

S1R72H74

テクニカルマニュアル

本資料のご使用につきましては、次の点にご留意願います。

本資料の内容については、予告無く変更することがあります。

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

適用範囲

本ドキュメントは、ATA ホストコントローラ LSI「S1R72H74」に適用されます。

目 次

1. 概要	1
2. 特長	2
3. ブロック図	3
3.1 PLL60	4
3.2 FIFO Endpoint / FIFO Controller / FIFO RAM	4
3.3 CPU I/F Controller	4
3.4 DMA Controller	4
3.5 IDE Master Controller	4
3.6 TestMUX	4
4. 端子配置図	5
4.1 端子配置図 (QFP パッケージ)	5
5. 端子機能説明	6
6. 機能説明	9
6.1 互換性	9
6.2 パワーマネージメント機能	9
6.2.1 SLEEP (スリープ)	10
6.2.2 SNOOZE (スヌーズ)	11
6.2.3 ACTIVE60 (アクティブ 60)	11
6.3 FIFO 管理	12
6.3.1 FIFO 管理	12
6.3.1.1 FIFO メモリマップ	12
6.3.2 FIFO へのアクセス方法	12
6.3.2.1 FIFO へのアクセス方法 (PIO アクセス)	12
6.3.2.2 FIFO へのアクセス方法 (DMA)	13
6.3.2.3 FIFO へのアクセス方法 (IDE)	13
6.3.2.4 FIFO へのアクセス制限	13
6.4 CPUIF	14
6.4.1 モード切り替え	14
6.4.2 モード切り替えに際して	14
6.4.2.1 16bit BE mode をご使用の場合	14
6.4.2.2 8bit mode をご使用の場合	16
6.4.3 ブロック構成	16
6.4.3.1 REG (S1R72H74 Registers)	16
6.4.3.1.1 同期レジスタアクセス (ライト)	16
6.4.3.1.2 同期レジスタアクセス (リード)	16
6.4.3.1.3 FIFO アクセス (ライト)	16
6.4.3.1.4 FIFO アクセス (リード)	17
6.4.3.1.5 FIFO アクセスの端数処理	17
6.4.3.1.6 非同期レジスタアクセス (ライト)	19
6.4.3.1.7 非同期レジスタアクセス (リード)	19
6.4.3.2 DMA0/DMA1 (DMA ch.0 / ch.1)	19
6.4.3.2.1 基本機能	19
6.4.3.2.2 端子設定	21
6.4.3.2.3 アクセスモード設定	22
6.4.3.2.4 カウントモード (ライト)	22
6.4.3.2.5 カウントモード (リード)	23
6.4.3.2.6 フリーランモード (ライト)	24

6.4.3.2.7 フリーランモード（リード）	25
6.4.3.2.8 REQ アサートカウントオプション（ライト）	25
6.4.3.2.9 REQ アサートカウントオプション（リード）	26
6.4.3.2.10 DMA の FIFO アクセス端数処理	26
6.5 IDE I/F	26
6.5.1 IDE タスクファイルレジスタへのアクセス	27
6.5.1.1 IDE タスクファイルレジスタからのリード	27
6.5.1.2 IDE タスクファイルレジスタへのライト	27
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	27
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	27
6.5.2 PIO アクセス	28
6.5.2.1 PIO リード DMA	28
6.5.2.2 PIO ライト DMA	28
6.5.3 Multi-Word DMA	28
6.5.3.1 Multi-Word DMA リード	28
6.5.3.2 Multi-Word DMA ライト	29
6.5.4 Ultra DMA	30
6.5.4.1 Ultra DMA リード	30
6.5.4.2 Ultra DMA ライト	30
6.5.5 IDE 転送モードの設定について	31
6.6 バウンダリスキャン（JTAG）	32
6.6.1 対応インストラクション	32
6.6.2 DEVICE_CODE に関して	32
6.6.3 バウンダリスキャン除外端子	32
7. レジスタ	33
7.1 レジスタマップ	33
7.2 レジスタ詳細説明	41
7.2.1 00h MainIntStat (Main Interrupt Status)	41
7.2.2 02h SIE_IntStat (SIE Interrupt Status)	42
7.2.3 03h CPU_IntStat (CPU Interrupt Status)	43
7.2.4 04h FIFO_IntStat (FIFO Interrupt Status)	44
7.2.5 06h IDE_IntStat (IDE Interrupt Status)	45
7.2.6 10h MainIntEnb (Main Interrupt Enable)	45
7.2.7 12h SIE_IntEnb (SIE Interrupt Enable)	45
7.2.8 13h CPU_IntEnb (CPU Interrupt Enable)	45
7.2.9 14h FIFO_IntEnb (FIFO Interrupt Enable)	45
7.2.10 16h IDE_IntEnb (IDE Interrupt Enable)	45
7.2.11 20h RevisionNum (Revision Number)	45
7.2.12 21h ChipReset (Chip Reset)	45
7.2.13 22h PM_Control (Power Management Control)	45
7.2.14 29h EPrFIFO_Clr (Endpoint FIFO Clear)	45
7.2.15 2Ah ClrAllJoin (Clear All Join)	45
7.2.16 2Eh WakeupTim_H (Wakeup Time High)	45
7.2.17 2Fh WakeupTim_L (Wakeup Time Low)	45
7.2.18 5Ah EPbConfig_0 (EPb Configuration 0)	45
7.2.19 5Dh EPbJoin (End Point b Join)	45
7.2.20 62h EPcConfig_0 (EPc Configuration 0)	45
7.2.21 65h EPcJoin (End Point c Join)	45
7.2.22 70h EPnFIFO_Rd_H (EPn FIFO Read High)	45
7.2.23 71h EPnFIFO_Rd_L (EPn FIFO Read Low)	45
7.2.24 72h EPnFIFO_Wr_H (EPn FIFO Write High)	45
7.2.25 73h EPnFIFO_Wr_L (EPn FIFO Write Low)	45
7.2.26 74h EPnRdRemain_H (EPn FIFO Read Remain High)	45
7.2.27 75h EPnRdRemain_L (EPn FIFO Read Remain Low)	45
7.2.28 76h EPnWrRemain_H (EPn FIFO Write Remain High)	45
7.2.29 77h EPnWrRemain_L (EPn FIFO Write Remain Low)	45

7.2.30	7Ch EPnFIFO_ByteRd(EPn FIFO Byte Read)	45
7.2.31	80h DMA0_FIFO_Control (DMA0 FIFO Control)	45
7.2.32	81h DMA0_Config (DMA0 Config)	45
7.2.33	82h DMA0_Control (DMA0 Control)	45
7.2.34	84h DMA0_Remain_H (DMA0 FIFO Remain High)	45
7.2.35	85h DMA0_Remain_L (DMA0 FIFO Remain Low)	45
7.2.36	88h DMA0_Count_HH (DMA0 Transfer Byte Counter High/High)	45
7.2.37	89h DMA0_Count_HL (DMA0 Transfer Byte Counter High/Low)	45
7.2.38	8Ah DMA0_Count_LH (DMA0 Transfer Byte Counter Low/High)	45
7.2.39	8Bh DMA0_Count_LL (DMA0 Transfer Byte Counter Low/Low)	45
7.2.40	8Ch DMA0_RdData_H (DMA0 Read Data High)	45
7.2.41	8Dh DMA0_RdData_L (DMA0 Read Data Low)	45
7.2.42	8Eh DMA0_WrData_H (DMA0 Write Data High)	45
7.2.43	8Fh DMA0_WrData_L (DMA0 Write Data Low)	45
7.2.44	90h DMA1_FIFO_Control (DMA1 FIFO Control)	45
7.2.45	91h DMA1_Config (DMA0 Config)	45
7.2.46	92h DMA1_Control (DMA1 Control)	45
7.2.47	94h DMA1_Remain_H (DMA1 FIFO Remain High)	45
7.2.48	95h DMA1_Remain_L (DMA1 FIFO Remain Low)	45
7.2.49	98h DMA1_Count_HH (DMA1 Transfer Byte Counter High/High)	45
7.2.50	99h DMA1_Count_HL (DMA1 Transfer Byte Counter High/Low)	45
7.2.51	9Ah DMA1_Count_LH (DMA1 Transfer Byte Counter Low/High)	45
7.2.52	9Bh DMA1_Count_LL (DMA1 Transfer Byte Counter Low/Low)	45
7.2.53	9Ch DMA1_RdData_H (DMA1 Read Data High)	45
7.2.54	9Dh DMA1_RdData_L (DMA1 Read Data Low)	45
7.2.55	9Eh DMA1_WrData_H (DMA1 Write Data High)	45
7.2.56	9Fh DMA1_WrData_L (DMA1 Write Data Low)	45
7.2.57	A0h IDE_Status (IDE Status)	45
7.2.58	A1h IDE_Control (IDE Control)	45
7.2.59	A2h IDE_Config_0 (IDE Configuration 0)	45
7.2.60	A3h IDE_Config_1 (IDE Configuration 1)	45
7.2.61	A4h IDE_Rmod (IDE Register Mode)	45
7.2.62	A5h IDE_Tmod (IDE Transfer Mode)	45
7.2.63	A6h IDE_Umod (IDE Ultra-DMA Transfer Mode)	45
7.2.64	AAh IDE_CRC_H (IDE CRC High)	45
7.2.65	ABh IDE_CRC_L (IDE CRC Low)	45
7.2.66	ADh IDE_Count_H (IDE Transfer Byte Counter High)	45
7.2.67	A Eh IDE_Count_M (IDE Transfer Byte Counter Middle)	45
7.2.68	AFh IDE_Count_L (IDE Transfer Byte Counter Low)	45
7.2.69	B0h IDE_RegAdrs (IDE Register Address)	45
7.2.70	B2h IDE_RdRegValue_H (IDE Register Read Value High)	45
7.2.71	B3h IDE_RdRegValue_L (IDE Register Read Value Low)	45
7.2.72	B4h IDE_WrRegValue_H (IDE Register Write Value High)	45
7.2.73	B5h IDE_WrRegValue_L (IDE Register Write Value Low)	45
7.2.74	B6h IDE_SeqWrRegControl (IDE Sequential Register Write Control)	45
7.2.75	B7h IDE_SeqWrRegCnt (IDE Sequential Register Write Counter)	45
7.2.76	B8h IDE_SeqWrRegAdrs (IDE Sequential Register Write Address FIFO)	45
7.2.77	B9h IDE_SeqWrRegValue (IDE Sequential Register Write Value FIFO)	45
7.2.78	BCh IDE_RegConfig (IDE Register Configuration)	45
7.2.79	EBh ModeProtect (Mode Protection)	45
7.2.80	EDh ClkSelect (Clock Select)	45
7.2.81	EFh ChipConfig (Chip Configuration)	45
8.	電気的特性	45
8.1	絶対最大定格	45
8.2	推奨動作条件	45
8.3	DC 特性	45

8.4 AC 特性	45
8.4.1 RESET タイミング	45
8.4.2 クロックタイミング	45
8.4.3 CPU/DMA I/F アクセスタイミング	45
8.4.4 IDE I/F タイミング	45
8.4.4.1 PIO Read Timing	45
8.4.4.2 PIO Write Timing.....	45
8.4.4.3 DMA Read Timing.....	45
8.4.4.4 DMA Write Timing	45
8.4.4.5 Ultra DMA Read Timing	45
8.4.4.6 Ultra DMA Write Timing.....	45
9. 接続例	45
9.1 CPU I/F 接続例.....	45
10. 外形寸法図	45
10.1 QFP パッケージ	45
11. Appendix A IDE_Config_1.Swap ビット設定	45
12. Appendix B リトルエンディアンの CPU への接続	45
改訂履歴表	45

1. 概要

S1R72H74**は ATA/ATAPI6 に対応する ATA ホストコントローラ LSI です。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

2. 特長

《 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

《 互換性 》

- USB デバイスコントローラ S1R72V03 と互換
 - USB を除く機能互換
 - USB を除くレジスタ互換
 - 端子互換

《 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 1 MΩ 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
- 動作保証温度範囲 -40 ~ 85

耐放射設計はされていません

3. ブロック図

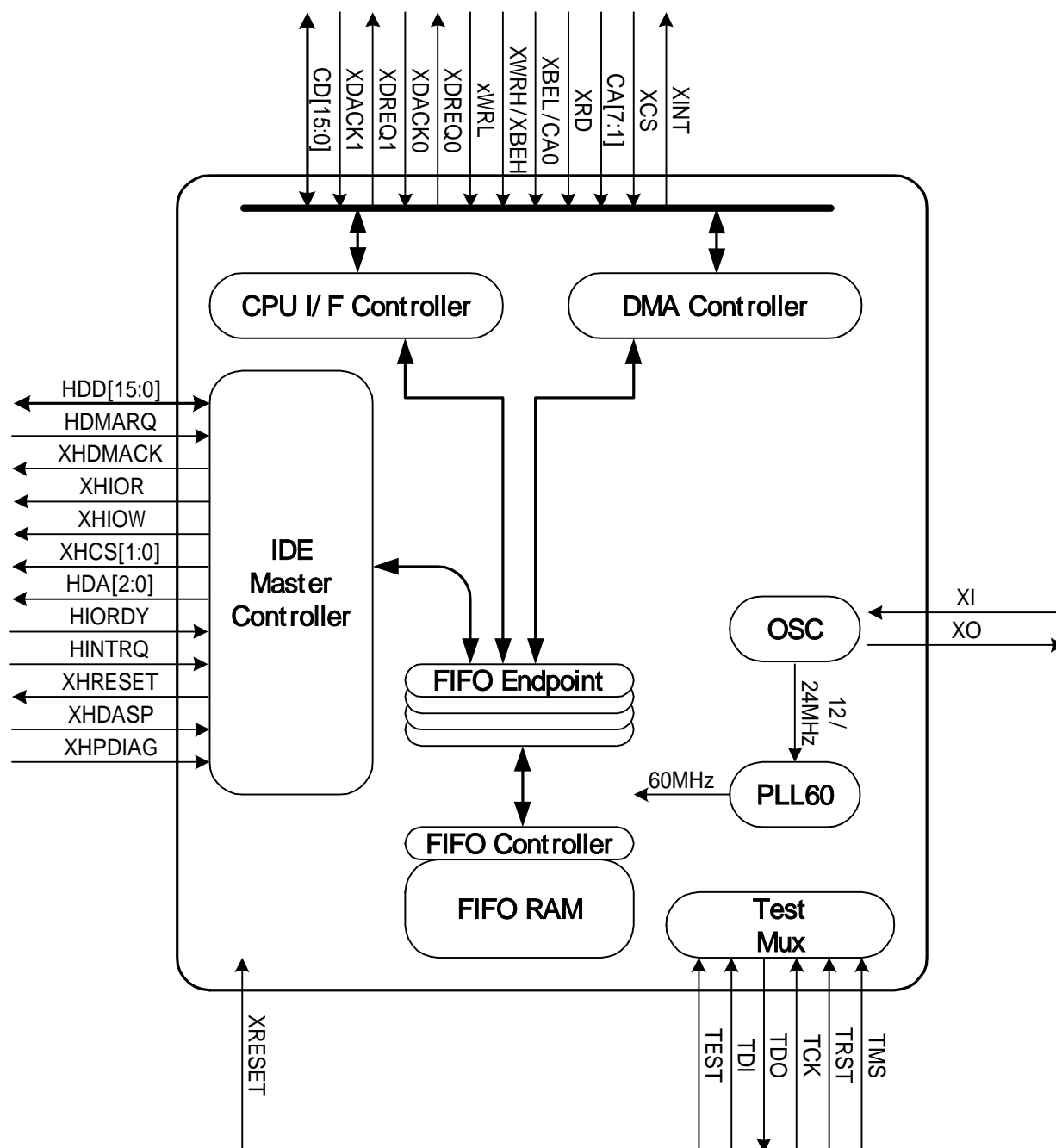


図 3.1 全体ブロック図

3. ブロック図

3.1 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。

3.2 FIFO Endpoint / FIFO Controller / FIFO RAM

データ転送用バッファ（計 2kB）です。

3.3 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。

3.4 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。

3.5 IDE Master Controller

ATA/ATAPI6 に対応する IDE インタフェースです。
PIO モード 0～4, Multi word DMA, UDMA モード 0～5 に対応します。

