
I/O Core Ver. 2.00
仕様書 第6版

平成 28 年 11 月 23 日

目 次

第 1 章	Abstract	7
第 2 章	Pin Assignment	9
第 3 章	Instruction	17
3.1	サポートする命令	17
3.2	サポートしていない命令	17
3.3	命令セットの符合化	18
3.3.1	命令の形式	18
3.3.2	Opecode	18
3.3.3	Function	19
3.3.4	Rt	19
3.3.5	Rs	19
3.4	命令セットの詳細	20
3.4.1	メモリアクセス命令	20
3.4.2	即値命令	25
3.4.3	ALU 命令	29
3.4.4	ジャンプ命令	37
3.4.5	分岐命令	39
3.4.6	System Call / Break Point 命令	43
3.4.7	特権命令	44
3.4.8	FPU 命令	47
第 4 章	Exception	61
4.1	発生する例外	61
4.2	割り込み	62
第 5 章	CPU Control Register	63
5.1	アドレスマップ	63
5.2	ビットマップ	64
第 6 章	FPU	71
6.1	制御レジスタ	71
6.1.1	アドレスマップ	71
6.1.2	ビットマップ	72
6.2	例外	72

第 7 章	Memory Management Unit	75
7.1	MMU の概要	75
7.2	MMU の制御	77
7.3	制御レジスタ	77
7.3.1	アドレスマップ	78
7.3.2	ビットマップ	78
第 8 章	Bus	83
8.1	仕様	83
8.2	バスマスタのマップ	83
8.3	バススレーブのアドレスマップ	84
8.4	バスコントローラ	86
8.4.1	アドレスマップ	86
8.4.2	ビットマップ	87
第 9 章	External Bus	89
9.1	仕様	89
9.1.1	外部バスのチャンネル切り替え	89
9.2	外部バスコントローラ	89
9.2.1	アドレスマップ	89
9.2.2	ビットマップ	90
第 10 章	Timer	93
10.1	概要	93
10.2	制御レジスタ	93
10.2.1	アドレスマップ	93
10.2.2	ビットマップ	94
第 11 章	Clock Generator	97
11.1	概要	97
11.2	制御レジスタ	97
11.2.1	アドレスマップ	97
11.2.2	Clock Divider	99
11.2.3	Clock Gating	99
11.2.4	Reset	100
第 12 章	Universal Asynchronous Receiver/Transmitter	101
12.1	アドレスマップ	101
12.1.1	Receiver Buffer (RB) / Transmitter Holding Register (THR)	101
12.1.2	Interrupt Enable Register (IER)	101
12.1.3	Interrupt Identification Register (IIR)	102
12.1.4	FIFO Control Register (FCR)	103
12.1.5	Line Control Register (LCR)	104
12.1.6	Modem Control Register (MCR)	105
12.1.7	Line Status Register (LSR)	106

12.1.8 Modem Status Register (MSR)	108
12.1.9 Divisor Latches (DL)	108
12.2 動作/使用方法	108
12.2.1 Initialization	109
第 13 章 General Purpose I/O Unit	111
13.1 Outline	111
13.2 Interface	111
13.2.1 Address Format	111
13.2.2 Control Register	111
13.3 Operation	113
第 14 章 Serial Peripheral Interface Unit	115
14.1 Outline	115
14.2 Interface	115
14.2.1 Address Format	115
14.2.2 Control Register	115
14.3 Operation	121
14.3.1 Manual Mode	122
14.3.2 Auto Mode	122
第 15 章 I2C Master Controller	123
15.1 Outline	123
15.2 Interface	123
15.2.1 Address Format	123
15.2.2 Control Register	123
15.3 Operation	126
15.3.1 System Configuration	126
15.3.2 I2C Protocol	127
15.3.3 Arbitration Procudure	128
15.3.4 Clock Stretching	128
第 16 章 PWM Generator	129
16.1 PWM 発生器概要	129
16.2 PWM コントロールレジスタ	130
16.3 PWM 周期制御レジスタ	132
16.4 PWM 反転制御レジスタ	133
16.5 デッドタイムレジスタ	133
第 17 章 PWM Input	137
17.1 PWM 入力器概要	137
17.2 PWMIN コントロールレジスタ	137
17.3 PWMIN HIGH レジスタ	138
17.4 PWMIN LOW レジスタ	138

第 18 章 Pulse Counter	139
18.1 パルスカウンタ概要	139
18.2 レジスタインタフェース	139
18.2.1 パルスカウンタ制御レジスタ	139
18.2.2 コンペアデータレジスタ	140
18.2.3 カウンタレジスタ	141
18.2.4 タイマレジスタ	141
第 19 章 Real Time Clock	143
19.1 Outline	143
19.2 Interface	143
19.2.1 Address Map	143
第 20 章 DMA Controller	151
20.1 レジスタマップ	151
20.1.1 DMA 制御レジスタ	151
20.1.2 DMA 割り込みクリアレジスタ	152
20.1.3 ポート/ソースアドレスレジスタ	152
20.1.4 メモリ/デスティネーションアドレスレジスタ	152
20.1.5 転送レングスレジスタ	153
20.1.6 データバッファレジスタ	153
20.1.7 転送モード制御レジスタ	153
20.1.8 ステータスレジスタ	155
第 21 章 PCI Host Controller	157
21.1 Outline	157
21.2 Bridge Control Register Block	157
21.2.1 PCI Configuration Space	158
21.2.2 Bridge ID Register (0x40)	160
21.2.3 Bridge Control Register (0x44)	161
21.2.4 Bridge status register (0x46)	161
21.2.5 Interrupt Status Register (0x4A)	161
21.2.6 Interrupt Mask Register (0x48)	162
21.2.7 PCI Address Pointer (0x4C)	162
21.2.8 PCI Transfer Counter (0x50)	164
21.2.9 PCI Command Register (0x54)	164
21.2.10 Initiator Dual-port Memory Data Pointer (0x58)	164
第 22 章 On-Chip Emulator	165
22.1 Outline	165
22.2 Operation	165
22.2.1 Single Write	165
22.2.2 Single Read	165

