

S2R72C05\*\*\*  
テクニカルマニュアル

本資料のご使用につきましては、次の点にご留意願います。  
本資料の内容については、予告無く変更することがあります。

---

1. 本資料の一部、または全部を弊社に無断で転載、または、複製など他の目的に使用することは堅くお断りいたします。
2. 本資料に掲載される応用回路、プログラム、使用方法等はあくまでも参考情報であり、これら起因する第三者の権利（工業所有権を含む）侵害あるいは損害の発生に対し、弊社はいかなる保証を行うものではありません。また、本資料によって第三者または弊社の工業所有権の実施権の許諾を行うものではありません。
3. 特性値の数値の大小は、数直線上の大小関係で表しています。
4. 本資料に掲載されている製品のうち「外国為替及び外国貿易法」に定める戦略物資に該当するものについては、輸出する場合、同法に基づく輸出許可が必要です。
5. 本資料に掲載されている製品は、生命維持装置その他、きわめて高い信頼性が要求される用途を前提としていません。よって、弊社は本（当該）製品をこれらの用途に用いた場合のいかなる責任についても負いかねます。

## 適用範囲

本ドキュメントは、USB2.0 デバイス・ホストコントローラ LSI「S2R72C05」に適用されます。



## 目次

1. 機能説明 .....	1
1.1. デバイス／ホスト・レジスタマップ選択及び機能選択 .....	1
1.1.1. レジスタマップ選択方法 .....	1
1.1.2. デバイス／ホスト機能選択方法 .....	1
1.1.3. ポート状態変化検出ステータス .....	2
1.1.3.1. VBUS_B端子変化ステータスの使用方法 .....	2
1.1.3.2. 信号ライン変化ステータスの使用方法 .....	3
1.2. USBデバイス制御 .....	5
1.2.1. エンドポイント .....	5
1.2.2. トランザクション .....	7
1.2.2.1. SETUPトランザクション .....	9
1.2.2.2. バルク／インタラプトOUTトランザクション .....	10
1.2.2.3. アイソクロナスOUTトランザクション .....	10
1.2.2.4. バルク／インタラプトINトランザクション .....	11
1.2.2.5. アイソクロナスINトランザクション .....	12
1.2.2.6. PINGトランザクション .....	12
1.2.3. コントロール転送 .....	13
1.2.3.1. セットアップステージ .....	14
1.2.3.2. データステージ／ステータスステージ .....	14
1.2.3.3. 自動アドレス設定機能 .....	14
1.2.3.4. デスクリプタ返信機能 .....	15
1.2.4. バルク転送／インタラプト転送／アイソクロナス転送 .....	15
1.2.5. データフロー .....	15
1.2.5.1. OUT転送 .....	15
1.2.5.2. IN転送 .....	16
1.2.6. バルクオンリーサポート .....	16
1.2.6.1. CBWサポート .....	16
1.2.6.2. CSWサポート .....	17
1.2.7. オート・ネゴシエーション機能 .....	18
1.2.7.1. DISABLE .....	18
1.2.7.2. IDLE .....	19
1.2.7.3. WAIT_TIM3US .....	19
1.2.7.4. WAIT_CHIRP .....	19
1.2.7.5. WAIT_RSTEND .....	19
1.2.7.6. DET_SUSPEND .....	19
1.2.7.7. IN_SUSPEND .....	19
1.2.7.8. CHK_EVENT .....	20
1.2.7.9. WAIT_RESTORE .....	20
1.2.7.10. ERR .....	20
1.2.7.11. 各ネゴシエーション機能の単体説明 .....	21
1.2.7.11.1. サスペンド検出(HSモード) .....	21
1.2.7.11.2. サスペンド検出(FSモード) .....	23
1.2.7.11.3. リセット検出(HSモード) .....	24
1.2.7.11.4. リセット検出(FSモード) .....	25
1.2.7.11.5. HS Detection Handshake .....	26
1.2.7.11.5.1. FSのホストダウンストリームポートに繋がれた場合 .....	27
1.2.7.11.5.2. HSのホストダウンストリームポートに繋がれた場合 .....	29
1.2.7.11.5.3. スヌーズ中にリセットされた場合 .....	32
1.2.7.11.6. レジュームの発行 .....	34

1.2.7.11.7.	レジュームの検出.....	37
1.2.7.11.8.	ケーブル挿入.....	39
1.3.	USBホスト制御.....	41
1.3.1.	チャンネル.....	41
1.3.1.1.	チャンネル概要.....	41
1.3.1.2.	コントロール専用チャンネル.....	43
1.3.1.3.	汎用チャンネル.....	44
1.3.1.4.	チャンネル使用例.....	46
1.3.1.4.1.	ストレージデバイスを1台接続した場合.....	46
1.3.1.4.2.	コミュニケーションデバイスを1台接続した場合.....	47
1.3.1.4.3.	ヒューマンインタフェイスデバイスを1台接続した場合.....	48
1.3.1.4.4.	ハブを介してストレージデバイスを2台接続した場合.....	49
1.3.2.	スケジューリング.....	50
1.3.3.	トランザクション.....	51
1.3.3.1.	SETUPトランザクション.....	52
1.3.3.2.	バルクOUTトランザクション.....	53
1.3.3.3.	インタラプトOUTトランザクション.....	54
1.3.3.4.	アイソクロナスOUTトランザクション.....	55
1.3.3.5.	バルクINトランザクション.....	56
1.3.3.6.	インタラプトINトランザクション.....	57
1.3.3.7.	アイソクロナスINトランザクション.....	59
1.3.3.8.	PINGトランザクション.....	60
1.3.3.9.	low-speed(LS)トランザクション.....	61
1.3.3.10.	スプリットトランザクション.....	63
1.3.4.	コントロール転送.....	64
1.3.4.1.	セットアップステージ.....	65
1.3.4.2.	データステージ/ステータスステージ.....	65
1.3.4.3.	コントロール転送サポート機能.....	65
1.3.5.	バルク転送/インタラプト転送/アイソクロナス転送.....	69
1.3.6.	データフロー.....	69
1.3.6.1.	OUT転送.....	69
1.3.6.2.	IN転送.....	69
1.3.7.	ゼロ長パケット自動発行機能.....	70
1.3.7.1.	バルク/インタラプトOUT転送のゼロ長パケット自動発行機能.....	70
1.3.8.	バルクオンリーサポート機能.....	71
1.3.9.	オーディオクラス・アシスト機能.....	75
1.3.10.	ホストステート管理サポート機能.....	76
1.3.10.1.	ホストステート.....	76
1.3.10.1.1.	IDLE.....	78
1.3.10.1.2.	WAIT_CONNECT.....	78
1.3.10.1.3.	DISABLED.....	79
1.3.10.1.4.	RESET.....	79
1.3.10.1.5.	OPERATIONAL.....	80
1.3.10.1.6.	SUSPEND.....	80
1.3.10.1.7.	RESUME.....	80
1.3.10.2.	検出機能.....	82
1.3.10.2.1.	VBUS異常検出.....	82
1.3.10.2.2.	切断検出.....	83
1.3.10.2.2.1.	HSデバイスが切断された場合.....	83
1.3.10.2.2.2.	FSまたはLSデバイスが切断された場合.....	84
1.3.10.2.3.	リモート・ウェークアップ検出.....	85

1.3.10.2.3.1.	HSデバイスが接続されている場合	85
1.3.10.2.3.2.	FSデバイスが接続されている場合	87
1.3.10.2.3.3.	LSデバイスが接続されている場合	88
1.3.10.2.4.	デバイスチャープ検出機能	89
1.3.10.2.4.1.	正しいデバイスのChirpを検出した場合	89
1.3.10.2.4.2.	異常なデバイスのChirpを検出した場合	90
1.3.10.2.5.	ポートエラー検出	91
1.3.10.3.	ホストステート管理サポート機能の単体説明	92
1.3.10.3.1.	GoIDLE	92
1.3.10.3.2.	GoWAIT_CONNECT	94
1.3.10.3.2.1.	FSデバイスが接続された場合	94
1.3.10.3.2.2.	LSデバイスが接続された場合	96
1.3.10.3.3.	GoDISABLED	98
1.3.10.3.3.1.	HSデバイスが接続されている場合	98
1.3.10.3.3.2.	FSデバイスが接続されている場合	99
1.3.10.3.3.3.	LSデバイスが接続されている場合	101
1.3.10.3.4.	GoRESET	103
1.3.10.3.4.1.	HSデバイスに対するリセット	103
1.3.10.3.4.2.	異常なデバイスのChirp検出	106
1.3.10.3.4.2.1.	チャープ完了ディセーブル(H_NegoControl_1.DisChirpFinish)設定が 0 の場合	106
1.3.10.3.4.2.2.	チャープ完了ディセーブル(H_NegoControl_1.DisChirpFinish)設定が 1 の場合	108
1.3.10.3.4.3.	FSデバイスに対するリセット	111
1.3.10.3.4.4.	LSデバイスに対するリセット	112
1.3.10.3.5.	GoOPERATIONAL	113
1.3.10.3.6.	GoSUSPEND	114
1.3.10.3.6.1.	HSデバイスが接続されている場合	114
1.3.10.3.6.2.	FSデバイスが接続されている場合	116
1.3.10.3.6.3.	LSデバイスが接続されている場合	118
1.3.10.3.7.	GoRESUME	119
1.3.10.3.7.1.	HSデバイスが接続されている場合	119
1.3.10.3.7.2.	FSデバイスが接続されている場合	121
1.3.10.3.7.3.	LSデバイスが接続されている場合	122
1.3.10.3.8.	GoWAIT_CONNECTtoDIS	124
1.3.10.3.9.	GoWAIT_CONNECTtoOP	125
1.3.10.3.9.1.	HSデバイスが接続されている場合	125
1.3.10.3.9.2.	FSまたはLSデバイスが接続されている場合	127
1.3.10.3.10.	GoRESETtoOP	128
1.3.10.3.10.1.	HSデバイスが接続されている場合	128
1.3.10.3.10.2.	FSまたはLSデバイスが接続されている場合	129
1.3.10.3.11.	GoSUSPENDtoOP	130
1.3.10.3.12.	GoRESUMEtOOP	131
1.4.	メディアデータ転送機能	132
1.4.1.	メディアデータ	132
1.4.2.	メディアデータ転送	132
1.4.3.	消費電力の削減	133
1.5.	パワーマネジメント機能	135
1.5.1.	2ポートモード時 (ClkSelect.Port1x2 が 0 にクリアされている場合)	135
1.5.1.1.	SLEEP(スリープ)	137
1.5.1.2.	SNOOZE(スヌーズ)	137

1.5.1.3.	ACTIVE60(アクティブ 60) .....	138
1.5.1.4.	ACT_DEVICE(アクトデバイス).....	138
1.5.1.5.	ACT_HOST(アクトホスト).....	138
1.5.1.6.	ACT_ALL(アクトオール) .....	139
1.5.2.	1 ポートモード時 (ClkSelect.Port1x2 が 1 にセットされている場合) .....	139
1.5.2.1.	SLEEP(スリープ) .....	140
1.5.2.2.	SNOOZE(スヌーズ) .....	141
1.5.2.3.	ACTIVE60(アクティブ 60) .....	141
1.5.2.4.	ACT_DEVICE(アクトデバイス).....	141
1.5.2.5.	ACT_HOST(アクトホスト).....	141
1.5.2.6.	ACT_ALL(アクトオール) .....	141
1.6.	FIFO管理.....	142
1.6.1.	FIFOメモリマップ .....	142
1.6.2.	デスクリプタエリア .....	143
1.6.2.1.	デスクリプタエリアへのデータの書き込み .....	143
1.6.2.2.	デスクリプタエリアでのデータステージ(IN)の実行 .....	143
1.6.3.	CBWエリア.....	143
1.6.3.1.	CBWエリア(USBデバイス時) .....	143
1.6.3.2.	CBWエリア(USBホスト時) .....	144
1.6.4.	CSWエリア .....	144
1.6.4.1.	CSWエリア(USBデバイス時).....	144
1.6.4.2.	CSWエリア(USBホスト時).....	144
1.6.5.	FIFOへのアクセス方法 .....	144
1.6.5.1.	RAMアクセスの方法(RAM_Rd) .....	145
1.6.5.2.	RAMアクセスの方法(RAM_WrDoor).....	145
1.6.5.3.	FIFOアクセスの方法(レジスタアクセス).....	145
1.6.5.4.	FIFOアクセスの方法(DMA) .....	145
1.6.5.5.	FIFOアクセスの方法(IDE).....	146
1.6.5.6.	FIFOアクセスの制限 .....	146
1.7.	メディアFIFO管理.....	147
1.7.1.	メディアFIFO .....	147
1.7.2.	メディアFIFOへのアクセス方法 .....	147
1.7.2.1.	メディアFIFOへのアクセス方法(レジスタアクセス).....	147
1.7.2.2.	メディアFIFOへのアクセス方法(DMA) .....	147
1.7.2.3.	メディアFIFOへのアクセス方法(IDE).....	147
1.7.2.4.	メディアFIFOへのアクセス制限 .....	148
1.8.	CPUIF .....	149
1.8.1.	CPUIFモード .....	149
1.8.2.	CPUIFのモード設定 .....	150
1.8.3.	ブロック構成.....	151
1.8.3.1.	REG(S2R72C05 Registers).....	151
1.8.3.1.1.	同期レジスタアクセス(ライト) .....	151
1.8.3.1.2.	同期レジスタアクセス(リード) .....	151
1.8.3.1.3.	FIFOアクセス(ライト) .....	151
1.8.3.1.4.	FIFOアクセス(リード) .....	152
1.8.3.1.5.	FIFOアクセスの端数処理 .....	153
1.8.3.1.6.	RAM_Rdアクセス .....	155
1.8.3.1.7.	非同期レジスタアクセス(ライト) .....	155
1.8.3.1.8.	非同期レジスタアクセス(リード) .....	155
1.8.3.2.	DMA0/DMA1(DMA ch.0 / ch.1).....	155



1.8.3.2.1.	基本機能.....	155
1.8.3.2.2.	端子設定.....	157
1.8.3.2.3.	カウントモード(ライト).....	157
1.8.3.2.4.	カウントモード(リード).....	159
1.8.3.2.5.	フリーランモード(ライト).....	161
1.8.3.2.6.	フリーランモード(リード).....	161
1.8.3.2.7.	REQアサートカウントオプション(ライト).....	161
1.8.3.2.8.	REQアサートカウントオプション(リード).....	162
1.8.3.2.9.	DMAのFIFOアクセス端数処理.....	162
1. 9.	Timer.....	163
1.9.1.	動作モード.....	163
1.9.2.	動作開始・停止.....	163
1. 10.	IDE I/F.....	164
1.10.1.	IDEタスクファイルレジスタへのアクセス.....	164
1.10.1.1.	IDEタスクファイルレジスタからのリード.....	164
1.10.1.2.	IDEタスクファイルレジスタへのライト.....	164
1.10.1.3.	IDEタスクファイルレジスタへのシーケンシャルライト.....	164
1.10.1.4.	IDEタスクファイルレジスタからのオートステータスレジスタリード.....	164
1.10.2.	PIOアクセス.....	166
1.10.2.1.	PIOリードDMA.....	166
1.10.2.2.	PIOライトDMA.....	166
1.10.3.	Multi-Word DMA.....	166
1.10.3.1.	Multi-Word DMA リード.....	166
1.10.3.2.	Multi-Word DMA ライト.....	167
1.10.4.	Ultra DMA.....	168
1.10.4.1.	Ultra DMA リード.....	168
1.10.4.2.	Ultra DMA ライト.....	168
1.10.5.	IDE転送モードの設定について.....	169
1. 11.	バウンダリスキャン(JTAG).....	170
1.11.1.	対応インストラクション.....	170
1.11.2.	DEVICE_CODEに関して.....	170
1.11.3.	バウンダリスキャン除外端子.....	170
2.	レジスタ.....	171
2. 1.	初期レジスタマップ.....	171
2. 2.	デバイス／ホスト共通レジスタマップ.....	172
2. 3.	デバイス・レジスタマップ.....	180
2. 4.	ホスト・レジスタマップ.....	186
2. 5.	初期レジスタ詳細説明.....	192
2.5.1.	xxxh CPUIF_MODE (CPU I/F Mode).....	192
2. 6.	デバイス／ホストレジスタ詳細説明.....	193
2.6.1.	000h MainIntStat (Main Interrupt Status).....	193
2.6.2.	001h DeviceIntStat (Device Interrupt Status).....	195
2.6.3.	002h HostIntStat (Host Interrupt Status).....	197
2.6.4.	003h CPU_IntStat (CPU Interrupt Status).....	199
2.6.5.	004h IDE_IntStat (IDE Interrupt Status).....	200
2.6.6.	005h MediaFIFO_IntStat (Media FIFO Interrupt Status).....	201
2.6.7.	010h MainIntEnb (Main Interrupt Enable).....	202
2.6.8.	011h DeviceIntEnb (Device Interrupt Enable).....	203
2.6.9.	012h HostIntEnb (Host Interrupt Enable).....	204
2.6.10.	013h CPU_IntEnb (CPU Interrupt Enable).....	205
2.6.11.	014h IDE_IntEnb (IDE Interrupt Enable).....	206

2.6.12.	015h MediaFIFO_IntEnb (Media FIFO Interrupt Enable).....	207
2.6.13.	020h <i>RevisionNum</i> ( <i>Revision Number</i> ) .....	208
2.6.14.	021h <i>ChipReset</i> ( <i>Chip Reset</i> ) .....	209
2.6.15.	022h PM_Control_0 (Power Management Control 0) .....	210
2.6.16.	023h PM_Control_1 ( <i>Power Management Control 1</i> ).....	213
2.6.17.	024h WakeupTim_H ( <i>Wakeup Time High</i> ).....	214
2.6.18.	025h WakeupTim_L ( <i>Wakeup Time Low</i> ) .....	214
2.6.19.	026h H_USB_Control ( <i>Host USB Control</i> ).....	215
2.6.20.	027h H_XcvrControl ( <i>Host Xcvr Control</i> ) .....	216
2.6.21.	028h D_USB_Status ( <i>Device USB Status</i> ) .....	217
2.6.22.	029h H_USB_Status ( <i>Host USB Status</i> ).....	218
2.6.23.	030h FIFO_Rd_0 (FIFO Read 0) .....	219
2.6.24.	031h FIFO_Rd_1 (FIFO Read 1) .....	219
2.6.25.	032h FIFO_Wr_0(FIFO Write 0) .....	220
2.6.26.	033h FIFO_Wr_1(FIFO Write 1) .....	220
2.6.27.	034h FIFO_RdRemain_H (FIFO Read Remain High) .....	221
2.6.28.	035h FIFO_RdRemain_L (FIFO Read Remain Low).....	221
2.6.29.	036h FIFO_WrRemain_H (FIFO Write Remain High) .....	222
2.6.30.	037h FIFO_WrRemain_L (FIFO Write Remain Low).....	222
2.6.31.	038h FIFO_ByteRd(FIFO Byte Read).....	223
2.6.32.	03Ah FIFO_ByteWr(FIFO Byte Write).....	224
2.6.33.	040h RAM_RdAdrs_H (RAM Read Address High).....	225
2.6.34.	041h RAM_RdAdrs_L (RAM Read Address Low).....	225
2.6.35.	042h RAM_RdControl (RAM Read Control).....	226
2.6.36.	043h RAM_RdCount (RAM Read Counter) .....	227
2.6.37.	044h RAM_WrAdrs_H (RAM Write Address High).....	228
2.6.38.	045h RAM_WrAdrs_L (RAM Write Address Low) .....	228
2.6.39.	046h RAM_WrDoor_0 (RAM Write Door 0).....	229
2.6.40.	047h RAM_WrDoor_1 (RAM Write Door 1).....	229
2.6.41.	048h MediaFIFO_Control (Media FIFO Control) .....	230
2.6.42.	049h ClrAllMediaFIFO_Join (Clear All Media FIFO Join).....	231
2.6.43.	04Ah MediaFIFO_Join (Media FIFO Join) .....	232
2.6.44.	050h RAM_Rd_00 (RAM Read 00) .....	233
2.6.45.	051h RAM_Rd_01 (RAM Read 01) .....	233
2.6.46.	052h RAM_Rd_02 (RAM Read 02) .....	233
2.6.47.	053h RAM_Rd_03 (RAM Read 03) .....	233
2.6.48.	054h RAM_Rd_04 (RAM Read 04) .....	233
2.6.49.	055h RAM_Rd_05 (RAM Read 05) .....	233
2.6.50.	056h RAM_Rd_06 (RAM Read 06) .....	233
2.6.51.	057h RAM_Rd_07 (RAM Read 07) .....	233
2.6.52.	058h RAM_Rd_08 (RAM Read 08) .....	233
2.6.53.	059h RAM_Rd_09 (RAM Read 09) .....	233
2.6.54.	05Ah RAM_Rd_0A (RAM Read 0A).....	233
2.6.55.	05Bh RAM_Rd_0B (RAM Read 0B) .....	233
2.6.56.	05Ch RAM_Rd_0C (RAM Read 0C) .....	233
2.6.57.	05Dh RAM_Rd_0D (RAM Read 0D) .....	233
2.6.58.	05Eh RAM_Rd_0E (RAM Read 0E).....	233
2.6.59.	05Fh RAM_Rd_0F (RAM Read 0F) .....	233
2.6.60.	060h RAM_Rd_10 (RAM Read 10) .....	233
2.6.61.	061h RAM_Rd_11 (RAM Read 11) .....	233
2.6.62.	062h RAM_Rd_12 (RAM Read 12) .....	233
2.6.63.	063h RAM_Rd_13 (RAM Read 13) .....	233
2.6.64.	064h RAM_Rd_14 (RAM Read 14) .....	233
2.6.65.	065h RAM_Rd_15 (RAM Read 15) .....	233
2.6.66.	066h RAM_Rd_16 (RAM Read 16) .....	233
2.6.67.	067h RAM_Rd_17 (RAM Read 17) .....	233

2.6.68.	068h RAM_Rd_18 (RAM Read 18) .....	233
2.6.69.	069h RAM_Rd_19 (RAM Read 19) .....	233
2.6.70.	06Ah RAM_Rd_1A (RAM Read 1A) .....	233
2.6.71.	06Bh RAM_Rd_1B (RAM Read 1B) .....	233
2.6.72.	06Ch RAM_Rd_1C (RAM Read 1C) .....	233
2.6.73.	06Dh RAM_Rd_1D (RAM Read 1D) .....	233
2.6.74.	06Eh RAM_Rd_1E (RAM Read 1E) .....	233
2.6.75.	06Fh RAM_Rd_1F (RAM Read 1F) .....	233
2.6.76.	071h DMA0_Config (DMA0 Config) .....	234
2.6.77.	072h DMA0_Control (DMA0 Control) .....	236
2.6.78.	074h DMA0_Remain_H (DMA0 FIFO Remain High) .....	237
2.6.79.	075h DMA0_Remain_L (DMA0 FIFO Remain Low) .....	237
2.6.80.	078h DMA0_Count_HH (DMA0 Transfer Byte Counter High/High) .....	238
2.6.81.	079h DMA0_Count_HL (DMA0 Transfer Byte Counter High/Low) .....	238
2.6.82.	07Ah DMA0_Count_LH (DMA0 Transfer Byte Counter Low/High) .....	238
2.6.83.	07Bh DMA0_Count_LL (DMA0 Transfer Byte Counter Low/Low) .....	238
2.6.84.	07Ch DMA0_RdData_0 (DMA0 Read Data 0) .....	240
2.6.85.	07Dh DMA0_RdData_1 (DMA0 Read Data 1) .....	240
2.6.86.	07Eh DMA0_WrData_0 (DMA0 Write Data 0) .....	241
2.6.87.	07Fh DMA0_WrData_1 (DMA0 Write Data 1) .....	241
2.6.88.	081h DMA1_Config (DMA1 Config) .....	242
2.6.89.	082h DMA1_Control (DMA1 Control) .....	244
2.6.90.	084h DMA1_Remain_H (DMA1 FIFO Remain High) .....	245
2.6.91.	085h DMA1_Remain_L (DMA1 FIFO Remain Low) .....	245
2.6.92.	088h DMA1_Count_HH (DMA1 Transfer Byte Counter High/High) .....	246
2.6.93.	089h DMA1_Count_HL (DMA1 Transfer Byte Counter High/Low) .....	246
2.6.94.	08Ah DMA1_Count_LH (DMA1 Transfer Byte Counter Low/High) .....	246
2.6.95.	08Bh DMA1_Count_LL (DMA1 Transfer Byte Counter Low/Low) .....	246
2.6.96.	08Ch DMA1_RdData_0 (DMA1 Read Data 0) .....	248
2.6.97.	08Dh DMA1_RdData_1 (DMA1 Read Data 1) .....	248
2.6.98.	08Eh DMA1_WrData_0 (DMA1 Write Data 0) .....	249
2.6.99.	08Fh DMA1_WrData_1 (DMA1 Write Data 1) .....	249
2.6.100.	090h IDE_Status (IDE Status) .....	250
2.6.101.	091h IDE_Control (IDE Control) .....	251
2.6.102.	092h IDE_Config_0 (IDE Configuration 0) .....	252
2.6.103.	093h IDE_Config_1 (IDE Configuration 1) .....	253
2.6.104.	094h IDE_Rmod (IDE Register Mode) .....	254
2.6.105.	095h IDE_Tmod (IDE Transfer Mode) .....	255
2.6.106.	096h IDE_Umod (IDE Ultra-DMA Transfer Mode) .....	256
2.6.107.	09Ah IDE_CRC_H (IDE CRC High) .....	257
2.6.108.	09Bh IDE_CRC_L (IDE CRC Low) .....	257
2.6.109.	09Dh IDE_Count_H (IDE Transfer Byte Counter High) .....	258
2.6.110.	09Eh IDE_Count_M (IDE Transfer Byte Counter Middle) .....	258
2.6.111.	09Fh IDE_Count_L (IDE Transfer Byte Counter Low) .....	258
2.6.112.	0A0h IDE_RegAdrs (IDE Register Address) .....	259
2.6.113.	0A2h IDE_RdRegValue_0 (IDE Register Read Value 0) .....	260
2.6.114.	0A3h IDE_RdRegValue_1 (IDE Register Read Value 1) .....	260
2.6.115.	0A4h IDE_WrRegValue_0 (IDE Register Write Value 0) .....	261
2.6.116.	0A5h IDE_WrRegValue_1 (IDE Register Write Value 1) .....	261
2.6.117.	0A6h IDE_SeqWrRegControl (IDE Sequential Register Write Control) .....	262
2.6.118.	0A7h IDE_SeqWrRegCnt (IDE Sequential Register Write Counter) .....	263
2.6.119.	0A8h IDE_SeqWrRegAdrs (IDE Sequential Register Write Address FIFO) .....	264
2.6.120.	0A9h IDE_SeqWrRegValue (IDE Sequential Register Write Value FIFO) .....	265
2.6.121.	0ACh IDE_RegConfig (IDE Register Configuration) .....	266
2.6.122.	0B1h HostDeviceSel (Host Device Select) .....	267
2.6.123.	0B3h ModeProtect (Mode Protection) .....	268

2.6.124.	0B5h <i>ClkSelect</i> ( <i>Clock Select</i> ) .....	269
2.6.125.	0B7h <i>ChipConfig</i> ( <i>Chip Configuration</i> ) .....	270
2.6.126.	0BDh <i>TimerConfig</i> ( <i>Timer Configuration</i> ) .....	272
2.6.127.	0BEh <i>TimerSet_H</i> ( <i>Timer Set High</i> ) .....	273
2.6.128.	0BFh <i>TimerSet_L</i> ( <i>Timer Set Low</i> ) .....	273
2.6.129.	0C0h <i>AREAnIntStat</i> ( <i>AREAn Interrupt Status</i> ) .....	274
2.6.130.	0C1h <i>AREA0IntStat</i> ( <i>AREA0 Interrupt Status</i> ) .....	275
2.6.131.	0C2h <i>AREA1IntStat</i> ( <i>AREA1 Interrupt Status</i> ) .....	275
2.6.132.	0C3h <i>AREA2IntStat</i> ( <i>AREA2 Interrupt Status</i> ) .....	275
2.6.133.	0C4h <i>AREA3IntStat</i> ( <i>AREA3 Interrupt Status</i> ) .....	275
2.6.134.	0C5h <i>AREA4IntStat</i> ( <i>AREA4 Interrupt Status</i> ) .....	275
2.6.135.	0C6h <i>AREA5IntStat</i> ( <i>AREA5 Interrupt Status</i> ) .....	275
2.6.136.	0C8h <i>AREAnIntEnb</i> ( <i>AREAn Interrupt Enable</i> ) .....	276
2.6.137.	0C9h <i>AREA0IntEnb</i> ( <i>AREA0 Interrupt Enable</i> ) .....	277
2.6.138.	0CAh <i>AREA1IntEnb</i> ( <i>AREA1 Interrupt Enable</i> ) .....	277
2.6.139.	0CBh <i>AREA2IntEnb</i> ( <i>AREA2 Interrupt Enable</i> ) .....	277
2.6.140.	0CCh <i>AREA3IntEnb</i> ( <i>AREA3 Interrupt Enable</i> ) .....	277
2.6.141.	0CDh <i>AREA4IntEnb</i> ( <i>AREA4 Interrupt Enable</i> ) .....	277
2.6.142.	0CEh <i>AREA5IntEnb</i> ( <i>AREA5 Interrupt Enable</i> ) .....	277
2.6.143.	0D0h <i>AREA0Join_0</i> ( <i>AREA 0 Join 0</i> ) .....	278
2.6.144.	0D2h <i>AREA1Join_0</i> ( <i>AREA 1 Join 0</i> ) .....	278
2.6.145.	0D4h <i>AREA2Join_0</i> ( <i>AREA 2 Join 0</i> ) .....	278
2.6.146.	0D6h <i>AREA3Join_0</i> ( <i>AREA3 Join 0</i> ) .....	278
2.6.147.	0D8h <i>AREA4Join_0</i> ( <i>AREA 4 Join 0</i> ) .....	278
2.6.148.	0DAh <i>AREA5Join_0</i> ( <i>AREA 5 Join 0</i> ) .....	278
2.6.149.	0D1h <i>AREA0Join_1</i> ( <i>AREA 0 Join 1</i> ) .....	279
2.6.150.	0D3h <i>AREA1Join_1</i> ( <i>AREA 1 Join 1</i> ) .....	279
2.6.151.	0D5h <i>AREA2Join_1</i> ( <i>AREA 2 Join 1</i> ) .....	279
2.6.152.	0D7h <i>AREA3Join_1</i> ( <i>AREA 3 Join 1</i> ) .....	279
2.6.153.	0D9h <i>AREA4Join_1</i> ( <i>AREA 4 Join 1</i> ) .....	279
2.6.154.	0DBh <i>AREA5Join_1</i> ( <i>AREA 5 Join 1</i> ) .....	279
2.6.155.	0DEh <i>ClrAREAnJoin_0</i> ( <i>Clear AREA n Join 0</i> ) .....	280
2.6.156.	0DFh <i>ClrAREAnJoin_1</i> ( <i>Clear AREA 1 Join 1</i> ) .....	281
2.6.157.	180h <i>AREA0StartAdrs_H</i> ( <i>AREA 0 Start Address High</i> ) .....	282
2.6.158.	181h <i>AREA0StartAdrs_L</i> ( <i>AREA 0 Start Address Low</i> ) .....	282
2.6.159.	184h <i>AREA1StartAdrs_H</i> ( <i>AREA 1 Start Address High</i> ) .....	282
2.6.160.	185h <i>AREA1StartAdrs_L</i> ( <i>AREA 1 Start Address Low</i> ) .....	282
2.6.161.	188h <i>AREA2StartAdrs_H</i> ( <i>AREA2 Start Address High</i> ) .....	282
2.6.162.	189h <i>AREA2StartAdrs_L</i> ( <i>AREA2 Start Address Low</i> ) .....	282
2.6.163.	18Ch <i>AREA3StartAdrs_H</i> ( <i>AREA3 Start Address High</i> ) .....	282
2.6.164.	18Dh <i>AREA3StartAdrs_L</i> ( <i>AREA3 Start Address Low</i> ) .....	282
2.6.165.	190h <i>AREA4StartAdrs_H</i> ( <i>AREA4 Start Address High</i> ) .....	282
2.6.166.	191h <i>AREA4StartAdrs_L</i> ( <i>AREA4 Start Address Low</i> ) .....	282
2.6.167.	194h <i>AREA5StartAdrs_H</i> ( <i>AREA5 Start Address High</i> ) .....	282
2.6.168.	195h <i>AREA5StartAdrs_L</i> ( <i>AREA5 Start Address Low</i> ) .....	282
2.6.169.	182h <i>AREA0EndAdrs_H</i> ( <i>AREA 0 End Address High</i> ) .....	283
2.6.170.	183h <i>AREA0EndAdrs_L</i> ( <i>AREA 0 End Address Low</i> ) .....	283
2.6.171.	186h <i>AREA1EndAdrs_H</i> ( <i>AREA1 End Address High</i> ) .....	283
2.6.172.	187h <i>AREA1EndAdrs_L</i> ( <i>AREA1 End Address Low</i> ) .....	283
2.6.173.	18Ah <i>AREA2EndAdrs_H</i> ( <i>AREA2 End Address High</i> ) .....	283
2.6.174.	18Bh <i>AREA2EndAdrs_L</i> ( <i>AREA2 End Address Low</i> ) .....	283
2.6.175.	18Eh <i>AREA3EndAdrs_H</i> ( <i>AREA3 End Address High</i> ) .....	283
2.6.176.	18Fh <i>AREA3EndAdrs_L</i> ( <i>AREA3 End Address Low</i> ) .....	283
2.6.177.	192h <i>AREA4EndAdrs_H</i> ( <i>AREA4 End Address High</i> ) .....	283
2.6.178.	193h <i>AREA4EndAdrs_L</i> ( <i>AREA4 End Address Low</i> ) .....	283
2.6.179.	196h <i>AREA5EndAdrs_H</i> ( <i>AREA5 End Address High</i> ) .....	283

2.6.180.	197h AREA5EndAdrs_L (AREA5 End Address Low) .....	283
2.6.181.	19Fh AREAnFIFO_Clr (AREAn FIFO Clear) .....	284
2. 7.	デバイス・レジスタ詳細説明 .....	285
2.7.1.	0E0h D_SIE_IntStat (Device SIE Interrupt Status) .....	285
2.7.2.	0E2h D_FIFO_IntStat (Device FIFO Interrupt Status) .....	288
2.7.3.	0E3h D_BulkIntStat (Device Bulk Interrupt Status) .....	289
2.7.4.	0E4h D_EPrIntStat (Device EPr Interrupt Status) .....	290
2.7.5.	0E5h D_EP0IntStat (Device EP0 Interrupt Status) .....	291
2.7.6.	0E6h D_EPaIntStat (Device EPa Interrupt Status) .....	291
2.7.7.	0E7h D_EPbIntStat (Device EPb Interrupt Status) .....	291
2.7.8.	0E8h D_EPcIntStat (Device EPc Interrupt Status) .....	291
2.7.9.	0E9h D_EPdIntStat (Device EPd Interrupt Status) .....	291
2.7.10.	0EAh D_EPeIntStat (Device EPe Interrupt Status) .....	291
2.7.11.	0F0h D_SIE_IntEnb (Device SIE Interrupt Enable) .....	292
2.7.12.	0F2h D_FIFO_IntEnb (Device FIFO Interrupt Enable) .....	293
2.7.13.	0F3h D_BulkIntEnb (Device Bulk Interrupt Enable) .....	294
2.7.14.	0F4h D_EPrIntEnb (Device EPr Interrupt Enable) .....	295
2.7.15.	0F5h D_EP0IntEnb (Device EP0 Interrupt Enable) .....	296
2.7.16.	0F6h D_EPaIntEnb (Device EPa Interrupt Enable) .....	296
2.7.17.	0F7h D_EPbIntEnb (Device EPb Interrupt Enable) .....	296
2.7.18.	0F8h D_EPcIntEnb (Device EPc Interrupt Enable) .....	296
2.7.19.	0F9h D_EPdIntEnb (Device EPd Interrupt Enable) .....	296
2.7.20.	0FAh D_EPeIntEnb (Device EPe Interrupt Enable) .....	296
2.7.21.	100h D_Reset (Device Reset) .....	297
2.7.22.	102h D_NegoControl (Device Negotiation Control) .....	298
2.7.23.	105h D_XcvrControl (Device Xcvr Control) .....	300
2.7.24.	106h D_USB_Test (Device USB_Test) .....	301
2.7.25.	108h D_EPnControl (Device Endpoint Control) .....	303
2.7.26.	10Ah D_BulkOnlyControl (Device BulkOnly Control) .....	304
2.7.27.	10Bh D_BulkOnlyConfig (Device BulkOnly Configuration) .....	305
2.7.28.	110h D_EP0SETUP_0 (Device EP0 SETUP 0) .....	307
2.7.29.	111h D_EP0SETUP_1 (Device EP0 SETUP 1) .....	307
2.7.30.	112h D_EP0SETUP_2 (Device EP0 SETUP 2) .....	307
2.7.31.	113h D_EP0SETUP_3 (Device EP0 SETUP 3) .....	307
2.7.32.	114h D_EP0SETUP_4 (Device EP0 SETUP 4) .....	307
2.7.33.	115h D_EP0SETUP_5 (Device EP0 SETUP 5) .....	307
2.7.34.	116h D_EP0SETUP_6 (Device EP0 SETUP 6) .....	307
2.7.35.	117h D_EP0SETUP_7 (Device EP0 SETUP 7) .....	307
2.7.36.	118h D_USB_Address (Device USB Address) .....	308
2.7.37.	11Ah D_SETUP_Control (Device SETUP Control) .....	309
2.7.38.	11Eh D_FrameNumber_H (Device FrameNumber High) .....	310
2.7.39.	11Fh D_FrameNumber_L (Device FrameNumber Low) .....	310
2.7.40.	120h D_EP0MaxSize (Device EP0 Max Packet Size) .....	311
2.7.41.	121h D_EP0Control (Device EP0 Control) .....	312
2.7.42.	122h D_EP0ControlIN (Device EP0 Control IN) .....	313
2.7.43.	123h D_EP0ControlOUT (Device EP0 Control OUT) .....	315
2.7.44.	130h D_EPaMaxSize_H (Device EPa Max Packet Size High) .....	316
2.7.45.	131h D_EPaMaxSize_L (Device EPa Max Packet Size Low) .....	316
2.7.46.	140h D_EPbMaxSize_H (Device EPb Max Packet Size High) .....	316
2.7.47.	141h D_EPbMaxSize_L (Device EPb Max Packet Size Low) .....	316
2.7.48.	150h D_EPcMaxSize_H (Device EPc Max Packet Size High) .....	316
2.7.49.	151h D_EPcMaxSize_L (Device EPc Max Packet Size Low) .....	316
2.7.50.	1A0h D_EPdMaxSize_H (Device EPd Max Packet Size High) .....	316
2.7.51.	1A1h D_EPdMaxSize_L (Device EPd Max Packet Size Low) .....	316
2.7.52.	1B0h D_EPeMaxSize_H (Device EPe Max Packet Size High) .....	316
2.7.53.	1B1h D_EPeMaxSize_L (Device EPe Max Packet Size Low) .....	316

2.7.54.	132h D_EPdConfig_0 (Device EPd Configuration 0).....	317
2.7.55.	142h D_EPbConfig_0 (Device EPb Configuration 0).....	317
2.7.56.	152h D_EPcConfig_0 (Device EPc Configuration 0).....	317
2.7.57.	1A2h D_EPdConfig_0 (Device EPd Configuration 0).....	317
2.7.58.	1B2h D_EPeConfig_0 (Device EPe Configuration 0).....	317
2.7.59.	134h D_EPdControl (Device EPd Control).....	318
2.7.60.	144h D_EPbControl (Device EPb Control).....	318
2.7.61.	154h D_EPcControl (Device EPc Control).....	318
2.7.62.	1A4h D_EPdControl (Device EPd Control).....	318
2.7.63.	1B4h D_EPeControl (Device EPe Control).....	318
2.7.64.	160h D_DescAdrs_H (Device Descriptor Address High).....	320
2.7.65.	161h D_DescAdrs_L (Device Descriptor Address Low).....	320
2.7.66.	162h D_DescSize_H (Device Descriptor Size High).....	321
2.7.67.	163h D_DescSize_L (Device Descriptor Size Low).....	321
2.7.68.	170h D_DMA0_FIFO_Control (Device DMA0 FIFO Control).....	322
2.7.69.	172h D_DMA1_FIFO_Control (Device DMA1 FIFO Control).....	323
2.7.70.	1BEh DTM_Config (Device Transceiver Macro Config).....	324
2.7.71.	1E1h D_ModeControl (Address Mode Control).....	325
2.8.	ホスト・レジスタ詳細説明.....	326
2.8.1.	0E0h H_SIE_IntStat_0 (Host SIE Interrupt Status 0).....	326
2.8.2.	0E1h H_SIE_IntStat_1 (SIE Host Interrupt Status 1).....	328
2.8.3.	0E2h H_FIFO_IntStat (Host FIFO Interrupt Status).....	329
2.8.4.	0E3h H_FrameIntStat (Host Frame Interrupt Status).....	330
2.8.5.	0E4h H_CHrIntStat (Host CHr Interrupt Status).....	331
2.8.6.	0E5h H_CH0IntStat (Host CH0 Interrupt Status).....	332
2.8.7.	0E6h H_CHaIntStat (Host CHa Interrupt Status).....	334
2.8.8.	0E7h H_CHbIntStat (Host CHb Interrupt Status).....	336
2.8.9.	0E8h H_CHcIntStat (Host CHc Interrupt Status).....	336
2.8.10.	0E9h H_CHdIntStat (Host CHd Interrupt Status).....	336
2.8.11.	0EAh H_CHeIntStat (Host CHe Interrupt Status).....	336
2.8.12.	0F0h H_SIE_IntEnb_0 (Host SIE Interrupt Enable).....	337
2.8.13.	0F1h H_SIE_IntEnb_1 (SIE Host Interrupt Enable 1).....	338
2.8.14.	0F2h H_FIFO_IntEnb (Host FIFO Interrupt Enable).....	339
2.8.15.	0F3h H_FrameIntEnb (Host Frame Interrupt Enable).....	340
2.8.16.	0F4h H_CHrIntEnb (Host CHr Interrupt Enable).....	341
2.8.17.	0F5h H_CH0IntEnb (Host CH0 Interrupt Enable).....	342
2.8.18.	0F6h H_CHaIntEnb (Host CHa Interrupt Enable).....	343
2.8.19.	0F7h H_CHbIntEnb (Host CHb Interrupt Enable).....	344
2.8.20.	0F8h H_CHcIntEnb (Host CHc Interrupt Enable).....	344
2.8.21.	0F9h H_CHdIntEnb (Host CHd Interrupt Enable).....	344
2.8.22.	0FAh H_CHeIntEnb (Host CHe Interrupt Enable).....	344
2.8.23.	100h H_Reset (Host Reset).....	345
2.8.24.	102h H_NegoControl_0 (Host NegoControl 0).....	346
2.8.25.	104h H_NegoControl_1 (Host NegoControl 1).....	348
2.8.26.	106h H_USB_Test (Host USB_Test).....	349
2.8.27.	110h H_CH0SETUP_0 (Host CH0 SETUP 0).....	351
2.8.28.	111h H_CH0SETUP_1 (Host CH0 SETUP 1).....	351
2.8.29.	112h H_CH0SETUP_2 (Host CH0 SETUP 2).....	351
2.8.30.	113h H_CH0SETUP_3 (Host CH0 SETUP 3).....	351
2.8.31.	114h H_CH0SETUP_4 (Host CH0 SETUP 4).....	351
2.8.32.	115h H_CH0SETUP_5 (Host CH0 SETUP 5).....	351
2.8.33.	116h H_CH0SETUP_6 (Host CH0 SETUP 6).....	351
2.8.34.	117h H_CH0SETUP_7 (Host CH0 SETUP 7).....	351
2.8.35.	11Eh H_FrameNumber_H (Host FrameNumber High).....	352
2.8.36.	11Fh H_FrameNumber_L (Host FrameNumber Low).....	352
2.8.37.	120h H_CH0Config_0 (Host Channel 0 Configuration0).....	353

2.8.38.	121h H_CH0Config_1 (Host Channel 0 Configuration1) .....	355
2.8.39.	123h H_CH0MaxPktSize (Host Channel 0 Max Packet Size).....	356
2.8.40.	126h H_CH0TotalSize_H (Host Channel 0 Total Size High) .....	357
2.8.41.	127h H_CH0TotalSize_L (Host Channel 0 Total Size Low) .....	357
2.8.42.	128h H_CH0HubAdrs (Host Channel 0 Hub Address).....	358
2.8.43.	129h H_CH0FuncAdrs (Host Channel 0 Function Address) .....	359
2.8.44.	12Bh H_CTL_SupportControl (Host ControlTransfer Support Control) .....	360
2.8.45.	12Eh H_CH0ConditionCode (Host Channel 0 Condition Code).....	361
2.8.46.	130h H_CHaConfig_0 (Host Channel a Configuration0) .....	362
2.8.47.	131h H_CHaConfig_1 (Host Channel a Configuration1) .....	364
2.8.48.	132h H_CHaMaxPktSize_H (Host Channel a Max Packet Size High) .....	365
2.8.49.	133h H_CHaMaxPktSize_L (Host Channel a Max Packet Size Low).....	365
2.8.50.	134h H_CHaTotalSize_HH (Host Channel a Total Size High-High) .....	366
2.8.51.	135h H_CHaTotalSize_HL (Host Channel a Total Size High-Low).....	366
2.8.52.	136h H_CHaTotalSize_LH (Host Channel a Total Size Low-High).....	366
2.8.53.	137h H_CHaTotalSize_LL (Host Channel a Total Size Low-Low).....	366
2.8.54.	138h H_CHaHubAdrs (Host Channel a Hub Address) .....	368
2.8.55.	139h H_CHaFuncAdrs (Host Channel a Function Address).....	369
2.8.56.	13Ah H_CHaBO_SupporotControl (Host Bulk Only Transfer Supporot Control).....	370
2.8.57.	13Bh H_CHaBO_CSW_RcvDataSize (Host CSW Receive Data Size) .....	371
2.8.58.	13Ch H_CHaBO_OUT_EP_Control (Host OUT Endpoint Control) .....	372
2.8.59.	13Dh H_CHaBO_IN_EP_Control (Host IN Endpoint Control) .....	373
2.8.60.	13Eh H_CHaConditionCode (Host Channel a Condition Code) .....	374
2.8.61.	140h H_CHbConfig_0 (Host Channel b Configuration0) .....	375
2.8.62.	150h H_CHcConfig_0 (Host Channel c Configuration0) .....	375
2.8.63.	160h H_CHdConfig_0 (Host Channel d Configuration0) .....	375
2.8.64.	170h H_CHeConfig_0 (Host Channel e Configuration0) .....	375
2.8.65.	141h H_CHbConfig_1 (Host Channel b Configuration1) .....	377
2.8.66.	151h H_CHcConfig_1 (Host Channel c Configuration1) .....	377
2.8.67.	161h H_CHdConfig_1 (Host Channel d Configuration1) .....	377
2.8.68.	171h H_CHeConfig_1 (Host Channel e Configuration1) .....	377
2.8.69.	142h H_CHbMaxPktSize_H (Host Channel b Max Packet Size High).....	379
2.8.70.	143h H_CHbMaxPktSize_L (Host Channel b Max Packet Size Low) .....	379
2.8.71.	152h H_CHcMaxPktSize_H (Host Channel c Max Packet Size High) .....	379
2.8.72.	153h H_CHcMaxPktSize_L (Host Channel c Max Packet Size Low).....	379
2.8.73.	162h H_CHdMaxPktSize_H (Host Channel d Max Packet Size High).....	379
2.8.74.	163h H_CHdMaxPktSize_L (Host Channel d Max Packet Size Low) .....	379
2.8.75.	172h H_CHeMaxPktSize_H (Host Channel e Max Packet Size High) .....	379
2.8.76.	173h H_CHeMaxPktSize_L (Host Channel e Max Packet Size Low).....	379
2.8.77.	144h H_CHbTotalSize_HH (Host Channel b Total Size High-High) .....	380
2.8.78.	145h H_CHbTotalSize_HL (Host Channel b Total Size High-Low) .....	380
2.8.79.	146h H_CHbTotalSize_LH (Host Channel b Total Size Low-High) .....	380
2.8.80.	147h H_CHbTotalSize_LL (Host Channel b Total Size Low-Low) .....	380
2.8.81.	154h H_CHcTotalSize_HH (Host Channel c Total Size High-High) .....	380
2.8.82.	155h H_CHcTotalSize_HL (Host Channel c Total Size High-Low).....	380
2.8.83.	156h H_CHcTotalSize_LH (Host Channel c Total Size Low-High).....	380
2.8.84.	157h H_CHcTotalSize_LL (Host Channel c Total Size Low-Low).....	380
2.8.85.	164h H_CHdTotalSize_HH (Host Channel d Total Size High-High) .....	380
2.8.86.	165h H_CHdTotalSize_HL (Host Channel d Total Size High-Low) .....	380
2.8.87.	166h H_CHdTotalSize_LH (Host Channel d Total Size Low-High) .....	380
2.8.88.	167h H_CHdTotalSize_LL (Host Channel d Total Size Low-Low) .....	380
2.8.89.	174h H_CHeTotalSize_HH (Host Channel e Total Size High-High) .....	380
2.8.90.	175h H_CHeTotalSize_HL (Host Channel e Total Size High-Low).....	380
2.8.91.	176h H_CHeTotalSize_LH (Host Channel e Total Size Low-High).....	380
2.8.92.	177h H_CHeTotalSize_LL (Host Channel e Total Size Low-Low).....	380
2.8.93.	148h H_CHbHubAdrs (Host Channel b Hub Address).....	383

2.8.94.	158h H_CHcHubAdrs (Host Channel c Hub Address) .....	383
2.8.95.	168h H_CHdHubAdrs (Host Channel d Hub Address) .....	383
2.8.96.	178h H_CHeHubAdrs (Host Channel e Hub Address) .....	383
2.8.97.	149h H_CHbFuncAdrs (Host Channel b Function Address) .....	384
2.8.98.	159h H_CHcFuncAdrs (Host Channel c Function Address) .....	384
2.8.99.	169h H_CHdFuncAdrs (Host Channel d Function Address) .....	384
2.8.100.	179h H_CHeFuncAdrs (Host Channel e Function Address) .....	384
2.8.101.	14Ah H_CHbInterval_H (Host Channel b Interval High) .....	385
2.8.102.	14Bh H_CHbInterval_L (Host Channel b Interval Low) .....	385
2.8.103.	15Ah H_CHcInterval_H (Host Channel c Interval High) .....	385
2.8.104.	15Bh H_CHcInterval_L (Host Channel c Interval Low) .....	385
2.8.105.	16Ah H_CHdInterval_H (Host Channel d Interval High) .....	385
2.8.106.	16Bh H_CHdInterval_L (Host Channel d Interval Low) .....	385
2.8.107.	17Ah H_CHeInterval_H (Host Channel e Interval High) .....	385
2.8.108.	17Bh H_CHeInterval_L (Host Channel e Interval Low) .....	385
2.8.109.	14Ch H_CHbTranPause(Host Channel b Transaction Pause) .....	386
2.8.110.	15Ch H_CHcTranPause(Host Channel c Transaction Pause) .....	386
2.8.111.	16Ch H_CHdTranPause(Host Channel d Transaction Pause) .....	386
2.8.112.	17Ch H_CHeTranPause(Host Channel e Transaction Pause) .....	386
2.8.113.	14Eh H_CHbConditionCode (Host Channel b Condition Code) .....	387
2.8.114.	15Eh H_CHcConditionCode (Host Channel c Condition Code) .....	387
2.8.115.	16Eh H_CHdConditionCode (Host Channel d Condition Code) .....	387
2.8.116.	17Eh H_CHeConditionCode (Host Channel e Condition Code) .....	387
2.8.117.	1B0h H_TriggerFrameNum_H (Host Trigger Frame Number High) .....	388
2.8.118.	1B1h H_TriggerFrameNum_L (Host Trigger Frame Number Low) .....	388
2.8.119.	1BEh <i>HTM_Config</i> (Host Transceiver Macro Config) .....	389
2.8.120.	1F5h H_Protect (Host Protect) .....	390
Appendix A	IDE_Config_1.Swapビット設定 .....	391
Appendix B	リトルエンディアンのCPUへの接続 .....	392
Appendix C	1 ポートモード .....	394
Appendix D	SetAddressリクエストの応答について(TSのみ適用) .....	396
	改訂履歴(Rev.1.00) .....	398



## 1. 機能説明

本 LSI の動作について説明します。

以下では、レジスタについて下記のルールで標記します。

- ・ 1 アドレス分のレジスタを指す場合の名称  
レジスタ名+レジスタと上記します。  
例：「MainIntStat レジスタ」
- ・ 個々のビットを指す場合の名称  
レジスタ名. ビット名+ビット、或いは、ビット名+ビットと標記します。  
例：「MainIntStat.CPU\_IntStat ビット」
- ・ エンドポイント毎にあるレジスタ  
D\_EPx{x=...}～レジスタと標記します。  
例：「D\_EPx{x=0,a-e}IntStat レジスタ」
- ・ チャネル毎にあるレジスタ  
H\_CHx{x=...}～レジスタと標記します。  
例：「H\_CHx{x=0,a-e}IntStat レジスタ」
- ・ DMA チャネル毎にあるレジスタ  
DMAx{x=...}～レジスタと標記します。  
例：「DMAx{x=0,1}Config レジスタ」

### 1.1. デバイス／ホスト・レジスタマップ選択及び機能選択

本 LSI では、USB を使用する際に、デバイス・レジスタとホスト・レジスタについてはいずれか一方のレジスタマップ、並びに機能を選択することが必要です。

デバイス・レジスタマップを選択した場合(デバイスマップ選択状態)、デバイス／ホスト・共通レジスタおよびデバイス・レジスタのレジスタ・ビットがアクセス可能となります。ホスト・レジスタマップを選択した場合(ホストマップ選択状態)、デバイス／ホスト・共通レジスタおよびホスト・レジスタのレジスタ・ビットがアクセス可能となります。

#### 1.1.1. レジスタマップ選択方法

本 LSI のデバイス・レジスタとホスト・レジスタとのレジスタマップ選択は、レジスタ・ビットの設定にて行います。

レジスタマップの選択は、SLEEP / SNOOZE / ACTIVE60 / ACT\_DEVICE / ACT\_HOST / ACT\_ALL(「1.5.パワーマネージメント機能」参照)の全パワーマネージメント状態で設定をすることができます。デバイス／ホスト・レジスタマップ選択の設定項目を表 1-1 に示します。

表 1-1 デバイス・モード／ホスト・モード選択の設定項目

項目	レジスタ／ビット	説明
デバイス／ホスト レジスタマップ選 択	HostDeviceSel.HOSTxDEVICE	デバイスとホストのいずれかのレジスタマッ プを選択し、選択した方のレジスタにアクセス できます。 1: ホストマップ選択状態 0: デバイスマップ選択状態

#### 1.1.2. デバイス／ホスト機能選択方法

本 LSI のデバイス／ホストの機能選択方法は、ClkSelect.PORT1x2 ビットの設定値によって異なります。

ClkSelect.PORT1x2 ビットがクリアされている場合(2 ポートモード時)は、パワーマネージメント機能と連動して動作しており、パワーマネージメントのステートによって、使用できる機能が変わります。例えば、GoActDevice により ACT\_DEVICE ステートに遷移した場合は、デバイス機能が使用可能となります。また、GoActHost により ACT\_HOST ステートに遷移した場合には、ホスト機能が使用可能とな

## 1. 機能説明

ります。また、この時、PM\_Control\_1.MonActFunc ビットにて、現在使用可能な機能が確認できます。MonActFunc ビットが 0 にクリアされている場合にはデバイス機能が、1 にセットされている場合にはホスト機能が使用可能となっています。GoActAllDev または GoActAllHost により、ACT\_ALL ステートに遷移した場合は、デバイスまたはホストの使用可能な機能が MonActFunc ビットに表示されます。

ClkSelect.PORT1x2 ビットがセットされている場合(1 ポートモード時)は、前述のレジスタマップ選択に準じて機能が選択されます。つまり、HostDeviceSel.HOSTxDEVICE ビットをクリアした場合にはデバイス機能が、セットした場合にはホスト機能が使用可能となります。

### 1.1.3. ポート状態変化検出ステータス

本 LSI には、USB ポートの状態検出機能があります。

本機能はSLEEP / SNOOZE / ACTIVE60 / ACT\_DEVICE / ACT\_HOST / ACT\_ALL(「1.5.パワーマネージメント機能」参照)の全パワーマネージメントステートで使用できます。また、デバイス／ホストのモードにかかわらず使用できますので、デバイス・モード時にDownstream port(ホストポート)の状態変化を、ホスト・モード時にUpstream port(デバイスポート)の状態変化を検出することができます。(2 ポートモード時)

ホストポートおよびデバイスポートの状態変化は、モード選択のきっかけとして使用することができます。

#### 1.1.3.1.VBUS\_B 端子変化ステータスの使用方法

本ステータスは、デバイスポートの VBUS\_B 端子の状態が変化したことを示します。

VBUS\_B端子変化ステータス関連レジスタを表 1-2 に示します。

表 1-2 VBUS\_B 端子変化ステータス関連レジスタ

項目	レジスタ／ビット	説明
VBUS_B 端子変化ステータス	DeviceIntStat.VBUS_Changed	デバイスポートのVBUS_B端子の状態が変化したことを示します。
VBUS_B 端子変化ステータス割り込みイネーブル	DeviceIntEnb.EnVBUS_Changed	VBUS_B 端子変化ステータスによる MainIntStat.DeviceIntStat ビットのアサートを許可／禁止します。
VBUS_B 端子状態	D_USB_Status.VBUS	VBUS_B端子の状態を示します。

VBUS\_B 端子変化ステータスを使用する場合、ファームウェアは(1)、(2)、(4)～(7)の処理を行います。

- (1) VBUS\_B 端子変化ステータスをクリアします。
- (2) VBUS\_B 端子変化ステータス割り込みイネーブルをセットします。また、MainIntEnb.EnDeviceIntStat ビットをセットします。(割り込みを使用する場合)
- (3) デバイスポートの VBUS\_B 端子の状態が変化すると、VBUS\_B 端子変化ステータスが発行されます。
- (4) VBUS\_B 端子変化ステータスを確認します。
- (5) VBUS\_B 端子変化ステータスをクリアします。
- (6) VBUS\_B 端子変化ステータス割り込みイネーブルをクリアします。
- (7) VBUS\_B 端子状態を確認します。VBUS\_B 端子状態ビットが"1"であれば、VBUS が供給されている(デバイスポートにホストまたはハブが接続されている)状態と判断します。

## 1.1.3.2. 信号ライン変化ステータスの使用方法

本ステータスは、ホストポートの DP 端子および DM 端子の状態が変化したことを示します。  
信号ライン変化ステータス関連レジスタを表 1-3 に示します。

表 1-3 信号ライン変化ステータス関連レジスタ

項目	レジスタ／ビット	説明
信号ライン変化ステータス	HostIntStat.LineStateChanged	ホストポートのDP端子およびDM端子の状態が変化したことを示します。
信号ライン変化ステータス割り込みイネーブル	HostIntEnb.EnLineStateChanged	信号ライン変化ステータスによる MainIntStat.HostIntStat ビットのアサートを許可／禁止します。
VBUS 異常検出ステータス	HostIntStat.VBUS_Err	VBUSFLG_A端子をモニタし、VBUSに異常が発生したことを示します。
VBUS 異常検出ステータス割り込みイネーブル	HostIntEnb.EnVBUS_Err	VBUS 異常検出ステータスによる MainIntStat.HostIntStat ビットのアサートを許可／禁止します。
オペレーションモード	H_XcvrControl.OpMode[1:0]	ホストランシーバマクロ(HTM)のオペレーションモードを設定します。
VBUS イネーブル	H_USB_Control.VBUS_Enb	外部USBパワースイッチをイネーブルにするよう、VBUSEN_A端子から信号を出力します。
ホストポート VBUS 状態	H_USB_Status.VBUS_State	ホストポートのVBUSFLG_A端子の状態(正常／異常)を示します。
ホストポート信号ライン状態	H_USB_Status.LineState[1:0]	USBケーブル上の信号状態を示します。
ホストステート遷移実行	H_NegoControl_1.AutoMode[3:0]	遷移させるホストステートを設定します。

信号ライン変化ステータスは、USBホストとして動作中の接続検出ステータスとは異なり、USBホスト機能を使用していないとき、すなわち、SLEEP / SNOOZE / ACTIVE60 / ACT\_DEVICE / ACT\_ALL(「1.5. パワーマネージメント機能」参照)のいずれかのステートにおいて、ホストポートの信号ライン変化を検出するための特殊なステータスです。USBホスト動作時にHostIntEnb.EnLineStateChangedがイネーブルとなっていると、本ステータスは頻繁にアサートされます。そのため、USBホストモードでの動作時はHostIntEnb.EnLineStateChangedをディセーブルにしてください。

また、USB ホストモードでの動作時以外で VBUS 異常検出ステータスを検出した場合には、直ちに VBUS イネーブルを OFF して VBUS のドライブを停止してください。

信号ライン変化ステータスを使用する際に、ファームウェアは(1)～(4)、(6)～(9)の処理を行います。

- (1) 信号ライン変化ステータスおよび VBUS 異常検出ステータスをクリアします。
- (2) 信号ライン変化ステータスイネーブルおよび VBUS 異常検出ステータスイネーブルをセットします。また、MainIntEnb.EnHostIntStat ビットをセットします。(割り込みを使用する場合)
- (3) H\_XcvrControl レジスタに 0x93 を設定します。
- (4) VBUS イネーブルを ON します。
- (5) ホストポートにデバイスの接続があるとラインステートが SE0 から変化し、信号ライン変化ステータスが発行されます。
- (6) 信号ライン変化ステータスを確認します。
- (7) 信号ライン変化ステータスをクリアします。
- (8) 信号ライン変化ステータスイネーブルをクリアします。
- (9) ホストポート信号ライン状態を確認します。"01"または"10"であれば、ホストポートにデバイ

## 1. 機能説明

---

スが接続されている状態と判断します。

このあと、ホスト・モードを選択した際に、H\_XcvrControlレジスタとVBUS\_Enbビットは、ここで設定した値が保持されています。これらの設定は、ファームウェアがホストステート遷移実行レジスタに適切なコードを設定することで、設定したホストステートに応じてハードウェアが自動設定します。詳細は、「1.3.10.ホストステート管理サポート機能」を参照してください。

## 1.2. USB デバイス制御

本 LSI の USB デバイス機能について説明します。

### 1.2.1. エンドポイント

本 LSI はコントロール転送用のエンドポイント(EP0)と、5 本の汎用エンドポイント(EPa, EPb, EPc, EPd, EPe)を持ちます。エンドポイント EPa, EPb, EPc, EPd, EPe は、それぞれに、バルク、インタラプトまたはアイソクロナス転送用のエンドポイントとして使用できます。

本 LSI のハードウェアは、エンドポイントを提供し、トランザクションの管理を行います。一方、USB に定義されるインタフェイス(以下、USB 定義インタフェイス)の管理機能を提供しません。USB 定義インタフェイスは、ファームウェアで実装して下さい。デバイス固有のデスクリプタ定義に沿って、エンドポイントを適宜設定し、組み合わせて、USB 定義インタフェイスを構成してください。

各エンドポイントには、USB 定義インタフェイスによって決定される固定の基本設定項目と、転送毎に制御を行う可変の制御項目及びステータスがあります。基本設定項目は、チップ初期化時、または、USB 定義インタフェイスの切り替え時等に設定して下さい。

表 1-4 にエンドポイント EP0(デフォルトコントロールパイプ)の基本設定項目を示します。

エンドポイント EP0 は、IN 方向と OUT 方向とで、レジスタセットや FIFO 領域を共有します。エンドポイント EP0 におけるデータステージ及びステータスステージでは、その実行に際して、ファームウェアによって、適宜データトランザクションの方向を設定して下さい。

後述する FIFO エリアに関連付けることにより、トランザクションを実行することが出来るようになります。まず、FIFO 領域を、AREA0StartAdrs\_H,L 及び AREA0EndAdrs\_H,L で確保し、AREA0FIFO\_Clr で初期化した後、AREA0Join\_1.JoinEP0CH0 をセットして下さい。このジョイン処理によって関連付けが行われるまで、FIFO を使用してのデータ転送が行えません。

表 1-4 エンドポイント EP0 の基本設定項目

項目	レジスタ/ビット	説明
マックスパケットサイズ	D_EP0MaxSize	マックスパケットサイズを、FS 動作時には 8,16,32,64 のいずれかの値に設定します。また、HS 動作時には 64 に設定します。エンドポイント EP0 を FIFO の AREA0 領域に関連付けて使用して下さい。
FIFO 領域	AREA0StartAdrs_H,L AREA0EndAdrs_H,L	AREA0 領域に割り当てる FIFO 領域を、アドレスで指定します。FIFO 領域には、マックスパケットサイズ以上を確保してください。
FIFO 関連付け	AREA0Join_1.JoinEP0CH0	エンドポイント EP0 を AREA0 領域に関連付け、EP0 におけるデータ転送を行えるようにします。

表 1-5 に汎用エンドポイント(EPa, EPb, EPc, EPd, EPe)の基本設定項目を示します。エンドポイント EPa, EPb, EPc, EPd, EPe は、トランザクション方向とエンドポイントナンバーを任意に設定出来ますので、5 本までの独立したエンドポイントを使用できます。USB 定義インタフェイスの定義内容に合わせて適宜設定し、また有効にすることによって、USB 定義インタフェイスを構成して下さい。

エンドポイント EPa, EPb, EPc, EPd, EPe は、それぞれ FIFO 領域に関連付けられることによって、トランザクションを実行できるようになります。FIFO 領域を AREA<sub>x</sub>{x=1-5}StartAdrs\_H,L 及び AREA<sub>x</sub>{x=1-5}EndAdrs\_H,L レジスタにより確保し、AREAnFIFO\_Clr にて FIFO を初期化した後、AREA<sub>x</sub>{x=1-5}Join\_1 レジスタを適宜セットして下さい。このジョイン処理によって関連付けが行われて居ないエンドポイントは、トランザクションを実行しません。従って、使用しないエンドポイントでは、ジョイン処理をしないで下さい。

## 1. 機能説明

表 1-5 汎用エンドポイントの基本設定項目

項目	レジスタ/ビット	説明
トランザクション方向	D_EPx{x=a-e}Config_0.INxOUT	各エンドポイントの転送方向を設定します。
マックスパケットサイズ	D_EPx{x=a-e}MaxSize_H, D_EPx{x=a-e}MaxSize_L	各エンドポイントのマックスパケットサイズを、8Byte,16Byte,32Byte,64Byte,512Byte のいずれかの値に設定します。 但し、バルク転送を行うエンドポイントでは、FS モード時は 8/16/32/64Byte のいずれか、HS モード時は 512Byte に設定して下さい。 また、アイソクロナス転送を行うエンドポイントでは、FS モード時は 1~1023Byte のいずれか、HS モード時は 1~1024Byte のいずれかに設定します。
エンドポイントナンバー	D_EPx{x=a-e}Config_0.EndpointNumber	各エンドポイントのエンドポイントナンバーを 0x1~0xF の間の任意の値に設定します。
トグルモード	D_EPx{x=a-e}Config_0.IntEP_Mode	インタラプト転送の動作モードを設定します。バルク転送を行うエンドポイントでは、方向に関らず、"0"に設定して下さい。 IN 方向のエンドポイントでは、トグルシーケンスのモードを設定します。 OUT 方向のエンドポイントでは、インタラプト転送を行う場合に、"1"にセットして下さい。
アイソクロナスモード	D_EPx{x=a-e}Config_0.ISO	アイソクロナス転送を行うエンドポイントで、1 にセットして下さい。
FIFO 領域	AREAx{x=1-5}StartAdrs_H,L AREAx{x=1-5}EndAdrs_H,L	領域をアドレスで指定して確保します。各領域は、関連付けるエンドポイントのマックスパケットサイズ以上のサイズを確保してください。また、FIFO のサイズは転送のスループットに影響する場合があります。
FIFO 関連付け	AREAx{x=1-5}Join_1	各エンドポイントと FIFO 領域とを関連付けます。 組み合わせは下記の通りとして、使用するエンドポイントをのみ関連付けて下さい。 エンドポイント EPa : AREA1 エンドポイント EPb : AREA2 エンドポイント EPc : AREA3 エンドポイント EPd : AREA4 エンドポイント EPe : AREA5

### 1.2.2. トランザクション

本LSIはH/Wでトランザクション実行機能と、ファームウェアに対するトランザクション実行のためのインタフェースを提供します。ファームウェアに対するインタフェースは、制御レジスタとステータスレジスタ、及び、ステータスによりアサートされる割り込み信号として実装されています。ステータスにより割り込みをアサートする設定については、「2.レジスタ」を参照して下さい。

本LSIは個々のトランザクション毎に、ファームウェアに対してステータスを発行します。しかしながら、ファームウェアは必ずしも、個々のトランザクションを管理する必要はありません。LSIはトランザクションへの応答を行う時に、FIFOを参照し、そのデータ数または空き数によって、データ転送を行えるか否かを判断して自動的に処理を行います。

例えば、OUTのエンドポイントであれば、ファームウェアは、CPUインタフェース(DMAリードまたはレジスタリード)、または、IDEインタフェース(IDEライト)により、FIFOからデータを読み出してFIFOに空き領域を作り出すことによって、OUTトランザクションを自動的に連続して実行させることが出来ます。また、INのエンドポイントであれば、ファームウェアは、CPUインタフェース(DMAライトまたはレジスタライト)または、IDEインタフェース(IDEリード)により、FIFOにデータを書き込んでFIFOに有効データを作り出すことによって、INトランザクションを自動的に連続して実行させることが出来ます。

表 1-6 にエンドポイントEP0のトランザクション制御に関する制御項目及びステータスを示します。

表 1-6 エンドポイント EP0 の制御項目及びステータス

項目	レジスタ/ビット	説明
トランザクション方向	D_EP0Control.INxOUT	データステージ及びステータスステージにおいて、転送方向を設定します。
デスクリプタ返信イネーブル	D_EP0Control.ReplyDescriptor	デスクリプタの自動応答を起動します。
デスクリプタ返信アドレス	D_DescAdrs_H, DescAdrs_L	デスクリプタの自動応答によって、返信を行うデータの FIFO 上の先頭アドレスを指定します。
デスクリプタサイズ	D_DescSize_H, DescSize_L	デスクリプタの自動応答によって、返信を行うデータ数を指定します。
制御禁止	D_SETUP_Control.ProtectEP0	このビットがセットされていると、EP0ControlIN 及び EP0ControlOUT レジスタの ForceNAK ビットと ForceSTALL ビットへのアクセスが行えません。 このビットは、RcvEP0SETUP ステータスが立つと、LSI の h/w によってセットされ、CPU によるレジスタアクセスでクリアできます。
ショートパケット送信イネーブル	D_EP0ControlIN.EnShortPkt	マックスパケットサイズに満たない、ショートパケットの送信を有効にします。ショートパケットを送信した IN トランザクションが完結すると、クリアされます。
トグルシーケンスビット	D_EP0ControlIN.ToggleStat, D_EP0ControlOUT.ToggleStat	トグルシーケンスビットの状態を示します。SETUP ステージにより、自動的に初期化されます。
トグルセット	D_EP0ControlIN.ToggleSet, D_EP0ControlOUT.ToggleSet	トグルシーケンスビットをセットします。
トグルクリア	D_EP0ControlIN.ToggleClr, D_EP0ControlOUT.ToggleClr	トグルシーケンスビットをクリアします。

(続く)

## 1. 機能説明

表 1-7 エンドポイント EP0 の制御項目及びステータス(続き)

項目	レジスタ/ビット	説明
強制 NAK 応答	D_EP0ControlIN.ForceNAK, D_EP0ControlOUT.ForceNAK	FIFO のデータ数/空き数に関わらず、IN 又は OUT(PING を含む)トランザクションに NAK 応答します。
STALL 応答	D_EP0ControlIN.ForceSTALL, D_EP0ControlOUT.ForceSTALL	IN 又は OUT(PING を含む)トランザクションに STALL 応答します。
自動 ForceNAK セット	D_EP0ControlOUT.AutoForceNAK	OUT トランザクションの完結毎に、D_EP0ControlOUT.ForceNAK ビットをセットします。
SETUP 受信ステータス	DeviceIntStat.RcvEP0SETUP	SETUP トランザクションが実行されたことを示します。
トランザクションステータス	D_EP0IntStat.OUT_ShortACK, D_EP0IntStat.IN_TranACK, D_EP0IntStat.OUT_TranACK, D_EP0IntStat.IN_TranNAK, D_EP0IntStat.OUT_TranNAK, D_EP0IntStat.IN_TranErr, D_EP0IntStat.OUT_TranErr	トランザクションの結果を示します。
デスクリプタ返信データステージ終了ステータス	D_FIFO_IntStat.DescriptorCmp	デスクリプタ自動応答のデータステージが終了したことを示します。

表 1-8 に、汎用エンドポイント EPa, EPb, EPc, EPd, EPe のトランザクション処理に関する制御項目とステータスを示します。

表 1-8 汎用エンドポイントの制御項目とステータス

項目	レジスタ/ビット	説明
自動 ForceNAK セット	D_EPx{x=a-e}Control.AutoForceNAK	OUT トランザクションの完結毎に、そのエンドポイントの D_EPx{x=a-e}Control.ForceNAK ビットをセットします。
ショートパケット送信イネーブル	D_EPx{x=a-e}Control.EnShortPkt	IN トランザクションに対し、マックスパケットサイズに満たない、ショートパケットの送信を有効にします。ショートパケットを送信した IN トランザクションが完結すると、クリアされます。
ショートパケット受信による自動 ForceNAK セットの禁止	D_EPx{x=a-e}Control.DisAF_NAK_Short	OUT トランザクションにおいて、ショートパケットを受信すると、自動的にそのエンドポイントの D_EPx{x=a-e}Control.ForceNAK ビットをセットする機能(※)を禁止します。 ※：このビットにより禁止しない場合は有効になっています。
トグルシーケンスビット	D_EPx{x=a-e}Control.ToggleStat	トグルシーケンスビットの状態を示します。
トグルセット	D_EPx{x=a-e}Control.ToggleSet	トグルシーケンスビットをセットします。

(続く)



表 1-9 汎用エンドポイントの制御項目とステータス(続き)

項目	レジスタ/ビット	説明
トグルクリア	D_EPx{x=a-e}Control.ToggleClr	トグルシーケンスビットをクリアします。
強制 NAK 応答	D_EPx{x=a-e}Control.ForceNAK	FIFO のデータ数/空き数に関わらず、トランザクションに NAK 応答します。
STALL 応答	D_EPx{x=a-e}Control.ForceSTALL	トランザクションに STALL 応答します。
トランザクションステータス	D_EPx{x=a-e}IntStat.OUT_ShortACK, D_EPx{x=a-e}IntStat.IN_TranACK, D_EPx{x=a-e}IntStat.OUT_TranACK, D_EPx{x=a-e}IntStat.IN_TranNAK, D_EPx{x=a-e}IntStat.OUT_TranNAK, D_EPx{x=a-e}IntStat.IN_TranErr, D_EPx{x=a-e}IntStat.OUT_TranErr	トランザクションの結果を示します。

### 1.2.2.1.SETUP トランザクション

本 LSI は、自ノードのエンドポイント EP0 宛ての SETUP トランザクションを、無条件に実施します。(D\_NegoControl.ActiveUSB ビットによって USB 機能は有効にされている必要が有ります。)

SETUP トランザクションが発行されると、データパケット(8Byte)の全ての内容を D\_EP0SETUP\_0～D\_EP0SETUP\_7 レジスタに格納し、ACK 応答します。また、SetAddress()リクエストを除き、ファームウェアに対し RcvEP0SETUP ステータスを発行します。

本 LSI は SETUP トランザクション中にエラーが発生した場合には、応答せず、ステータスを発行しません。

本 LSI は SETUP トランザクションが完結すると、D\_EP0ControlIN レジスタ及び D\_EP0ControlOUT レジスタの ForceNAK ビットをセットし、ForceSTALL ビットをクリアします。また、ToggleStat ビットをセットします。また、D\_SETUP\_Control.ProtectEP0 ビットをセットします。ファームウェアは、エンドポイント EP0 の設定を終え、次のステージに移行可能になったら、SETUP\_Control.ProtectEP0 ビットをクリアし、D\_EP0ControlIN レジスタまたは D\_EP0ControlOUT レジスタにおいて、該当する方向の ForceNAK ビットをクリアして下さい。

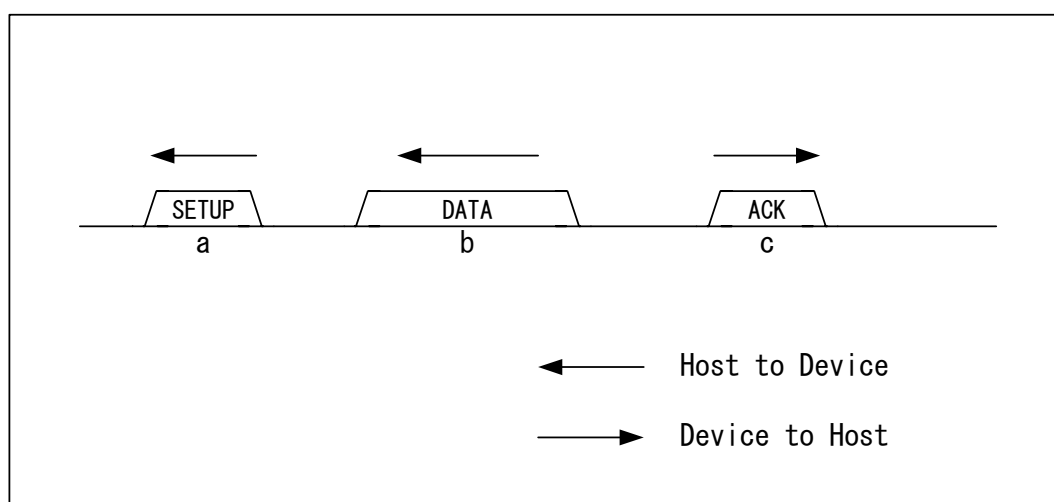


図 1-1 SETUP トランザクション

図 1-1 にSETUPトランザクションの様子を図示します。(a)ホストが、このノードのエンドポイント 0 に宛てたSETUPトークンを発行します。(b)ホストは続けて、8Byte長のデータパケットを送信します。LSIはこのデータをD\_EP0SETUP\_0～D\_EP0SETUP\_7 レジスタに書き込みます。(c)LSIは自動的にACK 応答します。また、自動設定するレジスタを設定し、ファームウェアに対しステータスを発行します。

## 1. 機能説明

### 1.2.2.2. バルク／インタラプト OUT トランザクション

バルク及びインタラプト OUT トランザクションでは、FIFO の空き容量がマックスパケットサイズ以上有ると、データの受信を開始します。

本 LSI は、バルク及びインタラプト OUT トランザクションにおいて、全てのデータが正常に受信できると、トランザクションを完結し、ACK または NYET 応答します。また、ファームウェアに対し、該当するエンドポイントの OUT\_TrانACK ステータス(D\_EPx{x=0,a-e}IntStat.OUT\_TrانACK ビット)を発行します。また、FIFO を更新して、データを受信済みとして、領域を確保します。

また、バルク及びインタラプト OUT トランザクションにおいて、ショートパケットの全てのデータを受信できると、上記のトランザクション完結処理に加え、OUT\_ShortACK ステータス(D\_EPx{x=0,a-e}IntStat.OUT\_ShortACK ビット)を発行します。また、D\_EPx{x=a-e}Control.DisAF\_NAK\_Short ビットがクリアされていると、そのエンドポイントの、D\_EPx{x=a-e}Control.ForceNAK ビットをセットします。

バルク及びインタラプト OUT トランザクションにおいて、トグルミスマッチが発生した場合、トランザクションに ACK 応答しますが、ステータスを発行しません。FIFO は更新されません。

バルク及びインタラプト OUT トランザクションにエラーが発生した場合、トランザクションに応答しません。また、OUT\_TrانErr ステータス(D\_EPx{x=0,a-e}IntStat.OUT\_TrانErr ビット)を発行します。FIFO は更新されません。

バルク及びインタラプト OUT トランザクションにおいて、全てのデータを受信できなかった場合、トランザクションに NAK 応答します。また、OUT\_TrانNAK ステータス(D\_EPx{x=0,a-e}IntStat.OUT\_TrانNAK ビット)を発行します。FIFO は更新されません。

図 1-2 に、完結する場合のバルクまたはインタラプト OUT トランザクションの様子を図示します。(a) ホストが、このノードに存在する OUT 方向のエンドポイントに宛てた OUT トークンを発行します。(b) ホストは続けて、マックスパケットサイズ以内のデータパケットを送信します。LSIはこのデータを、該当するエンドポイントの FIFO に書き込みます。(c) LSI はデータを受信できると、自動的に ACK 応答します。また、自動設定するレジスタを設定し、ファームウェアに対しステータスを発行します。

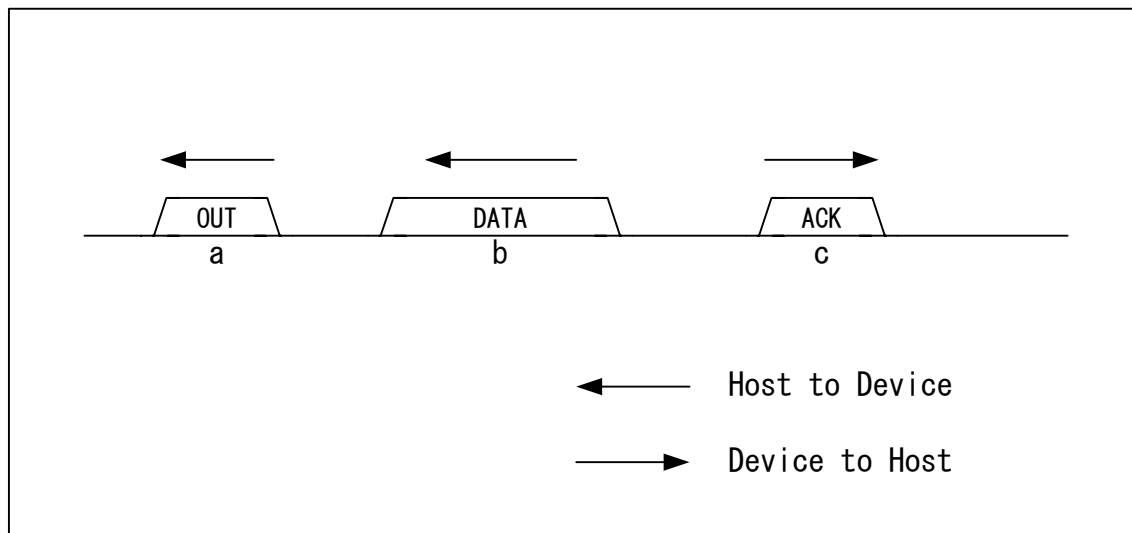


図 1-2 OUT トランザクション

### 1.2.2.3. アイソクロナス OUT トランザクション

アイソクロナス OUT トランザクションでは、FIFO の空き容量がマックスパケットサイズ以上有ると、データの受信を開始します。従いまして、CPU インタフェイスによるレジスタリードや DMA リード等により FIFO のデータを読み出して、空き領域を作りながら受信をすることが出来るよう、FIFO をマックスパケットサイズの倍程度割り当てることにより、良好なスループットを提供します。

アイソクロナス OUT トランザクションにおいて、全てのデータが正常に受信できると、F/W に対し、該当するエンドポイントの OUT\_TrانACK ステータス(EPx{x=a-e}IntStat.OUT\_TrانACK ビット)を発行

します。また、FIFO を更新して、データを受信済みとして、領域を確保します。

アイソクロナス OUT トランザクションにおいて、マックスパケットサイズ未満の全てのデータを受信できると、上記のトランザクション完結処理に加え、OUT\_ShortACK ステータス (EPx{x=a-e}IntStat.OUT\_ShortACK ビット)を発行します。また、EPx{x=a-e}Control.DisAF\_NAK\_Short ビットがクリアされていると、そのエンドポイントの、EPx{x=a-e}ForceNAK ビットをセットします。

アイソクロナス OUT トランザクションにエラーが発生した場合、データを受信せず、FIFO を更新しません。また、OUT\_TrnErr ステータス (EPx{x=a-e}IntStat.OUT\_TrnErr ビット)を発行します。

アイソクロナス OUT トランザクションにおいて、1 パケット分の全てのデータを受信できなかった場合、OUT\_TrnNAK ステータス (EPx{x=a-e}IntStat.OUT\_TrnNAK ビット)を発行します。FIFO は更新されません。

#### 1.2.2.4. バルク／インタラプト IN トランザクション

IN 方向のバルク及びインタラプトのエンドポイントにおいて、FIFO にマックスパケットサイズ分のデータが有るか、または、ファームウェアによってショートパケットの送信が許可されていると、IN トランザクションに応答して、データパケットを返信します。

ショートパケット(データ長ゼロのパケットを含む)の送信許可は、D\_EP0ControlIN.EnShortPkt ビットまたは D\_EPx{x=a-e}Control.EnShortPkt ビットをセットすることで行います。ショートパケットを送信する場合、送信許可後、トランザクションが完結するまでの間、新たなデータをそのエンドポイントの FIFO に書き込まないようにして下さい。

エンドポイント EP0 では、ショートパケットを送信する IN トランザクションが完結すると、D\_EP0ControlIN.ForceNAK ビットがセットされます。

データ返信した IN トランザクションで、ACK を受信すると、トランザクションを完結し、ファームウェアに対し、IN\_TrnACK ステータス (D\_EPx{x=0,a-e}IntStat.IN\_TrnACK ビット)を発行します。また、FIFO を更新して、送信したデータを送信済みとして領域を開放します。

データ返信した IN トランザクションで、ACK を受信しないと、トランザクションを失敗と見なし、ファームウェアに対し、IN\_TrnErr ステータス (D\_EPx{x=0,a-e}IntStat.IN\_TrnErr ビット)を発行します。また、FIFO を更新せず、領域を開放しません。

バルク及びインタラプトの IN 方向のエンドポイントにおいて、FIFO にマックスパケットサイズ分のデータが無く、かつ、ショートパケットの送信が許可されていないと、IN トランザクションに NAK 応答し、ファームウェアに対し、IN\_TrnNAK ステータス (D\_EPx{x=0,a-e}IntStat.IN\_TrnNAK ビット)を発行します。また、FIFO を更新せず、領域を開放しません。

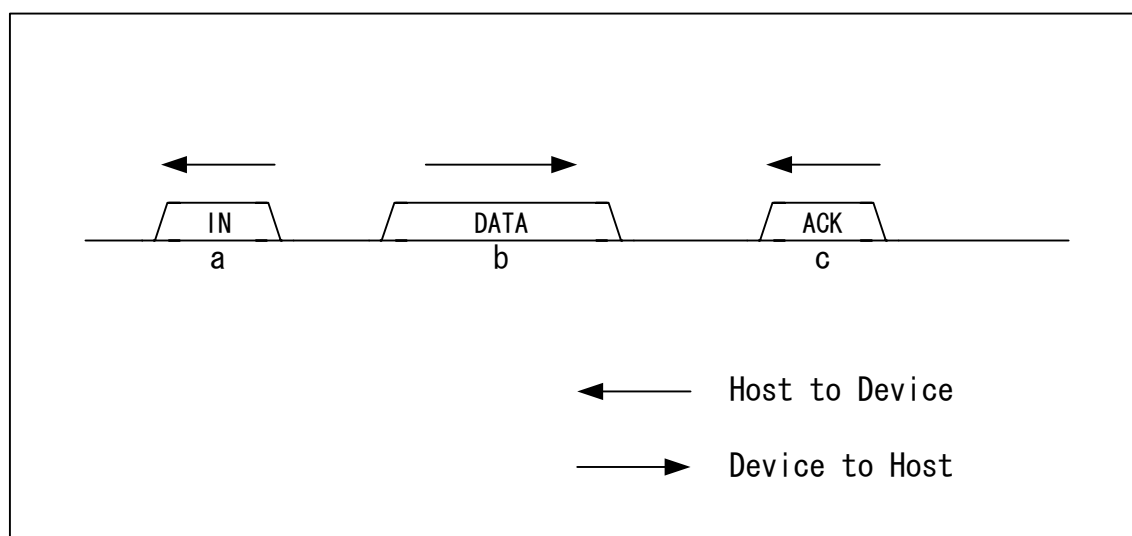


図 1-3 IN トランザクション

図 1-3 に、完結する場合のバルクまたはインタラプト IN トランザクションの様子を図示します。(a)

## 1. 機能説明

ホストが、このノードに存在するIN方向のエンドポイントに宛てたINトークンを発行します。(b)LSIは、このINトランザクションに応答できる場合、マックスパケットサイズ以内のデータパケットを送信します。(c)ホストはACK応答します。LSIはACKを受信すると、自動設定するレジスタを設定し、ファームウェアに対しステータスを発行します。

### 1.2.2.5.アイソクロナス IN トランザクション

IN 方向のアイソクロナスのエンドポイントにおいて、FIFO にマックスパケットサイズ分のデータがあるか、または、F/W によってショートパケットの送信が許可されていると、IN トランザクションに応答して、データパケットを返信します。

ショートパケット(データ長ゼロのパケットを含む)の送信許可は、EPx{x=a-e}Control.EnShortPkt ビットをセットすることで行います。ショートパケットを送信する場合、送信許可後、トランザクションが完結するまでの間、新たなデータをそのエンドポイントの FIFO に書き込まないようにして下さい。

アイソクロナス IN トランザクションにデータパケットを返信すると、トランザクションを完結し、F/W に対し、IN\_TrانACK ステータス(EPx{x=a-e}IntStat.IN\_TrانACK ビット)を発行します。また、FIFO を更新して、送信したデータを送信済みとして領域を開放します。

アイソクロナスの IN 方向のエンドポイントにおいて、FIFO にマックスパケットサイズ分のデータが無く、かつ、ショートパケットの送信が許可されていないと、IN トランザクションにゼロ長データパケットで応答し、F/W に対し、IN\_TrانNAK ステータス(EPx{x=a-e}IntStat.IN\_TrانNAK ビット)を発行します。また、FIFO を更新せず、領域を開放しません。

### 1.2.2.6.PING トランザクション

バルクの OUT 方向のエンドポイントでは、HS 動作時に、PING トランザクションを実行します。

該当するエンドポイントの FIFO 空き容量がマックスパケットサイズ以上であった場合に、PING トランザクションに対して ACK 応答します。また、ファームウェアに対して、ステータスを発行しません。

該当するエンドポイントの FIFO の空き容量がマックスパケットサイズ未満であった場合に、PING トランザクションに対して NAK 応答します。また、ファームウェアに対し、OUT\_TrانNAK ステータス(D\_EPx{x=0,a-e}IntStat.OUT\_TrانNAK ビット)を発行します。

PING トランザクションにおいては、FIFO が更新されることはありません。

図 1-4 に、PING トランザクションに対してACK応答する様子を図示します。(a)ホストが、このノードに存在するOUT方向のエンドポイントに宛てたPINGトークンを発行します。(b)LSIは、FIFOにマックスパケットサイズ分の空きがある場合、このPINGトランザクションに対しACK応答します。また、ファームウェアに対しステータスを発行します。

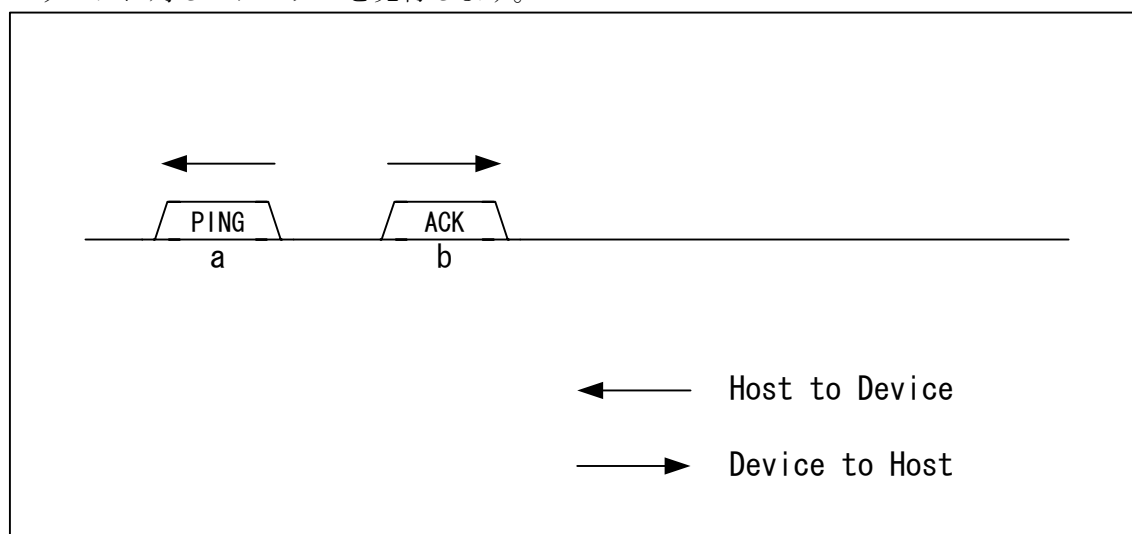


図 1-4 PING トランザクション

### 1.2.3. コントロール転送

エンドポイント EP0 におけるコントロール転送は、**SetAddress()**リクエストを除き、個々のトランザクションの組み合わせとして制御します。**SetAddress()**リクエストは、後述の自動アドレス設定機能により、自動的に処理されます。

図 1-5 にデータステージがOUT方向である場合のコントロール転送の様子を図示します。(a)ホストは、**SETUP**トランザクションによって、コントロール転送を開始します。デバイスのファームウェアはリクエストの内容を解析して、データステージに応答する準備をします。(b)ホストはOUTトランザクションを発行して、データステージを行い、デバイスはデータを受信します。(c)ホストはINトランザクションを発行して、ステータスステージを行い、デバイスはデータ長ゼロの packets を返信します。

データステージの無いコントロール転送は、この例におけるデータステージが無い状態で実施されます。

ステータスステージへの移行は、ホストがデータステージと逆方向のトランザクションを発行することによってなされます。ファームウェアは、**IN\_TranNAK** ステータス(**D\_EP0IntStat.IN\_TranNAK** ビット)を監視して、データステージからステータスステージに移行するきっかけとして下さい。

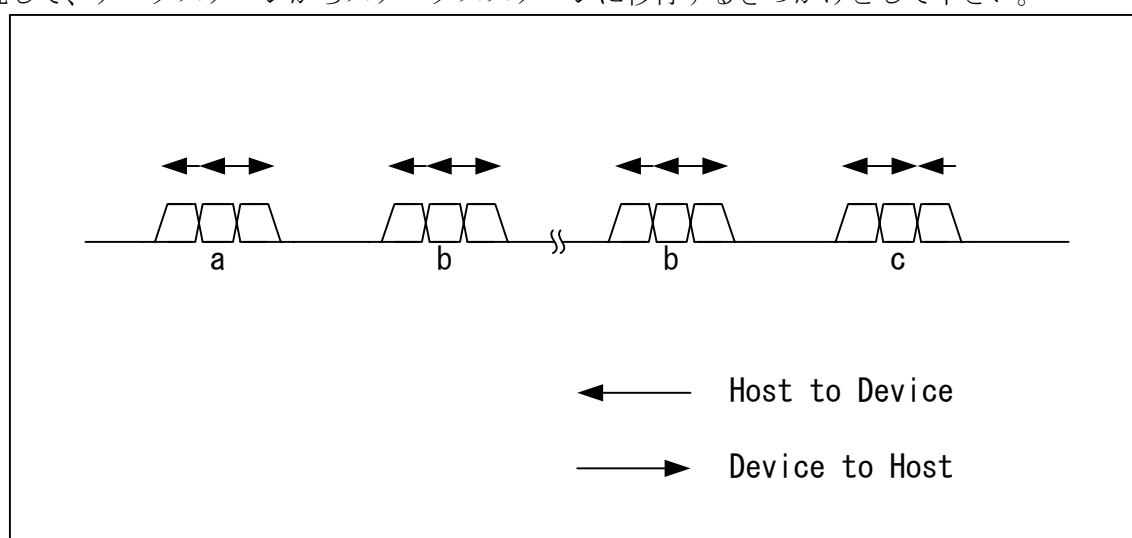


図 1-5 データステージが OUT 方向のコントロール転送

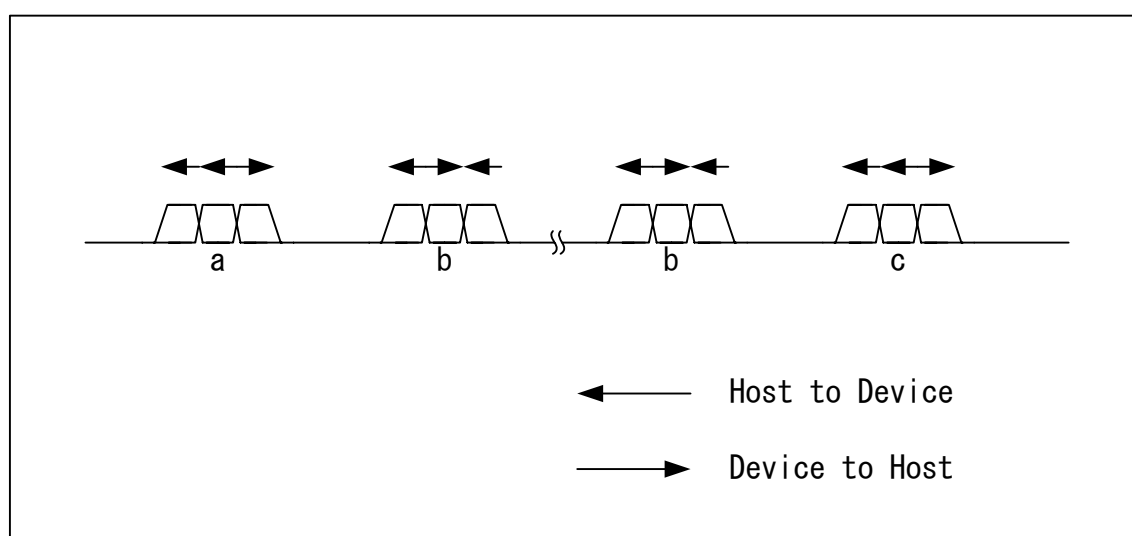


図 1-6 データステージが IN 方向のコントロール転送

図 1-6 にデータステージがIN方向である場合のコントロール転送の様子を図示します。(a)ホストは、

## 1. 機能説明

---

SETUPトランザクションによって、コントロール転送を開始します。デバイスのファームウェアはリクエストの内容を解析して、データステージに応答する準備をします。(b)ホストはINトランザクションを発行して、データステージを行い、デバイスはデータを送信します。(c)ホストはOUTトランザクションを発行して、ステータスステージを行い、デバイスはACK応答します。

ステータスステージへの移行は、ホストがデータステージと逆方向のトランザクションを発行することによってなされます。ファームウェアは、OUT\_TrانNAK ステータス(D\_EP0IntStat.OUT\_TrانNAK ビット)を監視して、データステージからステータスステージに移行するきっかけとして下さい。

コントロール転送のデータステージ及びステータスステージは、通常の OUT 及び IN トランザクションを行いますので、NAK によるフロー制御が有効です。デバイスは定められた時間内に、応答する準備をすることが許されています。

### 1.2.3.1. セットアップステージ

自ノードに宛てられた SETUP トークンを受信すると、自動的にセットアップトランザクションを実行します。

ファームウェアは、RcvEP0SETUP ステータスをモニタし、D\_EP0SETUP\_0～D\_EP0SETUP\_7 レジスタによりリクエストを解析して、コントロール転送を制御して下さい。

受信したリクエストが、OUT 方向のデータステージが有るものであった場合、データステージに移行するため、D\_EP0Control レジスタの INxOUT ビットをクリアして、エンドポイント EP0 を OUT 方向に設定して下さい。

受信したリクエストが、IN 方向のデータステージが有るものであった場合、データステージに移行するため、D\_EP0Control レジスタの INxOUT ビットをセットして、エンドポイント EP0 を IN 方向に設定して下さい。

受信したリクエストが、データステージの無いものであった場合、ステータスステージに移行するため、D\_EP0Control レジスタの INxOUT ビットをセットして、エンドポイント EP0 を IN 方向に設定して下さい。

### 1.2.3.2. データステージ/ステータスステージ

D\_EP0SETUP\_0～D\_EP0SETUP\_7 レジスタを読み出してリクエストを解析した内容に従って、次のステージに移行して下さい。

そのステージが OUT 方向である場合、D\_EP0Control レジスタの INxOUT をクリアして OUT 方向に設定し、D\_EP0ControlOUT レジスタを適宜設定して、ステージをコントロールして下さい。SETUP ステージ終了時は、ForceNAK ビットがセットされています。また、D\_SETUP\_Control.ProtectEP0 ビットがセットされています。

そのステージが IN 方向である場合、D\_EP0Control レジスタの INxOUT をセットして IN 方向に設定し、D\_EP0ControlIN レジスタを適宜設定して、ステージをコントロールして下さい。SETUP ステージ終了時は、ForceNAK ビットがセットされています。また、D\_SETUP\_Control.ProtectEP0 ビットがセットされています。

### 1.2.3.3. 自動アドレス設定機能

本 LSI には、エンドポイント EP0 におけるコントロール転送において、SetAddress()リクエストの処理を自動化する機能があります。

LSI の h/w は D\_EP0SETUP\_0～D\_EP0SETUP\_7 レジスタによってリクエストの内容を確認し、有効な SetAddress()リクエストで有った場合には、ファームウェアに通知することなく、このリクエストのステータスステージの処理に移行します。ステータスステージが完了すると、USB\_Address レジスタにアドレスを設定し、ファームウェアに対し SetAddressCmp ステータス(D\_SIE\_IntStat.SetAddressCmp ビット)を発行します。

ファームウェアはSetAddressCmpステータスを監視し、これが発行されたらUSB\_Addressレジスタにより、アドレスを確認できます。自動アドレス設定機能に関しては、「Appendix D SetAddressリクエストの応答について」を併せて参照して下さい。

### 1.2.3.4. デスクリプタ返信機能

本 LSI には、エンドポイント EP0 におけるコントロール転送において、GetDescriptor()等の複数回発行され同じデータを要求するリクエストに有効な、デスクリプタ返信機能があります。

データステージが IN 転送であるリクエストにおいて、ファームウェアはこの機能を使用することが出来ます。

D\_EP0Control.IN.ForceNAK ビットをクリアして、データステージへの応答を開始する前に、D\_DescAdrs\_H,L レジスタに FIFO のデスクリプタ領域内の、返信するデータの先頭アドレスを、また、D\_DescSize\_H,L レジスタに返信するデータの総バイト数を設定し、D\_EP0Control.ReplyDescriptor ビットをセットして下さい。

デスクリプタ返信機能は、設定数のデータを送り終わるまで、データステージの IN トランザクションに返信してデータパケットを返信し、IN トランザクションを実行します。設定数のデータを送り終えた後に IN トランザクションが発行されると、NAK 応答します。マックスパケットサイズに対し端数のデータが存在すると、デスクリプタ返信機能は、D\_EP0Control.IN.EnShortPkt をセットし、全てのデータを返信するまで、IN トランザクションに返信出来るようにします。

OUT トークンを受信し、ステータスステージへの移行を検知すると、D\_EP0Control.ReplyDescriptor ビットをクリアし、ファームウェアに対し DescriptorCmp ステータス(D\_FIFO\_IntStat.DescriptorCmp ビット)を発行します。DescriptorCmp ステータスを検知したら、ファームウェアはステータスステージを行って下さい。

デスクリプタ領域については、「1.6.FIFO管理」を参照して下さい。

### 1.2.4. バルク転送／インタラプト転送／アイソクロナス転送

汎用エンドポイント EPa, EPb, EPc, EPd, EPeにおけるバルク転送、インタラプト転送、アイソクロナス転送は、データフロー(「1.2.5.データフロー」参照)としても、連続する個々のトランザクション(「1.2.2.トランザクション」参照)としても制御できます。

### 1.2.5. データフロー

OUT 転送及び IN 転送の一般的なデータフローの制御について、説明します。

#### 1.2.5.1. OUT 転送

OUT 転送によって受信したデータは、各エンドポイントに関連付けられた FIFO 上に書き込まれます。FIFO のデータを読み出すには、CPU インタフェイスによるレジスタ読み出し、CPU インタフェイスによる DMA 読み出し、及び IDE へのライト転送の方法があります。

CPU インタフェイスのレジスタリードにより、FIFO のデータを読み出すには、AREAx{x=0-5}.Join\_0.JoinCPU\_Rd ビットにより、ただ一つの FIFO 領域を選択して下さい。選択した FIFO 領域は、FIFO\_Rd レジスタ、または、FIFO\_ByteRd レジスタにより、受信順に読み出すことが出来ます。また、読み出し可能な FIFO のデータ数を、FIFO\_RdRemain\_H,L レジスタにより参照できます。空の FIFO を読み出すことはできませんので、必ず FIFO\_RdRemain\_H,L レジスタによりデータ数を確認し、その数を超えないように読み出して下さい。

CPU インタフェイスの DMA リードにより、FIFO のデータを読み出すには、AREAx{x=0-5}.Join\_0.JoinDMAx{x=0,1}ビットにより、DMA のチャンネル毎にただ一つの FIFO 領域を選択し、DMAx{x=0,1}\_Control.Dir ビットに"1"を設定して下さい。選択した FIFO 領域は、CPU インタフェイスにおいて DMA 手順を実行することにより、受信順に読み出されます。また、FIFO の残りデータ数を、DMAx{x=0,1}\_Remain\_H,L レジスタで参照できます。FIFO が空になると、CPU インタフェイスは自動的に DMA を一時停止してフロー制御を行います。

IDE インタフェイスのライト転送により、FIFO のデータを読み出すには、AREAx{x=0-5}.Join\_0.JoinIDE ビットにより、ただ一つの FIFO 領域を選択し、IDE\_Control.Dir ビットに"1"を設定して下さい。選択した FIFO 領域は、IDE\_Control.IDE\_Go による IDE 転送を実行することにより、受信順に読み出されます。FIFO が空になると、IDE インタフェイスは自動的にライト転送を一時停止して、フロー制御を行います。

FIFO にデータパケットを受信できる空き領域が有れば、OUT トランザクションに自動的に応答して、データを受信できます。従って、ファームウェアによって、個々のトランザクションについての制御を

## 1. 機能説明

---

行うことなく、OUT 転送を行うことができます。但し、D\_EPx{x=a-e}Control.DisAF\_NAK\_Short ビットがクリアされている場合(初期値)、ショートパケット(データ長ゼロのパケットを含む)を受信した場合、そのエンドポイントの D\_EPx{x=a-e}Control.ForceNAK ビットをセットしますので、次のデータ転送を行う準備が出来たら、D\_EPx{x=a-e}Control.ForceNAK ビットをクリアして下さい。

### 1.2.5.2.IN 転送

IN 転送により送信するデータを、各エンドポイントに関連付けられた FIFO 領域に書き込んで下さい。FIFO にデータを書き込むには、CPU インタフェイスによるレジスタ書き込みと、CPU インタフェイスによる DMA 書き込み、及び IDE からのリード転送の方法があります。

CPU インタフェイスのレジスタライトにより、FIFO にデータを書き込むには、AREAx{x=0-5}Join\_0.JoinCPU\_Wr ビットにより、ただ一つの FIFO 領域を選択して下さい。選択した FIFO 領域には、FIFO\_Wr レジスタにより書き込むことができ、書き込み順にデータパケットで送信されます。また、FIFO の空き容量を、FIFO\_WrRemain\_H,L レジスタにより参照できます。フル状態の FIFO へ書き込むことは出来ません。必ず FIFO\_WrRemain\_H,L レジスタにより空き数を確認し、その数を超えないように書き込んで下さい。

CPU インタフェイスの DMA ライトにより、FIFO にデータを書き込むには、AREAx{x=0-5}Join.JoinDMAx{x=0,1}ビットにより、DMA のチャンネル毎にただ一つの FIFO 領域を選択し、DMAx{x=0,1}\_Control.Dir ビットに"0"を設定して下さい。選択した FIFO 領域には、CPU インタフェイスにおいて DMA の手順を実行することにより、書き込まれ、書き込み順にデータパケットで送信されます。FIFO がフルになると、CPU インタフェイスは自動的に DMA を一時停止してフロー制御を行います。

IDE インタフェイスのリード転送により、FIFO へデータを書き込むには、AREAx{x=0-5}Join\_0.JoinIDE ビットにより、ただ一つの FIFO 領域を選択し、IDE\_Control.Dir ビットに"0"を設定して下さい。選択した FIFO 領域は、IDE\_Control.IDE\_Go による IDE 転送を実行することにより、IDE からリードした順に書き込まれ、書き込み順にデータパケットで送信されます。FIFO がフルになると、IDE インタフェイスは自動的にリード転送を一時停止して、フロー制御を行います。

FIFO にマックスパケットサイズ以上のデータが有れば、IN トランザクションに自動的に応答して、データを送信できます。従って、ファームウェアによって、個々のトランザクションについての制御を行うことなく、IN 転送を行うことが出来ます。但し、データ転送の最後にショートパケットを送信する必要がある場合、EnShortPkt ビットをセットして下さい。このビットはショートパケットを送信した IN トランザクションが完結することによってクリアされます。FIFO へのデータ書き込みが終了した時点でセットすることが可能です。また、DMAx{x=0,1}\_FIFO\_Control.AutoEnShort ビットがセットされていると、CPU インタフェイスの DMA 書き込みが終了した時に、マックスパケットサイズに満たない端数データが FIFO に有ると、そのエンドポイントの EnShortPkt ビットを自動的にセットします。

### 1.2.6. バルクオンリーサポート

本 LSI には、エンドポイント EPa, EPb, EPc, EPd, EPe におけるバルク転送において、USB Mass Storage Class(BulkOnly Transport Protocol)に固有の Command Block Wrapper(CBW)の受信及び Command Status Wrapper(CSW)の送信を補助する、バルクオンリーサポート機能があります。

D\_BulkOnlyConfig.EPx{x=a-e}BulkOnly ビットをセットすると、対象となるエンドポイントで、バルクオンリーサポート機能が有効になります。

バルクオンリーサポート機能が有効にされ、CBW サポート、または CSW サポートが実行されている間は、エンドポイントに通常割り当てられた FIFO 領域ではなく、CBW 領域または CSW 領域として割り当てられている領域を使用して、パケットの受信(CBW)または送信(CSW)を行います。

#### 1.2.6.1.CBW サポート

ファームウェアは BulkOnly Transport Protocol のコマンドトランスポートを行うときに、CBW サポートを使用することが出来ます。D\_BulkOnlyConfig.EPx{x=a-e}BulkOnly ビットがセットされると、対応する OUT のエンドポイントで CBW サポートが有効になります。CBW サポートは、ただひとつのエンドポイントで有効になるように設定して下さい。CBW サポートが有効であるときに、



D\_BulkOnlyControl.GoCBW\_Mode ビットをセットすると、CBW サポートが実行され、対象となるエンドポイントにおける OUT トランザクションで受信したデータを CBW として扱います。

データパケットのデータ長が CBW として期待される 31 バイト長であった場合には、データを CBW 領域に保存し、ファームウェアに対し CBW 完了ステータス(D\_BulkIntStat.CBW\_Cmp ビット)を発行します。また、D\_BulkOnlyControl.GoCBW\_Mode ビットを自動的にクリアし、CBW サポートの実行が終了します。また、このとき D\_BulkOnlyControl.GoCSW\_Mode ビットがセットされていると、同時にクリアします。

データパケットのデータ長が、31 バイト長未満か、または、31 バイト長を超えた場合には、データを受信せず、ファームウェアに対し CBW データ長エラーステータス(D\_BulkIntStat.CBW\_LengthErr ビット)を発行します。また、D\_BulkOnlyControl.GoCBW\_Mode ビットを自動的にクリアし、CBW サポートの実行を終了します。また、このとき D\_BulkOnlyControl.GoCSW\_Mode ビットがセットされていると、同時にクリアします。CBW\_Err ステータスが発行された場合、BulkOnly Transport Protocol でフェーズミスマッチが発生していますので、ファームウェアはエンドポイントを STALL するなどして、通信の復旧を行ってください。

対象となるエンドポイントで D\_EPx{x=a-e}Control.ForceSTALL がセットされ、OUT トランザクションに STALL 応答した場合には、ファームウェアに対し CBW エラーステータス(D\_BulkIntStat.CBW\_Err ビット)を発行し、D\_BulkOnlyControl.GoCBW\_Mode ビットをクリアし、CBW サポートの実行が終了します。また、このとき D\_BulkOnlyControl.GoCSW\_Mode ビットがセットされていると、同時にクリアします。

OUT トランザクションに CRC エラーなどのトランザクションエラーが発生した場合は、データを受信せずファームウェアに対し CBW トランザクションエラーステータス(D\_BulkIntStat.CBW\_TranErr ビット)ステータスを発行します。この場合には、D\_BulkOnlyControl.GoCBW\_Mode ビットがクリアされず、CBW サポートの実行が継続します。また、このとき D\_BulkOnlyControl.GoCSW\_Mode ビットがセットされていてもクリアされません。

CBW 領域に受信したデータは、RAM\_Rd 機能を用いて読み出すことができます。

#### 1.2.6.2.CSW サポート

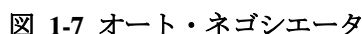
ファームウェアは BulkOnly Transport Protocol のステータストランスポートを行うときに、CSW サポートを使用することが出来ます。D\_BulkOnlyConfig.EPx{x=a-e}BulkOnly ビットがセットされると、対応する IN のエンドポイントで CSW サポートが有効になります。CSW サポートは、ただひとつのエンドポイントで有効になるように制御して下さい。CSW サポートが有効であるときに、D\_BulkOnlyControl.GoCSW\_Mode ビットをセットすると、CSW サポートが実行され、対象となるエンドポイントにおける IN トランザクションを CSW として扱い、CSW 領域から 13 バイトのデータを送信します。

IN トランザクションにおいて、13 バイトの CSW データをホストへ返信した後に、ホストからの ACK を受信してトランザクションを完結した場合には、ファームウェアに対し CSW 完了ステータス(D\_BulkIntStat.CSW\_Cmp ビット)を発行します。また、D\_BulkOnlyControl.GoCSW\_Mode ビットを自動的にクリアして CSW サポートの実行を終了します。また、同時に D\_BulkOnlyControl.GoCBW\_Mode ビットをセットして CBW サポートの実行を開始します。

IN トランザクションにおいて、13 バイトのデータをホストへ返信した後に、ホストからの ACK が受信できなかった場合は、ファームウェアに対し CSW エラーステータス(D\_BulkIntStat.CSW\_Err ビット)を発行します。この時、D\_BulkOnlyControl.GoCSW\_Mode ビットをクリアせずに CSW サポートの実行を継続します。また、同時にハードウェアが D\_BulkOnlyControl.GoCBW\_Mode ビットをセットして CBW サポートの実行を開始します。即ち、この場合には、CSW サポートの実行と、CBW サポートの実行が同時に行われている状態となります。もし、ホストが CSW を受信できずエラーとなっていた場合には、CSW のリトライが行われますが、CSW サポートが実行中なので応答することができます。また、デバイスが ACK を受信できずにエラーとなった場合には、次の CBW が行われますが、CBW サポートが実行中なので、応答することができ、また、CBW サポートが行われることによって CSW サポートの実行が終了されます。

CSW 領域へは、RAM\_WrDoor 機能を用いてデータを書き込むことができます。

サスペンド検出、リセット検出、HS Detection Handshake 実行、レジューム検出、リストア実行を USB バスの状態を逐次チェックしながら自動的に行います。実際に何が行われたかは、各割り込み (DetectRESET、DetectSUSPEND、ChirpCmp、RestoreCmp) をチェックすることにより確認することが出来ます。



D\_NegoControl.EnAutoNego ビットをクリアしている時に、このステートに入ります。

オート・ネゴシエーション機能を有効にする時には、D\_NegoControl.EnAutoNego ビットをセットする前にリセット検出割り込み許可ビット(D\_SIE\_IntEnb.EnDetectRESET)、サスペンド検出割り込み許可ビット(D\_SIE\_IntEnb.EnDetectSUSPEND)をセットし、両イベント検出割り込みを許可してください。

オート・ネゴシエーション機能を有効にすると、内部イベント検出機能を有効にします。オート・ネゴシエーション機能を有効にしている間は、D\_NegoControl.DisBusDetect ビットを絶対にセットしないでください。

#### 1.2.7.2.IDLE

リセット検出待ち、サスペンド検出待ちをするステートです。

現在の USB スピードが HS の時には、USB バス上にバス・アクティビティが 3ms 以上検出できなかった場合に、一旦 FS のターミネーションを有効にし、FS-J が検出された場合はサスペンド、SE0 が検出された場合はリセットと判断します。現在のスピードが FS の時には、2.5us 以上の SE0 が検出された場合はリセット、3ms 以上バス・アクティビティが検出できなかった場合はサスペンドと判断します。これらの判断と同時に、リセット検出割り込み、またはサスペンド検出割り込みが発生し、D\_SIE\_IntStat.DetectRESET ビット、または D\_SIE\_IntStat.DetectSUSPEND ビットがセットされます。

サスペンドと判断した場合、一旦イベント検出機能を停止し、DET\_SUSPEND ステートに入ります。

リセットと判断した場合、一旦イベント検出機能を停止し、WAIT\_TIM3US ステートに入ります。

#### 1.2.7.3.WAIT\_TIM3US

リセット検出後、HS Detection Handshake を実行するまでの時間を調整しています。一定時間経過後(約 3us 後)、WAIT\_CHIRP ステートに入ります。

#### 1.2.7.4.WAIT\_CHIRP

D\_NegoControl.GoChirp ビットを自動的にセットし、HS Detection Handshake を実行します。HS Detection Handshake が終了すると、Chirp 終了割り込みステータス(D\_SIE\_IntStat.ChirpCmp)がセットされ、WAIT\_RSTEND ステートに入ります。HS Detection Handshake の詳細については、「1.2.7.11.5.HS Detection Handshake」を参照してください。

また、D\_NegoControl.DisableHS ビットをセットしている場合は、HS Detection Handshake を実行せずに、Chirp 終了割り込みステータス(D\_SIE\_IntStat.ChirpCmp)がセットされ、WAIT\_RSTEND ステートに入ります。

なお、このステート終了後は、D\_USB\_Status.FSxHS ビットに設定された転送スピードにて動作することとなります。転送スピードが変化したことを検出する必要がある場合は、前述の Chirp 終了割り込みを有効にするために、D\_SIE\_IntEnb.EnChirpCmp ビットをセットしてください。

#### 1.2.7.5.WAIT\_RSTEND

リセット期間が終了するまで、このステートにて待ちます。HS 時はホストからの Chirp 送信(この IC にとっては受信)が終了したこと、FS 時は SE0 から J に遷移したことをもって、リセット期間の終了と判断します。

リセット期間終了と判断した後、イベント検出機能を有効にし、再度 IDLE ステートに入ります。

#### 1.2.7.6.DET\_SUSPEND

サスペンドと判断された場合に、自動的に D\_NegoControl.InSUSPEND ビットがセットされ、IN\_SUSPEND ステートに入ります。この D\_NegoControl.InSUSPEND ビットによって、FS-J からのバスの遷移を検出する機能を有効にし、ホストからのレジューム及びリセットを検出ようになります。

サスペンド中に実際に消費電流を軽減するかどうかは、アプリケーションに依存します。本 LSI では、2 段階の消費電流軽減策(スヌーズ、スリープ)を持っています。詳しい内容、さらには制御方法につきましては、「1.5.パワーマネジメント機能」を参照してください。

また、この時サスペンド終了指示であるレジューム(FS-K)を検出するために、F/W にて D\_SIE\_IntEnb.EnNonJ ビットをセットし、NonJ 割り込みを許可してください。

#### 1.2.7.7.IN\_SUSPEND

NonJ 割り込みステータス(D\_SIE\_IntStat.NonJ)がセットされた場合サスペンドからの復帰指示であると判断し、D\_NegoControl.InSUSPEND ビットを F/W にてクリアすると、CHK\_EVENT ステートに入ります。

## 1. 機能説明

---

リモート・ウェイクアップ機能を有効にしているアプリケーションで、自発的にサスペンドから復帰する場合には、このステートの中で `D_NegoControl.SendWakeup` ビットをセットし、1ms 以上、15ms 以下の間 `FS-K` を出力してください。

### 1.2.7.8.CHK\_EVENT

USB ケーブル上をチェックし、`FS-K` を検出した場合レジュームであると判断し、`SE0` を検出した場合リセットであると判断します。レジュームと判断した場合は、`D_NegoControl.RestoreUSB` ビットをセットし、サスペンド前の転送スピード(`D_USB_Status.FSxHS` の値に従う)に戻ります。リセットと判断した場合は、`IDLE` ステートからの遷移と同じく、一旦イベント検出機能を停止し、`WAIT_TIM3US` ステートに入ります。

もし、`FS-K` でも `SE0` でも無いステートを検出した場合には、オート・ネゴシエーション・エラー割り込みステータス(`D_SIE_IntStat.AutoNegoErr`)ビットをセットし、`ERR` ステートに入ります。

### 1.2.7.9.WAIT\_RESTORE

`D_SIE_IntStat.RestoreCmp` ビットがセットされると、イベント検出機能を有効にし、`IDLE` ステートに入ります。

### 1.2.7.10.ERR

一旦このステートに遷移すると、オート・ネゴシエーション機能を停止させない限り、このステートから抜けません。このステートは、USB 規格上在りえません。

なお、どのステートにおいても、USB ケーブルが抜かれたことによる判断を行っていないため、もし USB ケーブルが抜かれた場合には、すぐにオート・ネゴシエーション機能を停止してください。

### 1.2.7.11. 各ネゴシエーション機能の単体説明

オートネゴシエーションにより実施されている、各バスイベントの処理について、個別にその実施方法を説明します。オートネゴシエーション機能は、本 LSI のレジスタインタフェースを通じて制御を行っていますので、通常はオートネゴシエーションで自動的に行われている処理を、f/w で実施することも出来ます。

#### 1.2.7.11.1. サスペンド検出(HS モード)

本 LSI が HS モードで動作している時に、3ms 以上送受信が何も検出されなかった場合(T1)、FS モードに自動的に移行します(HS のターミネーションを無効にし、FS のターミネーション(Rpu)を有効にします)。この動作により DP は"H"になり、D\_USB\_Status.LineState[1:0]ビットで"J"を確認することができます(もし、"SE0"を検出した場合は、リセット(後述)となることに注意)。その後 T2 の時点で依然"J"が検出された場合、D\_SIE\_IntStat.DetectSUSPEND ビットがセットされますので、USB のサスペンドステートであると判断します。

この時、D\_SIE\_IntEnb.EnDetectSUSPEND ビットおよび DeviceIntEnb.EnD\_SIE\_IntStat ビットがセットされ、MainIntEnb.EnDeviceIntStat がセットされている場合には、同時に XINT 信号がアサートされます。以下の図では、USB サスペンド中にスヌーズを行う時の動作を表しています。

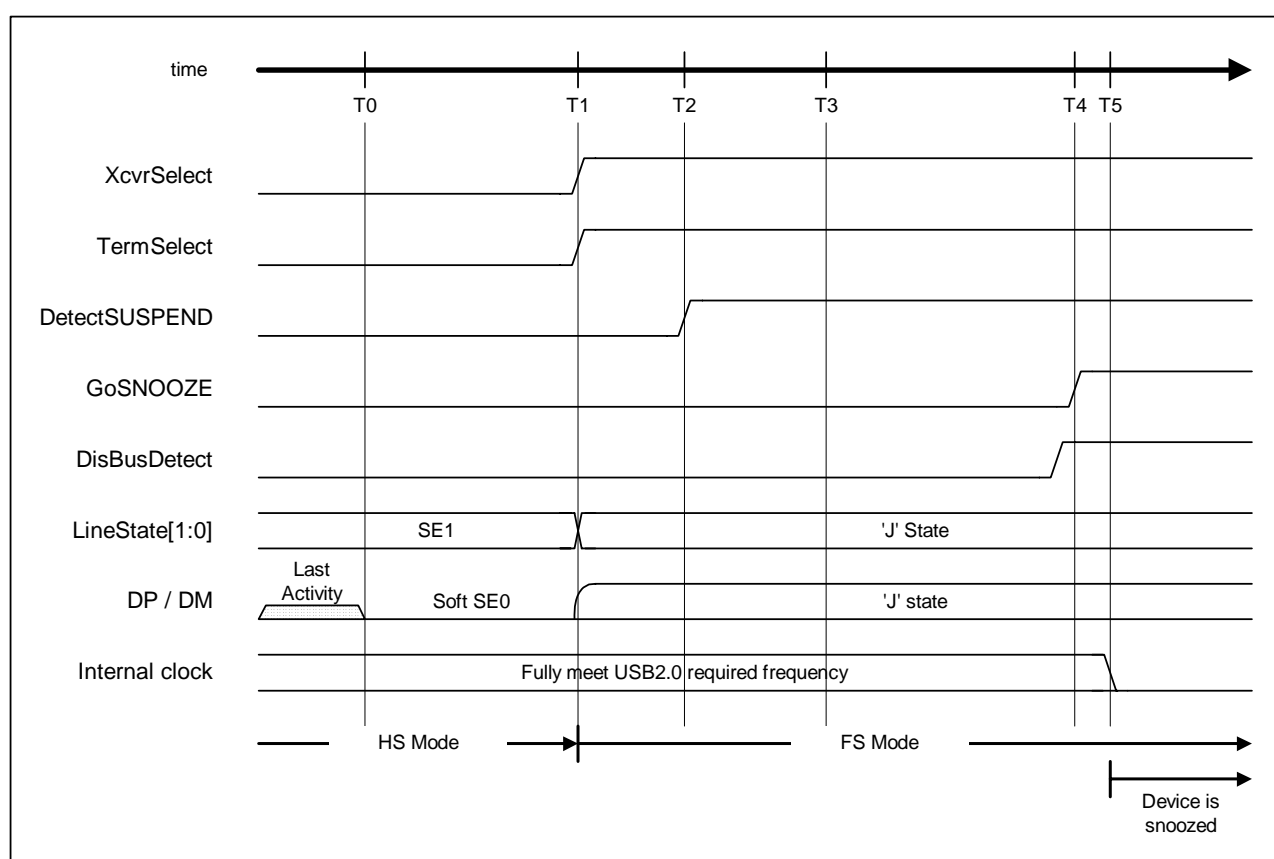


図 1-8 Suspend Timing (HS mode)

## 1. 機能説明

表 1-10 Suspend Timing Values (HS mode)

Timing Parameter	Description	Value
T0	最後のバス・アクティビティ。	0 (reference)
T1	この時点で依然バス・アクティビティが無い場合、XcvtSelect、TermSelect を"1"にセットし、HS モードから FS モードに切り替える。	$HS\_Reset\ T0 + 3.0ms < T1$ $\{T_{WTREV}\} < HS\_Reset\ T0 + 3.125ms$
T2	LineState[1:0]をサンプリングする。この時"J"なら、DetectSUSPEND が"1"になり、USB のサスペンドステートと判断する。	$T1 + 100\mu s < T2 \{T_{WTRSTHS}\} < T1 + 875\mu s$
T3	これより前では、RESUME を発行してはいけない。	$HS\_Reset\ T0 + 5ms \{T_{WTRSM}\}$
T4	GoSNOOZE を"1"にセットし、完全にスヌーズに移行。これ以降はVBUSからUSBで規定されたサスペンド電流以上を引っ張ってはいけない。 (スヌーズ移行前に、DisBusDetect を"1"にセット)	$HS\_Reset\ T0 + 10ms \{T_{2SUSP}\}$
T5	内部クロックが完全停止。	$T5 < T4 + 10\mu s$

注： {} は、USB2.0 規格書で規格されている名称である。

## 1.2.7.11.2. サスペンド検出(FS モード)

本 LSI が FS モードで動作している時に、3ms 以上送受信が何も検出されなかった場合、または D\_USB\_Status.LineState[1:0] ビットに "J" を検出し続け (T1)、さらに T2 の時点で依然 "J" が検出された場合、USB のサスペンドステートであると判断し、SIE\_IntStat.DetectSUSPEND ビットがセットされます。

この時、D\_SIE\_IntEnb.EnDetectSUSPEND ビットおよび DeviceIntEnb.EnD\_SIE\_IntStat ビットがセットされ、MainIntEnb.EnDeviceIntStat がセットされている場合には、同時に XINT 信号がアサートされます。以下の図では、USB サスペンド中にスヌーズを行う時の動作を表しています。