3.6 TestMUX

テスト回路です。

4. 端子配置図

4.1 端子配置図 (QFP パッケージ)

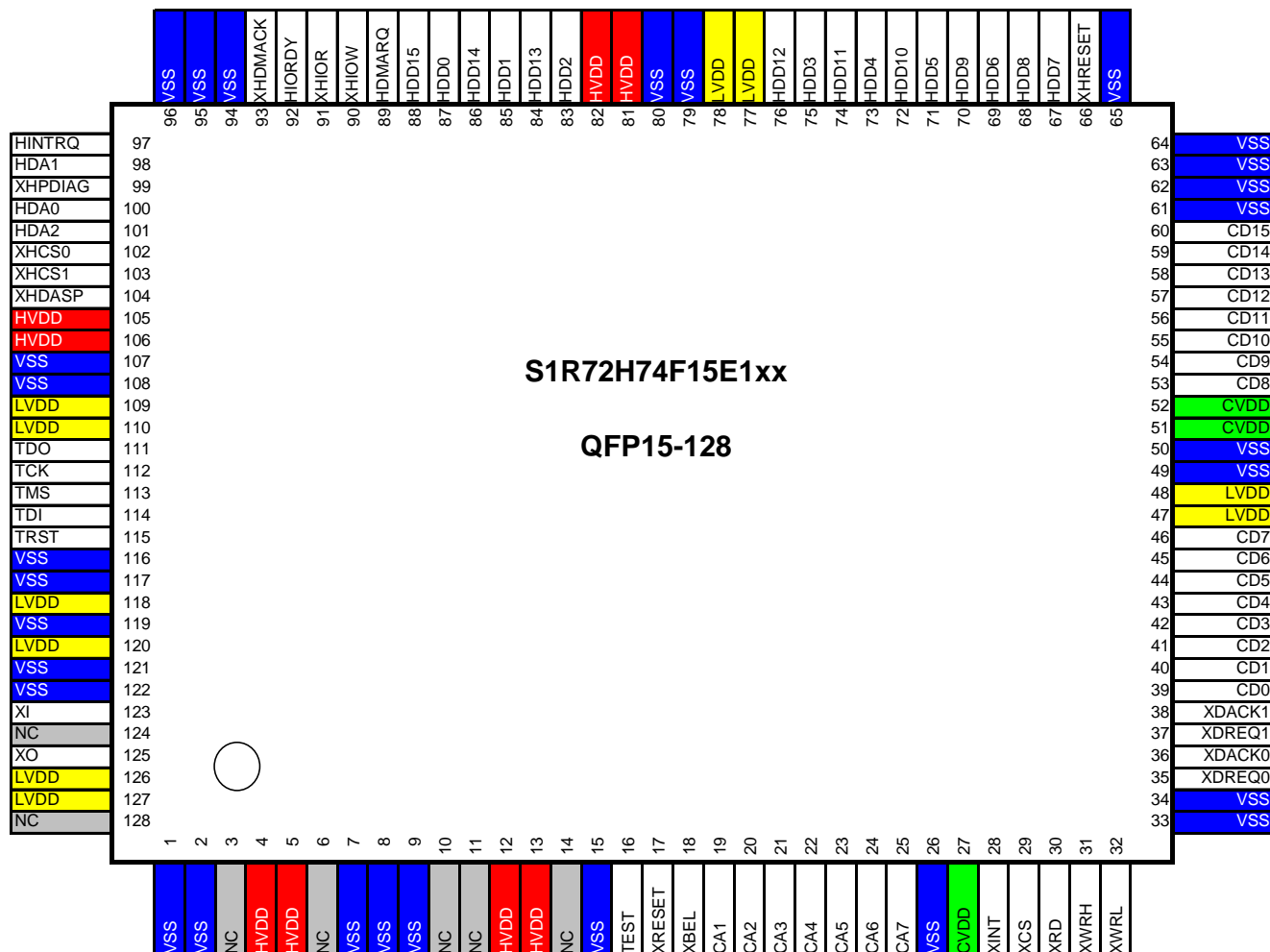


図 4.1 QFP パッケージ端子配置図

5. 端子機能説明

5. 端子機能説明

Pin	名称	I/O	RESET	端子タイプ	端子説明
17	XRESET	IN	-	-	リセット信号
123	XI	IN	-	Analog	内部発振回路用入力 12MHz/24MHz
125	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 LVDD 電源系に属します。

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

5. 端子機能説明

Pin	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	XRD	IN	-	-	リード・ストロブ		
32	XWRL (XWR)	IN	-	-	ライト・ストロブ 下位	ライト・ストロブ	
31	XWRH (XBEH)	IN	-	-	ライト・ストロブ 上位	ハイ・バイト イネーブル	High 固定
29	XCS	IN	-	-	チップセレクト信号		
28	XINT	OUT	High	2mA TriState	割り込み出力信号		
35	XDREQ0	OUT	High	2mA	DMA0 リクエスト		
36	XDACK0	IN	-	-	DMA0 アクノリッジ		
37	XDREQ1	OUT	High	2mA	DMA1 リクエスト		
38	XDACK1	IN	-	-	DMA1 アクノリッジ		
18	XBEL (CA0)	IN	-	-	High or Low 固定	ロー・バイト イネーブル	アドレス 0
19	CA1	IN	-	-	CPU バスアドレス		
20	CA2	IN	-	-			
21	CA3	IN	-	-			
22	CA4	IN	-	-			
23	CA5	IN	-	-			
24	CA6	IN	-	-			
25	CA7	IN	-	-			
39	CD0	BI	Hi-Z	2mA	CPU データ・バス		
40	CD1	BI	Hi-Z	2mA			
41	CD2	BI	Hi-Z	2mA			
42	CD3	BI	Hi-Z	2mA			
43	CD4	BI	Hi-Z	2mA			
44	CD5	BI	Hi-Z	2mA			
45	CD6	BI	Hi-Z	2mA			
46	CD7	BI	Hi-Z	2mA			
53	CD8	BI	Hi-Z	2mA			
54	CD9	BI	Hi-Z	2mA			
55	CD10	BI	Hi-Z	2mA			
56	CD11	BI	Hi-Z	2mA			
57	CD12	BI	Hi-Z	2mA			
58	CD13	BI	Hi-Z	2mA			
59	CD14	BI	Hi-Z	2mA			
60	CD15	BI	Hi-Z	2mA			

上記の端子は CVDD 電源系に属します。

5. 端子機能説明

Pin	名称	I/O	RESET	端子タイプ	端子説明
101	HDA2	OUT	Hi-Z	4mA	IDE レジスタ・アドレス
98	HDA1	OUT	Hi-Z	4mA	
100	HDA0	OUT	Hi-Z	4mA	
103	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	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	XHIOR	OUT	Hi-Z	4mA	IDE リード・ストローク
90	XHIOW	OUT	Hi-Z	4mA	IDE ライト・ストローク
89	HDMAHQ	IN (PD)	(PD)	-	DMA 転送要求
93	XHDMACK	OUT	Hi-Z	4mA	DMA 転送許可
92	HIORDY	IN (PU)	(PU)	-	IDE レジスタレディ信号
97	HINTRQ	IN (PU)	(PU)	-	IDE 割り込み要求
66	XHRESET	OUT	Hi-Z	4mA	IDE バスリセット
104	XHDASP	IN (PU)	(PU)	-	ドライブ有効 / スレーブドライブ有り
99	XHPDIAG	IN (PU)	(PU)	-	診断シーケンス終了信号
88	HDD15	BI (PU)	(PU)	4mA	IDE データ・バス
86	HDD14	BI (PU)	(PU)	4mA	
84	HDD13	BI (PU)	(PU)	4mA	
76	HDD12	BI (PU)	(PU)	4mA	
74	HDD11	BI (PU)	(PU)	4mA	
72	HDD10	BI (PU)	(PU)	4mA	
70	HDD9	BI (PU)	(PU)	4mA	
68	HDD8	BI (PU)	(PU)	4mA	
67	HDD7	BI (PD)	(PD)	4mA	
69	HDD6	BI (PU)	(PU)	4mA	
71	HDD5	BI (PU)	(PU)	4mA	
73	HDD4	BI (PU)	(PU)	4mA	
75	HDD3	BI (PU)	(PU)	4mA	
83	HDD2	BI (PU)	(PU)	4mA	
85	HDD1	BI (PU)	(PU)	4mA	
87	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	名称	電圧	端子説明
4, 5, 12, 13, 81, 82, 105, 106	HVDD	3.3V	IDE, TEST I/O 用電源
27, 51, 52	CVDD	3.3V/1.8V	CPU I/F I/O 用電源
47, 48, 77, 78, 109, 110, 118, 120, 126, 127	LVDD	1.8V	内部電源
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, 121, 122	VSS	0V	GND
3, 6, 10, 11, 14, 124, 128	NC	0V	NC (GND に固定してください。)

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.FIFO_IntStat ビット」「MainIntStat レジスタの FIFO_IntStat ビット」
- ・ FIFO エンドポイント毎にあるレジスタ
EPx{x=b-c} ~ レジスタ、EPx{x=b-c} ~ レジスタ等と標記します。
例：「EPx{x=b-c}Config 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=b-c}Join.JoinDMAx{x=0,1}_Wr ビット」

6.1 互換性

本 LSI は、USB デバイスコントローラ LSI 「S1R72V03」との互換性を持ちます。S1R72V03 は、USB デバイス機能と ATA ホスト機能を持ちます。本 LSI は、物理的 I/F 及び論理的 I/F において、S1R72V03 から USB 機能を除いた構成となっています。

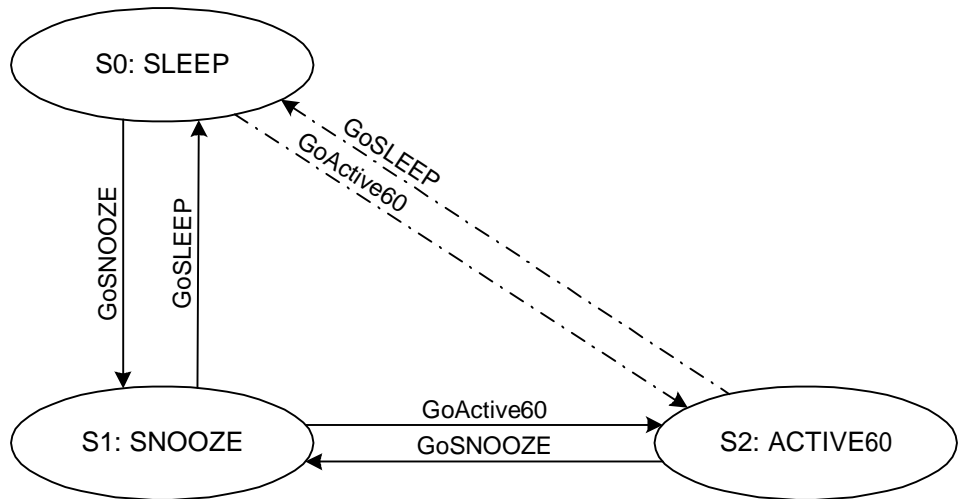
また、ソフトウェアのソースコードの互換性を保持するため、S1R72V03 の USB に関連するレジスタ名称を継承しています。

6.2 パワーマネージメント機能

オシレータ及び PLL の動作を制御し、スリープ、スヌーズ、アクティブ 60 の 3 ステートを遷移します。他のステータに遷移する場合には、PM_Control.GoSLEEP、PM_Control.GoSNOOZE、PM_Control.GoActive60 ビットをセットすることで開始され、任意の処理を行います。現在、どのステータ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、PM_IntStat(SIE_IntStat).FinishedPM イベントが発生します。この時、PM_IntEnb(SIE_IntEnb).EnFinishedPM ビットをセットし、かつ MainIntEnb.EnPM_IntStat(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。

全てのステータから他のステータへ遷移することが可能で、アクティブ 60 ステータ時に PM_Control.GoSLEEP ビットをセットした場合は、スヌーズステータを経由してスリープステータに遷移し、遷移終了した時点で PM_IntStat(SIE_IntStat).FinishedPM イベントが発生します。またスリープステータ時に PM_Control.GoActive60 ビットをセットした場合も、スヌーズステータを経由して、アクティブ 60 ステータに遷移し、遷移終了した時点で PM_IntStat(SIE_IntStat).FinishedPM イベントが発生します。

6. 機能説明



一点鎖線の遷移は、実際には実線を通して遷移する。

図 6.1 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。

スヌーズステート及びアクティブ 60 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、まず PLL60 を停止し、次に、OSCCLK の出力を停めた後にオシレータの発振を停止します。

逆に、スリープステート中に PM_Control.GoSNOOZE または PM_Control.GoActive60 ビットをセットすると、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。

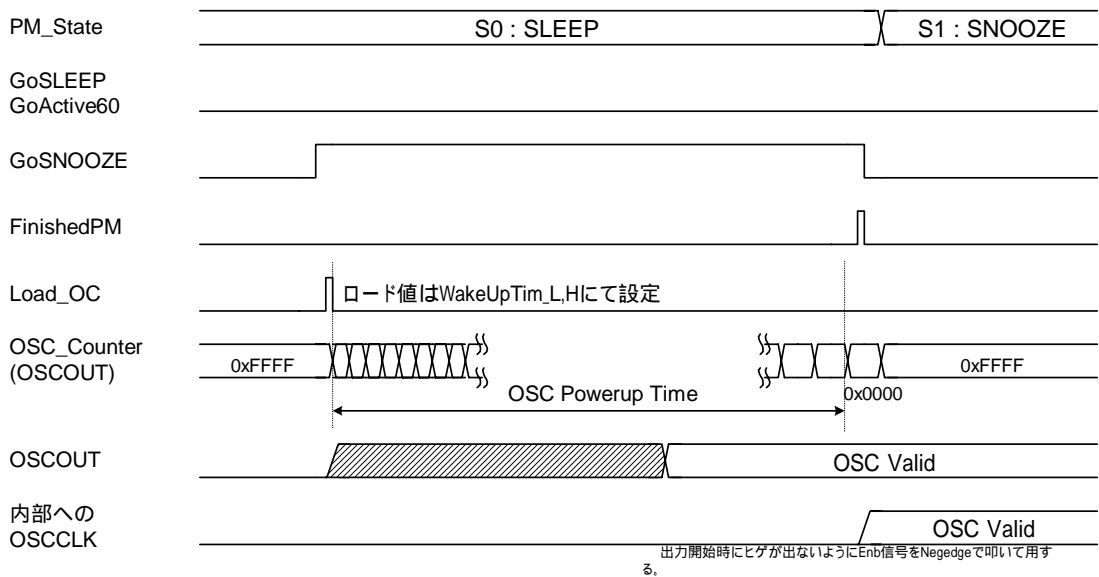


図 6.2 SLEEP ステートからの離脱 (GoSNOOZE 時)

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、PLL60 を停止します。

また、スヌーズステート中に 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。

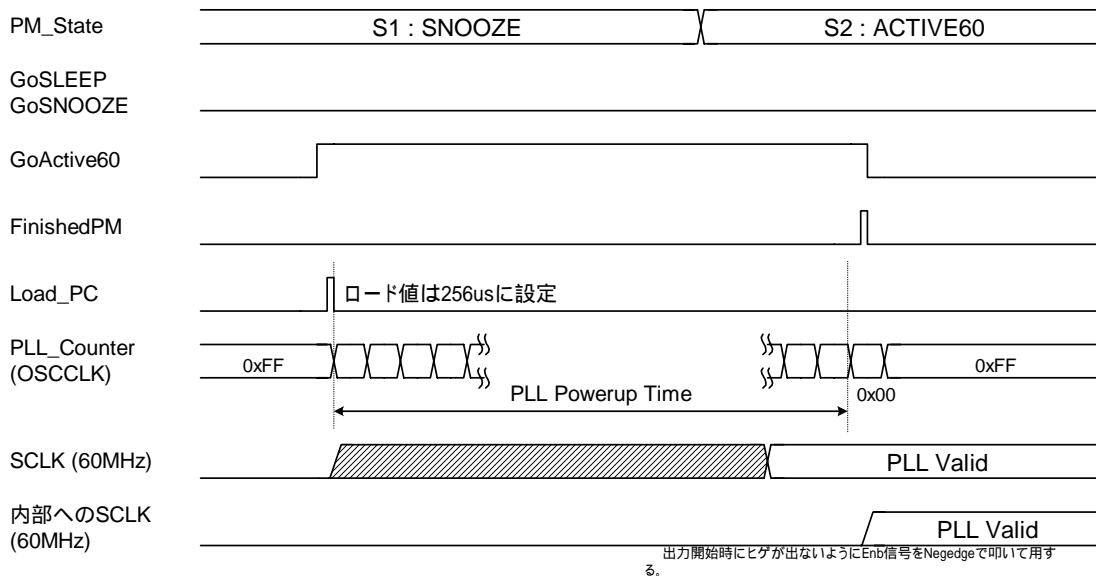


図 6.3 SNOOZE ステートからの離脱 (GoActive60 時)