第 23 章 Responsive Link	167
23.1 概要	167
23.2 インタフェース	168
23.3 パケットフォーマット	169
23.3.1 固定長 (64B) のデータパケット	170
23.3.2 固定長 (16B) のイベントパケット	170
23.3.3 優先度による追い越し機構	171
23.4 フレームフォーマット	173
23.5 ルーティング・テーブル	173
23.6 パケットの加減速制御	174
23.7 優先度に従った経路制御	175
23.8 低レベル通信	177
23.8.1 CODEC	177
23.8.2 巡回組織ハミング符号化	178
23.8.3 Bit Stuffing	178
23.8.4 NRZI 符号化	178
23.8.5 セットアップパターン	178
23.8.6 DPLL を用いたビット同期	179
23.8.7 エラーの取扱い	179
23.8.8 通信速度	179
23.9 メモリマップ	180
23.10 レジスタマップ	180
23.10.1 SDRAM モードレジスタ	180
23.10.2 レスポンシブリンク速度設定レジスタ	181
23.10.3 レスポンシブリンク初期化レジスタ	181
23.10.4 レスポンシブリンク割り込みクリアレジスタ	182
23.10.5 デコーダリセット割り込みクリアレジスタ	183
23.10.6 レスポンシブリンク送信停止割り込みクリアレジスタ	184
23.10.7 レスポンシブリンク継続割り込みクリアレジスタ	184
23.10.8 レスポンシブリンク致命的エラー割り込みクリアレジスタ	185
23.10.9 レスポンシブリンクルーティングテーブル割り込みクリアレジスタ	186
23.10.10 レスポンシブリンク SDRAM バスリクエストレジスタ	186
23.10.11 レスポンシブリンク SDRAM バスグラントレジスタ	186
23.10.12 レスポンシブリンクルーティングテーブルバスリクエストレジスタ	187
23.10.13 レスポンシブリンクルーティングテーブルバスグラントレジスタ	188
23.10.14 イベントリンク LRU アドレスレジスタ	189
23.10.15 データリンク LRU アドレスレジスタ	189
23.10.16 レスポンシブリンク用割り込みコントローララインエーブルレジスタ	189
23.10.17 イベントリンク用 SDRAM ループカウンタレジスタ	189
23.10.18 データリンク用 SDRAM ループカウンタレジスタ	190
23.10.19 レスポンシブリンクスイッチモードレジスタ	190
23.10.20 レスポンシブリンク用オフラインレジスタ	190
23.10.21 パラレルモードレジスタ	191
23.10.22 エラーパケットヘッダレジスタ	192

23.10.2エラーヘッダポインタレジスタ	192
23.10.2エラーパケットモードレジスタ	192
23.10.2SDRAM 回復イネーブルレジスタ	192
23.10.2通信コーデック設定レジスタ	193
23.11DPM (Dual Port Memory)	193
23.11.1 Event Output	193
23.11.2 Event Input	196
23.11.3 Data Output	199
23.11.4 Data Input	201
23.12通信方法	203
23.12.1 手順	203
23.12.2 相互通信の際の注意点	204
23.13Responsive Link の割り込みコントローラ	204
23.13.1 レジスタマップ	205

第 24 章 更新履歴	209
-------------	-----

1

Abstract

I/O Core は、ロボット制御の I/O 処理を専用に行うチップとして開発された System-on-Chip である。MIPS 互換の制御用プロセッサと、制御用 I/O とコンピュータ用 I/O (Responsive Link , PCI, GPIO, SPI, I2C, UART, PWM, Encoder 等) を搭載している。

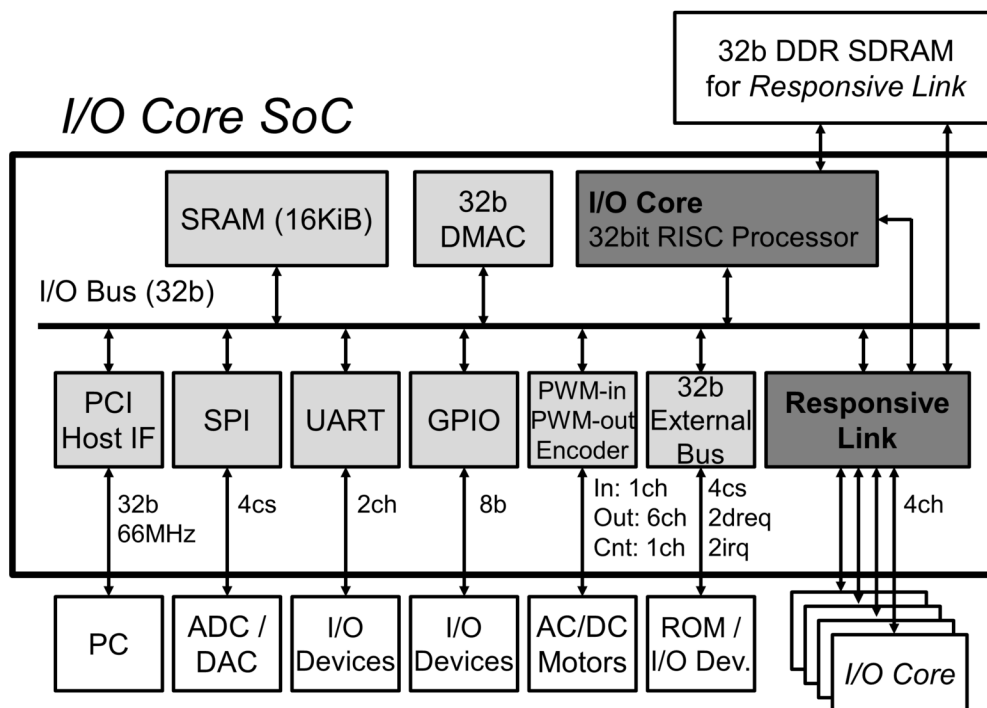


図 1.1: I/O Core のブロック図

以下に I/O Core が持つ I/O を示す。

- *Responsive Link* (4ch)
- PCI Controller (1ch)

- Serial Peripheral Interface ($4cs \times 1ch$)
- UART (2ch)
- I2C Master Controller(1ch)
- Digital Port (8bit)
- PWM Generator (6ch)
- PWM Input (2ch)
- Pulse Counter (2ch)
- Real-Time Clock (1ch)
- Timer (4ch)
- On-Chip Emulator (1ch)
- External Bus I/F (4ch)
- DMA Controller ($4ch \times 2$)
- DDR SDRAM I/F (1ch)
- SRAM (16kB)