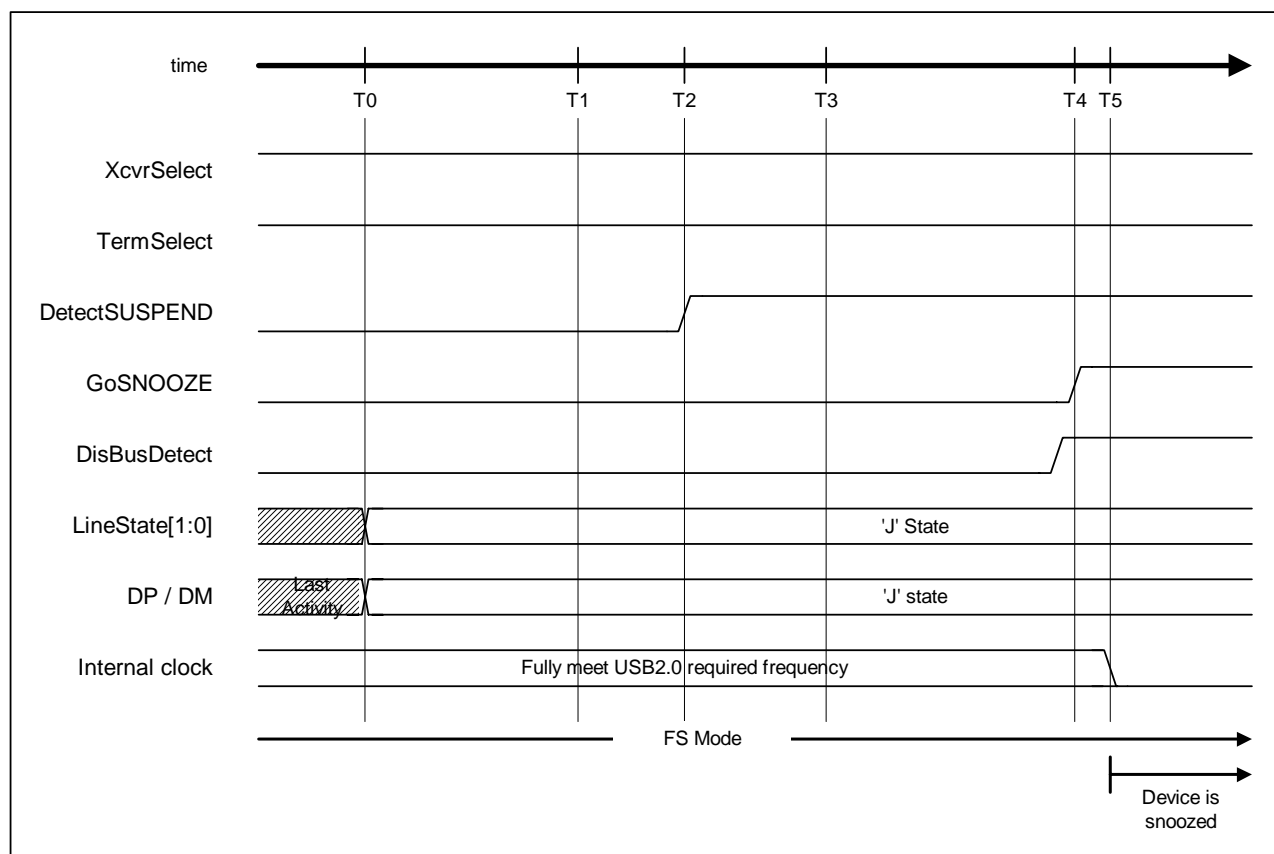


図 1-9 Suspend Timing (FS mode)

表 1-11 Suspend Timing Values (FS mode)

Timing Parameter	Description	Value
T0	最後のバス・アクティビティ。	0 (reference)
T1	この時点で依然バス・アクティビティが無い。	$T0 + 3.0\text{ms} < T1 \{T_{WTREV}\} < T0 + 3.125\text{ms}$
T2	LineState[1:0] をサンプリングする。この時 "J" なら、DetectSUSPEND が "1" になり、USB のサスペンドステートと判断する。	$T1 + 100\text{us} < T2 \{T_{WTWRSTHS}\} < T1 + 875\text{us}$
T3	これより前では、RESUME を発行してはいけない。	$T0 + 5\text{ms} \{T_{WTRSM}\}$
T4	GoSNOOZE を "1" にセットし、完全にスヌーズに移行。これ以降は VBUS から USB で規定されたサスペンド電流以上を引っ張ってはいけない。(スヌーズ移行前に、DisBusDetect を "1" にセット)	$T0 + 10\text{ms} \{T_{2SUSP}\}$
T5	内部クロックが完全停止。	$T5 < T4 + 10\text{us}$

注： {} は、USB2.0 規格書で規格されている名称である。

## 1. 機能説明

### 1.2.7.11.3. リセット検出(HS モード)

本 LSI が HS モードで動作している時に、3ms 以上送受信が何も検出されなかった場合、FS モードに自動的に移行します(HS のターミネーションを無効にし、FS のターミネーション(Rpu)を有効にします)。リセットされている場合、この動作が行われても DP ラインは"L"になったままで、この結果 D\_USB\_Status.LineState[1:0]ビットでも"SE0"を検出することができます。T2 の時点で依然"SE0"が検出された場合には、D\_SIE\_IntStat.DetectRESET ビットがセットされます。

この時、D\_SIE\_IntEnb.EnDetectRESET ビットおよび DeviceIntEnb.EnD\_SIE\_IntStat ビットがセットされ、MainIntEnb.EnDeviceIntStat がセットされている場合には、同時に XINT 信号がアサートされます。

以降は、D\_NegoControl.DisBusDetect ビットをセットした後に、HS Detection Handshake(後述)を行います。

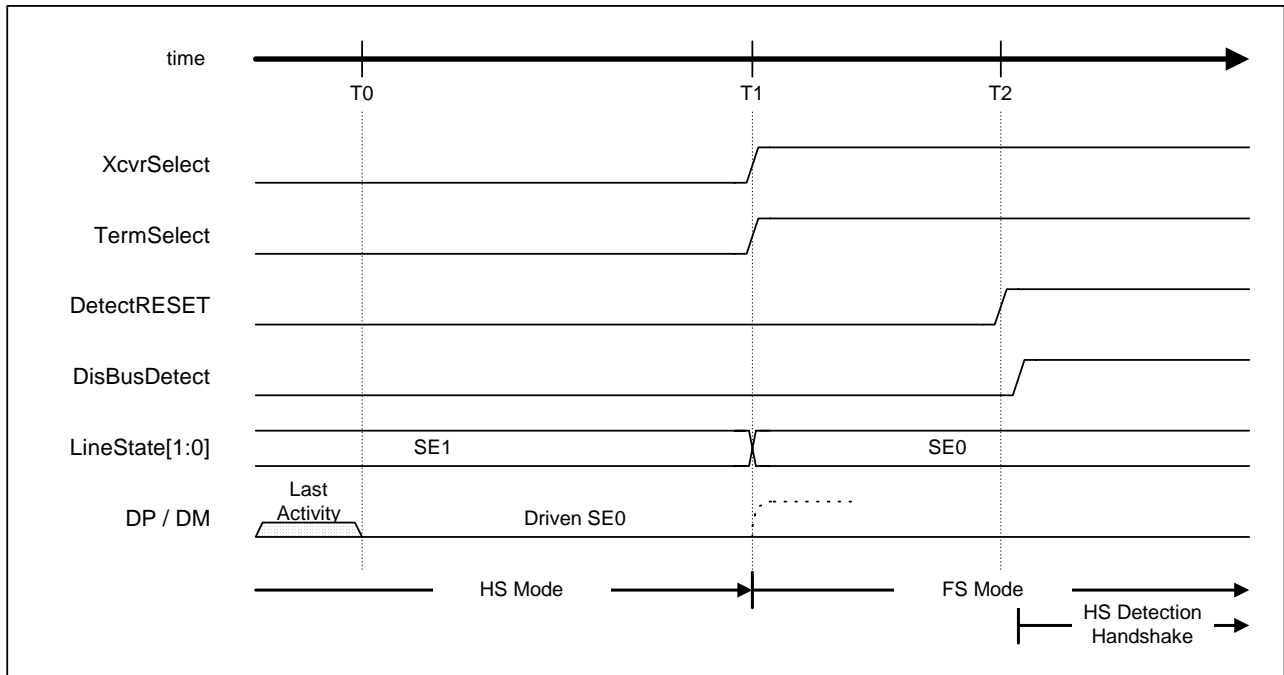


図 1-10 Reset Timing (HS mode)

表 1-12 Reset Timing Values (HS mode)

Timing Parameter	Description	Value
T0	最後のバス・アクティビティ。	0 (reference)
T1	この時点で依然バス・アクティビティが無い場合、XcvtSelect、TermSelect を"1"にセットし、HS モードから FS モードに切り替える。	$HS\_Reset\ T0 + 3.0ms < T1$ $\{T_{WTREV}\} <$ $HS\_Reset\ T0 + 3.125ms$
T2	LineState[1:0]をサンプリングする。この時"SE0"なら、DetectRESET が"1"になり、リセットへの移行と判断する。 リセット指示の検出後、DisBusDetect を"1"にセットし、以降 HS Detection Handshake を行う。	$T1 + 100us < T2 \{T_{WTWRSTHS}\} <$ $T1 + 875us$

注： {} は、USB2.0 規格書で規格されている名称である。



## 1.2.7.11.4. リセット検出(FS モード)

本 LSI が FS モードで動作している時に、2.5us 以上 D\_USB\_Status.LineState[1:0] ビットに "SE0" を検出し続けた場合には (T1)、D\_SIE\_IntStat.DetectRESET ビットがセットされます。

この時、D\_SIE\_IntEnb.EnDetectRESET ビットおよび DeviceIntEnb.EnD\_SIE\_IntStat ビットがセットされ、MainIntEnb.EnDeviceIntStat がセットされている場合には、同時に XINT 信号がアサートされます。

以降は D\_NegoControl.DisBusDetect ビットをセットした後に、HS Detection Handshake(後述)を行います。

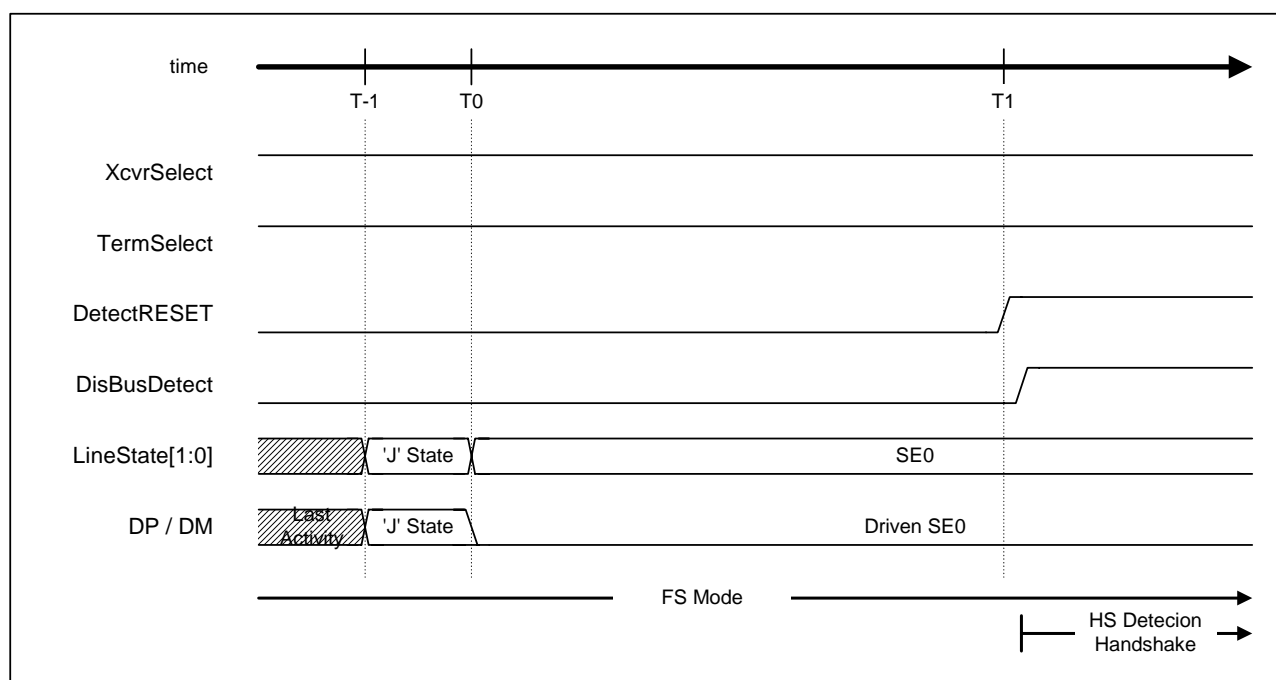


図 1-11 Reset Timing (FS mode)

表 1-13 Reset Timing Values (FS mode)

Timing Parameter	Description	Value
T-1	最後のバス・アクティビティ。	
T0	ホスト Downstream port からのリセットの指示開始。	0 (reference)
T1	"SE0"が継続されている場合、DetectRESET が"1"になり、リセットへの移行と判断する。 リセット指示の検出後、DisBusDetect を"1"にセットし、以降 HS Detection Handshake を行う。	HS Reset $T0 + 2.5\mu s < T1$ {T <sub>WTREV</sub> }

注： {} は、USB2.0 規格書で規格されている名称である。

## 1. 機能説明

---

### 1.2.7.11.5. HS Detection Handshake

HS Detection Handshake は、サスペンド中、FS 動作中、或いは HS 動作中の 3 状態のいずれかから、ホスト Downstream port からの"SE0"のアサートにより開始されます(上記状態からのリセットが開始された時)。詳細は、USB2.0 規格書を参照してください。

ここで、上記 3 状態から HS Detection Handshake に移行する方法について説明します。

本 LSI がサスペンド状態では、バス上に"SE0"を検出後直ちに HS Detection Handshake に移行します。

本 LSI が FS モードで動作している状態では、2.5us 以上の"SE0"を検出後、HS Detection Handshake に移行します。

本 LSI が HS モードで動作している状態では、3.0ms 以上の"SE0"を検出後、まず USB のサスペンド状態なのかリセットなのかを判断しなければならない為、一旦 FS モードに切り替わります。この時動作としては、D\_XcvrControl.XcvrSelect、D\_XcvrControl.TermSelect の両ビットを FS モードに切り替え、HS ターミネーションを無効にし、FS ターミネーションを有効にします。これらのモード切り替えは、h/w によって 3.125ms 以内に行われます。h/w は、このモード切り替えから 100us 以上 875us 以内に D\_USB\_Status.LineState[1:0]ビットをチェックし、"J"なら USB のサスペンド状態として判断し、"SE0"ならリセットと判断します。この時、リセットと判断された場合には、その後 HS Detection Handshake に移行します。

いずれの場合も、リセットは最小 10ms 存在しますが、移行する前の状態(HS もしくは FS)により、タイミングが多少異なります。ここでは、リセットが開始された時間を"HS Reset T0"として定義し、以降は、この"HS Reset T0"からの動作について説明します。

動作中の場合は内部クロックも充分静定しており問題ありませんが、サスペンド中にスリープ/スヌーズさせていた場合には、リセット検出時には内部クロックが出力されていません。このため HS Detection Handshake を行うために、必ず PM\_Control\_0.GoActAllDev ビットまたは PM\_Control\_0.GoActDevice ビットを"1"にセットし、内部クロックを動作させてください。この動作の詳細は、「1.5.パワーマネージメント機能」を参照してください。

この時、D\_SIE\_IntEnb.EnChirpCmp ビットおよび DeviceIntEnb.EnD\_SIE\_IntStat ビットがセットされ、MainIntEnb.EnDeviceIntStat がセットされている場合には、同時に XINT 信号がアサートされますので、HS Detection Handshake が終了したと判断してください。



## 1. 機能説明

表 1-14 HS Detection Handshake Timing Values (FS mode)

Timing Parameter	Description	Value
T0	HS Detection Handshake 開始。	0 (reference)
T1	HS トランシーバをイネーブルにし、GoChirp を"1"にセットして、Chirp K を送出開始。	$T0 < T1 < HS\ Reset\ T0 + 6.0ms$
T2	Chirp K 送出終了。最小 1ms の間は送出しなければならない。	$T1 + 1.0ms \{T_{UCH}\} < T2 < HS\ Reset\ T0 + 7.0ms \{T_{UCHEND}\}$
T3	ホスト downstream port が HS をサポートしている場合、ここから Chirp K を送出開始する。	$T2 < T3 < T2 + 100us \{T_{WTDCH}\}$
T4	Chirp を検出出来ない場合、この時点で FS モードに戻り、ChirpCmp が"1"にセットされ、リセットシーケンスが終了するのを待つ。	$T2 + 1.0ms < T4 \{T_{WTFS}\} < T2 + 2.5ms$
T5	リセットシーケンスの終了。	$HS\ Reset\ T0 + 10ms \{T_{DRST} (Min)\}$
T6	FS モードでの通常動作。	T6

注： {} は、USB2.0 規格書で規格されている名称である。

注： 最小 1ms の Chirp K を生成するために、66000 サイクル(内部クロック：60MHz)で判断する。

#### 1.2.7.11.5.2. HS のホストダウンストリームポートに繋がれた場合

本 LSI が、HS をサポートしているホスト downstream port に接続された時の動作を示します。HS Detection Handshake の開始時(T0)では、D\_XcvrControl.XcvrSelect と D\_XcvrControl.TermSelect は両ビットともに FS モードでなければなりません(FS ターミネーション、即ち DP のプルアップ抵抗(Rpu)を有効にし、HS ターミネーションを無効にします)。

まずは D\_NegoControl.GoChirp ビットをセットします。すると D\_XcvrControl.OpMode[1:0]ビットが "Disable Bit Stuffing and NRZI encoding" になり、"0" で埋め尽くされたデータが準備されます(T1)。これは、バス上に "HS K" (Device Chirp) を送出するためのものです。また同時に、D\_XcvrControl.XcvrSelect ビットが HS モードに設定され、かつ送信可能状態に設定されることで、downstream port に "HS K" (Device Chirp) が送出されます。送出終了後、downstream port からの Host Chirp を待ちます(T2)。ここでは downstream port は HS をサポートしているので、"HS K" (Chirp K)、"HS J" (Chirp J) を交互に連続して送出してきます(T3)。この状態を USB\_Status.LineState[1:0] ビットで Chirp K-J-K-J-K-J と最低 6 回検出したところで(T6)、XcvrControl.TermSelect ビットを HS モードに自動的に切り替え(T7)、完全な HS モードに移行します。これと同時に、D\_NegoControl.GoChirp ビットがクリアされるとともに D\_NegoStatus.FSxHS ビットがクリアされ、さらに D\_SIE\_IntStat.ChirpCmp ビットがセットされます。

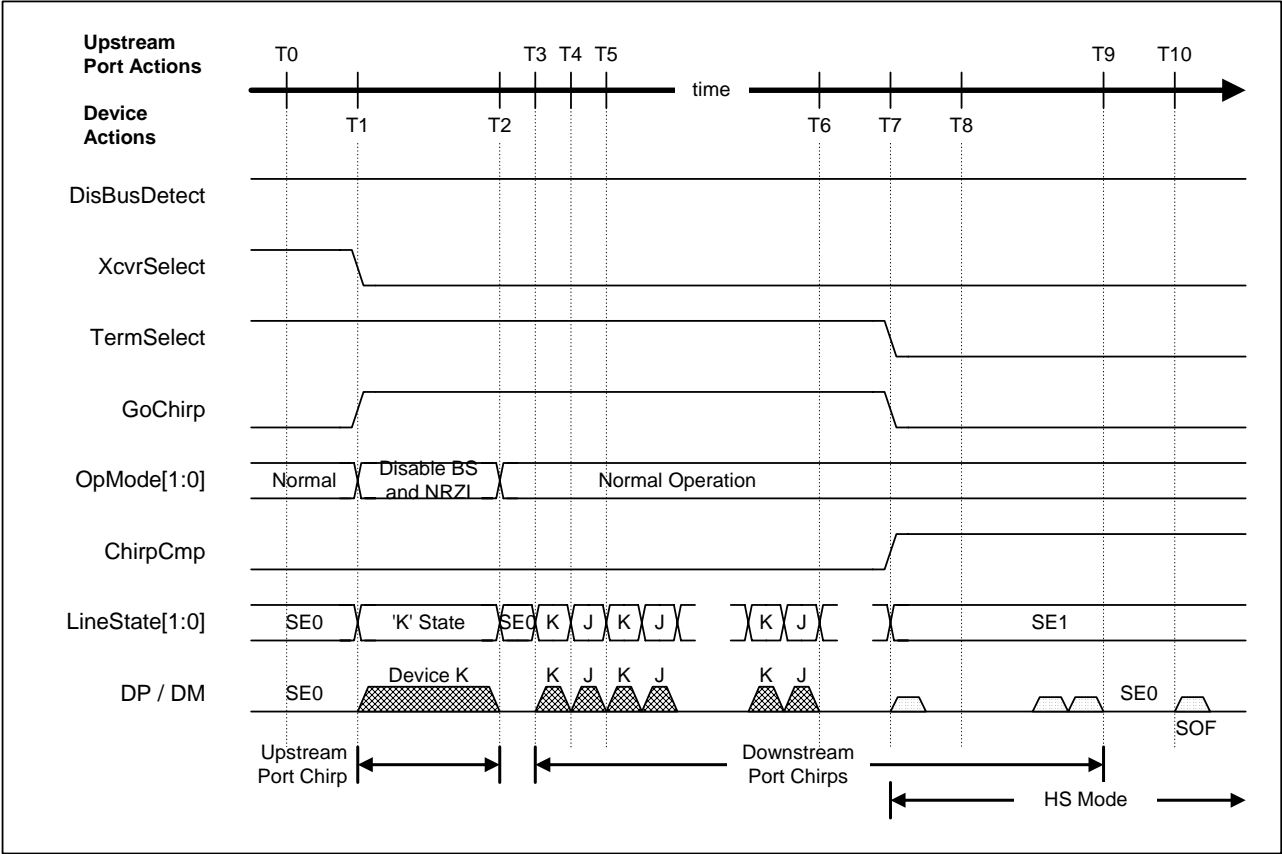
この時、D\_SIE\_IntEnb.EnChirpCmp ビットおよび DeviceIntEnb.EnD\_SIE\_IntStat ビットがセットされ、MainIntEnb.EnDeviceIntStat がセットされている場合には、同時に XINT 信号がアサートされますので、HS Detection Handshake が終了したと判断してください。

このホスト downstream port からの Chirp K、Chirp J はバス・アクティビティとして認識し、USB のサスペンドステートと判断しないようにしなければなりません。そこで、HS モードでは、この Chirp K、Chirp J を逐次検出し、内部の Suspend Timer に取り込んでいます。

なお、Chirp K-J-K-J-K-J を検出する為に、USB\_Status.LineState[1:0] ビットを使用しています。通常の HS パケットと違い、Chirp K、Chirp J は非常に遅い為、D\_USB\_Status.LineState[1:0] ビットを使用できません。しかし、本来のパケット受信時に D\_USB\_Status.LineState[1:0] ビットにバスの信号を載せると非常にノイジーな為、D\_XcvrControl.TermSelect ビットが HS モードの時、バス・アクティビティがあると判断される場合には、D\_USB\_Status.LineState[1:0] ビットは "J" を、バス・アクティビティが無いと判断される場合には "SE0" を出力します。

次図で、T6 の時点から Chirp の高さが変わっているのは、D\_XcvrControl.TermSelect ビットによりデバイス側の HS ターミネーションが有効になったことを表しています。通常、D\_XcvrControl.TermSelect が FS モード時の Chirp は約 800mV、D\_XcvrControl.TermSelect ビットが HS モード時の Chirp (HS の通常送受信パケットも同様) では、約 400mV となります。

# 1. 機能説明



**図 1-13 HS Detection Handshake Timing (HS mode)**

表 1-15 HS Detection Handshake Timing Values (HS mode)

Timing Parameter	Description	Value
T0	HS Detection Handshake 開始。	0 (reference)
T1	HS トランシーバをイネーブルにし、GoChirp を"1"にセットして、Chirp K を送出開始。	$T0 < T1 < HS\ Reset\ T0 + 6.0ms$
T2	Chirp K 送出終了。最小 1ms の間は送出しなければならない。	$T1 + 1.0ms \{T_{UCH}\} < T2 < HS\ Reset\ T0 + 7.0ms \{T_{UCHEND}\}$
T3	ホスト downstream port が最初の Chirp K をバスに送出。	$T2 < T3 < T2 + 100us \{T_{WTDCH}\}$
T4	ホスト downstream port が Chirp K から Chirp J に切り替え送出。	$T3 + 40us \{T_{DCHBIT} (Min)\} < T4 < T3 + 60us \{T_{DCHBIT} (Max)\}$
T5	ホスト downstream port が Chirp J から Chirp K に切り替え送出。	$T4 + 40us \{T_{DCHBIT} (Min)\} < T5 < T4 + 60us \{T_{DCHBIT} (Max)\}$
T6	Chirp K-J-K-J-K-J を検出。	T6
T7	Chirp K-J-K-J-K-J を検出したことを受けて、FS ターミネーションを無効に、HS ターミネーションを有効にする。ChirpCmp が"1"にセットされる。さらに、リセットの終了を待つ。	$T6 < T7 < T6 + 500us$
T8	Chirp K、Chirp J によりバス・アクティビティと認識される。ただし SYNC が検出できない為、パケット受信中和認識されることは無い。	T8
T9	ホスト downstream port からの Chirp K、Chirp J の送出終了。	$T10 - 500us \{T_{DCHSE0} (Max)\} < T9 < T10 - 100us \{T_{DCHSE0} (Min)\}$
T10	リセットシーケンスの終了。	$HS\ Reset\ T0 + 10ms \{T_{DRST} (Min)\}$

注： {} は、USB2.0 規格書で規格されている名称である。

注： 最小 1ms の Chirp K を生成するために、66000 サイクル(内部クロック：60MHz)で判断する。

# 1. 機能説明

## 1.2.7.11.5.3. スヌーズ中にリセットされた場合

本 LSI は、スヌーズ状態では PLL による内部システムクロックは出力されていません。ここでは、スヌーズ状態からの復帰について述べます。

スヌーズ状態で、リセットが検出された場合(T0)、D\_SIE\_IntStat.NonJ ビットがセットされます。さらに、D\_SIE\_IntEnb.EnNonJ ビットおよび DeviceIntEnb.EnD\_SIE\_IntStat ビットがセットされ、MainIntEnb.EnDeviceIntStat ビットがセットされている場合には、同時に XINT 信号がアサートされます。この時、すぐさまスヌーズから復帰しリセットシーケンスに移行させる為に、f/w により、PM\_Control\_0.GoActDevice ビットまたは PM\_Control\_0.GoActAllDev ビットを"1"にセットしてください(T1)。PLL パワーアップ時間経過後(T2)、PM\_Control\_1.PM\_State[3:0] が "ACT\_ALL" または "ACT\_DEVICE"になり、内部クロックが出力され始めます。この後 HS Detection Handshake(前述)を行います。

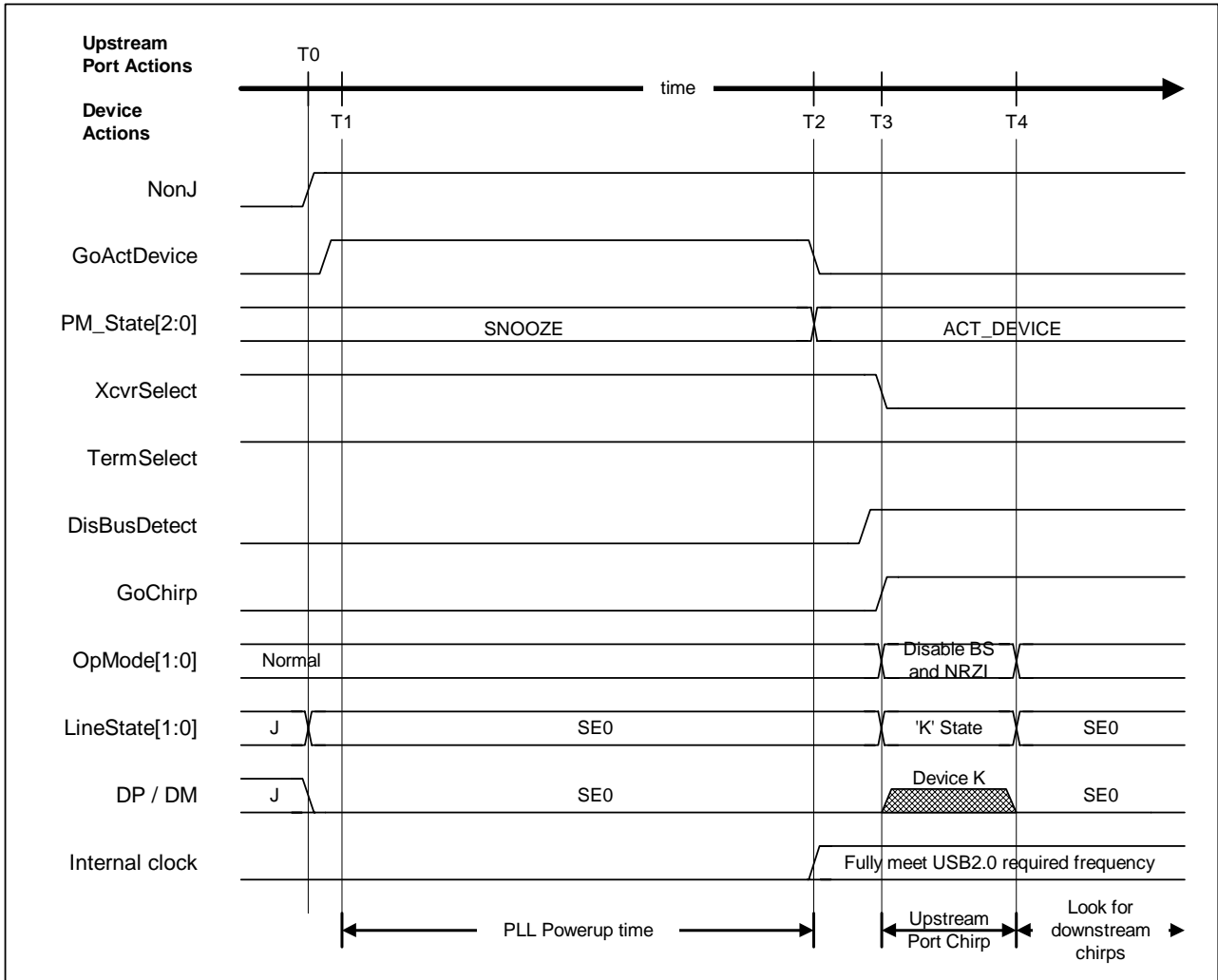


図 1-14 HS Detection Handshake Timing from Suspend



表 1-16 HS Detection Handshake Timing Values from Suspend

Timing Parameter	Description	Value
T0	NonJ が"1"にセットされ、LineState[1:0]で"SE0"を確認すると、スヌーズ中のリセットを検出。	0 (HS Reset T0)
T1	リセット検出後、GoActDevice を"1"にセット。	T1
T2	PM_State が"ACT_DEVICE"になる。内部クロック出力安定。	$T1 + 250\mu s < T2$
T3	GoChirp を"1"にセットし、chirp K をバスに送出。 (chirp K 送出前には DisBusDetect を"1"にセットする)	$T2 < T3 < \text{HS Reset T0} + 5.8\text{ms}$
T4	chirp K 送出を終了。	$T3 + 1.0\text{ms} \{T_{\text{UCH}}\} < T4 < \text{HS Reset T0} + 7.0\text{ms} \{T_{\text{UCHEND}}\}$

注： {} は、USB2.0 規格書で規格されている名称である。

注： 最小 1ms の Chirp K を生成するために、66000 サイクル(内部クロック：60MHz)で判断する。

注： 発振回路も停止していた場合(スリープ状態)は、後述する(PLL パワーアップ時間の他に、OSC パワーアップ時間が必要)。

## 1. 機能説明

---

### 1.2.7.11.6. レジュームの発行

ここでは、リモート・ウェイクアップがサポートされていて、かつホストからこのリモート・ウェイクアップを有効にされている時に、何らかの要因で、自らレジュームする方法を説明します。ただしリモート・ウェイクアップを行うことが可能なのは、バスがアイドルになってから少なくとも 5ms 経過してからでなくてはなりません。さらに、レジューム信号を出力してから 10ms 経過以前は、USB のサスペンド状態に入る前の電流を VBUS から消費することはできません。

デバイスは、リモート・ウェイクアップする為に、まずスリープ/スヌーズから復帰します。D\_SIE\_IntEnb.EnNonJ ビットをクリアし、PM\_Control\_0.GoActDevice ビット、または、PM\_Control\_0.GoActAllDev ビットをセットし(T0)、PLL パワーアップ時間経過後(T1)、PM\_Control\_1.PM\_State[3:0]ビットが"ACT\_DEVICE"または"ACT\_ALL"になると同時に内部クロックが出力され始めます。

その後、D\_NegoControl.SendWakeup ビットをセットし、レジューム信号を送出します(T2)。この時、内部では、D\_XcvtControl.OpMode[1:0]を"Disable Bit Stuffing and NRZI encoding"に設定し、送信データとして"0"を準備し、パケット送信状態にして、"K"(Resume 信号)を送出します。ホスト downstream port は、このレジューム信号を検出し、バス上に"K"(レジューム信号)を返してきます(T3)。

レジューム信号を送出し始めてから約 1ms 後、D\_NegoControl.SendWakeup ビットをクリアすることによってバスに送出していたレジューム信号が停止されますが(T4)、この時点ではホスト downstream port が依然バスをレジューム信号にホールドしています。

そこで、D\_NegoControl.RestoreUSB ビットをセットします。一定時間経過後、ホスト downstream port はレジューム信号の送出を停止し(T5)、2 ビットの LS-EOP(2\*SE0)を送出し、USB のサスペンド以前のスピードモードに切り替わります。これを検出した("K"でなくなった)ところで、D\_XcvtControl.XcvtSelect、D\_XcvtControl.TermSelect の両ビットが所望のモード(今回の場合 HS モード)に切り替えられ、D\_NegoControl.RestoreUSB ビットがクリアされるとともに D\_SIE\_IntStat.RestoreCmp ビットがセットされます。この時、D\_SIE\_IntEnb.EnRestoreCmp ビットおよび DeviceIntEnb.EnD\_SIE\_IntStat ビットがセットされ、MainIntEnb.EnDeviceIntStat ビットがセットされている場合には、同時に XINT 信号がアサートされます。

ここで、USB のサスペンド開始時には、スピードモード(HS 又は FS)を、USB\_Status.FSxHS ビットにて保存しており、レジュームにて復帰する場合には、この D\_USB\_Status.FSxHS ビットが示すモードに戻ります。この時、レジュームごとに HS Detection Handshake は行いません。ここでは USB のサスペンド以前のモードが HS モードであった場合についてのみ説明していることに注意してください。FS モードであった場合には、T5 以降が通常の FS モードとなり、特に大きなシーケンスの違いはありません。

本 LSI が、スヌーズ状態(PM\_Control\_1.PM\_State[3:0]ビットが"SNOOZE")では、内部クロックは出力されていません。ここでは、発振回路は動作しているものとして(スリープ状態ではなく、スヌーズ状態)、その時の動作を説明しています。

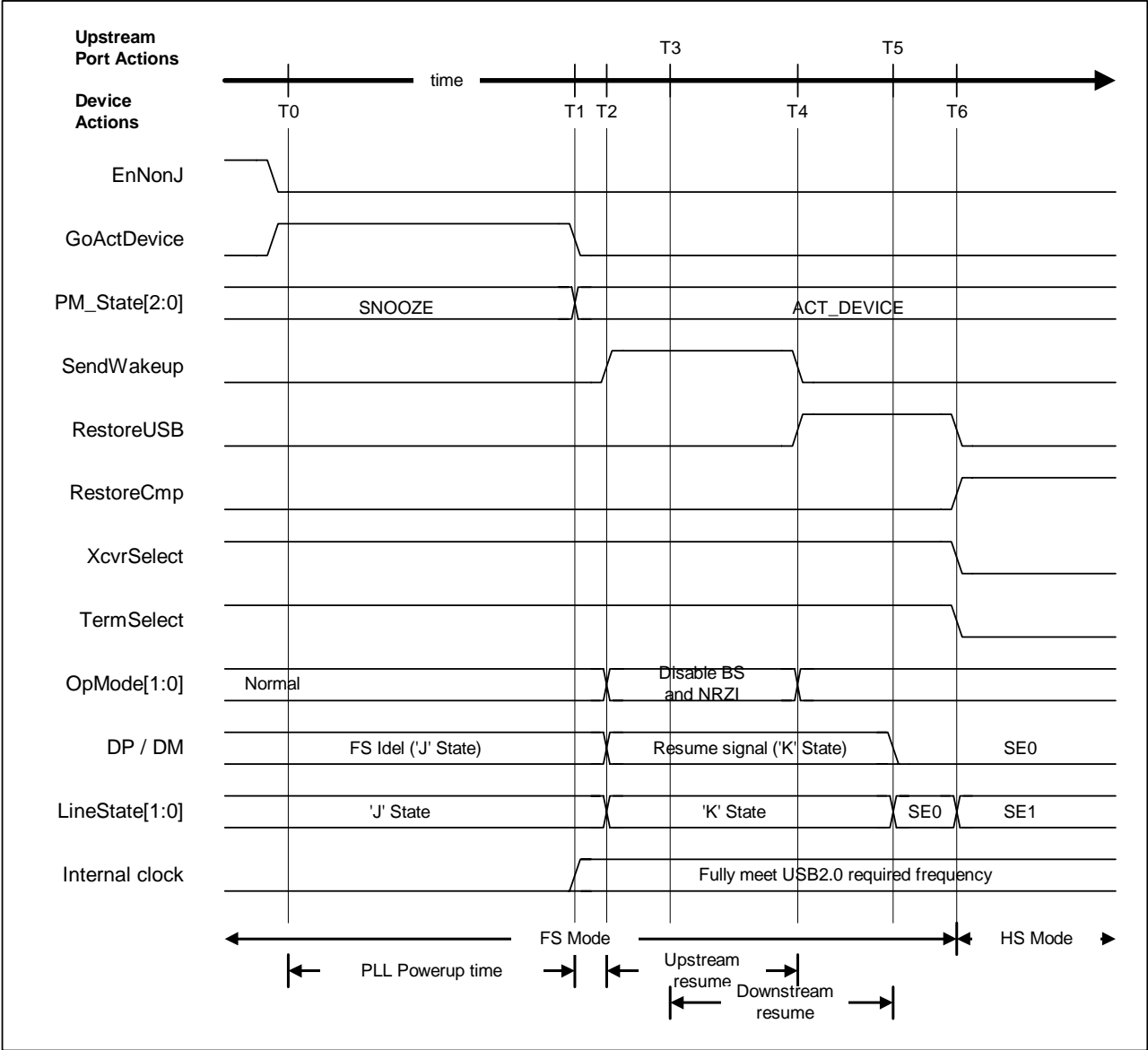


図 1-15 Assert Resume Timing (HS mode)

## 1. 機能説明

表 1-17 Assert Resume Timing Values (HS mode)

Timing Parameter	Description	Value
T0	レジューム開始。GoActDevice を"1"にセット。 (レジューム開始前に EnNonJ を"0"にクリアすること)	0 (reference)
T1	PM_State が"ACT_DEVICE"になる。内部クロック出力の安定。	$T0 + 250\mu s < T1$
T2	SendWakeup を"1"にセットし、FS の"K"を送出開始。ここで、10ms 以内は USB のサスペンド以前の電流を引っ張ってはいけない。	$T0 < T2 < T0 + 10\text{ms}$
T3	ホスト downstream port が FS の"K"を返す。	$T2 < T3 < T2 + 1.0\text{ms}$
T4	SendWakeup を"0"にクリアし、FS の"K"送出を終了。LineState[1:0]により"K"を確認後、RestoreUSB を"1"にセットする。	$T2 + 1.0\text{ms} \{T_{\text{DRSMUP}}(\text{Min})\} < T4 < T2 + 15\text{ms} \{T_{\text{DRSMUP}}(\text{Max})\}$
T5	ホスト downstream port が FS の"K"送出を終了。	$T2 + 20\text{ms} \{T_{\text{DRSMDN}}\}$
T6	RestoreCmp が"1"になる。USB のサスペンド以前が HS モードであった場合、自動的に HS モードに移行。	$T5 + 1.33\mu s \{2 \text{ Low-speed bit times}\}$

注： {} は、USB2.0 規格書で規格されている名称である。

## 1.2.7.11.7. レジュームの検出

本 LSI がサスペンドしている時、バス上には"J"(D\_USB\_Status.LineState[1:0]は"J")が観測されます。バス上に"K"が観測された時は、ホスト downstream port からのウェイクアップの指示(レジューム指示)を受け取ったことになります(T0)。この時、SIE\_IntStat.NonJ ビットがセットされます。この時、D\_SIE\_IntEnb.EnNonJ ビットおよび DeviceIntEnb.EnD\_SIE\_IntStat ビットがセットされ、MainIntEnb.EnDeviceIntStat ビットがセットされている場合には、同時に XINT 信号がアサートされます。

まず、PM\_Control\_0.GoActDevice ビット、または、PM\_Control\_0.GoActAllDev ビットを"1"にセットし(T1)、PLL パワーアップ時間経過後(T2)、PM\_Control\_1.PM\_State[3:0]が"ACT\_DEVICE"または"ACT\_ALL"になると同時に内部クロックが出力され始めます。

そこで、D\_NegoControl.RestoreUSB をセットします。一定時間経過後、ホスト downstream port はレジューム信号の送出を停止し(T3)、USB のサスペンド以前のスピードモードに切り替わります。これを検出した("K"でなくなった)ところで、D\_XcvtControl.XcvtSelect、D\_XcvtControl.TermSelect の両ビットが所望のモード(今回の場合 HS モード)に切り替えられ、D\_NegoControl.RestoreUSB ビットがクリアされるとともに D\_SIE\_IntStat.RestoreCmp ビットがセットされます。この時、D\_SIE\_IntEnb.EnRestoreCmp ビットおよび DeviceIntEnb.EnD\_SIE\_IntStat ビットがセットされ、MainIntEnb.EnDeviceIntStat ビットがセットされている場合には、同時に XINT 信号がアサートされます。

ここでは、発振回路は動作しているものとして(スリープ状態ではなく、スヌーズ状態)、その時の動作を説明しています。

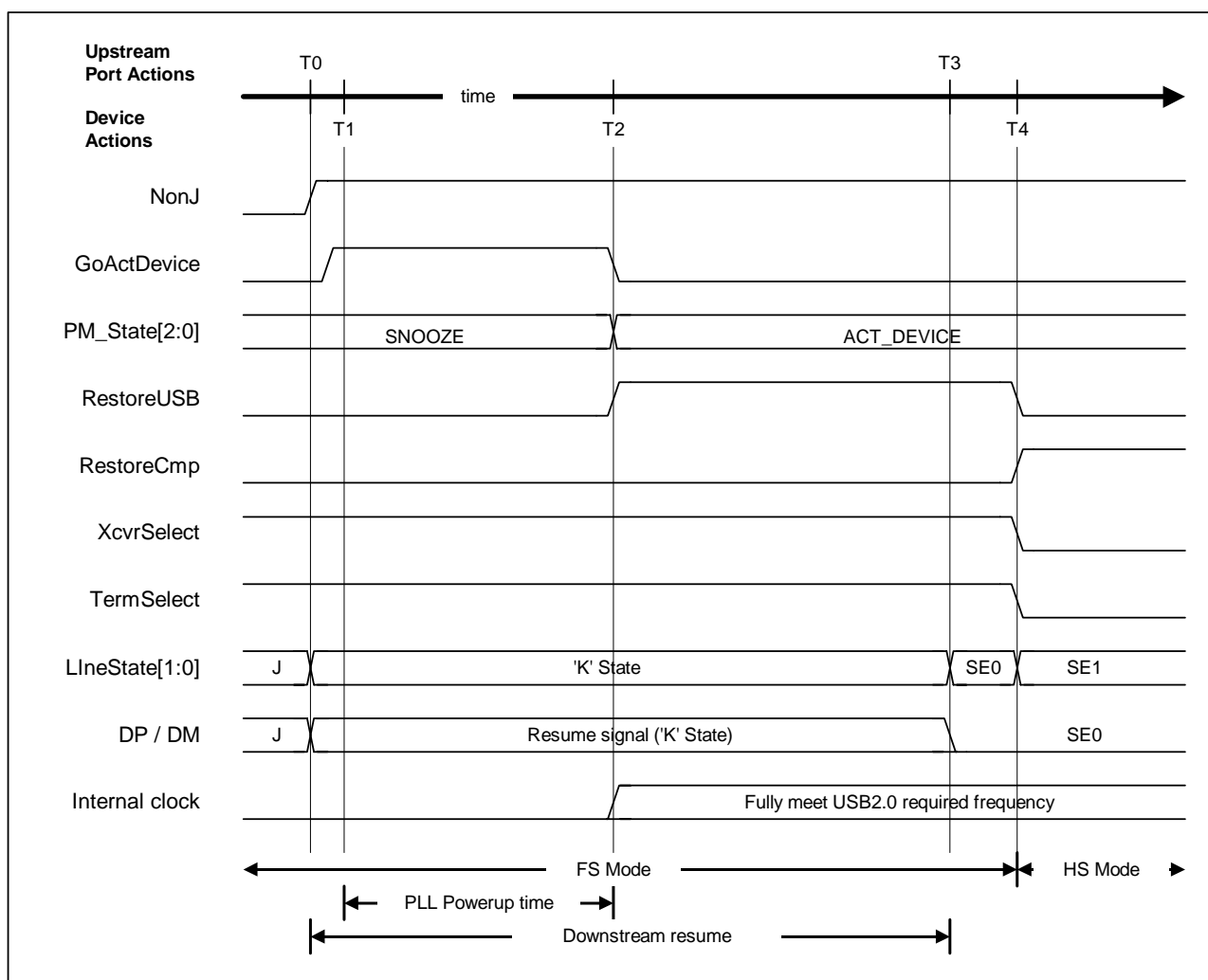


図 1-16 Detect Resume Timing (HS mode)

## 1. 機能説明

---

表 1-18 Detect Resume Timing Values (HS mode)

Timing Parameter	Description	Value
T0	ホスト downstream port が FS の”K”を送出。NonJ が”1”になる。	0 (reference)
T1	GoActDevice を”1”にセット。	T1
T2	PM_State が”ACT_DEVICE”になる。内部クロック出力安定。 LineState[1:0]で”K”を確認後、RestoreUSB を”1”にセットする。	$T1 + 250\mu s < T2$
T3	ホスト downstream port が FS の”K”送出生了。同時にホスト downstream port は USB のサスペンド以前の HS モードに移行。	$T2 + 20\text{ms} \{T_{\text{DRSMDN}}\}$
T4	USB のサスペンド以前が HS モードであった場合、自動的に HS モードに移行。	$T5 + 1.33\mu s \{2 \text{ Low-speed bit times}\}$

注： {} は、USB2.0 規格書で規格されている名称である。

## 1.2.7.11.8. ケーブル挿入

ここでは、ハブ又はホストに接続された場合、すなわちケーブルが挿入された場合を説明します。

ケーブルが抜かれている状態、もしくは繋がっていない状態に故意にしている時には、D\_XcvrControl.XcvrSelect ビットは FS モード、D\_XcvrControl.TermSelect ビットは HS モードを初期値としてください。

ケーブルが接続されていない状態(T0)でケーブルが接続されると、VBUS が"H"になり、同時に D\_USB\_Status.VBUS ビットがセットされます(T1)。この時、例えばスヌーズ状態にしていた場合は、PM\_Control\_0.GoActDevice ビット、または、PM\_Control\_0.GoActAllDev ビットを"1"にセットし(T2)、PLL パワーアップ時間経過後(T3)、PM\_Control\_1.PM\_State[3:0]が"ACT\_DEVICE"または"ACT\_ALL"になると同時に内部クロックが出力され始めます。その後、まずは FS デバイスが接続されたことにしなければならぬため、一旦は FS モードになるために、D\_XcvrControl.TermSelect ビットを FS モードにしてください(T4)。

その後、ホスト downstream port はリセットを送出し(T5)、ここから HS Detection Handshake が開始されます。

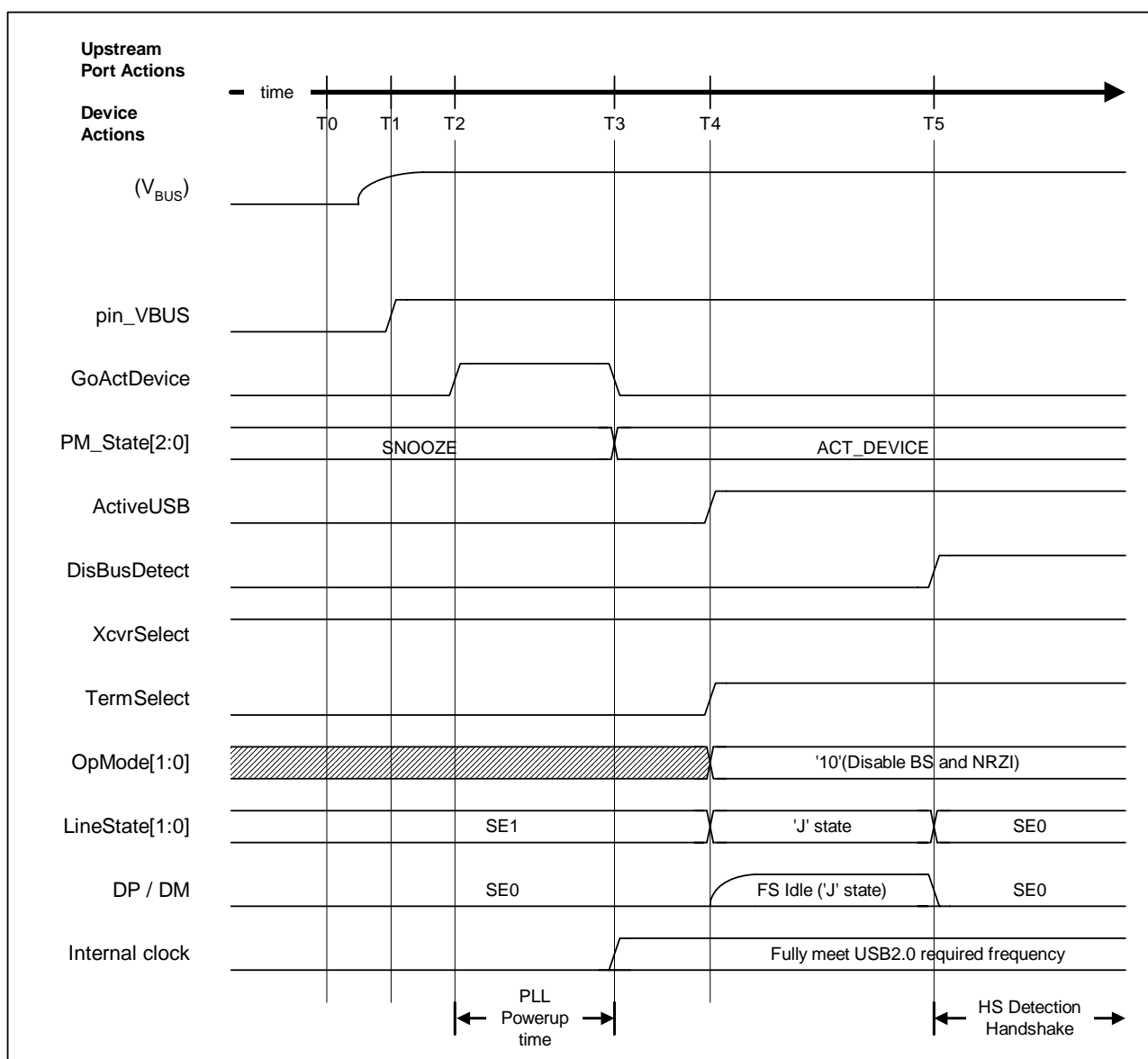


図 1-17 Device Attach Timing

## 1. 機能説明

---

表 1-19 Device Attach Timing Values

Timing Parameter	Description	Value
T0	ケーブルは挿入されていない。	0 (reference)
T1	ケーブルが挿入され、入力ピン VBUS_B が H になる。	T1
T2	GoActDevice を"1"にセット。	T2
T3	PM_State が"ACT_DEVICE"になる。内部クロック出力の安定。	$T2 + 250\mu s < T3$
T4	ActiveUSB を"1"にセット。TermSelect を"1"にセット。 OpMode[1:0]を"00"に設定。 FS モードに移行。FS ターミネーションが有効。	$T1 + 100ms \{T_{SIGATT}\} < T4$
T5	ホスト downstream port からリセットが送出される。 DisBusDetect を"1"にセット。	$T4 + 100ms \{T_{ATTDB}\} < T5$

注： {} は、USB2.0 規格書で規格されている名称である。



### 1.3. USB ホスト制御

#### 1.3.1. チャンネル

##### 1.3.1.1. チャンネル概要

本 LSI では、パイプと 1 対 1 で対応するホスト側のバッファと、そのバッファを介して行われる転送の為の設定レジスタとを、まとめてチャンネルと呼びます。

チャンネルには、IRP(I/O Request Packet)単位で転送情報を設定します。またチャンネルのバッファとして FIFO 領域を割り当てます。

チャンネルは設定された情報に基づき、IRP を自動的に複数のトランザクションに分割して実行します。チャンネルは IRP 単位で設定を切り替えることができますので、1 本のチャンネルで複数のエンドポイントに対応することが可能です。

図 1-18 にチャンネルの概念を示します。

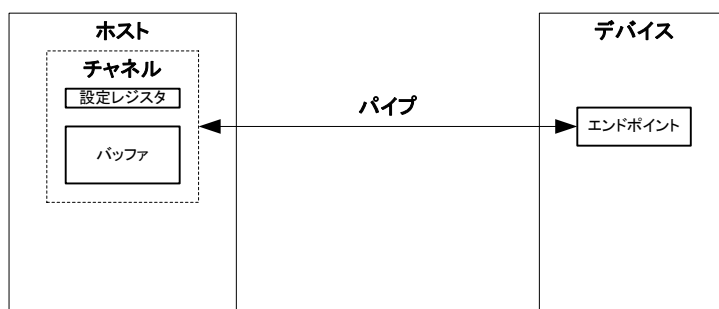


図 1-18 チャンネル概念図

ファームウェアは、バッファおよび転送情報の設定を行ったうえで転送実行(H\_CHx{x=0,a-e}Config\_0.TranGo)をセットします。転送実行をセットした後は、ファームウェアは IRP データ数の処理が完了するまで、バッファにデータを書き込む(OUT 転送時)、バッファからデータを読み出す(IN 転送時)といった処理を行います。

一方、ハードウェア(チャンネル)は IRP を自動的に複数のトランザクションに分割して実行します。転送が完了すると、割り込みによってファームウェアに通知します。

図 1-19 に転送を行う際の基本的な流れを示します。

# 1. 機能説明

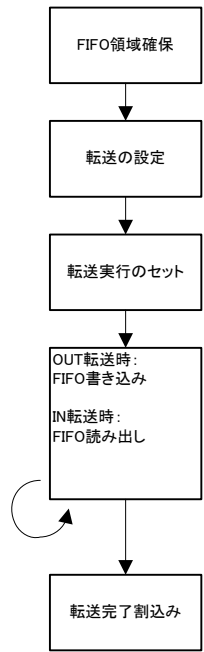


図 1-19 チャンネルの基本的な転送の手順

本 LSI はコントロール転送のみを行うチャンネル(CH0)、バルク転送のみを行うチャンネル(CHa)、バルク / インタラプト / アイソクロナス転送を行うチャンネル(CHb, CHc, CHd, CHe)の計 6 本のチャンネルを持ちます。ここでは、チャンネル CH0 をコントロール専用チャンネル、チャンネル CHa, CHb, CHc, CHd, CHe を汎用チャンネルと呼びます。

各チャンネルには、USB 定義インタフェースによって決定される固定の基本設定項目と、転送毎に制御を行う可変の制御項目及びステータスがあります。基本設定項目は、チップ初期化時、または、USB 定義インタフェースの切り替え時等に設定して下さい。

なお、同時に設定できるインタラプト転送とアイソクロナス転送の最大数は合わせて 4 本です。

各チャンネルが対応する転送タイプを表 1-20 に示します。

表 1-20 対応する転送タイプ

チャンネル	対応する転送タイプ	備考
CH0	コントロール転送	コントロール転送サポート 1 機能(1.3.4.3)使用可
CHa	バルク転送	バルクオンリーサポート機能(1.3.8)使用可
CHb, CHc CHd, CHe	バルク転送 インタラプト転送 アイソクロナス転送	オーディオクラスアシスト機能(1.3.9)使用可

## 1.3.1.2. コントロール専用チャネル

本 LSI では、コントロール専用チャネル(CH0)にてコントロール転送を行います。その為、複数のエンドポイントに対してコントロール転送を行う場合はチャネル CH0 を時分割して使用します。

表 1-21 にコントロール専用チャネル(CH0)の基本設定項目を示します。

表 1-21 コントロール専用チャネルの基本設定項目

項目	レジスタ/ビット	説明
転送速度	H_CH0Config_0.SpeedMode	チャネルCH0に対応するエンドポイントの転送速度(HS/FS/LS)を設定します。
トグルシーケンスビット	H_CH0Config_0.Toggle	トランザクションを開始する時のトグルシーケンスビットの初期値を設定します。また、トランザクション実行中、およびトランザクション完了後はトグルシーケンスビットの状態を示します。
トランザクション種別	H_CH0Config_1.TID	チャネル CH0 で発行するトランザクションの種別(SETUP/IN/OUT)を設定します。
マックスパケットサイズ	H_CH0MaxPktSize	マックスパケットサイズを、LS 動作時には 8 に、FS 動作時には 8,16,32,64 のいずれかの値に設定します。また、HS 動作時には 64 に設定します。
USB アドレス	H_CH0FuncAdrs.FuncAdrs	チャネルCH0で実行するIRPのあて先となるUSBアドレスを設定します。
エンドポイントナンバー	H_CH0FuncAdrs.EP_Number	チャネル CH0 で実行する IRP のあて先となるエンドポイントナンバーを設定します。
ハブアドレス	H_CH0HubAdrs.HubAdrs	スプリットトランザクションを行うハブのUSBアドレスを設定します。
ポート番号	H_CH0HubAdrs.Port	スプリットトランザクションを行うハブのポート番号を設定します。
IRP データ数	H_CH0TotalSize_H, H_CH0TotalSize_L	チャネルCH0で実行するIRPのデータ転送数をバイト単位で設定します。
FIFO 領域	AREAn{n=0-5}StartAdrs_H, AREAn{n=0-5}StartAdrs_L, AREAn{n=0-5}EndAdrs_H, AREAn{n=0-5}EndAdrs_L,	チャネル CH0 に割り当てる領域を、FIFO のアドレスで設定します。 FIFO 領域は、チャネル CH0 のマックスパケットサイズ以上の領域を割り当ててください。また、FIFO 領域のサイズがデータ転送のスループットに影響します。 FIFOの領域割り当ての詳細は、「1.6.FIFO 管理」を参照してください。
FIFO 領域ジョイン	AREAn{n=0-5}Join_1.JoinEP0CH0	チャネル CH0 を FIFO 領域に結合します。コントロール転送サポート機能を使用する場合、チャネル CH0 を AREA0 に結合してください。
セットアップデータ	H_CH0SETUP_x(x=0-7)	セットアップトランザクションで送信する 8 バイトのデータを設定します。

## 1. 機能説明

### 1.3.1.3. 汎用チャネル

汎用チャネルは、トランザクション方向と USB アドレスおよびエンドポイントナンバーを任意に設定出来ますので、同時に5本までのエンドポイントに対して1対1に対応させることが出来ます。また、コントロール専用チャネルと同様にチャネルを IRP 単位で時分割して使用することで、5本を越える数のエンドポイントとも転送を行うことが出来ます。

各チャネルには、USB 定義インタフェースによって決定される固定の基本設定項目と、転送毎に制御を行う可変の制御項目及びステータスがあります。基本設定項目は、チップ初期化時、または、USB 定義インタフェースの切り替え時等に設定して下さい。

表 1-22 に汎用チャネルの基本設定項目を示します。USB定義インタフェースの定義内容に合わせて適宜設定し、また有効にすることによって、USB定義インタフェースを構成して下さい。

表 1-22 汎用チャネルの基本設定項目

項目	レジスタ/ビット	説明
転送速度	H_CHx{x=a-e}Config_0.SpeedMode	各チャネルに対応するエンドポイントの転送速度(HS/FS/LS)を設定します。
トグルシーケンスビット	H_CHx{x=a-e}Config_0.Toggle	トランザクションを開始する時のトグルシーケンスビットの初期値を設定します。また、トランザクション実行中、およびトランザクション完了後はトグルシーケンスビットの状態を示します。
トランザクション種別	H_CHx{x=a-e}Config_1.TID	各チャネルで発行するトランザクションの種別(IN/OUT)を設定します。
転送種別	H_CHx{x=b-e}Config_1.TranType	各チャネルの転送種別(バルク、インタラプト、アイソクロナス)を設定します。
マックスパケットサイズ	H_CHx{x=a-e}MaxPktSize_H, H_CHx{x=a-e}MaxPktSize_L	各チャネルで実行するトランザクションのマックスパケットサイズを、1Byte～1024Byte の間の任意の値に設定します。
ハブアドレス	H_CHx{x=a-e}HubAdrs.HubAdrs	スプリットトランザクションを行うハブのUSBアドレスを設定します。
ポート番号	H_CHx{x=a-e}HubAdrs.Port	スプリットトランザクションを行うハブのポート番号を設定します。
USB アドレス	H_CHx{x=a-e}FuncAdrs.FuncAdrs	各チャネルで実行するIRPのあて先となるUSBアドレスを設定します。
エンドポイントナンバー	H_CHx{x=a-e}FuncAdrs.EP_Number	各チャネルで実行する IRP のあて先となるエンドポイントナンバーを設定します。
IRP データ数	H_CHx{x=a-e}TotalSize_HH, H_CHx{x=a-e}TotalSize_HL, H_CHx{x=a-e}TotalSize_LH, H_CHx{x=a-e}TotalSize_LL	各チャネルで実行するIRPのデータ転送数をバイト単位で設定します。
トークン発行間隔	H_CHx{x=b-e}Interval_H, H_CHx{x=b-e}Interval_L	インタラプト転送及びアイソクロナス転送におけるトークン発行間隔(周期)を設定します。

(続く)

表 1-23 汎用チャネルの基本設定項目(続き)

項目	レジスタ／ビット	説明
FIFO 領域	AREAn{n=0-5}StartAdrs_H, AREAn{n=0-5}StartAdrs_L, AREAn{n=0-5}EndAdrs_H, AREAn{n=0-5}EndAdrs_L,	各チャネルに割り当てる領域を、FIFO のアドレスで設定します。 FIFO 領域は、各チャネルのマックスパ ケットサイズ以上の領域を割り当てて 下さい。また、FIFO 領域のサイズがデ ータ転送のスループットに影響しま す。 FIFO の領域割り当ての詳細は、 「1.6.FIFO管理」を参照して下さい。
FIFO 領域ジョイ ン	AREAn{n=0-5}Join_1.JoinEPxCHx{x=a-e}	各チャネルを FIFO 領域に結合します。 バルクオンリーサポート機能を使用す る場合、チャネル CHa を AREA1 に結 合して下さい。

## 1. 機能説明

### 1.3.1.4. チャンネル使用例

#### 1.3.1.4.1. ストレージデバイスを1台接続した場合

USB Mass Storage Class(BulkOnly Transport Protocol)に対応したストレージデバイス(例：ハードディスク)を接続した場合のチャンネル使用例を図 1-20 に示します。このクラスに対応したデバイスでは、コントロール転送、バルクIN転送、バルクOUT転送を使用します。

コントロール転送に対してはCH0を使用します。

一方、バルク IN 転送およびバルク OUT 転送については汎用チャンネルを使用しますが、特に本 LSI のバルクオンリーサポート機能を使用する場合にはCHaを使用します。

バルクオンリーサポート機能とはUSB Mass Storage Class(BulkOnly Transport Protocol)のコマンドトランスポート(CBW)、データトランスポート、ステータストランスポート(CSW)の一連のトランスポート管理を自動で行う機能(「1.3.8.バルクオンリーサポート機能」参照)であり、CHaに実装しています。

バルクオンリーサポート機能を使用しない場合は、バルク IN 転送およびバルク OUT 転送を他の汎用チャンネルに個別(例えばCHbとCHc)に割り当てることも可能です。この場合、トランスポート管理はソフトウェアで行う必要があります。

ハードウェアは、チャンネルに対し転送のスケジューリング(「1.3.2.スケジューリング」参照)を行い、トランザクションを実行します。

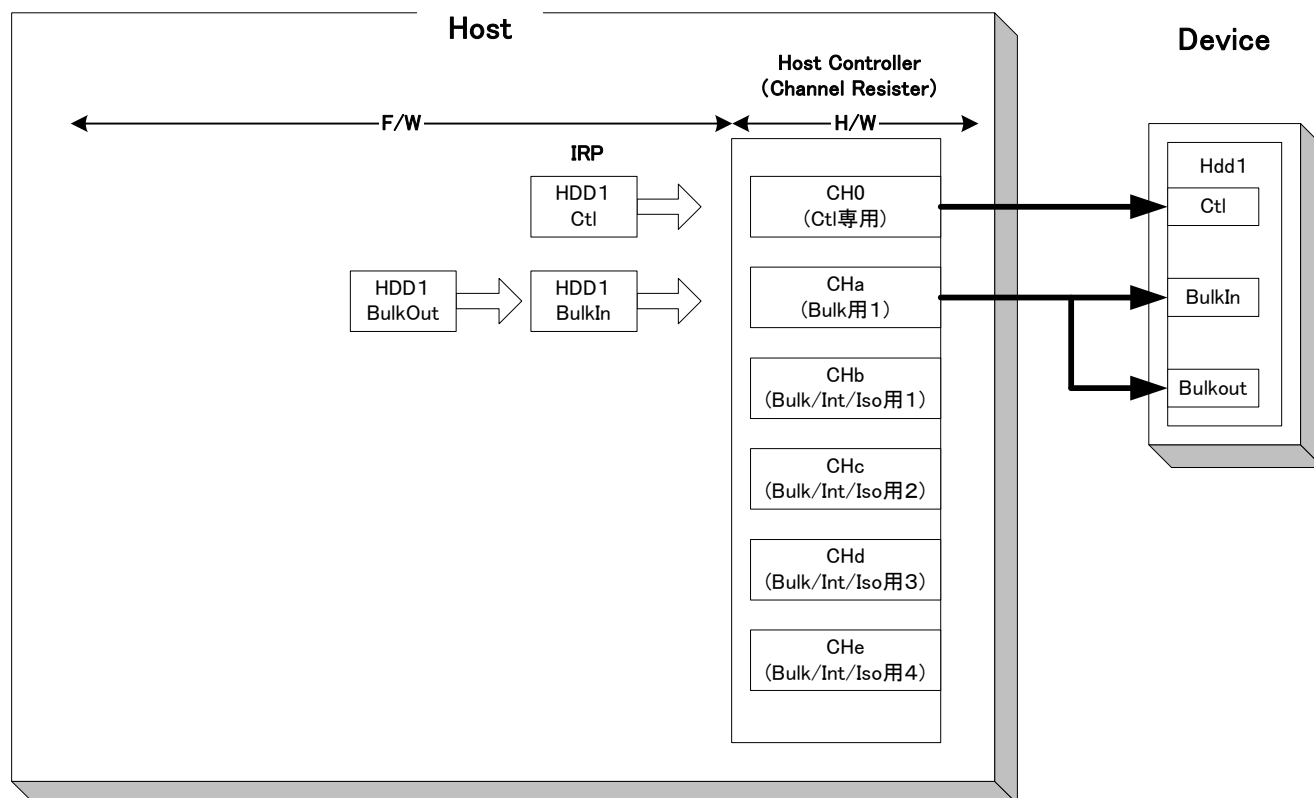


図 1-20 チャンネル使用例(ストレージデバイスを1台接続した場合)

#### 1.3.1.4.2. コミュニケーションデバイスを1台接続した場合

USB Communication Device Classに対応したコミュニケーションデバイス(例: 無線LANアダプタ)を接続した場合のチャンネル使用例を図 1-21 に示します。このクラスに対応したデバイスではコントロール転送、バルクIN転送、バルクOUT転送、インタラプトIN転送を使用します。

コントロール転送にはCH0を使用します。

このクラスではバルクIN転送とバルクOUT転送は並列に動作する必要があるため、それぞれの転送に汎用チャンネル(例えばCHb, CHc)を個別に割り当てて使用します。

インタラプトIN転送も同様に汎用チャンネル(例えばCHd)を個別に割り当てて使用します。

ハードウェアは、チャンネルに対し転送のスケジューリング(「1.3.2.スケジューリング」参照)を行い、トランザクションを実行します。

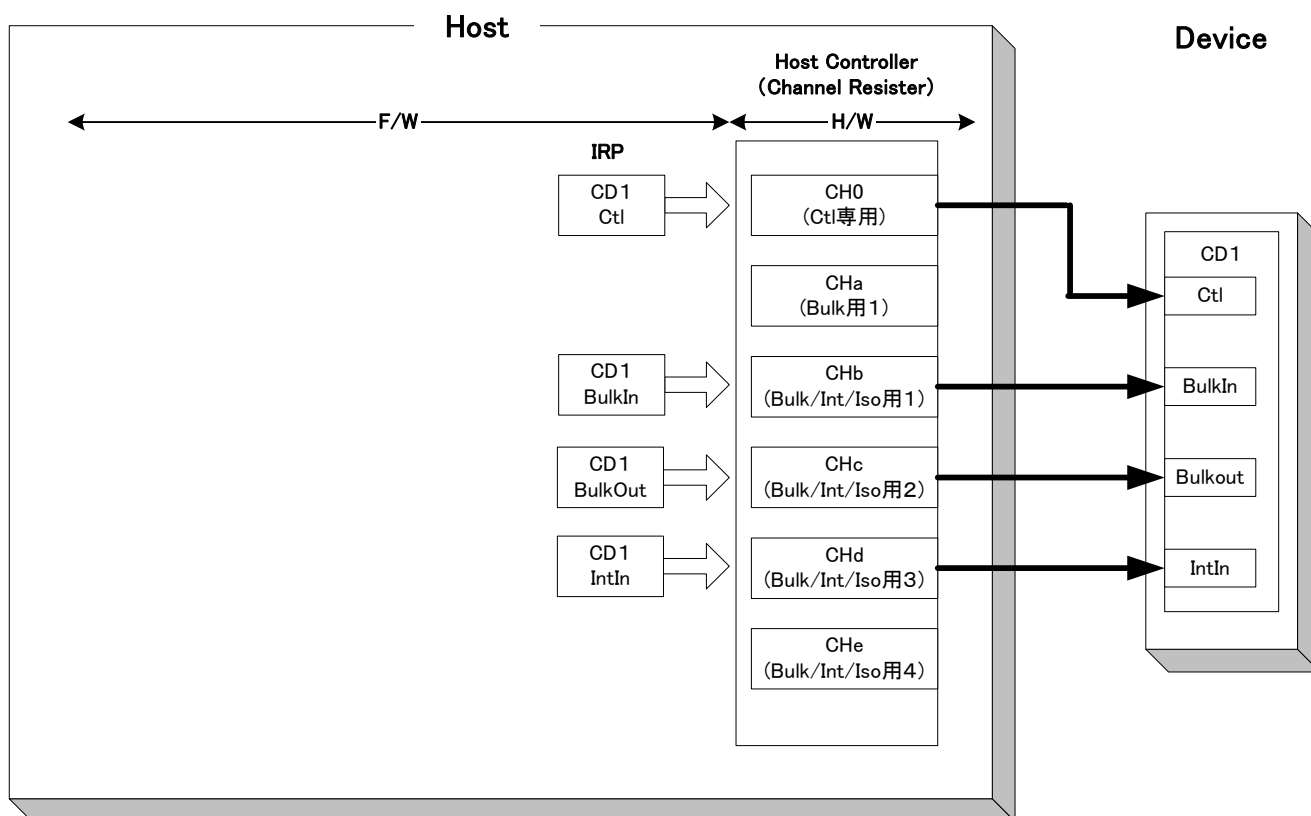


図 1-21 チャンネル使用例(コミュニケーションデバイスを1台接続した場合)

## 1. 機能説明

### 1.3.1.4.3. ヒューマンインタフェースデバイスを1台接続した場合

USB Human Interface Device Classに対応したデバイス(例：マウス)を接続した場合のチャンネル使用例を図 1-22 に示します。このクラスに対応したデバイスではコントロール転送、インタラプトIN転送を使用します。

コントロール転送にはCH0を使用します。

インタラプトIN転送には汎用チャンネル(例えばCHd)を割り当てて使用します。

ハードウェアは、チャンネルに対し転送のスケジューリング(「1.3.2.スケジューリング」参照)を行い、トランザクションを実行します。

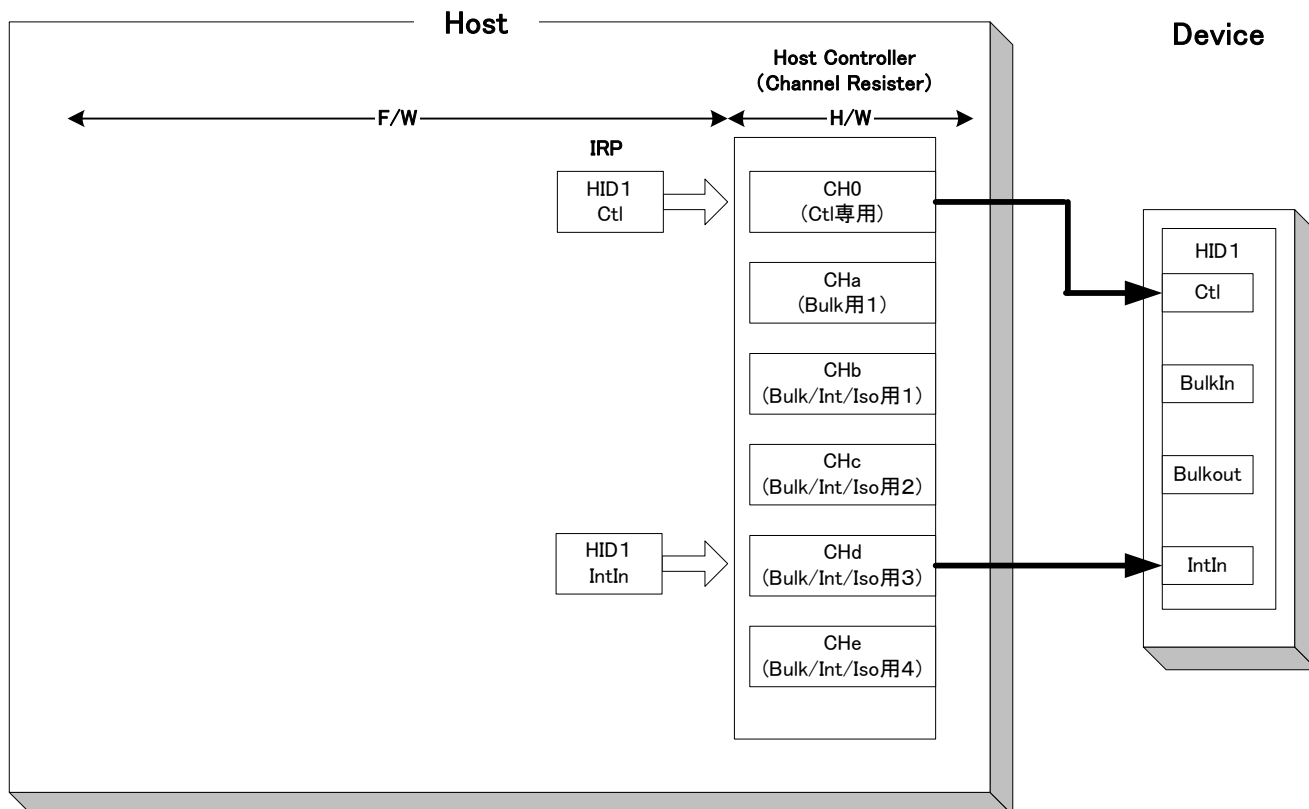


図 1-22 チャンネル使用例(ヒューマンインタフェースデバイスを1台接続した場合)



#### 1.3.1.4.4. ハブを介してストレージデバイスを2台接続した場合

USB Mass Storage Class(BulkOnly Transport Protocol)に対応したストレージデバイス(例：ハードディスク、USBメモリ)を、ハブを介して2台接続した場合のチャンネル使用例を図 1-23 に示します。USB Mass Storage ClassではバルクIN転送、バルクOUT転送を使用します。

ハブに関してはコントロール転送、インタラプト IN 転送を使用します。

USB メモリに関してはこの例ではハブを内蔵しているものを扱っています。

すべてのデバイスに対するコントロール転送は CH0 を時分割して使用します。

ハブ及び USB メモリのインタラプト IN 転送には、汎用チャンネル(例えば CHd, CHe)を個別に割り当てて使用します。

一方、ハードディスク及びUSBメモリのバルクIN転送およびバルクOUT転送に対しては、本LSIのバルクオンリーサポート機能を使用する場合はCHaを使用します。バルクオンリーサポート機能とはUSB Mass Storage Class(BulkOnly Transport Protocol)のコマンドトランスポート(CBW)、データトランスポート、ステータストランスポート(CSW)の一連のトランスポート管理を自動で行う機能(「1.3.8.バルクオンリーサポート機能」参照)であり、本LSIではCHaに実装しています。バルクオンリーサポート機能を使用する場合は転送相手(この例ではハードディスクとUSBメモリ)をコマンドトランスポートからステータストランスポートまでを単位として切り替えて使用します。

バルクオンリーサポート機能を使用しない場合は、バルク IN 転送およびバルク OUT 転送を汎用チャンネルに個別に割り当てることも可能です。この場合、トランスポート管理はソフトウェアで行う必要があります。

ハードウェアは、チャンネルに対し転送のスケジューリング(「1.3.2.スケジューリング」参照)を行い、トランザクションを実行します。

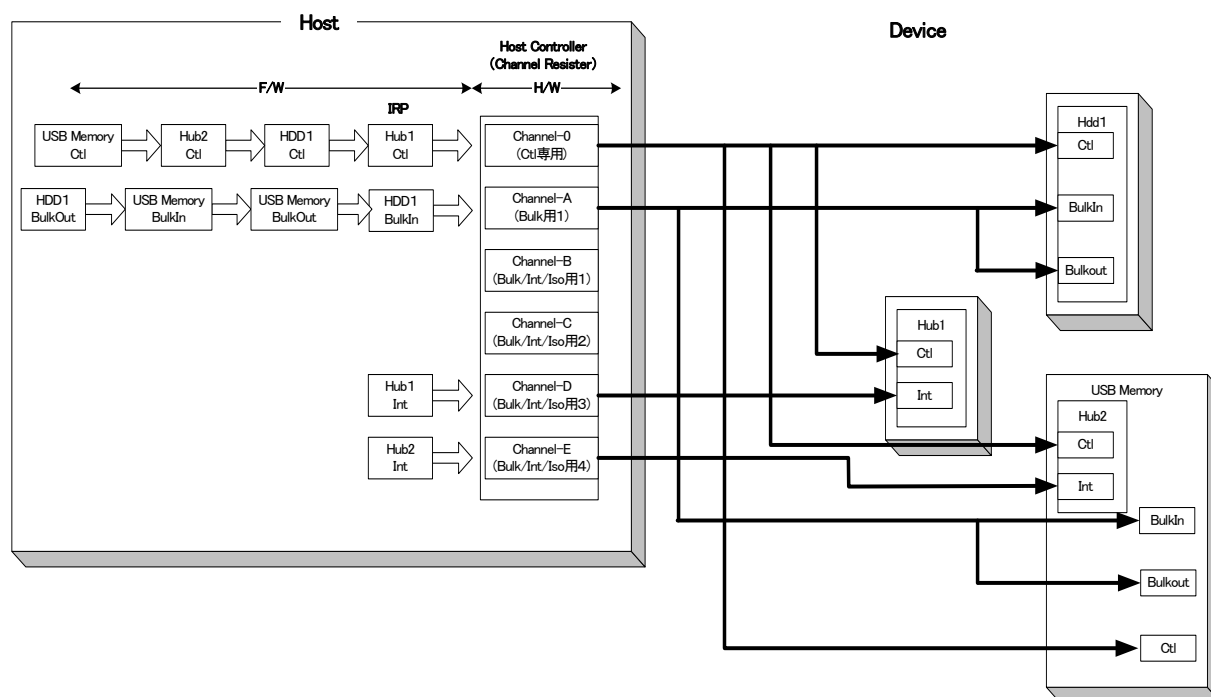


図 1-23 チャンネル使用例(ハブを介してストレージデバイスを2台接続した場合)

## 1. 機能説明

### 1.3.2. スケジューリング

ハードウェアは転送実行(H\_CHx{x=0,a-e}Config\_0.TranGo)がセットされたチャンネルの中から一つを選択し、そのチャンネルに設定された転送の実行可否判断を行います。転送が実行可能と判断すると、設定に従いトランザクションを一つ実行します。トランザクションが終了すると、ハードウェアは別のチャンネルを選択し、同様に実行可否判断を行い、トランザクションを実行します。

このようなチャンネルの選択、実行可否判断、トランザクションの実行を繰り返すことにより、ハードウェアは複数のチャンネルを使用して、複数のエンドポイントに対して転送を行います。

表 1-24 にチャンネルCH0 のスケジューリング制御に関する制御項目を示します。

表 1-24 チャンネル CH0 のスケジューリング設定項目

項目	レジスタ／ビット	説明
転送実行	H_CH0Config_0.TranGo	チャンネルCH0の転送実行を設定します。 チャンネルの設定に従い転送を行います。

表 1-25 に汎用チャンネル(CHA, CHb, CHc, CHd, CHe)のスケジューリング制御に関する制御項目を示します。

表 1-25 汎用チャンネルのスケジューリング設定項目

項目	レジスタ／ビット	説明
転送実行	H_CHx{x=a-e}Config_0.TranGo	各チャンネルの転送実行を設定します。 各チャンネルの設定に従い転送を行います。

### 1.3.3. トランザクション

本LSIは、ハードウェアでトランザクション実行機能と、ファームウェアに対するトランザクション実行のためのインタフェースを提供します。ファームウェアに対するインタフェースは、制御レジスタとステータスレジスタ、及び、ステータスによりアサートされる割り込み信号として実装されています。ステータスにより割り込みをアサートする設定については、「2.レジスタ」を参照して下さい。

ハードウェアはチャンネルを選択し、転送可否判断を行い、転送可能と判断した場合、チャンネルの設定に従ってトランザクションを実行します。

本LSIは個々のトランザクション毎に、ファームウェアに対してステータスを発行します。しかしながら、ファームウェアは必ずしも、個々のトランザクションを管理する必要はありません。

例えば、INのチャンネルであれば、ファームウェアは、CPUインタフェース(DMAリードまたはレジスタリード)により、FIFOからデータを読み出してFIFOに空き領域を作り出すことによって、INトランザクションを自動的に連続して実行させることが出来ます。また、OUTのチャンネルであれば、ファームウェアは、CPUインタフェース(DMAライトまたはレジスタライト)により、FIFOにデータを書き込んでFIFOに有効データを作り出すことによって、OUTトランザクションを自動的に連続して実行させることが出来ます。

表 1-26 にチャンネルCH0のトランザクション制御に関する制御項目及びステータスを示します。

表 1-26 チャンネル CH0 の制御項目及びステータス

項目	レジスタ/ビット	説明
トランザクションステータス	H_CH0IntStat.TotalSizeCmp, H_CH0IntStat.TranACK, H_CH0IntStat.TranErr, H_CH0IntStat.ChangeCondition	トランザクションの結果を示します。
トランザクションコンディションコード	H_CH0ConditionCode	トランザクション結果の詳細を示します。

表 1-27 に、汎用チャンネル(CHa, CHb, CHc, CHd, CHe)のトランザクション処理に関する制御項目とステータスを示します。

表 1-27 汎用チャンネルの制御項目とステータス

項目	レジスタ/ビット	説明
トランザクションステータス	H_CHx{x=a-e}IntStat.TotalSizeCmp, H_CHx{x=a-e}IntStat.TranACK, H_CHx{x=a-e}IntStat.TranErr, H_CHx{x=a-e}IntStat.ChangeCondition	トランザクションの結果を示します。
トランザクションコンディションコード	H_CHx{x=a-e}ConditionCode	トランザクション結果の詳細を示します。

## 1. 機能説明

### 1.3.3.1.SETUP トランザクション

CH0 の基本設定レジスタにおいて、トランザクション種別(H\_CH0Config\_1.TID)を"SETUP"に設定します。その他の基本設定項目を適宜設定し、H\_CH0SETUP\_0~7 レジスタにセットアップデータ(8Byte)を書き込み、転送実行(H\_CH0Config\_0.TranGo)をセットすることにより、チャンネル CH0 が USB 転送スケジューリングの対象となります。スケジューリングにより該当チャンネルが選択されると、フレーム残り時間を判断し、SETUP トランザクションが実行されます。

SETUP トランザクションでは H\_CH0SETUP\_0~7 レジスタのデータが用いられ、データパケットのデータ長は 8Byte となります。

SETUP トランザクションに対して ACK を受信すると、ファームウェアに対し、ACK ステータス(H\_CH0IntStat.TranACK ビット)を発行します。

SETUP トランザクションに対して正常な応答を受信しない場合、コンディションコード(H\_CH0ConditionCode)を"RetryError"に設定し、ファームウェアに対し TranErr ステータス(H\_CH0IntStat.TranErr ビット)を発行します。そしてリトライ処理を行います。エラーが 3 回連続して続く場合、H\_CH0Control.TranGo を自動クリアすることで転送を終了し、ファームウェアに対し ChangeCondition ステータス(H\_CH0IntStat.ChangeCondition ビット)を発行します。

図 1-24 にSETUPトランザクションの様子を図示します。(a)LSIが、宛先ノードに存在するエンドポイント 0 に宛てたSETUPトークンを発行します。(b)LSIは続けて、8Byte長のデータパケットを送信します。(c)LSIはACK受信により、該当するレジスタを自動設定し、ファームウェアに対しステータスを発行します。

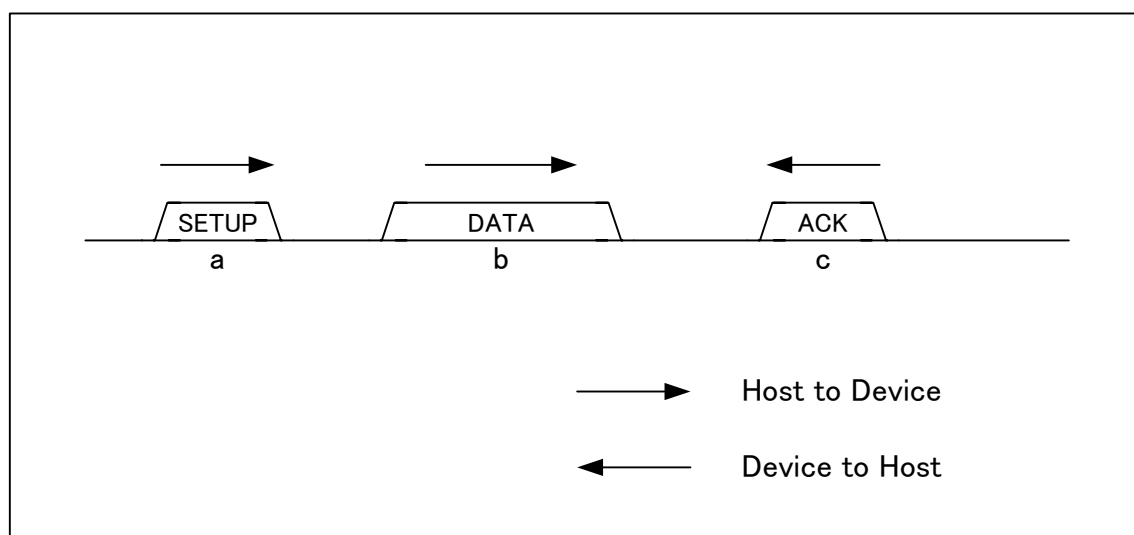


図 1-24 SETUP トランザクション

### 1.3.3.2. バルク OUT トランザクション

CHx{x=a-e}の基本設定レジスタにおいて、転送種別(H\_CHx{x=b-e}Config\_1.TranType)を"Bulk"、トランザクション種別(H\_CHx{x=a-e}Config\_1.TID)を"OUT"に設定します。その他の基本設定項目を適宜設定し、転送実行(H\_CHx{x=a-e}Config\_0.TranGo)をセットすることにより、チャンネルはハードウェアが行う USB 転送のスケジューリングの対象となります。スケジューリングにより該当チャンネルが選択されると、FIFO の有効データ数及びフレーム残り時間を判断し、バルク OUT トランザクションが実行されます。なお、チャンネル CHa はバルク転送専用となっている為に基本設定レジスタのうち転送種別は存在しません。

個々のデータパケットのデータ長は H\_CHx{x=a-e}MaxPacketSize\_H,L と H\_CHx{x=a-e}TotalSize\_HH,HL,LH,LL の値のうち小さい方の値となります。

バルク OUT トランザクションに対して ACK を受信すると、ファームウェアに対して ACK ステータス(H\_CHx{x=a-e}IntStat.TranACK ビット)を発行します。また、FIFO を更新し、送信したデータを送信済みとして領域を開放します。

バルク OUT トランザクションに対して NAK を受信した場合、FIFO は更新せず、領域も開放しません。その為、該当チャンネルが再び選択された場合、同じトランザクションを実行します。

バルク OUT トランザクションに対して STALL を受信すると、H\_CHx{x=a-e}Config\_0.TranGo を自動クリアすることで転送を終了し、コンディションコード(H\_CHx{x=a-e}ConditionCode)を"STALL"に設定します。そしてファームウェアに対し、ChangeCondition ステータス(H\_CHx{x=a-e}IntStat.ChangeCondition ビット)を発行します。また、FIFO は更新せず、領域も開放しません。

バルク OUT トランザクションに対して正常な応答を受信しない場合、FIFO は更新せず、領域も開放しません。コンディションコード(H\_CHx{x=a-e}ConditionCode)を"RetryError"に設定し、ファームウェアに対し、TranErr ステータス(H\_CHx{x=a-e}IntStat.TranErr ビット)を発行します。そしてリトライ処理を行います。エラーが 3 回連続して続いた場合は H\_CHx{x=a-e}Control.TranGo を自動クリアすることで転送を終了し、ファームウェアに対し ChangeCondition ステータス(H\_CHx{x=a-e}IntStat.ChangeCondition ビット)を発行します。

図 1-25 に、完結する場合のバルクOUTトランザクションの様子を図示します。(a)LSIが、宛先ノードに存在するOUT方向のエンドポイントに宛てたOUTトークンを発行します。(b)LSIは続けて、マックスパケットサイズ以内のデータパケットを送信します。(c)LSIはACK受信により、該当するレジスタを自動設定し、ファームウェアに対しステータスを発行します。

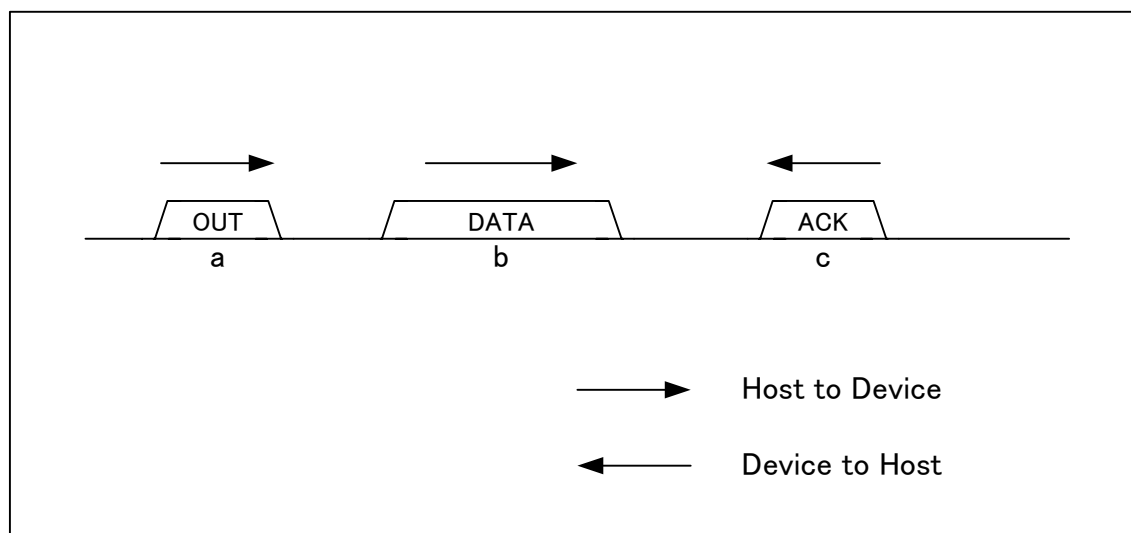


図 1-25 OUT トランザクション

### 1.3.3.3. インタラプト OUT トランザクション

CHx{x=b-e}の基本設定レジスタにおいて、転送種別(H\_CHx{x=b-e}Config\_1.TranType)を"Interrupt"、トランザクション種別(H\_CHx{x=b-e}Config\_1.TID)を"OUT"に設定します。また、トークン発行間隔(H\_CHx{x=b-e}Interval\_H,L)を設定し、その他の基本設定項目を適宜設定した後に転送実行(H\_CHx{x=b-e}Config\_0.TranGo)をセットすることにより、チャンネルはハードウェアが行う USB 転送のスケジューリングの対象となります。スケジューリングにより該当チャンネルが選択されると、トークン発行間隔(H\_CHx{x=b-e}Interval\_H,L)、FIFO の有効データ数及びフレーム残り時間を判断し、インタラプト OUT トランザクションが実行されます。

データパケットのデータ長は H\_CHx{x=b-e}MaxPacketSize\_H,L と H\_CHx{x=b-e}TotalSize\_HH,HL,LH,LL の値のうち小さい方の値となります。

インタラプト OUT トランザクションに対して ACK を受信すると、ファームウェアに対し、ACK ステータス(H\_CHx{x=b-e}IntStat.TranACK ビット)を発行します。また、FIFO を更新して、送信したデータを送信済みとして領域を開放します。

インタラプト OUT トランザクションに対して NAK を受信した場合、FIFO は更新せず、領域も開放しません。その為、該当チャンネルが再び選択された場合、同じトランザクションを実行します。

インタラプト OUT トランザクションに対して STALL を受信すると、H\_CHx{x=b-e}Config\_0.TranGo を自動クリアすることで転送を終了し、そしてコンディションコード(H\_CHx{x=b-e}ConditionCode)を"STALL"に設定します。そして、ファームウェアに対し、ChangeCondition ステータス(H\_CHx{x=b-e}IntStat.ChangeCondition ビット)を発行します。また、FIFO は更新せず、領域も開放しません。

インタラプト OUT トランザクションに対して正常な応答を受信しない場合、FIFO は更新せず、領域も開放しません。コンディションコード(H\_CHx{x=b-e}ConditionCode)を"RetryError"に設定し、ファームウェアに対し、TranErr ステータス(H\_CHx{x=b-e}IntStat.TranErr ビット)を発行します。そしてリトライ処理を行います。エラーが 3 回連続して続いた場合は H\_CHx{x=b-e}Control.TranGo を自動クリアすることで転送を終了し、ファームウェアに対し、ChangeCondition ステータス(H\_CHx{x=b-e}IntStat.ChangeCondition ビット)を発行します。

### 1.3.3.4. アイソクロナス OUT トランザクション

CHx{x=b-e}の基本設定レジスタにおいて、転送種別(H\_CHx{x=b-e}Config\_1.TranType)を"Isochronous"、トランザクション種別(H\_CHx{x=b-e}Config\_1.TID)を"OUT"に設定します。また、トークン発行間隔(H\_CHx{x=b-e}Interval\_H,L)を設定し、その他の基本設定項目を適宜設定した後に転送実行(H\_CHx{x=b-e}Config\_0.TranGo)をセットすることにより、チャンネルはハードウェアが行うUSB転送のスケジューリングの対象となります。スケジューリングにより該当チャンネルが選択された場合、トークン発行間隔(H\_CHx{x=b-e}Interval\_H,L)、FIFOの有効データ数及びフレーム残り時間を判断し、アイソクロナスOUTトランザクションが実行されます。

データパケットのデータ長は H\_CHx{x=b-e}MaxPacketSize\_H,L と H\_CHx{x=b-e}TotalSize\_HH,HL,LH,LL の値のうち小さい方の値となります。

アイソクロナス OUT トランザクションが終了すると、ファームウェアに対し、ACK ステータス(H\_CHx{x=b-e}IntStat.TranACK ビット)を発行します。また、FIFOを更新して、送信したデータを送信済みとして領域を開放します。

FIFOの有効データ数がデータパケットのデータ長に満たない場合はトランザクションを発行しません。その為FIFOは更新せず、領域も開放しません。そして、ファームウェアに対しTranErrステータスを発行し、コンディションコード(H\_CHx{x=b-e}ConditionCode)を"BufferUnderrun"に設定します。

図 1-26 に、完結する場合のアイソクロナスOUTトランザクションの様子を図示します。(a)LSIが、宛先ノードに存在するOUT方向のエンドポイントに宛てたOUTトークンを発行します。(b)LSIは続けて、マックスパケットサイズ以内のデータパケットを送信します。LSIはデータパケット送信後に該当するレジスタを自動設定し、ファームウェアに対しステータスを発行します。

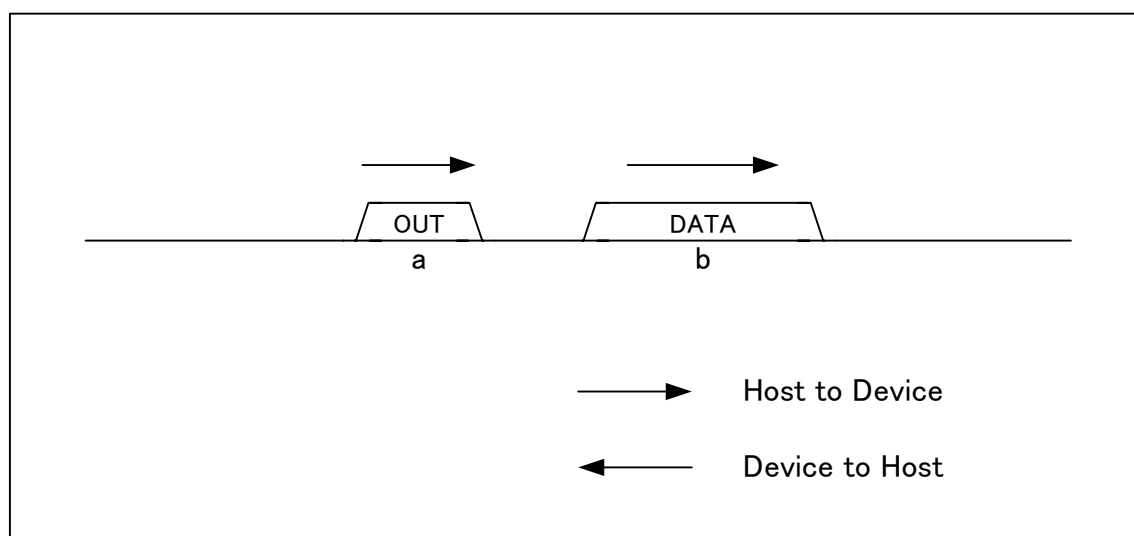


図 1-26 アイソクロナス OUT トランザクション

## 1. 機能説明

### 1.3.3.5. バルク IN トランザクション

CHx{x=a-e}の基本設定レジスタにおいて、転送種別(H\_CHx{x=b-e}Config\_1.TranType)を"Bulk"、トランザクション種別(H\_CHx{x=a-e}Config\_1.TID)を"IN"に設定します。その他の基本設定項目を適宜設定し、転送実行(H\_CHx{x=a-e}Config\_0.TranGo)をセットすることにより、チャンネルはハードウェアが行う USB 転送のスケジューリングの対象となります。スケジューリングにより該当チャンネルが選択された場合、FIFO の空き容量及びフレーム残り時間を判断し、バルク IN トランザクションが実行されます。なお、チャンネル CHa はバルク転送専用となっている為に基本設定レジスタのうち転送種別は存在しません。

受信するデータパケットの期待データ長は H\_CHx{x=a-e}MaxPacketSize\_H,L と H\_CHx{x=a-e}TotalSize\_HH,HL,LH,LL の値のうち小さい方の値となります。

バルク IN トランザクションにおいて、全てのデータが正常に受信できると、ACK 応答し、トランザクションを完結します。また、ファームウェアに対し、ACK ステータス H\_CHx{x=a-e}IntStat.TranACK ビット)を発行します。更に、FIFO を更新して、データを受信済みとして領域を確保します。

バルク IN トランザクションにおいて、受信データ長が期待データ長より小さい場合、H\_CHx{x=a-e}Config\_0.TranGo を自動クリアすることで転送を終了し、ACK 応答します。また、コンディションコード(H\_CHx{x=a-e}ConditionCode)を"DataUnderrun"に設定します。そして、ファームウェアに対し、ChangeCondition ステータス(H\_CHx{x=a-e}IntStat.ChangeCondition ビット)を発行します。更に、FIFO を更新して、データを受信済みとして領域を確保します。

バルク IN トランザクションにおいて、NAK を受信した場合、ステータスは発行しません。また、FIFO は更新しません。

バルク IN トランザクションに対して STALL を受信すると、H\_CHx{x=a-e}Config\_0.TranGo を自動クリアすることで転送を終了し、コンディションコード(H\_CHx{x=a-e}ConditionCode)を"STALL"に設定します。そしてファームウェアに対し、ChangeCondition ステータス(H\_CHx{x=a-e}IntStat.ChangeCondition ビット)を発行します。また、FIFO は更新しません。

バルク IN トランザクションにおいて、受信データ長が期待データ長より大きい場合、H\_CHx{x=a-e}Config\_0.TranGo を自動クリアすることで転送を終了します。応答は行いません。また、コンディションコード H\_CHx{x=a-e}ConditionCode)を"DataOverrun"に設定します。そして、ファームウェアに対し、ChangeCondition ステータス(H\_CHx{x=a-e}IntStat.ChangeCondition ビット)を発行します。FIFO は更新しません。

バルク IN トランザクションにおいて、トグルミスマッチが発生した場合、ACK 応答します。また、コンディションコード(H\_CHx{x=a-e}ConditionCode)を"RetryError"に設定します。そして、ファームウェアに対し、TranErr ステータス(H\_CHx{x=a-e}IntStat.TranErr ビット)を発行します。FIFO は更新しません。

バルク IN トランザクションにおいて、タイムアウトエラー、CRC エラー、ビットスタッフィングエラー、PID エラー(予期せぬ PID 含む)が発生した場合、応答は行いません。また、コンディションコード(H\_CHx{x=a-e}ConditionCode)を"RetryError"に設定します。そして、ファームウェアに対し、TranErr ステータス(H\_CHx{x=a-e}IntStat.TranErr ビット)を発行します。FIFO は更新しません。

コンディションコード(H\_CHx{x=a-e}ConditionCode)が"RetryError"に設定されるエラーが発生した場合、リトライ処理を行います。そして、エラーが 3 回連続して続く場合 H\_CHx{x=a-e}Control.TranGo を自動クリアすることで転送を終了し、ファームウェアに対し、ChangeCondition ステータス(H\_CHx{x=a-e}IntStat.ChangeCondition ビット)を発行します。

図 1-27 に、完結する場合のバルク IN トランザクションの様子を図示します。(a)LSIが、宛先ノードに存在するIN方向のエンドポイントに宛てたINトークンを発行します。(b)エンドポイントは、このIN トランザクションに応答できる場合、マックスパケットサイズ以内のデータパケットを送信します。(c)LSIはACK応答します。そして該当するレジスタを自動設定し、ファームウェアに対しステータスを発行します。



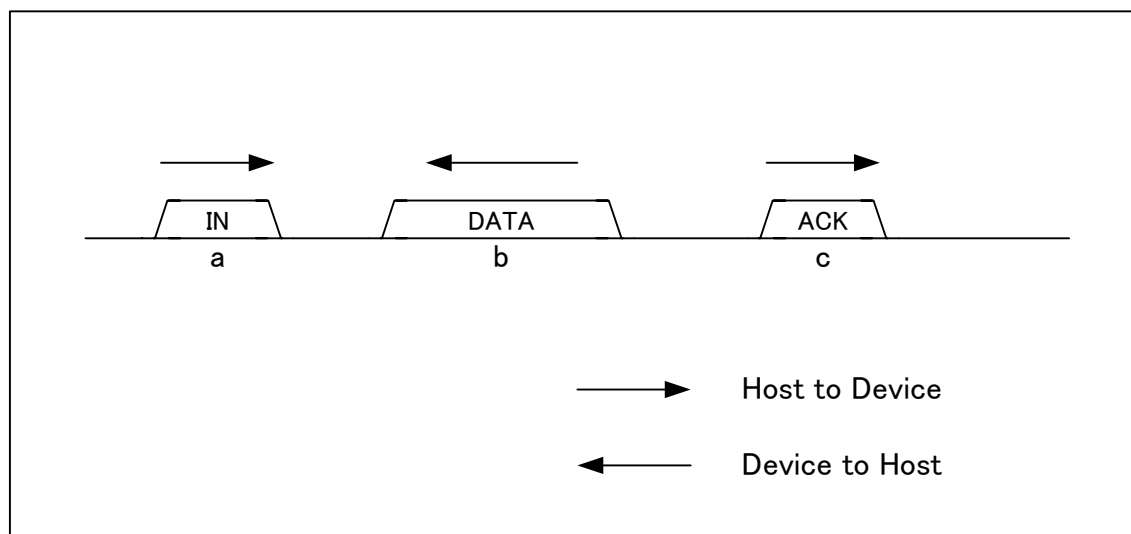


図 1-27 IN トランザクション

### 1.3.3.6. インタラプト IN トランザクション

CHx{x=b-e}の基本設定レジスタにおいて、転送種別(H\_CHx{x=b-e}Config\_1.TranType)を"Interrupt"、トランザクション種別(H\_CHx{x=b-e}Config\_1.TID)を"IN"に設定します。また、トークン発行間隔(H\_CHx{x=b-e}Interval\_H,L)を設定し、その他の基本設定項目を適宜設定し、転送実行(H\_CHx{x=b-e}Config\_0.TranGo)をセットすることにより、チャンネルはハードウェアが行う USB 転送のスケジューリングの対象となります。スケジューリングにより該当チャンネルが選択された場合、トークン発行間隔(H\_CHx{x=b-e}Interval\_H,L)、FIFO の空き容量及びフレーム残り時間を判断し、インタラプト IN トランザクションが実行されます。

受信するデータパケットの期待データ長は H\_CHx{x=b-e}MaxPacketSize\_H,L と H\_CHx{x=b-e}TotalSize\_HH,HL,LH,LL の値のうち小さい方の値となります。

インタラプト IN トランザクションにおいて、全てのデータが正常に受信できると、ACK 応答し、トランザクションを完結します。また、ファームウェアに対し、ACK ステータス(H\_CHx{x=b-e}IntStat.TranACK ビット)を発行します。また、FIFO を更新して、データを受信済みとして領域を確保します。

インタラプト IN トランザクションにおいて、受信データ長が期待データ長より小さい場合、H\_CHx{x=b-e}Config\_0.TranGo を自動クリアすることで転送を終了し、ACK 応答します。また、そしてコンディションコード(H\_CHx{x=b-e}ConditionCode)を"DataUnderrun"に設定します。そして、ファームウェアに対し、ChangeCondition ステータス(H\_CHx{x=b-e}IntStat.ChangeCondition ビット)を発行します。また、FIFO を更新して、データを受信済みとして領域を確保します。

インタラプト IN トランザクションにおいて、NAK を受信した場合、ステータスは発行しません。また、FIFO は更新しません。次のトランザクションは次の周期にて行います。

インタラプト IN トランザクションに対して STALL を受信すると、H\_CHx{x=b-e}Config\_0.TranGo を自動クリアすることで転送を終了し、コンディションコード(H\_CHx{x=b-e}ConditionCode)を"STALL"に設定します。そしてファームウェアに対し、ChangeCondition ステータス(H\_CHx{x=b-e}IntStat.ChangeCondition ビット)を発行します。また、FIFO は更新しません。

バルク IN トランザクションにおいて、受信データ長が期待データ長より大きい場合、H\_CHx{x=b-e}Config\_0.TranGo を自動クリアすることで転送を終了します。応答は行いません。また、コンディションコード H\_CHx{x=b-e}ConditionCode)を"DataOverrun"に設定します。そして、ファームウェアに対し、ChangeCondition ステータス(H\_CHx{x=b-e}IntStat.ChangeCondition ビット)を発行します。FIFO は更新しません。

インタラプト IN トランザクションにおいて、トグルミスマッチが発生した場合、ACK 応答します。また、コンディションコード(H\_CHx{x=b-e}ConditionCode)を"RetryError"に設定します。そして、ファームウェアに対し、TranErr ステータス(H\_CHx{x=b-e}IntStat.TranErr ビット)を発行します。FIFO は更

## 1. 機能説明

---

新しません。

インタラプト IN トランザクションにおいて、タイムアウトエラー、CRC エラー、ビットスタッフィングエラー、PID エラー(予期せぬ PID 含む)が発生した場合、応答は行いません。また、コンディションコード(H\_CHx{x=b-e}ConditionCode)を"RetryError"に設定します。そして、ファームウェアに対し、TranErr ステータス(H\_CHx{x=b-e}IntStat.TranErr ビット)を発行します。FIFO は更新しません。

コンディションコード(H\_CHx{x=b-e}ConditionCode)が"RetryError"に設定されるエラーが発生した場合、次の周期にてリトライ処理を行います。そして、エラーが 3 回連続して続く場合 H\_CHx{x=b-e}Control.TranGo を自動クリアすることで転送を終了し、ファームウェアに対し、ChangeCondition ステータス(H\_CHx{x=b-e}IntStat.ChangeCondition ビット)を発行します。

### 1.3.3.7. アイソクロナス IN トランザクション

CHx{x=b-e}の基本設定レジスタにおいて、転送種別(H\_CHx{x=b-e}Config\_1.TranType)を"Isochronous"、トランザクション種別(H\_CHx{x=b-e}Config\_1.TID)を"IN"に設定します。また、トークン発行間隔(H\_CHx{x=b-e}Interval\_H,L)を設定し、その他の基本設定項目を適宜設定し、転送実行(H\_CHx{x=b-e}Config\_0.TranGo)をセットすることにより、チャンネルはハードウェアが行う USB 転送のスケジューリングの対象となります。スケジューリングにより該当チャンネルが選択された場合、トークン発行間隔(H\_CHx{x=b-e}Interval\_H,L)、FIFO の空き容量及びフレーム残り時間を判断し、アイソクロナス IN トランザクションが実行されます。

受信するデータパケットの期待データ長は H\_CHx{x=b-e}MaxPacketSize\_H,L と H\_CHx{x=b-e}TotalSize\_HH,HL,LH,LL の値のうち小さい方の値となります。

アイソクロナス IN トランザクションにおいて、全てのデータが正常に受信できると、トランザクションを完結します。また、ファームウェアに対し、ACK ステータス(H\_CHx{x=b-e}IntStat.TranACK ビット)を発行します。また、FIFO を更新して、データを受信済みとして領域を確保します。

アイソクロナス IN トランザクションにおいて、受信データ長が期待データ長より小さい場合、ファームウェアに対し、ACK ステータス(H\_CHx{x=b-e}IntStat.TranACK ビット)及び ChangeCondition ステータス(H\_CHx{x=b-e}IntStat.ChangeCondition ビット)を発行します。そしてコンディションコード(H\_CHx{x=b-e}ConditionCode)を"DataUnderrun"に設定します。また、FIFO を更新して、データを受信済みとして領域を確保します。

アイソクロナス IN トランザクションにおいて、受信データ長が期待データ長より大きい場合、ファームウェアに対し、ChangeCondition ステータス(H\_CHx{x=b-e}IntStat.ChangeCondition ビット)を発行します。そしてコンディションコード(H\_CHx{x=b-e}ConditionCode)を"DataOverrun"に設定します。FIFO は更新しません。

アイソクロナス IN トランザクションにおいて、タイムアウトエラー、CRC エラー、ビットスタッフィングエラー、PID エラー(予期せぬ PID 含む)が発生した場合、ファームウェアに対し、TranErr ステータス(H\_CHx{x=b-e}IntStat.TranErr ビット)を発行します。また、コンディションコード(H\_CHx{x=b-e}ConditionCode)を"RetryError"に設定します。FIFO は更新しません。

FIFO の空き領域のサイズが H\_CHx{x=b-e}MaxPacketSize\_H,L が示す値に満たない場合はトランザクションを発行しません。そして、ファームウェアに対し TranErr ステータスを発行し、コンディションコード(H\_CHx{x=b-e}ConditionCode)を"BufferOverrun"に設定します。

図 1-28 に、完結する場合のアイソクロナス IN トランザクションの様子を図示します。(a)LSIが、宛先ノードに存在するIN方向のエンドポイントに宛てたINトークンを発行します。(b)エンドポイントは、このINトランザクションに応答できる場合、マックスパケットサイズ以内のデータパケットを送信します。LSIはデータパケット受信後に該当するレジスタを自動設定し、ファームウェアに対しステータスを発行します。

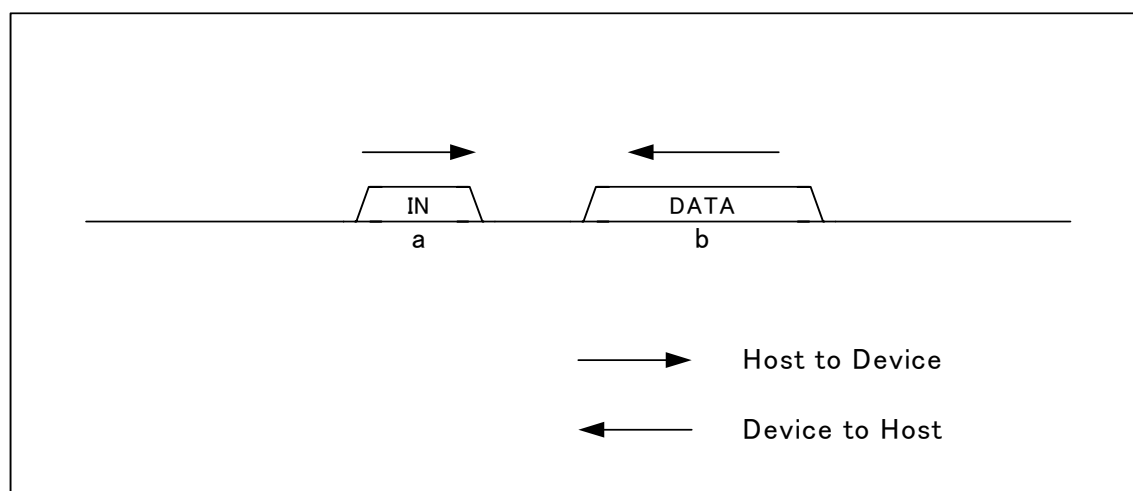


図 1-28 アイソクロナス IN トランザクション

## 1. 機能説明

### 1.3.3.8. PING トランザクション

バルク OUT またはコントロール OUT トランザクションを行うチャネルでは、HS 動作時に、以下に示す条件で PING トランザクションを実行する状態に移行します。チャネルが PING トランザクションを実行する状態の時にスケジューリングにより選択されると PING トランザクションを実行します。

OUT トランザクションに対して、NYET を受信した場合、NAK を受信した場合、または応答を受信しない場合、チャネルは PING トランザクションを実行する状態に移行します。

PING トランザクションに対して NAK を受信した場合、チャネルは引き続き PING トランザクションを実行する状態となります。また、ファームウェアに対しステータスは発行しません。

PING トランザクションに対して ACK を受信した場合、チャネルは PING トランザクションを実行する状態から OUT トランザクションを行う状態に戻ります。ファームウェアに対しステータスは発行しません。

PING トランザクションに対して STALL を受信した場合、 $H\_CHx\{x=0,a-e\}Config\_0.TranGo$  を自動クリアすることで転送を終了し、コンディションコード( $H\_CHx\{x=0,a-e\}ConditionCode$ )を"STALL"に設定します。そして、ファームウェアに対し、ChangeCondition ステータス( $H\_CHx\{x=0,a-e\}IntStat.ChangeCondition$  ビット)を発行します。

PING トランザクションに対して正常な応答を受信しない場合、コンディションコード( $H\_CHx\{x=0,a-e\}ConditionCode$ )を"RetryError"に設定します。そして、ファームウェアに対し、TranErr ステータス( $H\_CHx\{x=0,a-e\}IntStat.TranErr$  ビット)を発行します。この場合、リトライ処理を行いますが、エラーが3回連続して続く場合、 $H\_CHx\{x=0,a-e\}Control.TranGo$  を自動クリアすることで転送を終了し、ファームウェアに対し ChangeCondition ステータス( $H\_CHx\{x=0,a-e\}IntStat.ChangeCondition$  ビット)を発行します。

PING トランザクションにおいては、FIFO が更新されることはありません。

図 1-29 に、PING トランザクションに対して ACK 応答する様子を図示します。(a)LSI は、このノードに存在する OUT 方向のエンドポイントに宛てた PING トークンを発行します。(b)デバイス は、エンドポイントにマックスパケットサイズ分の空きがある場合、この PING トランザクションに対し ACK 応答します。

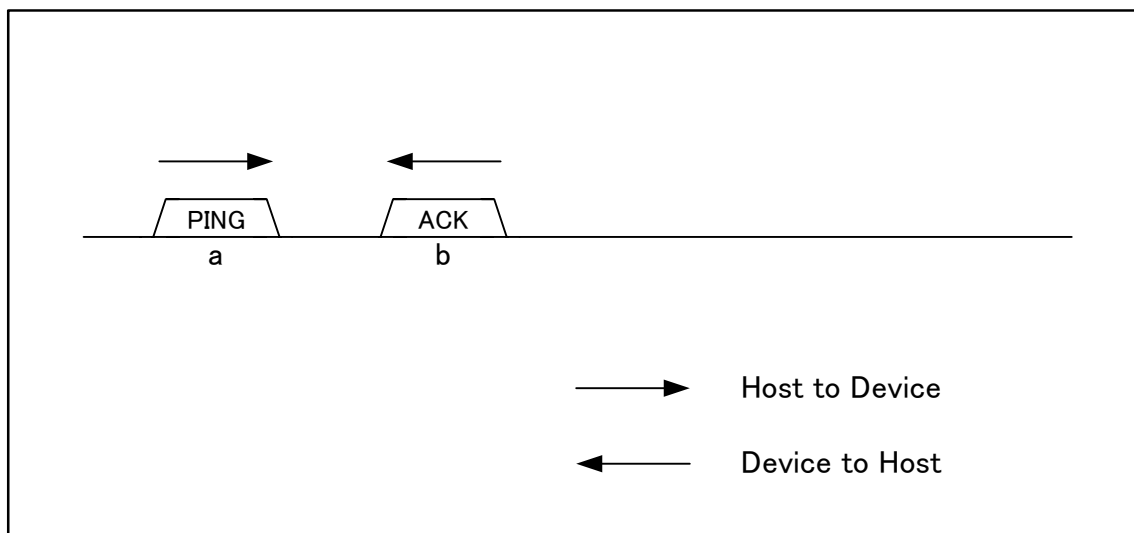


図 1-29 PING トランザクション

### 1.3.3.9.low-speed(LS)トランザクション

LS デバイスとの転送にはコントロール転送、またはインタラプト転送を使用します。

ダウンストリームポートに LS デバイスが接続された場合、ホストは LS にて動作します。使用するチャンネルの転送速度(H\_CHx{x=0,a-e}Config\_0.SpeedMode)を"LS"に設定することで LS bit time にてトランザクションを実行します。

一方、ダウンストリームポートに full-speed(FS)ハブが接続され、そのハブのダウンストリームポートに LS デバイスが接続された場合、ホストは FS にて動作します。使用するチャンネルの転送速度(H\_CHx{x=0,a-e}Config\_0.SpeedMode)を LS に設定することで、対応するエンドポイントへの全てのダウンストリーム・パケットの先頭にプリアンプルを伴って送信します。プリアンプルは FS bit time にて送信し、プリアンプルに続くダウンストリーム・パケットは LS bit time にて送信します。

図 1-30 に、完結する場合のインタラプトOUTトランザクションの様子を図示します。(a)LSIは、宛先ノードに存在するOUT方向のエンドポイントに宛てたOUTトークンを先頭にプリアンプルを付与して発行します。(b)LSIは続けて、マックスパケットサイズ以内のデータパケットを先頭にプリアンプルを付与して送信します。(c)LSIはACK受信により、該当するレジスタを自動設定し、ファームウェアに対しステータスを発行します。

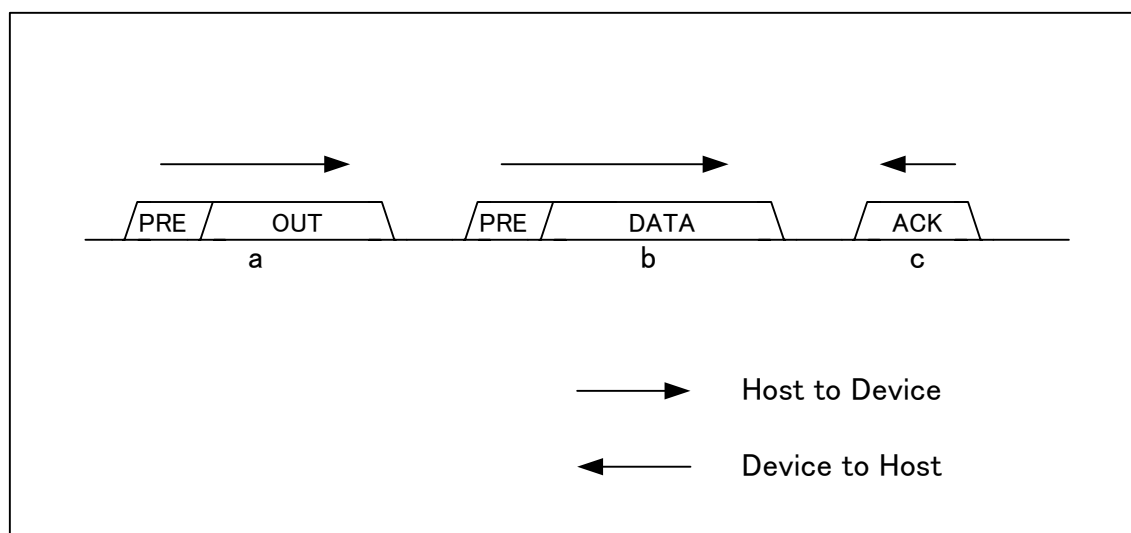


図 1-30 Preamble が付与された OUT トランザクション

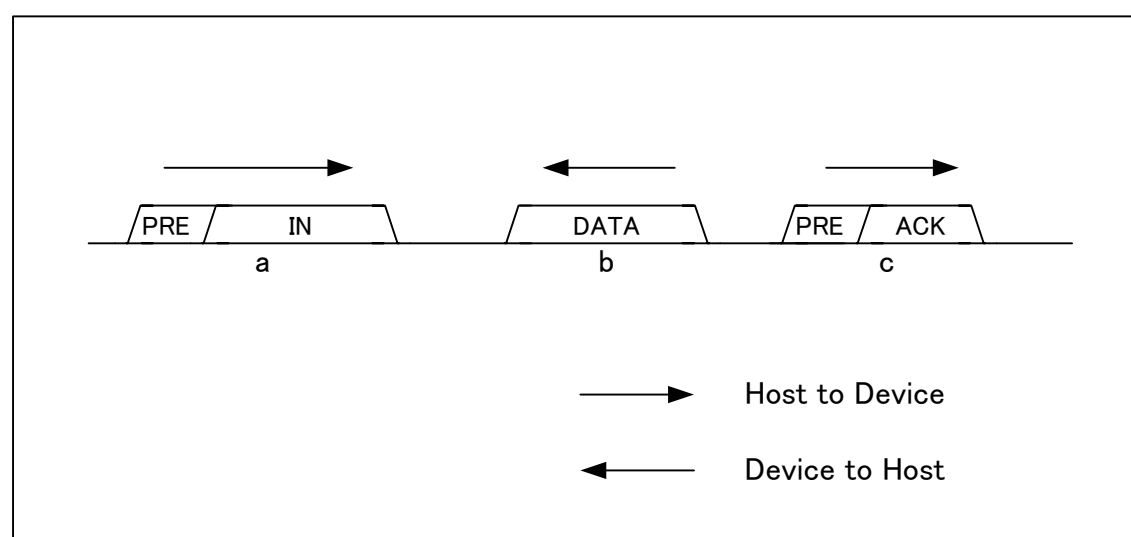


図 1-31 Preamble が付与された IN トランザクション

図 1-31 に、完結する場合のインタラプトINトランザクションの様子を図示します。(a)LSIは、宛先

## 1. 機能説明

---

ノードに存在するIN方向のエンドポイントに宛てたINトークンを先頭にプリアンプルを付与して発行します。(b)デバイスはマックスパケットサイズ以内のデータパケットを送信してきます。LSIはこのデータを該当するチャンネルのFIFOに書き込みます。(c)LSIはデータを受信出来るとACK応答を先頭にプリアンプルを付与して行います。また、該当するレジスタを自動設定し、ファームウェアに対しステータスを発行します。

### 1.3.3.10. スプリットトランザクション

ダウンストリームポートに high-speed(HS)ハブが接続され、そのハブのダウンストリームポートに FS または LS デバイスが接続された場合、ホストは HS にて動作します。使用するチャンネルの転送速度 ( $H\_CHx\{x=0,a-e\}Config\_0.SpeedMode$ )を FS または LS に設定することで、ハブとの間にて、対応するエンドポイントへのトランザクションをスプリットトランザクションで実行します。

該当チャンネルにはハブアドレス ( $H\_CHx\{x=0,a-e\}HubAdrs.HubAdrs$ )、ポート番号 ( $H\_CHx\{x=0,a-e\}HubAdrs.Port$ )を適切な値に設定します。

スプリットトランザクションにおけるスタートスプリットトランザクションからコンプリートスプリットトランザクションのシーケンスは、ハードウェアが制御します。その為ファームウェアはスプリットトランザクションの個々のトランザクションを意識する必要はありません。

コントロール転送、バルク転送、インタラプト転送、及び IN 方向のアイソクロナス転送の場合、スタートスプリットトランザクションからコンプリートスプリットトランザクションのシーケンスにおいて、最終のコンプリートスプリットトランザクションが正常に完結した場合、ファームウェアに対し ACK ステータス ( $H\_CHx\{x=0,a-e\}IntStat.TranACK$  ビット)を発行し、FIFO を更新します。

一方、OUT 方向のアイソクロナス転送の場合、最終のスタートスプリットトランザクションが正常に完結した場合、ファームウェアに対し ACK ステータス ( $H\_CHx\{x=b-e\}IntStat.TranACK$  ビット)を発行し、FIFO を更新します。

上記以外のスタートスプリットトランザクション及びコンプリートスプリットトランザクションが正常に完結した場合、ファームウェアに対してのステータスの発行は行いません。

スタートスプリットトランザクションからコンプリートスプリットトランザクションのシーケンスにおいて、個々のスプリットトランザクションでエラーが発生した場合、コンディションコード ( $H\_CHx\{x=0,a-e\}ConditionCode$ )を "RetryError" に設定し、ファームウェアに対し、TranErr ステータス ( $H\_CHx\{x=0,a-e\}IntStat.TranErr$  ビット)を発行します。FIFO は更新しません。そしてリトライ処理を行います。コントロール転送、バルク転送、インタラプト転送ではエラーが 3 回連続して続く場合、 $H\_CHx\{x=0,a-e\}Control.TranGo$  を自動クリアすることで転送を終了し、ファームウェアに対し、ChangeCondition ステータス ( $H\_CHx\{x=0,a-e\}IntStat.ChangeCondition$  ビット)を発行します。

## 1. 機能説明

### 1.3.4. コントロール転送

コントロール転送は、その各ステージを個々のトランザクションとして制御するか、または、後述(「1.3.4.3.コントロール転送サポート機能」)のコントロール転送サポート機能によって、自動的行えます。

図 1-32 にコントロール転送の制御の様子を図示します。ファームウェアはSETUP、DATA、STATUSの各ステージを適宜設定することでコントロール転送をハードウェアに実行させます。

