

S1R72V03

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件の許諾を行うものではありません。
3. 特性値のの数値の大小は、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、一般民生用です。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。

総則

適用範囲

本仕様書は、セイコーエプソン株式会社 半導体事業部製 USB2.0 コントローラ「S1R720V03」に適用します。

目 次

1.	概要	1
2.	特長	1
3.	ブロック図	2
3.1.	USB Transceiver Macro (UTM)	3
3.2.	PLL60	3
3.3.	Serial Interface Engine (SIE)	3
3.4.	FIFO / FIFO Controller	3
3.5.	CPU I/F Controller	3
3.6.	DMA Controller	3
3.7.	IDE Master Controller	3
3.8.	TestMUX	3
4.	端子配置図	4
4.1.	端子配置図(QFPパッケージ)	4
4.2.	端子配置図(BGAパッケージ)	5
5.	端子機能説明	6
6.	機能説明	10
6.1.	USBデバイス制御	10
6.1.1.	エンドポイント	10
6.1.2.	トランザクション	11
6.1.2.1.	SETUPトランザクション	13
6.1.2.2.	バルク／インタラプトOUTトランザクション	14
6.1.2.3.	バルク／インタラプトINトランザクション	16
6.1.2.4.	PINGトランザクション	16
6.1.3.	コントロール転送	17
6.1.3.1.	セットアップステージ	18
6.1.3.2.	データステージ／ステータスステージ	18
6.1.3.3.	自動アドレス設定機能	18
6.1.3.4.	デスクリプタ返信機能	19
6.1.4.	バルク転送／インタラプト転送	19
6.1.5.	データフロー	19
6.1.5.1.	OUT転送	19
6.1.5.2.	IN転送	19
6.1.6.	バルクオンリーサポート	20
6.1.6.1.	CBWサポート	20
6.1.6.2.	CSWサポート	21
6.1.7.	オート・ネゴシエーション機能	22
6.1.7.1.	DISABLE	22
6.1.7.2.	IDLE	24
6.1.7.3.	WAIT_TIM3US	24
6.1.7.4.	WAIT_CHIRP	24
6.1.7.5.	WAIT_RSTEND	24
6.1.7.6.	DET_SUSPEND	24
6.1.7.7.	IN_SUSPEND	24
6.1.7.8.	CHK_EVENT	24
6.1.7.9.	WAIT_RESTORE	25
6.1.7.10.	ERR	25
6.1.7.11.	各ネゴシエーション機能の単体説明	26
6.1.7.11.1.	サスペンド検出 (HSモード)	26

6.1.7.11.2.	サスペンド検出 (FSモード)	28
6.1.7.11.3.	リセット検出 (HSモード)	29
6.1.7.11.4.	リセット検出 (FSモード)	30
6.1.7.11.5.	HS Detection Handshake	31
6.1.7.11.5.1.	FSのダウンストリームポートに繋がれた場合	32
6.1.7.11.5.2.	HSのダウンストリームポートに繋がれた場合	34
6.1.7.11.5.3.	スヌーズ中にリセットされた場合	37
6.1.7.11.6.	レジュームの発行	39
6.1.7.11.7.	レジュームの検出	41
6.1.7.11.8.	ケーブル挿入	43
6.2.	パワーマネジメント機能	45
6.2.1.	SLEEP (スリープ)	45
6.2.2.	SNOOZE (スヌーズ)	46
6.2.3.	ACTIVE60 (アクティブ 60)	47
6.2.4.	ACTIVEALL (アクティブ 480)	47
6.3.	FIFO管理	48
6.3.1.	FIFO管理	48
6.3.1.1.	FIFOメモリマップ	48
6.3.1.2.	デスクリプタエリアの使用方法	49
6.3.1.2.1.	デスクリプタエリアへのデータの書き込み	49
6.3.1.2.2.	デスクリプタエリアでのデータステージ (IN) の実行	49
6.3.1.3.	CBWエリアの使用方法	49
6.3.1.3.1.	CBWエリアへの受信	49
6.3.1.3.2.	CBWエリアからのデータの読み出し	49
6.3.1.4.	CSWエリアの使用方法	49
6.3.1.4.1.	CSWエリアからの送信	49
6.3.1.4.2.	CSWエリアへのデータの書き込み	50
6.3.2.	FIFOへのアクセス方法	50
6.3.2.1.	FIFOへのアクセス方法 (RAM_Rd)	50
6.3.2.2.	FIFOへのアクセス方法 (RAM_WrDoor)	50
6.3.2.3.	FIFOへのアクセス方法 (レジスタアクセス)	50
6.3.2.4.	FIFOへのアクセス方法 (DMA)	50
6.3.2.5.	FIFOへのアクセス方法 (IDE)	51
6.3.2.6.	FIFOへのアクセス制限	51
6.4.	CPUIF	52
6.4.1.	モード切り替え	52
6.4.2.	モード切り替えに際して	52
6.4.2.1.	16bit BE modeをご使用の場合	52
6.4.2.2.	8bit modeをご使用の場合	54
6.4.3.	ブロック構成	54
6.4.3.1.	REG (S1R72V03 Registers)	54
6.4.3.1.1.	同期レジスタアクセス (ライト)	55
6.4.3.1.2.	同期レジスタアクセス (リード)	55
6.4.3.1.3.	FIFOアクセス (ライト)	55
6.4.3.1.4.	FIFOアクセス (リード)	55
6.4.3.1.5.	FIFOアクセスの端数処理	55
6.4.3.1.6.	RAM_Rdアクセス	58
6.4.3.1.7.	非同期レジスタアクセス (ライト)	58
6.4.3.1.8.	非同期レジスタアクセス (リード)	58
6.4.3.2.	DMA0/DMA1 (DMA ch.0 / ch.1)	58

6.4.3.2.1.	基本機能	58
6.4.3.2.2.	端子設定	59
6.4.3.2.3.	アクセスモード設定	59
6.4.3.2.4.	カウントモード（ライト）	60
6.4.3.2.5.	カウントモード（リード）	62
6.4.3.2.6.	フリーランモード（ライト）	64
6.4.3.2.7.	フリーランモード（リード）	64
6.4.3.2.8.	REQアサートカウントオプション（ライト）	64
6.4.3.2.9.	REQアサートカウントオプション（リード）	65
6.4.3.2.10.	DMAのFIFOアクセス端数処理	65
6.5.	IDE I/F	66
6.5.1.	IDEタスクファイルレジスタへのアクセス	66
6.5.1.1.	IDEタスクファイルレジスタからのリード	66
6.5.1.2.	IDEタスクファイルレジスタへのライト	66
6.5.1.3.	IDEタスクファイルレジスタへのシーケンシャルライト	66
6.5.1.4.	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	66
6.5.2.	PIOアクセス	67
6.5.2.1.	PIOリードDMA	67
6.5.2.2.	PIOライトDMA	67
6.5.3.	Multi-Word DMA	67
6.5.3.1.	Multi-Word DMAリード	67
6.5.3.2.	Multi-Word DMAライト	67
6.5.4.	Ultra DMA	69
6.5.4.1.	Ultra DMAリード	69
6.5.4.2.	Ultra DMAライト	69
6.5.5.	IDE転送モードの設定について	70
6.6.	バウンダリスキャン(JTAG)	71
6.6.1.	対応インストラクション	71
6.6.2.	DEVICE_CODEに関して	71
6.6.3.	バウンダリスキャン除外端子	71
7.	レジスタ	72
7.1.	レジスタマップ	72
7.2.	レジスタ詳細説明	80
7.2.1.	00h <i>MainIntStat (Main Interrupt Status)</i>	80
7.2.2.	01h <i>EPrIntStat (EPr Interrupt Status)</i>	82
7.2.3.	02h <i>SIE_IntStat (SIE Interrupt Status)</i>	83
7.2.4.	03h <i>CPU_IntStat (CPU Interrupt Status)</i>	86
7.2.5.	04h <i>FIFO_IntStat (FIFO Interrupt Status)</i>	87
7.2.6.	05h <i>BulkIntStat (Bulk Interrupt Status)</i>	88
7.2.7.	06h <i>IDE_IntStat (IDE Interrupt Status)</i>	89
7.2.8.	07h <i>DBG_IntStat (DEBUG Interrupt Status)</i>	90
7.2.9.	08h <i>EP0IntStat (EP0 Interrupt Status)</i>	91
7.2.10.	09h <i>EPaIntStat (EPa Interrupt Status)</i>	92
7.2.11.	0Ah <i>EPbIntStat (EPb Interrupt Status)</i>	93
7.2.12.	0Bh <i>EPcIntStat (EPc Interrupt Status)</i>	94
7.2.13.	0Ch~0Fh Reserved ()	95
7.2.14.	10h <i>MainIntEnb (Main Interrupt Enable)</i>	96
7.2.15.	11h <i>EPrIntEnb (EPr Interrupt Enable)</i>	97
7.2.16.	12h <i>SIE_IntEnb (SIE Interrupt Enable)</i>	98
7.2.17.	13h <i>CPU_IntEnb (CPU Interrupt Enable)</i>	99

7.2.18.	14h FIFO_IntEnb (FIFO Interrupt Enable)	100
7.2.19.	15h BulkIntEnb (Bulk Interrupt Enable)	101
7.2.20.	16h IDE_IntEnb (IDE Interrupt Enable)	102
7.2.21.	17h Reserved ()	103
7.2.22.	18h EP0IntEnb (EP0 Interrupt Enable)	104
7.2.23.	19h EPaIntEnb (EPa Interrupt Enable)	105
7.2.24.	1Ah EPbIntEnb (EPb Interrupt Enable)	106
7.2.25.	1Bh EPcIntEnb (EPc Interrupt Enable)	107
7.2.26.	1Ch~1Fh Reserved ()	108
7.2.27.	20h <i>RevisionNum</i> (<i>Revision Number</i>)	109
7.2.28.	21h <i>ChipReset</i> (<i>Chip Reset</i>)	110
7.2.29.	22h <i>PM_Control</i> (<i>Power Management Control</i>)	111
7.2.30.	23h <i>USB_Control</i> (<i>USB_Control</i>)	113
7.2.31.	24h <i>USB_Status</i> (<i>USB_Status</i>)	115
7.2.32.	25h <i>XcvrControl</i> (<i>Xcvr Control</i>)	116
7.2.33.	26h <i>USB_Test</i> (<i>USB_Test</i>)	117
7.2.34.	27h Reserved ()	119
7.2.35.	28h <i>EPnControl</i> (<i>Endpoint Control</i>)	120
7.2.36.	29h <i>EPnFIFO_Clr</i> (<i>Endpoint FIFO Clear</i>)	121
7.2.37.	2Ah <i>ClrAllJoin</i> (<i>Clear All Join</i>)	122
7.2.38.	2Bh Reserved ()	123
7.2.39.	2Ch <i>BulkOnlyControl</i> (<i>BulkOnly Control</i>)	124
7.2.40.	2Dh <i>BulkOnlyConfig</i> (<i>BulkOnly Configuration</i>)	125
7.2.41.	2Eh <i>WakeupTim_H</i> (<i>Wakeup Time High</i>)	126
7.2.42.	2Fh <i>WakeupTim_L</i> (<i>Wakeup Time Low</i>)	126
7.2.43.	30h <i>EPOSETUP_0</i> (<i>EP0 SETUP 0</i>)	127
7.2.44.	31h <i>EPOSETUP_1</i> (<i>EP0 SETUP 1</i>)	127
7.2.45.	32h <i>EPOSETUP_2</i> (<i>EP0 SETUP 2</i>)	127
7.2.46.	33h <i>EPOSETUP_3</i> (<i>EP0 SETUP 3</i>)	127
7.2.47.	34h <i>EPOSETUP_4</i> (<i>EP0 SETUP 4</i>)	127
7.2.48.	35h <i>EPOSETUP_5</i> (<i>EP0 SETUP 5</i>)	127
7.2.49.	36h <i>EPOSETUP_6</i> (<i>EP0 SETUP 6</i>)	127
7.2.50.	37h <i>EPOSETUP_7</i> (<i>EP0 SETUP 7</i>)	127
7.2.51.	38h <i>USB_Address</i> (<i>USB Address</i>)	128
7.2.52.	39h Reserved ()	129
7.2.53.	3Ah <i>SETUP_Control</i> (<i>SETUP Control</i>)	130
7.2.54.	3Bh~3Dh Reserved ()	131
7.2.55.	3Eh <i>FrameNumber_H</i> (<i>FrameNumber High</i>)	132
7.2.56.	3Fh <i>FrameNumber_L</i> (<i>FrameNumber Low</i>)	132
7.2.57.	40h <i>EP0MaxSize</i> (<i>EP0 Max Packet Size</i>)	133
7.2.58.	41h <i>EP0Control</i> (<i>EP0 Control</i>)	134
7.2.59.	42h <i>EP0ControlIN</i> (<i>EP0 Control IN</i>)	135
7.2.60.	43h <i>EP0ControlOUT</i> (<i>EP0 Control OUT</i>)	137
7.2.61.	44h Reserved ()	138
7.2.62.	45h <i>EP0Join</i> (<i>End Point0 Join</i>)	139
7.2.63.	46h~4Fh Reserved ()	140
7.2.64.	50h Reserved ()	141
7.2.65.	51h <i>EPaMaxSize_L</i> (<i>EPa Max Packet Size Low</i>)	141
7.2.66.	52h <i>EPaConfig_0</i> (<i>EPa Configuration 0</i>)	142
7.2.67.	53h Reserved ()	143

7.2.68.	54h EPaControl (EPa Control)	144
7.2.69.	55h EPaJoin (End Point a Join)	146
7.2.70.	56h~57h Reserved ()	148
7.2.71.	58h EPbMaxSize_H (EPb Max Packet Size High)	149
7.2.72.	59h EPbMaxSize_L (EPb Max Packet Size Low)	149
7.2.73.	5Ah EPbConfig_0 (EPb Configuration 0)	150
7.2.74.	5Bh Reserved ()	151
7.2.75.	5Ch EPbControl (EPb Control)	152
7.2.76.	5Dh EPbJoin (End Point b Join)	154
7.2.77.	5Eh~5Fh Reserved ()	156
7.2.78.	60h EPcMaxSize_H (EPc Max Packet Size High)	157
7.2.79.	61h EPcMaxSize_L (EPc Max Packet Size Low)	157
7.2.80.	62h EPcConfig_0 (EPc Configuration 0)	158
7.2.81.	63h Reserved ()	159
7.2.82.	64h EPcControl (EPc Control)	160
7.2.83.	65h EPcJoin (End Point c Join)	162
7.2.84.	66h~6Bh Reserved ()	164
7.2.85.	6Ch RAM_WrAdrs_H (RAM Write Address High)	165
7.2.86.	6Dh RAM_WrAdrs_L (RAM Write Address Low)	165
7.2.87.	6Eh RAM_WrDoor_H (RAM Write Door High)	166
7.2.88.	6Fh RAM_WrDoor_L (RAM Write Door Low)	166
7.2.89.	70h EPnFIFO_Rd_H (EPn FIFO Read High)	167
7.2.90.	71h EPnFIFO_Rd_L (EPn FIFO Read Low)	167
7.2.91.	72h EPnFIFO_Wr_H (EPn FIFO Write High)	168
7.2.92.	73h EPnFIFO_Wr_L (EPn FIFO Write Low)	168
7.2.93.	74h EPnRdRemain_H (EPn FIFO Read Remain High)	169
7.2.94.	75h EPnRdRemain_L (EPn FIFO Read Remain Low)	169
7.2.95.	76h EPnWrRemain_H (EPn FIFO Write Remain High)	170
7.2.96.	77h EPnWrRemain_L (EPn FIFO Write Remain Low)	170
7.2.97.	78h DescAdrs_H (Descriptor Address High)	171
7.2.98.	79h DescAdrs_L (Descriptor Address Low)	171
7.2.99.	7Ah DescSize_H (Descriptor Size High)	172
7.2.100.	7Bh DescSize_L (Descriptor Size Low)	172
7.2.101.	7Ch EPnFIFO_ByteRd (EPn FIFO Byte Read)	173
7.2.102.	7Dh~7Fh Reserved ()	174
7.2.103.	80h DMA0_FIFO_Control (DMA0 FIFO Control)	175
7.2.104.	81h DMA0_Config (DMA0 Config)	176
7.2.105.	82h DMA0_Control (DMA0 Control)	177
7.2.106.	83h Reserved ()	178
7.2.107.	84h DMA0_Remain_H (DMA0 FIFO Remain High)	179
7.2.108.	85h DMA0_Remain_L (DMA0 FIFO Remain Low)	179
7.2.109.	86h~87h Reserved ()	180
7.2.110.	88h DMA0_Count_HH (DMA0 Transfer Byte Counter High/High)	181
7.2.111.	89h DMA0_Count_HL (DMA0 Transfer Byte Counter High/Low)	181
7.2.112.	8Ah DMA0_Count_LH (DMA0 Transfer Byte Counter Low/High)	181
7.2.113.	8Bh DMA0_Count_LL (DMA0 Transfer Byte Counter Low/Low)	181
7.2.114.	8Ch DMA0_RdData_H (DMA0 Read Data High)	182
7.2.115.	8Dh DMA0_RdData_L (DMA0 Read Data Low)	182
7.2.116.	8Eh DMA0_WrData_H (DMA0 Write Data High)	183
7.2.117.	8Fh DMA0_WrData_L (DMA0 Write Data Low)	183

7.2.118.	90h DMA1_FIFO_Control (DMA1 FIFO Control)	184
7.2.119.	91h DMA1_Config (DMA0 Config)	185
7.2.120.	92h DMA1_Control (DMA1 Control)	186
7.2.121.	93h Reserved ()	187
7.2.122.	94h DMA1_Remain_H (DMA1 FIFO Remain High)	188
7.2.123.	95h DMA1_Remain_L (DMA1 FIFO Remain Low)	188
7.2.124.	96h~97h Reserved ()	189
7.2.125.	98h DMA1_Count_HH (DMA1 Transfer Byte Counter High/High)	190
7.2.126.	99h DMA1_Count_HL (DMA1 Transfer Byte Counter High/Low)	190
7.2.127.	9Ah DMA1_Count_LH (DMA1 Transfer Byte Counter Low/High)	190
7.2.128.	9Bh DMA1_Count_LL (DMA1 Transfer Byte Counter Low/Low)	190
7.2.129.	9Ch DMA1_RdData_H (DMA1 Read Data High)	192
7.2.130.	9Dh DMA1_RdData_L (DMA1 Read Data Low)	192
7.2.131.	9Eh DMA1_WrData_H (DMA1 Write Data High)	193
7.2.132.	9Fh DMA1_WrData_L (DMA1 Write Data Low)	193
7.2.133.	A0h IDE_Status (IDE Status)	194
7.2.134.	A1h IDE_Control (IDE Control)	195
7.2.135.	A2h IDE_Config_0 (IDE Configuration 0)	196
7.2.136.	A3h IDE_Config_1 (IDE Configuration 1)	197
7.2.137.	A4h IDE_Rmod (IDE Register Mode)	198
7.2.138.	A5h IDE_Tmod (IDE Transfer Mode)	199
7.2.139.	A6h IDE_Umod (IDE Ultra-DMA Transfer Mode)	200
7.2.140.	A7h~A9h Reserved ()	201
7.2.141.	AAh IDE_CRC_H (IDE CRC High)	202
7.2.142.	ABh IDE_CRC_L (IDE CRC Low)	202
7.2.143.	ACH Reserved ()	203
7.2.144.	ADh IDE_Count_H (IDE Transfer Byte Counter High)	204
7.2.145.	A Eh IDE_Count_M (IDE Transfer Byte Counter Middle)	204
7.2.146.	AFh IDE_Count_L (IDE Transfer Byte Counter Low)	204
7.2.147.	B0h IDE_RegAdrs (IDE Register Address)	205
7.2.148.	B1h Reserved ()	206
7.2.149.	B2h IDE_RdRegValue_H (IDE Register Read Value High)	207
7.2.150.	B3h IDE_RdRegValue_L (IDE Register Read Value Low)	207
7.2.151.	B4h IDE_WrRegValue_H (IDE Register Write Value High)	208
7.2.152.	B5h IDE_WrRegValue_L (IDE Register Write Value Low)	208
7.2.153.	B6h IDE_SeqWrRegControl (IDE Sequential Register Write Control)	209
7.2.154.	B7h IDE_SeqWrRegCnt (IDE Sequential Register Write Counter)	210
7.2.155.	B8h IDE_SeqWrRegAdrs (IDE Sequential Register Write Address FIFO)	211
7.2.156.	B9h IDE_SeqWrRegValue (IDE Sequential Register Write Value FIFO)	212
7.2.157.	BAh~BBh Reserved ()	213
7.2.158.	BCh IDE_RegConfig (IDE Register Configuration)	214
7.2.159.	BDh~BFh Reserved ()	215
7.2.160.	C0h RAM_Rd_00 (RAM Read 00)	216
7.2.161.	C1h RAM_Rd_01 (RAM Read 01)	216
7.2.162.	C2h RAM_Rd_02 (RAM Read 02)	216
7.2.163.	C3h RAM_Rd_03 (RAM Read 03)	216
7.2.164.	C4h RAM_Rd_04 (RAM Read 04)	216
7.2.165.	C5h RAM_Rd_05 (RAM Read 05)	216
7.2.166.	C6h RAM_Rd_06 (RAM Read 06)	216
7.2.167.	C7h RAM_Rd_07 (RAM Read 07)	216

7. 2. 168.	C8h RAM_Rd_08 (RAM Read 08)	216
7. 2. 169.	C9h RAM_Rd_09 (RAM Read 09)	216
7. 2. 170.	CAh RAM_Rd_0A (RAM Read 0A)	216
7. 2. 171.	CBh RAM_Rd_0B (RAM Read 0B)	216
7. 2. 172.	CCh RAM_Rd_0C (RAM Read 0C)	216
7. 2. 173.	CDh RAM_Rd_0D (RAM Read 0D)	216
7. 2. 174.	CEh RAM_Rd_0E (RAM Read 0E)	216
7. 2. 175.	CFh RAM_Rd_0F (RAM Read 0F)	216
7. 2. 176.	D0h RAM_Rd_10 (RAM Read 10)	216
7. 2. 177.	D1h RAM_Rd_11 (RAM Read 11)	216
7. 2. 178.	D2h RAM_Rd_12 (RAM Read 12)	216
7. 2. 179.	D3h RAM_Rd_13 (RAM Read 13)	216
7. 2. 180.	D4h RAM_Rd_14 (RAM Read 14)	216
7. 2. 181.	D5h RAM_Rd_15 (RAM Read 15)	216
7. 2. 182.	D6h RAM_Rd_16 (RAM Read 16)	216
7. 2. 183.	D7h RAM_Rd_17 (RAM Read 17)	216
7. 2. 184.	D8h RAM_Rd_18 (RAM Read 18)	216
7. 2. 185.	D9h RAM_Rd_19 (RAM Read 19)	216
7. 2. 186.	DAh RAM_Rd_1A (RAM Read 1A)	216
7. 2. 187.	DBh RAM_Rd_1B (RAM Read 1B)	216
7. 2. 188.	DCh RAM_Rd_1C (RAM Read 1C)	216
7. 2. 189.	DDh RAM_Rd_1D (RAM Read 1D)	216
7. 2. 190.	DEh RAM_Rd_1E (RAM Read 1E)	216
7. 2. 191.	DFh RAM_Rd_1F (RAM Read 1F)	216
7. 2. 192.	E0h RAM_RdAdrs_H (RAM Read Address High)	217
7. 2. 193.	E1h RAM_RdAdrs_L (RAM Read Address Low)	217
7. 2. 194.	E2h RAM_RdControl (RAM Read Control)	218
7. 2. 195.	E3h RAM_RdCount (RAM Read Counter)	219
7. 2. 196.	E4h~EAh Reserved ()	220
7. 2. 197.	EBh <i>ModeProtect (Mode Protection)</i>	221
7. 2. 198.	ECh Reserved ()	222
7. 2. 199.	EDh <i>ClkSelect (Clock Select)</i>	223
7. 2. 200.	EEh Reserved ()	224
7. 2. 201.	EFh <i>ChipConfig (Chip Configuration)</i>	225
7. 2. 202.	F0h Reserved ()	226
7. 2. 203.	F1h D_ModeControl (Device Mode Control)	226
7. 2. 204.	F2h~FFh Reserved ()	226
8.	電気的特性	227
8. 1.	絶対最大定格	227
8. 2.	推奨動作条件	227
8. 3.	DC特性	228
8. 4.	AC特性	232
8. 4. 1.	RESETタイミング	232
8. 4. 2.	クロックタイミング	232
8. 4. 3.	CPU/DMA I/Fアクセスタイミング	233
8. 4. 4.	IDE I/Fタイミング	235
8. 4. 4. 1.	PIO Read Timing	235
8. 4. 4. 2.	PIO Write Timing	236
8. 4. 4. 3.	DMA Read Timing	237
8. 4. 4. 4.	DMA Write Timing	238

8.4.4.5. Ultra DMA Read Timing.....	239
8.4.4.6. Ultra DMA Write Timing.....	241
8.4.5. USB I/Fタイミング.....	242
9. 接続例.....	243
9.1. CPU I/F接続例.....	243
9.2. USB I/F接続例.....	244
9.2.1. QFP15-128 の場合	244
9.2.2. PFBGA7UX100 の場合.....	245
10. 外形寸法図.....	246
10.1. QFPパッケージ.....	246
10.2. BGAパッケージ.....	247
Appendix A IDE_Config_1.Swapビット設定	248
Appendix B リトルエンディアンのCPUへの接続	249
Appendix C SetAddressリクエストの応答について	250
改訂履歴.....	252

1. 概要

S1R72V03**は USB2.0 準拠のハイスピードモードに対応する USB デバイスコントローラ LSI です。

IDE I/F を備え、HDD を内蔵するポータブル機器に適した特長を持ちます。

2. 特長

《 USB2.0 デバイス機能 》

- HS(480Mbps)及び FS(12Mbps)転送サポート
- FS/HS ターミネーション内蔵 (外付け回路不要)
- VBUS 5V 入力
- コントロール、バルク及びインタラプト転送をサポート
- Bulk 転送用 2 本、Interrupt転送用 1 本、及び、Endpoint 0 をサポート
- Bulk 転送用に各 1024B、Interrupt 転送用に 64B、及び、Control 転送用に 64B のFIFO領域を確保

《 CPU I/F 》

- 16bit 又は 8bit 幅の、汎用 CPU I/F に対応
- DMA 2ch. 搭載 (Multi-word 手順)
- Big Endian(Little Endian の CPU に対応するためのバススワップ機能を搭載)
- I/F 電圧変更可能(3.3V or 1.8V)

《 IDE I/F 》

- ATA/ATAPI6 に対応
- PIO モード 0~4, Multi word DMA, UDMA モード 0~5

《 その他 》

- クロック入力 12MHz,24MHz、水晶発振子対応。(発振回路及び帰還抵抗1MΩ内蔵)
- 電源電圧 3.3V と 1.8V、及び CPU I/F 電源の3電圧系統
- パッケージタイプ QFP15-128 , PFBGA7UX100
- 動作保証温度範囲 -40℃~85℃

※耐放射設計はされておりません

3. ブロック図

3. ブロック図

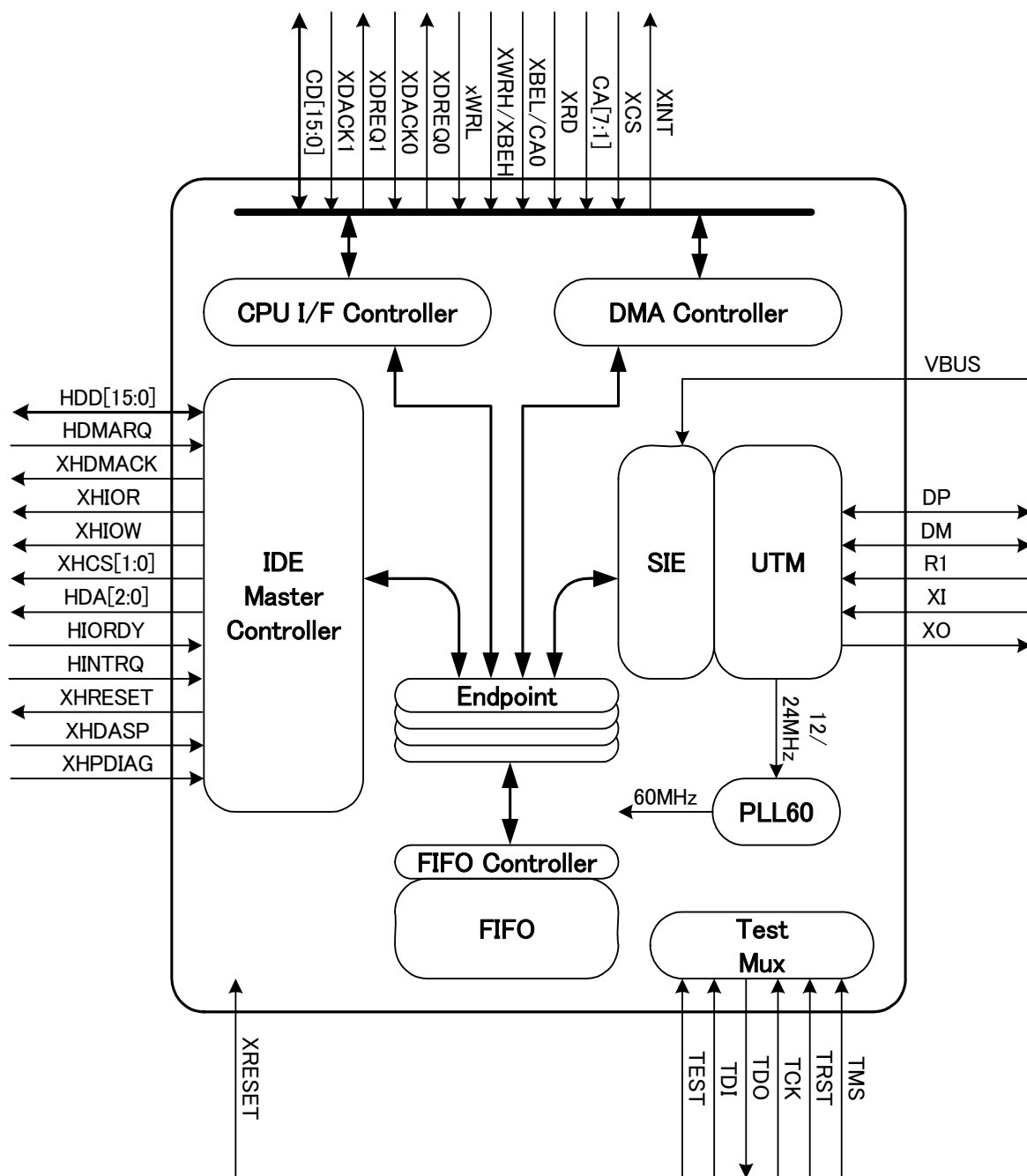


図 3-1 全体ブロック図

3.1. USB Transceiver Macro (UTM)

UTMI1.03 仕様準拠の USB2.0 トランシーバマクロです。アナログ回路及び高速ロジック回路を持ち、HS モード (480Mbps)、FS モード (12Mbps) をサポートします。

HS/FS に対応するトランスミッタ、レシーバ、ターミネーション、スケルチ回路や、エラステシティバッファ、シリパラ/パラシリ変換回路、ビットスタッフ/アンスタッフ回路、SYNC/EOP の付加/除去回路等を内蔵し、USB インタフェイスを実現します。

また、内蔵する PLL で USB 転送に必要な 480MHz クロックを生成します。**PLL は Oscillator を発振源とします。**

3.2. PLL60

内部ロジック動作に必要な 60MHz クロックを生成します。発振回路の入力クロックは 12/24MHz の水晶発振子に対応可能です。

3.3. Serial Interface Engine (SIE)

トランザクションの管理と、パケット生成を行います。

また、サスペンド・リジューム・リセット等の、バスイベントを制御します。

3.4. FIFO / FIFO Controller

エンドポイント用バッファ (2.5kB) です。

3.5. CPU I/F Controller

CPU I/F のタイミングを制御し、レジスタアクセスを可能にします。

3.6. DMA Controller

CPU I/F の DMA タイミングを制御し、FIFO アクセスを可能にします。2 チャンネルの DMA を搭載します。

3.7. IDE Master Controller

ATA/ATAPI6 に対応する IDE インタフェイスです。

PIO モード 0~4, Multi word DMA, UDMA モード 0~5 に対応します。

3.8. TestMUX

テスト回路です。

4.2. 端子配置図 (BGA パッケージ)

PFBGA7UX100

	1	2	3	4	5	6	7	8	9	10	
A	NC	R1	PVDD	XO	XI	LVDD	LVDD	HVDD	HDA0	NC	A
B	HVDD	VSS	PVSS	LVDD	VSS	TCK	XHDASP	HDA2	HINTRQ	XHDMACK	B
C	DM	VSS	VSS	TRST	TMS	VSS	XHCS0	HDA1	XHIOW	HIORDY	C
D	DP	VBUS	XRESET	TDI	TDO	XHCS1	XHPDIAG	XHIOR	HDD15	HDD14	D
E	HVDD	TEST	XBEL	CA3	CD1	CD3	HDMAHQ	HDD1	HDD13	HVDD	E
F	CA2	CA1	CA6	CA5	CD0	VSS	HDD0	HDD2	HDD3	HDD12	F
G	CA7	CA4	XWRH	XDREQ0	CD5	CD10	CD11	HDD11	HDD4	HDD10	G
H	CVDD	XCS	XWRL	XDREQ1	CD6	CD9	CD14	HDD5	HDD9	HDD8	H
J	XINT	XRD	XDACK0	CD2	CD7	CD8	CD13	VSS	HDD6	XHRESET	J
K	NC	VSS	XDACK1	CD4	LVDD	CVDD	CD12	CD15	HDD7	NC	K
	1	2	3	4	5	6	7	8	9	10	

図 4-2 BGA パッケージ端子配置図(TOP View)

5. 端子機能説明

5. 端子機能説明

GENERAL						
Pin	Ball	名称	I/O	RESET	端子タイプ	端子説明
17	D3	XRESET	IN	-	-	リセット信号
123	A5	XI	IN	-	Analog	内部発振回路用入力 12MHz/24MHz
125	A4	XO	OUT	-	Analog	内部発振回路用出力

XRESET は CVDD 電源系に属します。

XI,XO は PVDD 電源系に属します。

TEST						
Pin	Ball	名称	I/O	RESET	端子タイプ	端子説明
16	E2	TEST	IN	-		テスト端子(Low 固定)
111	D5	TDO	OUT	Hi-Z	2mA	JTAG TDO 端子
112	B6	TCK	IN	-		JTAG TCK 端子
113	C5	TMS	IN	-		JTAG TMS 端子
114	D4	TDI	IN	-		JTAG TDI 端子
115	C4	TRST	IN	-		JTAG TRST 端子

TEST は CVDD 電源系に属します。

上記の他の端子は HVDD 電源系に属します。

JTAG 機能を使用しない場合は、TEST, TCK, TMS, TDI, TRST の各端子は Low 固定、TDO 端子はオープンとして処理してください。

PD: Pull Down, PU: Pull Up

USB						
Pin	Ball	名称	I/O	RESET	端子タイプ	端子説明
128	A2	R1	IN	-	Analog	内部動作設定端子 6.2k Ω \pm 1%の抵抗を VSS 間に接続
10	D1	DP	BI	Hi-Z	Analog	USB デバイス・データライン Data+
6	C1	DM	BI	Hi-Z	Analog	USB デバイス・データライン Data-
14	D2	VBUS	IN(PD)	(PD)	Analog	USB デバイス・バス検出信号

5. 端子機能説明

CPU I/F								
Pin	Ball	名称	I/O	RESET	端子タイプ	端子説明		
Bus Mode ⇒						16bit Strobe mode	16bit BE mode	8bit mode
30	J2	XRD	IN	-	-	リード・ストローブ		
32	H3	XWRL (XWR)	IN	-	-	ライト・ストローブ 下位	ライト・ストローブ	
31	G3	XWRH (XBEH)	IN	-	-	ライト・ストローブ 上位	ハイ・バイト イネーブル	High 固定
29	H2	XCS	IN	-	-	チップセレクト信号		
28	J1	XINT	OUT	High	2mA TriState	割り込み出力信号		
35	G4	XDREQ0	OUT	High	2mA	DMA0 リクエスト		
36	J3	XDACK0	IN	-	-	DMA0 アクノリッジ		
37	H4	XDREQ1	OUT	High	2mA	DMA1 リクエスト		
38	K3	XDACK1	IN	-	-	DMA1 アクノリッジ		
18	E3	XBEL (CA0)	IN	-	-	High or Low 固定	ロー・バイト イネーブル	アドレス 0
19	F2	CA1	IN	-	-	CPU バスアドレス		
20	F1	CA2	IN	-	-			
21	E4	CA3	IN	-	-			
22	G2	CA4	IN	-	-			
23	F4	CA5	IN	-	-			
24	F3	CA6	IN	-	-			
25	G1	CA7	IN	-	-			
39	F5	CD0	BI	Hi-Z	2mA	CPU データ・バス		
40	E5	CD1	BI	Hi-Z	2mA			
41	J4	CD2	BI	Hi-Z	2mA			
42	E6	CD3	BI	Hi-Z	2mA			
43	K4	CD4	BI	Hi-Z	2mA			
44	G5	CD5	BI	Hi-Z	2mA			
45	H5	CD6	BI	Hi-Z	2mA			
46	J5	CD7	BI	Hi-Z	2mA			
53	J6	CD8	BI	Hi-Z	2mA			
54	H6	CD9	BI	Hi-Z	2mA			
55	G6	CD10	BI	Hi-Z	2mA			
56	G7	CD11	BI	Hi-Z	2mA			
57	K7	CD12	BI	Hi-Z	2mA			
58	J7	CD13	BI	Hi-Z	2mA			
59	H7	CD14	BI	Hi-Z	2mA			
60	K8	CD15	BI	Hi-Z	2mA			

上記の端子は CVDD 電源系に属します。

PD: Pull Down

PU: Pull Up

5. 端子機能説明

IDE I/F						
Pin	Ball	名称	I/O	RESET	端子タイプ	端子説明
101	B8	HDA2	OUT	Hi-Z	4mA	IDE レジスタ・アドレス
98	C8	HDA1	OUT	Hi-Z	4mA	
100	A9	HDA0	OUT	Hi-Z	4mA	
103	D6	XHCS1	OUT	Hi-Z	4mA	コントロールレジスタアクセス用チップセレクト
102	C7	XHCS0	OUT	Hi-Z	4mA	コマンドブロックレジスタアクセス用チップセレクト
91	D8	XHIOR	OUT	Hi-Z	4mA	IDE リード・ストローブ
90	C9	XHIOW	OUT	Hi-Z	4mA	IDE ライト・ストローブ
89	E7	HDMARQ	IN (PD)	(PD)	-	DMA 転送要求
93	B10	XHDMACK	OUT	Hi-Z	4mA	DMA 転送許可
92	C10	HIORDY	IN (PU)	(PU)	-	IDE レジスタレディ信号
97	B9	HINTRQ	IN (PU)	(PU)	-	IDE 割り込み要求
66	J10	XHRESET	OUT	Hi-Z	4mA	IDE バスリセット
104	B7	XHDASP	IN (PU)	(PU)	-	ドライブ有効／スレーブドライブ有り
99	D7	XHPDIAG	IN (PU)	(PU)	-	診断シーケンス終了信号
88	D9	HDD15	BI (PU)	(PU)	4mA	IDE データ・バス
86	D10	HDD14	BI (PU)	(PU)	4mA	
84	E9	HDD13	BI (PU)	(PU)	4mA	
76	F10	HDD12	BI (PU)	(PU)	4mA	
74	G8	HDD11	BI (PU)	(PU)	4mA	
72	G10	HDD10	BI (PU)	(PU)	4mA	
70	H9	HDD9	BI (PU)	(PU)	4mA	
68	H10	HDD8	BI (PU)	(PU)	4mA	
67	K9	HDD7	BI (PD)	(PD)	4mA	
69	J9	HDD6	BI (PU)	(PU)	4mA	
71	H8	HDD5	BI (PU)	(PU)	4mA	
73	G9	HDD4	BI (PU)	(PU)	4mA	
75	F9	HDD3	BI (PU)	(PU)	4mA	
83	F8	HDD2	BI (PU)	(PU)	4mA	
85	E8	HDD1	BI (PU)	(PU)	4mA	
87	F7	HDD0	BI (PU)	(PU)	4mA	

上記の端子は HVDD 電源系に属します。

PD: Pull Down

PU: Pull Up

POWER				
Pin	Ball	名称	電圧	端子説明
4, 5, 12, 13, 81, 82, 105, 106	B1, E1, A8, E10	HVDD	3.3V	IDE, USB, TEST I/O 用電源
27, 51, 52	K6, H1	CVDD	3.3V/1.8V	CPU I/F I/O 用電源
47, 48, 77, 78, 109, 110, 118, 120	A7, A6, B4, K5	LVDD	1.8V	内部電源
126, 127	A3	PVDD	1.8V	PLL、オシレータ用電源
1, 2, 7, 8, 9, 15, 26, 33, 34, 49, 50, 61, 62, 63, 64, 65, 79, 80, 94, 95, 96, 107, 108, 116, 117, 119	B2, C2, K2, C3, B5, C6, F6, J8	VSS	0V	GND
121, 122	B3	PVSS	0V	PLL 用 GND
3, 11, 124	A1, K1, A10, K10	NC	0V	NC (GND に固定してください。)

6. 機能説明

6. 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.RcvEP0SETUP ビット」「EP0ControlOUT レジスタの ForceNAK ビット」

- ・エンドポイント毎にあるレジスタ

EPx{x=0,a-c}~レジスタ、EPx{x=a-c}~レジスタ等と標記します。

例:「EPx{x=0,a-c}IntStat レジスタ」「EPx{x=a-c}Control レジスタ」

- ・DMA チャンネル毎にあるレジスタ

DMAx{x=0,1}~レジスタ等と標記します。

例:「DMAx{x=0,1}Config レジスタ」「EPx{x=0,a-c}Join.JoinDMAx{x=0,1}_Wr ビット」

6.1. USB デバイス制御

ここでは、USB デバイス機能について説明します。

6.1.1. 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、3本の汎用エンドポイント(EPa, EPb, EPc)を持ちます。エンドポイント EPa, EPb, EPc は、それぞれに、バルクまたはインタラプト転送用のエンドポイントとして使用できます。

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립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下さい。

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 6-1 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。

表 6-1 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	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 には、FIFO の 0 番地から 64 バイトの領域が割り当てられます。

表 6-2 に汎用エンドポイント(EPa, EPb, EPc)の基本設定項目を示します。エンドポイントEPa, EPb, EPc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、3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。

エンドポイント EPa の FIFO 領域は 64Byte、エンドポイント EPb, EPc の FIFO 領域は 1024Byte がそれぞれ割り当てられています。

表 6-2 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	EPx{x=a-c}Config_0.INxOUT	各エンドポイントの転送方向を設定します。
マックスバケットサイズ	EPx{x=b-c}MaxSize_H, EPx{x=a-c}MaxSize_L	各エンドポイントのマックスバケットサイズを、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。(注:EPa のマックスバケットサイズは 64Byte までしか設定出来ません)
エンドポイントナンバー	EPx{x=a-c}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	EPx{x=a-c}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、"0"にセットして下さい。
エンドポイント有効	EPx{x=a-c}Config_0.EnEndpoint	各エンドポイントを有効にします。 そのエンドポイントを使用する USB 定義インタフェースが有効にされた時に設定して下さい。

6.1.2. トランザクション

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、レジスタ説明の章を参照して下さい。

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。

6. 機能説明

表 6-3 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 6-3 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	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	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	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	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	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	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	EP0ControlIN.ToggleStat, 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	EP0ControlIN.ToggleSet, 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	EP0ControlIN.ToggleClr, EP0ControlOUT.ToggleClr	トグルシーケンスビットをクリアします。
強制 NAK 応答	EP0ControlIN.ForceNAK, 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	EP0ControlIN.ForceSTALL, 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	EP0ControlOUT.AutoForceNAK	OUT トランザクションの完結毎に、EP0ControlOUT.ForceNAK ビットをセットします。
SETUP 受信ステータス	MainIntStat.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	EP0IntStat.OUT_ShortACK, EP0IntStat.IN_TrانACK, EP0IntStat.OUT_TrانACK, EP0IntStat.IN_TrانNAK, EP0IntStat.OUT_TrانNAK, EP0IntStat.IN_TrانErr, EP0IntStat.OUT_TrانErr	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	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 6-4 に、汎用エンドポイントEPa, EPb, EPc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 6-4 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	EPx{x=a-c}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 EPx{x=a-c}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	EPx{x=a-c}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	EPx{x=a-c}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 EPx{x=a-c}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	EPx{x=a-c}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	EPx{x=a-c}Control.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	EPx{x=a-c}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	EPx{x=a-c}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	EPx{x=a-c}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	EPx{x=a-c}IntStat.OUT_ShortACK, EPx{x=a-c}IntStat.IN_TranACK, EPx{x=a-c}IntStat.OUT_TranACK, EPx{x=a-c}IntStat.IN_TranNAK, EPx{x=a-c}IntStat.OUT_TranNAK, EPx{x=a-c}IntStat.IN_TranErr, EPx{x=a-c}IntStat.OUT_TranErr	トランザクションの結果を示します。