6.2.3 ACTIVE60 (アクティブ 60)

オシレータ及び PLL60 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 ステートで無ければ読み書きできません。

IDE 並びに CPU 回路は、アクティブ 60 ステートで動作します。

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。

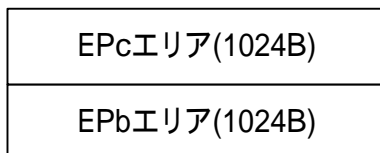


図 6.4 FIFO メモリマップ

FIFO のメモリは、EPb エリア、EPc 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。

EPb エリア及び EPc エリアは、IN(IDE Read)または OUT(IDE Write)の方向を任意に設定できる汎用 FIFO エンドポイントのエリアです。EPb エリア、EPc エリアは 1024 バイトが確保されており、IDE のリードまたはライト転送に使用することが出来ます。

IN(IDE Read)方向に設定された FIFO エンドポイントでは、IDE リードのデータ転送が可能です。また、OUT(IDE Write)方向に設定された FIFO エンドポイントでは、IDE ライトのデータ転送が可能です。CPU I/F からの PIO 及び DMA による FIFO アクセスは、IN または OUT の設定に依りません。(6.3.2.1, 6.3.2.2 を参照)

EPb エリア及び 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、EPnFIFO_Clr.EPx{x=b-c}FIFO_Clr の各ビットをセットしてください。

6.3.2 FIFO へのアクセス方法

FIFO へのアクセス要因には、CPU(PIO), CPU(DMA), IDE が有ります。

6.3.2.1 FIFO へのアクセス方法 (PIO 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 FIFO エンドポイントに対して、EPx{x=b-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 FIFO エンドポイントに対して、EPx{x=b-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=b-c}Join.JoinCPU_Rd にてただ 1 つ設定された FIFO 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=b-c}Join.JoinCPU_Wr にてただ 1 つ設定された FIFO 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=b-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。

6.3.2.2 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 FIFO エンドポイントの、EPn{n=b-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 FIFO エンドポイントの、EPn{n=b-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=b-c}Join.JoinDMAx{x=0,1}_Rd にて、ただ 1 つ設定された FIFO 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=b-c}Join.JoinDMAx{x=0,1}_Wr にて、ただ 1 つ設定された FIFO 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。

6.3.2.3 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 FIFO エンドポイントに対して、EPx{x=b-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 FIFO エンドポイントの方向に応じて設定されます。FIFO エンドポイントの方向設定が IN(EPn{n=b-c}Config_0.INxOUT=1)である時、その FIFO エンドポイントでは、IDE リードアクセスが行われます。また、FIFO エンドポイントの方向設定が OUT(EPn{n=b-c}Config_0.INxOUT=0)である時、その FIFO エンドポイントでは、IDE ライトアクセスが行われます。

6.3.2.4 FIFO へのアクセス制限

本 LSI の FIFO には、CPU バスからの PIO 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 FIFO 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。

- ・ 1 つの FIFO 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 FIFO 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。

6. 機能説明

6.4 CPUIF

6.4.1 モード切り替え

本 LSI 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 6.1 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ください。その後、ModeProtect レジスタにより設定を保護してください。

また、本 LSI 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 ビットを設定してください。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

本 LSI 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ください。

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、0 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。また、その際には図 6.5 に示す通り、必ず EFh 番地に対するバイトライトを行ってください。この時、本 LSI 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。本 LSI 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) を内蔵しておりますが、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ください。

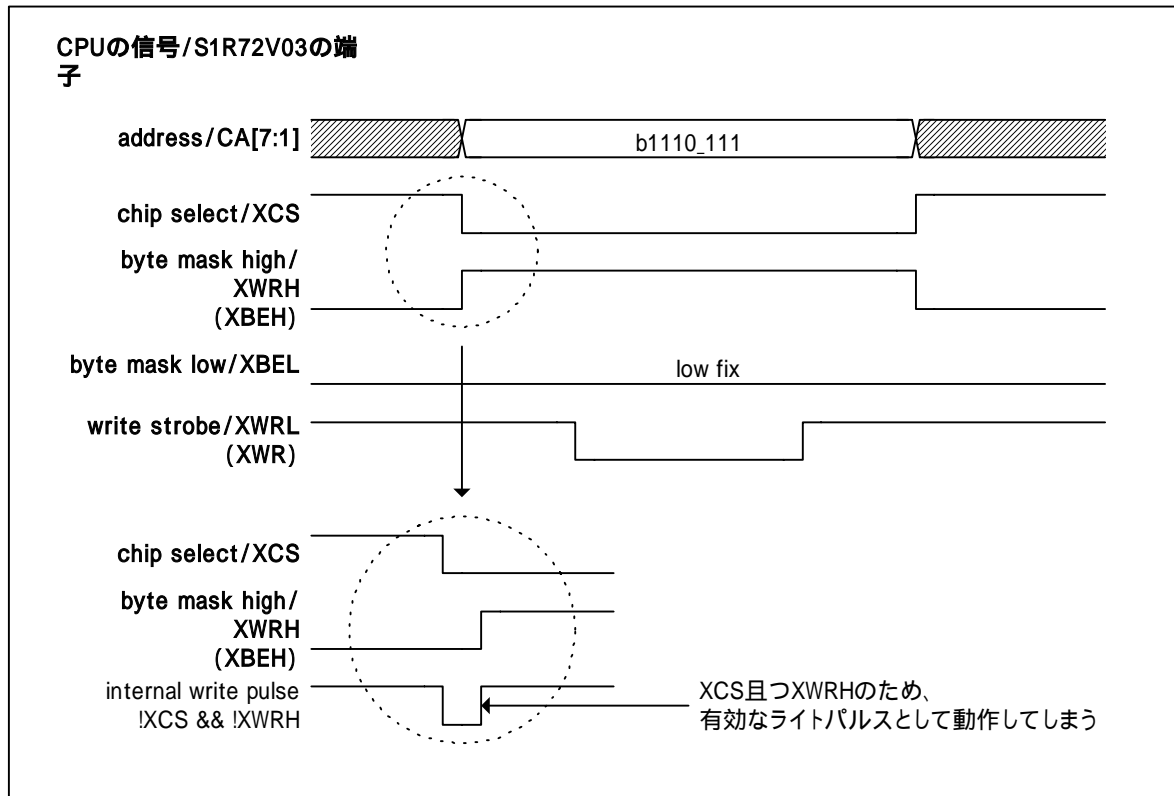


図 6.5 ChipConfig レジスタの初期設定

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、この様な制限はございません。

また、ChipConfig レジスタの設定前に、本 LSI にリードアクセスを行った場合、図 6.6 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。

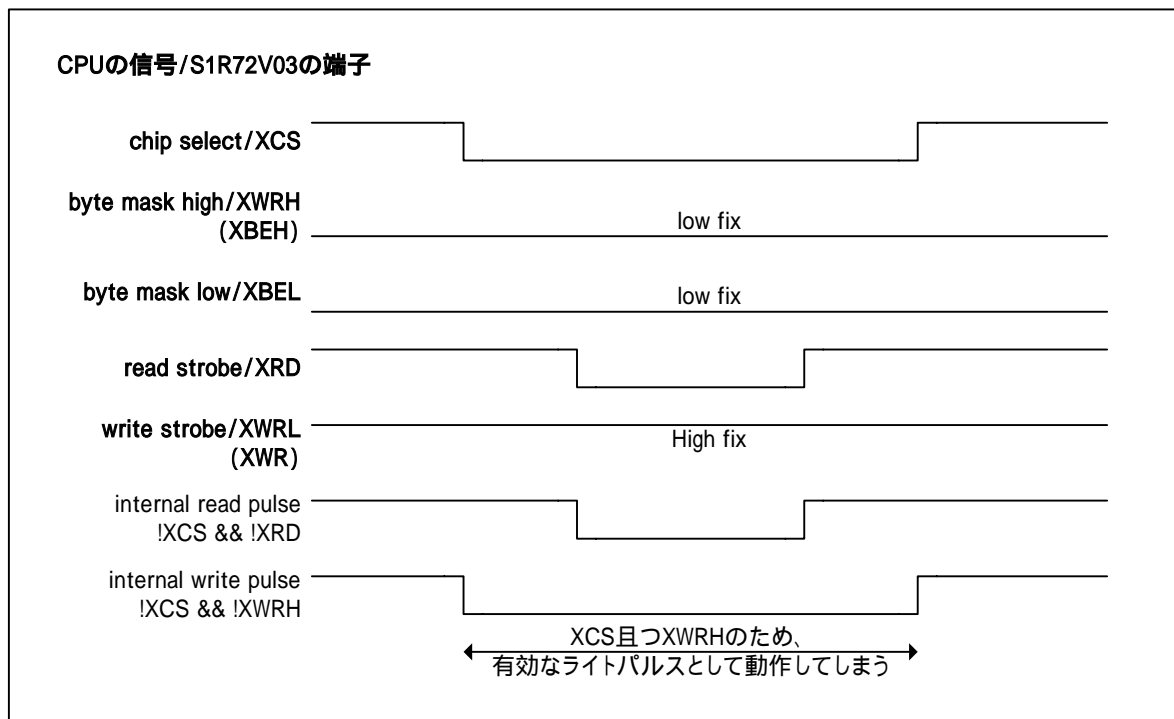


図 6.6 ChipConfig レジスタの初期設定前のリードアクセス

6. 機能説明

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、0 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、本 LSI にリードアクセスを行った場合、本 LSI 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

本 LSI の CPUIF (以下、CPUIF) のブロック構成を図 6.7 に示します。

REG/DMA0/DMA1 の 3 ブロックから構成されます。

- ・ REG : 本 LSI 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

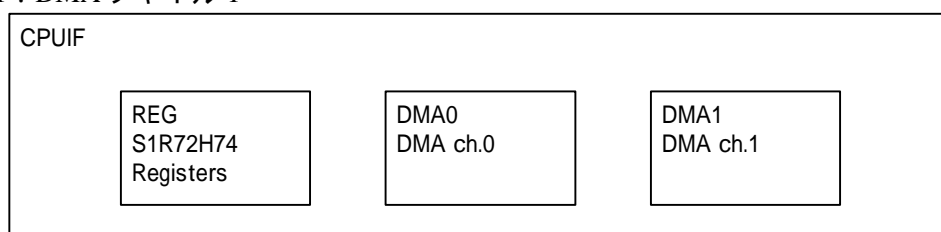


図 6.7 ブロック構成

6.4.3.1 REG (S1R72H74 Registers)

本 LSI のレジスタ領域へ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
- ・ 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) 期間を出力イネーブル期間としてレジスタのデータを外部バスへ出力します。

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 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。

FIFO アクセス (ライト) には以下の制限事項があります。

- ・ EPx{x=b-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ください。

- ・ 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ください。端数（奇数）バイトの書き込みを行う場合は FIFO のバイト境界を意識してストロブ信号を制御してください。詳細は“FIFO アクセスの端数処理”をご参照ください。
- ・ 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。

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=b-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ください。
- ・ 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ください。バイト読みを行う場合は EPnFIFO_ByteRd レジスタを用いて行ってください。バイト境界が存在する場合はバイト読みを行ってください。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”をご参照ください。

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=b-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 側に書き込んでください。この状態を図 6.8 の（1）に示します。IDE IF 等からはデータ A, B, C, D, … X, Y, Z の順に出力されます。

FIFO にバイト境界がある状態から書き込みを行う場合は、最初に Low 側にデータを書き込み（データ K の書き込み）バイト境界を解消した後、ワード書き込み（データ L, M）を行ってください。この状態を図 6.8 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

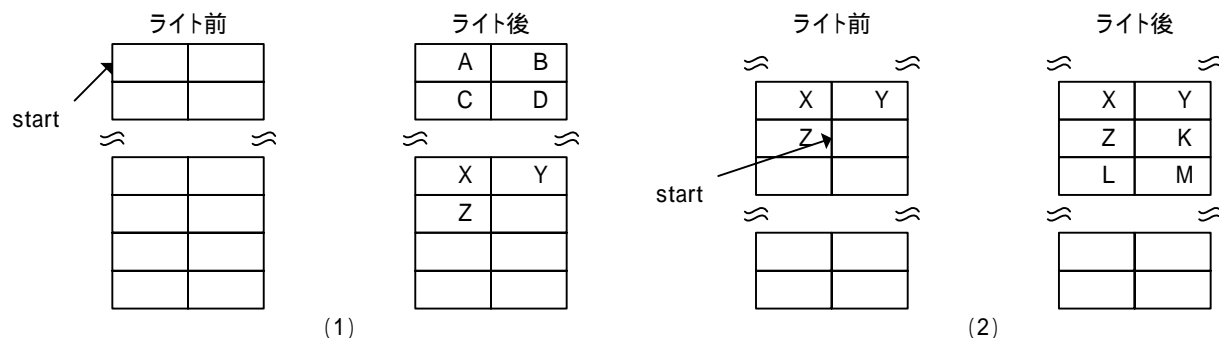


図 6.8 FIFO ライト処理（正常動作）

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

6. 機能説明

側のみ書き込みが行われます (図 6.9 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 6.9 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 6.9 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 側のみの書き込みが行われます (図 6.9 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。

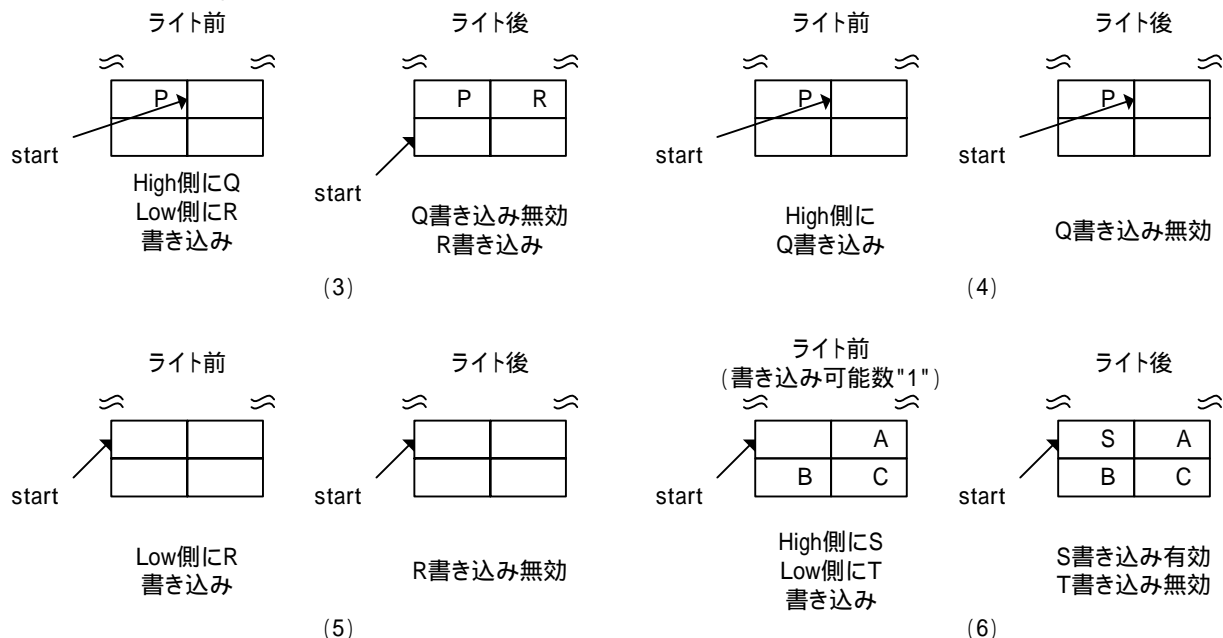


図 6.9 FIFO ライト処理 (注意が必要な動作)