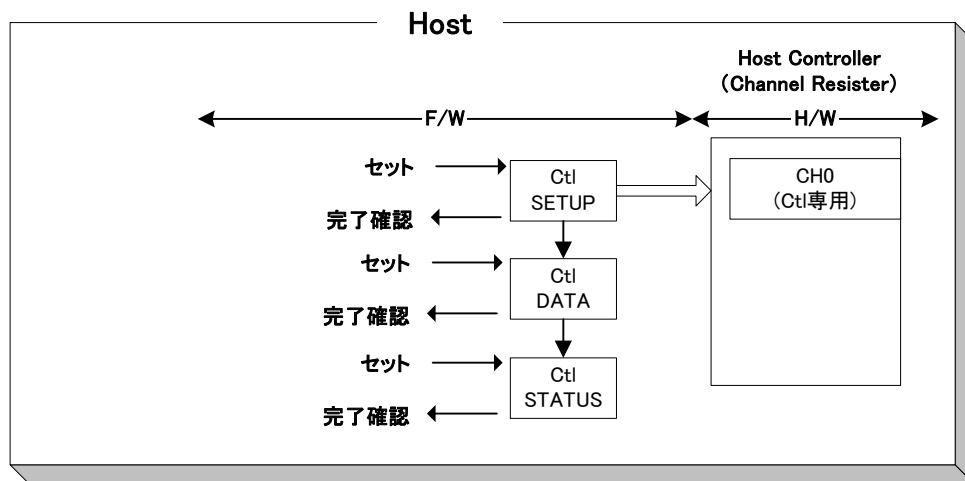


図 1-32 コントロール転送の制御

図 1-33 にデータステージがOUT方向である場合のコントロール転送の様子を図示します。(a)ホストは、SETUPトランザクションによって、コントロール転送を開始します。(b)ホストはOUTトランザクションを発行して、データステージを行います。(c)ホストはINトランザクションを発行して、ステータスステージを行います。

データステージの無いコントロール転送は、この例におけるデータステージが無い状態で実施されます。

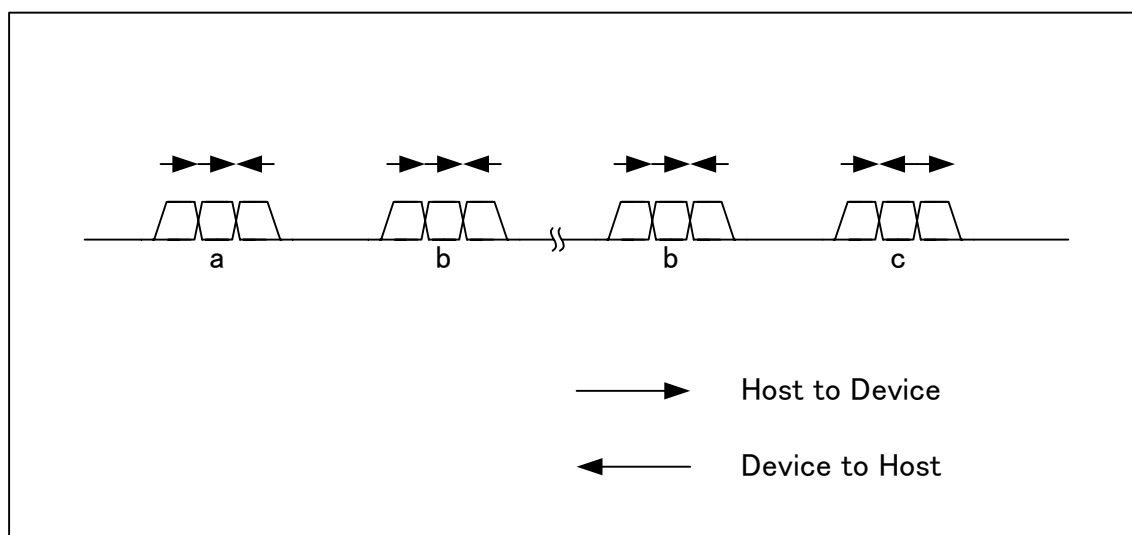


図 1-33 データステージが OUT 方向のコントロール転送



図 1-34 にデータステージがIN方向である場合のコントロール転送の様子を図示します。(a)ホストは、SETUPトランザクションによって、コントロール転送を開始します。(b)ホストはINトランザクションを発行して、データステージを行います。(c)ホストはOUTトランザクションを発行して、ステータスステージを行います。

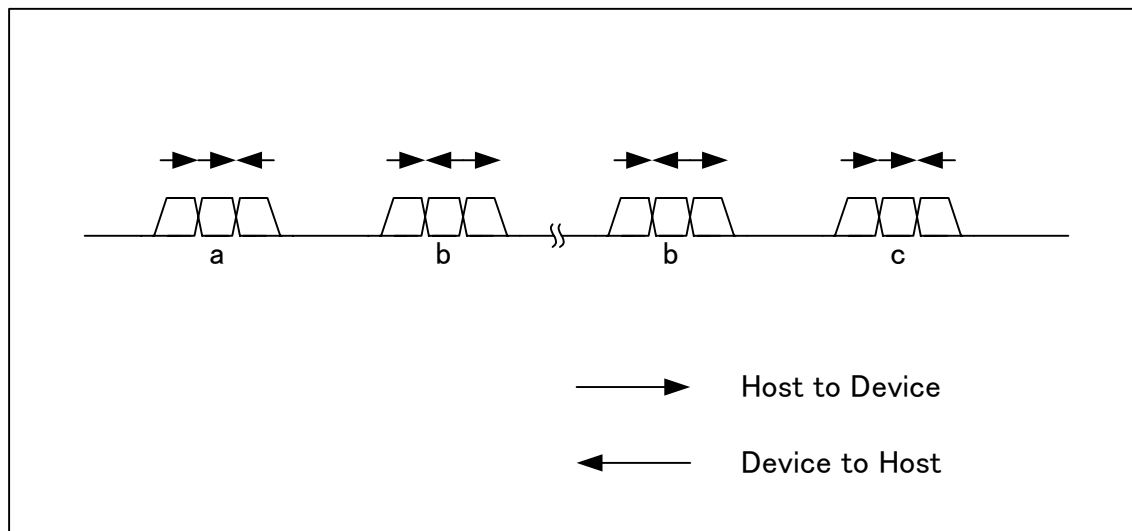


図 1-34 データステージが IN 方向のコントロール転送

#### 1.3.4.1. セットアップステージ

セットアップステージはセットアップトランザクションによって行います。セットアップトランザクションの詳細は「1.3.3.1.SETUPトランザクション」を参照して下さい。

#### 1.3.4.2. データステージ/ステータスステージ

セットアップステージ終了後、次のステージに移行してください。

そのステージが IN 方向である場合、トランザクション種別(H\_CH0Config\_1.TID)を"IN"に設定し、その他の基本設定レジスタを適宜設定して、トランザクションを実行してください。

一方、そのステージが OUT 方向である場合、トランザクション種別(H\_CH0Config\_1.TID)を"OUT"に設定し、その他の基本設定レジスタを適宜設定して、トランザクションを実行してください。

なお、ステータスステージを行う場合、IRP データ数(H\_CH0TotalSize\_H,L)を 0x0 に設定してトランザクションを実行してください。

#### 1.3.4.3. コントロール転送サポート機能

本 LSI にはコントロール転送の一連のステージ管理を自動で行う機能があります。この機能を用いることにより、ファームウェアでは各ステージを個々のトランザクションとして管理する必要がなくなります。

## 1. 機能説明

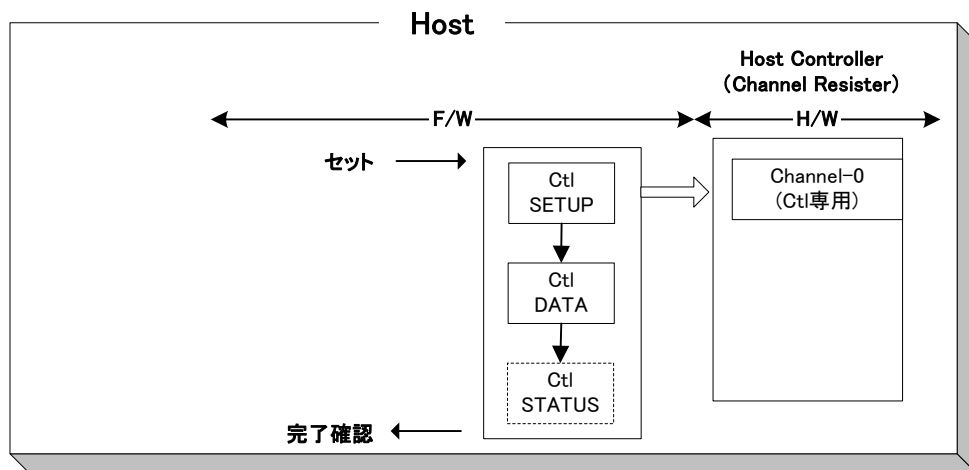


図 1-35 コントロール転送サポート機能の制御

コントロール転送サポート機能はチャンネル CH0 にて有効です。この機能を使用する場合はチャンネル CH0 にジョインする FIFO 領域として AREA0 を選択してください。この機能を使用した場合のコントロール転送は以下のようになります。ファームウェアは(1)～(4)、(7)の処理を行います。

- (1) チャンネル CH0 の以下の基本設定レジスタに適宜設定を行います。  
転送速度(H\_CH0Config\_0.SpeedMode)、マックスパケットサイズ(H\_CH0MaxPktSize)、  
USBアドレス(H\_CH0FuncAdrs.FuncAdrs)、  
エンドポイントナンバー(H\_CH0FuncAdrs.EP\_Number)、FIFO領域  
(AREA0StartAdrs\_H,L,,AREA0EndAdrs\_H,L)、  
FIFO領域ジョイン(AREA0Join\_1.JoinEP0CH0)
- (2) セットアップレジスタ(H\_CH0SETUP\_0～7にセットアップデータ(8Byte)を書き込みます。
- (3) データステージが OUT 方向の場合は CH0 にジョインした FIFO 領域に送信すべきデータを書き込みます。データステージが IN 方向の場合は CH0 にジョインした FIFO 領域をクリアします。
- (4) コントロール転送サポート実行(H\_CTL\_SupportControl.CTL\_SupportGo)をセットします。  
この時、コントロール転送ステージ(H\_CTL\_SupportControl.CTL\_SupportState)の値は"Idle(00b)"として H\_CTL\_SupportControl レジスタにライトを行います。
- (5) SETUP レジスタのデータ(8Byte)を用いて SETUP トランザクションを実行します(セットアップステージ)。
- (6) SETUP データに基づき、データステージを実行します。
  - SETUP データの bmRequestType の Bit7 が 0 の場合、CH0 にジョインした FIFO 領域にあるデータを OUT トランザクションによって送信します(OUT 方向のデータステージ)。OUT 方向のデータステージはSETUP データの wLength で示されるデータ数を OUT トランザクションで送信すると終了します。
  - SETUP データの bmRequestType の Bit7 が 1 の場合、IN トランザクションを発行し、受信したデータを CH0 にジョインした FIFO 領域に書き込みます(IN 方向のデータステージ)。IN 方向のデータステージはSETUP データの wLength で示されるデータ数を IN トランザクションで受信すると終了します。また、IN トランザクションでショートパケットを受信した場合も IN 方向のデータステージは終了します。
  - SETUP データの wLength の値が 0x0000 の場合はデータステージは実行しません。
- (7) CH0 にジョインした FIFO 領域が SETUP データの wLength で示される値より小さい場合、ファームウェアはデータステージのデータを分割して処理する必要があります。
  - データステージが OUT 方向の場合、CH0 にジョインした FIFO 領域に送信するデータが無くなるとトランザクションは発行されなくなります。従ってファームウェアは FIFO の空き領域を確認しながら、順次残りの送信データを FIFO に書き込む必要があります。

- ・ データステージが IN 方向の場合、CH0 にジョインした FIFO 領域に空き領域が無くなるとトランザクションは発行されなくなります。従ってファームウェアは FIFO の有効データ数を確認しながら順次受信したデータを FIFO から読み出し、FIFO に空き領域を作る必要があります。
- (8) SETUP データに基づき、ステータスステージを実行します。
- ・ OUT 方向のデータステージが終了すると、IN トランザクションを発行します(IN 方向のステータスステージ)。
- ・ IN 方向のデータステージが終了し、CH0 にジョインした FIFO 領域に存在する受信データがすべて読み出され FIFO が空になると、ゼロ長パケットの OUT トランザクションを発行します(OUT 方向のステータスステージ)。
- (9) コントロール転送が正常に完了するとコントロール転送サポート実行(H\_CTL\_SupportControl.CTL\_SupportGo)を自動クリアし、コントロール転送終了ステータス(H\_CH0IntStat.CTL\_SupportCmp)を発行します。
- (10) コントロール転送の途中でトランザクションのエラーを検出すると、コントロール転送サポート実行(CTL\_SupportControl.CTL\_SupportGo)を自動クリアすることでコントロール転送を中止し、コントロール転送停止ステータス(H\_CH0IntStat.CTL\_SupportStop)を発行します。そして、コントロール転送ステージ(H\_CTL\_SupportControl.CTL\_SupportState)により、エラーが発生したステージを示します。また、コンディションコード(H\_CH0ConditionCode)を適切な値に設定し、ChangeCondition ステータス(H\_CH0IntStat.ChangeCondition ビット)を発行します。

コントロール転送を中断する場合は、コントロール転送サポート実行(H\_CTL\_SupportControl.CTL\_SupportGo)をクリアします。コントロール転送の中断処理が終了するとステータスを発行します。

この時、コントロール転送がステータスステージまで完了して中断処理が終了した場合はコントロール転送終了ステータス(H\_CH0IntStat.CTL\_SupportCmp)を発行します。

一方、コントロール転送がステータスステージまで完了せずに中断処理が終了した場合はコントロール転送停止ステータス(H\_CH0IntStat.CTL\_SupportStop)を発行します。

コントロール転送の中断したステージはコントロール転送ステージ(H\_CTL\_SupportControl.CTL\_SupportState)により示します。

中断したステージよりコントロール転送を再開する場合、コントロール転送ステージ(H\_CTL\_SupportControl.CTL\_SupportState)を再開するステージに設定(即ち中断したステージに設定を保持)し、コントロール転送サポート実行(H\_CTL\_SupportControl.CTL\_SupportGo)をセットします。

一方、新たなコントロール転送を行う場合は、コントロール転送ステージ(H\_CTL\_SupportControl.CTL\_SupportState)に"Idle(00b)"を設定し、コントロール転送サポート実行(H\_CTL\_SupportControl.CTL\_SupportGo)をセットします。

コントロール転送サポート機能を実行中は、転送実行ビット(H\_CH0Config\_0.TranGo)、トグルシーケンスビット(H\_CH0Config\_0.Toggle)、トランザクション種別(H\_CH0Config\_1.TID)、IRP データ数(H\_CH0TotalSize\_H,L)はハードウェアにより設定および更新しますので、これらに対する書き込みは行わないで下さい。

トランザクションのエラーについては、「1.3.3.トランザクション」を参照して下さい。

コントロール転送サポート機能の設定項目、ステータスを表 1-28 に示します。

## 1. 機能説明

---

表 1-28 コントロール転送サポート機能の制御項目及びステータス

項目	レジスタ／ビット	説明
コントロール転送サポート 実行	H_CTL_SupportControl.CTL_SupportGo	コントロール転送のステージの管理を自動で行います。 詳細については、「1.3.4.3.コントロール転送サポート機能」を参照して下さい。
コントロール転送 ステージ	H_CTL_SupportControl.CTL_SupportState	コントロール転送サポート機能において、実行中のステージを示します。また、コントロール転送がエラーで中止された場合は、エラーが発生したステージを示します。
コントロール転送 実行結果	H_CH0IntStat.CTL_SupportCmp H_CH0IntStat.CTL_SupportStop	コントロール転送サポート機能によるコントロール転送の実行結果を示します。
トランザクション ステータス	H_CH0IntStat.TotalSizeCmp, H_CH0IntStat.TranACK, H_CH0IntStat.TranErr, H_CH0IntStat.ChangeCondition	トランザクションの結果を示します。
トランザクション コンディションコード	H_CH0ConditionCode.ConditionCode	トランザクション結果の詳細を示します。

### 1.3.5. バルク転送／インタラプト転送／アイソクロナス転送

CHaにおけるバルク転送、CHb, CHc, CHd, CHeにおけるバルク転送、インタラプト転送及びアイソクロナス転送はデータフロー(「1.3.6.データフロー」参照)としても、連続する個々のトランザクション(「1.3.3.トランザクション」参照)としても制御できます。

### 1.3.6. データフロー

OUT 転送及び IN 転送の一般的なデータフローの制御について、説明します。

#### 1.3.6.1. OUT 転送

H\_CH0TotalSize\_H,L または H\_CHx{x=a-e}TotalSize\_HH,HL,LH,LL に OUT 転送の総データ数を設定し、OUT 転送により送信するデータを、各チャンネルにジョインした FIFO 領域上に書き込んで下さい。FIFO にデータを書き込むには、CPU インタフェイスによるレジスタ書き込みと、CPU インタフェイスによる DMA 書き込み、及び IDE からのリード転送の方法があります。

CPU インタフェイスのレジスタライトにより FIFO にデータを書き込むには、AREAn{n=0-5}Join\_0.JoinCPU\_Wr ビットにより、チャンネルにジョインした FIFO 領域を 1 つ選択して下さい。選択した FIFO 領域には、FIFO\_Wr\_0,1 レジスタ、または FIFO\_ByteWr レジスタにより書き込むことができ、書き込み順にデータパケットで送信されます。また、FIFO の空き容量を、FIFO\_WrRemain\_H,L レジスタにより参照できます。フル状態の FIFO へ書き込むことは出来ません。必ず FIFO\_WrRemain\_H,L レジスタにより空き数を確認し、その数を超えないように書き込んで下さい。

IDE インタフェイスのリード転送により FIFO へデータを書き込むには、AREAn{n=0-5}Join\_0.JoinIDE ビットにより、ただ一つのチャンネルを選択し、IDE\_Control.Dir ビットに 0 を設定して下さい。選択したチャンネルの FIFO は、IDE\_Control.IDE\_Go による IDE 転送を実行することにより、IDE からリードした順に書き込まれ、書き込み順にデータパケットで送信されます。FIFO がフルになると、IDE インタフェイスは自動的にリード転送を一時停止して、フロー制御を行います。

OUT トランザクションで送信するデータパケットのサイズは、チャンネル CH0 では H\_CH0TotalSize\_H,L と H\_CH0MaxPktSize の値のうち小さい方となります。チャンネル CHx{x=a-e}では H\_CHx{x=a-e}TotalSize\_HH,HL,LH,LL と H\_CHx{x=a-e}MaxPktSize\_H,L の値のうち小さい方となります。

FIFO にデータパケットのデータサイズ以上のデータが有れば、OUT トランザクションを実行し、データを送信します。また、H\_CH0TotalSize\_H,L または H\_CHx{x=a-e}TotalSize\_HH,HL,LH,LL は送信したデータサイズ分減算されます。そして、TotalSize がゼロになると、H\_CHx{x=0,a-e}Config\_0.TranGo を自動クリアすることで転送を終了し、ファームウェアに対し、TotalSizeCmp ステータス(H\_CHx{x=0,a-e}IntStat.TotalSizeCmp ビット)を発行します。

このように、ファームウェアによって個々のトランザクションについての制御を行うことなく、OUT 転送を行うことが出来ます。

#### 1.3.6.2. IN 転送

H\_CH0TotalSize\_H,L または H\_CHx{x=a-e}TotalSize\_HH,HL,LH,LL に IN 転送の総データ数を設定して下さい。

IN トランザクションで受信するデータパケットの期待サイズは、チャンネル CH0 では H\_CH0TotalSize\_H,L と H\_CH0MaxPktSize の値のうち小さい方となります。チャンネル CHx{x=a-e}では H\_CHx{x=a-e}TotalSize\_HH,HL,LH,LL と H\_CHx{x=a-e}MaxPktSize\_H,L の値のうち小さい方となります。FIFO にマックスパケットサイズ以上の空き領域が有れば、IN トランザクション実行して、データを受信します。また、H\_CH0TotalSize\_H,L または H\_CHx{x=0,a-e}TotalSize\_HH,HL,LH,LL は受信したデータサイズ分減算されます。そして、TotalSize がゼロになると、H\_CHx{x=0,a-e}Config\_0.TranGo を自動クリアすることで転送を終了し、ファームウェアに対し、TotalSizeCmp ステータス(H\_CHx{x=0,a-e}IntStat.TotalSizeCmp ビット)を発行します。

受信したデータサイズが期待データ長より大きい場合、コンディションコード H\_CHx{x=0,a-e}ConditionCode)を "DataOverrun" に設定します。そして、ファームウェアに対し、ChangeCondition ステータス(H\_CHx{x=0,a-e}IntStat.ChangeCondition ビット)を発行します。FIFO は更新しません。コントロール転送、バルク転送、インタラプト転送の場合は更に H\_CHx{x=0,a-e}Config\_0.TranGo を自動クリアし、そのチャンネルでの転送を終了します。

## 1. 機能説明

受信したデータサイズがデータパケットの期待サイズ未満であれば、コンディションコード (H\_CHx{x=0,a-e}ConditionCode) を "DataUnderrun" に設定します。そしてファームウェアに対し、ChangeCondition ステータス (H\_CHx{x=0,a-e}IntStat.ChangeCondition ビット) を発行します。また、FIFO を更新して、データを受信済みとして領域を確保します。コントロール転送、バルク転送、インタラプト転送の場合は更に H\_CHx{x=0,a-e}Config\_0.TranGo を自動クリアし、そのチャンネルでの転送を終了します。

このように、ファームウェアによって個々のトランザクションについての制御を行うことなく、IN 転送を行うことが出来ます。

IN 転送によって受信したデータは、各チャンネルのジョインした FIFO 領域上に書き込まれます。FIFO のデータを読み出すには、CPU インタフェイスによるレジスタ読み出し、CPU インタフェイスによる DMA 読み出し、及び IDE へのライト転送の方法があります。

CPU インタフェイスのレジスタリードにより FIFO のデータを読み出すには、該当するチャンネルとジョインされた FIFO 領域を AREAn{n=0-5}Join\_0.JoinCPU\_Rd ビットによりただ一つ選択して下さい。選択した FIFO 領域は、FIFO\_Rd\_0,1 レジスタ、または FIFO\_ByteRd レジスタにより、受信順に読み出すことが出来ます。また、読み出し可能な FIFO のデータ数を、FIFO\_RdRemain\_H,L レジスタにより参照できます。空の FIFO を読み出すことはできませんので、必ず FIFO\_RdRemain\_H,L レジスタによりデータ数を確認し、その数を超えないように読み出して下さい。

CPU インタフェイスの DMA リードにより FIFO のデータを読み出すには、AREAn{n=0-5}Join\_1.JoinDMAx{x=0,1} ビットにより、DMA のチャンネル毎にただ一つの FIFO 領域を選択し、DMAx{x=0,1}\_Control.Dir ビットに 1 を設定して下さい。選択した FIFO 領域は、CPU インタフェイスにおいて DMA 手順を実行することにより、受信順に読み出されます。また、FIFO の残りデータ数を、DMAx{x=0,1}\_Remain\_H,L レジスタで参照できます。FIFO が空になると、CPU インタフェイスは自動的に DMA を一時停止してフロー制御を行います。

IDE インタフェイスのライト転送により、FIFO のデータを読み出すには、AREAn{n=0-5}Join\_1.IDE ビットにより、ただ一つの FIFO 領域を選択し IDE\_Control.Dir ビットに 1 を設定して下さい。選択したチャンネルの FIFO は、IDE\_Control.IDE\_Go による IDE 転送を実行することにより、受信順に読み出されます。FIFO が空になると、IDE インタフェイスは自動的にライト転送を一時停止して、フロー制御を行います。

### 1.3.7. ゼロ長パケット自動発行機能

OUT 転送を行うチャンネルにおいて、H\_CHx{x=a-e}Config\_1.AutoZeroLen ビットをセットすることにより、ゼロ長パケットを自動で発行する機能が有効になります。

ゼロ長パケット自動発行機能の設定項目を表 1-29 に示します。

表 1-29 ゼロ長パケット自動発行機能の制御項目

項目	レジスタ／ビット	説明
ゼロ長パケット 自動発行	H_CHx{x=a-e}Config_1.AutoZeroLen	ゼロ長パケットの自動発行機能を有効にします。 OUT 転送の場合のみこのビットは有効です。

#### 1.3.7.1. バルク／インタラプト OUT 転送のゼロ長パケット自動発行機能

バルク／インタラプト OUT 転送を実行しているチャンネルにおいて、H\_CHx{x=a-e}TotalSize\_HH,HL,LH,LL レジスタで設定したデータサイズの転送がちょうどマックスパケットサイズで完了しても H\_CHx{x=a-e}Config\_0.TranGo の自動クリアは行わず、転送を継続します。そして、再びこのチャンネルがスケジューリングされると OUT トランザクションをゼロ長パケットで実行します。このトランザクションが正常に完結すると、H\_CHx{x=a-e}Config\_0.TranGo を自動クリアすることで転送を終了し、ファームウェアに対し、TotalSizeCmp ステータス (H\_CHx{x=a-e}IntStat.TotalSizeCmp) を発行します。

### 1.3.8. バルクオンリーサポート機能

本LSIには、USB Mass Storage Class(BulkOnly Transport Protocol)のコマンドトランスポート(CBW)、データトランスポート、ステータストランスポート(CSW)の一連のトランスポート管理を自動で行う機能があります。この機能を用いることにより、ファームウェアは各トランスポートの制御を個別に行う必要がなくなります。バルクオンリーサポート機能を用いた場合の制御の例を図 1-36 に、また、使用せずに各トランスポートを個別のトランザクションとして制御した場合の例を図 1-37 に示します。

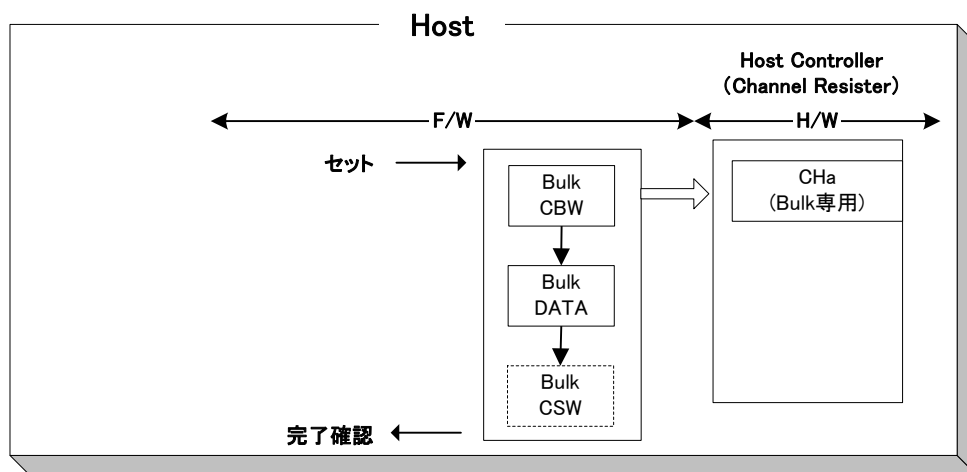


図 1-36 バルクオンリーサポート機能の制御

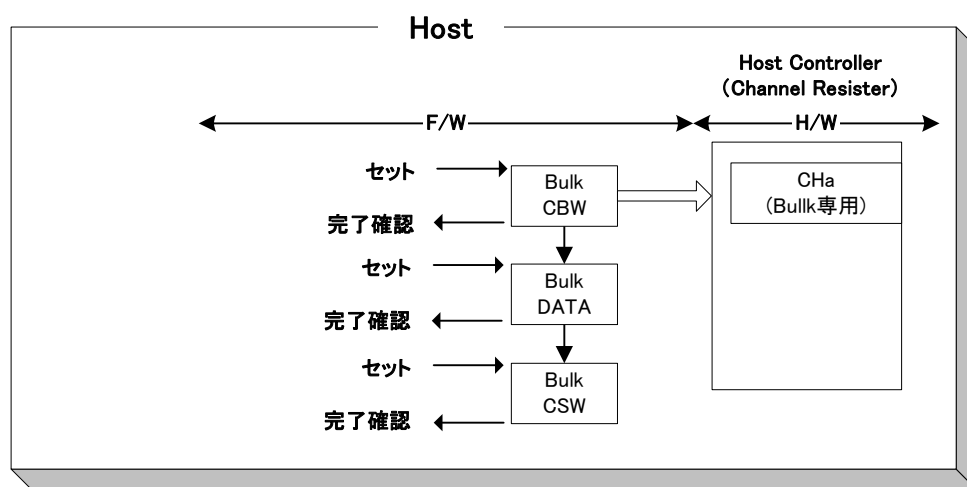


図 1-37 バルクオンリーサポート機能を使用しない場合の制御(参考)

バルクオンリーサポート機能はチャンネル CHa にて有効です。この機能を使用する場合はチャンネル CHa にジョインする FIFO 領域として AREA1 を選択してください。この機能を使用した場合のトランスポートの処理は以下のようになります。ファームウェアは(1)～(5)、(8)の処理を行います。

- (1) チャンネル CHa の以下の基本設定レジスタに適宜設定を行います。

転送速度(H\_CHaConfig\_0.SpeedMode)、

マックスパケットサイズ(H\_CHaMaxPktSize)、USB アドレス(H\_CHaFuncAdrs.FuncAdrs)、

FIFO 領域 (AREA1StartAdrs\_H,L,AREA1EndAdrs\_H,L) 、 FIFO 領域 ジョ イ ン (AREA1Join\_1.JoinEPaCHa)

- (2) バルクオンリーサポート機能の以下の制御レジスタに適宜設定を行います。

OUT エンドポイントトグルシーケンス(H\_CHaBO\_OUT\_EP\_Control.OUT\_Toggle)、

OUT エンドポイントナンバー(H\_CHaBO\_OUT\_EP\_Control.OUT\_EP\_Number)、

IN エンドポイントトグルシーケンス(H\_CHaBO\_IN\_EP\_Control.OUT\_Toggle)、

## 1. 機能説明

---

IN エンドポイントナンバー(H\_CHaBO\_IN\_EP\_Control.IN\_EP\_Number)

(3) FIFO の CBW エリアに CBW データ(31Byte)を書き込みます。

(4) CHa にジョインした FIFO 領域に DMA、または IDE の設定を行います。

(5) バルクオンリーサポート実行(H\_CHaBO\_SupportControl.BO\_SupportGo)をセットします。

この時、トランスポート状態(H\_CHaBO\_SupportControl.BO\_TransportState)の値は"Idle(00b)"として H\_CHaBO\_SupportControl レジスタにライトを行います。

(6) CBW エリアのデータ(31Byte)を、OUT エンドポイントナンバー(H\_CHaBO\_OUT\_EP\_Control.OUT\_EP\_Number)で示される OUT 方向のエンドポイントに対して、バルク OUT トランザクションによって送信します(コマンドトランスポート)。

(7) CBW データに基づき、データトランスポートを実行します。

- CBW データの bmCBWFlags の Bit7 が 0 の場合、CHa にジョインした FIFO 領域にあるデータを、OUT エンドポイントナンバー(H\_CHaBO\_OUT\_EP\_Control.OUT\_EP\_Number)で示される OUT 方向のエンドポイントに対して、バルク OUT トランザクションによって送信します(OUT 方向のデータトランスポート)。OUT 方向のデータトランスポートは CBW データの dCBWDataTransferLength で示されるデータ数を OUT トランザクションで送信すると終了します。
- CBW データの bmCBWFlags の Bit7 が 1 の場合、IN エンドポイントナンバー(H\_CHaBO\_IN\_EP\_Control.IN\_EP\_Number)で示される IN 方向のエンドポイントに対して、バルク IN トランザクションを発行し、受信したデータを CHa にジョインした FIFO 領域に書き込みます(IN 方向のデータトランスポート)。IN 方向のデータトランスポートは CBW データの dCBWDataTransferLength で示されるデータ数を IN トランザクションで受信すると終了します。また、IN トランザクションでショートパケットを受信した場合も IN 方向のデータトランスポートは終了します
- CBW データの dCBWDataTransferLength の値が 0x00000000 の場合はデータトランスポートは実行しません。

(8) CHa にジョインした FIFO 領域が CBW データの wCBWDataTransferLength で示される値より小さい場合、ファームウェアはデータトランスポートのデータを分割して処理する必要があります。

- データトランスポートが OUT 方向の場合、FIFO に送信するデータが無くなるとトランザクションは発行されなくなります。従ってファームウェアは FIFO の空き領域を確認しながら、順次残りの送信データを FIFO に書き込む必要があります。
- データトランスポートが IN 方向の場合、FIFO に空き領域がなくなるとトランザクションは発行されなくなります。従ってファームウェアは FIFO の有効データ数を確認しながら順次受信したデータを FIFO から読み出し、FIFO に空き領域を作る必要があります。

(9) IN エンドポイントナンバー(H\_CHaBO\_IN\_EP\_Control.IN\_EP\_Number)で示される IN 方向のエンドポイントに対して、バルク IN トランザクションを発行し、受信したデータを FIFO の CSW エリアに書き込みます(ステータストランスポート)。ステータストランスポートにおいて受信したデータ数はステータストランスポート受信データ数(H\_CHaBO\_CSW\_RcvDataSize.CSW\_RcvDataSize)に反映します。

- OUT 方向のデータトランスポートが終了した場合、ステータストランスポートを実行する状態になります。
- IN 方向のデータトランスポートが終了した場合、FIFO に存在する受信データがすべて読み出され FIFO が空になると、ステータストランスポートを実行する状態になります。

(10) ステータストランスポートで受信した CSW データをチェックします。チェック内容は以下の通りです。

- 受信した CSW のデータ長が 13Byte であること。
- CSW の dCSWSignature が 0x53425355 であること。
- CSW の dCSWTag が CBW の dCBWTag と一致すること。
- BCSWStatus の値が 0x00 であること。

以上を一つでも満たさない場合、バルクオンリーサポート実行(H\_CHaBO\_SupportControl.BO\_SupportGo)を自動クリアし、バルクオンリーサポート機能を停止し



ます。またバルクオンリーサポート停止ステータス(H\_CHaIntStat.BO\_SupportStop)を発行します。CSW エリアに受信したデータは、RAM\_Monitor 機能を用いて読み出すことが出来ます。

- (1 1) ステータストランスポートが正しく完結すると、バルクオンリーサポート実行(H\_CHaBO\_SupportControl.BO\_SupportGo)を自動クリアし、バルクオンリーサポート終了ステータス(H\_CHaIntStat.BO\_SupportCmp)を発行します。
- (1 2) それぞれのトランスポートにおいて、トランザクションのエラーを検出すると、バルクオンリーサポート実行(H\_CHaBO\_SupportControl.BO\_SupportGo)を自動クリアすることでバルクオンリーサポート機能を停止し、バルクオンリーサポート停止ステータス(H\_CHaIntStat.BO\_SupportStop)を発行します。そして、トランスポート状態(H\_CHaBO\_SupportControl.BO\_TransportState)により、エラーが発生したトランスポートを示します。また、コンディションコード(H\_CHaConditionCode)を適切な値に設定し、ChangeCondition ステータス(H\_CHaIntStat.ChangeCondition ビット)を発行します。

バルクオンリーサポート機能を中断する場合は、バルクオンリーサポート実行(H\_CHaBO\_SupportControl.BO\_SupportGo)をクリアします。バルクオンリーサポート機能の中断処理が終了するとステータスを発行します。

この時、トランスポートがステータストランスポートまで完了して中断処理が終了した場合はバルクオンリーサポート完了ステータス(CHaIntStat.BO\_SupportCmp)を発行します。

一方、トランスポートがステータストランスポートまで完了せずに中断処理が終了した場合はバルクオンリーサポート停止ステータス(H\_CHaIntStat.BO\_SupportStop)を発行します。

中断したトランスポートはトランスポート状態(H\_CHaBO\_SupportControl.BO\_TransportState)に示します。

中断したトランスポートよりバルクオンリーサポート機能を再開する場合、トランスポート状態(H\_CHaBO\_SupportControl.BO\_TransportState)を再開するトランスポートに設定(即ち中断したトランスポートに設定を保持)し、バルクオンリーサポート実行(H\_CHaBO\_SupportControl.BO\_SupportGo)をセットします。

一方、新たにバルクオンリーサポート機能を実行する場合は、トランスポート状態(H\_CHaBO\_SupportControl.BO\_TransportState)に Idle(00b)を設定し、バルクオンリーサポート実行(H\_CHaBO\_SupportControl.BO\_SupportGo)をセットします。

バルクオンリーサポート機能実行中は、転送実行ビット(H\_CHaConfig\_0.TranGo)、トグルシーケンスビット(H\_CHaConfig\_0.Toggle)、トランザクション種別(H\_CHaConfig\_1.TID)、トータルサイズフリービット(H\_CHaConfig\_1.TotalSizeFree)、エンドポイント番号(H\_CHaFuncAdrs.EP\_Number)、IRP データ数(H\_CHaTotalSize\_HH,HL,LH,LL)はハードウェアにより設定及び更新されていきますので、これらに対する書き込みは行わないで下さい。

トランザクションのエラーについては、「1.3.3.トランザクション」を参照して下さい。

FIFOのCBWエリア、CSWエリアについては、「1.6.FIFO管理」を参照して下さい。

DMAについては、「1.8.3.2.DMA0/DMA1(DMA ch.0 / ch.1)」を参照して下さい。

IDEについては、「1.10.IDE I/F」を参照して下さい。

バルクオンリーサポート機能の設定項目、ステータスを表 1-30 に示します。

## 1. 機能説明

表 1-30 バルクオンリーサポート機能の設定項目、ステータス

項目	レジスタ/ビット	説明
バルクオンリーサポート実行	H_CHaBO_SupportControl.BO_SupportGo	バルクオンリーサポート機能を実行します。 詳細については、「1.3.8.バルクオンリーサポート機能」を参照して下さい。
OUT エンドポイントトグルシーケンス	H_CHaBO_OUT_EP_Control.OUT_Toggle	OUT エンドポイントのトグルシーケンスビットの初期値を設定します。また、トランザクション実行中、およびトランザクション完了後はOUTエンドポイントのトグルシーケンスビットの状態を示します。
OUT エンドポイントナンバー	H_CHaBO_OUT_EP_Control.OUT_EP_Number	OUT エンドポイントのエンドポイントナンバーを 0x0～0xF の間の任意の値に設定します。
IN エンドポイントトグルシーケンス	H_CHaBO_IN_EP_Control.IN_Toggle	IN エンドポイントのトグルシーケンスビットの初期値を設定します。また、トランザクション実行中、およびトランザクション完了後はIN エンドポイントのトグルシーケンスビットの状態を示します。
IN エンドポイントナンバー	H_CHaBO_IN_EP_Control.IN_EP_Number	IN エンドポイントのエンドポイントナンバーを 0x0～0xF の間の任意の値に設定します。
バルクオンリーサポート実行結果	H_CHaIntStat.BO_SupportCmp H_CHaIntStat.BO_SupportStop	バルクオンリーサポートの実行結果を示します。
トランザクションステータス	H_CHaIntStat.TotalSizeCmp, H_CHaIntStat.TranACK, H_CHaIntStat.TranErr, H_CHaIntStat.ChangeCondition	トランザクションの結果を示します。
トランザクションコンディションコード	H_CHaConditionCode	トランザクション結果の詳細を示します。
トランスポート状態	H_CHaBO_SupportControl.BO_TransportState	バルクオンリーサポート機能の実行において、実行中のトランスポートを示します。また、トランスポートがエラーで中止された場合は、エラーが発生したトランスポートを示します。
ステータストラנסポート受信データ数	H_CSW_RcvDataSize	ステータストラנסポートでの受信データ数を示します。

### 1.3.9. オーディオクラス・アシスト機能

オーディオクラス・アシスト機能はサンプリング周波数 44.1kHz の 16bit 2 チャンネル PCM データをアイソクロナス転送で 1ms 周期で送信する際に使用します。この機能が有効の時、送信するデータパケットのサイズが 176Byte で 9 回連続トランザクションを行った後、180Byte で 1 回トランザクションを行います。その後再び 176Byte で 9 回連続トランザクションを行い、180Byte で 1 回トランザクションを行う、というデータパケットのサイズを自動で変更する動作を繰り返します。

このデータパケットサイズ変更のシーケンスは H\_CHx{x=b-e}Config\_1.Audio441 ビットが“1”である間保たれます。よってこのシーケンスを初期化する場合は H\_CHx{x=b-e}Config\_1.Audio441 ビットをクリアしてください。

表 1-31 オーディオクラスアシスト機能の制御項目

項目	レジスタ／ビット	説明
オーディオクラス アシスト機能	H_CHx{x=b-e}Config_1.Audio441	オーディオクラスアシスト機能を有効にします。OUT 方向のアイソクロナス転送の場合のみこの機能は使用できます。他の転送を行う場合はこのビットを“1”にセットしないで下さい。

## 1. 機能説明

### 1.3.10. ホストステート管理サポート機能

#### 1.3.10.1. ホストステート

ホストでは、上位からの要求およびバスの状態によりステートを遷移させる必要があります。そのため、ステート管理はファームウェアが行います。ハードウェアは各ステートにおける各種設定およびネゴシエーションをサポートします。

ホストステートの遷移図を図 1-38 に示します。

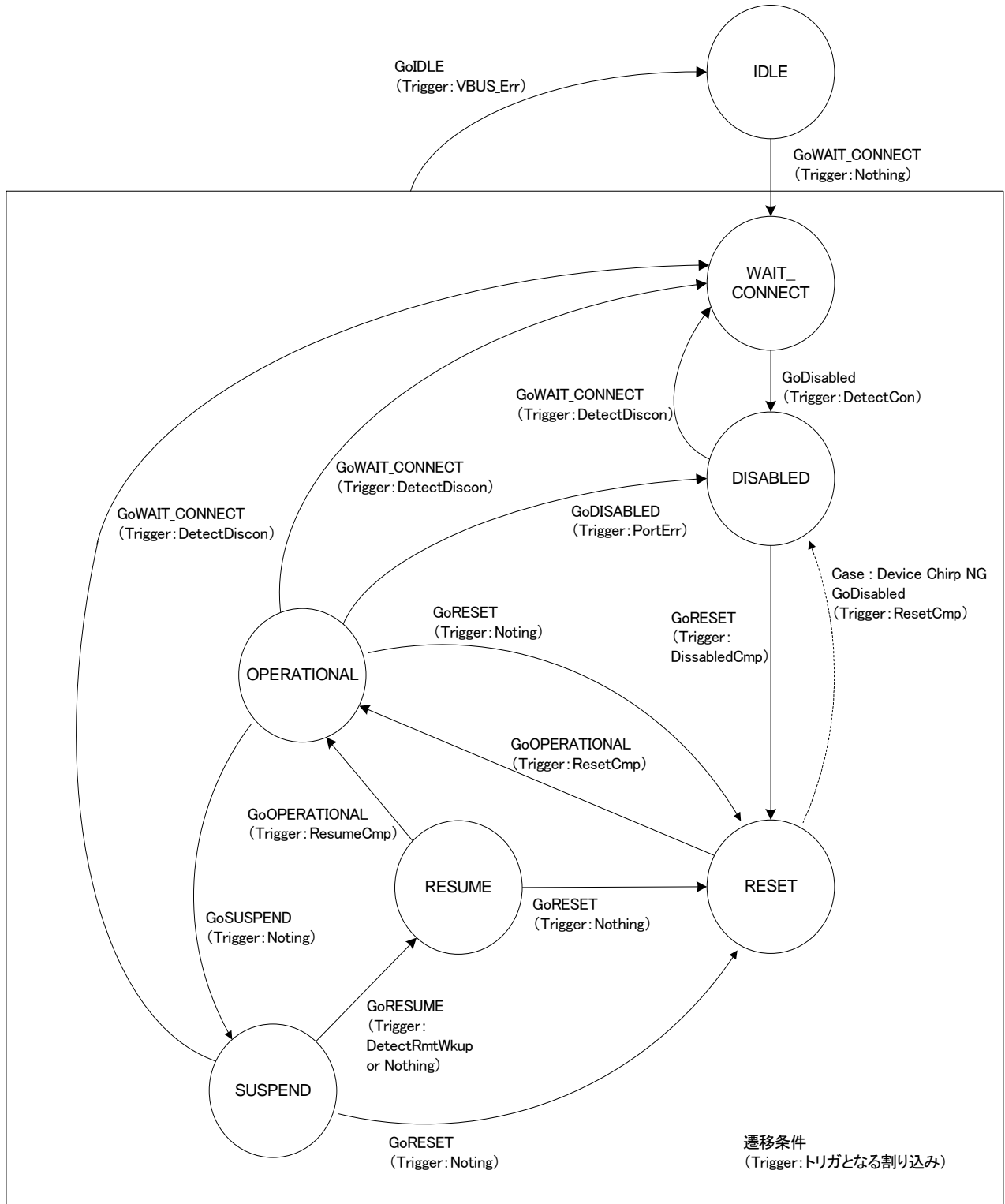


図 1-38 ホストステート遷移図

ホストステート管理サポート機能の設定項目およびステータスを表 1-32、表 1-33 に示します。

表 1-32 ホストステート管理サポート機能の設定項目およびステータス

項目	レジスタ／ビット	説明
ホストステート遷移実行	H_NegoControl_0.AutoMode	遷移させるホストステートを設定します。 設定は以下のいずれかの項目になります。 GoIDLE GoWAIT_CONNECT GoDISABLED GoRESET GoOPERATIONAL GoSUSPEND GoRESUME GoWAIT_CONNECTtoDIS GoWAIT_CONNECTtoOP GoRESETtoOP GoRESUMEtoOP GoSUSPENDtoOP
ホストステート遷移実行キャンセル	H_NegoControl_0.AutoModeCancel	現ホストステートでの処理を停止し、その状態で留まります。
ホストステートモニタ	H_NegoControl_0.HostState	現在のホストステート(下記)を示します。 IDLE WAIT_CONNECT DISABLED RESET OPERATIONAL SUSPEND RESUME
VBUS ステートモニタ	H_USB_Status.VBUS_State	VBUS の状態(正常／異常)を示します。
リモート・ウェークアップ受付許可	H_NegoControl_1.RmtWkupDetEnb	リモート・ウェークアップ受付を許可します。
チャープ完了ディセーブル	H_NegoControl_1.DisChirpFinish	デバイス Chirp が規定時間に完了しないときの動作モードを設定します。
VBUS 異常検出ステータス	H_SIE_IntStat_0.VBUS_Err	VBUS に異常が発生したことを示します。
接続検出ステータス	H_SIE_IntStat_0.DetectCon	ダウンストリームポートにデバイスが接続されたことを示します。
切断検出ステータス	H_SIE_IntStat_0.DetectDisCon	ダウンストリームポートからデバイスが切断されたことを示します。
リモート・ウェークアップ検出ステータス	H_SIE_IntStat_0.DetectRmtWkup	デバイスからのリモート・ウェークアップ信号を検出したことを示します。
デバイスチャープ正常検出ステータス	H_SIE_IntStat_0.DetectDevChirpOK	デバイスからのチャープ信号が正常であったことを示します。
デバイスチャープ異常検出ステータス	H_SIE_IntStat_0.DetectDevChirpNG	デバイスからのチャープ信号が異常であったことを示します。

(続く)

## 1. 機能説明

表 1-33 ホストステート管理サポート機能の設定項目およびステータス(続き)

項目	レジスタ/ビット	説明
リセット完了ステータス	H_SIE_IntStat_1.ResetCmp	USBリセットが正常に完了したことを示します。
サスペンド遷移完了ステータス	H_SIE_IntStat_1.SuspendCmp	サスペンドへの遷移が完了したことを示します。
レジューム完了ステータス	H_SIE_IntStat_1.ResumeCmp	レジュームが正常に完了したことを示します。
ポートスピード	H_NegoControl_1.PortSpeed	ダウンストリームポートの動作速度 (HS/FS/LS)を示します。ポートに接続されたデバイスの動作
ラインステート	H_USB_Status.LineState	USBケーブル上の信号状態を示します。
トランシーバ選択	H_XcvrControl.XcvrSelect	HS/FS/LSのいずれかのトランシーバを選択して有効にします。
ターミナル選択	H_XcvrControl.TermSelect	HS/FSのいずれかのターミナルを選択して有効にします。
オペレーションモード	H_XcvrControl.OpMode	HTMのオペレーションモードを設定します。

### 1.3.10.1.1. IDLE

USB ホスト機能を初期化するステートです。ホスト機能をイネーブルとしたときのデフォルトステートです。

本ステート以外のステートにおいて、VBUS\_Err を検出した時に、本ステートに遷移させる必要があります。

遷移を実行させるためには、H\_NegoControl\_0 レジスタに 0x80(H\_NegoControl\_0.AutoModeCancel に"1"、かつ H\_NegoControl\_0.AutoMode に"0x0")をライトして実行中ステートの動作を停止させます。H\_NegoControl\_0.AutoModeCancel ビットは停止処理が完了することで"0"になります(60MHz クロックで 6 サイクル程度必要)ので、H\_NegoControl\_0.AutoModeCancel ビットが"0"になるのを確認してから、同レジスタに"0x01"をライト(すなわちホストステート 遷移実行 (H\_NegoControl\_0.AutoMode)に "GoIDLE"を設定)してください。そうすることで本ステートに入ります。

本ステートでは、以下の設定を自動的に実行します。

- ・ USB ホストのトランザクション実行機能を即時停止する。
- ・ ポートを FS モードにして"NonDriving"とする。
- ・ VBUSEN\_A をオフする。
- ・ 接続検出、切断検出、リモート・ウェークアップ検出、デバイスチャープ検出の全検出機能をオフする。

### 1.3.10.1.2. WAIT\_CONNECT

ダウンストリームポートにデバイスが接続されるのを待つステートです。

"IDLE"ステートにおいては上位からの要求がある時に、また"DISABLED"、"OPERATIONAL"および"SUSPEND"の各ステートではデバイスの切断を検出した時に、一旦、本ステートに入り、相手の接続を待つ必要があります。

ホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoWAIT\_CONNECT"を設定した時に、本ステートに入ります。

本ステートでは、まず以下の設定を自動的に実行します。

- ・ USB ホストのトランザクション実行機能を即時停止する。
- ・ ポートを FS モードにして"PowerDown"とする。
- ・ VBUSEN\_A をオンする。
- ・ 接続検出、切断検出、リモート・ウェークアップ検出、デバイスチャープ検出の全検出機能をオ

フする。

次に、バスパワーデバイスの内部電源安定時間を待って、接続検出機能を自動的にオンにし、デバイスの接続を待ちます。このとき、VBUSEN\_A をオンしてからデバイスの接続検出までの時間は、ハードウェアでは管理しておりません。この時間は、必要に応じてファームウェアにて管理してください。

FS または HS デバイスが接続された場合、ラインステートが"J"として参照できます。一方、LS デバイスが接続された場合、ラインステートが"K"として参照できます。このいずれかの状態が 2.5us 以上継続されることでデバイスの接続を検出し、LS デバイスの接続を検出した場合には、ポートを LS モードにします。

接続が検出されると、切断検出機能を自動的にオンします。

その後、デバウンスインターバル期間に切断が検出されない場合、ファームウェアに対して接続検出ステータス(H\_SIE\_IntStat\_0.DetectCon)を発行し、接続検出機能および切断検出機能を自動的にオフします。一方、切断が検出された場合、切断検出機能を自動的にオフし、自動的に接続検出からやり直します。

### 1.3.10.1.3. DISABLED

ダウンストリームポートにデバイスが接続された状態で、バス上での信号送受信を行わないステートです。

"WAIT\_CONNECT"ステートで接続を検出した時、"RESET"ステートで異常なデバイスからの Chirp を検出した時、"OPERATIONAL"ステートでポートエラーを検出した時は、本ステートに入ります。

ホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoDISABLED"を設定した時に、本ステートに入ります。

本ステートでは、以下の設定を自動的に実行します。

- ・ 現トランザクションの完了を待って、USB ホストのトランザクション実行機能を停止する。
- ・ 本ステートに入る際に、HS モードであった場合はポートを FS モードにし、FS または LS モードの場合はポートは入る際のモードを持続する。
- ・ ポートを"PowerDown"とする。

次に、切断検出ディセーブル期間完了後に、以下の処理を自動的に実行します。

- ・ 切断検出機能をオンする。
- ・ ディセーブルド遷移完了ステータス(H\_SIE\_IntStat\_1.DisabledCmp)を発行する。

### 1.3.10.1.4. RESET

ダウンストリームポートに USB リセットを発行するステートです。

"DISABLED"ステートでディセーブルド遷移完了ステータスが発行された時は、本ステートに入り USB リセットを発行します。

また、上位から要求があった場合には、USB のいかなるステート("OPERATIONAL", "SUSPEND", "RESUME")からでも、本ステートに遷移することができます。

ホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoRESET"を設定した時に、本ステートに入ります。

本ステートでは、まず以下の設定を自動的に実行します。

- ・ 現トランザクションの完了を待って、USB ホストのトランザクション実行機能を停止する。
- ・ ポートを HS モードにして"NormalOperation"とする(USB ケーブル上の信号にリセット信号 SE0 をドライブします)。
- ・ 接続検出、切断検出およびリモート・ウェークアップ検出機能をオフする。
- ・ デバイスチャープ検出機能をオンする。

デバイスからの Chirp は"HS K"により検出します。ラインステートが"K"として 2.5us 以上継続することにより検出され、USB リセットの発行から規定時間以内に終了された場合に正常な Chirp として検出します。また、規定時間以内に終了されない場合には、異常な Chirp として検出します。

検出結果により以下の処理を自動的に行います。

(1) 正常なデバイスからの Chirp を検出した場合

デバイスからの Chirp 完了に伴い、ホストより"HS K"(Chirp K)、"HS J"(Chirp J)を交互に連続し

## 1. 機能説明

---

て送出します。ホストが Chirp の送出を完了した時点で、ファームウェアに対し、リセット完了ステータス(H\_SIE\_IntStat\_1.ResetCmp)を発行します。

ポートは HS モードのままです。

(2) 異常なデバイスからの Chirp を検出した場合

規定時間を過ぎた時点で、ファームウェアに対し、デバイスチャープ異常検出ステータス(H\_SIE\_IntStat\_0.DetectDevChirpNG)を発行します。その後の動作は、チャープ完了ディセーブル(H\_NegoControl\_1.DisChirpFinish)の設定により 2 つの動作モードが選択できます。詳細は「1.3.10.2.4.2.異常なデバイスのChirpを検出した場合」を参照してください。

(3) デバイスからの Chirp を検出せず、接続相手が FS の場合

規定時間の USB リセットを発行した後、ポートを FS モードにします。

ファームウェアに対し、リセット完了ステータス(H\_SIE\_IntStat\_1.ResetCmp)を発行します。

(4) 接続相手が LS の場合

規定時間の USB リセットを発行した後、ポートを LS モードにします。

ファームウェアに対し、リセット完了ステータス(H\_SIE\_IntStat\_1.ResetCmp)を発行します。

### 1.3.10.1.5. OPERATIONAL

USB のトランザクションを実行するステートです。

"RESET"または"RESUME"の完了後、本ステートに入りトランザクションを実行します。

ホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoOPERATIONAL"を設定した時に、本ステートに入ります。

本ステートでは、以下の設定を自動的に実行します。

- ・ ポートを"NormalOperation"とする。
- ・ USB ホストのトランザクション実行機能をイネーブルにする。
- ・ 切断検出機能をオンする。

### 1.3.10.1.6. SUSPEND

USB をサスペンドにするステートです。

USB のバスの使用を停止させるときに、"OPERATIONAL"から本ステートに遷移させます。

ホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoSUSPEND"を設定した時に、本ステートに入ります。

本ステートでは、まず以下の設定を自動的に実行します。

- ・ 切断検出機能およびリモート・ウェークアップ検出機能をオフする。
- ・ 現トランザクションの完了を待って、USB ホストのトランザクション実行機能を停止する。
- ・ ポートが HS モードであった場合、FS モードに設定する。一方、FS モードまたは LS モードであった場合は、そのモードを維持する。
- ・ ポートを"PowerDown"とする。

次に、切断およびリモート・ウェークアップ検出ディセーブル期間完了後に、以下の処理を自動的に実行します。

- ・ 切断検出機能をオンする。
- ・ リモート・ウェークアップ受付許可(H\_NegoControl\_1.RmtWkupDetEnb)がイネーブルの場合、リモート・ウェークアップ検出機能をオンする。
- ・ サスペンド遷移完了ステータス(H\_SIE\_IntStat\_1.SuspendCmp)を発行する。

さらに、リモート・ウェークアップ受付許可(H\_NegoControl\_1.RmtWkupDetEnb)がイネーブルの場合、リモート・ウェークアップ信号(2.5us 以上の"K"の継続)を検出することで、ファームウェアに対し、リモート・ウェークアップ検出ステータス(H\_SIE\_IntStat\_0.DetectRmtWkup)を発行します。

### 1.3.10.1.7. RESUME

ダウンストリームポートに USB レジューム信号を発行するステートです。

USB デバイスをサスペンド状態から復帰させるために、"SUSPEND"から本ステートに遷移させます。

ホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoRESUME"を設定した時に、本ステートに



入ります。

本ステートでは、切断およびリモート・ウェイクアップ検出機能を自動的にオフします。

そして、規定時間のレジューム信号("K")を発行します。

レジューム信号の発行完了時に、ポートを"SUSPEND"に入る前のモード設定に戻し、  
"NormalOperation"とします。

また、ファームウェアに対し、レジューム完了ステータス(H\_SIE\_IntStat\_1.ResumeCmp)を発行します。

## 1. 機能説明

### 1.3.10.2. 検出機能

#### 1.3.10.2.1. VBUS 異常検出

VBUSFLG\_A 入力端子のレベル変化(H→L)により VBUS の異常を検出します。VBUS 異常を検出した場合の実行手順は以下のとおりです。下記(2)は本 LSI のハードウェアが自動的に実行します。

- (1) VBUSFLG\_A(外付け USB パワースイッチのエラー発生フラグ)入力端子が L(エラー発生)になります(T0)。
- (2) ファームウェアに対し、VBUS 異常検出ステータス(HostIntStat.VBUS\_Err)を発行します(T0)。

なお、ホストは VBUS の異常を検出すると、直ちに VBUS をオフする必要があります。そのため、ファームウェアは、VBUS 異常検出ステータスを認識したら、H\_NegoControl\_0 レジスタに 0x80(H\_NegoControl\_0.AutoModeCancel に"1"、かつ H\_NegoControl\_0.AutoMode に"0x0")をライトして実行中ステートの動作を停止させます。H\_NegoControl\_0.AutoModeCancel ビットは停止処理が完了することで"0"になります(60MHz クロックで 6 サイクル程度必要)ので、H\_NegoControl\_0.AutoModeCancel ビットが"0"になるのを確認してから、同レジスタに 0x01 をライト(すなわちホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoIDLE"を設定)してください。それにより IDLE ステートに遷移し、VBUSEN\_A 端子論理がディセーブルとなり、VBUS をオフすることができます。

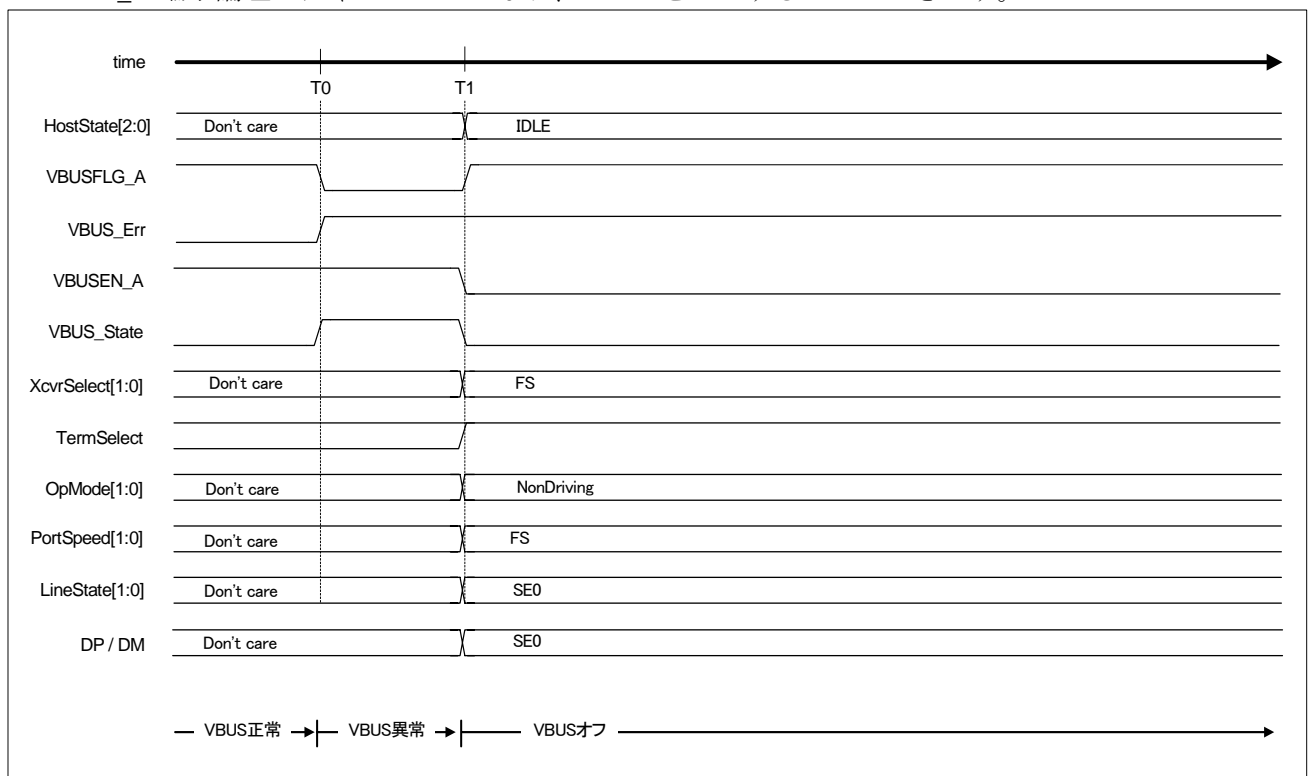


図 1-39 VBUS 異常検出 Timing

表 1-34 VBUS 異常検出 Timing Values

Timing Parameter	Description	Value
T0	VBUSFLG_A(外付け USB パワースイッチのエラー発生フラグ)入力端子が L(エラー)になる。 VBUS 異常検出ステータス(HostIntStat.VBUS_Err)を発行する。 (H/W)	0 (reference)
T1(参考)	H_NegoControl_0 に 0x80 をライト後、0x01 をライトして "IDLE"ステートに遷移させる。(F/W)	T1

### 1.3.10.2.2. 切断検出

デバイスの切断検出を"DISABLED"、"OPERATIONAL"、"SUSPEND"の各ステートにおいて行います。

切断が検出された時、VBUS をオフせず、接続検出からやり直す場合にはホストステートを"WAIT\_CONNECT"に遷移させてください。一方、VBUS をオフする場合にはホストステートを"IDLE"に遷移させてください。

#### 1.3.10.2.2.1. HS デバイスが切断された場合

HS デバイスの切断は、"OPERATIONAL"ステートにて検出します。

HS デバイスが切断された場合の実行手順は以下のとおりです。下記(2)～(3)は本 LSI のハードウェアが自動的に実行します。

- (1) デバイスが切断されます(T0)。
- (2) uSOF(HS\_SOF)の EOP 期間において切断検出を行い、3 回連続で切断を検出した場合に切断状態と判断します(T1)。
- (3) ファームウェアに対し、切断検出ステータス(H\_SIE\_IntStat\_0.DetectDiscon)を発行します(T1)。

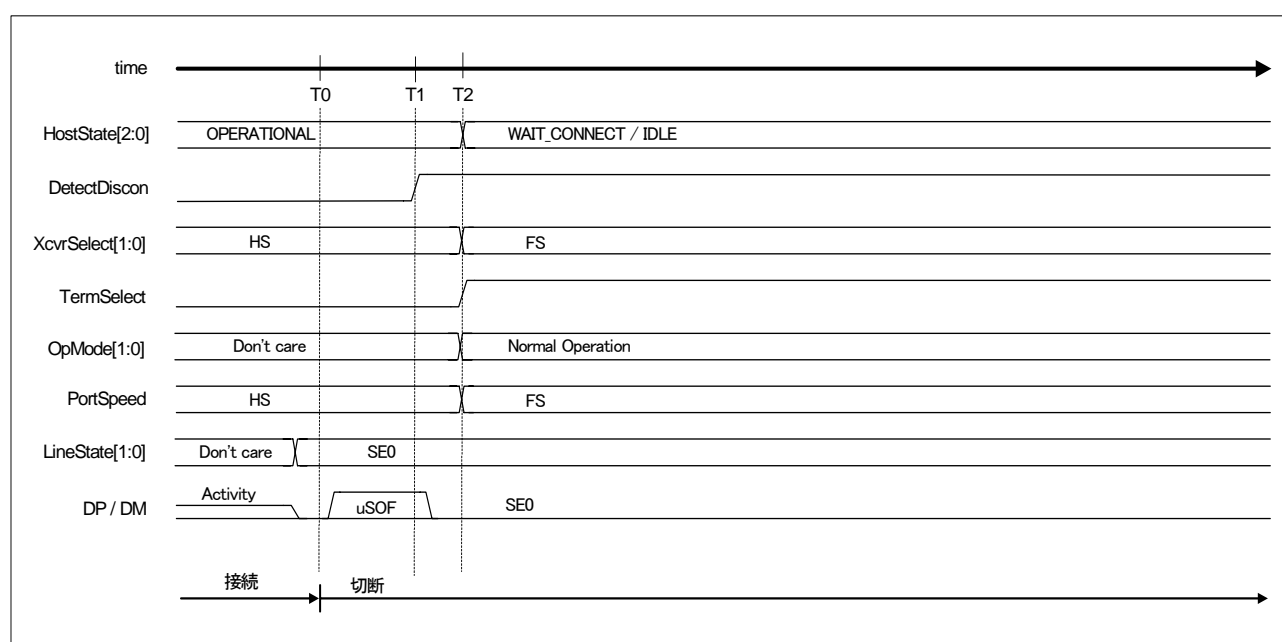


図 1-40 切断検出 Timing (HS mode)

表 1-35 切断検出 Timing Values (HS mode)

Timing Parameter	Description	Value
T0	デバイスが切断される。	0 (reference)
T1	切断検出ステータス(H_SIE_IntStat_0.DetectDiscon)を発行する。(H/W)	T1
T2(参考)	ホストステート遷移実行(H_NegoControl_0.AutoMode)に"GoWAIT_CONNECT"を設定する。(F/W)	T2

# 1. 機能説明

## 1.3.10.2.2.2. FS または LS デバイスが切断された場合

FS または LS デバイスの切断は、"DISABLED"、"OPERATIONAL"、"SUSPEND"ステートにて検出します。

FS または LS デバイスが切断された場合の実行手順は以下のとおりです。下記(2)～(3)は本 LSI のハードウェアが自動的に実行します。

- (1) デバイスが切断されます(T0)。
- (2) 信号線の状態から切断を検出します(T1)。
- (3) ファームウェアに対し、切断検出ステータス(H\_SIE\_IntStat\_0.DetectDiscon)を発行します(T1)。

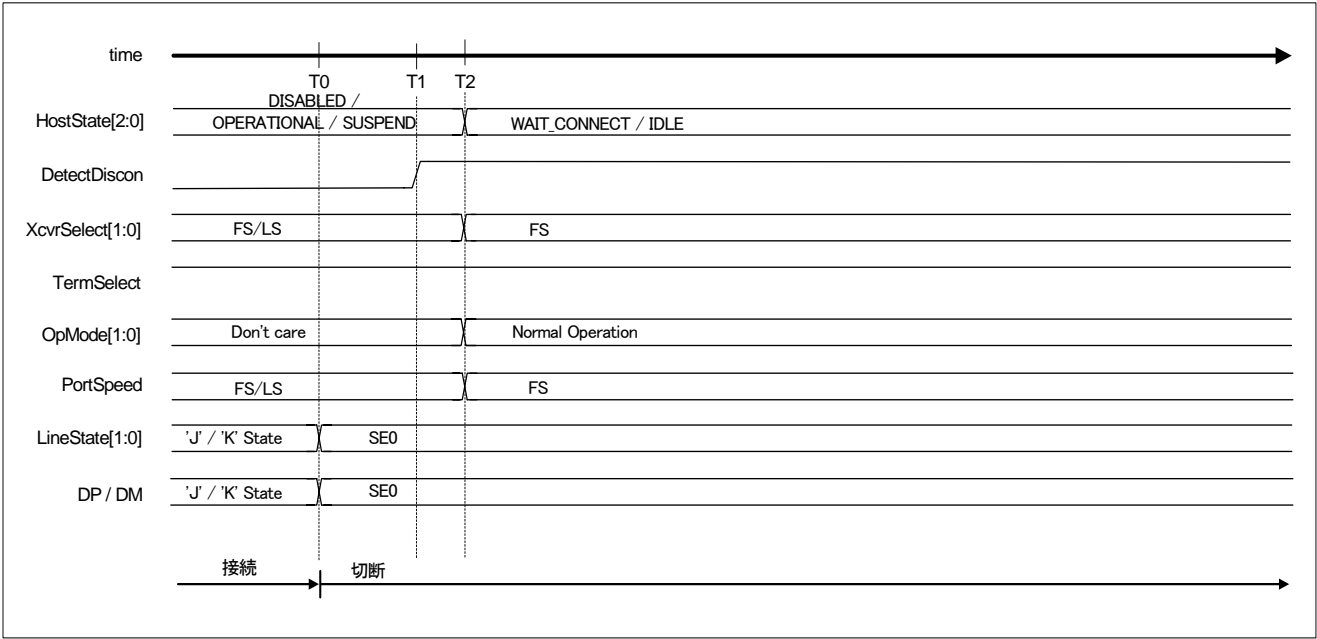


図 1-41 切断検出 Timing (FS または LS mode)

表 1-36 切断検出 Timing Values (FS または LS mode)

Timing Parameter	Description	Value
T0	デバイスが切断される。	0 (reference)
T1	切断検出ステータス(H_SIE_IntStat_0.DetectDiscon)を発行する。(H/W)	$T0 + 2.5\mu s < T1 \{T_{DDIS}\}$
T2(参考)	ホストステート遷移実行(H_NegoControl_0.AutoMode)に "GoWAIT_CONNECT"を設定する。(F/W)	規定なし

### 1.3.10.2.3. リモート・ウェークアップ検出

リモート・ウェークアップ受付許可(H\_NegoControl\_1.RmtWkupDetEnb)がイネーブルの場合、"SUSPEND"ステートにおいてリモート・ウェークアップ検出を行います。

#### 1.3.10.2.3.1. HS デバイスが接続されている場合

HS デバイスが接続されている場合の実行手順は以下のとおりです。下記(2)～(3)は本 LSI のハードウェアが自動的に実行します。

- (1) デバイスがリモート・ウェークアップ信号(K)を送出開始します(T0)。
- (2) ホストがリモート・ウェークアップ信号(K)を検出します(T1)。
- (3) ファームウェアに対し、リモート・ウェークアップ検出ステータス(H\_SIE\_IntStat\_0.DetectRmtWkup)を発行します(T1)。

なお、ホストはデバイスのリモート・ウェークアップ検出から 1ms 以内にレジューム信号("K")を発行する必要があります。そのため、ファームウェアは、リモート・ウェークアップ検出ステータスを直ちに認識し、ホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoRESUME"を 900us 以内に設定してください。

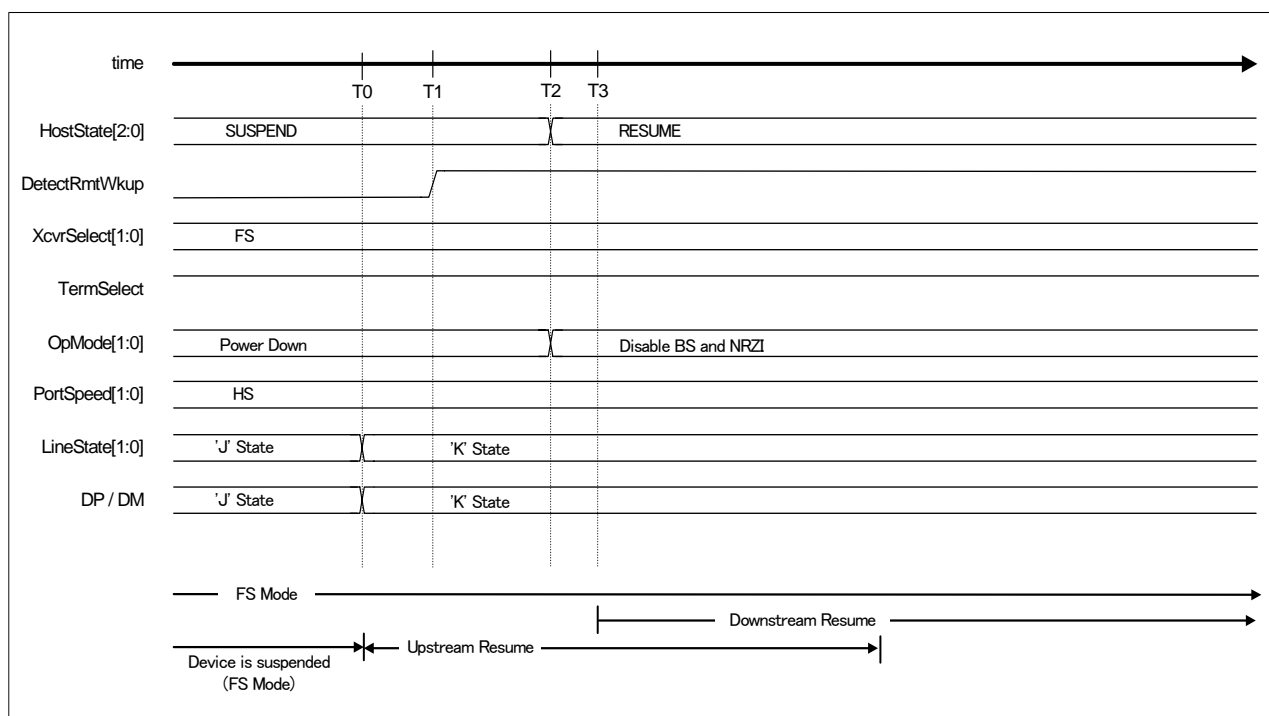


図 1-42 Remote Wakeup Timing (HS mode)

## 1. 機能説明

---

表 1-37 Remote Wakeup Timing Values (HS mode)

Timing Parameter	Description	Value
T0	デバイスがリモート・ウェークアップ信号(K)を送出開始する。	0 (reference)
T1	リモート・ウェークアップ信号(K)を検出する。 リモート・ウェークアップ検出ステータスを発行する。 (H/W)	$T0 + 2.5\mu s\{T_{URLK}\} < T1$
T2(参考)	ホストステート遷移実行(H_NegoControl_0.AutoMode)に "GoRESUME"を設定する。(F/W)	$T2 < T1 + 900\mu s$
T3(参考)	ホストがレジューム信号("K")の発行を開始する。(H/W)	$T3 < T0 + 1ms\{T_{URSM}\}$

注： {} は、USB2.0 規格書で規格されている名称である。

### 1.3.10.2.3.2. FS デバイスが接続されている場合

FS デバイスが接続されている場合の実行手順は、HS デバイスが接続されている場合の実行手順と同じです。

実行手順は、「1.3.10.2.3.1.HSデバイスが接続されている場合」を参照してください。

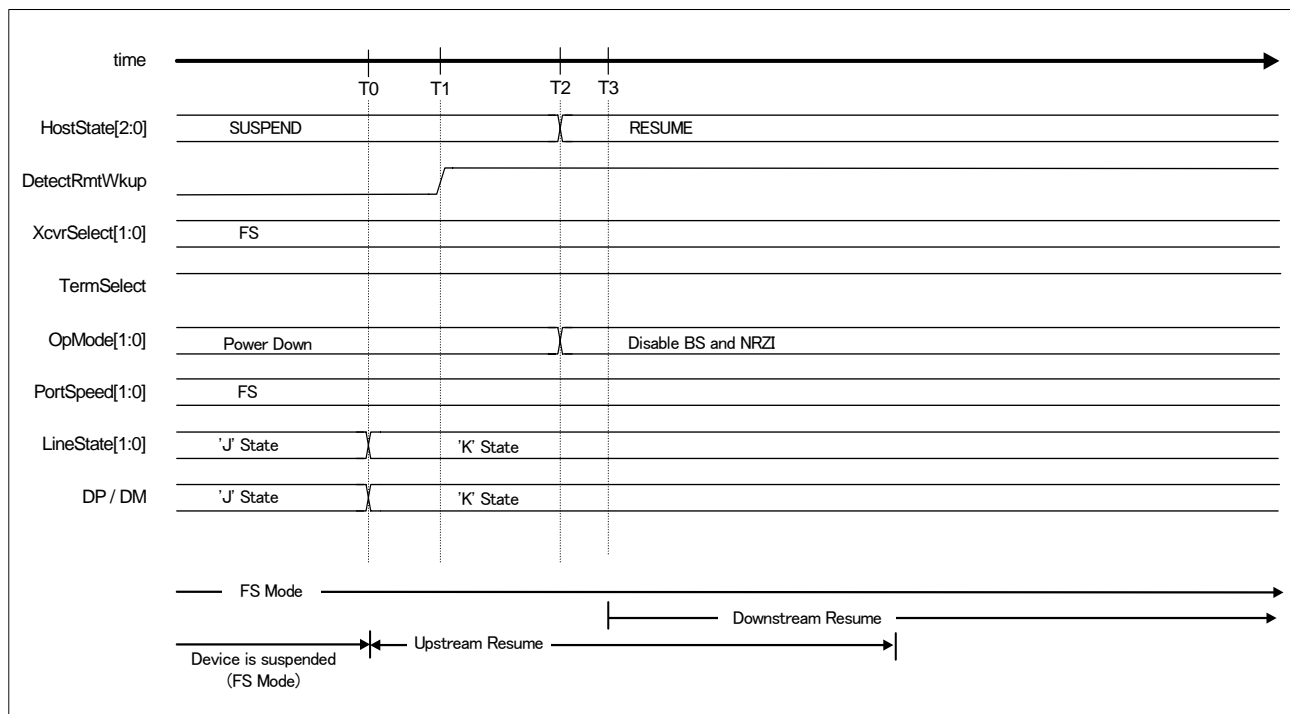


図 1-43 Remote Wakeup Timing (FS mode)

表 1-38 Remote Wakeup Timing Values (FS mode)

Timing Parameter	Description	Value
T0	デバイスがリモート・ウェークアップ信号(K)を送出開始する。	0 (reference)
T1	リモート・ウェークアップ信号(K)を検出する。 リモート・ウェークアップ検出ステータスを発行する。 (H/W)	$T0 + 2.5\mu s < T1 \{T_{URLK}\}$
T2(参考)	ホストステート遷移実行(H_NegoControl_0.AutoMode)に "GoRESUME"を設定する。(F/W)	$T2 < T1 + 900\mu s$
T3(参考)	ホストがレジューム信号("K")の発行を開始する。(H/W)	$T3 < T0 + 1ms \{T_{URSM}\}$

注： {} は、USB2.0 規格書で規格されている名称である。

# 1. 機能説明

## 1.3.10.2.3.3. LS デバイスが接続されている場合

LS デバイスが接続されている場合の実行手順は、HS デバイスが接続されている場合の実行手順と同じです。

実行手順は、「1.3.10.2.3.1.HSデバイスが接続されている場合」を参照してください。

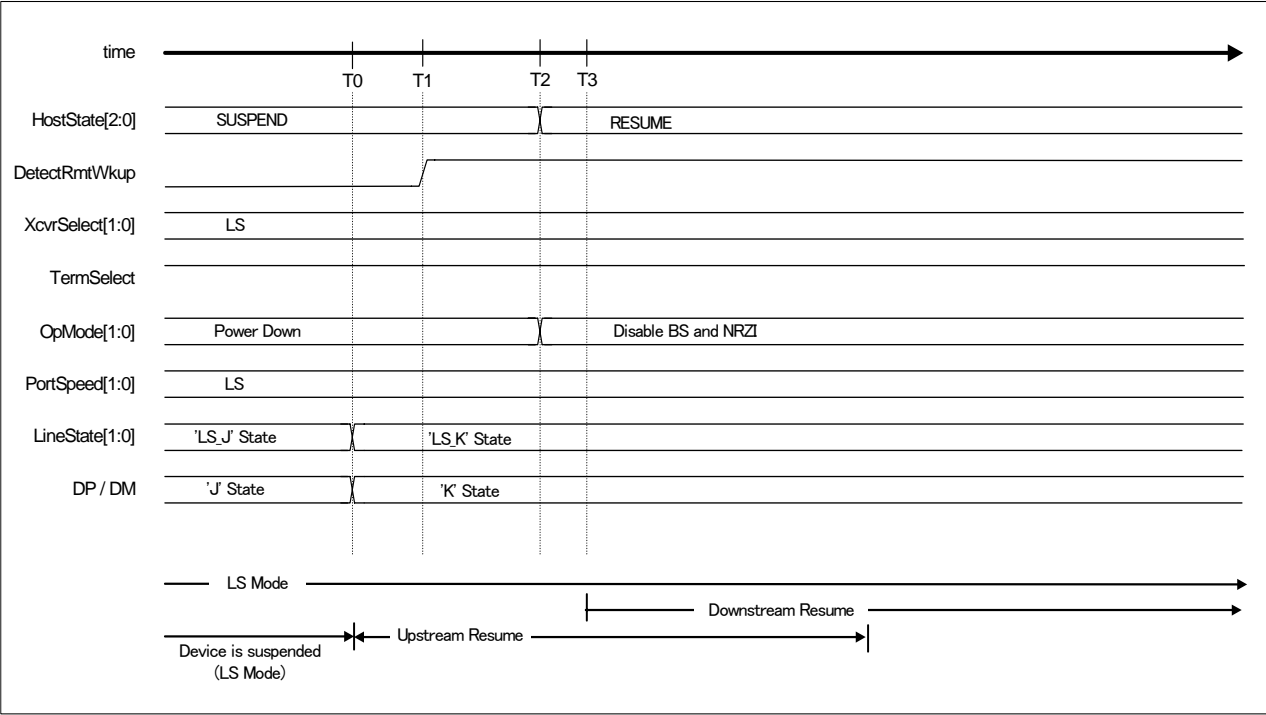


図 1-44 Remote Wakeup Timing (LS mode)

表 1-39 Remote Wakeup Timing Values (LS mode)

Timing Parameter	Description	Value
T0	デバイスがリモート・ウェークアップ信号("K")を送出開始する。	0 (reference)
T1	リモート・ウェークアップ信号("K")を検出する。 リモート・ウェークアップ検出ステータスを発行する。 (H/W)	$T0 + 2.5\mu s < T1 \{T_{URLK}\}$
T2(参考)	ホストステート遷移実行(H_NegoControl_0.AutoMode)に "GoRESUME"を設定する。(F/W)	$T2 < T1 + 900\mu s$
T3(参考)	ホストがレジューム信号("K")の発行を開始する。(H/W)	$T3 < T0 + 1ms \{T_{URSM}\}$

注： {} は、USB2.0 規格書で規格されている名称である。



### 1.3.10.2.4. デバイスチャープ検出機能

デバイスの Chirp を検出します。

デバイスチャープ検出機能は"RESET"ステートにてオンします。

#### 1.3.10.2.4.1. 正しいデバイスの Chirp を検出した場合

デバイスの Chirp を検出する手順は以下の通りです。

- (1) ファームウェアがホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoRESET"を設定します(T0)。
- (2) デバイスチャープ検出機能をオンします(T0)。
- (3) デバイスが Chirp を送出します(T1)。
- (4) ラインステート(H\_USB\_Status.LineState[1:0])に"K"が規定時間以上継続することでデバイスの Chirp を認識します(T2)。
- (5) デバイスの Chirp がリセット開始から規定時間以内に終了する(ラインステート(H\_USB\_Status.LineState[1:0])が"SE0"になる)ことをもってデバイスチャープ正常検出ステータス(H\_SIE\_IntStat0.DetectDevChirpOK)を発行します(T3)。
- (6) デバイスの Chirp を検出するとデバイスチャープ検出機能をオフします(T3)。

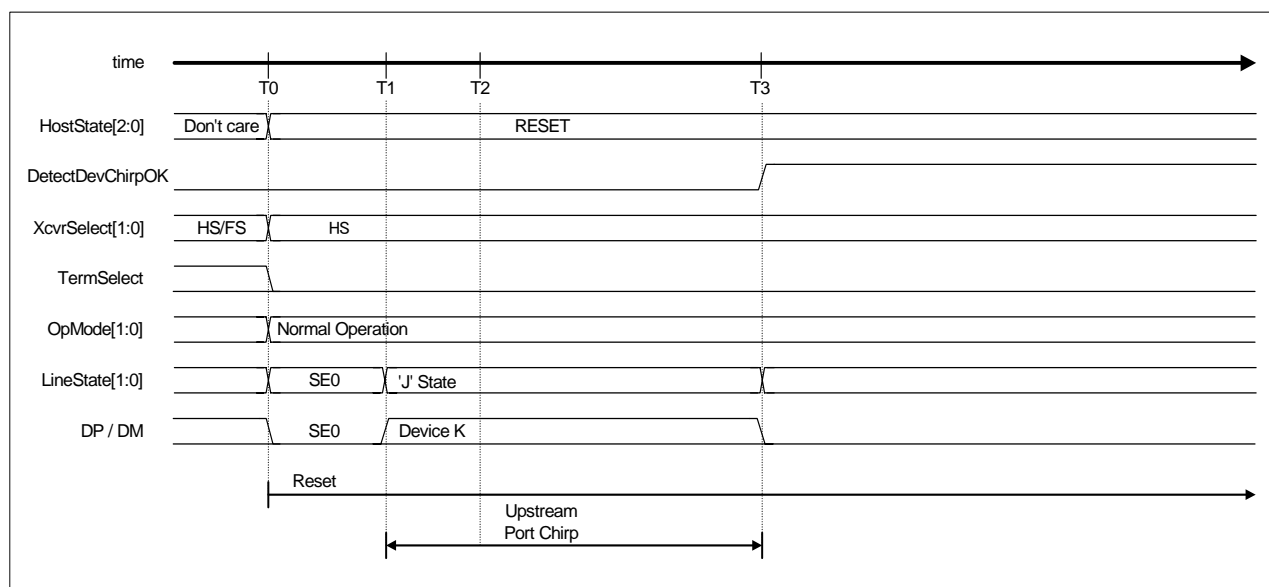


図 1-45 Device Chirp Timing

表 1-40 Device Chirp Timing Values

Timing Parameter	Description	Value
T0	H_NegoControl_0.AutoMode に"GoRESET"を設定する。(F/W)	0 (reference)
T1	デバイスが Chirp を開始。	$T0 < T1 < T0 + 6.0\text{ms}$
T2	デバイスの Chirp を認識する。(H/W)	$T1 + 2.5\mu\text{s} \{T_{\text{FILT}}\} < T2$
T3	デバイスが Chirp を終了。 デバイスチャープ検出機能をオフする。 デバイスチャープ正常検出ステータス (DetectDevChirpOK)を発行する。(H/W)	$T1 + 1.0\text{ms} \{T_{\text{UCH}}\} < T3 < T0 + 7.0\text{ms} \{T_{\text{UCHEND}}\}$

注： {} は、USB2.0 規格書で規格されている名称である。

## 1. 機能説明

### 1.3.10.2.4.2. 異常なデバイスの Chirp を検出した場合

デバイスチャープ検出機能は、デバイスの Chirp が規定時間に終了しない場合を異常と見なし、ステータスを発行します。

- (1) ファームウェアがホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoRESET"を設定します(T0)。
- (2) デバイスチャープ検出機能をオンします(T0)。
- (3) デバイスが Chirp を送出します(T1)。
- (4) ラインステート(H\_USB\_Status.LineState[1:0])に"K"が規定時間以上継続することでデバイスの Chirp を認識します(T2)。
- (5) リセットの開始から規定時間以内にデバイスの Chirp が終了しないために異常と見なし、デバイスチャープ異常検出ステータス(H\_SIE\_IntStat\_0.DetectDevChirpNG)を発行します(T3)
- (6) デバイスチャープ検出機能をオフします(T3)。

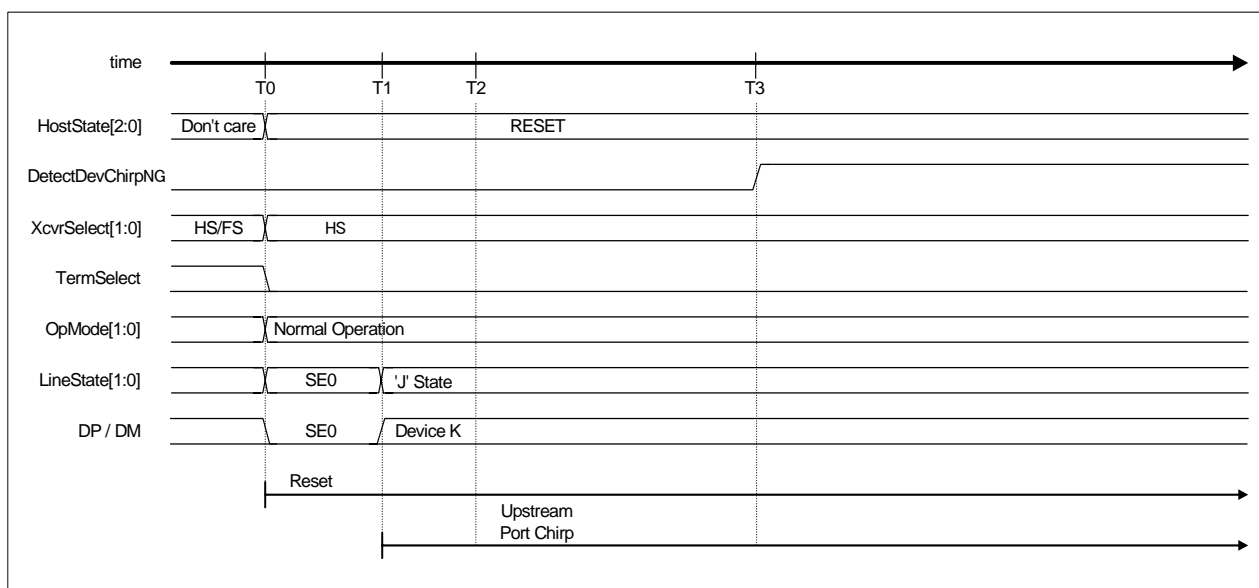


図 1-46 Device Chirp Timing(NG)

表 1-41 Device Chirp Timing(NG) Value

Timing Parameter	Description	Value
T0	H_NegoControl_0.AutoMode に"GoRESET"を設定する。(F/W)	0 (reference)
T1	デバイスが Chirp を開始。	$T0 < T1 < T0 + 6.0\text{ms}$
T2	デバイスの Chirp を認識する。(H/W)	$T1 + 2.5\mu\text{s} \{T_{\text{FILT}}\} < T2$
T3	デバイスチャープ異常検出ステータス (DetectDevChirpNG)を発行する。(H/W)	$T0 + 7\text{ms} \{T_{\text{UCHEND}}\} < T3$

注： {} は、USB2.0 規格書で規格されている名称である。

#### 1.3.10.2.5. ポートエラー検出

"OPERATIONAL"ステートにおいてポートエラーを検出します。

受信中のパケットにおいて、(マイクロ)フレームの終端に達しても EOP を検出できない場合に、ポートエラーと判断します。

ホストはポートエラーを検出すると、ファームウェアに対してポートエラー検出ステータス(H\_FrameIntStat.PortErr)を発行し、トランザクションを即時停止します。以降、SOF を含めトランザクションの発行は行いません。

ポートエラーが発生した場合、ファームウェアは以下の処理を行ってください。

- (1) H\_NegoControl\_0.AutoMode に"GoDISABLED"を設定します。
- (2) H\_NegoControl\_0.ResetHTM に"1"を設定し、ホストトランシーバマクロをリセットします。
- (3) 60MHz クロックで 3 サイクル以上の時間経過後、H\_NegoControl\_0.ResetHTM に"0"を設定し、ホストトランシーバマクロのリセットを解除します。

## 1. 機能説明

### 1.3.10.3. ホストステート管理サポート機能の単体説明

#### 1.3.10.3.1. GoIDLE

H\_NegoControl\_0 レジスタに 0x80(H\_NegoControl\_0.AutoModeCancel に "1"、かつ H\_NegoControl\_0.AutoMode に 0x0)をライトして実行中ステートの動作を停止させます。H\_NegoControl\_0.AutoModeCancel ビットは停止処理が完了することで"0"になります(60MHz クロックで 6 サイクル程度必要)ので、H\_NegoControl\_0.AutoModeCancel ビットが"0"になるのを確認してから、同レジスタに 0x01 をライト(すなわちホストステート遷移実行(H\_NegoControl\_0.AutoMode)に GoIDLE を設定)してください。そうすることで、"IDLE"遷移に必要な処理を本 LSI のハードウェアが自動的に実行します。

下記(3)～(8)は本 LSI のハードウェアが自動的に実行します。

- (1) H\_NegoControl\_0.AutoModeCancel に 0x80(H\_NegoControl\_0.AutoModeCancel に "1"、かつ H\_NegoControl\_0.AutoMode に 0x0)をライトします(T0)。
- (2) H\_NegoControl\_0.AutoModeCancel ビットが"0"になったことを確認し、H\_NegoControl\_0 に 0x01(H\_NegoControl\_0.AutoMode に 0x1)をライトします(T1)。
- (3) ホストステートモニタ(H\_NegoControl\_0.HostState)を"IDLE"に設定します(T1)。
- (4) VBUSEN\_A をオフします(T1)。
- (5) トランシーバ選択(H\_XcvrControl.XcvrSelect)とターミナル選択(H\_XcvrControl.TermSelect)を FS モードに設定します(T1)。
- (6) オペレーションモード(H\_XcvrControl.OpMode[1:0])を"NonDriving"に設定します(T1)。
- (7) USB ホストのトランザクション実行機能を即時停止します(T1)。
- (8) 接続検出、切断検出、リモート・ウェークアップ検出、デバイスチャープ検出の全検出機能をオフします(T1)。

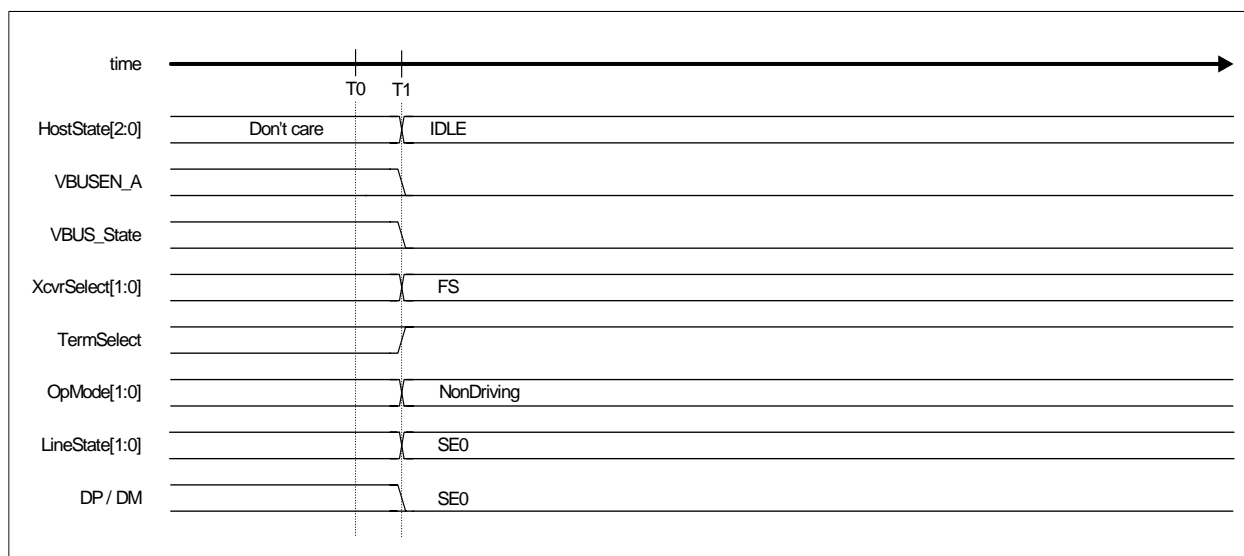


図 1-47 GoIDLE Timing

表 1-42 GoIDLE Timing Value

Timing Parameter	Description	Value
T0	実行中ステートの動作を停止する。(F/W)	0 (reference)
T1	ホストステート遷移実行キャンセルビットが”0”になったことを確認し、H_NegoControl_0.AutoMode に "GoIDLE"を設定する。(F/W) VBUSEN_A をオフする。 トランシーバ選択を FS モードに設定する。 ターミナル選択を FS モードに設定する。 オペレーションモードを”NonDrivinvg”に設定する。 トランザクション実行機能を即時停止する。 接続検出、切断検出、リモート・ウェークアップ検出、デバイスチャープ検出をオフする。(H/W)	$T0 + 5\text{cycle}(60\text{MHz}) < T1$

## 1. 機能説明

### 1.3.10.3.2. GoWAIT\_CONNECT

ホストステート遷移実行(H\_NegoControl0.AutoMode)に "GoWAIT\_CONNECT" を設定すると、 "WAIT\_CONNECT"遷移に必要な処理を本 LSI のハードウェアが自動的に実行します。

なお、HS デバイスはこの時点では FS デバイスとして接続します。その後に行われるリセット動作の HS Detection Handshake によって HS デバイスとして動作するようになります。

#### 1.3.10.3.2.1. FS デバイスが接続された場合

FS デバイスが接続された場合の実行手順は以下のとおりです。下記(2)～(12)は本 LSI のハードウェアが自動的に実行します。

- (1) ファームウェアがホストステート遷移実行(H\_NegoControl\_0.AutoMode)に "GoWAIT\_CONNECT"を設定します(T0)。
- (2) ホストステートモニタ(H\_NegoControl\_0.HostState)を"WAIT\_CONNECT"に設定します(T0)。
- (3) VBUSEN\_A をオンします(T0)。
- (4) トランシーバ選択(H\_XcvrControl.XcvrSelect)とターミナル選択(H\_XcvrControl.TermSelect)を FS モードに設定します(T0)。
- (5) オペレーションモード(H\_XcvrControl.OpMode[1:0])を"PowerDown"に設定します(T0)。
- (6) ポートスピード(H\_NegoControl\_1.PortSpeed[1:0])を"FS"に設定します(T0)。
- (7) デバイス内部電源の安定化の為に一定期間待ち、接続検出機能をオンします(T1)。
- (8) FS デバイスが接続されるとラインステート(H\_USB\_Status.LineState[1:0])に"J"が現れます(T2)。
- (9) ラインステート(H\_USB\_Status.LineState[1:0])の"J"が 2.5us 以上継続することをもって FS デバイスの接続と見なします(T3)。
- (10) 切断検出機能をオンします(T3)。
- (11) デバウンスインターバル期間中に切断が検出されない場合に接続検出ステータス(H\_SIE\_IntStat\_0.DetectCon)を発行します(T4)。もし、この期間中に切断が検出された場合には、切断検出機能をオフにし、(8)に戻っての接続検出からやり直します。切断検出ステータス(H\_SIE\_IntStat\_0.DetectDiscon)は発行しません。
- (12) 切断検出機能と接続検出機能をオフします(T4)。

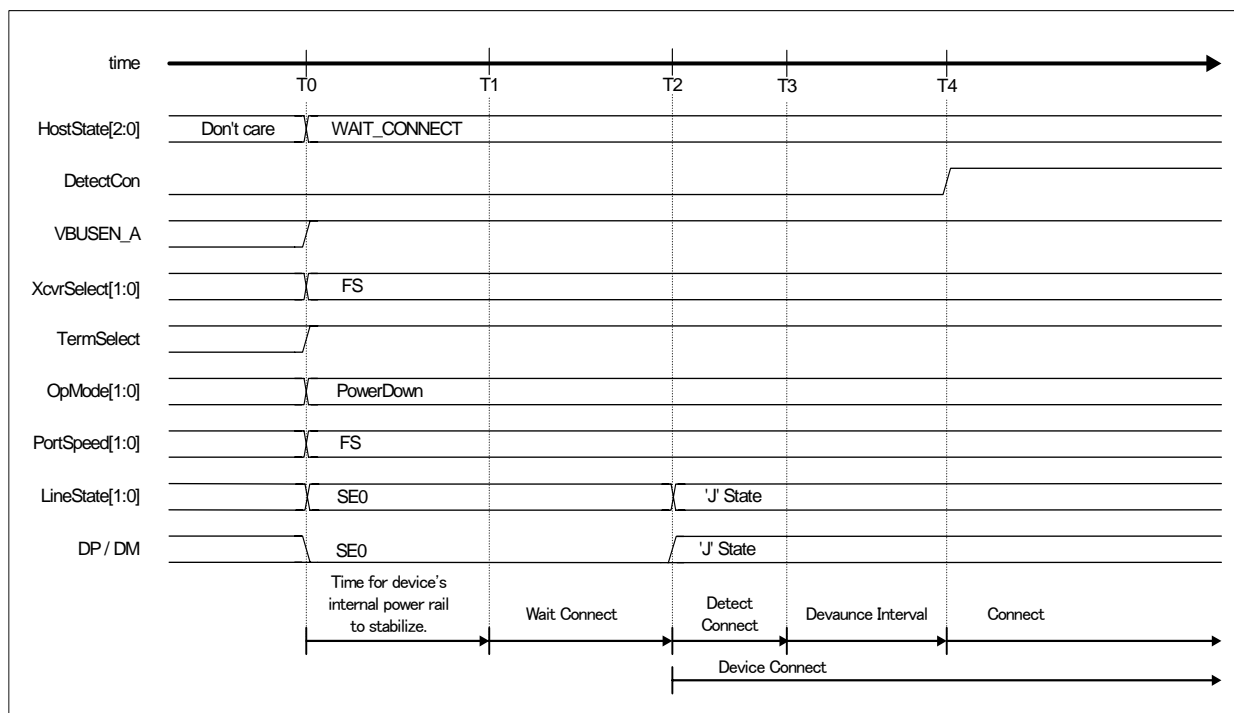


図 1-48 Device Attach Timing(FS mode)

表 1-43 Device Attach Timing Values(FS mode)

Timing Parameter	Description	Value
T0	H_NegoControl_0.AutoMode に"GoWAIT_CONNECT"を設定する。(F/W)	0 (reference)
T1	接続検出機能をオンする。(H/W)	$T0 + 100\text{ms}\{T_{\text{SIGATT}}\} < T1$
T2	デバイスが接続される。	T2
T3	切断検出機能をオンする。(H/W)	$T2 + 2.5\mu\text{s}\{T_{\text{DCNN}}\} < T3$
T4	接続検出ステータス(DetectCon)を発行する。 切断検出機能と接続検出機能をオフする。(H/W)	$T3 + 100\text{ms}\{T_{\text{ATTDB}}\} < T4$

注： {} は、USB2.0 規格書で規格されている名称である。

## 1. 機能説明

### 1.3.10.3.2.2. LS デバイスが接続された場合

LS デバイスが接続された場合の実行手順は以下のとおりです。下記(2)～(14)は本 LSI のハードウェアが自動的に実行します。

- (1) ファームウェアがホストステート遷移実行 (H\_NegoControl\_0.AutoMode) に "GoWAIT\_CONNECT"を設定します(T0)。
- (2) ホストステートモニタ(H\_NegoControl\_0.HostState)を "WAIT\_CONNECT"に設定します(T0)。
- (3) VBUSEN\_A をオンします(T0)。
- (4) トランシーバ選択(H\_XcvrControl.XcvrSelect)とターミナル選択(H\_XcvrControl.TermSelect)を FS モードに設定します(T0)。
- (5) オペレーションモード(H\_XcvrControl.OpMode[1:0])を "PowerDown"に設定します(T0)。
- (6) ポートスピード(H\_NegoControl\_1.PortSpeed[1:0])を "FS"に設定します(T0)。
- (7) デバイス内部電源の安定化の為に 100ms 待ち、接続検出機能をオンします(T1)。
- (8) LS デバイスが接続されるとラインステート(H\_USB\_Status.LineState[1:0])に "K"が現れます(T2)。
- (9) ラインステート(H\_USB\_Status.LineState[1:0])の "K"が 2.5us 以上継続することをもって LS デバイスの接続と見なします(T3)。
- (10) トランシーバ選択(H\_XcvrControl.XcvrSelect[1:0])を "LS"に設定します(T3)。このため、ラインステート (H\_USB\_Status.LineState[1:0]) の極性が LS になり、ラインステート (H\_USB\_Status.LineState[1:0])には "J"が現れます。
- (11) ポートスピード(H\_NegoControl\_1.PortSpeed[1:0])を "LS"に設定します(T3)。
- (12) 切断検出機能をオンします(T3)。
- (13) デバウンスインターバル期間中に切断が検出されない場合に接続検出ステータス (H\_SIE\_IntStat\_0.DetectCon)を発行します(T4)。もし、この期間中に切断が検出された場合には、切断検出機能をオフにし、トランシーバ選択 (H\_XcvrControl.XcvrSelect)とポートスピード (H\_NegoControl\_1.PortSpeed[1:0])をとともに "FS"に設定し、(8)に戻っての接続検出からやり直します。切断検出ステータス(H\_SIE\_IntStat\_0.DetectDiscon)は発行しません。
- (14) 切断検出機能と接続検出機能をオフします(T4)。

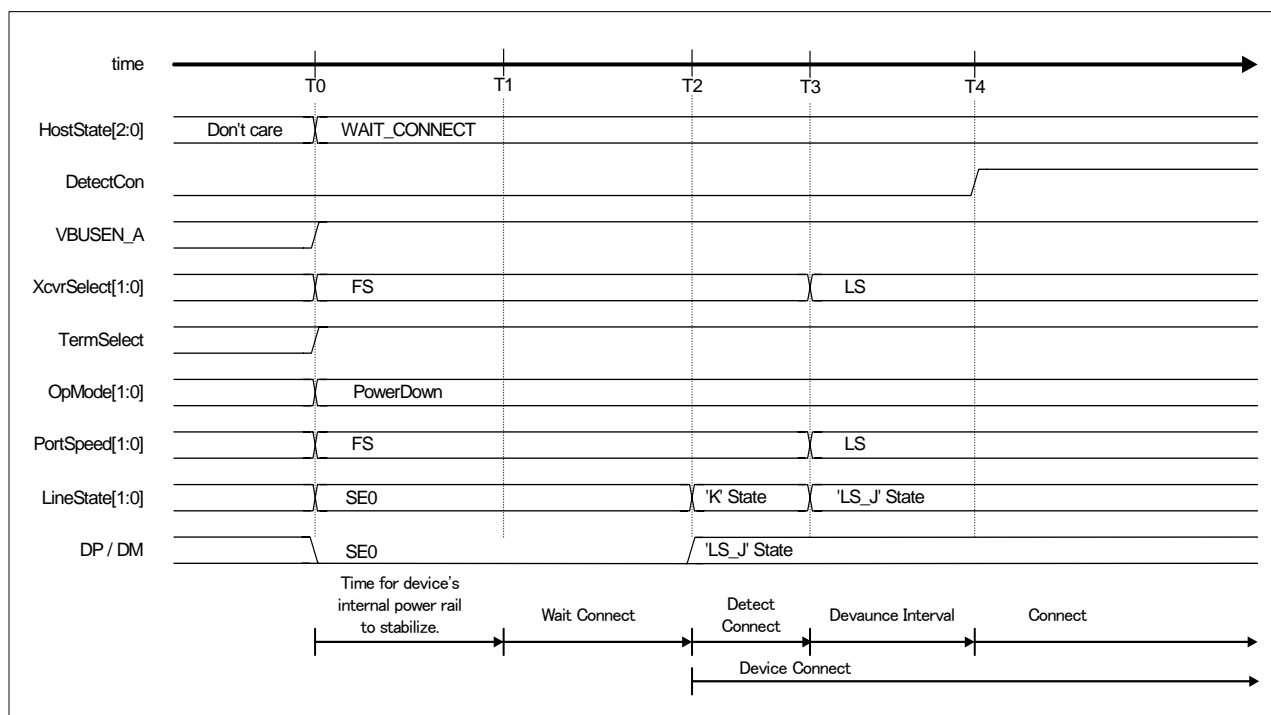


図 1-49 Device Attach Timing(LS mode)



表 1-44 Device Attach Timing Values(LS mode)

Timing Parameter	Description	Value
T0	H_NegoControl_0.AutoMode に GoWAIT_CONNECT を設定する。(F/W)	0 (reference)
T1	接続検出機能をオンする。(H/W)	$T0 + 100\text{ms}\{T_{\text{SIGATT}}\} < T1$
T2	デバイスが接続される。	T2
T3	切断検出機能をオンする。(H/W)	$T2 + 2.5\mu\text{s}\{T_{\text{DCNN}}\} < T3$
T4	接続検出ステータス(DetectCon)を発行する。(H/W) 切断検出機能と接続検出機能をオフする。(H/W)	$T3 + 100\text{ms}\{T_{\text{ATTDB}}\} < T4$

注： {} は、USB2.0 規格書で規格されている名称である。

## 1. 機能説明

### 1.3.10.3.3. GoDISABLED

ホストステート遷移実行(H\_NegoControl0.AutoMode)に"GoDISABLED"を設定すると、ディセーブルド遷移に必要な処理を本 LSI のハードウェアが自動的に実行します。

"WAIT\_CONNECT"ステートで接続を検出した時、"RESET"ステートで異常なデバイスからの Chirpを検出した時、"OPERATIONAL"ステートでポートエラーを検出した時は、本ステートに入ります。

#### 1.3.10.3.3.1. HS デバイスが接続されている場合

HS デバイスが接続されている場合の実行手順は以下のとおりです。下記(2)～(6)は本 LSI のハードウェアが自動的に実行します。

- (1) ファームウェアがホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoDISABLED"を設定します(T0)。
- (2) ホストステートモニタ(H\_NegoControl\_0.HostState)を"DISABLED"に設定します(T0)。
- (3) 切断検出機能をオフします(T0)。
- (4) 現在実行中のトランザクション完了を待って、トランシーバ選択(H\_XcvrControl.XcvrSelect)とターミナル選択(H\_XcvrControl.TermSelect)とポートスピード(H\_NegoControl\_1.PortSpeed)を"FS"モードに設定し、オペレーションモード(H\_XcvrControl.OpMode[1:0])を"PowerDown"に設定します(T1)。
- (5) 切断検出機能をオンします(T3)。
- (6) ファームウェアに対し、ディセーブルド遷移完了ステータス(H\_SIE\_IntStat\_1.DisabledCmp)を発行します(T3)。

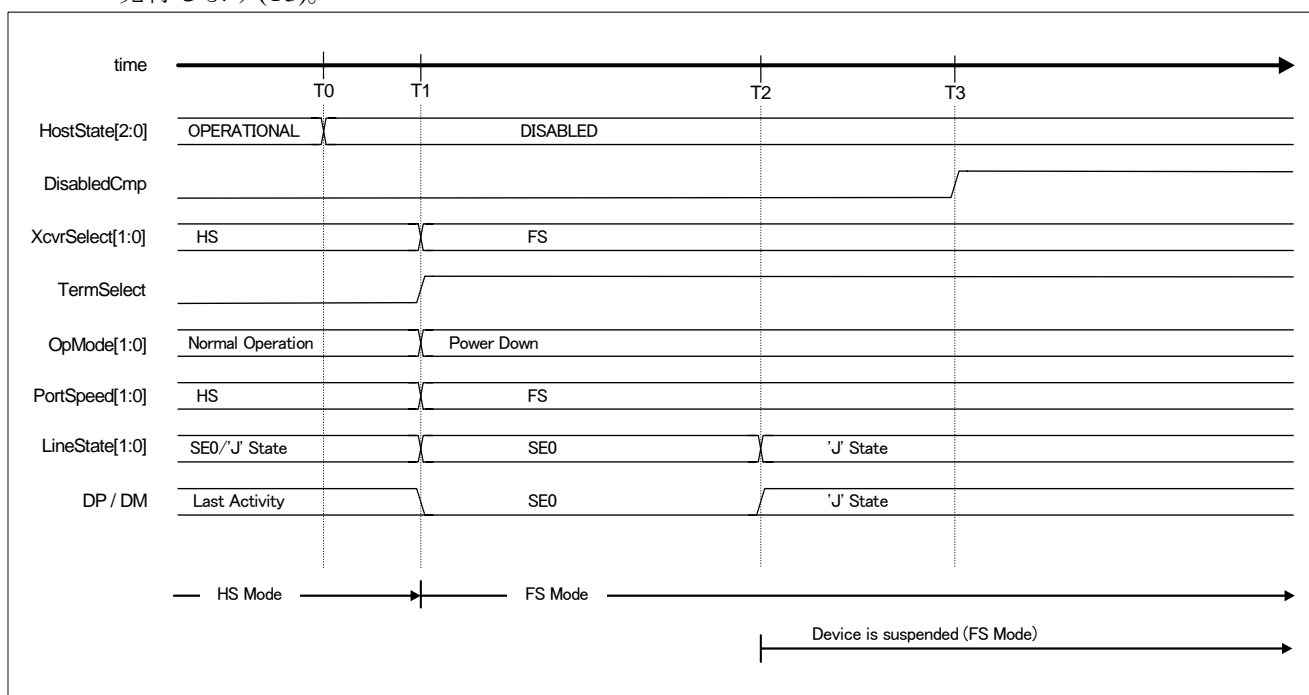


図 1-50 Disabled Timing (HS mode)

表 1-45 Disabled Timing Values (HS mode)

Timing Parameter	Description	Value
T0	H_NegoControl_0.AutoMode に"GoDISABLED"を設定する。(F/W) 切断検出機能をオフする。(H/W)	0 (reference)
T1	最後のバス・アクティビティ。完了後、トランシーバ選択とターミナル選択ポートスピードを"FS"モードに設定し、オペレーションモード(H_XcvrControl.OpMode[1:0])を"PowerDown"に設定する。(H/W)	T1
T2	デバイスがサスペンドを検出し、FS mode になる。	$T1 + 3.0\text{ms} < T2 \{T_{\text{WTREV}}\} < T1 + 3.125\text{ms}$
T3	切断検出機能をオンする。(H/W) ディセーブルド遷移完了ステータスを発行する。(H/W)	$T1 + 4\text{ms} < T3$

注： {} は、USB2.0 規格書で規格されている名称である。

### 1.3.10.3.3.2. FS デバイスが接続されている場合

FS デバイスが接続されている場合の実行手順は以下のとおりです。下記(2)～(6)は本 LSI のハードウェアが自動的に実行します。

- (1) ファームウェアがホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoDISABLED"を設定します(T0)。
- (2) ホストステートモニタ(H\_NegoControl\_0.HostState)を"DISABLED"に設定します(T0)。
- (3) 切断検出機能をオフします(T0)。
- (4) 現在実行中のトランザクション完了を待って、オペレーションモード(H\_XcvrControl.OpMode[1:0])を"PowerDown"に設定します(T1)。
- (5) 切断検出機能をオンします(T3)。
- (6) ファームウェアに対し、ディセーブルド遷移完了ステータス(H\_SIE\_IntStat\_1.DisabledCmp)を発行します(T3)。

# 1. 機能説明

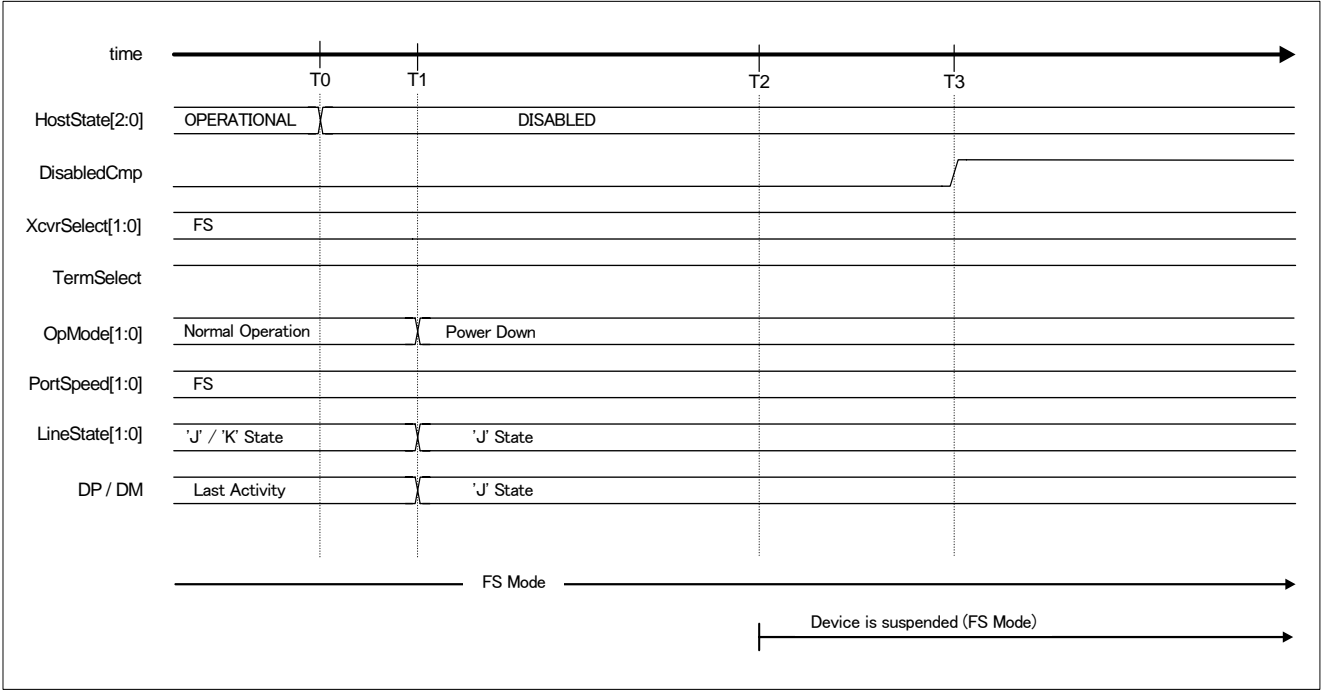


図 1-51 Disabled Timing (FS mode)

表 1-46 Disabled Timing Values (FS mode)

Timing Parameter	Description	Value
T0	H_NegoControl_0.AutoMode に GoDisabled を設定する。(F/W) 切断検出機能をオフする。(H/W)	0 (reference)
T1	最後のバス・アクティビティ。完了後、オペレーションモード(H_XcvrControl.OpMode[1:0])を”PowerDown”に設定する。(H/W)	T1
T2	デバイスがサスペンドを検出する。	$T1 + 3.0\text{ms} < T2 \{T_{\text{WTREV}}\} < T1 + 3.125\text{ms}$
T3	切断検出機能をオンする。 ディセーブルド遷移完了ステータスを発行する。(H/W)	$T1 + 4\text{ms} < T3$

注： {} は、USB2.0 規格書で規格されている名称である。

### 1.3.10.3.3.3. LS デバイスが接続されている場合

LS デバイスが接続されている場合の実行手順は以下のとおりです。下記(2)～(6)は本 LSI のハードウェアが自動的に実行します。

- (1) ファームウェアがホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoDISABLED"を設定します(T0)。
- (2) ホストステートモニタ(H\_NegoControl\_0.HostState)を"DISABLED"に設定します(T0)。
- (3) 切断検出機能をオフします(T0)。
- (4) 現在実行中のトランザクション完了を待って、オペレーションモード(H\_XcvtControl.OpMode[1:0])を"PowerDown"に設定します(T1)。
- (5) 切断検出機能をオンします(T3)。
- (6) ファームウェアに対し、ディセーブルド遷移完了ステータス(H\_SIE\_IntStat\_1.DisabledCmp)を発行します(T3)。

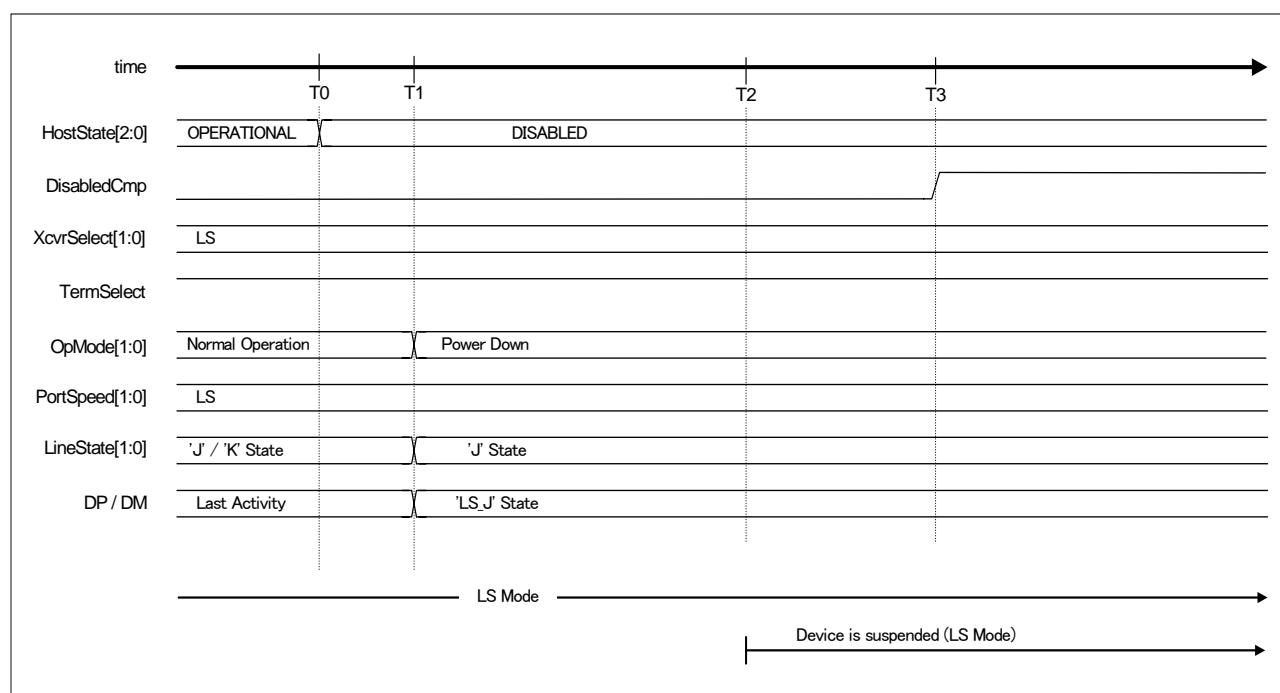


図 1-52 Disabled Timing (LS mode)

## 1. 機能説明

---

表 1-47 Disabled Timing Values (LS mode)

Timing Parameter	Description	Value
T0	H_NegoControl_0.AutoMode に"GoDISABLED"を設定する。(F/W) 切断検出機能をオフする。(H/W)	0 (reference)
T1	最後のバス・アクティビティ。完了後、オペレーションモード(H_XcvrControl.OpMode[1:0])を”PowerDown”に設定する。(H/W)	T1
T2	デバイスがサスペンドを検出する。	$T1 + 3.0\text{ms} < T2 \{T_{\text{WTREV}}\} < T1 + 3.125\text{ms}$
T3	切断検出機能をオンする。 ディセーブルド遷移完了ステータスを発行する。(H/W)	$T1 + 4\text{ms} < T3$

注： {} は、USB2.0 規格書で規格されている名称である。

### 1.3.10.3.4. GoRESET

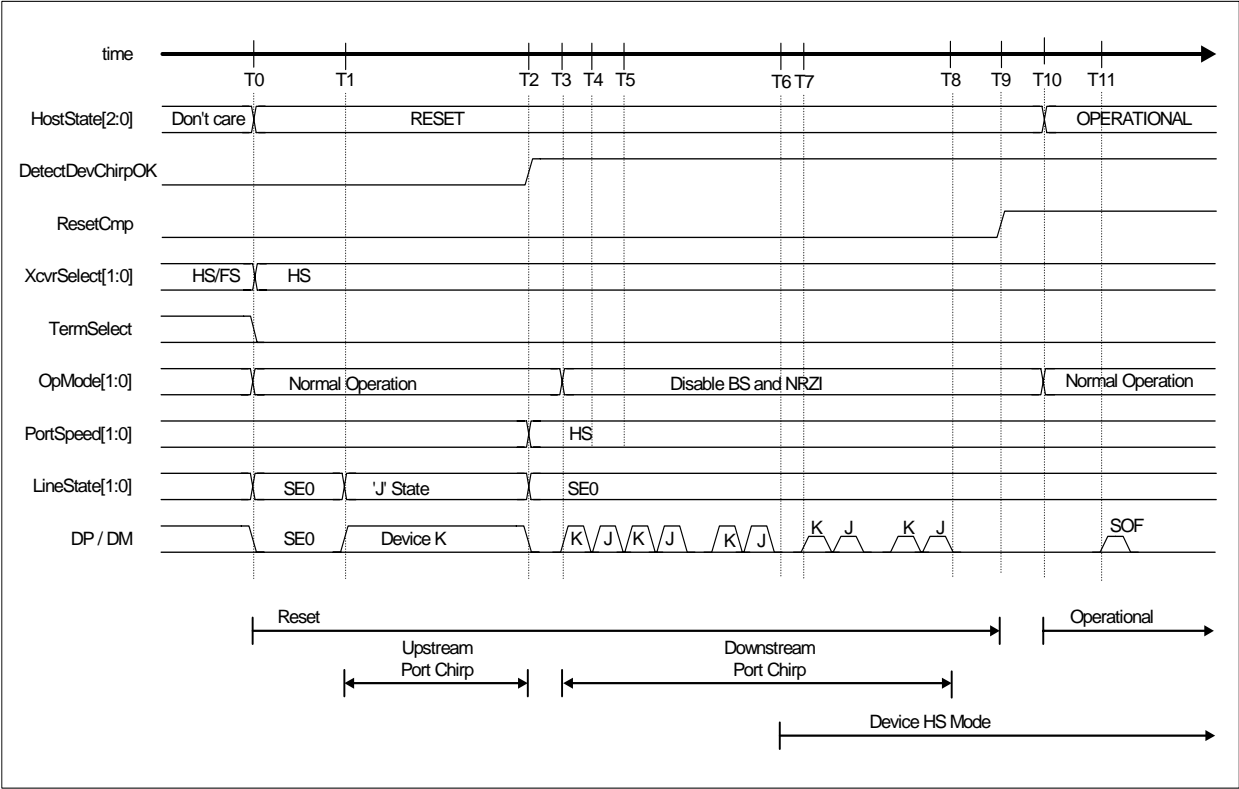
ホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoRESET"を設定すると、"RESET"遷移に必要な処理を本 LSI のハードウェアが自動的に実行します。なお、"OPERATIONAL"から本ステートに遷移する場合には、H/W が現在実行中のトランザクション完了を待って"RESET"処理を開始します。

#### 1.3.10.3.4.1. HS デバイスに対するリセット

HS デバイスに対するリセットの実行手順は以下のとおりです。下記(2)～(14)は本 LSI のハードウェアが自動的に実行します。

- (1) ファームウェアがホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoRESET"を設定します(T0)。
- (2) ホストステートモニタ(H\_NegoControl\_0.HostState)を"RESET"に設定します(T0)。
- (3) トランシーバ選択(H\_XcvrControl.XcvrSelect)とターミナル選択(H\_XcvrControl.TermSelect)を"HS"モードに設定します(T0)。
- (4) オペレーションモード(H\_XcvrControl.OpMode[1:0])を"Normal"に設定します(T0)。
- (5) デバイスチャープ検出機能をオンします(T0)。
- (6) ラインステート(H\_USB\_Status.LineState[1:0])にアクティビティ有("J"ステートとして見える)が 2.5us 以上継続することでデバイスの Chirp を認識します。そしてデバイスの Chirp がリセット開始から規定時間以内に終了する(ラインステート(H\_USB\_Status.LineState[1:0])がアクティビティ無("SE0"として見える)となる)ことをもってデバイスチャープ正常検出ステータス(H\_SIE\_IntStat\_0.DetectDevChirpOK)を発行します(T2)。  
ハードウェアはデバイスチャープ正常検出ステータス(H\_SIE\_IntStat\_0.DetectDevChirpOK)が"1"にセットされている場合にデバイスから正常な Chirp 信号が送信されたと判断し以降の処理を行います。その為、HS デバイスが切断された時には必ずデバイスチャープ正常検出ステータス(H\_SIE\_IntStat\_0.DetectDevChirpOK)を"0"にクリアしてください。クリアを行わずデバイスチャープ正常検出ステータス(H\_SIE\_IntStat\_0.DetectDevChirpOK)が"1"にセットされた状態で FS デバイスが接続されると、そのデバイスを HS デバイスと誤認識して以降の処理を行ってしまいます。
- (7) デバイスチャープ検出機能をオフします(T2)。
- (8) デバイスの Chirp 終了後、ホストは"Chirp K"を出力し始めます(T3)。
- (9) ホストは"Chirp K"から"Chirp J"に切り替えて出力します(T4)。
- (10) ホストは"Chirp J"から"Chirp K"に切り替えて出力します(T5)。以降ホストは"Chirp K"と"Chirp J"のシーケンスを交互に出力します。
- (11) デバイスはホストの Chirp を検出すると HS モードに移行します(T6)。T7 の時点から Chirp の高さが変わっているのは、デバイス側の HS ターミネーションが有効になったことを表しています。通常、デバイスが FS モード時の Chirp は約 800mV、HS モード時の Chirp では、約 400mV となります。
- (12) ホストは Chirp を終了します(T8)。
- (13) リセットを終了します(T9)。
- (14) リセット完了ステータス(H\_SIE\_IntStat\_1.ResetCmp)を発行します(T9)。