6.1.2.1.SETUP トランザクション

自ノードのエンドポイント EP0 宛ての SETUP トランザクションは、無条件に実施します。(USB_Control.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 EP0SETUP_0~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。

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。

SETUP トランザクションが完結すると、EP0ControlIN レジスタ及び 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、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、EP0ControlIN レジスタまたは 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 6-1 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EP0SETUP_0~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。

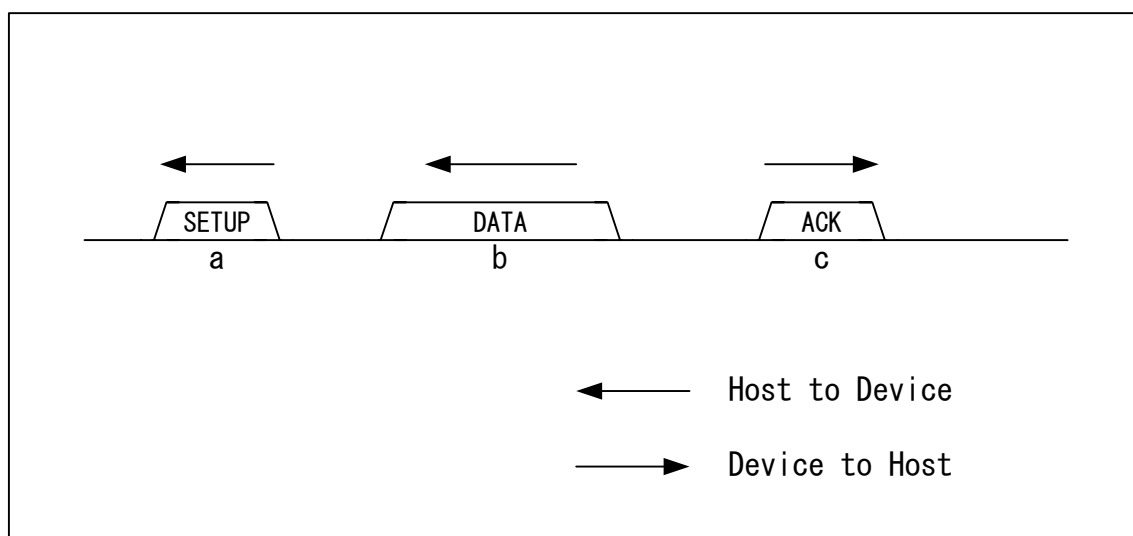


図 6-1 SETUP トランザクション

6.1.2.2. 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。

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_TransACK ステータス (EPx{x=0,a-c}IntStat.OUT_TransACK 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 (EPx{x=0,a-c}IntStat.OUT_ShortACK ビット)を発行します。また、EPx{x=a-c}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-c}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_TransErr ステータス (EPx{x=0,a-c}IntStat.OUT_TransErr 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_TransNAK ステータス (EPx{x=0,a-c}IntStat.OUT_TransNAK ビット)を発行します。FIFO は更新されません。

図 6-2 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。

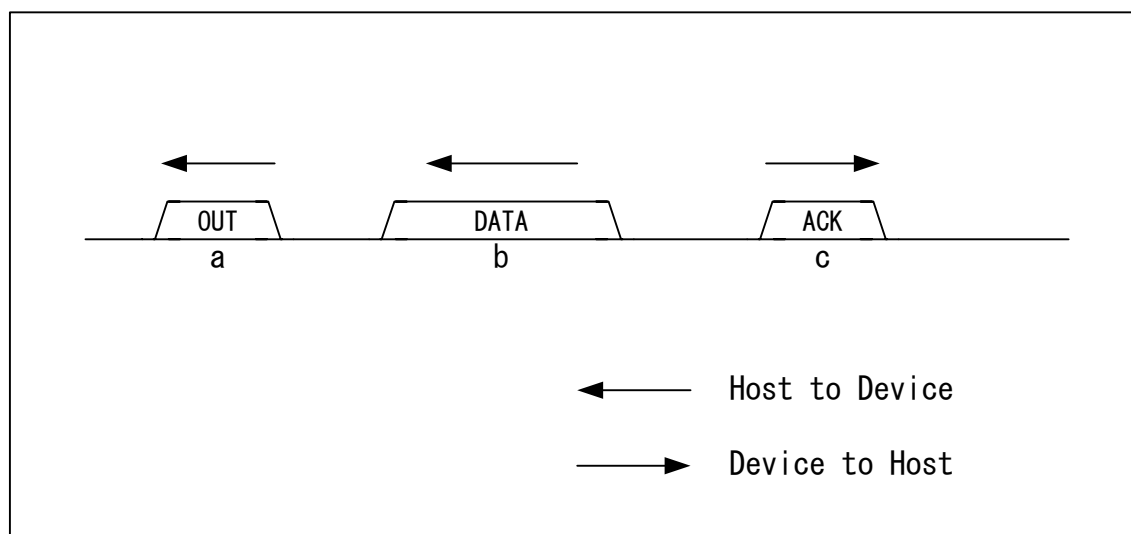


図 6-2 OUT トランザクション

6. 機能説明

6.1.2.3. 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ตอบสนอง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、EP0ControlIN.EnShortPkt ビットまたは EPx{x=a-c}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、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_TranACK ステータス (EPx{x=0,a-c}IntStat.IN_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。

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_TranErr ステータス (EPx{x=0,a-c}IntStat.IN_TranErr 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_TranNAK ステータス (EPx{x=0,a-c}IntStat.IN_TranNAK 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 6-3 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) 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ตอบสนอง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。

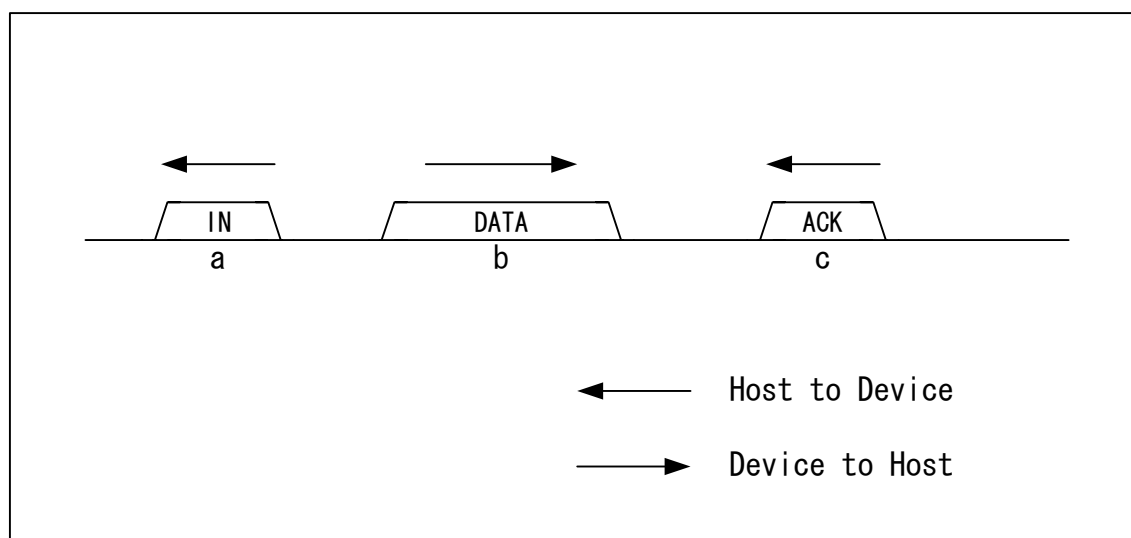


図 6-3 IN トランザクション

6.1.2.4. 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_TranNAK ステータス (EPx{x=0,a-c}IntStat.OUT_TranNAK 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 6-4 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。

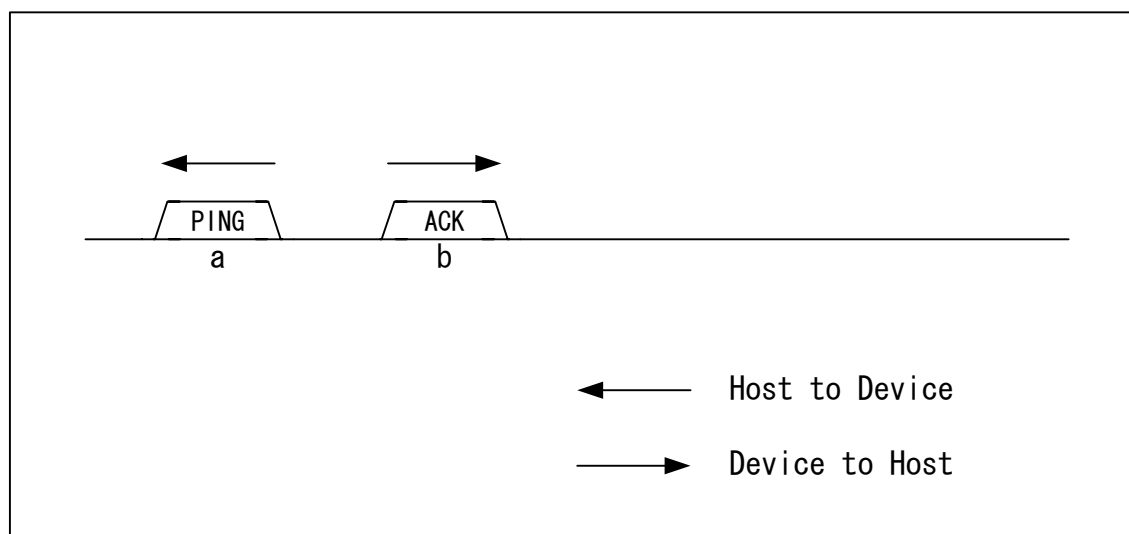


図 6-4 PING トランザクション

6.1.3. 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 6-5 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응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INT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バケットを返信します。

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(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。

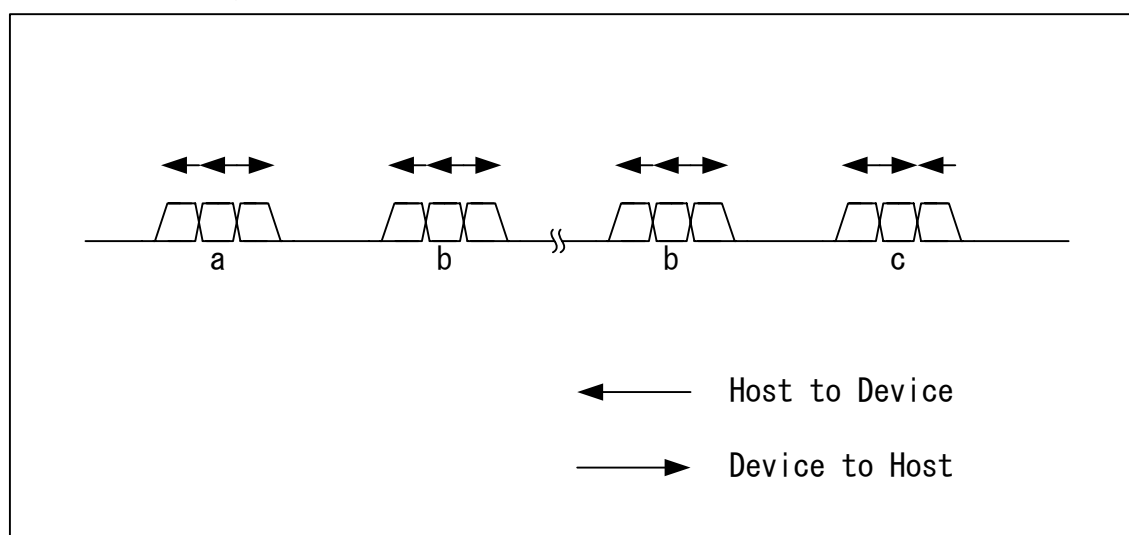


図 6-5 データステージが OUT 方向のコントロール転送

図 6-6 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、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응答する準備をします。(b)ホストはINT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。

ファームウェアは、RcvEP0SETUP ステータスをモニタし、EP0SETUP 0~EP0SETUP 7 レジスタによりリクエストを解析し

そのステージが OUT 方向である場合、EP0Control レジスタの INxOUT をクリアして OUT 方向に設定し、EP0ControlOUT

そのステージが IN 方向である場合、EP0Control レジスタの INxOUT をセットして IN 方向に設定し、EP0ControlIN レジスタ

LSI の h/w は EP0SETUP_0~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

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。

6.1.3.4. 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。

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、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、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、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응答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、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응答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、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 (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、機能説明の FIFO の項を参照して下さい。

6.1.4. バルク転送／インタラプト転送

汎用エンドポイントEPa, EPb, EPcにおけるバルク転送、インタラプト転送は、データフロー(6.1.5 参照)としても、連続する個々のトランザクション(6.1.2 参照)としても制御できます。

6.1.5. 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。

6.1.5.1. OUT 転送

OUT 転送によって受信したデータは、各エンドポイントの 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、EPx{x=0,a-c}Join.JoinCPU_Rd ビットにより、ただ一つのエンドポイントを選択して下さい。選択したエンドポイントの FIFO は、EPnFIFO_Rd レジスタ、または、EPnFIFO_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、EPnRdRemain_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 EPnRdRemain_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、EPx{x=0,a-c}Join.JoinDMAx{x=0,1}_Rd ビットにより、DMA のチャンネル毎にただ一つのエンドポイントを選択して下さい。選択したエンドポイントの FIFO 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、EPx{x=a-c}Join.JoinIDE ビットにより、ただ一つの OUT のエンドポイントを選択して下さい。選択したエンドポイントの 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。

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응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行うことなく、OUT 転送を行うことができます。但し、EPx{x=a-c}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 EPx{x=a-c}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、EPx{x=a-c}Control.ForceNAK ビットをクリアして下さい。

6.1.5.2. IN 転送

IN 転送により送信するデータを、各エンドポイントの FIFO 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

6. 機能説明

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、EPx{x=0,a-c}Join.JoinCPU_Wr ビットにより、ただ一つのエンドポイントを選択して下さい。選択したエンドポイントの FIFO には、EPnFIFO_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、EPnWrRemain_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 EPnWrRemain_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、EPx{x=0,a-c}Join.JoinDMAx{x=0,1}_Wr ビットにより、DMA チャンネル毎にただ一つのエンドポイントを選択して下さい。選択したエンドポイントの FIFO 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、EPx{x=a-c}Join.JoinIDE ビットにより、ただ一つの IN のエンドポイントを選択して下さい。選択したエンドポイントの 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。

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。

6.1.6. バルクオンリーサポート

本 LSI には、エンドポイント EPb, EPc 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。

BulkOnlyConfig.EPx{x=b,c}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の CBW サポート、または CSW サポートが実行されている間、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。

6.1.6.1. 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。BulkOnlyConfig.EPx{x=b,c}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制御して下さい。CBWサポートが有効であるときに、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(BulkIntStat.CBW_Cmpビット)を発行します。また、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、ファームウェアに対しCBWデータ長エラーステータス(BulkIntStat.CBW_LengthErrビット)を発行します。また、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_LengthErr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EPx{x=b,c}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、及び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エラーステータス(BulkIntStat.CBW_Errビット)を発行します。この場合には、BulkOnlyControl.GoCBW_Modeビットをクリアせず、CBWサポートの実行が継続されます。また、このとき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。

6.1.6.2.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できます。BulkOnlyConfig.EPx{x=b,c}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、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INTランザクションにおいて送信するデータをCSWとします。

INT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(BulkIntStat.CSW_Cmpビット)を発行します。また、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。

INT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(BulkIntStat.CSW_Errビット)を発行します。この時、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_Door 機能を用いてデータを書き込むことができます。

6. 機能説明

6.1.7. オート・ネゴシエーション機能

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。

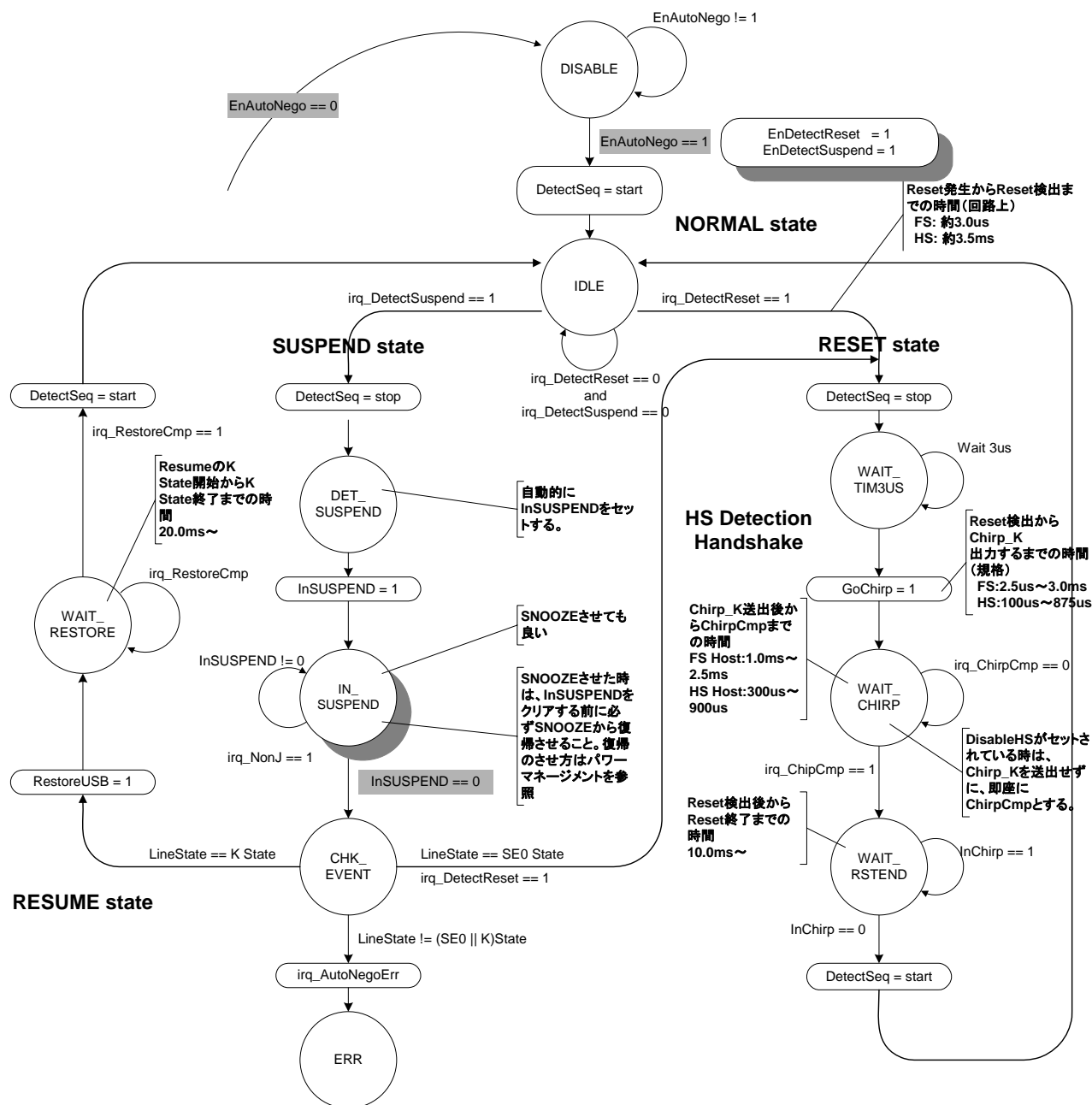


図 6-7 オート・ネゴシエータ

6.1.7.1.DISABLE

USB_Control.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、USB_Control.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 (SIE_IntEnb.EnDetectRESET)、サスペンド検出割り込み許可ビット (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、USB_Control.DisBusDetect ビットを絶対にセットしないでください。

6. 機能説明

6.1.7.2.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、SIE_IntStat.DetectRESET ビット、または 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。

6.1.7.3.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。

6.1.7.4.WAIT_CHIRP

USB_Control.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(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、6.1.7.11.5を参照してください。

また、USB_Control.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(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、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、SIE_IntEnb.EnChirpCmp ビットをセットしてください。

6.1.7.5.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。

6.1.7.6.DET_SUSPEND

サスペンドと判断された場合に、自動的に USB_Control.InSUSPENDビットがセットされ、IN_SUSPEND ステートに入ります。この USB_Control.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、6.2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 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。

6.1.7.7.IN_SUSPEND

NonJ 割り込みステータス(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、USB_Control.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。

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 USB_Control.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。

6.1.7.8.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、USB_Control.RestoreUSB ビットをセットし、サスペンド前の転送スピード(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

(DBG_IntStat.AutoNegoErr)ビットをセットし、ERR ステートに入ります。

6.1.7.9.WAIT_RESTORE

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。

6.1.7.10.ERR

一旦このステートに突入すると、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。

6. 機能説明

6.1.7.11. 各ネゴシエーション機能の単体説明

6.1.7.11.1. 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、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、SIE_IntStat.DetectSUSPEND ビットがセットされます。

この時、SIE_IntEnb.EnDetectSUSPEND ビットがセットされ、MainIntEnb.EnSIE_IntStat 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、USB のサスペンドステートであると判断します。以下の図では、スヌーズを行った時の動作を表しています。

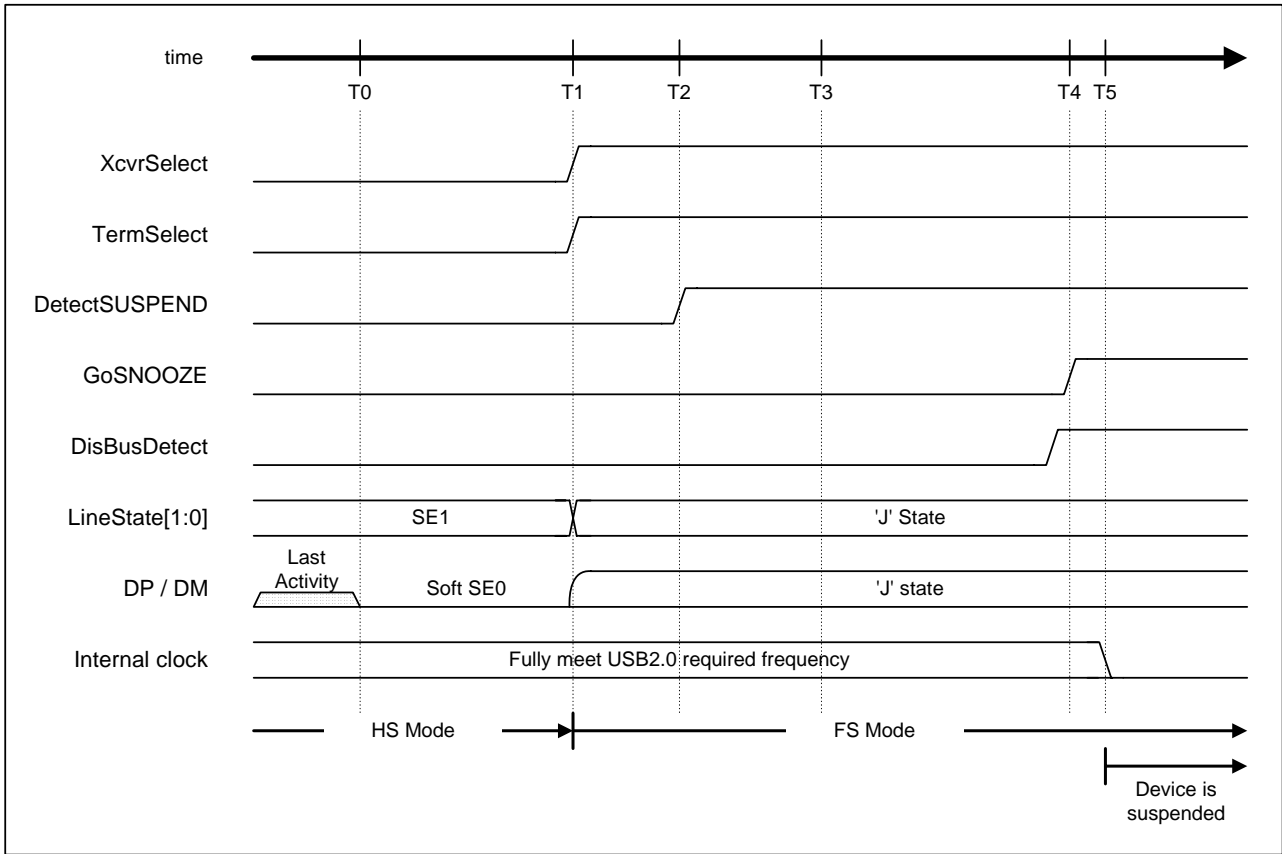


図 6-8 Suspend Timing (HS mode)

表 6-5 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 + 100us < T2 { $T_{WTWRSTHS}$ } < T1 + 875us
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	SnoozeUTM 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。(スヌーズ電流 8mA (typ))	T5 < T4 + 10us

注: {} は、USB2.0 規格書で規格されている名称である。

6. 機能説明

6.1.7.11.2. 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 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、SIE_IntEnb.EnDetectSUSPEND ビットがセットされ、MainIntEnb.EnSIE_IntStat 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、スヌーズを行った時の動作を表しています。

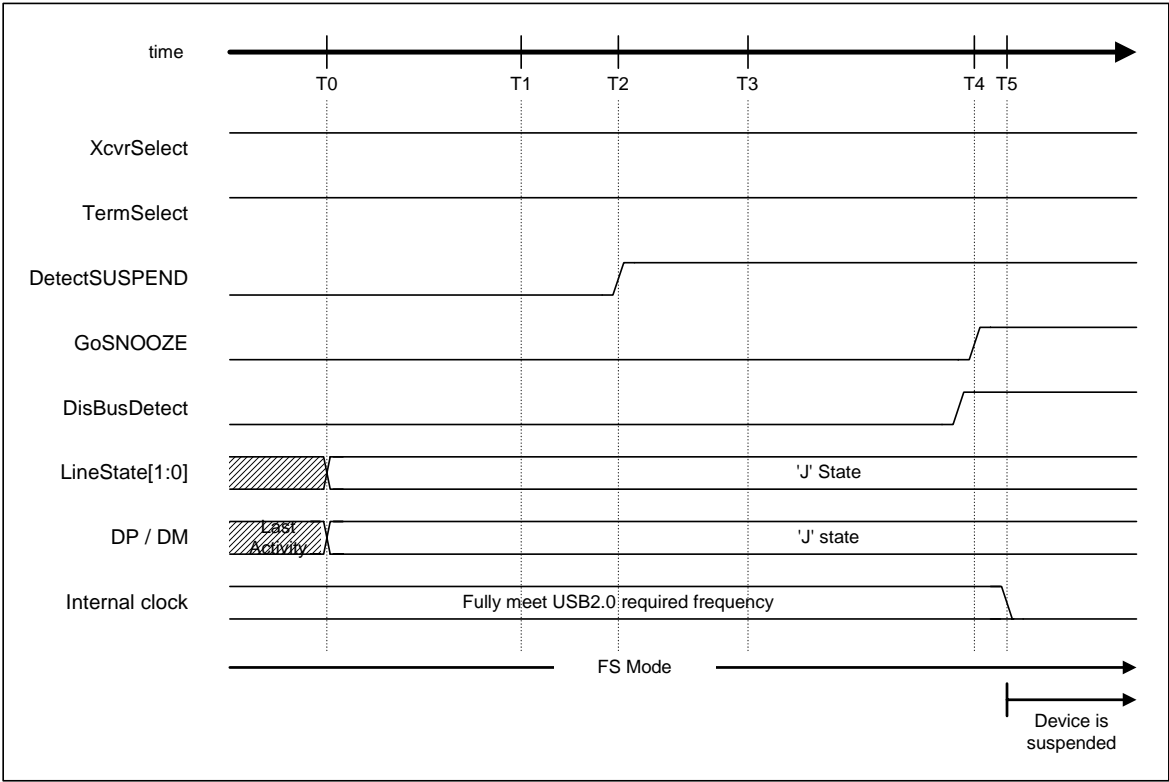


図 6-9 Suspend Timing (FS mode)

表 6-6 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。	$T1 + 100\mu\text{s} < T2 \{T_{WTWRSTHS}\} < T1 + 875\mu\text{s}$
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	SnoozeUTM 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。(スヌーズ電流 8mA (typ))	$T5 < T4 + 10\mu\text{s}$

注: {} は、USB2.0 規格書で規格されている名称である。

6.1.7.11.3. 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)。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 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、SIE_IntStat.DetectRESET ビットがセットされます。

この時、SIE_IntEnb.EnDetectRESET ビットがセットされ、MainIntEnb.EnSIE_IntStat 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、リセットの指示であると判断し、以降は、USB_Control.DisBusDetect ビットをセットした後に、HS Detection Handshake (後述) を行ってください。

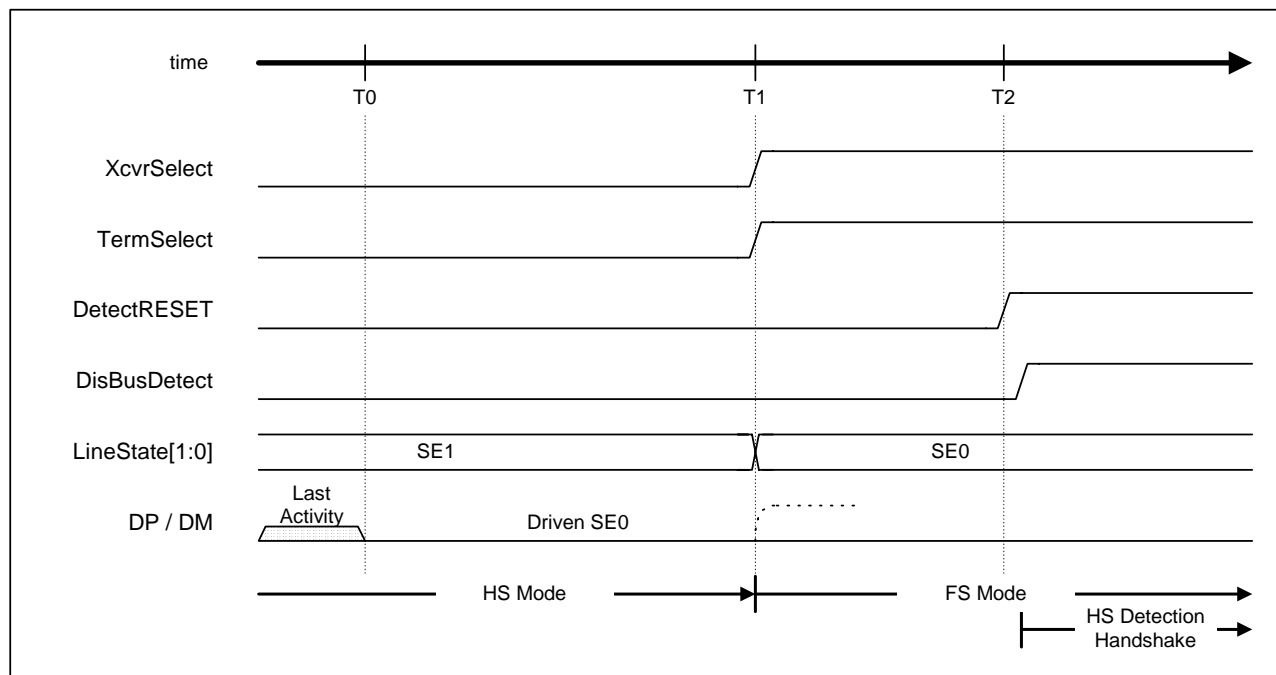


図 6-10 Reset Timing (HS mode)

表 6-7 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。

6. 機能説明

6.1.7.11.4. 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 USB_Status.LineState [1:0]ビットに”SE0”を検出し続けた場合には(T1)、SIE_IntStat.DetectRESET ビットがセットされます。

この時、SIE_IntEnb.EnDetectRESET ビットがセットされ、MainIntEnb. EnSIE_IntStat 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、リセットの指示であると判断し、以降は USB_Control.DisBusDetect ビットをセットした後に、HS Detection Handshake(後述)を行ってください。

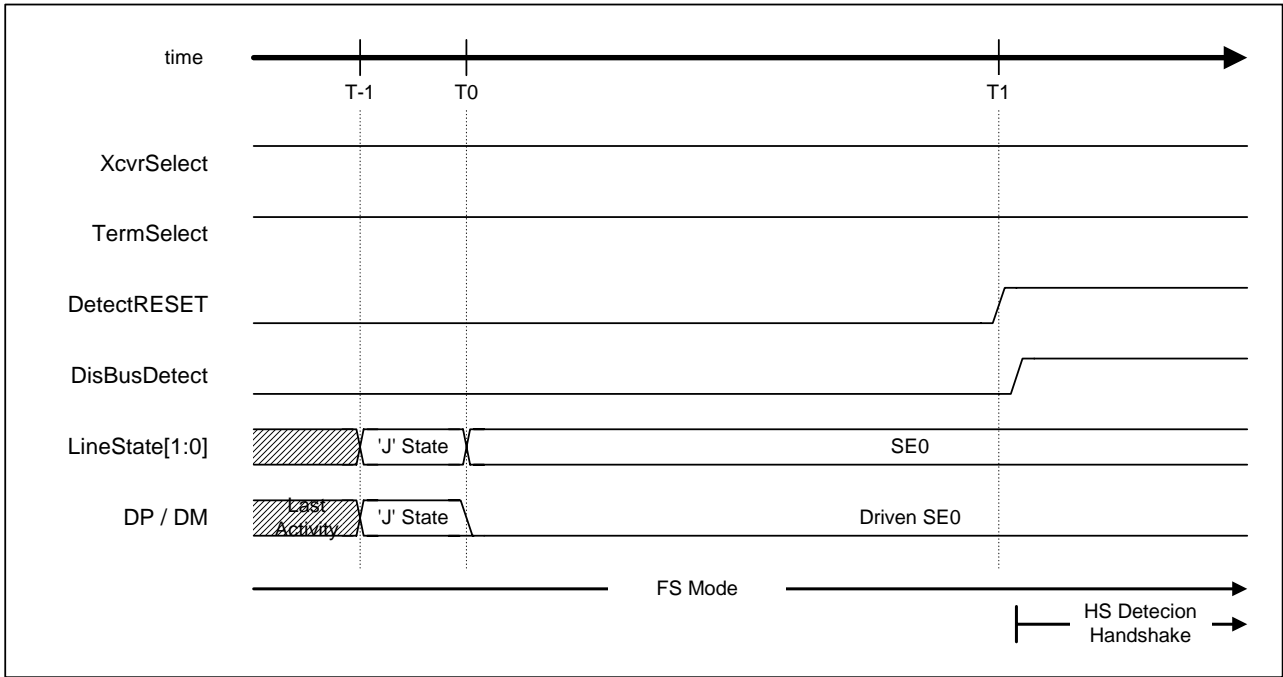


図 6-11 Reset Timing (FS mode)

表 6-8 Reset Timing Values (FS mode)

Timing Parameter	Description	Value
T-1	最後のバス・アクティビティ。	
T0	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 + 2.5us < T1 {T _{WTREV} }

注: {} は、USB2.0 規格書で規格されている名称である。

6.1.7.11.5. 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、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 に移行してください。

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 に移行してください。

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 のサスペンドステート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、XcvtControl.XcvtSelect、XcvtControl.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、3.125ms 以内に行われなければなりません。このモード切り替えから 100us 以上 875us 以内に USB_Status.LineState [1:0]ビットをチェックし、"J"なら USB のサスペンドステートとして判断し、"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 に移行してください。

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 が行うために、必ず PM_Control.GoActiveALL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、6.2 のパワーマネージメントを参照してください。

6. 機能説明

6.1.7.11.5.1. FS のダウンストリームポートに繋がれた場合

本 LSI が、HS をサポートしていない 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、XcvrControl.XcvrSelect と XcvrControl.TermSelect は両ビットともに FS モードでなければいけません (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、USB_Control.GoChirp ビットをセットします。すると 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" (chirp) を送出するためのものです。また同時に、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" (chirp) が送出されます。送出終了後、downstream port からの chirp を待ちます (T2)。通常、HS をサポートしている downstream port は、T3 から "HS K" "HS J" を連続的に送出してきますが (後述)、downstream port が HS をサポートしていない場合 (今回の場合) は、T4 の時点でも chirp を送出してこないため、XcvrControl.XcvrSelect ビットを FS モードに自動的に切り替え、USB_Control.GoChirp ビットがクリアされるとともに USB_Status.FSxHS ビットがセットされ、さらに SIE_IntStat.ChirpCmp ビットがセットされます。

この時、SIE_IntEnb.EnChirpCmp ビットがセットされ、MainIntEnb.EnSIE_IntStat 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。

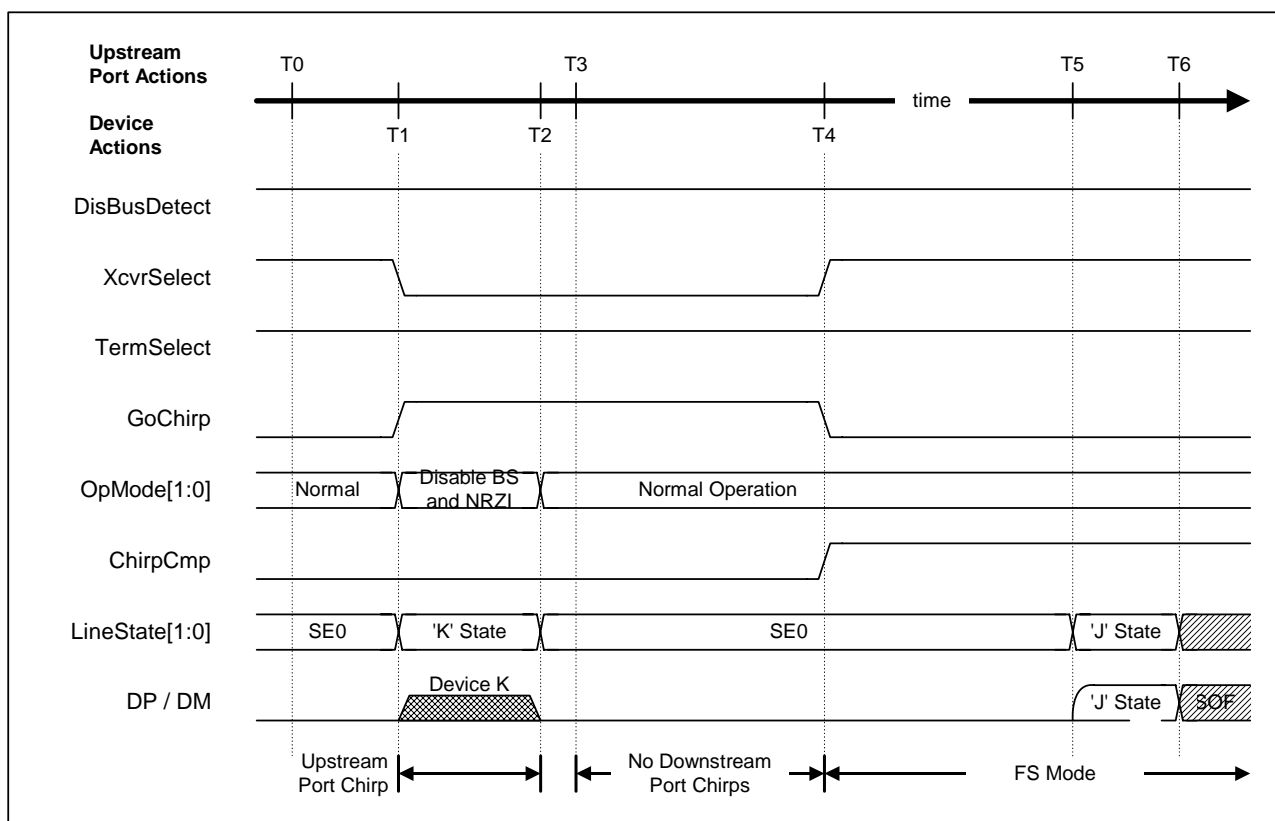


図 6-12 HS Detection Handshake Timing (FS mode)

表 6-9 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。	$T0 < T1 < \text{HS Reset } T0 + 6.0\text{ms}$
T2	Chirp K 送出終了。最小 1ms の間は送出しなければならない。	$T1 + 1.0\text{ms} \{T_{\text{UCH}}\} < T2 < \text{HS Reset } T0 + 7.0\text{ms} \{T_{\text{UCHEND}}\}$
T3	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。	$T2 < T3 < T2 + 100\mu\text{s} \{T_{\text{WTDCH}}\}$
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。	$T2 + 1.0\text{ms} < T4 \{T_{\text{WTFs}}\} < T2 + 2.5\text{ms}$
T5	リセットシーケンスの終了。	$\text{HS Reset } T0 + 10\text{ms} \{T_{\text{DRST}}(\text{Min})\}$
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。

6. 機能説明

6.1.7.11.5.2. HS の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 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、XcvrControl.XcvrSelect と 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 USB_Control.GoChirp ビットをセットします。すると 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" (chirp) を送出するためのものです。また同時に、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" (chirp) が送出されます。送出終了後、downstream port からの 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、USB_Control.GoChirp ビットがクリアされるとともに USB_Status.FSxHS ビットがクリアされ、さらに SIE_IntStat.ChirpCmp ビットがセットされます。

この時、SIE_IntEnb.EnChirpCmp ビットがセットされ、MainIntEnb.EnSIE_IntStat 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 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、USB_Status.LineState [1:0] ビットを使用できます。しかし、本来のパケット受信時に USB_Status.LineState [1:0] ビットにバスの信号を載せると非常にノイズな為、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、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、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、XcvrControl.TermSelect が FS モード時の Chirp は約 800mV、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。

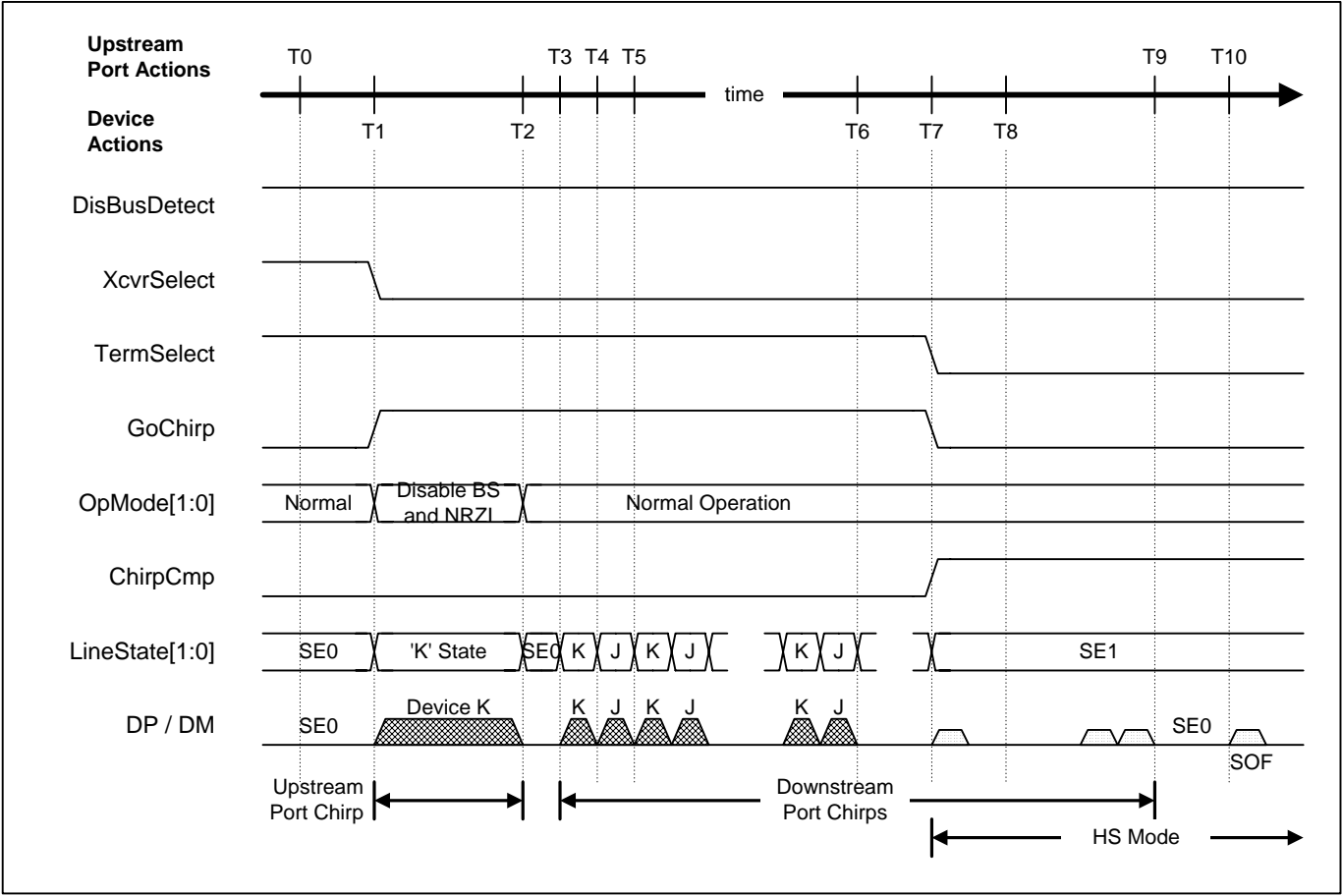


図 6-13 HS Detection Handshake Timing (HS mode)