2

Pin Assignment

IO コアのピン配置の一覧を以下に示す．

表 2.1: ピンリスト

Pin	Bump	Pin Name	Master Cell	Level	Dir.	Description	Termination	Polarity
0	AC22	PVSS2DGZ.b00	PVSS2DGZ	-	-	VSSIO	-	-
1	AA20	pci.clk	PCI66SDGZ	LVTTL	Input	PCI Clock	-	-
2	AC21	PVDD2DGZ.b00	PVDD2DGZ	3.3V	-	VDDIO	-	-
3	Y19	pci.rst_	PCI66SDGZ	LVTTL	Input	PCI Reset	-	Active Low
4	W18	pci.idsel	PCI66SDGZ	LVTTL	Input	Init Dev Sel	-	-
5	V17	pci.frame_	PCI66SDGZ	LVTTL	In/Out	Transaction Frame	-	Active Low
6	AB20	PVSS2DGZ.b01	PVSS2DGZ	-	-	VSSIO	-	-
7	U17	pci.irdy_	PCI66SDGZ	LVTTL	In/Out	Initiator Ready	-	Active Low
8	AC20	pci.trdy_	PCI66SDGZ	LVTTL	In/Out	Target Ready	-	Active Low
9	AA19	pci.stop_	PCI66SDGZ	LVTTL	In/Out	Stop Output	-	Active Low
10	T16	pci.devsel_	PCI66SDGZ	LVTTL	In/Out	Device Select	-	Active Low
11	Y18	PVDD2DGZ.b01	PVDD2DGZ	3.3V	-	VDDIO	-	-
12	W17	pci.ad31	PCI66SDGZ	LVTTL	In/Out	A/D Bus 31	-	-
13	AB19	pci.ad30	PCI66SDGZ	LVTTL	In/Out	A/D Bus 30	-	-
14	AC19	pci.ad29	PCI66SDGZ	LVTTL	In/Out	A/D Bus 29	-	-
15	U16	pci.ad28	PCI66SDGZ	LVTTL	In/Out	A/D Bus 28	-	-
16	AA18	PVSS2DGZ.b02	PVSS2DGZ	-	-	VSSIO	-	-
17	Y17	pci.ad27	PCI66SDGZ	LVTTL	In/Out	A/D Bus 27	-	-
18	V16	pci.ad26	PCI66SDGZ	LVTTL	In/Out	A/D Bus 26	-	-
19	AB18	pci.ad25	PCI66SDGZ	LVTTL	In/Out	A/D Bus 25	-	-
20	W16	pci.ad24	PCI66SDGZ	LVTTL	In/Out	A/D Bus 24	-	-
21	AC18	PVDD2DGZ.b02	PVDD2DGZ	3.3V	-	VDDIO	-	-
22	T15	pci.ad23	PCI66SDGZ	LVTTL	In/Out	A/D Bus 23	-	-
23	AA17	pci.ad22	PCI66SDGZ	LVTTL	In/Out	A/D Bus 22	-	-
24	U15	pci.ad21	PCI66SDGZ	LVTTL	In/Out	A/D Bus 21	-	-
25	Y16	pci.ad20	PCI66SDGZ	LVTTL	In/Out	A/D Bus 20	-	-
26	AB17	PVSS2DGZ.b03	PVSS2DGZ	-	-	VSSIO	-	-
27	AC17	pci.ad19	PCI66SDGZ	LVTTL	In/Out	A/D Bus 19	-	-
28	V15	pci.ad18	PCI66SDGZ	LVTTL	In/Out	A/D Bus 18	-	-
29	AA16	PVSS1DGZ.b00	PVSS1DGZ	-	-	VSS	-	-

Pin	Bump	Pin Name	Master Cell	Level	Dir.	Description	Termination	Polarity
30	W15	pci.ad17	PCI66SDGZ	LVTTL	In/Out	A/D Bus 17	-	-
31	T14	pci.ad16	PCI66SDGZ	LVTTL	In/Out	A/D Bus 16	-	-
32	AB16	PVDD1DGZ_b00	PVDD1DGZ	1.0V	-	VDD	-	-
33	AC16	pci.ad15	PCI66SDGZ	LVTTL	In/Out	A/D Bus 15	-	-
34	Y15	pci.ad14	PCI66SDGZ	LVTTL	In/Out	A/D Bus 14	-	-
35	AA15	PVSS1DGZ_b01	PVSS1DGZ	-	-	VSS	-	-
36	U14	pci.ad13	PCI66SDGZ	LVTTL	In/Out	A/D Bus 13	-	-
37	V14	pci.ad12	PCI66SDGZ	LVTTL	In/Out	A/D Bus 12	-	-
38	AB15	PVDD1DGZ_b01	PVDD1DGZ	1.0V	-	VDD	-	-
39	AC15	pci.ad11	PCI66SDGZ	LVTTL	In/Out	A/D Bus 11	-	-
40	T13	pci.ad10	PCI66SDGZ	LVTTL	In/Out	A/D Bus 10	-	-
41	W14	PVSS1DGZ_b02	PVSS1DGZ	-	-	VSS	-	-
42	Y14	pci.ad9	PCI66SDGZ	LVTTL	In/Out	A/D Bus 9	-	-
43	AA14	pci.ad8	PCI66SDGZ	LVTTL	In/Out	A/D Bus 8	-	-
44	AB14	PVDD1DGZ_b02	PVDD1DGZ	1.0V	-	VDD	-	-
45	AC14	pci.ad7	PCI66SDGZ	LVTTL	In/Out	A/D Bus 7	-	-
46	U13	pci.ad6	PCI66SDGZ	LVTTL	In/Out	A/D Bus 6	-	-
47	V13	PCI66SDGZ_b03	PVSS1DGZ	-	-	VSS	-	-
48	W13	pci.ad5	PCI66SDGZ	LVTTL	In/Out	A/D Bus 5	-	-
49	Y13	pci.ad4	PCI66SDGZ	LVTTL	In/Out	A/D Bus 4	-	-
50	AA13	PVDD2DGZ_b03	PVDD2DGZ	3.3V	-	VDDIO	-	-
51	AB13	pci.ad3	PCI66SDGZ	LVTTL	In/Out	A/D Bus 3	-	-
52	AC13	pci.ad2	PCI66SDGZ	LVTTL	In/Out	A/D Bus 2	-	-
53	V12	PVSS2DGZ_b04	PVSS2DGZ	-	-	VSSIO	-	-
54	W12	pci.ad1	PCI66SDGZ	LVTTL	In/Out	A/D Bus 1	-	-
55	Y12	PVSS1DGZ_b04	PVSS1DGZ	-	-	VSS	-	-
56	AA12	pci.ad0	PCI66SDGZ	LVTTL	In/Out	A/D Bus 0	-	-
57	AB12	PVDD1DGZ_b03	PVDD1DGZ	1.0V	-	VDD	-	-
58	AC12	pci.cbe3_	PCI66SDGZ	LVTTL	In/Out	C/B En Bus 3	-	Active Low
59	U12	pci.cbe2_	PCI66SDGZ	LVTTL	In/Out	C/B En Bus 2	-	Active Low
60	W11	PVSS1DGZ_b05	PVSS1DGZ	-	-	VSS	-	-
61	Y11	pci.cbe1_	PCI66SDGZ	LVTTL	In/Out	C/B En Bus 1	-	Active Low
62	AA11	pci.cbe0_	PCI66SDGZ	LVTTL	In/Out	C/B En Bus 0	-	Active Low
63	AB11	PVSS1DGZ_b06	PVSS1DGZ	-	-	VSS	-	-
64	AC11	pci.par	PCI66SDGZ	LVTTL	In/Out	Parity	-	-
65	V11	PVDD2DGZ_b04	PVDD2DGZ	3.3V	-	VDDIO	-	-
66	U11	pci.perr_	PCI66SDGZ	LVTTL	In/Out	Parity Error	-	Active Low
67	T11	pci.serr	PCI66SDGZ	LVTTL	In/Out	System Error	-	Active Low
68	AA10	PVSS2DGZ_b05	PVSS2DGZ	-	-	VSSIO	-	-
69	AB10	pci.req_	PCI66SDGZ	LVTTL	Input	Master Request	-	Active Low
70	AC10	pci.gnt_	PCI66SDGZ	LVTTL	Output	Master Grant	-	Active Low
71	Y10	PVDD1DGZ_b04	PVDD1DGZ	1.0V	-	VDD	-	-
72	W10	pci.inta_	PCI66SDGZ	LVTTL	Input	Interrupt A	-	Active Low
73	V10	PVSS1DGZ_b07	PVSS1DGZ	-	-	VSS	-	-
74	AA9	pci.intb_	PCI66SDGZ	LVTTL	Input	Interrupt B	-	Active Low
75	AB9	PVSS1DGZ_b08	PVSS1DGZ	-	-	VSS	-	-
76	AC9	pci.intc_	PCI66SDGZ	LVTTL	Input	Interrupt C	-	Active Low
77	Y9	PVDD1DGZ_b05	PVDD1DGZ	1.0V	-	VDD	-	-
78	W9	pci.intd_	PCI66SDGZ	LVTTL	Input	Interrupt D	-	Active Low
79	V9	link_sdram.oe_	pnl_sstl2classi	SSTL2	Output	Output Enable	-	Active Low
80	U10	link_sdram.dir	pnl_sstl2classi	SSTL2	Output	Direction	-	-
81	AC8	link_sdram.addr04	pnl_sstl2classi	SSTL2	Output	SDRAM Address 4	-	-
82	AB8	link_sdram.addr03	pnl_sstl2classi	SSTL2	Output	SDRAM Address 3	-	-
83	AA8	link_sdram.addr05	pnl_sstl2classi	SSTL2	Output	SDRAM Address 5	-	-
84	Y8	pnl_sstl_gcs_b00	pnl_sstl_gcs	-	-	VSS	-	-
85	AB7	link_sdram.addr02	pnl_sstl2classi	SSTL2	Output	SDRAM Address 2	-	-
86	AC7	pnl_sstl_vc_b00	pnl_sstl_vc	1.0V	-	VDD	-	-
87	W8	link_sdram.addr06	pnl_sstl2classi	SSTL2	Output	SDRAM Address 6	-	-
88	AA7	pnl_sstl_go_b00	pnl_sstl_go	-	-	VSSIO	-	-
89	U9	link_sdram.addr01	pnl_sstl2classi	SSTL2	Output	SDRAM Address 1	-	-
90	Y7	pnl_sstl_vq_b00	pnl_sstl_vq	2.5V	-	VDDIO	-	-
91	W7	link_sdram.addr07	pnl_sstl2classi	SSTL2	Output	SDRAM Address 7	-	-
92	AC6	pnl_sstl_gcs_b01	pnl_sstl_gcs	-	-	VSS	-	-
93	AB6	link_sdram.addr00	pnl_sstl2classi	SSTL2	Output	SDRAM Address 0	-	-