# 1. 機能説明



**図 1-53 Reset Timing (HS mode)**



表 1-48 Reset Timing Values (HS mode)

Timing Parameter	Description	Value
T0	H_NegoControl_0.AutoMode に "GoRESET" を設定する。 (F/W) デバイスチャープ検出機能をオンする。(H/W)	0 (reference)
T1	デバイスが Chirp を開始。	$T0 < T1 < T0 + 6.0\text{ms}$
T2	デバイスが Chirp を終了。 ポートスピードを HS に設定する。 デバイスチャープ検出機能をオフする。 デバイスチャープ正常検出ステータス (DetectDevChirpOK)を発行する。(H/W)	$T1 + 1.0\text{ms} \{T_{UCH}\} < T2 < T0 + 7.0\text{ms} \{T_{UCHEND}\}$
T3	ホストが最初の Chirp("Chirp K")を出力する。(H/W)	$T2 < T3 < T2 + 100\mu\text{s} \{T_{WTDCH}\}$
T4	ホストが"Chirp K"から"Chirp J"に切り替えて出力する。 (H/W)	$T3 + 40\mu\text{s} \{T_{DCHBIT}\} < T4 < T3 + 60\mu\text{s} \{T_{DCHBIT}\}$
T5	ホストが"Chirp J"から"Chirp K"に切り替えて出力する。 (H/W)	$T4 + 40\mu\text{s} \{T_{DCHBIT}\} < T5 < T4 + 60\mu\text{s} \{T_{DCHBIT}\}$
T6	デバイスがホストの Chirp を検出する。	T6
T7	デバイスが HS モードに遷移。	$T6 < T7 < T6 + 500\mu\text{s}$
T8	ホストが Chirp を終了。(H/W)	$T3 + 50\text{ms} \{T_{DRSTR}\} < T8$
T9	リセットの終了。リセット終了ステータス(ResetCmp)を発行する。(H/W)	$T8 < T9 < T8 + 150\mu\text{s}$
T10( 参考)	H_NegoControl_0.AutoMode に "GoOPERATIONAL" を設定する。(F/W)	$T10 < T9 + 200\mu\text{s}$
T11( 参考)	最初の SOF が送出。(H/W)	$T10 + 120\mu\text{s} < T11 < T10 + 130\mu\text{s}$ $T8 + 100\mu\text{s} \{T_{DCHSE0}\} < T11 < T8 + 500\mu\text{s} \{T_{DCHSE0}\}$

注： {} は、USB2.0 規格書で規格されている名称である。

## 1. 機能説明

### 1.3.10.3.4.2. 異常なデバイスの Chirp 検出

HS Detection Handshake において、デバイスの Chirp が異常だった場合の動作を示します。チャープ完了ディセーブル(H\_NegoControl\_1.DisChirpFinish)の設定により 2 つの動作モードが選択できます。

#### 1.3.10.3.4.2.1. チャープ完了ディセーブル(H\_NegoControl\_1.DisChirpFinish)設定が 0 の場合

異常検出後にホストの Chirp を行いません。デバイスチャープ異常検出ステータスが発行された場合、ファームウェアはリセット完了ステータス(H\_SIE\_IntStat\_1.ResetCmp)の発行を待って、ホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoDISABLED"を設定し、ホストステートを"DISABLED"ステートに遷移させます。下記(2)～(9)は本 LSI のハードウェアが自動的に実行します。

- (1) ファームウェアがホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoRESET"を設定します(T0)。
- (2) ホストステートモニタ(H\_NegoControl\_0.HostState)を"RESET"に設定します(T0)。
- (3) トランシーバ選択(H\_XcvrControl.XcvrSelect)とターミナル選択(H\_XcvrControl.TermSelect)を HS モードに設定します(T0)。
- (4) オペレーションモード(H\_XcvrControl.OpMode[1:0])を"Normal"に設定します(T0)。
- (5) デバイスチャープ検出機能をオンします(T0)。
- (6) ラインステート(H\_USB\_Status.LineState[1:0])にアクティビティ有("J"ステートとして見える)が 2.5us 以上継続することでデバイスの Chirp を認識します。しかし、リセット開始から規定時間以内にデバイスの Chirp が終了しないため異常とみなし、デバイスチャープ異常検出ステータス(H\_SIE\_IntStat\_0.DetectDevChirpNG)を発行します(T2)。
- (7) デバイスチャープ検出機能をオフします(T2)。
- (8) リセットを終了します(T3)。
- (9) リセット完了ステータス(H\_SIE\_IntStat\_1.ResetCmp)を発行します(T3)。

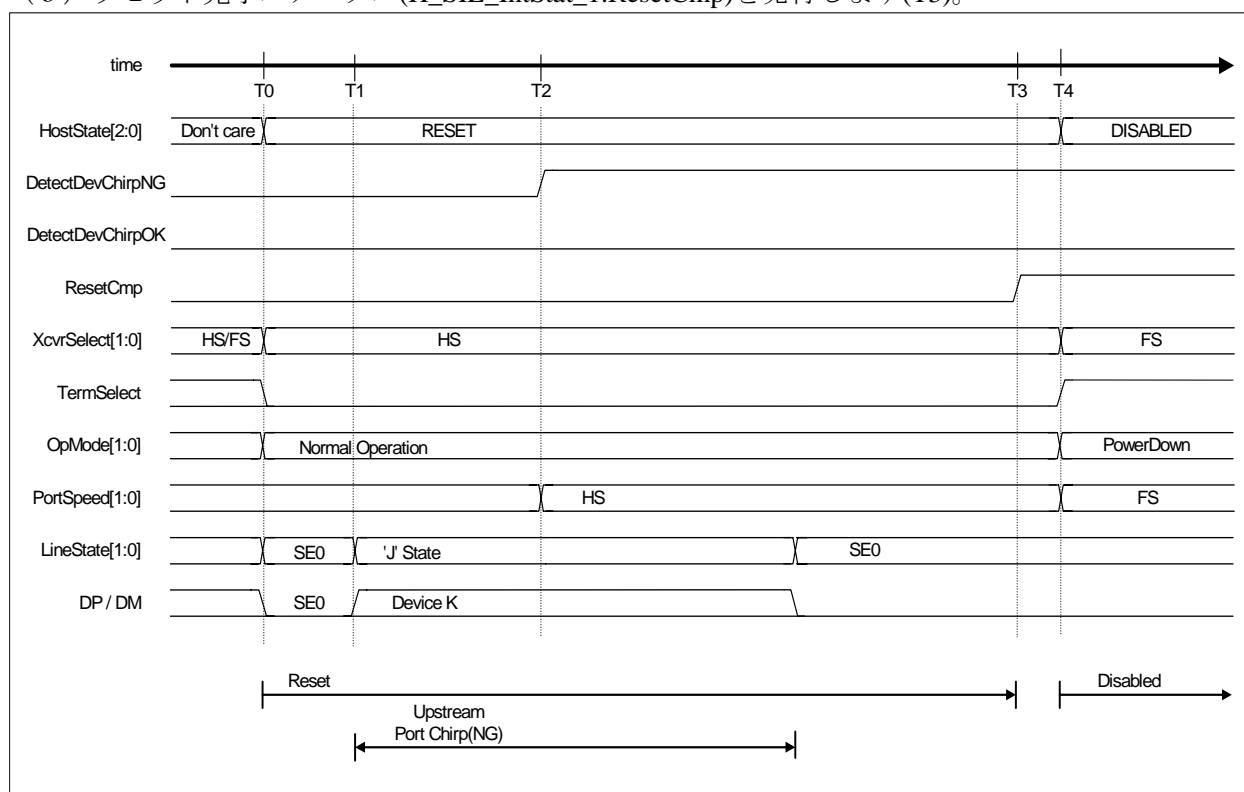


図 1-54 Detect Device Chirp NG Timing(チャープ完了ディセーブル設定 0)

表 1-49 Detect Device Chirp Timing Values(チャープ完了ディセーブル設定 0)

Timing Parameter	Description	Value
T0	H_NegoControl_0.AutoMode に"GoRESET"を設定する。(F/W) デバイスチャープ検出機能をオンする。(H/W)	0 (reference)
T1	デバイスが Chirp を開始。	$T0 < T1 < T0 + 6.0\text{ms}$
T2	デバイスチャープ異常検出ステータス (DetectDevChirpNG)を発行する。デバイスチャープ検出機能をオフする。(H/W)	$T0 + 7\text{ms}\{T_{\text{UCHEND}}\} < T2$
T3	リセットの終了。リセット終了ステータス(ResetCmp)を発行する。(H/W)	$T2 + 50\text{ms}\{T_{\text{DRSTR}}\} < T3$
T4(参考)	H_NegoControl_0.AutoMode に"GoDISABLED"を設定する。(F/W)	T4

注： {} は、USB2.0 規格書で規格されている名称である。

## 1. 機能説明

---

### 1.3.10.3.4.2.2. チャープ完了ディセーブル(H\_NegoControl\_1.DisChirpFinish)設定が 1 の場合

異常検出後にデバイスの Chirp 完了を待ってホストの Chirp を行います。

本モード使用時において、リセット完了ステータス(H\_SIE\_IntStat\_1.ResetCmp)を発行を待たずにホストステートを "DISABLED" ステータスに遷移させる場合には、H\_NegoControl\_0 レジスタに 0x80(H\_NegoControl\_0.AutoModeCancel に"1"、かつ H\_NegoControl\_0.AutoMode に"0x0")をライトして実行中ステータスの動作を停止させます。H\_NegoControl\_0.AutoModeCancel ビットは停止処理が完了することで"0"になります(60MHz クロックで 6 サイクル程度必要)ので、H\_NegoControl\_0.AutoModeCancel ビットが"0"になるのを確認してから、同レジスタに 0x03 をライト(すなわちホストステータス遷移実行(H\_NegoControl\_0.AutoMode)に"GoDISABLED"を設定)してください。

下記(2)～(15)は本 LSI のハードウェアが自動的に実行します。

- (1) ファームウェアがホストステータス遷移実行(H\_NegoControl\_0.AutoMode)に"GoRESET"を設定します(T0)。
- (2) ホストステータスモニタ(H\_NegoControl\_0.HostState)を"RESET"に設定します(T0)。
- (3) トランシーバ選択(H\_XcvrControl.XcvrSelect)とターミナル選択(H\_XcvrControl.TermSelect)を HS モードに設定します(T0)。
- (4) オペレーションモード(H\_XcvrControl.OpMode[1:0])を"Normal"に設定します(T0)。
- (5) デバイスチャープ検出機能をオンします(T0)。
- (6) ラインステータス(USB\_Host\_Status.LineState[1:0])にアクティビティ有("J"ステータスとして見える)が 2.5us 以上継続することでデバイスの Chirp を認識します。しかし、リセット開始から規定時間以内にデバイスの Chirp が終了しないため異常とみなし、デバイスチャープ異常検出ステータス(H\_SIE\_IntStat\_0.DetectDevChirpNG)を発行します(T2)。
- (7) デバイスチャープ検出機能をオフします(T2)。
- (8) デバイスの Chirp が終了する(ラインステータス(H\_USB\_Status.LineState[1:0])がアクティビティ無("SE0"として見える)になる)ことをもってデバイスチャープ正常検出ステータス(H\_SIE\_IntStat\_0.DetectDevChirpOK)を発行します(T3)。
- (9) デバイスの Chirp 終了後、ホストは"Chirp K"を出力し始めます(T4)。
- (10) ホストは"Chirp K"から"Chirp J"に切り替えて出力します(T5)。
- (11) ホストは"Chirp J"から"Chirp K"に切り替えて出力します(T6)。以降ホストは"Chirp K"と"Chirp J"のシーケンスを交互に出力します。
- (12) デバイスはホストの Chirp を検出すると HS モードに移行します(T7)。T8 の時点から Chirp の高さが変わっているのは、デバイス側の HS ターミネーションが有効になったことを表しています。通常、デバイスが FS モード時の Chirp は約 800mV、HS モード時の Chirp では、約 400mV となります。
- (13) ホストは Chirp を終了します(T9)。
- (14) リセットを終了します(T10)。
- (15) リセット完了ステータス(H\_SIE\_IntStat\_1.ResetCmp)を発行します(T10)。

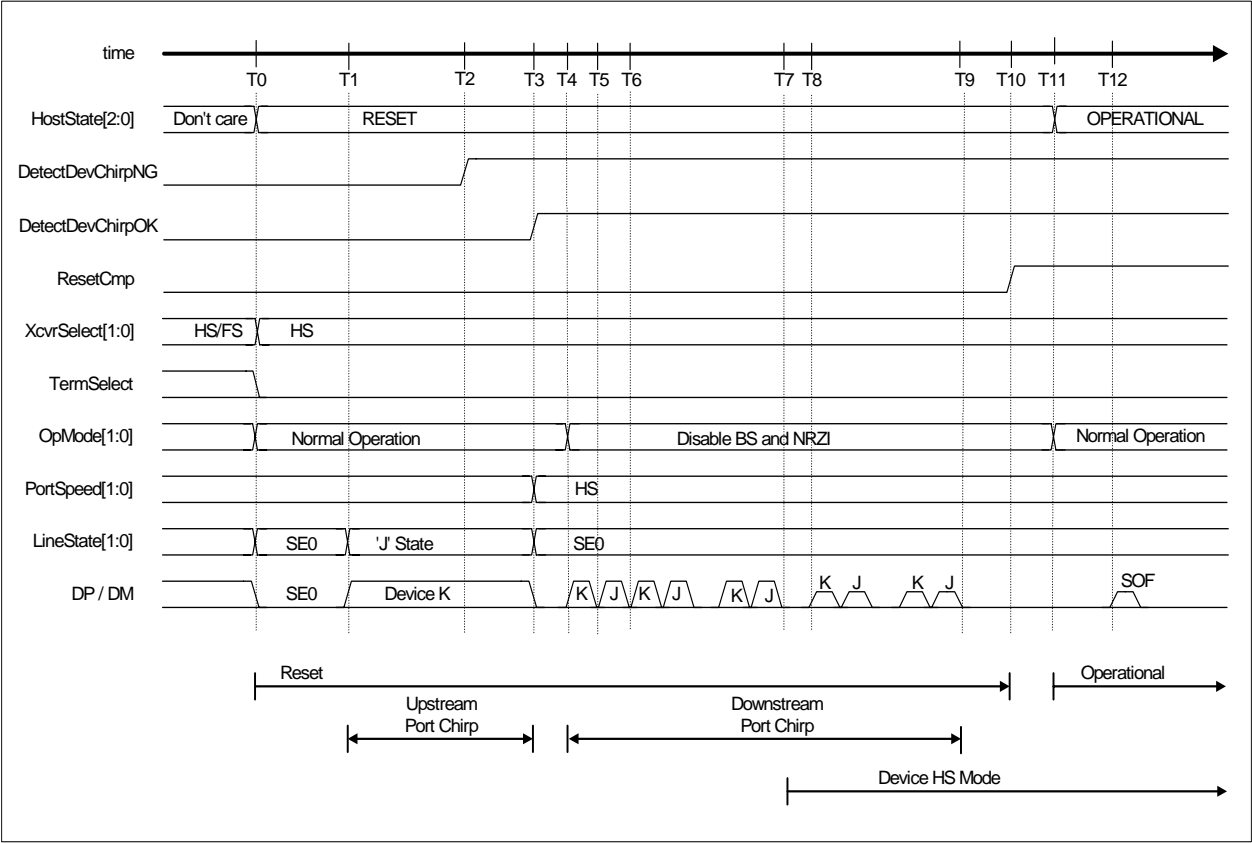


図 1-55 Detect Device Chirp NG Timing(チャープ完了ディセーブル設定 1)

## 1. 機能説明

表 1-50 Detect Device Chirp Timing Values(チャープ完了ディセーブル設定 1)

Timing Parameter	Description	Value
T0	H_NegoControl_0.AutoMode に"GoRESET"を設定する。 (F/W) デバイスチャープ検出機能をオンする。(H/W)	0 (reference)
T1	デバイスが Chirp を開始。	$T0 < T1 < T0 + 6.0\text{ms}$
T2	デバイスチャープ異常検出ステータス (DetectDevChirpNG)を発行する。 デバイスチャープ検出機能をオフする。(H/W)	$T0 + 7\text{ms}\{T_{\text{UCHEND}}\} < T2$
T3	デバイスが Chirp を終了。 ポートスピードを HS に設定する。 デバイスチャープ正常検出ステータス (DetectDevChirpOK)を発行する。(H/W)	T3
T4	ホストが最初の Chirp("Chirp K")を出力する。(H/W)	$T3 < T4 < T3 + 100\mu\text{s}\{T_{\text{WTDCH}}\}$
T5	ホストが"Chirp K"から"Chirp J"に切り替えて出力する。 (H/W)	$T4 + 40\mu\text{s}\{T_{\text{DCHBIT}}\} < T45 < T4 + 60\mu\text{s}\{T_{\text{DCHBIT}}\}$
T6	ホストが"Chirp J"から"Chirp K"に切り替えて出力する。 (H/W)	$T5 + 40\mu\text{s}\{T_{\text{DCHBIT}}\} < T6 < T5 + 60\mu\text{s}\{T_{\text{DCHBIT}}\}$
T7	デバイスがホストの Chirp 検出する。	T7
T8	デバイスが HS モードに遷移。	$T7 < T8 < T6 + 500\mu\text{s}$
T9	ホストが Chirp を終了。(H/W)	$T4 + 50\text{ms}\{T_{\text{DRSTR}}\} < T9$
T10	リセットの終了。リセット終了ステータス(ResetCmp)を発行する。(H/W)	$T9 < T10 < T9 + 150\mu\text{s}$
T11( 参考)	H_NegoControl0.AutoMode に"GoOPERATIONAL"を設定する。(F/W)	$T11 < T10 + 200\mu\text{s}$
T12( 参考)	最初の SOF が送出。(H/W)	$T11 + 120\mu\text{s} < T12 < T11 + 130\mu\text{s}$ $T9 + 100\mu\text{s}\{T_{\text{DCHSE0}}\} < T12 < T9 + 500\mu\text{s}\{T_{\text{DCHSE0}}\}$

注： {} は、USB2.0 規格書で規格されている名称である。

### 1.3.10.3.4.3. FS デバイスに対するリセット

FS デバイスに対するリセットの実行手順は以下のとおりです。下記(2)～(9)は本 LSI のハードウェアが自動的に実行します。

- (1) ファームウェアがホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoRESET"を設定します(T0)。
- (2) ホストステートモニタ(H\_NegoControl\_0.HostState)を"RESET"に設定します(T0)。
- (3) トランシーバ選択(H\_XcvrControl.XcvrSelect)とターミナル選択(H\_XcvrControl.TermSelect)を HS モードに設定します(T0)。
- (4) オペレーションモード(H\_XcvrControl.OpMode[1:0])を"Normal"に設定します(T0)。
- (5) デバイスチャープ検出機能をオンします(T0)。
- (6) デバイスの Chirp を検出せず、ポートスピード(H\_NegoControl\_1.PortSpeed[1:0])が HS/FS の為、相手デバイスが FS デバイスであると判断し、トランシーバ選択(H\_XcvrControl.XcvrSelect)とポートスピード(H\_NegoControl\_1.PortSpeed[1:0])を"FS"に設定します(T1)。
- (7) デバイスチャープ検出機能をオフします(T1)。
- (8) ターミナル選択(H\_XcvrControl.TermSelect)を"FS"に設定します(T2)。
- (9) リセット終了ステータス(H\_SIE\_IntStat\_1.ResetCmp)を発行します(T3)。

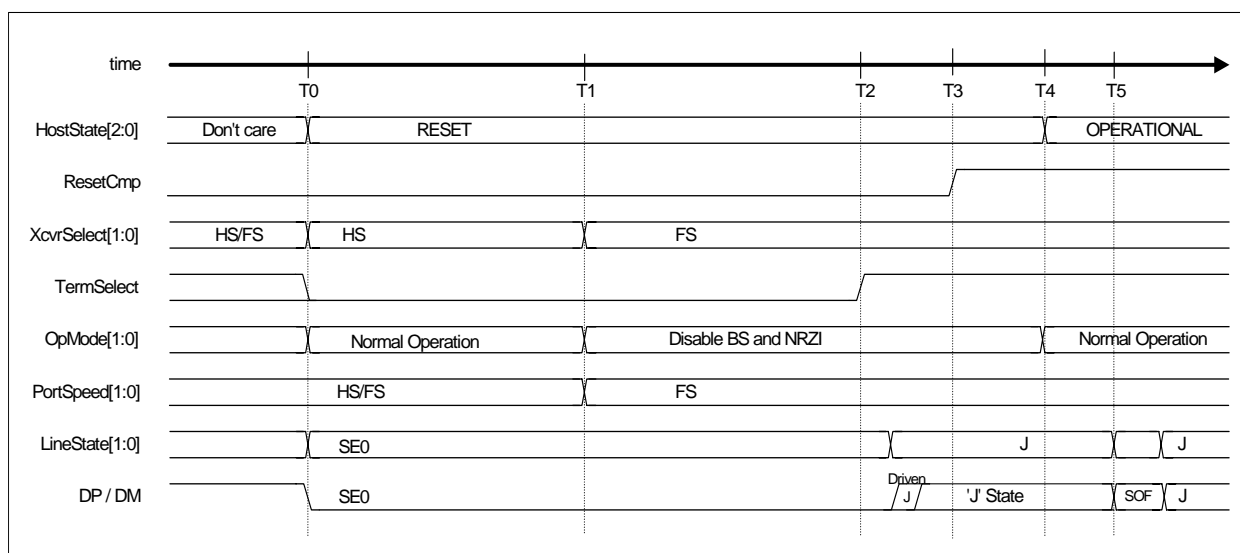


図 1-56 Reset Timing (FS mode)

表 1-51 Reset Timing Values (FS mode)

Timing Parameter	Description	Value
T0	H_NegoControl_0.AutoMode に"GoRESET"を設定する。(F/W) デバイスチャープ検出機能をオンする。(H/W)	0 (reference)
T1	トランシーバ選択を"FS"に設定する。 ポートスピードを"FS"に設定する。 デバイスチャープ検出機能をオフする。(H/W)	$T0 + 7.0ms\{T_{UCHEND}\} < T1$
T2	ターミナル選択を"FS"に設定する。(H/W)	$T0 + 50ms\{T_{DRSTR}\} < T2$
T3	リセット完了ステータスを発行する。(H/W)	$T2 + 150us < T3$
T4(参考)	H_NegoControl_0.AutoMode に"GoOPERATIONAL"を設定する。(F/W)	T4
T5(参考)	最初の SOF が送出。(H/W)	$T4 + 0.9ms < T5 < T4 + 1.1ms$ ( $T5 < T2 + 3ms$ )

注： {} は、USB2.0 規格書で規格されている名称である。

## 1. 機能説明

### 1.3.10.3.4.4. LS デバイスに対するリセット

LS デバイスに対するリセットの実行手順は以下のとおりです。下記(2)～(7)は本 LSI のハードウェアが自動的に実行します。

- (1) ファームウェアがホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoRESET"を設定します(T0)。
- (2) ホストステートモニタ(H\_NegoControl\_0.HostState)を"RESET"に設定します(T0)。
- (3) トランシーバ選択(H\_XcvrControl.XcvrSelect)とターミナル選択(H\_XcvrControl.TermSelect)を HS モードに設定します(T0)。
- (4) オペレーションモード(H\_XcvrControl.OpMode[1:0])を"Normal"に設定します(T0)。
- (5) ポートスピード (H\_NegoControl\_1.PortSpeed[1:0]) が "LS" の為、 トランシーバ選択 (H\_XcvrControl.XcvrSelect)を"LS"に設定します(T1)。
- (6) ターミナル選択(H\_XcvrControl\_1.TermSelect)を"FS"に設定します(T2)。
- (7) リセット終了ステータス(H\_SIE\_IntStat\_1.ResetCmp)を発行します(T3)。

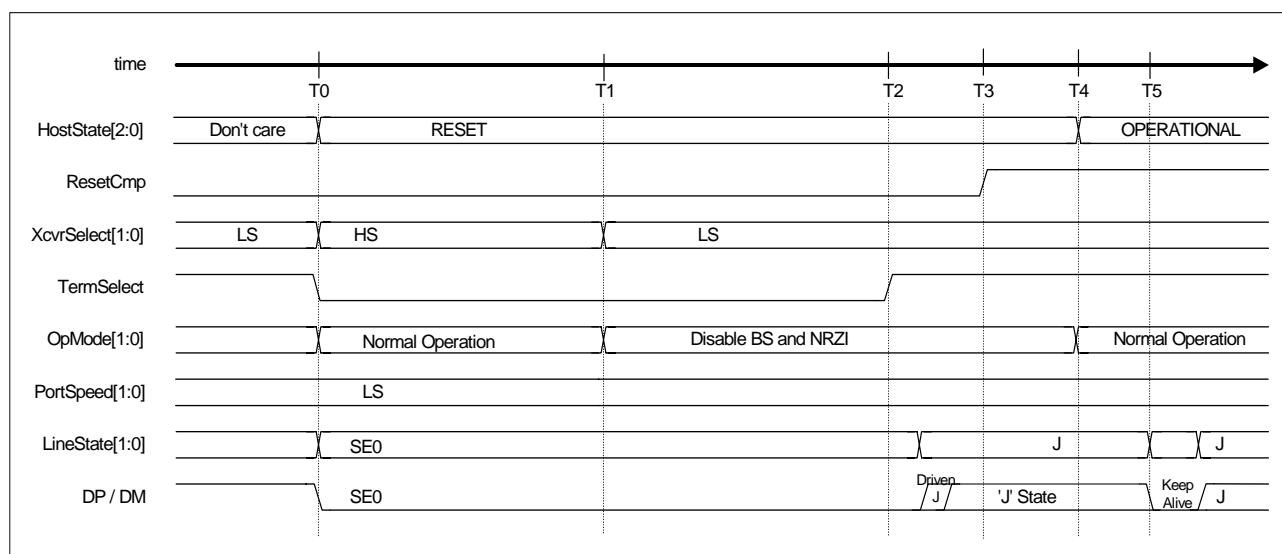


図 1-57 Reset Timing (LS mode)

表 1-52 Reset Timing Values (LS mode)

Timing Parameter	Description	Value
T0	H_NegoControl_0.AutoMode に"GoRESET"を設定する。(F/W)	0 (reference)
T1	トランシーバ選択を"LS"に設定する。(H/W)	$T0 + 7.0ms\{T_{UCHEND}\} < T1$
T2	ターミナル選択を"FS"に設定する。(H/W)	$T0 + 50ms\{T_{DRSTR}\} < T2$
T3	リセット完了ステータスを発行する。(H/W)	$T2 + 150us < T3$
T4(参考)	H_NegoControl_0.AutoMode に"GoOPERATIONAL"を設定する。(F/W)	T4
T5(参考)	最初の KeepAlive が送出。(H/W)	$T4 + 0.9ms < T5 < T4 + 1.1ms$ ( $T5 < T2 + 3ms$ )

注： {} は、USB2.0 規格書で規格されている名称である。



### 1.3.10.3.5. GoOPERATIONAL

ホストステート遷移実行 (H\_NegoControl\_0.AutoMode) に "GoOPERATIONAL" を設定すると、"OPERATIONAL" 遷移に必要な処理を本 LSI のハードウェアが自動的に実行します。

- (1) ファームウェアがホストステート遷移実行 (H\_NegoControl\_0.AutoMode) に "GoOPERATIONAL" を設定します (T0)。
- (2) ホストステートモニタ (H\_NegoControl\_0.HostState) を "OPERATIONAL" に設定します (T0)。
- (3) オペレーションモード (H\_XcvtControl.OpMode[1:0]) を "Normal" に設定し、USB のトランザクションを実行する状態になります (T0)。
- (4) 切断検出機能をオンします (T0)。
- (5) ポートスピード (H\_NegoControl\_1.PortSpeed[1:0]) が "HS" または "FS" に設定されている場合は最初の SOF を発行します (T1)。"LS" に設定されている場合は最初の KeepAlive が発行します (T1)。以降はチャンネルの設定に従って転送が行われます。

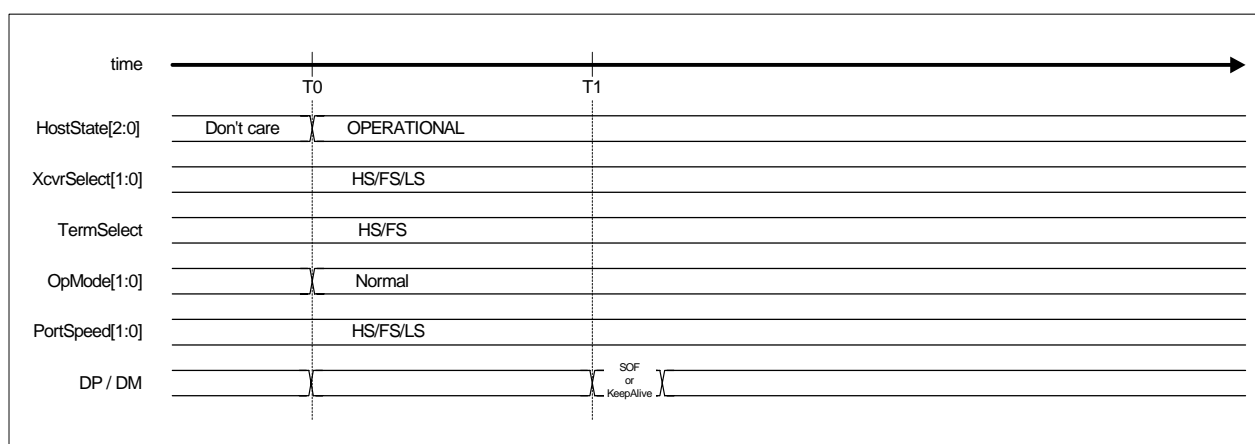


図 1-58 GoOPERATIONAL Timing

表 1-53 GoOPERATIONAL Timing Values

Timing Parameter	Description	Value
T0	H_NegoControl_0.AutoMode に "GoOPERATIONAL" を設定する。(F/W)	0 (reference)
T1	最初の SOF(HS/FS)、または KeepAlive(LS)が送出。	$T0 + 120\mu s < T1(HS) < T0 + 130\mu s$ $T0 + 0.9ms < T1(FS, LS) < T0 + 1.1ms$

## 1. 機能説明

### 1.3.10.3.6. GoSUSPEND

ホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoSUSPEND"を設定すると、サスペンド遷移に必要な処理を本 LSI のハードウェアが自動的に実行します。

#### 1.3.10.3.6.1. HS デバイスが接続されている場合

HS デバイスが接続されている場合の実行手順は以下のとおりです。下記(2)～(7)は本 LSI のハードウェアが自動的に実行します。

- (1) ファームウェアがホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoSUSPEND"を設定します(T0)。
- (2) ホストステートモニタ(H\_NegoControl\_0.HostState)を"SUSPEND"に設定します(T0)。
- (3) 切断検出機能およびリモート・ウェークアップ検出機能をオフします(T0)。
- (4) 現在実行中のトランザクション完了を待って、トランシーバ選択(H\_XcvrControl.XcvrSelect)とターミナル選択(H\_XcvrControl.TermSelect)を FS モードに設定し、オペレーションモード(H\_XcvrControl.OpMode[1:0])を"PowerDown"に設定します(T1)。
- (5) 切断検出機能をオンします(T3)。
- (6) リモート・ウェークアップ受付許可(H\_NegoControl\_1.RmtWkupDetEnb)がイネーブルの場合、リモート・ウェークアップ検出機能をオンします(T3)。
- (7) ファームウェアに対し、サスペンド遷移完了ステータス(H\_SIE\_IntStat\_1.SuspendCmp)を発行します(T3)。

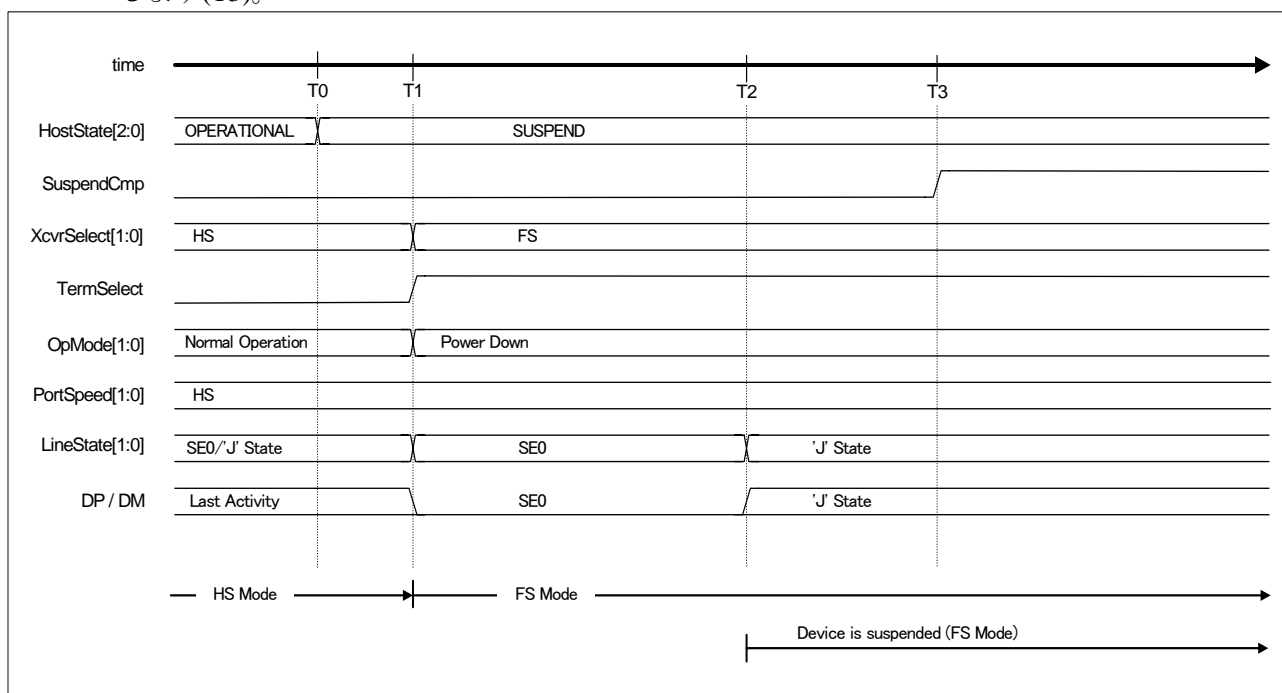


図 1-59 Suspend Timing (HS mode)

表 1-54 Suspend Timing Values (HS mode)

Timing Parameter	Description	Value
T0	H_NegoControl_0.AutoMode に"GoSUSPEND"を設定する。(F/W) 切断検出機能およびリモート・ウェークアップ検出機能をオフする。(H/W)	0 (reference)
T1	最後のバス・アクティビティ。完了後、トランシーバ選択とターミナル選択を FS モードに設定し、オペレーションモード (H_XcvtControl.OpMode[1:0]) を"PowerDown"に設定する。(H/W)	T1
T2	デバイスがサスペンドを検出し、FS mode になる。	$T1 + 3.0\text{ms} < T2 \{T_{\text{WTREV}}\} < T1 + 3.125\text{ms}$
T3	切断検出機能をオンする。 リモート・ウェークアップ受付許可がイネーブルの場合、リモート・ウェークアップ検出機能をオンする。 サスペンド遷移完了ステータスを発行する。(H/W)	$T1 + 5\text{ms}\{T_{\text{WTRSM}}\} < T3$

注： {} は、USB2.0 規格書で規格されている名称である。

## 1. 機能説明

### 1.3.10.3.6.2. FS デバイスが接続されている場合

FS デバイスが接続されている場合の実行手順は以下のとおりです。下記(2)～(7)は本 LSI のハードウェアが自動的に実行します。

- (1) ファームウェアがホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoSUSPEND"を設定します(T0)。
- (2) ホストステートモニタ(H\_NegoControl\_0.HostState)を"SUSPEND"に設定します(T0)。
- (3) 切断検出機能およびリモート・ウェークアップ検出機能をオフします(T0)。
- (4) 現在実行中のトランザクション完了を待って、オペレーションモード(H\_XcvtControl.OpMode[1:0])を"PowerDown"に設定します(T1)。
- (5) 切断検出機能をオンします(T3)。
- (6) リモート・ウェークアップ受付許可(H\_NegoControl\_1.RmtWkupDetEnb)がイネーブルの場合、リモート・ウェークアップ検出機能をオンします(T3)。
- (7) ファームウェアに対し、サスペンド遷移完了ステータス(H\_SIE\_IntStat\_1.SuspendCmp)を発行します(T3)。

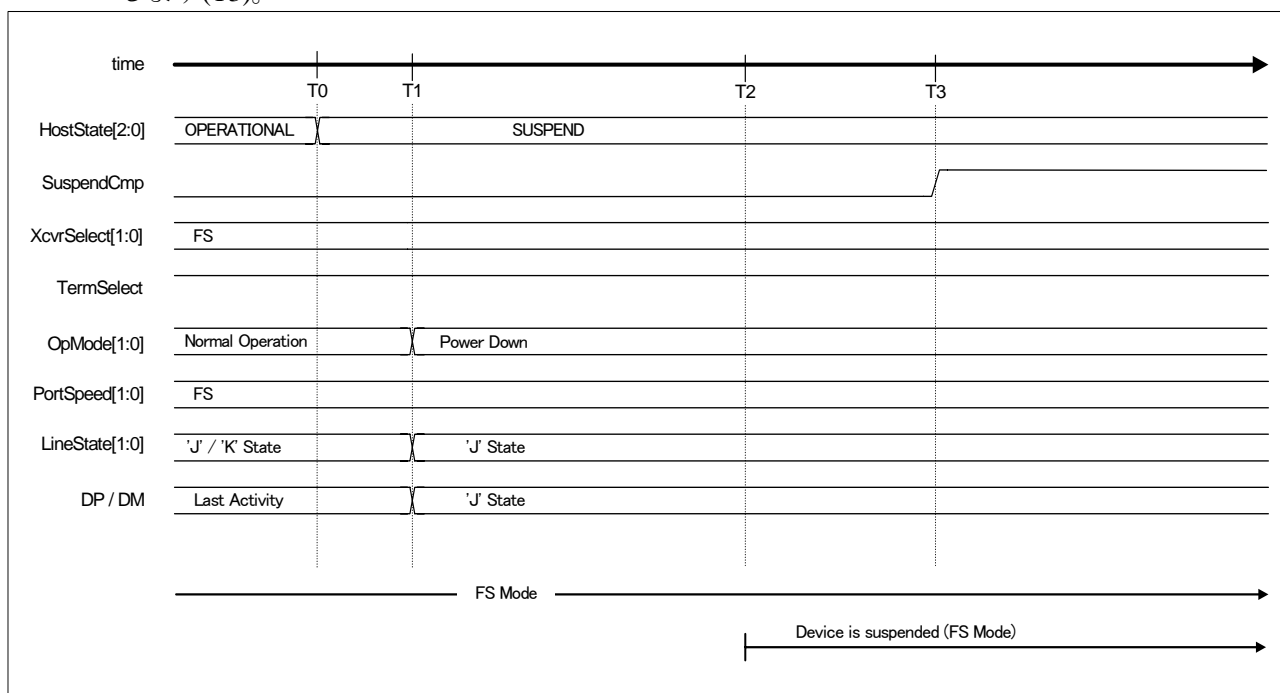


図 1-60 Suspend Timing (FS mode)

表 1-55 Suspend Timing Values (FS mode)

Timing Parameter	Description	Value
T0	H_NegoControl_0.AutoMode に"GoSUSPEND"を設定する。(F/W) 切断検出機能およびリモート・ウェークアップ検出機能をオフする。(H/W)	0 (reference)
T1	最後のバス・アクティビティ。完了後、オペレーションモード(H_XcvtControl.OpMode[1:0])を"PowerDown"に設定する。(H/W)	T1
T2	デバイスがサスペンドを検出する。	$T1 + 3.0\text{ms} < T2 \{T_{\text{WTREV}}\} < T1 + 3.125\text{ms}$
T3	切断検出機能をオンする。 リモート・ウェークアップ受付許可がイネーブルの場合、リモート・ウェークアップ検出機能をオンする。 サスペンド遷移完了ステータスを発行する。(H/W)	$T1 + 5\text{ms}\{T_{\text{WTRSM}}\} < T3$

注： {} は、USB2.0 規格書で規格されている名称である。

## 1. 機能説明

### 1.3.10.3.6.3. LS デバイスが接続されている場合

LS デバイスが接続されている場合の実行手順は以下のとおりです。下記(2)～(7)は本 LSI のハードウェアが自動的に実行します。

- (1) ファームウェアがホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoSUSPEND"を設定します(T0)。
- (2) ホストステートモニタ(H\_NegoControl0.HostState)を"SUSPEND"に設定します(T0)。
- (3) 切断検出機能およびリモート・ウェークアップ検出機能をオフします(T0)。
- (4) 現在実行中のトランザクション完了を待って、オペレーションモード(H\_XcvrControl.OpMode[1:0])を"PowerDown"に設定します(T1)。
- (5) 切断検出機能をオンします(T3)。
- (6) リモート・ウェークアップ受付許可(H\_NegoControl\_1.RmtWkupDetEnb)がイネーブルの場合、リモート・ウェークアップ検出機能をオンします(T3)。
- (7) ファームウェアに対し、サスペンド遷移完了ステータス(H\_SIE\_IntStat\_1.SuspendCmp)を発行します(T3)。

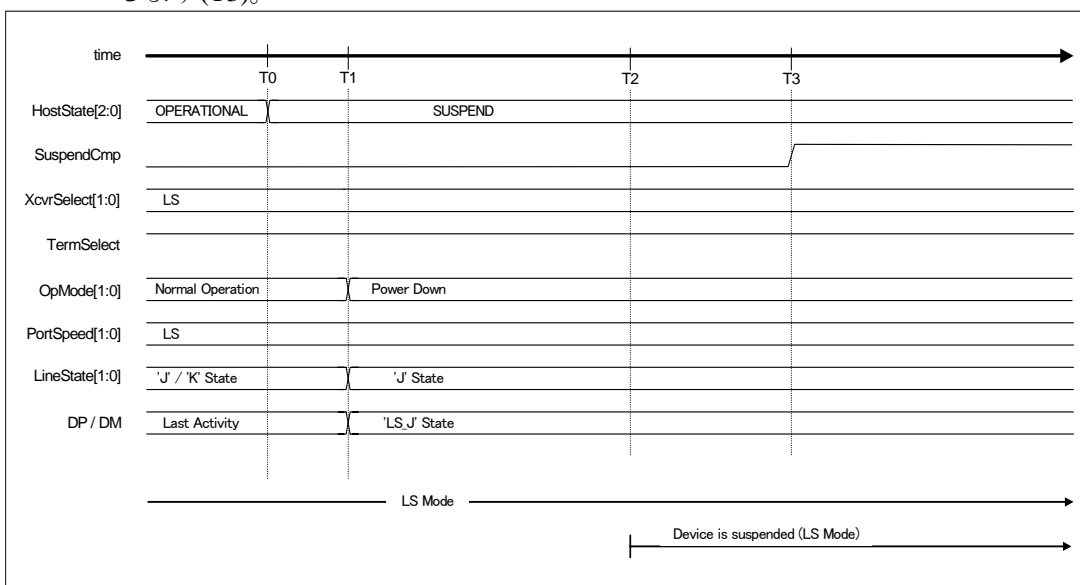


図 1-61 Suspend Timing (LS mode)

表 1-56 Suspend Timing Values (LS mode)

Timing Parameter	Description	Value
T0	H_NegoControl_0.AutoMode に"GoSUSPEND"を設定する。(F/W) 切断検出機能およびリモート・ウェークアップ検出機能をオフする。(H/W)	0 (reference)
T1	最後のバス・アクティビティ。完了後、オペレーションモード(H_XcvrControl.OpMode[1:0])を"PowerDown"に設定する。(H/W)	T1
T2	デバイスがサスペンドを検出する。	$T1 + 3.0\text{ms} < T2 \{T_{WTREV}\} < T1 + 3.125\text{ms}$
T3	切断検出機能をオンする。 リモート・ウェークアップ受付許可がイネーブルの場合、リモート・ウェークアップ検出機能をオンする。 サスペンド遷移完了ステータスを発行する。(H/W)	$T1 + 5\text{ms}\{T_{WTRSM}\} < T3$

注： {} は、USB2.0 規格書で規格されている名称である。

### 1.3.10.3.7. GoRESUME

ホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoRESUME"を設定すると、レジューム遷移に必要な処理を本 LSI のハードウェアが自動的に実行します。

#### 1.3.10.3.7.1. HS デバイスが接続されている場合

HS デバイスが接続されている場合の実行手順は以下のとおりです。下記(2)～(8)は本 LSI のハードウェアが自動的に実行します。

- (1) ファームウェアがホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoRESUME"を設定します(T0)。
- (2) ホストステートモニタ(H\_NegoControl\_0.HostState)を"RESUME"に設定します(T0)。
- (3) 切断検出機能およびリモート・ウェークアップ検出機能をオフします(T0)。
- (4) オペレーションモード(H\_XcvtControl.OpMode)を"Disable BS and NRZI"に設定し、レジューム"K"信号発行を開始します(T0)。
- (5) レジューム"K"信号発行を完了します(T1)。
- (6) ターミナル選択(H\_XcvtControl.TermSelect)を"HS"に設定します(T2)。
- (7) トランシーバ選択(H\_XcvtControl.XcvtSelect)を"HS"に設定します(T3)。
- (8) ファームウェアに対し、レジューム完了ステータス(H\_SIE\_IntStat\_1.ResumeCmp)を発行します(T3)。

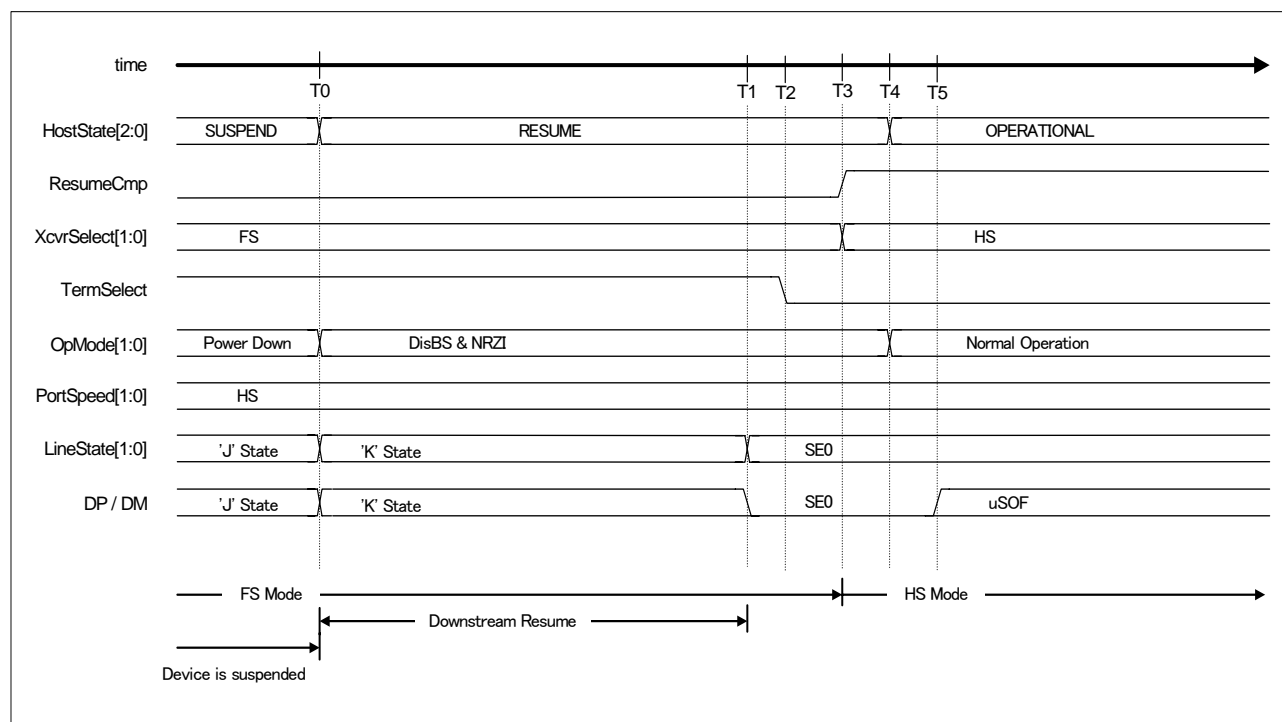


図 1-62 Resume Timing (HS mode)

## 1. 機能説明

表 1-57 Resume Timing Values (HS mode)

Timing Parameter	Description	Value
T0	H_NegoControl_0.AutoMode に"GoRESUME"を設定する。(F/W) 切断検出機能およびリモート・ウェークアップ検出機能をオフする。 オペレーションモードを"Disable BS and NRZI"に設定し、レジューム"K"信号の発行を開始する。(H/W)	0 (reference)
T1	レジューム"K"信号発行を完了する。 ターミナル選択を"HS"に設定する。(H/W)	$T0 + 20\text{ms}\{T_{\text{DRSMDN}}\} < T1$
T2	トランシーバ選択を"HS"に設定する。(H/W)	$T1 + 100\text{ns} < T2 < T1 + 2.0\mu\text{s}$
T3	レジューム完了ステータス(H_SIE_IntStat_1.ResumeCmp)を発行する。(H/W)	$T1 + 90\mu\text{s} < T3 < T1 + 110\mu\text{s}$
T4(参考)	GoOPERATIONAL を設定する。(F/W) オペレーションモードを"NomalOperation"に設定する。(H/W)	T4
T5(参考)	最初のマイクロ SOF を発行する。(H/W)	$T5 < T1 + 3\text{ms}$ $T4 + 120\mu\text{s} < T5 < T4 + 130\mu\text{s}$

注： {} は、USB2.0 規格書で規格されている名称である。



### 1.3.10.3.7.2. FS デバイスが接続されている場合

FS デバイスが接続されている場合の実行手順は以下のとおりです。下記(2)～(6)は本 LSI のハードウェアが自動的に実行します。

- (1) ファームウェアがホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoRESUME"を設定します(T0)。
- (2) ホストステートモニタ(H\_NegoControl\_0.HostState)を"RESUME"に設定します(T0)。
- (3) 切断検出機能およびリモート・ウェークアップ検出機能をオフします(T0)。
- (4) オペレーションモード(H\_XcvrControl.OpMode)を"Disable BS and NRZI"に設定し、レジューム"K"信号の発行を開始します(T0)。
- (5) レジューム"K"信号の発行を完了し(T1)、最後に LS bit time の EOP を付与します。
- (6) ファームウェアに対し、レジューム完了ステータス(H\_SIE\_IntStat\_1.ResumeCmp)を発行します(T2)。

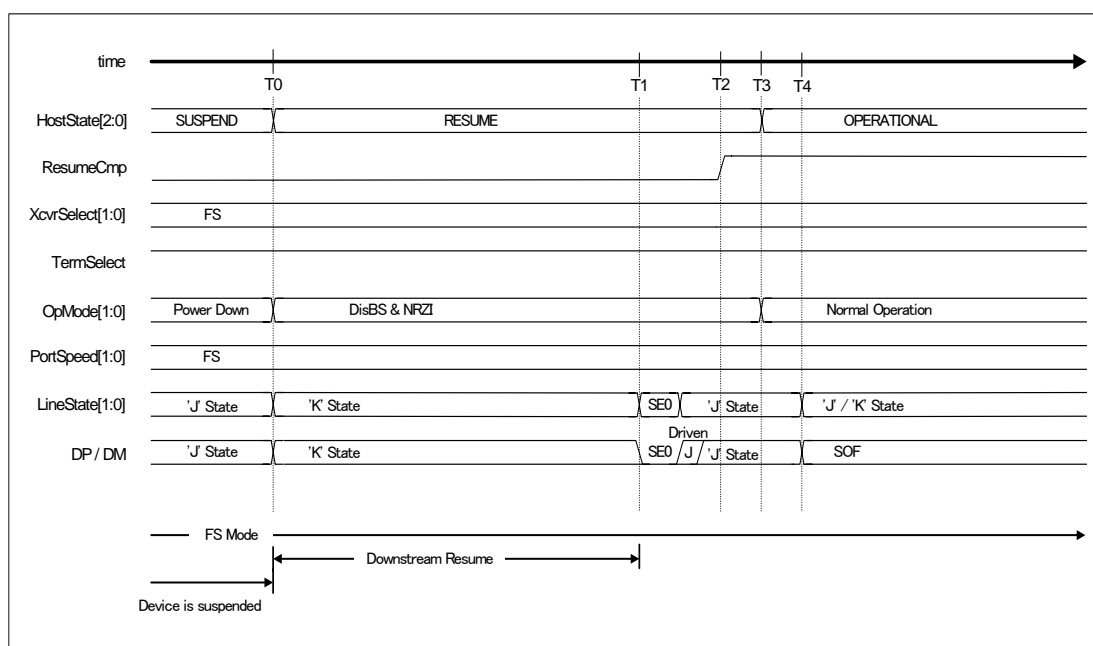


図 1-63 Resume Timing (FS mode)

## 1. 機能説明

表 1-58 Resume Timing Values (FS mode)

Timing Parameter	Description	Value
T0	H_NegoControl_0.AutoMode に"GoRESUME"を設定する。(F/W) 切断検出機能およびリモート・ウェークアップ検出機能をオフする。 オペレーションモードを"Disable BS and NRZI"に設定し、レジューム"K"信号の発行を開始する。(H/W)	0 (reference)
T1	レジューム"K"信号の発行を完了し、最後に LS bit time の EOP を付与する。(H/W)	$T0 + 20ms\{T_{DRSMDN}\} < T1$
T2	レジューム完了ステータス(H_SIE_IntStat_1.ResumeCmp)を発行する。(H/W)	$T1 + 90\mu s < T2 < T1 + 110\mu s$
T3(参考)	GoOPERATIONAL を設定する。(F/W) オペレーションモードを"NomalOperation"に設定する。(H/W)	T3
T4(参考)	最初の SOF を発行する。(H/W)	$T4 < T1 + 3ms$ $T3 + 0.9ms < T4 < T3 + 1.1ms$

注： {} は、USB2.0 規格書で規格されている名称である。

### 1.3.10.3.7.3. LS デバイスが接続されている場合

LS デバイスが接続されている場合の実行手順は以下のとおりです。下記(2)～(6)は本 LSI のハードウェアが自動的に実行します。

- (1) ファームウェアがホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoRESUME"を設定します(T0)。
- (2) ホストステートモニタ(H\_NegoControl\_0.HostState)を"RESUME"に設定します(T0)。
- (3) 切断検出機能およびリモート・ウェークアップ検出機能をオフします(T0)。
- (4) オペレーションモード(H\_XcvrControl.OpMode)を"Disable BS and NRZI"に設定し、レジューム"K"信号の発行を開始します(T0)。
- (5) レジューム"K"信号の発行を完了し(T1)、最後に LS bit time の EOP を付与します。
- (6) ファームウェアに対し、レジューム完了ステータス(H\_SIE\_IntStat\_1.ResumeCmp)を発行します(T2)。

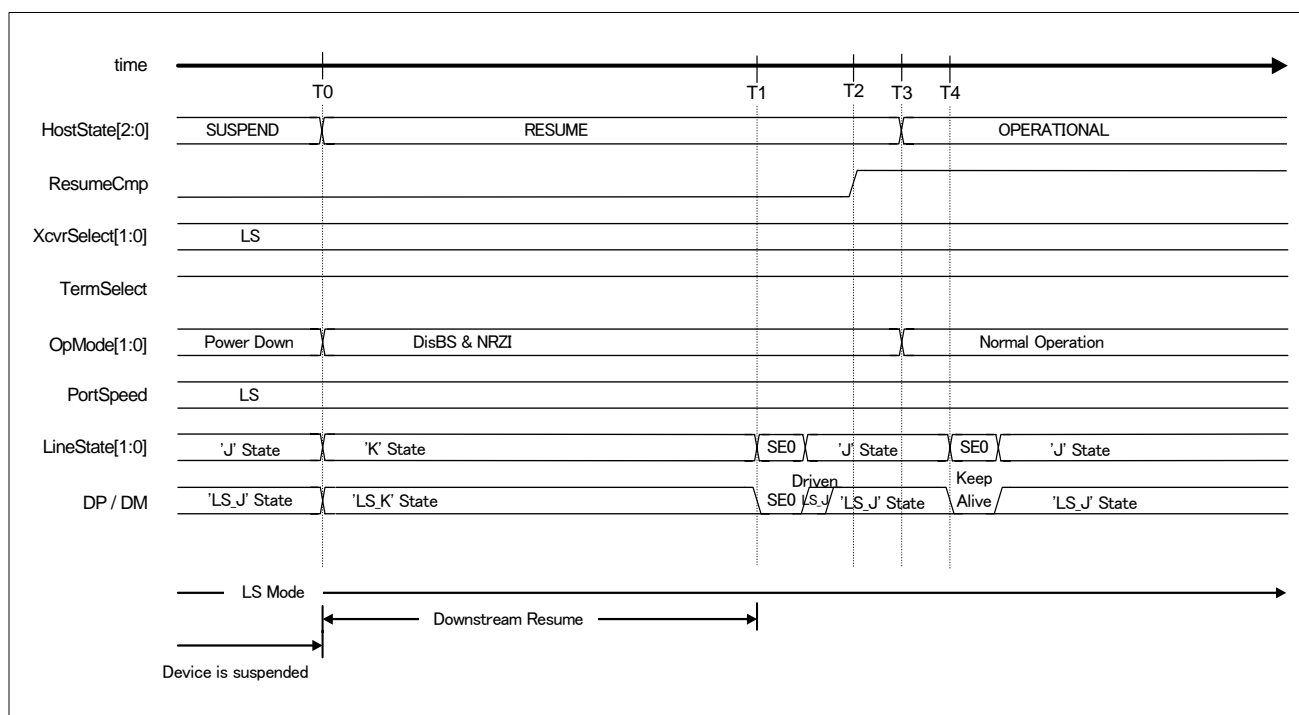


図 1-64 Resume Timing (LS mode)

表 1-59 Resume Timing Values (LS mode)

Timing Parameter	Description	Value
T0	H_NegoControl_0.AutoMode に"GoRESUME"を設定する。(F/W) 切断検出機能およびリモート・ウェークアップ検出機能をオフする。 オペレーションモードを"Disable BS and NRZI"に設定し、レジューム"K"信号の発行を開始する。(H/W)	0 (reference)
T1	レジューム"K"信号の発行を完了し、最後に LS bit time の EOP を付与する。(H/W)	$T0 + 20ms\{T_{DRSMDN}\} < T1$
T2	レジューム完了ステータス(H_SIE_IntStat_1.ResumeCmp)を発行する。(H/W)	$T1 + 90\mu s < T2 < T1 + 110\mu s$
T3(参考)	"GoOPERATIONAL"を設定する。(F/W) オペレーションモードを"NormalOperation"に設定する。(H/W)	T3
T4(参考)	最初の Keep Alive を発行する。(H/W)	$T4 < T1 + 3ms$ $T3 + 0.9ms < T4 < T3 + 1.1ms$

注： {} は、USB2.0 規格書で規格されている名称である。

# 1. 機能説明

## 1.3.10.3.8. GoWAIT\_CONNECTtoDIS

ホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoWAIT\_CONNECTtoDIS"を設定すると、"WAIT\_CONNECT"ステートから"DISABLED"ステートに至る必要な処理を本 LSI のハードウェアが自動的に実行します。

実行手順は以下のとおりです。下記(2)～(5)は本 LSI のハードウェアが自動的に実行します。

- (1) ファームウェアがホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoWAIT\_CONNECTtoDIS"を設定します(T0)。
- (2) "GoWAIT\_CONNECT"と同等の処理を実行します(T0)。
- (3) 接続検出を行い、接続検出ステータス(H\_SIE\_IntStat\_0.DetectCon)を発行します(T1)。
- (4) "GoDISABLED"と同等の処理を実行します(T1)。
- (5) ディセーブルド完了ステータス(H\_SIE\_IntStat\_1.DisabledCmp)を発行します(T2)。

なお、各ステートにおけるタイミングは、"GoWAIT\_CONNECT"、"GoDISABLED"を実行した場合と同じです。タイミングの詳細は、それぞれ「1.3.10.3.2.GoWAIT\_CONNECT」、「1.3.10.3.3.GoDISABLED」を参照してください。

また、途中で異常(切断、VBUS異常)を検出した場合の実行手順およびタイミングについては、「1.3.10.2.2.切断検出」、「1.3.10.2.1.VBUS異常検出」を参照してください。

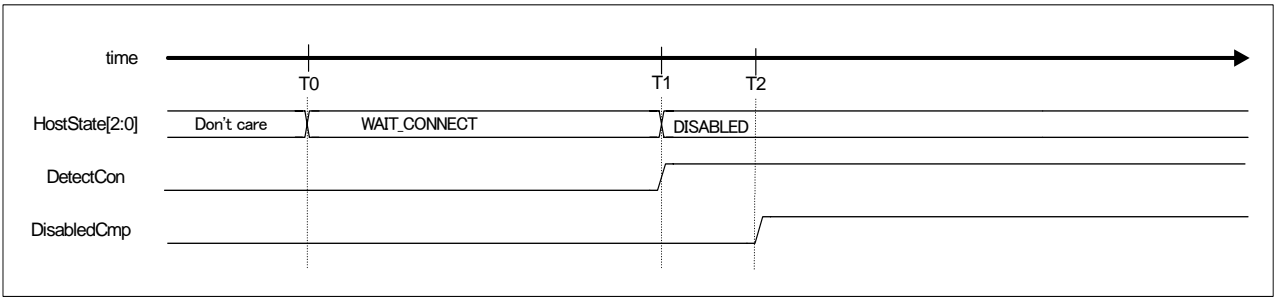


図 1-65 GoWAIT\_CONNECTtoDIS Timing (HS mode)

表 1-60 GoWAIT\_CONNECTtoDIS Timing Values (HS mode)

Timing Parameter	Description	Value
T0	H_NegoControl_0.AutoMode に "GoWAIT_CONNECTtoDIS"を設定する。(F/W) GoWAIT_CONNECT と同等の処理を実行する。(H/W)	0 (reference)
T1	接続検出を行い、接続検出ステータスを発行する。 GoDISABLED と同等の処理を実行する。(H/W)	T1
T2	ディセーブルド完了ステータスを発行する。(H/W)	T2

### 1.3.10.3.9. GoWAIT\_CONNECTtoOP

ホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoWAIT\_CONNECTtoOP"を設定すると、"WAIT\_CONNECT"ステートから"OPERATIONAL"ステートに至る必要な処理を本 LSI のハードウェアが自動的に実行します。

#### 1.3.10.3.9.1. HS デバイスが接続されている場合

HS デバイスが接続されている場合の実行手順は以下のとおりです。下記(2)～(9)は本 LSI のハードウェアが自動的に実行します。

- (1) ファームウェアがホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoWAIT\_CONNECTtoOP"を設定します(T0)。
- (2) "GoWAIT\_CONNECT"と同等の処理を実行します(T0)。
- (3) 接続検出を行い、接続検出ステータス(H\_SIE\_IntStat\_0.DetectCon)を発行します(T1)。
- (4) "GoDISABLED"と同等の処理を実行します(T1)。
- (5) ディセーブルド完了ステータス(H\_SIE\_IntStat\_1.DisabledCmp)を発行します(T2)。
- (6) "GoRESET"と同等の処理を実行します(T2)。
- (7) デバイスの Chirp を検出し、デバイスチャープ正常検出ステータス(H\_SIE\_IntStat\_0.DetectDevChirpOK)を発行します(T3)。
- (8) リセット完了ステータス(H\_SIE\_IntStat\_1.ResetCmp)を発行します(T4)。
- (9) "GoOPERATIONAL"と同等の処理を実行します(T4)。

なお、各ステートにおけるタイミングは、"GoWAIT\_CONNECT"、"GoDISABLED"、"GoRESET"、"GoOPERATIONAL"を実行した場合と同じです。タイミングの詳細は、それぞれ「1.3.10.3.2.GoWAIT\_CONNECT」、「1.3.10.3.3.GoDISABLED」、「1.3.10.3.4.GoRESET1.3.10.3.5.GoOPERATIONAL」を参照してください。

また、途中で異常(切断、VBUS異常、デバイスチャープ異常)を検出した場合の実行手順およびタイミングについては、「1.3.10.2.2.切断検出」、「1.3.10.2.1.VBUS異常検出」を参照してください。

# 1. 機能説明

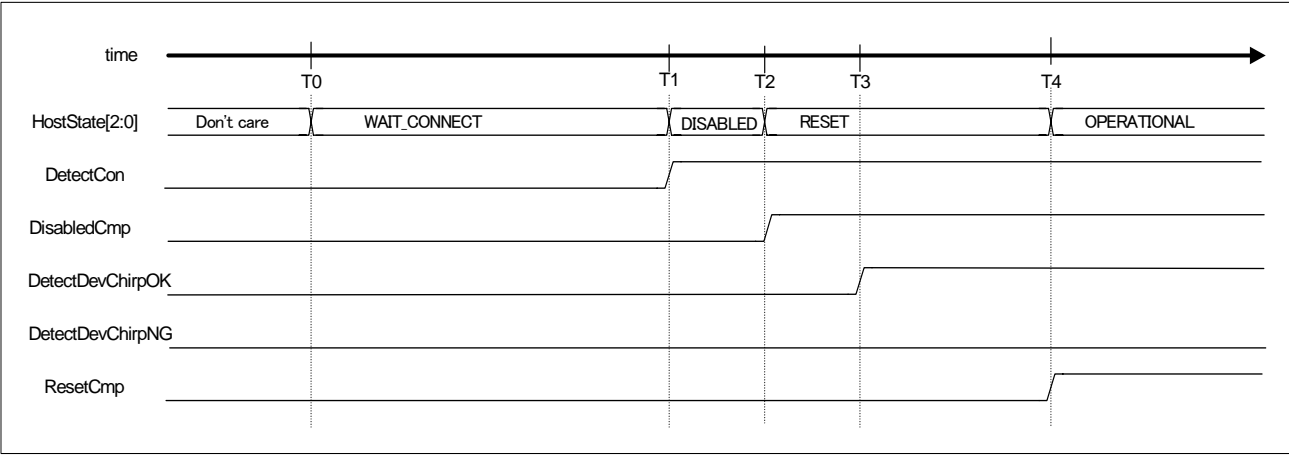


図 1-66 GoWAIT\_CONNECTtoOP Timing (HS mode)

表 1-61 GoWAIT\_CONNECTtoOP Timing Values (HS mode)

Timing Parameter	Description	Value
T0	H_NegoControl_0.AutoMode に "GoWAIT_CONNECTtoOP"を設定する。(F/W) "GoWAIT_CONNECT"と同等の処理を実行する。(H/W)	0 (reference)
T1	接続検出を行い、接続検出ステータスを発行する。 "GoDISABLED"と同等の処理を実行する。(H/W)	T1
T2	ディセーブルド完了ステータスを発行する。 "GoRESET"と同等の処理を実行する。(H/W)	T2
T3	デバイスチャープ検出を行い、デバイスチャープ正常検出ステータスを発行する。(H/W)	T3
T4	リセット完了ステータスを発行する。 "GoOPERATIONAL"と同等の処理を実行する。(H/W)	T4

### 1.3.10.3.9.2. FS または LS デバイスが接続されている場合

FS または LS デバイスが接続されている場合の実行手順は以下のとおりです。下記(2)～(9)は本 LSI のハードウェアが自動的に実行します。

- (1) ファームウェアがホストステート遷移実行 (H\_NegoControl\_0.AutoMode) に "GoWAIT\_CONNECTtoOP"を設定します(T0)。
- (2) "GoWAIT\_CONNECT"と同等の処理を実行します(T0)。
- (3) 接続検出を行い、接続検出ステータス(H\_SIE\_IntStat\_0.DetectCon)を発行します(T1)。
- (4) "GoDISABLED"と同等の処理を実行します(T1)。
- (5) ディセーブルド完了ステータス(H\_SIE\_IntStat\_1.DisabledCmp)を発行します(T2)。
- (6) "GoRESET"と同等の処理を実行します(T2)。
- (7) デバイスの Chirp を検出しないため、デバイスチャープ正常／異常検出ステータス (H\_SIE\_IntStat\_0.DetectDevChirpOK/NG)は発行しません(T3)。
- (8) リセット完了ステータス(H\_SIE\_IntStat\_1.ResetCmp)を発行します(T4)。
- (9) "GoOPERATIONAL"と同等の処理を実行します(T4)。

なお、各ステートにおけるタイミングは、"GoWAIT\_CONNECT"、"GoDISABLED"、"GoRESET"、"GoOPERATIONAL" を実行した場合と同じです。タイミングの詳細はそれぞれ「1.3.10.3.2.GoWAIT\_CONNECT」、「1.3.10.3.3.GoDISABLED」、「1.3.10.3.4.GoRESET1.3.10.3.5.GoOPERATIONAL」を参照してください。

また、途中で異常(切断、VBUS異常)を検出した場合の実行手順およびタイミングについては、「1.3.10.2.2.切断検出」、「1.3.10.2.1.VBUS異常検出」を参照してください。

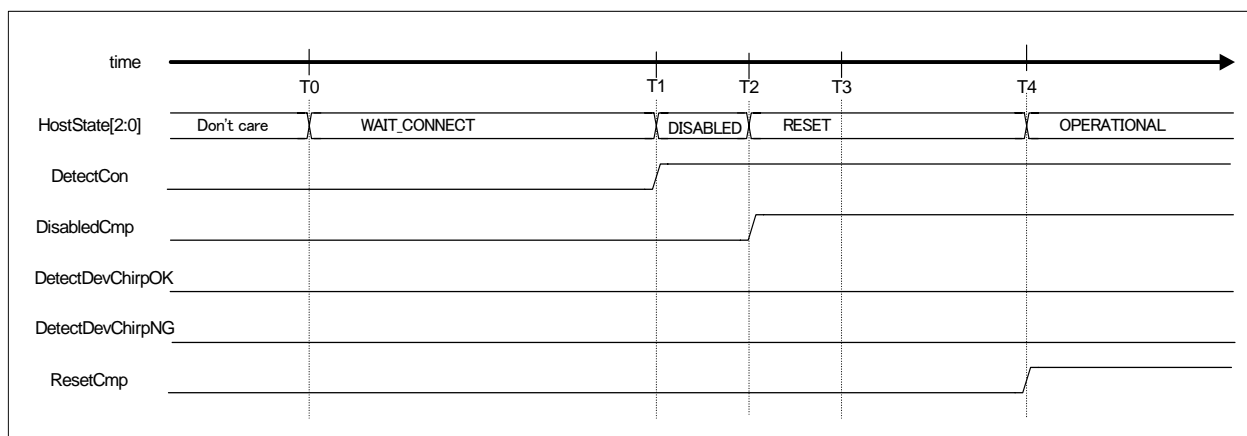


図 1-67 GoWAIT\_CONNECTtoOP Timing (FS または LS mode)

表 1-62 GoWAIT\_CONNECTtoOP Timing Values (FS または LS mode)

Timing Parameter	Description	Value
T0	H_NegoControl_0.AutoMode に GoWAIT_CONNECTtoOP を設定する。(F/W) GoWAIT_CONNECT と同等の処理を実行する。(H/W)	0 (reference)
T1	接続検出を行い、接続検出ステータスを発行する。 GoDISABLED と同等の処理を実行する。(H/W)	T1
T2	ディセーブルド完了ステータスを発行する。 GoRESET と同等の処理を実行する。(H/W)	T2
T3	デバイスチャープ非検出のため、デバイスチャープ正常／異常検出ステータスを発行しない。(H/W)	T3
T4	リセット完了ステータスを発行する。 GoOPERATIONAL と同等の処理を実行する。(H/W)	T4

## 1. 機能説明

### 1.3.10.3.10. GoRESETtoOP

ホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoRESETtoOP"を設定すると、"RESET"ステートから"OPERATIONAL"ステートに至る必要な処理を本 LSI のハードウェアが自動的に実行します。

#### 1.3.10.3.10.1. HS デバイスが接続されている場合

HS デバイスが接続されている場合の実行手順は以下のとおりです。下記(2)～(5)は本 LSI のハードウェアが自動的に実行します。

- (1) ファームウェアがホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoRESETtoOP"を設定します(T0)。
- (2) "GoRESET"と同等の処理を実行します(T0)。
- (3) デバイスの Chirp を検出し、デバイスチャープ正常検出ステータス(H\_SIE\_IntStat\_0.DetectDevChirpOK)を発行します(T1)。
- (4) リセット完了ステータス(H\_SIE\_IntStat\_1.ResetCmp)を発行します(T2)。
- (5) "GoOPERATIONAL"と同等の処理を実行します(T2)。

なお、各ステートにおけるタイミングは、"GoRESET"、"GoOPERATIONAL"を実行した場合と同じです。タイミングの詳細はそれぞれ「1.3.10.3.4.GoRESET1.3.10.3.5.GoOPERATIONAL」を参照してください。

また、途中で異常(VBUS異常およびデバイスチャープ異常)を検出した場合の実行手順およびタイミングについては、「1.3.10.2.2.切断検出」、「1.3.10.2.1.VBUS異常検出」を参照してください。

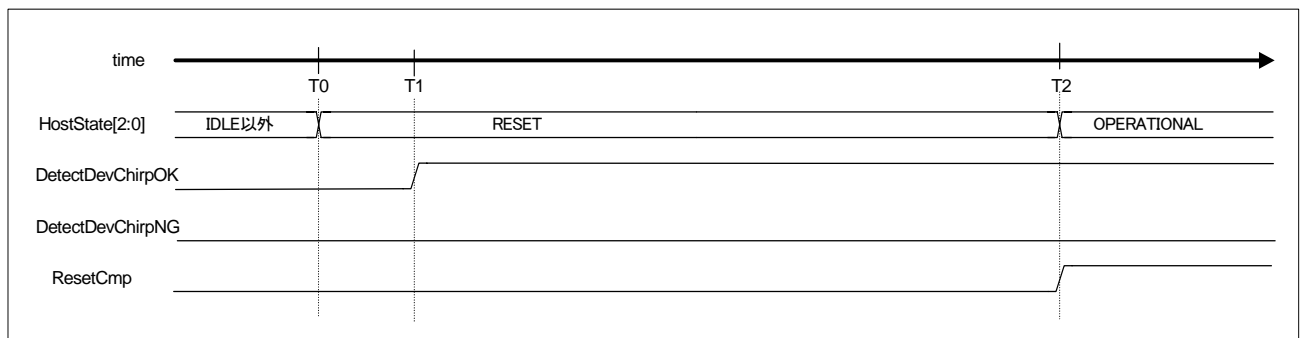


図 1-68 GoRESETtoOP Timing (HS mode)

表 1-63 GoRESETtoOP Timing Values (HS mode)

Timing Parameter	Description	Value
T0	H_NegoControl_0.AutoMode に"GoRESETtoOP"を設定する。(F/W) "GoRESET"と同等の処理を実行する。(H/W)	0 (reference)
T1	デバイスチャープ検出を行い、デバイスチャープ正常検出ステータスを発行する。(H/W)	T1
T2	リセット完了ステータスを発行する。 "GoOPERATIONAL"と同等の処理を実行する。(H/W)	T2



### 1.3.10.3.10.2. FS または LS デバイスが接続されている場合

FS または LS デバイスが接続されている場合の実行手順は以下のとおりです。下記(2)～(5)は本 LSI のハードウェアが自動的に実行します。

- (1) ファームウェアがホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoRESETtoOP"を設定します(T0)。
- (2) "GoRESET"と同等の処理を実行します(T0)。
- (3) デバイスの Chirp を検出しないため、デバイスチャープ正常／異常検出ステータス(H\_SIE\_IntStat\_0.DetectDevChirpOK/NG)は発行しません(T1)。
- (4) リセット完了ステータス(H\_SIE\_IntStat\_1.ResetCmp)を発行します(T2)。
- (5) "GoOPERATIONAL"と同等の処理を実行します(T2)。

なお、各ステートにおけるタイミングは、GoRESET、GoOPERATIONALを実行した場合と同じです。タイミングの詳細は、それぞれ「1.3.10.3.4.GoRESET1.3.10.3.5.GoOPERATIONAL」を参照してください。

また、途中で異常(VBUS 異常)を検出した場合の実行手順およびタイミングについては、「1.3.10.2.1.VBUS異常検出」を参照してください。

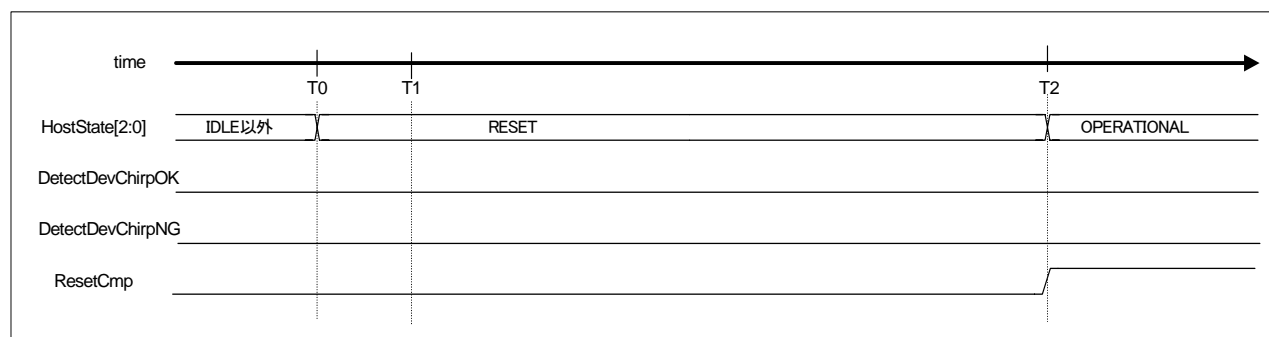


図 1-69 GoRESETtoOP Timing (FS または LS mode)

表 1-64 GoRESETtoOP Timing Values (FS または LS mode)

Timing Parameter	Description	Value
T0	H_NegoControl_0.AutoMode に"GoRESETtoOP"を設定する。(F/W) "GoRESET"と同等の処理を実行する。(H/W)	0 (reference)
T1	デバイスチャープ非検出のため、デバイスチャープ正常／異常検出ステータスを発行しない。(H/W)	T1
T2	リセット完了ステータスを発行する。 "GoOPERATIONAL"と同等の処理を実行する。(H/W)	T2

## 1. 機能説明

### 1.3.10.3.11. GoSUSPENDtoOP

ホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoSUSPENDtoOP"を設定すると、"SUSPEND"ステートから"OPERATIONAL"ステートに至る必要な処理を本 LSI のハードウェアが自動的に実行します。

"GoSUSPENDtoOP"を設定した場合は、リモート・ウェークアップ検出機能をハードウェアが自動的にオン／オフします(ただしリモート・ウェークアップ受付許可(H\_NegoControl\_1.RmtWkupDetEnb)には反映しません)ので、ファームウェアがリモート・ウェークアップ受付許可(H\_NegoControl\_1.RmtWkupDetEnb)を操作する必要はありません。

なお、本設定を使用する際には、パワーマネジメント機能を使用しないでください。

本設定による実行手順は以下のとおりです。下記(2)～(7)は本 LSI のハードウェアが自動的に実行します。

- (1) ファームウェアがホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoSUSPENDtoOP"を設定します(T0)。
- (2) "GoSUSPEND"と同等の処理を実行します(T0)。
- (3) サスペンド遷移完了ステータス(H\_SIE\_IntStat\_1.SuspendCmp)を発行します(T1)。
- (4) リモート・ウェークアップを検出し、リモート・ウェークアップを検出ステータス(H\_SIE\_IntStat\_0.DetectRmtWkup)を発行します(T2)。
- (5) "GoRESUME"と同等の処理を実行します(T2)。
- (6) レジューム完了ステータス(H\_SIE\_IntStat\_1.ResumeCmp)を発行します(T3)。
- (7) "GoOPERATIONAL"と同等の処理を実行します(T3)。

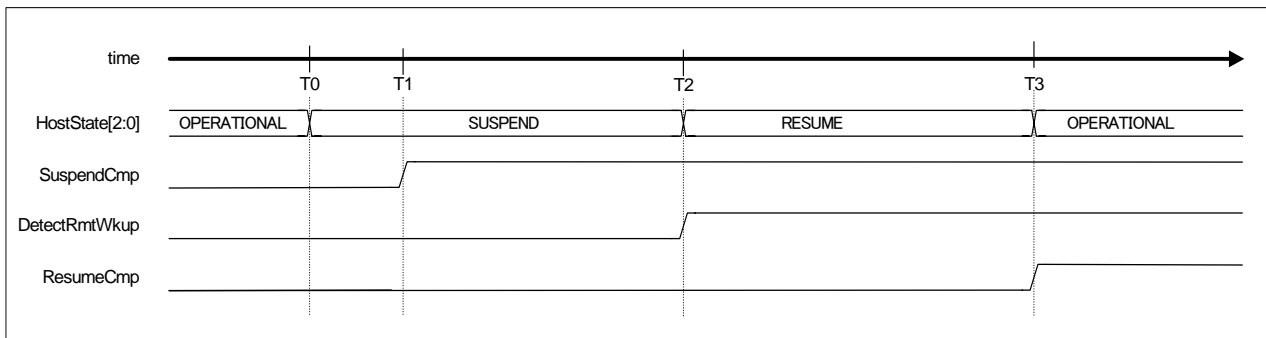


図 1-70 GoSUSPENDtoOP Timing

表 1-65 GoSUSPENDtoOP Timing Value

Timing Parameter	Description	Value
T0	H_NegoControl_0.AutoMode に"GoSUSPENDtoOP"を設定する。(F/W) "GoSUSPEND"と同等の処理を実行する。(H/W)	0 (reference)
T1	サスペンド遷移完了ステータスを発行する。(H/W)	T1
T2	リモート・ウェークアップを検出し、リモート・ウェークアップ検出ステータスを発行する。"GoRESUME"と同等の処理を実行する。(H/W)	T2
T3	レジューム完了ステータスを発行する。 "GoOPERATIONAL"と同等の処理を実行する。(H/W)	T3

1.3.10.3.12. GoRESUMEtoOP

ホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoRESUMEtoOP"を設定すると、"RESUME"ステートから"OPERATIONAL"ステートに至る必要な処理を本 LSI のハードウェアが自動的に実行します。

本設定による実行手順は以下のとおりです。下記(2)～(4)は本 LSI のハードウェアが自動的に実行します。

- (1) ファームウェアがホストステート遷移実行(H\_NegoControl\_0.AutoMode)に"GoRESUMEtoOP"を設定します(T0)。
- (2) "GoRESUME"と同等の処理を実行します(T0)。
- (3) レジューム完了ステータス(H\_SIE\_IntStat\_1.ResumeCmp)を発行します(T1)。
- (4) "GoOPERATIONAL"と同等の処理を実行します(T1)。

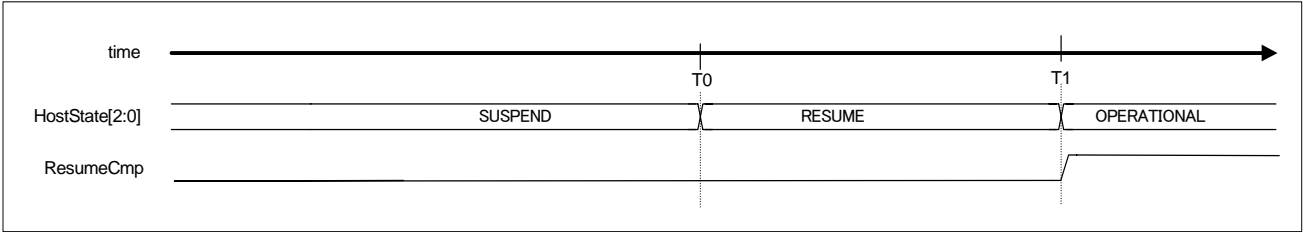


図 1-71 GoRESUMEtoOP

表 1-66 GoRESUMEtoOP Timing Value

Timing Parameter	Description	Value
T0	H_NegoControl_0.AutoMode に"GoRESUMEtoOP"を設定する。(F/W) "GoRESUME"と同等の処理を実行する。(H/W)	0 (reference)
T1	レジューム完了ステータスを発行する。 "GoOPERATIONAL"と同等の処理を実行する。(H/W)	T1

## 1. 機能説明

### 1.4. メディアデータ転送機能

#### 1.4.1. メディアデータ

機器が扱うデータはアプリケーションにより様々であり、例えば音楽データや映像データなどがあります。これらのデータを総称して、本書ではメディアデータと呼びます。

#### 1.4.2. メディアデータ転送

本 LSI は、USB 転送に使用する FIFO とは別に、メディアデータの転送に使用できる 64 バイトの FIFO(メディア FIFO)を有します。この FIFO を使用して、HDD とメモリ間でのメディアデータの転送が容易に実現できます。

このように、メディア FIFO を介して HDD とメモリ間の転送を行う本 LSI の機能を、本書ではメディアデータ転送機能と呼びます。

メディアデータ転送の様子を再生時(HDD→メモリ)を例に示します。

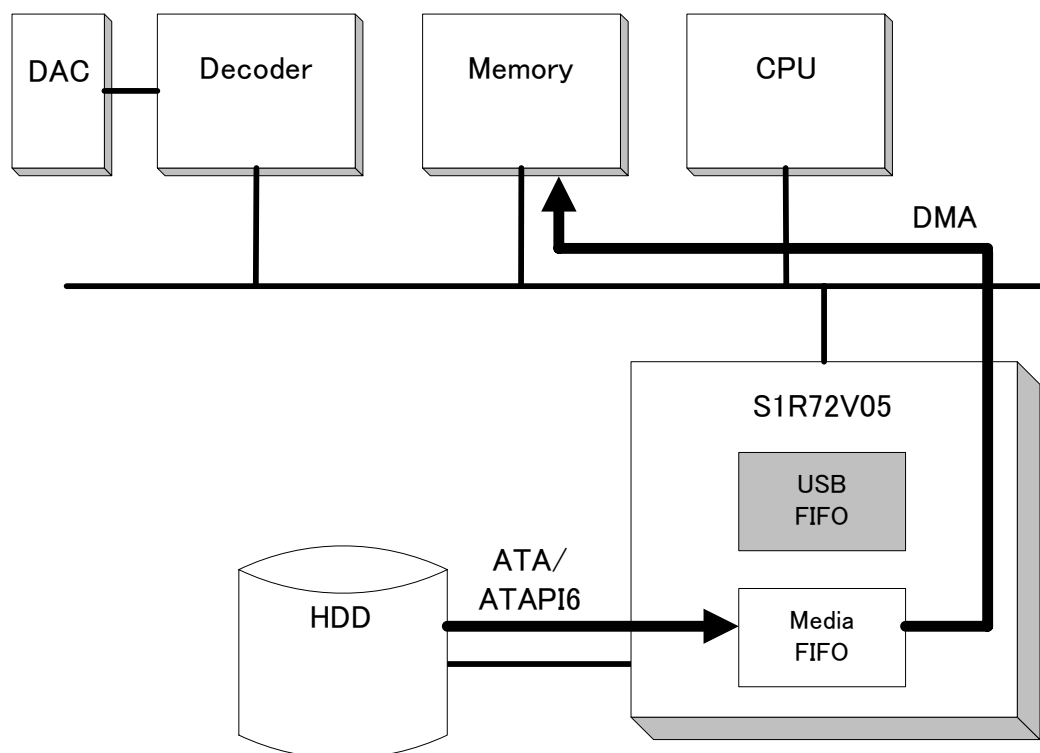


図 1-72 メディアデータ転送例(再生時)

メディア FIFO へのアクセスは、MediaFIFO\_Join レジスタにより、IDE ポートと、もう一方のポート (DMA0/1, CPU\_Rd, CPU\_Wr のいずれか)を指定することで行います。

メディアFIFOへのアクセス方法の詳細については、「1.7.メディアFIFO管理」を参照してください。

### 1.4.3. 消費電力の削減

メディアデータ転送機能は、本 LSI の電力モードステートのうち、ACTIVE60、ACT\_DEVICE、ACT\_HOST、ACT\_ALL のいずれかのステートで使用することができます。このうち、ACTIVE60 ステートでは、USB デバイス部および USB ホスト部の機能を停止しますので、消費電力を大きく削減することができます。

パワーマネジメント機能の詳細は、「1.5.パワーマネジメント機能」を参照してください。

USB 転送とメディアデータ転送を行う場合、およびメディアデータ転送のみを行う場合のデータフローの様子をそれぞれ下図に示します。

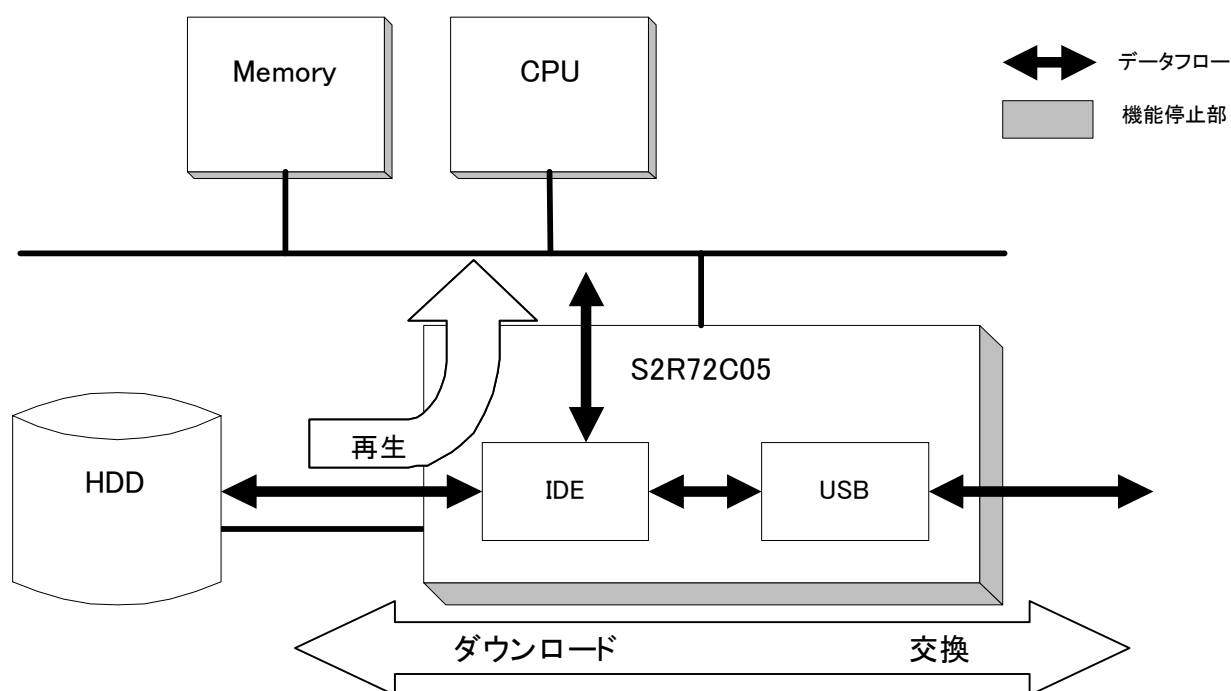


図 1-73 USB 転送とメディアデータ転送(例：再生時)

## 1. 機能説明

---

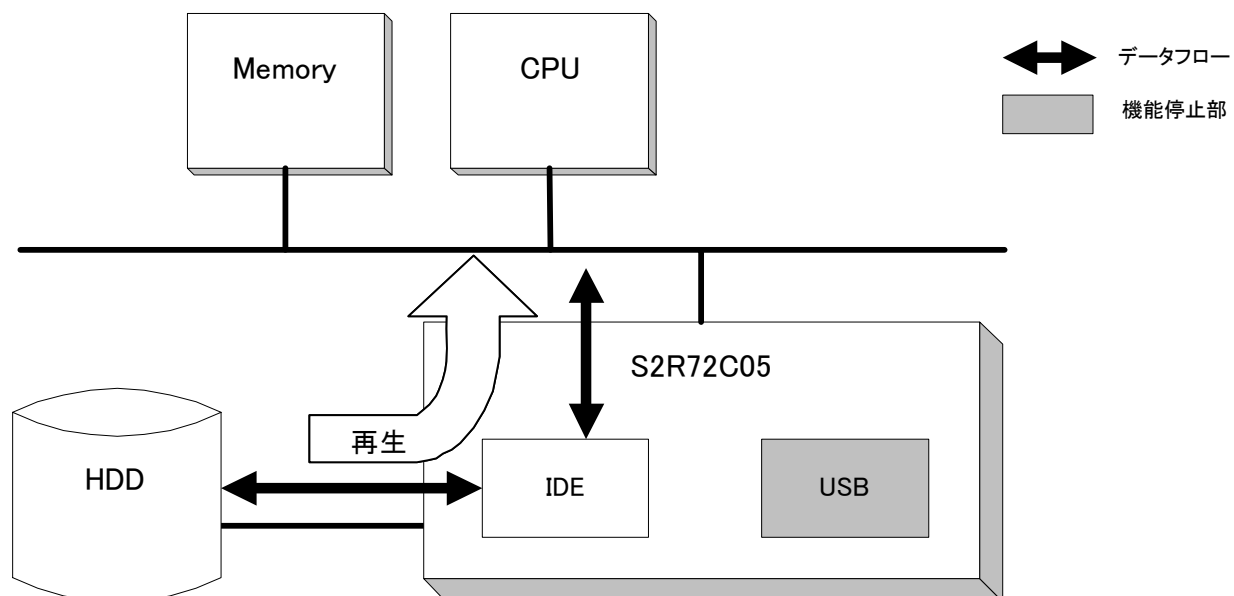


図 1-74 メディアデータ転送(例：再生時)のみ

## 1.5. パワーマネージメント機能

まず、本 IC を使用する場合には、2 ポートモードにて使用するか、1 ポートモードにて使用するかを ClkSelect.Port1x2 ビットにより適切に設定してください。2 ポートモードとは、USB ポート A をホスト、USB ポート B をデバイスとして使用するモードです。1 ポートモードとは、USB ポート B を、ホスト、デバイスで共用します。初期値は 0 にクリアされているため 2 ポートモードとなっており、1 にセットすることで 1 ポートモードに設定できます。なお、ClkSelect.Port1x2 ビットの切り替えは、後述の SLEEP ステートもしくは SNOOZE ステートにて行ってください。

### 1.5.1. 2 ポートモード時 (ClkSelect.Port1x2 が 0 にクリアされている場合)

パワーマネージメント機能は、オシレータ、PLL(DevicePLL480, HostPLL480, PLL60)の動作を制御し、SLEEP ステート、SNOOZE ステート、ACTIVE60 ステート、ACT\_DEVICE ステート、ACT\_HOST ステート、ACT\_ALL ステートの 6 ステートを遷移します。他のステートに遷移する場合には、PM\_Control\_0.GoSLEEP、PM\_Control\_0.GoSNOOZE、PM\_Control\_0.GoActive60、PM\_Control\_0.GoActDevice、PM\_Control\_0.GoActHost、PM\_Control\_0.GoActA 1 1 Dev、PM\_Control\_0.GoActA 1 1 Host ビットをセットすることで開始され、任意の処理が行われた後終了します。現在、どのステートにいるかを確認するには、PM\_Control\_1.PM\_State[3:0]を確認してください。また遷移終了時、MainIntStat.FinishedPM イベントを発生します。この時、MainIntEnb.EnFinishedPM ビットをセットしていれば、割り込み XINT が発生します。

全てのステートから他のステートへ遷移することが可能で、ACT\_DEVICE ステートまたは ACT\_HOST ステート時に PM\_Control\_0.GoSLEEP ビットをセットした場合は、ACTIVE60 ステート、SNOOZE ステートを経由し、SLEEP ステートに遷移し、完全に遷移終了した時点で MainIntStat.FinishedPM イベントを発生します。また SLEEP ステート時に PM\_Control\_0.GoActDevice ビットまたは PM\_Control\_0.GoActHost ビットをセットした場合も、SNOOZE ステート、ACTIVE60 ステートを経由し、ACT\_DEVICE ステートまたは ACT\_HOST ステートに遷移し、完全に遷移終了した時点で MainIntStat.FinishedPM イベントを発生します。同様に、ACTIVE60 ステート時に PM\_Control\_0.GoSLEEP ビットをセットした場合も、SNOOZE ステートを経由し、SLEEP ステートに遷移し、完全に遷移終了した時点で MainIntStat.FinishedPM イベントを発生します。また SLEEP ステート時に PM\_Control\_0.GoActive60 ビットをセットした場合も、SNOOZE ステートを経由し、ACTIVE60 ステートに遷移し、完全に遷移終了した時点で MainIntStat.FinishedPM イベントを発生します。また、ACT\_DEVICE ステート時に PM\_Control\_0.GoActHost ビットをセットした場合は、ACTIVE60 ステートを経由して ACT\_HOST ステートに遷移し、完全に遷移終了した時点で MainIntStat.FinishedPM イベントを発生します。同様に、ACT\_HOST ステート時に PM\_Control\_0.GoActDevice ビットをセットした場合は、ACTIVE60 ステートを経由して ACT\_DEVICE ステートに遷移し、完全に遷移終了した時点で MainIntStat.FinishedPM イベントを発生します。

同様に、SLEEP ステート、SNOOZE ステート、ACTIVE60 ステートから ACT\_ALL ステートへの遷移には、PM\_Control\_0.GoActA 1 1 Dev または PM\_Control\_0.GoActAllHost ビットをセットすることで、各途中ステートを遷移し、完全に ACT\_ALL ステートに遷移終了した時点で MainIntStat.FinishedPM イベントを発生します。逆に、ACT\_ALL ステートから SLEEP ステート、SNOOZE ステート、ACTIVE60 ステートへの遷移には、各々 PM\_Control\_0.GoSLEEP、PM\_Control\_0.GoSnooze、PM\_Control\_0.GoActive60 ビットをセットすることで可能です。この場合も、完全に遷移が終了した時点で MainIntStat.FinishedPM イベントを発生します。

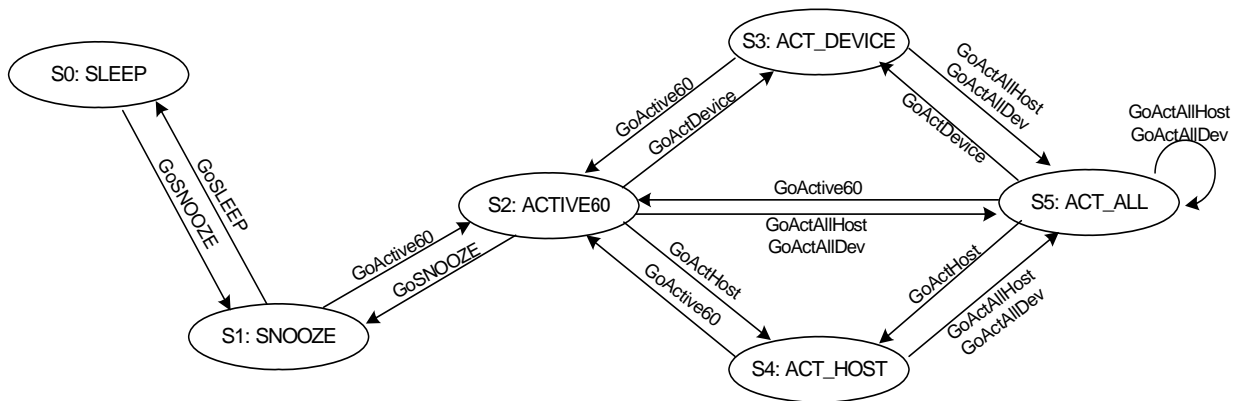
なお、ACT\_ALL ステートを除いて、現在のステートから同じステートへの遷移を行うことは出来ないことに注意してください。このとき、セットした PM\_Control\_0.GoXXXX ビットはクリアされません。例えば、ACTIVE60 ステートで動作している状態で、PM\_Control\_0.GoActive60 をセットした場合には、遷移を行わず(見かけ上正しいが)、PM\_Control\_0.GoActive60 ビットがセットされたままとなります。したがって、動作が正常に行われなくなります。

ACT\_ALL ステートにおいては、DevicePLL480 及び HostPLL480 の両 PLL が動作しており、両機能が使用できます。ただし、転送はどちらか一方しか行えません。ホストにて転送を行っていて、デバイスの転送を行いたい場合には、PM\_Control\_0.GoActAllDev ビットをセットし、デバイス転送モードにしてください。遷移終了後(瞬時に)、その他のステートへの遷移と同様に、MainIntStat.FinishedPM イベント

## 1. 機能説明

トを発生し、同時に PM\_Control\_1.MonActFunc ビットをクリアします。同様に、デバイスにて転送を行っていて、ホストの転送を行いたい場合には、PM\_Control\_0.GoActAllHost ビットをセットし、ホスト転送モードにしてください。遷移終了後(瞬時に)、その他のステートへの遷移と同様に、MainIntStat.FinishedPM イベントを発生し、同時に PM\_Control\_1.MonActFunc ビットをセットします。したがって、パワーマネジメントとしてのステートとしては、ACT\_ALL ステートのみが存在しており、ACT\_ALL\_DEVICE ステート及び ACT\_ALL\_HOST ステートは存在せず、PM\_Control\_1.MonActFunc によるサブステートとなります。

この PM\_Control\_1.MonActFunc ビットは、ACT\_ALL ステートにおけるセット／クリア以外に、ACT\_DEVICE ステートでもクリアされ、また ACT\_HOST ステートでもセットされます。したがって、2ポートモード時には、デバイス／ホストのどちらの転送が実行可能かを表しています。ただし、SLEEP ステート、SNOOZE ステート、ACTIVE60 ステートにおいては、このビットは意味がありません。



※ 最終目標ステートのGoXXXXを行うことで、全てのステートから他のステートへの遷移が可能

※ ACT\_ALLステートは、ACT\_ALL.DEVステート、ACT\_ALL.HOSTステートを含んでいます。

図 1-75 2ポートモード時のパワーマネジメント



### 1.5.1.1.SLEEP(スリープ)

オシレータが発振していないステートです。したがって、このステートでは、PLL も発振していません。

SNOOZE ステート、ACTIVE60 ステート、ACT\_DEVICE ステート、ACT\_HOST ステート及び ACT\_ALL ステート中に PM\_Control\_0.GoSLEEP ビットをセットし、スリープに遷移する場合は、動作している PLL(DevicePLL480 または HostPLL480、もしくはその両方)を停止し、続いて PLL60 を停止し、最後に OSCCLK の出力を停めた後に発振を停止します。

逆に、SLEEP ステート中に PM\_Control\_0.GoSNOOZE、PM\_Control\_0.GoActive60 ビット、PM\_Control\_0.GoActDevice ビット、PM\_Control\_0.GoActHost ビット、PM\_Control\_0.GoActAllDev ビット、または、PM\_Control\_0.GoActAllHost ビットをセットし、スリープからスヌーズへ遷移する時には、オシレータが安定して発振するまで内部回路には OSCCLK を与えないように、発振安定時間ゲートしています。この発振安定時間は、発振セル、発振子、周辺回路及び基板によって変化しますので、WakeUpTim\_H,L レジスタを用いて設定してください。この WakeUpTim\_H,L レジスタは、非同期でアクセス可能ですので、SLEEP ステートでも読み出し、書き込みが可能です。

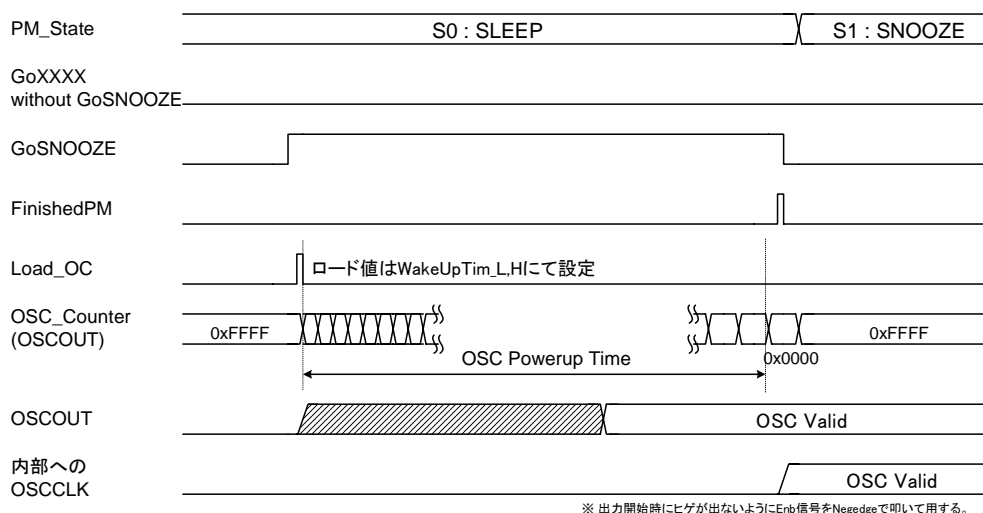


図 1-76 SLEEP ステートからの離脱(GoSNOOZE 時)

### 1.5.1.2.SNOOZE(スヌーズ)

オシレータは発振している状態で、PLL が発振していないステートです。

ACTIVE60 ステート、ACT\_DEVICE ステート、ACT\_HOST ステート及び ACT\_ALL ステート中に PM\_Control\_0.GoSNOOZE ビットをセットし、スヌーズに遷移する場合は、出力しているクロックを停止した後、DevicePLL480 または HostPLL480、もしくはその両方を停止し、最後に PLL60 を停止します。

逆に、SNOOZE ステート中に PM\_Control\_0.GoActive60 ビット、PM\_Control\_0.GoActDevice ビット、PM\_Control\_0.GoActHost ビット、PM\_Control\_0.GoActAllHost ビット、または、PM\_Control\_0.GoActAllDev ビットをセットし、スヌーズからアクティブへ遷移する時には、PLL が安定して発振するまで内部回路には SCLK を与えないように、PLL 安定時間(約 250us)ゲートします。

## 1. 機能説明

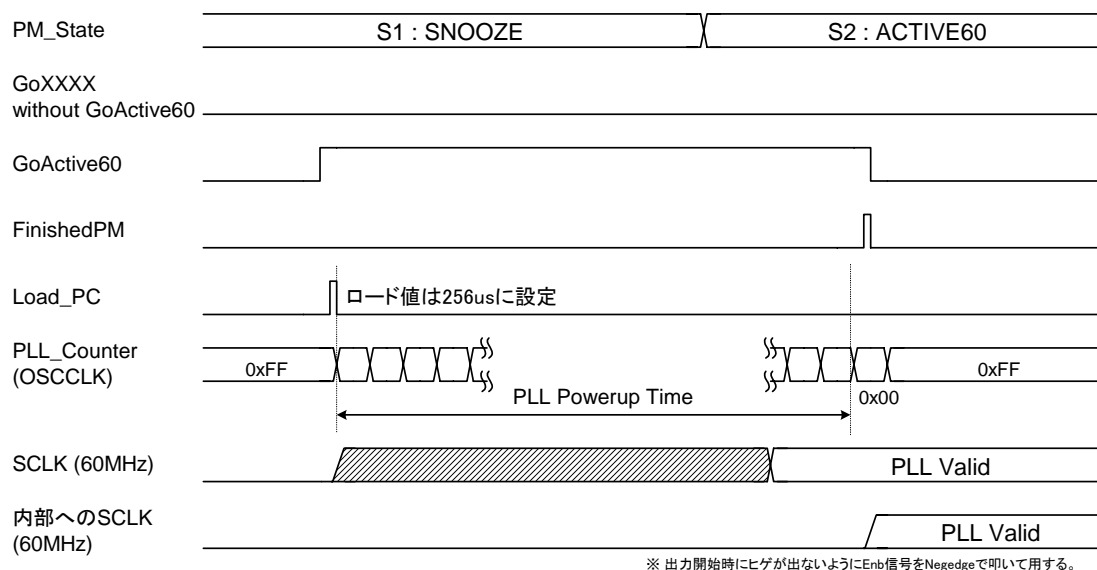


図 1-77 SNOOZE ステートからの離脱(GoActive60 時)

### 1.5.1.3.ACTIVE60(アクティブ 60)

オシレータ、PLL60 が動作し、DevicePLL480 およびHostPLL480 が停止しているステートです。レジスタマップの**斜体太字**のレジスタ及びビットはスヌーズ及びスリープ時でも読み書き可能なレジスタです。**斜体文字**以外のレジスタについては、ACTIVE60 ステートでは、「2.2.デバイス／ホスト共通レジスタマップ」のレジスタは読み書きでき、「2.3.デバイス・レジスタマップ」および「2.4.ホスト・レジスタマップ」のレジスタは読むことができます。

USB 回路は、周波数 480MHz の SCLK480 を必要としているため、ホスト、デバイスともに、このステートでは動作しません。

IDE 並びに CPU 回路は、周波数 480MHz の SCLK480 が不必要です。周波数 480MHz の PLL の消費電流は、周波数 60MHz の消費電流に比較して大きいので、USB を使用しない場合には ACTIVE60 ステートに遷移させることで、低消費電力化が図れます。

### 1.5.1.4.ACT\_DEVICE(アクトデバイス)

オシレータ、PLL60、DevicePLL480 が動作しているステートです。レジスタマップの**斜体太字**のレジスタ及びビットはスヌーズ及びスリープ時でも読み書き可能なレジスタです。**斜体文字**以外のレジスタについては、ACT\_DEVICEステートでは、「2.2.デバイス／ホスト共通レジスタマップ」および「2.3.デバイス・レジスタマップ」のレジスタは読み書きでき、「2.4.ホスト・レジスタマップ」のレジスタは読むことができます。ACT\_DEVICEステートに遷移が終了した時点で、PM\_Control\_1.MonActFuncビットはクリアされ、USBデバイスとしてデータ転送が可能となります。

USB デバイス回路は、周波数 480MHz の SCLK480 を必要としているため、このステートで動作しません。

IDE 並びに CPU 回路は、周波数 480MHz の SCLK480 が不必要です。周波数 480MHz の PLL の消費電流は、周波数 60MHz の消費電流に比較して大きいので、USB を使用しない場合には ACTIVE60 ステートに遷移することで、低消費電力化が図れます。

### 1.5.1.5.ACT\_HOST(アクトホスト)

オシレータ、PLL60、HostPLL480 が動作しているステートです。レジスタマップの**斜体太字**のレジスタ及びビットはスヌーズ及びスリープ時でも読み書き可能なレジスタです。**斜体文字**以外のレジスタについては、ACT\_HOSTステートでは、「2.2.デバイス／ホスト共通レジスタマップ」および「2.4.ホスト・レジスタマップ」のレジスタは読み書きでき、「2.3.デバイス・レジスタマップ」のレジスタは読むことができます。ACT\_HOSTステートに遷移が終了した時点で、PM\_Control\_1.MonActFuncビットはセ

ットされ、USBホストとしてデータ転送が可能となります。

USB ホスト回路は、周波数 480MHz の SCLK480 を必要としているため、このステートで動作します。

IDE 並びに CPU 回路は、周波数 480MHz の SCLK480 が不必要です。周波数 480MHz の PLL の消費電流は、周波数 60MHz の消費電流に比較して大きいので、USB を使用しない場合には ACTIVE60 ステートに遷移することで、低消費電力化が図れます。

#### 1.5.1.6. ACT\_ALL(アクトオール)

オシレータ、PLL60、DevicePLL480、HostPLL480 と全てのクロック源が動作しているステートです。レジスタマップの**斜体太字**のレジスタ及びビットはスヌーズ及びスリープ時でも読み書き可能なレジスタです。**斜体文字**以外のレジスタについては、ACT\_ALLステートでは、2 ポートモードの場合 (ClkSelect.Port1x2 ビットがクリアされている場合)には、全てのレジスタを読み書きすることができます。すなわち、「2.2.デバイス/ホスト共通レジスタマップ」、「2.4.ホスト・レジスタマップ」及び「2.3.デバイス・レジスタマップ」のレジスタを読み書きすることができます。ただし、ホスト・レジスタマップ、デバイス・レジスタマップは、同じアドレス空間に存在していますので、レジスタを読み書きする場合には、HostDeviceSel.HOSTxDEVICEビットを適切に設定してください。

USB デバイス及びホスト回路は、周波数 480MHz の SCLK480 を必要としており、このステートで動作します。この ACT\_ALL ステートへの遷移開始ビットとして、PM\_Control\_0.GoActAllDev ビットをセットした場合には、ACT\_ALL ステートに遷移が終了した時点で、PM\_Control\_1.MonActFunc ビットがクリアされ、USB デバイスとしてデータ転送が可能となります。ここでは、このステートを、ACT\_ALL ステートのサブステートとして、ACT\_ALL\_DEV ステートと定義しています。この時、USB ホストはデータ転送は行えませんが、周辺機器との接続を切断することなく、SOF を送出し続けたり、SUSPEND させたりすることは可能です。

逆に ACT\_ALL ステートへの遷移開始ビットとして、PM\_Control\_0.GoActAllHost ビットをセットした場合には、ACT\_ALL ステートに遷移が終了した時点で、PM\_Control\_1.MonActFunc ビットがセットされ、USB ホストとしてデータ転送が可能となります。ここでは、このステートを、ACT\_ALL ステートのサブステートとして、ACT\_ALL\_HOST ステートと定義しています。この時、USB デバイスはデータ転送は行えませんが、PC との接続を切断することなく、接続を保持し続けるために、PC からのトークンに対しては、NAK 応答を行います。また、エンドポイント EP0 に対する SETUP トランザクションには、正常に ACK 応答します。

この ACT\_ALL ステートを使用することで、デバイスとしての動作を完全に止めることなく、ホスト動作を実現することが可能です。例えば、PC との接続を切断することなく、すなわちマストレージデバイスの場合には、ディスクマウントを維持したまま、ホストとして USB メモリ等への読み書きが行えます。

#### 1.5.2. 1 ポートモード時 (ClkSelect.Port1x2 が 1 にセットされている場合)

USB ポート B のみを使用して、USB のデバイス、ホストのどちらか一方の機能を実現できます。機能の切り替えには、HostDeviceSel.HOSTxDEVICE ビットを使用し、HostDeviceSel.HOSTxDEVICE ビットが 0 にクリアされている場合には、レジスタマップ選択とともに、デバイス機能が選択され、デバイス機能ブロックへのクロックゲートがオープンし、ホスト機能ブロックへのクロックゲートがクローズします。逆に HostDeviceSel.HOSTxDEVICE ビットが 1 にセットされている場合には、レジスタマップ選択とともに、ホスト機能が選択され、ホスト機能ブロックへのクロックゲートがオープンし、デバイス機能ブロックへのクロックゲートがクローズします。実際の機能ブロックへのクロック供給は、後述の SLEEP ステート、SNOOZE ステート、ACTIVE60 ステート、ACT\_DEVICE ステートにより開始、停止されますので、この HostDeviceSel.HOSTxDEVICE ビットのセット、クリアは、各モジュールにクロックが供給されていない SLEEP ステート及び SNOOZE ステートにて行ってください。

2 ポートモード時に使用できた ACT\_HOST ステート及び ACT\_ALL ステートは、1 ポートモードでは使用できませんので、注意してください。また、1 ポートモードにて使用していない USB ポート A の HTM(Host Transceiver Macro)は、動作しません。接続例にしたがって端子処理してください。

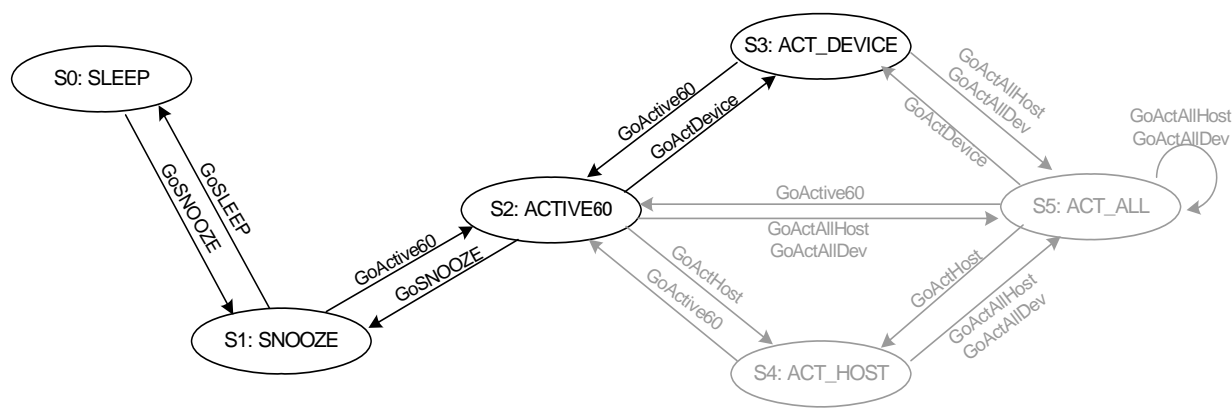
パワーマネジメント機能は、オシレータ、PLL(DevicePLL480、PLL60)の動作を制御し、SLEEP ステート、SNOOZE ステート、ACTIVE60 ステート、ACT\_DEVICE ステートの 4 ステートを遷移します。

## 1. 機能説明

他のステートに遷移する場合には、PM\_Control\_0.GoSLEEP、PM\_Control\_0.GoSNOOZE、PM\_Control\_0.GoActive60、PM\_Control\_0.GoActDevice ビットをセットすることで開始され、任意の処理が行われた後終了します。現在、どのステートにいるかを確認するには、PM\_Control\_1.PM\_State[3:0]を確認してください。また遷移終了時、MainIntStat.FinishedPM イベントが発生します。この時、MainIntEnb.EnFinishedPM ビットをセットすれば、割り込み XINT が発生します。

全てのステートから他のステートへ遷移することが可能で、ACT\_DEVICE ステート時に PM\_Control\_0.GoSLEEP ビットをセットした場合は、ACTIVE60 ステート、SNOOZE ステートを経由し、SLEEP ステートに遷移し、完全に遷移終了した時点で MainIntStat.FinishedPM イベントが発生します。また SLEEP ステート時に PM\_Control\_0.GoActDevice ビットをセットした場合も、SNOOZE ステート、ACTIVE60 ステートを経由し、ACT\_DEVICE ステートに遷移し、完全に遷移終了した時点で MainIntStat.FinishedPM イベントが発生します。同様に、ACTIVE60 ステート時に PM\_Control\_0.GoSLEEP ビットをセットした場合も、SNOOZE ステートを経由し、SLEEP ステートに遷移し、完全に遷移終了した時点で MainIntStat.FinishedPM イベントが発生します。また SLEEP ステート時に PM\_Control\_0.GoActive60 ビットをセットした場合も、SNOOZE ステートを経由し、ACTIVE60 ステートに遷移し、完全に遷移終了した時点で MainIntStat.FinishedPM イベントが発生します。

なお、現在のステートから、同じステートへの遷移を行うことは出来ないことに注意してください。このとき、セットした PM\_Control\_0.GoXXXX ビットはクリアされません。例えば、ACTIVE60 ステートで動作している状態で、PM\_Control\_0.GoActive60 をセットした場合には、遷移を行わず(見かけ上正しいが)、PM\_Control\_0.GoActive60 ビットがセットされたままとなります。



※ 最終目標ステートのGoXXXXを行うことで、全てのステートから他のステートへの遷移が可能

※ ACT\_ALLステートは、ACT\_ALL\_DEVステート、ACT\_ALL\_HOSTステートを含んでいます。

図 1-78 1 ポートモード時のパワーマネージメント

### 1.5.2.1.SLEEP(スリープ)

オシレータが発振していないステートです。したがって、このステートでは、PLL も発振していません。

SNOOZE ステート、ACTIVE60 ステート、ACT\_DEVICE ステート中に PM\_Control\_0.GoSLEEP ビットをセットし、スリープに遷移する場合は、DevicePLL480 を停止し、続いて PLL60 を停止し、最後に OSCCLK の出力を停めた後に発振を停止します。

逆に、SLEEP ステート中に PM\_Control\_0.GoSNOOZE、PM\_Control\_0.GoActive60 ビット、PM\_Control\_0.GoActDevice ビットをセットし、スリープからスヌーズへ遷移する時には、オシレータが安定して発振するまで内部回路には OSCCLK を与えないように、発振安定時間ゲートしています。この発振安定時間は、発振セル、発振子、周辺回路及び基板によって変化しますので、WakeUpTim\_H,L レジスタを用いて設定してください。この WakeUpTim\_H,L レジスタは、非同期でアクセス可能ですので、SLEEP ステートでも読み出し、書き込みが可能です。

### 1.5.2.2.SNOOZE(スヌーズ)

オシレータは発振している状態で、PLL が発振していないステートです。

ACTIVE60 ステート、ACT\_DEVICE ステート中に PM\_Control\_0.GoSNOOZE ビットをセットし、スヌーズに遷移する場合は、出力しているクロックを停止した後、DevicePLL480 を停止し、最後に PLL60 を停止します。

逆に、SNOOZE ステート中に PM\_Control\_0.GoActive60 ビットまたは PM\_Control\_0.GoActDevice ビットをセットし、スヌーズからアクティブへ遷移する時には、PLL が安定して発振するまで内部回路には SCLK を与えないように、PLL 安定時間(約 250us)ゲートします。

### 1.5.2.3.ACTIVE60(アクティブ 60)

オシレータ、PLL60 が動作し、DevicePLL480 が停止しているステートです。レジスタマップの**斜体太字**のレジスタ及びビットはスヌーズ及びスリープ時でも読み書き可能なレジスタです。**斜体文字**以外のレジスタについては、ACTIVE60 ステートでは、「2.2.デバイス／ホスト共通レジスタマップ」のレジスタは読み書きでき、「2.3.デバイス・レジスタマップ」および「2.4.ホスト・レジスタマップ」のレジスタは読むことができます。

USB 回路は、周波数 480MHz の SCLK480 を必要としているため、ホスト、デバイスともに、このステートでは動作しません。

IDE 並びに CPU 回路は、周波数 480MHz の SCLK480 が不必要です。周波数 480MHz の PLL の消費電流は、周波数 60MHz の消費電流に比較して大きいので、USB を使用しない場合には ACTIVE60 ステートに遷移させることで、低消費電力化が図れます。

### 1.5.2.4.ACT\_DEVICE(アクトデバイス)

オシレータ、PLL60、DevicePLL480(USBポートBの 480MHzのPLL)が動作しているステートです。レジスタマップの**斜体太字**のレジスタ及びビットはスヌーズ及びスリープ時でも読み書き可能なレジスタです。**斜体文字**以外のレジスタについては、ACT\_DEVICEステートでは、「2.2.デバイス／ホスト共通レジスタマップ」は読み書きできます。「2.3.デバイス・レジスタマップ」及び「2.4.ホスト・レジスタマップ」、HostDeviceSelect.HOSTxDEVICEビットの設定によって、読み書き可能かどうか切り替わります。HostDeviceSelect.HOSTxDEVICEビットが 0 にクリアされている場合には、「2.3.デバイス・レジスタマップ」が読み書きできます。逆にHostDeviceSelect.HOSTxDEVICEビットが 1 にセットされている場合には、「2.4.ホスト・レジスタマップ」は読み書きできます。

USB デバイス及びホスト回路は、周波数 480MHz の SCLK480 を必要としているため、このステートで動作します。

IDE 並びに CPU 回路は、周波数 480MHz の SCLK480 が不必要です。周波数 480MHz の PLL の消費電流は、周波数 60MHz の消費電流に比較して大きいので、USB を使用しない場合には ACTIVE60 ステートに遷移することで、低消費電力化が図れます。

### 1.5.2.5.ACT\_HOST(アクトホスト)

この ACT\_HOST ステート、及びこのステートに遷移させるための PM\_Control\_0.GoActHost ビットは使用できません。

### 1.5.2.6.ACT\_ALL(アクトオール)

この ACT\_ALL ステート、及びこのステートに遷移させるための PM\_Control\_0.GoActALL ビットは使用できません。

## 1. 機能説明

### 1.6. FIFO 管理

FIFO 管理について説明します。

#### 1.6.1. FIFO メモリマップ

FIFO のメモリマップを以下に示します。

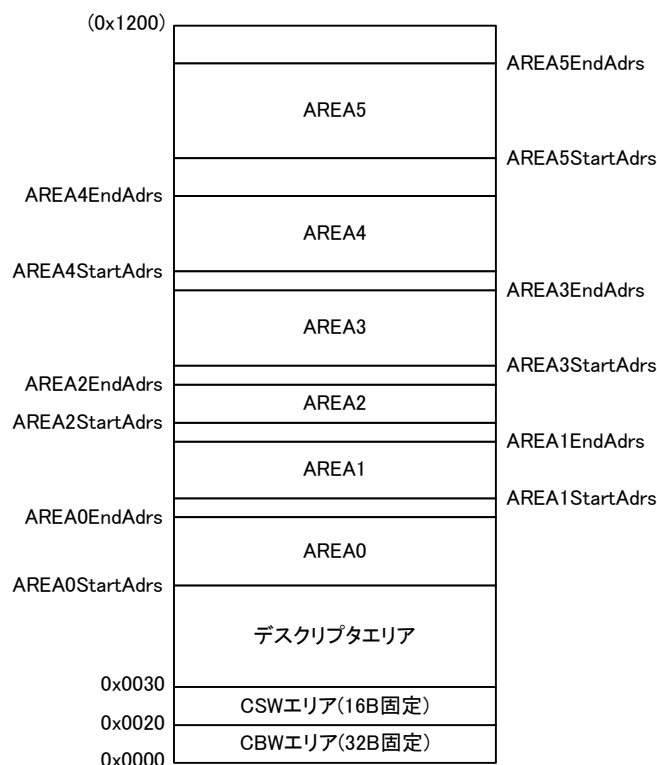


図 1-79 FIFO メモリマップ例

FIFOメモリは、CBWエリア、CSWエリア、デスクリプタエリア、AREA0、AREA1、AREA2、AREA3、AREA4、AREA5に分割して使用できます。CBWエリア、CSWエリアについては図 1-79 のように固定領域が割り当てられます。一方その他の $AREA_x\{x=0-5\}$ についてはFIFOエリア設定レジスタ( $AREA_x\{x=0-5\}StartAdrs\_H,L$ ,  $AREA_x\{x=0-5\}EndAdrs\_H,L$ )で領域を柔軟に設定することが可能です。デスクリプタエリアは未使用の領域を自由に使用できます。

デスクリプタエリアは、USBデバイス時にデスクリプタ返信機能で使用する領域です。FIFOの未使用領域のどの位置からでも使用することが出来ます。実際の使用方法については、「1.6.2.デスクリプタエリア」に後述します。全てのFIFO領域をデスクリプタ返信機能に使用設定することが可能となっていますが、競合を避けるために、ここで表したエリアでの使用を推奨します。

CBWエリアは、USBデバイス時にバルクオンリーサポート機能のCBWサポートで使用する領域です。32 バイト確保されていますが、そのうち、アドレス 0x0000 から始まる 31 バイトの領域を使用します。実際の使用方法については、「1.6.3.1.CBWエリア(USBデバイス時)」に後述します。また、このCBWエリアはUSBホスト時にはCHaのバルクオンリーサポート機能で使用する領域です。実際の使用方法については「1.6.3.2.CBWエリア(USBホスト時)」に後述します。

CSWエリアは、USBデバイス時にバルクオンリーサポート機能のCSWサポートで使用する領域です。16 バイト確保されていますが、そのうち、アドレス 0x0020 から始まる 13 バイトの領域を使用します。実際の使用方法については、「1.6.4.1.CSWエリア(USBデバイス時)」に後述します。また、このCSWエリアはUSBホスト時にはCHaのバルクオンリーサポート機能で使用する領域です。実際の使用方法については「1.6.4.2. CSWエリア(USBホスト時)」に後述します。

AREA0、AREA1、AREA2、AREA3、AREA4、AREA5 は、USB デバイス時はエンドポイント( $EP_x\{x=0, a-e\}$ )とジョインすることにより使用可能な、エンドポイントのエリアです。USB ホスト時には、同様

にチャンネル(CHx{x=0, a-e})に Join することで使用可能となります。ジョインするには、AREA ジョイン設定レジスタ(AREAx{x=0-5}Join\_1)の JoinEPxCHx{x=0,a-e}ビットを設定します。なお、同じエンドポイント、チャンネルを複数のエリアに同時にジョインしないで下さい。

AREA0, AREA1, AREA2, AREA3, AREA4, AREA5 それぞれのエリアは、FIFO として制御されており、データ格納数が保持されています。この保持された状態をクリアするためには、AREAnFIFO\_Clr.ClearAREAx{x=0-5}の各ビットをセットして下さい。

なお、これらの状態クリアは、データ保持情報の初期化動作のみを行い、データは書き込みまたはクリアをしません。従って、このビットにより RAM 上のデータがクリアされることはありませんので、デスクリプタエリア内に記録された情報は消えることは無く、クリア後に再度データを書き込み直す必要はありません。

### 1.6.2. デスクリプタエリア

デスクリプタエリアは、USB デバイス時にデスクリプタ返信機能に使用するエリアです。デスクリプタ返信機能は、エンドポイント 0 において、データステージが IN 転送で行われる場合に使用することが出来ます。

IN 方向のデータステージを行う場合に、この領域内に書き込まれたデータの先頭アドレスと、返信するデータサイズを設定し、デスクリプタ返信機能を実行すると、自動的にデータステージが行われます。