6. 機能説明

表 6-10 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。	$T0 < T1 < \text{HS Reset } T0 + 6.0\text{ms}$
T2	Chirp K 送出終了。最小 1ms の間は送出しなければならない。	$T1 + 1.0\text{ms} \{T_{\text{UCH}}\} < T2 < \text{HS Reset } T0 + 7.0\text{ms} \{T_{\text{UCHEND}}\}$
T3	downstream port が最初の Chirp K をバスに送出。	$T2 < T3 < T2 + 100\text{us} \{T_{\text{WTDCH}}\}$
T4	downstream port が Chirp K から Chirp J に切り替え送出。	$T3 + 40\text{us} \{T_{\text{DCHBIT}}(\text{Min})\} < T4 < T3 + 60\text{us} \{T_{\text{DCHBIT}}(\text{Max})\}$
T5	downstream port が Chirp J から Chirp K に切り替え送出。	$T4 + 40\text{us} \{T_{\text{DCHBIT}}(\text{Min})\} < T5 < T4 + 60\text{us} \{T_{\text{DCHBIT}}(\text{Max})\}$
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。	$T6 < T7 < T6 + 500\text{us}$
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、パケット受信中与認識されることは無い。	T8
T9	downstream port からの Chirp K、Chirp J の送出終了。	$T10 - 500\text{us} \{T_{\text{DCHSE0}}(\text{Max})\} < T9 < T10 - 100\text{us} \{T_{\text{DCHSE0}}(\text{Min})\}$
T10	リセットシーケンスの終了。	$\text{HS Reset } T0 + 10\text{ms} \{T_{\text{DRST}}(\text{Min})\}$

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。

6.1.7.11.5.3. 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、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ます。

スヌーズ状態で、リセットが検出された場合（T0）、SIE_IntStat.NonJ ビットがセットされます。さらに、SIE_IntEnb.EnNonJ ビットがセットされ、MainIntEnb.EnSIE_IntStat 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、PM_Control.GoActiveALL 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.PM_State [2:0]が"ACTIVEALL"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）を行ってください。

この時、発振回路を停めていなければ（スリープ状態からの復帰でなければ）、内部クロックは USB2.0 規格に則った周波数精度で出力されています。

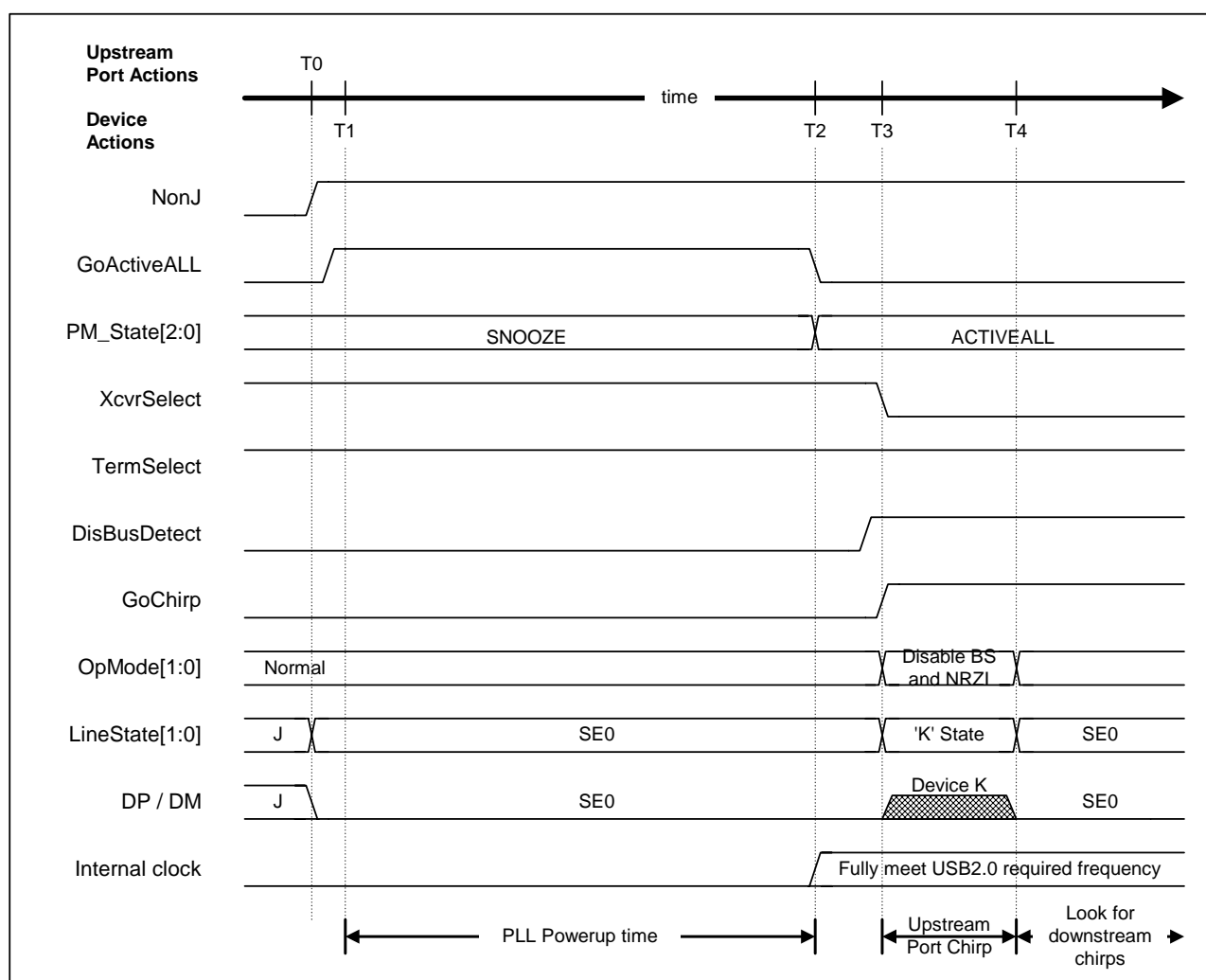


図 6-14 HS Detection Handshake Timing from Suspend

6. 機能説明

表 6-11 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、GoActiveALL を'1'にセット。	T1
T2	PM_State が'ACTIVEALL'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。	$T3 + 1.0\text{ms} \{T_{UCH}\} < T4 < \text{HS Reset T0} + 7.0\text{ms} \{T_{UCHEND}\}$

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）。

6.1.7.11.6. 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 から引っ張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。SIE_IntEnb.EnNonJ ビットをクリアし、PM_Control.GoActiveALL ビットをセットし (T0)、PLL パワーアップ時間経過後 (T1)、PM_Control.PM_State [2:0] ビットが "ACTIVEALL" になると同時に内部クロックが出力され始めます。この時、発振回路を停めていなければ、この内部クロックは、USB2.0 規格に則った周波数精度で出力されています。

その後、USB_Control.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、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) を送ります。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、USB_Control.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 (T4)、この時点では 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、USB_Control.RestoreUSB ビットをセットします。一定時間経過後、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、XcvtControl.XcvtSelect、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、USB_Control.RestoreUSB ビットがクリアされるとともに SIE_IntStat.RestoreCmp ビットがセットされます。この時、SIE_IntEnb.EnRestoreCmp ビットがセットされ、MainIntEnb.EnSIE_IntStat 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 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 の必要はありません。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。実際、FS モードの時には、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変化はありません。

本 LSI が、スヌーズ状態 (PM_Control.PM_State [2:0] 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。

6. 機能説明

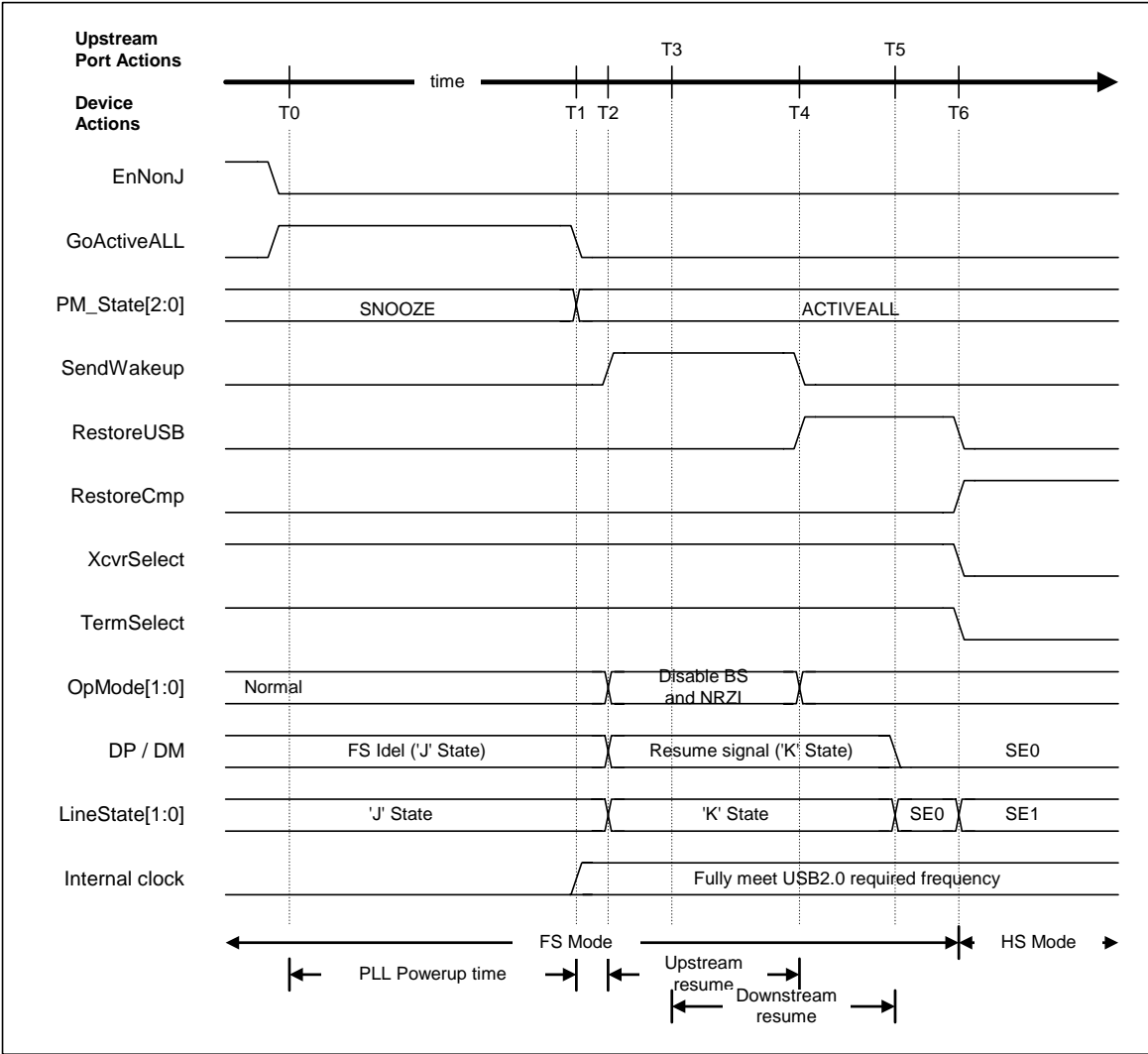


図 6-15 Assert Resume Timing (HS mode)

表 6-12 Assert Resume Timing Values (HS mode)

Timing Parameter	Description	Value
T0	レジューム開始。GoActiveALL 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'ACTIVEALL'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。	$T0 < T2 < T0 + 10ms$
T3	downstream port が FS の"K"を返す。	$T2 < T3 < T2 + 1.0ms$
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。	$T2 + 1.0ms \{T_{DRSMUP} (Min)\} < T4 < T2 + 15ms \{T_{DRSMUP} (Max)\}$
T5	downstream port が FS の"K"送出を終了。	$T2 + 20ms \{T_{DRSMDN}\}$
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。

6.1.7.11.7. レジュームの検出

本 LSI がスヌーズしている時、バス上には"J" (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、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、発振回路が動作を停止していなければ(スリープ状態でなければ)、SIE_IntStat.NonJビットがセットされます。この時、SIE_IntEnb.EnNonJビットがセットされ、MainIntEnb.EnSIE_IntStat 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.GoActiveALL 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.PM_State [2:0]が"ACTIVEALL"になると同時に内部クロックが出力され始めます。この時、発振回路を停めていなければ、この内部クロックは USB2.0 規格に則った周波数精度で出力されています。

そこで、USB_Control.RestoreUSB をセットします。一定時間経過後、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、XcvtControl.XcvtSelect、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、USB_Control.RestoreUSB ビットがクリアされるとともに SIE_IntStat.RestoreCmp ビットがセットされます。この時、SIE_IntEnb.EnRestoreCmp ビットがセットされ、MainIntEnb.EnSIE_IntStat 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。

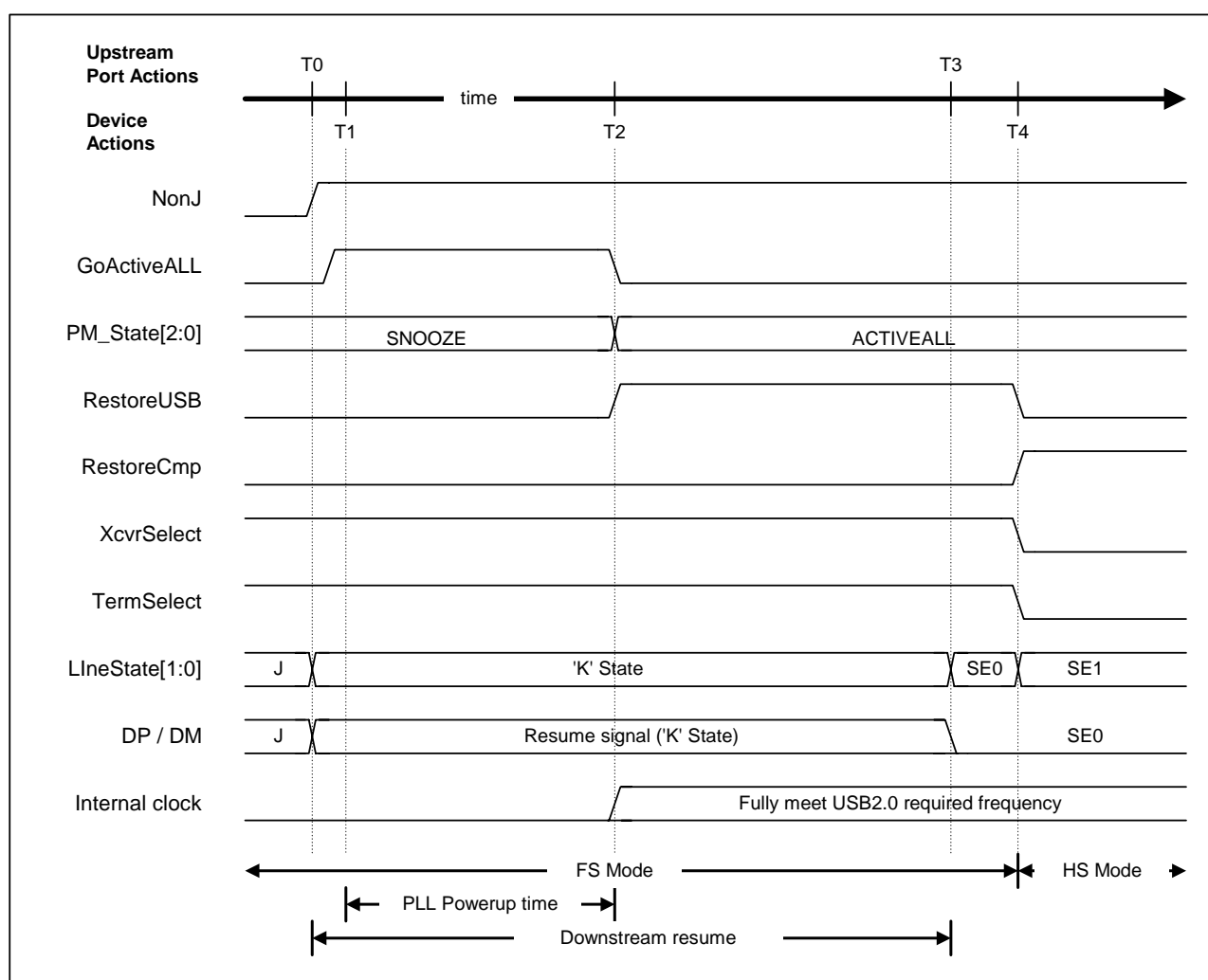


図 6-16 Detect Resume Timing (HS mode)

6. 機能説明

表 6-13 Detect Resume Timing Values (HS mode)

Timing Parameter	Description	Value
T0	downstream port が FS の "K" を送出。NonJ が '1' になる。	0 (reference)
T1	GoActiveALL を '1' にセット。	T1
T2	PM_State が 'ACTIVEALL' 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	downstream port が FS の "K" 送出を終了。同時に downstream port は USB のサスペンド以前の HS モードに移行。	$T2 + 20ms \{T_{DRSMDN}\}$
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。

6.1.7.11.8. 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、XcvtControl.XcvtSelect ビットは FS モード、XcvtControl.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、 V_{BUS} が "H" になり、同時に USB_Status.VBUS ビットがセットされます (T1)。この時スヌーズ状態にしていた場合は、PM_Control.GoActiveALL 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.PM_State [2:0] が "ACTIVEALL" 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ないため、一旦は FS モードになるために、XcvtControl.TermSelect ビットを FS モードにしてください (T4)。

その後、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。

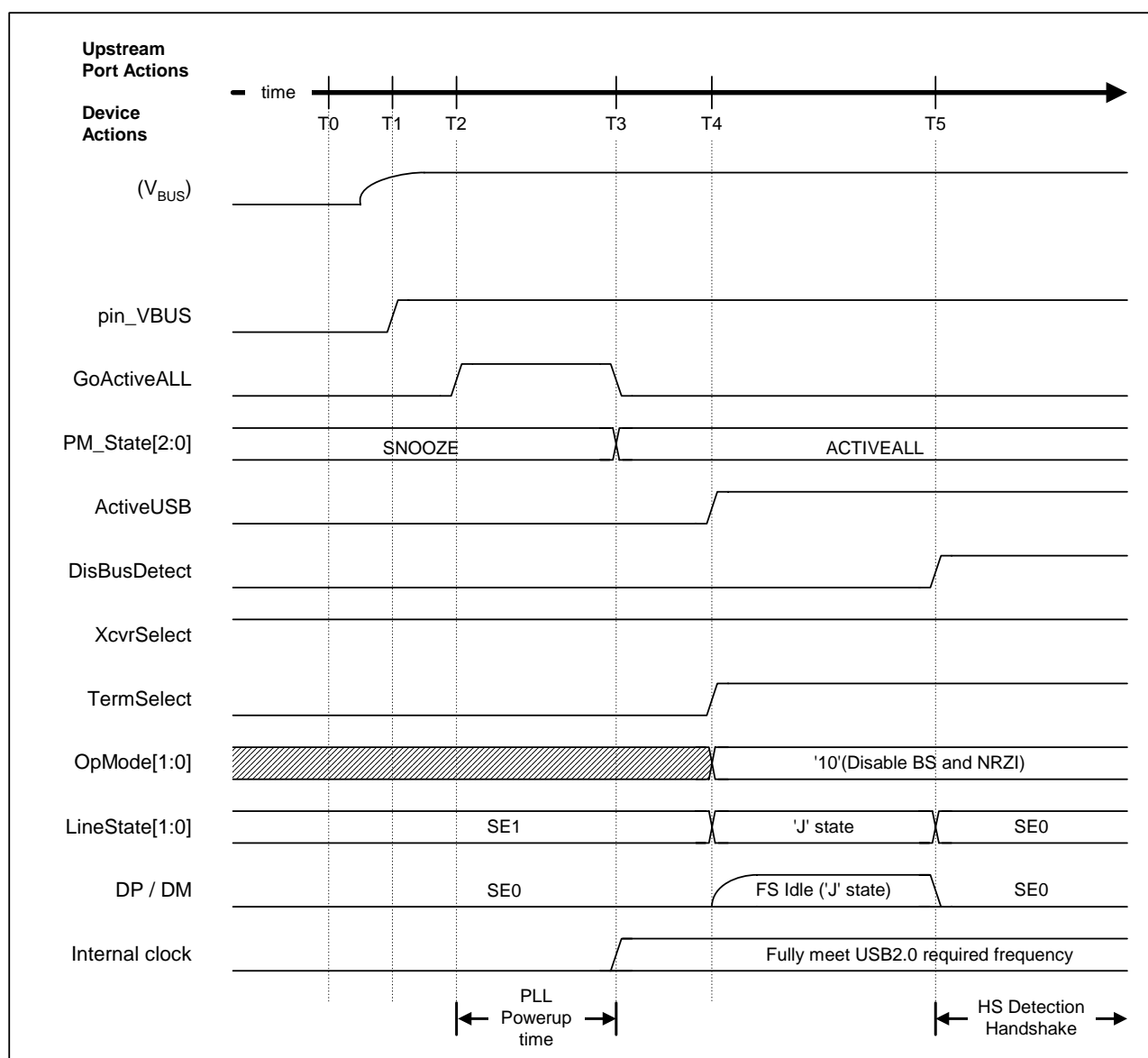


図 6-17 Device Attach Timing

6. 機能説明

表 6-14 Device Attach Timing Values

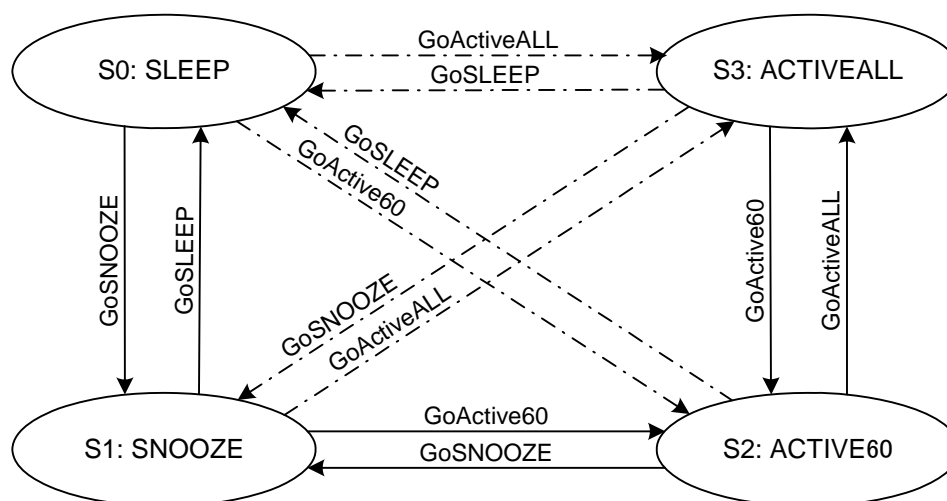
Timing Parameter	Description	Value
T0	ケーブルは挿入されていない。	0 (reference)
T1	ケーブルが挿入され、入力ピン VBUS が H になる。	T1
T2	GoActiveALL を '1' にセット。	T2
T3	PM_State が 'ACTIVEALL' になる。内部クロック出力の安定。	$T2 + 250\mu s < T3$
T4	ActiveUSB を '1' にセット。TermSelect を '1' にセット。OpMode [1:0] を '00' に設定。 FS モードに移行。FS ターミネーションが有効。	$T1 + 100ms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

注: {} は、USB2.0 規格書で規格されている名称である。

6.2. パワーマネージメント機能

オシレータ、PLL(PLL480, PLL60)の動作を制御し、スリープ、スヌーズ、アクティブ 60、アクティブオール の 4 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.GoSLEEP、PM_Control.GoSNOOZE、PM_Control.GoActive60、PM_Control.GoActiveALL 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.PM_State [2:0]を確認してください。また遷移終了時、SIE_IntStat.FinishedPM イベントが発生します。この時、SIE_IntEnb.EnFinishedPM ビットをセットし、かつ MainIntEnb.EnSIE_IntStat 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、アクティブオール・ステート時に PM_Control.GoSLEEP ビットをセットした場合は、アクティブ 60・ステート、スヌーズ・ステートを経由し、スリープ・ステートに遷移し、完全に遷移終了した時点で SIE_IntStat.FinishedPM イベントが発生します。またスリープ・ステート時に PM_Control.GoActiveALL ビットをセットした場合も、スヌーズ・ステート、アクティブ 60・ステートを経由し、アクティブオール・ステートに遷移し、完全に遷移終了した時点で SIE_IntStat.FinishedPM イベントが発生します。同様に、アクティブ 60・ステート時に PM_Control.GoSLEEP ビットをセットした場合も、スヌーズ・ステートを経由し、スリープ・ステートに遷移し、完全に遷移終了した時点で SIE_IntStat.FinishedPM イベントが発生します。またスリープ・ステート時に PM_Control.GoActive60 ビットをセットした場合も、スヌーズ・ステートを経由し、アクティブ 60・ステートに遷移し、完全に遷移終了した時点で SIE_IntStat.FinishedPM イベントが発生します。



※ 一点鎖線の遷移は、実際には実線を通して遷移する。

図 6-18 パワーマネージメント

6.2.1. 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。

スヌーズ・ステート、アクティブ 60・ステート及びアクティブオール・ステート中に PM_Control.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、OSC を PLL480、PLL60 の順で停止し、最終的に、OSCCLK の出力を停めた後に発振を停止します。

逆に、スリープ・ステート中に PM_Control.GoSNOOZE、PM_Control.GoActive60 ビット及び PM_Control.GoActiveALL ビットをセットし、スリープからスヌーズへ離脱する時には、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。

6. 機能説明

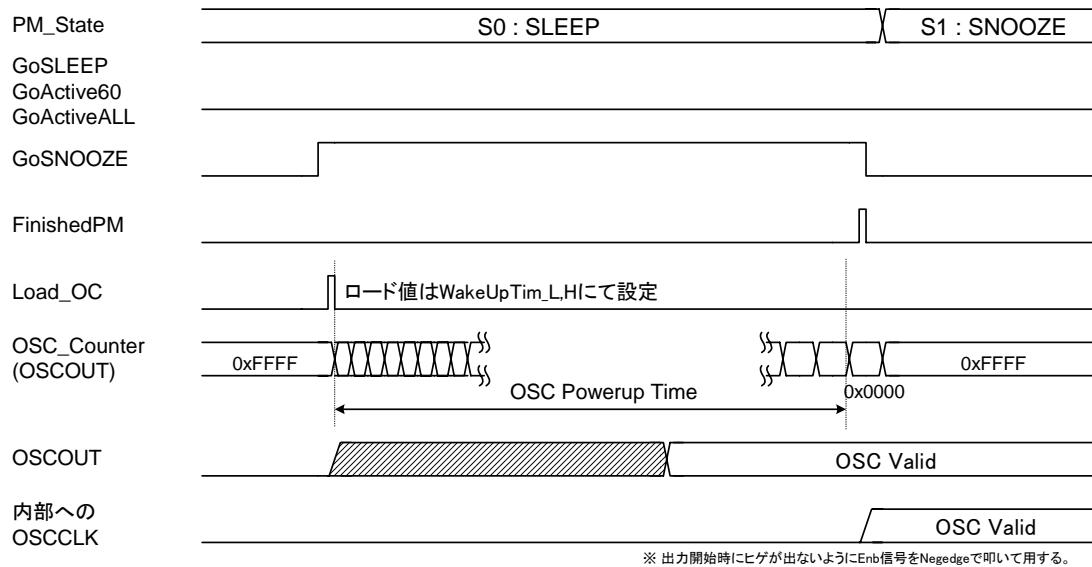


図 6-19 SLEEP ステートからの離脱 (GoSNOOZE 時)

6.2.2. 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。

アクティブ 60・ステート及びアクティブオール・ステート中に PM_Control.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、PLL480、PLL60 の順で停止します。

しかし、スヌーズ・ステート中に PM_Control.GoActiveALL ビット及び PM_Control.GoActive60 ビットをセットし、スヌーズからアクティブへ離脱する時には、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。

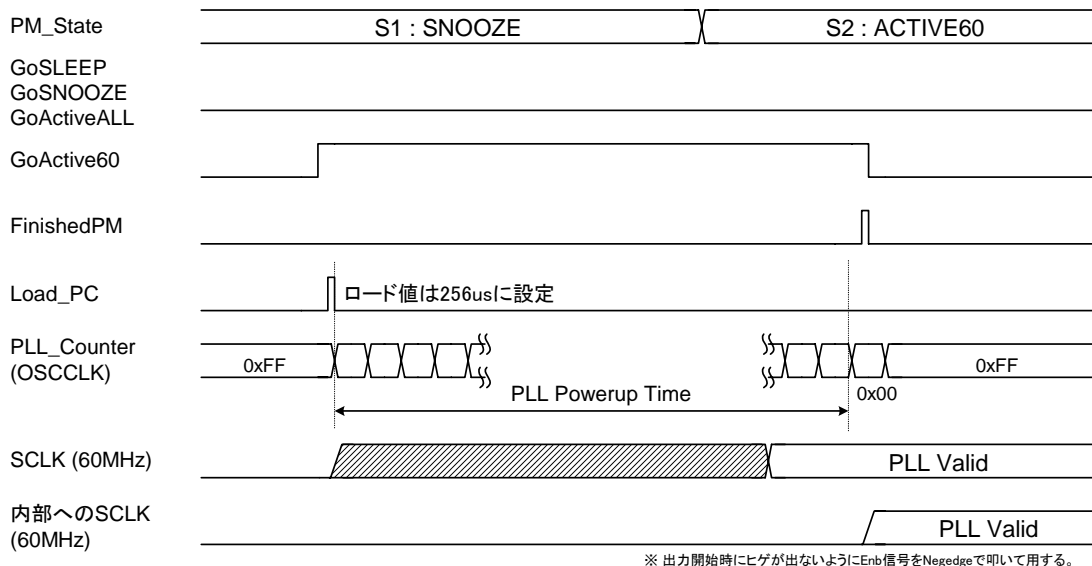


図 6-20 SNOOZE ステートからの離脱 (GoActive60 時)

6.2.3. ACTIVE60（アクティブ 60）

オシレータ、PLL60 が動作し、PLL480 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きましては、アクティブ 60・ステート及びアクティブオール・ステートで無ければ読み書きできず、読むと全て 0 として読み出されてしまいます。

また、USB 回路は、SCLK480 を必要としているためアクティブオール・ステートでのみ動作します。IDE 並びに CPU 回路は、SCLK480 が不必要なため、アクティブ 60・ステートもしくはアクティブオール・ステートの両方で動作します。480MHz の PLL の消費電流は 60MHz の消費電流に比較して大きいので、USB を必要としない場合にはアクティブ 60・ステートに遷移することで、低消費電力化が図れます。

6.2.4. ACTIVEALL（アクティブ 480）

オシレータ、PLL60、PLL480 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きましては、アクティブ 60・ステート及びアクティブ 480・ステートで無ければ読み書きできず、読むと全て 0 として読み出されてしまいます。

また、USB 回路は、SCLK480 を必要としているためアクティブオール・ステートでのみ動作します。IDE 並びに CPU 回路は、SCLK480 が不必要なため、アクティブ 60・ステートもしくはアクティブオール・ステートの両方で動作します。480MHz の PLL の消費電流は 60MHz の消費電流に比較して大きいので、USB を必要としない場合にはアクティブ 60・ステートに遷移することで、低消費電力化が図れます。

6. 機能説明

6.3. FIFO 管理

6.3.1. FIFO 管理

FIFO 管理について説明します。

6.3.1.1. FIFO メモリマップ

FIFO のメモリマップを以下に示します。

(0xA00)	
0x600	EPcエリア(1024B)
0x200	EPbエリア(1024B)
0x1C0	EPaエリア(64B)
0x1B0	CSWエリア(16B)
0x190	CBWエリア(32B)
0x040	デスクリプタエリア(336B)
0x000	EP0エリア(64B)

図 6-21 FIFO メモリマップ

FIFO のメモリは、EP0 エリア、デスクリプタエリア、CBW エリア、CSW エリア、EPa エリア、EPb エリア、EPc エリアの、7 のエリアに分割され、それぞれに固定領域が割り当てられています。

EP0 エリアは、USB で必須のエンドポイント 0 に使用するエリアで、IN 及び OUT の両方向に使用されます。このエリアには、64 バイトが確保されていますが、そのうち使用できるのは、アドレス 0x000 から始まる、エンドポイント 0 のマックスパケットサイズ分の領域です。したがって、エンドポイント 0 は常にシングルバッファとなります。

デスクリプタエリアは、デスクリプタ返信機能で使用される領域です。332 バイトが確保されており、そ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、6.3.1.2 に後述します。実際には全ての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をデスクリプタエリアと定め、使用を推奨します。

CBWエリアは、バルクオンリーサポート機能のCBWサポートで使用される領域です。32 バイト確保されていますが、そのうち、アドレス 0x190 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、6.3.1.3 に後述します。

CSWエリアは、バルクオンリーサポート機能のCSWサポートで使用される領域です。16 バイト確保されていますが、そのうち、アドレス 0x1B0 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、6.3.1.4 に後述します。

EPa エリア、EPb エリア、EPc エリアは、エンドポイント番号と、IN または OUT の方向を任意に設定できる、汎用エンドポイントのエリアです。EPa エリアは 64 バイトが確保されていますので、FS モードのバルク転送や HS/FS モードのインタラプト転送に使用することが出来ます。EPb エリア、EPc エリアは 1024 バイトが確保されていますので、加えて HS モードのバルク転送にも使用することが出来ます。

EP0 エリア、EPa エリア、EPb エリア、EPc 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、EPnControl.AllFIFO_Clr ビットか、または、各領域に対応した、EPnControl.EP0FIFO_Clr、EPnFIFO_Clr.EPx(x=a-c)FIFO_Clr 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。

6.3.1.2. デスクリプタエリアの使用方法

デスクリプタエリアは、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 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。

6.3.1.2.1. 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_H,L 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_H,L レジスタに連続して書き込むことが出来ます。

なお、RAM_WrDoor_H,L レジスタは書き込みのみとなっています。

6.3.1.2.2. 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、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 DescSize_H,L レジスタに設定した後、EP0Control.ReplyDescriptor ビットに 1 をセットします。また、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、SETUP_Control.ProtectEP0 をクリアしてから、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 (EP0MaxSize にて設定) に自動的に分割しながら、ホストに 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、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、EP0Control.ReplyDescriptor をクリアし、EPrintStat.DescriptorCmp がセットされます。ファームウェアはステータスステージの処理に移行して下さい。

6.3.1.3. CBW エリアの使用方法

CBW エリアは、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 EPb, EPc) で、Bulk Only Transport 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 にはデータだけを受信することが出来、IDE や DMA 等の転送の制御を容易にします。

6.3.1.3.1. CBW エリアへの受信

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。

6.3.1.3.2. CBW エリアからのデータの読み出し

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 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 (FIFO_IntStat.RAM_RdCmp ビット) が発行されます。

6.3.1.4. CSW エリアの使用方法

CSW エリアは、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 EPb, EPc) 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 からはデータだけを送信することが出来、IDE や DMA 等の転送の制御を容易にします。

6.3.1.4.1. CSW エリアからの送信

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

6. 機能説明

のデータをデータパケットとして送信します。

6.3.1.4.2. CSW エリアへのデータの書き込み

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(0x1B0)を書き、RAM_WrDoor_H,L 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なく問題ありません。

6.3.2. FIFO へのアクセス方法

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

6.3.2.1.FIFO へのアクセス方法 (RAM_Rd)

FIFO に CPUIF の 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 FIFO_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 レジスタからデータを読み出ししてください。RAM_Rd レジスタのデータは 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、エンドポイントの 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_Adrs_H,L レジスタ、RAM_Count 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 FIFO_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。

6.3.2.2.FIFO へのアクセス方法 (RAM_WrDoor)

FIFO に CPUIF の RAM_WrDoor_H,L 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_H,L 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_H,L レジスタに連続して書き込むことが出来ます。

RAM_WrDoor_H,L レジスタによる FIFO への書き込みは、エンドポイントの FIFO 領域の設定に関係なく、任意に行う事が出来ます。

6.3.2.3.FIFO へのアクセス方法 (レジスタアクセス)

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、いずれか 1 つのエンドポイントに対して、EPx{x=0,a-c}Join.JoinCPU_Rd に1をセットし、EPnFIFO_Rd レジスタ、または、EPnFIFO_ByteRd レジスタにて読み出しを行います。

また、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、いずれか 1 つのエンドポイントに対して、EPx{x=0,a-c}Join.JoinCPU_Wr に1をセットし、EPnFIFO_Wr レジスタに書き込みを行います。

EPnRdRemain_H,L レジスタは、EPx{x=0,a-c}Join.JoinCPU_Rd にてただ 1 つ設定されたエンドポイントにおいて、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、EPnWrRemain_H,L レジスタは、EPx{x=0,a-c}Join.JoinCPU_Wr にてただ 1 つ設定されたエンドポイントにおいて、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、EPx{x=0,a-c}Join.JoinCPU_Rd レジスタ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。

6.3.2.4.FIFO へのアクセス方法 (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、DMA の各チャンネルにつき、いずれか 1 つのエンドポイントの、EPn{n=0,a-c}Join.JoinDMAx{x=0,1}_Rd レジスタに1をセットし、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 の各チャンネルにつき、いずれか1つのエンドポイントの、EPn{n=0,a-c}Join.JoinDMAx{x=0,1}_Wr レジスタに1をセットし、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、EPn{n=0,a-c}Join.JoinDMAx{x=0,1}_Rd にて、ただ

1つ設定されたエンドポイントにおいて、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、EPn{n=0,a-c}Join.JoinDMAx{x=0,1}_Wrにて、ただ1つ設定されたエンドポイントにおいて、FIFOに書き込み可能なエリアの残り数を表しています。

6.3.2.5.FIFO へのアクセス方法(IDE)

FIFOにIDEがアクセスする場合には、いずれか1つのエンドポイントに対して、EPx{x=0,a-c}Join.JoinIDE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転送の方向は接続されているエンドポイントの方向に応じて設定されます。

6.3.2.6.FIFO へのアクセス制限

本LSIのFIFOには、USBとの送受信、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、それぞれのエンドポイントにおける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。

- 同一のエンドポイントに対し、複数の書き込み要因を同時に設定してはいけません。
- 同一のエンドポイントに対し、複数の読み出し要因を同時に設定してはいけません。
- 1つのエンドポイントには、JoinCPU_Wr, JoinCPU_Rd, JoinDMAx{x=0,1}_Wr, JoinDMAx{x=0,1}_Rdのうちの1つしか設定できない。
- JoinIDE, JoinCPU_Wr, JoinCPU_Rd, JoinDMAx{x=0,1}_Wr, JoinDMAx{x=0,1}_Rdは、それぞれ、同時に1つのエンドポイントにしか設定できない。
- JoinCPU_WrとJoinCPU_Rdは同時に設定出来ない。

同一のエンドポイントに対する書き込み／読み出し要因の重複に関し、USB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}_Wrを設定して書き込むことができます。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}_Wr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}_Rdを設定して読み出すことができます。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}_Rd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、各エンドポイントのEnEndpointがクリアされている場合や、ForceNAKがセットされている場合などがあります。

6. 機能説明

6.4. CPUIF

6.4.1. モード切り替え

S1R72V03 の CPUIF は非同期 CPU に対応し、以下の 3 つの動作モードを持っております。

表 6-15 CPUIF 動作モード設定

動作モード	BusMode	Bus8x16	備考
16bit Strobe mode	0	0	デフォルト
16bit BE mode	1	*	BusMode ビットの設定優先
8bit mode	0	1	

各動作モードの切り替えは ChipConfig レジスタの BusMode ビット、Bus8x16 ビットの設定によって行います。ChipConfig レジスタの値は ModeProtect レジスタの設定によって誤書き込みに対するプロテクトをかけることができます。

実使用時には、電源投入直後、第一に ChipConfig レジスタの設定を行い動作モードを決定して下さい。その後、ModeProtect レジスタにより設定を保護して下さい。

また、S1R72V03 の CPUIF はバスの Swap 機能を持ちます。この機能を使用する場合は ChipConfig レジスタの初期設定時に ChipConfig.CPU_BusSwap ビットを設定して下さい。Swap 機能は CPU_BusSwap ビットを設定した後、E9h 番地をリードする事で有効になります。ChipConfig レジスタの設定では上記に加え、XINT の論理レベル及び出力モードの設定、XDREQ0,1、XDACK0,1 の論理レベル設定、DMA0,1 の CS_Mode の設定を行う事が出来ます。

以下の説明においては、特に断りの無い限りデフォルト設定の 16bit Strobe mode、Bus Swap なしを基本として説明致します。

6.4.2. モード切り替えに際して

S1R72V03 は ChipConfig レジスタにより、ご使用の CPU に適合する CPU バスの動作モードに設定することが可能です。チップの初期状態は 16bit Strobe mode で動作しておりますので、16bit BE mode、または 8bit mode に切り替える際には以下の点にご注意下さい。

6.4.2.1. 16bit BE mode をご使用の場合

ご使用のCPUに合わせて 16bit BE modeを使用する場合、6.4.1 に示す通り、ChipConfigレジスタへの設定を第一に行って下さい。また、その際には 図 6-22 に示す通り、必ずEFh番地に対するバイトライトを行って下さい。この時、S1R72V03 は初期状態の 16bit Strobe modeで動作しており、CPUのchip select信号とbyte mask high信号(XCSとXWRH)に下図に示すようなスキューがある場合、有効なライト期間と見なして内部的に動作してしまう可能性があります。S1R72V03 ではこのようなスキューを除去する為のフィルタ回路(min:1ns)を搭載しておりますが、ご使用のCPUのAC特性をご確認の上、基板上での処理等により、このようなスキューが生じないようにして下さい。

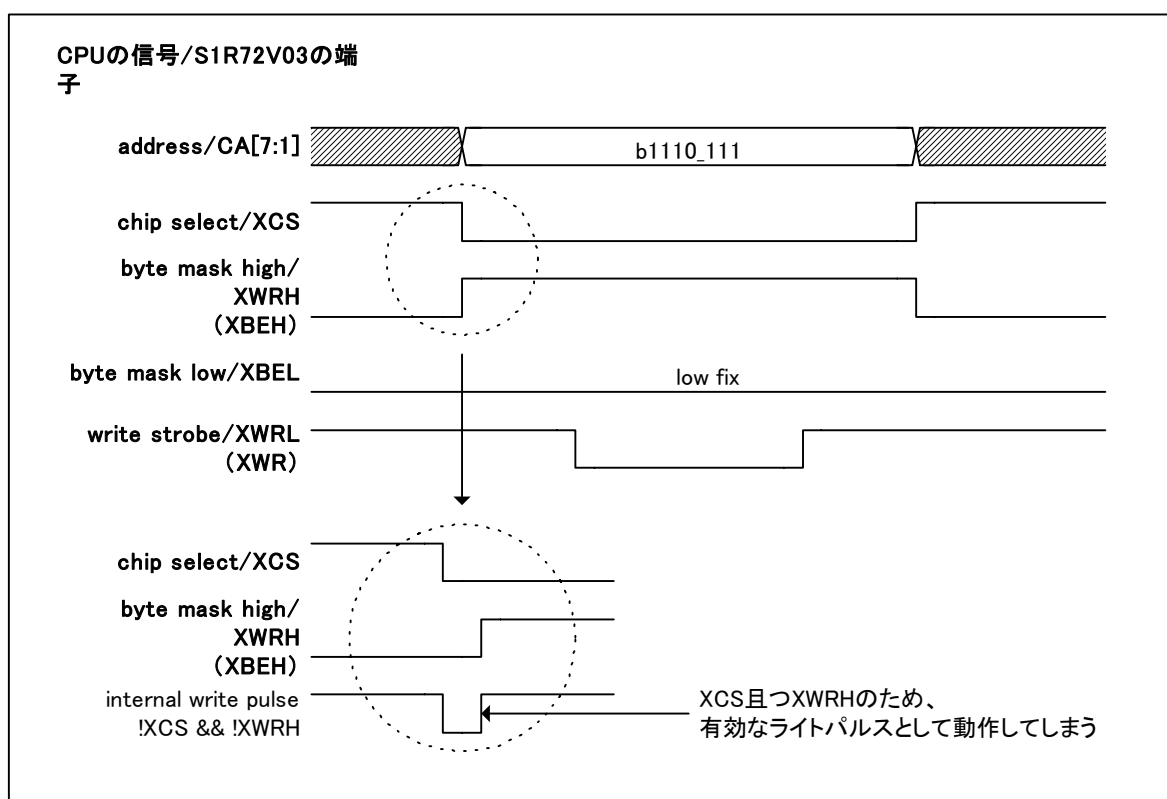


図 6-22 ChipConfig レジスタの初期設定

動作モードの設定完了後は内部的なライトパルス (internal write pulse) の生成条件が更新されますので、このような制限はございません。

また、ChipConfigレジスタの設定前に、S1R72V03 にリードアクセスを行った場合、図 6-23 に示す通り、チップの内部ではリード動作とライト動作を同時に行ってしまいます。この場合の動作は保証されませんので、必ずChipConfigレジスタの設定を第一に行ってください。