Pin	Bump	Pin Name	Master Cell	Level	Dir.	Description	Termination	Polarity
94	V8	link_sdram_addr08	pnl_sstl2classi	SSTL2	Output	SDRAM Address 8	-	-
95	AA6	link_sdram_addr10	pnl_sstl2classi	SSTL2	Output	SDRAM Address 10	-	-
96	V7	pnl_sstl_vc.b01	pnl_sstl_vc	1.0V	-	VDD	-	-
97	AC5	link_sdram_addr09	pnl_sstl2classi	SSTL2	Output	SDRAM Address 9	-	-
98	Y6	link_sdram_bank1	pnl_sstl2classi	SSTL2	Output	SDRAM Bank Address 1	-	-
99	AB5	pnl_sstl_go.b01	pnl_sstl_go	-	-	VSSIO	-	-
100	T10	link_sdram_addr11	pnl_sstl2classi	SSTL2	Output	SDRAM Address 11	-	-
101	W6	pnl_sstl_vq.b01	pnl_sstl_vq	2.5V	-	VDDIO	-	-
102	AA5	link_sdram_bank0	pnl_sstl2classi	SSTL2	Output	SDRAM Bank Address 0	-	-
103	AC4	link_sdram_addr12	pnl_sstl2classi	SSTL2	Output	SDRAM Address 12	-	-
104	U8	link_sdram_cs1	pnl_sstl2classi	SSTL2	Output	SDRAM CS 1	-	Active Low
105	AB4	link_sdram_cs0	pnl_sstl2classi	SSTL2	Output	SDRAM CS0	-	Active Low
106	V6	link_sdram_cke	pnl_sstl2classi	SSTL2	Output	SDRAM Clock Enable	-	-
107	Y5	link_sdram_ras	pnl_sstl2classi	SSTL2	Output	SDRAM RAS	-	Active Low
108	AC3	pnl_sstl_go.b02	pnl_sstl_go	-	-	VSSIO	-	-
109	U7	link_sdram_cas	pnl_sstl2classi	SSTL2	Output	SDRAM CAS	-	Active Low
110	AA4	pnl_sstl_vq.b02	pnl_sstl_vq	2.5V	-	VDDIO	-	-
111	U6	link_sdram_we	pnl_sstl2classi	SSTL2	Output	SDRAM Write Enable	-	Active Low
112	AB3	pnl_sstl_gcs.b02	pnl_sstl_gcs	-	-	VSS	-	-
113	AC2	link_sdram_dqm3	pnl_sstl2classi	SSTL2	Output	SDRAM DQM 3	-	-
114	W5	link_sdram_dqm1	pnl_sstl2classi	SSTL2	Output	SDRAM DQM 2	-	-
115	Y4	link_sdram_dqm2	pnl_sstl2classi	SSTL2	Output	SDRAM DQM 1	-	-
116	T9	link_sdram_dqm0	pnl_sstl2classi	SSTL2	Output	SDRAM DQM 0	-	-
117	AB2	link_sdram_clk	pnl_sstl2classi	SSTL2	Output	SDRAM Clock	-	-
118	AA3	pnl_sstl_go.r00	pnl_sstl_go	-	-	VSSIO	-	-
119	AB1	link_sdram_clk_	pnl_sstl2classi	SSTL2	Output	SDRAM Clock X	-	-
120	T8	sstl_vref	pnl_sstl_vref	-	-	SDRAM VREF	-	-
121	V5	link_sdram_dqs3	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQS 3	-	-
122	AA2	pnl_sstl_vp.r00	pnl_sstl_vp	2.5V	-	VDDIO	-	-
123	AA1	link_sdram_dq31	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQ 31	-	-
124	Y3	pnl_sstl_vc.r00	pnl_sstl_vc	1.0V	-	VDD	-	-
125	W4	link_sdram_dq30	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQ 30	-	-
126	Y2	pnl_sstl_gcs.r00	pnl_sstl_gcs	-	-	VSS	-	-
127	T7	link_sdram_dq29	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQ 29	-	-
128	U5	link_sdram_dq28	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQ 28	-	-
129	Y1	pnl_sstl_gcs.r01	pnl_sstl_gcs	-	-	VSS	-	-
130	T6	link_sdram_dq27	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQ 27	-	-
131	W3	link_sdram_dq26	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQ 26	-	-
132	V4	pnl_sstl_vc.r01	pnl_sstl_vc	1.0V	-	VDD	-	-
133	W2	link_sdram_dq25	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQ 25	-	-
134	R8	link_sdram_dq24	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQ 24	-	-
135	W1	pnl_sstl_go.r01	pnl_sstl_go	-	-	VSSIO	-	-
136	T5	link_sdram_dqs2	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQS 2	-	-
137	V3	pnl_sstl_vq.r01	pnl_sstl_vq	2.5V	-	VDDIO	-	-
138	U4	link_sdram_dq23	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQ 23	-	-
139	R6	link_sdram_dq22	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQ 22	-	-
140	V2	pnl_sstl_gcs.r02	pnl_sstl_gcs	-	-	VSS	-	-
141	V1	link_sdram_dq21	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQ 21	-	-
142	R7	link_sdram_dq20	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQ 20	-	-
143	U3	pnl_sstl_gcs.r03	pnl_sstl_gcs	-	-	VSS	-	-
144	R5	link_sdram_dq19	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQ 19	-	-
145	T4	link_sdram_dq18	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQ 18	-	-
146	U2	pnl_sstl_vc.r02	pnl_sstl_vc	1.0V	-	VDD	-	-
147	U1	link_sdram_dq17	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQ 17	-	-
148	P8	link_sdram_dq16	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQ 16	-	-
149	P7	pnl_sstl_go.r02	pnl_sstl_go	-	-	VSSIO	-	-
150	P6	link_sdram_dqs1	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQS 1	-	-
151	T3	pnl_sstl_vq.r02	pnl_sstl_vq	2.5V	-	VDDIO	-	-
152	T2	link_sdram_dq15	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQ 15	-	-
153	T1	link_sdram_dq14	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQ 14	-	-
154	R4	pnl_sstl_gcs.r04	pnl_sstl_gcs	-	-	VSS	-	-
155	N8	link_sdram_dq13	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQ 13	-	-
156	P5	link_sdram_dq12	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQ 12	-	-
157	R3	pnl_sstl_vc.r03	pnl_sstl_vc	1.0V	-	VDD	-	-