デバイスデスクリプタ等、機器で一意に決定されるような内容を、電源投入後の初期設定時等にこのエリアに書き込んでおくことにより、リクエストを受け付けるとこのエリアのデータを返信するよう指示することが出来ます。リクエスト毎に EP0 の FIFO エリアにデータを書き込む必要が無く、高速に応答することが出来ます。

#### 1.6.2.1. デスクリプタエリアへのデータの書き込み

デスクリプタエリアへデータを書き込むには、RAM\_WrDoor 機能を使用します。書き込み開始アドレスを RAM\_WrAdrs\_H,L レジスタに設定し、RAM\_WrDoor\_0,1 レジスタにデータを書き込むことによって行います。RAM\_WrAdrs\_H,L レジスタの値は、一回の書き込み毎に、書き込みデータ数ずつ更新されますので、連続したアドレスにデータを書き込む場合には、RAM\_WrDoor\_0,1 レジスタに連続して書き込むことが出来ます。

なお、RAM\_WrDoor\_0,1 レジスタは書き込みのみ可能となっています。

#### 1.6.2.2. デスクリプタエリアでのデータステージ(IN)の実行

書き込まれたデータを、デスクリプタ返信機能にて使用する場合には、D\_DescAdrs\_H,L レジスタに、データステージに送信するデータの先頭アドレスを設定し、返信するデータサイズを D\_DescSize\_H,L レジスタに設定した後、D\_EP0Control.ReplyDescriptor ビットに 1 をセットします。また、D\_EP0Control.INxOUT ビットに"1"をセットして、IN トランザクションを行えるようにします。また、データステージの IN トランザクションにデータパケットを返信出来るよう、D\_SETUP\_Control.ProtectEP0 をクリアしてから、D\_EP0Control.IN.ForceNAK ビットをクリアするようご注意ください。

設定後、ホストからの IN トランザクションに応答し、マックスパケットサイズ(D\_EP0MaxSize にて設定)に自動的に分割しながら、ホストに D\_DescSize\_H,L レジスタに設定されたデータ数までデータパケットを返信します。また、D\_DescSize\_H,L レジスタの値がマックスパケットサイズに満たない場合や、分割した後の残りのデータ数がマックスパケットサイズに満たない場合には、自動的にショートパケットとして送信します。

ホストから OUT トランザクションが発行されると、D\_EP0Control.ReplyDescriptor をクリアし、D\_EP0IntStat.DescriptorCmp がセットされます。F/W はステータスステージの処理に移行して下さい。

### 1.6.3. CBW エリア

#### 1.6.3.1. CBW エリア(USB デバイス時)

CBW エリアは、USB デバイス時にバルクオンリーサポート機能の CBW サポートに使用するエリアです。Bulk OUT のエンドポイント(エンドポイント EPa, EPb, EPc, EPd, EPe)で、Bulk Only Transport

## 1. 機能説明

---

Protocol のコマンドトランスポートを行うときに、このエリアに受信することが出来ます。こうすることにより、エンドポイントの FIFO では、データトランスポートで受信したデータのみを扱うことができます。

CBW サポートを実行している時に、対象となるエンドポイントで OUT トランザクションが行われ、データサイズが 31 バイトであると、CBW エリアに受信します。データ長が 31 バイトで無かった場合は、エラーステータスを発行し、データは破棄します。

CBW エリアに受信したデータを読み出すには、RAM\_Rd 機能を使用します。RAM\_RdControl.RAM\_GoRdCBW\_CSW ビットをセットすると、CBW エリアのデータが読み出されて RAM\_Rd\_00~RAM\_Rd\_1E レジスタにコピーされ、完了ステータス(CPU\_IntStat.RAM\_RdCmp ビット)が発行されます。

### 1.6.3.2.CBW エリア(USB ホスト時)

CBW エリアは、USB ホスト時にバルクオンリーサポート機能で使用するエリアです。チャンネル CHa で、Bulk Only Transport Protocol のコマンドトランスポートを行う際に、これらのエリアから CBW データをデータパケットとして送信します。こうすることにより、チャンネルの FIFO では、データトランスポートで送信するデータのみを扱うことができます。

データパケットを送信する前に、CBW エリアには 0x0000 から CBW データ(31 バイト)を用意して下さい。

CBW エリアにデータを書き込むには、RAM\_WrDoor 機能を使用します。RAM\_WrAdrs\_H,L レジスタに、CBW エリアの先頭アドレス(0x0000)を書き、RAM\_WrDoor\_0,1 レジスタを介して 31 バイトの有効データを書き込みます。CBW エリアは 32 バイト確保されていますので、ワードアクセスで 32 バイト書き込んでも、他の領域を侵すことは無く、問題ありません。

### 1.6.4. CSW エリア

#### 1.6.4.1.CSW エリア(USB デバイス時)

CSW エリアは、USB デバイス時、バルクオンリーサポート機能の CSW サポートに使用するエリアです。Bulk IN のエンドポイント(エンドポイント EPa, EPb, EPc, EPd, EPe)で、Bulk Only Transport Protocol のステータストランスポートを行うときに、このエリアから送信することが出来ます。こうすることにより、エンドポイントの FIFO では、データトランスポートで送信するデータのみを扱うことができます。

CSW サポートを実行している時に、対象となるエンドポイントで IN トランザクションが行われると、CSW エリアから 13 バイトのデータをデータパケットとして送信します。

CSW エリアにデータを書き込むには、RAM\_WrDoor 機能を使用します。RAM\_WrAdrs\_H,L レジスタに、CSW エリアの先頭アドレス(0x0020)を書き、RAM\_WrDoor\_0,1 レジスタを介して 13 バイトの有効データを書き込みます。CSW エリアは 16Byte 確保されていますので、ワードアクセスで 14 バイト書き込んでも、他の領域を侵すことは無く、問題ありません。

#### 1.6.4.2.CSW エリア(USB ホスト時)

CSW エリアは、USB ホスト時バルクオンリーサポート機能で使用するエリアです。チャンネル CHa で、Bulk Only Transport Protocol のステータストランスポートを行う際に、このエリアに CSW データを受信します。こうすることにより、チャンネルの FIFO では、データトランスポートで受信したデータのみを扱うことができます。

CSW エリアに受信したデータを読み出すには、RAM\_Rd 機能を使用します。RAM\_RdControl.RAM\_GoRdCBW\_CSW ビットをセットすると、CSW エリアのデータが読み出され、RAM\_Rd\_00~RAM\_Rd\_0C レジスタにコピーされ、完了ステータス(CPU\_IntStat.RAM\_RdCmp ビット)が発行されます。

### 1.6.5. FIFO へのアクセス方法

FIFO のデータにアクセスする方法は、RAM アクセスする方法と、FIFO アクセスする方法とがあります。FIFO アクセスの要因としては、CPU(レジスタ), CPU(DMA), USB, IDE が有ります。

RAM アクセスによって、FIFO の任意の場所からデータを読むことができ、また、任意の場所にデー



タを書くことができます。RAM アクセスによっては、FIFO のデータ数は更新されません。つまり、RAM\_Rd 機能によって FIFO のデータを読み出しても、FIFO 領域は解放されません。また、RAM\_WrDoor 機能によって FIFO にデータを書き込んでも、FIFO 領域を占有しません。

#### 1.6.5.1.RAM アクセスの方法(RAM\_Rd)

FIFO に RAM\_Rd レジスタによってリードアクセスする場合には、読み出しを行いたい FIFO 領域の先頭アドレスとデータサイズを RAM\_RdAdrs\_H,L レジスタ、RAM\_RdCount レジスタに設定し、RAM\_RdControl.RAM\_GoRd ビットを設定して下さい。指定された FIFO 領域のデータを RAM\_Rd レジスタから読み出し可能になると CPU\_IntStat.RAM\_RdCmp ビットが"1"にセットされます。RAM\_RdCmp ビットを確認した後 RAM\_Rd\_00 ~ RAM\_Rd\_1F レジスタからデータを読み出してください。読み出されるデータは RAM\_Rd\_00 から順に格納されます。RAM\_RdCount レジスタに設定したサイズが 32 バイトより小さい場合は、設定サイズ以降の RAM\_Rd レジスタの値は無効となります。

RAM\_Rd レジスタによる FIFO データの読み出しは、FIFO 領域の設定に関係なく、任意に行う事が出来ます。

RAM\_Rd 機能動作中は、RAM\_RdAdrs\_H,L レジスタ、RAM\_RdCount レジスタの値が逐次更新されます。RAM\_Rd 機能を開始した後は CPU\_IntStat.RAM\_RdCmp ビットがセットされるまで、これらのレジスタへアクセスしないで下さい。RAM\_Rd 機能動作中にこれらのレジスタを読み出した場合の値は保証されません。また、これらのレジスタに書き込みを行った場合、誤動作の原因となります。

#### 1.6.5.2.RAM アクセスの方法(RAM\_WrDoor)

FIFO に RAM\_WrDoor\_0,1 レジスタによってライトアクセスする場合には、書き込み開始アドレスを RAM\_WrAdrs\_H,L レジスタに設定し、RAM\_WrDoor\_0,1 レジスタにてデータを書き込むことによって行います。一回のライトアクセス毎に、自動的に RAM\_WrAdrs\_H,L レジスタは書き込み数ずつインクリメントされますので、連続したアドレスにデータを書き込む場合には、RAM\_WrDoor\_0,1 レジスタに連続して書き込むことが出来ます。

RAM\_WrDoor\_0,1 レジスタによる FIFO への書き込みは、FIFO 領域の設定に関係なく、任意に行う事が出来ます。

#### 1.6.5.3.FIFO アクセスの方法(レジスタアクセス)

FIFO にレジスタアクセスによってリードアクセスする場合には、いずれか 1 つのエリアに対して、AREAx{x=0-5}Join\_0.JoinCPU\_Rd に 1 をセットし、FIFO\_Rd\_0,1 レジスタ、または FIFO\_ByteRd レジスタにて読み出しを行います。

また、FIFO にレジスタアクセスによってライトアクセスする場合には、いずれか 1 つのエリアに対して、AREAx{x=0-5}Join\_0.JoinCPU\_Wr に 1 をセットし、FIFO\_Wr\_0,1 レジスタ、または FIFO\_ByteWr レジスタに書き込みを行います。

FIFO\_RdRemain\_H,L レジスタは、JoinCPU\_Rd にてただ 1 つ設定されたエリアにおいて、FIFO から読み出し可能なデータの残り数を表しています。また、FIFO\_WrRemain\_H,L レジスタは、JoinCPU\_Wr にてただ 1 つ設定されたエリアにおいて、FIFO に書き込み可能なエリアの残り数を表しています。

ICE 等を使用して F/W のデバッグを行うに際し、レジスタのダンプ等を行う場合に、JoinCPU\_Rd ビットのいずれかがセットされていると、レジスタのダンプ時に FIFO からデータを読み出されてしまうことに注意してください。

CPU のレジスタアクセスは、FIFO とメディア FIFO とで排他的に Join する必要があります。つまり、メディア FIFO と FIFO のいずれかの領域の両方に、JoinCPU\_Wr を同時に設定することは出来ません。同様に、メディア FIFO と FIFO のいずれかの領域の両方に、JoinCPU\_Rd を同時に設定することは出来ません。

#### 1.6.5.4.FIFO アクセスの方法(DMA)

FIFO に CPU の DMA アクセスによってリードアクセスする場合には、AREAx{x=0-5}Join\_0.JoinDMAx{x=0,1} ビットによりただ一つのエリアを選択し、DMAx{x=0,1}\_Control.Dir ビットに 1 を設定し、DMA 手順を実行して読み出しを行います。

また、FIFO に CPU の DMA アクセスによってライトアクセスする場合には、DMA の各チャンネルに

## 1. 機能説明

---

付き、AREAx{x=0-5}Join\_0.JoinDMAx{x=0,1}ビットによりただ一つのエリアを選択し、DMAx{x=0,1}\_Control.Dirビットに0を設定し、DMA手順を実行して書き込みを行います。

DMAx{x=0,1}\_Remain\_H,Lレジスタは、DMAの各チャンネルについて、AREAx{x=0-5}Join\_0.JoinDMAx{x=0,1}ビットによりただ一つ選択されたエリアにおいて、FIFOから読み出し可能なデータの残り数を表しています。また、DMAの各チャンネルについて、AREAx{x=0-5}Join\_0.JoinDMAx{x=0,1}ビットによりただ一つ選択されたエリアにおいて、FIFOに書き込み可能な残り数を表しています。

CPUのDMAアクセスは、FIFOとメディアFIFOとで排他的にJoinする必要があります。つまり、メディアFIFOとFIFOのいずれかの領域の両方に、JoinDMA0またはJoinDMA1を同時に設定することは出来ません。

### 1.6.5.5.FIFO アクセスの方法(IDE)

FIFOにIDEがアクセスする場合には、いずれか1つのエリアに対して、AREAx{x=0-5}Join\_0.Join.JoinIDEビットによりただ一つのエリアを選択し、IDE手順を実行してデータ転送を行います。IDE転送の方向はIDE\_Control.Dirビットによります。

IDEアクセスは、FIFOとメディアFIFOとで排他的にJoinする必要があります。つまり、メディアFIFOとFIFOのいずれかの領域の両方に、JoinIDEを同時に設定することは出来ません。

### 1.6.5.6.FIFO アクセスの制限

本LSIのFIFOには、USBとの送受信、CPUバスからのレジスタ書き込み、読み出し、DMAを用いた書き込み、読み出し、及び、IDEによる書き込み、読み出しが同時に行われます。また、CPUバスからの読み出しについては、先読み処理を行っています。

これらのことから、それぞれのエリアにおけるFIFOへのアクセス設定方法(Join)に対し、基本的に下記の排他ルールがあります。

- ・同一のFIFOエリアに対し、複数の書き込み要因を同時に設定してはいけません。
- ・同一のFIFOエリアに対し、複数の読み出し要因を同時に設定してはいけません。
- ・一つのエリアには、JoinCPU\_Wr, JoinCPU\_Rd, JoinDMAのうちの一つしか設定できない。
- ・JoinIDE, JoinCPU\_Wr, JoinCPU\_Rd, JoinDMAx{x=0,1}は、それぞれ、同時に一つのエリアにしか設定できない。
- ・JoinIDE, JoinCPU\_Wr, JoinCPU\_Rd, JoinDMAx{x=0,1}は、FIFOとメディアFIFOに同時には設定できない。

同一のエリアに対する書き込み／読み出し要因の重複に関し、USB(デバイス時)からのアクセス要因に関しては、例外的に下記の使用方法が可能です。例えば、OUTのエンドポイントのFIFOエリアに対し、JoinCPU\_WrやJoinDMAx{x=0,1}(DMAx{x=0,1}Control.Dir==0)を設定して書き込むことが出来ます。この場合、必ずOUTトランザクションが行われない状況にした後に、JoinCPU\_WrまたはJoinDMAx{x=0,1}を設定し、CPUからの書き込みを行う必要があります。また、INのエンドポイントのFIFOエリアに対し、JoinCPU\_RdやJoinDMAx{x=0,1}(DMAx{x=0,1}Control.Dir==1)を設定して読み出すことが出来ます。この場合、必ずINトランザクションが行われない状況にした後、JoinCPU\_RdまたはJoinDMAx{x=0,1}を設定し、CPUからの読み出しを行う必要があります。

トランザクションが行われない状況には、ActiveUSBビットがクリアされている場合、各エンドポイントがFIFO領域にジョインされていない場合や、ForceNAKがセットされている場合などがあります。

## 1.7. メディア FIFO 管理

メディアデータ転送機能における FIFO 管理について説明します。

### 1.7.1. メディア FIFO

メディア FIFO は IDE およびメモリ間でのメディアデータの転送用に使用します。

メディア FIFO は、USB FIFO とは独立しており、自動的に 64 バイトが使用エリアの固定領域として割り当てられます。そのため、F/W による領域設定は必要ありません。

使用エリアは、FIFO として制御されており、データ格納数が保持されています。この保持された状態をクリアするためには、MediaFIFO\_Control.MediaFIFO\_Clr をセットして下さい。

### 1.7.2. メディア FIFO へのアクセス方法

メディア FIFO へのアクセス要因には、CPU(レジスタ)、CPU(DMA)、IDE が有ります。

#### 1.7.2.1. メディア FIFO へのアクセス方法(レジスタアクセス)

メディア FIFO に CPU のレジスタアクセスによってリードアクセスする場合には、MediaFIFO\_Join.JoinCPU\_Rd に 1 をセットし、FIFO\_Rd\_0,1 レジスタ、または FIFO\_ByteRd レジスタにて読み出しを行います。

また、メディア FIFO に CPU のレジスタアクセスによってライトアクセスする場合には、MediaFIFO\_Join.JoinCPU\_Wr に 1 をセットし、FIFO\_Wr\_0,1 レジスタに書き込みを行います。

FIFO\_RdRemain\_H,L レジスタは、MediaFIFO\_Join.JoinCPU\_Rd にて設定された場合に、FIFO から読み出し可能なデータの残り数を表しています。また、FIFO\_WrRemain\_H,L レジスタは、MediaFIFO\_Join.JoinCPU\_Wr にて設定された場合に、FIFO に書き込み可能なエリアの残り数を表しています。

ICE 等を使用してファームウェアのデバッグを行うに際し、レジスタのダンプ等を行う場合に、MediaFIFO\_Join.JoinCPU\_Rd レジスタがセットされていると、レジスタのダンプ時に FIFO からデータを読み出されてしまうことに注意してください。

CPU のレジスタアクセスは、FIFO とメディア FIFO とで排他的に Join する必要があります。つまり、メディア FIFO と FIFO のいずれかの領域の両方に、JoinCPU\_Wr を同時に設定することは出来ません。同様に、メディア FIFO と FIFO のいずれかの領域の両方に、JoinCPU\_Rd を同時に設定することは出来ません。

#### 1.7.2.2. メディア FIFO へのアクセス方法(DMA)

メディア FIFO に CPU の DMA アクセスによってリードアクセスする場合には、DMA の各チャンネルにつき、MediaFIFO\_Join.JoinDMAx{x=0,1} ビットによりただ一つのチャンネルを選択し、DMAx{x=0,1}\_Control.Dir ビットに 1 を設定し、DMA 手順を実行して読み出しを行います。

また、メディア FIFO に CPU の DMA アクセスによってライトアクセスする場合には、DMA の各チャンネルにつき、MediaFIFO\_Join.JoinDMAx{x=0,1} ビットによりただ一つのチャンネルを選択し、DMAx{x=0,1}\_Control.Dir ビットに 0 を設定し、DMA 手順を実行して書き込みを行います。

DMAx{x=0,1}\_Remain\_H,L レジスタは、DMA の各チャンネルについて、MediaFIFO\_Join.JoinDMAx{x=0,1} ビットによりただ一つ選択されたチャンネルにおいて、FIFO から読み出し可能なデータの残り数を表しています。また、DMA の各チャンネルについて、MediaFIFO\_Join.JoinDMAx{x=0,1} ビットによりただ一つ選択されたチャンネルにおいて、FIFO に書き込み可能なエリアの残り数を表しています。

CPU の DMA アクセスは、FIFO とメディア FIFO とで排他的に Join する必要があります。つまり、メディア FIFO と FIFO のいずれかの領域の両方に、JoinDMA0 または JoinDMA1 を同時に設定することは出来ません。

#### 1.7.2.3. メディア FIFO へのアクセス方法(IDE)

メディア FIFO に IDE がアクセスする場合には、MediaFIFO\_Join.JoinIDE ビットに 1 をセットし、IDE 手順を実行してデータ転送を行います。IDE 転送の方向は IDE\_Control.Dir ビットによります。

IDE アクセスは、FIFO とメディア FIFO とで排他的に Join する必要があります。つまり、メディア

## 1. 機能説明

---

FIFO と FIFO のいずれかの領域の両方に、JoinIDE を同時に設定することは出来ません。

### 1.7.2.4. メディア FIFO へのアクセス制限

本 LSI のメディア FIFO には、CPU バスからのレジスタまたは DMA 書き込み、読み出し、IDE との送受信が同時に行われます。また、CPU バスからの読み出しについては、先読み処理を行っています。

これらのことから、メディア FIFO へのアクセス設定方法(Join)に対し、下記の排他ルールがあります。

- ・複数の書き込み要因を同時に設定してはいけない。
- ・複数の読み出し要因を同時に設定してはいけない。
- ・メディア FIFO には、JoinCPU\_Wr,JoinCPU\_Rd,JoinDMAx{x=0,1}のうちの一つしか設定できない。
- ・JoinIDE,JoinCPU\_Wr,JoinCPU\_Rd,JoinDMAx{x=0,1}は、FIFO とメディア FIFO に同時には設定できない。

## 1.8. CPUIF

### 1.8.1. CPUIF モード

S2R72C05 のCPUIFはBEモード／Strobeモードの 2 つのBUSモードをサポートします。各々のBUSモードで、ライトアクセスの方法が異なります。BEモードとは、ライトタイミングを示すライトストロブ信号、及び上位(CD[15:8])／下位(CD[7:0])バイトのイネーブル信号によってライトアクセスを行う方法です。一方、Strobeモードとは上位／下位バイトそれぞれに個別のライトストロブ信号を持ち、それらのライトストロブ信号によってライトアクセスを行う方法です(図 1-80 参照)。リードアクセスの方法は何れのBUSモードでも同一で、リードストロブ信号を用いて行います。

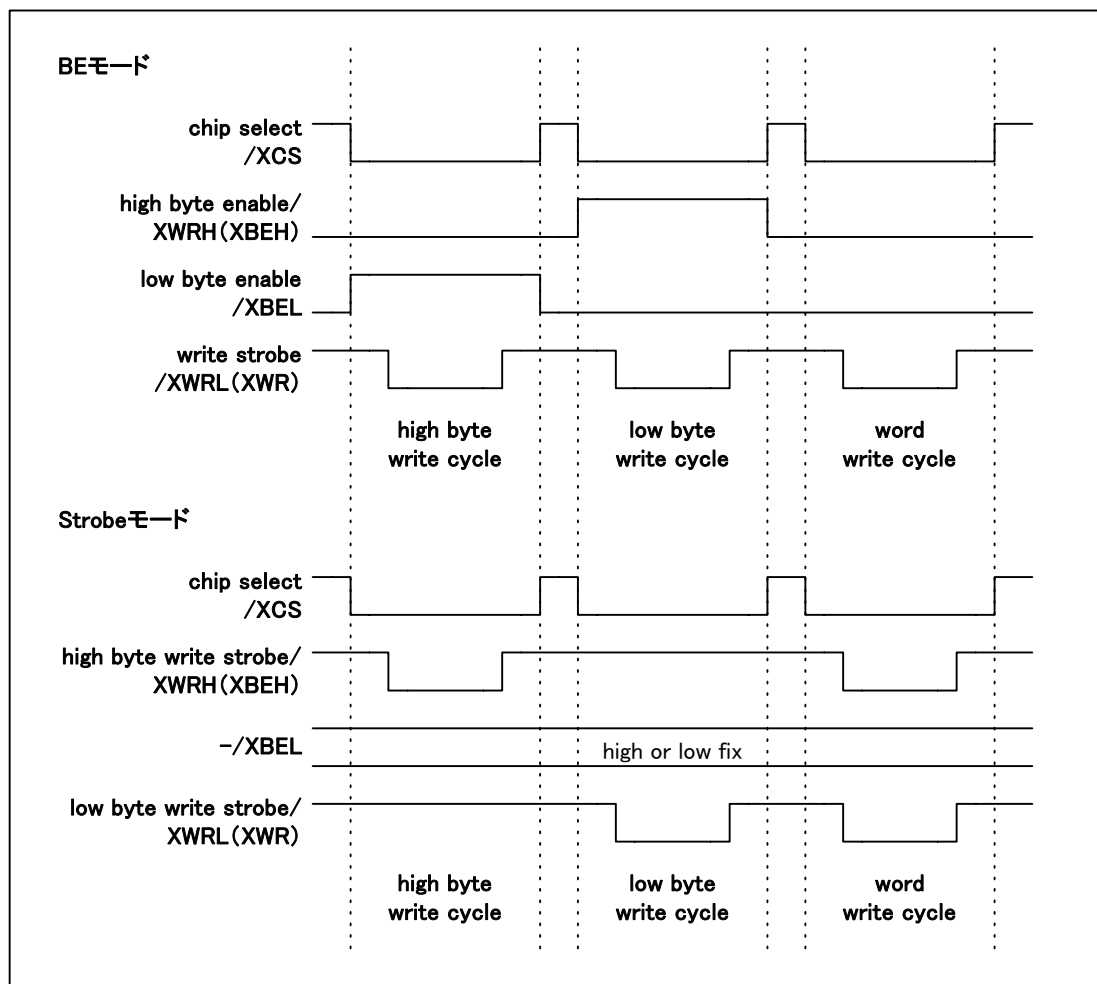


図 1-80 BUS モード

上述のBUSモードに加え、CPUのエンディアンに対応するため、S2R72C05 のCPUIFはBigEndianモード／LittleEndianモードの 2 つのENDIANモードをサポートします。BigEndianモードとは、偶数アドレスのレジスタが上位、奇数アドレスのレジスタが下位、また、1<sup>st</sup>バイトのデータが上位、2<sup>nd</sup>バイトのデータが下位になるモードです。LittleEndianモードとは、BigEndianモードの反対に、偶数アドレスのレジスタが下位、奇数アドレスのレジスタが上位、また、1<sup>st</sup>バイトのデータが下位、2<sup>nd</sup>バイトのデータが上位になるモードです。ENDIANモードの詳細は「Appendix B リトルエンディアンのCPUへの接続」をご参照ください。

接続する CPU に適合した、BUS モード、ENDIAN モードに設定した上で、S2R72C05 をご使用下さい。

モードの設定方法に関しては、次項で説明します。

## 1. 機能説明

### 1.8.2. CPUIF のモード設定

S2R72C05 は、ハードウェアリセット後 CPUIF のモード設定待ちの状態(未初期化期間)となり、この状態においては、モード設定以外のアクセスを受け付けません。モード設定終了後にレジスタアクセスが可能になります(初期化済み期間)。モードの設定方法は以下の通りです。

- 1) XCS 及び XWRL 信号をアサートし、特定のデータパターンを用いて、モード設定データを書き込みます(モード設定)。例えば、C05 をマッピングしたメモリ空間にワードでデータライトします。
- 2) XCS 信号を一旦ネゲートします。例えば、NOP を挿入する、或いは C05 をマッピングしたメモリ空間以外にアクセスします。
- 3) XCS 信号を再度アサートします(モード確定)。例えば、1)と同様の動作をします。
- 4) XCS 信号を一旦ネゲートします。例えば、2)と同様の動作をします。

以上でモード設定は完了し、通常のレジスタアクセスが可能になります。モード設定の確認のため設定完了後に ChipConfig レジスタを読み出す事を推奨致します。

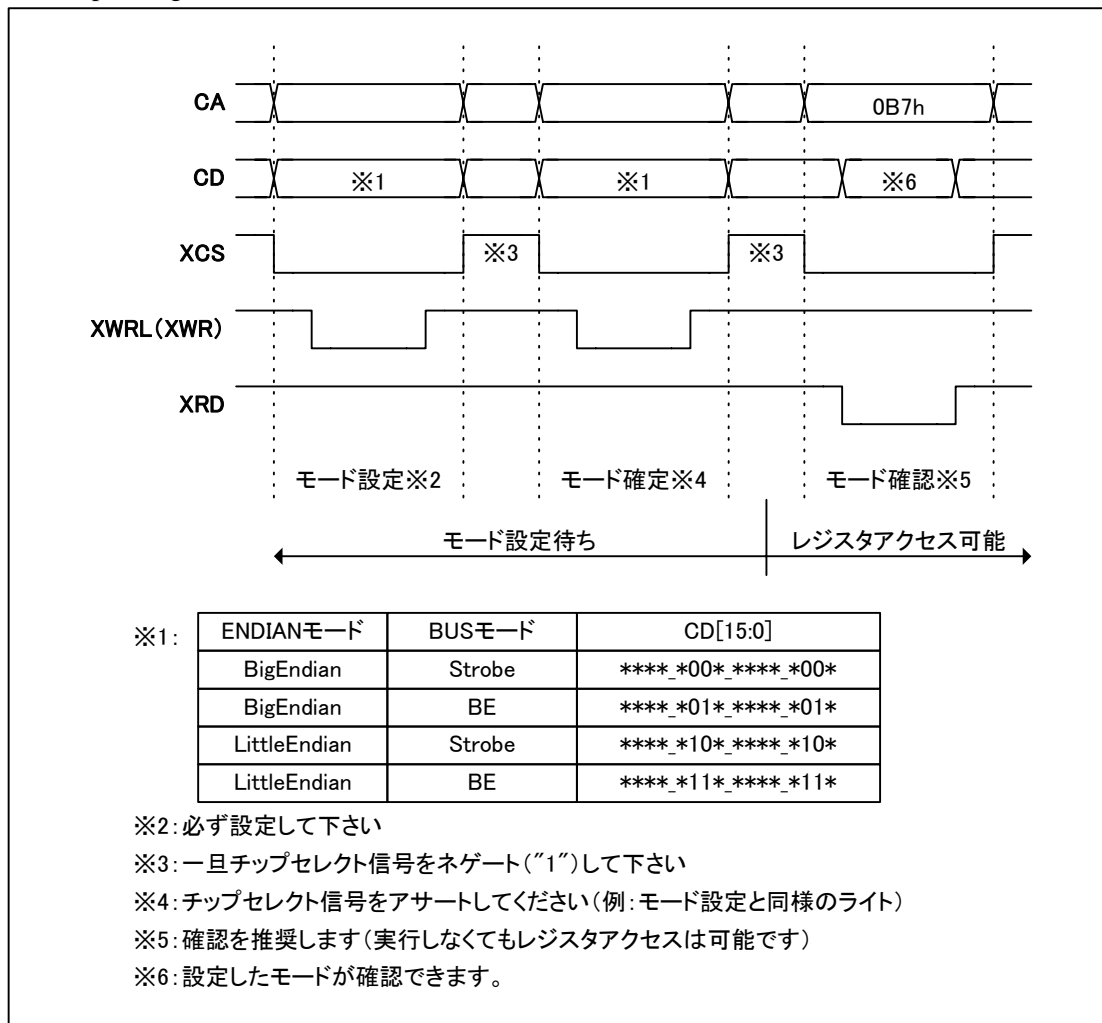


図 1-81 CPUIF のモード設定

モード設定の状態において、図 1-81 に記載以外の信号は、AC規格を満足する限り、High/Low何れの値でも問題ありません。

CPUIF のモード設定は、ハードリセット後に必ず行う必要があります、また、ハードリセット後の未初期化期間でのみ行えます。

以下の CPUIF の説明においては、Strobe モード/BigEndian モードを基本に説明致します。

### 1.8.3. ブロック構成

S2R72C05 CPUIF(以下、CPUIF)のブロック構成を図 1-82 に示します。  
REG/DMA0/DMA1 の3 ブロックから構成されます。

- ・ REG : S2R72C05 レジスタ領域へのアクセス制御
- ・ DMA0 : DMA チャンネル 0
- ・ DMA1 : DMA チャンネル 1

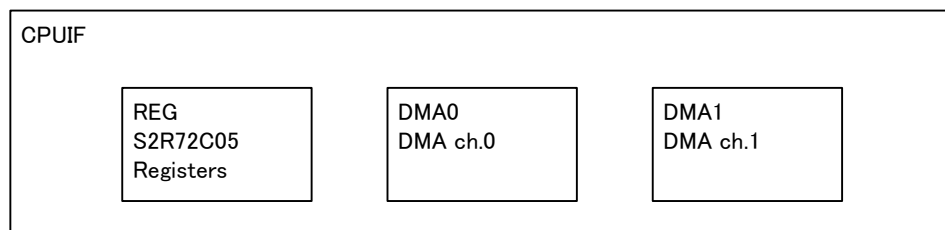


図 1-82 ブロック構成

#### 1.8.3.1.REG(S2R72C05 Registers)

S2R72C05 レジスタ領域のアクセスを制御します。以下のアクセス機能が含まれます。

- ・ 同期レジスタアクセス
- ・ FIFO アクセス
- ・ RAM\_Rd アクセス
- ・ 非同期レジスタアクセス

##### 1.8.3.1.1. 同期レジスタアクセス(ライト)

内部クロックに同期して外部バスのデータをレジスタへ書き込みます。

##### 1.8.3.1.2. 同期レジスタアクセス(リード)

リード(XCS、XRD 共にアサート)期間を出カイネーブル期間としてレジスタのデータを外部バスへ出力します。

レジスタのリード動作において、カウント値など3バイト以上で意味を成すレジスタでは、アクセスサイクル間のカウント値の桁上がり等により誤ったカウント値をリードしない様、最上位バイトのリードタイミングで下位バイトのレジスタの値を保持し、下位バイトのリード時には、この値を外部バスへ出力するレジスタもあります。

##### 1.8.3.1.3. FIFO アクセス(ライト)

FIFO ライトアクセスは FIFO\_Wr\_0,1、FIFO\_ByteWr、RAM\_WrDoor\_0,1 レジスタへの書き込みを意味します。

FIFO アクセス(ライト)には以下の制限事項があります。

- ・ AREAx{x=0-5}Join\_0.JoinCPU\_Wr ビット、MediaFIFO\_Join.JoinCPU\_Wr のいずれかを設定した後、FIFO\_WrRemain\_H,L レジスタで書き込み可能なデータ数を確認した上でアクセスを行って下さい。RAM\_WrDoor\_0,1 レジスタにはこの制限はございません。
- ・ 基本的にワード(2 バイト)単位でアクセスを行って下さい。端数(奇数)バイトの書き込みを行う場合はFIFOのバイト境界を意識してストロブ信号を制御して下さい。または、FIFO\_ByteWrレジスタをご使用下さい。詳細は「1.8.3.1.5.FIFO アクセスの端数処理」をご参照下さい。
- ・ FIFO\_Wr\_0,1、FIFO\_ByteWr レジスタへのライト直後に FIFO\_WrRemain\_H,L レジスタを確認しても正確な FIFO の空き領域を確認することが出来ません。必ず1CPU サイクル以上の間隔を空けて確認してください。
- ・ RAM\_WrDoor\_0,1 レジスタへのライト直後に RAM\_WrDoorAdrs\_H,L レジスタを

## 1. 機能説明

---

確認しても正確なアドレスを確認することが出来ません。必ず 1CPU サイクル以上の間隔を空けて確認してください。

### 1.8.3.1.4. FIFO アクセス(リード)

FIFO リードアクセスは FIFO\_Rd\_0,1 レジスタ、FIFO\_ByteRd レジスタの読み出しを意味します。

FIFO リードアクセスには、以下の制限事項があります。

- AREA<sub>x</sub>{x=0-5}.Join\_0.JoinCPU\_Rd ビット、MediaFIFO\_Join.JoinCPU\_Rd のいずれかを設定した後、FIFO\_RdRemain\_H,L レジスタで読み出し可能データ数及び RdRemainValid ビットを確認した上でアクセスを行って下さい。
- ワード読みを行う場合はFIFO\_Rd\_0,1 レジスタを用いて行って下さい。バイト読みを行う場合はFIFO\_ByteRdレジスタを用いて行って下さい。バイト境界が存在する場合はバイト読みを行って下さい。この場合にFIFO\_Rd\_0,1 レジスタを用いてワード読みを行った場合は片側にのみ有効なデータが出力されます。詳細は「1.8.3.1.5.FIFOアクセスの端数処理」ををご参照下さい。



## 1.8.3.1.5. FIFO アクセスの端数処理

端数(奇数)データを扱う場合の FIFO へのデータの格納状態と FIFO アクセスの関係を説明します。実際の FIFO は 4byte 幅ですが、この章の説明では簡易化のため 2byte 幅で表記致します。4byte/2byte による動作の相違はありません。

## 【ライト動作】

基本的にはバイト境界の存在しない状態から書き込み動作を行う事を推奨致します。

AREAnFIFO\_Clr.ClrAREAx{x=0-5} ビットやMediaFIFO\_Control.MediaFIFO\_Clrビット(メディアデータ転送時)をセットする等行って、バイト境界の存在しない状態からワード書き込みを行い、奇数データが存在する場合は、連続するデータの最終バイト(データZ)のみHigh側に書き込んで下さい。この状態を図 1-83 の(1)に示します。USB等からはデータA, B, C, D, ・ ・ ・ X, Y, Zの順に出力されます。

FIFOにバイト境界がある状態から書き込みを行う場合は、最初にLow側にデータを書き込み(データKの書き込み)、バイト境界を解消した後、ワード書き込み(データL, M)を行って下さい。この状態を図 1-83 の(2)に示します。

以上は、正常な書き込み動作です。

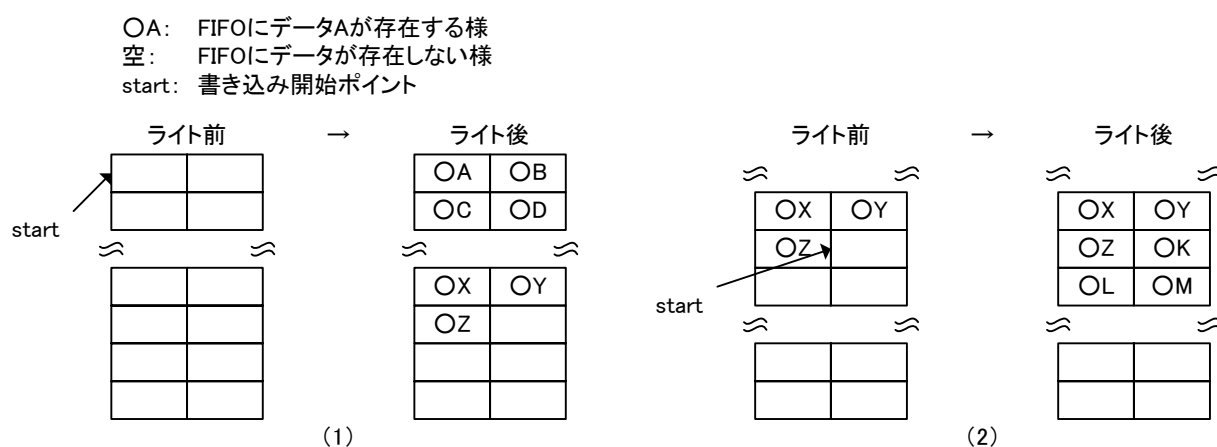


図 1-83 FIFO ライト処理(正常動作)

以下は注意が必要な書き込みを行った場合の動作です。

FIFOにバイト境界がある状態からワード書き込みを行った場合、High側の書き込みは無視されLow側のみ書き込みが行われます(図 1-84 の(3))。つまり、Low側にバイト書き込みを行ったときと同じ動作を行います。また、FIFOにバイト境界のある状態からHigh側にのみ書き込みを行った場合、その書き込みは無視されます(図 1-84 の(4))。

FIFOにバイト境界が無い状態からLow側にのみ書き込みを行った場合、その書き込みは無視されます(図 1-84 の(5))。また、FIFOにバイト境界が無く、且つ書き込み可能数が”1”の状態からワード書き込みを行った場合、Low側の書き込みは無視され、High側のみ書き込みが行われます(図 1-84 の(6))。つまり、High側にバイト書き込みを行ったときと同じ動作を行います。

## 1. 機能説明

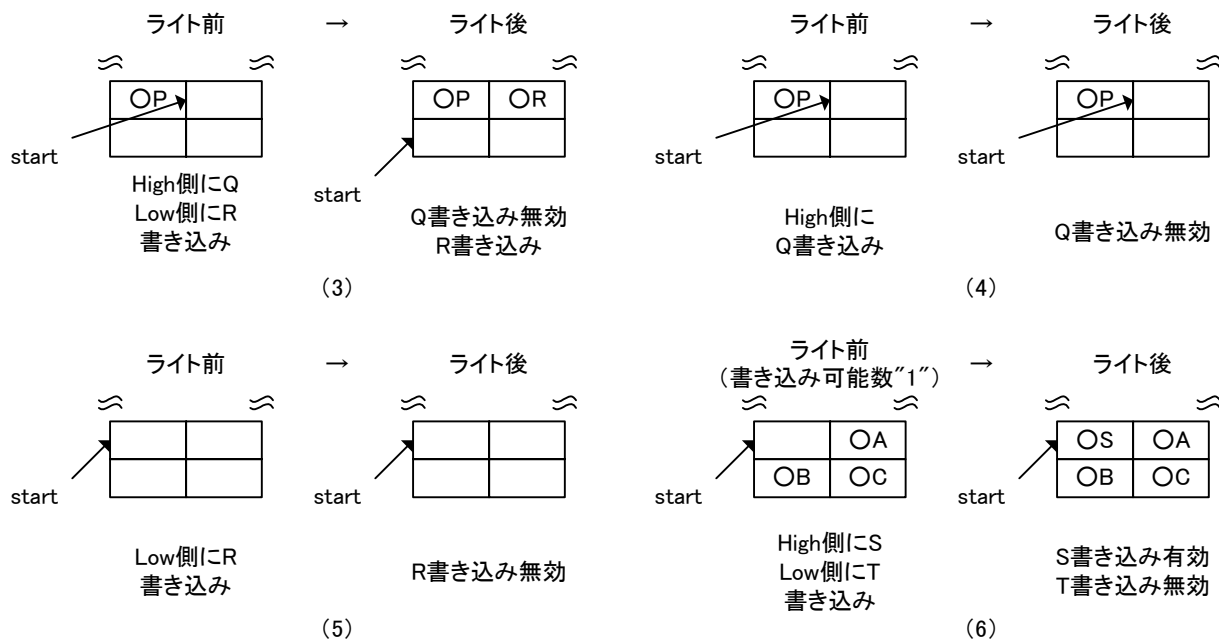


図 1-84 FIFO ライト処理(注意が必要な動作)

### 【リード動作】

バイト境界が無い場合は、FIFO\_Rd\_0,1 レジスタを用いたワード読み出し、FIFO\_ByteRd レジスタを用いたバイト読み出し何れを行っても問題ありません。バイト境界がある場合は、FIFO\_ByteRd レジスタを用いたバイト読み出しを行って下さい。一旦バイト境界を解消した後はワード読み出し、バイト読み出し何れを行っても問題ありません。

バイト境界が無い状態からのワード読み出しの様子を図 1-85 の(1)に示します。アクセス毎にデータ A, B→データ C, D と読み出されます。また、バイト読み出しの様子を図 1-85 の(2)に示します。アクセス毎にデータ A→データ B→データ C→データ D と読み出されます。以上は正常な読み出し動作です。

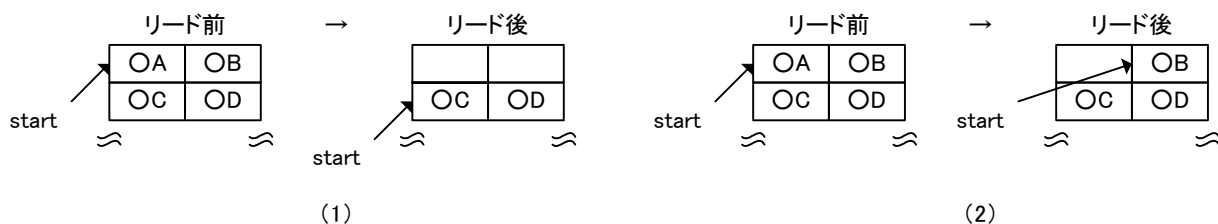


図 1-85 FIFO リード処理(正常動作)

以下は注意が必要な読み出しを行った場合の動作です。

図 1-86 の(3)はバイト境界がある状態から、FIFO\_Rd\_0,1 レジスタを用いてワード読み出しを行った場合の動作です。High側には不定のデータが出力され、Low側にデータJが出力されます。リードのポインタは1バイト分のみ進みます。図 1-86 の(4)はバイト境界が無いが残りデータが1バイトの状態から、FIFO\_Rd\_0,1 レジスタを用いてワード読み出しを行った場合の動作です。High側にはデータXが出力され、Low側には不定データが出力されます。リードのポインタは1バイト分のみ進みます。

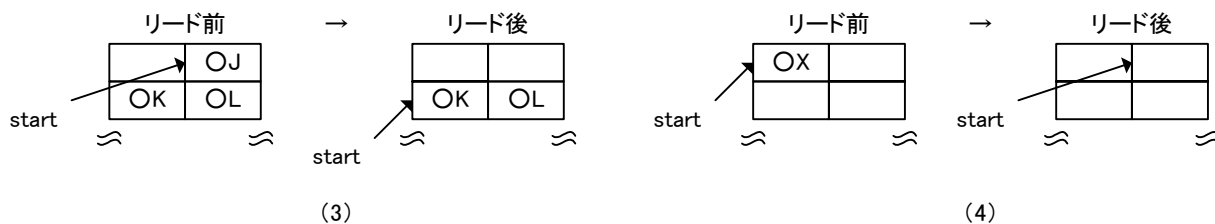


図 1-86 FIFO リード処理(注意が必要な動作)

上記より、端数処理のリード動作を行う場合の例を説明します。

- 1) USB から 64 バイト送られてきたデータを 31 バイト→33 バイトで読み出したい場合。
  - ① CPUIF は 64 バイトのレディをラッチして一連のリードシーケンスを開始する。
  - ② 30 バイト分のデータを FIFO\_Rd\_0,1 レジスタでワード読み、もしくは FIFO\_ByteRd レジスタでバイト読みする。
  - ③ 31 バイト目のデータを FIFO\_ByteRd レジスタでバイト読みする。→バイト境界発生。
  - ④ 32 バイト目のデータをバイト読みする。この場合 FIFO\_ByteRd レジスタでのバイト読みを推奨します。FIFO\_Rd\_0,1 レジスタでのワード読みを行った場合は Low 側にデータが出力されます。→バイト境界解消。
  - ⑤ 残り 32 バイトのデータを FIFO\_Rd\_0,1 レジスタでワード読み、もしくは FIFO\_ByteRd でバイト読みする。
- 2) JoinCPU\_Rd がセットされている状態で USB から 31 バイト、33 バイトと送られてきたデータ 64 バイトを全て FIFO\_Rd\_0,1 レジスタでワード読みしたい場合。
  - ① USB から 31 バイトのデータを受信した時点で CPUIF は 31 バイトのレディをラッチして一連の動作シーケンスを開始する。
  - ② 30 バイト分のデータをワード読みする。
  - ③ キャッシュされている 31 バイト目のデータ(バイト境界)を解消するため、一旦ジョインを切り離す。
  - ④ 33 バイトのデータが USB から送られてきた後、再度ジョインする。(1+33 バイト)
  - ⑤ CPUIF は 34 バイトのレディをラッチして一連の動作シーケンスを開始する。
  - ⑥ 34 バイト分のデータをワード読みする。

#### 1.8.3.1.6. RAM\_Rd アクセス

同期レジスタリードと同様、リード(XCS、XRD共にアサート)期間を出力イネーブル期間としてデータを外部バスへ出力します。詳細は「1.6.5.1.RAMアクセスの方法(RAM\_Rd)」をご参照下さい。

#### 1.8.3.1.7. 非同期レジスタアクセス(ライト)

外部ライト信号(XCS、XWRL,H)からライトパルスを作成して外部バスのデータをレジスタへ書き込みます。

#### 1.8.3.1.8. 非同期レジスタアクセス(リード)

同期レジスタリードと同様、リード(XCS、XRD共にアサート)期間を出力イネーブル期間としてレジスタのデータを外部バスへ出力します。

### 1.8.3.2.DMA0/DMA1(DMA ch.0 / ch.1)

#### 1.8.3.2.1. 基本機能

DMA の基本動作は以下の通りです。

##### 【ライト動作】

FIFO に書き込み可能な空き領域がある場合に XDREQ をアサートし DMA 転送が可能になります。

##### 【リード動作】

FIFO に読み出し可能なデータがあり、読み出し動作可能になると XDREQ をアサ

## 1. 機能説明

---

とし DMA 転送が可能になります。

DMA のアクセス方法は以下の 3 通りあります。

- XDACK でアクセスする

ChipConfig.CS\_Mode="0", DMAx{x=0,1}\_Config.DMA\_Mode="0" のとき、このアクセス方法となります。ChipConfig.DACK\_Level に設定された論理レベルに XDACK 信号がアサートされると、DMA アクセスと見なして転送を行います。

- XDACK&XCS でアクセスする

ChipConfig.CS\_Mode="1", DMAx{x=0,1}\_Config.DMA\_Mode="0" のとき、このアクセス方法となります。ChipConfig.DACK\_Level に設定された論理レベルに XDACK 信号がアサートされ、かつ XCS がアサート("0")されると、DMA アクセスと見なして転送を行います。

- DMA\_Mode でアクセスする

ChipConfig.CS\_Mode="0", DMAx{x=0,1}\_Config.DMA\_Mode="1" のとき、このアクセス方法となります。DMAx{x=0,1}\_Rd/WrData\_H/L レジスタへのアクセスを DMA アクセスと見なして転送を行います。

何れの場合も、DMAx{x=0,1}\_Config.ActiveDMA を"1"に設定して下さい。

DMA は二つの動作モードと一つの動作オプションを持ちます。

- カウントモード

設定されたカウント数分の DMA 転送を行います。

内部 FIFO に書き込み可能な空き領域／読み出し可能なデータがあり、且つ DMAx{x=0,1}\_Count\_HH,HL,LH,LL レジスタに残りカウントがある場合に XDREQ をアサートし DMA 転送が可能になります。

- フリーランモード

内部 FIFO に書き込み可能な空き領域／読み出し可能なデータがある場合に XDREQ をアサートし DMA 転送が可能になります。

- REQ アサートカウントオプション

CPU のバーストリード／バーストライトに対応するためのオプションです。このオプションは、カウントモード／フリーランモード何れでも使用が可能です。DMAx{x=0,1}\_Config.ReqAssertCount[1:0] ビットに設定されたアサートカウント数以上の、書き込み可能な空き領域／読み出し可能なデータが FIFO にある場合に XDREQ をアサートし DMA 転送が可能になります。従って、基本的には一旦 XDREQ がアサートされると、設定されたアサートカウント数分の転送が保証されることになります。ただし、FIFO の空き領域／データが設定されたアサートカウント数未満の場合も、カウントモードに設定され、且つ FIFO の空き領域／データが残りカウント数以上の場合、XDREQ がアサートされます。この場合の保証される転送数は残りカウント数となります。

DMA の基本はワード単位でのデータ処理となります。バイト単位でのデータ処理はカウントモードで且つ残りカウント数が"1"の時のみ行うことが出来ます。各々の動作モード、オプションにおける XDREQ のアサート条件、また XDREQ アサート時の転送可能数の関係を下表に示します。

表 1-67 動作モード、オプションと転送開始条件一覧

## カウントモード・ReqAssertCountオプション使用時

条件	カウントモード (Count>0)			
	Count ≥ Req		Count < Req	
	Ready ≥ Req	Ready < Req	Ready ≥ Count	Ready < Count
XDREQ	アサート	ネゲート	アサート	ネゲート
転送可能数	Req	–	Count	–

## フリーランモード・ReqAssertCountオプション使用時

条件	フリーランモード	
	–	
	Ready ≥ Req	Ready < Req
XDREQ	アサート	ネゲート
転送可能数	Req	–

## カウントモード・ReqAssertCountオプション未使用時

条件	カウントモード (Count>0)		
	Count ≥ Ready		Count < Ready
	Ready ≥ 2	Ready < 2	Ready ≥ Count
XDREQ	アサート	ネゲート	アサート
転送可能数	Ready (Readyが奇数の場合はReady-1)	–	Count

## フリーランモード・ReqAssertCountオプション未使用時

条件	フリーランモード	
	–	
	Ready ≥ 2	Ready < 2
XDREQ	アサート	ネゲート
転送可能数	Ready (Readyが奇数の場合はReady-1)	–

※ 表中の Req は DMA<sub>x</sub>{x=0,1}Config.ReqAssertCount の設定値を、Ready は FIFO の空き領域／データ数を、Count は DMA x {x=0,1} Count\_HH,HL,LH,LL の値を指しています。

## 1.8.3.2.2. 端子設定

ChipConfig レジスタの設定により、XDREQ<sub>x</sub>{x=1,0}の論理レベル、XDACK<sub>x</sub>{x=1,0}の論理レベルを設定する事が出来ます。以下の説明では特に断りの無い限り、XDREQ,XDACK 共に負論理で記述しております。

## 1.8.3.2.3. カウントモード(ライト)

## 【動作開始】

DMA<sub>x</sub>{x=0,1}\_Count\_HH,HL,LH,LL レジスタにカウント値を設定した後、DMA<sub>x</sub>{x=0,1}\_Control.DMA\_Go ビットに"1"をセットして下さい。内部 FIFO に書き込み可能な空き領域(DMA\_Ready)が2バイト以上あり、且つ残りカウントがある場合に XDREQ をアサートし DMA 転送が可能になります。FIFO に残り1バイトの空き領域しかない場合は、カウントモードに設定され、且つ残りカウント数が"1"の時のみ XDREQ がアサートされます。

奇数バイトの書き込みを行って FIFO にバイト境界が発生した場合には、USB 等からデータが転送された後に、FIFO クリアを行ってバイト境界を解消し、次のライト動作を開始して下さい。例えば DMA から 31 バイト毎に書き込みを行い、USB から 31 バイト毎にデータを転送したい場合、(1)DMA カウント値を31に設定し31バイトデータを書き込む、(2)USB に31バイト分のデータが転送されるのを待つ、

## 1. 機能説明

(3)USB から 31 バイト分のデータが転送されたのを確認した後 FIFO クリアする、以上の動作を繰り返して行って下さい。

動作停止となるまで DMAx{x=0,1}\_Control.DMA\_Running ビットに”1”が読めます。

### 【動作停止】

動作停止条件は以下の 2 つです。

- ・ DMAx{x=0,1}\_Count\_HH,HL,LH,LL レジスタに設定したカウント数分の DMA 転送が完了
- ・ DMAx{x=0,1}\_Control.DMA\_Stop ビットに”1”をライト

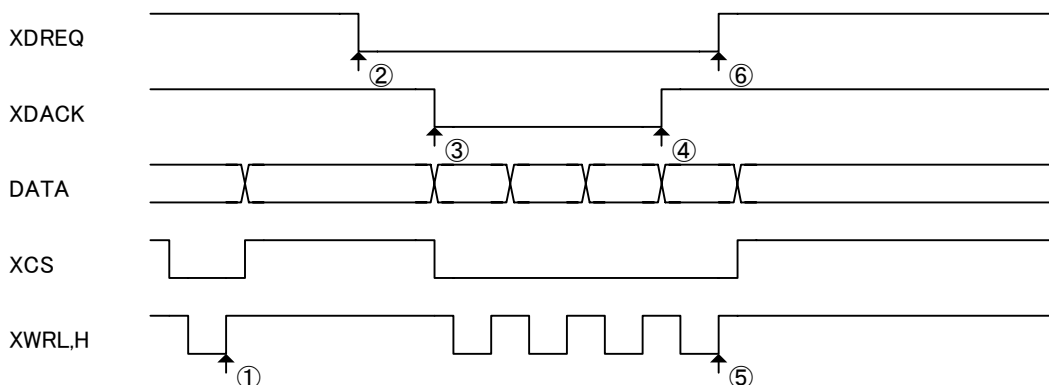
DMA 動作停止時に CPU\_IntStat.DMAx{x=0,1}\_Cmp ビットがセットされます。

DMAx{x=0,1}\_Count\_HH,HL,LH,LL レジスタによる転送停止時は最終アクセスのストローブアサート期間に XDREQ をネゲートします。

DMA\_Stop ビットによる転送停止時は、同期レジスタアクセスのライトタイミングでチップ内部動作を停止し、XDREQ をネゲートします。DMA\_Stop ビットによる DMA の停止を行う場合は、CPU 側の DMAC(マスタ)を先に停止して下さい。

カウントモードで転送を開始し、設定されたカウント数の転送完了前に DMAx{x=0,1}\_Control.DMA\_Stop ビットで転送を停止した場合の動作タイミングを図 1-87 に示します。

ex1.【転送開始条件】カウント(8バイト)<FIFOの空き容量(16バイト) 【転送停止条件】DMA\_Stop



- ①DMA\_Control.DMA\_Goビットへの”1”書き込みによりDMA回路動作開始
- ②USB等からデータが転送される事によりFIFOに空き領域(DMA\_Ready)が出来  
DMA\_Readyを受けてXDREQをアサート
- ③XDACKがアサートされDMA転送開始
- ④カウントモードの転送数完了前に、マスタ側を停止、XDACKをネゲート
- ⑤DMA\_Control.DMA\_Stopビットへの”1”書き込みによりDMA回路停止
- ⑥DMA回路停止を受けてXDREQネゲート

図 1-87 カウントモードライトタイミング 1

カウントモードで転送を開始し、設定されたカウント数分の転送を完了しDMA転送が終了する場合の動作タイミングを図 1-88 に示します。

ex2.【転送開始条件】カウント(8バイト)>FIFOの空き容量(4バイト) 【転送停止条件】カウント0

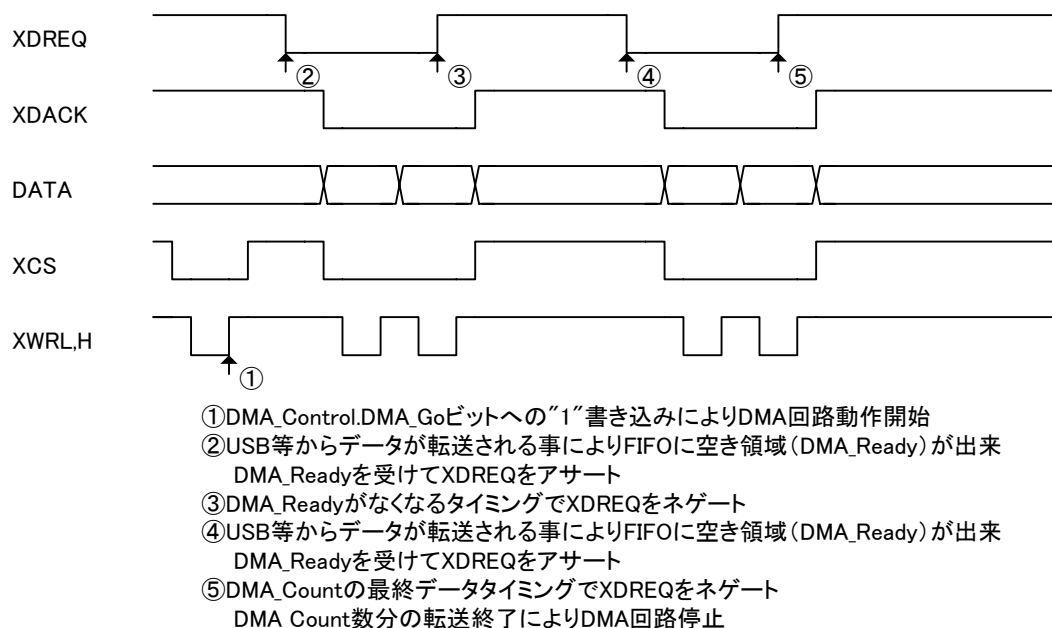


図 1-88 カウントモードライトタイミング 2

#### 1.8.3.2.4. カウントモード(リード)

##### 【動作開始】

DMAx{x=0,1}\_Count\_HH,HL,LH,LL レジスタにカウント値を設定した後、DMAx{x=0,1}\_Control.DMA\_Go ビットに"1"をセットして下さい。内部 FIFO に読み出し可能なデータが2バイト以上あり、且つ残りカウントがある場合に、外部からのリードに対応可能になると XDREQ をアサートします。FIFO に残り1バイトしかデータがない場合はカウントモードに設定され、且つ残りカウント数が"1"の時のみ XDREQ がアサートされます。

デバイス動作時を例にしますと、カウントモードのリード動作では DMAx{x=0,1}\_Count\_HH,HL,LH,LL レジスタに設定されたカウント数以上のデータが、そのDMAが接続されているエンドポイントのFIFOに貯まった時点で、ForceNAK ビットを自動的に"1"にセットし NAK 応答します。また、USB からショートパケットを受信した場合にも DisAF\_NAK\_Short ビットをセットしていなければ、該当するエンドポイントの ForceNAK ビットを自動的に"1"にセットし NAK 応答します。

奇数バイトのリードを行ってバイト境界が発生した場合には、FIFO クリアを行ってバイト境界を解消してから次の転送を行って下さい。例えば、USB から31バイト毎にデータが転送され、DMA から31バイト毎にデータを読み出す場合、(1)USB から31バイトのデータを受信する(この時点で ForceNAK がセットされ該当するエンドポイントは NAK 応答する)、(2)DMA から31バイトのデータを読み出す、(3)FIFO クリアした後 ForceNAK をクリアし USB からの転送を受信可能にする、以上の動作を繰り返して行なって下さい。

動作停止となるまで DMAx{x=0,1}\_Control.DMA\_Running ビットに"1"が読めます。

##### 【動作停止】

動作停止条件は以下の2つです。

- DMAx{x=0,1}\_Count\_HH,HL,LH,LL レジスタに設定したカウント数分のDMA転送が完了
- DMAx{x=0,1}\_Control.DMA\_Stop ビットに"1"をライト

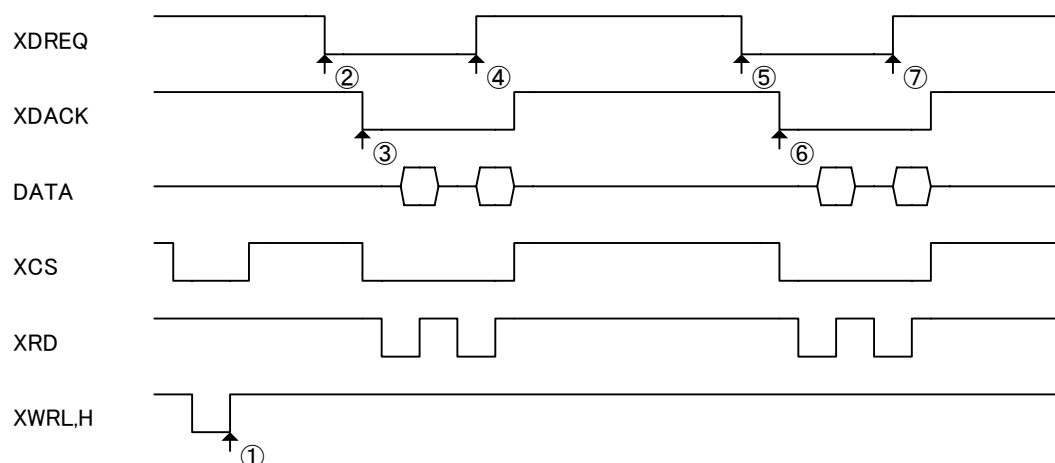
## 1. 機能説明

DMAx{x=0,1}\_Count\_HH,HL,LH,LL レジスタによる転送停止は最終アクセスのストローブ信号アサート期間に XDREQ をネゲートします。

DMA\_Stop ビットによる転送停止は、同期レジスタアクセスのライトタイミングでチップ内部動作を停止し、XDREQ をネゲートします。DMA\_Stop ビットによる DMA の停止を行う場合は、CPU 側の DMAC(マスタ)を先に停止して下さい。

カウントモードで転送を開始し、設定されたカウント数分の転送を完了してDMA転送が終了する場合の動作タイミングを図 1-89 に示します。

ex.【転送開始】カウント(8バイト)>FIFOのデータ(4バイト) 【転送停止】カウント0



- ① DMA\_Control.DMA\_Goビットへの“1”書き込みによりDMA回路動作開始
- ② USB等からFIFOにデータが書き込まれ、  
外部からデータ読み出し可能になることでXDREQをアサート
- ③ XDACKがアサートされDMA転送開始
- ④ FIFOデータが空になるタイミングでXDREQネゲート
- ⑤ USB等からFIFOにデータが書き込まれ、  
外部からデータ読み出し可能になることでXDREQをアサート
- ⑥ XDACKがアサートされDMA転送開始
- ⑦ DMA\_Countの最終データタイミングでXDREQネゲート

図 1-89 カウントモードリードタイミング



### 1.8.3.2.5. フリーランモード(ライト)

#### 【動作開始】

DMAx{x=0,1}\_Config.FreeRunビットをセットした後、DMAx{x=0,1}\_Control.DMA\_Goビットに”1”を書き込んで下さい。内部FIFOに書き込み可能な空き領域が2バイト以上ある場合にXDREQをアサートしDMA転送が可能になります。FIFOに残り1バイトの空き領域しかない場合はフリーランモードではXDREQがアサートされません。転送を行う場合は「1.8.3.2.3.カウントモード(ライト)」をご参照ください。

動作停止となるまでDMAx{x=0,1}\_Control.DMA\_Runningビットに”1”が読めます。

#### 【動作停止】

動作停止条件は以下です。

- ・ DMAx{x=0,1}\_Control.DMA\_Stopビットに”1”をライト

DMA\_Stopビットによる転送停止は、同期レジスタアクセスのライトタイミングでチップ内部動作を停止し、XDREQをネゲートします。DMA\_StopビットによるDMAの停止を行う場合は、CPU側のDMAC(マスタ)を先に停止して下さい。

フリーランモードのDMA転送中にDMAx{x=0,1}\_Count\_HH,HL,LH,LLレジスタの値がオーバーフローすると、CPU\_IntStat.DMAx{x=0,1}\_Countupビットがセットされます。この場合もDMA転送は継続され、DMAx{x=0,1}\_Count\_HH,HL,LH,LLも継続してカウントされます。

動作タイミングはDMAx{x=0,1}\_Count\_HH,HL,LH,LLによる制限が無い事を除き、カウントモードと同等です。

### 1.8.3.2.6. フリーランモード(リード)

#### 【動作開始】

DMAx{x=0,1}\_Config.FreeRunビットをセットした後、DMAx{x=0,1}\_Control.DMA\_Goビットに”1”をセットして下さい。内部FIFOに読み出し可能なデータが2バイト以上あり、外部からのリードに対応可能になるとXDREQをアサートします。FIFOに残り1バイトしか有効なデータがない場合はDMA動作を開始しません。転送を行う場合は「1.8.3.2.4.カウントモード(リード)」をご参照ください。

動作停止となるまでDMAx{x=0,1}\_Control.DMA\_Runningビットに”1”が読めます。

#### 【動作停止】

動作停止条件は以下です。

- ・ DMAx{x=0,1}\_Control.DMA\_Stopビットに”1”をライト

DMA\_Stopビットによる転送停止は、同期レジスタアクセスのライトタイミングでチップ内部動作を停止し、XDREQをネゲートします。DMA\_StopビットによるDMAの停止を行う場合は、CPU側のDMAC(マスタ)を先に停止して下さい。

フリーランモードのDMA転送中にDMAx{x=0,1}\_Count\_HH,HL,LH,LLレジスタの値がオーバーフローすると、CPU\_IntStat.DMAx{x=0,1}\_Countupビットがセットされます。この場合もDMA転送は継続され、DMAx{x=0,1}\_Count\_HH,HL,LH,LLも継続してカウントされます。

動作タイミングはDMAx{x=0,1}\_Count\_HH,HL,LH,LLによる制限が無い事を除き、カウントモードと同等です。

### 1.8.3.2.7. REQアサートカウントオプション(ライト)

#### 【動作開始】

DMAx{x=0,1}\_Config.ReqAssertCount[1:0]ビットでアサートカウント数を設定した後、DMAx{x=0,1}\_Control.DMA\_Goビットに”1”をセットして下さい。内部FIFOに設定されたアサートカウント数以上の書き込み可能な空き領域がある場合にXDREQをアサートしDMA転送が可能になります。従って、一旦XDREQがアサートされると、設定されたアサートカウント数分の転送が保証されることになります。ただし、空き領域がアサートカウント数に満たない場合も、カウントモードに設定され、且つ残りカウント数以上の空き領域がある場合はXDREQをアサートします。この場合、転送可能数は、残りカウント数となります。

## 1. 機能説明

このモードでは、ReqAssertCount[1:0]ビットに設定された転送数毎に一旦 XDREQ がネゲートされます。

動作停止となるまで DMAx{x=0,1}\_Control.DMA\_Running ビットに"1"が読めます。

### 【動作停止】

動作停止条件は「1.8.3.2.3.カウントモード(ライト)」、「1.8.3.2.5.フリーランモード(ライト)」をご参照ください。

ex.【転送開始】REQアサートカウント(8beat: 16byte)

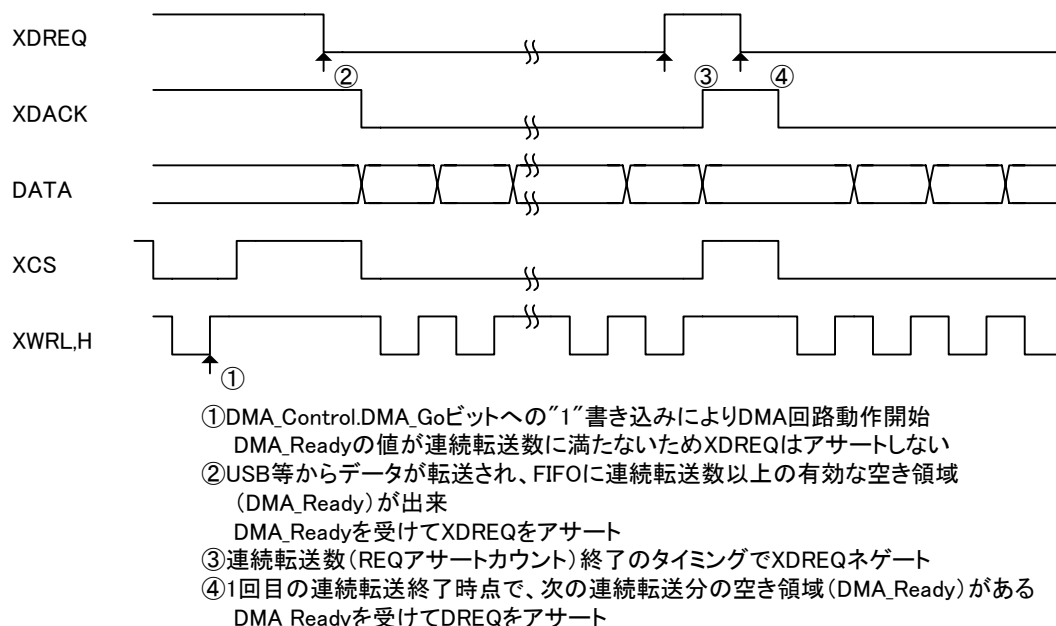


図 1-90 REQ アサートカウントオプションライトタイミング

### 1.8.3.2.8. REQ アサートカウントオプション(リード)

#### 【動作開始】

DMAx{x=0,1}\_Config.ReqAssertCount[[1:0] ビットでアサートカウント数を設定した後、DMAx{x=0,1}\_Control.DMA\_Go ビットに"1"をセットして下さい。内部 FIFO に設定されたアサートカウント数以上の読み出し可能なデータがあり外部からのリードに対応可能になると XDREQ をアサートし DMA 転送が可能になります。従って、一旦 XDREQ がアサートされると、設定されたアサートカウント数分の転送が保証されることになります。ただし、FIFO のデータが REQ アサートカウント数に満たない場合も、カウントモードに設定され且つ残りカウント数以上のデータがある場合は、XDREQ をアサートします。この場合、転送可能数は、残りカウント数となります。

このモードでは、ReqAssertCount[1:0]ビットに設定された転送数毎に一旦 XDREQ がネゲートされます。

動作停止となるまで DMAx{x=0,1}\_Control.DMA\_Running ビットに"1"が読めます。

#### 【動作停止】

動作停止条件は「1.8.3.2.4.カウントモード(リード)」、「1.8.3.2.6.フリーランモード(リード)」をご参照ください。

動作タイミングは図 1-89、図 1-90 をご参照下さい。

### 1.8.3.2.9. DMA の FIFO アクセス端数処理

「1.8.3.1.5.FIFOアクセスの端数処理」をご参照ください。DMAにはバイト読み出し/バイト書き込みの口はございませんのでご注意ください。

## 1.9. Timer

S2R72C05 は 16bit タイマを搭載しております。タイマの分解能(クロック)は約 1.1ms(60MHz の 16bit 分周)です。

TimerSet\_H,L レジスタに設定されたカウント値をダウンカウンタにロードし、ダウンカウントを行います。カウント値が 0x0 になると割り込みステータスを発行します。

### 1.9.1. 動作モード

シングルモード／フリーランモードの 2 つの動作モードをサポートします。

シングルモード：ダウンカウンタの値が 0x0 になるとタイマが停止します。

フリーランモード：カウント値が 0x0 になると、TimerSet\_H,L レジスタの値をダウンカウンタにロードして、ダウンカウントを続けます。

### 1.9.2. 動作開始・停止

タイマの動作を開始させるには、TimerSet\_H,L レジスタへダウンカウント値を書き込んで下さい。書き込みに応じて、ダウンカウンタへ値をロードし大塚が動作を開始します。

タイマ動作を停止させるには、TimerConfig.TimerStop ビットを”1”に設定して下さい。

## 1. 機能説明

---

### 1.10. IDE I/F

ここでは、IDE I/F 機能について説明します。

IDE I/F 機能を使用する際は、IDE\_Config.1.ActiveIDE ビットにあらかじめ”1”を書き込む必要があります。

#### 1.10.1. IDE タスクファイルレジスタへのアクセス

IDE タスクファイルレジスタへのアクセス方法について説明します。

ファームウェアは IDE\_RegAdrs レジスタから IDE\_RegConfig レジスタまでのレジスタ群を介して、IDE タスクファイルレジスタにアクセスすることが出来ます。IDE バス上のリードライトシーケンスは、動作完了まで LSI のハードウェアが行うため、ファームウェアはコマンドをセットした後ポーリングまたは割り込みでシーケンス終了が通知されるまで、この IDE タスクファイルレジスタへのアクセス動作から解放されます。

IDE タスクファイルレジスタへアクセスする際には、IDE バスの XHCS0=0/HDA[2:0]=0 の時は IDE\_Tmod レジスタの設定値が、それ以外のアドレスでは IDE\_Rmod レジスタの設定値が使われて転送が行われるため、IDE の転送モードに応じて IDE\_Rmod/IDE\_Tmod レジスタにあらかじめ適切な値を設定する必要があります。

##### 1.10.1.1. IDE タスクファイルレジスタからのリード

IDE\_RegAdrs.IDE\_RegAddress[3:0] ビットにアクセスしたいアドレスをあらかじめまたは同時にセットし、IDE\_RegAdrs.IDE\_RdReg ビットに”1”をライトすることにより、IDE タスクファイルレジスタからのリード動作が行われます。

IDE\_RegAdrs.IDE\_RdReg ビットが”0”になり、かつ IDE\_IntStat.IDE\_RegCmp ビットが”1”になりリード動作が終了します。

IDE タスクファイルレジスタからリードされたデータは IDE\_RdRegValue レジスタに格納されます。

##### 1.10.1.2. IDE タスクファイルレジスタへのライト

IDE\_WrRegValue レジスタにあらかじめライトしたいデータをセットし、IDE\_RegAdrs.IDE\_RegAddress[3:0] ビットにアクセスしたいアドレスをあらかじめまたは同時にセットし、IDE\_RegAdrs.IDE\_WrReg ビットに”1”をライトすることにより、IDE タスクファイルレジスタへのライト動作が行われます。

IDE\_RegAdrs.IDE\_WrReg ビットが”0”になり、かつ IDE\_IntStat.IDE\_RegCmp ビットが”1”になりライト動作が終了します。

##### 1.10.1.3. IDE タスクファイルレジスタへのシーケンシャルライト

IDE\_SeqWrRegControl レジスタを使うことにより、あらかじめセットしておいた最大 16 組のアドレス・データで IDE タスクファイルレジスタへシーケンシャルライトすることが出来ます。

あらかじめ IDE\_SeqWrRegAdrs.IDE\_SeqWrRegAddress[3:0] ビットにライトしたいアドレスを、IDE\_SeqWrRegValue レジスタにライトしたいバイトデータを対で最大 16 組ライトします。XHCS0=0/HDA[2:0]=0 のアドレスの時はワードアクセスになるため、下位／上位の順にデータを 2 回書き込む必要があります、アドレス・データ対は 2 組使用されます。その後 IDE\_SeqWrRegControl.IDE\_SeqWrReg ビットに”1”を書き込むとシーケンシャルライト動作が行われます。IDE\_SeqWrRegControl.IDE\_SeqWrReg ビットが”0”になり、かつ IDE\_IntStat.IDE\_SeqWrRegCmp ビットが”1”になりシーケンシャルライト動作が終了します。

シーケンシャルライト動作の起動前に、あらかじめセットしておいたアドレス・データが不要になった場合は、IDE\_SeqWrRegControl.IDE\_SeqWrRegClr ビットに”1”をライトしてアドレス・データを破棄することが出来ます。

シーケンシャルライト動作中にファームウェアが通常の IDE タスクファイルレジスタのリードライト動作を行うと、IDE\_IntStat.IDE\_RegErr ビットが”1”になり、そのリードライトは無視されます。

##### 1.10.1.4. IDE タスクファイルレジスタからのオートステータスレジスタリード

IDE バスの HINTRQ がセットされた場合、IDE\_RegConfig.EnAutoStsRd ビットに”1”が書き込まれて

いると、LSI は自動で IDE ステータスレジスタ(XHCS0=0,HDA[2:0]=7)をリードし、リードしたデータを IDE\_RdRegValue レジスタに格納した後、IDE\_IntStat.IDE\_CompleteINTRQ ビットが”1”になります。

オートステータスレジスタリード動作からファームウェアが IDE\_RdRegValue\_1 レジスタを読み出すまでの期間に、ファームウェアが通常の IDE タスクファイルレジスタのリードライト動作を行うと、IDE\_IntStat.IDE\_RegErr ビットが”1”になり、そのリードライトは無視されます。

## 1. 機能説明

---

### 1.10.2. PIO アクセス

PIO モードによる DMA 機能について説明します。

PIO モードによる DMA では、IDE\_Tmod レジスタの設定値が使われて DMA 転送が行われるため、IDE の転送モードに応じて IDE\_Tmod レジスタにあらかじめ適切な値を設定する必要があります。

#### 1.10.2.1. PIO リード DMA

PIO リード DMA は以下の手順で動作します。

IDE\_Config\_0.DMA ビット及び IDE\_Config\_0.Ultra ビット両方に”0”を書き込みます。

IDE\_CountH/M/L レジスタに転送バイト数を設定します。

IDE\_Control.IDE\_Clr ビットに”1”を書き込み、IDE 回路を初期状態にします。(必ずしも必要ではありません)

IDE\_Control.IDE\_Go ビットに”1”を書き込むと DMA 動作が開始され、転送とともに IDE\_CountH/M/L レジスタの内容が減っていき、カウントが 0 になって IDE の DMA 動作と IDE からリードしたデータの内部 FIFO への書き込みが全て終了すると、IDE\_Control.IDE\_Go ビットが”0”になり、かつ IDE\_IntStat.IDE\_Cmp ビットが”1”になり DMA 動作が終了します。

DMA 転送中に IDE\_Control.IDE\_Go ビットに”0”を書き込むと DMA 動作が中断されて DMA が終了します。その時内部 FIFO まで転送したデータは有効ですが、回路の中間バッファに溜まったデータは捨てられるため、IDE の転送バイト数を正確に管理することは出来ません。

#### 1.10.2.2. PIO ライト DMA

PIO ライト DMA は以下の手順で動作します。

IDE\_Config\_0.DMA ビット及び IDE\_Config\_0.Ultra ビット両方に”0”を書き込みます。

IDE\_CountH/M/L レジスタに転送バイト数を設定します。

IDE\_Control.IDE\_Clr ビットに”1”を書き込み、IDE 回路を初期状態にします。(必ずしも必要ではありません)

IDE\_Control.IDE\_Go ビットに”1”を書き込むと DMA 動作が開始され、転送とともに IDE\_CountH/M/L レジスタの内容が減っていき、カウントが 0 になって IDE への DMA ライトが全て終了すると、IDE\_Control.IDE\_Go ビットが”0”になり、かつ IDE\_IntStat.IDE\_Cmp ビットが”1”になり DMA 動作が終了します。

DMA 転送中に IDE\_Control.IDE\_Go ビットに”0”を書き込むと DMA 動作が中断されて DMA が終了します。その時まで IDE にライトしたデータは有効ですが、回路の中間バッファに溜まったデータは捨てられるため、IDE の転送バイト数を正確に管理することは出来ません。

### 1.10.3. Multi-Word DMA

Multi-Word DMA モードによる DMA 機能について説明します。

Multi-Word DMA モードによる DMA では、IDE\_Tmod レジスタの設定値が使われて DMA 転送が行われるため、IDE の転送モードに応じて IDE\_Tmod レジスタにあらかじめ適切な値を設定する必要があります。

#### 1.10.3.1. Multi-Word DMA リード

Multi-Word DMA リードは以下の手順で動作します。

IDE\_Config\_0.DMA ビットに”1”を、IDE\_Config\_0.Ultra ビットに”0”を書き込みます。

IDE\_Config\_1.DelayStrobe ビット、IDE\_Config\_1.InterLock ビットに適切な値を書き込みます。

IDE\_CountH/M/L レジスタに転送バイト数を設定します。

IDE\_Control.IDE\_Clr ビットに”1”を書き込み、IDE 回路を初期状態にします。(必ずしも必要ではありません)

IDE\_Control.IDE\_Go ビットに”1”を書き込むと DMA 動作が開始され、転送とともに IDE\_CountH/M/L レジスタの内容が減っていき、カウントが 0 になって IDE の DMA 動作と IDE からリードしたデータの内部 FIFO への書き込みが全て終了すると、IDE\_Control.IDE\_Go ビットが”0”になり、かつ IDE\_IntStat.IDE\_Cmp ビットが”1”になり DMA 動作が終了します。

DMA 転送中に IDE\_Control.IDE\_Go ビットに”0”を書き込むと DMA 動作が中断されて DMA が終了し

ます。その時内部 FIFO まで転送したデータは有効ですが、回路の中間バッファに溜まったデータは捨てられるため、IDE の転送バイト数を正確に管理することは出来ません。

#### 1.10.3.2. Multi-Word DMA ライト

Multi-Word DMA ライトは以下の手順で動作します。

IDE\_Config\_0.DMA ビットに”1”を、IDE\_Config\_0.Ultra ビットに”0”を書き込みます。

IDE\_Config\_1.DelayStrobe ビット、IDE\_Config\_1.InterLock ビットに適切な値を書き込みます。

IDE\_CountH/M/L レジスタに転送バイト数を設定します。

IDE\_Control.IDE\_Clr ビットに”1”を書き込み、IDE 回路を初期状態にします。(必ずしも必要ではありません)

IDE\_Control.IDE\_Go ビットに”1”を書き込むと DMA 動作が開始され、転送とともに IDE\_CountH/M/L レジスタの内容が減っていき、カウントが 0 になって IDE への DMA ライトが全て終了すると、IDE\_Control.IDE\_Go ビットが”0”になり、かつ IDE\_IntStat.IDE\_Cmp ビットが”1”になり DMA 動作が終了します。

DMA 転送中に IDE\_Control.IDE\_Go ビットに”0”を書き込むと DMA 動作が中断されて DMA が終了します。その時まで IDE にライトしたデータは有効ですが、回路の中間バッファに溜まったデータは捨てられるため、IDE の転送バイト数を正確に管理することは出来ません。

## 1. 機能説明

---

### 1.10.4. Ultra DMA

Ultra DMA モードによる DMA 機能について説明します。

Ultra DMA モードによる DMA では、IDE\_Umod レジスタの設定値が使われて DMA 転送が行われるため、IDE の転送モードに応じて IDE\_Umod レジスタにあらかじめ適切な値を設定する必要があります。

#### 1.10.4.1. Ultra DMA リード

Ultra DMA リードは以下の手順で動作します。

IDE\_Config\_0.DMA ビット及び IDE\_Config\_0.Ultra ビットに”1”を書き込みます。

IDE\_CountH/M/L レジスタに転送バイト数を設定します。

IDE\_Control.IDE\_Clr ビットに”1”を書き込み、IDE 回路を初期状態にします。(必ずしも必要ではありません)

IDE\_Control.IDE\_Go ビットに”1”を書き込むと DMA 動作が開始され、転送とともに IDE\_CountH/M/L レジスタの内容が減っていき、カウントが 0 になって IDE の DMA 動作と IDE からリードしたデータの内部 FIFO への書き込みが全て終了すると、IDE\_Control.IDE\_Go ビットが”0”になり、かつ IDE\_IntStat.IDE\_Cmp ビットが”1”になり DMA 動作が終了します。

DMA 転送中に IDE\_Control.IDE\_Go ビットに”0”を書き込むと DMA 動作が中断されて DMA が終了します。その時内部 FIFO まで転送したデータは有効ですが、回路の中間バッファに溜まったデータは捨てられるため、IDE の転送バイト数を正確に管理することは出来ません。また、この場合接続された IDE デバイスにはホストターミネートを通知します。

DMA 転送中にデバイスターミネートが発生した場合、IDE\_IntStat.DetectTerm ビットが”1”になり接続された IDE デバイスは転送を止めてしまいましたが、起動した DMA は終了しませんので IDE\_Control.IDE\_Go ビットに”0”を書き込んで DMA を終了させる必要があります。

#### 1.10.4.2. Ultra DMA ライト

Ultra DMA ライトは以下の手順で動作します。

IDE\_Config\_0.DMA ビット及び IDE\_Config\_0.Ultra ビットに”1”を書き込みます。

IDE\_CountH/M/L レジスタに転送バイト数を設定します。

IDE\_Control.IDE\_Clr ビットに”1”を書き込み、IDE 回路を初期状態にします。(必ずしも必要ではありません)

IDE\_Control.IDE\_Go ビットに”1”を書き込むと DMA 動作が開始され、転送とともに IDE\_CountH/M/L レジスタの内容が減っていき、カウントが 0 になって IDE への DMA ライトが全て終了すると、IDE\_Control.IDE\_Go ビットが”0”になり、かつ IDE\_IntStat.IDE\_Cmp ビットが”1”になり DMA 動作が終了します。

DMA 転送中に IDE\_Control.IDE\_Go ビットに”0”を書き込むと DMA 動作が中断されて DMA が終了します。その時まで IDE にライトしたデータは有効ですが、回路の中間バッファに溜まったデータは捨てられるため、IDE の転送バイト数を正確に管理することは出来ません。また、この場合接続された IDE デバイスにはホストターミネートを通知します。

DMA 転送中にデバイスターミネートが発生した場合、IDE\_IntStat.DetectTerm ビットが”1”になり接続された IDE デバイスは転送を止めてしまいましたが、起動した DMA は終了しませんので IDE\_Control.IDE\_Go ビットに”0”を書き込んで DMA を終了させる必要があります。



## 1.10.5. IDE 転送モードの設定について

IDE 転送の各モードにする設定値を下表に示します。

●Register Mode (XHCS0=0,HDA=0 以外のファームウェアによる IDE タスクファイルレジスタへのアクセス時)

Mode	レジスタ設定値(必須)		レジスタ設定値(推奨)
	IDE_Config_0.Ultra	IDE_Config_0.DMA	IDE_Rmod
0	影響しない	影響しない	FFh
1	影響しない	影響しない	F1h
2	影響しない	影響しない	F0h
3	影響しない	影響しない	22h
4	影響しない	影響しない	10h

●PIO Mode (XHCS0=0,HDA=0 のファームウェアによる IDE タスクファイルレジスタへのアクセス時)

Mode	レジスタ設定値(必須)		レジスタ設定値(推奨)
	IDE_Config_0.Ultra	IDE_Config_0.DMA	IDE_Tmod
0	影響しない	影響しない	FFh
1	影響しない	影響しない	88h
2	影響しない	影響しない	44h
3	影響しない	影響しない	22h
4	影響しない	影響しない	10h

●PIO Mode (DMA 転送時)

Mode	レジスタ設定値(必須)		レジスタ設定値(推奨)
	IDE_Config_0.Ultra	IDE_Config_0.DMA	IDE_Tmod
0	0	0	FFh
1	0	0	88h
2	0	0	44h
3	0	0	22h
4	0	0	10h

●Multi-word DMA Mode (DMA 転送時)

Mode	レジスタ設定値(必須)		レジスタ設定値(推奨)
	IDE_Config_0.Ultra	IDE_Config_0.DMA	IDE_Tmod
0	0	1	BBh
1	0	1	20h
2	0	1	10h

●Ultra Mode (DMA 転送時、DATA-OUT 時)

Mode	レジスタ設定値(必須)		レジスタ設定値(推奨)
	IDE_Config_0.Ultra	IDE_Config_0.DMA	IDE_Umod
0	1	1	06h
1	1	1	04h
2	1	1	03h
3	1	1	02h
4	1	1	01h
5	1	1	00h

注：Ultra Mode において IDE バスからデータを入力する DATA-IN 時は、IDE\_Umod レジスタの設定によらず全てのモードでデータ受信可能です。

## 1. 機能説明

### 1.11. バウンダリスキャン(JTAG)

バウンダリスキャン(JTAG)は、TEST 端子が L(デフォルト)の時に使用可能です。バウンダリスキャンは、JTAG(IEEE 1149.1)仕様に準拠した、BSR(Boundary Scan Register)、それを繋ぐスキャンパスおよび TAP コントローラで構成されます。バウンダリスキャンの接続情報は、BSDL フォーマットにて提供可能です。

#### 1.11.1. 対応インストラクション

本 LSI の JTAG インストラクションビット幅は 4 ビットとなっており、以下の JTAG インストラクションに対応致します。

表 1-68 JTAG インストラクションコード

インストラクション	説明	コード
SAMPLE/PRELOAD	LSI 内部状態の BSR への取り込み及びデータ設定。	0010
BYPASS	BSR によるスキャンパスをバイパスする。	1111
EXTEST	デバイスの物理的な接続チェック。	0000
CLAMP	出力値を保持しながら、スキャンパスをバイパスする。	0011
HIGHZ	出力を全て Hi-Z に固定する。	0100
IDCODE	定められた DEVICE_CODE の出力。	0001

#### 1.11.2. DEVICE\_CODE に関して

IDCODE インストラクションに対する DEVICE\_CODE の構成要素は以下のとおりとなります。

表 1-69 DEVICE\_CODE

Version	1
Part Number	0x0015
Manufacturer	0x0BE

従って、IDCODE インストラクションに対する DEVICE\_CODE 応答は、  
0001\_00000000000010101\_00010111110\_1  
となります。

#### 1.11.3. バウンダリスキャン除外端子

本 LSI の端子のうち、DP\_A、DM\_A、DP\_B、DM\_B、R1\_A、R1\_B、XI、XO、VBUS\_B、および、TEST にはバウンダリスキャンセルが挿入されていないため、スキャン対象外となります。

2. レジスタ

レジスタは、初期レジスタ、デバイス／ホスト共通レジスタ、デバイス・レジスタ、ホスト・レジスタに分類され、デバイス・レジスタとホスト・レジスタとは HostDeviceSel.HOSTxDEVICE ビットの設定で、マップの切替を行います。同ビットが”0”のときにデバイス・レジスタが選択され、”1”のときにホスト・レジスタが選択されます。同ビットの設定を変えても、レジスタの設定値はクリアされません。

リザーブレジスタ・ビットには”1”を書き込まないでください。

本LSIのレジスタマップはビッグエンディアンで定義されています。リトルエンディアンCPUとの接続については、「Appendix B リトルエンディアンのCPUへの接続」を参照して下さい。

2.1. 初期レジスタマップ

S2R72C05 は、ハードリセットされてから、本「初期レジスタ」が設定されてその設定が有効にされるまでの間(未初期化期間)は、本「初期レジスタ」のみがアクセス可能です。詳細は「1.8.2.CPUIFのモード設定」を参照して下さい。

Word Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0xXXX *	<b>CPUIF_MODE</b>	W	0xXXXX						CPU_Endian	BusMode	
									CPU_Endian	BusMode	

\* アドレスは Don't care となります。未初期化期間において、本 LSI へのライトアクセスは、全てこのレジスタへのライトアクセスと見なして処理します。

## 2. レジスタ

### 2.2. デバイス／ホスト共通レジスタマップ

SLEEP / SNOOZE 時にも読み書きできるレジスタを**太字斜体**で示します。

それ以外のレジスタは、ACTIVE60 / ACT\_DEVICE / ACT\_HOST / ACT\_ALL時に読み書きできます。  
(各ステートについては、「1.5.パワーマネージメント機能」を参照して下さい。)

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x000	<b><i>MainIntStat</i></b>	R/(W)	0x00	<b><i>DeviceIntStat</i></b>	<b><i>HostIntStat</i></b>	CPU_IntStat	IDE_IntStat	MediaFIFO_IntStat	AREAnIntStat		<b><i>FinishedPM</i></b>
0x001	<b><i>DeviceIntStat</i></b>	R/(W)	0x00	<b><i>VBUS_Changed</i></b>		<b><i>D_SIE_IntStat</i></b>	D_BulkIntStat	RcvEP0SETUP	D_FIFO_IntStat	D_EP0IntStat	D_EPrIntStat
0x002	<b><i>HostIntStat</i></b>	R/(W)	0x00	<b><i>VBUS_Err</i></b>	<b><i>LineStateChanged</i></b>	H_SIE_IntStat_1	H_SIE_IntStat_0	H_FrameIntStat	H_FIFO_IntStat	H_CH0IntStat	H_CHrIntStat
0x003	CPU_IntStat	R/(W)	0x00	RAM_RdCmp	C_TimerCmp			DMA1_Countup	DMA1_Cmp	DMA0_Countup	DMA0_Cmp
0x004	IDE_IntStat	R/(W)	0x00	IDE_RegCmp	IDE_RegErr	IDE_SeqWrRegCmp	CompleteINTRQ		IDE_Cmp	DetectINTRQ	DetectTerm
0x005	MediaFIFO_IntStat	R/(W)	0x01		MediaIDE_Cmp				FIFO_NotEmpty	FIFO_Full	FIFO_Empty
0x006			0xXX								
0x007			0xXX								
0x008			0xXX								
0x009			0xXX								
0x00A			0xXX								
0x00B			0xXX								
0x00C			0xXX								
0x00D			0xXX								
0x00E			0xXX								
0x00F			0xXX								

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x010	<b><i>MainIntEnb</i></b>	R/W	0x00	<b><i>EnDeviceIntStat</i></b>	<b><i>EnHostIntStat</i></b>	EnCPU_IntStat	EnIDE_IntStat	EnMediaFIFO_IntStat	EnAREAnIntStat		<b><i>EnFinishedPM</i></b>
0x011	<b><i>DeviceIntEnb</i></b>	R/W	0x00	<b><i>EnVBUS_Changed</i></b>		<b><i>EnD_SIE_IntStat</i></b>	EnD_BulkIntStat	EnRcvEP0SETUP	EnD_FIFO_IntStat	EnD_EP0IntStat	EnD_EPrIntStat
0x012	<b><i>HostIntEnb</i></b>	R/W	0x00	<b><i>EnVBUS_Err</i></b>	<b><i>EnLineStateChanged</i></b>	EnH_SIE_IntStat_1	EnH_SIE_IntStat_0	EnH_FrameIntStat	EnH_FIFO_IntStat	EnH_CH0IntStat	EnH_CHrIntStat
0x013	CPU_IntEnb	R/W	0x00	EnRAM_RdCmp	EnC_TimerCmp			EnDMA1_Countup	EnDMA1_Cmp	EnDMA0_Countup	EnDMA0_Cmp
0x014	IDE_IntEnb	R/W	0x00	EnIDE_RegCmp	EnIDE_RegErr	EnIDE_SeqWrRegCmp	EnCompleteINTRQ		EnIDE_Cmp	EnDetectINTRQ	EnDetectTerm
0x015	MediaFIFO_IntEnb	R/W	0x00		EnMediaIDE_Cmp				EnFIFO_NotEmpty	EnFIFO_Full	EnFIFO_Empty
0x016			0xXX								
0x017			0xXX								
0x018			0xXX								
0x019			0xXX								
0x01A			0xXX								
0x01B			0xXX								
0x01C			0xXX								
0x01D			0xXX								
0x01E			0xXX								
0x01F			0xXX								

## 2. レジスタ

SLEEP / SNOOZE 時にでも読み書きできるレジスタを**太字斜体**で示します。

それ以外のレジスタは、ACTIVE60 / ACT\_DEVICE / ACT\_HOST / ACT\_ALL時に読み書きできます。  
(各状態については、「1.5. パワーマネージメント機能」を参照して下さい。)

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x020	<u>RevisionNum</u>	R	0x60	RevisionNumer							
0x021	<u>ChipReset</u>	W	0xXX								AllReset
0x022	<u>PM_Control_0</u>	R/W	0x00	GoSLEEP	GoSNOOZE	GoActive60	GoActDevice	GoActHost	GoActAllDev	GoActAllHost	
0x023	<u>PM_Control_1</u>	R	0x00	MonActFunc				PM_State[3:0]			
0x024	<u>WakeupTim_H</u>	R/W	0x00	WakeupTim[15:8]							
0x025	<u>WakeupTim_L</u>	R/W	0x00	WakeupTim[7:0]							
0x026	<u>H_USB_Control</u>	R/W	0x00	VBUS_Enb							
0x027	<u>H_XcvrControl</u>	R/W	0x91	TermSelect	RemoveRPD	XcvrSelect[1:0]				OpMode[1:0]	
0x028	<u>D_USB_Status</u>	R/W	0xXX	VBUS	FSxHS					LineState[1:0]	
0x029	<u>H_USB_Status</u>	R		VBUS_State						LineState[1:0]	
0x02A			0xXX								
0x02B			0xXX								
0x02C			0xXX								
0x02D			0xXX								
0x02E			0xXX								
0x02F			0xXX								

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x030	FIFO_Rd_0	R	0xXX	FIFO_Rd_0[7:0]							
0x031	FIFO_Rd_1	R	0xXX	FIFO_Rd_1[7:0]							
0x032	FIFO_Wr_0	W	0xXX	FIFO_Wr_0[7:0]							
0x033	FIFO_Wr_1	W	0xXX	FIFO_Wr_1[7:0]							
0x034	FIFO_RdRemain_H	R	0x00	RdRemainValid				RdRemain[12:8]			
0x035	FIFO_RdRemain_L	R	0x00	RdRemain[7:0]							
0x036	FIFO_WrRemain_H	R	0x00					WrRemain[12:8]			
0x037	FIFO_WrRemain_L	R	0x00	WrRemain[7:0]							
0x038	FIFO_ByteRd	R	0xXX	FIFO_ByteRd[7:0]							
0x039			0xXX								
0x03A	FIFO_ByteWr	W	0xXX	FIFO_ByteWr[7:0]							
0x03B			0xXX								
0x03C			0xXX								
0x03D			0xXX								
0x03E			0xXX								
0x03F			0xXX								

## 2. レジスタ

SLEEP / SNOOZE 時にでも読み書きできるレジスタを**太字斜体**で示します。

それ以外のレジスタは、ACTIVE60 / ACT\_DEVICE / ACT\_HOST / ACT\_ALL時に読み書きできます。  
(各状態については、「1.5. パワーマネージメント機能」を参照して下さい。)

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x040	RAM_RdAdrs_H	R/W	0x00					RAM_RdAdrs[12:8]			
0x041	RAM_RdAdrs_L	R/W	0x00	RAM_RdAdrs[7:2]							
0x042	RAM_RdControl	R/W	0x00	RAM_GoRdCBW_CSW	RAM_GoRd						
0x043	RAM_RdCount	R/W	0x00			RAM_RdCount[5:2]					
0x044	RAM_WrAdrs_H	R/W	0x00					RAM_WrAdrs[12:8]			
0x045	RAM_WrAdrs_L	R/W	0x00	RAM_WrAdrs[7:0]							
0x046	RAM_WrDoor_0	W	0xXX	RAM_WrDoor_0[7:0]							
0x047	RAM_WrDoor_1	W	0xXX	RAM_WrDoor_1[7:0]							
0x048	MediaFIFO_Control	W	0xXX								MediaFIFO_Clr
0x049	ClrAllMediaFIFO_Join	W	0xXX	ClrJoinIDE				ClrJoinDMA1	ClrJoinDMA0	ClrJoinCPU_Rd	ClrJoinCPU_Wr
0x04A	MediaFIFO_Join	R/W	0x00	JoinIDE				JoinDMA1	JoinDMA0	JoinCPU_Rd	JoinCPU_Wr
0x04B			0xXX								
0x04C			0xXX								
0x04D			0xXX								
0x04E			0xXX								
0x04F			0xXX								

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x050	<b>RAM_Rd_00</b>	R	0x00	RAM_Rd_00[7:0]							
0x051	<b>RAM_Rd_01</b>	R	0x00	RAM_Rd_01[7:0]							
0x052	<b>RAM_Rd_02</b>	R	0x00	RAM_Rd_02[7:0]							
0x053	<b>RAM_Rd_03</b>	R	0x00	RAM_Rd_03[7:0]							
0x054	<b>RAM_Rd_04</b>	R	0x00	RAM_Rd_04[7:0]							
0x055	<b>RAM_Rd_05</b>	R	0x00	RAM_Rd_05[7:0]							
0x056	<b>RAM_Rd_06</b>	R	0x00	RAM_Rd_06[7:0]							
0x057	<b>RAM_Rd_07</b>	R	0x00	RAM_Rd_07[7:0]							
0x058	<b>RAM_Rd_08</b>	R	0x00	RAM_Rd_08[7:0]							
0x059	<b>RAM_Rd_09</b>	R	0x00	RAM_Rd_09[7:0]							
0x05A	<b>RAM_Rd_0A</b>	R	0x00	RAM_Rd_0A[7:0]							
0x05B	<b>RAM_Rd_0B</b>	R	0x00	RAM_Rd_0B[7:0]							
0x05C	<b>RAM_Rd_0C</b>	R	0x00	RAM_Rd_0C[7:0]							
0x05D	<b>RAM_Rd_0D</b>	R	0x00	RAM_Rd_0D[7:0]							
0x05E	<b>RAM_Rd_0E</b>	R	0x00	RAM_Rd_0E[7:0]							
0x05F	<b>RAM_Rd_0F</b>	R	0x00	RAM_Rd_0F[7:0]							

## 2. レジスタ

SLEEP / SNOOZE 時にでも読み書きできるレジスタを**太字斜体**で示します。

それ以外のレジスタは、ACTIVE60 / ACT\_DEVICE / ACT\_HOST / ACT\_ALL時に読み書きできます。  
(各状態については、「1.5. パワーマネージメント機能」を参照して下さい。)

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x060	<b><i>RAM_Rd_10</i></b>	R	0x00	RAM_Rd_10[7:0]							
0x061	<b><i>RAM_Rd_11</i></b>	R	0x00	RAM_Rd_11[7:0]							
0x062	<b><i>RAM_Rd_12</i></b>	R	0x00	RAM_Rd_12[7:0]							
0x063	<b><i>RAM_Rd_13</i></b>	R	0x00	RAM_Rd_13[7:0]							
0x064	<b><i>RAM_Rd_14</i></b>	R	0x00	RAM_Rd_14[7:0]							
0x065	<b><i>RAM_Rd_15</i></b>	R	0x00	RAM_Rd_15[7:0]							
0x066	<b><i>RAM_Rd_16</i></b>	R	0x00	RAM_Rd_16[7:0]							
0x067	<b><i>RAM_Rd_17</i></b>	R	0x00	RAM_Rd_17[7:0]							
0x068	<b><i>RAM_Rd_18</i></b>	R	0x00	RAM_Rd_18[7:0]							
0x069	<b><i>RAM_Rd_19</i></b>	R	0x00	RAM_Rd_19[7:0]							
0x06A	<b><i>RAM_Rd_1A</i></b>	R	0x00	RAM_Rd_1A[7:0]							
0x06B	<b><i>RAM_Rd_1B</i></b>	R	0x00	RAM_Rd_1B[7:0]							
0x06C	<b><i>RAM_Rd_1C</i></b>	R	0x00	RAM_Rd_1C[7:0]							
0x06D	<b><i>RAM_Rd_1D</i></b>	R	0x00	RAM_Rd_1D[7:0]							
0x06E	<b><i>RAM_Rd_1E</i></b>	R	0x00	RAM_Rd_1E[7:0]							
0x06F	<b><i>RAM_Rd_1F</i></b>	R	0x00	RAM_Rd_1F[7:0]							

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x070			0xXX								
0x071	DMA0_Config	R/W	0x00	FreeRun	DMA_Mode			ActiveDMA		ReqAssertCount[1:0]	
0x072	DMA0_Control	R/W	0x00	DMA_Running			CounterClr	Dir		DMA_Stop	DMA_Go
0x073			0xXX								
0x074	DMA0_Remain_H	R	0x00				DMA_Remain[12:8]				
0x075	DMA0_Remain_L	R	0x00	DMA_Remain[7:0]							
0x076			0xXX								
0x077			0xXX								
0x078	DMA0_Count_HH	R/W	0x00	DMA_Count[31:24]							
0x079	DMA0_Count_HL	R/W	0x00	DMA_Count[23:16]							
0x07A	DMA0_Count_LH	R/W	0x00	DMA_Count[15:8]							
0x07B	DMA0_Count_LL	R/W	0x00	DMA_Count[7:0]							
0x07C	DMA0_RdData_0	R	0xXX	DMA_RdData_0[7:0]							
0x07D	DMA0_RdData_1	R	0xXX	DMA_RdData_1[7:0]							
0x07E	DMA0_WrData_0	W	0xXX	DMA_WrData_0[7:0]							
0x07F	DMA0_WrData_1	W	0xXX	DMA_WrData_1[7:0]							

## 2. レジスタ

SLEEP / SNOOZE 時にでも読み書きできるレジスタを**太字斜体**で示します。

それ以外のレジスタは、ACTIVE60 / ACT\_DEVICE / ACT\_HOST / ACT\_ALL時に読み書きできます。  
(各ステートについては、「1.5.パワーマネージメント機能」を参照して下さい。)

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x080			0xXX								
0x081	DMA1_Config	R/W	0x00	FreeRun	DMA_Mode			ActiveDMA		ReqAssertCount[1:0]	
0x082	DMA1_Control	R/W	0x00	DMA_Running			CounterClr	Dir		DMA_Stop	DMA_Go
0x083			0xXX								
0x084	DMA1_Remain_H	R	0x00				DMA_Remain[12:8]				
0x085	DMA1_Remain_L	R	0x00	DMA_Remain[7:0]							
0x086			0xXX								
0x087			0xXX								
0x088	DMA1_Count_HH	R/W	0x00	DMA_Count[31:24]							
0x089	DMA1_Count_HL	R/W	0x00	DMA_Count[23:16]							
0x08A	DMA1_Count_LH	R/W	0x00	DMA_Count[15:8]							
0x08B	DMA1_Count_LL	R/W	0x00	DMA_Count[7:0]							
0x08C	DMA1_RdData_0	R	0xXX	DMA_RdData_0[7:0]							
0x08D	DMA1_RdData_1	R	0xXX	DMA_RdData_1[7:0]							
0x08E	DMA1_WrData_0	W	0xXX	DMA_WrData_0[7:0]							
0x08F	DMA1_WrData_1	W	0xXX	DMA_WrData_1[7:0]							

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x090	<b>IDE_Status</b>	R/W	0x00	DMARQ	DMACK	INTRQ	IORDY			PDIAG	DASP
0x091	<b>IDE_Control</b>	R/W	0x00		IDE_Clr			Dir			IDE_Go
0x092	<b>IDE_Config_0</b>	R/W	0x00	IDE_BusReset	IDE_LongBusReset					Ultra	DMA
0x093	<b>IDE_Config_1</b>	R/W	0x04	ActiveIDE	DelayStrobe		InterLock		Swap		
0x094	<b>IDE_Rmod</b>	R/W	0x00	RegisterAssertPulseWidth[3:0]				RegisterNegatePulseWidth[3:0]			
0x095	<b>IDE_Tmod</b>	R/W	0x00	TransferAssertPulseWidth[3:0]				TransferNegatePulseWidth[3:0]			
0x096	<b>IDE_Umod</b>	R/W	0x00					UltraDMA_Cycle[3:0]			
0x097			0xXX								
0x098			0xXX								
0x099			0xXX								
0x09A	<b>IDE_CRC_H</b>	R	0x00	IDE_CRC[15:8]							
0x09B	<b>IDE_CRC_L</b>	R	0x00	IDE_CRC[7:0]							
0x09C			0xXX								
0x09D	<b>IDE_Count_H</b>	R/W	0x00	IDE_Count[23:16]							
0x09E	<b>IDE_Count_M</b>	R/W	0x00	IDE_Count[15:8]							
0x09F	<b>IDE_Count_L</b>	R/W	0x00	IDE_Count[7:1]							



## 2. レジスタ

SLEEP / SNOOZE 時にでも読み書きできるレジスタを**太字斜体**で示します。

それ以外のレジスタは、ACTIVE60 / ACT\_DEVICE / ACT\_HOST / ACT\_ALL時に読み書きできます。  
(各ステートについては、「1.5. パワーマネージメント機能」を参照して下さい。)

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x0A0	IDE_RegAdrs	R/W	0x00	IDE_WrReg	IDE_RdReg			IDE_RegAddress[3:0]			
0x0A1			0xXX								
0x0A2	IDE_RdRegValue_0	R	0x00	IDE_RdRegValue_0[7:0]							
0x0A3	IDE_RdRegValue_1	R	0x00	IDE_RdRegValue_1[7:0]							
0x0A4	IDE_WrRegValue_0	R/W	0x00	IDE_WrRegValue_0[7:0]							
0x0A5	IDE_WrRegValue_1	R/W	0x00	IDE_WrRegValue_1[7:0]							
0x0A6	IDE_SeqWrRegControl	R/W	0x00	IDE_SeqWrReg	IDE_SeqWrRegClr						
0x0A7	IDE_SeqWrRegCnt	R	0x00				IDE_SeqWrRegCnt[4:0]				
0x0A8	IDE_SeqWrRegAdrs	W	0xXX					IDE_SeqRegAddress[3:0]			
0x0A9	IDE_SeqWrRegValue	W	0xXX	IDE_SeqWrRegValue[7:0]							
0x0AA			0xXX								
0x0AB			0xXX								
0x0AC	IDE_RegConfig	R/W	0x00	EnAutoSt s Rd							
0x0AD			0xXX								
0x0AE			0xXX								
0x0AF			0xXX								

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x0B0			0xXX								
0x0B1	<b>HostDeviceSel</b>	R/W	0x00								<b>HOSTxDEVICE</b>
0x0B2			0xXX								
0x0B3	<b>ModeProtect</b>	R/W	0x56	<b>ModeProtect[7:0](56 以外を書くとプロテクト、0x56 で解除)</b>							
0x0B4			0xXX								
0x0B5	<b>ClkSelect</b>	R/W	0x41	<b>xActIDE_Term</b>	<b>xActIDE_DD_Term</b>					<b>PORT1x2</b>	<b>ClkSelect</b>
0x0B6			0xXX								
0x0B7	<b>ChipConfig</b>	R/W	0x00	<b>IntLevel</b>	<b>IntMode</b>	<b>DREQ_Level</b>	<b>DACK_Level</b>	<b>CS_Mode</b>	<b>CPU_Endian</b>	<b>BusMode</b>	<b>Initialized</b>
0x0B8			0xXX								
0x0B9			0xXX								
0x0BA			0xXX								
0x0BB			0xXX								
0x0BC			0xXX								
0x0BD	<b>TimerConfig</b>	R/W	0x00	TimerStop	TimerRunning		FreeRun				
0x0BE	<b>TimerSet_H</b>	R/W	0xFF	TimerSet[15:8]							
0x0BF	<b>TimerSet_L</b>	R/W	0xFF	TimerSet[7:0]							

## 2. レジスタ

SLEEP / SNOOZE 時にでも読み書きできるレジスタを**太字斜体**で示します。

それ以外のレジスタは、ACTIVE60 / ACT\_DEVICE / ACT\_HOST / ACT\_ALL時に読み書きできます。  
(各ステートについては、「1.5. パワーマネージメント機能」を参照して下さい。)

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x0C0	<b>AREAnIntStat</b>	R	0x00			AREA5IntStat	AREA4IntStat	AREA3IntStat	AREA2IntStat	AREA1IntStat	AREA0IntStat
0x0C1	<b>AREA0IntStat</b>	R/(W)	0x01						FIFO_NotEmpty	FIFO_Full	FIFO_Empty
0x0C2	<b>AREA1IntStat</b>	R/(W)	0x01						FIFO_NotEmpty	FIFO_Full	FIFO_Empty
0x0C3	<b>AREA2IntStat</b>	R/(W)	0x01						FIFO_NotEmpty	FIFO_Full	FIFO_Empty
0x0C4	<b>AREA3IntStat</b>	R/(W)	0x01						FIFO_NotEmpty	FIFO_Full	FIFO_Empty
0x0C5	<b>AREA4IntStat</b>	R/(W)	0x01						FIFO_NotEmpty	FIFO_Full	FIFO_Empty
0x0C6	<b>AREA5IntStat</b>	R/(W)	0x01						FIFO_NotEmpty	FIFO_Full	FIFO_Empty
0x0C7											
0x0C8	<b>AREAnIntEnb</b>	R/W	0x00			EnAREA5IntStat	EnAREA4IntStat	EnAREA3IntStat	EnAREA2IntStat	EnAREA1IntStat	EnAREA0IntStat
0x0C9	<b>AREA0IntEnb</b>	R/W	0x00						EnFIFO_NotEmpty	EnFIFO_Full	EnFIFO_Empty
0x0CA	<b>AREA1IntEnb</b>	R/W	0x00						EnFIFO_NotEmpty	EnFIFO_Full	EnFIFO_Empty
0x0CB	<b>AREA2IntEnb</b>	R/W	0x00						EnFIFO_NotEmpty	EnFIFO_Full	EnFIFO_Empty
0x0CC	<b>AREA3IntEnb</b>	R/W	0x00						EnFIFO_NotEmpty	EnFIFO_Full	EnFIFO_Empty
0x0CD	<b>AREA4IntEnb</b>	R/W	0x00						EnFIFO_NotEmpty	EnFIFO_Full	EnFIFO_Empty
0x0CE	<b>AREA5IntEnb</b>	R/W	0x00						EnFIFO_NotEmpty	EnFIFO_Full	EnFIFO_Empty
0x0CF											

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x0D0	<b>AREA0Join_0</b>	R/W	0x00	JoinIDE	JoinFIFO_Stat			JoinDMA1	JoinDMA0	JoinCPU_Rd	JoinCPU_Wr
0x0D1	<b>AREA0Join_1</b>	R/W	0x00			JoinEPeCHe	JoinEPdCHd	JoinEPcCHc	JoinEPbCHb	JoinEPaCHa	JoinEP0CH0
0x0D2	<b>AREA1Join_0</b>	R/W	0x00	JoinIDE	JoinFIFO_Stat			JoinDMA1	JoinDMA0	JoinCPU_Rd	JoinCPU_Wr
0x0D3	<b>AREA1Join_1</b>	R/W	0x00			JoinEPeCHe	JoinEPdCHd	JoinEPcCHc	JoinEPbCHb	JoinEPaCHa	JoinEP0CH0
0x0D4	<b>AREA2Join_0</b>	R/W	0x00	JoinIDE	JoinFIFO_Stat			JoinDMA1	JoinDMA0	JoinCPU_Rd	JoinCPU_Wr
0x0D5	<b>AREA2Join_1</b>	R/W	0x00			JoinEPeCHe	JoinEPdCHd	JoinEPcCHc	JoinEPbCHb	JoinEPaCHa	JoinEP0CH0
0x0D6	<b>AREA3Join_0</b>	R/W	0x00	JoinIDE	JoinFIFO_Stat			JoinDMA1	JoinDMA0	JoinCPU_Rd	JoinCPU_Wr
0x0D7	<b>AREA3Join_1</b>	R/W	0x00			JoinEPeCHe	JoinEPdCHd	JoinEPcCHc	JoinEPbCHb	JoinEPaCHa	JoinEP0CH0
0x0D8	<b>AREA4Join_0</b>	R/W	0x00	JoinIDE	JoinFIFO_Stat			JoinDMA1	JoinDMA0	JoinCPU_Rd	JoinCPU_Wr
0x0D9	<b>AREA4Join_1</b>	R/W	0x00			JoinEPeCHe	JoinEPdCHd	JoinEPcCHc	JoinEPbCHb	JoinEPaCHa	JoinEP0CH0
0x0DA	<b>AREA5Join_0</b>	R/W	0x00	JoinIDE	JoinFIFO_Stat			JoinDMA1	JoinDMA0	JoinCPU_Rd	JoinCPU_Wr
0x0DB	<b>AREA5Join_1</b>	R/W	0x00			JoinEPeCHe	JoinEPdCHd	JoinEPcCHc	JoinEPbCHb	JoinEPaCHa	JoinEP0CH0
0x0DC			0xXX								
0x0DD			0xXX								
0x0DE	<b>ClrAREAnJoin_0</b>	W	0xXX	ClrJoinIDE	ClrJoinFIFO_Stat			ClrJoinDMA1	ClrJoinDMA0	ClrJoinCPU_Rd	ClrJoinCPU_Wr
0x0DF	<b>ClrAREAnJoin_1</b>	W	0xXX			ClrJoinEPeCHe	ClrJoinEPdCHd	ClrJoinEPcCHc	ClrJoinEPbCHb	ClrJoinEPaCHa	ClrJoinEP0CH0

## 2. レジスタ

SLEEP / SNOOZE 時にでも読み書きできるレジスタを**太字斜体**で示します。

それ以外のレジスタは、ACTIVE60 / ACT\_DEVICE / ACT\_HOST / ACT\_ALL時に読み書きできます。  
(各ステートについては、「1.5. パワーマネージメント機能」を参照して下さい。)

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x180	<b>AREA0StartAdrs_H</b>	R/W	0x00				StartAdrs[12:8]				
0x181	<b>AREA0StartAdrs_L</b>	R/W	0x00	StartAdrs[7:2]							
0x182	<b>AREA0EndAdrs_H</b>	R/W	0x00				EndAdrs[12:8]				
0x183	<b>AREA0EndAdrs_L</b>	R/W	0x00	EndAdrs[7:2]							
0x184	<b>AREA1StartAdrs_H</b>	R/W	0x00				StartAdrs[12:8]				
0x185	<b>AREA1StartAdrs_L</b>	R/W	0x00	StartAdrs[7:2]							
0x186	<b>AREA1EndAdrs_H</b>	R/W	0x00				EndAdrs[12:8]				
0x187	<b>AREA1EndAdrs_L</b>	R/W	0x00	EndAdrs[7:2]							
0x188	<b>AREA2StartAdrs_H</b>	R/W	0x00				StartAdrs[12:8]				
0x189	<b>AREA2StartAdrs_L</b>	R/W	0x00	StartAdrs[7:2]							
0x18A	<b>AREA2EndAdrs_H</b>	R/W	0x00				EndAdrs[12:8]				
0x18B	<b>AREA2EndAdrs_L</b>	R/W	0x00	EndAdrs[7:2]							
0x18C	<b>AREA3StartAdrs_H</b>	R/W	0x00				StartAdrs[12:8]				
0x18D	<b>AREA3StartAdrs_L</b>	R/W	0x00	StartAdrs[7:2]							
0x18E	<b>AREA3EndAdrs_H</b>	R/W	0x00				EndAdrs[12:8]				
0x18F	<b>AREA3EndAdrs_L</b>	R/W	0x00	EndAdrs[7:2]							

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x190	<b>AREA4StartAdrs_H</b>	R/W	0x00				StartAdrs[12:8]				
0x191	<b>AREA4StartAdrs_L</b>	R/W	0x00	StartAdrs[7:2]							
0x192	<b>AREA4EndAdrs_H</b>	R/W	0x00				EndAdrs[12:8]				
0x193	<b>AREA4EndAdrs_L</b>	R/W	0x00	EndAdrs[7:2]							
0x194	<b>AREA5StartAdrs_H</b>	R/W	0x00				StartAdrs[12:8]				
0x195	<b>AREA5StartAdrs_L</b>	R/W	0x00	StartAdrs[7:2]							
0x196	<b>AREA5EndAdrs_H</b>	R/W	0x00				EndAdrs[12:8]				
0x197	<b>AREA5EndAdrs_L</b>	R/W	0x00	EndAdrs[7:2]							
			0xXX								
			0xXX								
			0xXX								
			0xXX								
			0xXX								
			0xXX								
			0xXX								
0x19F	<b>AREAnFIFO_Clr</b>	W	0xXX			ClrAREA5	ClrAREA4	ClrAREA3	ClrAREA2	ClrAREA1	ClrAREA0

## 2. レジスタ

### 2.3. デバイス・レジスタマップ

SLEEP / SNOOZE 時にも読み書きできるレジスタを**太字斜体**で示します。

それ以外のレジスタは、ACTIVE60 / ACT\_DEVICE / ACT\_HOST / ACT\_ALL時に読め、ACT\_DEVICE / ACT\_ALL時に書けます。(各ステートについては、「1.5.パワーマネージメント機能」を参照して下さい。)

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x0E0	<b><i>D_SIE_IntStat</i></b>	R/(W)	0x00		<b><i>NonJ</i></b>	RcvSOF	DetectRESET	DetectSUSPEND	ChirpCmp	RestoreCmp	SetAddressCmp
0x0E1			0xXX								
0x0E2	<b><i>D_FIFO_IntStat</i></b>	R/(W)	0x00	DescriptorCmp	FIFO_IDE_Cmp	FIFO1_Cmp	FIFO0_Cmp		FIFO_NotEmpty	FIFO_Full	FIFO_Empty
0x0E3	<b><i>D_BulkIntStat</i></b>	R/(W)	0x00	CBW_Cmp	CBW_LengthErr	CBW_Err		CSW_Cmp	CSW_Err		
0x0E4	<b><i>D_EPrIntStat</i></b>	R	0x00				D_EPeIntStat	D_EPdIntStat	D_EPCIntStat	D_EPbIntStat	D_EPAIntStat
0x0E5	<b><i>D_EPOIntStat</i></b>	R/(W)	0x00		OUT_ShortACK	IN_TranACK	OUT_TranACK	IN_TranNAK	OUT_TranNAK	IN_TranErr	OUT_TranErr
0x0E6	<b><i>D_EPAIntStat</i></b>	R/(W)	0x00		OUT_ShortACK	IN_TranACK	OUT_TranACK	IN_TranNAK	OUT_TranNAK	IN_TranErr	OUT_TranErr
0x0E7	<b><i>D_EPbIntStat</i></b>	R/(W)	0x00		OUT_ShortACK	IN_TranACK	OUT_TranACK	IN_TranNAK	OUT_TranNAK	IN_TranErr	OUT_TranErr
0x0E8	<b><i>D_EPCIntStat</i></b>	R/(W)	0x00		OUT_ShortACK	IN_TranACK	OUT_TranACK	IN_TranNAK	OUT_TranNAK	IN_TranErr	OUT_TranErr
0x0E9	<b><i>D_EPdIntStat</i></b>	R/(W)	0x00		OUT_ShortACK	IN_TranACK	OUT_TranACK	IN_TranNAK	OUT_TranNAK	IN_TranErr	OUT_TranErr
0x0EA	<b><i>D_EPeIntStat</i></b>	R/(W)	0x00		OUT_ShortACK	IN_TranACK	OUT_TranACK	IN_TranNAK	OUT_TranNAK	IN_TranErr	OUT_TranErr
0x0EB			0xXX								
0x0EC			0xXX								
0x0ED			0xXX								
0x0EE			0xXX								
0x0EF			0xXX								

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x0F0	<b><i>D_SIE_IntEnb</i></b>	R/W	0x00		<b><i>EnNonJ</i></b>	EnRcvSOF	EnDetectRESET	EnDetectSUSPEND	EnChirpCmp	EnRestoreCmp	EnSetAddressCmp
0x0F1			0xXX								
0x0F2	<b><i>D_FIFO_IntEnb</i></b>	R/W	0x00	EnDescriptorCmp	EnFIFO_IDE_Cmp	EnFIFO1_Cmp	EnFIFO0_Cmp		EnFIFO_NotEmpty	EnFIFO_Full	EnFIFO_Empty
0x0F3	<b><i>D_BulkIntEnb</i></b>	R/W	0x00	EnCBW_Cmp	EnCBW_LengthErr	EnCBW_Err		EnCSW_Cmp	EnCSW_Err		
0x0F4	<b><i>D_EPrIntEnb</i></b>	R/W	0x00				EnD_EPeIntStat	EnD_EPdIntStat	EnD_EPCIntStat	EnD_EPbIntStat	EnD_EPAIntStat
0x0F5	<b><i>D_EPOIntEnb</i></b>	R/W	0x00		EnOUT_ShortACK	EnIN_TranACK	EnOUT_TranACK	EnIN_TranNAK	EnOUT_TranNAK	EnIN_TranErr	EnOUT_TranErr
0x0F6	<b><i>D_EPAIntEnb</i></b>	R/W	0x00		EnOUT_ShortACK	EnIN_TranACK	EnOUT_TranACK	EnIN_TranNAK	EnOUT_TranNAK	EnIN_TranErr	EnOUT_TranErr
0x0F7	<b><i>D_EPbIntEnb</i></b>	R/W	0x00		EnOUT_ShortACK	EnIN_TranACK	EnOUT_TranACK	EnIN_TranNAK	EnOUT_TranNAK	EnIN_TranErr	EnOUT_TranErr
0x0F8	<b><i>D_EPCIntEnb</i></b>	R/W	0x00		EnOUT_ShortACK	EnIN_TranACK	EnOUT_TranACK	EnIN_TranNAK	EnOUT_TranNAK	EnIN_TranErr	EnOUT_TranErr
0x0F9	<b><i>D_EPdIntEnb</i></b>	R/W	0x00		EnOUT_ShortACK	EnIN_TranACK	EnOUT_TranACK	EnIN_TranNAK	EnOUT_TranNAK	EnIN_TranErr	EnOUT_TranErr
0x0FA	<b><i>D_EPeIntEnb</i></b>	R/W	0x00		EnOUT_ShortACK	EnIN_TranACK	EnOUT_TranACK	EnIN_TranNAK	EnOUT_TranNAK	EnIN_TranErr	EnOUT_TranErr
0x0FB			0xXX								
0x0FC			0xXX								
0x0FD			0xXX								
0x0FE			0xXX								
0x0FF			0xXX								

## 2. レジスタ

SLEEP / SNOOZE 時にでも読み書きできるレジスタを**太字斜体**で示します。

それ以外のレジスタは、ACTIVE60 / ACT\_DEVICE / ACT\_HOST / ACT\_ALL時に読め、ACT\_DEVICE / ACT\_ALL時に書けます。(各ステートについては、「1.5.パワーマネージメント機能」を参照して下さい。)

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x100	<b><i>D_Reset</i></b>	R/W	0x01								<b><i>ResetDTM</i></b>
0x101			0xXX								
0x102	<b><i>D_NegoControl</i></b>	R/W	0x00	DisBusDetect	EnAutoNego	InSUSPEND	DisableHS	SendWakeup	RestoreUSB	GoChirp	ActiveUSB
0x103			0xXX								
0x104			0xXX								
0x105	<b><i>D_XcvrControl</i></b>	R/W	0x41	TermSelect	XcvrSelect					OpMode[1:0]	
0x106	<b><i>D_USB_Test</i></b>	R/W	0x00	EnHS_Test				Test_SE0_NAK	Test_J	Test_K	Test_Packet
0x107			0xXX								
0x108	<b><i>D_EPnControl</i></b>	W	0xXX	AllForceNAK	EPrForceSTALL						
0x109			0xXX								
0x10A	<b><i>D_BulkOnlyControl</i></b>	R/W	0x00	AutoForceNAK_CBW					GoCBW_Mode	GoCSW_Mode	
0x10B	<b><i>D_BulkOnlyConfig</i></b>	R/W	0x00				EPeBulkOnly	EPdBulkOnly	EPcBulkOnly	EPbBulkOnly	EPaBulkOnly
0x10C			0xXX								
0x10D			0xXX								
0x10E			0xXX								
0x10F			0xXX								

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x110	D_EP0SETUP_0	R	0x00	SETUP0[7:0]							
0x111	D_EP0SETUP_1	R	0x00	SETUP1[7:0]							
0x112	D_EP0SETUP_2	R	0x00	SETUP2[7:0]							
0x113	D_EP0SETUP_3	R	0x00	SETUP3[7:0]							
0x114	D_EP0SETUP_4	R	0x00	SETUP4[7:0]							
0x115	D_EP0SETUP_5	R	0x00	SETUP5[7:0]							
0x116	D_EP0SETUP_6	R	0x00	SETUP6[7:0]							
0x117	D_EP0SETUP_7	R	0x00	SETUP7[7:0]							
0x118	D_USB_Address	R/W	0x00	SetAddress	USB_Address[6:0]						
0x119			0xXX								
0x11A	D_SETUP_Control	R/W	0x00								ProtectEP0
0x11B			0xXX								
0x11C			0xXX								
0x11D			0xXX								
0x11E	D_FrameNumber_H	R	0x80	Fn_Invalid					FrameNumber[10:8]		
0x11F	D_FrameNumber_L	R	0x00	FrameNumber[7:0]							

## 2. レジスタ

SLEEP / SNOOZE 時にでも読み書きできるレジスタを**太字斜体**で示します。

それ以外のレジスタは、ACTIVE60 / ACT\_DEVICE / ACT\_HOST / ACT\_ALL時に読め、ACT\_DEVICE / ACT\_ALL時に書けます。(各ステートについては、「1.5.パワーマネージメント機能」を参照して下さい。)