6. 機能説明

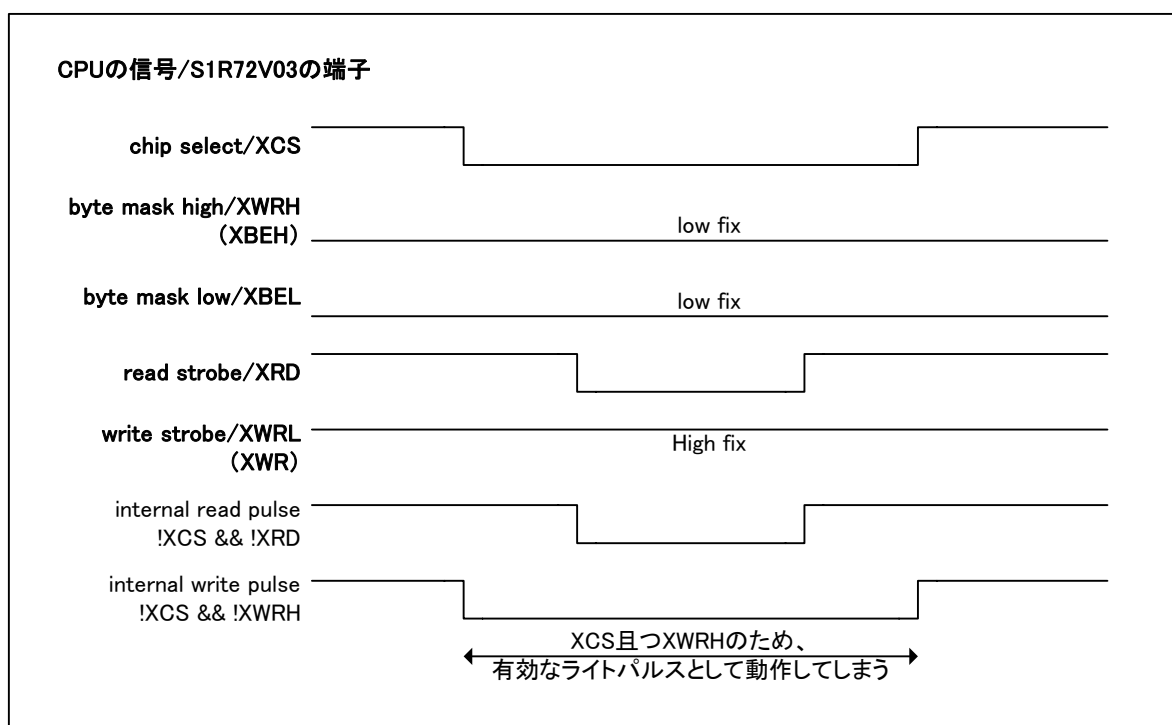


図 6-23 ChipConfig レジスタの初期設定前のリードアクセス

6.4.2.2. 8bit mode をご使用の場合

ご使用のCPUに合わせて8bit modeを使用する場合、6.4.1に示す通り、ChipConfigレジスタへの設定を第一に行ってください。ChipConfigレジスタの設定の前に、S1R72V03 にリードアクセスを行った場合、S1R72V03 は初期状態の 16bit Strobe mode で動作しておりますので、CD[15:0]全端子が出力状態になります。CD[15:8]をPull Up/Pull Down処理されている場合は特に問題ありませんが、これらの端子をVDD/GNDに直結している場合、電流消費が大幅に増大します。これを回避する為、必ず ChipConfigレジスタの設定を第一に行ってください。

6.4.3. ブロック構成

S1R72V03 CPUIF(以下、CPUIF)のブロック構成を図 6-24 に示します。

REG/DMA0/DMA1 の3ブロックから構成されます。

- ・ REG: S1R72V03 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

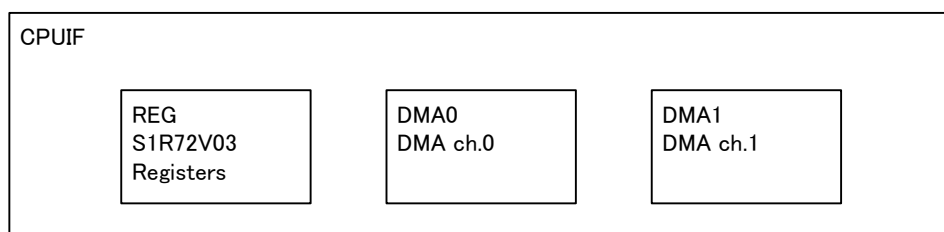


図 6-24 ブロック構成

6.4.3.1. REG (S1R72V03 Registers)

S1R72V03 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
- ・

6.4.3.1.1. 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。

6.4.3.1.2. 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 (8bit mode の場合 2 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します。

6.4.3.1.3. FIFO アクセス（ライト）

FIFO ライトアクセスは EPnFIFO_Wr_H,L、RAM_WrDoor_H,L レジスタへの書き込みを意味します。8bit mode 動作の場合は EPnFIFO_Wr_H,L レジスタのどちらにアクセスを行っても FIFO への書き込みを行うことができます。RAM_WrDoor_H,L に関しても同様です。

FIFO アクセス(ライト)には以下の制限事項があります。

- ・ EPx{x=0,a-c}.Join.JoinCPU_Wr ビットを設定した後、EPnWrRemain_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_H,L レジスタにはこの制限はございません。
- ・ 16bit CPU を使用の場合は、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ーブ信号を制御して下さい。詳細は「FIFO アクセスの端数処理」をご参照下さい。
- ・ EPnFIFO_Wr_H,L レジスタへのライト直後に EPnWrRemain_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_H,L レジスタへのライト直後に RAM_WrDoorAdrs_H,L レジスタを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。

6.4.3.1.4. FIFO アクセス（リード）

FIFO リードアクセスは EPnFIFO_Rd_H,L レジスタ、EPnFIFO_ByteRd レジスタの読み出しを意味します。8bit mode 動作の場合は EPnFIFO_Rd_H,L レジスタ、EPnFIFO_ByteRd レジスタのどのレジスタにアクセスを行っても同様に FIFO からの読み出しを行うことができます。

FIFO リードアクセスには、以下の制限事項があります。

- ・ EPx{x=0,a-c}.Join.JoinCPU_Rd ビットを設定した後、EPnRdRemain_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。
- ・ 16bit mode で使用の場合、ワード読みを行う場合は EPnFIFO_Rd_H,L 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 EPnFIFO_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 EPnFIFO_Rd_H,L 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「FIFO アクセスの端数処理」をご参照下さい。

6.4.3.1.5. 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。

EPnFIFO_Clr.EPx{x=a-c}.FIFO_Clr ビット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 6-25 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。

6. 機能説明

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 6-25 の(2)に示します。

以上は、正常な書き込み動作です。

○A: FIFOにデータAが存在する様

空: FIFOにデータが存在しない様

start: 書き込み開始ポイント

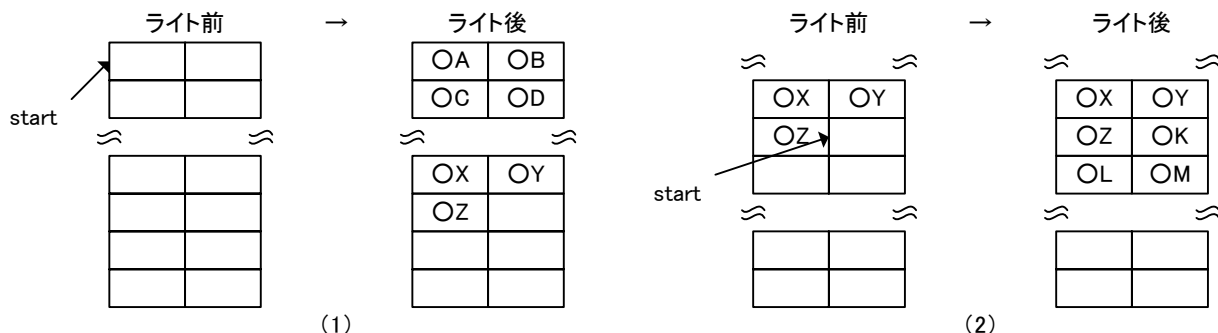


図 6-25 FIFO ライト処理(正常動作)

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 6-26 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 6-26 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 6-26 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 6-26 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。

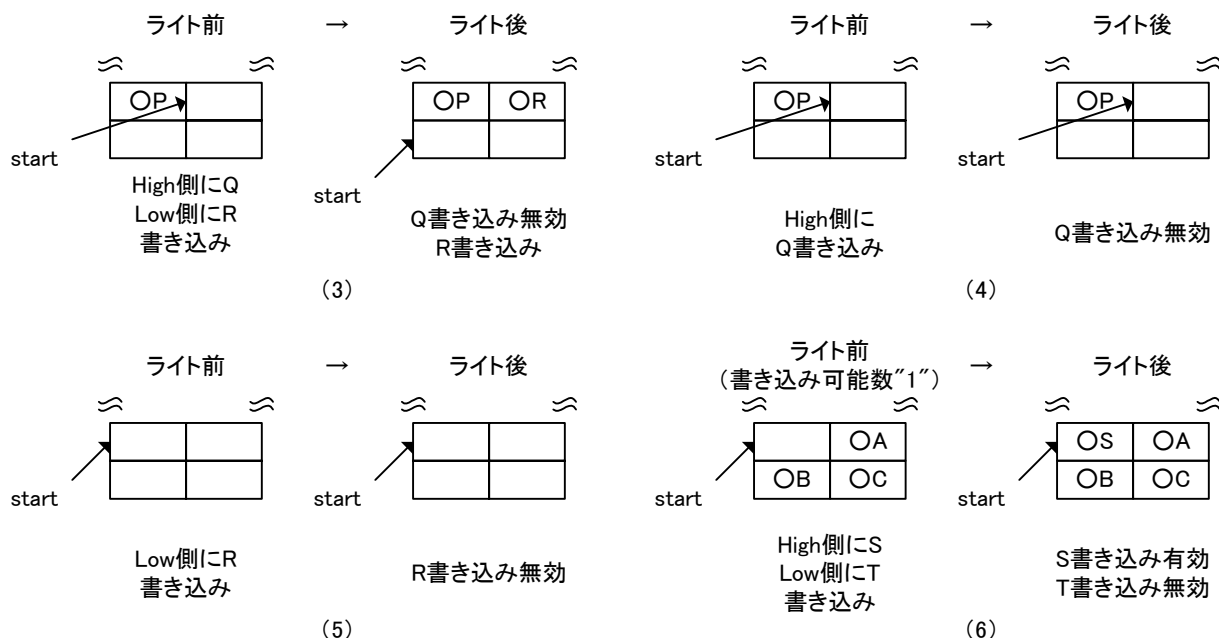


図 6-26 FIFO ライト処理(注意が必要な動作)

【リード動作】

バイト境界が無い場合は、EPnFIFO_Rd_H,L レジスタを用いたワード読み出し、EPnFIFO_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、EPnFIFO_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 6-27 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 6-27 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。

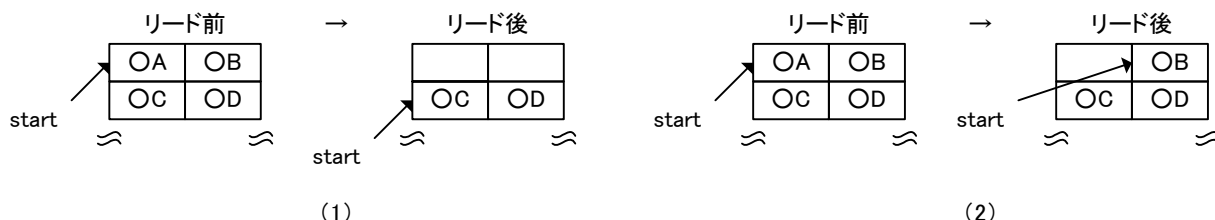


図 6-27 FIFO リード処理(正常動作)

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 6-28 の(3)はバイト境界がある状態から、EPnFIFO_Rd_H,L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 6-28 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、EPnFIFO_Rd_H,L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。

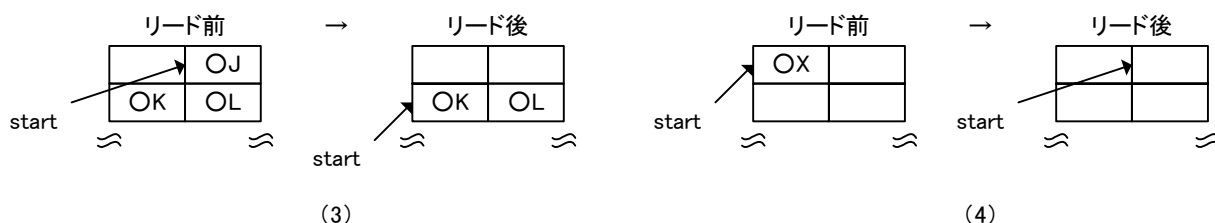


図 6-28 FIFO リード処理(注意が必要な動作)

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 EPnFIFO_Rd_H,L レジスタでワード読み、もしくは EPnFIFO_ByteRd レジスタでバイト読みする。
 - ③ 31 バイト目のデータを EPnFIFO_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 EPnFIFO_ByteRd レジスタでのバイト読みを推奨します。EPnFIFO_Rd_H,L 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 EPnFIFO_Rd_H,L レジスタでワード読み、もしくは EPnFIFO_ByteRd レジスタでバイト読みする。
- 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 EPnFIFO_Rd_H,L 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。

6. 機能説明

6.4.3.1.6. 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)期間を出カ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 FIFO へのアクセス方法(RAM_Rd)をご参照下さい。

6.4.3.1.7. 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。

6.4.3.1.8. 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。

6.4.3.2. DMA0/DMA1 (DMA ch.0 / ch.1)

6.4.3.2.1. 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 をアサートし DMA 転送が可能になります。

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。

16bit mode の場合、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できます。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 6-16 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(16bit/8bitモード動作時共)

条件	カウントモード(Count>0)			
	Count \geq Req		Count<Req	
	Ready \geq Req	Ready<Req	Ready \geq 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(16bit/8bitモード動作時共)

条件	フリーランモード	
	–	
	Ready \geq 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(16bitモード動作時)

条件	カウントモード(Count>0)		
	Count \geq Ready		Count<Ready
	Ready \geq 2	Ready<2	Ready \geq 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(16bitモード動作時)

条件	フリーランモード	
	–	
	Ready \geq 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)	–

カウントモード・ReqAssertCountオプション未使用時(8bitモード動作時)

条件	カウントモード(Count>0)		
	Count \geq Ready		Count<Ready
	Ready \geq 1	Ready<1	Ready \geq Count
XDREQ	アサート	ネゲート	アサート
転送可能数	Ready	–	Count

フリーランモード・ReqAssertCountオプション未使用時(8bitモード動作時)

条件	フリーランモード	
	–	
	Ready \geq 1	Ready<1
XDREQ	アサート	ネゲート
転送可能数	Ready	–

※ 表中の Req は DMAx{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 DMAx{x=0,1}Count_HH,HL,LH,LL の値を指しています。

6.4.3.2.2. 端子設定

ChipConfig レジスタの設定により、XDREQx{x=1,0}の論理レベル、XDACKx{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。

6.4.3.2.3. アクセスモード設定

ChipConfig.CS_Modeビット及び DMAx{x=1,0}_Config.DMA_Modeビットの設定により、DMA アクセスと見なす動作条件を設定する事が出来ます。

6. 機能説明

表 6-17 アクセスモード設定

CS_Mode	DMA_Mode	説明
0	0	XDREQ に対し XDACK がアサートされることで DMA アクセスと見なします
1	0	XDREQ に対し XDACK 及び XCS がアサートされることで DMA アクセスと見なします
*	1	XDREQ に対し DMAx{x=1,0}_Rd/WrData レジスタにアクセスされる事で DMA アクセスと見なします

以下の説明では特に断りの無い限り、XDACK 及び XCS のアサートを動作条件と見なすアクセスモードで記述しております。

6.4.3.2.4. カウントモード（ライト）

【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 に書き込み可能な空き領域(DMA_Ready)が 2 バイト以上(8bit mode の場合 1 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、(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

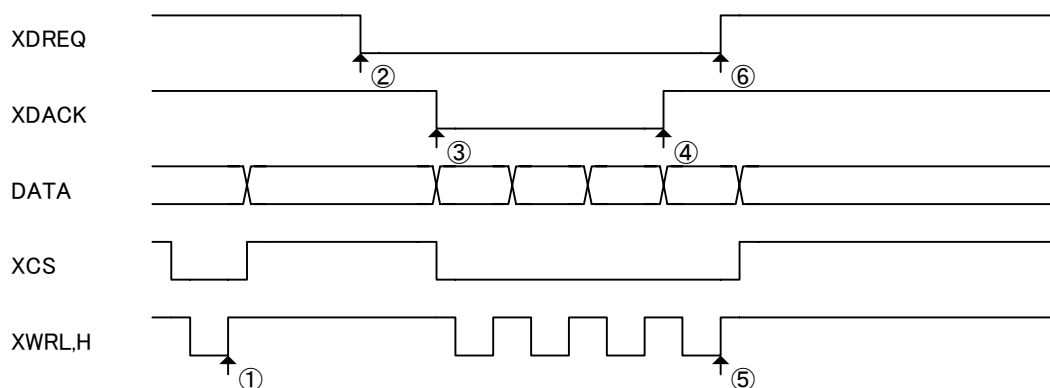
DMA 動作停止時に CPU_IntStat..DMAx{x=0,1}_Cmp ビットがセットされます。

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 6-29 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 6-29 カウントモードライトタイミング 1

6. 機能説明

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 6-30 に示します。

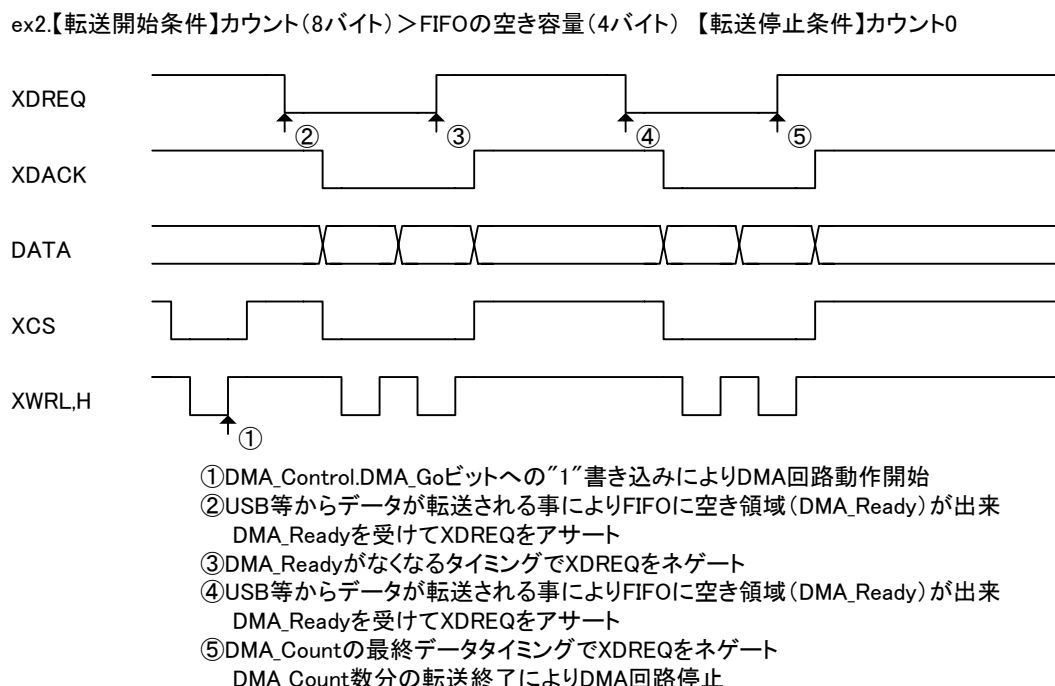


図 6-30 カウントモードライトタイミング 2

6.4.3.2.5. 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(8bit mode 動作時は 1 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。

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

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

6-31 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

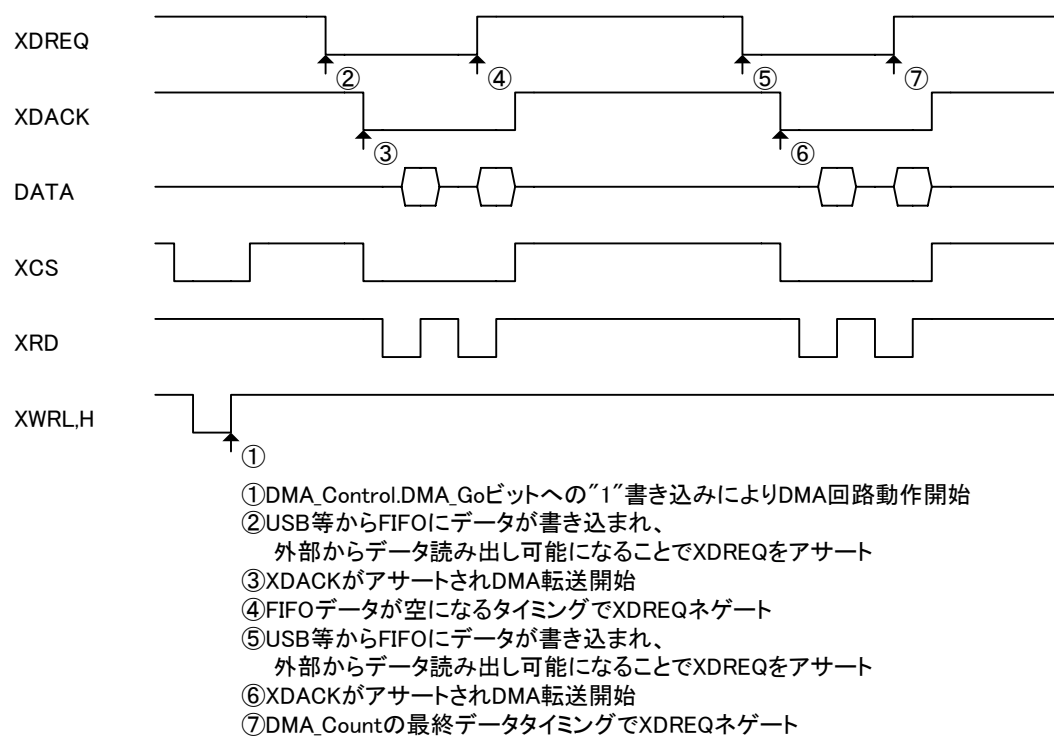


図 6-31 カウントモードリードタイミング

6. 機能説明

6.4.3.2.6. 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 (8bit mode 動作時は 1 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。転送を行う場合はカウントモードの説明をご参照ください。

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。

6.4.3.2.7. 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 (8bit mode 動作時は 1 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。転送を行う場合はカウントモードの説明をご参照ください。

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。

6.4.3.2.8. 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。

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】

動作停止条件はカウントモード、フリーランモードの説明をご参照ください。

ex.【転送開始】REQアサートカウント(8beat:16byte)

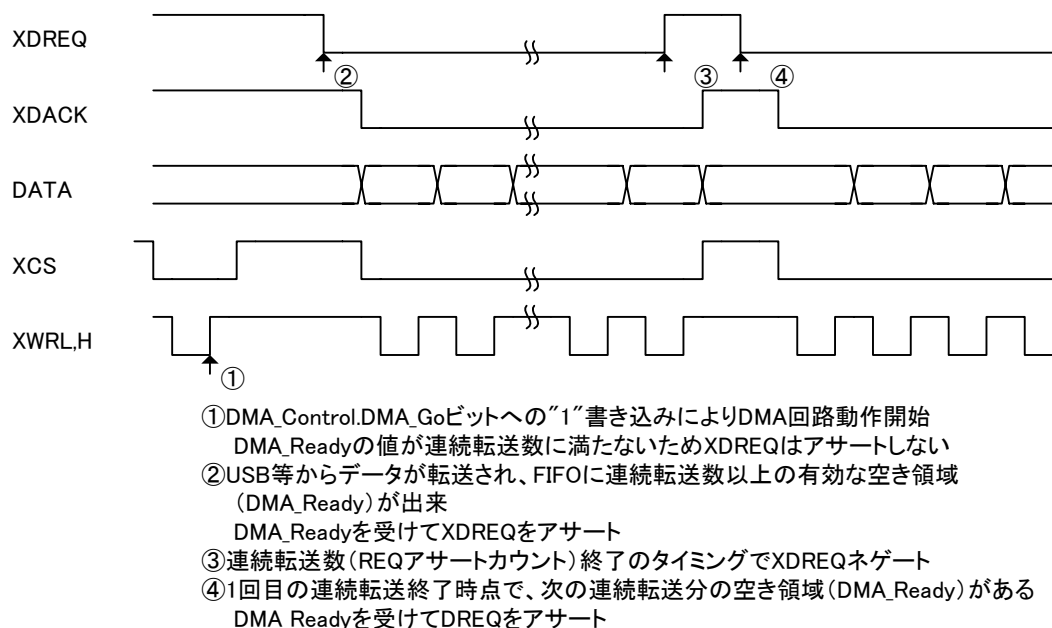


図 6-32 REQ アサートカウントオプションライトタイミング

6.4.3.2.9. 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】

動作停止条件はカウントモード、フリーランモードの説明をご参照ください。

動作タイミングは 図 6-31、図 6-32 をご参照下さい。

6.4.3.2.10. DMA の FIFO アクセス端数処理

6.4.3.1.5“FIFOアクセスの端数処理”をご参照ください。DMAIにはバイト読み出しの口はございませんのでご注意下さい。

6.5. 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。また、IDE デバイスとの通信におけるエンディアンを正しく設定するために、IDE_Config_1.Swap ビットに"1"を書き込んでください。以下では、特に断らない限り、IDE_Config_1.Swap ビットが"1"にセットされている前提で説明します。なお、IDE_Config_1.Swap ビットにつきましては、レジスタ詳細説明及び Appendix A をご参照ください。

6.5.1. 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。

6.5.1.1. 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。

6.5.1.2. 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。

6.5.1.3. 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(HDD[15:8]側)／上位(HDD[7:0]側)の順(IDE_Config_1.Swap="1"の時)にデータを 2 回書き込む必要があり、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。

6.5.1.4. 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_L 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。

6.5.2. 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。

6.5.2.1. 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。

6.5.2.2. 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。

6.5.3. 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。

6.5.3.1. 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。

6.5.3.2. 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

6. 機能説明

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。

6.5.4. 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。

6.5.4.1. 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 デバイスにはホスターミネートを通知します。

DMA 転送中にデバ이스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すが、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。

6.5.4.2. 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 デバイスにはホスターミネートを通知します。

DMA 転送中にデバ이스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すが、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。

6. 機能説明

6.5.5. 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。

6.6. バウンダリスキャン (JTAG)

バウンダリスキャン(JTAG)は、TEST 端子を L (デフォルト)の時使用可能です。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。

6.6.1. 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 6-18 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

6.6.2. 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 6-19 DEVICE_CODE

Version	1
Part Number	0x000E
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_0000000000001110_00010111110_1

となります。

6.6.3. バウンダリスキャン除外端子

本 LSI の端子のうち、DP、DM、XI、XO、VBUS、R1、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。従って、DP、DM、XO 端子は、HIGHZ インストラクションを実行することによって High-Z にはなりません。

7. レジスタ

7. レジスタ

7.1. レジスタマップ

スリープ／スヌーズ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。

Byte Addr.	Register Name	R/W	Reset	bit15 bit7	bit14 bit6	bit13 bit5	bit12 bit4	bit11 bit3	bit10 bit2	bit9 bit1	bit8 bit0
0x00	MainIntStat	R/(W)	0x00	SIE_IntStat	EPrintStat	CPU_IntStat	FIFO_IntStat	BulkIntStat	IDE_IntStat	EP0IntStat	RcvEP0SETUP
0x01	EPrintStat	R	0x00						EPcIntStat	EPbIntStat	EPaIntStat
0x02	SIE_IntStat	R/(W)	0x00	VBUS_Changed	NonJ	FinishedPM	DetectRESET	DetectSUSPEND	ChirpCmp	RestoreCmp	SetAddressCmp
0x03	CPU_IntStat	R/(W)	0x00		RcvSOF	DMA1_Countup	DMA1_Cmp			DMA0_Countup	DMA0_Cmp
0x04	FIFO_IntStat	R/(W)	0x00	DescriptorCmp	RAM_RdCmp	FIFO_IDE_Cmp	FIFO1_Cmp	FIFO_Full	FIFO_Empty		FIFO0_Cmp
0x05	BulkIntStat	R/(W)	0x00	CBW_Cmp	CBW_LengthErr	CBW_Err		CSW_Cmp	CSW_Err		
0x06	IDE_IntStat	R/(W)	0x00	IDE_RegCmp	IDE_RegErr	IDE_SeqWrRegCmp	CompleteINTRQ		IDE_Cmp	DetectINTRQ	DetectTerm
0x07	(DBG_IntStat)	R/(W)	0x00	CPU_CacheMiss	SIE_AutoNegoErr	FIFO_JoinErr		CPU_BufWrMiss			
0x08	EP0IntStat	R/(W)	0x00		OUT_ShortACK	IN_TranACK	OUT_TranACK	IN_TranNAK	OUT_TranNAK	IN_TranErr	OUT_TranErr
0x09	EPaIntStat	R/(W)	0x00		OUT_ShortACK	IN_TranACK	OUT_TranACK	IN_TranNAK	OUT_TranNAK	IN_TranErr	OUT_TranErr
0x0A	EPbIntStat	R/(W)	0x00		OUT_ShortACK	IN_TranACK	OUT_TranACK	IN_TranNAK	OUT_TranNAK	IN_TranErr	OUT_TranErr
0x0B	EPcIntStat	R/(W)	0x00		OUT_ShortACK	IN_TranACK	OUT_TranACK	IN_TranNAK	OUT_TranNAK	IN_TranErr	OUT_TranErr
0x0C			0xXX								
0x0D			0xXX								
0x0E			0xXX								
0x0F			0xXX								

Byte Addr.	Register Name	R/W	Reset	bit15 bit7	bit14 bit6	bit13 bit5	bit12 bit4	bit11 bit3	bit10 bit2	bit9 bit1	bit8 bit0
0x10	MainIntEnb	R/W	0x00	EnSIE_IntStat	EnEPrintStat	EnCPU_IntStat	EnFIFO_IntStat	EnBulkIntStat	EnIDE_IntStat	EnEP0IntStat	EnRcvEP0SETUP
0x11	EPaIntEnb	R/W	0x00						EnEPcIntStat	EnEPbIntStat	EnEPaIntStat
0x12	SIE_IntEnb	R/W	0x00	EnVBUS_Changed	EnNonJ	EnFinishedPM	EnDetectRESET	EnDetectSUSPEND	EnChirpCmp	EnRestoreCmp	EnSetAddressCmp
0x13	CPU_IntEnb	R/W	0x00		EnRcvSOF	EnDMA1_Countup	EnDMA1_Cmp			EnDMA0_Countup	EnDMA0_Cmp
0x14	FIFO_IntEnb	R/W	0x00	EnDescriptorCmp	EnRAM_RdCmp	EnFIFO_IDE_Cmp	EnFIFO1_Cmp	EnFIFO_Full	EnFIFO_Empty		EnFIFO0_Cmp
0x15	BulkIntEnb	R/W	0x00	EnCBW_Cmp	EnCBW_LengthErr	EnCBW_Err		EnCSW_Cmp	EnCSW_Err		
0x16	IDE_IntEnb	R/W	0x00	EnIDE_RegCmp	EnIDE_RegErr	EnSeqWrRegCmp	EnCompleteINTRQ		EnIDE_Cmp	EnDetectINTRQ	EnDetectTerm
0x17			0xXX								
0x18	EP0IntEnb	R/W	0x00		EnOUT_ShortACK	EnIN_TranACK	EnOUT_TranACK	EnIN_TranNAK	EnOUT_TranNAK	EnIN_TranErr	EnOUT_TranErr
0x19	EPaIntEnb	R/W	0x00		EnOUT_ShortACK	EnIN_TranACK	EnOUT_TranACK	EnIN_TranNAK	EnOUT_TranNAK	EnIN_TranErr	EnOUT_TranErr
0x1A	EPbIntEnb	R/W	0x00		EnOUT_ShortACK	EnIN_TranACK	EnOUT_TranACK	EnIN_TranNAK	EnOUT_TranNAK	EnIN_TranErr	EnOUT_TranErr
0x1B	EPcIntEnb	R/W	0x00		EnOUT_ShortACK	EnIN_TranACK	EnOUT_TranACK	EnIN_TranNAK	EnOUT_TranNAK	EnIN_TranErr	EnOUT_TranErr
0x1C			0xXX								
0x1D			0xXX								
0x1E			0xXX								
0x1F			0xXX								

7. レジスタ

スリープ／スヌーズ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。

Byte Addr.	Register Name	R/W	Reset	bit15 bit7	bit14 bit6	bit13 bit5	bit12 bit4	bit11 bit3	bit10 bit2	bit9 bit1	bit8 bit0
0x20	RevisionNum	R	0x30	Revision Number							
0x21	ChipReset	W	0xXX	ResetUTM						—	AllReset
0x22	PM_Control	R/W	0x00		PM_State [2:0]			GoSLEEP	GoSNOOZE	GoActive60	GoActiveAL
0x23	USB_Control	R/W	0x00	DisBusDetect	EnAutoNego	InSUSPEND	DisableHS	SendWakeup	RestoreUSB	GoChirp	ActiveUSB
0x24	USB_Status	R	0xXX	VBUS	FSxHS					LineState [1:0]	
0x25	XcvtControl	R/W	0x41	TermSelect	XcvtSelect					OpMode [1:0]	
0x26	USB_Test	R/W	0x00	EnHS_Test				Test_SE0_NAK	Test_J	Test_K	Test_Packet
0x27			0xXX								
0x28	EPnControl	W	0xXX	AllForceNAK	EPrForceSTALL	AllFIFO_Clr					EP0FIFO_Clr
0x29	EPrFIFO_Clr	W	0xXX						EPcFIFO_Clr	EPbFIFO_Clr	EPaFIFO_Clr
0x2A	ClrAllJoin	W	0xXX	ClrJoinIDE	ClrJoin_FIFOStatus	ClrJoinDMA0_Rd	ClrJoinDMA0_Wr	ClrJoinDMA1_Rd	ClrJoinDMA1_Wr	ClrJoinCPU_Rd	ClrJoinCPU_Wr
0x2B			0xXX								
0x2C	BulkOnlyControl	R/W	0x00	AutoForceNAK_CBW					GoCBW_Mode	GoCSW_Mode	
0x2D	BulkOnlyConfig	R/W	0x00	(Reserved)	(Reserved)	(Reserved)	(Reserved)	(Reserved)	EPcBulkOnly	EPbBulkOnly	(Reserved)
0x2E	WakeupTim_H	R/W	0x00	WakeupTim [15:8]							
0x2F	WakeupTim_L	R/W	0x00	WakeupTim [7:0]							

Byte	Register	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Addr.	Name			bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x30	EP0SETUP_0	R	0x00	SETUP 0							
0x31	EP0SETUP_1	R	0x00	SETUP 1							
0x32	EP0SETUP_2	R	0x00	SETUP 2							
0x33	EP0SETUP_3	R	0x00	SETUP 3							
0x34	EP0SETUP_4	R	0x00	SETUP 4							
0x35	EP0SETUP_5	R	0x00	SETUP 5							
0x36	EP0SETUP_6	R	0x00	SETUP 6							
0x37	EP0SETUP_7	R	0x00	SETUP 7							
0x38	USB_Address	R(W)	0x00	SetAddress	USB_Address [6:0]						
0x39			0xXX								
0x3A	SETUP_Control	R/W	0x00								ProtectEP0
0x3B			0xXX								
0x3C			0xXX								
0x3D			0xXX								
0x3E	FrameNumber_H	R	0x80	FN_Invalid					FrameNumber [10:8]		
0x3F	FrameNumber_L	R	0x00	FrameNumber [7:0]							

7. レジスタ

スリープ／スヌーズ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。

Byte Addr.	Register Name	R/W	Reset	bit15 bit7	bit14 bit6	bit13 bit5	bit12 bit4	bit11 bit3	bit10 bit2	bit9 bit1	bit8 bit0
0x40	EP0MaxSize	R/W	0x40		EP0MaxSize[6:3]						
0x41	EP0Control	R/W	0x00	INxOUT							ReplyDescript or
0x42	EP0ControlIN	R/W	0x00		EnShortPkt		ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL
0x43	EP0ControlOUT	R/W	0x00	AutoForceNAK			ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL
0x44			0xXX								
0x45	EP0Join	R/W	0x00			JoinDMA0_Rd	JoinDMA0_Wr	JoinDMA1_Rd	JoinDMA1_Wr	JoinCPU_Rd	JoinCPU_Wr
0x46			0xXX								
0x47			0xXX								
0x48			0xXX								
0x49			0xXX								
0x4A			0xXX								
0x4B			0xXX								
0x4C			0xXX								
0x4D			0xXX								
0x4E			0xXX								
0x4F			0xXX								

Byte Addr.	Register Name	R/W	Reset	bit15 bit7	bit14 bit6	bit13 bit5	bit12 bit4	bit11 bit3	bit10 bit2	bit9 bit1	bit8 bit0
0x50			0xXX								
0x51	EPaMaxSize_L	R/W	0x00		EPaMaxSize [6:3]						
0x52	EPaConfig_0	R/W	0x00	INxOUT	IntEP_Mode	EnEndpoint		EndpointNumber[3:0]			
0x53			0xXX								
0x54	EPaControl	R/W	0x00	AutoForceNAK	EnShortPkt	DisAF_NAK_Short	ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL
0x55	EPaJoin	R/W	0x00	JoinIDE	JoinFIFO_Stat	JoinDMA0_Rd	JoinDMA0_Wr	JoinDMA1_Rd	JoinDMA1_Wr	JoinCPU_Rd	JoinCPU_Wr
0x56			0xXX								
0x57			0xXX								
0x58	EPbMaxSize_H	R/W	0x00							EPbMaxSize[9:8]	
0x59	EPbMaxSize_L	R/W	0x00	EPbMaxSize[7:3]							
0x5A	EPbConfig_0	R/W	0x00	INxOUT	IntEP_Mode	EnEndpoint		EndpointNumber[3:0]			
0x5B			0xXX								
0x5C	EPbControl	R/W	0x00	AutoForceNAK	EnShortPkt	DisAF_NAK_Short	ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL
0x5D	EPbJoin	R/W	0x00	JoinIDE	JoinFIFO_Stat	JoinDMA0_Rd	JoinDMA0_Wr	JoinDMA1_Rd	JoinDMA1_Wr	JoinCPU_Rd	JoinCPU_Wr
0x5E			0xXX								
0x5F			0xXX								

7. レジスタ

スリープ／スヌーズ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。

Byte Addr.	Register Name	R/W	Reset	bit15 bit7	bit14 bit6	bit13 bit5	bit12 bit4	bit11 bit3	bit10 bit2	bit9 bit1	bit8 bit0		
0x60	EPcMaxSize_H	R/W	0x00							EPcMaxSize[9:8]			
0x61	EPcMaxSize_L	R/W	0x00	EPcMaxSize[7:3]									
0x62	EPcConfig_0	R/W	0x00	INxOUT	IntEP_Mode	EnEndpoint		EndpointNumber[3:0]					
0x63			0xXX										
0x64	EPcControl	R/W	0x00	AutoForceNAK	EnShortPkt	DisAF_NAK_Short	ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL		
0x65	EPcJoin	R/W	0x00	JoinIDE	JoinFIFO_Stat	JoinDMA0_Rd	JoinDMA0_Wr	JoinDMA1_Rd	JoinDMA1_Wr	JoinCPU_Rd	JoinCPU_Wr		
0x66			0xXX										
0x67			0xXX										
0x68			0xXX										
0x69			0xXX										
0x6A			0xXX										
0x6B			0xXX										
0x6C	RAM_WrAdrs_H	R/W	0x00					RAM_Adrs[11:8]					
0x6D	RAM_WrAdrs_L	R/W	0x00	RAM_Adrs[7:0]									
0x6E	RAM_WrDoor_H	W	0xXX	RAM_Door[15:8]									
0x6F	RAM_WrDoor_L	W	0xXX	RAM_Door[7:0]									

Byte Addr.	Register Name	R/W	Reset	bit15 bit7	bit14 bit6	bit13 bit5	bit12 bit4	bit11 bit3	bit10 bit2	bit9 bit1	bit8 bit0
0x70	EPnFIFO_Rd_H	R	0xXX	EPnFIFO_Rd [15:8]							
0x71	EPnFIFO_Rd_L	R	0xXX	EPnFIFO_Rd [7:0]							
0x72	EPnFIFO_Wr_H	W	0xXX	EPnFIFO_Wr [15:8]							
0x73	EPnFIFO_Wr_L	W	0xXX	EPnFIFO_Wr [7:0]							
0x74	EPnRdRemain_H	R	0x00	RdRemainValid			EPnRdRemain [12:8]				
0x75	EPnRdRemain_L	R	0x00	EPnRdRemain [7:0]							
0x76	EPnWrRemain_H	R	0x00				EPnWrRemain [12:8]				
0x77	EPnWrRemain_L	R	0x00	EPnWrRemain [7:0]							
0x78	DescAdrs_H	R/W	0x00					DescAdrs[11:8]			
0x79	DescAdrs_L	R/W	0x00	DescAdrs [7:0]							
0x7A	DescSize_H	R/W	0x00							DescSize [9:8]	
0x7B	DescSize_L	R/W	0x00	DescSize [7:0]							
0x7C	EPnFIFO_ByteRd	R	0xXX	EPnFIFO_ByteRd [7:0]							
0x7D			0xXX								
0x7E			0xXX								
0x7F			0xXX								

7. レジスタ

スリープ／スヌーズ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。

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x80	DMA0_FIFO_Control	R/W	0x00	FIFO_Running	AutoEnShort						
0x81	DMA0_Config	R/W	0x00	FreeRun	DMA_Mode			ActiveDMA		ReqAssertCount [1:0]	
0x82	DMA0_Control	R/W	0x00	DMA_Running			CounterClr			DMA_Stop	DMA_Go
0x83			0xFF								
0x84	DMA0_Remain_H	R	0x00				DMA_Remain [12:8]				
0x85	DMA0_Remain_L	R	0x00	DMA_Remain [7:0]							
0x86			0xFF								
0x87			0xFF								
0x88	DMA0_Count_HH	R/W	0x00	DMA_Count [31:24]							
0x89	DMA0_Count_HL	R/W	0x00	DMA_Count [23:16]							
0x8A	DMA0_Count_LH	R/W	0x00	DMA_Count [15:8]							
0x8B	DMA0_Count_LL	R/W	0x00	DMA_Count [7:0]							
0x8C	DMA0_RdData_H	R	0xFF	DMA_RdData[15:8]							
0x8D	DMA0_RdData_L	R	0xFF	DMA_RdData[7:0]							
0x8E	DMA0_WrData_H	W	0xFF	DMA_WrData[15:8]							
0x8F	DMA0_WrData_L	W	0xFF	DMA_WrData[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
0x90	DMA1_FIFO_Control	R/W	0x00	FIFO_Running	AutoEnShort						
0x91	DMA1_Config	R/W	0x00	FreeRun	DMA_Mode			ActiveDMA		ReqAssertCount [1:0]	
0x92	DMA1_Control	R/W	0x00	DMA_Running			CounterClr			DMA_Stop	DMA_Go
0x93			0xFF								
0x94	DMA1_Remain_H	R	0x00				DMA_Remain [12:8]				
0x95	DMA1_Remain_L	R	0x00	DMA_Remain [7:0]							
0x96			0xFF								
0x97			0xFF								
0x98	DMA1_Count_HH	R/W	0x00	DMA_Count [31:24]							
0x99	DMA1_Count_HL	R/W	0x00	DMA_Count [23:16]							
0x9A	DMA1_Count_LH	R/W	0x00	DMA_Count [15:8]							
0x9B	DMA1_Count_LL	R/W	0x00	DMA_Count [7:0]							
0x9C	DMA1_RdData_H	R	0xFF	DMA_RdData[15:0]							
0x9D	DMA1_RdData_L	R	0xFF	DMA_RdData[7:0]							
0x9E	DMA1_WrData_H	W	0xFF	DMA_WrData[15:8]							
0x9F	DMA1_WrData_L	W	0xFF	DMA_WrData[7:0]							