Pin	Bump	Pin Name	Master Cell	Level	Dir.	Description	Termination	Polarity
158	R2	link_sdram_dq11	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQ 11	-	-
159	R1	link_sdram_dq10	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQ 10	-	-
160	N7	pnl_sstl_gcs_r05	pnl_sstl_gcs	-	-	VSS	-	-
161	N6	link_sdram_dq9	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQ 9	-	-
162	P4	link_sdram_dq8	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQ 8	-	-
163	P3	pnl_sstl_go_r03	pnl_sstl_go	-	-	VSSIO	-	-
164	P2	link_sdram_dqs0	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQS 0	-	-
165	P1	pnl_sstl_vq_r03	pnl_sstl_vq	2.5V	-	VDDIO	-	-
166	M7	link_sdram_dq7	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQ 7	-	-
167	N5	link_sdram_dq6	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQ 6	-	-
168	N4	pnl_sstl_vc_r04	pnl_sstl_vc	1.0V	-	VDD	-	-
169	N3	link_sdram_dq5	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQ 5	-	-
170	N2	link_sdram_dq4	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQ 4	-	-
171	N1	pnl_sstl_gcs_r06	pnl_sstl_gcs	-	-	VSS	-	-
172	M6	link_sdram_dq3	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQ 3	-	-
173	M5	link_sdram_dq2	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQ 2	-	-
174	M4	pnl_sstl_vq_r04	pnl_sstl_vq	2.5V	-	VDDIO	-	-
175	M3	link_sdram_dq1	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQ 1	-	-
176	M2	link_sdram_dq0	pnl_sstl2classi	SSTL2	In/Out	SDRAM DQ 0	-	-
177	M1	pnl_sstl_go_r04	pnl_sstl_go	-	-	VSSIO	-	-
178	L4	lvds_vref	pnl_vref_lvds	-	-	LVDS VREF	-	-
179	L3	pnl_vc_lvds_r00	pnl_vc_lvds	3.3V	-	VDD	-	-
180	L5	link_data_s_out1_p	pnl_lvds85_out_gcs	LVDS	Output	Data Link OutP	-	-
181	L2	pnl_lvds85_out_gcs_r00	pnl_lvds85_out_gcs	-	-	VSS	-	-
182	L6	link_data_s_out1_n	pnl_lvds85_out_gcs	LVDS	Output	Data Link OutN	-	-
183	L1	pnl_go_lvds_r00	pnl_go_lvds	-	-	VSSIO	-	-
184	K5	link_data_s_out2_p	pnl_lvds85_out_vop	LVDS	Output	Data Link OutP	-	-
185	K4	pnl_lvds85_out_vop_r00	pnl_lvds85_out_vop	2.5V	-	VDDIO	-	-
186	K6	link_data_s_out2_n	pnl_lvds85_out_vop	LVDS	Output	Data Link OutN	-	-
187	K3	pnl_gcs_lvds_r00	pnl_gcs_lvds	-	-	VSS	-	-
188	K1	link_data_s.in1_p	pnl_lvds85_se.in	LVDS	Input	Data Link In P	-	-
189	K2	link_data_s.in1_n	pnl_lvds85_se.in	LVDS	Input	Data Link In N	-	-
190	J3	link_data_s_out3_p	pnl_lvds85_out_vc	LVDS	Output	Data Link OutP	-	-
191	J2	pnl_lvds85_out_vc_r00	pnl_lvds85_out_vc	3.3V	-	VDD	-	-
192	J4	link_data_s_out3_n	pnl_lvds85_out_vc	LVDS	Output	Data Link OutN	-	-
193	J1	pnl_gcs_lvds_r01	pnl_gcs_lvds	-	-	VSS	-	-
194	H3	link_data_s_out4_p	pnl_lvds85_out_go	LVDS	Output	Data Link OutP	-	-
195	H2	pnl_lvds85_out_go_r00	pnl_lvds85_out_go	-	-	VSSIO	-	-
196	H4	link_data_s_out4_n	pnl_lvds85_out_go	LVDS	Output	Data Link OutN	-	-
197	H1	pnl_vop_lvds_r00	pnl_vop_lvds	2.5V	-	VDDIO	-	-
198	J5	link_data_s.in2_p	pnl_lvds85_se.in	LVDS	Input	Data Link In P	-	-
199	J6	link_data_s.in2_n	pnl_lvds85_se.in	LVDS	Input	Data Link In N	-	-
200	H5	pnl_gcs_lvds_r02	pnl_gcs_lvds	-	-	VSS	-	-
201	G2	link_data_s.in3_p	pnl_lvds85_se.in	LVDS	Input	Data Link In P	-	-
202	G1	link_data_s.in3_n	pnl_lvds85_se.in	LVDS	Input	Data Link In N	-	-
203	G3	pnl_vc_lvds_r01	pnl_vc_lvds	3.3V	-	VDD	-	-
204	G4	link_data_s.in4_p	pnl_lvds85_se.in	LVDS	Input	Data Link In P	-	-
205	G5	link_data_s.in4_n	pnl_lvds85_se.in	LVDS	Input	Data Link In N	-	-
206	H6	pnl_gcs_lvds_r03	pnl_gcs_lvds	-	-	VSS	-	-
207	F1	link_event_s.in4_p	pnl_lvds85_se.in	LVDS	Input	Event Link In P	-	-
208	F2	link_event_s.in4_n	pnl_lvds85_se.in	LVDS	Input	Event Link In N	-	-
209	F3	pnl_go_lvds_r01	pnl_go_lvds	-	-	VSSIO	-	-
210	G6	link_event_s.out4_p	pnl_lvds85_out_vop	LVDS	Output	Event Link Out P	-	-
211	F4	pnl_lvds85_out_vop_r01	pnl_lvds85_out_vop	2.5V	-	VDDIO	-	-
212	F6	link_event_s.out4_n	pnl_lvds85_out_vop	LVDS	Output	Event Link Out N	-	-
213	E1	link_event_s.in3_p	pnl_lvds85_se.in	LVDS	Input	Event Link In P	-	-
214	E2	link_event_s.in3_n	pnl_lvds85_se.in	LVDS	Input	Event Link In N	-	-
215	E3	pnl_gcs_lvds_r04	pnl_gcs_lvds	-	-	VSS	-	-
216	F5	link_event_s.in2_p	pnl_lvds85_se.in	LVDS	Input	Event Link In P	-	-
217	E5	link_event_s.in2_n	pnl_lvds85_se.in	LVDS	Input	Event Link In N	-	-
218	D3	link_event_s.out3_p	pnl_lvds85_out_vc	LVDS	Output	Event Link Out P	-	-
219	D1	pnl_lvds85_out_vc_r01	pnl_lvds85_out_vc	3.3V	-	VDD	-	-
220	D2	link_event_s.out3_n	pnl_lvds85_out_vc	LVDS	Output	Event Link Out N	-	-
221	E4	pnl_gcs_lvds_r05	pnl_gcs_lvds	-	-	VSS	-	-