【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ください。一旦バイト境界を解消した後にはワード読み出し、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 6.10 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 6.10 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。

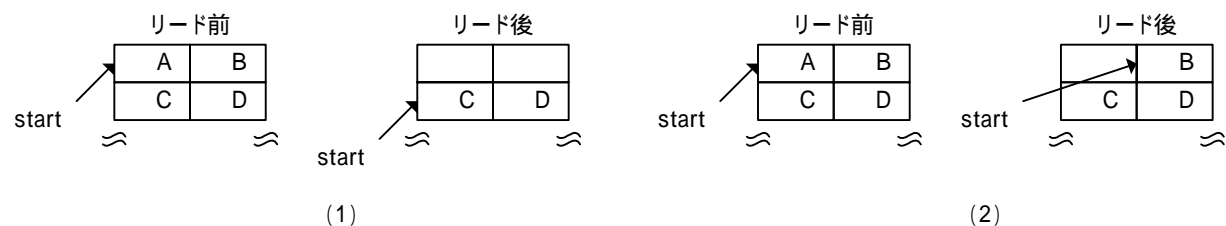


図 6.10 FIFO リード処理 (正常動作)

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 6.11 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 6.11 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。

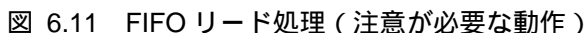


図 6.11 FIFO リード処理（注意が必要な動作）

- 1) FIFO に 64 バイト有るデータを 31 バイト 33 バイトで読み出したい場合。
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。
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。
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。
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。
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) 31 バイトのデータが有る時点で JoinCPU_Rd をセットして読み出しを開始した後、続けて書き込まれたデータを EPnFIFO_Rd_H,L レジスタでワード読みしたい場合。
31 バイトのデータが有る状態で JoinCPU_Rd を設定した時点で、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。
30 バイト分のデータをワード読みする。
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。
再度ジョインする。
CPUIF は残るデータ数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。
残りのデータをワード読みする。

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。

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

6.4.3.2.1 基本機能

設定されたカウント数分の DMA 転送を行います。

6. 機能説明

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出来ます。各々の動作モード、オプションにおける XDREQ のアサート条件、また XDREQ アサート時の転送可能数の関係を下表に示します。

表 6.2 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 Req		Count < Req	
	Ready Req	Ready < Req	Ready Count	Ready < Count
XDREQ	アサート	ネゲート	アサート	ネゲート
転送可能数	Req	-	Count	-

フリーランモード・ReqAssertCountオプション使用時(16bit / 8bitモード動作時共)

条件	フリーランモード	
	-	
	Ready Req	Ready < Req
XDREQ	アサート	ネゲート
転送可能数	Req	-

カウントモード・ReqAssertCountオプション未使用時(16bitモード動作時)

条件	カウントモード (Count > 0)		
	Count Ready		Count < Ready
	Ready 2	Ready < 2	Ready Count
XDREQ	アサート	ネゲート	アサート
転送可能数	Ready (Readyが奇数の場合はReady-1)	-	Count

フリーランモード・ReqAssertCountオプション未使用時(16bitモード動作時)

条件	フリーランモード	
	-	
	Ready 2	Ready < 2
XDREQ	アサート	ネゲート
転送可能数	Ready (Readyが奇数の場合はReady-1)	-

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 Ready		Count < Ready
	Ready 1	Ready < 1	Ready 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 1	Ready < 1
XDREQ	アサート	ネゲート
転送可能数	Ready	-

表中の Req は $\text{DMAx}\{x=0,1\}\text{Config.ReqAssertCount}$ の設定値を、Ready は FIFO の空き領域 / データ数を、Count は $\text{DMAx}\{x=0,1\}\text{Count_HH,HL,LH,LL}$ 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、 $\text{XDREQ}_x\{x=1,0\}$ の論理レベル、 $\text{XDACK}_x\{x=1,0\}$ 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. 機能説明

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。

表 6.3 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"をセットしてください。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、FIFO が空になった後、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

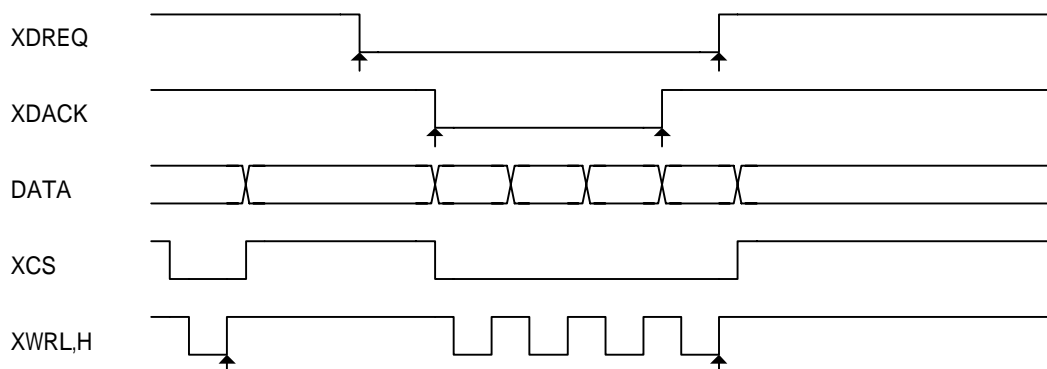
DMA 動作停止時に CPU_IntStat..DMAx{x=0,1}_Cmp ビットがセットされます。

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) を先に停止してください。

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 6.12 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

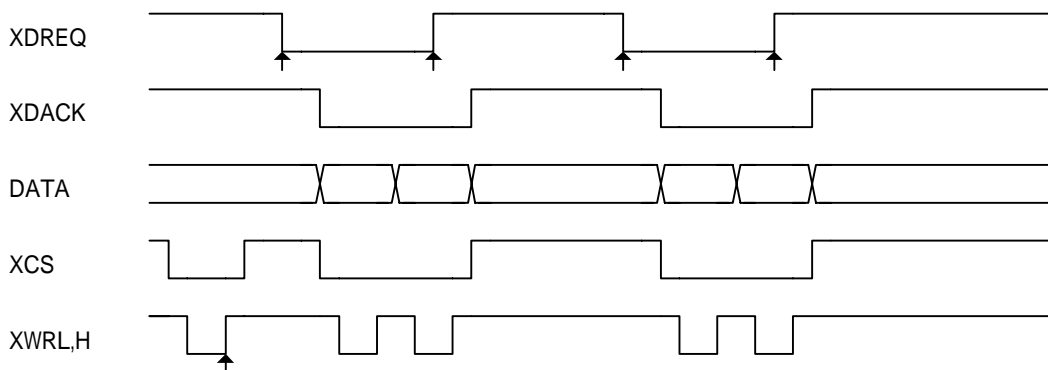


DMA_Control.DMA_Goビットへの"1"書き込みによりDMA回路動作開始
 データが転送されて、FIFOに空き領域(DMA_Ready)が出来、
 DMA_Readyを受けてXDREQをアサート
 XDACKがアサートされDMA転送開始
 カウントモードの転送数完了前に、マスタ側を停止、XDACKをネゲート
 DMA_Control.DMA_Stopビットへの"1"書き込みによりDMA回路停止
 DMA回路停止を受けてXDREQネゲート

図 6.12 カウントモードライトタイミング 1

カウントモードで転送を開始し、設定されたカウント数分の転送を完了し DMA 転送が終了する場合の動作タイミングを図 6.13 に示します。

ex2.【転送開始条件】カウント(8バイト) > FIFOの空き容量(4バイト) 【転送停止条件】カウント0



DMA_Control.DMA_Goビットへの"1"書き込みによりDMA回路動作開始
 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
 DMA_ReadyがなくなるタイミングでXDREQをネゲート
 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
 DMA_Countの最終データタイミングでXDREQをネゲート
 DMA_Count数分の転送終了によりDMA回路停止

図 6.13 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"をセットしてください。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、外部が

6. 機能説明

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。

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。

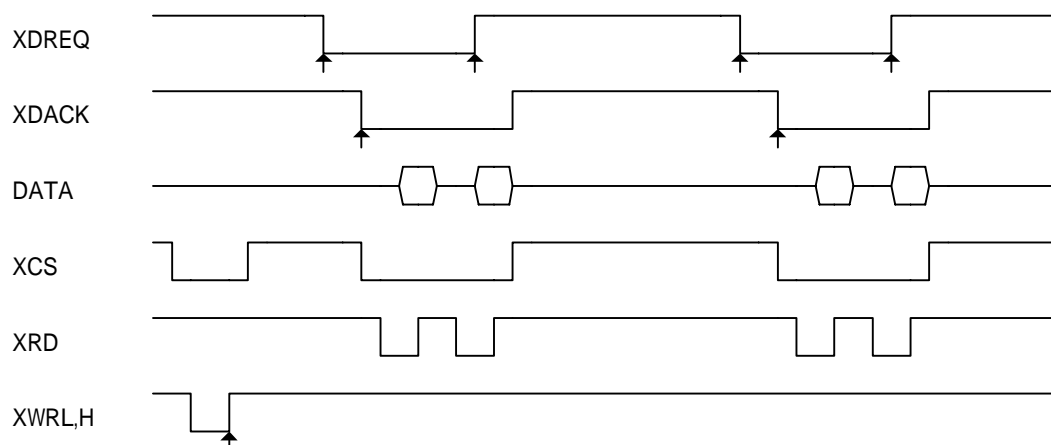
- ・ DMAx{x=0,1}_Count_HH,HL,LH,LL レジスタに設定したカウント数分の DMA 転送が完了
- ・ DMAx{x=0,1}_Control.DMA_Stop ビットに"1"をライト

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) を先に停止してください。

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 6.14 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



DMA_Control.DMA_Goビットへの"1"書き込みによりDMA回路動作開始
FIFOにデータが書き込まれ、
外部からデータ読み出し可能になることでXDREQをアサート
XDACKがアサートされDMA転送開始
FIFOデータが空になるタイミングでXDREQネゲート
FIFOにデータが書き込まれ、
外部からデータ読み出し可能になることでXDREQをアサート
XDACKがアサートされDMA転送開始
DMA_Countの最終データタイミングでXDREQネゲート

図 6.14 カウントモードリードタイミング

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"を書き込んでください。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) を先に停止してください。

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”をセットしてください。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) を先に停止してください。

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”をセットしてください。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でも、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。

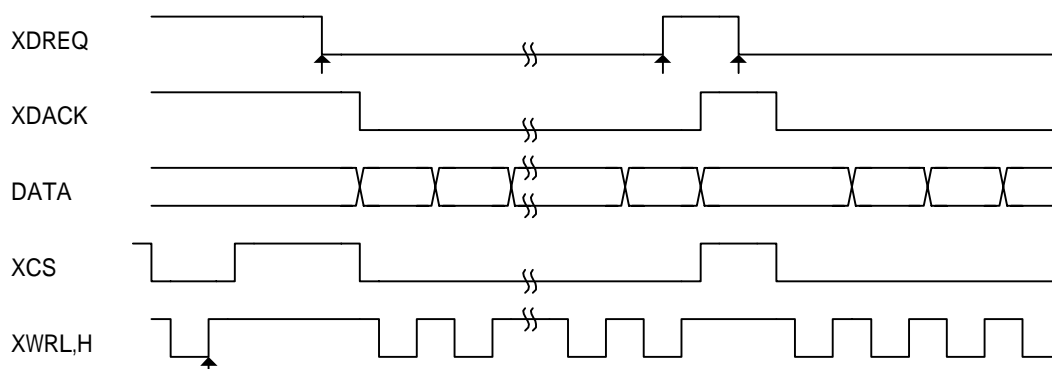
動作停止となるまで DMAx{x=0,1}_Control.DMA_Running ビットに”1”が読めます。

【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。

6. 機能説明

ex.【転送開始】REQアサートカウント(8beat:16byte)



DMA_Control.DMA_Goビットへの"1"書き込みによりDMA回路動作開始
DMA_Readyの値が連続転送数に満たないためXDREQはアサートしない
データが転送され、FIFOに連続転送数以上の有効な空き領域
(DMA_Ready)が出来、DMA_Readyを受けてXDREQをアサート
連続転送数(REQアサートカウント)終了のタイミングでXDREQネゲート
1回目の連続転送終了時点で、次の連続転送分の空き領域(DMA_Ready)がある
DMA_Readyを受けてDREQをアサート

図 6.15 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"をセットしてください。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。したがって、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 6.14、図 6.15 をご参照ください。

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。DMAにはバイト読み出しの口はございませんのでご注意ください。

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

6. 機能説明

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 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。その時までに 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. 機能説明

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 デバイスにはホストターミネートを通知します。

DMA 転送中にデバイスターミネートが発生した場合、IDE_IntStat.DetectTerm ビットが”1”になり接続された IDE デバイスは転送を止めてしまいますが、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 デバイスにはホストターミネートを通知します。

DMA 転送中にデバイスターミネートが発生した場合、IDE_IntStat.DetectTerm ビットが”1”になり接続された IDE デバイスは転送を止めてしまいますが、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。

6.5.5 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. 機能説明

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) 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 6.4 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 6.5 DEVICE_CODE

Version	1
Part Number	0x000E
Manufacturer	0x00BE

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、XI、XO、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 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

S1R72H74 は、S1R72V03 との互換性を保持するため、レジスタ名称を継承します。そのため、USB 固有の名称が使用されている場合があります。S1R72V03 でレジスタ定義され、S1R72H74 ではレジスタ定義されていないレジスタ・ビットについては、"Don't care"または"Don't touch"と明示しています。

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	(Don't care)	CPU_IntStat	FIFO_IntStat	(Don't care)	IDE_IntStat	(Don't care)	(Don't care)
0x01	(Don't care)		0xXX								
0x02	SIE_IntStat	R/(W)	0x00	(Don't care)	(Don't care)	FinishedPM	(Don't care)	(Don't care)	(Don't care)	(Don't care)	(Don't care)
0x03	CPU_IntStat	R/(W)	0x00		(Don't care)	DMA1_Countup	DMA1_Cmp			DMA0_Countup	DMA0_Cmp
0x04	FIFO_IntStat	R/(W)	0x00	(Don't care)	(Don't care)	FIFO_IDE_Cmp	FIFO1_Cmp	FIFO_Full	FIFO_Empty		FIFO0_Cmp
0x05	(Don't care)		0xXX								
0x06	IDE_IntStat	R/(W)	0x00	IDE_RegCmp	IDE_RegErr	IDE_SeqWrRegCmp	CompleteINTRQ		IDE_Cmp	DetectINTRQ	DetectTerm
0x07	(Don't care)		0xXX								
0x08	(Don't care)		0xXX								
0x09	(Don't care)		0xXX								
0x0A	(Don't care)		0xXX								
0x0B	(Don't care)		0xXX								
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	(Don't touch)	EnCPU_IntStat	EnFIFO_IntStat	(Don't touch)	EnIDE_IntStat	(Don't touch)	(Don't touch)
0x11	(Don't touch)		0xXX								
0x12	SIE_IntEnb	R/W	0x00	(Don't touch)	(Don't touch)	EnFinishedPM	(Don't touch)	(Don't touch)	(Don't touch)	(Don't touch)	(Don't touch)
0x13	CPU_IntEnb	R/W	0x00		(Don't touch)	EnDMA1_Countup	EnDMA1_Cmp			EnDMA0_Countup	EnDMA0_Cmp
0x14	FIFO_IntEnb	R/W	0x00	(Don't touch)	(Don't touch)	EnFIFO_IDE_Cmp	EnFIFO1_Cmp	EnFIFO_Full	EnFIFO_Empty		EnFIFO0_Cmp
0x15	(Don't touch)		0xXX								
0x16	IDE_IntEnb	R/W	0x00	EnIDE_RegCmp	EnIDE_RegErr	En_SeqWrRegCmp	EnCompleteINTRQ		EnIDE_Cmp	EnDetectINTRQ	EnDetectTerm
0x17	(Don't touch)		0xXX								
0x18	(Don't touch)		0xXX								
0x19	(Don't touch)		0xXX								
0x1A	(Don't touch)		0xXX								
0x1B	(Don't touch)		0xXX								
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	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x20	<i>RevisionNum</i>	R	0x30	<i>Revision Number</i>							
0x21	<i>ChipReset</i>	W	0xXX	(Don't touch)							<i>AllReset</i>
0x22	<i>PM_Control</i>	R/W	0x00		<i>PM_State [2:0]</i>			<i>GoSLEEP</i>	<i>GoSNOOZE</i>	<i>GoActive60</i>	(Don't touch)
0x23	(Don't touch)		0xXX								
0x24	(Don't touch)		0xXX								
0x25	(Don't touch)		0xXX								
0x26	(Don't touch)		0xXX								
0x27			0xXX								
0x28	(Don't touch)		0xXX								
0x29	<i>EPcFIFO_Clr</i>	W	0xXX						<i>EPcFIFO_Clr</i>	<i>EPbFIFO_Clr</i>	(Don't touch)
0x2A	<i>ClrAllJoin</i>	W	0xXX	<i>ClrJoinIDEs</i>	<i>ClrJoinDMA0_Rd</i>	<i>ClrJoinDMA0_Wr</i>	<i>ClrJoinDMA1_Rd</i>	<i>ClrJoinDMA1_Wr</i>	<i>ClrJoinCPU_Rd</i>	<i>ClrJoinCPU_Wr</i>	
0x2B											
0x2C	(Don't touch)		0xXX								
0x2D	(Don't touch)		0xXX								
0x2E	<i>WakeupTim_H</i>	R/W	0x00	<i>WakeupTim [15:8]</i>							
0x2F	<i>WakeupTim_L</i>	R/W	0x00	<i>WakeupTim [7:0]</i>							