7. レジスタ

スリープ／スヌーズ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。

Byte Addr.	Register Name	R/W	Reset	bit15 bit7	bit14 bit6	bit13 bit5	bit12 bit4	bit11 bit3	bit10 bit2	bit9 bit1	bit8 bit0
0xA0	IDE_Status	R/W	0x00	DMARQ	DMACK	INTRQ	IORDY			PDIAG	DASP
0xA1	IDE_Control	R/W	0x00		IDE_Clr						IDE_Go
0xA2	IDE_Config_0	R/W	0x00	IDE_BusReset	IDE_LongBusReset					Ultra	DMA
0xA3	IDE_Config_1	R/W	0x00	ActiveIDE	DelayStrobe		InterLock		Swap		
0xA4	IDE_Rmod	R/W	0x00	RegisterAssertPulseWidth[3:0]				RegisterNegatePulseWidth[3:0]			
0xA5	IDE_Tmod	R/W	0x00	TransferAssertPulseWidth[3:0]				TransferNegatePulseWidth[3:0]			
0xA6	IDE_Umod	R/W	0x00					UltraDMA_Cycle[3:0]			
0xA7			0xFF								
0xA8			0xFF								
0xA9			0xFF								
0xAA	IDE_CRC_H	R/W	0x00	IDE_CRC[15:8]							
0xAB	IDE_CRC_L	R/W	0x00	IDE_CRC[7:0]							
0xAC			0xFF								
0xAD	IDE_Count_H	R/W	0x00	IDE_Count[23:16]							
0xAE	IDE_Count_M	R/W	0x00	IDE_Count[15:8]							
0xAF	IDE_Count_L	R/W	0x00	IDE_Count[7:1]							

Byte Addr.	Register Name	R/W	Reset	bit15 bit7	bit14 bit6	bit13 bit5	bit12 bit4	bit11 bit3	bit10 bit2	bit9 bit1	bit8 bit0
0xB0	IDE_RegAdrs	R/W	0x00	IDE_WrReg	IDE_RdReg			IDE_RegAddress[3:0]			
0xB1			0xFF								
0xB2	IDE_RdRegValue_H	R	0x00	IDE_RdRegValue[15:8]							
0xB3	IDE_RdRegValue_L	R	0x00	IDE_RdRegValue[7:0]							
0xB4	IDE_WrRegValue_H	R/W	0x00	IDE_WrRegValue[15:8]							
0xB5	IDE_WrRegValue_L	R/W	0x00	IDE_WrRegValue[7:0]							
0xB6	IDE_SeqWrRegControl	R/W	0x00	IDE_SeqWrReg	IDE_SeqWrRegClr						
0xB7	IDE_SeqWrRegCnt	R	0x00				IDE_SeqWrRegCnt[4:0]				
0xB8	IDE_SeqWrRegAdrs	W	0xFF					IDE_SeqRegAddress[3:0]			
0xB9	IDE_SeqWrRegValue	W	0xFF	IDE_SeqWrRegValue[7:0]							
0xBA			0xFF								
0xBB			0xFF								
0xBC	IDE_RegConfig	R/W	0x00	EnAutoStatusRd							
0xBD			0xFF								
0xBE			0xFF								
0xBF			0xFF								

7. レジスタ

スリープ／スヌーズ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。

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0xC0	RAM_Rd_00	R	0x00								
0xC1	RAM_Rd_01	R	0x00								
0xC2	RAM_Rd_02	R	0x00								
0xC3	RAM_Rd_03	R	0x00								
0xC4	RAM_Rd_04	R	0x00								
0xC5	RAM_Rd_05	R	0x00								
0xC6	RAM_Rd_06	R	0x00								
0xC7	RAM_Rd_07	R	0x00								
0xC8	RAM_Rd_08	R	0x00								
0xC9	RAM_Rd_09	R	0x00								
0xCA	RAM_Rd_0A	R	0x00								
0xCB	RAM_Rd_0B	R	0x00								
0xCC	RAM_Rd_0C	R	0x00								
0xCD	RAM_Rd_0D	R	0x00								
0xCE	RAM_Rd_0E	R	0x00								
0xCF	RAM_Rd_0F	R	0x00								

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0xD0	RAM_Rd_10	R	0x00								
0xD1	RAM_Rd_11	R	0x00								
0xD2	RAM_Rd_12	R	0x00								
0xD3	RAM_Rd_13	R	0x00								
0xD4	RAM_Rd_14	R	0x00								
0xD5	RAM_Rd_15	R	0x00								
0xD6	RAM_Rd_16	R	0x00								
0xD7	RAM_Rd_17	R	0x00								
0xD8	RAM_Rd_18	R	0x00								
0xD9	RAM_Rd_19	R	0x00								
0xDA	RAM_Rd_1A	R	0x00								
0xDB	RAM_Rd_1B	R	0x00								
0xDC	RAM_Rd_1C	R	0x00								
0xDD	RAM_Rd_1D	R	0x00								
0xDE	RAM_Rd_1E	R	0x00								
0xDF	RAM_Rd_1F	R	0x00								

7. レジスタ

スリープ／スヌーズ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。

Byte Addr.	Register Name	R/W	Reset	bit15 bit7	bit14 bit6	bit13 bit5	bit12 bit4	bit11 bit3	bit10 bit2	bit9 bit1	bit8 bit0
0xE0	RAM_RdAdrs_H	R/W	0x00					RAM_RdAdrs[11:8]			
0xE1	RAM_RdAdrs_L	R/W	0x00	RAM_RdAdrs[7:2]							
0xE2	RAM_RdControl	R/W	0x00	RAM_GoRdCBW	RAM_GoRd						
0xE3	RAM_RdCount	R/W	0x00			RAM_RdCount[5:2]					
0xE4			0xXX								
0xE5			0xXX								
0xE6			0xXX								
0xE7			0xXX								
0xE8			0xXX								
0xE9			0xXX								
0xEA			0xXX								
0xEB	ModeProtect	R/W	0x56	<i>Protected[7:0](56 以外を書くとプロテクト、0x56 で解除)</i>							
0xEC			0xXX								
0xED	ClkSelect	R/W	0x01	<i>xActIDE_Term</i>							<i>ClkSelect</i>
0xEE			0xXX								
0xEF	ChipConfig	R/W	0x00	<i>IntLevel</i>	<i>IntMode</i>	<i>DREQ_Level</i>	<i>DACK_Level</i>	<i>CS_Mode</i>	<i>CPU_Swap</i>	<i>BusMode</i>	<i>Bus8x16</i>

Byte Addr.	Register Name	R/W	Reset	bit15 bit7	bit14 bit6	bit13 bit5	bit12 bit4	bit11 bit3	bit10 bit2	bit9 bit1	bit8 bit0
0xF0	(Reserved)	R/W	0xXX								
0xF1	D_ModeControl	W	0xXX	(Reserved)	(Reserved)	(Reserved)	SetAddressMode	(Reserved)	(Reserved)	(Reserved)	(Reserved)
0xF2	(Reserved)	R/W	0xXX								
0xF3	(Reserved)	R/W	0xXX								
0xF4	(Reserved)	R/W	0xXX								
0xF5	(Reserved)	R/W	0xXX								
0xF6	(Reserved)	R/W	0xXX								
0xF7	(Reserved)	R/W	0xXX								
0xF8	(Reserved)	R/W	0xXX								
0xF9	(Reserved)	R/W	0xXX								
0xFA	(Reserved)	R/W	0xXX								
0xFB	(Reserved)	R/W	0xXX								
0xFE	(Reserved)	R/W	0xXX								
0xFD	(Reserved)	R/W	0xXX								
0xFE	(Reserved)	R/W	0xXX								
0xFF	(Reserved)	R/W	0xXX								

0xF1 D_ModeControl レジスタの詳細は、Appendix C を参照して下さい。

7. レジスタ

7.2. レジスタ 詳細説明

7.2.1. 00h MainIntStat (Main Interrupt Status)

Address	Register Name	R / W	Bit Symbol	Description		Reset
00h	MainIntStat	R	7: SIE_IntStat	0: None	1: SIE Interrupts	00h
		R	6: EPrIntStat	0: None	1: EPr Interrupts	
		R	5: CPU_IntStat	0: None	1: CPU Interrupts	
		R	4: FIFO_IntStat	0: None	1: FIFO Interrupts	
		R	3: BulkIntStat	0: None	1: Bulk Interrupts	
		R	2: IDE_IntStat	0: None	1: IDE Interrupts	
		R	1: EP0IntStat	0: None	1: EP0 Interrupts	
		R (W)	0: RcvEP0SETUP	0: None	1: Receive EP0 SETUP	

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 SIE_IntStat

割り込み要因を間接指示します。

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 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、スリープ／スヌーズ中もリード有効です。

Bit6 EPrIntStat

割り込み要因を間接指示します。

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 EPrIntEnb 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。

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 FIFO_IntStat

割り込み要因を間接指示します。

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 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。

Bit3 BulkIntStat

割り込み要因を間接指示します。

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 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。

Bit2 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。

Bit1 EP0IntStat

割り込み要因を間接指示します。

EP0_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 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 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EP0Setup_0～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 EP0ControlIN,EP0ControlOUT レジスタの ForceSTALL ビットが"0"に EP0ControlIN,EP0ControlOUT レジスタの ForceNAK ビット、ToggleStat ビット、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。

7. レジスタ

7.2.2. 01h EPrIntStat (EPr Interrupt Status)

Address	Register Name	R / W	Bit Symbol	Description		Reset
01h	EPrIntStat		7:	0:	1:	00h
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
		R	2: EPcIntStat	0: None	1: EPc Interrupt	
		R	1: EPbIntStat	0: None	1: EPb Interrupt	
		R	0: 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。

Bit7-3 Reserved

Bit2 EPcIntStat

割り込み要因を間接指示します。

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 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 EPbIntStat

割り込み要因を間接指示します。

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 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 EPaIntStat

割り込み要因を間接指示します。

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 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。

7.2.3. 02h SIE_IntStat (SIE Interrupt Status)

Address	Register Name	R / W	Bit Symbol	Description	Reset
02h	SIE_IntStat	R (W)	7: VBUS_Changed	0: None	1: VBUS is Changed
		R (W)	6: NonJ	0: None	1: Detect Non J state
		R (W)	5: FinishedPM	0: None	1: Detect FinishedPM
		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

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 VBUS_Changed

割り込み要因を直接指示します。

VBUS 端子の状態が変化した時に "1" にセットされます。

USB_Statusレジスタの VBUSビットによって VBUS 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。このビットはスリープ／スヌーズ中も有効です。

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 がスヌーズ状態 (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 FinishedPM

割り込み要因を直接指示します。

PM_Control レジスタで、GoSLEEP,GoSNOOZE,GoActive60,GoActiveALL 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。

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、USB_Control 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、USB_Control 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下さい。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下さい。

リセット検出時、USB_Control 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、USB_Control レジスタの EnAutoNego ビットの項を参照して下さい。

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／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、PMControl レジスタの Snooze 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。

7. レジスタ

Bit2 ChirpCmp

割り込み要因を直接指示します。

USB_Control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。

USB_Control 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(USB_Address レジスタ参照)が、そのコントロール転送の処理を自動的に行います。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 USB_Address レジスタにアドレスがセットされます。

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。

ACTIVE60 ステートに入る際には、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。

＜ACTIVE60 ステートに入るとき＞

- 1) 割り込みステータスを処理し、クリアする(SIE_IntStat.Bit4～0)
- 2) 割り込みステータスをディスエーブルにする(SIE_IntEnb.Bit4～0)

＜ACTIVE60 ステートから出るとき＞

- 3) 割り込みステータスをクリアする(SIE_IntStat.Bit4～0)
- 4) 割り込みステータスをイネーブルにする(SIE_IntEnb.Bit4～0)

7. レジスタ

7.2.4. 03h CPU_IntStat (CPU Interrupt Status)

Address	Register Name	R / W	Bit Symbol	Description		Reset
03h	CPU_IntStat		7:	0:	1:	00h
		R (W)	6: RcvSOF	0: None	1: Receive SOF Token	
		R (W)	5: DMA1_Countup	0: None	1: DMA1 Counter Overflow	
		R (W)	4: DMA1_Cmp	0: None	1: DMA1 Complete	
			3:	0:	1:	
			2:	0:	1:	
		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 **Reserved**

Bit6 **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。

Bit5 **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。

Bit4 **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。

Bit3-2 **Reserved**

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。

7.2.5. 04h FIFO_IntStat (FIFO Interrupt Status)

Address	Register Name	R / W	Bit Symbol	Description	Reset
04h	FIFOIntStat	R (W)	7: DescriptorCmp	0: None	1: Descriptor Complete
		R (W)	6: RAM_RdCmp	0: None	1: RAM Read Complete
		R (W)	5: FIFO_IDE_Cmp	0: None	1: FIFO-IDE Complete
		R (W)	4: FIFO1_Cmp	0: None	1: FIFO1 Complete
		R (W)	3: FIFO_Full	0: None	1: Selected FIFO is Full
		R (W)	2: FIFO_Empty	0: None	1: Selected FIFO is Empty
			1:	0:	1:
		R (W)	0: FIFO0_Cmp	0: None	1: FIFO0 Complete

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 EP0IntStat レジスタの OUT_TransNAK ビットと共に、"1"にセットされます。

Bit6 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。

Bit5 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。

Bit4 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。

Bit3 FIFO_Full

割り込み要因を直接指示します。

EPx{x=a,b,c}Join.JoinFIFO_Stat によって選択されたエンドポイントの FIFO が満状態である時に、"1"にセットされます。

Bit2 FIFO_Empty

割り込み要因を直接指示します。

EPx{x=a,b,c}Join.JoinFIFO_Stat によって選択されたエンドポイントの FIFO が空状態である時に、"1"にセットされます。

Bit1 Reserved**Bit0 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。

7. レジスタ

7.2.6. 05h BulkIntStat (Bulk Interrupt Status)

Address	Register Name	R / W	Bit Symbol	Description		Reset
05h	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

7.2.7. 06h IDE_IntStat (IDE Interrupt Status)

Address	Register Name	R / W	Bit Symbol	Description	Reset
06h	IDE_IntStat	R (W)	7: IDE_RegCmp	0: None	00h
		R (W)	6: IDE_RegErr	0: None	
		R (W)	5: IDE_SeqWrRegCmp	0: None	
		R (W)	4: CompleteINTRQ	0: None	
			3:	0:	
		R (W)	2: IDE_Cmp	0: None	
		R (W)	1: DetectINTRQ	0: None	
		R (W)	0: DetectTerm	0: None	

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 Reseved**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 信号の立ち上がりが出検された時に"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。

7. レジスタ

7.2.8. 07h DBG_IntStat (DEBUG Interrupt Status)

Address	Register Name	R / W	Bit Symbol	Description		Reset
07h	DBG_IntStat	R	7: CPU_CacheMiss	0: None	1: Detected CPU Cashe Miss	00h
		R (W)	6: SIE_AutoNegoErr	0: None	1: Detected AutoNego Error	
		R (W)	5: FIFO_JoinErr	0: None	1: Detected FIFO Join Error	
			4:	0:	1:	
		R	3: CPU_BufWrMiss	0: None	1: Detected CPU BufWr Error	
			2:	0:	1:	
			1:	0:	1:	
			0:	0:	1:	

デバック用レジスタの割り込みステータスを表示します。割り込み要因を直接指示しているビットは"1"を書き込む事で割り込み要因をクリアできます。

Bit7 CPU_CacheMiss

割り込み要因を間接指示します。

CPUIF の REG/DMA0/DMA1 ブロックの何れかが、キャッシュ動作を完了していない状態でリードされた時に、このビットが"1" にセットされます。

ソフトウェアのデバッグ用ビットです。

Bit6 SIE_AutoNegoErr

割り込み要因を直接指示します。

オート・ネゴシエーション機能のシーケンスでエラーがあった場合に、"1"にセットされます。

ハードウェアのデバッグ用ビットです。

Bit5 FIFO_JoinErr

割り込み要因を直接指示します。

EPx{x=0,a-h}Join レジスタにてエンドポイントへのアクセス権を設定する際に排他処理違反をしたときに"1"にセットされます。

ソフトウェアのデバッグ用ビットです。

Bit4 Reserved

Bit3 CPU_BufWrMiss

割り込み要因を間接指示します。

CPUIF の REG/DMA0/DMA1 ブロックの何れかが、FIFO への書き込み動作を完了していない状態でライトされた時に、このビットが"1" にセットされます。

ハードウェアのデバッグ用ビットです。

Bit2-0 Reserved

7.2.9. 08h EP0IntStat (EP0 Interrupt Status)

Address	Register Name	R / W	Bit Symbol	Description	Reset
08h	EP0IntStat		7:	0:	00h
		R (W)	6: OUT_ShortACK	0: None 1: OUT Short-Packet ACK	
		R (W)	5: IN_TranACK	0: None 1: IN Transaction ACK	
		R (W)	4: OUT_TranACK	0: None 1: OUT Transaction ACK	
		R (W)	3: IN_TranNAK	0: None 1: IN Transaction NAK	
		R (W)	2: OUT_TranNAK	0: None 1: OUT Transaction NAK	
		R (W)	1: IN_TranErr	0: None 1: IN Transaction Error	
		R (W)	0: OUT_TranErr	0: None 1: OUT Transaction Error	

エンドポイント EP0 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_TranACK と同時に"1" にセットされます。

Bit5 **IN_TranACK**

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_TranACK**

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_TranNAK**

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_TranNAK**

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_TranErr**

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_TranErr**

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。

7. レジスタ

7.2.10. 09h EPaintStat (EPa Interrupt Status)

Address	Register Name	R / W	Bit Symbol	Description	Reset
09h	EPaintStat		7:	0:	1:
		R (W)	6: OUT_ShortACK	0: None	1: OUT Short Packet ACK
		R (W)	5: IN_TranACK	0: None	1: IN Transaction ACK
		R (W)	4: OUT_TranACK	0: None	1: OUT Transaction ACK
		R (W)	3: IN_TranNAK	0: None	1: IN Transaction NAK
		R (W)	2: OUT_TranNAK	0: None	1: OUT Transaction NAK
		R (W)	1: IN_TranErr	0: None	1: IN Transaction Error
		R (W)	0: OUT_TranErr	0: None	1: OUT Transaction Error

エンドポイント EPa 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_TranACK と同時に"1" にセットされます。

Bit5 **IN_TranACK**

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_TranACK**

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_TranNAK**

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_TranNAK**

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_TranErr**

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_TranErr**

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。

7.2.11. 0Ah EPbIntStat (EPb Interrupt Status)

Address	Register Name	R / W	Bit Symbol	Description	Reset
0Ah	EPbIntStat		7:	0:	1:
		R (W)	6: OUT_ShortACK	0: None	1: OUT Short Packet ACK
		R (W)	5: IN_TranACK	0: None	1: IN Transaction ACK
		R (W)	4: OUT_TranACK	0: None	1: OUT Transaction ACK
		R (W)	3: IN_TranNAK	0: None	1: IN Transaction NAK
		R (W)	2: OUT_TranNAK	0: None	1: OUT Transaction NAK
		R (W)	1: IN_TranErr	0: None	1: IN Transaction Error
		R (W)	0: OUT_TranErr	0: None	1: OUT Transaction Error

エンドポイント EPb 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_TranACK と同時に"1" にセットされます。

Bit5 **IN_TranACK**

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_TranACK**

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_TranNAK**

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_TranNAK**

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_TranErr**

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_TranErr**

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。

7. レジスタ

7.2.12. 0Bh EPcIntStat (EPc Interrupt Status)

Address	Register Name	R / W	Bit Symbol	Description		Reset
0Bh	EPcIntStat		7:	0:	1:	00h
		R (W)	6: OUT_ShortACK	0: None	1: OUT Short Packet ACK	
		R (W)	5: IN_TrانACK	0: None	1: IN Transaction ACK	
		R (W)	4: OUT_TrانACK	0: None	1: OUT Transaction ACK	
		R (W)	3: IN_TrانNAK	0: None	1: IN Transaction NAK	
		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 EPc 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。

7.2.13. 0Ch~0Fh Reserved ()

Address	Register Name	R / W	Bit Symbol	Description		Reset
0Ch -0Fh	Reserved		7:	0:	1:	XXh
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
			0:	0:	1:	

7. レジスタ

7.2.14. 10h *MainIntEnb* (Main Interrupt Enable)

Address	Register Name	R / W	Bit Symbol	Description		Reset
10h	MainIntEnb	R / W	7: EnSIE_IntStat	0: Disable	1: Enable	00h
		R / W	6: EnEPIntStat	0: Disable	1: Enable	
		R / W	5: EnCPU_IntStat	0: Disable	1: Enable	
		R / W	4: EnFIFO_IntStat	0: Disable	1: Enable	
		R / W	3: EnBulkIntStat	0: Disable	1: Enable	
		R / W	2: EnIDE_IntStat	0: Disable	1: Enable	
		R / W	1: EnEP0IntStat	0: Disable	1: Enable	
		R / W	0: EnRcvEP0SETUP	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。

EnSIE_IntStat ビットはスリープ／スヌーズ中も有効です。

7.2.15. 11h EPrintEnb (EP Interrupt Enable)

Address	Register Name	R / W	Bit Symbol	Description		Reset
11h	EPrintEnb		7:	0:	1:	00h
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
		R / W	2: EnEPcIntStat	0: Disable	1: Enable	
		R / W	1: EnEPbIntStat	0: Disable	1: Enable	
		R / W	0: EnEPaIntStat	0: Disable	1: Enable	

EPrintStat レジスタの割り込み要因による、MainIntStat レジスタの EPrintStat ビットのアサートを許可／禁止します。

7. レジスタ

7.2.16. 12h *SIE_IntEnb* (*SIE Interrupt Enable*)

Address	Register Name	R / W	Bit Symbol	Description		Reset
12h	<i>SIE_IntEnb</i>	R / W	7: <i>EnVBUS_Changed</i>	0: Disable	1: Enable	00h
		R / W	6: <i>EnNonJ</i>	0: Disable	1: Enable	
		R / W	5: <i>EnFinishedPM</i>	0: Disable	1: Enable	
		R / W	5: EnDetectRESET	0: Disable	1: Enable	
		R / W	4: 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	

SIE_IntStat レジスタの割り込み要因による、MainIntStat レジスタの SIE_IntStat ビットのアサートを許可／禁止します。

EnVBUS_Changed、EnNonJ、EnFinishedPM ビットはスリープ／スヌーズ中も有効です。

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、書き込みできません。これら同期ビットの ACTIVE60 ステート時の処理に関しては、SIE_IntStat レジスタの説明をご参照下さい。

7.2.17. 13h CPU_IntEnb (CPU Interrupt Enable)

Address	Register Name	R / W	Bit Symbol	Description		Reset
13h	CPU_IntEnb		7:	0:	1:	00h
		R / W	6: EnRcvSOF	0: Disable	1: Enable	
		R / W	5: EnDMA1_Countup	0: Disable	1: Enable	
		R / W	4: EnDMA1_Cmp	0: Disable	1: Enable	
			3:	0:	1:	
			2:	0:	1:	
		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。

7. レジスタ

7.2.18. 14h FIFO_IntEnb (FIFO Interrupt Enable)

Address	Register Name	R / W	Bit Symbol	Description		Reset
14h	FIFO_IntEnb	R / W	7: EnDescriptorCmp	0: Disable	1: Enable	00h
		R / W	6: EnRAM_RdCmp	0: Disable	1: Enable	
		R / W	5: EnFIFO_IDE_Cmp	0: Disable	1: Enable	
		R / W	4: EnFIFO1_Cmp	0: Disable	1: Enable	
		R / W	3: EnFIFO_Full	0: Disable	1: Enable	
		R / W	2: EnFIFO_Empty	0: Disable	1: Enable	
			1:	0:	1:	
		R / W	0: EnFIFO0_Cmp	0: Disable	1: Enable	

FIFO_IntStat レジスタの割り込み要因による、MainIntStat レジスタの FIFO_IntStat ビットのアサートを許可／禁止します。

7.2.19. 15h BulkIntEnb (Bulk Interrupt Enable)

Address	Register Name	R / W	Bit Symbol	Description		Reset
15h	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	
		R / W	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:	

BulkIntStat レジスタの割り込み要因による、MainIntStat レジスタの BulkIntStat ビットのアサートを許可／禁止します。

7. レジスタ

7.2.20. 16h IDE_IntEnb (IDE Interrupt Enable)

Address	Register Name	R / W	Bit Symbol	Description		Reset
16h	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。

7.2.21. 17h Reserved ()

Address	Register Name	R / W	Bit Symbol	Description		Reset
17h	Reserved		7:	0:	1:	XXh
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
			0:	0:	1:	

7. レジスタ

7.2.22. 18h EP0IntEnb (EP0 Interrupt Enable)

Address	Register Name	R / W	Bit Symbol	Description		Reset
18h	EP0IntEnb		7:	0:	1:	00h
		R / W	6: EnOUT_ShortACK	0: Disable	1: Enable	
		R / W	5: EnIN_TranACK	0: Disable	1: Enable	
		R / W	4: EnOUT_TranACK	0: Disable	1: Enable	
		R / W	3: EnIN_TranNAK	0: Disable	1: Enable	
		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	

EP0IntStat レジスタの割り込み要因による、MainIntStat レジスタの EP0IntStat ビットのアサートを許可／禁止します。

7.2.23. 19h EPalntEnb (EPa Interrupt Enable)

Address	Register Name	R / W	Bit Symbol	Description		Reset
19h	EPalntEnb		7:	0:	1:	00h
		R / W	6: EnOUT_ShortACK	0: Disable	1: Enable	
		R / W	5: EnIN_TranACK	0: Disable	1: Enable	
		R / W	4: EnOUT_TranACK	0: Disable	1: Enable	
		R / W	3: EnIN_TranNAK	0: Disable	1: Enable	
		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	

EPalntStat レジスタの割り込み要因による、EPrntStat レジスタの EPalntStat ビットのアサートを許可／禁止します。

7. レジスタ

7.2.24. 1Ah EPbIntEnb (EPb Interrupt Enable)

Address	Register Name	R / W	Bit Symbol	Description		Reset
1Ah	EPbIntEnb		7:	0:	1:	00h
		R / W	6: EnOUT_ShortACK	0: Disable	1: Enable	
		R / W	5: EnIN_TranACK	0: Disable	1: Enable	
		R / W	4: EnOUT_TranACK	0: Disable	1: Enable	
		R / W	3: EnIN_TranNAK	0: Disable	1: Enable	
		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	

EPbIntStat レジスタの割り込み要因による、EPrIntStat レジスタの EPbIntStat ビットのアサートを許可／禁止します。

7.2.25. 1Bh EPcIntEnb (EPc Interrupt Enable)

Address	Register Name	R / W	Bit Symbol	Description		Reset
1Bh	EPcIntEnb		7:	0:	1:	00h
		R / W	6: EnOUT_ShortACK	0: Disable	1: Enable	
		R / W	5: EnIN_TranACK	0: Disable	1: Enable	
		R / W	4: EnOUT_TranACK	0: Disable	1: Enable	
		R / W	3: EnIN_TranNAK	0: Disable	1: Enable	
		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	

EPcIntStat レジスタの割り込み要因による、EPrIntStat レジスタの EPcIntStat ビットのアサートを許可／禁止します。

7. レジスタ

7.2.26. 1Ch~1Fh Reserved ()

Address	Register Name	R / W	Bit Symbol	Description		Reset
1Ch -1Fh	Reserved		7:	0:	1:	XXh
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
			0:	0:	1:	

7.2.27. 20h RevisionNum (Revision Number)

Address	Register Name	R / W	Bit Symbol	Description	Reset
20h	RevisionNum	R	7: RevisionNum [7]	Revision Number	30h
			6: RevisionNum [6]		
			5: RevisionNum [5]		
			4: RevisionNum [4]		
			3: RevisionNum [3]		
			2: RevisionNum [2]		
			1: RevisionNum [1]		
			0: RevisionNum [0]		

本 LSI のリビジョンナンバーを示します。このレジスタはスリープ／スヌーズ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、0x30 となっています。

7. レジスタ

7.2.28. 21h *ChipReset* (*Chip Reset*)

Address	Register Name	R / W	Bit Symbol	Description		Reset
21h	ChipReset	R / W	7: ResetUTM	0: Do nothing	1: Reset UTM	80h
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
		W	0: AllReset	0: None	1: Reset	

本 LSI をリセットします。スリープ／スヌーズ時でもアクセス有効です。

Bit7 **ResetUTM**

このビットに"1"をセットすると、本 LSI の UTM ブロックを初期化します。

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。

Bit6-1 **Reserved**

Bit0 **AllReset**

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。

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(8.4.3)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下さい。

7.2.29. 22h *PM_Control* (Power Management Control)

Address	Register Name	R / W	Bit Symbol	Description	Reset
22h	<i>PM_Control</i>		7:	0:	1:
		R	6: <i>PM_State</i> [2]	<i>PM_State</i> [2:0]	
			5: <i>PM_State</i> [1]		
			4: <i>PM_State</i> [0]		
		R / W	3: <i>GoSLEEP</i>	0: Do nothing	1: Go to Sleep
		R / W	2: <i>GoSNOOZE</i>	0: Do nothing	1: Go to Snooze
		R / W	1: <i>GoActive60</i>	0: Do nothing	1: Go to Active60
		R / W	0: <i>GoActiveALL</i>	0: Do nothing	1: Go to ActiveAll

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。
このレジスタはスリープ／スヌーズ中でも有効です。

Bit7 **Reserved**

Bit6-4 ***PM_State* [2:0]**

電力モードのステートを表します。

000: スリープ・ステート (OSC オフ、PLL60 オフ、PLL480 オフ)

001: スヌーズ・ステート (OSC オン、PLL60 オフ、PLL480 オフ)

011: アクティブ 60・ステート (OSC オン、PLL60 オン、PLL480 オフ)

111: アクティブオール・ステート (OSC オン、PLL60 オン、PLL480 オン)

その他: 未使用

レジスタマップのうち、斜体太字のビット以外は、

アクティブ 60・ステートまたはアクティブオール・ステートでのみ読み書き可能です。

なお、USB 関連のレジスタ・ビットは、アクティブオール・ステートでのみ読み書き可能です。

なお本ステートは、*PM_Control.GoXXXX* をセットしてから *SIE_IntStat.FinishedPM* 割り込みステータスがセットされ、*PM_Control.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。

Bit3 ***GoSLEEP***

スリープ・ステート以外のステートから、スリープ・ステートへの移行を開始します。

スヌーズ・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スリープ・ステートに移行します。

アクティブ 60・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スリープ・ステートに移行します。

アクティブオール・ステート時に本ビットを"1"にセットすると、最初に PLL480 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スリープ・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 *SIE_IntStat.FinishedPM* ビットがセットされます。

Bit2 ***GoSNOOZE***

スヌーズ・ステート以外のステートから、スヌーズ・ステートへの移行を開始します。

スリープ・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、スヌーズ・ステートに移行します。

アクティブ 60・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、スヌーズ・ステートに移行します。

アクティブオール・ステート時に本ビットを"1"にセットすると、最初に PLL480 の発振を停止し、次に PLL60 の発振を停止し、スヌーズ・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 *SIE_IntStat.FinishedPM* ビットがセットされます。

7. レジスタ

Bit1 GoActive60

アクティブ 60・ステート以外のステートから、アクティブ 60・ステートへの移行を開始します。

スリープ・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、アクティブ 60・ステートに移行します。

スヌーズ・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、アクティブ 60・ステートに移行します。

アクティブオール・ステート時に本ビットを"1"にセットすると、PLL480 の発振を停止し、アクティブ 60・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 SIE_IntStat.FinishedPM ビットがセットされます。

Bit0 GoActiveALL

アクティブオール・ステート以外のステートから、アクティブオール・ステートへの移行を開始します。

スリープ・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、PLL480 の発振を開始し、PLL480 発振安定時間 (約 250us) 経過後、アクティブオール・ステートに移行します。

スヌーズ・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、PLL480 の発振を開始し、PLL480 発振安定時間 (約 250us) 経過後、アクティブオール・ステートに移行します。

アクティブ 60・ステート時に本ビットを"1"にセットすると、PLL480 の発振を開始し、PLL480 発振安定時間 (約 250us) 経過後、アクティブオール・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 SIE_IntStat.FinishedPM ビットがセットされます。

※本 LSI はスリープ／スヌーズ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、スヌーズ中に XINT 信号がアサートされないようマスクしておりますが、スヌーズ解除と同時に 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。

＜スリープ／スヌーズ開始前＞

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)。

＜スリープ／スヌーズ解除後＞

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)。

7.2.30. 23h USB_Control (USB_Control)

Address	Register Name	R / W	Bit Symbol	Description	Reset
23h	USB_Control	R / W	7: DisBusDetect	0: Enable BusDetect	1: Disable BusDetect
		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

USB に関する動作設定を行います。このレジスタは ACTIVE60 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。

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 μ 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、動作説明の章を参照して下さい。

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、“機能説明 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

7. レジスタ

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、XcvtControl レジスタの TermSelect ビット、XcvtSelect 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 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 としての動作が可能となります。

7.2.31. 24h *USB_Status* (*USB_Status*)

Address	Register Name	R / W	Bit Symbol	Description		Reset
24h	USB_Status	R	7: VBUS	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: LineState [1]	Line State [1:0]		
			0: LineState [0]			

USB に関するステータスを表示します。

Bit7 *VBUS*

VBUS 端子の状態が表示されます。このビットはスリープ／スヌーズ中でも有効です。

Bit6 FSxHS

現在の動作モードを示します。USB_Control.GoChirp ビットにより"HS Detection Handshake" (機能説明参照) 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。このビットは ACTIVE60 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。

ケーブルアタッチ時に、“FS(1)”にセットして下さい。

Bit5-2 *Reserved***Bit1-0** *LineState* [1:0]

USB ケーブル上の信号状態を示します。このビットはスリープ／スヌーズ中でも有効です。

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。

LineState		
TermSelect	DP / DM	LineState [1:0]
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

7. レジスタ

7.2.32. 25h XcvtControl (Xcvt Control)

Address	Register Name	R / W	Bit Symbol	Description		Reset
25h	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]			

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 ACTIVE60 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。

Bit7 TermSelect

FS 又は 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、USB_Control 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 XcvtSelect

FS 又は HS いずれかのトランシーバ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、USB_Control 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、
又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。

7.2.33. 26h USB_Test (USB_Test)

Address	Register Name	R / W	Bit Symbol	Description		Reset
26h	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	

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。このレジスタは ACTIVE60 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。

Bit7 EnHS_Test

このビットに"1"をセットすると、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、USB_Control 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、USB_Control 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、XcvtControl レジスタの、TermSelect 及び XcvtSelect 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、XcvtControl レジスタの、TermSelect 及び XcvtSelect 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) テストモードで使用するエンドポイントの MaxPacketSize を 64 以上、転送方向を IN、EndpointNumber を "0xF", EnEndPoint ビットを"1"に設定して、使用可能として下さい。

2) その他のエンドポイントの設定を、上記エンドポイントの設定と重複しないようにして下さい。又は、EnEndPoint ビットをクリアして下さい。

3) テストモードで使用するエンドポイントの 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

7. レジスタ

込んで下さい。また、割り込みイネーブルレジスタの EnIN_TranErr 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_TranErr 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。

7.2.34. 27h Reserved ()

Address	Register Name	R / W	Bit Symbol	Description		Reset
27h	Reserved		7:	0:	1:	XXh
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
			0:	0:	1:	

7. レジスタ

7.2.35. 28h EPnControl (Endpoint Control)

Address	Register Name	R / W	Bit Symbol	Description		Reset
28h	EPnControl	W	7: AllForceNAK	0: Do nothing	1: Set All ForceNAK	XXh
		W	6: EPrForceSTALL	0: Do nothing	1: Set EP's ForceSTALL	
		W	5: AllFIFO_Clr	0: Do nothing	1: Clear All FIFO	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
		W	0: EP0FIFO_Clr	0: Do nothing	1: Clear EP0 FIFO	

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 EPa,EPb,EPc の ForceSTALL ビットを"1"にセットします。

Bit5 AllFIFO_Clr

全てのエンドポイントの FIFO がクリアされます。各エンドポイントの領域設定を行った時は、設定終了後に一度必ずこのビットに"1"をセットして、全てのエンドポイントの FIFO をクリアして下さい。このビットは、FIFO クリア完了後自動的に"0"にクリアされます。

エンドポイントに DMAx{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、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。

Bit4-1 Reserved

Bit0 EP0FIFO_Clr

エンドポイント EP0 の FIFO 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。

エンドポイント EP0 に DMAx{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。

7.2.36. 29h EPrFIFO_Clr (Endpoint FIFO Clear)

Address	Register Name	R / W	Bit Symbol	Description		Reset
29h	EPrFIFO_Clr		7:	0:	1:	XXh
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
		W	2: EPcFIFO_Clr	0: Do nothing	1: Clear EPc FIFO	
		W	1: EPbFIFO_Clr	0: Do nothing	1: Clear EPb FIFO	
		W	0: EPaFIFO_Clr	0: Do nothing	1: Clear EPa FIFO	

該当するエンドポイントの 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。

エンドポイントに DMA_x{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、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。

7. レジスタ

7.2.37. 2Ah ClrAllJoin (Clear All Join)

Address	Register Name	R / W	Bit Symbol	Description		Reset
2Ah	ClrAllJoin	W	7: ClrJoinIDE	0: Do nothing	1: Clear Join IDE	XXh
			6: ClrJoinFIFO_Status	0: Do nothing	1: Clear Join FIFO_Status	
		W	5: ClrJoinDMA0_Rd	0: Do nothing	1: Clear Join DMA0_Rd	
		W	4: ClrJoinDMA0_Wr	0: Do nothing	1: Clear Join DMA0_Wr	
		W	3: ClrJoinDMA1_Rd	0: Do nothing	1: Clear Join DMA1_Rd	
		W	2: ClrJoinDMA1_Wr	0: Do nothing	1: Clear Join DMA1_Wr	
		W	1: ClrJoinCPU_Rd	0: Do nothing	1: Clear Join CPU_Rd	
		W	0: ClrJoinCPU_Wr	0: Do nothing	1: Clear Join CPU_Wr	

該当するポートと各エンドポイン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。

エンドポイントがポートに接続 (EPx{x=0,a-c}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。

7.2.38. 2Bh Reserved ()

Address	Register Name	R / W	Bit Symbol	Description		Reset
2Bh	Reserved		7:	0:	1:	XXh
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
			0:	0:	1:	

7. レジスタ

7.2.39. 2Ch BulkOnlyControl (BulkOnly Control)

Address	Register Name	R / W	Bit Symbol	Description		Reset
2Ch	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:	1: Begin CBW Mode	
		R / W	1: GoCSW_Mode	0:	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、BulkOnlyConfig レジスタの項を参照して下さい。

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、BulkOnlyConfig レジスタの項を参照して下さい。

Bit0 Reserved

7.2.40. 2Dh BulkOnlyConfig (BulkOnly Configuration)

Address	Register Name	R / W	Bit Symbol	Description		Reset
2Dh	BulkOnlyConfig		7:	0:	1:	00h
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
		R / W	2: EPcBulkOnly	0: None	1: Enable BulkOnly on EPc	
		R / W	1: EPbBulkOnly	0: None	1: Enable BulkOnly on EPb	
			0:	0:	1:	

バルクオンリーサポート機能を有効にします。

Bit7-3 Reserved**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 Reserved

7. レジスタ

7.2.41. 2Eh WakeupTim_H (Wakeup Time High)

7.2.42. 2Fh WakeupTim_L (Wakeup Time Low)

Address	Register Name	R / W	Bit Symbol	Description	Reset
2Eh -2Fh	<i>WakeupTim_H</i> <i>WakeupTim_L</i>	R / W	<i>WakeupTim [15:0]</i>	Wakeup Time [15:0]	0000h

スリープ・ステートからスヌーズ・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。本レジスタはスリープ時にもアクセスが可能です。

スリープ・ステート時に、PM_Control.GoActiveALL、PM_Control.GoActive60、PM_Control.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、USB の SUSPEND 時にスリープ・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。

- 7.2.43. 30h EP0SETUP_0 (EP0 SETUP 0)
 7.2.44. 31h EP0SETUP_1 (EP0 SETUP 1)
 7.2.45. 32h EP0SETUP_2 (EP0 SETUP 2)
 7.2.46. 33h EP0SETUP_3 (EP0 SETUP 3)
 7.2.47. 34h EP0SETUP_4 (EP0 SETUP 4)
 7.2.48. 35h EP0SETUP_5 (EP0 SETUP 5)
 7.2.49. 36h EP0SETUP_6 (EP0 SETUP 6)
 7.2.50. 37h EP0SETUP_7 (EP0 SETUP 7)

Address	Register Name	R / W	Bit Symbol	Description	Reset
30h	EP0SETUP_0	R	7: EP0SETUP_n [7]	Endpoint 0 SETUP Data 0 -Endpoint 0 SETUP Data 7	00h
-37h	-EP0SETUP_7		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。

7. レジスタ

7.2.51. 38h USB_Address (USB Address)

Address	Register Name	R / W	Bit Symbol	Description		Reset	
38h	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。このレジスタは ACTIVE60 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。

Bit7 SetAddress

SetAddress リクエストの受信時にセットすると、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。

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。

7.2.52. 39h Reserved ()

Address	Register Name	R / W	Bit Symbol	Description		Reset
39h	Reserved		7:	0:	1:	XXh
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
			0:	0:	1:	

7. レジスタ

7.2.53. 3Ah SETUP_Control(SETUP Control)

Address	Register Name	R / W	Bit Symbol	Description		Reset
39h	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 EP0SETUP_0～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 EP0ControlIN,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。

7.2.54. 3Bh～3Dh Reserved ()

Address	Register Name	R / W	Bit Symbol	Description		Reset
3Bh -3Dh	Reserved		7:	0:	1:	XXh
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
			0:	0:	1:	

7. レジスタ

7.2.55. 3Eh FrameNumber_H (FrameNumber High)

7.2.56. 3Fh FrameNumber_L (FrameNumber Low)

Address	Register Name	R / W	Bit Symbol	Description		Reset
3Eh	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]			

Address	Register Name	R / W	Bit Symbol	Description	Reset
3Fh	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。その際に FrameNumber_H レジスタを先にアクセスして下さい。

3Eh.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。

3Eh.Bit6-3 Reserved

3Eh.Bit2-0, 3Fh.Bit7-0 FrameNumber [10:0]

受信した SOF パケットの FrameNumber が表示されます。

7.2.57. 40h EP0MaxSize (EP0 Max Packet Size)

Address	Register Name	R / W	Bit Symbol	Description		Reset	
40h	EP0MaxSize		7:	0:	1:	40h	
		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**

7. レジスタ

7.2.58. 41h EP0Control (EP0 Control)

Address	Register Name	R / W	Bit Symbol	Description		Reset
41h	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、EP0ControlIN 及び 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、又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 又は 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 EP0ControlIN または 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、EPnIntStat レジスタの DescriptorCmp ビットと 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、動作説明の章を参照して下さい。

7.2.59. 42h EP0ControlIN (EP0 Control IN)

Address	Register Name	R / W	Bit Symbol	Description	Reset
42h	EP0ControlIN		7:	0:	1:
		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。

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、

7. レジスタ

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。

7.2.60. 43h EP0ControlOUT (EP0 Control OUT)

Address	Register Name	R / W	Bit Symbol	Description		Reset
43h	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 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、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"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。

7. レジスタ

7.2.61. 44h Reserved ()

Address	Register Name	R / W	Bit Symbol	Description		Reset
44h	Reserved		7:	0:	1:	XXh
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
			0:	0:	1:	

7.2.62. 45h EP0Join (End Point0 Join)

Address	Register Name	R / W	Bit Symbol	Description	Reset
45h	EP0Join		7:	0:	1:
			6:	0:	1
		R / W	5: JoinDMA0_Rd	0: Do nothing	1: Join EP0 to DMA0_Rd
		R / W	4: JoinDMA0_Wr	0: Do nothing	1: Join EP0 to DMA0_Wr
		R / W	3: JoinDMA1_Rd	0: Do nothing	1: Join EP0 to DMA1_Rd
		R / W	2: JoinDMA1_Wr	0: Do nothing	1: Join EP0 to DMA1_Wr
		R / W	1: JoinCPU_Rd	0: Do nothing	1: Join EP0 to CPU_Rd
		R / W	0: JoinCPU_Wr	0: Do nothing	1: Join EP0 to CPU_Wr

エンドポイント 0 とデータ転送を行うポートを指定します。

Bit7-6 **Reserved**

Bit5 **JoinDMA0_Rd**

エンドポイント EP0 の FIFO で DMA0 のリード転送を行います。即ち DMA0 でリードが行われると、このエンドポイントの FIFO からデータが読み出されます。このリード転送は、エンドポイントの方向設定に依存せず、任意に行うことができます。

Bit4 **JoinDMA0_Wr**

エンドポイント EP0 の FIFO で DMA0 のライト転送を行います。即ち DMA0 で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。このライト転送は、エンドポイントの方向設定に依存せず、任意に行うことができます。

Bit3 **JoinDMA1_Rd**

エンドポイント EP0 の FIFO で DMA1 のリード転送を行います。即ち DMA1 でリードが行われると、このエンドポイントの FIFO からデータが読み出されます。このリード転送は、エンドポイントの方向設定に依存せず、任意に行うことができます。

Bit2 **JoinDMA1_Wr**

エンドポイント EP0 の FIFO で DMA1 のライト転送を行います。即ち DMA1 で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。このライト転送は、エンドポイントの方向設定に依存せず、任意に行うことができます。

Bit1 **JoinCPU_Rd**

エンドポイント EP0 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 EPnFIFO_Rd_H,L レジスタ、または、EPnFIFO_ByteRd レジスタのリードが行われると、このエンドポイントの FIFO からデータが読み出されます。このリード転送は、エンドポイントの方向設定に依存せず、任意に行うことができます。