Pin	Bump	Pin Name	Master Cell	Level	Dir.	Description	Termination	Polarity
222	E6	link_event_s_out2_p	pnl_lvds85_out_go	LVDS	Output	Event Link Out P	-	-
223	D4	pnl_lvds85_out_go_r01	pnl_lvds85_out_go	-	-	VSSIO	-	-
224	D6	link_event_s_out2_n	pnl_lvds85_out_go	LVDS	Output	Event Link Out N	-	-
225	C1	pnl_vop_lvds_r01	pnl_vop_lvds	2.5V	-	VDDIO	-	-
226	C2	link_event_s_in1_p	pnl_lvds85_se_in	LVDS	Input	Event Link In P	-	-
227	C3	link_event_s_in1_n	pnl_lvds85_se_in	LVDS	Input	Event Link In N	-	-
228	D5	pnl_vc_lvds_r02	pnl_vc_lvds	3.3V	-	VDD	-	-
229	C5	link_event_s_out1_p	pnl_lvds85_out_gcs	LVDS	Output	Event Link Out P	-	-
230	B1	pnl_lvds85_out_gcs_r01	pnl_lvds85_out_gcs	-	-	VSS	-	-
231	C4	link_event_s_out1_n	pnl_lvds85_out_gcs	LVDS	Output	Event Link Out N	-	-
232	B2	uart_stx_pad1	pnl_tf04it0nn2	LVTTL	Output	UART CH1 TxD	-	-
233	L7	uart_srx_pad1	pnl_it2nn2	LVTTL	Output	UART CH1 RxD	-	-
234	L8	uart_dtr_pad0	pnl_tf04it0nn2	LVTTL	Input	UART CH0 DSR	-	-
235	B3	uart_rts_pad0	pnl_tf04it0nn2	LVTTL	Output	UART CH0 RTS	-	-
236	A2	pnl_vc_t00	pnl_vc	1.0V	-	VDD	-	-
237	K7	uart_stx_pad0	pnl_tf04it0nn2	LVTTL	Output	UART CH0 TxD	-	-
238	K8	uart_dcd_pad0	pnl_it2nn2	LVTTL	Input	UART CH0T DCD	-	-
239	B4	pnl_gcs_t00	pnl_gcs	-	-	VSS	-	-
240	J7	uart_ri_pad0	pnl_it2nn2	LVTTL	Input	UART CH0 RI	-	-
241	H7	uart_dsr_pad0	pnl_it2nn2	LVTTL	Input	UART CH0 DSR	-	-
242	A3	uart_srx_pad0	pnl_it2nn2	LVTTL	Input	UART CH0 RxD	-	-
243	C6	uart_cts_pad0	pnl_it2nn2	LVTTL	Input	UART CH0 CTS	-	-
244	B5	pnl_go_t00	pnl_go	-	-	VSSIO	-	-
245	D7	spi_mosi	pnl_tf04it0nn2	LVTTL	-	SPI MOSI	Pull-Down	-
246	E7	spi_miso	pnl_it2nn2	LVTTL	-	SPI MISO	Pull-Down	-
247	A4	pnl_vop_t00	pnl_vop	3.3V	-	VDDIO	-	-
248	F7	spi_sck	pnl_tf04it0nn2	LVTTL	-	SPI Clock	Pull-Down	-
249	C7	pnl_gcs_t01	pnl_gcs	-	-	VSS	-	-
250	B6	spi_ss0_	pnl_tf04it0nn2	LVTTL	-	SPI SS 0	Pull-Up	Active Low
251	G7	spi_ss1_	pnl_tf04it0nn2	LVTTL	-	SPI SS 1	Pull-Up	Active Low
252	A5	pnl_vc_t01	pnl_vc	1.0V	-	VDD	-	-
253	J8	spi_ss2_	pnl_tf04it0nn2	LVTTL	-	SPI SS 2	Pull-Up	Active Low
254	H8	spi_ss3_	pnl_tf04it0nn2	LVTTL	-	SPI SS 2	Pull-Up	Active Low
255	G8	pnl_go_t01	pnl_go	-	-	VSSIO	-	-
256	D8	gpio_data7	pnl_tf04it0nn2	LVTTL	-	GPIO Data 7	-	-
257	A6	gpio_data6	pnl_tf04it0nn2	LVTTL	-	GPIO Data 6	-	-
258	B7	pnl_vop_t01	pnl_vop	3.3V	-	VDDIO	-	-
259	E8	gpio_data5	pnl_tf04it0nn2	LVTTL	-	GPIO Data 5	-	-
260	C8	gpio_data4	pnl_tf04it0nn2	LVTTL	-	GPIO Data 4	-	-
261	F8	pnl_gcs_t02	pnl_gcs	-	-	VSS	-	-
262	H9	gpio_data3	pnl_tf04it0nn2	LVTTL	-	GPIO Data 3	-	-
263	A7	gpio_data2	pnl_tf04it0nn2	LVTTL	-	GPIO Data 2	-	-
264	G9	gpio_data1	pnl_tf04it0nn2	LVTTL	-	GPIO Data 1	-	-
265	B8	gpio_data0	pnl_tf04it0nn2	LVTTL	-	GPIO Data 0	-	-
266	E9	pnl_gcs_t03	pnl_gcs	-	-	VSS	-	-
267	D9	i2c_scl	pnl_tf04it0nn2	LVTTL	In/Out	I2C SCL	Pull-Up	-
268	C9	i2c_sda	pnl_tf04it0nn2	LVTTL	In/Out	I2C SDA	Pull-Up	-
269	A8	pnl_vc_t02	pnl_vc	1.0V	-	VDD	-	-
270	F9	ext_bus_data31	pnl_tf12it0nn2	LVTTL	-	Ext Data 31	-	-
271	H10	ext_bus_data30	pnl_tf12it0nn2	LVTTL	-	Ext Data 30	-	-
272	B9	pnl_go_t02	pnl_go	-	-	VSSIO	-	-
273	G10	ext_bus_data29	pnl_tf12it0nn2	LVTTL	-	Ext Data 29	-	-
274	F10	ext_bus_data28	pnl_tf12it0nn2	LVTTL	-	Ext Data 28	-	-
275	A9	pnl_vop_t02	pnl_vop	3.3V	-	VDDIO	-	-
276	H11	ext_bus_data27	pnl_tf12it0nn2	LVTTL	-	Ext Data 27	-	-
277	E10	ext_bus_data26	pnl_tf12it0nn2	LVTTL	-	Ext Data 26	-	-
278	D10	ext_bus_data25	pnl_tf12it0nn2	LVTTL	-	Ext Data 25	-	-
279	C10	ext_bus_data24	pnl_tf12it0nn2	LVTTL	-	Ext Data 24	-	-
280	B10	pnl_gcs_t04	pnl_gcs	-	-	VSS	-	-
281	A10	ext_bus_data23	pnl_tf12it0nn2	LVTTL	-	Ext Data 23	-	-
282	G11	ext_bus_data22	pnl_tf12it0nn2	LVTTL	-	Ext Data 22	-	-
283	F11	ext_bus_data21	pnl_tf12it0nn2	LVTTL	-	Ext Data 21	-	-
284	E11	ext_bus_data20	pnl_tf12it0nn2	LVTTL	-	Ext Data 20	-	-