Byte Addr	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x30	(Don't care)		0xXX								
0x31	(Don't care)		0xXX								
0x32	(Don't care)		0xXX								
0x33	(Don't care)		0xXX								
0x34	(Don't care)		0xXX								
0x35	(Don't care)		0xXX								
0x36	(Don't care)		0xXX								
0x37	(Don't care)		0xXX								
0x38	(Don't care)		0xXX								
0x39			0xXX								
0x3A	(Don't care)		0xXX								
0x3B			0xXX								
0x3C			0xXX								
0x3D			0xXX								
0x3E	(Don't care)		0xXX								
0x3F	(Don't care)		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
0x40	(Don't care)		0xXX								
0x41	(Don't touch)		0xXX								
0x42	(Don't care)		0xXX								
0x43	(Don't care)		0xXX								
0x44											
0x45	(Don't care)		0xXX								
0x46											
0x47											
0x48											
0x49											
0x4A											
0x4B											
0x4C											
0x4D											
0x4E											
0x4F											

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	(Don't care)		0xXX								
0x52	(Don't care)		0xXX								
0x53			0xXX								
0x54	(Don't care)		0xXX								
0x55	(Don't care)		0xXX								
0x56			0xXX								
0x57			0xXX								
0x58	(Don't care)		0xXX								
0x59	(Don't care)		0xXX								
0x5A	EPbConfig_0	R/W	0x00	INxOUT	(Don't care)	(Don't care)		(Don't care)	(Don't care)	(Don't care)	(Don't care)
0x5B			0xXX								
0x5C	(Don't care)		0xXX								
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	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x60	(Don't care)		0xXX								
0x61	(Don't care)		0xXX								
0x62	EPcConfig_0	R/W	0x00	INxOUT	(Don't care)	(Don't care)		(Don't care)	(Don't care)	(Don't care)	(Don't care)
0x63			0xXX								
0x64	(Don't care)		0xXX								
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	(Don't touch)		0xXX								
0x6D	(Don't touch)		0xXX								
0x6E	(Don't touch)		0xXX								
0x6F	(Don't touch)		0xXX								

Byte Addr	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	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	(Don't care)		0xXX								
0x79	(Don't care)		0xXX								
0x7A	(Don't care)		0xXX								
0x7B	(Don't care)		0xXX								
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	(Don't care)						
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			0xXX								
0x84	DMA0_Remain_H	R	0x00				DMA_Remain [12:8]				
0x85	DMA0_Remain_L	R	0x00	DMA_Remain [7:0]							
0x86			0xXX								
0x87			0xXX								
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	0xXX	DMA_RdData[15:8]							
0x8D	DMA0_RdData_L	R	0xXX	DMA_RdData[7:0]							
0x8E	DMA0_WrData_H	W	0xXX	DMA_WrData[15:8]							
0x8F	DMA0_WrData_L	W	0xXX	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	(Don't care)						
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			0xXX								
0x94	DMA1_Remain_H	R	0x00				DMA_Remain [12:8]				
0x95	DMA1_Remain_L	R	0x00	DMA_Remain [7:0]							
0x96			0xXX								
0x97			0xXX								
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	0xXX	DMA_RdData[15:0]							
0x9D	DMA1_RdData_L	R	0xXX	DMA_RdData[7:0]							
0x9E	DMA1_WrData_H	W	0xXX	DMA_WrData[15:8]							
0x9F	DMA1_WrData_L	W	0xXX	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_Comfig_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	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	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								

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	(Don't care)		0xFF								
0xC1	(Don't care)		0xFF								
0xC2	(Don't care)		0xFF								
0xC3	(Don't care)		0xFF								
0xC4	(Don't care)		0xFF								
0xC5	(Don't care)		0xFF								
0xC6	(Don't care)		0xFF								
0xC7	(Don't care)		0xFF								
0xC8	(Don't care)		0xFF								
0xC9	(Don't care)		0xFF								
0xCA	(Don't care)		0xFF								
0xCB	(Don't care)		0xFF								
0xCC	(Don't care)		0xFF								
0xCD	(Don't care)		0xFF								
0xCE	(Don't care)		0xFF								
0xCF	(Don't care)		0xFF								

Byte Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0xD0	(Don't care)		0xFF								
0xD1	(Don't care)		0xFF								
0xD2	(Don't care)		0xFF								
0xD3	(Don't care)		0xFF								
0xD4	(Don't care)		0xFF								
0xD5	(Don't care)		0xFF								
0xD6	(Don't care)		0xFF								
0xD7	(Don't care)		0xFF								
0xD8	(Don't care)		0xFF								
0xD9	(Don't care)		0xFF								
0xDA	(Don't care)		0xFF								
0xDB	(Don't care)		0xFF								
0xDC	(Don't care)		0xFF								
0xDD	(Don't care)		0xFF								
0xDE	(Don't care)		0xFF								
0xDF	(Don't care)		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 bit7	bit14 bit6	bit13 bit5	bit12 bit4	bit11 bit3	bit10 bit2	bit9 bit1	bit8 bit0
0xE0	(Don't care)		0xFF								
0xE1	(Don't care)		0xFF								
0xE2	(Don't touch)		0xFF								
0xE3	(Don't care)		0xFF								
0xE4			0xFF								
0xE5			0xFF								
0xE6			0xFF								
0xE7			0xFF								
0xE8			0xFF								
0xE9			0xFF								
0xEA			0xFF								
0xEB	ModeProtect	R/W	0x56	<i>Protected[7:0](56 以外を書くとはプロテクト、0x56 で解除)</i>							
0xEC			0xFF								
0xED	ClkSelect	R/W	0x01	<i>xActID_E_Term</i>							<i>ClkSelect</i>
0xEE			0xFF								
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	(Don't touch)		0xFF								
0xF1	(Don't touch)		0xFF								
0xF2	(Don't touch)		0xFF								
0xF3	(Don't touch)		0xFF								
0xF4	(Don't touch)		0xFF								
0xF5	(Don't touch)		0xFF								
0xF6	(Don't touch)		0xFF								
0xF7	(Don't touch)		0xFF								
0xF8	(Don't touch)		0xFF								
0xF9	(Don't touch)		0xFF								
0xFA	(Don't touch)		0xFF								
0xFB	(Don't touch)		0xFF								
0xFE	(Don't touch)		0xFF								
0xFD	(Don't touch)		0xFF								
0xFE	(Don't touch)		0xFF								
0xFF	(Don't touch)		0xFF								

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: (Don't care)	0:	1: Undefined	
		R	5: CPU_IntStat	0: None	1: CPU Interrupts	
		R	4: FIFO_IntStat	0: None	1: FIFO Interrupts	
		R	3: (Don't care)	0:	1: Undefined	
		R	2: IDE_IntStat	0: None	1: IDE Interrupts	
		R	1: (Don't care)	0:	1: Undefined	
		R (W)	0: (Don't care)	0:	1: Undefined	

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** **(Don't care)**
- 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** **(Don't care)**
- 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。
- Bit1-0** **(Don't care)**

7. レジスタ

7.2.2 02h SIE_IntStat (SIE Interrupt Status)

Address	Register Name	R / W	Bit Symbol	Description		Reset
02h	<i>SIE_IntStat</i>	R (W)	7: (Don't care)	0:	1: Undefined	00h
		R (W)	6: (Don't care)	0:	1: Undefined	
		R (W)	5: FinishedPM	0: None	1: Detect FinishedPM	
		R (W)	4: (Don't care)	0:	1: Undefined	
		R (W)	3: (Don't care)	0:	1: Undefined	
		R (W)	2: (Don't care)	0:	1: Undefined	
		R (W)	1: (Don't care)	0:	1: Undefined	
		R (W)	0: (Don't care)	0:	1: Undefined	

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。

Bit7-6 (Don't care)

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。

Bit4-0 (Don't care)

7.2.3 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: (Don't care)	0:	1: Undefined	
		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 **(Don't care)**

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. レジスタ

7.2.4 04h FIFO_IntStat (FIFO Interrupt Status)

Address	Register Name	R / W	Bit Symbol	Description		Reset
04h	FIFOIntStat	R (W)	7: (Don't care)	0:	1: Undefined	00h
		R (W)	6: (Don't care)	0:	1: Undefined	
		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。

Bit7-6 (Don't care)

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 FIFO 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 FIFO 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 FIFO 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 FIFO 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 FIFO 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 FIFO 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 FIFO 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 FIFO 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。

7.2.5 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	1: Register Access Complete	00h
		R (W)	6: IDE_RegErr	0: None	1: Register Access Error	
		R (W)	5: IDE_SeqWrRegCmp	0: None	1: Sequence Write Complete	
		R (W)	4: CompleteINTRQ	0: None	1: Auto Status Read Compl.	
			3:	0:	1:	
		R (W)	2: IDE_Cmp	0: None	1: DMA Complete	
		R (W)	1: DetectINTRQ	0: None	1: Detected Interrupt	
		R (W)	0: DetectTerm	0: None	1: Detected Device terminate	

IDE 関連の割り込みステータスを表示します。全てのビットは"1"を書き込む事で割り込み要因をクリアできます。

- Bit7 IDE_RegCmp**
割り込み要因を直接指示します。
IDE_RegAdrs レジスタによる IDE レジスタへのリードライトアクセスが完了した時に"1"にセットされます。
- Bit6 IDE_RegErr**
割り込み要因を直接指示します。
以下の場合に"1"にセットされます。
1) IDE_RegConfig レジスタによる自動ステータスリードの動作シーケンス中に IDE_RegAdrs レジスタによる IDE レジスタへのリードライトアクセスを行った。
2) IDE_SeqWrRegControl レジスタによるシーケンスライトの動作シーケンス中に IDE_RegAdrs レジスタによる IDE レジスタへのリードライトアクセスを行った。
- Bit5 IDE_SeqWrRegCmp**
割り込み要因を直接指示します。
IDE_SeqWrRegControl レジスタによるシーケンスライトの動作が完了した時に"1"にセットされます。
- Bit4 CompleteINTRQ**
割り込み要因を直接指示します。
IDE_RegConfig レジスタによる自動ステータスリードの動作が完了した時に"1"にセットされます。
- Bit3 Reserved**
- Bit2 IDE_Cmp**
割り込み要因を直接指示します。
IDE_Control レジスタによる DMA 動作が完了した時に"1"にセットされます。
- Bit1 DetectINTRQ**
割り込み要因を直接指示します。
IDE_RegConfig レジスタの EnAutoStsRd ビットがセットされていないとき、IDE の HINTRQ 信号の立ち上がりが検出された時に"1"にセットされます。
- Bit0 DetectTerm**
割り込み要因を直接指示します。
IDE_Control レジスタによる IDE の DMA 動作中で転送モードが Ultra-DMA の場合、デバイスターミネートを検出した時に"1"にセットされます。

7. レジスタ

7.2.6 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: (Don't touch)	0:	1: Undefined	
		R / W	5: EnCPU_IntStat	0: Disable	1: Enable	
		R / W	4: EnFIFO_IntStat	0: Disable	1: Enable	
		R / W	3: (Don't touch)	0:	1: Undefined	
		R / W	2: EnIDE_IntStat	0: Disable	1: Enable	
		R / W	1:	0:	1:	
		R / W	0:	0:	1:	

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。
EnPM_IntStat / EnSIE_IntStat ビットはスリープ / スヌーズ中も有効です。

7.2.7 12h SIE_IntEnb (SIE Interrupt Enable)

Address	Register Name	R / W	Bit Symbol	Description		Reset
12h	SIE_IntEnb	R / W	7: (Don't touch)	0:	1: Undefined	00h
		R / W	6: (Don't touch)	0:	1: Undefined	
		R / W	5: EnFinishedPM	0: Disable	1: Enable	
		R / W	5: (Don't touch)	0:	1: Undefined	
		R / W	4: (Don't touch)	0:	1: Undefined	
		R / W	2: (Don't touch)	0:	1: Undefined	
		R / W	1: (Don't touch)	0:	1: Undefined	
		R / W	0: (Don't touch)	0:	1: Undefined	

SIE_IntStat レジスタの割り込み要因による、MainIntStat レジスタの PM_IntStat / SIE_IntStat ビットのアサートを許可 / 禁止します。
EnFinishedPM ビットはスリープ / スヌーズ中も有効です。

7.2.8 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: (Don't touch)	0:	1: Undefined	
		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。

7.2.9 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。

7.2.10 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。

7.2.11 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

7.2.12 21h ChipReset (Chip Reset)

Address	Register Name	R / W	Bit Symbol	Description		Reset
21h	ChipReset	R / W	7: (Don't touch)	0:	1: Undefined	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 (Don't touch)

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、書き込みしないでください。
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ください。

7.2.13 22h PM_Control (Power Management Control)

Address	Register Name	R / W	Bit Symbol	Description		Reset
22h	PM_Control		7:	0:	1:	80h
		R	6: PM_State [2]	PM_State [2:0]		
			5: PM_State [1]			
			4: PM_State [0]			
		R / W	3: GoSLEEP	0: Do nothing	1: Go to Sleep	
		R / W	2: GoSNOOZE	0: Do nothing	1: Go to Snooze	
		R / W	1: GoActive60	0: Do nothing	1: Go to Active60	
		R / W	0: (Don't touch)	0:	1: Undefined	

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_Contrl.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。
- 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 Don't touch**

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ください。

<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. レジスタ

7.2.14 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: (Don't touch)	0:	1: Undefined	

該当する FIFO 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。
FIFO 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 FIFO エンドポイントのビットを"1"にセットしないでください。

7.2.15 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: ClrJionDMA0_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 FIFO 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。
FIFO エンドポイントがポートに接続 (EP_x{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"にセットしないでください。誤動作の原因となります。

7.2.16 2Eh WakeupTim_H (Wakeup Time High)

7.2.17 2Fh WakeupTim_L (Wakeup Time Low)

Address	Register Name	R / W	Bit Symbol	Description	Reset
2Eh -2Fh	WakeupTim_H WakeupTim_L	R / W	WakeupTim [15:0]	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。

7.2.18 5Ah EPbConfig_0 (EPb Configuration 0)

Address	Register Name	R / W	Bit Symbol	Description	Reset
5Ah	EPbConfig_0	R / W	7: INxOUT	0: OUT(IDE Write) 1: IN(IDE Read)	00h
			6: (Don't care)	0: 1: Undefined	
			5: (Don't care)	0: 1: Undefined	
			4:	0: 1:	
			3: (Don't care)	0: 1: Undefined	
			2: (Don't care)	0: 1: Undefined	
			1: (Don't care)	0: 1: Undefined	
			0: (Don't care)	0: 1: Undefined	