Bit0 **JoinCPU_Wr**

エンドポイント EP0 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 EPnFIFO_Wr_H,L 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。このライト転送は、エンドポイントの方向設定に依存せず、任意に行うことができます。

JoinDMAx{x=0,1}_Rd, JoinDMAx{x=0,1}_Wr ビットを設定した場合は、DMAx{x=0,1}_Remain_H,L レジスタにより、エンドポイントが OUT 方向の場合は残りデータ数、エンドポイントが IN 方向の場合は空き容量が、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、EPnRdRemain_H,L、EPnWrRemain_H,L を参照し、EPnFIFO_Rd_H,L、EPnFIFO_ByteRd、EPnFIFO_Wr_H,L 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。

このレジスタの Bit5-0 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"を書き込んだ場合、上位ビットが有効とされます。

7. レジスタ

7.2.63. 46h~4Fh Reserved ()

Address	Register Name	R / W	Bit Symbol	Description		Reset
46h -4Fh	Reserved		7:	0:	1:	XXh
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
			0:	0:	1:	

7.2.64. 50h Reserved ()

7.2.65. 51h EPaMaxSize_L (EPa Max Packet Size Low)

Address	Register Name	R / W	Bit Symbol	Description	Reset
50h	Reserved		7:	0:	XXh
			6:	0:	
			5:	0:	
			4:	0:	
			3:	0:	
			2:	0:	
			1:	0:	
			0	0:	

Address	Register Name	R / W	Bit Symbol	Description		Reset
51h	EPaMaxSize_L		7:	0:	1:	00h
		R / W	6: MaxSize [6]	Endpoint [a] Max Packet Size		
			5: MaxSize [5]			
			4: MaxSize [4]			
			3: MaxSize [3]			
			2:	0:	1:	
			1:	0:	1:	
			0:	0:	1:	

MaxPacketSize を設定します。

50h.Bit7-0 **Reserved**

51h.Bit7 **Reserved**

51h.Bit6-3 **EPaMaxSize [6:3]**

エンドポイント EPa の MaxPacketSize を設定します。
 このエンドポイントを FS の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
 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 64 バイトまで
 の任意の転送数が設定可能です。

51h.Bit2-0 **Reserved**

7. レジスタ

7.2.66. 52h EPaConfig_0 (EPa Configuration 0)

Address	Register Name	R / W	Bit Symbol	Description		Reset
52h	EPaConfig_0	R / W	7: INxOUT	0: OUT	1: IN	00h
		R / W	6: IntEP_Mode	0: Normal Toggle (IN) 0: Bulk OUT (OUT)	1: Always Toggle (IN) 1: Interrupt OUT (OUT)	
		R / W	5: EnEndpoint	0: Disable Endpoint	1: Enable Endpoint	
			4:	0:	1:	
		R / W	3: EndpointNumber [3]	Endpoint Number		
			2: EndpointNumber [2]			
			1: EndpointNumber [1]			
			0: EndpointNumber [0]			

エンドポイント EPa 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 EnEndpoint

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を有効にします。

このビットが"0"の時は、エンドポイントへのアクセスを無視します。

ホストからの SetConfiguration リクエストに従って設定して下さい。

Bit4 Reserved

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。

7.2.67. 53h Reserved ()

Address	Register Name	R / W	Bit Symbol	Description		Reset
53h	Reserved		7:	0:	1:	XXh
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
			0:	0:	1:	

7. レジスタ

7.2.68. 54h EPaControl (EPa Control)

Address	Register Name	R / W	Bit Symbol	Description		Reset
54h	EPaControl	R / W	7: AutoForceNAK	0: Do nothing	1: Auto Force NAK	00h
		R / W	6: EnShortPkt	0: Do nothing	1: Enable Short Packet	
		R / W	5: DisAF_NAK_Short	0: Auto Force NAK Short	1 Disable Auto Force	
		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 EPa の動作設定を行います。

Bit7 AutoForceNAK

エンドポイント EPa 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 EPa 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 EPa 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 EPa 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 EPa 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 EPa 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 EPa 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。

7. レジスタ

7.2.69. 55h EPaJoin (End Point a Join)

Address	Register Name	R / W	Bit Symbol	Description		Reset
55h	EPaJoin	R / W	7: JoinIDE	0: Do nothing	1: Join EPa to IDE	00h
		R / W	6: JoinFIFO_Stat	0: Do nothing	1: Join EPa to show Status	
		R / W	5: JoinDMA0_Rd	0: Do nothing	1: Join EPa to DMA0_Rd	
		R / W	4: JoinDMA0_Wr	0: Do nothing	1: Join EPa to DMA0_Wr	
		R / W	3: JoinDMA1_Rd	0: Do nothing	1: Join EPa to DMA1_Rd	
		R / W	2: JoinDMA1_Wr	0: Do nothing	1: Join EPa to DMA1_Wr	
		R / W	1: JoinCPU_Rd	0: Do nothing	1: Join EPa to CPU_Rd	
		R / W	0: JoinCPU_Wr	0: Do nothing	1: Join EPa to CPU_Wr	

エンドポイント EPa とデータ転送を行うポートを指定します。

Bit7 JoinIDE

エンドポイント EPa の FIFO で IDE のデータ転送を行います。

このビットをセットすると、IDE の転送方向は、EPa の方向に設定されます。即ち、エンドポイント EPa が OUT 方向であったばあい、IDE ではライト転送を行います。また、エンドポイント EPa が IN 方向であった場合、IDE ではリード転送を行います。

Bit6 JoinFIFO_Stat

エンドポイント EPa の FIFO の Full 及び Empty の状態を、FIFO_IntStat.FIFO_Full 及び FIFO_IntStat.FIFO_Empty でモニタできるようにします。

Bit5 JoinDMA0_Rd

エンドポイント EPa の FIFO で DMA0 のリード転送を行います。即ち DMA0 でリードが行われると、このエンドポイントの FIFO からデータが読み出されます。このリード転送は、エンドポイントの方向設定に依存せず、任意に行うことができます。

Bit4 JoinDMA0_Wr

エンドポイント EPa の FIFO で DMA0 のライト転送を行います。即ち DMA0 で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。このライト転送は、エンドポイントの方向設定に依存せず、任意に行うことができます。

Bit3 JoinDMA1_Rd

エンドポイント EPa の FIFO で DMA1 のリード転送を行います。即ち DMA1 でリードが行われると、このエンドポイントの FIFO からデータが読み出されます。このリード転送は、エンドポイントの方向設定に依存せず、任意に行うことができます。

Bit2 JoinDMA1_Wr

エンドポイント EPa の FIFO で DMA1 のライト転送を行います。即ち DMA1 で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。このライト転送は、エンドポイントの方向設定に依存せず、任意に行うことができます。

Bit1 JoinCPU_Rd

エンドポイント EPa 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 EPnFIFO_Rd_H,L レジスタ、または、EPnFIFO_ByteRd レジスタのリードが行われると、このエンドポイントの FIFO からデータが読み出されます。このリード転送は、エンドポイントの方向設定に依存せず、任意に行うことができます。

Bit0 JoinCPU_Wr

エンドポイント EPa 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 EPnFIFO_Wr_H,L 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。このライト転送は、エンドポイントの方向設定に依存せず、任意に行うことができます。

JoinDMAx{x=0,1}_Rd, JoinDMAx{x=0,1}_Wr ビットを設定した場合は、DMAx{x=0,1}_Remain_H,L レジスタにより、エンド

ポイントが OUT 方向の場合は残りデータ数、エンドポイントが IN 方向の場合は空き容量が、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、EPnRdRemain_H,L、EPnWrRemain_H,L を参照し、EPnFIFO_Rd_H,L、EPnFIFO_ByteRd、EPnFIFO_Wr_H,L 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。

このレジスタの Bit5-0 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"を書き込んだ場合、上位ビットが有効とされます。

7. レジスタ

7.2.70. 56h～57h Reserved ()

Address	Register Name	R / W	Bit Symbol	Description		Reset
56h -57h	Reserved		7:	0:	1:	XXh
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
			0:	0:	1:	

7.2.71. 58h EPbMaxSize_H (EPb Max Packet Size High)

7.2.72. 59h EPbMaxSize_L (EPb Max Packet Size Low)

Address	Register Name	R / W	Bit Symbol	Description		Reset
58h	EPbMaxSize_H		7:	0:	1:	00h
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
		R / W	1: MaxSize [9]			
			0: MaxSize [8]			

Address	Register Name	R / W	Bit Symbol	Description	Reset
59h	EPbMaxSize_L	R / W	7: MaxSize [7]	Endpoint [b] Max Packet Size	00h
			6: MaxSize [6]		
			5: MaxSize [5]		
			4: MaxSize [4]		
			3: MaxSize [3]		
			2:	0:	
			1:	0:	
			0:	0:	

MaxPacketSize を設定します。

58h.Bit7-2 **Reserved**

58h.Bit1-0, 59h.Bit7-3 **EPbMaxSize [9:3]**

エンドポイント EPb 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 512 バイトまで

の任意の転送数が設定可能です。

59h.Bit2-0 **Reserved**

7. レジスタ

7.2.73. 5Ah EPbConfig_0 (EPb Configuration 0)

Address	Register Name	R / W	Bit Symbol	Description		Reset
5Ah	EPbConfig_0	R / W	7: INxOUT	0: OUT	1: IN	00h
		R / W	6: IntEP_Mode	0: Normal Toggle (IN) 0: Bulk OUT (OUT)	1: Always Toggle (IN) 1: Interrupt OUT (OUT)	
		R / W	5: EnEndpoint	0: Disable Endpoint	1: Enable Endpoint	
			4:	0:	1:	
		R / W	3: EndpointNumber [3]	Endpoint Number		
			2: EndpointNumber [2]			
			1: EndpointNumber [1]			
			0: EndpointNumber [0]			

エンドポイント EPb 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 EnEndpoint

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を有効にします。

このビットが"0"の時は、エンドポイントへのアクセスを無視します。

ホストからの SetConfiguration リクエストに従って設定して下さい。

Bit4 Reserved

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。

7.2.74. 5Bh Reserved ()

Address	Register Name	R / W	Bit Symbol	Description		Reset
5Bh	Reserved		7:	0:	1:	XXh
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
			0:	0:	1:	

7. レジスタ

7.2.75. 5Ch EPbControl (EPb Control)

Address	Register Name	R / W	Bit Symbol	Description		Reset
5Ch	EPbControl	R / W	7: AutoForceNAK	0: Do nothing	1: Auto Force NAK	00h
		R / W	6: EnShortPkt	0: Do nothing	1: Enable Short Packet	
		R / W	5: DisAF_NAK_Short	0: Auto Force NAK Short	1 Disable Auto Force	
		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 EPb の動作設定を行います。

Bit7 AutoForceNAK

エンドポイント EPb 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 EPb 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 EPb 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 EPb 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 EPb 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 EPb 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 EPb 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。

7. レジスタ

7.2.76. 5Dh EPbJoin (End Point b Join)

Address	Register Name	R / W	Bit Symbol	Description		Reset
5Dh	EPbJoin	R / W	7: JoinIDE	0: Do nothing	1: Join EPb to IDE	00h
		R / W	6: JoinFIFO_Stat	0: Do nothing	1: Join EPb to show Status	
		R / W	5: JoinDMA0_Rd	0: Do nothing	1: Join EPb to DMA0_Rd	
		R / W	4: JoinDMA0_Wr	0: Do nothing	1: Join EPb to DMA0_Wr	
		R / W	3: JoinDMA1_Rd	0: Do nothing	1: Join EPb to DMA1_Rd	
		R / W	2: JoinDMA1_Wr	0: Do nothing	1: Join EPb to DMA1_Wr	
		R / W	1: JoinCPU_Rd	0: Do nothing	1: Join EPb to CPU_Rd	
		R / W	0: JoinCPU_Wr	0: Do nothing	1: Join EPb to CPU_Wr	

エンドポイント EPb とデータ転送を行うポートを指定します。

Bit7 JoinIDE

エンドポイント EPb の FIFO で IDE のデータ転送を行います。

このビットをセットすると、IDE の転送方向は、EPb の方向に設定されます。即ち、エンドポイント EPb が OUT 方向であったばあい、IDE ではライト転送を行います。また、エンドポイント EPb が IN 方向であった場合、IDE ではリード転送を行います。

Bit6 JoinFIFO_Stat

エンドポイント EPb の FIFO の Full 及び Empty の状態を、FIFO_IntStat.FIFO_Full 及び FIFO_IntStat.FIFO_Empty でモニタできるようにします。

Bit5 JoinDMA0_Rd

エンドポイント EPb の FIFO で DMA0 のリード転送を行います。即ち DMA0 でリードが行われると、このエンドポイントの FIFO からデータが読み出されます。このリード転送は、エンドポイントの方向設定に依存せず、任意に行うことができます。

Bit4 JoinDMA0_Wr

エンドポイント EPb の FIFO で DMA0 のライト転送を行います。即ち DMA0 で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。このライト転送は、エンドポイントの方向設定に依存せず、任意に行うことができます。

Bit3 JoinDMA1_Rd

エンドポイント EPb の FIFO で DMA1 のリード転送を行います。即ち DMA1 でリードが行われると、このエンドポイントの FIFO からデータが読み出されます。このリード転送は、エンドポイントの方向設定に依存せず、任意に行うことができます。

Bit2 JoinDMA1_Wr

エンドポイント EPb の FIFO で DMA1 のライト転送を行います。即ち DMA1 で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。このライト転送は、エンドポイントの方向設定に依存せず、任意に行うことができます。

Bit1 JoinCPU_Rd

エンドポイント EPb 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 EPnFIFO_Rd_H,L レジスタ、または、EPnFIFO_ByteRd レジスタのリードが行われると、このエンドポイントの FIFO からデータが読み出されます。このリード転送は、エンドポイントの方向設定に依存せず、任意に行うことができます。

Bit0 JoinCPU_Wr

エンドポイント EPb 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 EPnFIFO_Wr_H,L 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。このライト転送は、エンドポイントの方向設定に依存せず、任意に行うことができます。

JoinDMAx{x=0,1}_Rd, JoinDMAx{x=0,1}_Wr ビットを設定した場合は、DMAx{x=0,1}_Remain_H,L レジスタにより、エンド

ポイントが OUT 方向の場合は残りデータ数、エンドポイントが IN 方向の場合は空き容量が、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、EPnRdRemain_H,L、EPnWrRemain_H,L を参照し、EPnFIFO_Rd_H,L、EPnFIFO_ByteRd、EPnFIFO_Wr_H,L 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。

このレジスタの Bit5-0 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"を書き込んだ場合、上位ビットが有効とされます。

7. レジスタ

7.2.77. 5Eh~5Fh Reserved ()

Address	Register Name	R / W	Bit Symbol	Description		Reset
5Eh -5Fh	Reserved		7:	0:	1:	XXh
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
			0:	0:	1:	

7.2.78. 60h EPcMaxSize_H (EPc Max Packet Size High)

7.2.79. 61h EPcMaxSize_L (EPc Max Packet Size Low)

Address	Register Name	R / W	Bit Symbol	Description		Reset
60h	EPcMaxSize_H		7:	0:	1:	00h
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
		R / W	1: MaxSize [9]			
			0: MaxSize [8]			

Address	Register Name	R / W	Bit Symbol	Description	Reset
61h	EPcMaxSize_L	R / W	7: MaxSize [7]	Endpoint [c] Max Packet Size	00h
			6: MaxSize [6]		
			5: MaxSize [5]		
			4: MaxSize [4]		
			3: MaxSize [3]		
			2:	0:	
			1:	0:	
			0:	0:	

MaxPacketSize を設定します。

60h.Bit7-2 **Reserved**

60h.Bit1-0, 61h.Bit7-3 **EPcMaxSize [9:3]**

エンドポイント EPc 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 512 バイトまで

の任意の転送数が設定可能です。

61h.Bit2-0 **Reserved**

7. レジスタ

7.2.80. 62h EPcConfig_0 (EPc Configuration 0)

Address	Register Name	R / W	Bit Symbol	Description		Reset
62h	EPcConfig_0	R / W	7: INxOUT	0: OUT	1: IN	00h
		R / W	6: IntEP_Mode	0: Normal Toggle (IN) 0: Bulk OUT (OUT)	1: Always Toggle (IN) 1: Interrupt OUT (OUT)	
		R / W	5: EnEndpoint	0: Disable Endpoint	1: Enable Endpoint	
			4:	0:	1:	
		R / W	3: EndpointNumber [3]	Endpoint Number		
			2: EndpointNumber [2]			
			1: EndpointNumber [1]			
			0: EndpointNumber [0]			

エンドポイント EPc 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 EnEndpoint

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を有効にします。

このビットが"0"の時は、エンドポイントへのアクセスを無視します。

ホストからの SetConfiguration リクエストに従って設定して下さい。

Bit4 Reserved

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。

7.2.81. 63h Reserved ()

Address	Register Name	R / W	Bit Symbol	Description		Reset
63h	Reserved		7:	0:	1:	XXh
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
			0:	0:	1:	

7. レジスタ

7.2.82. 64h EPcControl (EPc Control)

Address	Register Name	R / W	Bit Symbol	Description		Reset
64h	EPcControl	R / W	7: AutoForceNAK	0: Do nothing	1: Auto Force NAK	00h
		R / W	6: EnShortPkt	0: Do nothing	1: Enable Short Packet	
		R / W	5: DisAF_NAK_Short	0: Auto Force NAK Short	1 Disable Auto Force	
		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 EPc の動作設定を行います。

Bit7 AutoForceNAK

エンドポイント EPc 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 EPc 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 EPc 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 EPc 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 EPc 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 EPc 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 EPc 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。

7. レジスタ

7.2.83. 65h EPcJoin (End Point c Join)

Address	Register Name	R / W	Bit Symbol	Description		Reset
65h	EPcJoin	R / W	7: JoinIDE	0: Do nothing	1: Join EPc to IDE	00h
		R / W	6: JoinFIFO_Status	0: Do nothing	1: Join EPc to show Status	
		R / W	5: JoinDMA0_Rd	0: Do nothing	1: Join EPc to DMA0_Rd	
		R / W	4: JoinDMA0_Wr	0: Do nothing	1: Join EPc to DMA0_Wr	
		R / W	3: JoinDMA1_Rd	0: Do nothing	1: Join EPc to DMA1_Rd	
		R / W	2: JoinDMA1_Wr	0: Do nothing	1: Join EPc to DMA1_Wr	
		R / W	1: JoinCPU_Rd	0: Do nothing	1: Join EPc to CPU_Rd	
		R / W	0: JoinCPU_Wr	0: Do nothing	1: Join EPc to CPU_Wr	

エンドポイント EPc とデータ転送を行うポートを指定します。

Bit7 JoinIDE

エンドポイント EPc の FIFO で IDE のデータ転送を行います。

このビットをセットすると、IDE の転送方向は、EPc の方向に設定されます。即ち、エンドポイント EPc が OUT 方向であったばあい、IDE ではライト転送を行います。また、エンドポイント EPc が IN 方向であった場合、IDE ではリード転送を行います。

Bit6 JoinFIFO_Stat

エンドポイント EPc の FIFO の Full 及び Empty の状態を、FIFO_IntStat.FIFO_Full 及び FIFO_IntStat.FIFO_Empty でモニタできるようにします。

Bit5 JoinDMA0_Rd

エンドポイント EPc の FIFO で DMA0 のリード転送を行います。即ち DMA0 でリードが行われると、このエンドポイントの FIFO からデータが読み出されます。このリード転送は、エンドポイントの方向設定に依存せず、任意に行うことができます。

Bit4 JoinDMA0_Wr

エンドポイント EPc の FIFO で DMA0 のライト転送を行います。即ち DMA0 で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。このライト転送は、エンドポイントの方向設定に依存せず、任意に行うことができます。

Bit3 JoinDMA1_Rd

エンドポイント EPc の FIFO で DMA1 のリード転送を行います。即ち DMA1 でリードが行われると、このエンドポイントの FIFO からデータが読み出されます。このリード転送は、エンドポイントの方向設定に依存せず、任意に行うことができます。

Bit2 JoinDMA1_Wr

エンドポイント EPc の FIFO で DMA1 のライト転送を行います。即ち DMA1 で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。このライト転送は、エンドポイントの方向設定に依存せず、任意に行うことができます。

Bit1 JoinCPU_Rd

エンドポイント EPc 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 EPnFIFO_Rd_H,L レジスタ、または、EPnFIFO_ByteRd レジスタのリードが行われると、このエンドポイントの FIFO からデータが読み出されます。このリード転送は、エンドポイントの方向設定に依存せず、任意に行うことができます。

Bit0 JoinCPU_Wr

エンドポイント EPc 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 EPnFIFO_Wr_H,L 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。このライト転送は、エンドポイントの方向設定に依存せず、任意に行うことができます。

JoinDMAx{x=0,1}_Rd, JoinDMAx{x=0,1}_Wr ビットを設定した場合は、DMAx{x=0,1}_Remain_H,L レジスタにより、エンド

ポイントが OUT 方向の場合は残りデータ数、エンドポイントが IN 方向の場合は空き容量が、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、EPnRdRemain_H,L、EPnWrRemain_H,L を参照し、EPnFIFO_Rd_H,L、EPnFIFO_ByteRd、EPnFIFO_Wr_H,L 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。

このレジスタの Bit5-0 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"を書き込んだ場合、上位ビットが有効とされます。

7. レジスタ

7.2.84. 66h~6Bh Reserved ()

Address	Register Name	R / W	Bit Symbol	Description		Reset
66h -6Bh	Reserved		7:	0:	1:	XXh
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
			0:	0:	1:	

7.2.85. 6Ch RAM_WrAdrs_H (RAM Write Address High)

7.2.86. 6Dh RAM_WrAdrs_L (RAM Write Address Low)

Address	Register Name	R / W	Bit Symbol	Description	Reset
6Ch	RAM_WrAdrs_H		7:		00h
			6:		
			5:		
			4:		
		R / W	3: RAM_WrAdrs [11]	RAM Write Address	
			2: RAM_WrAdrs [10]		
			1: RAM_WrAdrs [9]		
			0: RAM_WrAdrs [8]		

Address	Register Name	R / W	Bit Symbol	Description	Reset
6Dh	RAM_WrAdrs_L	R / W	7: RAM_WrAdrs [7]	RAM Write Address	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_H,L 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。

6Ch.Bit7-4 **Reserved**

6Ch.Bit3-0, 6Dh.Bit7-0 **RAM_WrAdrs[11:0]**

RAM への書き込みを行う際のアドレスを指定します。RAM_WrDoor_H,L 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_H,L 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 RAM_WrDoor_H,L レジスタの項を参照して下さい。

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。

7. レジスタ

7.2.87. 6Eh RAM_WrDoor_H (RAM Write Door High)

7.2.88. 6Fh RAM_WrDoor_L (RAM Write Door Low)

Address	Register Name	R / W	Bit Symbol	Description	Reset
06Eh	RAM_WrDoor_H	W	7: RAM_WrDoor [15]	RAM Write Door	XXh
			6: RAM_WrDoor [14]		
			5: RAM_WrDoor [13]		
			4: RAM_WrDoor [12]		
			3: RAM_WrDoor [11]		
			2: RAM_WrDoor [10]		
			1: RAM_WrDoor [9]		
			0: RAM_WrDoor [8]		

Address	Register Name	R / W	Bit Symbol	Description	Reset
06Fh	RAM_WrDoor_L	W	7: RAM_WrDoor [7]	RAM Write Door	XXh
			6: RAM_WrDoor [6]		
			5: RAM_WrDoor [5]		
			4: RAM_WrDoor [4]		
			3: RAM_WrDoor [3]		
			2: RAM_WrDoor [2]		
			1: RAM_WrDoor [1]		
			0: RAM_WrDoor [0]		

6Eh.Bit7-0, 6Fh.Bit7-0 RAM_WrDoor [15:0]

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。

RAM_WrDoor_H,L レジスタによりデスクリプタエリアへのデータの書き込みを行えます。このレジスタから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。

7.2.89. 70h EPnFIFO_Rd_H (EPn FIFO Read High)

7.2.90. 71h EPnFIFO_Rd_L (EPn FIFO Read Low)

Address	Register Name	R / W	Bit Symbol	Description	Reset
70h	EPnFIFO_Rd_H	R	7: EPnFIFO_Rd [15]	Endpoint n FIFO Read	XXh
			6: EPnFIFO_Rd [14]		
			5: EPnFIFO_Rd [13]		
			4: EPnFIFO_Rd [12]		
			3: EPnFIFO_Rd [11]		
			2: EPnFIFO_Rd [10]		
			1: EPnFIFO_Rd [9]		
			0: EPnFIFO_Rd [8]		

Address	Register Name	R / W	Bit Symbol	Description	Reset
71h	EPnFIFO_Rd_L	R	7: EPnFIFO_Rd [7]	Endpoint n FIFO Read	XXh
			6: EPnFIFO_Rd [6]		
			5: EPnFIFO_Rd [5]		
			4: EPnFIFO_Rd [4]		
			3: EPnFIFO_Rd [3]		
			2: EPnFIFO_Rd [2]		
			1: EPnFIFO_Rd [1]		
			0: EPnFIFO_Rd [0]		

70h.Bit7-0, 71h.Bit7-0 EPnFIFO_Rd [15:0]

EPx{x=0,a-c}Join.JoinCPU_Rd ビットがセットされているエンドポイント FIFO のデータを読み出せます。

8bit mode 時には EPnFIFO_Rd_H,L いずれのレジスタにアクセスしても FIFO のデータを読み出せます。

16bit mode 時の 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。詳細は機能説明”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 EPnRdRemain_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。

7. レジスタ

7.2.91. 72h EPnFIFO_Wr_H(EPn FIFO Write High)

7.2.92. 73h EPnFIFO_Wr_L(EPn FIFO Write Low)

Address	Register Name	R / W	Bit Symbol	Description	Reset
72h	EPnFIFO_Wr_H	W	7: EPnFIFO_Wr [15]	Endpoint n FIFO Write	XXh
			6: EPnFIFO_Wr [14]		
			5: EPnFIFO_Wr [13]		
			4: EPnFIFO_Wr [12]		
			3: EPnFIFO_Wr [11]		
			2: EPnFIFO_Wr [10]		
			1: EPnFIFO_Wr [9]		
			0: EPnFIFO_Wr [8]		

Address	Register Name	R / W	Bit Symbol	Description	Reset
73h	EPnFIFO_Wr_L	W	7: EPnFIFO_Wr [7]	Endpoint n FIFO Write	XXh
			6: EPnFIFO_Wr [6]		
			5: EPnFIFO_Wr [5]		
			4: EPnFIFO_Wr [4]		
			3: EPnFIFO_Wr [3]		
			2: EPnFIFO_Wr [2]		
			1: EPnFIFO_Wr [1]		
			0: EPnFIFO_Wr [0]		

72h.Bit7-0, 73h.Bit7-0 EPnFIFO_Wr [15:0]

EPx{x=0,a-c}Join.JoinCPU_Wr ビットがセットされているエンドポイント FIFO ヘデータを書き込みます。

8bit mode 時には EPnFIFO_Wr_H,L いずれのレジスタにアクセスしても FIFO ヘデータを書き込みます。

16bit mode 時の 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。詳細は機能説明”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 EPnWrRemain_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。

7.2.93. 74h EPnRdRemain_H (EPn FIFO Read Remain High)

7.2.94. 75h EPnRdRemain_L (EPn FIFO Read Remain Low)

Address	Register Name	R / W	Bit Symbol	Description		Reset
74h	EPnRdRemain_H	R	7: RdRemainValid	0:None	1: Read Remain Valid	00h
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
		R	3: EPnRdRemain [11]	Endpoint n FIFO Read Remain High		
			2: EPnRdRemain [10]			
			1: EPnRdRemain [9]			
			0: EPnRdRemain [8]			

Address	Register Name	R / W	Bit Symbol	Description	Reset
75h	EPnRdRemain_L	R	7: EPnRdRemain [7]	Endpoint n FIFO Read Remain Low	00h
			6: EPnRdRemain [6]		
			5: EPnRdRemain [5]		
			4: EPnRdRemain [4]		
			3: EPnRdRemain [3]		
			2: EPnRdRemain [2]		
			1: EPnRdRemain [1]		
			0: EPnRdRemain [0]		

74h.Bit7

RdRemainValid

EPx{x=0,a-c}Join.JoinCPU_Rd ビットによって、CPU I/F にエンドポイントがジョインされかつ、EPnRdRemain 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 EPnRdRemain の値は無効です。

74h.Bit6-4

Reserved

74h.Bit3-0, 75h.Bit7-0

EPnRdRemain [11:0]

EPx{x=0,a-c}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、EPnRdRemain_H レジスタと EPnRdRemain_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、EPnRdRemain_H レジスタを先にアクセスして下さい。

7. レジスタ

7.2.95. 76h EPnWrRemain_H (EPn FIFO Write Remain High)

7.2.96. 77h EPnWrRemain_L (EPn FIFO Write Remain Low)

Address	Register Name	R / W	Bit Symbol	Description		Reset
76h	EPnWrRemain_H		7:	0:	1:	00h
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
		R	3: EPnWrRemain [11]	Endpoint n FIFO Write Remain High		
			2: EPnWrRemain [10]			
			1: EPnWrRemain [9]			
			0: EPnWrRemain [8]			

Address	Register Name	R / W	Bit Symbol	Description	Reset
77h	EPnWrRemain_L	R	7: EPnWrRemain [7]	Endpoint n FIFO Write Remain Low	00h
			6: EPnWrRemain [6]		
			5: EPnWrRemain [5]		
			4: EPnWrRemain [4]		
			3: EPnWrRemain [3]		
			2: EPnWrRemain [2]		
			1: EPnWrRemain [1]		
			0: EPnWrRemain [0]		

77h.Bit7-4

Reserved

77h.Bit3-0, 78h.Bit7-0

EPnWrRemain [11:0]

EPx{x=0,a-c}Join.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の 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、EPnWrRemain_H レジスタと EPnWrRemain_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、EPnWrRemain_H レジスタを先にアクセスして下さい。

7.2.97. 78h DescAdrs_H (Descriptor Address High)

7.2.98. 79h DescAdrs_L (Descriptor Address Low)

Address	Register Name	R / W	Bit Symbol	Description		Reset
78h	DescAdrs_H		7:	0:	1:	00h
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
		R / W	3: DescAdrs [11]	Descriptor Address		
			2: DescAdrs [10]			
			1: DescAdrs [9]			
			0: DescAdrs [8]			

Address	Register Name	R / W	Bit Symbol	Description	Reset
79h	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。

78h.Bit7-4 Reserved

78h.Bit4-0, 79h.Bit7-0 DescAdrs [11:0]

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 0x9FF(2.5kByte)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、EP0Control レジスタの ReplyDescriptor の項を参照して下さい。

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、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。エンドポイント EP0 の予約された領域の終了アドレス(0x040)から CBW 領域の先頭アドレス(0x190)までの間が適切です。

Descriptor Address を参照する場合は、DescAdrs_H,DescAdrs_L の順に読み出して下さい。

7. レジスタ

7.2.99. 7Ah DescSize_H (Descriptor Size High)

7.2.100. 7Bh DescSize_L (Descriptor Size Low)

Address	Register Name	R / W	Bit Symbol	Description		Reset
7Ah	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]			

Address	Register Name	R / W	Bit Symbol	Description	Reset
7Bh	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。

7Ah.Bit7-2

Reserved

7Ah.Bit1-0, 7Bh.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。エンドポイント EP0 の予約された領域の終了アドレス(0x040)から CBW 領域の先頭アドレス(0x190)までの間を使用するようにして下さい。

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。

7.2.101. 7Ch EPnFIFO_ByteRd(EPn FIFO Byte Read)

Address	Register Name	R / W	Bit Symbol	Description	Reset
7Ch	EPnFIFO_ByteRd	R	7: EPnFIFO_ByteRd [7]	Endpoint n FIFO Byte Read	XXh
			6: EPnFIFO_ByteRd [6]		
			5: EPnFIFO_ByteRd [5]		
			4: EPnFIFO_ByteRd [4]		
			3: EPnFIFO_ByteRd [3]		
			2: EPnFIFO_ByteRd [2]		
			1: EPnFIFO_ByteRd [1]		
			0: EPnFIFO_ByteRd [0]		

Bit7-0 EPnFIFO_ByteRd [7:0]

EPx{x=0,a-c}Join.JoinCPU_Rd ビットがセットされているエンドポイント 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 EPnRdRemain_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。

7. レジスタ

7.2.102. 7Dh～7Fh Reserved ()

Address	Register Name	R / W	Bit Symbol	Description		Reset
7Dh -7Fh	Reserved		7:	0:	1:	XXh
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
			0:	0:	1:	

7.2.103. 80h DMA0_FIFO_Control (DMA0 FIFO Control)

Address	Register Name	R / W	Bit Symbol	Description		Reset
80h	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

7. レジスタ

7.2.104. 81h DMA0_Config (DMA0 Config)

Address	Register Name	R / W	Bit Symbol	Description		Reset
81h	DMA0_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: DMA0 Inactive	1: DMA0 Active	
			2:	0:	1:	
		R / W	1: ReqAssertCount [1]	Request Assert Count		
			0: ReqAssertCount [0]			

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リッジとして DMA 動作します。

Bit5-4 Reserved

Bit3 ActiveDMA

DMA0 をアクティブにします。DMA0 を使用する際にはこのビットを"1"にセットして下さい。

0: DMA0 無効

1: DMA0 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	8bit mode
0b00	Normal	Normal
0b01	16Byte(8Count)	16Byte(16Count)
0b10	32Byte(16Count)	32Byte(32Count)
0b11	64Byte(32Count)	64Byte(64Count)

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。

7.2.105. 82h DMA0_Control (DMA0 Control)

Address	Register Name	R / W	Bit Symbol	Description		Reset
82h	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	
			3:	0:	1:	
			2:	0:	1:	
		W	1: DMA_Stop	0: Do nothing	1: Finish DMA	
			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、EPx{x=0,a-c}Join.JoinDMA0_Rd,Wr 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。

Bit3-2 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。

7. レジスタ

7.2.106. 83h Reserved ()

Address	Register Name	R / W	Bit Symbol	Description		Reset
83h	Reserved		7:	0:	1:	XXh
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
			0:	0:	1:	

7.2.107. 84h DMA0_Remain_H (DMA0 FIFO Remain High)

7.2.108. 85h DMA0_Remain_L (DMA0 FIFO Remain Low)

Address	Register Name	R / W	Bit Symbol	Description		Reset	
84h	DMA0_Remain_H		7:	0:	1:	00h	
			6:	0:	1:		
			5:	0:	1:		
			4:	0:	1:		
		R	3: DMA_Remain [11]		DMA FIFO Remain High		
			2: DMA_Remain [10]				
			1: DMA_Remain [9]				
			0: DMA_Remain [8]				

Address	Register Name	R / W	Bit Symbol	Description	Reset
85h	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]		

84h.Bit7-4

Reserved

84h.Bit3-0, 85h.Bit7-0

DMA_Remain [11:0]

EPx{x=0,a-c}Join.JoinDMA0_Rd ビットによって DMA に接続しているエンドポイントの FIFO 内の残りデータ数を示します。

または、EPx{x=0,a-c}Join.JoinDMA0_Wr ビットによって DMA に接続しているエンドポイントの FIFO 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。

7. レジスタ

7.2.109. 86h～87h Reserved ()

Address	Register Name	R / W	Bit Symbol	Description		Reset
86h -87h	Reserved		7:	0:	1:	XXh
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
			0:	0:	1:	

7.2.110. 88h DMA0_Count_HH (DMA0 Transfer Byte Counter High/High)

7.2.111. 89h DMA0_Count_HL (DMA0 Transfer Byte Counter High/Low)

7.2.112. 8Ah DMA0_Count_LH (DMA0 Transfer Byte Counter Low/High)

7.2.113. 8Bh DMA0_Count_LL (DMA0 Transfer Byte Counter Low/Low)

Address	Register Name	R / W	Bit Symbol	Description	Reset
88h	DMA0_Count_HH	R / W	7: DMA_Count [31]	DMA Transfer Byte Counter	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]		

Address	Register Name	R / W	Bit Symbol	Description	Reset
89h	DMA0_Count_HL	R / W	7: DMA_Count [23]	DMA Transfer Byte Counter	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]		

Address	Register Name	R / W	Bit Symbol	Description	Reset
8Ah	DMA0_Count_LH	R / W	7: DMA_Count [15]	DMA Transfer Byte Counter	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]		

Address	Register Name	R / W	Bit Symbol	Description	Reset
8Bh	DMA0_Count_LL	R / W	7: DMA_Count [7]	DMA Transfer Byte Counter	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

7. レジスタ

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。

7.2.114. 8Ch DMA0_RdData_H (DMA0 Read Data High)

7.2.115. 8Dh DMA0_RdData_L (DMA0 Read Data Low)

Address	Register Name	R / W	Bit Symbol	Description	Reset
8Ch	DMA0_RdData_H	R	7: DMA_RdData [15]	DMA Read Data High	XXh
			6: DMA_RdData [14]		
			5: DMA_RdData [13]		
			4: DMA_RdData [12]		
			3: DMA_RdData [11]		
			2: DMA_RdData [10]		
			1: DMA_RdData [9]		
			0: DMA_RdData [8]		

Address	Register Name	R / W	Bit Symbol	Description	Reset
8Dh	DMA0_RdData_L	R	7: DMA_RdData [7]	DMA Read Data Low	XXh
			6: DMA_RdData [6]		
			5: DMA_RdData [5]		
			4: DMA_RdData [4]		
			3: DMA_RdData [3]		
			2: DMA_RdData [2]		
			1: DMA_RdData [1]		
			0: DMA_RdData [0]		

8Ch.Bit7-0, 8Dh.Bit7-0 DMA_RdData [15:0]

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 EPx{x=0,a-c}Join.JoinDMA0_Rd ビットによって DMA に接続しているエンドポイントの FIFO データを読み出す事が出来ます。

8bit Mode で動作する場合は、DMA0_RdData_H, DMA0_RdData_L いずれにアクセスしても同様に DMA アクセスする事が出来ます。

7.2.116. 8Eh DMA0_WrData_H (DMA0 Write Data High)

7.2.117. 8Fh DMA0_WrData_L (DMA0 Write Data Low)

Address	Register Name	R / W	Bit Symbol	Description	Reset
8Eh	DMA0_WrData_H	W	7: DMA_WrData [15]	DMA Write Data High	XXh
			6: DMA_WrData [14]		
			5: DMA_WrData [13]		
			4: DMA_WrData [12]		
			3: DMA_WrData [11]		
			2: DMA_WrData [10]		
			1: DMA_WrData [9]		
			0: DMA_WrData [8]		

Address	Register Name	R / W	Bit Symbol	Description	Reset
8Fh	DMA0_WrData_L	W	7: DMA_WrData [7]	DMA Write Data Low	XXh
			6: DMA_WrData [6]		
			5: DMA_WrData [5]		
			4: DMA_WrData [4]		
			3: DMA_WrData [3]		
			2: DMA_WrData [2]		
			1: DMA_WrData [1]		
			0: DMA_WrData [0]		

8Eh.Bit7-0, 8Fh.Bit7-0 DMA_WrData [15:0]

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 EPx{x=0,a-c}Join.JoinDMA0_Wr ビットによって DMA に接続しているエンドポイントの FIFO にデータを書き込む事が出来ます。

8bit Mode で動作する場合は、DMA0_WrData_H, DMA0_WrData_L いずれにアクセスしても同様に DMA アクセスする事が出来ます。

7. レジスタ

7.2.118. 90h DMA1_FIFO_Control (DMA1 FIFO Control)

Address	Register Name	R / W	Bit Symbol	Description		Reset
90h	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

7.2.119. 91h DMA1_Config (DMA0 Config)

Address	Register Name	R / W	Bit Symbol	Description		Reset
91h	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

DMA1 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 DMA1_RdData/DMA1_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 を **アクティブ** にします。DMA0 を使用する際にはこのビットを"1"にセットして下さい。

- 0: DMA1 無効
1: DMA1 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	8bit mode
0b00	Normal	Normal
0b01	16Byte(8Count)	16Byte(16Count)
0b10	32Byte(16Count)	32Byte(32Count)
0b11	64Byte(32Count)	64Byte(64Count)

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。

7. レジスタ

7.2.120. 92h DMA1_Control (DMA1 Control)

Address	Register Name	R / W	Bit Symbol	Description	Reset
92h	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	
			3:	0: 1:	
			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、EPx{x=0,a-c}Join.JoinDMA1_Rd,Wr 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。

Bit3-2 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。

7.2.121. 93h Reserved ()

Address	Register Name	R / W	Bit Symbol	Description		Reset
93h	Reserved		7:	0:	1:	XXh
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
			0:	0:	1:	

7. レジスタ

7.2.122. 94h DMA1_Remain_H (DMA1 FIFO Remain High)

7.2.123. 95h DMA1_Remain_L (DMA1 FIFO Remain Low)

Address	Register Name	R / W	Bit Symbol	Description		Reset	
94h	DMA1_Remain_H		7:	0:	1:	00h	
			6:	0:	1:		
			5:	0:	1:		
			4:	0:	1:		
		R	3: DMA_Remain [11]		DMA FIFO Remain High		
			2: DMA_Remain [10]				
			1: DMA_Remain [9]				
			0: DMA_Remain [8]				

Address	Register Name	R / W	Bit Symbol	Description	Reset
95h	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]		

94h.Bit7-4

Reserved

94h.Bit3-0, 95h.Bit7-0

DMA_Remain [11:0]

EPx{x=0,a-c}Join.JoinDMA1_Rd ビットによって DMA1 に接続しているエンドポイントの FIFO 内の残りデータ数を示します。

または、EPx{x=0,a-c}Join.JoinDMA1_Wr ビットによって DMA1 に接続しているエンドポイントの FIFO 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。

7.2.124. 96h~97h Reserved ()

Address	Register Name	R / W	Bit Symbol	Description		Reset
96h -97h	Reserved		7:	0:	1:	XXh
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
			0:	0:	1:	

7. レジスタ

7.2.125. 98h DMA1_Count_HH (DMA1 Transfer Byte Counter High/High)

7.2.126. 99h DMA1_Count_HL (DMA1 Transfer Byte Counter High/Low)

7.2.127. 9Ah DMA1_Count_LH (DMA1 Transfer Byte Counter Low/High)

7.2.128. 9Bh DMA1_Count_LL (DMA1 Transfer Byte Counter Low/Low)

Address	Register Name	R / W	Bit Symbol	Description	Reset
98h	DMA1_Count_HH	R / W	7: DMA_Count [31]	DMA Transfer Byte Counter	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]		

Address	Register Name	R / W	Bit Symbol	Description	Reset
99h	DMA1_Count_HL	R / W	7: DMA_Count [23]	DMA Transfer Byte Counter	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]		

Address	Register Name	R / W	Bit Symbol	Description	Reset
9Ah	DMA1_Count_LH	R / W	7: DMA_Count [15]	DMA Transfer Byte Counter	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]		

Address	Register Name	R / W	Bit Symbol	Description	Reset
9Bh	DMA1_Count_LL	R / W	7: DMA_Count [7]	DMA Transfer Byte Counter	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]		

98h-9Bh.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。

7. レジスタ

7.2.129. 9Ch DMA1_RdData_H (DMA1 Read Data High)

7.2.130. 9Dh DMA1_RdData_L (DMA1 Read Data Low)

Address	Register Name	R / W	Bit Symbol	Description	Reset
9Ch	DMA1_RdData_H	R	7: DMA_RdData [15]	DMA Read Data High	XXh
			6: DMA_RdData [14]		
			5: DMA_RdData [13]		
			4: DMA_RdData [12]		
			3: DMA_RdData [11]		
			2: DMA_RdData [10]		
			1: DMA_RdData [9]		
			0: DMA_RdData [8]		

Address	Register Name	R / W	Bit Symbol	Description	Reset
9Dh	DMA1_RdData_L	R	7: DMA_RdData [7]	DMA Read Data Low	XXh
			6: DMA_RdData [6]		
			5: DMA_RdData [5]		
			4: DMA_RdData [4]		
			3: DMA_RdData [3]		
			2: DMA_RdData [2]		
			1: DMA_RdData [1]		
			0: DMA_RdData [0]		

9Ch.Bit7-0, 9Dh.Bit7-0 DMA_RdData [15:0]

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 EPx{x=0,a-c}Join.JoinDMA1_Rd ビットによって DMA に接続しているエンドポイントの FIFO データを読み出す事が出来ます。

8bit Mode で動作する場合は、DMA1_RdData_H, DMA1_RdData_L いずれにアクセスしても同様に DMA アクセスする事が出来ます。

7.2.131. 9Eh DMA1_WrData_H (DMA1 Write Data High)

7.2.132. 9Fh DMA1_WrData_L (DMA1 Write Data Low)

Address	Register Name	R / W	Bit Symbol	Description	Reset
9Eh	DMA1_WrData_H	W	7: DMA_WrData [15]	DMA Write Data High	XXh
			6: DMA_WrData [14]		
			5: DMA_WrData [13]		
			4: DMA_WrData [12]		
			3: DMA_WrData [11]		
			2: DMA_WrData [10]		
			1: DMA_WrData [9]		
			0: DMA_WrData [8]		