Pin	Bump	Pin Name	Master Cell	Level	Dir.	Description	Termination	Polarity
285	D11	pnl_vc.t03	pnl_vc	1.0V	-	VDD	-	-
286	C11	ext_bus_data19	pnl_tf12it0nn2	LVTTL	-	Ext Data 19	-	-
287	A11	ext_bus_data18	pnl_tf12it0nn2	LVTTL	-	Ext Data 18	-	-
288	B11	ext_bus_data17	pnl_tf12it0nn2	LVTTL	-	Ext Data 17	-	-
289	E12	ext_bus_data16	pnl_tf12it0nn2	LVTTL	-	Ext Data 16	-	-
290	D12	pnl_gcs.t05	pnl_gcs	-	-	VSS	-	-
291	C12	ext_bus_data15	pnl_tf12it0nn2	LVTTL	-	Ext Data 15	-	-
292	B12	ext_bus_data14	pnl_tf12it0nn2	LVTTL	-	Ext Data 14	-	-
293	A12	ext_bus_data13	pnl_tf12it0nn2	LVTTL	-	Ext Data 13	-	-
294	F12	ext_bus_data12	pnl_tf12it0nn2	LVTTL	-	Ext Data 12	-	-
295	G12	pnl_go.t03	pnl_go	-	-	VSSIO	-	-
296	H13	ext_bus_data11	pnl_tf12it0nn2	LVTTL	-	Ext Data 11	-	-
297	G13	ext_bus_data10	pnl_tf12it0nn2	LVTTL	-	Ext Data 10	-	-
298	F13	ext_bus_data9	pnl_tf12it0nn2	LVTTL	-	Ext Data 9	-	-
299	A13	ext_bus_data8	pnl_tf12it0nn2	LVTTL	-	Ext Data 8	-	-
300	B13	pnl_vop.t03	pnl_vop	3.3V	-	VDDIO	-	-
301	C13	ext_bus_data7	pnl_tf12it0nn2	LVTTL	-	Ext Data 7	-	-
302	D13	ext_bus_data6	pnl_tf12it0nn2	LVTTL	-	Ext Data 6	-	-
303	E13	pnl_gcs.t06	pnl_gcs	-	-	VSS	-	-
304	G14	ext_bus_data5	pnl_tf12it0nn2	LVTTL	-	Ext Data 5	-	-
305	A14	ext_bus_data4	pnl_tf12it0nn2	LVTTL	-	Ext Data 4	-	-
306	B14	ext_bus_data3	pnl_tf12it0nn2	LVTTL	-	Ext Data 3	-	-
307	C14	ext_bus_data2	pnl_tf12it0nn2	LVTTL	-	Ext Data 2	-	-
308	D14	pnl_vc.t04	pnl_vc	1.0V	-	VDD	-	-
309	E14	ext_bus_data1	pnl_tf12it0nn2	LVTTL	-	Ext Data 1	-	-
310	F14	ext_bus_data0	pnl_tf12it0nn2	LVTTL	-	Ext Data 0	-	-
311	A15	pnl_gcs.t07	pnl_gcs	-	-	VSS	-	-
312	B15	ext_bus_addr31	pnl_tf12it0nn2	LVTTL	-	Ext Address 31	-	-
313	C15	ext_bus_addr30	pnl_tf12it0nn2	LVTTL	-	Ext Address 30	-	-
314	D15	pnl_go.t04	pnl_go	-	-	VSSIO	-	-
315	E15	ext_bus_addr29	pnl_tf12it0nn2	LVTTL	-	Ext Address 29	-	-
316	F15	ext_bus_addr28	pnl_tf12it0nn2	LVTTL	-	Ext Address 28	-	-
317	A16	pnl_vc.t05	pnl_vc	1.0V	-	VDD	-	-
318	B16	ext_bus_addr27	pnl_tf12it0nn2	LVTTL	-	Ext Address 27	-	-
319	C16	ext_bus_addr26	pnl_tf12it0nn2	LVTTL	-	Ext Address 26	-	-
320	D16	pnl_vop.t04	pnl_vop	3.3V	-	VDDIO	-	-
321	E16	ext_bus_addr25	pnl_tf12it0nn2	LVTTL	-	Ext Address 25	-	-
322	G15	ext_bus_addr24	pnl_tf12it0nn2	LVTTL	-	Ext Address 24	-	-
323	A17	pnl_go.t05	pnl_go	-	-	VSSIO	-	-
324	B17	ext_bus_addr23	pnl_tf12it0nn2	LVTTL	-	Ext Address 23	-	-
325	C17	ext_bus_addr22	pnl_tf12it0nn2	LVTTL	-	Ext Address 22	-	-
326	D17	pnl_gcs.t08	pnl_gcs	-	-	VSS	-	-
327	E17	ext_bus_addr21	pnl_tf12it0nn2	LVTTL	-	Ext Address 21	-	-
328	F16	ext_bus_addr20	pnl_tf12it0nn2	LVTTL	-	Ext Address 20	-	-
329	A18	pnl_vc.t06	pnl_vc	1.0V	-	VDD	-	-
330	B18	ext_bus_addr19	pnl_tf12it0nn2	LVTTL	-	Ext Address 19	-	-
331	C18	ext_bus_addr18	pnl_tf12it0nn2	LVTTL	-	Ext Address 18	-	-
332	D18	pnl_gcs.t09	pnl_gcs	-	-	VSS	-	-
333	E18	ext_bus_addr17	pnl_tf12it0nn2	LVTTL	-	Ext Address 17	-	-
334	F17	ext_bus_addr16	pnl_tf12it0nn2	LVTTL	-	Ext Address 16	-	-
335	A19	pnl_go.t06	pnl_go	-	-	VSSIO	-	-
336	B19	ext_bus_addr15	pnl_tf12it0nn2	LVTTL	-	Ext Address 15	-	-
337	C19	ext_bus_addr14	pnl_tf12it0nn2	LVTTL	-	Ext Address 14	-	-
338	D19	pnl_gcs.t10	pnl_gcs	-	-	VSS	-	-
339	E19	ext_bus_addr13	pnl_tf12it0nn2	LVTTL	-	Ext Address 13	-	-
340	F18	ext_bus_addr12	pnl_tf12it0nn2	LVTTL	-	Ext Address 12	-	-
341	A20	pnl_vop.t05	pnl_vop	3.3V	-	VDDIO	-	-
342	B20	ext_bus_addr11	pnl_tf12it0nn2	LVTTL	-	Ext Address 11	-	-
343	C20	ext_bus_addr10	pnl_tf12it0nn2	LVTTL	-	Ext Address 10	-	-
344	D20	pnl_vc.t07	pnl_vc	1.0V	-	VDD	-	-
345	E20	ext_bus_addr9	pnl_tf12it0nn2	LVTTL	-	Ext Address 9	-	-
346	A21	ext_bus_addr8	pnl_tf12it0nn2	LVTTL	-	Ext Address 8	-	-
347	B21	pnl_gcs.t11	pnl_gcs	-	-	VSS	-	-
348	C21	ext_bus_addr7	pnl_tf12it0nn2	LVTTL	-	Ext Address 7	-	-
349	A22	ext_bus_addr6	pnl_tf12it0nn2	LVTTL	-	Ext Address 6	-	-