FIFO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FIFOエンドポイントと重複しないように設定してください。

Bit7 **INxOUT**
FIFOエンドポイントの転送方向を設定します。

Bit6-5 **(Don't care)**

Bit4 **Reserved**

Bit3-0 **(Don't care)**

7. レジスタ

7.2.19 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	

FIFO 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**
FIFO 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、FIFO 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、FIFO 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**
FIFO 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**
FIFO 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 FIFO 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、FIFO 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**
FIFO 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 FIFO 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、FIFO 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**
FIFO 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 FIFO 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、FIFO 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**
FIFO 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 FIFO 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、FIFO 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**
FIFO 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 EPn FIFO_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 FIFO 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、FIFO 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**
FIFO 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 EPn FIFO_Wr_H,L レジスタへのライトが行われると、この FIFO 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、FIFO 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、FIFO エンドポイントが OUT 方向の場合は残りデータ数、FIFO 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" を書きこんだ場合、上位ビットが有効とされます。

7.2.20 62h EPcConfig_0 (EPc Configuration 0)

Address	Register Name	R / W	Bit Symbol	Description		Reset
62h	EPcConfig_0	R / W	7: INxOUT	0: OUT(IDE Write)	1: IN(IDE Read)	00h
			6: (Don't care)	0:	1: Undefined	
			5: (Don't care)	0:	1: Undefined	
			4:	0:	1:	
			3: (Don't care)	0:	1: Undefined	
			2: (Don't care)	0:	1: Undefined	
			1: (Don't care)	0:	1: Undefined	
			0: (Don't care)	0:	1: Undefined	

FIFO 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 FIFO エンドポイントと重複しないように設定してください。

Bit7 **INxOUT**
FIFO エンドポイントの転送方向を設定します。

Bit6-5 **(Don't care)**

Bit4 **Reserved**

Bit3-0 **(Don't care)**

7. レジスタ

7.2.21 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	

FIFO 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**
FIFO 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、FIFO 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、FIFO 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**
FIFO 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**
FIFO 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 FIFO 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、FIFO 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**
FIFO 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 FIFO 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、FIFO 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**
FIFO 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 FIFO 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、FIFO 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**
FIFO 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 FIFO 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、FIFO 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**
FIFO 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 EPn FIFO_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 FIFO 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、FIFO 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**
FIFO 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 EPn FIFO_Wr_H,L レジスタへのライトが行われると、この FIFO 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、FIFO 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、FIFO エンドポイントが OUT 方向の場合は残りデータ数、FIFO 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"を書きこんだ場合、上位ビットが有効とされます。

7.2.22 70h EPnFIFO_Rd_H (EPn FIFO Read High)

7.2.23 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 FIFO 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

7.2.24 72h EPnFIFO_Wr_H(EPn FIFO Write High)

7.2.25 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 FIFO 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。

7.2.26 74h EPnRdRemain_H (EPn FIFO Read Remain High)

7.2.27 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 FIFO 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 FIFO 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ください。

7. レジスタ

7.2.28 76h EPnWrRemain_H (EPn FIFO Write Remain High)

7.2.29 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 FIFO 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ください。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ください。

7.2.30 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 FIFO 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。

7.2.31 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 FIFO 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 FIFO 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 FIFO 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

7.2.32 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 をアクノリッジとして DMA 動作します。
- 1 : ホストからの DMA0_RdData/DMA0_WrData レジスタに対するアクセスをアクノリッジとして DMA 動作します。

Bit5-4

Reserved

Bit3

ActiveDMA

DMA0 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"にセットしてください。

- 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。

7.2.33 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	
		W	0: DMA_Go	0: Do nothing	1: Start DMA	

DMA0 の制御、及び状態を表示します。

- Bit7 DMA_Running**
DMA0 の転送中、このビットが"1"にセットされます。このビットが"1"である間は、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ください。
- 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

7.2.34 84h DMA0_Remain_H (DMA0 FIFO Remain High)

7.2.35 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 FIFO 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 FIFO 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 の順に読み出して下さい。

7.2.36 88h DMA0_Count_HH (DMA0 Transfer Byte Counter High/High)

7.2.37 89h DMA0_Count_HL (DMA0 Transfer Byte Counter High/Low)

7.2.38 8Ah DMA0_Count_LH (DMA0 Transfer Byte Counter Low/High)

7.2.39 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. レジスタ

7.2.40 8Ch DMA0_RdData_H (DMA0 Read Data High)

7.2.41 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 FIFO 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。

7.2.42 8Eh DMA0_WrData_H (DMA0 Write Data High)

7.2.43 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 FIFO 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。

7.2.44 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 FIFO 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 FIFO 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 FIFO 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

7.2.45 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"にセットしてください。

- 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。

7.2.46 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ください。
- 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. レジスタ

7.2.47 94h DMA1_Remain_H (DMA1 FIFO Remain High)

7.2.48 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 FIFO 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 FIFO 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。

7.2.49 98h DMA1_Count_HH (DMA1 Transfer Byte Counter High/High)

7.2.50 99h DMA1_Count_HL (DMA1 Transfer Byte Counter High/Low)

7.2.51 9Ah DMA1_Count_LH (DMA1 Transfer Byte Counter Low/High)

7.2.52 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

7.2.53 9Ch DMA1_RdData_H (DMA1 Read Data High)

7.2.54 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 FIFO 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。

7.2.55 9Eh DMA1_WrData_H (DMA1 Write Data High)

7.2.56 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 FIFO 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。

7.2.57 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”を読み出すことが出来ます。

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. レジスタ

7.2.58 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_Cm p ビットが"1"になります。
DMA の転送中はこのビットが"1"にセットされて、終了するとこのビットは"0"に戻ります。
このビットが"1"である間に"0"を書き込むと実行中の DMA 転送は中断されて終了しますが、IDE_IntStat レジスタの IDE_Cm p ビットはセットされません。

7.2.59 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 μ 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 μ 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 μ 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 μ 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. レジスタ

7.2.60 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ください。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 を参照してください。
- Bit1-0** **Reserved**

7.2.61 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 XHIO_R/XHIO_W アサート・ネゲート時のストローブ幅を設定します。

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 × 16.67nS = 67nS
 0001: 5 × 16.67nS = 83nS

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 × 16.67nS = 67nS
 0001: 5 × 16.67nS = 83nS

7. レジスタ

7.2.62 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ーブ幅を設定します。

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

7.2.63 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. レジスタ

7.2.64 AAh IDE_CRC_H (IDE CRC High)

7.2.65 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ください。

7.2.66 ADh IDE_Count_H (IDE Transfer Byte Counter High)

7.2.67 AEh IDE_Count_M (IDE Transfer Byte Counter Middle)

7.2.68 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ください。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. レジスタ

7.2.69 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

7.2.70 B2h IDE_RdRegValue_H (IDE Register Read Value High)

7.2.71 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ください。

7. レジスタ

7.2.72 B4h IDE_WrRegValue_H (IDE Register Write Value High)

7.2.73 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。

7.2.74 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

7.2.75 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. レジスタ

7.2.76 B8h IDE_SeqWrRegAdrs (IDE Sequential Register Write Address FIFO)

Address	Register Name	R / W	Bit Symbol	Description	Reset
B8h	IDE_SeqWrRegAdrs		7:	0:	1:
			6:	0:	1:
			5:	0:	1:
			4:	0:	1:
		W	3: IDE_SeqRegAddress [3]	IDE_SeqRegAddress [3:0]	XXh
			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。

7.2.77 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。

7.2.78 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**

7.2.79 EBh ModeProtect(Mode Protection)

Address	Register Name	R / W	Bit Symbol	Description	Reset
EBh	<i>ModeProtect</i>	R / W	7: <i>ModeProtect [7]</i>	Mode Protection	56h
			6: <i>ModeProtect [6]</i>		
			5: <i>ModeProtect [5]</i>		
			4: <i>ModeProtect [4]</i>		
			3: <i>ModeProtect [3]</i>		
			2: <i>ModeProtect [2]</i>		
			1: <i>ModeProtect [1]</i>		
			0: <i>ModeProtect [0]</i>		

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ください。

7. レジスタ

7.2.80 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

7.2.81 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 端子を固定してください。

- 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ください。

- 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

8. 電気的特性

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
入力電圧	HVI	VSS-0.3 ~ HVDD+0.5	V
	CVI	VSS-0.3 ~ CVDD+0.5	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	

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
入力電圧	HVI	-0.3	-	HVDD+0.3	V
	CVI	-0.3	-	CVDD+0.3	V
周囲温度	T _a	-40	25	85	

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(内部) HVDD, CVDD(IO 部)

また、本 IC は下記手順で電源オフを行ってください。

HVDD, CVDD(IO 部) LVDD(内部)

注)

LVDD 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。

8.3 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)	-	TBD	TBD	mA	
		IDDH	HVDD=3.3V(typ), 3.6V(max)	-	TBD	TBD	mA
		IDDCH	CVDD=3.3V(typ), 3.6V(max)	-	TBD	TBD	mA
		IDDCL	CVDD=1.8V(typ), 1.95V(max)	-	TBD	TBD	mA
静止電流 2							
電源電流	IDDS	VIN = HVDD or VSS	-	-	80	μA	
		LVDD = 1.95V					
		HVDD=3.6V					
		CVDD=3.6					
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 を接続しデータを送受信している状態(実転送レート TBD MB/s)での測定値。max 値は同値からの見積値。

2: Ta=25℃、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
入力特性 (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 (LVTTTL) 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
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 IOL = 2.7mA	-	-	VSS+0.4	V

8. 電気的特性

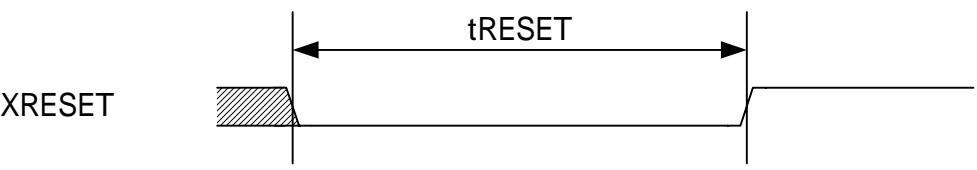
項目	記号	条件	MIN	TYP	MAX	単位
出力特性	端子名: CA [15:0], XINT, HDD [15:0], HDA [2:0], XHCS1, XHCS0, XHDMACK, XHRESET					
OFF-STATE リーク電流	IOZ	HVDD = 3.3V CVDD = 3.3V CVDD = 1.8V VOH = HVDD/CVDD VOL = VSS	-2	-	2	μA
入力特性	端子名: HDD[15:8], HDD[6:0], HIORDY, HINTRQ, XHDASP, XHPDIAG					
ブルアップ抵抗	RPLU	HVDD = 3.0V VIL = VSS	-16.4	-	-38.5	μA
入力特性	端子名: HDD7, HDMARQ,					
ブルダウン抵抗	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					
入出力端子容量 1	CIO1	f = 1MHz HVDD = LVDD = VSS PVDD = PVSS	-	-	8	pF

8. 電気的特性

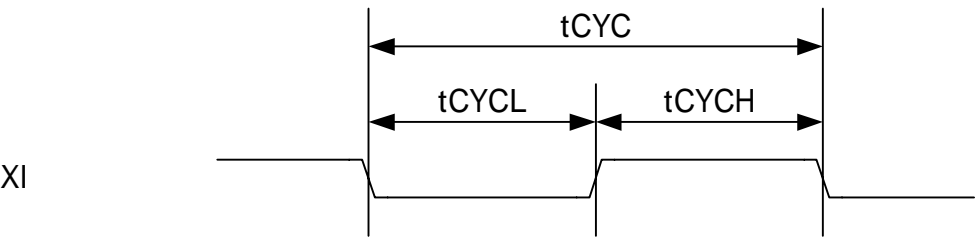
8.4 AC 特性

8.4.1 RESET タイミング



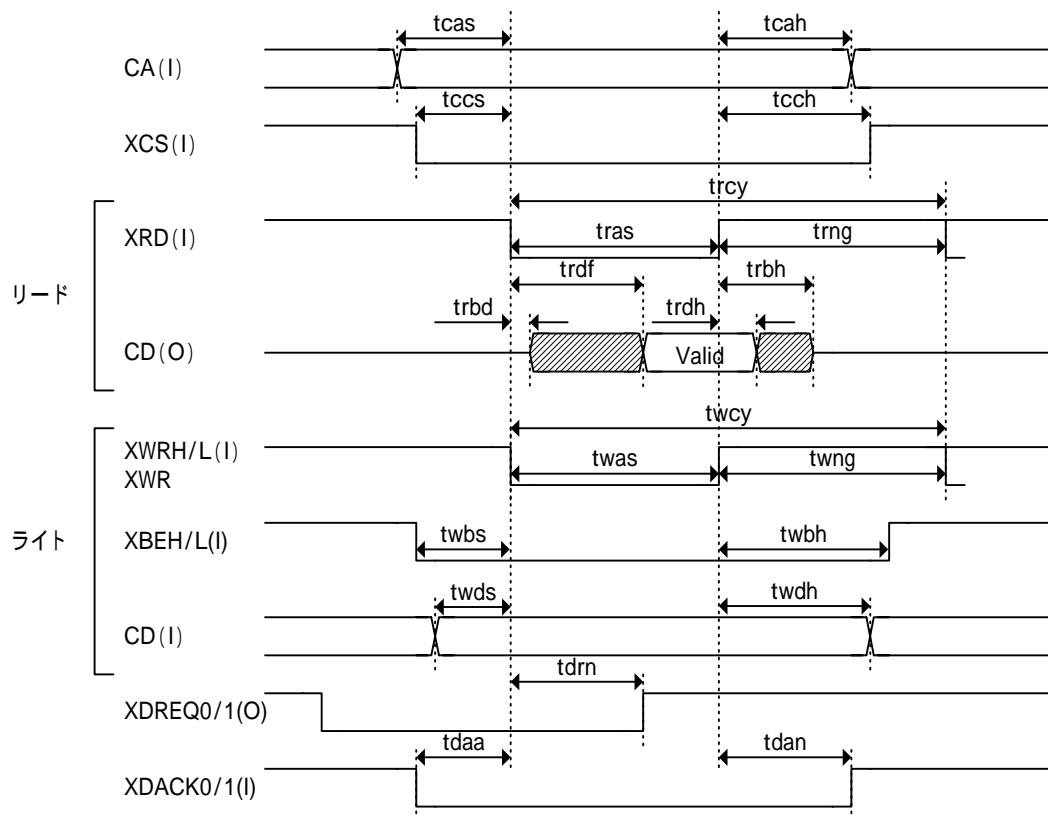
記号	説明	min	typ	max	単位
t_{RESET}	リセットパルス幅	40	-	-	ns