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x120	<b><i>D_EP0MaxSize</i></b>	R/W	0x40	EP0MaxSize[6:3]							
0x121	<b><i>D_EP0Control</i></b>	R/W	0x00	INxOUT							ReplyDescriptor
0x122	<b><i>D_EP0ControlIN</i></b>	R/W	0x00		EnShortPkt		ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL
0x123	<b><i>D_EP0ControlOUT</i></b>	R/W	0x00	AutoForceNAK			ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL
0x124			0xXX								
0x125			0xXX								
0x126			0xXX								
0x127			0xXX								
0x128			0xXX								
0x129			0xXX								
0x12A			0xXX								
0x12B			0xXX								
0x12C			0xXX								
0x12D			0xXX								
0x12E			0xXX								
0x12F			0xXX								

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x130	D_EPaMaxSize_H	R/W	0x00						EPaMaxSize[10:8]		
0x131	D_EPaMaxSize_L	R/W	0x00	EPaMaxSize[7:0]							
0x132	D_EPaConfig_0	R/W	0x00	INxOUT	IntEP_Mode		ISO	EndpointNumber[3:0]			
0x133			0xXX								
0x134	D_EPaControl	R/W	0x00	AutoForceNAK	EnShortPkt	DisAF_NAK_Short	ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL
0x135			0xXX								
0x136			0xXX								
0x137			0xXX								
0x138			0xXX								
0x139			0xXX								
0x13A			0xXX								
0x13B			0xXX								
0x13C			0xXX								
0x13D			0xXX								
0x13E			0xXX								
0x13F			0xXX								

## 2. レジスタ

SLEEP / SNOOZE 時にでも読み書きできるレジスタを**太字斜体**で示します。

それ以外のレジスタは、ACTIVE60 / ACT\_DEVICE / ACT\_HOST / ACT\_ALL時に読み、ACT\_DEVICE / ACT\_ALL時に書けます。(各ステートについては、「1.5.パワーマネージメント機能」を参照して下さい。)

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x140	D_EPbMaxSize_H	R/W	0x00						EPbMaxSize[10:8]		
0x141	D_EPbMaxSize_L	R/W	0x00	EPbMaxSize[7:0]							
0x142	D_EPbConfig_0	R/W	0x00	INxOUT	IntEP_Mode		ISO	EndpointNumber[3:0]			
0x143			0xXX								
0x144	D_EPbControl	R/W	0x00	AutoForceNAK	EnShortPkt	DisAF_NAK_Short	ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL
0x145			0xXX								
0x146			0xXX								
0x147			0xXX								
0x148			0xXX								
0x149			0xXX								
0x14A			0xXX								
0x14B			0xXX								
0x14C			0xXX								
0x14D			0xXX								
0x14E			0xXX								
0x14F			0xXX								

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x150	D_EPcMaxSize_H	R/W	0x00						EPcMaxSize[10:8]		
0x151	D_EPcMaxSize_L	R/W	0x00	EPcMaxSize[7:0]							
0x152	D_EPcConfig_0	R/W	0x00	INxOUT	IntEP_Mode		ISO	EndpointNumber[3:0]			
0x153			0xXX								
0x154	D_EPcControl	R/W	0x00	AutoForceNAK	EnShortPkt	DisAF_NAK_Short	ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL
0x155			0xXX								
0x156			0xXX								
0x157			0xXX								
0x158			0xXX								
0x159			0xXX								
0x15A			0xXX								
0x15B			0xXX								
0x15C			0xXX								
0x15D			0xXX								
0x15E			0xXX								
0x15F			0xXX								

## 2. レジスタ

SLEEP / SNOOZE 時にでも読み書きできるレジスタを**太字斜体**で示します。

それ以外のレジスタは、ACTIVE60 / ACT\_DEVICE / ACT\_HOST / ACT\_ALL時に読め、ACT\_DEVICE / ACT\_ALL時に書けます。(各ステートについては、「1.5.パワーマネージメント機能」を参照して下さい。)

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x160	D_DescAdrs_H	R/W	0x00				DescAdrs[12:8]				
0x161	D_DescAdrs_L	R/W	0x00	DescAdrs[7:0]							
0x162	D_DescSize_H	R/W	0x00							DescSize[9:8]	
0x163	D_DescSize_L	R/W	0x00	DescSize[7:0]							
0x164			0xXX								
0x165			0xXX								
0x166			0xXX								
0x167			0xXX								
0x168			0xXX								
0x169			0xXX								
0x16A			0xXX								
0x16B			0xXX								
0x16C			0xXX								
0x16D			0xXX								
0x16E			0xXX								
0x16F			0xXX								

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x170	<b>D_DMA0_FIFO_Control</b>	R/W	0x00	FIFO_Running	AutoEnShort						
0x171			0xXX								
0x172	<b>D_DMA1_FIFO_Control</b>	R/W	0x00	FIFO_Running	AutoEnShort						
0x173			0xXX								
0x174			0xXX								
0x175			0xXX								
0x176			0xXX								
0x177			0xXX								
0x178			0xXX								
0x179			0xXX								
0x17A			0xXX								
0x17B			0xXX								
0x17C			0xXX								
0x17D			0xXX								
0x17E			0xXX								
0x17F			0xXX								



## 2. レジスタ

SLEEP / SNOOZE 時にでも読み書きできるレジスタを**太字斜体**で示します。

それ以外のレジスタは、ACTIVE60 / ACT\_DEVICE / ACT\_HOST / ACT\_ALL時に読め、ACT\_DEVICE / ACT\_ALL時に書けます。(各ステートについては、「1.5.パワーマネージメント機能」を参照して下さい。)

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x1A0	D_EPdMaxSize_H	R/W	0x00						EPdMaxSize[10:8]		
0x1A1	D_EPdMaxSize_L	R/W	0x00	EPdMaxSize[7:0]							
0x1A2	D_EPdConfig_0	R/W	0x00	INxOUT	IntEP_Mode		ISO	EndpointNumber[3:0]			
0x1A3			0xXX								
0x1A4	D_EPdControl	R/W	0x00	AutoForceNAK	EnShortPkt	DisAF_NAK_Short	ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL
0x1A5			0xXX								
0x1A6			0xXX								
0x1A7			0xXX								
0x1A8			0xXX								
0x1A9			0xXX								
0x1AA			0xXX								
0x1AB			0xXX								
0x1AC			0xXX								
0x1AD			0xXX								
0x1AE			0xXX								
0x1AF			0xXX								

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x1B0	D_EPeMaxSize_H	R/W	0x00						EPeMaxSize[10:8]		
0x1B1	D_EPeMaxSize_L	R/W	0x00	EPeMaxSize[7:0]							
0x1B2	D_EPeConfig_0	R/W	0x00	INxOUT	IntEP_Mode		ISO	EndpointNumber[3:0]			
0x1B3			0xXX								
0x1B4	D_EPeControl	R/W	0x00	AutoForceNAK	EnShortPkt	DisAF_NAK_Short	ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL
0x1B5			0xXX								
0x1B6			0xXX								
0x1B7			0xXX								
0x1B8			0xXX								
0x1B9			0xXX								
0x1BA			0xXX								
0x1BB			0xXX								
0x1BC			0xXX								
0x1BD			0xXX								
0x1BE	DTM_Config	R/W	0x00			DTM_SlopeValue[1:0]				DTM_TermValue[1:0]	
0x1BF			0xXX								

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x1E0	(Reserved)		0xXX								
0x1E1	<b>D_ModeControl</b>	W	0xXX	(Reserved)	(Reserved)	(Reserved)	SetAddressMode	(Reserved)	(Reserved)	(Reserved)	(Reserved)

## 2. レジスタ

### 2.4. ホスト・レジスタマップ

SLEEP / SNOOZE 時にでも読み書きできるレジスタを**太字斜体**で示します。

それ以外のレジスタは、ACTIVE60 / ACT\_DEVICE / ACT\_HOST / ACT\_ALL時に読め、ACT\_HOST / ACT\_ALL時に書けます。(各ステートについては、「1.5.パワーマネージメント機能」を参照して下さい。)

Byte Addr.	Register Name	R/W	Reset	bit15 bit7	bit14 bit6	bit13 bit5	bit12 bit4	bit11 bit3	bit10 bit2	bit9 bit1	bit8 bit0
0x0E0	<b><i>H_SIE_IntStat_0</i></b>	R/(W)	0x00				DetectCon	DetectDiscon	DetectRmtWkup	DetectDevChirpOK	DetectDevChirpNG
0x0E1	<b><i>H_SIE_IntStat_1</i></b>	R/(W)	0x00					DisabledCmp	ResumeCmp	SuspendCmp	ResetCmp
0x0E2	<b><i>H_FIFO_IntStat</i></b>	R/(W)	0x00		FIFO_IDE_Cmp	FIFO1_Cmp	FIFO0_Cmp		FIFO_NotEmpty	FIFO_Full	FIFO_Empty
0x0E3	<b><i>H_FrameIntStat</i></b>	R/(W)	0x00	TriggerFrame					PortErr	FrameNumOver	SOF
0x0E4	<b><i>H_CHrIntStat</i></b>	R	0x00				H_CHeIntStat	H_CHdIntStat	H_CHcIntStat	H_CHbIntStat	H_CHaIntStat
0x0E5	<b><i>H_CH0IntStat</i></b>	R/(W)	0x00	TotalSizeCmp	TranACK	TranErr	ChangeCondition			CTL_SupportCmp	CTL_SupportStop
0x0E6	<b><i>H_CHaIntStat</i></b>	R/(W)	0x00	TotalSizeCmp	TranACK	TranErr	ChangeCondition			BO_SupportCmp	BO_SupportStop
0x0E7	<b><i>H_CHbIntStat</i></b>	R/(W)	0x00	TotalSizeCmp	TranACK	TranErr	ChangeCondition				
0x0E8	<b><i>H_CHcIntStat</i></b>	R/(W)	0x00	TotalSizeCmp	TranACK	TranErr	ChangeCondition				
0x0E9	<b><i>H_CHdIntStat</i></b>	R/(W)	0x00	TotalSizeCmp	TranACK	TranErr	ChangeCondition				
0x0EA	<b><i>H_CHeIntStat</i></b>	R/(W)	0x00	TotalSizeCmp	TranACK	TranErr	ChangeCondition				
0x0EB			0xXX								
0x0EC			0xXX								
0x0ED			0xXX								
0x0EE			0xXX								
0x0EF			0xXX								

Byte Addr.	Register Name	R/W	Reset	bit15 bit7	bit14 bit6	bit13 bit5	bit12 bit4	bit11 bit3	bit10 bit2	bit9 bit1	bit8 bit0
0x0F0	<b><i>H_SIE_IntEnb_0</i></b>	R/W	0x00				EnDetectCon	EnDetectDiscon	EnDetectRmtWkup	EnDetectDevChirpOK	EnDetectDevChirpNG
0x0F1	<b><i>H_SIE_IntEnb_1</i></b>	R/W	0x00					EnDisabledCmp	EnResumeCmp	EnSuspendCmp	EnResetCmp
0x0F2	<b><i>H_FIFO_IntEnb</i></b>	R/W	0x00		EnFIFO_IDE_Cmp	EnFIFO1_Cmp	EnFIFO0_Cmp		EnFIFO_NotEmpty	EnFIFO_Full	EnFIFO_Empty
0x0F3	<b><i>H_FrameIntEnb</i></b>	R/W	0x00	EnTriggerFrame					EnPortErr	EnFrameNumOver	EnSOF
0x0F4	<b><i>H_CHrIntEnb</i></b>	R/W	0x00				EnH_CHeIntStat	EnH_CHdIntStat	EnH_CHcIntStat	EnH_CHbIntStat	EnH_CHaIntStat
0x0F5	<b><i>H_CH0IntEnb</i></b>	R/W	0x00	EnTotalSizeCmp	EnTranACK	EnTranErr	EnChangeCondition			EnCTL_SupportCmp	EnCTL_SupportStop
0x0F6	<b><i>H_CHaIntEnb</i></b>	R/W	0x00	EnTotalSizeCmp	EnTranACK	EnTranErr	EnChangeCondition			EnBO_SupportCmp	EnBO_SupportStop
0x0F7	<b><i>H_CHbIntEnb</i></b>	R/W	0x00	EnTotalSizeCmp	EnTranACK	EnTranErr	EnChangeCondition				
0x0F8	<b><i>H_CHcIntEnb</i></b>	R/W	0x00	EnTotalSizeCmp	EnTranACK	EnTranErr	EnChangeCondition				
0x0F9	<b><i>H_CHdIntEnb</i></b>	R/W	0x00	EnTotalSizeCmp	EnTranACK	EnTranErr	EnChangeCondition				
0x0FA	<b><i>H_CHeIntEnb</i></b>	R/W	0x00	EnTotalSizeCmp	EnTranACK	EnTranErr	EnChangeCondition				
0x0FB			0xXX								
0x0FC			0xXX								
0x0FD			0xXX								
0x0FE			0xXX								
0x0FF			0xXX								

## 2. レジスタ

SLEEP / SNOOZE 時にでも読み書きできるレジスタを**太字斜体**で示します。

それ以外のレジスタは、ACTIVE60 / ACT\_DEVICE / ACT\_HOST / ACT\_ALL時に読め、ACT\_HOST / ACT\_ALL時に書けます。(各ステートについては、「1.5.パワーマネージメント機能」を参照して下さい。)

(注)H\_NegoControl\_1、H\_FrameNumber\_H、H\_FrameNumber\_L の Reset 値は ACT\_HOST 時に読める値です。それ以外のステートでは、Reset 値は 0x00 が読めます。

Byte Addr.	Register Name	R/W	Reset	bit15 bit7	bit14 bit6	bit13 bit5	bit12 bit4	bit11 bit3	bit10 bit2	bit9 bit1	bit8 bit0
0x100	<b>H_Reset</b>	R/W	0x01								<b>ResetHTM</b>
0x101			0xXX								
0x102	H_NegoControl_0	R/W	0x1X	AutoModeCancel	HostState[2:0]			AutoMode[3:0]			
0x103											
0x104	H_NegoControl_1	R/W	0x10			PortSpeed[1:0]				DisChirpFinish	RmtWkupDetEnb
0x105			0xXX								
0x106	H_USB_Test	R/W	0x00	EnHS_Test			Test_Force_Enable	Test_SE0_NAK	Test_J	Test_K	Test_Packet
0x107			0xXX								
0x108			0xXX								
0x109			0xXX								
0x10A			0xXX								
0x10B			0xXX								
0x10C			0xXX								
0x10D			0xXX								
0x10E			0xXX								
0x10F			0xXX								

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x110	H_CH0SETUP_0	R/W	0x00	SETUP_0[7:0]							
0x111	H_CH0SETUP_1	R/W	0x00	SETUP_1[7:0]							
0x112	H_CH0SETUP_2	R/W	0x00	SETUP_2[7:0]							
0x113	H_CH0SETUP_3	R/W	0x00	SETUP_3[7:0]							
0x114	H_CH0SETUP_4	R/W	0x00	SETUP_4[7:0]							
0x115	H_CH0SETUP_5	R/W	0x00	SETUP_5[7:0]							
0x116	H_CH0SETUP_6	R/W	0x00	SETUP_6[7:0]							
0x117	H_CH0SETUP_7	R/W	0x00	SETUP_7[7:0]							
0x118			0xXX								
0x119			0xXX								
0x11A			0xXX								
0x11B			0xXX								
0x11C	(Reserved)	R/W	0xXX								(Reserved)
0x11D			0xXX								
0x11E	H_FrameNumber_H	R	0x07						FrameNumber[10:8]		
0x11F	H_FrameNumber_L	R	0xFF	FrameNumber[7:0]							

## 2. レジスタ

SLEEP / SNOOZE 時にでも読み書きできるレジスタを**太字斜体**で示します。

それ以外のレジスタは、ACTIVE60 / ACT\_DEVICE / ACT\_HOST / ACT\_ALL時に読め、ACT\_HOST / ACT\_ALL時に書けます。(各ステートについては、「1.5.パワーマネージメント機能」を参照して下さい。)

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x120	<u>H_CH0Config_0</u>	R/W	0x00	ACK_Cnt[3:0]				SpeedMode[1:0]		Toggle	TranGo
0x121	<u>H_CH0Config_1</u>	R/W	0x00	TID[1:0]							
0x122			0xXX								
0x123	<u>H_CH0MaxPktSize</u>	R/W	0x00	MaxPktSize[6:0]							
0x124			0xXX								
0x125			0xXX								
0x126	<u>H_CH0TotalSize_H</u>	R/W	0x00	TotalSize[15:8]							
0x127	<u>H_CH0TotalSize_L</u>	R/W	0x00	TotalSize[7:0]							
0x128	<u>H_CH0HubAdrs</u>	R/W	0x00	HubAdrs[3:0]					Port[2:0]		
0x129	<u>H_CH0FuncAdrs</u>	R/W	0x00	FuncAdrs[3:0]				EP_Number[3:0]			
0x12A			0xXX								
0x12B	<u>H_CTL_SupportControl</u>	R/W	0xXX	CTL_SupportState[1:0]							CTL_SupportGo
0x12C			0xXX								
0x12D			0xXX								
0x12E	<u>H_CH0ConditionCode</u>	R	0x00	ConditonCode[2:0]							
0x12F			0xXX								

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x130	<u>H_CHaConfig_0</u>	R/W	0x00	ACK_Cnt[3:0]				SpeedMode[1:0]		Toggle	TranGo
0x131	<u>H_CHaConfig_1</u>	R/W	0x00	TID[1:0]				AutoZeroLen			TotalSizeFree
0x132	<u>H_CHaMaxPktSize_H</u>	R/W	0x00						MaxPktSize[10:8]		
0x133	<u>H_CHaMaxPktSize_L</u>	R/W	0x00	MaxPktSize[7:0]							
0x134	<u>H_CHaTotalSize_HH</u>	R/W	0x00	TotalSize[31:24]							
0x135	<u>H_CHaTotalSize_HL</u>	R/W	0x00	TotalSize[23:16]							
0x136	<u>H_CHaTotalSize_LH</u>	R/W	0x00	TotalSize[15:8]							
0x137	<u>H_CHaTotalSize_LL</u>	R/W	0x00	TotalSize[7:0]							
0x138	<u>H_CHaHubAdrs</u>	R/W	0x00	HubAdrs[3:0]					Port[2:0]		
0x139	<u>H_CHaFuncAdrs</u>	R/W	0x00	FuncAdrs[3:0]				EP_Number[3:0]			
0x13A	<u>H_CHaBO_SupportControl</u>	R/W	0x00			BO_TransportState[1:0]					BO_SupportGo
0x13B	<u>H_CHaCSW_RcvDataSize</u>	R/W	0x00					CSW_RcvDataSize[3:0]			
0x13C	<u>H_CHaBO_OUT_EP_Control</u>	R/W	0x00				OUT_Toggle	OUT_EP_Number[3:0]			
0x13D	<u>H_CHaBO_IN_EP_Control</u>	R/W	0x00				IN_Toggle	IN_EP_Number[3:0]			
0x13E	<u>H_CHaConditionCode</u>	R	0x00		ConditonCode[2:0]						
0x13F			0xXX								

## 2. レジスタ

SLEEP / SNOOZE 時にでも読み書きできるレジスタを**太字斜体**で示します。

それ以外のレジスタは、ACTIVE60 / ACT\_DEVICE / ACT\_HOST / ACT\_ALL時に読め、ACT\_HOST / ACT\_ALL時に書けます。(各ステートについては、「1.5.パワーマネージメント機能」を参照して下さい。)

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x140	<u>H_CHbConfig_0</u>	R/W	0x00	ACK_Cnt[3:0]				SpeedMode[1:0]		Toggle	TranGo
0x141	<u>H_CHbConfig_1</u>	R/W	0x00	TID[1:0]		TranType[1:0]		AutoZeroLen	Audio441	TotalSizeFree[1:0]	
0x142	<u>H_CHbMaxPktSize_H</u>	R/W	0x00						MaxPktSize[10:8]		
0x143	<u>H_CHbMaxPktSize_L</u>	R/W	0x00	MaxPktSize[7:0]							
0x144	<u>H_CHbTotalSize_HH</u>	R/W	0x00	TotalSize[31:24]							
0x145	<u>H_CHbTotalSize_HL</u>	R/W	0x00	TotalSize[23:16]							
0x146	<u>H_CHbTotalSize_LH</u>	R/W	0x00	TotalSize[15:8]							
0x147	<u>H_CHbTotalSize_LL</u>	R/W	0x00	TotalSize[7:0]							
0x148	<u>H_CHbHubAdrs</u>	R/W	0x00	HubAdrs[3:0]					Port[2:0]		
0x149	<u>H_CHbFuncAdrs</u>	R/W	0x00	FuncAdrs[3:0]				EP_Number[3:0]			
0x14A	<u>H_CHbInterval_H</u>	R/W	0x00						Interval[10:8]		
0x14B	<u>H_CHbInterval_L</u>	R/W	0x00	Interval[7:0]							
0x14C	<u>H_CHbTranPause</u>	R/W	0xXX							EnTranPause	TranPause
0x14D			0xXX								
0x14E	<u>H_CHbConditionCode</u>	R	0x00		ConditonCode[2:0]						
0x14F			0xXX								

Byte Addr.	Register Name	R/W	Reset	bit15 bit7	bit14 bit6	bit13 bit5	bit12 bit4	bit11 bit3	bit10 bit2	bit9 bit1	bit8 bit0
0x150	<u>H_CHcConfig_0</u>	R/W	0x00	ACK_Cnt[3:0]				SpeedMode[1:0]		Toggle	TranGo
0x151	<u>H_CHcConfig_1</u>	R/W	0x00	TID[1:0]		TranType[1:0]		AutoZeroLen	Audio441	TotalSizeFree[1:0]	
0x152	<u>H_CHcMaxPktSize_H</u>	R/W	0x00						MaxPktSize[10:8]		
0x153	<u>H_CHcMaxPktSize_L</u>	R/W	0x00	MaxPktSize[7:0]							
0x154	<u>H_CHcTotalSize_HH</u>	R/W	0x00	TotalSize[31:24]							
0x155	<u>H_CHcTotalSize_HL</u>	R/W	0x00	TotalSize[23:16]							
0x156	<u>H_CHcTotalSize_LH</u>	R/W	0x00	TotalSize[15:8]							
0x157	<u>H_CHcTotalSize_LL</u>	R/W	0x00	TotalSize[7:0]							
0x158	<u>H_CHcHubAdrs</u>	R/W	0x00	HubAdrs[3:0]						Port[2:0]	
0x159	<u>H_CHcFuncAdrs</u>	R/W	0x00	FuncAdrs[3:0]				EP_Number[3:0]			
0x15A	<u>H_CHcInterval_H</u>	R/W	0x00						Interval[10:8]		
0x15B	<u>H_CHcInterval_L</u>	R/W	0x00	Interval[7:0]							
0x15C	<u>H_CHcTranPause</u>	R/W	0x00							EnTranPause	TranPause
0x15D			0xFF								
0x15E	<u>H_CHcConditionCode</u>	R	0x00	ConditonCode[2:0]							
0x15F			0xFF								

## 2. レジスタ

SLEEP / SNOOZE 時にでも読み書きできるレジスタを**太字斜体**で示します。

それ以外のレジスタは、ACTIVE60 / ACT\_DEVICE / ACT\_HOST / ACT\_ALL時に読め、ACT\_HOST / ACT\_ALL時に書けます。(各ステートについては、「1.5.パワーマネージメント機能」を参照して下さい。)

Byte Addr.	Register Name	R/W	Reset	bit15 bit7	bit14 bit6	bit13 bit5	bit12 bit4	bit11 bit3	bit10 bit2	bit9 bit1	bit8 bit0
0x160	<u>H_CHdConfig_0</u>	R/W	0x00	ACK_Cnt[3:0]				SpeedMode[1:0]		Toggle	TranGo
0x161	<u>H_CHdConfig_1</u>	R/W	0x00	TID[1:0]		TranType[1:0]		AutoZeroLen	Audio441	TotalSizeFree[1:0]	
0x162	<u>H_CHdMaxPktSize_H</u>	R/W	0x00						MaxPktSize[10:8]		
0x163	<u>H_CHdMaxPktSize_L</u>	R/W	0x00	MaxPktSize[7:0]							
0x164	<u>H_CHdTotalSize_HH</u>	R/W	0x00	TotalSize[31:24]							
0x165	<u>H_CHdTotalSize_HL</u>	R/W	0x00	TotalSize[23:16]							
0x166	<u>H_CHdTotalSize_LH</u>	R/W	0x00	TotalSize[15:8]							
0x167	<u>H_CHdTotalSize_LL</u>	R/W	0x00	TotalSize[7:0]							
0x168	<u>H_CHdHubAdrs</u>	R/W	0x00	HubAdrs[3:0]					Port[2:0]		
0x169	<u>H_CHdFuncAdrs</u>	R/W	0x00	FuncAdrs[3:0]				EP_Number[3:0]			
0x16A	<u>H_CHdInterval_H</u>	R/W	0x00						Interval[10:8]		
0x16B	<u>H_CHdbnterval_L</u>	R/W	0x00	Interval[7:0]							
0x16C	<u>H_CHdTranPause</u>	R/W	0x00							EnTranPause	TranPause
0x16D			0xFF								
0x16E	<u>H_CHdConditionCode</u>	R	0x00	ConditonCode[2:0]							
0x16F			0xFF								

Byte Addr.	Register Name	R/W	Reset	bit15 bit7	bit14 bit6	bit13 bit5	bit12 bit4	bit11 bit3	bit10 bit2	bit9 bit1	bit8 bit0
0x170	<u>H_CHeConfig_0</u>	R/W	0x00	ACK_Cnt[3:0]				SpeedMode[1:0]		Toggle	TranGo
0x171	<u>H_CHeConfig_1</u>	R/W	0x00	TID[1:0]		TranType[1:0]		AutoZeroLen	Audio441	TotalSizeFree[1:0]	
0x172	<u>H_CHeMaxPktSize_H</u>	R/W	0x00						MaxPktSize[10:8]		
0x173	<u>H_CHeMaxPktSize_L</u>	R/W	0x00	MaxPktSize[7:0]							
0x174	<u>H_CHeTotalSize_HH</u>	R/W	0x00	TotalSize[31:24]							
0x175	<u>H_CHeTotalSize_HL</u>	R/W	0x00	TotalSize[23:16]							
0x176	<u>H_CHeTotalSize_LH</u>	R/W	0x00	TotalSize[15:8]							
0x177	<u>H_CHeTotalSize_LL</u>	R/W	0x00	TotalSize[7:0]							
0x178	<u>H_CHeHubAdrs</u>	R/W	0x00	HubAdrs[3:0]						Port[2:0]	
0x179	<u>H_CHeFuncAdrs</u>	R/W	0x00	FuncAdrs[3:0]				EP_Number[3:0]			
0x17A	<u>H_CHeInterval_H</u>	R/W	0x00						Interval[10:8]		
0x17B	<u>H_CHeInterval_L</u>	R/W	0x00	Interval[7:0]							
0x17C	<u>H_CHeTranPause</u>	R/W	0xXX							EnTranPause	TranPause
0x17D			0xFF								
0x17E	<u>H_CHeConditionCode</u>	R	0x00	ConditonCode[2:0]							
0x17F			0xFF								

## 2. レジスタ

SLEEP / SNOOZE 時にでも読み書きできるレジスタを**太字斜体**で示します。  
それ以外のレジスタは、ACTIVE60 / ACT\_DEVICE / ACT\_HOST / ACT\_ALL時に読め、ACT\_HOST / ACT\_ALL時に書けます。(各ステートについては、「1.5.パワーマネージメント機能」を参照して下さい。)

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x1B0	H_TriggerFrameNum_H	R/W	0x00	UseTriggerFrame					TriggerFrameNum[10:8]		
0x1B1	H_TriggerFrameNum_L	R/W	0x00	TriggerFrameNum[7:0]							
0x1B2			0xXX								
0x1B3			0xXX								
0x1B4			0xXX								
0x1B5			0xXX								
0x1B6			0xXX								
0x1B7			0xXX								
0x1B8			0xXX								
0x1B9			0xXX								
0x1BA			0xXX								
0x1BB			0xXX								
0x1BC			0xXX								
0x1BD			0xXX								
0x1BE	<u>HTM_Config</u>	R/W	0x00	HTM_SlopeValue[1:0]					HTM_TermValue[1:0]		
0x1BF			0xXX								

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x1F4	(Reserved)		0xXX	(Reserved)	(Reserved)	(Reserved)	(Reserved)	(Reserved)	(Reserved)	(Reserved)	(Reserved)
0x1F5	<b>H_Protect</b>	R/W	0x00	(Reserved)	(Reserved)	(Reserved)	(Reserved)	PortSpeedWrEnb	(Reserved)	(Reserved)	(Reserved)

## 2. レジスタ

### 2.5. 初期レジスタ 詳細説明

#### 2.5.1. xxxh CPUIF\_MODE (CPU I/F Mode)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
uninitialized	xxxh	CPUIF_MODE		15,7:	0:	1:	xxh
				14,6:	0:	1:	
				13,5:	0:	1:	
				12,4:	0:	1:	
				11,3:	0:	1:	
			W	10,2: <b>CPU_Endain</b>	0: BigEndian Mode	1: LittleEndian Mode	
			W	9,1: <b>BusMode</b>	0: Strobe Mode	1: BE Mode	
				8,0:	0:	1:	

未初期化期間に有効なレジスタです。ご使用のCPUにあわせて設定を行って下さい。詳細は「1.8.1.CPUIFモード」及び「1.8.2.CPUIFのモード設定」をご参照ください。

上位バイト／下位バイトに同じものがミラーで存在するものとしてアクセスして下さい。

#### Bit15-11, Bit7-3    **Reserved**

#### Bit10, Bit2        **CPU\_Endian**

CPUIF のエンディアンを設定します。

0 : BigEndian モード

1 : LittleEndian モード

初期化済み期間において、このビットの設定は ChipConfig.CPU\_Endian ビットで確認できます。

#### Bit9, Bit1         **BusMode**

CPUIF の Bus モードを設定します。

0 : Strobe モード

1 : BE モード

初期化済み期間において、このビットの設定は ChipConfig.BusMode ビットで確認できます。

#### Bit8, Bit0         **Reserved**



## 2.6. デバイス／ホストレジスタ詳細説明

2.6.1. 000h *MainIntStat* (Main Interrupt Status)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	000h	<i>MainIntStat</i>	R	7: <i>DeviceIntStat</i>	0: None	1: Device Interrupts	00h
			R	6: <i>HostIntStat</i>	0: None	1: Host Interrupts	
			R	5: CPU_IntStat	0: None	1: CPU Interrupts	
			R	4: IDE_IntStat	0: None	1: IDE Interrupts	
			R	3: MediaFIFO_IntStat	0: None	1: MediaFIFO Interrupts	
			R	2: AREAnIntStat	0: None	1: AREA Interrupts	
				1:	0:	1:	
			R (W)	0: <i>FinishedPM</i>	0: None	1: Detect FinishedPM	

本 LSI の割り込み要因を表示します。

このレジスタには割り込み要因を間接指示するビットと直接指示するビットがあります。割り込み要因を間接指示するビットは、それぞれに対応する割り込みステータスレジスタをリードすることにより、割り込み要因を直接指示するビットまで辿ることができます。割り込み要因を間接指示するビットは、リードオンリーであり、大元の割り込み要因を直接指示するビットをクリアすることにより、自動的にクリアされます。割り込み要因を直接指示しているビットは、書き込み可能であり、該当ビットに"1"を書き込むことにより、割り込み要因をクリアすることができます。MainIntEnb レジスタにより、対応するビットの割り込みがイネーブルにされている場合は、割り込み要因が"1"にセットされると XINT 端子がアサートされ、CPU に対して割り込みが発生します。該当する割り込み要因が全てクリアされると、XINT 端子がネゲートされます。

**Bit7 DeviceIntStat**

割り込み要因を間接指示します。

DeviceIntStat レジスタに割り込み要因があり、かつその割り込み要因に対応する DeviceIntEnb レジスタのビットがイネーブルにされている時に"1"にセットされます。このビットは、SLEEP / SNOOZE 中もリード有効です。

**Bit6 HostIntStat**

割り込み要因を間接指示します。

HostIntStat レジスタに割り込み要因があり、かつその割り込み要因に対応する HostIntEnb レジスタのビットがイネーブルにされている時に"1"にセットされます。このビットは、SLEEP / SNOOZE 中もリード有効です。

**Bit5 CPU\_IntStat**

割り込み要因を間接指示します。

CPU\_IntStat レジスタに割り込み要因があり、かつその割り込み要因に対応する CPU\_IntEnb レジスタのビットがイネーブルにされている時に"1"にセットされます。

**Bit4 IDE\_IntStat**

割り込み要因を間接指示します。

IDE\_IntStat レジスタに割り込み要因があり、かつその割り込み要因に対応する IDE\_IntEnb レジスタのビットがイネーブルにされている時に"1"にセットされます。

**Bit3 MediaFIFO\_IntStat**

割り込み要因を間接指示します。

MediaFIFO\_IntStat レジスタに割り込み要因があり、かつその割り込み要因に対応する MediaFIFO\_IntEnb レジスタのビットがイネーブルにされている時に"1"にセットされます。

**Bit2 AREAnIntStat**

割り込み要因を間接指示します。

AREAnIntStat レジスタに割り込み要因があり、かつその割り込み要因に対応する AREAnIntEnb レジスタのビットがイネーブルにされている時に"1"にセットされます。

**Bit1 Reserved****Bit0 FinishedPM**

## 2. レジスタ

---

割り込み要因を直接指示します。

PM\_Control\_0 レジスタで GoSLEEP, GoSNOOZE, GoActive60, GoActDevice, GoActHost, GoActAllDev, GoActAllHost を設定した場合、指示したそれぞれの状態に達したら、このビットは"1"にセットされます。このビットは SLEEP / SNOOZE 中も有効です。

## 2.6.2. 001h DeviceIntStat (Device Interrupt Status)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	001h	<b>DeviceIntStat</b>	R (W)	7: <b>VBUS_Changed</b>	0: None	1: VBUS is Changed	00h
				6:	0:	1:	
			R	5: <b>D_SIE_IntStat</b>	0: None	1: SIE Interrupts	
			R	4: <b>D_BulkIntStat</b>	0: None	1: Bulk Interrupts	
			R (W)	3: <b>RcvEP0SETUP</b>	0: None	1: Receive EP0 SETUP	
			R	2: <b>D_FIFO_IntStat</b>	0: None	1: FIFO Interrupts	
			R	1: <b>D_EP0IntStat</b>	0: None	1: EP0 Interrupts	
			R	0: <b>D_EPrIntStat</b>	0: None	1: EPr Interrupts	

デバイス関連の割り込みを表示します。

このレジスタには割り込み要因を間接指示するビットと直接指示するビットがあります。割り込み要因を間接指示するビットは、それぞれに対応する割り込みステータスレジスタをリードすることにより、割り込み要因を直接指示するビットまで辿ることができます。割り込み要因を間接指示するビットは、リードオンリーであり、大元の割り込み要因を直接指示するビットをクリアすることにより、自動的にクリアされます。割り込み要因を直接指示しているビットは、書き込み可能であり、該当ビットに"1"を書き込むことにより、割り込み要因をクリアすることができます。

**Bit7 VBUS\_Changed**

割り込み要因を直接指示します。

VBUS\_B 端子の状態が変化した時に"1"にセットされます。

D\_USB\_Status レジスタの VBUS ビットによって VBUS\_B 端子の状態を確認して下さい。VBUS が"0"であれば、ケーブルが抜かれたことを示します。このビットは SLEEP / SNOOZE 中も有効です。

**Bit6 Reserved****Bit5 D\_SIE\_IntStat**

割り込み要因を間接指示します。

D\_SIE\_IntStat レジスタに割り込み要因があり、かつその割り込み要因に対応する D\_SIE\_IntEnb レジスタのビットがイネーブルにされている時に"1"にセットされます。このビットは、SLEEP / SNOOZE 中もリード有効です。

**Bit4 D\_BulkIntStat**

割り込み要因を間接指示します。

D\_BulkIntStat レジスタに割り込み要因があり、かつその割り込み要因に対応する D\_BulkIntEnb レジスタのビットがイネーブルにされている時に"1"にセットされます。

**Bit3 RcvEP0SETUP**

割り込み要因を直接指示します。

コントロール転送のセットアップステージが終了し、受信したデータが D\_EP0SETUP\_0～D\_EP0SETUP\_7 レジスタに格納されると"1"にセットされます。同時に D\_EP0ControlIN, D\_EP0ControlOUT レジスタの ForceSTALL ビットが"0"に D\_EP0ControlIN, D\_EP0ControlOUT レジスタの ForceNAK ビット、ToggleStat ビット、D\_SETUP\_Control レジスタの ProtectEP0 ビットが"1"に、自動的に設定されます。SetAddress() リクエストに対しては、AutoSetAddress 機能が自動応答し、このステータスはセットされません。

**Bit2 D\_FIFO\_IntStat**

割り込み要因を間接指示します。

D\_FIFO\_IntStat レジスタに割り込み要因があり、かつその割り込み要因に対応する D\_FIFO\_IntEnb レジスタのビットがイネーブルにされている時に"1"にセットされます。

**Bit1 D\_EP0IntStat**

## 2. レジスタ

---

割り込み要因を間接指示します。

D\_EP0IntStat レジスタに割り込み要因があり、かつその割り込み要因に対応する  
D\_EP0IntEnb レジスタのビットがイネーブルにされている時に"1"にセットされます。

### Bit0 D\_EPrIntStat

割り込み要因を間接指示します。

D\_EPrIntStat レジスタに割り込み要因があり、かつその割り込み要因に対応する  
D\_EPaIntEnb レジスタのビットがイネーブルにされている時、"1"にセットされます。

## 2.6.3. 002h HostIntStat (Host Interrupt Status)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device  / Host	002h	<b>HostIntStat</b>	R/(W )	7: <b>VBUS_Err</b>	0: None 1: VBUS Error	00h
			R/(W )	6: <b>LineStatusChanged</b>	0: None 1: Line Status Changed	
			R	5: H_SIE_IntStat1	0: None 1: SIE Interrupts1	
			R	4: H_SIE_IntStat0	0: None 1: SIE Interrupts0	
			R	3: H_FrameIntStat	0: None 1: Frame Interrupts	
			R	2: H_FIFOIntStat	0: None 1: FIFO Interrupts	
			R	1: H_CH0IntStat	0: None 1: CH0 Interrupts	
			R	0: H_CHrIntStat	0: None 1: CHr Interrupts	

ホスト関連の割り込みを表示します。

このレジスタには割り込み要因を間接指示するビットと直接指示するビットがあります。割り込み要因を間接指示するビットは、それぞれに対応する割り込みステータスレジスタをリードすることにより、割り込み要因を直接指示するビットまで辿ることができます。割り込み要因を間接指示するビットは、リードオンリーであり、大元の割り込み要因を直接指示するビットをクリアすることにより、自動的にクリアされます。割り込み要因を直接指示しているビットは、書き込み可能であり、該当ビットに"1"を書き込むことにより、割り込み要因をクリアすることができます。

**Bit7 VBUS\_Err**

割り込み要因を直接指示します。このビットは SLEEP / SNOOZE 中も有効です。

VBUSFLG\_A 端子に外部接続の VBUS パワースイッチから VBUS 異常を知らせる信号(High から Low への変化 Edge)が入力された時に"1"にセットされます。

H\_USB\_Status レジスタの VBUS\_State ビットによって VBUSFLG\_A 端子の状態を確認して下さい。

上記の異常信号とは、外部接続の VBUS パワースイッチの仕様によって異なりますので、その仕様をご確認下さい。

**Bit6 LineStateChanged**

割り込み要因を直接指示します。このビットは SLEEP / SNOOZE 中も有効です。

ホストポートの DP 端子および DM 端子の状態が変化したことを示します。

この割り込みは、USB ホスト機能を使用していないとき、あるいは USB ホスト機能を使用していて SLEEP / SNOOZE を行ったときのホストポートの信号ライン変化を検出するために使用します。

**Bit5 H\_SIE\_IntStat1**

割り込み要因を間接指示します。

H\_SIE\_IntStat1 レジスタに割り込み要因があり、かつその割り込み要因に対応する H\_SIE\_IntEnb1 レジスタのビットがイネーブルにされている時に"1"にセットされます。

**Bit4 H\_SIE\_IntStat0**

割り込み要因を間接指示します。

H\_SIE\_IntStat0 レジスタに割り込み要因があり、かつその割り込み要因に対応する H\_SIE\_IntEnb0 レジスタのビットがイネーブルにされている時に"1"にセットされます。

**Bit3 H\_FrameIntStat**

割り込み要因を間接指示します。

H\_FrameIntStat レジスタに割り込み要因があり、かつその割り込み要因に対応する H\_FrameIntEnb レジスタのビットがイネーブルにされている時に"1"にセットされます。

**Bit2 H\_FIFO\_IntStat**

割り込み要因を間接指示します。

## 2. レジスタ

---

H\_FIFO\_IntStat レジスタに割り込み要因があり、かつその割り込み要因に対応する  
H\_FIFO\_IntEnb レジスタのビットがイネーブルにされている時に"1"にセットされます。

**Bit1 H\_CH0\_IntStat**

割り込み要因を間接指示します。

H\_CH0\_IntStat レジスタに割り込み要因があり、かつその割り込み要因に対応する  
H\_CH0\_IntEnb レジスタのビットがイネーブルにされている時に"1"にセットされます。

**Bit0 H\_CHr\_IntStat**

割り込み要因を間接指示します。

H\_CHr\_IntStat レジスタに割り込み要因があり、かつその割り込み要因に対応する  
H\_CHr\_IntEnb レジスタのビットがイネーブルにされている時に"1"にセットされます。

## 2.6.4. 003h CPU\_IntStat (CPU Interrupt Status)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	003h	CPU_IntStat	R (W)	7: RAM_RdCmp	0: None	1: RAM Read Complete	00h
			R (W)	6: C_TimerCmp	0: None	1: Timer Interrupts	
				5:	0:	1:	
				4:	0:	1:	
			R (W)	3: DMA1_Countup	0: None	1: DMA1 Counter Overflow	
			R (W)	2: DMA1_Cmp	0: None	1: DMA1 Complete	
			R (W)	1: DMA0_CountUp	0: None	1: DMA0 Counter Overflow	
			R (W)	0: DMA0_Cmp	0: None	1: DMA0 Complete	

CPU インタフェイス関連の割り込みを表示します。  
 全てのビットは"1"を書き込む事で割り込み要因をクリアできます。

**Bit7 RAM\_RdCmp**

割り込み要因を直接指示します。

RAM\_Rd 機能において、RAM からデータを読み出し、RAM\_Rd\_XX のデータが有効になったところで、"1"にセットされます。

**Bit6 C\_TimerCmp**

割り込み要因を直接指示します。

TimerSet\_H,L に設定した値からタイマの値がオーバーフローした時に、"1"にセットされます。

**Bit5-4 Reserved****Bit3 DMA1\_CountUp**

割り込み要因を直接指示します。

転送モードがフリーランモードで動作している時に、DMA1\_Count\_HH,HL,LH,LL の値がオーバーフローした時に、"1"にセットされます。DMA1\_Count\_HH,HL,LH,LL の値は 0 に戻り、DMA 動作は継続します。

**Bit2 DMA1\_Cmp**

割り込み要因を直接指示します。

DMA 転送が停止されるか、或いは、指定された転送数が終了し、終了処理が完了した時に"1"にセットされます。

**Bit1 DMA0\_CountUp**

割り込み要因を直接指示します。

転送モードがフリーランモードで動作している時に、DMA0\_Count\_HH,HL,LH,LL の値がオーバーフローした時に、"1"にセットされます。DMA0\_Count\_HH,HL,LH,LL の値は 0 に戻り、DMA 動作は継続します。

**Bit0 DMA0\_Cmp**

割り込み要因を直接指示します。

DMA 転送が停止されるか、或いは、指定された転送数が終了し、終了処理が完了した時に"1"にセットされます。

## 2. レジスタ

### 2.6.5. 004h IDE\_IntStat (IDE Interrupt Status)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	004h	IDE_IntStat	R (W)	7: IDE_RegCmp	0: None	1: Register Access Complete	00h
			R (W)	6: IDE_RegErr	0: None	1: Register Access Error	
			R (W)	5: IDE_SeqWrRegCmp	0: None	1: Sequence Write Complete	
			R (W)	4: CompleteINTRQ	0: None	1: Auto Status Read Compl.	
				3:	0:	1:	
			R (W)	2: IDE_Cmp	0: None	1: DMA Complete	
			R (W)	1: DetectINTRQ	0: None	1: Detected Interrupt	
			R (W)	0: DetectTerm	0: None	1: Detected Device terminate	

IDE 関連の割り込みステータスを表示します。

全てのビットは"1"を書き込む事で割り込み要因をクリアできます。

#### Bit7 IDE\_RegCmp

割り込み要因を直接指示します。

IDE\_RegAdrs レジスタによる IDE レジスタへのリードライトアクセスが完了した時に"1"にセットされます。

#### Bit6 IDE\_RegErr

割り込み要因を直接指示します。

以下の場合に"1"にセットされます。

- 1) IDE\_RegConfig レジスタによる自動ステータスリードの動作シーケンス中に  
IDE\_RegAdrs レジスタによる IDE レジスタへのリードライトアクセスを行った。
- 2) IDE\_SeqWrRegControl レジスタによるシーケンスライトの動作シーケンス中に  
IDE\_RegAdrs レジスタによる IDE レジスタへのリードライトアクセスを行った。

#### Bit5 IDE\_SeqWrRegCmp

割り込み要因を直接指示します。

IDE\_SeqWrRegControl レジスタによるシーケンスライトの動作が完了した時に"1"にセットされます。

#### Bit4 CompleteINTRQ

割り込み要因を直接指示します。

IDE\_RegConfig レジスタによる自動ステータスリードの動作が完了した時に"1"にセットされます。

#### Bit3 Reserved

#### Bit2 IDE\_Cmp

割り込み要因を直接指示します。

IDE\_Control レジスタによる DMA 動作が完了した時に"1"にセットされます。

#### Bit1 DetectINTRQ

割り込み要因を直接指示します。

IDE\_RegConfig レジスタの EnAutoStsRd ビットがセットされていないとき、IDE の HINTRQ 信号の立ち上がり

検出された時に"1"にセットされます。

#### Bit0 DetectTerm

割り込み要因を直接指示します。

IDE\_Control レジスタによる IDE の DMA 動作中で転送モードが Ultra-DMA の場合、 デバイスターミネートを

検出した時に"1"にセットされます。



## 2.6.6. 005h MediaFIFO\_IntStat (Media FIFO Interrupt Status)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	005h	MediaFIFO_IntStat		7:	0:	1:	00h
			R (W)	6: MediaIDE_Cmp	0: None	1: Media FIFO IDE Complete	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
			R (W)	2: FIFO_NotEmpty	0: None	1: FIFO NotEmpty	
			R (W)	1: FIFO_Full	0: None	1: FIFO Full	
			R (W)	0: FIFO_Empty	0: None	1: FIFO Empty	

Media FIFO 関連の割り込みステータスを表示します。  
 全てのビットは"1"を書き込む事で割り込み要因をクリアできます。

**Bit7**      **Reserved**

**Bit6**      **MediaIDE\_Cmp**

割り込み要因を直接指示します。

IDE が Join され Dir が"0"の場合、IDE の転送が終了した後に MediaFIFO が空になると、"1"にセットされます。Dir が"1"の場合、IDE の転送が終了すると、"1"にセットされます。

**Bit5-3**    **Reserved**

**Bit2**      **FIFO\_NotEmpty**

割り込み要因を直接指示します。

Media FIFO にデータが存在する状態(NotEmpty)になると"1"にセットされます。

**Bit1**      **FIFO\_Full**

割り込み要因を直接指示します。

Media FIFO が Full 時に"1"にセットされます。

**Bit0**      **FIFO\_Empty**

割り込み要因を直接指示します。

Media FIFO が Empty 時に"1"にセットされます。

## 2. レジスタ

---

### 2.6.7. 010h *MainIntEnb* (Main Interrupt Enable)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	010h	<b>MainIntEnb</b>	R / W	7: <b>EnDeviceIntStat</b>	0: Disable	1: Enable	00h
			R / W	6: <b>EnHostIntStat</b>	0: Disable	1: Enable	
			R / W	5: EnCPU_IntStat	0: Disable	1: Enable	
			R / W	4: En IDE_IntStat	0: Disable	1: Enable	
			R / W	3: EnMediaFIFO_IntStat	0: Disable	1: Enable	
			R / W	2: EnAREAnIntStat	0: Disable	1: Enable	
				1:	0:	1	
			R / W	0: <b>EnFinishedPM</b>	0: Disable	1: Enable	

**MainIntStat** レジスタの割り込み要因による割り込み信号(XINT)のアサートを許可／禁止するレジスタです。

対応するビットを"1"にセットすることで割り込みを許可します。

EnDeviceIntStat, EnHostIntStat, EnFinishedPM ビットは SLEEP / SNOOZE 中も有効です。

2.6.8. 011h *DeviceIntEnb (Device Interrupt Enable)*

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	011h	<b><i>DeviceIntEnb</i></b>	R / W	7: <b><i>EnVBUS_Changed</i></b>	0: Disable	1: Enable	00h
				6:	0:	1:	
			R / W	5: <b><i>EnD_SIE_IntStat</i></b>	0: Disable	1: Enable	
			R / W	4: <i>EnD_BulkIntStat</i>	0: Disable	1: Enable	
			R / W	3: <i>EnRcvEP0SETUP</i>	0: Disable	1: Enable	
			R / W	2: <i>EnD_FIFO_IntStat</i>	0: Disable	1: Enable	
			R / W	1: <i>EnD_EP0IntStat</i>	0: Disable	1: Enable	
			R / W	0: <i>EnD_EP1IntStat</i>	0: Disable	1: Enable	

DeviceIntStat レジスタの割り込み要因による、MainIntStat レジスタの DeviceIntStat ビットのアサート  
を許可／禁止します。

EnVBUS\_Changed, EnD\_SIE\_IntStat ビットは SLEEP / SNOOZE 中も有効です。

## 2. レジスタ

---

### 2.6.9. 012h *HostIntEnb (Host Interrupt Enable)*

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	012h	<b>HostIntEnb</b>	R / W	7: <b>EnVBUS_Err</b>	0: Disable	1: Enable	00h
			R / W	6: <b>EnLineStatusChanged</b>	0: Disable	1: Enable	
			R / W	5: EnH_SIE_IntStat1	0: Disable	1: Enable	
			R / W	4: EnH_SIE_IntStat0	0: Disable	1: Enable	
			R / W	3: EnH_FrameIntStat	0: Disable	1: Enable	
			R / W	2: EnH_FIFOIntStat	0: Disable	1: Enable	
			R / W	1: EnH_CH0IntStat	0: Disable	1: Enable	
			R / W	0: EnH_CHrIntStat	0: Disable	1: Enable	

HostIntStat レジスタの割り込み要因による、MainIntStat レジスタの HostIntStat ビットのアサートを許可／禁止します。

EnVBUS\_Err, EnLineStatusChanged ビットは SLEEP / SNOOZE 中も有効です。

## 2.6.10. 013h CPU\_IntEnb (CPU Interrupt Enable)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	013h	CPU_IntEnb	R / W	7: EnRAM_RdCmp	0: Disable	1: Enable	00h
			R / W	6: EnC_TimerCmp	0: Disable	1: Enable	
				5:	0:	1:	
				4:	0:	1:	
			R / W	3: EnDMA1_Countup	0: Disable	1: Enable	
			R / W	2: EnDMA1_Cmp	0: Disable	1: Enable	
			R / W	1: EnDMA0_CountUp	0: Disable	1: Enable	
			R / W	0: EnDMA0_Cmp	0: Disable	1: Enable	

CPU\_IntStat レジスタの割り込み要因による、MainIntStat レジスタの CPU\_IntStat ビットのアサートを許可／禁止します。

## 2. レジスタ

---

### 2.6.11. 014h IDE\_IntEnb (IDE Interrupt Enable)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	014h	IDE_IntEnb	R / W	7: EnIDE_RegCmp	0: Disable	1: Enable	00h
			R / W	6: EnIDE_RegErr	0: Disable	1: Enable	
			R / W	5: En_SeqWrRegCmp	0: Disable	1: Enable	
			R / W	4: EnCompleteINTRQ	0: Disable	1: Enable	
				3:	0:	1:	
			R / W	2: EnIDE_Cmp	0: Disable	1: Enable	
			R / W	1: EnDetectINTRQ	0: Disable	1: Enable	
			R / W	0: EnDetectTerm	0: Disable	1: Enable	

IDE\_IntStat レジスタの割り込み要因による、MainIntStat レジスタの IDE\_IntStat ビットのアサートを許可／禁止します。

## 2.6.12. 015h MediaFIFO\_IntEnb (Media FIFO Interrupt Enable)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	015h	MediaFIFO_IntEnb		7:	0:	1:	00h
			R / W	6: EnMediaIDE_Cmp	0: Disable	1: Enable	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
			R / W	2: EnFIFO_NotEmpty	0: Disable	1: Enable	
			R / W	1: EnFIFO_Full	0: Disable	1: Enable	
			R / W	0: EnFIFO_Empty	0: Disable	1: Enable	

MediaFIFO\_IntStat レジスタの割り込み要因による、MainIntStat レジスタの MediaFIFO\_IntStat ビットのアサートを許可／禁止します。

## 2. レジスタ

---

### 2.6.13. 020h *RevisionNum* (*Revision Number*)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	020h	<i>RevisionNum</i>	R	7: <i>RevisionNum</i> [7]	Revision Number	60h
				6: <i>RevisionNum</i> [6]		
				5: <i>RevisionNum</i> [5]		
				4: <i>RevisionNum</i> [4]		
				3: <i>RevisionNum</i> [3]		
				2: <i>RevisionNum</i> [2]		
				1: <i>RevisionNum</i> [1]		
				0: <i>RevisionNum</i> [0]		

本 LSI のリビジョンナンバーを示します。このレジスタは SLEEP / SNOOZE 中でもアクセス有効です。

現仕様に関してのリビジョンナンバーは、0x60 となっています。



2.6.14. 021h *ChipReset* (*Chip Reset*)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	021h	<b>ChipReset</b>		7:	0:	1:	XXh
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
			W	0: <b>AllReset</b>	0: None	1: Reset	

本 LSI をリセットします。  
SLEEP / SNOOZE 時でもアクセス有効です。

**Bit7-1**    **Reserved**

**Bit0**     **AllReset**

CPUIF\_MODE レジスタを除く全回路を初期化します。本レジスタによるソフトリセットでは、未初期化期間には遷移しません。

なお、このレジスタにリセット用途以外で、書き込みしないで下さい。

AC スペックに違反してこのレジスタにリセット用途以外の書き込みを行った場合、誤動作の原因となりますのでご注意ください。

## 2. レジスタ

### 2.6.15. 022h PM\_Control\_0 (Power Management Control 0)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	022h	PM_Control_0	R / W	7: <b>GoSLEEP</b>	0: Do nothing	1: Go to SLEEP	00h
			R / W	6: <b>GoSNOOZE</b>	0: Do nothing	1: Go to SNOOZE	
			R / W	5: <b>GoActive60</b>	0: Do nothing	1: Go to ACTIVE60	
			R / W	4: <b>GoActDevice</b>	0: Do nothing	1: Go to ACT_DEVICE	
			R / W	3: <b>GoActHost</b>	0: Do nothing	1: Go to ACT_HOST	
			R / W	2: <b>GoActAllDev</b>	0: Do nothing	1: Go to ACT_ALL_DEV	
			R / W	1: <b>GoActAllHost</b>	0: Do nothing	1: Go to ACT_ALL_HOST	
				0:	0:	1:	

本 LSI のパワーマネージメント関連の動作設定を行います。  
このレジスタは SLEEP / SNOOZE 中も有効です。

#### Bit7 GoSLEEP

SLEEP ステート以外のステートから、SLEEP ステートへの移行を開始します。

SNOOZE ステート時に本ビットを"1"にセットすると、オシレータの発振を停止し SLEEP ステートに移行します。ACTIVE60 ステート時に本ビットを"1"にセットすると、まず PLL60 の発振を停止し、その後オシレータの発振を停止し SLEEP ステートに移行します。ACT\_DEVICE ステート時に本ビットを"1"にセットすると、最初に DevicePLL480 の発振を停止し、次に PLL60 の発振を停止し、その後オシレータの発振を停止し SLEEP ステートに移行します。ACT\_HOST ステート時に本ビットを"1"にセットすると、最初に HostPLL480 の発振を停止し、次に PLL60 の発振を停止し、その後オシレータの発振を停止し SLEEP ステートに移行します。ACT\_ALL ステート時に本ビットを"1"にセットすると、最初に DevicePLL480 及び HostPLL480 の発振を停止し、次に PLL60 の発振を停止し、その後オシレータの発振を停止し、SLEEP ステートに移行します。

どのステートからの移行であっても、移行が完了し次第、本ビットは自動的にクリアされ、同時に MainIntStat.FinishedPM ビットがセットされます。

#### Bit6 GoSNOOZE

SNOOZE ステート以外のステートから、SNOOZE ステートへの移行を開始します。

SLEEP ステート時に本ビットを"1"にセットすると、オシレータの発振を開始し、オシレータ発振安定時間(WakeupTim\_H, L に設定された時間)経過後、SNOOZE ステートに移行します。ACTIVE60 ステート時に本ビットを"1"にセットすると、PLL60 の発振を停止し、SNOOZE ステートに移行します。ACT\_DEVICE ステート時に本ビットを"1"にセットすると、最初に DevicePLL480 の発振を停止し、次に PLL60 の発振を停止し、SNOOZE ステートに移行します。ACT\_HOST ステート時に本ビットを"1"にセットすると、最初に HostPLL480 の発振を停止し、次に PLL60 の発振を停止し、SNOOZE ステートに移行します。ACT\_ALL ステート時に本ビットを"1"にセットすると、最初に DevicePLL480 及び HostPLL480 の発振を停止し、次に PLL60 の発振を停止し、SNOOZE ステートに移行します。

どのステートからの移行であっても、移行が完了し次第、本ビットは自動的にクリアされ、同時に MainIntStat.FinishedPM ビットがセットされます。

#### Bit5 GoActive60

ACTIVE60 ステート以外のステートから、ACTIVE60 ステートへの移行を開始します。

SLEEP ステート時に本ビットを"1"にセットすると、オシレータの発振を開始し、オシレータ発振安定時間(WakeupTim\_H, L に設定された時間)経過後、PLL60 の発振を開始し、PLL60 発振安定時間(約 250us)経過後、ACTIVE60 ステートに移行します。SNOOZE ステート時に本ビットを"1"にセットすると、PLL60 の発振を開始し、PLL 発振安定時間(約 250us)経過後、ACTIVE60 ステートに移行します。ACT\_DEVICE ステート時に本ビットを"1"にセットすると、DevicePLL480 の発振を停止し、ACTIVE60 ステートに移行します。ACT\_HOST ステート時に

本ビットを”1”にセットすると、HostPLL480 の発振を停止し、ACTIVE60 ステートに移行します。ACT\_ALL ステート時に本ビットを”1”にセットすると、DevicePLL480 及び HostPLL480 の発振を停止し、ACTIVE60 ステートに移行します。

どのステートからの移行であっても、移行が完了し次第、本ビットは自動的にクリアされ、同時に MainIntStat.FinishedPM ビットがセットされます。

#### Bit4 GoActDevice

ACT\_DEVICE ステート以外のステートから、ACT\_DEVICE ステートへの移行を開始します。

SLEEP ステート時に本ビットを”1”にセットすると、オシレータの発振を開始し、オシレータ発振安定時間(WakeupTim\_H, L に設定された時間)経過後、PLL60 の発振を開始し、PLL60 発振安定時間(約 250us)経過後、DevicePLL480 の発振を開始し、PLL480 発振安定時間(約 250us)経過後、ACT\_DEVICE ステートに移行します。SNOOZE ステート時に本ビットを”1”にセットすると、PLL60 の発振を開始し、PLL 発振安定時間(約 250us)経過後、DevicePLL480 の発振を開始し、PLL480 発振安定時間(約 250us)経過後、ACT\_DEVICE ステートに移行します。ACTIVE60 ステート時に本ビットを”1”にセットすると、DevicePLL480 の発振を開始し、PLL480 発振安定時間(約 250us)経過後、ACT\_DEVICE ステートに移行します。ACT\_HOST ステート時に本ビットを”1”にセットすると、HostPLL480 の発振を停止し、ACTIVE60 ステートに移行します。その後 DevicePLL480 の発振を開始し、PLL480 発振安定時間(約 250us)経過後、ACT\_DEVICE ステートに移行します。ACT\_ALL ステート時に本ビットを”1”にセットすると、HostPLL480 の発振を停止し、ACT\_DEVICE ステートに移行します。

どのステートからの移行であっても、移行が完了し次第、本ビットは自動的にクリアされ、同時に MainIntStat.FinishedPM ビットがセットされます。

#### Bit3 GoActHost

ACT\_HOST ステート以外のステートから、ACT\_HOST ステートへの移行を開始します。

SLEEP ステート時に本ビットを”1”にセットすると、オシレータの発振を開始し、オシレータ発振安定時間(WakeupTim\_H, L に設定された時間)経過後、PLL60 の発振を開始し、PLL60 発振安定時間(約 250us)経過後、HostPLL480 の発振を開始し、PLL480 発振安定時間(約 250us)経過後、ACT\_HOST ステートに移行します。SNOOZE ステート時に本ビットを”1”にセットすると、PLL60 の発振を開始し、PLL 発振安定時間(約 250us)経過後、HostPLL480 の発振を開始し、PLL480 発振安定時間(約 250us)経過後、ACT\_HOST ステートに移行します。ACTIVE60 ステート時に本ビットを”1”にセットすると、HostPLL480 の発振を開始し、PLL480 発振安定時間(約 250us)経過後、ACT\_HOST ステートに移行します。ACT\_DEVICE ステート時に本ビットを”1”にセットすると、DevicePLL480 の発振を停止し、ACTIVE60 ステートに移行します。その後 HostPLL480 の発振を開始し、PLL480 発振安定時間(約 250us)経過後、ACT\_HOST ステートに移行します。ACT\_ALL ステート時に本ビットを”1”にセットすると、DevicePLL480 の発振を停止し、ACT\_HOST ステートに移行します。

どのステートからの移行であっても、移行が完了し次第、本ビットは自動的にクリアされ、同時に MainIntStat.FinishedPM ビットがセットされます。

#### Bit2 GoActAllDev

ACT\_ALL\_DEV サブステート以外のステートから、ACT\_ALL\_DEV サブステートへの移行を開始します。

この ACT\_ALL\_DEV サブステートは、ACT\_ALL ステートへ移行後、PM\_Control\_1.MonActFunc ビットを”0”にクリアして移行するサブステートとして定義しています。

SLEEP ステート時に本ビットを”1”にセットすると、オシレータの発振を開始し、オシレータ発振安定時間(WakeupTim\_H, L に設定された時間)経過後、PLL60 の発振を開始し、PLL60 発振安定時間(約 250us)経過後、DevicePLL480 及び HostPLL480 の発振を開始し、PLL480 発振安定時間(約 250us)経過後、PM\_Control\_1.MonActFunc ビットを”0”にクリアし、ACT\_ALL\_DEV サブステートに移行します。SNOOZE ステート時に本ビットを”1”にセットすると、PLL60 の発振を開始し、PLL 発振安定時間(約 250us)経過後、DevicePLL480 及び HostPLL480 の発振を開始し、PLL480 発振安定時間(約 250us)経過後、PM\_Control\_1.MonActFunc ビットを”0”にク

## 2. レジスタ

リアし、ACT\_ALL\_DEV サブステートに移行します。ACTIVE60 ステート時に本ビットを”1”にセットすると、DevicePLL480 及び HostPLL480 の発振を開始し、PLL480 発振安定時間(約 250us)経過後、PM\_Control\_1.MonActFunc ビットを”0”にクリアし、ACT\_ALL\_DEV サブステートに移行します。ACT\_DEVICE ステート時に本ビットを”1”にセットすると、HostPLL480 の発振を開始し、PLL480 発振安定時間(約 250us)経過後、PM\_Control\_1.MonActFunc ビットを”0”にクリアし、ACT\_ALL\_DEV サブステートに移行します。ACT\_ALL\_HOST サブステート時に本ビットを”1”にセットすると、PM\_Control\_1.MonActFunc ビットを”0”にクリアし、ACT\_ALL\_DEV サブステートに移行します。

どのステートからの移行であっても、移行が完了し次第、本ビットは自動的にクリアされ、同時に MainIntStat.FinishedPM ビットがセットされます。

この ACT\_ALL\_DEV サブステートでは、DevicePLL480 及び HostPLL480 の両 PLL が動作しており、両機能が使用できますが、データ転送は USB Device 側でしか行えません。

### Bit1 GoActAllHost

ACT\_ALL\_HOST サブステート以外のステートから、ACT\_ALL\_HOST サブステートへの移行を開始します。

この ACT\_ALL\_HOST サブステートは、ACT\_ALL ステートへ移行後、PM\_Control\_1.MonActFunc ビットを”1”にセットして移行するサブステートとして定義しています。

SLEEP ステート時に本ビットを”1”にセットすると、オシレータの発振を開始し、オシレータ発振安定時間(WakeupTim\_H, L に設定された時間)経過後、PLL60 の発振を開始し、PLL60 発振安定時間(約 250us)経過後、DevicePLL480 及び HostPLL480 の発振を開始し、PLL480 発振安定時間(約 250us)経過後、PM\_Control\_1.MonActFunc ビットを”1”にセットし、ACT\_ALL\_HOST サブステートに移行します。SNOOZE ステート時に本ビットを”1”にセットすると、PLL60 の発振を開始し、PLL 発振安定時間(約 250us)経過後、DevicePLL480 及び HostPLL480 の発振を開始し、PLL480 発振安定時間(約 250us)経過後、PM\_Control\_1.MonActFunc ビットを”1”にセットし、ACT\_ALL\_HOST サブステートに移行します。ACTIVE60 ステート時に本ビットを”1”にセットすると、DevicePLL480 及び HostPLL480 の発振を開始し、PLL480 発振安定時間(約 250us)経過後、PM\_Control\_1.MonActFunc ビットを”1”にセットし、ACT\_ALL\_HOST サブステートに移行します。ACT\_DEVICE ステート時に本ビットを”1”にセットすると、HostPLL480 の発振を開始し、PLL480 発振安定時間(約 250us)経過後、PM\_Control\_1.MonActFunc ビットを”1”にセットし、ACT\_ALL\_HOST サブステートに移行します。ACT\_ALL\_DEV サブステート時に本ビットを”1”にセットすると、PM\_Control\_1.MonActFunc ビットを”1”にセットし、ACT\_ALL\_HOST サブステートに移行します。

どのステートからの移行であっても、移行が完了し次第、本ビットは自動的にクリアされ、同時に MainIntStat.FinishedPM ビットがセットされます。

この ACT\_ALL\_HOST サブステートでは、DevicePLL480 及び HostPLL480 の両 PLL が動作しており、両機能が使用できますが、データ転送は Host 側でしか行えません。

※本 LSI は SLEEP / SNOOZE 中にアクセスできない割り込みステータス(以下、同期ステータス)により、SNOOZE 中に XINT 信号がアサートされないようマスクしておりますが、SNOOZE 解除と同時に XINT 端子がアサートされるのを回避するため、F/W により以下の処理を行って下さい。

＜SLEEP / SNOOZE 開始前＞

同期ステータスを処理し、クリアする(～IntStat)。

同期ステータスをディスエーブルにする(～IntEnb)。

＜SLEEP / SNOOZE 解除後＞

同期ステータスをクリアする(～IntStat)。

同期ステータスをイネーブルにする(～IntEnb)。

### Bit0 Reserved

2.6.16. 023h *PM\_Control\_1* (Power Management Control 1)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset	
Device / Host	023h	<i>PM_Control_1</i>	R	7: <i>MonActFunc</i>	0: Device mode	1: Host mode	00h	
				6:	0:	1:		
				5:	0:	1:		
				4:	0:	1:		
			R	3: <i>PM_State[3]</i>		PM_State[3:0]		
				2: <i>PM_State[2]</i>				
				1: <i>PM_State[1]</i>				
				0: <i>PM_State[0]</i>				

本 LSI のパワーマネジメント関連の動作設定を行います。  
このレジスタは、SLEEP / SNOOZE 中も有効です。

**Bit7      MonActFunc**

ACT\_ALL ステート時に、転送が行える USB 機能を表します。

0: USB Device でデータ転送可能

1: USB Host でデータ転送可能

**Bit7-4      Reserved****Bit3-0      PM\_State[3:0]**

パワーマネジメントのステートを表します。

0000: SLEEP ステート                      (OSC オフ、PLL60 オフ、DevicePLL480 オフ、HostPLL480 オフ)

0001: SNOOZE ステート                      (OSC オン、PLL60 オフ、DevicePLL480 オフ、HostPLL480 オフ)

0011: ACTIVE60 ステート      (OSC オン、PLL60 オン、DevicePLL480 オフ、HostPLL480 オフ)

0111: ACT\_DEVICE ステート (OSC オン、PLL60 オン、DevicePLL480 オン、HostPLL480 オフ)

1011: ACT\_HOST ステート      (OSC オン、PLL60 オン、DevicePLL480 オフ、HostPLL480 オン)

1111: ACT\_ALL ステート                      (OSC オン、PLL60 オン、DevicePLL480 オン、HostPLL480 オン)

その他: 未使用

なお本ステートは、PM\_Control\_0.GoXXXX をセットしてから

MainIntStat.FinishedPM 割り込みステータスがセットされ、

PM\_Control\_0.GoXXXX ビットがクリアされるまでは、

該当ステートに向かい逐次変化致しますので、参照しないでください。

ACT\_ALL ステートで、当レジスタの MonActFunc ビットが”0”にクリアされている場合は、ACT\_ALL\_DEV サブステート、”1”にセットされている場合は、ACT\_ALL\_HOST サブステートと定義しています。

## 2. レジスタ

### 2.6.17. 024h *WakeupTim\_H* (Wakeup Time High)

### 2.6.18. 025h *WakeupTim\_L* (Wakeup Time Low)

Mode	Addresses	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	024h -025h	<i>WakeupTim_H</i> <i>WakeupTim_L</i>	R / W	<i>WakeupTim[15:0]</i>	Wakeup Time[15:0]	0000h

SLEEP ステートから SNOOZE ステートへ復帰する際のオシレータ発振安定時間を設定します。本レジスタは SLEEP 時にもアクセスが可能です。

SLEEP ステート時に、PM\_Control\_0.GoActDevice、PM\_Control\_0.GoActHost、PM\_Control\_0.GoActive60、PM\_Control\_0.GoSNOOZE ビットに"1"が書き込まれた際に、発振セルをイネーブルにし、オシレータの発振を開始します。この時、カウンタにこの *WakeupTim\_H,L* の設定値をロードし、OSC の立ち上がりにてカウントダウンを始めます。カウントダウン終了後、内部 OSCCLK のゲートを開き、PLL 等の回路に CLK を送出開始します。

このオシレータ発振安定時間は、発振子、発振セル、基板、負荷容量等により変化します。もし、デバイス動作時において、USB の SUSPEND 時に SLEEP ステートまで落とす場合は、USB の RESET 検出から 5.1ms 以内に 60MHz±10%に内部 SCLK が安定していなければなりません。

したがって、オシレータ発振安定時間 + PLL60 安定時間(250us 未満)+ PLL480 安定時間(250us 未満)の合計が 5.1ms 以下とならなければなりません。

2.6.19. 026h *H\_USB\_Control (Host USB Control)*

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	026h	<i>H_USB_Control</i>	R / W	7: <i>VBUS_Enb</i>	0: Disable	1: Enable	00h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
				0:	0:	1:	

ホスト に関する動作設定を行います。  
このレジスタは SLEEP / SNOOZE 中も有効です。

**Bit7**     *VBUS\_Enb*

VBUSEN\_A 端子(出力)の状態を設定します。デフォルトでローレベルです。

**Bit6-0**     **Reserved**

## 2. レジスタ

### 2.6.20. 027h H\_XcvrControl (Host Xcvr Control)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	027h	<i>H_XcvrControl</i>	R / W	7: <i>TermSelect</i>	0: HS Termination	1: FS Termination	91h
			R / W	6: <i>RemovedRPD</i>	0: RPD ON	1: RPD OFF	
			R / W	5: <i>XcvrSelect[1]</i>	XcvrSelect[1:0]		
			R / W	4: <i>XcvrSelect[0]</i>			
				3:	0:	1:	
				2:	0:	1:	
			R / W	1: <i>OpMode[1]</i>	OpMode[1:0]		
				0: <i>OpMode[0]</i>			

ホストトランシーバマクロに関する設定を行います。  
このレジスタは SLEEP / SNOOZE 中も有効です。

#### Bit7 TermSelect

HS または FS いずれかのターミネーションを選択して有効にします。

本ビットは、H\_NogoControl\_0.AutoMode を設定することにより H/W が自動的に設定を行いますので、設定を行わないでください。

#### Bit6 RemovedRPD

ホストデータラインである DP\_A および DM\_A 用内蔵プルダウン抵抗を ON/OFF します。

0: RPD ON

1: RPD OFF

本ビットは、通常は”0”(ON)の設定のままで使用してください。

特にホスト動作時(SUSPEND も含む)においては、必ず”0”(ON)の設定のままで使用してください。本ビットを”0”から変化させますと、ホストデータラインの特性が変わり、USB における誤動作を引き起こす可能性がありますのでご注意ください。

#### Bit5-4 XcvrSelect[1:0]

HS、FS、LS のいずれかのトランシーバを選択して有効にします。

00: High Speed トランシーバ

01: Full Speed トランシーバ

10: Reserved

11: Low Speed トランシーバ

本ビットは、H\_NogoControl\_0.AutoMode を設定することにより H/W が自動的に設定を行いますので、設定を行わないでください。

#### Bit3-2 Reserved

#### Bit1-0 OpMode

HTM のオペレーションモードを設定します。

本ビットは、H\_NogoControl\_0.AutoMode を設定することにより H/W が自動的に設定を行いますので、設定を行わないでください。

ただし、ACT\_HOSTステート以外の状態でホストポートの信号ライン変化ステータスを検出する場合は、「1.1.3.2.信号ライン変化ステータスの使用方法」をご参照のうえ、本ビットの設定を行ってください。

OpMode		
00	"Normal Operation"	通常使用状態
01	"Non-Driving"	非使用状態
10	"Disable Bitstuffing and NRZI encoding"	通常使用状態で Bitstuffing と NRZI encoding 機能ディセーブル状態
11	"Power-Down"	シングルエンドレシーバのみ使用する状態



2.6.21. 028h *D\_USB\_Status* (Device USB Status)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	028h	<i>D_USB_Status</i>	R	7: <i>VBUS</i>	0: VBUS = L	1: VBUS = H	XXh
			R / W	6: FSxHS	0: HS mode	1: FS mode	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
			R	1: <i>LineState[1]</i>	Line State[1:0]		
				0: <i>LineState[0]</i>			

デバイスに関するステータスを表示します。

**Bit7** *VBUS*

VBUS\_B 端子の状態が表示されます。このビットは SLEEP / SNOOZE 中でも有効です。

**Bit6** *FSxHS*

現在の動作モードを示します。D\_NegoControl.GoChirp ビットにより "HS Detection Handshake" (「1.2.7.11.5.HS Detection Handshake」参照)を実行すると、自動的に設定されます。このビットを書き込むことにより動作モードを強制的に変更することも可能ですが、シミュレーション等で "HS Detection Handshake"を行わずに動作モードを切り替えたい場合にのみ、このビットを操作してください。

ケーブルアタッチ時に、"FS(1)"にセットして下さい。

このビットは ACTIVE60 / ACT\_DEVICE / ACT\_HOST 時に読むことができ、ACT\_DEVICE 時に書くことができます。

**Bit5-2** *Reserved***Bit1-0** *LineState[1:0]*

USB ケーブル上の信号状態を示します。このビットは SLEEP / SNOOZE 中でも有効です。

D\_XcvrControl レジスタの TermSelect ビットが"1"である時(FS ターミネーション選択時)、XcvrSelect ビットが"1"

(FS トランシーバ選択時)であれば、DP/DM の FS レシーバの受信値を、XcvrSelect が"0"(HS トランシーバ選択時)であれば、HS レシーバの受信値を示します。

TermSelect が"0"である時は、USB のバス・アクティビティを示します。

<b>LineState</b>		
<b>TermSelect</b>	<b>DP / DM</b>	<b>LineState[1:0]</b>
0	Don't Care	バス・アクティビティ
1	SE0	0b00
1	J	0b01
1	K	0b10
1	SE1	0b11

## 2. レジスタ

### 2.6.22. 029h *H\_USB\_Status* (Host USB Status)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	029h	<b><i>H_USB_Status</i></b>	R	7: <b><i>VBUS_State</i></b>	0: VBUSFLG_A = High	1: VBUSFLG _A= Low	XXh
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
			R	1: <b><i>LineState[1]</i></b>	Line State[1:0]		
				0: <b><i>LineState[0]</i></b>			

ホストに関するステータスを表示します。

#### Bit7 *VBUS\_State*

VBUSFLG\_A 端子の状態が表示されます。このビットは SLEEP / SNOOZE 中も有効です。

#### Bit6-2 *Reserved*

#### Bit1-0 *LineState[1:0]*

USB ケーブル上の信号状態を示します。このビットは SLEEP / SNOOZE も有効です。

H\_XcvrControl レジスタの XcvrSelect[1:0]が"01"である時(FS トランシーバ選択時)は DP/DM の FS レシーバの受信値を、"11"である時(LS トランシーバ選択時)は DP/DM の LS レシーバの受信値を示します。

XcvrSelect[1:0]が"00"である時(HS トランシーバ選択時)は、USB のバス・アクティビティを示します。

<i>LineState</i>		
XcvrSelect[1:0]	DP / DM	LineState[1:0]
00	Don't Care	バス・アクティビティ アクティビティ有: 0b01 アクティビティ無: 0b00
01 または 11	SE0	0b00
01 または 11	J	0b01
01 または 11	K	0b10
01 または 11	SE1	0b11

(注)XcvrSelect[1:0]="10"のコードは Reserved のため、そのときの動作は保証いたしません。

## 2.6.23. 030h FIFO\_Rd\_0 (FIFO Read 0)

## 2.6.24. 031h FIFO\_Rd\_1 (FIFO Read 1)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	030h	FIFO_Rd_0	R	7: FIFO_Rd_0[7]	Endpoint n / Channel n / Media FIFO Read	XXh
				6: FIFO_Rd_0[6]		
				5: FIFO_Rd_0[5]		
				4: FIFO_Rd_0[4]		
				3: FIFO_Rd_0[3]		
				2: FIFO_Rd_0[2]		
				1: FIFO_Rd_0[1]		
				0: FIFO_Rd_0[0]		

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	031h	FIFO_Rd_1	R	7: FIFO_Rd_1[7]	Endpoint n / Channel n / Media FIFO Read	XXh
				6: FIFO_Rd_1[6]		
				5: FIFO_Rd_1[5]		
				4: FIFO_Rd_1[4]		
				3: FIFO_Rd_1[3]		
				2: FIFO_Rd_1[2]		
				1: FIFO_Rd_1[1]		
				0: FIFO_Rd_1[0]		

## 030h.Bit7-0, 031h.Bit7-0 FIFO\_Rd\_0[7:0], FIFO\_Rd\_1[7:0]

AREAx{x=0-5}Join\_0.JoinCPU\_Rd ビットがセットされているエンドポイント FIFO、または MediaFIFO\_Join.JoinCPU\_Rd ビットがセットされているメディア FIFO のデータを読み出せます。

FIFOにバイト境界がある場合に、このレジスタを読み出した場合は片側にのみ有効なデータが出力されます。詳細は「1.8.3.1.5.FIFOアクセスの端数処理」をご参照ください。

このレジスタを用いて FIFO のデータを読み出す場合は、必ず FIFO\_RdRemain\_H,L レジスタにより読み出し可能データ数を確認した後、リードを行ってください。

## 2. レジスタ

### 2.6.25. 032h FIFO\_Wr\_0(FIFO Write 0)

### 2.6.26. 033h FIFO\_Wr\_1(FIFO Write 1)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	032h	FIFO_Wr_0	W	7: FIFO_Wr_0[7]	Endpoint n / Channel n / Media FIFO Write	XXh
				6: FIFO_Wr_0[6]		
				5: FIFO_Wr_0[5]		
				4: FIFO_Wr_0[4]		
				3: FIFO_Wr_0[3]		
				2: FIFO_Wr_0[2]		
				1: FIFO_Wr_0[1]		
				0: FIFO_Wr_0[0]		

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	033h	FIFO_Wr_1	W	7: FIFO_Wr_1[7]	Endpoint n / Channel n / Media FIFO Write	XXh
				6: FIFO_Wr_1[6]		
				5: FIFO_Wr_1[5]		
				4: FIFO_Wr_1[4]		
				3: FIFO_Wr_1[3]		
				2: FIFO_Wr_1[2]		
				1: FIFO_Wr_1[1]		
				0: FIFO_Wr_1[0]		

### 32h.Bit7-0, 33h.Bit7-0 FIFO\_Wr\_0[7:0], FIFO\_Wr\_1[7:0]

AREAx{x=0-5}Join\_0.JoinCPU\_Wr ビットがセットされているエンドポイント FIFO、または MediaFIFO\_Join.JoinCPU\_Wr ビットがセットされているメディア FIFO ヘデータを書き込みます。

FIFOにバイト境界がある場合に、このレジスタへ書き込んだ場合は片側のみ書き込みが行われます。詳細は「1.8.3.1.5.FIFOアクセスの端数処理」をご参照ください。

このレジスタを用いて FIFO ヘデータを書き込む場合は、必ず FIFO\_WrRemain\_H,L レジスタにより、書き込み可能データ数を確認した後、ライトを行ってください。

## 2.6.27. 034h FIFO\_RdRemain\_H (FIFO Read Remain High)

## 2.6.28. 035h FIFO\_RdRemain\_L (FIFO Read Remain Low)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device       / Host	034h	FIFO_RdRemain_H	R	7: RdRemainValid	0:None	1: Read Remain Valid	00h
				6:	0:	1:	
				5:	0:	1:	
			R	4: RdRemain[12]	Endpoint n / Channel n / Media FIFO Read Remain High		
				3: RdRemain[11]			
				2: RdRemain[10]			
				1: RdRemain[9]			
				0: RdRemain[8]			

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	035h	FIFO_RdRemain_L	R	7: RdRemain[7]	Endpoint n / Channel n / Media FIFO Read Remain Low	00h
				6: RdRemain[6]		
				5: RdRemain[5]		
				4: RdRemain[4]		
				3: RdRemain[3]		
				2: RdRemain[2]		
				1: RdRemain[1]		
				0: RdRemain[0]		

34h.Bit7 **RdRemainValid**

AREAx{x=0-5}Join\_0.JoinCPU\_Rd ビット、または MediaFIFO\_Join.JoinCPU\_Rd ビットによって、CPU I/F にエンドポイント／チャネル／メディアがジョインされかつ、FIFO\_RdRemain の値が有効な場合に"1"にセットされます。このビットがクリアされている場合の RdRemain の値は無効です。

34h.Bit6-5 **Reserved**34h.Bit4-0, 35h.Bit7-0 **RdRemain[12:0]**

AREAx{x=0-5}Join\_0.JoinCPU\_Rd ビット、または MediaFIFO\_Join.JoinCPU\_Rd ビットによって CPU I/F に接続しているエンドポイント／チャネルの FIFO 内のリード可能なデータ数を示します。FIFO のリード可能なデータ数を取得する場合は、FIFO\_RdRemain\_H レジスタと FIFO\_RdRemain\_L レジスタを対でアクセスする必要があります。

## 2. レジスタ

### 2.6.29. 036h FIFO\_WrRemain\_H (FIFO Write Remain High)

### 2.6.30. 037h FIFO\_WrRemain\_L (FIFO Write Remain Low)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	036h	WrRemain_H		7:	0:	1:	00h
				6:	0:	1:	
				5:	0:	1:	
			R	4: WrRemain[12]	Endpoint n / Channel n / Media FIFO Write Remain High		
				3: WrRemain[11]			
				2: WrRemain[10]			
				1: WrRemain[9]			
				0: WrRemain[8]			

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	037h	WrRemain_L	R	7: WrRemain[7]	Endpoint n / Channel n / Media FIFO Write Remain Low	00h
				6: WrRemain[6]		
				5: WrRemain[5]		
				4: WrRemain[4]		
				3: WrRemain[3]		
				2: WrRemain[2]		
				1: WrRemain[1]		
				0: WrRemain[0]		

36h.Bit7-5

Reserved

36h.Bit4-0, 37h.Bit7-0

WrRemain[12:0]

AREAx{x=0-5}Join\_0.JoinCPU\_Wr ビットにより CPU I/F に接続しているエンドポイント FIFO、または MediaFIFO\_Join.JoinCPU\_Wr ビットにより CPU I/F に接続しているチャネル FIFO の空き容量を示します。ただし、FIFO への書き込み直後には正確な FIFO の空き容量を確認することは出来ません。1CPU サイクル以上の間隔を開けて FIFO の空き容量を確認して下さい。FIFO の空き容量を取得する場合は、FIFO\_WrRemain\_H レジスタと FIFO\_WrRemain\_L レジスタを対でアクセスする必要があります。

## 2.6.31. 038h FIFO\_ByteRd(FIFO Byte Read)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	038h	FIFO_ByteRd	R	7: FIFO_ByteRd[7]	Endpoint n / Channel n / Media FIFO Byte Read	XXh
				6: FIFO_ByteRd[6]		
				5: FIFO_ByteRd[5]		
				4: FIFO_ByteRd[4]		
				3: FIFO_ByteRd[3]		
				2: FIFO_ByteRd[2]		
				1: FIFO_ByteRd[1]		
				0: FIFO_ByteRd[0]		

**Bit7-0 FIFO\_ByteRd[7:0]**

AREAx{x=0-5}Join\_0.JoinCPU\_Rd ビットがセットされているエンドポイント FIFO、または MediaFIFO\_Join.JoinCPU\_Rd ビットがセットされているチャネル FIFO のデータをバイト単位で読み出せます。このレジスタを用いて FIFO のデータを読み出す場合は、必ず FIFO\_RdRemain\_H,L レジスタにより読み出し可能データ数を確認した後、リードを行ってください。

## 2. レジスタ

---

### 2.6.32. 03Ah FIFO\_ByteWr(FIFO Byte Write)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	03Ah	FIFO_ByteWr	W	7: FIFO_ByteWr[7]	Endpoint n / Channel n / Media FIFO Byte Write	XXh
				6: FIFO_ByteWr[6]		
				5: FIFO_ByteWr[5]		
				4: FIFO_ByteWr[4]		
				3: FIFO_ByteWr[3]		
				2: FIFO_ByteWr[2]		
				1: FIFO_ByteWr[1]		
				0: FIFO_ByteWr[0]		

#### Bit7-0 FIFO\_ByteWr[7:0]

AREAx{x=0-5}Join\_0.JoinCPU\_Wr ビットがセットされているエンドポイント FIFO、または MediaFIFO\_Join.JoinCPU\_Wr ビットがセットされている FIFO にデータをバイト単位で書き込みます。このレジスタを用いて FIFO ヘデータを書き込む場合は、必ず FIFO\_WrRemain\_H,L レジスタにより、書き込み可能データ数を確認した後、ライトを行ってください。



**2.6.33. 040h RAM\_RdAdrs\_H (RAM Read Address High)****2.6.34. 041h RAM\_RdAdrs\_L (RAM Read Address Low)**

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	040h	RAM_RdAdrs_H		7:	0:	1:	00h
				6:	0:	1:	
				5:	0:	1:	
			R / W	4: RAM_RdAdrs[12]	RAM Read Address		
				3: RAM_RdAdrs[11]			
				2: RAM_RdAdrs[10]			
				1: RAM_RdAdrs[9]			
				0: RAM_RdAdrs[8]			

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	041h	RAM_RdAdrs_L	R / W	7: RAM_RdAdrs[7]	RAM Read Address	00h
				6: RAM_RdAdrs[6]		
				5: RAM_RdAdrs[5]		
				4: RAM_RdAdrs[4]		
				3: RAM_RdAdrs[3]		
				2: RAM_RdAdrs[2]		
				1:		
				0:		

**040h.Bit7-5                      Reserved****040h.Bit4-0, 041h.Bit7-2      RAM\_RdAdrs[12:2]**

RAM\_Rd を行う先頭アドレスを設定します。このレジスタを設定した後、RAM\_RdCount レジスタを設定し、RAM\_RdControl レジスタのビットをセットして下さい。RAM\_Rd 機能が開始します。このレジスタの値は、RAM\_Rd 機能作動中は内部動作に応じて変化します。従って、一旦 RAM\_RdControl レジスタのビットをセットして、RAM\_Rd 機能を開始させた後は CPU\_IntStat.RAM\_RdCmp ビットがセットされるまでこのレジスタの値を読み出さないで下さい。RAM\_Rd 機能作動中にこのレジスタを読み出した場合の値は保証されません。また、RAM\_Rd 機能作動中にこのレジスタに書き込んだ場合、誤動作の原因となりますのでご注意ください。

**041h.Bit1-0                      Reserved**

## 2. レジスタ

### 2.6.35. 042h RAM\_RdControl (RAM Read Control)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	042h	RAM_RdControl	R / W	7: RAM_GoRdCBW_CSW	0: Do nothing 1: RAM Read CBW_CSW start	00h
			R / W	6: RAM_GoRd	0: Do nothing 1: RAM Read start	
				5:	0: 1:	
				4:	0: 1:	
				3:	0: 1:	
				2:	0: 1:	
				1:	0: 1:	
				0:	0: 1:	

#### Bit7 RAM\_GoRdCBW\_CSW

デバイス動作(ACT\_DEVICE ステート)時は CBW エリア、ホスト動作(ACT\_HOST ステート)時は CSW エリアに受信したデータを読み出すために RAM\_Rd 機能を開始するビットです。

デバイス動作時にこのビットに”1”を書き込むと、RAM\_Rd 機能を開始し、CBW エリアからデータをリードします。RAM\_Rd\_00～RAM\_Rd\_1E レジスタの値が有効になると、CPU\_IntStat.RAM\_RdCmp ビットが”1”にセットされ、このビットが自動的にクリアされます。

ホスト動作時にこのビットに”1”を書き込むと、RAM\_Rd 機能を開始し、CSW エリアからデータをリードします。RAM\_Rd\_00～RAM\_Rd\_0C レジスタの値が有効になると、CPU\_IntStat.RAM\_RdCmp ビットが”1”にセットされ、このビットが自動的にクリアされます。

いずれの場合も RAM\_RdAdrs\_H,L レジスタ、RAM\_RdCount レジスタの設定は必要ありません。

RAM\_GoRd ビットと同時にセットすると、本ビットの機能が優先されます。

#### Bit6 RAM\_GoRd

RAM\_Rd 機能を開始するビットです。

RAM\_RdAdrs\_H,L レジスタに RAM\_Rd を行う先頭アドレスを設定した後、RAM\_RdCount レジスタを設定し、このビットに”1”を書き込むと RAM\_Rd 機能を開始します。指定された先頭アドレスから、指定されたカウント数分のデータをリードし RAM\_Rd\_xx{xx=00-1F} レジスタの値が有効になると CPU\_IntStat.RAM\_RdCmp ビットが”1”にセットされ、このビットが自動的にクリアされます。

RAM\_GoRdCBW\_CSW ビットと同時にセットすると、RAM\_GoRdCBW\_CSW ビットの機能が優先されます。

#### Bit5-0 Reserved

## 2.6.36. 043h RAM\_RdCount (RAM Read Counter)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	043h	RAM_RdCount	R / W	7:	RAM Read Counter	00h
				6:		
				5: RAM_RdCount[5]		
				4: RAM_RdCount[4]		
				3: RAM_RdCount[3]		
				2: RAM_RdCount[2]		
				1:		
				0:		

**Bit7-0 RAM\_RdCount[5:2]**

RAM\_Rd 機能を用いて RAM\_Rd\_xx{xx=00~1F} レジスタにリードするデータ数を設定します。RAM\_RdAdrs\_H,L レジスタを設定した後、このレジスタをセットし、RAM\_RdControl レジスタのビットをセットして RAM\_Rd 機能を開始して下さい。このレジスタの値は、RAM\_Rd 機能作動中は内部動作に応じて変化します。従って、一旦 RAM\_RdControl レジスタのビットをセットして、RAM\_Rd 機能を開始させた後は CPU\_IntStat.RAM\_RdCmp ビットがセットされるまでこのレジスタの値を読み出さないで下さい。RAM\_Rd 機能作動中にこのレジスタを読み出した場合の値は保証されません。また、RAM\_Rd 機能作動中にこのレジスタに書き込んだ場合、誤動作の原因となりますのでご注意ください。

このレジスタの最大設定数は 32 バイトです。32 バイトを超えるデータ数の設定は誤動作の原因となりますのでご注意ください。

## 2. レジスタ

### 2.6.37. 044h RAM\_WrAdrs\_H (RAM Write Address High)

### 2.6.38. 045h RAM\_WrAdrs\_L (RAM Write Address Low)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	044h	RAM_WrAdrs_H		7:		00h
				6:		
				5:		
			R / W	4: RAM_WrAdrs[12]	RAM Write Address High	
				3: RAM_WrAdrs[11]		
				2: RAM_WrAdrs[10]		
				1: RAM_WrAdrs[9]		
				0: RAM_WrAdrs[8]		

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	045h	RAM_WrAdrs_L	R / W	7: RAM_WrAdrs[7]	RAM Write Address Low	00h
				6: RAM_WrAdrs[6]		
				5: RAM_WrAdrs[5]		
				4: RAM_WrAdrs[4]		
				3: RAM_WrAdrs[3]		
				2: RAM_WrAdrs[2]		
				1: RAM_WrAdrs[1]		
				0: RAM_WrAdrs[0]		

RAM\_WrDoor\_0,1 レジスタによる RAM へのライトを行うアドレスを指定します。

#### 44h.Bit7-5

#### Reserved

#### 44h.Bit4-0, 45h.Bit7-0

#### RAM\_WrAdrs[12:0]

RAMへの書き込みを行う際のアドレスを指定します。RAM\_WrDoor\_0,1 レジスタへの書き込みバイト数に応じてアドレスはインクリメントされます。RAM\_WrDoor\_0,1 レジスタ書き込み直後には正確なRAM\_WrAdrsを確認する事は出来ませんので、1CPUサイクル以上の間隔を開けて、RAM\_WrAdrsを確認して下さい。データの書き込みに関しては「2.6.39.046h RAM\_WrDoor\_0 (RAM Write Door 0)」 「2.6.40.047h RAM\_WrDoor\_1 (RAM Write Door 1)」を参照して下さい。

**2.6.39. 046h RAM\_WrDoor\_0 (RAM Write Door 0)****2.6.40. 047h RAM\_WrDoor\_1 (RAM Write Door 1)**

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	046h	RAM_WrDoor_0	W	7: RAM_WrDoor_0[7]	RAM Write Door 0	XXh
				6: RAM_WrDoor_0[6]		
				5: RAM_WrDoor_0[5]		
				4: RAM_WrDoor_0[4]		
				3: RAM_WrDoor_0[3]		
				2: RAM_WrDoor_0[2]		
				1: RAM_WrDoor_0[1]		
				0: RAM_WrDoor_0[0]		

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	047h	RAM_WrDoor_1	W	7: RAM_WrDoor_1[7]	RAM Write Door 1	XXh
				6: RAM_WrDoor_1[6]		
				5: RAM_WrDoor_1[5]		
				4: RAM_WrDoor_1[4]		
				3: RAM_WrDoor_1[3]		
				2: RAM_WrDoor_1[2]		
				1: RAM_WrDoor_1[1]		
				0: RAM_WrDoor_1[0]		

**046h.Bit7-0, 047h.Bit7-0 RAM\_WrDoor\_0[7:0], RAM\_WrDoor1[7:0]**

RAM へのライトを行う際のアクセスレジスタです。ライトオンリーのレジスタです。

書き込み開始前に、RAM\_WrAdrs\_H,L レジスタに、RAM のデータを書き込む先頭アドレスを設定して下さい。その後、このレジスタに書き込みを行えば、RAM\_WrAdrs\_H,L が自動的に書き込みバイト数に応じてインクリメントされ、順次書き込みが行えます。

デバイス・モードの場合には、RAM\_WrDoor\_0,1 レジスタにより、デスクリプタエリアおよび CSW エリアへのデータの書き込みを行えます。RAM\_WrDoor\_0,1 レジスタによりデスクリプタエリアへ書き込んだデータは、ReplyDescriptor の機能によって何度でも使用できます。即ち、このデータは Descriptor 返信機能によって、消される、または上書きされることは有りません。但し、Descriptor データを書き込んだ領域が、他のエンドポイントで確保されている領域と重なる場合には、データは上書きされることがあります。

ホスト・モードの場合には、RAM\_WrDoor\_0,1 レジスタにより、CBW エリアへのデータの書き込みを行えます。

## 2. レジスタ

### 2.6.41. 048h MediaFIFO\_Control (Media FIFO Control)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device  / Host	048h	MediaFIFO_Contr ol		7:	0:	1:	XXh
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
			W	0: MediaFIFO_Clr	0: Do nothing	1: Clear Media FIFO	

MediaFIFO の動作設定を行います。ライトオンリーのレジスタです。

#### Bit7-1 Reserved

#### Bit0 MediaFIFO\_Clr

MediaFIFO をクリアします。

このビットは、"1"をセットされると FIFO をクリアする動作のみ行い、セットされた値は保持しません。

MediaFIFO に DMA<sub>x</sub>{x=0,1}がジョインされ、かつ、該当する DMA が起動中(DMA\_Running ビットが"1"の間)に、このビットを"1"にセットしないで下さい。

## 2.6.42. 049h ClrAllMediaFIFO\_Join (Clear All Media FIFO Join)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	049h	ClrAllMediaFIFO_Join	W	7:ClrJoinIDE	0: Do nothing	1: Clear Join IDE	XXh
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
			W	3:ClrJoinDMA1	0: Do nothing	1: Clear Join DMA1	
			W	2:ClrJoinDMA0	0: Do nothing	1: Clear Join DMA0	
			W	1:ClrJoinCPU_Rd	0: Do nothing	1: Clear Join CPU_Rd	
			W	0:ClrJoinCPU_Wr	0: Do nothing	1: Clear Join CPU_Wr	

該当するポートと MediaFIFO の接続をクリアします。ライトオンリーのレジスタです。

このレジスタのビットは、接続クリア後、自動的に"0"にクリアされます。

MediaFIFO がポートに接続(MediaFIFO\_Join レジスタの該当するビットが"1"にセット)され、且つ各ポートの起動中に、このレジスタのビットを"1"にセットしないで下さい。誤動作の原因となります。

## 2. レジスタ

### 2.6.43. 04Ah MediaFIFO\_Join (Media FIFO Join)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	04Ah	MediaFIFO_Join	R / W	7:JoinIDE	0: Do nothing	1: Join MediaFIFO to IDE	00h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
			R / W	3: JoinDMA1	0: Do nothing	1: Join MediaFIFO to DMA1	
			R / W	2: JoinDMA0	0: Do nothing	1: Join MediaFIFO to DMA0	
			R / W	1: JoinCPU_Rd	0: Do nothing	1: Join MediaFIFO to CPU_Rd	
			R / W	0: JoinCPU_Wr	0: Do nothing	1: Join MediaFIFO to CPU_Wr	

MediaFIFO とデータ転送を行うポートを指定します。

#### Bit7 JoinIDE

MediaFIFO で DMA1 の転送を行います。転送の方向は、IDE\_Control.Dir ビットの設定によります。

#### Bit6-4 Reserved

#### Bit3 JoinDMA1

MediaFIFO で DMA1 の転送を行います。転送の方向は、DMA1\_Control.Dir ビットの設定によります。

#### Bit2 JoinDMA0

MediaFIFO で DMA0 の転送を行います。転送の方向は、DMA0\_Control.Dir ビットの設定によります。

#### Bit1 JoinCPU\_Rd

MediaFIFO で CPU レジスタアクセスのリード転送を行います。即ち FIFO\_Rd\_0,1 レジスタ、または、FIFO\_ByteRd レジスタのリードが行われると、このチャンネルの FIFO からデータが読み出されます。

#### Bit0 JoinCPU\_Wr

MediaFIFO で CPU レジスタアクセスのライト転送を行います。即ち FIFO\_Wr\_0,1 レジスタ、または、FIFO\_ByteWr レジスタへのライトが行われると、このエンドポイントの FIFO にデータが書き込まれます。

JoinDMAx[x=0,1] ビットを設定した場合は、DMA0\_Control.Dir ビットが 1 のときは残りデータ数、0 のときは空き容量が、DMAx[x=0,1]\_Remain\_H,L レジスタにより、それぞれ参照できます。

JoinCPU\_Rd, JoinCPU\_Wr ビットを設定した場合は、FIFO\_RdRemain\_H,L、FIFO\_WrRemain\_H,L を参照し、FIFO\_Rd\_0,1、FIFO\_ByteRd、FIFO\_Wr\_0,1、FIFO\_ByteWr レジスタからデータを読み出し、または書き込みできます。

JoinDMA1 ビット、JoinDMA0 ビット、JoinCPU\_Rd ビット、JoinCPU\_Wr ビットは、同時に 1 ビットのみを"1"にセットすることが可能です。複数のビットに対して、同時に"1"を書きこんだ場合、上位ビットが有効とされます。



- 2.6.44. 050h RAM\_Rd\_00 (RAM Read 00)  
 2.6.45. 051h RAM\_Rd\_01 (RAM Read 01)  
 2.6.46. 052h RAM\_Rd\_02 (RAM Read 02)  
 2.6.47. 053h RAM\_Rd\_03 (RAM Read 03)  
 2.6.48. 054h RAM\_Rd\_04 (RAM Read 04)  
 2.6.49. 055h RAM\_Rd\_05 (RAM Read 05)  
 2.6.50. 056h RAM\_Rd\_06 (RAM Read 06)  
 2.6.51. 057h RAM\_Rd\_07 (RAM Read 07)  
 2.6.52. 058h RAM\_Rd\_08 (RAM Read 08)  
 2.6.53. 059h RAM\_Rd\_09 (RAM Read 09)  
 2.6.54. 05Ah RAM\_Rd\_0A (RAM Read 0A)  
 2.6.55. 05Bh RAM\_Rd\_0B (RAM Read 0B)  
 2.6.56. 05Ch RAM\_Rd\_0C (RAM Read 0C)  
 2.6.57. 05Dh RAM\_Rd\_0D (RAM Read 0D)  
 2.6.58. 05Eh RAM\_Rd\_0E (RAM Read 0E)  
 2.6.59. 05Fh RAM\_Rd\_0F (RAM Read 0F)  
 2.6.60. 060h RAM\_Rd\_10 (RAM Read 10)  
 2.6.61. 061h RAM\_Rd\_11 (RAM Read 11)  
 2.6.62. 062h RAM\_Rd\_12 (RAM Read 12)  
 2.6.63. 063h RAM\_Rd\_13 (RAM Read 13)  
 2.6.64. 064h RAM\_Rd\_14 (RAM Read 14)  
 2.6.65. 065h RAM\_Rd\_15 (RAM Read 15)  
 2.6.66. 066h RAM\_Rd\_16 (RAM Read 16)  
 2.6.67. 067h RAM\_Rd\_17 (RAM Read 17)  
 2.6.68. 068h RAM\_Rd\_18 (RAM Read 18)  
 2.6.69. 069h RAM\_Rd\_19 (RAM Read 19)  
 2.6.70. 06Ah RAM\_Rd\_1A (RAM Read 1A)  
 2.6.71. 06Bh RAM\_Rd\_1B (RAM Read 1B)  
 2.6.72. 06Ch RAM\_Rd\_1C (RAM Read 1C)  
 2.6.73. 06Dh RAM\_Rd\_1D (RAM Read 1D)  
 2.6.74. 06Eh RAM\_Rd\_1E (RAM Read 1E)  
 2.6.75. 06Fh RAM\_Rd\_1F (RAM Read 1F)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	050h ~ -06Fh	RAM_Rd_00 ~ RAM_Rd_1F	R	7: RAM_Rd_xx[7]	RAM Read	00h
				6: RAM_Rd_xx[6]		
				5: RAM_Rd_xx[5]		
				4: RAM_Rd_xx[4]		
				3: RAM_Rd_xx[3]		
				2: RAM_Rd_xx[2]		
				1: RAM_Rd_xx[1]		
				0: RAM_Rd_xx[0]		

#### 50h-6Fh.Bit7-0 RAM\_Rd\_xx[7:0]

RAM\_Rd 機能を用いて RAM からリードしたデータを格納するレジスタです。RAM\_RdAdrs\_H,L レジスタ、RAM\_RdCount レジスタを設定し、RAM\_RdControl レジスタのビットを用いて RAM\_Rd 機能を開始して下さい。本レジスタの値が有効になると FIFO\_IntaStat.RAM\_RdCmp ビットが”1”にセットされます。RAM\_RdCount レジスタに設定した値が 32 バイト未満の場合、RAM からリードしたデータは RAM\_Rd\_00 から順に格納されます。RAM\_RdCount レジスタに設定したカウント数以降のレジスタの値(例えば、カウント設定が”16”の場合、RAM\_Rd\_10~RAM\_Rd\_1F)は無効となります。

## 2. レジスタ

### 2.6.76. 071h DMA0\_Config (DMA0 Config)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	071h	DMA0_Config	R / W	7: FreeRun	0: Count mode	1: FreeRun mode
			R / W	6: DMA_Mode	0: Normal mode	1: Address Decode mode
				5:	0:	1:
				4:	0:	1:
			R / W	3: ActiveDMA	0: DMA0 Inactive	1: DMA0 Active
				2:	0:	1:
			R / W	1: ReqAssertCount[1]	Request Assert Count	
				0: ReqAssertCount[0]		
						00h

DMA0 の動作モードを設定します。

#### Bit7 FreeRun

DMA0 の動作モードを設定します。

0 : カウントモード

1 : フリーランモード

#### Bit6 DMA\_Mode

DMA0 のモードを設定します。

0 : ホストからの XDACK をアクノリッジとして DMA 動作します。

1 : ホストからの DMA0\_RdData/DMA0\_WrData レジスタに対するアクセスをアクノリッジとして DMA 動作します。

#### Bit5-4 Reserved

#### Bit3 ActiveDMA

DMA0 のアクセスを有効にします。

0 : DMA0 アクセス無効

1 : DMA0 アクセス有効

#### Bit2 Reserved

#### Bit1-0 ReqAssertCount[1:0]

CPU のバーストリード／バーストライトに対応するための、REQ アサートカウントオプション設定ビットです。

XDREQ0 のアサートカウント数(転送バイト数)を設定します。設定されたアサートカウント数以上の書き込み可能な空き領域／読み出し可能なデータが FIFO にある場合に XDREQ0 をアサートします。

設定されたアサートカウント数の DMA 転送を終了すると一旦 XDREQ0 をネゲートし、再度アサートカウント数以上の空き領域／データを確認した時点で XDREQ0 をアサートします。

つまり、一回の XDREQ0 のアサートに対し、設定されたアサートカウント数分の転送を保証します。

ただし、カウントモードに設定され、且つ DMA0\_Count\_HH,HL,LH,LL の残りカウント数が設定されたアサートカウント数より小さい場合は DMA0\_Count\_HH,HL,LH,LL のカウント数が優先され、DMA0\_Count\_HH,HL,LH,LL のカウント数以上の空き領域／データが FIFO にある場合に XDREQ0 をアサートします。

下表に DMA0\_Count\_HH,HL,LH,LL(表では Count)、ReqAssertCount(表では Req)、FIFO の空き領域／データ(表では Ready)と XDREQ0 信号及び転送可能数の関係を示します。

DMA0\_Count\_HH,HL,LH,LL の残りカウントが"1"以上である事が必須条件です。

	Count>=Req		Count<Req	
	Ready>=Req	Ready<Req	Ready>=Count	Ready<Count
XDREQ0	アサート	ネゲート	アサート	ネゲート
転送可能数	Req	-	Req	-

ReqAssertCount[1:0]	モード
	16bit mode
0b00	Normal
0b01	16Byte(8Count)
0b10	32Byte(16Count)
0b11	64Byte(32Count)

00(Normal)設定時には REQ アサートカウントオプション未使用となります。

## 2. レジスタ

### 2.6.77. 072h DMA0\_Control (DMA0 Control)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	072h	DMA0_Control	R	7: DMA_Running	0: DMA is not running      1: DMA is running	00h
				6:	0:                              1:	
				5:	0:                              1:	
			W	4: CounterClr	0: Do nothing              1: Clear DMA counter	
			R / W	3: Dir	0: CPU-IF → FIFO RAM    1: CPU-IF ← FIFO RAM	
				2:	0:                              1:	
			W	1: DMA_Stop	0: Do nothing              1: Finish DMA	
			W	0: DMA_Go	0: Do nothing              1: Start DMA	

DMA0 の制御、及び状態を表示します。

#### Bit7 DMA\_Running

DMA0 の転送中、このビットが"1"にセットされます。このビットが"1"である間は、AREAx{x=0-5}Join\_0.JoinDMA0、MediaFIFO\_Join.JoinDMA0 ビットを書き換えることはできません。

Bit6-5 Reserved

#### Bit4 CounterClr

このビットに"1"をセットすると、DMA0\_Count\_HH,HL,LH,LL レジスタが 0x00 にクリアされます。DMA\_Running ビットが"1"である時は、このビットへの書き込みは無視されます。

#### Bit3 Dir

DMA0 の転送方向を設定します。

0: CPU-IF → FIFO RAM(DMA ライト)

1: CPU-IF ← FIFO RAM(DMA リード)

#### Bit2 Reserved

#### Bit1 DMA\_Stop

このビットに"1"をセットすると、DMA0 の転送を終了します。DMA0 の転送を停止すると、DMA\_Running ビットを"0"にクリアします。また、CPU\_IntStat レジスタの DMA0\_Cmp ビットに"1"をセットします。DMA0 の転送を再開する場合、DMA\_Running ビットまたは DMA0\_Cmp ビットを確認し、DMA が終了するのを待って行って下さい。

#### Bit0 DMA\_Go

このビットを"1"にセットすると、DMA0 の転送を開始します。

**2.6.78. 074h DMA0\_Remain\_H (DMA0 FIFO Remain High)****2.6.79. 075h DMA0\_Remain\_L (DMA0 FIFO Remain Low)**

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	074h	DMA0_Remain_H		7:	0:	1:
				6:	0:	1:
				5:	0:	1:
			R	4: DMA_Remain[12]	DMA FIFO Remain High	00h
				3: DMA_Remain[11]		
				2: DMA_Remain[10]		
				1: DMA_Remain[9]		
				0: DMA_Remain[8]		

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	075h	DMA0_Remain_L	R	7: DMA_Remain[7]	DMA FIFO Remain Low	00h
				6: DMA_Remain[6]		
				5: DMA_Remain[5]		
				4: DMA_Remain[4]		
				3: DMA_Remain[3]		
				2: DMA_Remain[2]		
				1: DMA_Remain[1]		
				0: DMA_Remain[0]		

**074h.Bit7-5                      Reserved****074h.Bit4-0, 075h.Bit7-0      DMA\_Remain[12:0]**

読み出しの場合、AREAx{x=0-5}Join\_0.JoinDMA0 ビットによって DMA に接続しているエンドポイント FIFO、または MediaFIFO\_Join.JoinDMA0 ビットによって DMA に接続している MediaFIFO の FIFO 内の残りデータ数を示します。

書き込みの場合、AREAx{x=0-5}Join\_0.JoinDMA0 ビットによって DMA に接続しているエンドポイントの FIFO、または MediaFIFO\_Join.JoinDMA0 ビットによって DMA に接続している MediaFIFO の空き容量を示します。DMA 書き込みを行った直後にはこのレジスタで正しい FIFO の空き容量を参照する事が出来ません。1CPU サイクル以上の間隔を空けて FIFO の空き容量を確認してください。

このレジスタを読み出す場合は DMA0\_Remain\_H,L の順に読み出してください。

## 2. レジスタ

### 2.6.80. 078h DMA0\_Count\_HH (DMA0 Transfer Byte Counter High/High)

### 2.6.81. 079h DMA0\_Count\_HL (DMA0 Transfer Byte Counter High/Low)

### 2.6.82. 07Ah DMA0\_Count\_LH (DMA0 Transfer Byte Counter Low/High)

### 2.6.83. 07Bh DMA0\_Count\_LL (DMA0 Transfer Byte Counter Low/Low)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	078h	DMA0_Count_HH	R / W	7: DMA_Count[31]	DMA Transfer Byte Counter High-High	00h
				6: DMA_Count[30]		
				5: DMA_Count[29]		
				4: DMA_Count[28]		
				3: DMA_Count[27]		
				2: DMA_Count[26]		
				1: DMA_Count[25]		
				0: DMA_Count[24]		

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	079h	DMA0_Count_HL	R / W	7: DMA_Count[23]	DMA Transfer Byte Counter High-Low	00h
				6: DMA_Count[22]		
				5: DMA_Count[21]		
				4: DMA_Count[20]		
				3: DMA_Count[19]		
				2: DMA_Count[18]		
				1: DMA_Count[17]		
				0: DMA_Count[16]		

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	07Ah	DMA0_Count_LH	R / W	7: DMA_Count[15]	DMA Transfer Byte Counter Low-High	00h
				6: DMA_Count[14]		
				5: DMA_Count[13]		
				4: DMA_Count[12]		
				3: DMA_Count[11]		
				2: DMA_Count[10]		
				1: DMA_Count[9]		
				0: DMA_Count[8]		

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	07Bh	DMA0_Count_LL	R / W	7: DMA_Count[7]	DMA Transfer Byte Counter Low-Low	00h
				6: DMA_Count[6]		
				5: DMA_Count[5]		
				4: DMA_Count[4]		
				3: DMA_Count[3]		
				2: DMA_Count[2]		
				1: DMA_Count[1]		
				0: DMA_Count[0]		

カウントモード時に DMA0 の、転送データ長をバイト単位で設定します。最大 0xFFFF\_FFFF バイトまで設定可能です。設定された値からダウンカウントします。本レジスタに転送数を設定した後、DMA0\_Control.DMA\_Go ビットに"1"をセットして DMA 転送を起動してください。本レジスタに設定された転送 Byte 数の転送が終了すると、DMA 転送は終了します。

フリーランモードの場合、設定された値からカウントアップします。DMA0\_Count\_HH,HL,LH,LL レ

レジスタの値がオーバーフローすると、CPU\_IntStat レジスタの DMA0\_CountUp ビットに"1"をセットします。オーバーフロー後もカウントは継続されます。このモードでは、DMA の転送数が参照できます。

DMA 書き込みを行った直後は、このレジスタで正確なカウント数を確認することは出来ません。1CPU サイクル以上の間隔をあけて、カウント数を確認してください。このレジスタをリードする場合は DMA0\_Count\_HH,HL,LH,LL の順に読み出してください。

## 2. レジスタ

### 2.6.84. 07Ch DMA0\_RdData\_0 (DMA0 Read Data 0)

### 2.6.85. 07Dh DMA0\_RdData\_1 (DMA0 Read Data 1)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	07Ch	DMA0_RdData_0	R	7: DMA_RdData_0[7]	DMA Read Data 0	XXh
				6: DMA_RdData_0[6]		
				5: DMA_RdData_0[5]		
				4: DMA_RdData_0[4]		
				3: DMA_RdData_0[3]		
				2: DMA_RdData_0[2]		
				1: DMA_RdData_0[1]		
				0: DMA_RdData_0[0]		

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	07Dh	DMA0_RdData_1	R	7: DMA_RdData_1[7]	DMA Read Data 1	XXh
				6: DMA_RdData_1[6]		
				5: DMA_RdData_1[5]		
				4: DMA_RdData_1[4]		
				3: DMA_RdData_1[3]		
				2: DMA_RdData_1[2]		
				1: DMA_RdData_1[1]		
				0: DMA_RdData_1[0]		

### 07Ch.Bit7-0, 07Dh.Bit7-0 DMA\_RdData\_0[7:0], DMA\_RdData\_1[7:0]

DMA0\_Config.DMA\_Mode ビットが”1”にセットされている場合、このレジスタにアクセスする事で

AREAx{x=0-5}Join\_0.JoinDMA0 ビット、または MediaFIFO\_Join.JoinDMA0 ビットによって DMA に接続しているエンドポイント／チャンネル／MediaFIFO の FIFO データを読み出す事が出来ます。このとき、DMA0\_Control.Dir ビットは DMA リードに設定されている必要があります。



**2.6.86. 07Eh DMA0\_WrData\_0 (DMA0 Write Data 0)****2.6.87. 07Fh DMA0\_WrData\_1 (DMA0 Write Data 1)**

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	07Eh	DMA0_WrData_0	W	7: DMA_WrData_0[7]	DMA Write Data 0	XXh
				6: DMA_WrData_0[6]		
				5: DMA_WrData_0[5]		
				4: DMA_WrData_0[4]		
				3: DMA_WrData_0[3]		
				2: DMA_WrData_0[2]		
				1: DMA_WrData_0[1]		
				0: DMA_WrData_0[0]		

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	07Fh	DMA0_WrData_1	W	7: DMA_WrData_1[7]	DMA Write Data 1	XXh
				6: DMA_WrData_1[6]		
				5: DMA_WrData_1[5]		
				4: DMA_WrData_1[4]		
				3: DMA_WrData_1[3]		
				2: DMA_WrData_1[2]		
				1: DMA_WrData_1[1]		
				0: DMA_WrData_1[0]		

**07Eh.Bit7-0, 07Fh.Bit7-0 DMA\_WrData\_0[7:0], DMA\_WrData\_1[7:0]**

DMA0\_Config.DMA\_Mode ビットが”1”にセットされている場合、このレジスタにアクセスする事で

AREAx{x=0-5}Join\_0.JoinDMA0 ビット、または MediaFIFO\_Join.JoinDMA0 ビットによって DMA に接続しているエンドポイント／チャンネル／MediaFIFO の FIFO にデータを書き込む事が出来ます。このとき、DMA0\_Control.Dir ビットは DMA ライトに設定されている必要があります。

## 2. レジスタ

### 2.6.88. 081h DMA1\_Config (DMA1 Config)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	081h	DMA1_Config	R / W	7: FreeRun	0: Count mode	1: FreeRun mode	00h
			R / W	6: DMA_Mode	0: Normal mode	1: Address Decode mode	
				5:	0:	1:	
				4:	0:	1:	
			R / W	3: ActiveDMA	0: DMA1 Inactive	1: DMA1 Active	
				2:	0:	1:	
			R / W	1: ReqAssertCount[1]	Request Assert Count		
				0: ReqAssertCount[0]			

DMA1 の動作モードを設定します。

#### Bit7 FreeRun

DMA1 の動作モードを設定します。

- 0 : カウントモード
- 1 : フリーランモード

#### Bit6 DMA\_Mode

DMA0 のモードを設定します。

- 0 : ホストからの XDACK をアクノリッジとして DMA 動作します。
- 1 : ホストからの DMA0\_RdData/DMA0\_WrData レジスタに対するアクセスをアクノリッジとして DMA 動作します。

#### Bit5-4 Reserved

#### Bit3 ActiveDMA

DMA1 のアクセスを有効にします。

- 0 : DMA1 アクセス無効
- 1 : DMA1 アクセス有効

#### Bit2 Reserved

#### Bit1-0 ReqAssertCount[1:0]

CPU のバーストリード／バーストライトに対応するための、REQ アサートカウントオプション設定ビットです。

XDREQ1 のアサートカウント数(転送バイト数)を設定します。設定されたアサートカウント数以上の書き込み可能な空き領域／読み出し可能なデータが FIFO にある場合に XDREQ1 をアサートします。

設定されたアサートカウント数の DMA 転送を終了すると一旦 XDREQ1 をネゲートし、再度アサートカウント数以上の空き領域／データを確認した時点で XDREQ1 をアサートします。

つまり、一回の XDREQ1 のアサートに対し、設定されたアサートカウント数分の転送を保証します。

ただし、カウントモードに設定され、且つ DMA1\_Count\_HH,HL,LH,LL の残りカウント数が設定されたアサートカウント数より小さい場合は DMA1\_Count\_HH,HL,LH,LL のカウント数が優先され、DMA1\_Count\_HH,HL,LH,LL のカウント数以上の空き領域／データが FIFO にある場合に XDREQ1 をアサートします。

下表に DMA1\_Count\_HH,HL,LH,LL(表では Count)、ReqAssertCount(表では Req)、FIFO の空き領域／データ(表では Ready)と XDREQ1 信号及び転送可能数の関係を示します。

DMA1\_Count\_HH,HL,LH,LL の残りカウントが"1"以上である事が必須条件です。

	Count>=Req		Count<Req	
	Ready>=Req	Ready<Req	Ready>=Count	Ready<Count
XDREQ1	アサート	ネゲート	アサート	ネゲート
転送可能数	Req	-	Req	-

ReqAssertCount[1:0]	モード
	16bit mode
0b00	Normal
0b01	16Byte(8Count)
0b10	32Byte(16Count)
0b11	64Byte(32Count)

00(Normal)設定時には REQ アサートカウントオプション未使用となります。

## 2. レジスタ

### 2.6.89. 082h DMA1\_Control (DMA1 Control)

Mode	Addresses	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	082h	DMA1_Control	R	7: DMA_Running	0: DMA is not running	1: DMA is running	00h
				6:	0:	1:	
				5:	0:	1:	
			W	4: CounterClr	0: Do nothing	1: Clear DMA counter	
			R / W	3: Dir	0: CPU-IF → FIFO RAM	1: CPU-IF ← FIFO RAM	
				2:	0:	1:	
			W	1: DMA_Stop	0: Do nothing	1: Finish DMA	
			W	0: DMA_Go	0: Do nothing	1: Start DMA	

DMA1 の制御、及び状態を表示します。

#### Bit7 DMA\_Running

DMA1 の転送中、このビットが"1"にセットされます。このビットが"1"である間は、AREAx{x=0-5}Join\_0.JoinDMA1 ビット、MediaFIFO\_Join.JoinDMA1 ビットを書き換えることはできません。

#### Bit6-5 Reserved

#### Bit4 CounterClr

このビットに"1"をセットすると、DMA1\_Count\_HH,HL,LH,LL レジスタが 0x00 にクリアされます。DMA\_Running ビットが"1"である時は、このビットへの書き込みは無視されます。

#### Bit3 Dir

DMA1 の転送方向を設定します。

0: CPU-IF → FIFO RAM(DMA ライト)

1: CPU-IF ← FIFO RAM(DMA リード)

#### Bit2 Reserved

#### Bit1 DMA\_Stop

このビットに"1"をセットすると、DMA1 の転送を終了します。DMA1 の転送を停止すると、DMA\_Running ビットを"0"にクリアします。また、CPU\_IntStat レジスタの DMA1\_Cmp ビットに"1"をセットします。DMA1 の転送を再開する場合、DMA\_Running ビットまたは DMA1\_Cmp ビットを確認し、DMA が終了するのを待って行って下さい。

#### Bit0 DMA\_Go

このビットを"1"にセットすると、DMA1 の転送を開始します。

**2.6.90. 084h DMA1\_Remain\_H (DMA1 FIFO Remain High)****2.6.91. 085h DMA1\_Remain\_L (DMA1 FIFO Remain Low)**

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	084h	DMA1_Remain_H		7:	0:	1:
				6:	0:	1:
				5:	0:	1:
			R	4: DMA_Remain[12]	DMA FIFO Remain High	00h
				3: DMA_Remain[11]		
				2: DMA_Remain[10]		
				1: DMA_Remain[9]		
				0: DMA_Remain[8]		

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	085h	DMA1_Remain_L	R	7: DMA_Remain[7]	DMA FIFO Remain Low	00h
				6: DMA_Remain[6]		
				5: DMA_Remain[5]		
				4: DMA_Remain[4]		
				3: DMA_Remain[3]		
				2: DMA_Remain[2]		
				1: DMA_Remain[1]		
				0: DMA_Remain[0]		

**084h.Bit7-5 Reserved****084h.Bit4-0, 085h.Bit7-0 DMA\_Remain[12:0]**

読み出しの場合、AREAx{x=0-5}Join\_0.JoinDMA1 ビットによって DMA に接続しているエンドポイント FIFO、または MediaFIFO\_Join.JoinDMA1 ビットによって DMA に接続している MediaFIFO 内の残りデータ数を示します。

書き込みの場合、AREAx{x=0-5}Join\_0.JoinDMA1 ビットによって DMA に接続しているエンドポイントの FIFO、または MediaFIFO\_Join.JoinDMA1 ビットによって DMA に接続している MediaFIFO の空き容量を示します。DMA 書き込みを行った直後にはこのレジスタで正しい FIFO の空き容量を参照する事が出来ません。1CPU サイクル以上の間隔を空けて FIFO の空き容量を確認してください。

このレジスタを読み出す場合は DMA1\_Remain\_H,L の順に読み出してください。

## 2. レジスタ

### 2.6.92. 088h DMA1\_Count\_HH (DMA1 Transfer Byte Counter High/High)

### 2.6.93. 089h DMA1\_Count\_HL (DMA1 Transfer Byte Counter High/Low)

### 2.6.94. 08Ah DMA1\_Count\_LH (DMA1 Transfer Byte Counter Low/High)

### 2.6.95. 08Bh DMA1\_Count\_LL (DMA1 Transfer Byte Counter Low/Low)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	088h	DMA1_Count_HH	R / W	7: DMA_Count[31]	DMA Transfer Byte Counter High-High	00h
				6: DMA_Count[30]		
				5: DMA_Count[29]		
				4: DMA_Count[28]		
				3: DMA_Count[27]		
				2: DMA_Count[26]		
				1: DMA_Count[25]		
				0: DMA_Count[24]		

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	089h	DMA1_Count_HL	R / W	7: DMA_Count[23]	DMA Transfer Byte Counter High-Low	00h
				6: DMA_Count[22]		
				5: DMA_Count[21]		
				4: DMA_Count[20]		
				3: DMA_Count[19]		
				2: DMA_Count[18]		
				1: DMA_Count[17]		
				0: DMA_Count[16]		

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	08Ah	DMA1_Count_LH	R / W	7: DMA_Count[15]	DMA Transfer Byte Counter Low-High	00h
				6: DMA_Count[14]		
				5: DMA_Count[13]		
				4: DMA_Count[12]		
				3: DMA_Count[11]		
				2: DMA_Count[10]		
				1: DMA_Count[9]		
				0: DMA_Count[8]		

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	08Bh	DMA1_Count_LL	R / W	7: DMA_Count[7]	DMA Transfer Byte Counter Low-Low	00h
				6: DMA_Count[6]		
				5: DMA_Count[5]		
				4: DMA_Count[4]		
				3: DMA_Count[3]		
				2: DMA_Count[2]		
				1: DMA_Count[1]		
				0: DMA_Count[0]		

#### 088h-08Bh.Bit7-0 DMA\_Count[31:0]

カウントモード時に DMA1 の、転送データ長をバイト単位で設定します。最大 0xFFFF\_FFFF バイトまで設定可能です。設定された値からダウンカウントします。本レジスタに転送数を設定した後、DMA1\_Control.DMA\_Go ビットに"1"をセットして DMA 転送を起動してください。本レジスタに設定された転送 Byte 数の転送が終了すると、DMA 転送は終了します。

フリーランモードの場合、設定された値からカウントアップします。

DMA1\_Count\_HH,HL,LH,LL レジスタの値がオーバーフローすると、CPU\_IntStat レジスタの DMA1\_CountUp ビットに"1"をセットします。オーバーフロー後もカウントは継続されます。このモードでは、DMA の転送数が参照できます。

DMA 書き込みを行った直後は、このレジスタで正確なカウント数を確認することは出来ません。1CPU サイクル以上の間隔をあけて、カウント数を確認してください。このレジスタをリードする場合は DMA1\_Count\_HH,HL,LH,LL の順に読み出してください。

## 2. レジスタ

### 2.6.96. 08Ch DMA1\_RdData\_0 (DMA1 Read Data 0)

### 2.6.97. 08Dh DMA1\_RdData\_1 (DMA1 Read Data 1)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	08Ch	DMA1_RdData_0	R	7: DMA_RdData_0[7]	DMA Read Data 0	XXh
				6: DMA_RdData_0[6]		
				5: DMA_RdData_0[5]		
				4: DMA_RdData_0[4]		
				3: DMA_RdData_0[3]		
				2: DMA_RdData_0[2]		
				1: DMA_RdData_0[1]		
				0: DMA_RdData_0[0]		

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	08Dh	DMA1_RdData_1	R	7: DMA_RdData_1[7]	DMA Read Data 1	XXh
				6: DMA_RdData_1[6]		
				5: DMA_RdData_1[5]		
				4: DMA_RdData_1[4]		
				3: DMA_RdData_1[3]		
				2: DMA_RdData_1[2]		
				1: DMA_RdData_1[1]		
				0: DMA_RdData_1[0]		