Address	Register Name	R / W	Bit Symbol	Description	Reset
9Fh	DMA1_WrData_L	W	7: DMA_WrData [7]	DMA Write Data Low	XXh
			6: DMA_WrData [6]		
			5: DMA_WrData [5]		
			4: DMA_WrData [4]		
			3: DMA_WrData [3]		
			2: DMA_WrData [2]		
			1: DMA_WrData [1]		
			0: DMA_WrData [0]		

9Eh.Bit7-0, 9Fh.Bit7-0 DMA_WrData [15:0]

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 EPx{x=0,a-c}Join.JoinDMA1_Wr ビットによって DMA に接続しているエンドポイントの FIFO にデータを書き込む事が出来ます。

8bit Mode で動作する場合は、DMA1_WrData_H, DMA1_WrData_L いずれにアクセスしても同様に DMA アクセスする事が出来ます。

7. レジスタ

7.2.133. A0h IDE_Status (IDE Status)

Address	Register Name	R / W	Bit Symbol	Description		Reset
A0h	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。

7.2.134. A1h IDE_Control (IDE Control)

Address	Register Name	R / W	Bit Symbol	Description		Reset
A1h	IDE_Control		7:	0:	1:	00h
		W	6: IDE_Clr	0: None	1: Clear IDE Circuit	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			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。

Bit5-1 **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_Cmp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_Cmpビットはセットされません。

7. レジスタ

7.2.135. A2h IDE_Config_0 (IDE Configuration 0)

Address	Register Name	R / W	Bit Symbol	Description		Reset
A2h	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 $50\mu\text{s}$ 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 $50\mu\text{s}$ 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 $400\mu\text{s}$ 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 $400\mu\text{s}$ 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

7.2.136. A3h IDE_Config_1 (IDE Configuration 1)

Address	Register Name	R / W	Bit Symbol	Description		Reset
A3h	IDE_Config_1	R/W	7: ActiveIDE	0: InActivated IDE Bus	1: Activate IDE Bus	00h
		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。初期状態では、このビットは"0"であることにご注意ください。Swap ビットの詳細につきまして、Appendix A を参照して下さい。

Bit1-0 Reserved

7. レジスタ

7.2.137. A4h IDE_Rmod (IDE Register Mode)

Address	Register Name	R / W	Bit Symbol	Description	Reset
A4h	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]		

CPU の 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: $5 \times 16.67\text{nS} = 83\text{nS}$

7.2.138. A5h IDE_Tmod (IDE Transfer Mode)

Address	Register Name	R / W	Bit Symbol	Description	Reset
A5h	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]		

CPU 及び DMA 転送時の 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: $5 \times 16.67\text{nS} = 83\text{nS}$

7. レジスタ

7.2.139. A6h IDE_Umod (IDE Ultra-DMA Transfer Mode)

Address	Register Name	R / W	Bit Symbol	Description		Reset
A6h	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]			

DMA 転送時の 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: $3 \times 16.67\text{nS} = 50\text{nS}$

7.2.140. A7h～A9h Reserved ()

Address	Register Name	R / W	Bit Symbol	Description		Reset
A7h -A9h	Reserved		7:	0:	1:	XXh
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
			0:	0:	1:	

7. レジスタ

7.2.141. AAh IDE_CRC_H (IDE CRC High)

7.2.142. ABh IDE_CRC_L (IDE CRC Low)

Address	Register Name	R / W	Bit Symbol	Description	Reset
AAh	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]		

Address	Register Name	R / W	Bit Symbol	Description	Reset
ABh	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。その際に IDE_CRC_H レジスタを先にアクセスして下さい。

7.2.143. ACh Reserved ()

Address	Register Name	R / W	Bit Symbol	Description		Reset
ACh	Reserved		7:	0:	1:	XXh
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
			0:	0:	1:	

7. レジスタ

7.2.144. ADh IDE_Count_H (IDE Transfer Byte Counter High)

7.2.145. AEh IDE_Count_M (IDE Transfer Byte Counter Middle)

7.2.146. AFh IDE_Count_L (IDE Transfer Byte Counter Low)

Address	Register Name	R / W	Bit Symbol	Description	Reset
ADh	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]		

Address	Register Name	R / W	Bit Symbol	Description	Reset
AEh	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]		

Address	Register Name	R / W	Bit Symbol	Description	Reset
AFh	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_Count_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。

7.2.147. B0h IDE_RegAdrs (IDE Register Address)

Address	Register Name	R / W	Bit Symbol	Description		Reset
B0h	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_H/L 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_H/L 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 CPU の 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

7. レジスタ

7.2.148. B1h Reserved ()

Address	Register Name	R / W	Bit Symbol	Description		Reset
B1h	Reserved		7:	0:	1:	XXh
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
			0:	0:	1:	

7.2.149. B2h IDE_RdRegValue_H (IDE Register Read Value High)**7.2.150. B3h IDE_RdRegValue_L (IDE Register Read Value Low)**

Address	Register Name	R / W	Bit Symbol	Description	Reset
B2h	IDE_RdRegValue_H	R	7: IDE_RdRegValue [15]	IDE_RdRegValue [15:8]	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]		

Address	Register Name	R / W	Bit Symbol	Description	Reset
B3h	IDE_RdRegValue_L	R	7: IDE_RdRegValue [7]	IDE_RdRegValue [7:0]	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 CPU の 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_H と IDE_RdRegValue_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_RdRegValue_H レジスタを先にアクセスして下さい。

7. レジスタ

7.2.151. B4h IDE_WrRegValue_H (IDE Register Write Value High)

7.2.152. B5h IDE_WrRegValue_L (IDE Register Write Value Low)

Address	Register Name	R / W	Bit Symbol	Description	Reset
B4h	IDE_WrRegValue_H	R/W	7: IDE_WrRegValue [15]	IDE_WrRegValue [15:8]	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]		

Address	Register Name	R / W	Bit Symbol	Description	Reset
B5h	IDE_WrRegValue_L	R/W	7: IDE_WrRegValue [7]	IDE_WrRegValue [7:0]	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。

7.2.153. B6h IDE_SeqWrRegControl (IDE Sequential Register Write Control)

Address	Register Name	R / W	Bit Symbol	Description		Reset
B6h	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

7. レジスタ

7.2.154. B7h IDE_SeqWrRegCnt (IDE Sequential Register Write Counter)

Address	Register Name	R / W	Bit Symbol	Description		Reset
B7h	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。

7.2.155. B8h IDE_SeqWrRegAdrs (IDE Sequential Register Write Address FIFO)

Address	Register Name	R / W	Bit Symbol	Description		Reset
B8h	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。

7. レジスタ

7.2.156. B9h IDE_SeqWrRegValue (IDE Sequential Register Write Value FIFO)

Address	Register Name	R / W	Bit Symbol	Description	Reset
B9h	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(データの書き込み順は IDE_Config_1.Swap ビットが"1"である時、下位バイト=HDD[15:8]・上位バイト=HDD[7:0]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。

7.2.157. BAh~BBh Reserved ()

Address	Register Name	R / W	Bit Symbol	Description		Reset
BAh -BBh	Reserved		7:	0:	1:	XXh
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
			0:	0:	1:	

7. レジスタ

7.2.158. BCh IDE_RegConfig (IDE Register Configuration)

Address	Register Name	R / W	Bit Symbol	Description		Reset
BCh	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

7.2.159. BDh~BFh Reserved ()

Address	Register Name	R / W	Bit Symbol	Description		Reset
BDh -BFh	Reserved		7:	0:	1:	XXh
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
			0:	0:	1:	

7. レジスタ

- 7.2.160. C0h RAM_Rd_00 (RAM Read 00)
- 7.2.161. C1h RAM_Rd_01 (RAM Read 01)
- 7.2.162. C2h RAM_Rd_02 (RAM Read 02)
- 7.2.163. C3h RAM_Rd_03 (RAM Read 03)
- 7.2.164. C4h RAM_Rd_04 (RAM Read 04)
- 7.2.165. C5h RAM_Rd_05 (RAM Read 05)
- 7.2.166. C6h RAM_Rd_06 (RAM Read 06)
- 7.2.167. C7h RAM_Rd_07 (RAM Read 07)
- 7.2.168. C8h RAM_Rd_08 (RAM Read 08)
- 7.2.169. C9h RAM_Rd_09 (RAM Read 09)
- 7.2.170. CAh RAM_Rd_0A (RAM Read 0A)
- 7.2.171. CBh RAM_Rd_0B (RAM Read 0B)
- 7.2.172. CCh RAM_Rd_0C (RAM Read 0C)
- 7.2.173. CDh RAM_Rd_0D (RAM Read 0D)
- 7.2.174. CEh RAM_Rd_0E (RAM Read 0E)
- 7.2.175. CFh RAM_Rd_0F (RAM Read 0F)
- 7.2.176. D0h RAM_Rd_10 (RAM Read 10)
- 7.2.177. D1h RAM_Rd_11 (RAM Read 11)
- 7.2.178. D2h RAM_Rd_12 (RAM Read 12)
- 7.2.179. D3h RAM_Rd_13 (RAM Read 13)
- 7.2.180. D4h RAM_Rd_14 (RAM Read 14)
- 7.2.181. D5h RAM_Rd_15 (RAM Read 15)
- 7.2.182. D6h RAM_Rd_16 (RAM Read 16)
- 7.2.183. D7h RAM_Rd_17 (RAM Read 17)
- 7.2.184. D8h RAM_Rd_18 (RAM Read 18)
- 7.2.185. D9h RAM_Rd_19 (RAM Read 19)
- 7.2.186. DAh RAM_Rd_1A (RAM Read 1A)
- 7.2.187. DBh RAM_Rd_1B (RAM Read 1B)
- 7.2.188. DCh RAM_Rd_1C (RAM Read 1C)
- 7.2.189. DDh RAM_Rd_1D (RAM Read 1D)
- 7.2.190. DEh RAM_Rd_1E (RAM Read 1E)
- 7.2.191. DFh RAM_Rd_1F (RAM Read 1F)

Address	Register Name	R / W	Bit Symbol	Description	Reset
C0h -DFh	RAM_Rd_00 ~ RAM_Rd_1F	R	7: RAM Read_xx [7]	RAM Read	00h
			6: RAM Read_xx [6]		
			5: RAM Read_xx [5]		
			4: RAM Read_xx [4]		
			3: RAM Read_xx [3]		
			2: RAM Read_xx [2]		
			1: RAM Read_xx [1]		
			0: RAM Read_xx [0]		

C0h-DFh.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。

7.2.192. E0h RAM_RdAdrs_H (RAM Read Address High)**7.2.193. E1h RAM_RdAdrs_L (RAM Read Address Low)**

Address	Register Name	R / W	Bit Symbol	Description		Reset
E0h	RAM_RdAdrs_H		7:	0:	1:	00h
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
		R / W	3: RAM Read Address [11]	RAM Read Address		
			2: RAM Read Address [10]			
			1: RAM Read Address [9]			
			0: RAM Read Address [8]			

Address	Register Name	R / W	Bit Symbol	Description	Reset
E1h	RAM_RdAdrs_L	R / W	7: RAM Read Address [7]	RAM Read Address	00h
			6: RAM Read Address [6]		
			5: RAM Read Address [5]		
			4: RAM Read Address [4]		
			3: RAM Read Address [3]		
			2: RAM Read Address [2]		
			1:		
			0:		

E0h.Bit7-4**Reserved****E0h.Bit3-0, 75h.Bit7-2****RAM Read Address [11:2]**

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 FIFO_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。

E1h.Bit1-0**Reserved**

7. レジスタ

7.2.194. E2h RAM_RdControl (RAM Read Control)

Address	Register Name	R / W	Bit Symbol	Description		Reset
E2h	RAM_RdControl	R / W	7: RAM_GoRdCBW	0: Do nothing	1: RAM Read CBW 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

CBW 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。

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_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_Rd_00~RAM_Rd_1E レジスタの値が有効になると、FIFO_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_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 FIFO_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 ビットと同時にセットすると、RAM_GoRdCBW ビットの機能が優先されます。

Bit5-0 Reserved

7.2.195. E3h RAM_RdCount (RAM Read Counter)

Address	Register Name	R / W	Bit Symbol	Description	Reset
E3h	RAM_RdCount		7:		00h
			6:		
		R / W	5: RAM_RdCount [5]	RAM Read Counter	
			4: RAM_RdCount [4]		
			3: RAM_RdCount [3]		
			2: RAM_RdCount [2]		
			1:		
			0:		

Bit7-0 RAM_RdCount [7:0]

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 FIFO_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。

7. レジスタ

7.2.196. E4h～EAh Reserved ()

Address	Register Name	R / W	Bit Symbol	Description		Reset
E4h -EAh	Reserved		7:	0:	1:	XXh
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
			0:	0:	1:	

7.2.197. EBh *ModeProtect*(*Mode Protection*)

Address	Register Name	R / W	Bit Symbol	Description	Reset
EBh	<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。

7. レジスタ

7.2.198. ECh Reserved ()

Address	Register Name	R / W	Bit Symbol	Description		Reset
ECh	Reserved		7:	0:	1:	XXh
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
			0:	0:	1:	

7.2.199. EDh *ClkSelect* (Clock Select)

Address	Register Name	R / W	Bit Symbol	Description		Reset
EDh	<i>ClkSelect</i>	R / W	7: <i>xActIDE_Term</i>	0: IDE Termination ON	1: IDE Termination OFF	01h
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
		R / W	0: <i>ClkSelect</i>	0: 12MHz	1: 24MHz	

Bit7 *xActIDE_Term*

IDE ポートのプルアップ、プルダウンを ON/OFF します。

0: IDE Termination ON

1: IDE Termination OFF

Bit6-1 **Reserved****Bit0** *ClkSelect*

本 LSI で使用するクロックを選択します。

0: 12MHz

1: 24MHz

7. レジスタ

7.2.200. EEh Reserved ()

Address	Register Name	R / W	Bit Symbol	Description		Reset
EEh	Reserved		7:	0:	1:	XXh
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
			0:	0:	1:	

7.2.201. EFh ChipConfig (Chip Configuration)

Address	Register Name	R / W	Bit Symbol	Description		Reset
EFh	ChipConfig	R / W	7: IntLevel	0: Low Active	1: High Active	00h
		R / W	6: IntMode	0: 1/0 mode	1: Hi-z/0 mode	
		R / W	5: DREQ_Level	0: Low Active	1: High Active	
		R / W	4: DACK_Level	0: Low Active	1: High Active	
		R / W	3: CS_Mode	0: DACK mode	1: CS mode	
		R / W	2: CPU_Swap	0: Do nothing	1: Bus Swap	
		R / W	1: BusMode	0: XWRH/L mode	1: XBEH/L mode	
		R / W	0: Bus8x16	0: 16bit mode	1: 8bit mode	

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。

- 0: 負論理
- 1: 正論理

Bit6 IntMode

XINT の出力モードを設定します。

- 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。

- 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。DMA を使用しない場合はこのビットの設定に対し、アクティブでないレベルに XDACK0,1 端子を固定して下さい。

- 0: 負論理
- 1: 正論理

Bit3 CS_Mode

XDACK0,1 信号を用いて DMA アクセスする場合の DMA0,1 の動作モードを設定します。DMAx{x=0,1}_Config, DMA_Mode="1" の場合、このビットの設定は不要です。デフォルト"0"のままでご使用ください。

- 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_Swap

16bit mode 時の CPU バスを設定します。8bit mode 時はこのビットをセットしないで下さい。

- 0: 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: 偶数アドレスを下位側、奇数アドレスを上位側とします。

このビットの設定は、レジスタ書き込み後、E9h 番地をリードする事により有効になります。ChipReset.ResetAll ビットにて、回路のリセットを行った場合、レジスタの値は初期化されますが、その設定が有効になるのは、上述同様に E9h 番地をリードした後になります。

Bit1-0 BusMode, Bus8x16

CPU の動作モードを設定します。

動作モード	bit1.BusMode	bit0.Bus8x16
16bit Strobe mode	0	0
16bit BE mode	1	*
8bit mode	0	1

7. レジスタ

7.2.202. F0h Reserved ()

Address	Register Name	R / W	Bit Symbol	Description		Reset
F0h	Reserved		7:	0:	1:	XXh
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
			0:	0:	1:	

7.2.203. F1h D_ModeControl (Device Mode Control)

Address	Register Name	R / W	Bit Symbol	Description		Reset
F1h	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"		

詳細は、Appendix C を参照して下さい。

7.2.204. F2h～FFh Reserved ()

Address	Register Name	R / W	Bit Symbol	Description		Reset
F2h -FFh	Reserved		7:	0:	1:	XXh
			6:	0:	1:	
			5:	0:	1:	
			4:	0:	1:	
			3:	0:	1:	
			2:	0:	1:	
			1:	0:	1:	
			0:	0:	1:	

8. 電気的特性

8.1. 絶対最大定格

(V_{SS}=0V)

項目	記号	定格値	単位
電源電圧	HVDD	VSS-0.3~4.0	V
	CVDD	VSS-0.3~4.0	V
	LVDD	VSS-0.3 ~ 2.5	V
	PVDD	VSS-0.3 ~ 2.5	V
入力電圧	HVI	VSS-0.3 ~ HVDD+0.5	V
	CVI	VSS-0.3 ~ CVDD+0.5	V
	PVI	VSS-0.3 ~ PVDD+0.5	V
	VVI ※1	VSS-0.3 ~ 6.0	V
出力電圧	HVO	VSS-0.3 ~ HVDD+0.5	V
出力電流／端子	IOUT	±10	mA
保存温度	Tstg	-65~150	°C

※1 VBUS

8.2. 推奨動作条件

項目	記号	MIN	TYP	MAX	単位
電源電圧	HVDD	3.00	3.30	3.60	V
	CVDD※1	3.00	3.30	3.60	V
	CVDD※2	1.65	1.80	1.95	V
	LVDD	1.65	1.80	1.95	V
	PVDD	1.65	1.80	1.95	V
入力電圧	HVI	-0.3	-	HVDD+0.3	V
	CVI	-0.3	-	CVDD+0.3	V
	PVI	-0.3		PVDD+0.5	V
	VVI	-0.3	-	6.0	V
周囲温度	T _a	-40	25	85	°C

※1 CPU IF 3.3V で使用時

※2 CPU IF 1.8V で使用時

本ICは下記順序で電源投入を行ってください。

LVDD,PVDD(内部) →HVDD, CVDD(IO 部)

また、本 IC は下記手順で電源オフを行って下さい。

HVDD, CVDD(IO 部) →LVDD, PVDD(内部)

注)

LVDD, PVDD が切断されている状態で HVDD, CVDD のみを継続的(1Sec 以上)に印可することは Chip の信頼性上問題がありますので避けてください。

8. 電氣的特性

8.3. DC 特性

DC 状態における入力特性(推奨動作条件による)

項目	記号	条件	MIN	TYP	MAX	単位
電源供給電流 ※1						
電源電流	IDDL	LVDD=1.8V(typ), 1.95V(max)	-	28	45	mA
	IDDH	HVDD=3.3V(typ), 3.6V(max)	-	14	23	mA
	IDDCH	CVDD=3.3V(typ), 3.6V(max)	-	1	4	mA
	IDDCL	CVDD=1.8V(typ), 1.95V(max)	-	0.7	2	mA
	IDDP	PVDD=1.8V(typ), 1.95V(max)	-	16	26	mA
静止電流 ※2						
電源電流	IDDS	VIN = HVDD or VSS LVDD = 1.95V HVDD=3.6V CVDD=3.6 PVDD=1.95V	-	-	80	μA
入力リーク						
入力リーク電流	IL	HVDD=3.3V CVDD=3.3V/1.8V LVDD=1.8V HVIH=HVDD CVIH=CVDD VIL=VSS	-5	-	5	μA

※1: typ 値は IDE ポートに HDD を接続し、PC との間でデータを送受信している状態(実転送レート 17MB/s)での測定値。max 値は同値からの見積値。

※2: Ta=25°C、双方向端子が入力状態である場合の静止電流値。

8. 電気的特性

DC 状態における入力特性(推奨動作条件による)(つづき)

項目	記号	条件	MIN	TYP	MAX	単位
力特性(LVCMOS)	端子名:	TDI, TCK, TRST, TMS				
"H"レベル入力電圧	VIH1	HVDD = 3.6V	2.1	-	-	V
"L"レベル入力電圧	VIL1	HVDD = 3.0V	-	-	0.9	V
入力特性(LVCMOS)	端子名:	CA [7:1], CD [15:0], XCS, XRD, XWRL, XWRH, XBEL, XDACK0, XDACK1, TEST, XRESET				
"H"レベル入力電圧	VIH2	CVDD=3.6V	2.1	-	-	V
"L"レベル入力電圧	VIL2	CVDD=3.0	-	-	0.9	V
"H"レベル入力電圧	VIH3	CVDD=1.95V	1.22	-	-	V
"L"レベル入力電圧	VIL3	CVDD=1.65V	-	-	0.62	V
入力特性(LVTTL)	端子名	HDD [15:0], HDMARQ, HIORDY, HINTRQ, XHDASP, XHPDIAG				
"H"レベル入力電圧	VIH4	HVDD=3.6V	1.9	-	-	V
"L"レベル入力電圧	VIL4	HVDD=3.0V	-	-	0.9	V
シュミット入力特性(USB:FS)	端子名:	DP, DM				
"H"レベルトリガ電圧	VT+ (USB)	HVDD = 3.6V	1.1	-	1.8	V
"L"レベルトリガ電圧	VT- (USB)	HVDD = 3.0V	1.0	-	1.5	V
ヒステリシス電圧	ΔV (USB)	HVDD= 3.0V	0.1	-	-	V
入力特性(USB:FS 差動入力)	端子名:	DP, DM のペア				
差動入力の感度	VDS (USB)	HVDD = 3.0V 差動入力電圧 0.8V~2.5V	-	-	0.2	V
入力特性(VBUS)	端子名:	VBUS				
"H"レベルトリガ電圧	VT+ (VBUS)	HVDD = 3.6V	1.86	-	2.85	V
"L"レベルトリガ電圧	VT- (VBUS)	HVDD = 3.0V	1.48	-	2.23	V
ヒステリシス電圧	ΔV (VBUS)	HVDD= 3.0V	0.31	-	0.64	V
プルダウン抵抗	RPLDV	VI=5.0V	110	125	150	k Ω
出力特性	端子名:	CD [15:0], XDREQ0, XDREQ1, XINT				
"H"レベル出力電圧	VOH1	CVDD = 3.0V IOH = -2.6mA	CVDD-0.4	-	-	V
"L"レベル出力電圧	VOL1	CVDD = 3.0V IOL = 2.7mA	-	-	VSS+0.4	V
"H"レベル出力電圧	VOH2	CVDD = 1.65V IOH = -1.3mA	CVDD-0.4	-	-	V
"L"レベル出力電圧	VOL2	CVDD = 1.65V IOL = 1.4mA	-	-	VSS+0.4	V
出力特性	端子名:	HDD [15:0], HDA [2:0], XHCS1, XHCS0, XHIOR, XHIOW, XHDMACK, XHRESET,				
"H"レベル出力電圧	VOH3	HVDD = 3.0V IOH = -5.2mA	HVDD-0.4	-	-	V
"L"レベル出力電圧	VOL3	HVDD = 3.0V IOL = 5.4mA	-	-	VSS+0.4	V
出力特性	端子名:	TDO				
"H"レベル出力電圧	VOH2	HVDD = 3.0V IOH = -2.6mA	HVDD-0.4	-	-	V
"L"レベル出力電圧	VOL2	HVDD = 3.0V	-	-	VSS+0.4	V

8. 電気的特性

			IOL = 2.7mA				
出力特性(USB:FS)		端子名:	DP, DM				
	"H"レベル出力電圧	VOH (USB)	HVDD = 3.0V	2.8	-	-	V
	"L"レベル出力電圧	VOL (USB)	HVDD = 3.6V	-	-	0.3	V

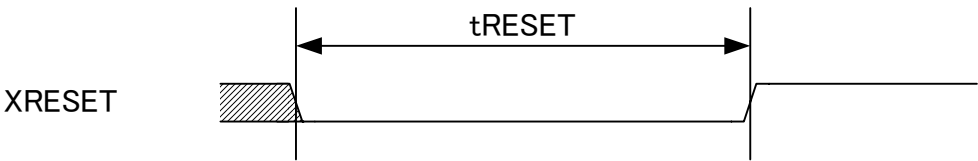
項目	記号	条件	MIN	TYP	MAX	単位
出力特性(USB:HS)		端子名:	DP, DM			
"H"レベル出力電圧	VHSOH (USB)	HVDD = 3.0V	360	-	-	mV
"L"レベル出力電圧	VHSOL (USB)	HVDD = 3.6V	-	-	10.0	mV
出力特性		端子名:	CA [15:0], XINT, HDD [15:0], HDA [2:0], XHCS1, XHCS0, XHDMACK, XHRESET			
OFF-STATE リーク電流	IOZ	HVDD = 3.3V CVDD=3.3V CVDD=1.8V VOH = HVDD/CVDD VOL = VSS	-2	-	2	μA
入力特性		端子名:	HDD[15:8], HDD[6:0], HIORDY, HINTRQ, XHDASP, XHPDIAG			
プルアップ抵抗	RPLU	HVDD=3.0V VIL=VSS	-16.4	-	-38.5	μA
入力特性		端子名:	HDD7, HDMARQ,			
プルダウン抵抗	RPLD	HVDD=3.0V VIH=VSS	17.2	-	38.7	μA

項目	記号	条件	MIN	TYP	MAX	単位
端子容量	端子名: 全入力端子					
入力端子容量	CI	f = 1MHz HVDD=CVDD=LVDD =VSS PVDD=PVSS	-	-	8	pF
端子容量	端子名: 全出力端子					
出力端子容量	CO	f = 1MHz HVDD=CVDD=LVDD =VSS PVDD=PVSS	-	-	8	pF
端子容量	端子名: 全入出力端子 (DP, DM を除く)					
入出力端子容量 1	CIO1	f = 1MHz HVDD=LVDD =VSS PVDD=PVSS	-	-	8	pF
端子容量	端子名: DP, DM					
入出力端子容量 2	CIO2	f = 1MHz HVDD=CVDD=LVDD =VSS PVDD=PVSS	-	-	8	pF

8. 電気的特性

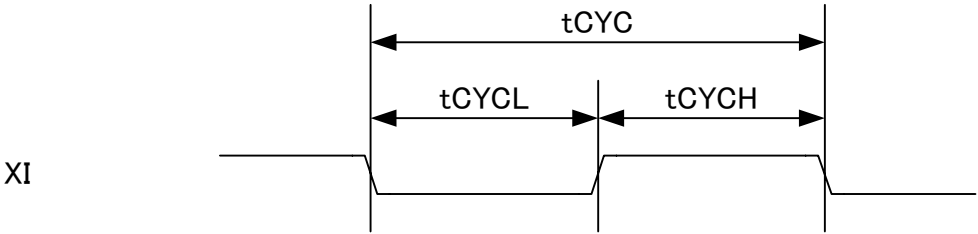
8.4. AC 特性

8.4.1. RESET タイミング



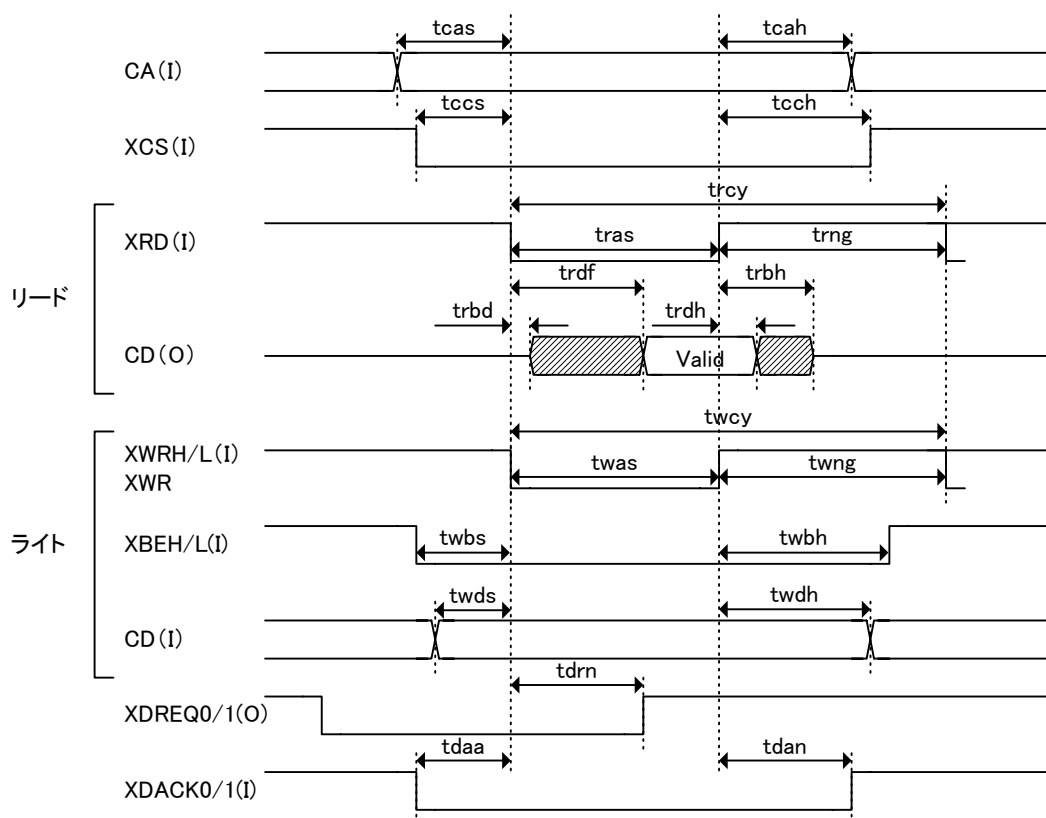
記号	説明	min	typ	max	単位
t_{RESET}	リセットパルス幅	40	–	–	ns

8.4.2. クロックタイミング



記号	説明	min	typ	max	単位
$t_{CYC}^{(*)}$	クロックサイクル(ClkSelect=0)	11.999	12	12.001	MHz
$t_{CYC}^{(*)}$	クロックサイクル(ClkSelect=1)	23.998	24	24.002	MHz
t_{CYCH} t_{CYCL}	クロックデューティ	45	–	55	%

8.4.3. CPU/DMA I/F アクセスタイミング



CLVDD=3.3V

(CL=30pF)

記号	項目	min	typ	max	unit
tcas	アドレスセットアップ時間	6	—	—	ns
tcah	アドレスホールド時間	6	—	—	ns
tccs	XCSセットアップ時間	6	—	—	ns
tcch	XCSホールド時間	6	—	—	ns
trcy	リードサイクル	75	—	—	ns
tras	リードストローブアサート時間	37	—	—	ns
trng	リードストローブネゲート時間	25	—	—	ns
trbd	リードデータ出力開始時間	1	—	—	ns
trdf	リードデータ確定時間	—	—	30	ns
trdh	リードデータホールド時間	1	—	—	ns
trbh	リードデータ出力遅延時間	—	—	7	ns
twcy	ライトサイクル	75	—	—	ns
twas	ライトストローブアサート時間	37	—	—	ns
twng	ライトストローブネゲート時間	25	—	—	ns
twbs	ライトバイトイネーブルセットアップ時間	6	—	—	ns
twbh	ライトバイトイネーブルホールド時間	6	—	—	ns
twds	ライトデータセットアップ時間	0	—	—	ns
twdh	ライトデータホールド時間	0	—	—	ns
tdrn	XDREQ0/1ネゲート遅延時間	—	—	50	ns
tdaa	XDACK0/1セットアップ時間	6	—	—	ns
tdan	XDACK0/1ホールド時間	6	—	—	ns

8. 電気的特性

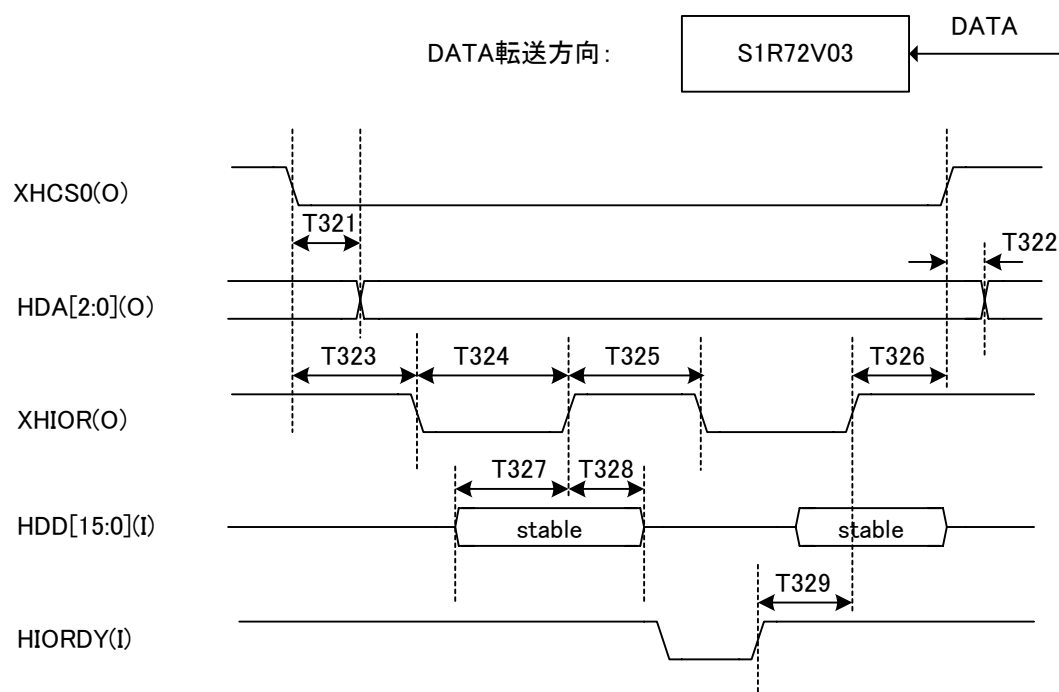
CLVDD=1.8V

(CL=30pF)

記号	項目	min	typ	max	unit
tcas	アドレスセットアップ時間	6	–	–	ns
tcah	アドレスホールド時間	6	–	–	ns
tccs	XCSセットアップ時間	6	–	–	ns
tcch	XCSホールド時間	6	–	–	ns
trcy	リードサイクル	80	–	–	ns
tras	リードストローブアサート時間	40	–	–	ns
trng	リードストローブネゲート時間	25	–	–	ns
trbd	リードデータ出力開始時間	1	–	–	ns
trdf	リードデータ確定時間	–	–	35	ns
trdh	リードデータホールド時間	1	–	–	ns
trbh	リードデータ出力遅延時間	–	–	8	ns
twcy	ライトサイクル	75	–	–	ns
twas	ライトストローブアサート時間	40	–	–	ns
twng	ライトストローブネゲート時間	25	–	–	ns
twbs	ライトバイトイネーブルセットアップ時間	6	–	–	ns
twbh	ライトバイトイネーブルホールド時間	6	–	–	ns
twds	ライトデータセットアップ時間	0	–	–	ns
twdh	ライトデータホールド時間	0	–	–	ns
tdrn	XDREQ0/1ネゲート遅延時間	–	–	60	ns
tdaa	XDACK0/1セットアップ時間	6	–	–	ns
tdan	XDACK0/1ホールド時間	6	–	–	ns

8.4.4. IDE I/F タイミング

8.4.4.1. PIO Read Timing

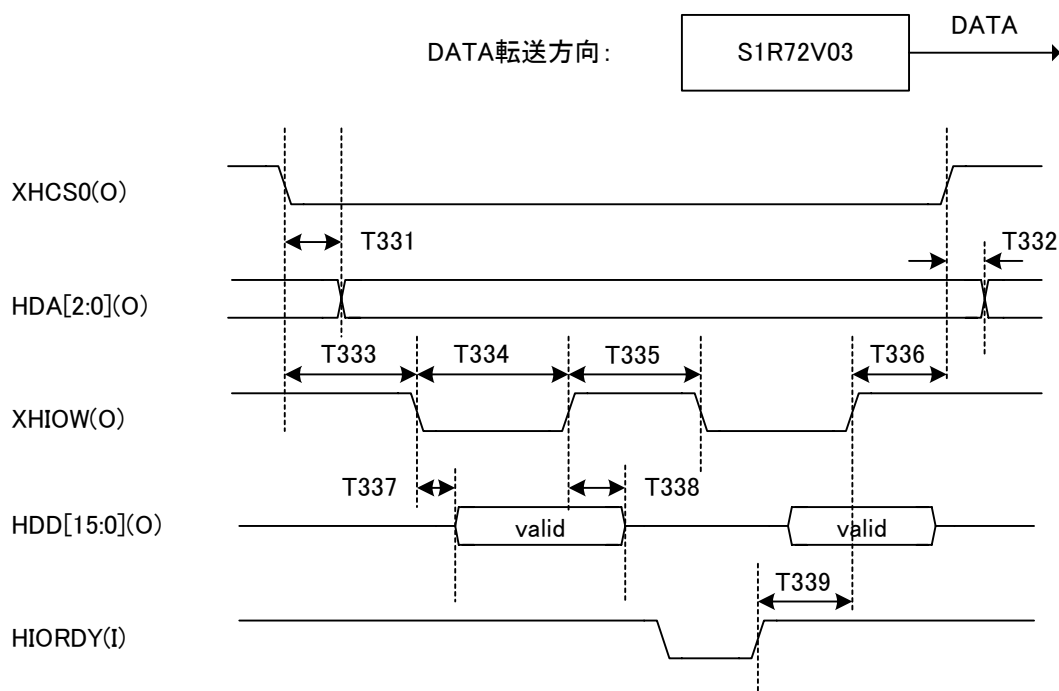


記号	説明	min	typ	max	単位
T321	XHCS0 ↓ → HDA HDA出力遅延時間	–	0	–	ns
T322	XHCS0 ↑ → HDA HDAホールド時間	–	0	–	ns
T323	XHCS0 ↓ → XHIOR ↓ XHCS0セットアップ時間	25	–	–	ns
T324	XHIOR ↓ → XHIOR ↑ XHIORアサートパルス幅	–	$(AP+4) * 16.7 - 3$	–	ns
T325	XHIOR ↑ → XHIOR ↓ XHIORネゲートパルス幅	–	$(NP+4) * 16.7 + 3$	–	ns
T326	XHIOR ↑ → XHCS0 ↑ XHCS0ホールド時間	10	–	–	ns
T327	HDD → XHIOR ↑ データセットアップ時間	10	–	–	ns
T328	XHIOR ↑ → HDD データホールド時間	0	–	–	ns
T329	HIORDYアサート → XHIOR ↑ XHIOR出力遅延時間	–	–	25	ns

*1: AP=IDE_Tmod.AssertPulseWidth, NP=IDE_Tmod.NegatePulseWidth
 詳細はレジスタ説明、“IDE Transfer Mode”を参照のこと

8. 電气的特性

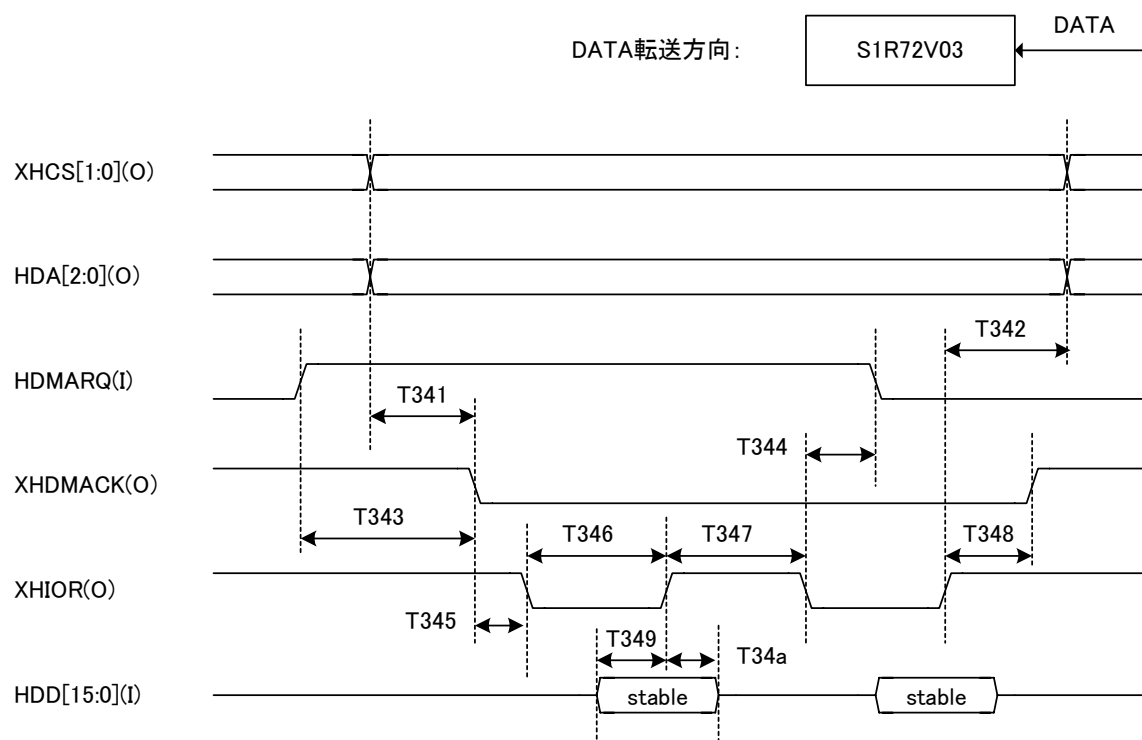
8.4.4.2. PIO Write Timing



記号	説明	min	typ	max	単位
T331	XHCS0 ↓ → HDA HDA出力遅延時間	–	0	–	ns
T332	XHCS0 ↑ → HDA HDAホールド時間	–	0	–	ns
T333	XHCS0 ↓ → XHIOW ↓ XHCS0セットアップ時間	25	–	–	ns
T334	XHIOW ↓ → XHIOW ↑ XHIOWアサートパルス幅	–	(AP+4) * 16.7 – 3	–	ns
T335	XHIOW ↑ → XHIOW ↓ XHIOWネゲートパルス幅	–	(NP+4) * 16.7 + 3	–	ns
T336	XHIOW ↑ → XHCS0 ↑ XHCS0ホールド時間	10	–	–	ns
T337	XHIOW ↓ → HDD データ出力遅延時間	0	–	10	ns
T338	XHIOW ↑ → HDD データバスネゲート時間	10	–	45	ns
T339	HIORDYアサート → XHIOW ↑ XHIOW出力遅延時間	–	–	25	ns

*1: AP=IDE_Tmod.AssertPulseWidth, NP=IDE_Tmod.NegatePulseWidth
詳細はレジスタ説明、“IDE Transfer Mode”を参照のこと

8.4.4.3.DMA Read Timing

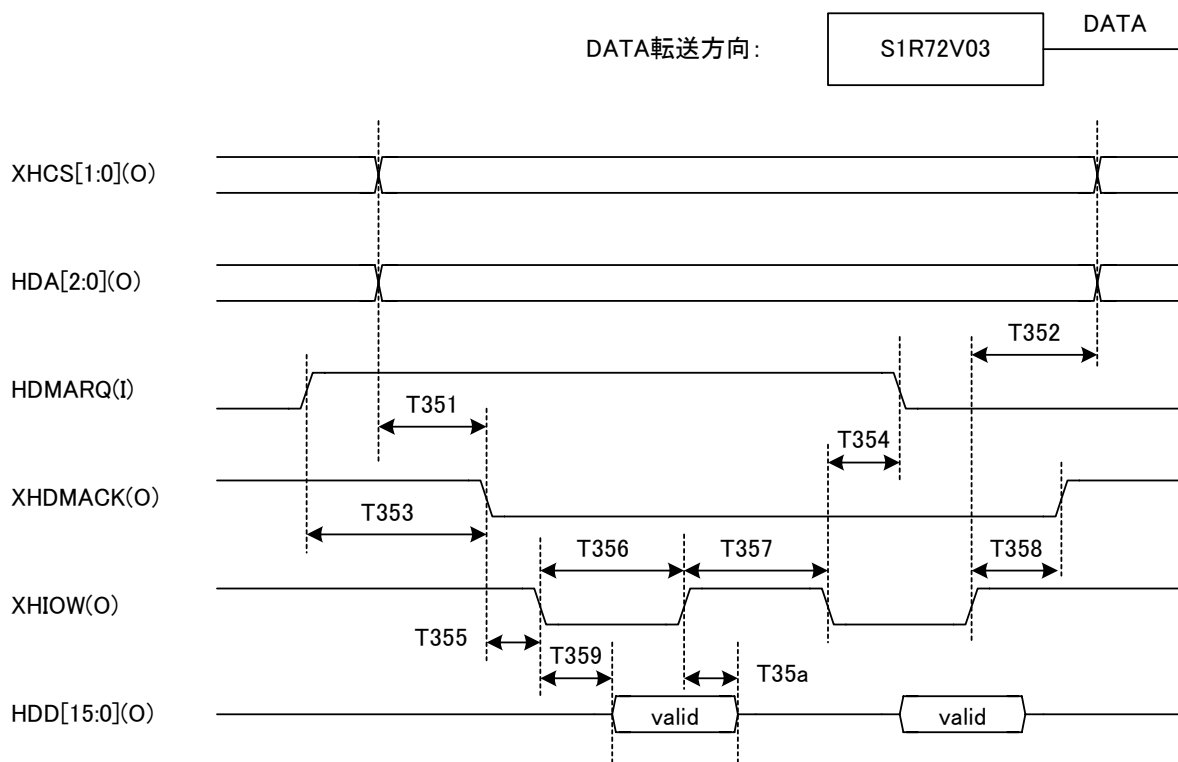


記号	説明	min	typ	max	単位
T341	XHCS ↑、HDA → XHDMACK ↓ アドレスセットアップ時間	70	–	–	ns
T342	XHIOR ↑ → XHCS ↑、HDA アドレスホールド時間	10	–	–	ns
T343	HDMARQ ↑ → XHDMACK ↓ XHDMACK応答時間	17	–	–	ns
T344	XHIOR ↓ → HDMARQネゲート HDMARQホールド時間	0	–	–	ns
T345	XHDMACK ↓ → XHIOR ↓ XHDMACKセットアップ時間	0	–	–	ns
T346	XHIOR ↓ → XHIOR ↑ XHIORアサートパルス幅	–	$(AP+4) * 16.7 - 3$	–	ns
T347	XHIOR ↑ → XHIOR ↓ XHIORネゲートパルス幅	–	$(NP+4) * 16.7 + 3$	–	ns
T348	XHIOR ↑ → XHDMACK ↑ XHDMACKホールド時間	5	–	90	ns
T349	HDD → XHIOR ↑ データセットアップ時間	10	–	–	ns
T34a	XHIOR ↑ → HDD データバスホールド時間	0	–	–	ns