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

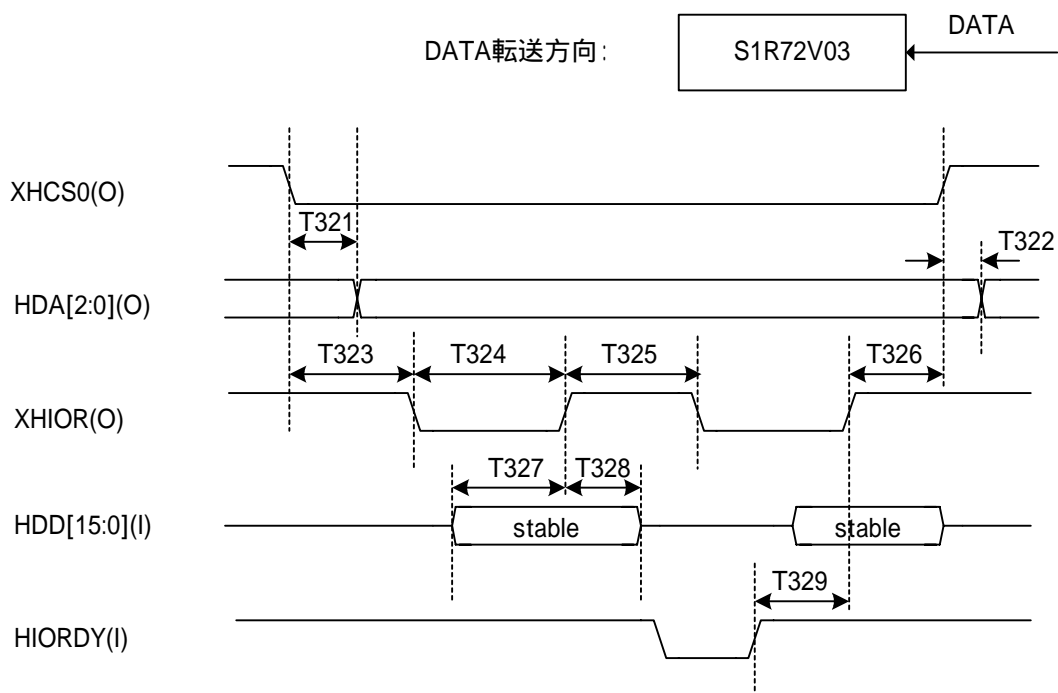
CLVDD=1.8V

(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



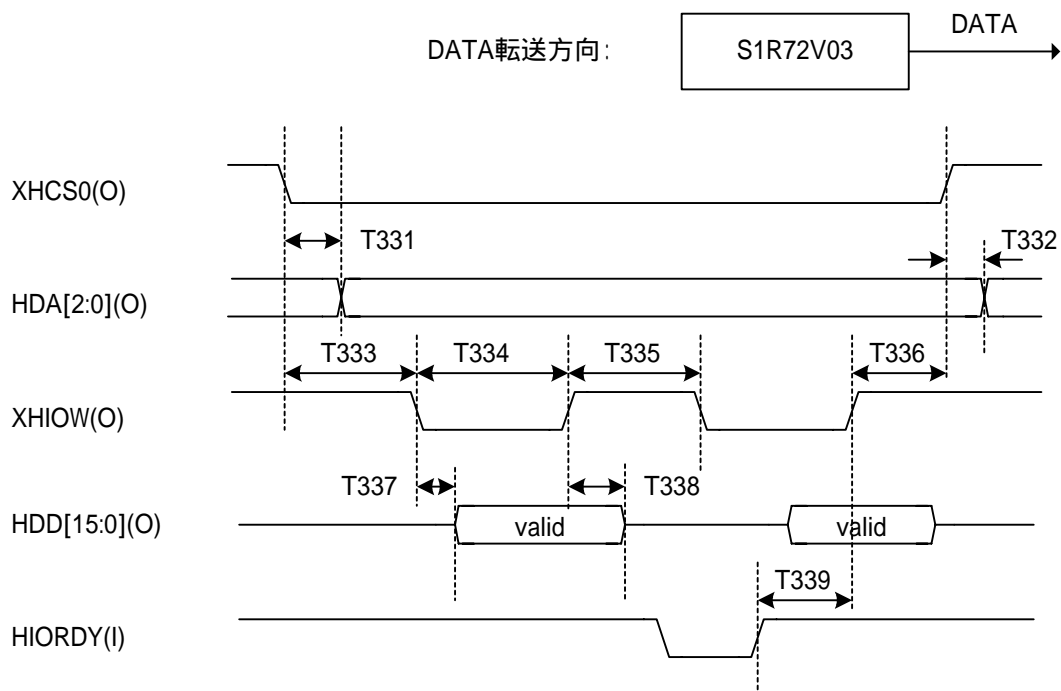
記号	説明	min	typ	max	単位
T321	XHCS0 HDA HDA出力遅延時間	-	0	-	ns
T322	XHCS0 HDA HDAホールド時間	-	0	-	ns
T323	XHCS0 XHIOR XHCS0セットアップ時間	25	-	-	ns
T324	XHIOR XHIOR XHIORアサートパルス幅	-	(AP+4) * 16.7 - 3	-	ns
T325	XHIOR XHIOR XHIORネゲートパルス幅	-	(NP+4) * 16.7 + 3	-	ns
T326	XHIOR XHCS0 XHCS0ホールド時間	10	-	-	ns
T327	HDD XHIOR データセ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

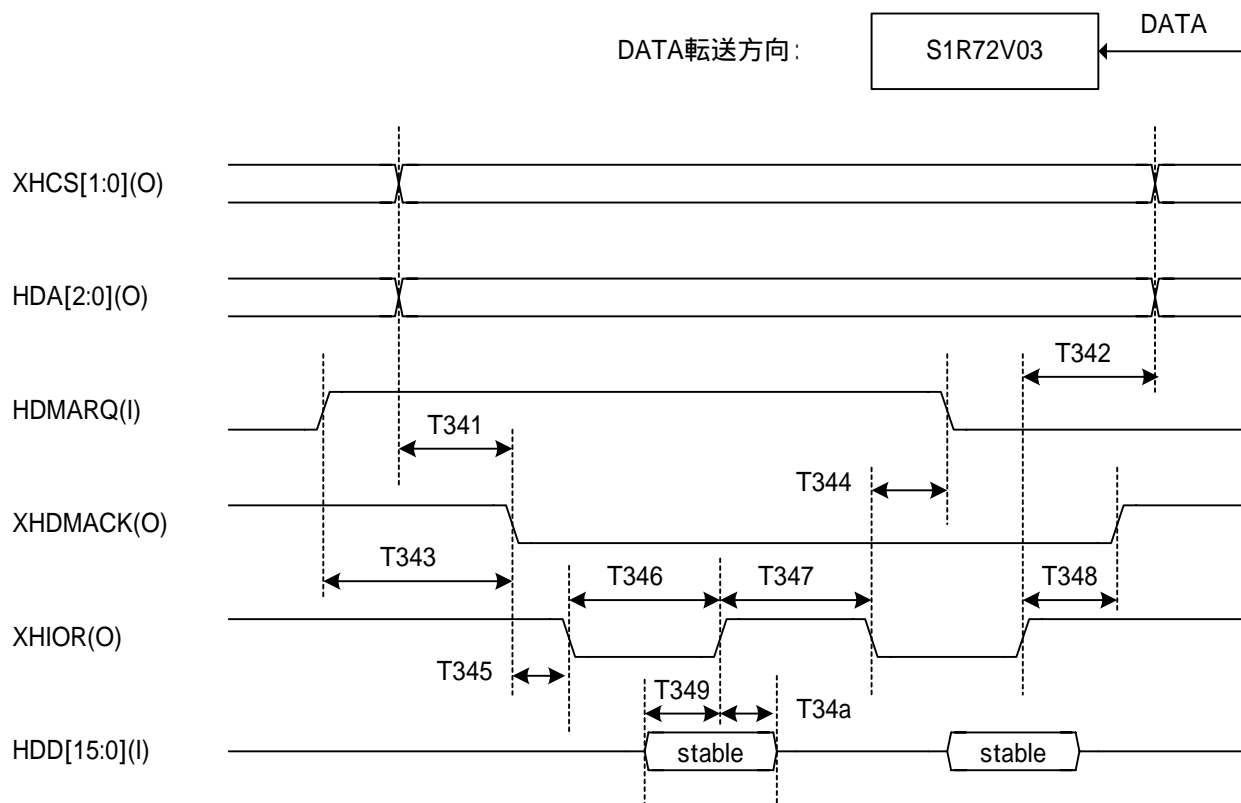
8.4.4.2 PIO Write Timing



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

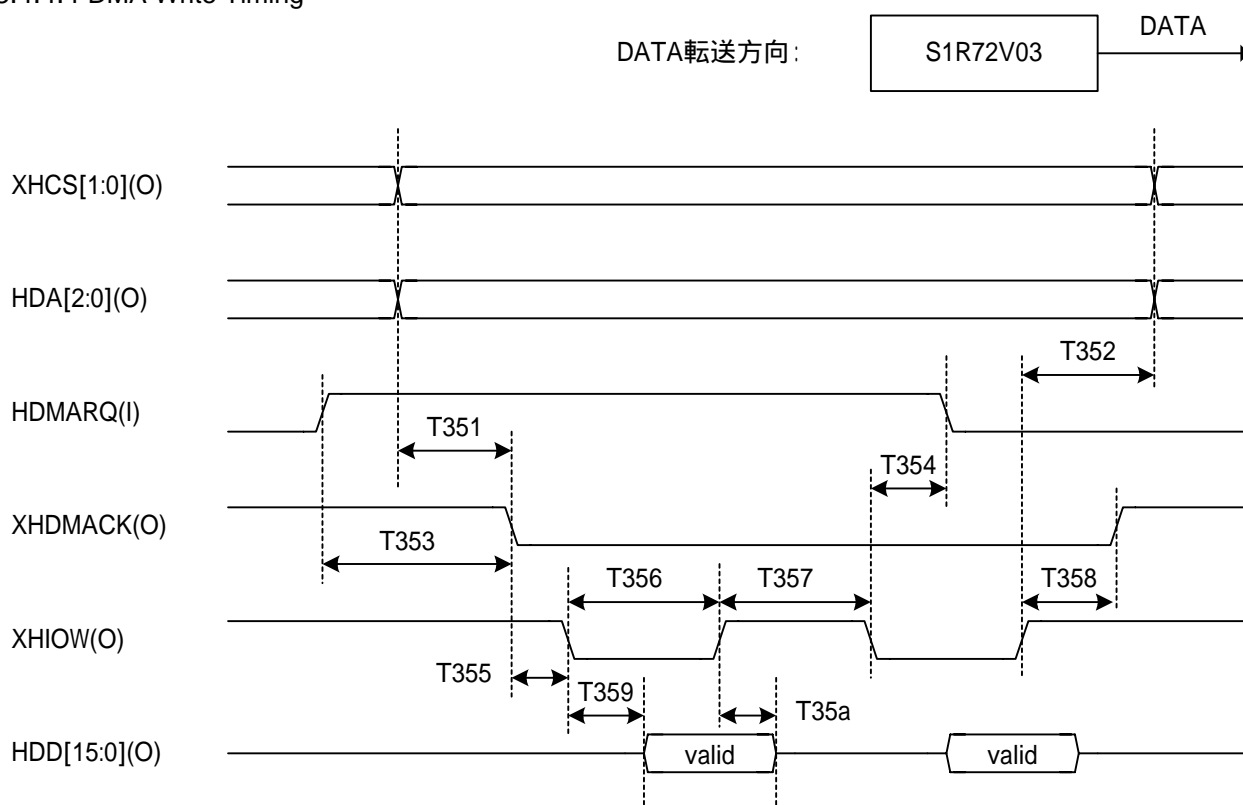


記号	説明	min	typ	max	単位
T341	XHCS、HDA、XHDMACK アドレスセットアップ時間	70	-	-	ns
T342	XHIOR、XHCS、HDA アドレスホールド時間	10	-	-	ns
T343	HDMARQ、XHDMACK XHDMACK応答時間	17	-	-	ns
T344	XHIOR、HDMARQネゲート HDMARQホールド時間	0	-	-	ns
T345	XHDMACK、XHIOR XHDMACKセットアップ時間	0	-	-	ns
T346	XHIOR、XHIOR XHIORアサートパルス幅	-	$(AP+4) * 16.7 - 3$	-	ns
T347	XHIOR、XHIOR 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

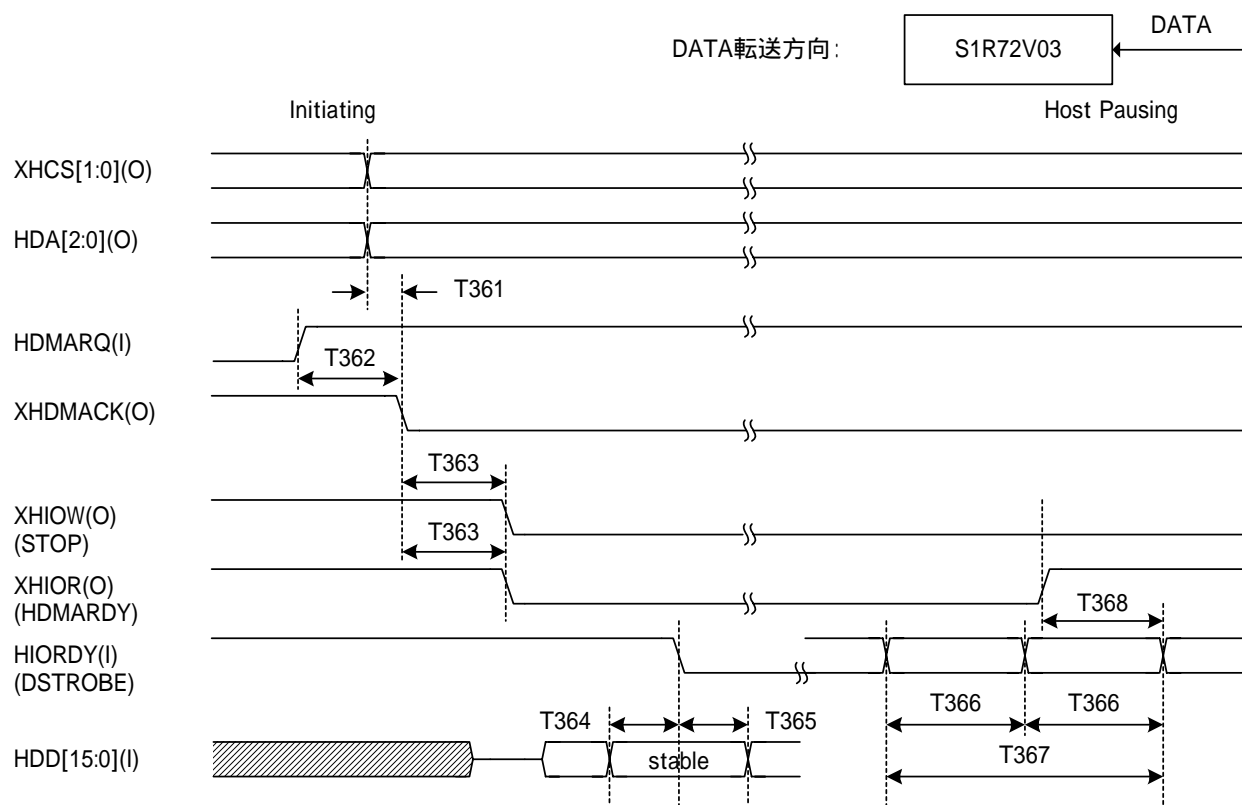
8.4.4.4 DMA Write Timing



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ネゲート 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ネゲートパルス幅	-	$(NP+4) * 16.7 + 3$	-	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 データバスネゲート時間	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