#### 08Ch.Bit7-0, 08Dh.Bit7-0 DMA\_RdData\_0[7:0], DMA\_RdData\_1[7:0]

DMA1\_Config.DMA\_Mode ビットが”1”にセットされている場合、このレジスタにアクセスする事で AREAx{x=0-5}Join\_0.JoinDMA1 ビット、または MediaFIFO\_Join.JoinDMA1 ビットによって DMA に接続しているエンドポイント／チャンネル／MediaFIFO の FIFO データを読み出す事が出来ます。このとき、DMA1\_Control.Dir ビットは DMA リードに設定されている必要があります。



**2.6.98. 08Eh DMA1\_WrData\_0 (DMA1 Write Data 0)****2.6.99. 08Fh DMA1\_WrData\_1 (DMA1 Write Data 1)**

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	08Eh	DMA1_WrData_0	W	7: DMA_WrData_0[7]	DMA Write Data 0	XXh
				6: DMA_WrData_0[6]		
				5: DMA_WrData_0[5]		
				4: DMA_WrData_0[4]		
				3: DMA_WrData_0[3]		
				2: DMA_WrData_0[2]		
				1: DMA_WrData_0[1]		
				0: DMA_WrData_0[0]		

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	08Fh	DMA1_WrData_1	W	7: DMA_WrData_1[7]	DMA Write Data 1	XXh
				6: DMA_WrData_1[6]		
				5: DMA_WrData_1[5]		
				4: DMA_WrData_1[4]		
				3: DMA_WrData_1[3]		
				2: DMA_WrData_1[2]		
				1: DMA_WrData_1[1]		
				0: DMA_WrData_1[0]		

**08Eh.Bit7-0, 08Fh.Bit7-0 DMA\_WrData\_0[7:0], DMA\_WrData\_1[7:0]**

DMA1\_Config.DMA\_Mode ビットが”1”にセットされている場合、このレジスタにアクセスする事で AREAx{x=0-5}Join\_0.JoinDMA1 ビット、または MediaFIFO\_Join.JoinDMA1 ビットによって DMA に接続しているエンドポイント／チャネル／MediaFIFO の FIFO にデータを書き込む事が出来ます。このとき、DMA1\_Control.Dir ビットは DMA ライトに設定されている必要があります。

## 2. レジスタ

---

### 2.6.100. 090h IDE\_Status (IDE Status)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	090h	IDE_Status	R	7: DMARQ	0:HDMARQ Not Asserted    1:HDMARQ Asserted	00h
			R	6: DMACK	0:XHDMACK Not Asserted    1:XHDMACK Asserted	
			R	5: INTRQ	0:HINTRQ Not Asserted    1:HINTRQ Asserted	
			R	4: IORDY	0:HIORDY Not Asserted    1:HIORDY Asserted	
				3:	0:                            1:	
				2:	0:                            1:	
			R	1: PDIAG	0:xHPDIAG Not Asserted    1:xHPDIAG Asserted	
			R	0: DASP	0:xHDASP Not Asserted    1:xHDASP Asserted	

IDE バスの信号状態を表示します。各信号がアサートされている時に”1”を読み出すことができます。  
XHDMACK、XHPDIAG、XHDASP の負論理信号は、電圧レベルが”0”の時に”1”が読み出されることに注意してください。

Bit3-2 はリザーブで常に”0”が読み出されます。

## 2.6.101. 091h IDE\_Control (IDE Control)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	091h	IDE_Control		7:	0:	1:	00h
			W	6: IDE_Clr	0: None	1: Clear IDE Circuit	
				5:	0:	1:	
				4:	0:	1:	
			R/W	3: Dir	0: IDE → FIFO RAM	1: IDE ← FIFO RAM	
				2:	0:	1:	
				1:	0:	1:	
			R/W	0: IDE_Go	0: None	1: IDE DMA Go	

IDE の DMA を制御します。

**Bit7**      **Reserved**

**Bit6**      **IDE\_Clr**

このビットに"1"をセットすると、IDE 回路が初期状態に戻ります。設定したレジスタの内容は変わりません。IDE の DMA 中やレジスタアクセスのシーケンス中に本ビットをセットしてはいけません。

**Bit5-4**    **Reserved**

**Bit3**      **Dir**

IDE の転送方向を設定します。

0: IDE → FIFO RAM(IDE リード)

1: IDE ← FIFO RAM(IDE ライト)

**Bit2-1**    **Reserved**

**Bit0**      **IDE\_Go**

このビットに"1"をセットすると IDE の DMA が起動され、終了すると IDE\_IntStat レジスタの IDE\_Cm p ビットが"1"になります。

DMA の転送中はこのビットが"1"にセットされて、終了するとこのビットは"0"に戻ります。

このビットが"1"である間に"0"を書き込むと実行中の DMA 転送は中断されて終了しますが、IDE\_IntStat レジスタの IDE\_Cm p ビットはセットされません。

## 2. レジスタ

### 2.6.102. 092h IDE\_Config\_0 (IDE Configuration 0)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	092h	IDE_Config_0	R/W	7: IDE_BusReset	0: None	1: XHRESET Asserted	00h
			R/W	6: IDE_LongBusReset	0: None	1: XHRESET Asserted	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
			R/W	1: Ultra	0: Non Ultra mode	1: Ultra mode	
			R/W	0: DMA	0: Non DMA mode	1: DMA mode	

IDE の DMA を制御します。

#### Bit7 IDE\_BusReset

このビットに"1"をセットすると、IDE の xHRESET 信号が 50  $\mu$ s アサートされます。このビットに"1"が表示されている間に再度このビットに"1"をセットすると、そこからさらに 50  $\mu$ s 間 IDE の XHRESET 信号がアサートされます。Bit7/Bit6 どちらかの XHRESET 信号アサートでこのビットには"1"が読み出されます。

#### Bit6 IDE\_LongBusReset

このビットに"1"をセットすると、IDE の xHRESET 信号が 400  $\mu$ s アサートされます。このビットに"1"が表示されている間に再度このビットに"1"をセットすると、そこからさらに 400  $\mu$ s 間 IDE の XHRESET 信号がアサートされます。Bit7/Bit6 どちらかの XHRESET 信号アサートでこのビットには"1"が読み出されます。

#### Bit5-2 Reserved

#### Bit1 Ultra

このビットと Bit0:DMA に同時に"1"をセットすると、IDE\_Control レジスタで起動される IDE の DMA が Ultra モードになります。

DMA の転送中はこのビットを書き換えてはいけません。下表にこのビットにより設定される IDE の DMA の転送モードを示します。

#### Bit0 DMA

このビットに"1"をセットすると、IDE\_Control レジスタで起動される IDE の DMA がマルチワード DMA モードになります。

DMA の転送中はこのビットを書き換えてはいけません。下表にこのビットにより設定される IDE の DMA の転送モードを示します。

Bit1-0	"00"	"01"	"10"	"11"
	PIO	Multiword DMA	設定禁止	Ultra

## 2.6.103. 093h IDE\_Config\_1 (IDE Configuration 1)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	093h	IDE_Config_1	R/W	7: ActiveIDE	0: InActivated IDE Bus	1: Activate IDE Bus	04h
			R/W	6: DelayStrobe	0: Not Delay Strobe Signal	1: Delay Strobe Signal	
				5:	0:	1:	
			R/W	4: InterLock	0: None	1: DMA InterLock	
				3:	0:	1:	
			R/W	2: Swap	0: Data Swap	1: None	
				1:	0:	1:	
				0:	0:	1:	

IDE バスの状態を制御します。

**Bit7 ActiveIDE**

このビットに"1"をセットすると、IDE の出力信号が有効になります。IDE バスへのレジスタリードライトコマンド発行や、IDE-DMA を実行するにはあらかじめこのビットに"1"をセットしておく必要があります。このビットが"0"の時は、IDE 信号は全て入力状態になっています。

**Bit6 DelayStrobe**

このビットに"1"をセットすると、IDE-DMA のマルチワード DMA 転送時、XHDMACK アサートから XHIOR/XHIOW のストロブ信号アサートまで 2 システムクロック間(約 33ns)セットアップ時間を確保します。このビットが"0"の時はIDE-DMA のマルチワードDMA 転送時、XHDMACK アサートと XHIOR/XHIOW のストロブ信号アサートは同時 (約 0ns)になります。

**Bit5 Reserved****Bit4 InterLock**

このビットに"1"をセットすると、IDE-DMA のマルチワード DMA 転送時、LSI 内部でデータが転送できなくなったことによる要因では XHDMACK をネゲートせずに、IDE バスを保持したまま内部のデータが準備できるまで待ちます。このビットが"0"の時は、内部のデータが準備できない時は一旦 XHDMACK を開放します。

**Bit3 Reserved****Bit2 Swap**

このビットを"0"にクリアすると、IDEバスのデータ上位 8 ビットと下位 8 ビットが反転して入出力されます。通常は、このビットが"1"にセットされているようにして下さい。Swapビットの詳細につきまして、「Appendix A IDE\_Config\_1.Swapビット設定」を参照して下さい。

**Bit1-0 Reserved**

## 2. レジスタ

### 2.6.104. 094h IDE\_Rmod (IDE Register Mode)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	094h	IDE_Rmod	R/W	7: RegisterAssertPulseWidth[3]	Register Assert Pulse Width	00h
				6: RegisterAssertPulseWidth[2]		
				5: RegisterAssertPulseWidth[1]		
				4: RegisterAssertPulseWidth[0]		
			R/W	3: RegisterNegatePulseWidth[3]	Register Negate Pulse Width	
				2: RegisterNegatePulseWidth[2]		
				1: RegisterNegatePulseWidth[1]		
				0: RegisterNegatePulseWidth[0]		

IDE バスへのレジスタモードでのアクセス時 XHIOR/XHIOW アサート・ネゲート時のストロブ幅を設定します。

IDE の転送モードに合わせた適切な値を選択する必要があります。

#### Bit7-4 RegisterAssertPulseWidth[3:0]

システムクロック(60MHz)周期の [RegisterAssertPulseWidth + 4] 倍になります。

ex 0000:  $4 \times 16.67\text{nS} = 67\text{nS}$

0001:  $5 \times 16.67\text{nS} = 83\text{nS}$

#### Bit3-0 RegisterNegatePulseWidth[3:0]

システムクロック(60MHz)周期の [RegisterNegatePulseWidth + 4] 倍になります。

ex 0000:  $4 \times 16.67\text{nS} = 67\text{nS}$

0001:  $5 \times 16.67\text{nS} = 83\text{nS}$

## 2.6.105. 095h IDE\_Tmod (IDE Transfer Mode)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	095h	IDE_Tmod	R/W	7: TransferAssertPulseWidth[3]	Transfer Assert Pulse Width	00h
				6: TransferAssertPulseWidth[2]		
				5: TransferAssertPulseWidth[1]		
				4: TransferAssertPulseWidth[0]		
			R/W	3: TransferNegatePulseWidth[3]	Transfer Negate Pulse Width	
				2: TransferNegatePulseWidth[2]		
				1: TransferNegatePulseWidth[1]		
				0: TransferNegatePulseWidth[0]		

IDE バスへの PIO モードアクセス時 XHIOR/XHIOW アサート・ネゲート時のストロブ幅を設定します。

IDE の転送モードに合わせた適切な値を選択する必要があります。

**Bit7-4 TransferAssertPulseWidth[3:0]**

システムクロック(60MHz)周期の [TransferAssertPulseWidth + 4] 倍になります。

ex 0000:  $4 \times 16.67\text{nS} = 67\text{nS}$

0001:  $5 \times 16.67\text{nS} = 83\text{nS}$

**Bit3-0 TransferNegatePulseWidth[3:0]**

システムクロック(60MHz)周期の [TransferNegatePulseWidth + 4] 倍になります。

ex 0000:  $4 \times 16.67\text{nS} = 67\text{nS}$

0001:  $5 \times 16.67\text{nS} = 83\text{nS}$

## 2. レジスタ

### 2.6.106. 096h IDE\_Umod (IDE Ultra-DMA Transfer Mode)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	096h	IDE_Umod		7:	0:	1:	00h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
			R/W	3: UltraDMA_Cycle[3]	UltraDMA_Cycle		
				2: UltraDMA_Cycle[2]			
				1: UltraDMA_Cycle[1]			
				0: UltraDMA_Cycle[0]			

IDE バスへの Ultra モードアクセス時のアクセスサイクル幅を設定します。

IDE の転送モードに合わせた適切な値を選択する必要があります。

**Bit7-4**    **Reserved**

**Bit3-0**    **UltraDMA\_Cycle[3:0]**

システムクロック(60MHz)周期の [UltraDMA\_Cycle + 2] 倍になります。

ex    0000:  $2 \times 16.67\text{nS} = 33\text{nS}$

      0001:  $3 \times 16.67\text{nS} = 50\text{nS}$



## 2.6.107. 09Ah IDE\_CRC\_H (IDE CRC High)

## 2.6.108. 09Bh IDE\_CRC\_L (IDE CRC Low)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	09Ah	IDE_CRC_H	R	7: IDE_CRC[15]	IDE_CRC[15:8]	00h
				6: IDE_CRC[14]		
				5: IDE_CRC[13]		
				4: IDE_CRC[12]		
				3: IDE_CRC[11]		
				2: IDE_CRC[10]		
				1: IDE_CRC[9]		
				0: IDE_CRC[8]		

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	09Bh	IDE_CRC_L	R	7: IDE_CRC[7]	IDE_CRC[7:0]	00h
				6: IDE_CRC[6]		
				5: IDE_CRC[5]		
				4: IDE_CRC[4]		
				3: IDE_CRC[3]		
				2: IDE_CRC[2]		
				1: IDE_CRC[1]		
				0: IDE_CRC[0]		

IDE の Ultra モードでの DMA 転送時の CRC 計算結果を逐次表示します。読み出す場合は、IDE\_CRC\_H と IDE\_CRC\_L レジスタを対でアクセスする必要があります。

## 2. レジスタ

### 2.6.109. 09Dh IDE\_Count\_H (IDE Transfer Byte Counter High)

### 2.6.110. 09Eh IDE\_Count\_M (IDE Transfer Byte Counter Middle)

### 2.6.111. 09Fh IDE\_Count\_L (IDE Transfer Byte Counter Low)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	09Dh	IDE_Count_H	R/W	7: IDE_Count[23]	IDE_Count[23:16]	00h
				6: IDE_Count[22]		
				5: IDE_Count[21]		
				4: IDE_Count[20]		
				3: IDE_Count[19]		
				2: IDE_Count[18]		
				1: IDE_Count[17]		
				0: IDE_Count[16]		

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	09Eh	IDE_Count_M	R/W	7: IDE_Count[15]	IDE_Count[15:8]	00h
				6: IDE_Count[14]		
				5: IDE_Count[13]		
				4: IDE_Count[12]		
				3: IDE_Count[11]		
				2: IDE_Count[10]		
				1: IDE_Count[9]		
				0: IDE_Count[8]		

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	09Fh	IDE_Count_L	R/W	7: IDE_Count[7]	IDE_Count[7:1]	00h
				6: IDE_Count[6]		
				5: IDE_Count[5]		
				4: IDE_Count[4]		
				3: IDE_Count[3]		
				2: IDE_Count[2]		
				1: IDE_Count[1]		
				0:	0: 1:	

IDE の DMA 転送時の転送バイト数を設定します。このレジスタ設定が 0 バイトで DMA を起動した場合は起動が無視されます。読み出す場合は、IDE\_Count\_H と IDE\_Count\_M と IDE\_Count\_L レジスタを対でアクセスする必要があります。その際に IDE\_CRC\_H レジスタを先にアクセスして下さい。尚、IDE\_Count\_L レジスタの最下位ビットは常に”0”が表示されます。

## 2.6.112. 0A0h IDE\_RegAdrs (IDE Register Address)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	0A0h	IDE_RegAdrs	R/W	7: IDE_WrReg	0: None	1: IDE Register Write Go	00h
			R/W	6: IDE_RdReg	0: None	1: IDE Register Read Go	
				5:	0:	1:	
				4:	0:	1:	
			R/W	3: IDE_RegAddress[3]	IDE_RegAddress[3:0]		
				2: IDE_RegAddress[2]			
				1: IDE_RegAddress[1]			
				0: IDE_RegAddress[0]			

CPU による IDE バスへのレジスタアクセスを制御します。

**Bit7 IDE\_WrReg**

このビットに"1"をセットすると、あらかじめセットしてある IDE\_WrRegValue\_0,1 レジスタの内容で、IDE バスに対して PIO モードまたはレジスタモードで IDE レジスタをライトします。動作中はこのビットに"1"が表示され、動作が終了すると IDE\_IntStat レジスタの IDE\_RegCmp ビットがセットされて、このビットは"0"に戻ります。IDE バスへのアドレスはあらかじめ、または同時に IDE\_RegAddress にセットする必要があります。あらかじめ IDE\_Rmod、IDE\_Tmod レジスタを適切なモードに設定しておく必要があります。

**Bit6 IDE\_RdReg**

このビットに"1"をセットすると、IDE バスに対して PIO モードまたはレジスタモードで IDE レジスタをリードして、リードした値を IDE\_RdRegValue\_0,1 レジスタにセットします。動作中はこのビットに"1"が表示され、動作が終了すると IDE\_IntStat レジスタの IDE\_RegCmp ビットがセットされて、このビットは"0"に戻ります。IDE バスへのアドレスはあらかじめ、または同時に IDE\_RegAddress にセットする必要があります。あらかじめ IDE\_Rmod、IDE\_Tmod レジスタを適切なモードに設定しておく必要があります。

**Bit5-4 Reserved****Bit3-0 IDE\_RegAddress[3:0]**

IDE\_WrReg、IDE\_RdReg ビットによる IDE バスへのレジスタアクセス時のアドレスを設定します。IDE バスに出力されるアドレスの対応は下表の通りです。

IDE_RegAddress[3]	0:XHCS0=0	1:XHCS1=0
IDE_RegAddress[2]	0:HDA2=0	1:HDA2=1
IDE_RegAddress[1]	0:HDA1=0	1:HDA1=1
IDE_RegAddress[0]	0:HDA0=0	1:HDA0=1

## 2. レジスタ

### 2.6.113. 0A2h IDE\_RdRegValue\_0 (IDE Register Read Value 0)

### 2.6.114. 0A3h IDE\_RdRegValue\_1 (IDE Register Read Value 1)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	0A2h	IDE_RdRegValue_0	R	7: IDE_RdRegValue[15]	IDE RdRegValue 0	00h
				6: IDE_RdRegValue[14]		
				5: IDE_RdRegValue[13]		
				4: IDE_RdRegValue[12]		
				3: IDE_RdRegValue[11]		
				2: IDE_RdRegValue[10]		
				1: IDE_RdRegValue[9]		
				0: IDE_RdRegValue[8]		

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	0A3h	IDE_RdRegValue_1	R	7: IDE_RdRegValue[7]	IDE RdRegValue1	00h
				6: IDE_RdRegValue[6]		
				5: IDE_RdRegValue[5]		
				4: IDE_RdRegValue[4]		
				3: IDE_RdRegValue[3]		
				2: IDE_RdRegValue[2]		
				1: IDE_RdRegValue[1]		
				0: IDE_RdRegValue[0]		

IDE\_RegAdrs レジスタの IDE\_RdReg ビットによる IDE バスに対する IDE レジスタのリードで、リードした値がこのレジスタにセットされます。また、IDE\_RegConfig レジスタの自動ステータスレジスタリードでもリードした値がこのレジスタにセットされます。読み出す場合は、IDE\_RdRegValue\_0 と IDE\_RdRegValue\_1 レジスタを対でアクセスする必要があります。

## 2.6.115. 0A4h IDE\_WrRegValue\_0 (IDE Register Write Value 0)

## 2.6.116. 0A5h IDE\_WrRegValue\_1 (IDE Register Write Value 1)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	0A4h	IDE_WrRegValue_0	R/W	7: IDE_WrRegValue[15]	IDE WrRegValue 0	00h
				6: IDE_WrRegValue[14]		
				5: IDE_WrRegValue[13]		
				4: IDE_WrRegValue[12]		
				3: IDE_WrRegValue[11]		
				2: IDE_WrRegValue[10]		
				1: IDE_WrRegValue[9]		
				0: IDE_WrRegValue[8]		

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	0A5h	IDE_WrRegValue_1	R/W	7: IDE_WrRegValue[7]	IDE WrRegValue 1	00h
				6: IDE_WrRegValue[6]		
				5: IDE_WrRegValue[5]		
				4: IDE_WrRegValue[4]		
				3: IDE_WrRegValue[3]		
				2: IDE_WrRegValue[2]		
				1: IDE_WrRegValue[1]		
				0: IDE_WrRegValue[0]		

IDE\_RegAdrs レジスタの IDE\_WrReg ビットによる CPU の IDE バスに対する IDE レジスタのライトで、ライトすべきデータをあらかじめここにセットします。

## 2. レジスタ

### 2.6.117. 0A6h IDE\_SeqWrRegControl (IDE Sequential Register Write Control)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device  / Host	0A6h	IDE_SeqWrRegControl	R/W	7: IDE_SeqWrReg	0:	1: IDE Sequence Write Go	00h
			W	6: IDE_SeqWrRegClr	0:	1: Clear IDE Sequence Write	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
				0:	0:	1:	

CPU による IDE バスへのレジスタへのシーケンスライト動作を制御します。

#### Bit7 IDE\_SeqWrReg

このビットに"1"をセットすると、あらかじめ IDE\_SeqWrRegAdrs / IDE\_SeqWrRegValue にセットされた最大 16 組のアドレス・データが、セットした順に IDE バスへレジスタライトされて、終了すると IDE\_IntStat レジスタの IDE\_SeqWrRegCmp ビットが"1"になります。

シーケンス動作中はこのビットが"1"にセットされて、終了するとこのビットは"0"に戻ります。

#### Bit6 IDE\_SeqWrRegClr

このビットに"1"をセットすると、あらかじめ IDE\_SeqWrRegAdrs / IDE\_SeqWrRegValue にセットされた最大 16 組のアドレス・データを破棄して初期状態に戻すことができます。シーケンス動作中に本ビットをセットしてはいけません。

#### Bit5-0 Reserved

## 2.6.118. 0A7h IDE\_SeqWrRegCnt (IDE Sequential Register Write Counter)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	0A7h	IDE_SeqWrRegCnt		7:	0:	1:	00h
				6:	0:	1:	
				5:	0:	1:	
			R	4: IDE_SeqWrRegCnt[4]	IDE_SeqWrRegCnt[4:0]		
				3: IDE_SeqWrRegCnt[3]			
				2: IDE_SeqWrRegCnt[2]			
				1: IDE_SeqWrRegCnt[1]			
	0: IDE_SeqWrRegCnt[0]						

IDE\_SeqWrRegValue レジスタに書き込んだデータ数を表示します。最大 10h までが表示されます。IDE バスへのシーケンスライトが行われるとともに値が減り、IDE\_SeqWrRegValue レジスタに書き込んだ全てのデータが IDE バスに書き込み完了すると”0”に戻ります。IDE\_SeqWrRegControl レジスタの IDE\_SeqWrRegClr ビットへの”1”書き込みでも”0”に戻ります。

## 2. レジスタ

---

### 2.6.119. 0A8h IDE\_SeqWrRegAdrs (IDE Sequential Register Write Address FIFO)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device  / Host	0A8h	IDE_SeqWrRegAdrs		7:	0:	1:	XXh
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
			W	3: IDE_SeqRegAddress[3]	IDE_SeqRegAddress[3:0]		
				2: IDE_SeqRegAddress[2]			
				1: IDE_SeqRegAddress[1]			
				0: IDE_SeqRegAddress[0]			

IDE\_SeqWrRegControl レジスタによる IDE バスへのシーケンスライト動作時、IDE バスへ出力するアドレスを IDE\_SeqWrRegValue レジスタのデータと対でセットします。同じアドレスが連続する場合には、そのアドレスを一度セットしたら再セットする必要はありません。IDE バスに出力されるアドレスとビットとの関係は IDE\_RegAdrs レジスタの IDE\_RegAddress ビットと同じです。



## 2.6.120. 0A9h IDE\_SeqWrRegValue (IDE Sequential Register Write Value FIFO)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	0A9h	IDE_SeqWrRegValue	W	7: IDE_SeqWrRegValue[7]	IDE_SeqWrRegValue[7:0]	XXh
				6: IDE_SeqWrRegValue[6]		
				5: IDE_SeqWrRegValue[5]		
				4: IDE_SeqWrRegValue[4]		
				3: IDE_SeqWrRegValue[3]		
				2: IDE_SeqWrRegValue[2]		
				1: IDE_SeqWrRegValue[1]		
				0: IDE_SeqWrRegValue[0]		

IDE\_SeqWrRegControl レジスタによる IDE バスへのシーケンスライト動作時、IDE バスへ出力するデータを IDE\_SeqWrRegAdrs レジスタのアドレスと対で順にセットします。最大 16 組セットすることが出来、それを超える書き込みは無視されます。IDE\_SeqWrRegAdrs が"0"の時(XHCS=0、HDA=0 のデータポートへのライト)は、IDE に対しては 16 ビットアクセスになりますので、このレジスタを 2 回セット(データは下位バイト・上位バイトの順です)する必要があり、その場合は 16 組中の 2 組が使われます。それ以外のアドレスでは IDE に対しては 8 ビットアクセスになりますので、IDE への 1 回のライトに対してこのレジスタを 1 回セットすることになります。

## 2. レジスタ

---

### 2.6.121. 0ACh IDE\_RegConfig (IDE Register Configuration)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	0ACh	IDE_RegConfig	R/W	7: EnAutoStsRd	0: None	1: Auto Status Read Enable	00h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
				0:	0:	1:	

IDE バスの HINTRQ 割り込みでの自動ステータスリード動作を制御します。

#### Bit7 EnAutoStsRd

このビットに"1"をセットすると、IDE バスの HINTRQ 割り込みが発生した際に IDE バスのステータスレジスタ(XHCS0=0、HDA=7)を自動で読みに行き、終了すると IDE\_RdRegValue レジスタにリードした値をセットして、IDE\_IntStat レジスタの CompleteINTRQ ビットが"1"になります。終了してもこのビットはセットされたままです。

#### Bit6-0 Reserved

2.6.122. 0B1h *HostDeviceSel* (*Host Device Select*)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	0B1h	<i>HostDeviceSel</i>		7:	0:	1:	00h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
			R / W	0: <i>HOSTxDEVICE</i>	0: Device mode	1: Host mode	

レジスタマップのデバイス・モードまたはホスト・モードの設定を行います。1Portモード時にはUSBの動作モードの設定を行います。詳細は「1.5.2.1 ポートモード時 (ClkSelect.Port1x2 が 1 にセットされている場合)」をご参照下さい。

**Bit7-1**    **Reserved**

**Bit0**     **HOSTxDEVICE**

デバイス・モードまたはホスト・モードの設定を行います。

0 : Device mode

1 : host mode

このビットの設定を変更しても、各レジスタの設定値はクリアされません。

このビットは、SLEEP / SNOOZE 中もアクセス有効です。

## 2. レジスタ

### 2.6.123. 0B3h *ModeProtect*(*Mode Protection*)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	B3h	<i>ModeProtect</i>	R / W	7: <i>ModeProtect</i> [7]	Mode Protection	56h
				6: <i>ModeProtect</i> [6]		
				5: <i>ModeProtect</i> [5]		
				4: <i>ModeProtect</i> [4]		
				3: <i>ModeProtect</i> [3]		
				2: <i>ModeProtect</i> [2]		
				1: <i>ModeProtect</i> [1]		
				0: <i>ModeProtect</i> [0]		

#### Bit7-0 *ModeProtect*[7:0]

ChipConfig レジスタ及び ClkSelect.ClkSelect ビットの値を保護します。このレジスタに 56h を書き込むと ChipConfig レジスタ及び ClkSelect.ClkSelect ビットへのライトアクセスが有効になります。

通常使用においては、ChipConfig レジスタ及び ClkSelect.ClkSelect ビットを任意に設定した後、このレジスタに 56h 以外の値(例えば 00h)を設定して、ChipConfig レジスタ及び ClkSelect.ClkSelect ビットの設定を保護して下さい。

このビットは、SLEEP / SNOOZE 中もアクセス有効です。

2.6.124. 0B5h *ClkSelect* (Clock Select)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	0B5h	<b><i>ClkSelect</i></b>	R / W	7: <b><i>xActIDE_Term</i></b>	0: Termination ON	1: Termination OFF	41h
			R / W	6: <b><i>xActIDE_DD_Term</i></b>	0: Termination ON	1: Termination OFF	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
			R / W	1: <b><i>PORT1x2</i></b>	0: 2 Port	1: 1 Port	
			R / W	0: <b><i>ClkSelect</i></b>	0: 12MHz	1: 24MHz	

**Bit7** ***xActIDE\_Term***

IDE ポートの端子のうち、  
HDMARQ、HIORDY、HINTRQ、XHDASP、HDD7 の Termination を ON/OFF します。  
このビットは、SLEEP / SNOOZE 中もアクセス有効です。

0 : Termination ON

1 : Termination OFF

**Bit6** ***xActIDE\_DD\_Term***

IDE ポート端子のうち、  
HDD15～HDD8、HDD6～HDD0 の Termination を ON/OFF します。  
このビットは、SLEEP / SNOOZE 中もアクセス有効です。

0 : Termination ON

1 : Termination OFF

**Bit6-2** **Reserved****Bit1** ***PORT1x2***

本 LSI で使用する USB ポートのモードを選択します。

2 Port 設定時は、A ポートに USB Host、B ポートに USB Device が割り当てられ、いずれかの機能を選択して使用します。

1 Port 設定時は、B ポートのみが使用でき、B ポートにて USB Host / Device のいずれかの機能を選択して使用します。

1 ポートモードの詳細は「Appendix C 1 ポートモード」を参照してください。

このビットは、SLEEP / SNOOZE 中もアクセス有効です。

0 : 2 Port

1 : 1 Port

**Bit0** ***ClkSelect***

本 LSI で使用するクロックを選択します。このビットは、SLEEP / SNOOZE 中もアクセス有効です。

0 : 12MHz

1 : 24MHz

## 2. レジスタ

### 2.6.125. 0B7h ChipConfig (Chip Configuration)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	0B7h	<b>ChipConfig</b>	R / W	7: <b>IntLevel</b>	0: Low Active	1: High Active	0Xh
			R / W	6: <b>IntMode</b>	0: 1/0 mode	1: Hi-z/0 mode	
			R / W	5: <b>DREQ_Level</b>	0: Low Active	1: High Active	
			R / W	4: <b>DACK_Level</b>	0: Low Active	1: High Active	
			R / W	3: <b>CS_Mode</b>	0: DACK mode	1: CS mode	
			R	2: <b>CPU_Endian</b>	0: Do nothing	1: Bus Swap	
			R	1: <b>BusMode</b>	0: XWRH/L mode	1: XBEH/L mode	
			R	0: <b>Initialized</b>	0: Uninitialized	1: Initialized	

本 LSI の動作モードを設定します。

#### Bit7 **IntLevel**

XINT の論理レベルを設定します。このビットは、SLEEP / SNOOZE 中もアクセス有効です。

- 0 : 負論理
- 1 : 正論理

#### Bit6 **IntMode**

XINT の出力モードを設定します。このビットは、SLEEP / SNOOZE 中もアクセス有効です。

- 0 : 1/0 モード
- 1 : Hi-z/0 モード

#### Bit5 **DREQ\_Level**

XDREQ0,1 の論理レベルを設定します。このビットは、SLEEP / SNOOZE 中もアクセス有効です。

- 0 : 負論理
- 1 : 正論理

#### Bit4 **DACK\_Level**

XDACK0,1 の論理レベルを設定します。このビットは、SLEEP / SNOOZE 中もアクセス有効です。

- 0 : 負論理
- 1 : 正論理

#### Bit3 **CS\_Mode**

DMA0,1 の動作モードを設定します。このビットは、SLEEP / SNOOZE 中もアクセス有効です。

- 0 : XDACK0,1 がアサートされているとき有効な DMA アクセスとして動作します。
- 1 : XCS 且つ XDACK0,1 がアサートされているとき有効な DMA アクセスとして動作します。

#### Bit2 **CPU\_Endian**

CPU バスの設定を表示します。このビットは、SLEEP / SNOOZE 中もアクセス有効です。

- 0 : BigEndian モード。偶数アドレスを上位側、奇数アドレスを下位側とします。
  - 1 : LittleEndian モード。偶数アドレスを下位側、奇数アドレスを上位側とします。
- リトルエンディアンのCPUへの接続についての詳細およびレジスタマップは、「Appendix B. リトルエンディアンのCPUへの接続」を参照してください。

#### Bit1 **BusMode**

CPU の動作モード設定を表示します。このビットは、SLEEP / SNOOZE 中もアクセス有効です。

- 0 : Strobe モード
- 1 : BE モード

#### Bit0 **Initialized**

初期化済み期間であることを示すフラグです。通常、”1”が読み出されます。

## 2. レジスタ

### 2.6.126. 0BDh TimerConfig (Timer Configuration)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	0BDh	TimerConfig	W	7: TimerStop	0: Low Active	1: High Active	00h
			R	6: TimerRunning	0: 1／0 mode	1: Hi-z／0 mode	
				5:	0:	1:	
			R / W	4: FreeRun	0: Single Mode	1: Free Run Mode	
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
				0:	0:	1:	

タイマを設定します。

#### Bit7 TimerStop

このビットに”1”をライトする事でタイマを停止させる事が出来ます。停止後、このビットはクリアされます。

#### Bit6 TimerRunning

タイマの動作状態を表示します。

0：タイマ停止

1：タイマ動作

#### Bit5 Reserved

#### Bit4 FreeRun

このビットを”1”にセットすると、タイマをフリーランで 사용할 ことが出来ます。

0: シングルモード、ダウンカウンタが 0x0 になるとタイマのカウントを停止します。

1: フリーランモード、ダウンカウンタが 0x0 になると、TimerSet\_H,L レジスタの値をロードし、カウントを続けます。

#### Bit3-0 Reserved



## 2.6.127. 0BEh TimerSet\_H (Timer Set High)

## 2.6.128. 0BFh TimerSet\_L (Timer Set Low)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	0BEh	TimerSet_H	R / W	TimerSet[15:8]	Timer Set[15:8]	FFh

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	0BFh	TimerSet_L	R / W	TimerSet[7:0]	Timer Set[7:0]	FFh

タイマのダウンカウント値を設定します。このレジスタへ書き込みを行うと、ダウンカウンタへ値がロードされて、タイマが動作を始めます。タイマ動作中にこのレジスタへ書き込みを行った場合は、その時点では値がロードされません。フリーランモードで動作中には、ダウンカウント値が 0x0 になると、新たな値がロードされてダウンカウントを行います。

タイマの周期は以下の計算式となります。

$$(60\text{MHz} \times 65536) \times (\text{本レジスタへの設定値} + 1)$$

## 2. レジスタ

### 2.6.129. 0C0h AREAnIntStat (AREAn Interrupt Status)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	0C0h	AREAnIntStat		7:	0:	1:
				6:	0:	1:
			R	5: AREA5IntStat	0: None	1: AREA5 Interrupts
			R	4: AREA4IntStat	0: None	1: AREA4 Interrupts
			R	3: AREA3IntStat	0: None	1: AREA3 Interrupts
			R	2: AREA2IntStat	0: None	1: AREA2 Interrupts
			R	1: AREA1IntStat	0: None	1: AREA1 Interrupts
			R	0: AREA0IntStat	0: None	1: AREA0 Interrupts

FIFO 領域の割り込み要因を間接指示します。

#### Bit7-6 Reserved

#### Bit5 AREA5IntStat

割り込み要因を間接指示します。

AREA5IntStat レジスタに割り込み要因があり、かつその割り込み要因に対応する AREA5IntEnb レジスタのビットがイネーブルにされている時に"1"にセットされます。

#### Bit4 AREA4IntStat

割り込み要因を間接指示します。

AREA4IntStat レジスタに割り込み要因があり、かつその割り込み要因に対応する AREA4IntEnb レジスタのビットがイネーブルにされている時に"1"にセットされます。

#### Bit3 AREA3IntStat

割り込み要因を間接指示します。

AREA3IntStat レジスタに割り込み要因があり、かつその割り込み要因に対応する AREA3IntEnb レジスタのビットがイネーブルにされている時に"1"にセットされます。

#### Bit2 AREA2IntStat

割り込み要因を間接指示します。

AREA2IntStat レジスタに割り込み要因があり、かつその割り込み要因に対応する AREA2IntEnb レジスタのビットがイネーブルにされている時に"1"にセットされます。

#### Bit1 AREA1IntStat

割り込み要因を間接指示します。

AREA1IntStat レジスタに割り込み要因があり、かつその割り込み要因に対応する AREA1IntEnb レジスタのビットがイネーブルにされている時に"1"にセットされます。

#### Bit0 AREA0IntStat

割り込み要因を間接指示します。

AREA0IntStat レジスタに割り込み要因があり、かつその割り込み要因に対応する AREA0IntEnb レジスタのビットがイネーブルにされている時に"1"にセットされます。

2.6.130. 0C1h AREA0IntStat (AREA0 Interrupt Status)

2.6.131. 0C2h AREA1IntStat (AREA1 Interrupt Status)

2.6.132. 0C3h AREA2IntStat (AREA2 Interrupt Status)

2.6.133. 0C4h AREA3IntStat (AREA3 Interrupt Status)

2.6.134. 0C5h AREA4IntStat (AREA4 Interrupt Status)

2.6.135. 0C6h AREA5IntStat (AREA5 Interrupt Status)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	0C1h	AREA0IntStat		7:	0:	1:
	0C2h	AREA1IntStat		6:	0:	1:
	0C3h	AREA2IntStat		5:	0:	1:
	0C4h	AREA3IntStat		4:	0:	1:
	0C5h	AREA4IntStat		3:	0:	1:
	0C6h	AREA5IntStat	R (W)	2: FIFO_NotEmpty	0: None	1: FIFO NotEmpty
			R (W)	1: FIFO_Full	0: None	1: FIFO Full
			R (W)	0: FIFO_Empty	0: None	1: FIFO Empty

FIFO エリア x{x=0-5}関連の割り込みステータスを表示します。  
全てのビットは"1"を書き込む事で割り込み要因をクリアできます。

**Bit7-3** Reserved**Bit2** FIFO\_NotEmpty

割り込み要因を直接指示します。

FIFO エリア x{x=0-5}にデータが存在する状態(NotEmpty)になると"1"にセットされます。

**Bit1** FIFO\_Full

割り込み要因を直接指示します。

FIFO エリア x{x=0-5}が Full 時に"1"にセットされます。

**Bit0** FIFO\_Empty

割り込み要因を直接指示します。

FIFO エリア x{x=0-5}が Empty 時に"1"にセットされます。

## 2. レジスタ

---

### 2.6.136. 0C8h AREAnIntEnb (AREAn Interrupt Enable)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	0C8h	AREAnIntEnb		7:	0:	1:	00h
				6:	0:	1:	
			R / W	5: EnAREA5IntStat	0: Disable	1: Enable	
			R / W	4: EnAREA4IntStat	0: Disable	1: Enable	
			R / W	3: EnAREA3IntStat	0: Disable	1: Enable	
			R / W	2: EnAREA2IntStat	0: Disable	1: Enable	
			R / W	1: EnAREA1IntStat	0: Disable	1: Enable	
			R / W	0: EnAREA0IntStat	0: Disable	1: Enable	

AREAnIntStat レジスタの割り込み要因による、MainIntStat レジスタの AREAnIntStat ビットのアサー  
トを許可／禁止します。

2.6.137. 0C9h AREA0IntEnb (AREA0 Interrupt Enable)

2.6.138. 0CAh AREA1IntEnb (AREA1 Interrupt Enable)

2.6.139. 0CBh AREA2IntEnb (AREA2 Interrupt Enable)

2.6.140. 0CCh AREA3IntEnb (AREA3 Interrupt Enable)

2.6.141. 0CDh AREA4IntEnb (AREA4 Interrupt Enable)

2.6.142. 0CEh AREA5IntEnb (AREA5 Interrupt Enable)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	0C9h	AREA0IntEnb		7:	0:	1:	00h
	0CAh	AREA1IntEnb		6:	0:	1:	
	0CBh	AREA2IntEnb		5:	0:	1:	
	0CCh	AREA3IntEnb		4:	0:	1:	
	0CDh	AREA4IntEnb		3:	0:	1:	
	0CEh	AREA5IntEnb	R / W	2: EnFIFO_NotEmpty	0: Disable	1: Enable	
			R / W	1: EnFIFO_Full	0: Disable	1: Enable	
			R / W	0: EnFIFO_Empty	0: Disable	1: Enable	

AREAx{x=0-5}IntStat レジスタの割り込み要因による、AREAnIntStat レジスタの AREAx{x=0-5}IntStat ビットのアサートを許可／禁止します。

## 2. レジスタ

- 2.6.143. 0D0h AREA0Join\_0 (AREA 0 Join 0)  
 2.6.144. 0D2h AREA1Join\_0 (AREA 1 Join 0)  
 2.6.145. 0D4h AREA2Join\_0 (AREA 2 Join 0)  
 2.6.146. 0D6h AREA3Join\_0 (AREA3 Join 0)  
 2.6.147. 0D8h AREA4Join\_0 (AREA 4 Join 0)  
 2.6.148. 0DAh AREA5Join\_0 (AREA 5 Join 0)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	0D0h	AREA0Join_0	R / W	7: JoinIDE	0: Do nothing	1: Join to IDE	00h
	0D2h	AREA1Join_0	R / W	6: JoinFIFO_Stat	0: Do nothing	1: Join to FIFO Status	
	0D4h	AREA2Join_0		5:	0:	1:	
	0D6h	AREA3Join_0		4:	0:	1:	
	0D8h	AREA4Join_0	R / W	3: JoinDMA1	0: Do nothing	1: Join to DMA1	
	0DAh	AREA5Join_0	R / W	2: JoinDMA0	0: Do nothing	1: Join to DMA0	
			R / W	1: JoinCPU_Rd	0: Do nothing	1: Join to CPU Read	
			R / W	0: JoinCPU_Wr	0: Do nothing	1: Join to CPU Write	

FIFO 領域 AREA<sub>x</sub>{x=0-5}に接続するポートを設定します。

### Bit7 JoinIDE

FIFO 領域 AREA<sub>x</sub>{x=0-5}の FIFO で IDE の転送を行います。転送の方向は、IDE\_Control.Dir ビットの設定によります。

### Bit6 JoinFIFO\_Stat

FIFO 領域 AREA<sub>x</sub>{x=0-5}の FIFO の Full 及び Empty の状態を、D\_FIFO\_IntStat.FIFO\_NotEmpty、D\_FIFO\_IntStat.FIFO\_Full 及び D\_FIFO\_IntStat.FIFO\_Empty、または、H\_FIFO\_IntStat.FIFO\_NotEmpty、H\_FIFO\_IntStat.FIFO\_Full 及び H\_FIFO\_IntStat.FIFO\_Empty でモニタできるようにします。

### Bit5-4 Reserved

### Bit3 JoinDMA1

FIFO 領域 AREA<sub>x</sub>{x=0-5}の FIFO で DMA1 の転送を行います。転送の方向は、DMA1\_Control.Dir ビットの設定によります。

### Bit2 JoinDMA0

FIFO 領域 AREA<sub>x</sub>{x=0-5}の FIFO で DMA0 の転送を行います。転送の方向は、DMA0\_Control.Dir ビットの設定によります。

### Bit1 JoinCPU\_Rd

FIFO 領域 AREA<sub>x</sub>{x=0-5}の FIFO で CPU レジスタアクセスのリード転送を行います。即ち FIFO\_Rd\_0,1 レジスタ、または、FIFO\_ByteRd レジスタのリードが行われると、この FIFO 領域からデータが読み出されます。

### Bit0 JoinCPU\_Wr

FIFO 領域 AREA<sub>x</sub>{x=0-5}の FIFO で CPU レジスタアクセスのライト転送を行います。即ち FIFO\_Wr\_0,1 レジスタ、または、FIFO\_ByteWr レジスタへのライトが行われると、この FIFO 領域にデータが書き込まれます。

JoinDMA<sub>x</sub>{x=0,1}ビットを設定した場合は、DMA<sub>x</sub>{x=0,1}\_Control.Dir ビットが 1 のときは残りデータ数、0 のときは空き容量が、DMA<sub>x</sub>{x=0,1}\_Remain\_H,L レジスタにより、それぞれ参照できます。

JoinCPU\_Rd, JoinCPU\_Wr ビットを設定した場合は、FIFO\_RdRemain\_H,L、FIFO\_WrRemain\_H,L を参照し、FIFO\_Rd\_0,1、FIFO\_ByteRd、FIFO\_Wr\_0,1、FIFO\_ByteWr レジスタからデータを読み出し、または書き込みできます。

JoinDMA<sub>x</sub>{x=0,1}ビット、JoinCPU\_Rd ビット、JoinCPU\_Wr ビットは、同時に 1 ビットのみを"1"にセットしてください。複数のビットに対して、同時に"1"を書きこんだ場合、動作が不安定になる恐れがあります。

FIFO領域の設定には、排他条件がありますので、「1.6.5.6.FIFOアクセスの制限」を参照して下さい。

- 2.6.149. 0D1h AREA0Join\_1 (AREA 0 Join 1)  
 2.6.150. 0D3h AREA1Join\_1 (AREA 1 Join 1)  
 2.6.151. 0D5h AREA2Join\_1 (AREA 2 Join 1)  
 2.6.152. 0D7h AREA3Join\_1 (AREA 3 Join 1)  
 2.6.153. 0D9h AREA4Join\_1 (AREA 4 Join 1)  
 2.6.154. 0DBh AREA5Join\_1 (AREA 5 Join 1)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	0D5h	AREA2Join_1		7:	0:	1:	00h
				6:	0:	1:	
			R / W	5: JoinEPeCHe	0: Do nothing	1: Join to EPe / CHe	
			R / W	4: JoinEPdCHd	0: Do nothing	1: Join to EPd / CHd	
			R / W	3: JoinEPcCHc	0: Do nothing	1: Join to EPc / CHc	
			R / W	2: JoinEPbCHb	0: Do nothing	1: Join to EPb / CHb	
			R / W	1: JoinEPaCHa	0: Do nothing	1: Join to EPa / CHa	
			R / W	0: JoinEP0CH0	0: Do nothing	1: Join to EP0 / CH0	

FIFO 領域 AREA<sub>x</sub>{x=0-5}に接続するエンドポイント及びチャンネルを関連付けます。

**Bit7-6 Reserved**

**Bit5 JoinEPeCHe**

FIFO 領域 AREA<sub>x</sub>{x=0-5}にエンドポイント EPe もしくはチャンネル CHe を接続します。接続することで、エンドポイント EPe、チャンネル EPe を使用して、データ転送を伴うトランザクションを実行可能となります。

**Bit4 JoinEPdCHd**

FIFO 領域 AREA<sub>x</sub>{x=0-5}にエンドポイント EPd もしくはチャンネル CHd を接続します。接続することで、エンドポイント EPd、チャンネル EPd を使用して、データ転送を伴うトランザクションを実行可能となります。

**Bit3 JoinEPcCHc**

FIFO 領域 AREA<sub>x</sub>{x=0-5}にエンドポイント EPc もしくはチャンネル CHc を接続します。接続することで、エンドポイント EPc、チャンネル EPc を使用して、データ転送を伴うトランザクションを実行可能となります。

**Bit2 JoinEPbCHb**

FIFO 領域 AREA<sub>x</sub>{x=0-5}にエンドポイント EPb もしくはチャンネル CHb を接続します。接続することで、エンドポイント EPb、チャンネル EPb を使用して、データ転送を伴うトランザクションを実行可能となります。

**Bit1 JoinEPaCHa**

FIFO 領域 AREA<sub>x</sub>{x=0-5}にエンドポイント EPa もしくはチャンネル CHa を接続します。接続することで、エンドポイント EPa、チャンネル EPa を使用して、データ転送を伴うトランザクションを実行可能となります。

**Bit0 JoinEP0CH0**

FIFO 領域 AREA<sub>x</sub>{x=0-5}にエンドポイント EP0 もしくはチャンネル CH0 を接続します。接続することで、エンドポイント EP0、チャンネル EP0 を使用して、データ転送を伴うトランザクションを実行可能となります。

同一の FIFO 領域に JoinEPxCHx{x=0,a-e}ビットを同時に複数設定した場合には、トランザクション順によっては、想定外の動作を行うことがありますので、ご注意願います。基本的には、同一の FIFO 領域に JoinEPxCHx{x=0,a-e}ビットを設定行わないことを推奨致します。

FIFOの関連付けには制限がありますので、「1.2.1.エンドポイント」及び「1.3.1.チャンネル」を参照して下さい。

## 2. レジスタ

---

### 2.6.155. 0DEh ClrAREAnJoin\_0 (Clear AREA n Join 0)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	0DEh	ClrAREAnJoin_0	W	7: ClrJoinIDE	0: Do nothing	1: Clear JoinIDE	00h
			W	6: ClrJoinFIFO_Stat	0: Do nothing	1: Clear JoinFIFO_Stat	
				5:	0:	1:	
				4:	0:	1:	
			W	3: ClrJoinDMA1	0: Do nothing	1: Clear JoinDMA1	
			W	2: ClrJoinDMA0	0: Do nothing	1: Clear JoinDMA0	
			W	1: ClrJoinCPU_Rd	0: Do nothing	1: Clear JoinCPU_Rd	
			W	0: ClrJoinCPU_Wr	0: Do nothing	1: Clear JoinCPU_Wr	

各 FIFO 領域と該当するポートの接続をクリアします。ライトオンリーのレジスタです。

このレジスタのビットは、接続クリア後、自動的に"0"にクリアされます。

FIFO 領域がポートに接続(AREAx{x=0-5}Join\_0 レジスタの該当するビットが"1"にセット)され、且つ各ポートの起動中に、このレジスタのビットを"1"にセットしないで下さい。誤動作の原因となります。



## 2.6.156. 0DFh ClrAREAnJoin\_1 (Clear AREA 1 Join 1)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	0DFh	ClrAREAnJoin_1		7:	0:	1:	00h
				6:	0:	1:	
			W	5: ClrJoinEPeCHe	0: Do nothing	1: Clear JoinEPeCHe	
			W	4: ClrJoinEPdCHd	0: Do nothing	1: Clear JoinEPdCHd	
			W	3: ClrJoinEPcCHc	0: Do nothing	1: Clear JoinEPcCHc	
			W	2: ClrJoinEPbCHb	0: Do nothing	1: Clear JoinEPbCHb	
			W	1: ClrJoinEPaCHa	0: Do nothing	1: Clear JoinEPaCHa	
			W	0: ClrJoinEP0CH0	0: Do nothing	1: Clear JoinEP0CH0	

各 FIFO 領域と該当するエンドポイント及びチャネルの接続をクリアします。ライトオンリーのレジスタです。

このレジスタのビットは、接続クリア後、自動的に"0"にクリアされます。

FIFO 領域がエンドポイント及びチャネルに接続(AREAx{x=0-5}Join\_1 レジスタの該当するビットが"1"にセット)され、且つ各エンドポイント及びチャネルのトランザクションが実行されている時に、このレジスタのビットを"1"にセットしないで下さい。誤動作の原因となります。

## 2. レジスタ

- 2.6.157. 180h AREA0StartAdrs\_H (AREA 0 Start Address High)  
 2.6.158. 181h AREA0StartAdrs\_L (AREA 0 Start Address Low)  
 2.6.159. 184h AREA1StartAdrs\_H (AREA 1 Start Address High)  
 2.6.160. 185h AREA1StartAdrs\_L (AREA 1 Start Address Low)  
 2.6.161. 188h AREA2StartAdrs\_H (AREA2 Start Address High)  
 2.6.162. 189h AREA2StartAdrs\_L (AREA2 Start Address Low)  
 2.6.163. 18Ch AREA3StartAdrs\_H (AREA3 Start Address High)  
 2.6.164. 18Dh AREA3StartAdrs\_L (AREA3 Start Address Low)  
 2.6.165. 190h AREA4StartAdrs\_H (AREA4 Start Address High)  
 2.6.166. 191h AREA4StartAdrs\_L (AREA4 Start Address Low)  
 2.6.167. 194h AREA5StartAdrs\_H (AREA5 Start Address High)  
 2.6.168. 195h AREA5StartAdrs\_L (AREA5 Start Address Low)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	180h	AREA0StartAdrs_H		7:	0:	1:
	184h	AREA1StartAdrs_H		6:	0:	1:
	188h	AREA2StartAdrs_H		5:	0:	1:
	18Ch	AREA3StartAdrs_H	R / W	4: StartAdrs[12]	AREAx{x=0-5} Start Address High	00h
	190h	AREA4StartAdrs_H		3: StartAdrs[11]		
	194h	AREA5StartAdrs_H		2: StartAdrs[10]		
				1: StartAdrs[9]		
				0: StartAdrs[8]		

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	181h	AREA0StartAdrs_L	R / W	7: StartAdrs[7]	AREAx{x=0-5} Start Address Low	00h
	185h	AREA1StartAdrs_L		6: StartAdrs[6]		
	189h	AREA2StartAdrs_L		5: StartAdrs[5]		
	18Dh	AREA3StartAdrs_L		4: StartAdrs[4]		
	191h	AREA4StartAdrs_L		3: StartAdrs[3]		
	195h	AREA5StartAdrs_L		2: StartAdrs[2]		
				1:		
				0:		

AREAx{x=0-5}で使用する FIFO の領域設定を行います。

### XX0h.Bit7-5 Reserved

### XX0h.Bit4-0, XX1h.Bit7-2 StartAdrs[12:2]

FIFO 領域 AREAx{x=0-5}に割り当てる FIFO の先頭アドレスを設定します。

アドレス値は、上位 12 ビット～2 ビットでの設定のため 4 バイト単位での指定になります。

FIFO 領域 AREAx{x=0-5}に割り当てられる領域は、EndAdrs にて設定されたアドレスの 1 バイト前までとなります。

StartAdrs、EndAdrs を設定した後は、必ず AREAnFIFO\_Clr レジスタの ClrAREAx{x=0-5} ビットを"1"にして FIFO 領域 AREAx{x=0-5}の FIFO をクリアして下さい。

なお、ここで設定した領域より、ジョインした USB のデバイス/ホストの MaxSize が大きい場合には、正常に動作しません。また、FIFO 領域 AREAx{x=0-5}と他の FIFO 領域が重なった場合も、正常に動作しません。

本 LSI の内蔵 RAM は 4.5kB ですので、最大 0x1200 番地のアドレスをサポートしています。

### XX1h.Bit1-0 Reserved

- 2.6.169. 182h AREA0EndAdrs\_H (AREA 0 End Address High)  
 2.6.170. 183h AREA0EndAdrs\_L (AREA 0 End Address Low)  
 2.6.171. 186h AREA1EndAdrs\_H (AREA1 End Address High)  
 2.6.172. 187h AREA1EndAdrs\_L (AREA1 End Address Low)  
 2.6.173. 18Ah AREA2EndAdrs\_H (AREA2 End Address High)  
 2.6.174. 18Bh AREA2EndAdrs\_L (AREA2 End Address Low)  
 2.6.175. 18Eh AREA3EndAdrs\_H (AREA3 End Address High)  
 2.6.176. 18Fh AREA3EndAdrs\_L (AREA3 End Address Low)  
 2.6.177. 192h AREA4EndAdrs\_H (AREA4 End Address High)  
 2.6.178. 193h AREA4EndAdrs\_L (AREA4 End Address Low)  
 2.6.179. 196h AREA5EndAdrs\_H (AREA5 End Address High)  
 2.6.180. 197h AREA5EndAdrs\_L (AREA5 End Address Low)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	182h	AREA0EndAdrs_H		7:	0:	1:
	186h	AREA1EndAdrs_H		6:	0:	1:
	18Ah	AREA2EndAdrs_H		5:	0:	1:
	18Eh	AREA3EndAdrs_H	R / W	4: EndAdrs[12]	AREAx{x=0-5} End Address High	00h
	192h	AREA4EndAdrs_H		3: EndAdrs[11]		
	196h	AREA5EndAdrs_H		2: EndAdrs[10]		
				1: EndAdrs[9]		
				0: EndAdrs[8]		

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device / Host	183h	AREA0EndAdrs_L	R / W	7: EndAdrs[7]	AREAx{x=0-5} End Address Low	00h
	187h	AREA1EndAdrs_L		6: EndAdrs[6]		
	18Bh	AREA2EndAdrs_L		5: EndAdrs[5]		
	18Fh	AREA3EndAdrs_L		4: EndAdrs[4]		
	193h	AREA4EndAdrs_L		3: EndAdrs[3]		
	197h	AREA5EndAdrs_L		2: EndAdrs[2]		
				1:		
				0:		

AREAx{x=0-5}で使用する FIFO の領域設定を行います。

#### XX0h.Bit7-5 Reserved

#### XX0h.Bit4-0, XX1h.Bit7-2 EndAdrs[12:2]

FIFO 領域 AREAx{x=0-5}に割り当てる FIFO の最終アドレスの次のバイトを設定します。アドレス値は、上位 12 ビット～2 ビットでの設定のため 4 バイト単位での指定になります。

FIFO 領域 AREAx{x=0-5}に割り当てられる領域は、EndAdrs にて設定されたアドレスの 1 バイト前までとなります。

StartAdrs、EndAdrs を設定した後は、必ず AREAnFIFO\_Clr レジスタの ClrAREAx{x=0-5} ビットを"1"にして FIFO 領域 AREAx{x=0-5}の FIFO をクリアして下さい。

なお、ここで設定した領域より、ジョインした USB のデバイス/ホストの MaxSize が大きい場合には、正常に動作しません。また、FIFO 領域 AREAx{x=0-5}と他の FIFO 領域が重なった場合も、正常に動作しません。

本 LSI の内蔵 RAM は 4.5kB です。最大 0x1200 番地のアドレスをサポートしています。

#### XX1h.Bit1-0 Reserved

## 2. レジスタ

### 2.6.181. 19Fh AREAnFIFO\_Clr (AREAn FIFO Clear )

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device / Host	19Fh	AREAnFIFO_Clr		7:	0:	1:	XXh
				6:	0:	1:	
			W	5: AREA5FIFO_Clr	0: Do nothing	1: Clear AREA5 FIFO	
			W	4: AREA4FIFO_Clr	0: Do nothing	1: Clear AREA4 FIFO	
			W	3: AREA3FIFO_Clr	0: Do nothing	1: Clear AREA3 FIFO	
			W	2: AREA2FIFO_Clr	0: Do nothing	1: Clear AREA2 FIFO	
			W	1: AREA1FIFO_Clr	0: Do nothing	1: Clear AREA1 FIFO	
			W	0: AREA0FIFO_Clr	0: Do nothing	1: Clear AREA0FIFO	

該当する FIFO 領域 AREAx{x=0-5}の FIFO をクリアします。ライトオンリーのレジスタです。

このレジスタの各ビットは、"1"をセットされると FIFO をクリアする動作のみ行い、セットされた値は保持しません。

FIFO 領域 AREAx{x=0-5}に DMA がジョインされ、かつ、該当する DMA が起動中(DMA\_Running ビットが"1"の間)に、該当するエンドポイントのビットを"1"にセットしないで下さい。

また、このレジスタは、データ保持情報の初期化動作のみを行い、データ自身を書き込みまたはクリアをしません。従って、このビットにより RAM 上のデータがクリアされることはありません。

## 2.7. デバイス・レジスタ詳細説明

## 2.7.1. 0E0h D\_SIE\_IntStat (Device SIE Interrupt Status)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device	0E0h	D_SIE_IntStat		7:	0:	1:	00h
			R (W)	6: <b>NonJ</b>	0: None	1: Detect Non J state	
			R (W)	5: RcvSOF	0: None	1: Received SOF	
			R (W)	4: DetectRESET	0: None	1: Detect USB Reset	
			R (W)	3: DetectSUSPEND	0: None	1: Detect USB Suspend	
			R (W)	2: ChirpCmp	0: None	1: Chirp Complete	
			R (W)	1: RestoreCmp	0: None	1: Restore Complete	
			R (W)	0: SetAddressCmp	0: None	1: AutoSetAddress Complete	

デバイス SIE 関連の割り込みを表示します。

全てのビットは"1"を書き込む事で割り込み要因をクリアできます。

**Bit7**      **Reserved**

**Bit6**      **NonJ**

割り込み要因を直接指示します。

USB バス上で J ステート以外の状態を検出すると"1"にセットされます。このビットは、本 LSI が SNOOZE 状態(PM\_Control レジスタの InSnooze ビットが"1")の時、及び AutoNegotiation 機能使用時に USB\_Control レジスタの InSUSPEND ビットが"1"にセットされている時に有効です。

**Bit5**      **RcvSOF**

割り込み要因を直接指示します。

SOF トークンを受信すると"1"にセットされます。

**Bit4**      **DetectRESET**

割り込み要因を直接指示します。

USB のリセットステートを検出すると"1"にセットされます。このビットがセットされている間は USB のサスペンドステートの検出ができません(DetectSUSPEND がセットされません)。

このリセット検出は、D\_NegoControl レジスタの ActiveUSB ビットが"1"にセットされている時に有効です。

"HS"動作モードの場合は、バス・アクティビティが一定時間無くなると、USB のリセット／サスペンド検出のために FS ターミネーションを自動的に設定し、SE0 が検出されるとリセットと判断して、このビットが"1"にセットされます。

AutoNegotiation 機能を使用しない場合には、このビットが"1"にセットされた場合、継続するリセットを誤検出しないよう、D\_NegoControl レジスタの DisBusDetect ビットを"1"にセットして USB のリセット／サスペンドステートの検出を無効にしてください。リセットに対する処理終了後に DisBusDetect ビットを"0"にクリアして USB のリセット／サスペンドステートの検出を有効にしてください。

リセット検出時、D\_NegoControl レジスタの GoChirp ビットにより、"HS Detection Handshake"を開始することができます。

AutoNegotiation 機能については、D\_NegoControl レジスタの EnAutoNego ビットの項 (「298.102h D\_NegoControl (Device Negotiation Control)」)を参照して下さい。

**Bit3**      **DetectSUSPEND**

割り込み要因を直接指示します。

USB のサスペンドステートを検出すると"1"にセットされます。このビットがセットされている間は USB のリセットステートの検出ができません(DetectRESET がセットされません)。

"HS"動作モードの場合は、バス・アクティビティが一定時間無くなると、USB のリセット

## 2. レジスタ

---

／サスペンド検出のために"FS"動作モードに自動的に設定されます。USB のサスペンドステートの検出後は、PM\_Control\_0 レジスタの GoSnooze ビットを"1"にセットすることにより、本 LSI をスヌーズモード(内蔵 PLL の発振を停止)にすることが出来ます。

### Bit2 ChirpCmp

割り込み要因を直接指示します。

D\_NegoControl レジスタの GoChirp ビットにより開始された"HS Detection Handshake"が完了すると"1"にセットされます。

割り込み発生後に D\_USB\_Status レジスタの FSxHS ビットをリードすることで、現在の動作モード(FS or HS)の判定をすることができます。

**Bit1 RestoreCmp**

割り込み要因を直接指示します。

D\_NegoControl レジスタの RestoreUSB ビットにより開始された Restore 処理が終了すると "1" にセットされます。このビットが "1" にセットされると動作モード (FS or HS) が Suspend する前の状態に戻ります。

**Bit0 SetAddressCmp**

割り込み要因を直接指示します。

SetAddress() リクエストを受信すると、AutoSetAddress機能(「2.7.36.118h D\_USB\_Address (Device USB Address)」参照)が、そのコントロール転送の処理を自動的行います。ステータスステージを行って SetAddress() リクエストに関わるコントロール転送が完了した時に、このステータスが "1" にセットされます。また、同時に D\_USB\_Address レジスタにアドレスがセットされます。

同期ビット (Bit5～0) は ACTIVE60 / ACT\_HOST ステートの時、読み出しはできますが、書き込み(割り込み要因クリア)できません。

ACT\_DEVICE ステートまたは ACT\_ALL ステートから、それ以外のステートに遷移する際には、これらの割り込みステータスにより割り込み信号 XINT がアサートされないよう、F/W にて以下の処理を行って下さい。

＜ACT\_DEVICE / ACT\_ALL ステートを出るとき＞

- 1) 割り込みステータスを処理し、クリアする (D\_SIE\_IntStat.Bit5～0)
- 2) 割り込みステータスをディスエーブルにする (D\_SIE\_IntEnb.Bit5～0)

＜ACT\_DEVICE / ACT\_ALL ステートに入るとき＞

- 3) 割り込みステータスをクリアする (D\_SIE\_IntStat.Bit5～0)
- 4) 割り込みステータスをイネーブルにする (D\_SIE\_IntEnb.Bit5～0)

## 2. レジスタ

### 2.7.2. 0E2h D\_FIFO\_IntStat (Device FIFO Interrupt Status)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device	0E2h	D_FIFO_IntStat	R (W)	7: DescriptorCmp	0: None	1: Descriptor Complete	00h
			R (W)	6: FIFO_IDE_Cmp	0: None	1: FIFO-IDE Complete	
			R (W)	5: FIFO1_Cmp	0: None	1: FIFO1 Complete	
			R (W)	4: FIFO0_Cmp	0: None	1: FIFO0 Complete	
				3:	0:	1:	
			R (W)	2: FIFO_NotEmpty	0: None	1: FIFO NotEmpty	
			R (W)	1: FIFO_Full	0: None	1: FIFO Full	
			R (W)	0: FIFO_Empty	0: None	1: FIFO Empty	

デバイス FIFO 関連の割り込みステータスを表示します。  
全てのビットは"1"を書き込む事で割り込み要因をクリアできます。

#### Bit7 DescriptorCmp

割り込み要因を直接指示します。

Descriptor 返信機能において、DescriptorSize レジスタの設定数のデータを返信し終わると、"1"にセットされます。

また、DescriptorSize レジスタの設定数まで送信する前にステータスステージへ移行(OUT トークンを受信)した場合には D\_EP0IntStat レジスタの OUT\_TranNAK ビットと共に、"1"にセットされます。

#### Bit6 FIFO\_IDE\_Cmp

割り込み要因を直接指示します。

IDE にジョインされているエンドポイントが IN 方向の場合、IDE の転送が終了した後に FIFO が空になると、"1"にセットされます。IDE にジョインされているエンドポイントが OUT 方向の場合、IDE の転送が終了すると、"1"にセットされます。

#### Bit5 FIFO1\_Cmp

割り込み要因を直接指示します。

DMA1 にジョインされているエンドポイントが IN 方向の場合、DMA1 の転送が終了した後に FIFO が空になると、"1"にセットされます。DMA1 にジョインされているエンドポイントが OUT 方向の場合、DMA1 の転送が終了すると、"1"にセットされます。

#### Bit4 FIFO0\_Cmp

割り込み要因を直接指示します。

DMA0 にジョインされているエンドポイントが IN 方向の場合、DMA0 の転送が終了した後に FIFO が空になると、"1"にセットされます。DMA0 にジョインされているエンドポイントが OUT 方向の場合、DMA0 の転送が終了すると、"1"にセットされます。

#### Bit3 Reserved

#### Bit2 FIFO\_NotEmpty

割り込み要因を直接指示します。

AREAx{x=0-5}Join.JoinFIFO\_Stat ビットが"1"にセットされているときに、該当する FIFO 領域にデータが存在する状態(NotEmpty)になると、"1"にセットされます。

#### Bit1 FIFO\_Full

割り込み要因を直接指示します。

AREAx{x=0-5}Join.JoinFIFO\_Stat ビットが"1"にセットされているときに、該当する FIFO 領域が Full になると、"1"にセットされます。

#### Bit0 FIFO\_Empty

割り込み要因を直接指示します。

AREAx{x=0-5}Join.JoinFIFO\_Stat ビットが"1"にセットされているときに、該当する FIFO 領域が Empty になると、"1"にセットされます。



## 2.7.3. 0E3h D\_BulkIntStat (Device Bulk Interrupt Status)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device	0E3h	D_BulkIntStat	R (W)	7: CBW_Cmp	0: None	1: CBW Complete	00h
			R (W)	6: CBW_LengthErr	0: None	1: CBW Length Error	
			R (W)	5: CBW_Err	0: None	1: CBW Transaction Error	
			R (W)	4:	0:	1:	
			R (W)	3: CSW_Cmp	0: None	1: CSW Complete	
			R (W)	2: CSW_Err	0: None	1: CSW Error	
				1:	0:	1:	
				0:	0:	1:	

Bulk 転送機能関連の割り込みステータスを表示します。全てのビットは"1"を書き込む事で割り込み要因をクリアできます。

**Bit7 CBW\_Cmp**

割り込み要因を直接指示します。

CBW の 31 バイトを正常に受信できた時に"1"にセットされます。

**Bit6 CBW\_LengthErr**

割り込み要因を直接指示します。

受信した CBW のパケット長が 31 バイト以外であった時に"1"にセットされます。

**Bit5 CBW\_Err**

割り込み要因を直接指示します。

受信した CBW に CRC エラー等のトランザクションエラーを検出した時に"1"にセットされます。

**Bit4 Reserved****Bit3 CSW\_Cmp**

割り込み要因を直接指示します。

CSW の 13 バイトを正常に送信できた時に"1"にセットされます。

**Bit2 CSW\_Err**

割り込み要因を直接指示します。

CSW の送信にエラーがあった時(ACK が返ってこなかった時)に"1"にセットされます。

**Bit1-0 Reserved**

## 2. レジスタ

### 2.7.4. 0E4h D\_EPrIntStat (Device EPr Interrupt Status)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device	0E4h	D_EPrIntStat		7:	0:	1:	00h
				6:	0:	1:	
				5:	0:	1:	
			R	4: E_EPeIntStat	0: None	1: EPe Interrupt	
			R	3: D_EPdIntStat	0: None	1: EPd Interrupt	
			R	2: D_EPcIntStat	0: None	1: EPc Interrupt	
			R	1: D_EPbIntStat	0: None	1: EPb Interrupt	
			R	0: D_EPaIntStat	0: None	1: EPa Interrupt	

エンドポイント EPr の割り込みを表示します。

#### Bit7-5 Reserved

#### Bit4 D\_EPeIntStat

割り込み要因を間接指示します。

D\_EPeIntStat レジスタに割り込み要因があり、かつその割り込み要因に対応する D\_EPeIntEnb レジスタのビットがイネーブルにされている時、"1"にセットされます。

#### Bit3 D\_EPdIntStat

割り込み要因を間接指示します。

D\_EPdIntStat レジスタに割り込み要因があり、かつその割り込み要因に対応する D\_EPdIntEnb レジスタのビットがイネーブルにされている時、"1"にセットされます。

#### Bit2 D\_EPcIntStat

割り込み要因を間接指示します。

D\_EPcIntStat レジスタに割り込み要因があり、かつその割り込み要因に対応する D\_EPcIntEnb レジスタのビットがイネーブルにされている時、"1"にセットされます。

#### Bit1 D\_EPbIntStat

割り込み要因を間接指示します。

D\_EPbIntStat レジスタに割り込み要因があり、かつその割り込み要因に対応する D\_EPbIntEnb レジスタのビットがイネーブルにされている時、"1"にセットされます。

#### Bit0 D\_EPaIntStat

割り込み要因を間接指示します。

D\_EPaIntStat レジスタに割り込み要因があり、かつその割り込み要因に対応する D\_EPaIntEnb レジスタのビットがイネーブルにされている時、"1"にセットされます。

**2.7.5. 0E5h D\_EP0IntStat (Device EP0 Interrupt Status)****2.7.6. 0E6h D\_EPaIntStat (Device EPa Interrupt Status)****2.7.7. 0E7h D\_EPbIntStat (Device EPb Interrupt Status)****2.7.8. 0E8h D\_EPCIntStat (Device EPc Interrupt Status)****2.7.9. 0E9h D\_EPdIntStat (Device EPd Interrupt Status)****2.7.10. 0EAh D\_EPeIntStat (Device EPe Interrupt Status)**

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device	0E5h	D_EP0IntStat		7:	0:	1:	00h
	0E6h	D_EPaIntStat	R (W)	6: OUT_ShortACK	0: None	1: OUT Short-Packet ACK	
	0E7h	D_EPbIntStat	R (W)	5: IN_TrانACK	0: None	1: IN Transaction ACK	
	0E8h	D_EPCIntStat	R (W)	4: OUT_TrانACK	0: None	1: OUT Transaction ACK	
	0E9h	D_EPdIntStat	R (W)	3: IN_TrانNAK	0: None	1: IN Transaction NAK	
	0EAh	D_EPeIntStat	R (W)	2: OUT_TrانNAK	0: None	1: OUT Transaction NAK	
			R (W)	1: IN_TrانErr	0: None	1: IN Transaction Error	
			R (W)	0: OUT_TrانErr	0: None	1: OUT Transaction Error	

エンドポイント EPx{x=0,a-e}の割り込みステータスを表示します。全てのビットは"1"を書き込む事で割り込み要因をクリアできます。

**Bit7 Reserved****Bit6 OUT\_ShortACK**

割り込み要因を直接指示します。

OUT トランザクションでショートパケットを受信し、ACK を返信した時、OUT\_TrانACK と同時に"1"にセットされます。

**Bit5 IN\_TrانACK**

割り込み要因を直接指示します。

IN トランザクションで ACK を受信した時、"1"にセットされます。

**Bit4 OUT\_TrانACK**

割り込み要因を直接指示します。

OUT トランザクションで ACK を返信した時、"1"にセットされます。

**Bit3 IN\_TrانNAK**

割り込み要因を直接指示します。

IN トランザクションで NAK を返信した時、"1"にセットされます。

**Bit2 OUT\_TrانNAK**

割り込み要因を直接指示します。

OUT トランザクション及び PING トランザクションに対して NAK を返信した時、"1"にセットされます。

**Bit1 IN\_TrانErr**

割り込み要因を直接指示します。

IN トランザクションにおいて STALL を返した場合、パケットにエラーがあった場合、及びハンドシェイクがタイムアウトになった場合に、"1"にセットされます。

**Bit0 OUT\_TrانErr**

割り込み要因を直接指示します。

OUT トランザクションにおいて STALL を返信した場合、及び、パケットにエラーがあった場合に、"1"にセットされます。

## 2. レジスタ

### 2.7.11. 0F0h *D\_SIE\_IntEnb* (Device SIE Interrupt Enable)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device	0F0h	<i>D_SIE_IntEnb</i>		7:	0:	1:	00h
			R / W	6: <i>EnNonJ</i>	0: Disable	1: Enable	
			R / W	5: EnRcvSOF	0: Disable	1: Enable	
			R / W	4: EnDetectRESET	0: Disable	1: Enable	
			R / W	3: EnDetectSUSPEND	0: Disable	1: Enable	
			R / W	2: EnChirpCmp	0: Disable	1: Enable	
			R / W	1: EnRestoreCmp	0: Disable	1: Enable	
			R / W	0: EnSetAddressCmp	0: Disable	1: Enable	

*D\_SIE\_IntStat* レジスタの割り込み要因による、*MainIntStat* レジスタの *D\_SIE\_IntStat* ビットのアサートを許可／禁止します。

*EnNonJ* ビットは SLEEP / SNOOZE 中も有効です。

同期ビット(Bit5～0)はACTIVE60 / ACT\_HOSTステートの時、読み出しはできますが、書き込みできません。ACT\_DEVICEステート、または、ACT\_ALLステートから、それ以外のステートに遷移する際の処理に関しては、「2.7.1.0E0h *D\_SIE\_IntStat* (Device SIE Interrupt Status)」をご参照下さい。

## 2.7.12. 0F2h D\_FIFO\_IntEnb (Device FIFO Interrupt Enable)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device	0F2h	D_FIFO_IntEnb	R / W	7: EnDescriptorCmp	0: Disable	1: Enable	00h
			R / W	6: EnFIFO_IDE_Cmp	0: Disable	1: Enable	
			R / W	5: EnFIFO1_Cmp	0: Disable	1: Enable	
			R / W	4: EnFIFO0_Cmp	0: Disable	1: Enable	
				3:	0:	1:	
			R / W	2: EnFIFO_NotEmpty	0: Disable	1: Enable	
			R / W	1: EnFIFO_Full	0: Disable	1: Enable	
			R / W	0: EnFIFO_Empty	0: Disable	1: Enable	

D\_FIFO\_IntStat レジスタの割り込み要因による、MainIntStat レジスタの D\_FIFO\_IntStat ビットのアサートを許可／禁止します。

## 2. レジスタ

---

### 2.7.13. 0F3h D\_BulkIntEnb (Device Bulk Interrupt Enable)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device	0F3h	D_BulkIntEnb	R / W	7: EnCBW_Cmp	0: Disable	1: Enable	00h
			R / W	6: EnCBW_LengthErr	0: Disable	1: Enable	
			R / W	5: EnCBW_Err	0: Disable	1: Enable	
				4:	0:	1:	
			R / W	3: EnCSW_Cmp	0: Disable	1: Enable	
			R / W	2: EnCSW_Err	0: Disable	1: Enable	
				1:	0:	1:	
				0:	0:	1:	

D\_BulkIntStat レジスタの割り込み要因による、MainIntStat レジスタの D\_BulkIntStat ビットのアサート許可／禁止します。

## 2.7.14. 0F4h D\_EPrIntEnb (Device EPr Interrupt Enable)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device	0F4h	D_EPrIntEnb		7: (Reserved)	0:	1:	00h
				6: (Reserved)	0:	1:	
				5: (Reserved)	0:	1:	
			R / W	4: EnD_EPeIntStat	0: Disable	1: Enable	
			R / W	3: EnD_EPdIntStat	0: Disable	1: Enable	
			R / W	2: EnD_EPcIntStat	0: Disable	1: Enable	
			R / W	1: EnD_EPbIntStat	0: Disable	1: Enable	
			R / W	0: EnD_EPaIntStat	0: Disable	1: Enable	

D\_EPrIntStat レジスタの割り込み要因による、MainIntStat レジスタの D\_EPrIntStat ビットのアサートを許可／禁止します。

## 2. レジスタ

2.7.15. 0F5h D\_EP0IntEnb (Device EP0 Interrupt Enable)

2.7.16. 0F6h D\_EPaIntEnb (Device EPa Interrupt Enable)

2.7.17. 0F7h D\_EPbIntEnb (Device EPb Interrupt Enable)

2.7.18. 0F8h D\_EPCIntEnb (Device EPc Interrupt Enable)

2.7.19. 0F9h D\_EPdIntEnb (Device EPd Interrupt Enable)

2.7.20. 0FAh D\_EPeIntEnb (Device EPe Interrupt Enable)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device	0F5h	D_EP0IntEnb		7:	0:	1:	00h
	0F6h	D_EPaIntEnb	R / W	6: EnOUT_ShortACK	0: Disable	1: Enable	
	0F7h	D_EPbIntEnb	R / W	5: EnIN_TranACK	0: Disable	1: Enable	
	0F8h	D_EPCIntEnb	R / W	4: EnOUT_TranACK	0: Disable	1: Enable	
	0F9h	D_EPdIntEnb	R / W	3: EnIN_TranNAK	0: Disable	1: Enable	
	0FAh	D_EPeIntEnb	R / W	2: EnOUT_TranNAK	0: Disable	1: Enable	
			R / W	1: EnIN_TranErr	0: Disable	1: Enable	
			R / W	0: EnOUT_TranErr	0: Disable	1: Enable	

D\_EP0IntStat レジスタの割り込み要因による、MainIntStat レジスタの D\_EP0IntStat ビットのアサートを許可／禁止します。

D\_EPx{x=a-e}IntStat レジスタの割り込み要因による、D\_EPrIntStat の D\_EPx{x=a-e}IntStat ビットのアサートを許可／禁止します。



2.7.21. 100h *D\_Reset* (Device Reset)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device	100h	<i>D_Reset</i>		7:	0:	1:	01h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
			R / W	0: <i>ResetDTM</i>	0: Do nothing	1: Reset DTM	

デバイスのトランシーバマクロをリセットします。  
SLEEP / SNOOZE 時でもアクセス有効です。

**Bit7-1**    **Reserved**

**Bit0**      **ResetDTM**

このビットに"1"をセットすると、本 LSI のデバイストランシーバマクロを初期化します。  
リセットを解除するには、このビットを"0"にクリアして下さい。

## 2. レジスタ

### 2.7.22. 102h D\_NegoControl (Device Negotiation Control)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device	102h	D_NegoControl	R / W	7: DisBusDetect	0: Enable BusDetect	1: Disable BusDetect	00h
			R / W	6: EnAutoNego	0: Disable AutoNegotiation	1: Enable AutoNegotiation	
			R / W	5: InSUSPEND	0: Do nothing	1: Monitor NonJ	
			R / W	4: DisableHS	0: HS mode	1: Disable HS mode	
			R / W	3: SendWakeup	0: Do nothing	1:Send       Remotewakeup Signal	
			R / W	2: RestoreUSB	0: Do nothing	1: Restore operation mode	
			R / W	1: GoChirp	0: Do nothing	1: Do Chirp sequence	
			R / W	0: ActiveUSB	0: Disactivate USB	1: Activate USB	

デバイスのネゴシエーション に関する動作設定を行います。

#### Bit7 DisBusDetect

このビットを"1"にセットすると、USB のリセット／サスペンドステートの自動検出を無効にします。このビットが"0"にクリアされている場合、USB のリセット／サスペンドステートの検出のため、USB バス上のバス・アクティビティを監視します。

"HS"モード時は、バス・アクティビティが 3ms の期間検出されない場合、自動的に"FS"モードに切り替えたのち、USB のリセットあるいはサスペンドステートの判定を行い、その後該当する割り込み要因(DetectReset、DetectSuspend)をセットします。"FS"モード時はバス・アクティビティが 3ms の期間検出されないと USB のサスペンドステートと判定し、また、2.5  $\mu$ s 以上の"SE0"を検出するとリセットと判断し、該当する割り込み要因をセットします。

DetectReset、DetectSuspend のビットが"1"にセットされたら DisBusDetect ビットを"1"にセットして USB のリセット／サスペンドステートが継続している間、検出を無効にして下さい。AutoNegotiation 機能を使用する場合、このビットに"1"をセットしないようにして下さい。

#### Bit6 EnAutoNego

AutoNegotiation機能を有効にします。AutoNegotiation機能は、リセット検出時に、スピードネゴシエーションが終了してスピードモードが決定するまでのシーケンスを自動化します。AutoNegotiation機能の詳細は、「1.2.7.オート・ネゴシエーション機能」を参照して下さい。

#### Bit5 InSUSPEND

AutoNegotiation 機能使用時に、USB のサスペンドステートを検出すると自動的に"1"にセットされ NonJ ステートの検出機能を有効にします。USB のサスペンドステートから復帰する場合には、このビットを"0"にクリアして下さい。

AutoNegotiation機能を使用する場合の説明は、「1.2.7.オート・ネゴシエーション機能」をご参照下さい。

#### Bit4 DisableHS

GoChirp が"1"にセットされた時に、このビットが"1"にセットされている時には、DeviceChirp を送出せずに強制的に FS モードとなり、ChirpCmp 割り込みを発生します。

#### Bit3 SendWakeup

このビットを"1"にセットすると、USB ポートに RemoteWakeup 信号(K)を出力します。

RemoteWakeup 信号の送出開始から 1ms 以上 15ms 以内経過後、このビットを"0"にクリアして送出を停止して下さい。

#### Bit2 RestoreUSB

USB のサスペンドステートからリジュームする際に、このビットを"1"にセットすると、USB のサスペンド前に保存された動作モード(FS or HS)に自動的に切り替えられ、該当する割り込み要因(RestoreCmp)がセットされます。

このビットは、動作終了後自動的に"0"にクリアされます。

AutoNegotiation 機能を使用する場合、このビットの機能は自動的に制御されますので、この

ビットをセット／クリアしないで下さい。

**Bit1 GoChirp**

USB バスがリセット状態である場合に、このビットに"1"をセットすると、ホスト／ハブとの間で"HS Detection Handshake"を行い、XcvrControl レジスタの TermSelect ビット、XcvrSelect ビット及び USB\_Status レジスタの FSxHS ビットが自動的に設定されます。動作終了と同時に割り込み要因(ChirpCmp)がセットされます。

このビットは、動作終了後自動的に"0"にクリアされます。動作終了後 USBStauts レジスタの FSxHS ビットを参照することで、"HS Detection Handshake"の結果が確認できます。

AutoNegotiation 機能を使用する場合、このビットの機能は自動的に制御されますので、このビットをセット／クリアしないで下さい。

**Bit0 ActiveUSB**

本 LSI では、このビットがハードリセット後"0"にクリアされているため、USB デバイスの全機能を停止しています。本 LSI の設定終了後に、本ビットを"1"にセットすることで、USB デバイスとしての動作が可能となります。

## 2. レジスタ

### 2.7.23. 105h D\_XcvrControl (Device Xcvr Control)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device	105h	D_XcvrControl	R / W	7: TermSelect	0: HS Termination	1: FS Termination	41h
			R / W	6: XcvrSelect	0: HS Transceiver	1: FS Transceiver	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
			R / W	1: OpMode[1]	OpMode[1:0]		
				0: OpMode[0]			

デバイスのトランシーバマクロに関する設定を行います。

#### Bit7 TermSelect

FS または HS いずれかのターミネーションを選択して有効にします。USB\_Control レジスタの GoChirp ビットによって"HS detection handshake"を実行した場合、または、D\_NegoControl レジスタの EnAutoNego ビットがセットされ、AutoNegotiation 機能が実行された場合、このビットは自動的に設定されます。

#### Bit6 XcvrSelect

FS または HS いずれかのトランシーバを選択して有効にします。D\_NegoControl レジスタの GoChirp ビットによって"HS detection handshake"を実行した場合、または、D\_NegoControl レジスタの EnAutoNego ビットがセットされ、AutoNegotiation 機能が実行された場合、このビットは自動的に設定されます。

#### Bit5-2 Reserved

#### Bit1-0 OpMode

UTM のオペレーションモードを設定します。

USB ケーブルが抜かれている時(※)、USB のサスペンド状態になる時、またはテストモード時以外には、通常設定する必要がありません。

OpMode		
00	"Normal Operation"	通常使用状態
01	"Non-Driving"	USB ケーブルが抜かれている時にはこの状態にして下さい。
10	"Disable Bitstuffing and NRZI encoding"	USB テストモード時にはこの状態にしてください。
11	"Power-Down"	USB のサスペンド時にはこの状態にしてください。

※USB ケーブルが抜けている時には、このレジスタを"41h"にセットすることを推奨します。

## 2.7.24. 106h D\_USB\_Test (Device USB\_Test)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device	106h	D_USB_Test	R / W	7: EnHS_Test	0: Do nothing	1: EnHS_Test	00h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
			R / W	3: Test_SE0_NAK	0: Do nothing	1: Test_SE0_NAK	
			R / W	2: Test_J	0: Do nothing	1: Test_J	
			R / W	1: Test_K	0: Do nothing	1: Test_K	
			R / W	0: Test_Packet	0: Do nothing	1: Test_Packett	

デバイスの USB 2.0 のテストモードに関する動作設定を行います。SetFeature リクエストで指定されたテストモードに対応するビットを設定し、ステータスステージ終了後に EnHS\_Test ビットに"1"をセットすることにより、USB2.0 の規格で定義されたテストモードの動作を行うようにして下さい。

**Bit7 EnHS\_Test**

このビットに"1"をセットすると、D\_USB\_Test レジスタの下位 4 ビットのいずれかのビットに"1"が設定されている場合、そのビットに対応するテストモードに入ります。テストモードを行う際には、D\_NegoControl レジスタの DisBusDetect ビットを"1"にして USB のサスペンドとリセットの検出を行わないようにする必要があります。また、D\_NegoControl レジスタの EnAutoNego ビットを"0"にクリアして、AutoNegotiation 機能を無効にして下さい。

また、テストモードへの移行は、SetFeature リクエストにおけるステータスステージの終了後に行うように、ご注意ください。

**Bit6-4 Reserved****Bit3 Test\_SE0\_NAK**

このビットを"1"に設定し、EnHS\_Test ビットに"1"をセットすることにより、Test\_SE0\_NAK テストモードに入ることができます。

**Bit2 TEST\_J**

このビットを"1"に設定し、EnHS\_Test ビットに"1"をセットすることにより、Test\_J テストモードに入ることができます。なお、このテストモードでは、EnHS\_Test ビットを"1"にセットする前に、XcvrControl レジスタの、TermSelect 及び XcvrSelect をスピードに従って設定し、また、OpMode を"10"(Disable Bitstuffing and NRZI encoding)にセットして下さい。

**Bit1 TEST\_K**

このビットを"1"に設定し、EnHS\_Test ビットに"1"をセットすることにより、Test\_K テストモードに入ることができます。なお、このテストモードでは、EnHS\_Test ビットを"1"にする前に、XcvrControl レジスタの、TermSelect 及び XcvrSelect をスピードに従って設定し、また、OpMode を"10"(Disable Bitstuffing and NRZI encoding)にセットして下さい。

**Bit0 Test\_Packet**

このビットを"1"に設定し、EnHS\_Test ビットに"1"をセットすることにより、Test\_Packet テストモードに入ることができます。

このテストモードは EP0 以外の任意のエンドポイントで使用できますので、下記の設定を行って下さい。

- 1) エンドポイント EPx{x=a-e}の MaxPacketSize を 64 以上、転送方向を IN に設定し、EndpointNumber を"0xF"に設定して、使用可能として下さい。また、エンドポイント EPx{x=a-e}の FIFO を 64Byte 以上、割り当てて下さい。
- 2) 他のエンドポイントの設定を、上記 EPx{x=a-e}の設定と重複しないようにして下さい。

または、AREAx{x=0-5}Join\_1 レジスタの各ビットをクリアして下さい。

## 2. レジスタ

---

3) EPx{x=a-e}の FIFO をクリアし、下記のテストパケット用のデータをこの FIFO に書き込んで下さい。

EPx{x=a-e}IntStat レジスタの IN\_TrnErr ビットを"0"にクリアして下さい。

4) Test Packet の送信完了の毎に、IN\_TrnErr ステータスが"1"にセットされます。パケット送信テストモード時に FIFO に書き込むデータは以下の 53 バイトです。

00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h,  
00h, AAh, AAh, AAh, AAh, AAh, AAh, AAh,  
AAh, EEh, EEh, EEh, EEh, EEh, EEh, EEh,  
EEh, FEh, FFh, FFh, FFh, FFh, FFh, FFh,  
FFh, FFh, FFh, FFh, FFh, 7Fh, BFh, DFh,  
EFh, F7h, FBh, FDh, FCh, 7Eh, BFh, DFh,  
EFh, F7h, FBh, FDh, 7Eh

テストパケット送出時に、SIE が PID と CRC を付加しますので、FIFO に書き込むデータは、USB 規格 Rev.2.0 に記載されているテストパケットデータのうち、DATA0 PID の次のデータから、CRC16 以外のデータまでとなります。

## 2.7.25. 108h D\_EPnControl (Device Endpoint Control)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device	108h	D_EPnControl	W	7: AllForceNAK	0: Do nothing	1: Set All ForceNAK	XXh
			W	6: EPrForceSTALL	0: Do nothing	1: Set EP's ForceSTALL	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
				0:	0:	1:	

エンドポイントの動作設定を行います。ライトオンリーのレジスタです。

**Bit7 AllForceNAK**

全てのエンドポイントの ForceNAK ビットを"1"にセットします。

**Bit6 EPrForceSTALL**

エンドポイント EP0 を除く全てのエンドポイントの ForceSTALL ビットを"1"にセットします。

**Bit5-0 Reserved**

## 2. レジスタ

### 2.7.26. 10Ah D\_BulkOnlyControl (Device BulkOnly Control)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device	10Ah	D_BulkOnlyControl	R / W	7:AutoForceNAK_CBW	0: None	1: AutoForceNAK after CBW	00h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
			R / W	2: GoCBW_Mode	0: None	1: Begin CBW Mode	
			R / W	1: GoCSW_Mode	0: None	1: Begin CSW Mode	
				0:	0:	1:	

バルクオンリーサポート機能を制御します。

#### Bit7 AutoForceNAK\_CBW

このビットを"1"にセットすると、CBW サポートによって CBW の受信する OUT トランザクションが完了すると、該当するエンドポイントの ForceNAK ビットを"1"にセットします。

#### Bit6-3 Reserved

#### Bit2 GoCBW\_Mode

このビットを"1"にセットすると、該当するエンドポイントでCBWサポートを実行します。CBWサポートを実行するエンドポイントについては、D\_BulkOnlyConfig レジスタの項(「2.7.26.10Ah D\_BulkOnlyControl (Device BulkOnly Control)」)を参照して下さい。

#### Bit1 GoCSW\_Mode

このビットを"1"にセットすると、該当するエンドポイントでCSWサポートを実行します。CSWサポートを実行するエンドポイントについては、D\_BulkOnlyConfig レジスタの項(「2.7.27.10Bh D\_BulkOnlyConfig (Device BulkOnly Configuration)」)を参照して下さい。

#### Bit0 Reserved



## 2.7.27. 10Bh D\_BulkOnlyConfig (Device BulkOnly Configuration)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device	10Bh	D_BulkOnlyConfig		7:	0:	1:
				6:	0:	1:
				5:	0:	1:
			R / W	4: EPeBulkOnly	0: None	1: Enable BulkOnly on EPe
			R / W	3: EPdBulkOnly	0: None	1: Enable BulkOnly on EPd
			R / W	2: EPcBulkOnly	0: None	1: Enable BulkOnly on EPc
			R / W	1: EPbBulkOnly	0: None	1: Enable BulkOnly on EPb
			R / W	0: EPaBulkOnly	0: None	1: Enable BulkOnly on EPa

バルクオンリーサポート機能を有効にします。

**Bit7-5 Reserved****Bit4 EPeBulkOnly**

このビットを"1"にセットすると、エンドポイント EPe でバルクオンリーサポート機能が有効になります。バルクオンリーサポートが有効にされると、エンドポイント EPe が OUT のエンドポイントである場合、BulkOnlyControl.GoCBW\_Mode ビットをセットすることによって、CBW サポートを行います。また、エンドポイント EPe が IN のエンドポイントである場合、BulkOnlyControl.GoCSW\_Mode ビットをセットすることによって、CSW サポートを行います。

同時に 2 つ以上の OUT のエンドポイントでバルクオンリーサポート機能を有効にしないで下さい。

**Bit3 EPdBulkOnly**

このビットを"1"にセットすると、エンドポイント EPd でバルクオンリーサポート機能が有効になります。バルクオンリーサポートが有効にされると、エンドポイント EPd が OUT のエンドポイントである場合、BulkOnlyControl.GoCBW\_Mode ビットをセットすることによって、CBW サポートを行います。また、エンドポイント EPd が IN のエンドポイントである場合、BulkOnlyControl.GoCSW\_Mode ビットをセットすることによって、CSW サポートを行います。

同時に 2 つ以上の OUT のエンドポイントでバルクオンリーサポート機能を有効にしないで下さい。

**Bit2 EPcBulkOnly**

このビットを"1"にセットすると、エンドポイント EPc でバルクオンリーサポート機能が有効になります。バルクオンリーサポートが有効にされると、エンドポイント EPc が OUT のエンドポイントである場合、BulkOnlyControl.GoCBW\_Mode ビットをセットすることによって、CBW サポートを行います。また、エンドポイント EPc が IN のエンドポイントである場合、BulkOnlyControl.GoCSW\_Mode ビットをセットすることによって、CSW サポートを行います。

同時に 2 つ以上の OUT のエンドポイントでバルクオンリーサポート機能を有効にしないで下さい。

**Bit1 EPbBulkOnly**

このビットを"1"にセットすると、エンドポイント EPb でバルクオンリーサポート機能が有効になります。バルクオンリーサポートが有効にされると、エンドポイント EPb が OUT のエンドポイントである場合、BulkOnlyControl.GoCBW\_Mode ビットをセットすることによって、CBW サポートを行います。また、エンドポイント EPb が IN のエンドポイントである場合、BulkOnlyControl.GoCSW\_Mode ビットをセットすることによって、CSW サポートを行います。

同時に 2 つ以上の OUT のエンドポイントでバルクオンリーサポート機能を有効にしないで下さい。

**Bit0 EPaBulkOnly**

このビットを"1"にセットすると、エンドポイント EPa でバルクオンリーサポート機能が有

## 2. レジスタ

---

効になります。バルクオンリーサポートが有効にされると、エンドポイント **EPa** が **OUT** のエンドポイントである場合、**BulkOnlyControl.GoCBW\_Mode** ビットをセットすることによって、**CBW** サポートを行います。また、エンドポイント **EPa** が **IN** のエンドポイントである場合、**BulkOnlyControl.GoCSW\_Mode** ビットをセットすることによって、**CSW** サポートを行います。

同時に 2 つ以上の **OUT** のエンドポイントでバルクオンリーサポート機能を有効にしないで下さい。

- 2.7.28. 110h D\_EP0SETUP\_0 (Device EP0 SETUP 0)  
 2.7.29. 111h D\_EP0SETUP\_1 (Device EP0 SETUP 1)  
 2.7.30. 112h D\_EP0SETUP\_2 (Device EP0 SETUP 2)  
 2.7.31. 113h D\_EP0SETUP\_3 (Device EP0 SETUP 3)  
 2.7.32. 114h D\_EP0SETUP\_4 (Device EP0 SETUP 4)  
 2.7.33. 115h D\_EP0SETUP\_5 (Device EP0 SETUP 5)  
 2.7.34. 116h D\_EP0SETUP\_6 (Device EP0 SETUP 6)  
 2.7.35. 117h D\_EP0SETUP\_7 (Device EP0 SETUP 7)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device	110h -117h	D_EP0SETUP_0 -D_EP0SETUP_7	R	7: EP0SETUP_n[7]	Endpoint 0 SETUP Data 0 -Endpoint 0 SETUP Data 7	00h
				6: EP0SETUP_n[6]		
				5: EP0SETUP_n[5]		
				4: EP0SETUP_n[4]		
				3: EP0SETUP_n[3]		
				2: EP0SETUP_n[2]		
				1: EP0SETUP_n[1]		
				0: EP0SETUP_n[0]		

エンドポイント EP0 のセットアップステージで受信した 8 バイトのデータが、EP0SETUP\_0 から順に格納されます。

#### EP0SETUP\_0

bmRequestType がセットされます。

#### EP0SETUP\_1

bRequest がセットされます。

#### EP0SETUP\_2

wValue の下位 8 ビットがセットされます。

#### EP0SETUP\_3

wValue の上位 8 ビットがセットされます。

#### EP0SETUP\_4

wIndex の下位 8 ビットがセットされます。

#### EP0SETUP\_5

wIndex の上位 8 ビットがセットされます。

#### EP0SETUP\_6

wLength の下位 8 ビットがセットされます。

#### EP0SETUP\_7

wLength の上位 8 ビットがセットされます。

## 2. レジスタ

### 2.7.36. 118h D\_USB\_Address (Device USB Address)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device	118h	D_USB_Address	R / W	7: SetAddress	0: None	1: Set USB Address	00h
			R ( W )	6: USB_Address[6]	USB Address		
				5: USB_Address[5]			
				4: USB_Address[4]			
				3: USB_Address[3]			
				2: USB_Address[2]			
				1: USB_Address[1]			
				0: USB_Address[0]			

AutoSetAddress 機能により、USB アドレスが設定されます。

SetAddress() リクエストを受信すると、AutoSetAddress 機能はそのコントロール転送を自動的に行います。AutoSetAddress 機能は、SetAddress() リクエストに関わるコントロール転送のステータスステージが完了し、USB\_Address をセットした後に、SetAddressCmp ステータスを発行します。

#### Bit7 SetAddress

SetAddress リクエストの受信時に、このビットに"1"をセットすると、同リクエストのステータスステージが完了した時に、USB\_Address が自動的にセットされます。自動アドレス設定モードが無効の場合に、このビットの設定が有効になります。詳細は、「Appendix D SetAddress リクエストの応答について」を参照して下さい。

#### Bit6-0 USB\_Address

USB アドレスが設定されます。

AutoSetAddress 機能によって自動的に書き込まれます。

また、書き込みが可能ですが、SetAddress() リクエストを受信すると、再度自動的に書き換えます。

## 2.7.37. 11Ah D\_SETUP\_Control(Device SETUP Control)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device	11Ah	D_SETUP_Control		7:	0:	1:	00h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
			R / W	0: ProtectEP0	0: None	1: Protect EP0	

コントロール転送関係の設定をします。

**Bit7-1**    **Reserved**

**Bit0**     **ProtectEP0**

コントロール転送のセットアップステージが終了し、受信したデータが D\_EP0SETUP\_0～D\_EP0SETUP\_7 レジスタに格納されると、"1"にセットされます。

同時に D\_EP0ControlIN,D\_EP0ControlOUT レジスタの ForceSTALL ビットが"0"に、ForceNAK ビットが"1"に、ToggleStat ビットが"1"に、自動的に設定されます。

ProtectEP0 ビットは SETUP トランザクションが行われるとセットされます。従って、SetAddress()リクエストに対してもセットされます。

このビットが"1"にセットされていると、EP0 の ForceNAK ビット、ForceSTALL ビットの設定変更ができません。

## 2. レジスタ

### 2.7.38. 11Eh D\_FrameNumber\_H (Device FrameNumber High)

### 2.7.39. 11Fh D\_FrameNumber\_L (Device FrameNumber Low)

Mode	Addresses	Register Name	R / W	Bit Symbol	Description		Reset
Device	11Eh	D_FrameNumber_H	R	7: FnInvalid	0: Frame number is valid	1: Frame number is not valid	80h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
			R	2: FrameNumber[10]	Frame Number High		
				1: FrameNumber[9]			
				0: FrameNumber[8]			

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device	11Fh	D_FrameNumber_L	R	7: FrameNumber[7]	Frame Number Low	00h
				6: FrameNumber[6]		
				5: FrameNumber[5]		
				4: FrameNumber[4]		
				3: FrameNumber[3]		
				2: FrameNumber[2]		
				1: FrameNumber[1]		
				0: FrameNumber[0]		

SOF トークンを受信する毎に更新される、USB のフレームナンバーが表示されます。フレームナンバーを取得する場合は、FrameNumber\_H と FrameNumber\_L レジスタを対でアクセスする必要があります。

#### 11Eh.Bit7 FnInvalid

受信した SOF パケットにエラーが発生した時に、このビットが"1"にセットされます。

#### 11Eh.Bit6-3 Reserved

#### 11Eh.Bit2-0, 11Fh.Bit7-0 FrameNumber[10:0]

受信した SOF パケットの FrameNumber が表示されます。

## 2.7.40. 120h D\_EP0MaxSize (Device EP0 Max Packet Size)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device	120h	D_EP0MaxSize		7:	0:	1:
			R / W	6: EP0MaxSize[6]	Endpoint[0] Max Packet Size	
				5: EP0MaxSize[5]		
				4: EP0MaxSize[4]		
				3: EP0MaxSize[3]		
				2:	0:	1:
				1:	0:	1:
				0:	0:	1:

エンドポイント EP0 の設定を行います。

**Bit7**      **Reserved**

**Bit6-3**    **EP0MaxSize[6:3]**

    エンドポイント EP0 の MaxPacketSize を設定します。

    このエンドポイントは、以下のサイズから任意のサイズを選択して使用可能です。

        FS 時      8, 16, 32, 64 バイト

        HS 時      64 バイト

**Bit2-0**    **Reserved**

## 2. レジスタ

### 2.7.41. 121h D\_EP0Control (Device EP0 Control)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device	121h	D_EP0Control	R / W	7: INxOUT	0: OUT	1: IN	00h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
			R / W	0: ReplyDescriptor	0: Do nothing	1: Reply Descriptor	

エンドポイント EP0 の設定を行います。

#### Bit7 INxOUT

エンドポイント EP0 の転送方向を設定します。

セットアップステージで受信したリクエストを判断して、このビットに値を設定して下さい。

データステージがある場合は、このビットにデータステージにおける転送方向をセットして下さい。セットアップステージが完了することにより、D\_EP0ControlIN 及び D\_EP0ControlOUT レジスタの ForceNAK ビットがセットされるので、データステージ及びステータスステージの実行時にクリアして下さい。

データステージが終了したら、ステータスステージの方向に合わせて、このビットを設定しなおして下さい。データステージの転送方向が IN の場合は、ステータスステージは OUT 方向となりますので、このビットに"0"を設定して下さい。また、データステージの転送方向が OUT、またはデータステージがない場合は、ステータスステージは IN 方向となりますので、エンドポイント EP0 の FIFO をクリアして、このビットに"1"を設定して下さい。

このビットの設定値と異なる方向の IN または OUT トランザクションに対しては、NAK 応答します。但し、そのトランザクション方向に対応する D\_EP0ControlIN または D\_EP0ControlOUT レジスタの ForceSTALL ビットがセットされていると STALL 応答します。

#### Bit6-1 Reserved

#### Bit0 ReplyDescriptor

Descriptor 返信機能を実行します。

このビットが"1"にセットされると、エンドポイント EP0 の IN トランザクションに応答して、FIFO から Descriptor データを、MaxPacketSize 分返信します。Descriptor データは、DescAdrs\_H,L レジスタの設定値のアドレスを先頭にする、DescSize\_H,L レジスタの設定サイズのデータを指します。これらの設定値は、Descriptor 返信機能の実行中に更新されますので、ReplyDescriptor ビットをセットする毎に設定して下さい。

1 つのトランザクション毎に、DescAdrs\_H,L レジスタは、送信したデータ数だけインクリメントされ、また、DescSize\_H,L レジスタは、送信したデータ数だけデクリメントされます。

DescSize\_H,L の設定数のデータを送信して終了した場合、及び、IN トランザクション以外のトランザクションが行われた場合には、Descriptor 返信機能は終了し、ReplyDescriptor ビットは"0"にクリアされ、D\_FIFO\_IntStat レジスタの DescriptorCmp ビットと D\_EP0IntStat レジスタの IN\_TranACK ビットに"1"がセットされます。

さらに詳細な説明は、「1.2.3.4.デスクリプタ返信機能」動作説明の章を参照して下さい。



## 2.7.42. 122h D\_EP0ControlIN (Device EP0 Control IN)

Mode	Addresses	Register Name	R / W	Bit Symbol	Description		Reset
Device	122h	D_EP0ControlIN		7:	0:	1:	00h
			R / W	6: EnShortPkt	0: Do nothing	1: Enable short Packet	
				5:	0:	1:	
			R	4: ToggleStat	Toggle sequence bit		
			W	3: ToggleSet	0: Do nothing	1: Set Toggle sequence bit	
			W	2: ToggleClr	0: Do nothing	1: Clear Toggle sequence bit	
			R / W	1: ForceNAK	0: Do nothing	1: Force NAK	
			R / W	0: ForceSTALL	0: Do nothing	1: Force STALL	

エンドポイント EP0 の IN トランザクションに関する動作設定及び状態表示を行います。

**Bit7 Reserved****Bit6 EnShortPkt**

このビットを"1"にセットすることで、エンドポイント EP0 の IN トランザクションに対して、MaxPacketSize に満たない FIFO 内のデータをショートパケットとして送信することができます。ショートパケットを送信した IN トランザクションが完了すると、自動的にこのビットが"0"にクリアされます。MaxPacketSize のパケットを送信した場合は、このビットはクリアされません。

FIFO 内にデータが無い場合にこのビットを"1"にセットすると、ホストからの IN トークンに対して Zero 長パケットを送信することができます。このビットをセットしてパケットを送信している最中に、該当 FIFO にデータを書き込むと、タイミングによりそのデータも含めて送信されることがあります。パケットの送信が終了し、このビットがクリアされるまで、FIFO へのデータ書き込みは行わないで下さい。

**Bit5 Reserved****Bit4 ToggleStat**

エンドポイント EP0 の、IN トランザクションのトグルシーケンスビットの状態を示します。

**Bit3 ToggleSet**

エンドポイント EP0 の、IN トランザクションのトグルシーケンスビットを"1"にセットします。ToggleClr ビットと同時にセットした場合、ToggleClr ビットの機能が優先されます。

**Bit2 ToggleClr**

エンドポイント EP0 の、IN トランザクションのトグルシーケンスビットを"0"にクリアします。ToggleSet ビットと同時にセットした場合、このビットの機能が優先されます。

**Bit1 ForceNAK**

このビットを"1"にセットすると、FIFO のデータ数に関わらずエンドポイント EP0 の IN トランザクションに対して NAK 応答します。

セットアップステージが完了することによって MainIntStat レジスタの RcvEP0SETUP ビットに"1"がセットされると、このビットは"1"にセットされ、RcvEP0SETUP ビットが"1"である間、このビットは"0"にクリアできません。また、ショートパケットを送信した IN トランザクションが完了した時、このビットは"1"にセットされます。

このビットを"1"にセットする際に、既にトランザクションが実行中である場合においては、そのトランザクションが終了するまでビットはセットされず、終了と同時にこのビットは"1"にセットされます。トランザクションが実行中で無い場合においては、即座に"1"にセットされます。

**Bit0 ForceSTALL**

このビットを"1"にセットすると、エンドポイント EP0 の IN トランザクションに対して STALL 応答します。このビットは、ForceNAK ビットの設定より優先されます。

## 2. レジスタ

---

セットアップステージが完了することによって、DeviceIntStat レジスタの RcvEP0SETUP ビットに"1"がセットされると、このビットは"0"にクリアされ、RcvEP0SETUP ビットが"1"である間はこのビットを"1"にセットできません。

現在実行中のトランザクションがある場合、トランザクション開始から一定時間後のこのビットの設定は、次のトランザクションから有効になります。

## 2.7.43. 123h D\_EP0ControlOUT (Device EP0 Control OUT)

Mode	Addresses	Register Name	R / W	Bit Symbol	Description		Reset
Device	123h	D_EP0ControlOUT	R / W	7: AutoForceNAK	0: Do nothing	1: Auto Force NAK	00h
				6:	0:	1:	
				5:	0:	1	
			R	4: ToggleStat	Toggle sequence bit		
			W	3: ToggleSet	0: Do nothing	1: Set Toggle sequence bit	
			W	2: ToggleClr	0: Do nothing	1: Clear Toggle sequence bit	
			R / W	1: ForceNAK	0: Do nothing	1: Force NAK	
			R / W	0: ForceSTALL	0: Do nothing	1: Force STALL	

エンドポイント EP0 の OUT トランザクションに関する動作設定及び状態表示を行います。

**Bit7 AutoForceNAK**

エンドポイント EP0 の OUT トランザクションが正常に完結すると、このレジスタの ForceNAK ビットを"1"にセットします。

**Bit6-5 Reserved****Bit4 ToggleStat**

エンドポイント EP0 の、OUT トランザクションのトグルシーケンスビットの状態を示します。

**Bit3 ToggleSet**

エンドポイント EP0 の、OUT トランザクションのトグルシーケンスビットを"1"にセットします。ToggleClr ビットと同時にセットした場合、ToggleClr ビットの機能が優先されます。

**Bit2 ToggleClr**

エンドポイント EP0 の、OUT トランザクションのトグルシーケンスビットを"0"にクリアします。ToggleSet ビットと同時にセットした場合、このビットの機能が優先されます。

**Bit1 ForceNAK**

このビットを"1"にセットすると、FIFO の空き容量に関わらずエンドポイント EP0 の OUT トランザクションに対して NAK 応答します。

セットアップステージが完了することによって DeviceIntStat レジスタの RcvEP0SETUP ビットに"1"がセットされると、このビットは"1"にセットされ、RcvEP0SETUP ビットが"1"である間はこのビットを"0"にクリアすることはできません。

このビットを"1"にセットする際に、既にトランザクションが実行中である場合においては、そのトランザクションが終了するまでビットはセットされず、終了と同時にこのビットは"1"にセットされます。トランザクションが実行中で無い場合においては、即座に"1"にセットされます。

**Bit0 ForceSTALL**

このビットを"1"にセットすると、エンドポイント EP0 の OUT トランザクションに対して STALL 応答します。このビットは、ForceNAK ビットの設定より優先されます。

セットアップステージが完了することによって、DeviceIntStat レジスタの RcvEP0SETUP ビットに"1"がセットされると、このビットは"0"にクリアされ、RcvEP0SETUP ビットが"1"である間はこのビットを"1"にセットすることはできません。

現在実行中のトランザクションがある場合、トランザクション開始から一定時間後のこのビットの設定は、次のトランザクションから有効になります。

## 2. レジスタ

- 2.7.44. 130h D\_EPbMaxSize\_H (Device EPb Max Packet Size High)  
 2.7.45. 131h D\_EPbMaxSize\_L (Device EPb Max Packet Size Low)  
 2.7.46. 140h D\_EPcMaxSize\_H (Device EPc Max Packet Size High)  
 2.7.47. 141h D\_EPcMaxSize\_L (Device EPc Max Packet Size Low)  
 2.7.48. 150h D\_EPdMaxSize\_H (Device EPd Max Packet Size High)  
 2.7.49. 151h D\_EPdMaxSize\_L (Device EPd Max Packet Size Low)  
 2.7.50. 1A0h D\_EPeMaxSize\_H (Device EPe Max Packet Size High)  
 2.7.51. 1A1h D\_EPeMaxSize\_L (Device EPe Max Packet Size Low)  
 2.7.52. 1B0h D\_EPeMaxSize\_H (Device EPe Max Packet Size High)  
 2.7.53. 1B1h D\_EPeMaxSize\_L (Device EPe Max Packet Size Low)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device	130h	D_EPbMaxSize_H		7:	0:	1:
	140h	D_EPcMaxSize_H		6:	0:	1:
	150h	D_EPdMaxSize_H		5:	0:	1:
	1A0h	D_EPeMaxSize_H		4:	0:	1:
	1B0h	D_EPeMaxSize_H		3:	0:	1:
			R / W	2: MaxSize[10] 1: MaxSize[9] 0: MaxSize[8]	Max Packet Size	
						00h

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device	131h	D_EPbMaxSize_L	R / W	7: MaxSize[7]	Max Packet Size	00h
	141h	D_EPcMaxSize_L		6: MaxSize[6]		
	151h	D_EPdMaxSize_L		5: MaxSize[5]		
	1A1h	D_EPeMaxSize_L		4: MaxSize[4]		
	1B1h	D_EPeMaxSize_L		3: MaxSize[3]		
				2: MaxSize[2]		
				1: MaxSize[1]		
				0: MaxSize[0]		

MaxPacketSize を設定します。

### XX0h.Bit7-3 Reserved

### XX0h.Bit2-0, XX1h.Bit7-0 MaxSize[10:0]

エンドポイント EPx{x=a-e}の MaxPacketSize を設定します。

このエンドポイントをバルク転送用として使用する場合には、

FS 時 8, 16, 32, 64 バイト

HS 時 512 バイト

のいずれかに設定して下さい。

このエンドポイントをインタラプト転送用として使用する場合は、

FS 時 64 バイトまで

HS 時 1024 バイトまで

の任意の転送数が設定可能です。

このエンドポイントをアイソクロナス転送用として使用する場合は、

FS 時 1023 バイトまで

HS 時 1024 バイトまで

の任意の転送数が設定可能です。

### XX1h.Bit2-0 Reserved

**2.7.54. 132h D\_EPaConfig\_0 (Device EPa Configuration 0)****2.7.55. 142h D\_EPbConfig\_0 (Device EPb Configuration 0)****2.7.56. 152h D\_EPcConfig\_0 (Device EPc Configuration 0)****2.7.57. 1A2h D\_EPdConfig\_0 (Device EPd Configuration 0)****2.7.58. 1B2h D\_EPeConfig\_0 (Device EPe Configuration 0)**

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device	132h	D_EPaConfig_0	R / W	7: INxOUT	0: OUT	1: IN	00h
	142h	D_EPbConfig_0	R / W	6: IntEP_Mode	0: Normal	1: Special	
	152h	D_EPcConfig_0		5:	0:	1:	
	1A2h	D_EPdConfig_0	R / W	4: ISO	0: Other than Isochronous	1: Isochronous	
	1B2h	D_EPeConfig_0	R / W	3: EndpointNumber[3]	Endpoint Number		
				2: EndpointNumber[2]			
				1: EndpointNumber[1]			
				0: EndpointNumber[0]			

エンドポイント EPx{x=a-e}の設定を行います。

EndpointNumber と INxOUT の組み合わせが、他のエンドポイントと重複しないように設定して下さい。

**Bit7 INxOUT**

エンドポイントの転送方向を設定します。

**Bit6 IntEP\_Mode**

Interrupt 転送に関する設定を行います。

Bulk のエンドポイントでは、このビットに"1"を設定しないで下さい。

このビットの設定は、エンドポイントの方向(IN/OUT)によって異なります(エンドポイントの方向は Bit7"INxOUT"によって設定されます)。

IN 方向(INxOUT = 1)の場合、トグルシーケンスビットの動作モードを設定します。トグルシーケンスの動作モードは、アプリケーションに依存します。Interrupt IN のエンドポイントに対し、どちらかの動作モードを選択して下さい。

0: Normal toggle — 通常のトグルシーケンスを行います。

1: Always toggle — トランザクション毎に常にトグルします。このモードについては、USB2.0 規格書 5.7.5 項をご参照下さい。

OUT 方向(INxOUT = 0)の場合、このエンドポイントにおいて PING フローコントロールを行うか否かを設定します。Interrupt OUT のエンドポイントでは、このビットを"1"にセットして下さい。

0: Bulk OUT — Bulk OUT のエンドポイントはこの設定にして下さい。

1: Interrupt OUT — Interrupt OUT のエンドポイントはこの設定にして下さい。

**Bit5 Reserved****Bit4 ISO**

このビットを"1"にセットすることで、このエンドポイントが Isochronous トランザクションを実行します。

**Bit3-0 EndpointNumber**

0x1~0xF の任意のエンドポイントナンバーを設定します。

## 2. レジスタ

- 2.7.59. 134h D\_EPaControl (Device EPa Control)  
 2.7.60. 144h D\_EPbControl (Device EPb Control)  
 2.7.61. 154h D\_EPcControl (Device EPc Control)  
 2.7.62. 1A4h D\_EPdControl (Device EPd Control)  
 2.7.63. 1B4h D\_EPeControl (Device EPe Control)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device	134h	D_EPaControl	R / W	7: AutoForceNAK	0: Do nothing	1: Auto Force NAK	00h
	144h	D_EPbControl	R / W	6: EnShortPkt	0: Do nothing	1: Enable Short Packet	
	154h	D_EPcControl	R / W	5: DisAF_NAK_Short	0: Auto Force NAK Short	1 Disable Auto Force	
	1A4h	D_EPdControl	R	4: ToggleStat	Toggle sequence bit		
	1B4h	D_EPeControl	W	3: ToggleSet	0: Do nothing	1: Set Toggle sequence bit	
			W	2: ToggleClr	0: Do nothing	1: Clear Toggle sequence bit	
			R / W	1: ForceNAK	0: Do nothing	1: Force NAK	
			R / W	0: ForceSTALL	0: Do nothing	1: Force STALL	

エンドポイント EPx{x=a-e}の動作設定を行います。

### Bit7 AutoForceNAK

エンドポイント EPx{x=a-e}のトランザクションが正常に完結すると、このレジスタの ForceNAK ビットを"1"にセットします。

### Bit6 EnShortPkt

このビットを"1"にセットすることで、エンドポイント EPx{x=a-e}の IN トランザクションに対して、MaxPacketSize に満たない FIFO 内のデータをショートパケットとして送信することができます。ショートパケットを送信した IN トランザクションが完了すると、自動的にこのビットが"0"にクリアされます。マックスパケットサイズのパケットを送信した場合は、このビットはクリアされません。

FIFO 内にデータが無い場合にこのビットを"1"にセットすると、ホストからの IN トークンに対して Zero 長パケットを送信することができます。このビットをセットしてパケットを送信している最中に、該当 FIFO にデータを書き込むと、タイミングによりそのデータも含めて送信されることがあります。パケットの送信が終了し、このビットがクリアされるまで、FIFO へのデータ書き込みは行わないで下さい。

### Bit5 DisAF\_NAK\_Short

Auto Force NAK Short(以下、AF\_NAK\_Short※)機能の有効／無効を設定します。

※正常な OUT トランザクション完結時に受信したパケットがショートパケットの場合、自動的に ForceNAK ビットを"1"にセットする。

デフォルトの設定は AF\_NAK\_Short 機能が有効です。

このビットを"1"にセットすると、AF\_NAK\_Short 機能が無効になります。

AutoForceNAK ビットが"1"にセットされている場合は、AutoForceNAK ビットが優先されます。

### Bit4 ToggleStat

エンドポイント EPx{x=a-e}のトグルシーケンスビットの状態を示します。

### Bit3 ToggleSet

エンドポイント EPx{x=a-e}トグルシーケンスビットを"1"にセットします。ToggleClr ビットと同時にセットした場合、ToggleClr ビットの機能が優先されます。

### Bit2 ToggleClr

エンドポイント EPx{x=a-e}のトグルシーケンスビットを"0"にクリアします。ToggleSet ビットと同時にセットした場合、このビットの機能が優先されます。

### Bit1 ForceNAK

このビットを"1"にセットすると、FIFO のデータ数または空き容量に関わらずエンドポイント EPx{x=a-e}のトランザクションに対して NAK 応答します。

このビットを"1"にセットする際に、既にトランザクションが実行中である場合には、そのトランザクションが終了するまでビットはセットされず、終了と同時にこのビットは"1"にセットされます。トランザクションが実行中で無い場合には、即座に"1"にセットされます。

**Bit0 ForceSTALL**

このビットを"1"にセットすると、エンドポイント EPx{x=a-e}のトランザクションに対して STALL 応答します。このビットは、ForceNAK ビットの設定より優先されます。

現在実行中のトランザクションがある場合、トランザクション開始から一定時間後のこのビットの設定は、次のトランザクションから有効になります。

## 2. レジスタ

### 2.7.64. 160h D\_DescAdrs\_H (Device Descriptor Address High)

### 2.7.65. 161h D\_DescAdrs\_L (Device Descriptor Address Low)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device	160h	D_DescAdrs_H		7:	0:	1:
				6:	0:	1:
				5:	0:	1:
			R / W	4: DescAdrs[12]	Descriptor Address	00h
				3: DescAdrs[11]		
				2: DescAdrs[10]		
				1: DescAdrs[9]		
				0: DescAdrs[8]		

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device	161h	D_DescAdrs_L	R / W	7: DescAdrs[7]	Descriptor Address	00h
				6: DescAdrs[6]		
				5: DescAdrs[5]		
				4: DescAdrs[4]		
				3: DescAdrs[3]		
				2: DescAdrs[2]		
				1: DescAdrs[1]		
				0: DescAdrs[0]		

Descriptor Address を指定します。

#### 160h.Bit7-5 Reserved

#### 160h.Bit4-0, 161h.Bit7-0 DescAdrs[12:0]

Descriptor 返信機能における、Descriptor 返信動作開始時の FIFO の先頭アドレスを指定します。

Descriptor Address は、Descriptor 返信機能に対して FIFO 領域を割り当てるものではありません。Descriptor Address は、FIFO の領域設定に関わらず、0x000 から 0x11FF(4.5kByte)までの FIFO の全領域を指定することが出来ます。

Descriptor返信時には、エンドポイントEP0 におけるINトランザクション完了毎に、送信データ数の分だけDescAdrsは更新されます。Descriptor返信機能については、D\_EP0ControlレジスタのReplyDescriptorの項(「2.7.41.121h D\_EP0Control (Device EP0 Control)」)を参照して下さい。

Descriptor 返信機能用の FIFO 領域は、明示的には割り当てませんので、D\_DescAdrs\_H,L レジスタと D\_DescSize\_H,L レジスタの指定によって、他のエンドポイントの FIFO との重複を避けて下さい。



## 2.7.66. 162h D\_DescSize\_H (Device Descriptor Size High)

## 2.7.67. 163h D\_DescSize\_L (Device Descriptor Size Low)

Mode	Addresses	Register Name	R / W	Bit Symbol	Description		Reset
Device	162h	D_DescSize_H		7:	0:	1:	00h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
		R / W	1: DescSize[9]	Descriptor Size			
			0: DescSize[8]				

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device	163h	D_DescSize_L	R / W	7: DescSize[7]	DescriptorSize	00h
				6: DescSize[6]		
				5: DescSize[5]		
				4: DescSize[4]		
				3: DescSize[3]		
				2: DescSize[2]		
				1: DescSize[1]		
				0: DescSize[0]		

Descriptor Size を指定します。

## 162h.Bit7-2 Reserved

## 162h.Bit1-0, 163h.Bit7-0 DescSize[9:0]

Descriptor Sizeには、Descriptor返信機能において、返信する総データ数を指定します。Descriptor返信機能については、EP0ControlレジスタのReplyDescriptorビットの項(「2.7.41.121h D\_EP0Control (Device EP0 Control)」)を参照して下さい。

Descriptor Size には、FIFO のサイズ及び領域設定に関わらず、0x000 から 0x3FF までの値を指定することが出来ます。Descriptor 返信時には、エンドポイント EP0 における IN トランザクション完了毎に、送信データ数の分だけ DescSize は更新されます。

Descriptor 返信機能用の FIFO 領域は、明示的には割り当てませんので、DescAdrs\_H,L レジスタと DescSize\_H,L レジスタの指定によって、他のエンドポイントの FIFO との重複を避けて下さい。

## 2. レジスタ

### 2.7.68. 170h D\_DMA0\_FIFO\_Control (Device DMA0 FIFO Control)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device	170h	D_DMA0_FIFO_Control	R	7: FIFO_Running	0: FIFO is not running	1: FIFO is running	00h
			R / W	6: AutoEnShort	0: Do nothing	1: Auto Enable Short Packet	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
				0:	0:	1:	

DMA0 転送時の、FIFO の状態の表示及び設定を行います。

#### Bit7 FIFO\_Running

DMA0 に接続されたエンドポイントの FIFO が動作中であることを示します。DMA0 を起動すると"1"にセットされ、DMA0 が終了した後、FIFO が空になると"0"にクリアされます。

#### Bit6 AutoEnShort

DMA0 の終了時に、マックスパケットサイズに満たないデータ数が FIFO に残る場合に、そのエンドポイントの EnShortPkt ビットを"1"にセットします。

DMA0 に接続されたエンドポイントが IN 方向である場合に有効です。

#### Bit5-0 Reserved

## 2.7.69. 172h D\_DMA1\_FIFO\_Control (Device DMA1 FIFO Control)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device	172h	D_DMA1_FIFO_Control	R	7: FIFO_Running	0: FIFO is not running	1: FIFO is running	00h
			R / W	6: AutoEnShort	0: Do nothing	1: Auto Enable Short Packet	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
				0:	0:	1:	

DMA1 転送時の、FIFO の状態の表示及び設定を行います。

**Bit7 FIFO\_Running**

DMA1 に接続されたエンドポイントの FIFO が動作中であることを示します。DMA1 を起動すると"1"にセットされ、DMA1 が終了した後、FIFO が空になると"0"にクリアされます。

**Bit6 AutoEnShort**

DMA1 の終了時に、マックスパケットサイズに満たないデータ数が FIFO に残る場合に、そのエンドポイントの EnShortPkt ビットを"1"にセットします。

DMA1 に接続されたエンドポイントが IN 方向である場合に有効です。

**Bit5-0 Reserved**

## 2. レジスタ

### 2.7.70. 1BEh DTM\_Config (Device Transceiver Macro Config)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Device	1BEh	<b><i>DTM_Config</i></b>		7:	0:	1:	XXh
				6:	0:	1:	
			R / W	5: <b><i>DTM_SlopeValue[1]</i></b>	DTM Slope Value[1:0]		
			R / W	4: <b><i>DTM_SlopeValue[0]</i></b>			
				3:	0:	1:	
				2:	0:	1:	
			R / W	1: <b><i>DTM_TermValue[1]</i></b>	DTM Termination Value[1:0]		
			R / W	0: <b><i>DTM_TermValue[0]</i></b>			

トランシーバマクロの調整値設定用レジスタです。

**Bit7-6**    **Reserved**

**Bit5-4**    **DTM\_SlopeValue[1:0]**

HS トランスミッタのスルーレートを調整します。4 段階の調整が可能です。

00 : 緩

01 : ↑

10 : ↓

11 : 急

**Bit3-2**    **Reserved**

**Bit1-0**    **DTM\_TermValue[1:0]**

HS 伝送路のターミネーションを調整します。4 段階の調整が可能です。

00 : 高

01 : ↑

10 : ↓

11 : 低

## 2.7.71. 1E1h D\_ModeControl (Address Mode Control)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Device	1E1h	D_ModeControl	W	7: (Reserved)	Don't set "1"	XXh
			W	6: (Reserved)	Don't set "1"	
			W	5: (Reserved)	Don't set "1"	
			W	4: SetAddressMode	0: Auto mode      1: Manual mode	
			W	3: (Reserved)	Don't set "1"	
			W	2: (Reserved)	Don't set "1"	
			W	1: (Reserved)	Don't set "1"	
			W	0: (Reserved)	Don't set "1"	

**Bit7-5    Reserved****Bit4      SetAddressMode**

自動アドレス設定機能を無効にします。

**Bit3-0    Reserved**

詳細は、「Appendix D SetAddressリクエストの応答について」を参照して下さい。

## 2. レジスタ

### 2.8. ホスト・レジスタ詳細説明

#### 2.8.1. 0E0h H\_SIE\_IntStat\_0 (Host SIE Interrupt Status 0)

Mode	Addresses	Register Name	R / W	Bit Symbol	Description		Reset
Host	0E0h	H_SIE_IntStat_0		7:	0:	1:	00h
				6:	0:	1:	
				5:	0:	1:	
			R (W)	4: DetectCon	0: None	1: Detect Connect	
			R (W)	3: DetectDiscon	0: None	1: Detect Disconnect	
			R (W)	2: DetectRmtWkup	0: None	1: Detect Remote WakeUp	
			R (W)	1: DetectDevChirpOK	0: None	1: Detect Device Chirp OK	
			R (W)	0: DetectDevChirpNG	0: None	1: Detect Device Chirp NG	