*1:AP=IDE_Tmod.AssertPulseWidth, NP=IDE_Tmod.NegatePulseWidth
 詳細はレジスタ説明、“IDE Transfer Mode”を参照のこと

8. 電気的特性

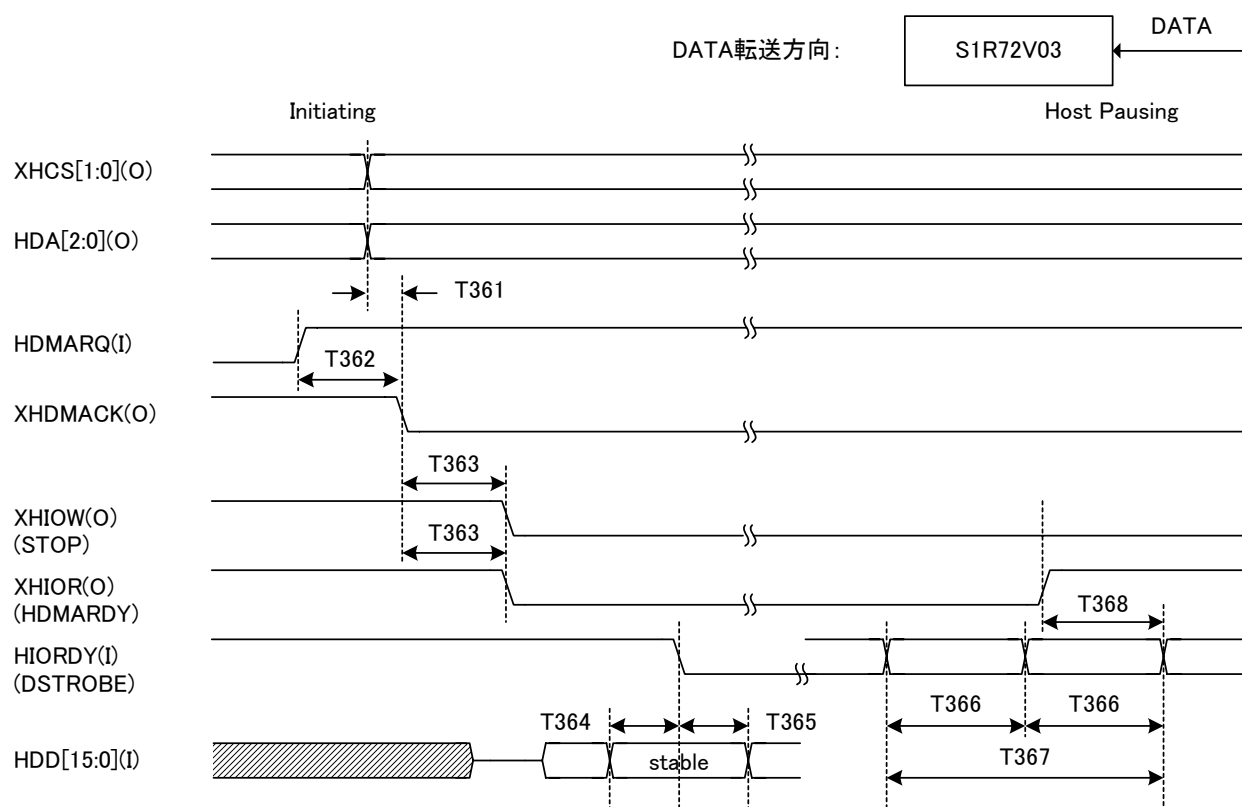
8.4.4.4.DMA Write Timing



記号	説明	min	typ	max	単位
T351	XHCS ↑、HDA → XHDMACK ↓ アドレスセットアップ時間	70	–	–	ns
T352	XHIOW ↑ → XHCS ↑、HDA アドレスホールド時間	10	–	–	ns
T353	HDMARQ ↑ → XHDMACK ↓ XHDMACK応答時間	17	–	–	ns
T354	XHIOW ↓ → HDMARQ ネグート HDMARQホールド時間	0	–	–	ns
T355	XHDMACK ↓ → XHIOW ↓ XHDMACKセットアップ時間	0	–	–	ns
T356	XHIOW ↓ → XHIOW ↑ XHIOWアサートパルス幅	–	(AP+4) * 16.7 – 3	–	ns
T357	XHIOW ↑ → XHIOW ↓ XHIOWネグートパルス幅	–	(NP+4) * 16.7 + 3	–	ns
T358	XHIOW ↑ → XHDMACK ↑ XHDMACKホールド時間	5	–	90	ns
T359	XHIOW ↓ → HDD データ出力遅延時間	0	–	10	ns
T35a	XHIOW ↑ → HDD データバスネグート時間	10	–	45	ns

*1:AP=IDE_Tmod.AssertPulseWidth, NP=IDE_Tmod.NegatePulseWidth
詳細はレジスタ説明、“IDE Transfer Mode”を参照のこと

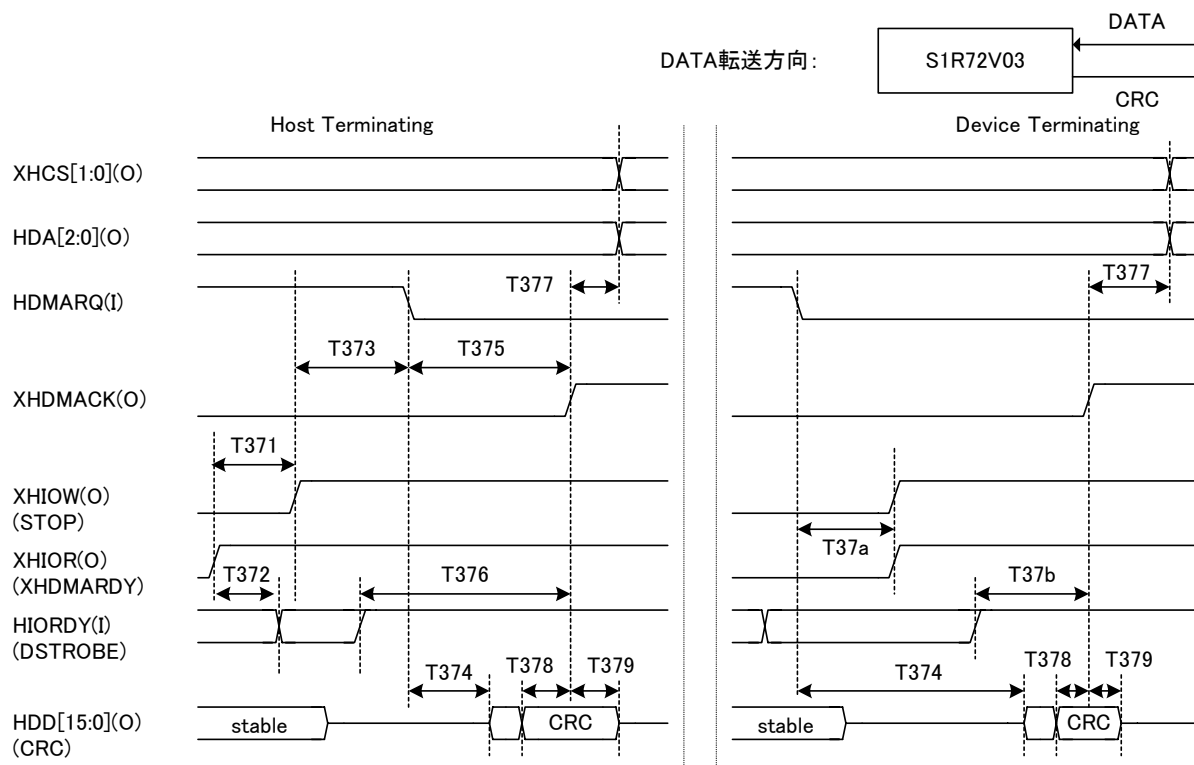
8.4.4.5. Ultra DMA Read Timing



記号	説明	min	typ	max	単位
T361	XHCS ↑、HDA → XHDMACK ↓ アドレスセットアップ時間	20	–	–	ns
T362	HDMARQ ↑ → XHDMACK ↓ XHDMACK応答時間	0	–	–	ns
T363	XHDMACK ↓ → XHIOR(W) ↓ エンベロップ時間	20	–	70	ns
T364	HDD → HIORDY データセットアップ時間	4	–	–	ns
T365	HIORDY → HDD データホールド時間	4	–	–	ns
T366	HIORDY → HIORDY HIORDYサイクル時間	15	–	–	ns
T367	HIORDY → HIORDY HIORDYサイクル時間 × 2	30	–	–	ns
T368	XHIOR ↑ → HIORDY 最終のSTROBE時間	–	–	IDE規格 t_{RFS}	ns

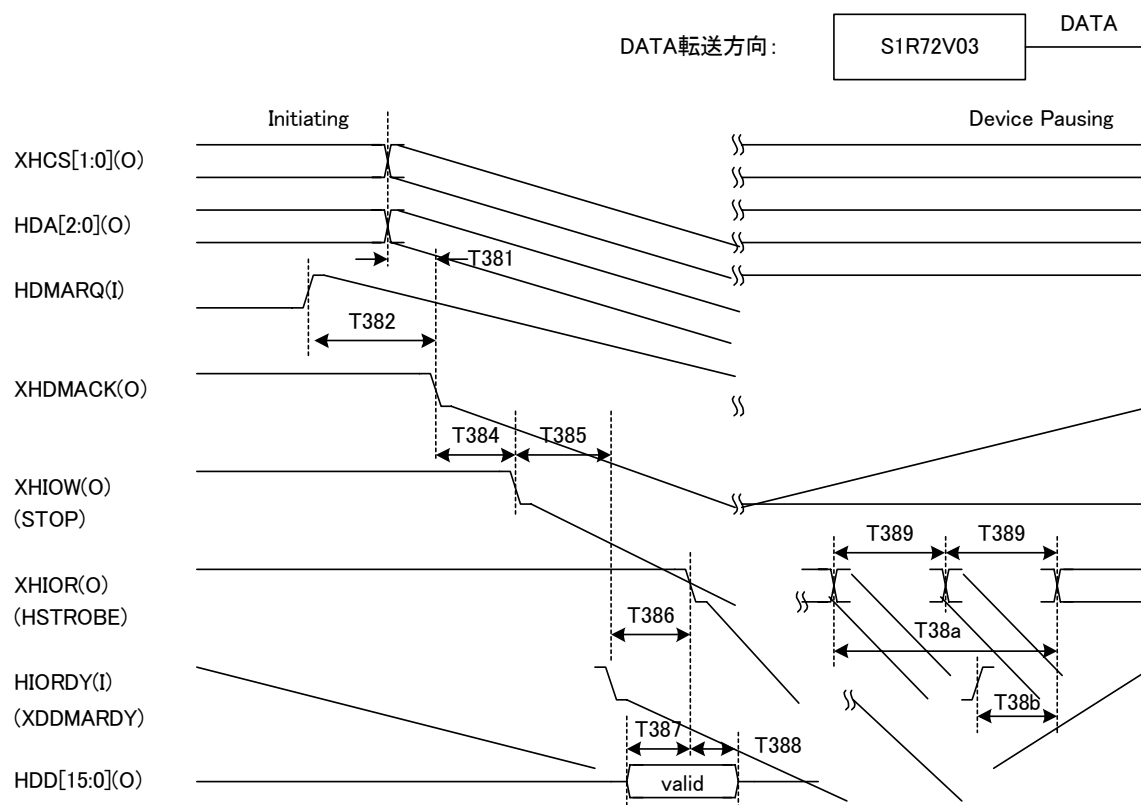
8. 電氣的特性

Ultra DMA Read Timing (つづき)



記号	説明	min	typ	max	単位
T371	XHIOR ↑ → XHIOW ↑ STOPアサートまでの時間	85	—	—	ns
T372	XHIOR ↑ → HIORDY 最終のSTROBE時間	—	—	IDE規格 t_{RFS}	ns
T373	XHIOW ↑ → HDMARQ ↓ 制限付きインターロック時間	—	—	IDE規格 t_{LI}	ns
T374	HDMARQ ↓ → HDD 出力ディレイ時間	20	—	—	ns
T375	HDMARQ ↓ → XHDMACK ↑ 最小インターロック時間	160	—	—	ns
T376	HIORDY → XHDMACK ↑ 最小インターロック時間	20	—	—	ns
T377	XHDMACK ↑ → XHCS0,1 XHCS0,1ホールド時間	20	—	—	ns
T378	HDD(CRC) → XHDMACK ↑ CRCデータセットアップ時間	6.7	—	—	ns
T379	XHDMACK ↑ → HDD(CRC) CRCデータホールド時間	6.2	—	—	ns
T37a	HDMARQ ↓ → XHIOR ↑ 制限付きインターロック時間	0	—	150	ns
T37b	HIORDY → XHDMACK ↑ 最小インターロック時間	110	—	—	ns

8.4.4.6. Ultra DMA Write Timing



記号	説明	min	typ	max	単位
T381	XHCS ↑、HDA → XHDMACK ↓ アドレスセットアップ時間	20	–	–	ns
T382	HDMARQ ↑ → XHDMACK ↓ XHDMACK応答時間	0	–	–	ns
T384	XHDMACK ↓ → XHIOW ↓ エンベロープ時間	20	–	70	ns
T385	XHIOW ↓ → HIORDY ↓ 制限付きインターロック時間	IDE規格 t_{L}	–	IDE規格 t_{L}	ns
T386	HIORDY ↓ → XHIOR ↓ 制限無しインターロック時間	20	–	–	ns
T387	HDD → XHIOR ↓ データセットアップ時間	–	(cyc+1) * 16.7	–	ns
T388	XHIOR ↓ → HDD データホールド時間	–	(cyc+1) * 16.7	–	ns
T389	XHIOR → XHIOR XHIORサイクル時間	–	(cyc+2) * 16.7	–	ns
T38a	XHIOR → XHIOR XHIORサイクル時間 × 2	–	T389 * 2	–	ns
T38b	HIORDY ↑ → XHIOR 最終のSTROBE時間	20	–	38	ns

*1:cyc=UltraDMAcycle

詳細はレジスタ説明、“IDE Ultra-DMA Transfer Mode”を参照のこと

8. 電気的特性

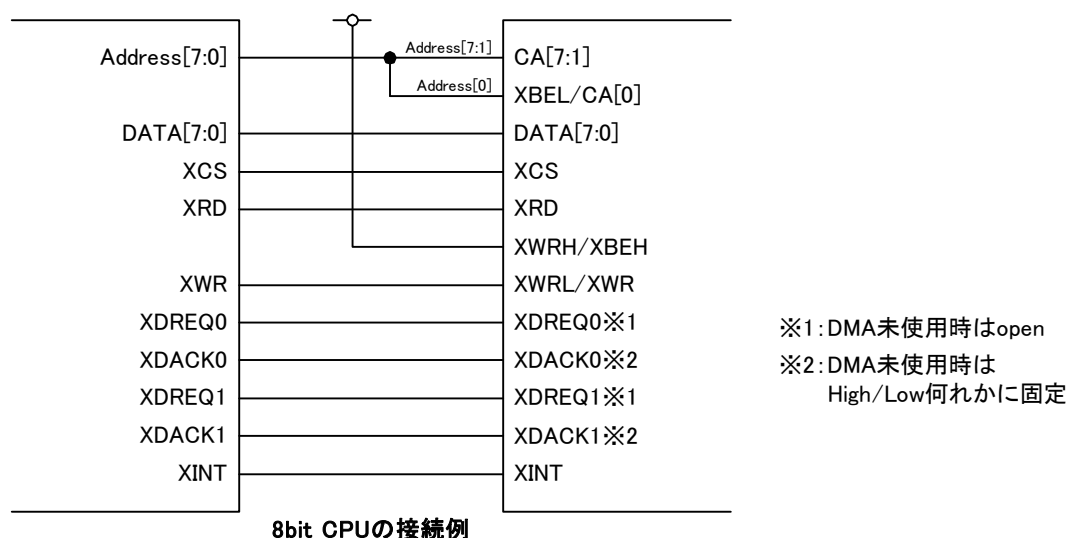
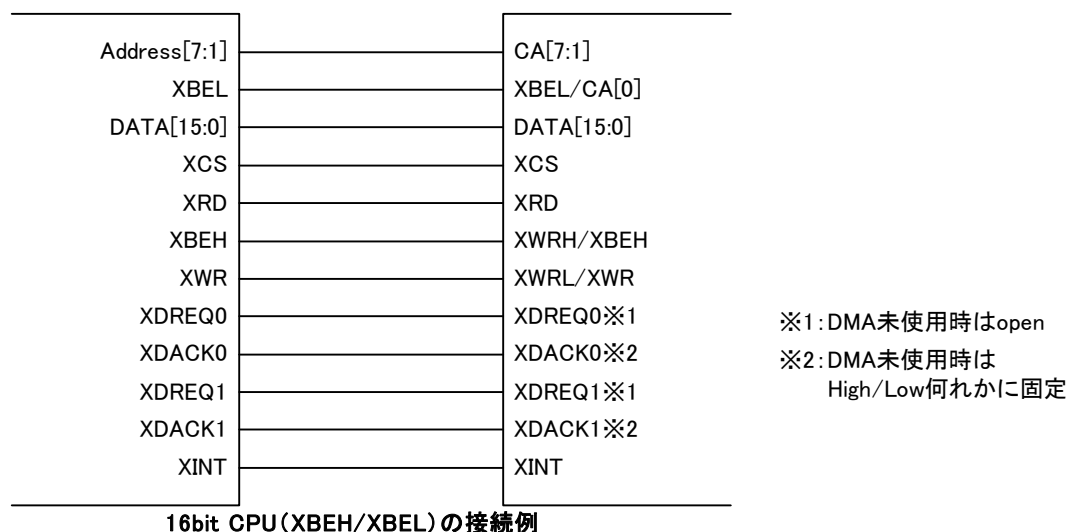
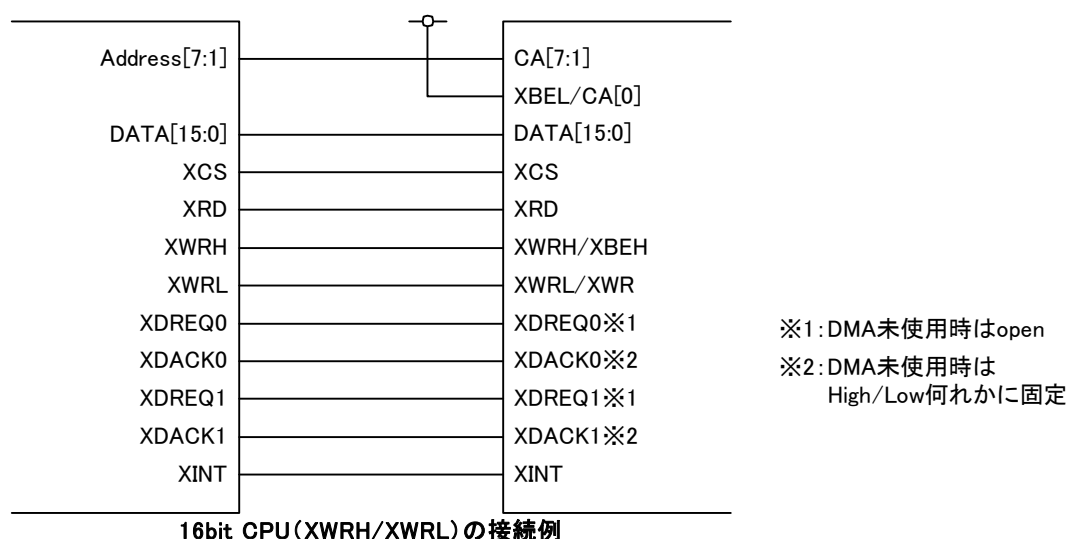
8.4.5. USB I/F タイミング

USB2.0 規格に準拠します。

< Universal Serial Bus Specification Revision 2.0 released on April 27, 2000 >

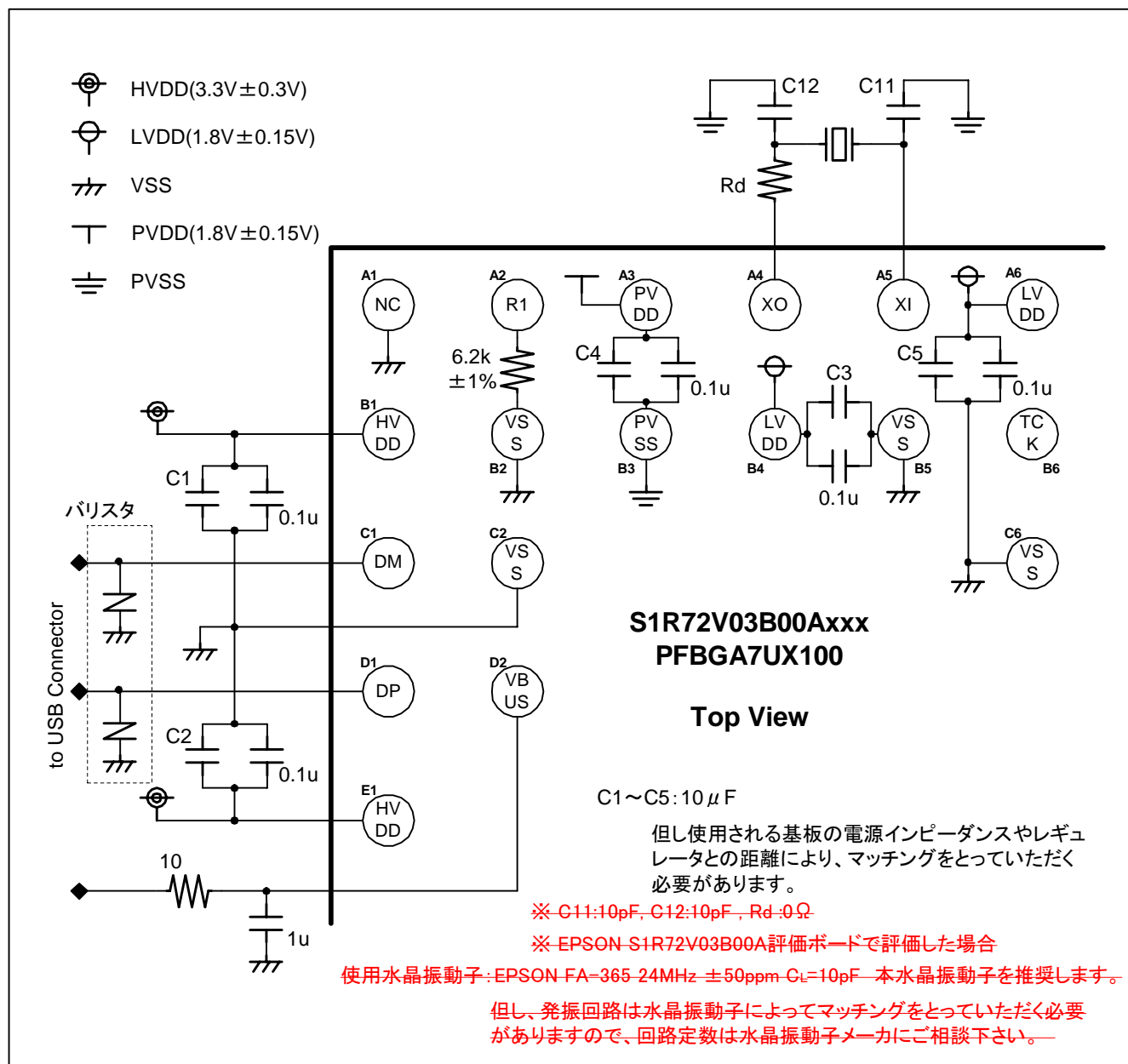
9. 接続例

9.1. CPU I/F 接続例



9.2.2. PFBGA7UX100 の場合

電源素子の性能は USB 信号波形品質に影響を与える為、その選定には注意してください。



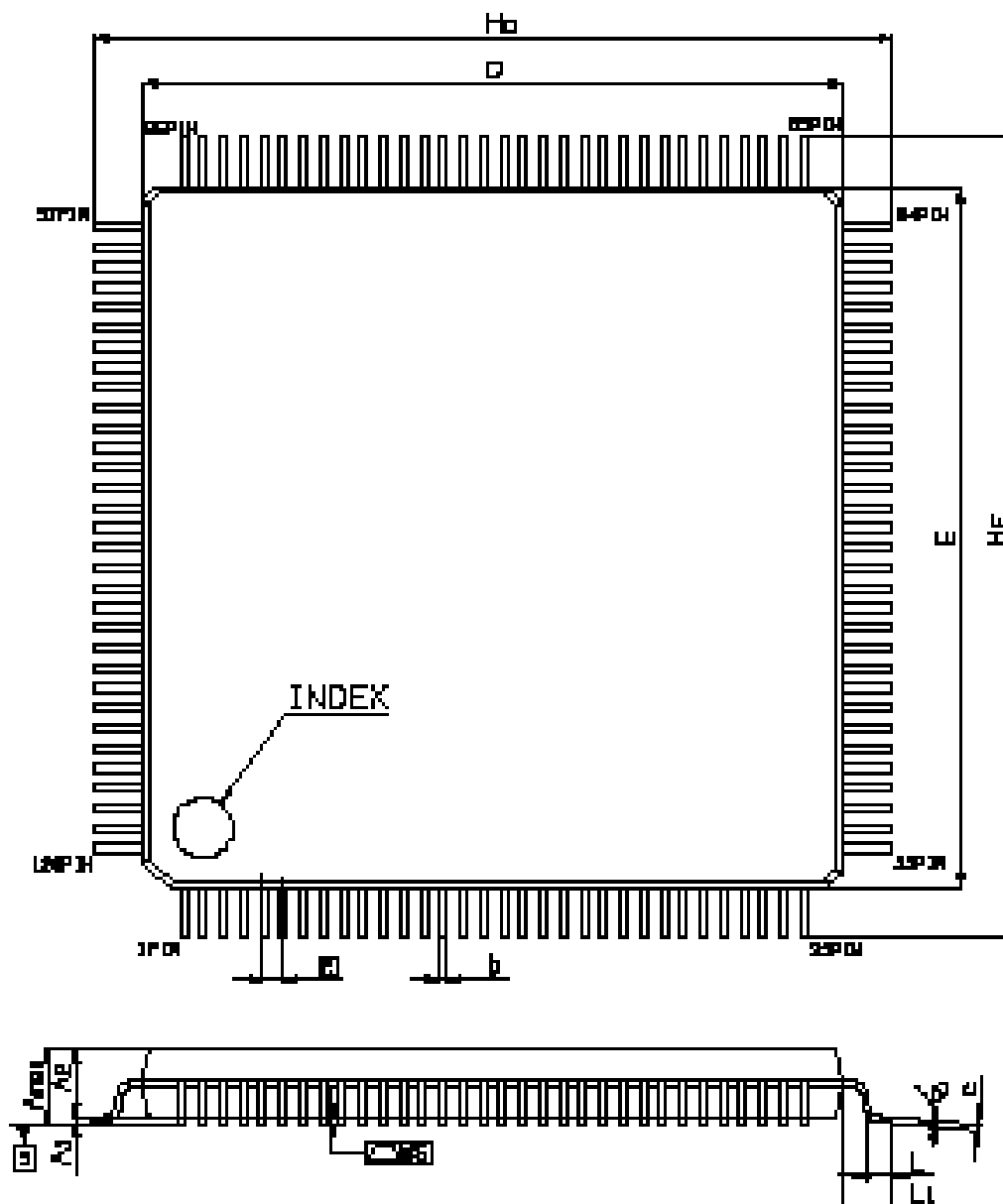
基板の設計に関しましては、別途、「PCB ガイドライン(PCB_revX.X_Japanese.pdf)」をご参照ください。

* 文書名の"X"はリビジョンナンバーです。

10. 外形寸法図

10. 外形寸法図

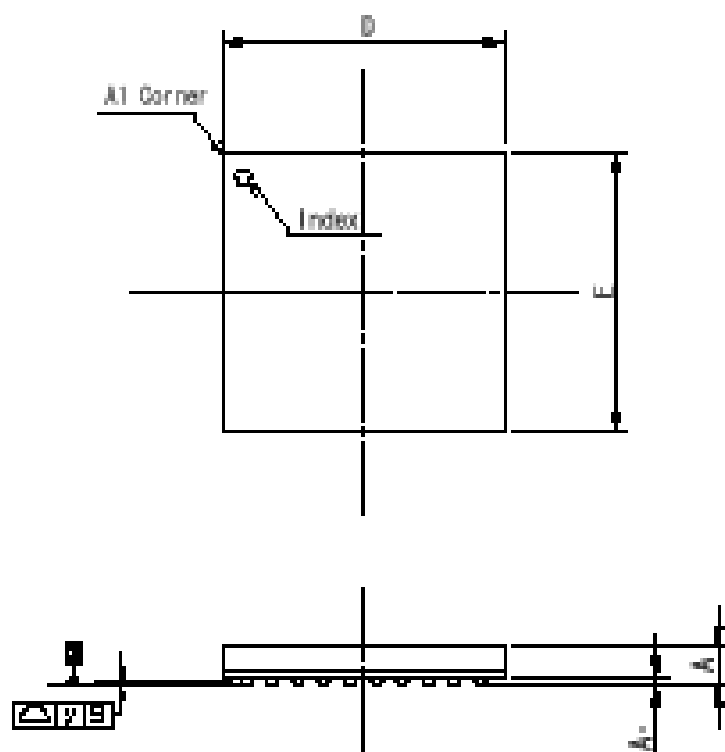
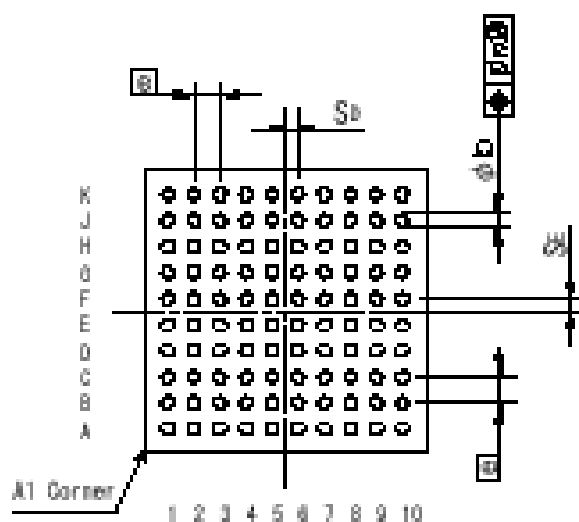
10.1.QFP パッケージ



Symbol	Dimension in Millimeters		
	Min	Max	Max
E	-	3	-
P	-	3	-
W	-	-	1.7
L	-	0.1	-
H	-	0.3	-
b	0.15	-	0.35
e	0.15	-	0.3
t	0.1	-	0.3
L	0.3	-	0.75
Hb	-	18	-
Hd	-	18	-
s	-	-	0.125

1 = 1mm

10.2. BGA パッケージ

Top ViewBottom View

Symbol	Dimension in Millimeters		
	Min	Max	
D	-	7	-
E	-	7	-
A	-	-	1.2
A ₁	-	0.22	-
a	-	0.65	-
b	0.27	-	0.37
x	-	-	0.08
y	-	-	0.1
S ₀	-	0.325	-
S _E	-	0.325	-

1 - 2mm

Appendix A IDE_Config_1.Swap ビット設定

S1R72V03 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。S1R72V03 では、IDE_Config_1.Swap ビットを使用して、S1R72V03 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	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_H HDD[7:0] ⇒ IDE_RdRegValue_L	なし	IDE_WrRegValue_H ⇒ HDD[15:8] IDE_WrRegValue_L ⇒ HDD[7:0]	IDE_SeqWrRegValue(1 st) ⇒ HDD[7:0] IDE_SeqWrRegValue(2 nd) ⇒ HDD[15:8]
1	HDD[15:8] ⇒ IDE_RdRegValue_L HDD[7:0] ⇒ IDE_RdRegValue_H	なし	IDE_WrRegValue_H ⇒ HDD[7:0] IDE_WrRegValue_L ⇒ HDD[15:8]	IDE_SeqWrRegValue(1 st) ⇒ HDD[15:8] IDE_SeqWrRegValue(2 nd) ⇒ 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_H HDD[7:0] ⇒ IDE_RdRegValue_L	同左	IDE_WrRegValue_L ⇒ HDD[7:0]	IDE_SeqWrRegValue ⇒ HDD[7:0]
1	同上	同上	IDE_WrRegValue_H ⇒ HDD[7:0]	同上

Appendix B リトルエンディアンの CPU への接続

S1R72V03 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 V03 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、即ち上位バイトには、V03 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、即ち下位バイトには V03 の CD7 から CD0 を接続してください。また、CPU によってはライト信号の仕様が異なりますが、これもハイ、ローをそのまま接続してください。

<F/W>

リトルエンディアンの CPU にて、V03 を動作させる時には、まず以下の手順を行ってください。

- ① ChipConfig.CPU_Swap ビットを"1"にセット

このレジスタのアドレス番地は V03 においては 0xEF に割り当てられていますが、②を行うまでは、リトルエンディアンの CPU においては、0xEE に割り当てられているように動作しています。これは、本 LSI の初期状態がビッグエンディアンであるために、ライト信号の上位、下位が逆転しているからです。

- ② 0xE9 番地をリード

このリード動作によって、本 LSI は CPU のバスの上位と下位を切り替えます。①を実行しただけでは切り替わっていないことにご注意願います。したがって、このリード動作の後には、全レジスタが 7.1 レジスタマップに示す通りのアドレス番地となります。

なお、この設定を行った後、EPnFIFO_Rd_H/L、EPnFIFO_Wr_H/L を除く、その他の全ての内部レジスタは、Char(8 ビット幅)にてアクセスしてください。Short(16 ビット)以上にて定義されるレジスタにつきましても、Char にてアクセスして頂き、CPU のメモリ上にてキャストしてご使用ください。

EPnFIFO_Rd_H/L、EPnFIFO_Wr_H/L につきましては、Short にてアクセス可能となっています。また DMAC を使用してのアクセスに対しても問題御座いません(以下の表を参照)。

- USB から 01_02_03_04_05_06 と順にデータを受け取った場合

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]
1st	01	02	02	01
2nd	03	04	04	03
3rd	05	06	06	05

Appendix C SetAddress リクエストの応答について

bmRequestType が 0(標準リクエスト)以外であり、bRequest が 0x05 であるリクエストを受信した場合に、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を使用し、USB_Address レジスタ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 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 SETUP_Control.ProtectEP0="0" EP0Control.INxOUT="1" EP0Control.IIN="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

① SetAddress 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_EP0SETUP0,1(※)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。

※: S1R72V03 のレジスタ定義は、"RcvEP0SETUP"となります。

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 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。

- SETUP_Control.ProtectEP0="0"

- EP0Control.INxOUT="1"

- 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。

尚、上記手順に含まれるレジスタのうち、下記"D_ModeContrl"を、開発仕様書の Rev.1.60 にて追加定義しました。また、USB_Address レジスタの bit7 を、開発仕様書の Rev.1.60 にて追加定義しました。

Address	Register Name	R / W	Bit Symbol	Description	Reset
F1h	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**

改訂履歴

年月日	改訂内容			
	Rev.	頁	種別	内 容
04/11/15	0.79	全頁	新規	新規制定
04/12/3	0.80	P4 P6 P61	修正	テスト端子標記を"TEST"に統一
		P2	修正	ブロック図 PLL60 ブロック追加
		P3	修正	3.2 項 Oscillator を PLL60 に変更
		P6	修正	5. 項 端子機能 PU/PD 修正
		P1	修正	CPU I/F 概要の Endian の記述を修正
		P18	修正	6.1.6.1 項 CBW_LengthErr に関する説明を修正
		P18	修正	6.1.6.1 項 CBW_Err に関する説明を修正
		P22	修正	6.1.8 項を 6.1.7.11 項に項番を修正 (6.1.8. ?項もこれに順ずる)
		P44	修正	6.3.1.3.1 項 誤記修正
		P66	削除	ビット削除 81h bit6-4, 91h bit6-4
		P66	追加	ビット追加 81h bit6, 91h bit6 レジスタ追加 8Ch-8Fh, 9Ch-9Fh
		P156, 165, 162, 163, 171, 172	追加	上記ビット追加、レジスタ追加に伴う、レジスタ詳細説明追加
		P51	追加	6.4.2.2.3 項 追加
		P5	追加	4.2 BGA パッケージ端子配置図追加
		P6, P7, P8	追加	Ball 名称追加
		P62	削除 修正	BulkIntStat.CBW_TranErr を削除。 BulkIntStat.CBW_Err を bit5 に移動。他の記述箇所も同様。
		P62	追加	FIFO_IntStat.FIFO_Full, FIFO_IntStat.FIFO_Empty 及び各イネーブル ビットを追加。詳細説明も同様。
		P64	追加	EPx {x=a, b, c} Join. JoinFIFO_Stat ビットを追加。詳細説明も同様。
		P63	追加	ClrAllJoin. ClrJoinFIFO_Status ビットを追加。詳細説明も同様。
		P207, P208, P209	修正	DC 特性修正
		P220, P221	修正	接続例修正
		P8 P209	修正	PU PD 追加修正
		P43	修正	デスク립タエリアの領域修正
		P66 P67 P69	追加 修正	レジスタマップ DMA0_Config.ActiveDMA ビット、DMA1_Config.ActiveDMA ビット追加 IDE_Count_L レジスタの bit0 削除 ClkSel.xActIDE_Term ビット追加 ChipConfig.Bus8x16 ビット名称修正

		P198	修正 追加	レジスタ説明 RAM_RdCount レジスタ、表修正。設定数の制限追記。
		P183	修正	レジスタ説明 IDE_Count_H レジスタを一番初めに読むように説明修正。
		P202	追加	xActIDE_Term ビット説明追加
		P204	修正	CPU_Swap ビットの説明修正
		P20	修正	6.1.7.8 オートネゴシエーションエラーステータスビット修正
		P73, P86	追加	ACTIVE60 時の割り込みステータスの処理説明
		P73, P86, P101, P102, P103, P104, P115	追加	ACTIVE60 時にライトアクセス不可の説明
		P200	追加	ModeProtect で保護されるレジスタ追加
		P47 P204	追加	CPU_BusSwap ビットに関する説明追加
		P156, P165	修正	ActiveDMA の説明、誤記修正
		P210	追加	CPUIF AC スペック規定
		P47, 48	追加	モード切り替えに際して追加
05/03/02	0.90	P18, P90	修正	RAM_Monitor を RAM_Rd に修正
		P17	修正	DTGo を IDE_Go に修正
		P8	修正	LVDD 欄に 118, 120 ピンを追加修正
		P106	修正	Bit0 の説明、誤記修正
		P127, P133, P139, P145	修正	Bit 名 誤記修正
		P174	修正	Bit 説明、誤記修正
		P64	修正	0x14h bit3, bit2 追加修正
		P90	修正	bit2 R/W 属性追加修正
		P9	修正	VSS J3 -> J8
		P158, P167	修正	DMAx_Config{x=0, 1}. ActiveDMA ビットの説明修正
		P127, P133, P139, P145	修正	EPxJoin{x=0, a-c} レジスタビット排他処理説明修正。 Bit5-0 のうち 1 ビットのみセット可能。
		P6	修正	XI A1 -> A5
05/4/1	0.91	P69.P178	修正	IDE_Config1.bit2 Swap ビットを Reserve ビットに修正
		P209	修正	入力リーク追記
		P61	修正	IDE_CRC_L/H レジスタに初期値を書くと言う記述を削除
		P183	修正	IDC_CRC_L/H を Read Only に修正。初期値の設定記述を削除
		P104	修正	USB_Status.bit6: FSxHS 説明に「ケーブルアタッチ時に、"FS(1)"にセットして下さい。」を追加
		P101	修正	PM_Control.bit6-4 PM_State 説明修正

改訂履歴

		P208	修正	電源投入順を追加修正
05/4/27	1.0	P212	挿入	RESET タイミング、クロックタイミング追記 P212 挿入のため以降ページ番号修正
		P6	修正	JTAG 機能未使用時の端子処理の説明を記載
		P209	修正	供給電流、静止電流 記入
		P213	修正	CPU アクセスタイミング 3. 3V修正 1. 8V追加
		P215 ~P221	修正	IDE 規格値修正
		P224,225	修正	電源素子の注意事項追記
05/5/30	1.1	P59	修正	IDE_Config_1.Swap ビットに関する記載を追加
		P59	修正	6.5.1.3 シーケンシャルライト時のデータ書き込み順を修正
		P69	追加	IDE_Config_1.Swap ビットを追加
		P178	追加	IDE_Config_1.Swap ビットを追加
		P193	追加	IDE_SeqWrRegValue レジスタの書き込み順を補足
		P228	追加	Appendix A を追加
		P229	追加	Appendix B を追加
05/6/2	1.2	P224,225	修正	C11,C12,Rd 値に参考値記入。また推奨水晶振動子記入
		P229	修正	Appendix B を修正(全面修正)
05/6/2	1.2	P8	修正	IDE I/F 端子の RESET 時の値を修正
		P229	修正	Appendix B を修正(全面修正)
05/10/3	1.3	P7	修正	XINT 端子の端子タイプを修正
		P208,211	削除	UVDD の記述を削除
		P208	追加	絶対最大定格に CVDD 系及び PVDD 系入力電圧を追加 推奨動作条件に CVDD 系及、PVDD 系及び VBUS の入力電圧を追加
		P6,P7,P8	追加	端子の属する電源系を明記
		P210	修正	IVDD を HVDD に名称修正
		P229	修正	B9h を E9h に誤記修正
		P223	修正	8bit CPU の接続例における DATA 幅の誤記修正
06/02/1	1.4	P47	修正	6. 4. 2. 1 項 ChipConfig レジスタ番地の誤記修正 : EDh 番地→EFh 番地
06/04/10	1.5	P151	削除	誤記につき、以下の説明を削除。 「EPx{x=0,a-c}Join.JoinCPU_Rd ビットがセットされたときに該当するエンドポイントにデータが無い場合、そのエンドポイントに USB 等からデータが書き込まれるまでは、このビットはセットされません(RdRemainValid な"0"という状態にはなりません)。」
		P224 P225	修正	DP/DM ラインに静電保護バリスタを負荷
07/07/31	1.6	P1	修正	電源系統を2系当から3系統に修正
		P3	追記	PLL のクロックソースを明記
		P6	修正	VBUS 端子に PD を記載
		P13	追加	OUT トランザクションの正常完了時の応答に NYET を追加
		P45, P46	修正	FIFO のアクセス制限、アクセス要因の重複に関する説明を修正
		P63	修正	DEVICE_CODE の間違いを修正 適用外端子の修正 説明の修正

	P65, P117	修正	0x38 USB_Address レジスタに SetAddress ビットを追加
	P71, P207, P230	修正	0xF1 に D_ModeControl レジスタを追加 同レジスタの bit 4 を SetAddressMode ビットと定義 Appendix C を追加し、D_ModeControl レジスタの説明を追加
	P158, P167	修正	ActiveDMA ビットの説明を修正
	P206	修正	DACK_Level ビットの説明を修正
	P206	修正	CS_Mode ビットの説明を修正
	P208	追加	電源オフ手順を追記
	P209	修正	電源電流を修正 静止電流を修正 説明を追記
	P210	追加	VBUS 端子のプルダウン抵抗値を記載
	P212	削除	不要なマーカ"※"を削除
	P222	追加	参照文章名を記載
	P224, P225	修正	特性値の記載を削除 推奨部品の記載を削除 「PCB ガイドライン参照」を追記

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 : 410538302
2006 年 06 月 作成
2006 年 10 月 改訂