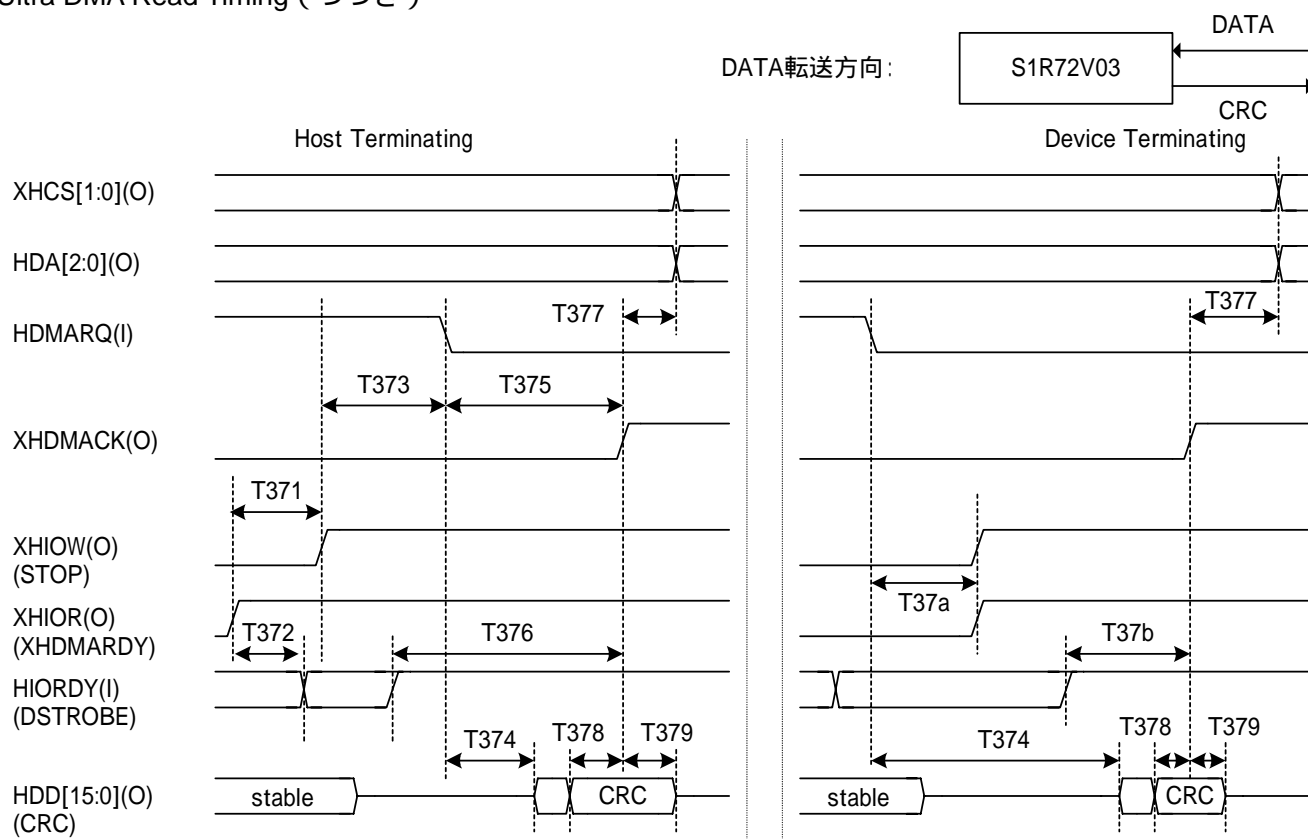
8.4.4.5 Ultra DMA Read Timing



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	HDMAHQ XHDMACK XHDMACK応答時間	0	-	-	ns
T363	XHDMACK XHIOW(W) エンベロープ時間	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

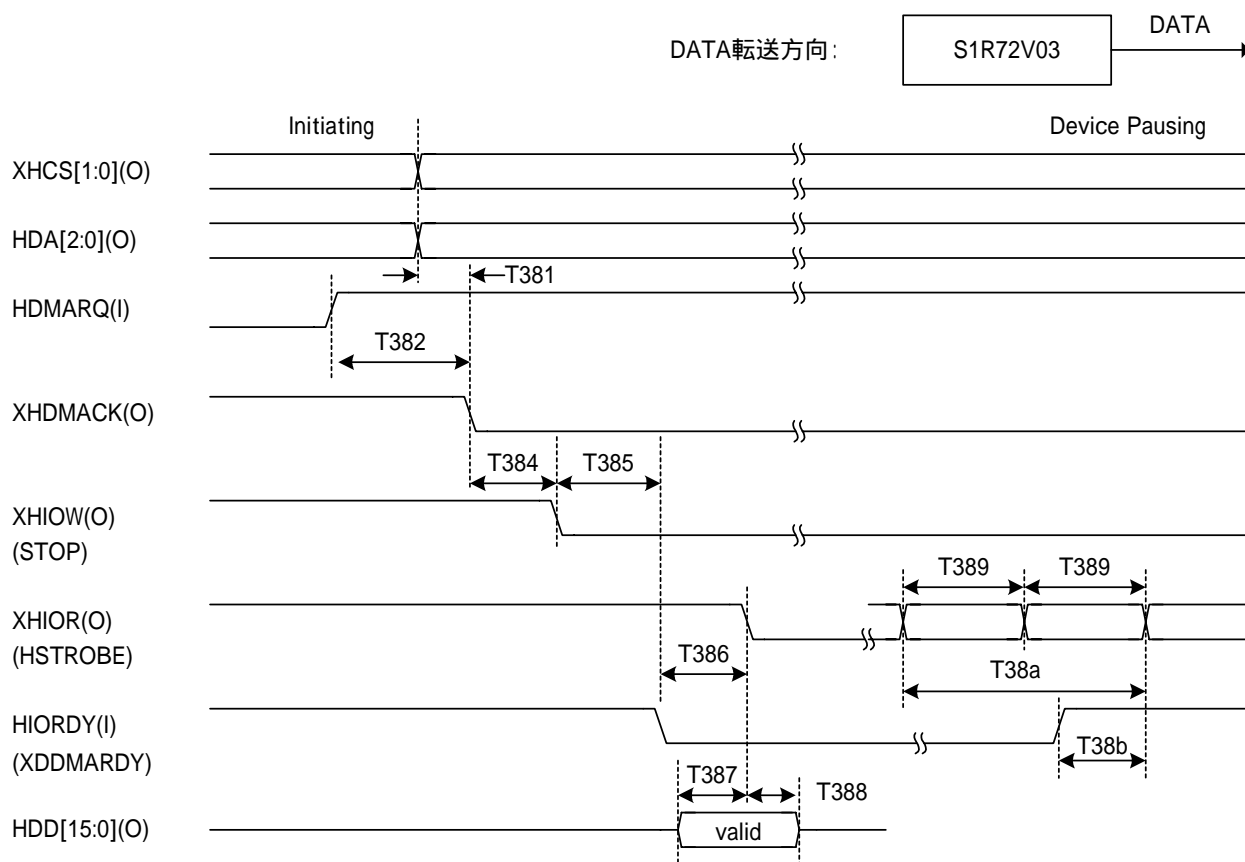
8. 電気的特性

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_{LI}	-	IDE規格 t_{LI}	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

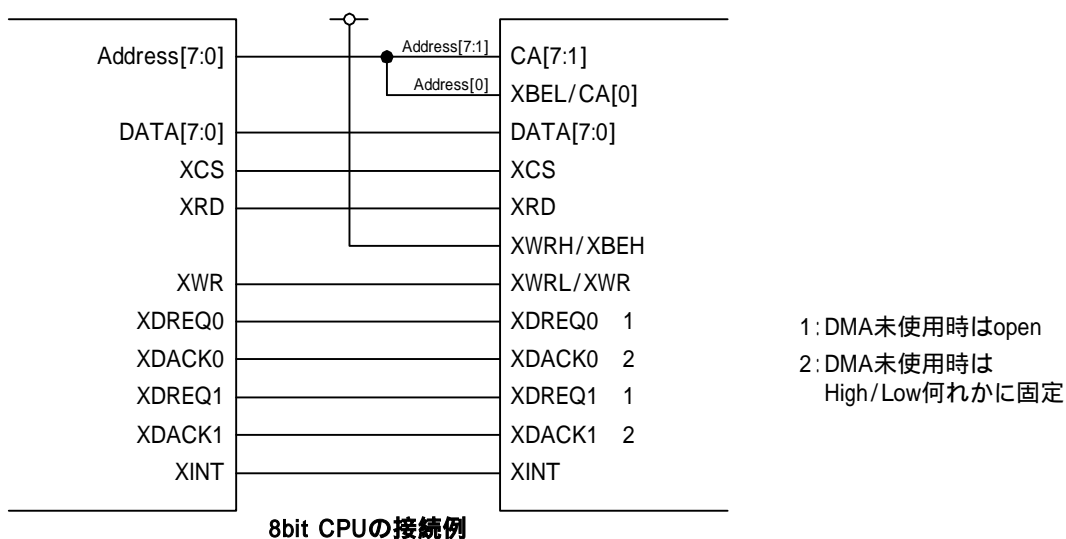
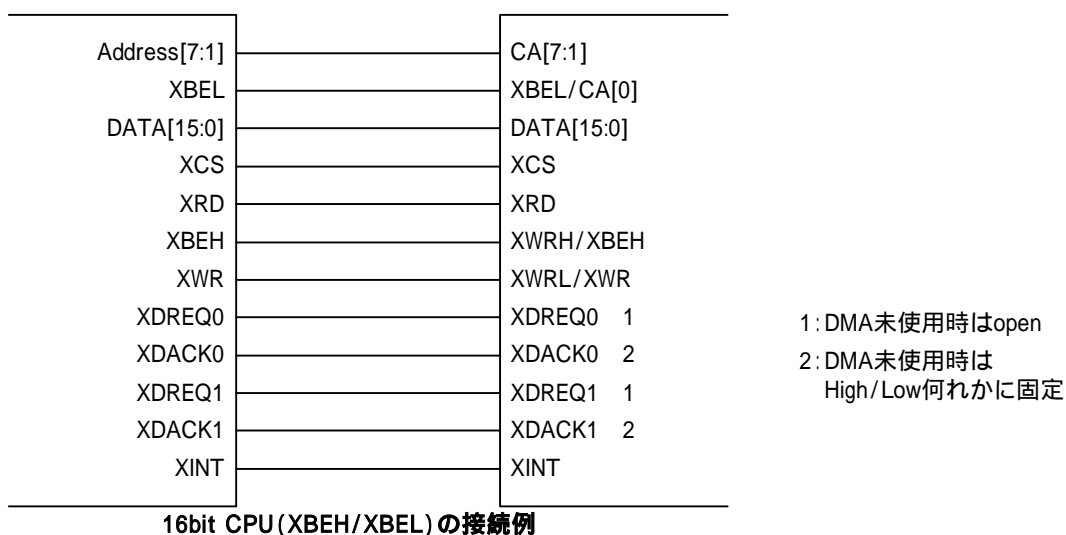
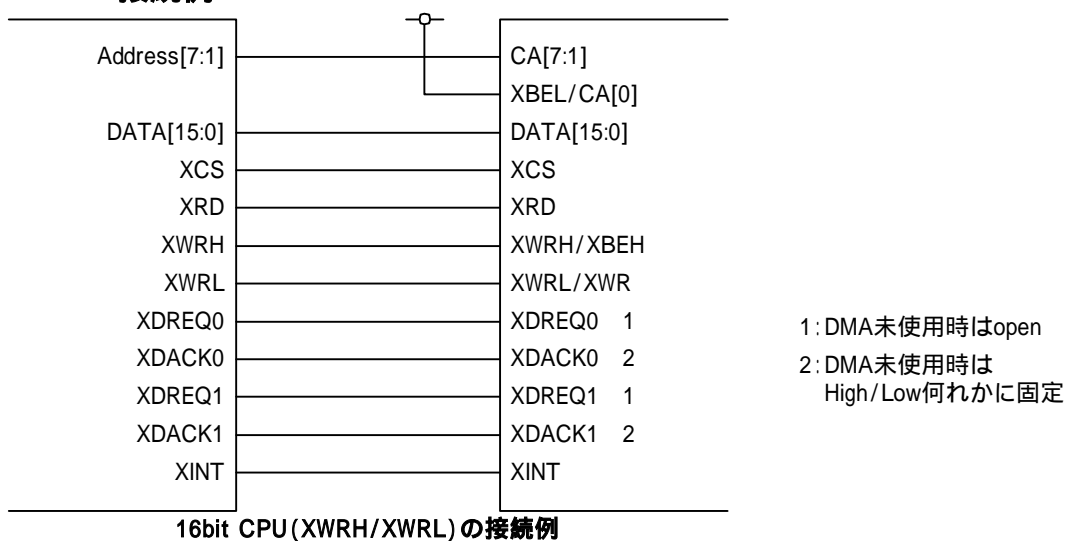
*1:cyc=UltraDMACycle

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

9. 接続例

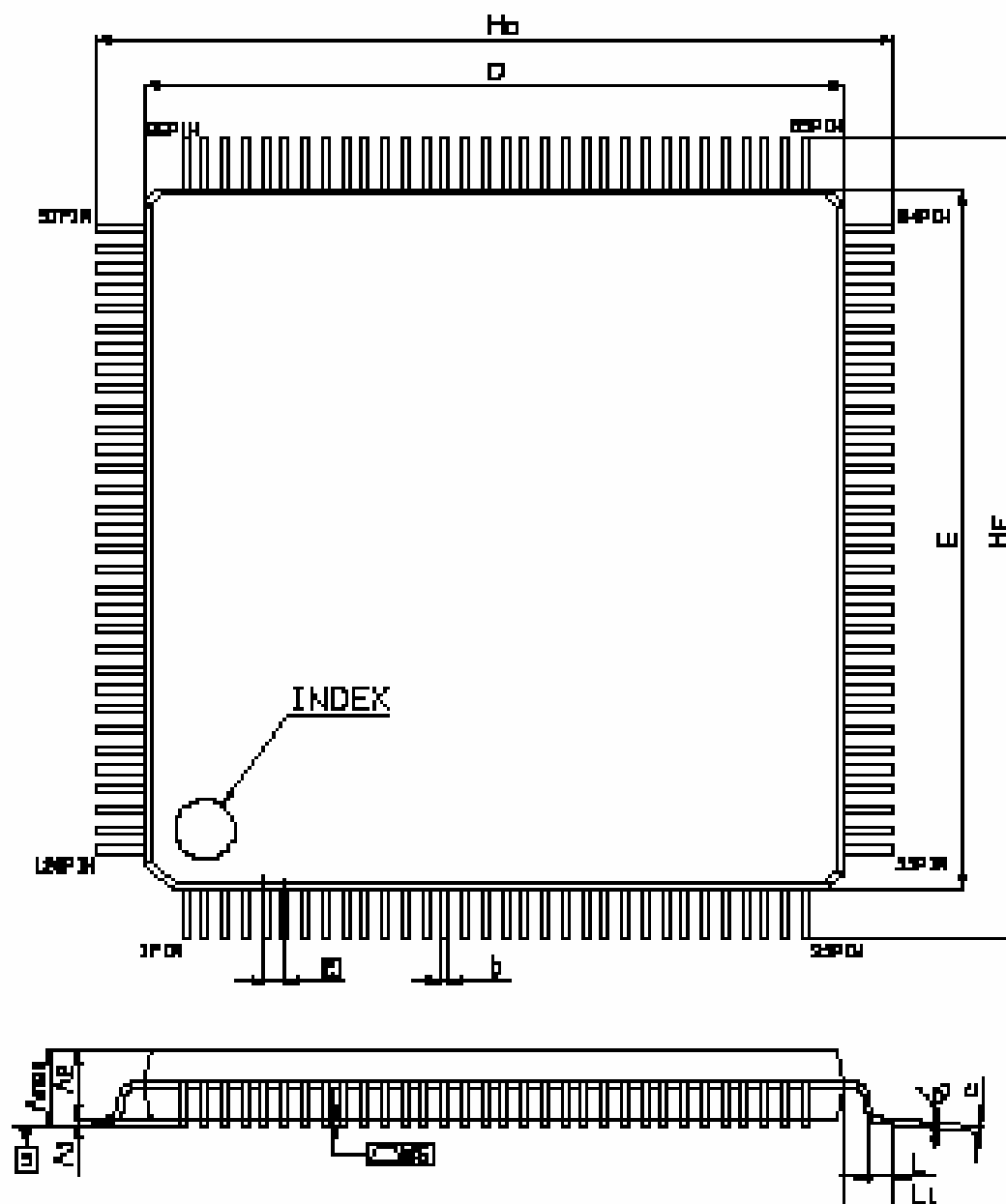
9. 接続例

9.1 CPU I/F 接続例



10. 外形寸法図

10.1 QFP パッケージ



Symbol	Dimension in Millimeters		
	Min	Max	Max
E	-	12	-
D	-	12	-
A_{max}	-	-	1.7
A_1	-	0.1	-
A_2	-	0.3	-
b	-	0.3	-
b	0.13	-	0.33
b	0.125	-	0.3
b	0.1	-	0.3
b	0.3	-	0.75
L	-	1	-
H_0	-	12	-
H_0	-	12	-
γ	-	-	0.125

1 = 1mm

11. Appendix A IDE_Config_1.Swap ビット設定

S1R72H74 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。S1R72H74 では、IDE_Config_1.Swap ビットを使用して、S1R72H74 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

12. Appendix B リトルエンディアンの CPU への接続

S1R72H74 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 S1R72H74 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ください。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、S1R72H74 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 S1R72H74 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、S1R72H74 を動作させる時には、まず以下の手順を行ってください。

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 S1R72H74 においては 0xEF に割り当てられていますが、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。

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。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

改訂履歴表

改訂履歴表

年月日	改訂内容			
	Rev.	頁	種別	内 容
07/05/11	0.90RC	全頁	新規	新規制定

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 : 411118100
2007 年 5 月 作成