ホスト SIE 関連の割り込みを表示します。

全てのビットは"1"を書き込む事で割り込み要因をクリアできます。

#### Bit7-5 Reserved

#### Bit4 DetectCon

割り込み要因を直接指示します。

USB ケーブルの接続が検出されると"1"にセットされます。

#### Bit3 DetectDiscon

割り込み要因を直接指示します。

USB ケーブルの切断が検出されると"1"にセットされます。

#### Bit2 DetectRmtWkup

割り込み要因を直接指示します。

Suspend 状態で、デバイスからの Remote WakeUp 信号を検出すると"1"にセットされます。

#### Bit1 DetectDevChirpOK

割り込み要因を直接指示します。

デバイスからのチャープ信号が正常であった場合に"1"にセットされます。

また、ホストステートが RESET の状態において、H/W はこのビットによりデバイスからのチャープ信号の有無を判断します。このビットが"1"にセットされている場合は H\_NegoControl\_1.PortSpeed を"00b(HS)"に設定し、以降 HS デバイスに対する処理を行います。この為、デバイスが切断された時には必ずこのビットを"0"にクリアしてください。

#### Bit0 DetectDevChirpNG

割り込み要因を直接指示します。

デバイスからのチャープ信号が異常であった場合に"1"にセットされます。

2ポートモードでは同期ビット(Bit4~0)は ACTIVE60 / ACT\_DEVICE ステートの時、読み出しはできますが、書き込み(割り込み要因クリア)できません。一方、1ポートモードでは同期ビット(Bit4~0)は ACTIVE60 ステートの時、読み出しはできますが、書き込み(割り込み要因クリア)できません。

2ポートモードで ACT\_HOST ステート、または、ACT\_ALL ステートから、それ以外のステートに遷移する際、あるいは1ポートモードで ACT\_DEVICE ステートから、それ以外のステートに遷移する際には、これらの割り込みステータスにより割り込み信号 XINT がアサートされないよう、F/W にて以下の処理を行って下さい。

<ACT\_HOST/ACT\_ALL ステートを出るとき(2ポートモード時)、ACT\_DEVICE ステートを出るとき(1ポートモード時)>

1) 割り込みステータスを処理し、クリアする(H\_SIE\_IntStat\_0.Bit4~0)

2) 割り込みステータスをディスエーブルにする(H\_SIE\_IntEnb\_0.Bit4~0)

＜ACT\_HOST／ACT\_ALL ステートに入るとき(2ポートモード時、ACT\_DEVICE ステートに入るとき(1ポートモード時)＞

- 3) 割り込みステータスをクリアする(H\_SIE\_IntStat\_0.Bit4～0)
- 4) 割り込みステータスをイネーブルにする(H\_SIE\_IntEnb\_0.Bit4～0)

## 2. レジスタ

### 2.8.2. 0E1h H\_SIE\_IntStat\_1 (SIE Host Interrupt Status 1)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Host	0E1h	H_SIE_IntStat_1		7:	0:	1:	00h
				6:	0:	1:	
				5:	0:	1	
				4:	0:	1:	
			R (W)	3: DisabledCmp	0: None	1: Disabled Complete	
			R (W)	2: ResumeCmp	0: None	1: Resume Complete	
			R (W)	1: SuspendCmp	0: None	1: Suspend Complete	
			R (W)	0: ResetCmp	0: None	1: Reset Complete	

ホスト SIE 関連の割り込みを表示します。全てのビットは"1"を書き込む事で割り込み要因をクリアできます。

#### Bit7-4 Reserved

#### Bit3 DisabledCmp

割り込み要因を直接指示します。

H\_NegoControl\_0.AutoMode[3:0]に GoDISABLED を設定してステート管理機能を実行させた場合に、DISABLED ステートへの遷移が完了した時に"1"にセットされます。

#### Bit2 ResumeCmp

割り込み要因を直接指示します。

H\_NegoControl\_0.AutoMode[3:0]に GoRESUME を設定してステート管理機能を実行させた場合に、レジュームが正常に完了した時に"1"にセットされます。

#### Bit1 SuspendCmp

割り込み要因を直接指示します。

H\_NegoControl\_0.AutoMode[3:0]に GoSUSPEND を設定してステート管理機能を実行させた場合に、サスペンドへの遷移が完了した時に"1"にセットされます。

#### Bit0 ResetCmp

割り込み要因を直接指示します。

H\_NegoControl\_0.AutoMode[3:0]に GoRESET を設定してステート管理機能を実行させた場合に、USB リセットが正常に完了した時に"1"にセットされます。

2ポートモードでは同期ビット(Bit3～0)は ACTIVE60 / ACT\_DEVICE ステートの時、読み出しはできませんが、書き込み(割り込み要因クリア)できません。一方、1ポートモードでは同期ビット(Bit4～0)は ACTIVE60 ステートの時、読み出しはできますが、書き込み(割り込み要因クリア)できません。

2ポートモードで ACT\_HOST ステート、または、ACT\_ALL ステートから、それ以外のステートに遷移する際、あるいは1ポートモードで ACT\_DEVICE ステートから、それ以外のステートに遷移する際には、これらの割り込みステータスにより割り込み信号 XINT がアサートされないよう、F/W にて以下の処理を行って下さい。

<ACT\_HOST / ACT\_ALL ステートを出るとき(2ポートモード時)、ACT\_DEVICE ステートを出るとき(1ポートモード時)>

- 1) 割り込みステータスを処理し、クリアする(H\_SIE\_IntStat\_1.Bit3～0)
- 2) 割り込みステータスをディセーブルにする(H\_SIE\_IntEnb\_1.Bit3～0)

<ACT\_HOST / ACT\_ALL ステートに入るとき(2ポートモード時、ACT\_DEVICE ステートに入るとき(1ポートモード時)>

- 3) 割り込みステータスをクリアする(H\_SIE\_IntStat\_1.Bit3～0)
- 4) 割り込みステータスをイネーブルにする(H\_SIE\_IntEnb\_1.Bit3～0)



## 2.8.3. 0E2h H\_FIFO\_IntStat (Host FIFO Interrupt Status)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Host	0E2h	H_FIFO_IntStat		7:	0:	1:	00h
			R (W)	6: FIFO_IDE_Cmp	0: None	1: FIFO-IDE Complete	
			R (W)	5: FIFO1_Cmp	0: None	1: FIFO1 Complete	
			R (W)	4: FIFO0_Cmp	0: None	1: FIFO0 Complete	
				3:	0:	1:	
			R (W)	2: FIFO_NotEmpty	0: None	1: FIFO NotEmpty	
			R (W)	1: FIFO_Full	0: None	1: FIFO Full	
			R (W)	0: FIFO_Empty	0: None	1: FIFO Empty	

ホストの FIFO 関連の割り込みを表示します。

全てのビットは"1"を書き込む事で割り込み要因をクリアできます。

**Bit7**      **Reserved**

**Bit6**      **FIFO\_IDE\_Cmp**

割り込み要因を直接指示します。

IDE にジョインされている FIFO 領域にジョインされているチャンネルが OUT 方向の場合、IDE の転送が終了した後に FIFO が空になると、"1"にセットされます。IDE にジョインされている FIFO 領域にジョインされているチャンネルが IN 方向の場合、IDE の転送が終了すると、"1"にセットされます。

**Bit5**      **FIFO1\_Cmp**

割り込み要因を直接指示します。

DMA1 にジョインされている FIFO 領域にジョインされているチャンネルが OUT 方向の場合、DMA1 の転送が終了した後に FIFO が空になると、"1"にセットされます。DMA1 にジョインされている FIFO 領域にジョインされているチャンネルが IN 方向の場合、DMA1 の転送が終了すると、"1"にセットされます。

**Bit4**      **FIFO0\_Cmp**

割り込み要因を直接指示します。

DMA0 にジョインされている FIFO 領域にジョインされているチャンネルが OUT 方向の場合、DMA0 の転送が終了した後に FIFO が空になると、"1"にセットされます。DMA0 にジョインされている FIFO 領域にジョインされているチャンネルが IN 方向の場合、DMA0 の転送が終了すると、"1"にセットされます。

**Bit3**      **Reserved**

**Bit2**      **FIFO\_NotEmpty**

割り込み要因を直接指示します。

AREAx{x=0-5}Join\_0.JoinFIFO\_Stat ビットが"1"にセットされているときに、該当する FIFO 領域にデータが存在する状態(NotEmpty)になると、"1"にセットされます。

**Bit1**      **FIFO\_Full**

割り込み要因を直接指示します。

AREAx{x=0-5}Join\_0.JoinFIFO\_Stat ビットが"1"にセットされているときに、該当する FIFO 領域が Full になると、"1"にセットされます。

**Bit0**      **FIFO\_Empty**

割り込み要因を直接指示します。

AREAx{x=0-5}Join\_0.JoinFIFO\_Stat ビットが"1"にセットされているときに、該当する FIFO 領域が Empty になると、"1"にセットされます。

## 2. レジスタ

### 2.8.4. 0E3h H\_FrameIntStat (Host Frame Interrupt Status)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Host	0E3h	H_FrameIntStat	R (W)	7: TriggerFrame	0: None	1: Trigger Frame	00h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
			R (W)	2: PortErr	0: None	1: Port Error	
			R (W)	1: FrameNumOver	0: None	1: Frame Number Over	
			R (W)	0: SOF	0: None	1: SOF	

ホストのフレーム関連の割り込みを表示します。  
全てのビットは"1"を書き込む事で割り込み要因をクリアできます。

#### Bit7 TriggerFrame

割り込み要因を直接指示します。

ホストコントローラが SOF トークンを送信する際に、フレームナンバー (H\_FrameNumber\_H,L) とトリガーフレームナンバー (H\_TriggerFrameNum\_H,L) が一致した場合に"1"にセットされます。

#### Bit6-3 Reserved

#### Bit2 PortErr

割り込み要因を直接指示します。

ホスト動作時にポートエラーが検出された場合に"1"にセットされます。

#### Bit1 FrameNumOver

割り込み要因を直接指示します。

フレームナンバーカウンタがオーバーフローした (FrameNumber\_H レジスタの MSb (ビット 2) が "1" から "0" に変化した) 時に "1" にセットされます。FrameNumber\_H、L レジスタではカウンタ桁数が不足する場合に、この割り込みをカウントすることによりそれを補うことができます。

#### Bit0 SOF

割り込み要因を直接指示します。

転送スピードに応じて、下記の場合に "1" にセットされます。

HS: ホストコントローラがマイクロフレーム 0 の SOF トークンを送信する時

FS: ホストコントローラが SOF トークンを送信する時

LS: ホストコントローラが keepalive を送信する時

## 2.8.5. 0E4h H\_CHrIntStat (Host CHr Interrupt Status)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Host	0E4h	H_CHrIntStat		7:	0:	1:
				6:	0:	1:
				5:	0:	1:
			R	4: H_CHeIntStat	0: None	1: CHe Interrupt
			R	3: H_CHdIntStat	0: None	1: CHd Interrupt
			R	2: H_CHcIntStat	0: None	1: CHc Interrupt
			R	1: H_CHbIntStat	0: None	1: CHb Interrupt
			R	0: H_CHaIntStat	0: None	1: CHa Interrupt

チャネル CHx{x=a-e}の割り込みを表示します。

**Bit7-5 Reserved****Bit4 H\_CHeIntStat**

割り込み要因を間接指示します。

H\_CHeIntStat レジスタに割り込み要因があり、かつその割り込み要因に対応する H\_CHeIntEnb レジスタのビットがイネーブルにされている時、"1"にセットされます。

**Bit3 H\_CHdIntStat**

割り込み要因を間接指示します。

H\_CHdIntStat レジスタに割り込み要因があり、かつその割り込み要因に対応する H\_CHdIntEnb レジスタのビットがイネーブルにされている時、"1"にセットされます。

**Bit2 H\_CHcIntStat**

割り込み要因を間接指示します。

H\_CHcIntStat レジスタに割り込み要因があり、かつその割り込み要因に対応する H\_CHcIntEnb レジスタのビットがイネーブルにされている時、"1"にセットされます。

**Bit1 H\_CHbIntStat**

割り込み要因を間接指示します。

H\_CHbIntStat レジスタに割り込み要因があり、かつその割り込み要因に対応する H\_CHbIntEnb レジスタのビットがイネーブルにされている時、"1"にセットされます。

**Bit0 H\_CHaIntStat**

割り込み要因を間接指示します。

H\_CHaIntStat レジスタに割り込み要因があり、かつその割り込み要因に対応する H\_CHaIntEnb レジスタのビットがイネーブルにされている時、"1"にセットされます。

## 2. レジスタ

### 2.8.6. 0E5h H\_CH0IntStat (Host CH0 Interrupt Status)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Host	0E5h	H_CH0IntStat	R (W)	7: TotalSizeCmp	0: None	1: TotalSize Complete	00h
			R (W)	6: TranACK	0: None	1: Tran ACK	
			R (W)	5: TranErr	0: None	1: Tran Error	
			R (W)	4: ChangeCondition	0: None	1: Change Condition	
				3:	0:	1:	
				2:	0:	1:	
			R (W)	1: CTL_SupportCmp	0: None	1: CTL_Support Cmpete	
			R (W)	0: CTL_SupportStop	0: None	1: CTL_Support Stop	

チャンネル CH0 の割り込みステータスを表示します。  
全てのビットは"1"を書き込む事で割り込み要因をクリアできます。

#### Bit7 TotalSizeCmp

割り込み要因を直接指示します。

IRP 単位の packets 転送を正常に転送完了した時に、このビットが"1"にセットされます。

コントロール転送サポート機能動作時はセットアップステージ、データステージ、ステータスステージの各ステージが正常終了した場合にこのビットが"1"にセットされます。

#### Bit6 TranACK

割り込み要因を直接指示します。

H\_CH0Config\_0 レジスタの ACK\_Cnt に設定された回数のトランザクションが正常に完了した時に、このビットが"1"にセットされます。コントロール転送サポート機能動作時はデータステージのトランザクションのみカウントされます。

#### Bit5 TranErr

割り込み要因を直接指示します。

個別トランザクションがリトライエラー、すなわちタイムアウトエラー、CRC エラー、ビットスタッフィングエラー、PID エラー(予期せぬ PID 含む)、トグルミスマッチエラーのいずれかで完了した時に、このビットが"1"にセットされます。

#### Bit4 ChangeCondition

割り込み要因を直接指示します。

トランザクションにおいてコンディションコードのストール、データ・オーバーラン、データ・アンダーラン、3回連続のリトライエラーが発生した時に、このビットが"1"にセットされます。

また、F/W により H\_CH0Config\_0.TranGo ビットを引き上げた際もこのビットが"1"にセットされます。そのときの ConditionCode は最終トランザクションの結果を示します。

#### Bit3-2 Reserved

#### Bit1 CTL\_SupportCmp

割り込み要因を直接指示します。

コントロール転送サポート機能によるコントロール転送の全ステージが正常に完了した時に、このビットが"1"にセットされます。

また、H\_CTL\_SupportControl レジスタの CTL\_SupportGo ビットをクリアすることによるコントロール転送サポート機能の停止処理において、ステータスステージが正常終了して停止処理が終了した場合、このビットが"1"にセットされます。

#### Bit0 CTL\_SupportStop

割り込み要因を直接指示します。

コントロール転送サポート機能によるコントロール転送の途中で異常終了した時に、このビ

ットが"1"にセットされます。

また、H\_CTL\_SupportControl レジスタの CTL\_SupportGo ビットをクリアすることによるコントロール転送サポート機能の停止処理において、ステータスステージ以外のステージで停止処理が終了するか、ステータスステージでトランザクションがエラー終了した場合、このビットが"1"にセットされます。

## 2. レジスタ

### 2.8.7. 0E6h H\_CHaIntStat (Host CHa Interrupt Status)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Host	0E6h	H_CHaIntStat	R (W)	7: TotalSizeCmp	0: None	1: TotalSize Complete	00h
			R (W)	6: TranACK	0: None	1: Tran ACK	
			R (W)	5: TranErr	0: None	1: Tran Error	
			R (W)	4: ChangeCondition	0: None	1: Change Condition	
				3:	0:	1:	
				2:	0:	1:	
			R (W)	1: BO_SupportCmp	0: None	1: BO Support Cmpete	
			R (W)	0: BO_SupportStop	0: None	1: BO Support Stop	

チャンネル CHa の割り込みステータスを表示します。  
全てのビットは"1"を書き込む事で割り込み要因をクリアできます。

#### Bit7 TotalSizeCmp

割り込み要因を直接指示します。

IRP 単位の packets 転送で正常に転送完了した時に、このビットが"1"にセットされます。  
バルクオンリーサポート機能動作時は CBW トランスポート、データトランスポート、CSW トランスポートの各トランスポートが正常終了した場合にこのビットが"1"にセットされます。

#### Bit6 TranACK

割り込み要因を直接指示します。

H\_CHaConfig\_0 レジスタの ACK\_Cnt に設定された回数のトランザクションが正常に完了した時に、このビットが"1"にセットされます。バルクオンリーサポート機能動作時はデータトランスポートのトランザクションのみカウントされます。

#### Bit5 TranErr

割り込み要因を直接指示します。個別トランザクションがリトライエラー、すなわちタイムアウトエラー、CRC エラー、ビットスタッフィングエラー、PID エラー(予期せぬ PID 含む)、トグルミスマッチエラーのいずれかで完了した時に、このビットが"1"にセットされます。

#### Bit4 ChangeCondition

割り込み要因を直接指示します。

トランザクションにおいてコンディションコードのストール、データ・オーバーラン、データ・アンダーラン、3回連続のリトライエラーが発生した時に、このビットが"1"にセットされます。

また、F/W により H\_CHaConfig\_0.TranGo ビットを引き上げた際もこのビットが"1"にセットされます。そのときの ConditionCode は最終トランザクションの結果を示します。

#### Bit3-2 Reserved

#### Bit1 BO\_SupportCmp

割り込み要因を直接指示します。

バルクオンリーサポート機能による転送で、ステータストランスポートが正常に完了した時に、このビットが"1"にセットされます。

また、H\_BO\_SupportControl レジスタの BO\_SupportGo ビットをクリアすることによるバルクオンリーサポート機能の停止処理において、CSW トランスポートが正常終了して停止処理が終了した場合、このビットが"1"にセットされます。

#### Bit0 BO\_SupportStop

割り込み要因を直接指示します。

バルクオンリーサポート機能による転送で、いずれかの転送が異常終了した時に、このビットが"1"にセットされます。

また、H\_BO\_SupportControl レジスタの BO\_SupportGo ビットをクリアすることによるバルクオンリーサポート機能の停止処理において、CSW トランスポート以外のトランスポートで

停止処理が終了するか、CSW トランスポートでエラーを検出した場合、このビットが”1”にセットされます。

## 2. レジスタ

### 2.8.8. 0E7h H\_CHbIntStat (Host CHb Interrupt Status)

### 2.8.9. 0E8h H\_CHcIntStat (Host CHc Interrupt Status)

### 2.8.10. 0E9h H\_CHdIntStat (Host CHd Interrupt Status)

### 2.8.11. 0EAh H\_CHeIntStat (Host CHe Interrupt Status)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Host	0E7h	H_ChbIntStat	R (W)	7: TotalSizeCmp	0: None	1: TotalSize Complete	00h
	0E8h	H_CHcIntStat	R (W)	6: TranACK	0: None	1: Tran ACK	
	0E9h	H_CHdIntStat	R (W)	5: TranErr	0: None	1: Tran Error	
	0EAh	H_CHeIntStat	R (W)	4: ChangeCondition	0: None	1: Change Condition	
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
				0:	0:	1:	

チャネル CHx{x=b-e}の割り込みステータスを表示します。  
全てのビットは"1"を書き込む事で割り込み要因をクリアできます。

#### Bit7 TotalSizeCmp

割り込み要因を直接指示します。

IRP 単位の packets 転送で正常に転送完了した時に、このビットが"1"にセットされます。

#### Bit6 TranACK

割り込み要因を直接指示します。

H\_CHx{x=b-e}Config\_0 レジスタの ACK\_Cnt に設定された回数のトランザクションが正常に完了した時に、このビットが"1"にセットされます。

#### Bit5 TranErr

割り込み要因を直接指示します。個別トランザクションがリトライエラー、すなわちタイムアウトエラー、CRC エラー、ビットスタッフィングエラー、PID エラー(予期せぬ PID 含む)、トグルミスマッチエラーのいずれかで完了した時に、このビットが"1"にセットされます。

アイソクロナス転送の場合はバッファ・オーバーラン、バッファ・アンダーランが発生したときもこのビットが"1"にセットされます。

#### Bit4 ChangeCondition

割り込み要因を直接指示します。

トランザクションにおいてコンディションコードのストール、データ・オーバーラン、データ・アンダーランが発生した時に、このビットが"1"にセットされます。

バルク転送及びインタラプト転送においては、3 回連続のリトライエラーが発生した時にもこのビットが"1"にセットされます。

また、F/W により H\_Chx{x=b-e}Config\_0.TranGo ビットを引き上げた際もこのビットが"1"にセットされます。そのときの ConditionCode は最終トランザクションの結果を示します。

#### Bit3-0 Reserved



## 2.8.12. 0F0h H\_SIE\_IntEnb\_0 (Host SIE Interrupt Enable)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Host	0F0h	H_SIE_IntEnb_0		7:	0:	1:	00h
				6:	0:	1:	
				5:	0:	1:	
			R / W	4: EnDetectCon	0: Disable	1: Enable	
			R / W	3: EnDetectDiscon	0: Disable	1: Enable	
			R / W	2: EnDetectRmtWkup	0: Disable	1: Enable	
			R / W	1: EnDetectDevChirpOK	0: Disable	1: Enable	
			R / W	0: EnDetectDevChirpNG	0: Disable	1: Enable	

H\_SIE\_IntStat\_0 レジスタの割り込み要因による、MainIntStat レジスタの H\_SIE\_IntStat\_0 ビットのアサートを許可／禁止します。

同期ビット(Bit4～0)はACTIVE60 / ACT\_DEVICEステートの時、読み出しはできますが、書き込みできません。ACT\_HOSTステート、または、ACT\_ALLステートから、それ以外のステートに遷移する際の処理に関しては、「2.8.1.0E0h H\_SIE\_IntStat\_0 (Host SIE Interrupt Status 0)」をご参照下さい。

## 2. レジスタ

---

### 2.8.13. 0F1h H\_SIE\_IntEnb\_1 (SIE Host Interrupt Enable 1)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Host	0F1h	H_SIE_IntEnb_1		7:	0:	1:	00h
				6:	0:	1:	
				5:	0:	1	
				4:	0:	1: F	
			R / W	3: EnDisabledCmp	0: Disable	1: Enable	
			R / W	2: EnResumeCmp	0: Disable	1: Enable	
			R / W	1: EnSuspendCmp	0: Disable	1: Enable	
			R / W	0: EnResetCmp	0: Disable	1: Enable	

H\_SIE\_IntStat\_1 レジスタの割り込み要因による、MainIntStat レジスタの H\_SIE\_IntStat\_1 ビットのアサートを許可／禁止します。

同期ビット(Bit3～0)はACTIVE60 / ACT\_DEVICEステートの時、読み出しはできますが、書き込みできません。ACT\_HOSTステート、または、ACT\_ALLステートから、それ以外のステートに遷移する際の処理に関しては、「2.8.2.0E1h H\_SIE\_IntStat\_1 (SIE Host Interrupt Status 1)」をご参照下さい。

## 2.8.14. 0F2h H\_FIFO\_IntEnb (Host FIFO Interrupt Enable)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Host	0F2h	H_FIFO_IntEnb		7:	0:	1:	00h
			R / W	6: EnFIFO_IDE_Cmp	0: Disable	1: Enable	
			R / W	5: EnFIFO1_Cmp	0: Disable	1: Enable	
			R / W	4: EnFIFO0_Cmp	0: Disable	1: Enable	
				3:	0:	1:	
			R / W	2: EnFIFO_NotEmpty	0: Disable	1: Enable	
			R / W	1: EnFIFO_Full	0: Disable	1: Enable	
			R / W	0: EnFIFO_Empty	0: Disable	1: Enable	

H\_FIFO\_IntStat レジスタの割り込み要因による、MainIntStat レジスタの H\_FIFO\_IntStat ビットのアサートを許可／禁止します。

## 2. レジスタ

---

### 2.8.15. 0F3h H\_FrameIntEnb (Host Frame Interrupt Enable)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Host	0F3h	H_FrameIntEnb	R / W	7: EnTriggerFrame	0: Disable      1: Enable	00h
				6:	0:                  1:	
				5:	0:                  1:	
				4:	0:                  1:	
				3:	0:                  1:	
			R / W	2: EnPortErr	0: Disable      1: Enable	
			R / W	1: EnFrameNumOver	0: Disable      1: Enable	
			R / W	0: EnSOF	0: Disable      1: Enable	

H\_FrameIntStat レジスタの割り込み要因による、MainIntStat レジスタの H\_FrameIntStat ビットのアサートを許可／禁止します。

## 2.8.16. 0F4h H\_CHrIntEnb (Host CHr Interrupt Enable)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Host	0F4h	H_CHrIntEnb		7:	0:	1:	00h
				6:	0:	1:	
				5:	0:	1:	
			R / W	4: EnH_CHeIntStat	0: Disable	1: Enable	
			R / W	3: EnH_CHdIntStat	0: Disable	1: Enable	
			R / W	2: EnH_CHcIntStat	0: Disable	1: Enable	
			R / W	1: EnH_CHbIntStat	0: Disable	1: Enable	
			R / W	0: EnH_CHaIntStat	0: Disable	1: Enable	

H\_CHrIntStat レジスタの割り込み要因による、MainIntStat レジスタの H\_CHrIntStat ビットのアサート  
を許可／禁止します。

## 2. レジスタ

---

### 2.8.17. 0F5h H\_CH0IntEnb (Host CH0 Interrupt Enable)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Host	0F5h	H_CH0IntEnb	R / W	7: EnTotalSizeCmp	0: Disable	1: Enable	00h
			R / W	6: EnTranACK	0: Disable	1: Enable	
			R / W	5: EnTranErr	0: Disable	1: Enable	
			R / W	4: EnChangeCondition	0: Disable	1: Enable	
				3:	0:	1:	
				2:	0:	1:	
			R / W	1: EnCTL_SupportCmp	0: Disable	1: Enable	
			R / W	0: EnCTL_SupportStop	0: Disable	1: Enable	

H\_CH0IntStat レジスタの割り込み要因による、MainIntStat レジスタの H\_CH0IntStat ビットのアサートを許可／禁止します。

## 2.8.18. 0F6h H\_CHaIntEnb (Host CHa Interrupt Enable)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Host	0F6h	H_CHaIntEnb	R / W	7: EnTotalSizeCmp	0: Disable	1: Enable	00h
			R / W	6: EnTranACK	0: Disable	1: Enable	
			R / W	5: EnTranErr	0: Disable	1: Enable	
			R / W	4: EnChangeCondition	0: Disable	1: Enable	
				3:	0:	1:	
				2:	0: Disable	1:	
			R / W	1: EnBO_Support_Cmp	0: Disable	1: Enable	
			R / W	0: EnBO_Support_Stop	0: Disable	1: Enable	

H\_CHaIntStat レジスタの割り込み要因による、H\_CHrIntStat レジスタの CHaIntStat ビットのアサート  
を許可／禁止します。

## 2. レジスタ

2.8.19. 0F7h H\_CHbIntEnb (Host CHb Interrupt Enable)

2.8.20. 0F8h H\_CHcIntEnb (Host CHc Interrupt Enable)

2.8.21. 0F9h H\_CHdIntEnb (Host CHd Interrupt Enable)

2.8.22. 0FAh H\_CHeIntEnb (Host CHe Interrupt Enable)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Host	0F7h	H_CHbIntEnb	R / W	7: EnTotalSizeCmp	0: Disable 1: Enable	00h
	0F8h	H_CHcIntEnb	R / W	6: EnTranACK	0: Disable 1: Enable	
	0F9h	H_CHdIntEnb	R / W	5: EnTranErr	0: Disable 1: Enable	
	0FAh	H_CHeIntEnb	R / W	4: EnChangeCondition	0: Disable 1: Enable	
				3:	0: 1:	
				2:	0: 1:	
				1:	0: 1:	
				0:	0: 1:	

H\_CHx{x=b-e}IntStat レジスタの割り込み要因による、H\_CHrIntStat レジスタの CHx{x=b-e}IntStat ビットのアサートを許可／禁止します。



2.8.23. 100h *H\_Reset* (Host Reset)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Host	100h	<i>H_Reset</i>		7:	0:	1:	01h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
			R / W	0: <i>ResetHTM</i>	0: Do nothing	1: Reset HTM	

ホストのトランシーバマクロをリセットします。  
SLEEP / SNOOZE 時でもアクセス有効です。

**Bit7-1**    **Reserved**  
**Bit0**     **ResetHTM**

このビットに"1"をセットすると、本 LSI のホストトランシーバマクロを初期化します。  
リセットを解除するには、このビットを"0"にクリアして下さい。

## 2. レジスタ

### 2.8.24. 102h H\_NegoControl\_0 (Host NegoControl 0)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Host	102h	H_NegoControl_0	R / W	7: AutoModeCancel	0: None	1: Cancel	1Xh
			R	6: HostState[2]	HostState[2:0]		
			R	5: HostState[1]			
			R	4: HostState[0]			
			W	3: AutoMode[3]	AutoMode[3:0]		
			W	2: AutoMode[2]			
			W	1: AutoMode[1]			
			W	0: AutoMode[0]			

ホストのネゴシエーション に関する動作設定を行います。

#### Bit7 AutoModeCancel

このビットを"1"にセットすると、ホストステート管理サポート機能の実行を停止して、そのステートに留まります(H\_NegoControl\_0.AutoMode および H\_XcvtControl の設定を保持、信号ラインの状態を保持、内蔵タイマを停止、接続・切断・デバイス Chirp・リモートウェークアップの各検出機能を OFF します)。

以下の操作を行うとき、事前にこのビットによりホストステート管理サポート機能の実行を停止する必要があります。

- ・ホストステートを IDLE ステートに遷移させるとき、
  - ・デバイスからの Chirp に異常を検出しリセット完了ステータス(H\_SIE\_IntStat\_1.ResetCmp)の発行を待たずにホストステートを DISABLED ステートに遷移させるとき、
  - ・H\_USB\_Test.EnHS\_Testによりテストモードを実行するとき、
- このビットに"1"をセットすると、ホストステート管理サポート機能の停止処理を行い、停止処理が完了することで本ビットは"0"になります(60MHz クロックで 6 サイクル程度必要)。上記の場合においては、本ビットが"0"になったことを確認してから、H\_NegoControl\_0.AutoMode の GoIDLE 設定または GoDISABLED 設定、あるいは H\_USB\_Test.EnHS\_Test セットを行ってください。

#### Bit6-4 HostState[2:0]

ホストステート管理サポート機能実行時の現在のホストステートを示します。ステートは以下のいずれかの項目になります。

000: Reserved  
001: IDLE  
010: WAIT\_CONNECT  
011: DISABLED  
100: USB\_RESET  
101: USB\_OPERATIONAL  
110: USB\_SUSPEND  
111: USB\_RESUME

#### Bit3-0 AutoMode[3:0]

ホストステート管理サポート機能の実行で、遷移させるホストステートを設定します。本レジスタはライトオンリーレジスタであり、以下のいずれかを設定します。

0001:GoIDLE (IDLE ステートに遷移させる)  
0010:GoWAIT\_CONNECT (WAIT\_CONNECT ステートに遷移させる)  
0011:GoDISABLED (DISABLED ステートに遷移させる)  
0100:GoRESET (RESET ステートに遷移させる)  
0101:GoOPERATIONAL (OPERATIONAL ステートに遷移させる)  
0110:GoSUSPEND (SUSPEND ステートに遷移させる)  
0111:GoRESUME (RESUME ステートに遷移させる)

1001: GoWAIT\_CONNECTtoDIS (WAIT\_CONNECT ステートから DISABLED ステートまで連続に遷移させる)

1010: GoWAIT\_CONNECTtoOP (WAIT\_CONNECT ステートから OPERATIONAL ステートまで連続に遷移させる)

1100:GoRESETtoOP (RESET ステートから OPERATIONAL ステートまで連続に遷移させる)

1110:GoSUSPENDtoOP (SUSPEND ステートから OPERATIONAL ステートまで連続に遷移させる)

1111:GoRESUMEtoOP (RESUME ステートから OPERATIONAL ステートまで連続に遷移させる)

上記以外: Reserved

あるステートから IDLE ステートに遷移させる(GoIDLE を実行する)場合には、以下の手順を実行してください。

- H\_NegoControl\_0 レジスタに 0x80 をライトします。
- H\_NegoControl\_0.AutoModeCancel ビットが 0 になることを確認します。
- H\_NegoControl\_0 レジスタに 0x01 をライトします。

## 2. レジスタ

### 2.8.25. 104h H\_NegoControl\_1 (Host NegoControl 1)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Host	104h	H_NegoControl_1		7:	0:	1:
				6:	0:	1:
			R / W	5: PortSpeed[1]	PortSpeed[1:0]	
			R / W	4: PortSpeed[0]		
				3:	0:	1:
				2:	0:	1:
			R / W	1: DisChirpFinish	0: Normal	1: DisableChirpFinish
			R / W	0: RmtWkupDetEnb	0: Disable	1: Enable

ホストのネゴシエーション に関する動作設定を行います。

(注)本レジスタの Reset 値は 2 ポートモードの ACT\_HOST 時と ACT\_ALL 時、及び 1 ポートモードの ACT\_DEVICE 時に読める値です。それ以外のステートでは、Reset 値は 00h が読めます。

**Bit7-6**    **Reserved**

**Bit5-4**    **PortSpeed[1:0]**

転送速度を表示、及び設定します。通常は、リードオンリーですが、H\_Protect.PortSpeedWrEnb ビットに"1"をセットとすると、書き込み可能になります。  
Test\_Force\_Enable テストモードを実行する場合以外には、書き込む必要はありません。

00: High Speed

01: Full Speed

10: Reserved

11: Low Speed

**Bit3-2**    **Reserved**

**Bit1**      **DisChirpFinish**

デバイス Chirp が規定時間に完了しないときの動作モードを設定します。

0: デバイス Chirp 異常のステータスを上げた後、規定時間の USB Reset 送出により USB Reset を完了します。

1: デバイス Chirp 異常のステータスを上げた後、デバイス Chirp の完了を待ち、デバイス Chirp が完了するとホスト Chirp を  
実行した後、USB Reset を完了します。

**Bit0**      **RmtWkupDetEnb**

リモートウェークアップ検出機能の有効／無効を設定します。

## 2.8.26. 106h H\_USB\_Test (Host USB\_Test)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Host	106h	H_USB_Test	R / W	7: EnHS_Test	0: Do nothing	1: EnHS_Test	00h
				6:	0:	1:	
				5:	0:	1:	
			R / W	4: Test_Force_Enable	0: Do nothing	1: Test_Force_Enable	
			R / W	3: Test_SE0_NAK	0: Do nothing	1: Test_SE0_NAK	
			R / W	2: Test_J	0: Do nothing	1: Test_J	
			R / W	1: Test_K	0: Do nothing	1: Test_K	
			R / W	0: Test_Packet	0: Do nothing	1: Test_Packett	

ホストの USB 2.0 のテストモードに関する動作設定を行います。

テストモードの実行は WAIT\_CONNECT、DISABLED、SUSPEND のいずれかのステートで有効です。

これらのステートからテストモードに入る前には、各ステートの処理を停止する必要があります。テストモードに入る際には以下の手順を実行してください。

- ・ 全チャネルの TranGo ビット (H\_CHx{x=0,a-e}Config\_0.TranGo) 、 H\_CTL\_SupportControl.CTL\_SupportGo、 H\_BO\_SupportControl.BOSupportGo を”0”に設定します。
- ・ H\_NegoControl\_0 レジスタに 0x80 をライトします。
- ・ H\_NegoControl\_0.AutoModeCancel ビットが”0”になるのを確認します。
- ・ 本レジスタの下位 5 ビットのいずれかのビットと、EnHS\_Test に同時に”1”をセットします。

また、あるテストモードから他のテストモードに切り替えるとき、あるいはテストモードを終了させるときは、本レジスタに 0x00 をライトしてください。テストモードを終了し、ホストステートは IDLE となります。

**Bit7 EnHS\_Test**

このビットと H\_USB\_Test レジスタの下位 5 ビットのいずれかのビットを同時に”1”を設定すると、そのビットに対応するテストモードに入ります。

**Bit6-5 Reserved****Bit4 Test\_Force\_Enable**

このビットは Test\_Force\_Enable テストモードを実行する時に使用します。Test\_Force\_Enable テストモードを実行する為には前述のテストモードに入る際の手順を実行した後に以下の処理を行ってください。

初めに H\_Protect.PortSpeedWrEnb に ”1” を セ ッ ト し ま す 。 そ し て H\_NegoControl\_1.PortSpeed=”HS(0b00)”と設定した後に H\_Protect.PortSpeedWrEnb を”0”にクリアします。次に H\_NegoControl\_0.AutoMode に GoOPERATIONAL を設定します。

上記処理を行った後に、Test\_Force\_Enable ビットと EnHS\_Test ビットに同時に”1”をセットすることにより、TestForceEnable テストモードに入ることができます。このテストモードでは、ホストポートは HS モードにて SOF を送出し、切断検出が可能となります。

**Bit3 Test\_SE0\_NAK**

このビットと EnHS\_Test ビットに同時に”1”をセットすることにより、Test\_SE0\_NAK テストモードに入ることができます。このテストモードでは、ホストポートは HS モードにて受信状態となります。

**Bit2 TEST\_J**

このビットと EnHS\_Test ビットに同時に”1”をセットすることにより、Test\_J テストモードに入ることができます。このテストモードでは、ホストポートは HS モードにて”J”を送出します。

**Bit1 TEST\_K**

このビットと EnHS\_Test ビットに同時に”1”をセットすることにより、Test\_K テストモード

## 2. レジスタ

---

に入ることができます。このテストモードでは、ホストポートは HS モードにて”K”を送出します。

### Bit0 Test\_Packet

このビットと EnHS\_Test ビットに同時に”1”をセットすることにより、Test\_Packet テストモードに入ることができます。このテストモードは CH0 のみで使用できます。テストモードに入る前に CH0 にジョインされた FIFO 領域を 64 バイトに設定し、FIFO 領域をクリアし、下記のテストパケット用のデータをこの FIFO 領域に書き込んで下さい。

パケット送信テストモード時に FIFO に書き込むデータは以下の 53 バイトです。

00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h,  
00h, AAh, AAh, AAh, AAh, AAh, AAh, AAh,  
AAh, EEh, EEh, EEh, EEh, EEh, EEh, EEh,  
EEh, FEh, FFh, FFh, FFh, FFh, FFh, FFh,  
FFh, FFh, FFh, FFh, FFh, 7Fh, BFh, DFh,  
EFh, F7h, FBh, FDh, FCh, 7Eh, BFh, DFh,  
EFh, F7h, FBh, FDh, 7Eh

テストパケット送出時に、SIE が PID と CRC を付加しますので、FIFO に書き込むデータは、USB 規格 Rev.2.0 に記載されているテストパケットデータのうち、DATA0 PID の次のデータから、CRC16 以外のデータまでとなります。

- 2.8.27. 110h H\_CH0SETUP\_0 (Host CH0 SETUP 0)  
 2.8.28. 111h H\_CH0SETUP\_1 (Host CH0 SETUP 1)  
 2.8.29. 112h H\_CH0SETUP\_2 (Host CH0 SETUP 2)  
 2.8.30. 113h H\_CH0SETUP\_3 (Host CH0 SETUP 3)  
 2.8.31. 114h H\_CH0SETUP\_4 (Host CH0 SETUP 4)  
 2.8.32. 115h H\_CH0SETUP\_5 (Host CH0 SETUP 5)  
 2.8.33. 116h H\_CH0SETUP\_6 (Host CH0 SETUP 6)  
 2.8.34. 117h H\_CH0SETUP\_7 (Host CH0 SETUP 7)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Host	110h -117h	H_CH0SETUP_0 -H_CH0SETUP_7	R/W	7: CH0SETUP_n[7]	Channel 0 SETUP Data 0 - Channel 0 SETUP Data 7	00h
				6: CH0SETUP_n[6]		
				5: CH0SETUP_n[5]		
				4: CH0SETUP_n[4]		
				3: CH0SETUP_n[3]		
				2: CH0SETUP_n[2]		
				1: CH0SETUP_n[1]		
				0: CH0SETUP_n[0]		

チャンネルCH0のセットアップステージで送信する8バイトのデータを順にセットするレジスタです。

#### CH0SETUP\_0

bmRequestType をセットします。

#### CH0SETUP\_1

bRequest をセットします。

#### CH0SETUP\_2

wValue の下位 8 ビットをセットします。

#### CH0SETUP\_3

wValue の上位 8 ビットをセットします。

#### CH0SETUP\_4

wIndex の下位 8 ビットをセットします。

#### CH0SETUP\_5

wIndex の上位 8 ビットをセットします。

#### CH0SETUP\_6

wLength の下位 8 ビットをセットします。

#### CH0SETUP\_7

wLength の上位 8 ビットをセットします。

## 2. レジスタ

### 2.8.35. 11Eh H\_FrameNumber\_H (Host FrameNumber High)

### 2.8.36. 11Fh H\_FrameNumber\_L (Host FrameNumber Low)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Host	11Eh	H_FrameNumber_H		7:	0:	1:	07h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
			R/W	2: FrameNumber[10]	Frame Number High		
				1: FrameNumber[9]			
				0: FrameNumber[8]			

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Host	11Fh	H_FrameNumber_L	R/W	7: FrameNumber[7]	Frame Number Low	FFh
				6: FrameNumber[6]		
				5: FrameNumber[5]		
				4: FrameNumber[4]		
				3: FrameNumber[3]		
				2: FrameNumber[2]		
				1: FrameNumber[1]		
				0: FrameNumber[0]		

SOF トークンを送信する毎に更新される、USB のフレームナンバーが表示されます。フレームナンバーを取得する場合は、H\_FrameNumber\_H と H\_FrameNumber\_L レジスタを対でアクセスする必要があります。

(注)本レジスタの Reset 値は2ポートモードの ACT\_HOST 時と ACT\_ALL 時、及び1ポートモードの ACT\_DEVICE 時に読める値です。それ以外のステートでは、Reset 値は 00h が読めます。

#### 11Eh.Bit7-3 Reserved

#### 11Eh.Bit2-0, 11Fh.Bit7-0 FrameNumber[10:0]

送信する SOF パケットの FrameNumber が表示されます。



## 2.8.37. 120h H\_CH0Config\_0 (Host Channel 0 Configuration0)

Mode	Addresses	Register Name	R / W	Bit Symbol	Description		Reset
Host	120h	H_CH0Config_0	R / W	7: ACK_Cnt[3]	ACK Count		00h
				6: ACK_Cnt[2]			
				5: ACK_Cnt[1]			
				4: ACK_Cnt[0]			
			R / W	3: SpeedMode[1]	Speed Mode		
				2: SpeedMode[0]			
			R / W	1: Toggle	0: Toggle0	1: Toggle1	
			R / W	0: TranGo	0: Stand by	1: Transaction Start	

ホスト動作時にチャンネル CH0 の基本設定を行います。

**Bit7-4 ACK\_Cnt[3:0]**

チャンネル CH0 で行われる転送において、ACK をカウントする数を設定します。

設定された値だけ ACK をカウントすると、H\_CH0IntStat レジスタの TranACK ビットがセットされます。

0000: 16 回の ACK をカウントします。

0001～1111: 1 回～15 回の ACK をカウントします。

コントロール転送サポート機能実行中においてはデータステージのトランザクションのみカウントされ、セットアップステージ、ステータスステージのトランザクションはカウントされません。

**Bit3-2 SpeedMode[1:0]**

チャンネル CH0 で転送を行うデバイスの動作モードを設定します。

00: HS モード — HS デバイスの時はこの設定にして下さい。

01: FS モード — FS デバイスの時はこの設定にして下さい。

10: Reserved — 本値の使用を禁止します。

11: LS モード — LS デバイスの時はこの設定にして下さい。

**Bit1 Toggle**

トランザクションを開始する時のトグルシーケンスビットの初期値を設定します。また、トランザクション実行後、およびトランザクション完了後はトグルシーケンスビットの状態を示します。

0: トグル 0

1: トグル 1

**Bit0 TranGo**

このビットを”1”に設定するとチャンネル CH0 のトランザクションを開始します。トランザクション開始後このビットを”0”にクリアするとトランザクション処理を停止させることができます。また、このビットはチャンネル CH0 がトランザクション実行中か否かのステータスの意味も有します。

0: トランザクションを停止します(トランザクション停止中です)

1: トランザクションを開始します(トランザクション実行中です)

転送が H\_CH0TotalSize\_H～L レジスタ で設定したバイト数を完了した時点で H\_CH0IntStat レジスタの TotalSizeCmp ビットが”1”にセットされ、本ビットは自動的に”0”に戻ります。H\_CH0IntStat レジスタの ChangeCondition ビットがセットされたような場合は”0”にリセットされます。この時は、H\_CH0ConditionCode レジスタにその原因がセットされていますので参照して下さい。

また、このビットのクリアによる停止を行った時は処理最中のトランザクションが終了した時点で、H\_CH0IntStat レジスタの ChangeCondition ビットがセットされます。トランザクシ

## 2. レジスタ

---

ンが停止されても、FIFO 中のデータ、(残りの)トータルサイズ、チャネルに関する設定はそのままの状態となります。従って再びこのビットを"1"にセットすることによって、トランザクションを停止された時の続きから再開させることが可能です。(新たなトランザクションを行う場合は FIFO をクリアし、チャネル情報の設定をし直してください。)

コントロール転送サポート機能を使用する際はこのビットを設定する必要はありません。

## 2.8.38. 121h H\_CH0Config\_1 (Host Channel 0 Configuration1)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Host	121h	H_CH0Config_1	R / W	7: TID[1]	Transaction ID	00h
				6: TD[0]		
				5:	0:	
				4:	0:	
				3:	0:	
				2:	0:	
				1:	0:	
				0:	0:	

ホスト動作時にチャンネル CH0 の基本設定を行います。

**Bit7-6 TID[1:0]**

チャンネル CH0 で発行するトランザクションの種類(SETUP 、OUT、IN)を設定します。このビットの設定は CTL\_SupportControl レジスタの CTL\_SupportGo ビットを"1"に設定してトランザクションを開始した時には無効になります。

00: SETUP — SETUP トークンを発行します。

01: OUT — OUT トークンを発行します。

10: IN — IN トークンを発行します。

11: Reserved — 本値の使用を禁止します。

コントロール転送サポート機能を使用する際はこのビットを設定する必要はありません。

**Bit5-0 Reserved**

## 2. レジスタ

### 2.8.39. 123h H\_CH0MaxPktSize (Host Channel 0 Max Packet Size)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Host	123h	H_CH0MaxPktSize		7:	0:	1:	00h
			R / W	6: MaxPktSize[6]	Max Packet Size		
				5: MaxPktSize[5]			
				4: MaxPktSize[4]			
				3: MaxPktSize[3]			
				2: MaxPktSize[2]			
				1: MaxPktSize[1]			
				0: MaxPktSize[0]			

ホスト動作時にチャンネル CH0 の MaxPacketSize の設定を行います。

**Bit7**      **Reserved**

**Bit6-0**    **MaxPktSize[6:0]**

チャンネル CH0 の MaxPacketSize を設定します。

LS 時      8 バイト

FS 時      8, 16, 32, 64 バイト

HS 時      64 バイト

のいずれかに設定してください。

それ以外の値の設定は禁止します。

**2.8.40. 126h H\_CH0TotalSize\_H (Host Channel 0 Total Size High)****2.8.41. 127h H\_CH0TotalSize\_L (Host Channel 0 Total Size Low)**

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Host	126h	H_CH0TotalSize_H	R / W	7: TotalSize[15]	Total Size High	00h
				6: TotalSize[14]		
				5: TotalSize[13]		
				4: TotalSize[12]		
				3: TotalSize[11]		
				2: TotalSize[10]		
				1: TotalSize[9]		
				0: TotalSize[8]		

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Host	127h	H_CH0TotalSize_L	R / W	7: TotalSize[7]	Total Size Low	00h
				6: TotalSize[6]		
				5: TotalSize[5]		
				4: TotalSize[4]		
				3: TotalSize[3]		
				2: TotalSize[2]		
				1: TotalSize[1]		
				0: TotalSize[0]		

ホスト動作時にチャンネル CH0 で転送を行うデータの Total Size を設定します。

**118h.Bit7-0, 119h.Bit7-0 TotalSize[15:0]**

チャンネルCH0における転送データの全バイト数(最大 65,535byte:約 64Kbyte)を設定します。

H\_CH0Config\_0 レジスタの TranGo ビットによりトランザクションが開始された後は、トランザクションが正常終了するたびにこのレジスタは更新されます。

また、TotalSize = 0 の時、H\_CH0Config\_0 レジスタの TranGo ビットにより OUT トランザクションを開始するとゼロ長パケットが送信されます。

SETUP トランザクションを行う場合、およびコントロール転送サポート機能を使用する場合は、このレジスタを設定する必要はありません。

## 2. レジスタ

### 2.8.42. 128h H\_CH0HubAdrs (Host Channel 0 Hub Address)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Host	128h	H_CH0HubAdrs	R / W	7: HubAdrs[3]	Hub Address		00h
				6: HubAdrs[2]			
				5: HubAdrs[1]			
				4: HubAdrs[0]			
				3:	0:	1:	
			R / W	2: Port[2]	Port Number		
				1: Port[1]			
				0: Port[0]			

ホスト動作時にチャンネル CH0 に接続するハブの設定を行います。

#### Bit7-4 HubAdrs[3:0]

チャンネル CH0 で転送を行うファンクションが接続されているハブの USB アドレスを設定します。

0～15 までの任意の値が設定できます。

#### Bit3 Reserved

#### Bit2-0 Port[2:0]

チャンネル CH0 で転送を行うファンクションが接続されているハブのポートナンバーを設定します。

0～7 までの任意の値が設定できます。

## 2.8.43. 129h H\_CH0FuncAdrs (Host Channel 0 Function Address)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Host	129h	H_CH0FuncAdrs	R / W	7: FuncAdrs[3]	Function Address	00h
				6: FuncAdrs[2]		
				5: FuncAdrs[1]		
				4: FuncAdrs[0]		
			R / W	3: EP_Number[3]	Endpoint Number	
				2: EP_Number[2]		
				1: EP_Number[1]		
				0: EP_Number[0]		

ホスト動作時にチャンネル CH0 で転送を行うファンクションのアドレス設定を行います。

**Bit7-4 FuncAdrs[3:0]**

チャンネル CH0 が管理するエンドポイントを含むファンクションの USB アドレスを設定します。

0～15 までの任意の値が設定できます。

**Bit3-0 EP\_Number[3:0]**

チャンネル CH0 で転送を行うエンドポイント番号を設定します。

0～15 までの任意の値が設定できます。

## 2. レジスタ

### 2.8.44. 12Bh H\_CTL\_SupportControl (Host ControlTransfer Support Control)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Host	12Bh	H_CTL_Support - Control		7:	0:	1:	00h
				6:	0:	1:	
			R / W	5: CTL_SupportState[1]	Control Transfer Support State		
				4: CTL_SupportState[0]			
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
			R / W	0: CTL_SupportGo	0: Stand by	1: Control Transfer Go	

ホスト動作時にチャンネル CH0 でコントロール転送を行う時の支援機能の設定を行います。

#### Bit7-6 Reserved

#### Bit5-4 CTL\_SupportState[1:0]

CTL\_SupportGo ビットを”1”に設定し、コントロール支援機能を使用して転送を行っている時にどのステージを実行しているかを示します。

00: Idle — 転送未実行、または転送が正常に終了したことを示します。

01: Setup Stage — セットアップステージを実行していることを示します。

10: Data Stage — データステージを実行していることを示します。

11: Status Stage — ステータスステージを実行していることを示します。

#### Bit3-1 Reserved

#### Bit0 CTL\_SupportGo

このビットを”1”に設定するとチャンネル CH0 でコントロール転送支援機能により、セットアップステージ～(データステージ)～ステータスステージが自動的に行われます。

セットアップステージでは、SETUP トークンを自動的に送出し H\_CH0SETUP\_0～7 に設定したリクエストが送信されます。

次にデータステージがある場合は、指定された方向およびサイズでトランザクションが自動的に実行されます。

最後にステータスステージでは、データステージの有無および方向により、自動的に適当な PID のトークンを発行しゼロ長パケットの送受信が行われます。

以上のトランザクションおよびステージシーケンスが正常完了すると、H\_CH0IntStat レジスタの CTL\_SupportCmp ビットがセットされます。シーケンスの途中でパケットにエラーを検出した場合は、H\_CH0IntStat レジスタの CTL\_SupportStop ビットがセットされ、トランザクションは停止します。この時は、ConditionCode レジスタにその原因がセットされているので参照して下さい。

コントロール転送が終了した場合(正常終了、エラー終了いずれの場合も)、このビットは自動的にクリアされます。

コントロール転送サポート機能を実行中にこのビットをクリアすることにより、コントロール転送を停止することが出来ます。コントロール転送がステータスステージで正常終了した場合は CTL\_SupportCmp ビットがセットされます。これ以外の場合は CTL\_SupportStop ビットがセットされます。コントロール転送が停止したステージは CTL\_SupportState を参照して下さい。



## 2.8.45. 12Eh H\_CH0ConditionCode (Host Channel 0 Condition Code)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Host	11Eh	H_CH0ConditionCode		7:	0:	1:
			R	6: ConditionCode[2]	Condition Code	
				5: ConditionCode[1]		
				4: ConditionCode[0]		
				3:	0:	1:
				2:	0:	1:
				1:	0:	1:
				0:	0:	1:

ホスト動作時にチャンネル CH0 の転送完了結果を示します。

**Bit7**      **Reserved**

**Bit6-4**    **ConditionCode[2:0]**

チャンネル CH0 で転送が完了した場合の結果を示します。

Code	Meaning	Description
000	NOERROR ノーエラー	トランザクションがエラーなく完了した
001	STALL ストール	エンドポイントがストールPIDを返した
010	DATAOVERRUN データ・オーバーラン	<ul style="list-style-type: none"> <li>最大パケットサイズを超えるデータパケットを受信した * CRCエラーやビット・スタッフィングエラーが同時に検出された場合はリトライエラーとして処理します</li> <li>IRP(TotalSize)を超えるデータ数を受信した * CRCエラーやビット・スタッフィングエラーが同時に検出された場合はリトライエラーとして処理します</li> <li>* データパケットが最大パケットサイズ以下で、データパケットに含まれたデータグルが予期した値と一致していない場合、データ・オーバーランではなくグルミスマッチとして処理します</li> </ul>
011	DATAUNDERRUN データ・アンダーラン	<ul style="list-style-type: none"> <li>最大パケットサイズ未満のデータパケットを受信し、且つデータ数がIRP(TotalSize)に満たない * CRCエラーやビット・スタッフィングエラーが同時に検出された場合はリトライエラーとして処理します</li> </ul>
100	RETRYERROR リトライエラー	<ul style="list-style-type: none"> <li>デバイスがトークンに規定時間以内に応答しない(IN)、または規定時間以内にハンドシェークを発信しない(OUT)</li> <li>エンドポイントからのデータパケットにCRCエラーが含まれている</li> <li>エンドポイントからのデータパケットにビット・スタッフィングエラーが含まれている</li> <li>エンドポイントからのPID検査ビットが、データPID(IN)またはハンドシェーク(OUT)に失敗した</li> <li>受信PIDが無効またはPID値が定義されていない</li> <li>エンドポイントからのデータパケットに含まれたデータグルが、予期した値と一致していない(グルミスマッチ)</li> </ul>
その他	Reserved	

**Bit3-0**    **Reserved**

## 2. レジスタ

### 2.8.46. 130h H\_CHaConfig\_0 (Host Channel a Configuration0)

Mode	Address	Register Name	R / W	Bit Symbol	Description			Reset
Host	130h	H_CHaConfig_0	R / W	7: ACK_Cnt[3]	ACK Count			00h
				6: ACK_Cnt[2]				
				5: ACK_Cnt[1]				
				4: ACK_Cnt[0]				
			R / W	3: SpeedMode[1]	Speed Mode			
				2: SpeedMode[0]				
			R / W	1: Toggle	0: Toggle0	1: Toggle1		
			R / W	0: TranGo	0: Stand by	1: Transaction Start		

ホスト動作時にチャネル CHa の基本設定を行います。

#### Bit7-4 ACK\_Cnt[3:0]

チャネル CHa で行われる転送において、ACK をカウントする数を設定します。

設定された値だけ ACK をカウントすると、H\_CHaIntStat レジスタの TranACK ビットがセットされます。

0000: 16 回の ACK をカウントします。

0001～1111: 1 回～15 回の ACK をカウントします。

バルクオンリーサポート機能実行中においてはデータトランスポートのトランザクションのみカウントされ、CBW トランスポート、CSW トランスポートのトランザクションはカウントされません。

#### Bit3-2 SpeedMode[1:0]

チャネル CHa で転送を行うデバイスの動作モードを設定します。

00: HS モード — HS デバイスの時はこの設定にして下さい。

01: FS モード — FS デバイスの時はこの設定にして下さい。

10-11: Reserved — 本値の使用を禁止します。

#### Bit1 Toggle

トランザクションを開始する時のトグルシーケンスビットの初期値を設定します。また、トランザクション実行後、およびトランザクション完了後はトグルシーケンスビットの状態を示します。

0: トグル 0

1: トグル 1

バルクオンリーサポート機能を使用する際はこのビットを設定する必要はありません。

#### Bit0 TranGo

このビットを”1”に設定するとチャネル CHa のトランザクションを開始します。トランザクション開始後このビットを”0”にクリアするとトランザクション処理を停止させることができます。また、このビットはチャネル CHa がトランザクション実行中か否かのステータスの意味も有します。

0: トランザクションを停止します(トランザクション停止中です)

1: トランザクションを開始します(トランザクション実行中です)

転送が H\_CHaTotalSize\_HH～LL レジスタ で設定したバイト数完了した時点で H\_CHaIntStat レジスタの TotalSizeCmp ビットが ”1”にセットされ、本ビットは自動的に”0”に戻ります。H\_CHaIntStat レジスタの ChangeCondition ビットがセットされたような場合は”0”にリセットされます。この時は、H\_CHaConditionCode レジスタにその原因がセットされていますので参照して下さい。

また、このビットのクリアによる停止を行った時は処理最中のトランザクションが終了した時点で、H\_CHaIntStat レジスタの ChangeCondition ビットがセットされます。トランザクションが停止されても、FIFO 中のデータ、(残りの)トータルサイズ、チャネルに関する設定はそ

のままの状態となります。従って再びこのビットを”1”にセットすることによって、トランザクションを停止された時の続きから再開させることが可能です。(新たなトランザクションを行う場合は **FIFO** をクリアし、チャネル情報の設定をし直してください。)

バルクオンリーサポート機能を使用する際はこのビットを設定する必要はありません。

## 2. レジスタ

### 2.8.47. 131h H\_CHaConfig\_1 (Host Channel a Configuration1)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Host	131h	H_CHaConfig_1	R / W	7: TID[1]	Transaction ID	00h
				6: TID[0]		
				5:	0:	1:
				4:	0:	1:
			R / W	3: AutoZerolen	0: Do nothing	1: Add Zerolen
				2:	0:	1:
				1:	0:	1:
			R / W	0: TotalSizeFree	0: Do nothing	1: Total Size Free

ホスト動作時にチャネル CHa の基本設定を行います。

#### Bit7-6 TID[1:0]

チャネル CHa で発行するトランザクションの種類(OUT、IN)を設定します。このビットの設定は H\_BO\_SupportControl レジスタの BO\_SupportGo ビットを"1"に設定してトランザクションを開始した時には無効になります。

00: Reserved — 本値の使用を禁止します。

01: OUT — OUT トークンを発行します。

10: IN — IN トークンを発行します。

11: Reserved — 本値の使用を禁止します。

バルクオンリーサポート機能を使用する際はこのビットを設定する必要はありません。

#### Bit5-4 Reserved

#### Bit3 AutoZerolen

このビットに"1"を設定すると H\_CHaTotalSizeHH～LL レジスタで設定したサイズの転送がちょうど Max Packet Size で完了した際に、ゼロ長パケットを最後に自動付与します。OUT 転送の場合のみこのビットは有効となります。

#### Bit2-1 Reserved

#### Bit0 TotalSizeFree

このビットに"1"を設定すると H\_CHaTotalSizeHH～LL レジスタの設定値に関わりなく転送サイズが無限になります。

**2.8.48. 132h H\_CHaMaxPktSize\_H (Host Channel a Max Packet Size High)****2.8.49. 133h H\_CHaMaxPktSize\_L (Host Channel a Max Packet Size Low)**

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Host	132h	H_CHaMaxPktSize_H		7:	0:	1:	00h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
			R / W	1: MaxPktSize[9]	Max Packet Size High		
				0: MaxPktSize[8]			

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Host	133h	H_CHaMaxPktSize_L	R / W	7: MaxPktSize[7]	Max Packet Size Low	00h
				6: MaxPktSize[6]		
				5: MaxPktSize[5]		
				4: MaxPktSize[4]		
				3: MaxPktSize[3]		
				2: MaxPktSize[2]		
				1: MaxPktSize[1]		
				0: MaxPktSize[0]		

ホスト動作時にチャネル CHa の MaxPacketSize の設定を行います。

**132h.Bit7-2 Reserved**

リザーブ・ビットには”1”を書き込まないで下さい。

**132h.Bit1-0, 133h.Bit7-0 MaxPktSize[9:0]**

チャネル CHa の MaxPacketSize を設定します。

FS 時 8, 16, 32, 64 バイト(バルクオンリーサポート機能を使用する場合は 32, 64 バイトのいずれか)

HS 時 512 バイト

のいずれかに設定してください。

それ以外の値の設定は禁止します。

## 2. レジスタ

2.8.50. 134h H\_CHaTotalSize\_HH (Host Channel a Total Size High-High)

2.8.51. 135h H\_CHaTotalSize\_HL (Host Channel a Total Size High-Low)

2.8.52. 136h H\_CHaTotalSize\_LH (Host Channel a Total Size Low-High)

2.8.53. 137h H\_CHaTotalSize\_LL (Host Channel a Total Size Low-Low)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Host	134h	H_CHaTotalSize_HH	R / W	7: TotalSize[31]	Total Size High-High	00h
				6: TotalSize[30]		
				5: TotalSize[29]		
				4: TotalSize[28]		
				3: TotalSize[27]		
				2: TotalSize[26]		
				1: TotalSize[25]		
				0: TotalSize[24]		

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Host	135h	H_CHaTotalSize_HL	R / W	7: TotalSize[23]	Total Size High-Low	00h
				6: TotalSize[22]		
				5: TotalSize[21]		
				4: TotalSize[20]		
				3: TotalSize[19]		
				2: TotalSize[18]		
				1: TotalSize[17]		
				0: TotalSize[16]		

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Host	136h	H_CHaTotalSize_LH	R / W	7: TotalSize[15]	Total Size Low-High	00h
				6: TotalSize[14]		
				5: TotalSize[13]		
				4: TotalSize[12]		
				3: TotalSize[11]		
				2: TotalSize[10]		
				1: TotalSize[9]		
				0: TotalSize[8]		

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Host	137h	H_CHaTotalSize_LL	R / W	7: TotalSize[7]	Total Size Low-Low	00h
				6: TotalSize[6]		
				5: TotalSize[5]		
				4: TotalSize[4]		
				3: TotalSize[3]		
				2: TotalSize[2]		
				1: TotalSize[1]		
				0: TotalSize[0]		

ホスト動作時にチャンネル CHa で転送を行うデータの Total Size を設定します。

### 134h.Bit7-0, 135h.Bit7-0, 136h.Bit7-0, 137h.Bit7-0      TotalSize[31:0]

チャンネル CHa における転送データの全バイト数(最大 4,294,967,295byte : 約 4Gbyte)を設定します。

H\_CHaConfig\_0 レジスタの TranGo ビットによりトランザクションが開始された後は、トランザクションが正常に終了するたびにこのレジスタは更新されます。

また、TotalSize = 0 の時、H\_CHaConfig\_0 レジスタの TranGo ビットにより OUT トランザクションを開始するとゼロ長パケットが送信されます。

バルクオンリーサポート機能を使用する際はこのレジスタを設定する必要はありません。

## 2. レジスタ

### 2.8.54. 138h H\_CHaHubAdrs (Host Channel a Hub Address)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Host	138h	H_CHaHubAdrs	R / W	7: HubAdrs[3]	Hub Address		00h
				6: HubAdrs[2]			
				5: HubAdrs[1]			
				4: HubAdrs[0]			
				3:	0:	1:	
			R / W	2: Port[2]	Channel a Port Number		
				1: Port[1]			
				0: Port[0]			

ホスト動作時にチャンネル CHa に接続するハブの設定を行います。

#### Bit7-4 HubAdrs[3:0]

チャンネル CHa で転送を行うファンクションが接続されているハブの USB アドレスを設定します。

0～15 までの任意の値が設定できます。

#### Bit3 Reserved

#### Bit2-0 Port[2:0]

チャンネル CHa で転送を行うファンクションが接続されているハブのポートナンバーを設定します。

0～7 までの任意の値が設定できます。



## 2.8.55. 139h H\_CHaFuncAdrs (Host Channel a Function Address)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Host	139h	H_CHaFuncAdrs	R / W	7: FuncAdrs[3]	Function Address	00h
				6: FuncAdrs[2]		
				5: FuncAdrs[1]		
				4: FuncAdrs[0]		
			R / W	3: EP_Number[3]	Endpoint Number	
				2: EP_Number[2]		
				1: EP_Number[1]		
				0: EP_Number[0]		

ホスト動作時にチャンネル CHa で転送を行うファンクションのアドレス設定を行います。

**Bit7-4 FuncAdrs[3:0]**

チャンネル CHa が管理するエンドポイントを含むファンクションの USB アドレスを設定します。

0～15 までの任意の値が設定できます。

**Bit3-0 EP\_Number[3:0]**

チャンネル CHa で転送を行うエンドポイント番号を設定します。

0～15 までの任意の値が設定できます。

バルクオンリーサポート機能を使用する際はこのビットを設定する必要はありません。

## 2. レジスタ

### 2.8.56. 13Ah H\_CHaBO\_SupportControl (Host Bulk Only Transfer Support Control)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Host	13Ah	H_CHaBO_Support- Control		7:	0:	1:	00h
				6:	0:	1:	
			R / W	5: BO_TransportState[1]	Bulk Only Transfer Transport State		
				4: BO_TransportState[0]			
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
			R / W	0: BO_SupportGo	0: Stand by	1: BO Transfer Go	

ホスト動作時にチャネル CHa のバルクオンリーサポート機能の設定を行います。

**Bit7-6**    **Reserved**

**Bit5-4**    **BO\_TransportState[1:0]**

BO\_SupportGo ビットを”1”に設定し、バルクオンリーサポート機能を使用して転送を行っている時にどのトランスポートを実行しているかを示します。

00: Idle    — 転送未実行、または転送が正常に終了したことを示します。

01: CBW Transport    — CBW トランスポートを実行していることを示します。

10: Data Transport    — データトランスポートを実行していることを示します。

11: CSW Transport    — CSW トランスポートを実行していることを示します。

**Bit3-1**    **Reserved**

**Bit0**      **BO\_SupportGo**

このビットを”1”に設定するとチャネル CHa でバルクオンリーサポート機能により、CBW トランスポート～(データトランスポート)～CSW トランスポートが自動的に行われます。

CBW トランスポートでは、OUT トークンを自動的に送出し FIFO の CBW エリアに設定したデータが送信されます。

次にデータステージがある場合は、指定された方向およびサイズでデータトランスポートが自動的に実行されます。

最後に CSW トランスポートでは、IN トークンを自動的に送出し FIFO の CSW エリアにデータを受信します。

以上のトランスポートが正常完了すると、H\_CHaIntStat レジスタの BO\_SupportCmp ビットがセットされます。トランスポートの途中でパケットにエラーを検出した場合、および CSW の値が適切でない場合は H\_CHaIntStat レジスタの BO\_SupportStop ビットがセットされ、トランザクションは停止します。この時は、H\_CHaConditionCode レジスタにその原因がセットされていますので参照して下さい。H\_CHaIntStat レジスタの BO\_SupportStop ビットが”1”に設定された時に ConditionCode の値が”000”である場合は、CSW の値が適切でないことを表しています。

一連のトランスポートが終了した場合(正常終了、エラー終了いずれの場合も)、このビットは自動的にクリアされます。

バルクオンリーサポート機能を実行中にこのビットをクリアすることにより、トランスポートの処理を停止することが出来ます。この場合、CSW トランスポートが正常終了した場合は BO\_SupportCmp ビットが、それ以外は BO\_SupportStop ビットがセットされます。停止したトランスポートは BO\_TransportState にて確認してください。

## 2.8.57. 13Bh H\_CHaBO\_CSW\_RcvDataSize (Host CSW Receive Data Size)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Host	13Bh	H_CHaBO_CSW_RcvDataSize		7:	0:	1:	00h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
			R	3: CSW_RcvDataSize[3]	CSW Resceive Data Size		
				2: CSW_RcvDataSize[2]			
				1: CSW_RcvDataSize[1]			
				0: CSW_RcvDataSize[0]			

ホスト動作時にチャネル CHa のバルクオンリーサポート機能において、CSW トランスポート実行時の受信データ数を示します。

**Bit7-4 Reserved****Bit3-0 CSW\_RcvDataSize[3:0]**

CSW の受信データ数を示します。

CSW トランスポートで13バイト未満のデータを受信した場合、このレジスタにより受信データ数を確認出来ます。

CSW トランスポートでハンドシェークを受信した場合、および CSW トランスポート以外の場合においては、このレジスタの値は意味を持ちません。

## 2. レジスタ

### 2.8.58. 13Ch H\_CHaBO\_OUT\_EP\_Control (Host OUT Endpoint Control)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Host	13Ch	H_CHaBO_OUT_EP_Contr ol		7:	0:	1:	00h
				6:	0:	1:	
				5:	0:	1:	
			R / W	4: OUT_Toggle	0: Toggle0	1: Toggle1	
			R / W	3: OUT_EP_Number[3]	OUT EP Number		
				2: OUT_EP_Number[2]			
				1: OUT_EP_Number[1]			
				0: OUT_EP_Number[0]			

ホスト動作時にチャンネル CHa のバルクオンリーサポート機能の設定を行います。

#### Bit7-5 Reserved

#### Bit4 OUT\_Toggle

H\_CHaBO\_SupportControl レジスタの BO\_SupportGo ビットを”1”に設定し、バルクオンリーサポート機能を使用して OUT 方向の転送(CBW トランスポート、Data OUT トランスポート)を行う時のトグルシーケンスビットの初期値を設定します。

0: トグル 0

1: トグル 1

また、OUT 方向のトランスポートが正常終了した場合、トグルシーケンスビットをこのビットに自動的に保持します。

#### Bit3-0 OUT\_EP\_Number[3:0]

H\_CHaBO\_SupportControl レジスタの BO\_SupportGo ビットを”1”に設定し、バルクオンリーサポート機能を使用して OUT 方向の転送(CBW トランスポート、Data OUT トランスポート)を行う時の転送先デバイスのエンドポイント番号を設定します。

0～15 までの任意の値が設定できます。

## 2.8.59. 13Dh H\_CHaBO\_IN\_EP\_Control (Host IN Endpoint Control)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Host	13Dh	H_CHaBO_IN_EP_Contr ol		7:	0:	1:	00h
				6:	0:	1:	
				5:	0:	1:	
			R / W	4: IN_Toggle	0: Toggle0	1: Toggle1	
			R / W	3: IN_EP_Number[3]	IN EP Number		
				2: IN_EP_Number[2]			
				1: IN_EP_Number[1]			
				0: IN_EP_Number[0]			

ホスト動作時にチャネル CHa のバルクオンリーサポート機能の設定を行います。

**Bit7-5** Reserved**Bit4** IN\_Toggle

H\_CHaBO\_SupportControl レジスタの BO\_SupportGo ビットを”1”に設定し、バルクオンリーサポート機能を使用して IN 方向の転送(CSW トランスポート、Data IN トランスポート)を行う時のトグルシーケンスビットの初期値を設定します。

0: トグル 0

1: トグル 1

また、IN 方向のトランスポートが正常終了した場合、トグルシーケンスビットをこのビットに自動的に保持します。

**Bit3-0** IN\_EP\_Number[3:0]

H\_CHaBO\_SupportControl レジスタの BO\_SupportGo ビットを”1”に設定し、バルクオンリーサポート機能を使用して IN 方向の転送(CSW トランスポート、Data IN トランスポート)を行う時の転送先デバイスのエンドポイント番号を設定します。

0～15 までの任意の値が設定できます。

## 2. レジスタ

### 2.8.60. 13Eh H\_CHaConditionCode (Host Channel a Condition Code)

Mode	Addresses	Register Name	R / W	Bit Symbol	Description	Reset
Host	13Eh	H_CHaConditionCode		7:	0:	1:
			R	6: ConditionCode[2]	Condition Code	
				5: ConditionCode[1]		
				4: ConditionCode[0]		
				3:	0:	1:
				2:	0:	1:
				1:	0:	1:
				0:	0:	1:

ホスト動作時にチャネル CHa の転送完了結果を示します。

**Bit7**      **Reserved**

**Bit6-4**    **ConditionCode[2:0]**

チャネル CHa で転送が完了した場合の結果を示します。

Code	Meaning	Description
000	NOERROR ノーエラー	トランザクションがエラーなく完了した
001	STALL ストール	エンドポイントがストールPIDを返した
010	DATAOVERRUN データ・オーバーラン	<ul style="list-style-type: none"> <li>最大パケットサイズを越えるデータパケットを受信した <ul style="list-style-type: none"> <li>* CRCエラーやビット・スタッフィングエラーが同時に検出された場合はリトライエラーとして処理します</li> </ul> </li> <li>IRP(TotalSize)を越えるデータ数を受信した <ul style="list-style-type: none"> <li>* CRCエラーやビット・スタッフィングエラーが同時に検出された場合はリトライエラーとして処理します</li> </ul> </li> <li>データパケットが最大パケットサイズ以下で、データパケットに含まれたデータグルが予期した値と一致していない場合、データ・オーバーランではなくグルミスマッチとして処理します</li> </ul>
011	DATAUNDERRUN データ・アンダーラン	<ul style="list-style-type: none"> <li>最大パケットサイズ未満のデータパケットを受信し、且つデータ数がIRP(TotalSize)に満たない <ul style="list-style-type: none"> <li>* CRCエラーやビット・スタッフィングエラーが同時に検出された場合はリトライエラーとして処理します</li> </ul> </li> </ul>
100	RETRYERROR リトライエラー	<ul style="list-style-type: none"> <li>デバイスがトークンに規定時間以内に応答しない(IN)、または規定時間以内にハンドシェークを発信しない(OUT)</li> <li>エンドポイントからのデータパケットにCRCエラーが含まれている</li> <li>エンドポイントからのデータパケットにビット・スタッフィングエラーが含まれている</li> <li>エンドポイントからのPID検査ビットが、データPID(IN)またはハンドシェーク(OUT)に失敗した</li> <li>受信PIDが無効またはPID値が定義されていない</li> <li>エンドポイントからのデータパケットに含まれたデータグルが、予期した値と一致していない(グルミスマッチ)</li> </ul>
その他	Reserved	

**Bit3-0**    **Reserved**

**2.8.61. 140h H\_CHbConfig\_0 (Host Channel b Configuration0)****2.8.62. 150h H\_CHcConfig\_0 (Host Channel c Configuration0)****2.8.63. 160h H\_CHdConfig\_0 (Host Channel d Configuration0)****2.8.64. 170h H\_CHeConfig\_0 (Host Channel e Configuration0)**

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Host	140h	H_CHbConfig_0	R / W	7: ACK_Cnt[3]	ACK Count		00h
	150h	H_CHcConfig_0		6: ACK_Cnt[2]			
	160h	H_CHdConfig_0		5: ACK_Cnt[1]			
	170h	H_CHeConfig_0		4: ACK_Cnt[0]			
			R / W	3: SpeedMode[1]	Speed Mode		
				2: SpeedMode[0]			
			R / W	1: Toggle	0: Toggle0	1: Toggle1	
			R / W	0: TranGo	0: Stand by	1: Transaction Start	

ホスト動作時にチャネル CHx{x=b-e}の基本設定を行います。

**Bit7-4 ACK\_Cnt[3:0]**

チャネル CHx{x=b-e}で行われる転送において、ACK をカウントする数を設定します。

設定された値だけ ACK をカウントすると、H\_CHx{x=b-e}IntStat レジスタの TranACK ビットがセットされます。

0000: 16 回の ACK をカウントします。

0001～1111: 1 回～15 回の ACK をカウントします。

**Bit3-2 SpeedMode[1:0]**

チャネル CHx{x=b-e}で転送を行うデバイスの動作モードを設定します。

00: HS モード — HS デバイスの時はこの設定にして下さい。

01: FS モード — FS デバイスの時はこの設定にして下さい。

10: Reserved — 本値の使用を禁止します。

11: LS モード — LS デバイスの時はこの設定にして下さい。

**Bit1 Toggle**

トランザクションを開始する時のトグルシーケンスビットの初期値を設定します。また、トランザクション実行後、およびトランザクション完了後はトグルシーケンスビットの状態を示します。

0: トグル 0

1: トグル 1

**Bit0 TranGo**

このビットを”1”に設定するとチャネル CHx{x=b-e}のトランザクションを開始します。トランザクション開始後このビットを”0”にクリアするとトランザクション処理を停止させることができます。また、このビットはチャネル CHx{x=b-e}がトランザクション実行中か否かのステータスの意味も有します。

0: トランザクションを停止します(トランザクション停止中です)

1: トランザクションを開始します(トランザクション実行中です)

転送が H\_CHx{x=b-e}TotalSize\_HH～LL レジスタ で設定したバイト数完了した時点で、H\_CHx{x=b-e}IntStat レジスタの TotalSizeCmp ビットが”1”にセットされ、本ビットは自動的に”0”に戻ります。H\_CHx{x=b-e}IntStat レジスタの ChangeCondition ビットがセットされた場合などは”0”にリセットされます。この時は、H\_CHx{x=b-e}ConditionCode レジスタにその原因がセットされていますので参照して下さい。

また、このビットのクリアによる停止を行った時は処理最中のトランザクションが終了した時点で、H\_CHx{x=b-e}IntStat レジスタの ChangeCondition ビットがセットされます。トランザクションが停止されても、FIFO 中のデータ、(残りの)トータルサイズ、チャネルに関する設

## 2. レジスタ

---

定はそのままの状態となります。従って再びこのビットを”1”にセットすることによって、トランザクションを停止された時の続きから再開させることが可能です。(新たなトランザクションを行う場合は **FIFO** をクリアし、チャネル情報の設定をし直してください。)



**2.8.65. 141h H\_CHbConfig\_1 (Host Channel b Configuration1)****2.8.66. 151h H\_CHcConfig\_1 (Host Channel c Configuration1)****2.8.67. 161h H\_CHdConfig\_1 (Host Channel d Configuration1)****2.8.68. 171h H\_CHeConfig\_1 (Host Channel e Configuration1)**

Mode	Addresses	Register Name	R / W	Bit Symbol	Description		Reset
Host	141h	H_CHbConfig_1	R / W	7: TID[1]	Transaction ID		00h
	151h	H_CHcConfig_1		6: TID[0]			
	161h	H_CHdConfig_1	R / W	5: TranType[1]	Transfer Type		
	171h	H_CHeConfig_1		4: TranType[0]			
			R / W	3: AutoZerolen	0: Do nothing	1: Add Zerolen	
			R / W	2: Audio441	0: Disable	1: Enable	
			R / W	1: TotalSizeFree[1]	TotalSizeFree[1:0]		
				0: TotalSizeFree[0]			

ホスト動作時にチャンネル CHx{x=b-e}の基本設定を行います。

**Bit7-6 TID[1:0]**

チャンネル CHx{x=b-e}で発行するトランザクションの種類(OUT、IN)を設定します。

- 00: Reserved — 本値の使用を禁止します。
- 01: OUT — OUT トークンを発行します。
- 10: IN — IN トークンを発行します。
- 11: Reserved — 本値の使用を禁止します。

**Bit5-4 TranType[1:0]**

チャンネル CHx{x=b-e}で行う転送の種別を設定します。

- 00: Reserved — 本値の使用を禁止します。
- 01: Isochronous — アイソクロナス転送を行います。
- 10: Bulk — バルク転送を行います。
- 11: Interrupt — インタラプト転送を行います。

**Bit3 AutoZerolen**

このビットに”1”を設定すると H\_CHx{x=b-e}TotalSizeHH~LL レジスタで設定したサイズの転送がちょうど MaxPacketSize で完了した際に、ゼロ長パケットを最後に自動付与します。OUT 転送の場合のみこのビットは有効となります。

**Bit2 Audio441**

このビットに”1”を設定するとオーディオクラス・アシスト機能が有効になります。オーディオクラス・アシスト機能はアイソクロナス転送でサンプリング周波数 44.1kHz の 16bit、2 チャンネル PCM データを 1ms 周期で送信する時に使用します。オーディオクラス・アシスト機能が有効の時は送信するデータパケットのサイズを自動で調整します。このときデータパケットのサイズを 176Byte として 9 回連続トランザクションを行った後、データパケットのサイズを 180Byte として 1 回トランザクションを行います。

**Bit1-0 TotalSizeFree[1:0]**

H\_CHx{x=b-e}TotalSize\_HH~LL レジスタによる転送の制御を設定します。

- 00 : H\_CHx{x=b-e}TotalSize\_HH~LL レジスタで設定したサイズの転送が完了すると TranGo ビットをクリアし転送を終了します。そして、この時 TotalSizeCmp 割り込みを発生します。OUT 転送の場合、最後のデータパケットのサイズは MaxPktSize と TotalSize の小さい方の値となります。IN 転送の場合、期待する最後のデータパケットのサイズは MaxPktSize と TotalSize の小さい方の値となります。
- 01 : H\_CHx{x=b-e}TotalSize\_HH~LL レジスタの値に関係なく転送が行われます。OUT 転送の場合、データパケットのサイズは MaxPktSize の値となります。IN 転送の場合、期待するデータパケットのサイズは MaxPktSize の値となります。この設定

## 2. レジスタ

---

では H\_CHx{x=b-e}TotalSize\_HH~LL レジスタで設定したサイズの転送が完了しても TranGo ビットのクリアは行わず、TotalSizeCmp 割り込みも発生しません。

- 10 : H\_CHx{x=b-e}TotalSize\_HH~LL レジスタで設定したサイズの転送が完了すると時 TotalSizeCmp 割り込みを発生します。この時 TranGo ビットはクリアされませんが TotalSize の値が “0”の間中このチャネルのトランザクションの発行は停止されます。OUT 転送の場合、データパケットのサイズは MaxPktSize と TotalSize の小さい方の値となります。IN 転送の場合、期待するデータパケットのサイズは MaxPktSize と TotalSize の小さい方の値となります。

11 : Reserved

- 2.8.69. 142h H\_CHbMaxPktSize\_H (Host Channel b Max Packet Size High)  
 2.8.70. 143h H\_CHbMaxPktSize\_L (Host Channel b Max Packet Size Low)  
 2.8.71. 152h H\_CHcMaxPktSize\_H (Host Channel c Max Packet Size High)  
 2.8.72. 153h H\_CHcMaxPktSize\_L (Host Channel c Max Packet Size Low)  
 2.8.73. 162h H\_CHdMaxPktSize\_H (Host Channel d Max Packet Size High)  
 2.8.74. 163h H\_CHdMaxPktSize\_L (Host Channel d Max Packet Size Low)  
 2.8.75. 172h H\_CHeMaxPktSize\_H (Host Channel e Max Packet Size High)  
 2.8.76. 173h H\_CHeMaxPktSize\_L (Host Channel e Max Packet Size Low)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Host	142h	H_CHbMaxPktSize_H		7:	0:	1:	00h
	152h	H_CHcMaxPktsize_H		6:	0:	1:	
	162h	H_CHdMaxPktSize_H		5:	0:	1:	
	172h	H_CHeMaxPktSize_H		4:	0:	1:	
				3:	0:	1:	
			R / W	2: MaxPktSize[10]	Max Packet Size High		
				1: MaxPktSize[9]			
				0: MaxPktSize[8]			

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Host	143h	H_CHbMaxPktSize_L	R / W	7: MaxPktSize[7]	Max Packet Size Low	00h
	153h	H_CHcMaxPktSize_L		6: MaxPktSize[6]		
	163h	H_CHdMaxPktSize_L		5: MaxPktSize[5]		
	173h	H_CHeMaxPktSize_L		4: MaxPktSize[4]		
				3: MaxPktSize[3]		
				2: MaxPktSize[2]		
				1: MaxPktSize[1]		
				0: MaxPktSize[0]		

ホスト動作時にチャネル CHx{x=b-e}の MaxPacketSize の設定を行います。

#### 142h.Bit7-3 Reserved

リザーブ・ビットには”1”を書き込まないで下さい。

#### 142h.Bit2-0, 143h.Bit7-0 MaxPktSize[10:0]

チャネル CHx{x=b-e}の MaxPacketSize を設定します。

このチャネルをバルク転送用として使用する場合には、

FS 時 8, 16, 32, 64 バイト

HS 時 512 バイト

のいずれかに設定して下さい。

このチャネルをインタラプト転送用として使用する場合は、

LS 時 8 バイトまで

FS 時 64 バイトまで

HS 時 512 バイトまで

の任意の転送数が設定可能です。

このチャネルをアイソクロナス転送用として使用する場合は、

FS 時 1023 バイトまで

HS 時 1024 バイトまで

の任意の転送数が設定可能です。

それ以外の値の設定は禁止します。

## 2. レジスタ

---

- 2.8.77. 144h H\_CHbTotalSize\_HH (Host Channel b Total Size High-High)
- 2.8.78. 145h H\_CHbTotalSize\_HL (Host Channel b Total Size High-Low)
- 2.8.79. 146h H\_CHbTotalSize\_LH (Host Channel b Total Size Low-High)
- 2.8.80. 147h H\_CHbTotalSize\_LL (Host Channel b Total Size Low-Low)
- 2.8.81. 154h H\_CHcTotalSize\_HH (Host Channel c Total Size High-High)
- 2.8.82. 155h H\_CHcTotalSize\_HL (Host Channel c Total Size High-Low)
- 2.8.83. 156h H\_CHcTotalSize\_LH (Host Channel c Total Size Low-High)
- 2.8.84. 157h H\_CHcTotalSize\_LL (Host Channel c Total Size Low-Low)
- 2.8.85. 164h H\_CHdTotalSize\_HH (Host Channel d Total Size High-High)
- 2.8.86. 165h H\_CHdTotalSize\_HL (Host Channel d Total Size High-Low)
- 2.8.87. 166h H\_CHdTotalSize\_LH (Host Channel d Total Size Low-High)
- 2.8.88. 167h H\_CHdTotalSize\_LL (Host Channel d Total Size Low-Low)
- 2.8.89. 174h H\_CHeTotalSize\_HH (Host Channel e Total Size High-High)
- 2.8.90. 175h H\_CHeTotalSize\_HL (Host Channel e Total Size High-Low)
- 2.8.91. 176h H\_CHeTotalSize\_LH (Host Channel e Total Size Low-High)
- 2.8.92. 177h H\_CHeTotalSize\_LL (Host Channel e Total Size Low-Low)

次頁へ

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Host	144h	H_CHbTotalSize_HH	R / W	7: TotalSize[31]	Total Size High-High	00h
	154h	H_CHcTotalSize_HH		6: TotalSize[30]		
	164h	H_CHdTotalSize_HH		5: TotalSize[29]		
	174h	H_CHeTotalSize_HH		4: TotalSize[28]		
				3: TotalSize[27]		
				2: TotalSize[26]		
				1: TotalSize[25]		
				0: TotalSize[24]		

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Host	145h	H_CHbTotalSize_HL	R / W	7: TotalSize[23]	Total Size High-Low	00h
	155h	H_CHcTotalSize_HL		6: TotalSize[22]		
	165h	H_CHdTotalSize_HL		5: TotalSize[21]		
	175h	H_CHeTotalSize_HL		4: TotalSize[20]		
				3: TotalSize[19]		
				2: TotalSize[18]		
				1: TotalSize[17]		
				0: TotalSize[16]		

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Host	146h	H_CHbTotalSize_LH	R / W	7: TotalSize[15]	Total Size Low-High	00h
	156h	H_CHcTotalSize_LH		6: TotalSize[14]		
	166h	H_CHdTotalSize_LH		5: TotalSize[13]		
	176h	H_CHeTotalSize_LH		4: TotalSize[12]		
				3: TotalSize[11]		
				2: TotalSize[10]		
				1: TotalSize[9]		
				0: TotalSize[8]		

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Host	147h	H_CHbTotalSize_LL	R / W	7: TotalSize[7]	Total Size Low-Low	00h
	157h	H_CHcTotalSize_LL		6: TotalSize[6]		
	167h	H_CHdTotalSize_LL		5: TotalSize[5]		
	177h	H_CHeTotalSize_LL		4: TotalSize[4]		
				3: TotalSize[3]		
				2: TotalSize[2]		
				1: TotalSize[1]		
				0: TotalSize[0]		

ホスト動作時にチャネル CHx{x=b-e} で転送を行うデータの Total Size を設定します。

#### 1X4h.Bit7-0, 1X5h.Bit7-0, 1X6h.Bit7-0, 1X7h.Bit7-0 TotalSize[31:0]

チャネル CHx{x=b-e} における転送データの全バイト数(最大 4,294,967,295byte : 約 4Gbyte)を設定します。

H\_CHx{x=b-e}Config\_0 レジスタの TranGo ビットによりトランザクションが開始された後は、トランザクションが正常に終了するたびにこのレジスタは更新されます。

## 2. レジスタ

---

また、TotalSize=0 の時、H\_CHx{x=b-e}Config\_0 レジスタの TranGo ビットにより OUT トランザクションを開始するとゼロ長パケットが送信されます。

2.8.93. 148h H\_CHbHubAdrs (Host Channel b Hub Address)

2.8.94. 158h H\_CHcHubAdrs (Host Channel c Hub Address)

2.8.95. 168h H\_CHdHubAdrs (Host Channel d Hub Address)

2.8.96. 178h H\_CHeHubAdrs (Host Channel e Hub Address)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Host	148h	H_CHbHubAdrs	R / W	7: HubAdrs[3]	Hub Address		00h
	158h	H_CHcHubAdrs		6: HubAdrs[2]			
	168h	H_CHdHubAdrs		5: HubAdrs[1]			
	178h	H_CHeHubAdrs		4: HubAdrs[0]			
				3:	0:	1:	
			R / W	2: Port[2]	Port Number		
				1: Port[1]			
				0: Port[0]			

ホスト動作時にチャンネル CHx{x=b-e}に接続するハブの設定を行います。

**Bit7-4 HubAdrs[3:0]**

チャンネル CHx{x=b-e}で転送を行うファンクションが接続されているハブのUSBアドレスを設定します。

0～15 までの任意の値が設定できます。

**Bit3 Reserved****Bit2-0 Port[2:0]**

チャンネル CHx{x=b-e}で転送を行うファンクションが接続されているハブのポートナンバーを設定します。

0～7 までの任意の値が設定できます。

## 2. レジスタ

2.8.97. 149h H\_CHbFuncAdrs (Host Channel b Function Address)

2.8.98. 159h H\_CHcFuncAdrs (Host Channel c Function Address)

2.8.99. 169h H\_CHdFuncAdrs (Host Channel d Function Address)

2.8.100. 179h H\_CHeFuncAdrs (Host Channel e Function Address)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Host	149h	H_CHbFuncAdrs	R / W	7: FuncAdrs[3]	Function Address	00h
	159h	H_CHcFuncAdrs		6: FuncAdrs[2]		
	169h	H_CHdFuncAdrs		5: FuncAdrs[1]		
	179h	H_CHeFuncAdrs		4: FuncAdrs[0]		
			R / W	3: EP_Number[3]	Endpoint Number	
				2: EP_Number[2]		
				1: EP_Number[1]		
				0: EP_Number[0]		

ホスト動作時にチャンネル CHx{x=b-e} で転送を行うファンクションのアドレス設定を行います。

### Bit7-4 FuncAdrs[3:0]

チャンネル CHx{x=b-e} が管理するエンドポイントを含むファンクションの USB アドレスを設定します。

0～15 までの任意の値が設定できます。

### Bit3-0 EP\_Number[3:0]

チャンネル CHx{x=b-e} で転送を行うエンドポイント番号を設定します。

0～15 までの任意の値が設定できます。



- 2.8.101. 14Ah H\_CHbInterval\_H (Host Channel b Interval High)  
 2.8.102. 14Bh H\_CHbInterval\_L (Host Channel b Interval Low)  
 2.8.103. 15Ah H\_CHcInterval\_H (Host Channel c Interval High)  
 2.8.104. 15Bh H\_CHcInterval\_L (Host Channel c Interval Low)  
 2.8.105. 16Ah H\_CHdInterval\_H (Host Channel d Interval High)  
 2.8.106. 16Bh H\_CHdInterval\_L (Host Channel d Interval Low)  
 2.8.107. 17Ah H\_CHeInterval\_H (Host Channel e Interval High)  
 2.8.108. 17Bh H\_CHeInterval\_L (Host Channel e Interval Low)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Host	14Ah	H_CHbInterval_H		7:	0:	1:
	15Ah	H_CHcInterval_H		6:	0:	1:
	16Ah	H_CHdInterval_H		5:	0:	1:
	17Ah	H_CHeInterval_H		4:	0:	1:
				3:	0:	1:
			R / W	2: Interval[10]	Interrupt Transfer Interval High	
				1: Interval[9]		
				0: Interval[8]		
						00h

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Host	14Bh	H_CHbInterval_L	R / W	7: Interval[7]	Interrupt Transfer Interval Low	
	15Bh	H_CHcInterval_L		6: Interval[6]		
	16Bh	H_CHdInterval_L		5: Interval[5]		
	17Bh	H_CHeInterval_L		4: Interval[4]		
				3: Interval[3]		
				2: Interval[2]		
				1: Interval[1]		
				0: Interval[0]		
						00h

ホスト動作時にチャネル CH<sub>x</sub>{x=b-e}でインタラプト転送及びアイソクロナス転送を行う時のインターバル値を設定します。

#### 1XAh.Bit7-3 Reserved

#### 1XAh.Bit2-0, 1XBh.Bit7-0 Interval[10:0]

インタラプト転送及びアイソクロナス転送のトークン発行間隔(周期)をこのレジスタにより指定します。下位 3 ビットはマイクロフレーム(125  $\mu$ s)単位で指定し、上位 7 ビットはフレーム(ms)単位で指定します。このレジスタの設定は H\_CH<sub>x</sub>{x=b-e}Config1 レジスタの TranType ビットが"11"(インタラプト転送)の場合、または、TranType ビットが"01" (アイソクロナス転送)の場合に有効です。

このレジスタの"0"の設定は無効となります。

トランザクションの再送時もこのレジスタの設定間隔で行います。

Interval[2:0]  $\mu$  Frame — インターバルを 125  $\mu$ s 単位で指定します。1、2、4 マイクロフレームのいずれかに設定してください。それ以外の値の設定は禁止します。また、本ビットを設定する際は、Interval[10:3]はすべて"0"に設定してください。

Interval[10:3] Frame — インターバルを ms 単位で指定します。1~255 フレームの任意の値を設定できます。また、本ビットを設定する際は、Interval[2:0]はすべて"0"に設定してください。

## 2. レジスタ

2.8.109. 14Ch H\_CHbTranPause(Host Channel b Transaction Pause)

2.8.110. 15Ch H\_CHcTranPause(Host Channel c Transaction Pause)

2.8.111. 16Ch H\_CHdTranPause(Host Channel d Transaction Pause)

2.8.112. 17Ch H\_CHeTranPause(Host Channel e Transaction Pause)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Host	14Ch	H_CHbTranPause		7:	0:	1:	00h
	15Ch	H_CHcTranPause		6:	0:	1:	
	16Ch	H_CHdTranPause		5:	0:	1:	
	17Ch	H_CHeTranPause		4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
			R/W	1: EnTranPause	0: Disable	1: Enable	
			R/W	0: TranPause	0: Do nothing	1: Transaction Pause	

**Bit7-2**    **Reserved**

**Bit1**      **EnTranPause**

このビットに”1”を設定すると、H\_CHx{x=b-e}IntStat レジスタの TranACK ビットがセットされる時、同時に H\_CHx{x=b-e}TranPause レジスタの TranPause ビットが”1”にセットされます。

**Bit0**      **TranPause**

このビットが”1”の時、H\_CHx{x=b-e}Config\_0 の TranGo ビットが”1”に設定されていてもこのチャンネルではトランザクションは実行されません。このチャンネルがインタラプト転送あるいはアイソクロナス転送に設定されている場合、このビットが”1”の為にトランザクションが実行されなくても転送の周期は維持されます。

## 2.8.113. 14Eh H\_CHbConditionCode (Host Channel b Condition Code)

## 2.8.114. 15Eh H\_CHcConditionCode (Host Channel c Condition Code)

## 2.8.115. 16Eh H\_CHdConditionCode (Host Channel d Condition Code)

## 2.8.116. 17Eh H\_CHeConditionCode (Host Channel e Condition Code)

Mode	Addresses	Register Name	R / W	Bit Symbol	Description	Reset
Host	14Eh	H_CHbConditionCode	R	7:	0:	1:
	15Eh	H_CHcConditionCode		6: ConditionCode[2]	Condition Code	
	16Eh	H_CHdConditionCode		5: ConditionCode[1]		
	17Eh	H_CHeConditionCode		4: ConditionCode[0]		
				3:	0:	1:
				2:	0:	1:
				1:	0:	1:
				0:	0:	1:

ホスト動作時にチャネル CHx{x=b-e}の転送完了結果を示します。

**Bit7**      **Reserved**

**Bit6-4**    **ConditionCode[2:0]**

チャネル CHx{x=b-e}で転送が完了した場合の結果を示します。

Code	Meaning	Description
000	NOERROR ノーエラー	トランザクションがエラーなく完了した
001	STALL ストール	エンドポイントがストールPIDを返した
010	DATAOVERRUN データ・オーバーラン	<ul style="list-style-type: none"> <li>最大パケットサイズを越えるデータパケットを受信した * CRCエラーやビットスタッフィングエラーが同時に検出された場合はリトライエラーとして処理します</li> <li>IRP(TotalSize)を越えるデータ数を受信した * CRCエラーやビットスタッフィングエラーが同時に検出された場合はリトライエラーとして処理します</li> <li>* データパケットが最大パケットサイズ以下で、データパケットに含まれたデータグルが予期した値と一致していない場合、データ・オーバーランではなくグルミスマッチとして処理します</li> </ul>
011	DATAUNDERRUN データ・アンダーラン	<ul style="list-style-type: none"> <li>最大パケットサイズ未満のデータパケットを受信し、且つデータ数がIRP(TotalSize)に満たない * CRCエラーやビットスタッフィングエラーが同時に検出された場合はリトライエラーとして処理します</li> </ul>
100	RETRYERROR リトライエラー	<ul style="list-style-type: none"> <li>デバイスがトークンに規定時間以内に応答しない(IN)、または規定時間以内にハンドシェークを発信しない(OUT)</li> <li>エンドポイントからのデータパケットにCRCエラーが含まれている</li> <li>エンドポイントからのデータパケットにビットスタッフィングエラーが含まれている</li> <li>エンドポイントからのPID検査ビットが、データPID(IN)またはハンドシェーク(OUT)に失敗した</li> <li>受信PIDが無効またはPID値が定義されていない</li> <li>エンドポイントからのデータパケットに含まれたデータグルが、予期した値と一致していない(トグルミスマッチ)</li> </ul>
110	BUFFEROVERRUN バッファ・オーバーラン	アイソクロナス転送においてFIFOの空き容量が最大パケットサイズ未満であるためトランザクションが行われなかった。
111	BUFFERUNDERRUN バッファ・アンダーラン	アイソクロナス転送においてFIFOの有効データ数が不足のためトランザクションが行われなかった。

**Bit3-0**    **Reserved**

## 2. レジスタ

### 2.8.117. 1B0h H\_TriggerFrameNum\_H (Host Trigger Frame Number High)

### 2.8.118. 1B1h H\_TriggerFrameNum\_L (Host Trigger Frame Number Low)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Host	1B0h	H_TriggerFrameNum_H	R / W	7: UseTriggerFrame	0: Don't Use	1: Use	00h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
			R / W	2: TriggerFrameNum[10]	TriggerFrameNum[10:8]		
				1: TriggerFrameNum[9]			
				0: TriggerFrameNum[8]			

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Host	1B1h	H_TriggerFrameNum_L	R / W	7: TriggerFrameNum[7]	TriggerFrameNum[7:0]	00h
				6: TriggerFrameNum[6]		
				5: TriggerFrameNum[5]		
				4: TriggerFrameNum[4]		
				3: TriggerFrameNum[3]		
				2: TriggerFrameNum[2]		
				1: TriggerFrameNum[1]		
				0: TriggerFrameNum[0]		

H\_FrameIntStat.TriggerFrame 要因がセットされるフレーム番号を設定します。

#### 1B0h.Bit7 UseTriggerFrame

このビットが“1”のとき、ホストコントローラが SOF トークンを送信する際に、フレームナンバー(H\_FameNumber\_H,L)と

トリガーフレームナンバー (H\_TriggerFrameNum\_H,L) が一致すると

H\_FrameIntStat.TriggerFrame を"1"にセットします。

#### 1B0h.Bit6-3 Reserved

#### 1B0h.Bit2-0, 1B1h.Bit7-0 TriggerFrameNum[10:0]

H\_FrameIntStat.TriggerFrame 割込みを発生させるフレーム番号を指定します。

2.8.119. 1BEh *HTM\_Config* (Host Transceiver Macro Config)

Mode	Address	Register Name	R / W	Bit Symbol	Description	Reset
Host	1BEh	<b>HTM_Config</b>		7:	0:	1:
				6:	0:	1:
			R / W	5: <b>HTM_SlopeValue[1]</b>	HTM Slope Value[1:0]	
			R / W	4: <b>HTM_SlopeValue[0]</b>		
				3:	0:	1:
				2:	0:	1:
			R / W	1: <b>HTM_TermValue[1]</b>	HTM Termination Value[1:0]	
			R / W	0: <b>HTM_TermValue[0]</b>		
						XXh

トランシーバマクロの調整値設定用レジスタです。

**Bit7-6**    **Reserved**

**Bit5-4**    **HTM\_SlopeValue[1:0]**

HS トランスミッタのスルーレートを調整します。4 段階の調整が可能です。

00 : 緩

01 : ↑

10 : ↓

11 : 急

**Bit3-2**    **Reserved**

**Bit1-0**    **HTM\_TermValue[1:0]**

HS 伝送路のターミネーションを調整します。4 段階の調整が可能です。

00 : 高

01 : ↑

10 : ↓

11 : 低

## 2. レジスタ

---

### 2.8.120. 1F5h H\_Protect (Host Protect)

Mode	Address	Register Name	R / W	Bit Symbol	Description		Reset
Host	1F5h	H_Protect		7:	Don't change values		XXh
				6:			
				5:			
				4:			
			R / W	3: PortSpeedWrEnb	0: Protect	1: Free	
				2:	Don't change values		
				1:			
				0:			

トランシーバマクロの調整値設定用レジスタです。

**Bit7-4**    **Reserved**

**Bit3**      **PortSpeedWrEnb**

"1"をセットすると、H\_NegoControl\_1.PortSpeed を書き換え可能にします。

"0"にクリアすると、H\_NegoControl\_1.PortSpeed を書き込めないように保護します。

**Bit2-0**    **Reserved**

## Appendix A IDE\_Config\_1.Swap ビット設定

S2R72C05 の内部バスはビッグエンディアンで構成されており、[15:8]側がファーストバイトとなっています。これに対し IDE I/F はリトルエンディアンであり、[7:0]側がファーストバイトとなります。S2R72C05 では、IDE\_Config\_1.Swap ビットを使用して、S2R72C05 の内部バスと IDE I/F とのデータ・バスの結線を切り替えることが出来ます。

以下では、IDE\_Config\_1.Swap ビットの設定による H/W の動作に関し説明致します。

なお、表中の A⇒B は、A の値が B に反映される、となります。

### ●IDE\_Control.IDE\_Go ビットによるデータの DMA 転送

Swap	HDD[15:0]	
	IDE リード	IDE ライト
0	HDD[15:0] ⇒ 内部バス[15:0]	内部バス[15:0] ⇒ HDD[15:0]
1(初期値)	HDD[15:0] ⇒ { 内部バス [7:0], 内部バス [15:8]}	内部バス [15:0] ⇒ {HDD[7:0], HDD [15:8]}

### ●IDE データレジスタアクセス

Swap	IDE データレジスタ HDD[15:0]			
	IDE リード		IDE ライト	
	IDE_RegAdrs. IDE_RdReg	IDE_RegConfig. EnAutoStsRd	IDE_RegAdrs. IDE_WrReg	IDE_SeqWrRegControl. IDE_SeqWrReg
0	HDD[15:8] ⇒ IDE_RdRegValue_0 HDD[7:0] ⇒ IDE_RdRegValue_1	なし	IDE_WrRegValue_0 ⇒ HDD[15:8] IDE_WrRegValue_1 ⇒ HDD[7:0]	IDE_SeqWrRegValue(1 <sup>st</sup> ) ⇒ HDD[7:0] IDE_SeqWrRegValue(2 <sup>nd</sup> ) ⇒ HDD[15:8]
1 (初期 値)	HDD[15:8] ⇒ IDE_RdRegValue_1 HDD[7:0] ⇒ IDE_RdRegValue_0	なし	IDE_WrRegValue_0 ⇒ HDD[7:0] IDE_WrRegValue_1 ⇒ HDD[15:8]	IDE_SeqWrRegValue(1 <sup>st</sup> ) ⇒ HDD[15:8] IDE_SeqWrRegValue(2 <sup>nd</sup> ) ⇒ HDD[7:0]

### ●IDE タスクファイルレジスタアクセス

Swap	IDE タスクファイルレジスタ HDD[7:0]			
	IDE リード		IDE ライト	
	IDE_RegAdrs. IDE_RdReg	IDE_RegConfig. EnAutoStsRd	IDE_RegAdrs. IDE_WrReg	IDE_SeqWrRegControl. IDE_SeqWrReg
0	HDD[7:0] ⇒ IDE_RdRegValue_0 HDD[7:0] ⇒ IDE_RdRegValue_1	同左	IDE_WrRegValue_1 ⇒ HDD[7:0]	IDE_SeqWrRegValue ⇒ HDD[7:0]
1 (初期 値)	同上	同上	IDE_WrRegValue_0 ⇒ HDD[7:0]	同上

### Appendix B リトルエンディアンの CPU への接続

S2R72C05 の内部バスはビッグエンディアンで構成されており、偶数アドレスが上位バイト、奇数アドレスが下位バイトとなっています。これに対して、リトルエンディアンの CPU に接続しての使用方法を説明致します。

#### <基板>

リトルエンディアンの CPU と 72C05 の端子は、データ・バスとライト制御信号につきましては端子名称のまま接続して下さい。すなわち CPU のデータ・バスのビット 15 からビット 8(上位バイト)には 72C05 の CD15 から CD8 を接続し、CPU のデータ・バスのビット 7 からビット 0(下位バイト)には 72C05 の CD7 から CD0 を接続してください。また、ライト信号につきましても、ハイ、ローをそのまま接続してください。

なお、ご使用する CPU によってはライト信号自身の仕様が異なりますので、注意してください。

#### <F/W>

リトルエンディアンの CPU にて、本 LSI を動作させる時には、未初期化期間に以下の設定を行ってください。

##### ① CPUIF\_MODE.CPU\_Endian ビットを”1”にセット

CPUIF\_MODEレジスタは、本LSIがハードリセットされた直後の未初期化期間にのみアクセス可能です。未初期化期間においては、72C05 へのアクセスは、アドレスに係わり無くこのレジスタへのアクセスとなります。詳細は「1.8.2.CPUIFのモード設定」をご参照ください。

CPU\_Endian を上記に設定することにより、バスの上位バイト・下位バイトが入れ替わり、偶数アドレスが下位バイト、奇数アドレスが上位バイト、となります。但し、16bit 単位で意味を持つレジスタ(名称が、\*\_H, \*\_M, \*\_L, \*\_HH, \*\_HL, \*\_LH, \*\_LL のもの)は、CPU\_Endian の設定により、アドレスの偶数・奇数が入れ替わります。

	Big Endian(CPU_Endian=0)		Little Endian(CPU_Endian=1)	
	Even Address CD[15:8]	Odd Address CD[7:0]	Even Address CD[7:0]	Odd Address CD[15:8]
例 1: 8bit 単位のレジスタ	MainIntStat	DeviceIntStat	MainIntStat	DeviceIntStat
例 2: 8bit 単位のレジスタ	FIFO_Rd_0	FIFO_Rd_1	FIFO_Rd_0	FIFO_Rd_1
例 3: 16bit 単位のレジスタ	FIFO_RdRemain_H	FIFO_RdRemain_L	FIFO_RdRemain_L	FIFO_RdRemain_H
例 4: 16bit 単位のレジスタ	DMA0Count_HH	DMA0Count_HL	DMA0Count_HL	DMA0Count_HH
例 5: 16bit 単位のレジスタ	DMA0Count_LH	DMA0Count_LL	DMA0Count_LL	DMA0Count_LH

上記により、この設定を行い初期化済み期間に移行すると、すべての内部レジスタが、Char もしくは Short にてアクセス可能となります。また CPU の DMAC を使用してのアクセスに対しても適切なエンディアンに設定されます。(以下の表を参照)。

例 USB から 01\_02\_03\_04\_05\_06 と順にデータを受け取った場合の FIFO\_Rd\_0/1 レジスタへのアクセス



Short にてアクセス	CPU のアクセス方法			
	ビッグエンディアン		リトルエンディアン	
	CD[15:8]	CD[7:0]	CD[15:8]	CD[7:0]
1 <sup>st</sup>	01	02	02	01
2 <sup>nd</sup>	03	04	04	03
3 <sup>rd</sup>	05	06	06	05

ただし、Short より大きなサイズのレジスタにつきましては、Short にて分割アクセスして頂き、CPU のメモリ上にてキャストしてご使用ください。

### Appendix C 1 ポートモード

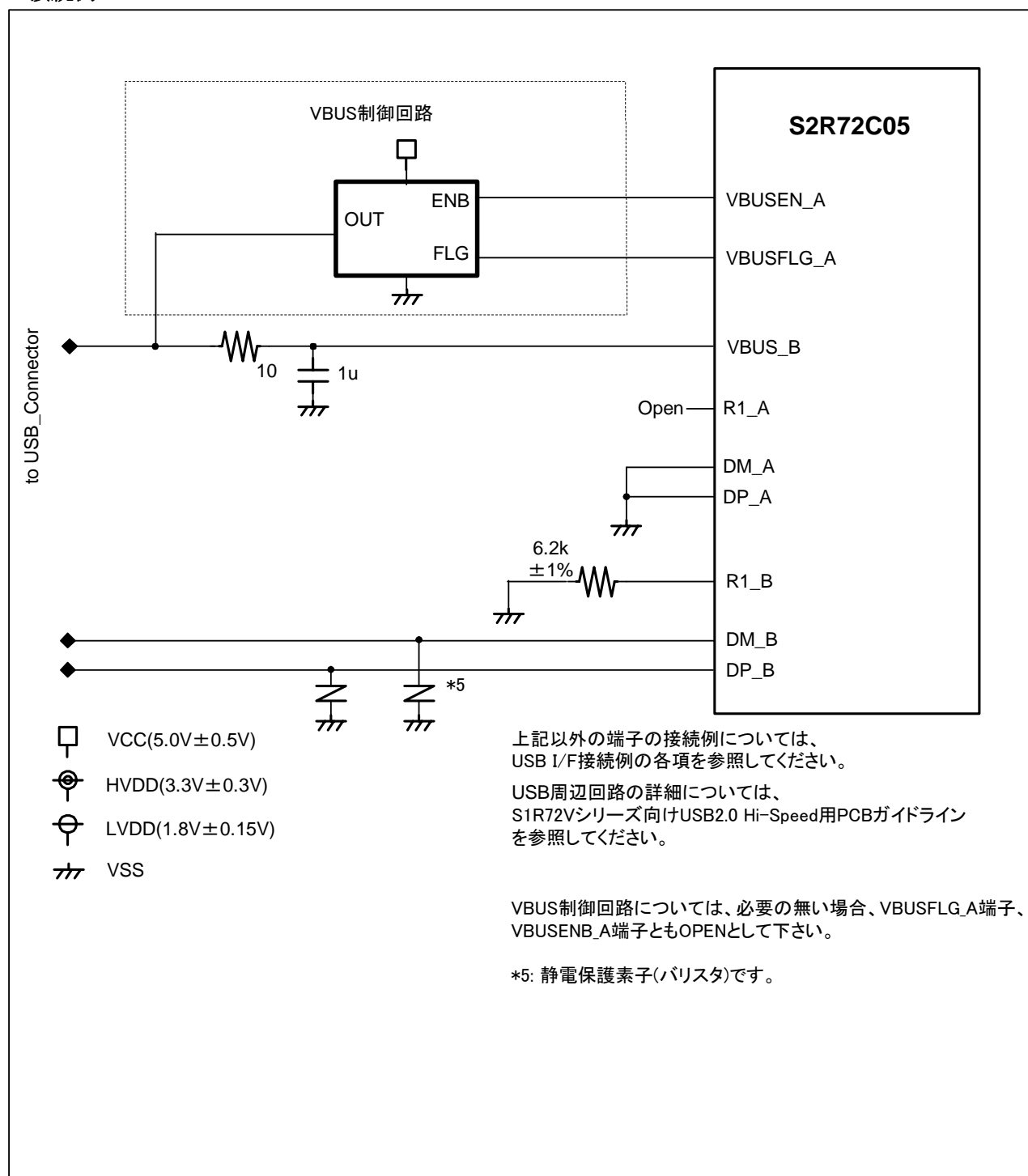
S2R72C05 は、初期設定時に ClkSelect.PORT1x2 ビットを設定することにより、2 ポートモード (ClkSelect.PORT1x2 = “0”) と 1 ポートモード (ClkSelect.PORT1x2 = “1”) を選択できます。2 ポートモードでは、A ポートにてホスト機能を、B ポートにてデバイス機能を実現します。対して、1 ポートモードでは、B ポートのみを使用してホスト／デバイスの両機能を実現します。

なお、1 ポートモードを選択する場合には、空きポートである A ポートを下図を参考に処理してください。また、ホスト機能時の VBUS 制御信号 (VBUSEN\_A、VBUSFLG\_A) は、末尾の\_A に関わらず、VBUS 制御回路にて使用してください。

さらに、1 ポートモードで使用する場合には、パワーマネジメント機能のうち、ACT\_HOST ステート、ACT\_ALL ステートを使用しないようにしてください。なお、PM\_Control\_0.GoActHost、PM\_Control\_0.GoActAllDevice、PM\_Control\_0.GoActAllHost ビットは、1 ポートモードでは使用できないようにマスクしており、セットできません。したがって、1 ポートモードを使用している場合において、ホスト機能、デバイス機能のどちらを実行する場合においても、パワーマネジメント機能としては、ACT\_DEVICE ステートで使用してください。

また、D\_Reset.ResetDTM および H\_Reset.ResetHTM は、両方のビットともに”0”を設定してください。

<接続例>



## Appendix D SetAddress リクエストの応答について(TS のみ適用)

自動アドレス設定機能が有効にされている場合に、bmRequestType が 0(標準リクエスト)以外であり、かつ、bRequest が 0x05 であるリクエストを受信すると、RcvEP0SETUP 割り込みステータスが発行されません。

この問題は、自動アドレス設定機能が、SetAddress リクエスト(bmRequestType==0, bRequest==0x05)を自動処理することに伴い、bRequest 値を以って RcvEP0SETUP 割り込みステータスをマスクしていることに起因しています。

この問題に対処するため、下記いずれかの措置をお願いします。

### 1. ベンダ及びクラスリクエストを制限する

bRequest==0x05 となるベンダリクエスト、またはクラスリクエストが使用されない場合、特別な対処は必要ありません。

### 2. 自動アドレス設定機能を無効にする

自動アドレス設定機能を無効にすることによって、本問題を対策することが出来ます。この場合、SetAddress リクエストを受信して、自動的にステータスステージを実行する機能が無効になり、他のリクエストと同様に f/w でステータスステージを実行する必要があります。但し、自動アドレス設定機能の一部の機能(D\_USB\_Address.SetAddress)を使用し、D\_USB\_Address レジスタへのアドレス値の設定を自動化することが出来ます。

以下に、自動アドレス設定機能を無効にする設定と、同機能が無効である場合の制御シーケンスを説明します。また、比較のため、自動アドレス設定機能が有効な場合の制御を説明します。

#### <自動アドレス設定機能を無効にする処理>

イベント／処理	自動アドレス設定機能＝有効	自動アドレス設定機能＝無効
① 自動アドレス設定機能を無効にする	—	f/w が D_ModeControl.SetAddressMode="1"とセットする。

#### ①自動アドレス設定機能を無効にする

D\_ModeControl.SetAddressMode ="1"と設定する。

この設定処理は、リセット解除後に一度行うことで、それ以降は設定する必要はありません。

#### <SetAddress リクエスト処理>

イベント／処理	自動アドレス設定機能＝有効	自動アドレス設定機能＝無効
① SetAddress リクエストを受信	—	h/w が RcvEP0SETUP 割り込みステータスを発行する
② リクエストの確認	—	f/w が EP0SETUP0, EP0SETUP1 により確認する
③ アドレス設定指示	—	f/w が D_USB_Address.SetAddress="1"とセットする
④ ステータスステージ応答準備	—	f/w が下記設定を行う D_SETUP_Control.ProtectEP0="0" D_EP0Control.INxOUT="1" D_EP0Control.IN="0x40"※ ※ForceNAK="0", EnShortPkt="1"
⑤ ステータスステージが行われる	h/w が SetAddressCmp 割り込みステータスを発行する	h/w が SetAddressCmp 割り込みステータスを発行する

### ①SetAddress リクエストを受信

h/w がリクエストを受信し、RcvEP0SETUP 割り込みステータスを発行します。

自動アドレス設定機能を無効化したことにより、SetAddress リクエストにおいても、他のリクエストと同様に、本ステータスによって SETUP トランザクションの受領を通知します。

### ②リクエストの確認

f/w が D\_EP0SETUP\_0,1 レジスタの内容により、bmRequestType, bRequest を確認します。

bmRequest==0, bRequest==0x05 であれば、SetAddress リクエストです。

### ③アドレス設定指示

f/w が USB\_Address.SetAddress="1"とセットします。

この設定により、この後にステータスステージが行われ完了すると、SetAddress リクエストで指示されたアドレスを、h/w が D\_USB\_Address レジスタに上書きします。また、その処理が完了したことを、SetAddressCmp 割り込みステータスで通知します。

### ④ステータスステージ応答準備

他のリクエストにおける、IN 方向のステータスステージと同じく、ゼロ長パケットを返信する処理をします。

- D\_SETUP\_Control.ProtectEP0="0"

- D\_EP0Control.INxOUT="1"

- D\_EP0Control.IN="0x40"(ForceNAK="0", EnShortPkt="1")

### ⑤ステータスステージが行われる

ステータスステージ(IN トランザクション)が行われると、h/w が SetAddressCmp 割り込みステータスを発行します。

## 改訂履歴(Rev.1.00)

年月日	改訂内容			
	Rev.	項(旧版)	種別	内 容
06/10/30	0.90	全項	新規	新規制定
06/12/27	0.91	1.6.5.6	修正	誤)各エンドポイントの <b>EnEndpoint</b> がクリアされている場合 正)各エンドポイントが <b>FIFO</b> 領域にジョインされていない場合
		2.7.24	修正	<b>Test_Packet</b> ビットの説明中の誤記訂正 誤)エンドポイントの設定を、上記 <b>EPx{x=a-e}</b> の設定と重複しないようにして下さい。 正)他のエンドポイントの設定を、上記 <b>EPx{x=a-e}</b> の設定と重複しないようにして下さい。 誤) <b>EPx{x=a-e}Config.EnEndpoint</b> ビットをクリアして下さい。 正) <b>AREAx{x=0-5}Join_1</b> レジスタの各ビットをクリアして下さい。
		2.7.54 2.7.55 2.7.56 2.7.57 2.7.58	修正	レジスタビットの誤定義を訂正 <b>Bit 5</b> が <b>EnEndpoint</b> ビットと定義されていたが、無定義ビットに訂正。それに伴い、説明を削除。 <b>Bit 4</b> が無定義ビットとされていたが、 <b>ISO</b> ビットに訂正。それに伴い、説明を追加。
		2.8.101 2.8.102 2.8.103 2.8.104 2.8.105 2.8.106 2.8.107 2.8.108	修正	<b>Interval</b> ビットの説明中の誤記修正 誤) <b>TranType</b> ビットが"11"(インタラプト転送)の場合のみ有効です。 正) <b>TranType</b> ビットが"11"(インタラプト転送)の場合、または、 <b>TranType</b> ビットが"00" (アイソクロナス転送)の場合に有効です。
		Appendix D	追記	「TS のみ適用」と付記した。 <b>Appendix D</b> は、TS 固有の問題を回避する方法を説明するものであるため。
07/06/05	1.00	1.2.1	追記	表 1-5、マックスパケットサイズの説明に、下記を追加。 「また、アイソクロナス転送を行うエンドポイントでは、 <b>FS</b> モード時は 1～1023Byte のいずれか、 <b>HS</b> モード時は 1～1024Byte のいずれかに設定します。」
		1.2.6	修正	バルクオンリーサポート機能を行えるエンドポイントに <b>EPa</b> を追加。
		1.5.1 1.5.2	修正	<b>XINT</b> の発生条件。 誤) この時、 <b>MainIntEnb.EnFinishedPM</b> ビットをセットし、かつ <b>MainIntEnb.EnSIE_IntStat</b> ビットをセットしていれば、割り込み <b>XINT</b> が発生します。 正) この時、 <b>MainIntEnb.EnFinishedPM</b> ビットをセットしていれば、割り込み <b>XINT</b> が発生します。
		1.5.1 1.5.2	修正	誤) <b>SIE_IntStat.FinishedPM</b> 正) <b>MainIntStat.FinishedPM</b>

		1.6.3.1	修正	以下捕捉した。 前)「こうすることにより、エンドポイントの FIFO にはデータだけを受信することが出来ます。」 後)「こうすることにより、エンドポイントの FIFO では、データトランスポートで受信したデータのみを扱うことができます。」
		1.6.3.2	追記	以下捕捉した。 「こうすることにより、チャンネルの FIFO では、データトランスポートで受信したデータのみを扱うことができます。」
		1.6.4.1	修正	以下捕捉した。 前)「こうすることにより、エンドポイントの FIFO からはデータだけを送信することが出来ます。」 後)「こうすることにより、エンドポイントの FIFO では、データトランスポートで送信するデータのみを扱うことができます。」
		1.6.4.2	修正	以下捕捉した。 前)「こうすることにより、チャンネルの FIFO にはデータだけを受信することが出来ます。」 後)「こうすることにより、チャンネルの FIFO では、データトランスポートで受信したデータのみを扱うことができます。」
		1.6.5	追加	捕捉追記。 「RAM アクセスによって、FIFO の任意の場所からデータを読むことができ、また、任意の場所にデータを書くことができます。RAM アクセスによっては、FIFO のデータ数は更新されません。つまり、RAM_Rd 機能によって FIFO のデータを読み出しても、FIFO 領域は解放されません。また、RAM_WrDoor 機能によって FIFO にデータを書き込んでも、FIFO 領域を占有しません。」
		1.6.5.6	修正	下記制限を削除。 <ul style="list-style-type: none"> <li>FIFO の他のエリアであっても、JoinCPU_Rd と JoinCPU_Wr とは、同時には設定できない。(FIFO と MediaFIFO とには可能)</li> </ul>
		1.7.2.1 1.7.2.2 1.7.2.3	修正	誤) MediaFIFOJoin 正) MediaFIFO_Join
		2.7.44-53	修正	MaxSize のビット幅。 誤) MaxSize[9:3] 正) MaxSize[10:0] MaxSize ビットの説明中、インタラプト転送時の設定値。 誤) HS 時 512 バイトまで 正) HS 時 1024 バイトまで MaxSize ビットの説明に、アイソクロナス転送時の設定値を追記。

		2	修正	<p>重複する記載をひとつにまとめた。</p> <p>修正前)</p> <p>レジスタ名称の末尾が下記の場合は、リトルエンディアンモード(CPU_Endian=1)にすると、アドレスの偶数・奇数が入れ替わります。</p> <p>_H _M _L _HH _HL _LH _LL</p> <p>修正後)</p> <p>本LSIのレジスタマップはビッグエンディアンで定義されています。リトルエンディアンCPUとの接続については、「Appendix B リトルエンディアンのCPUへの接続」を参照して下さい。</p>
		2.8.65-68 2.8.101-108	修正	<p>TranType ビットの定義誤記を修正した。</p> <p>誤) 00 Isochronous, 01 Reserved 正) 00 Reserved, 01 Isochronous</p>
		1.8.2	追記	<p>下記を追記した。</p> <p>CPUIF のモード設定は、ハードリセット後に必ず行う必要があります。また、ハードリセット後の未初期化期間でのみ行えます。</p>
		2.6.14	修正	<p>下記を修正した。</p> <p>誤)本 LSI の全回路をリセットします。外部リセット端子(XRST)と同義です。</p> <p>正) CPUIF_MODE レジスタを除く全回路を初期化します。本レジスタによるソフトリセットでは、未初期化期間には遷移しません。</p>
		2.6.37 2.6.38	修正	<p>削除) RAM_WrAdrs を参照する場合は、RAM_WrAdrs_H,RAM_WrAdrs_L の順に読み出してください。</p>
		2.6.104	修正	<p>誤)CPU の IDE バスへの 正)IDE バスへの</p>
		2.6.105	修正	<p>誤)CPU 及び DMA 転送時の IDE バスへの 正)IDE バスへの</p>
		2.6.106	修正	<p>誤)DMA 転送時の IDE バスへの 正)IDE バスへの</p>
		2.6.107 2.6.108	修正	<p>削除) その際に IDE_CRC_H レジスタを先にアクセスして下さい。</p>
		2.6.112	修正	<p>Bit3-0 の説明中、 誤)CPU の IDE バスへの 正)IDE バスへの</p>
		2.6.113 2.6.114	修正	<p>誤)CPU の IDE バスに対する 正)IDE バスに対する</p>
		2.6.113 2.6.114	修正	<p>削除) その際に IDE_RdRegValue_0 レジスタを先にアクセスして下さい。</p>
		2.6.115 2.6.116	修正	<p>誤)CPU の IDE バスに対する 正)IDE バスに対する</p>



	2.6.143 ~ 2.6.148	修正	誤) FIFO_Rd_0,1、FIFO_ByteRd、FIFO_Wr_0,1 レジスタからデータを読み出し、または書き込みできます。 正) FIFO_Rd_0,1、FIFO_ByteRd、FIFO_Wr_0,1、FIFO_ByteWr レジスタからデータを読み出し、または書き込みできます。
	2.7.3	修正	誤)CBW_Comp 正)CBW_Cmp
	2.7.38 2.7.39	修正	削除) その際に FrameNumber_H レジスタを先にアクセスして下さい。
	2.7.64 2.7.65	修正	削除) Descriptor Address を参照する場合は、D_DescAdrs_H,D_DescAdrs_L の順に読み出して下さい。
	2.7.66 2.7.67	修正	削除) Descriptor Size を参照する場合は、DescSize_H,DescSize_L の順に読み出して下さい。
	2.8.27~ 2.8.34	修正	誤) BmRequestType 正) bmRequestType 誤) Brequest 正) bRequest 誤) Wvalue 正) wValue 誤) Windex 正) wIndex 誤) WLength 正) wLengh
	2.8.35 2.8.36	修正	削除) その際に H_FrameNumber_H レジスタを先にアクセスして下さい。
	2.8.67 2.8.68	修正	誤) データパケットのサイズ 正) 最後のデータパケットのサイズ



**セイコーエプソン株式会社**  
半導体事業部 IC 営業部

---

<IC 国内営業グループ>

東京 〒191-8501 東京都日野市日野 421-8  
TEL (042) 587-5313 (直通) FAX (042) 587-5116

大阪 〒541-0059 大阪市中央区博労町 3-5-1 エプソン大阪ビル 15F  
TEL (06) 6120-6000 (代表) FAX (06) 6120-6100

---

ドキュメントコード : 411140101  
2007 年 6 月 作成