Pin	Bump	Pin Name	Master Cell	Level	Dir.	Description	Termination	Polarity
350	B22	ext_bus_addr5	pnl_tf12it0nn2	LVTTL	-	Ext Address 5	-	-
351	B23	ext_bus_addr4	pnl_tf12it0nn2	LVTTL	-	Ext Address 4	-	-
352	C22	ext_bus_addr3	pnl_tf12it0nn2	LVTTL	-	Ext Address 3	-	-
353	D21	ext_bus_addr2	pnl_tf12it0nn2	LVTTL	-	Ext Address 2	-	-
354	F19	pnl_gcs_l00	pnl_gcs	-	-	VSS	-	-
355	G19	ext_bus_mask3	pnl_tf12it0nn2	LVTTL	-	Ext Mask 3	Pull-Up	-
356	C23	ext_bus_mask2	pnl_tf12it0nn2	LVTTL	-	Ext Mask 2	Pull-Up	-
357	F20	ext_bus_mask1	pnl_tf12it0nn2	LVTTL	-	Ext Mask 1	Pull-Up	-
358	D22	ext_bus_mask0	pnl_tf12it0nn2	LVTTL	-	Ext Mask 0	Pull-Up	-
359	E21	pnl_vc_l00	pnl_vc	1.0V	-	VDD	-	-
360	G18	chip_id0_	pnl_tf12it0nn2	LVTTL	-	Chip ID 0	Pull-Up	-
361	G17	ext_bus_bus_req2_	pnl_tf12it0nn2	LVTTL	-	Ext Request 2	Pull-Up	Active Low
362	D23	ext_bus_bus_req1_	pnl_tf12it0nn2	LVTTL	-	Ext Request 1	Pull-Up	Active Low
363	G16	ext_bus_bus_req0_	pnl_tf12it0nn2	LVTTL	-	Ext Request 0	Pull-Up	Active Low
364	E22	pnl_gcs_l01	pnl_gcs	-	-	VSS	-	-
365	F21	chip_id1	pnl_tf12it0nn2	LVTTL	-	Chip ID 1	Pull-Down	-
366	G20	ext_bus_bus_grnt2_	pnl_tf12it0nn2	LVTTL	-	Ext Grant 2	Pull-Up	Active Low
367	H18	ext_bus_bus_grnt1_	pnl_tf12it0nn2	LVTTL	-	Ext Grant 1	Pull-Up	Active Low
368	E23	ext_bus_bus_grnt0_	pnl_tf12it0nn2	LVTTL	-	Ext Grant 0	Pull-Up	Active Low
369	H19	pnl_go_l00	pnl_go	-	-	VSSIO	-	-
370	H17	ext_bus_cs3_	pnl_tf12it0nn2	LVTTL	-	Ext CS 3	Pull-Up	Active Low
371	F22	ext_bus_cs2_	pnl_tf12it0nn2	LVTTL	-	Ext CS 2	Pull-Up	Active Low
372	H20	ext_bus_cs1_	pnl_tf12it0nn2	LVTTL	-	Ext CS 1	Pull-Up	Active Low
373	G21	ext_bus_cs0_	pnl_tf12it0nn2	LVTTL	-	Ext CS 0	Pull-Up	Active Low
374	F23	pnl_vop_l00	pnl_vop	3.3V	-	VDDIO	-	-
375	J18	ext_bus_oe_	pnl_tf12it0nn2	LVTTL	-	Ext Output Enable	Pull-Up	Active Low
376	J17	ext_bus_as_	pnl_tf12it0nn2	LVTTL	-	Ext Address Strobe	Pull-Up	Active Low
377	G22	ext_bus_rw	pnl_tf12it0nn2	LVTTL	-	Ext Read/Write	Pull-Up	-
378	J19	ext_bus_rdy_	pnl_tf12it0nn2	LVTTL	-	Ext Ready	Pull-Up	Active Low
379	H21	pnl_vc_l01	pnl_vc	1.0V	-	VDD	-	-
380	G23	ext_bus_br_req2	pnl_tf12it0nn2	LVTTL	-	Ext Burst Req 2	Pull-Up	-
381	J20	ext_bus_br_req1	pnl_tf12it0nn2	LVTTL	-	Ext Burst Req 1	Pull-Up	-
382	K17	ext_bus_br_req0	pnl_tf12it0nn2	LVTTL	-	Ext Burst Req 0	Pull-Up	-
383	K18	ext_bus_br_ack2	pnl_tf12it0nn2	LVTTL	-	Ext Burst Ack 2	Pull-Up	-
384	H22	ext_bus_br_ack1	pnl_tf12it0nn2	LVTTL	-	Ext Burst Ack 1	Pull-Up	-
385	J21	ext_bus_br_ack0	pnl_tf12it0nn2	LVTTL	-	Ext Burst Ack 0	Pull-Up	-
386	H23	pnl_gcs_l02	pnl_gcs	-	-	VSS	-	-
387	L16	ext_init_size_mode0	pnl_tf12it0nn2	LVTTL	-	Ext Init Size 0	Pull-Down	-
388	K19	ext_init_size_mode1	pnl_tf12it0nn2	LVTTL	-	Ext Init Size 1	Pull-Down	-
389	L17	ext_bus_auto_rdy_en_	pnl_tf12it0nn2	LVTTL	-	Auto Ready Enable	Pull-Up	Active Low
390	K20	debug_en_	pnl_it2pu8	LVTTL	-	Debug Enable	Pull-Up	Active Low
391	J22	ext_bus_dma_ack1_	pnl_tf12it0nn2	LVTTL	-	Ext DMA Ack 1	Pull-Up	Active Low
392	J23	ext_bus_dma_ack0_	pnl_tf12it0nn2	LVTTL	-	Ext DMA Ack 0	Pull-Up	Active Low
393	M17	ext_bus_dma_req1_	pnl_tf12it0nn2	LVTTL	-	Ext DMA Req 1	Pull-Up	Active Low
394	K21	ext_bus_dma_req0_	pnl_tf12it0nn2	LVTTL	-	Ext DMA Req 0	Pull-Up	Active Low
395	L18	pnl_vc_l02	pnl_vc	1.0V	-	VDD	-	-
396	K22	ext_bus_irq1	pnl_tf12it0nn2	LVTTL	-	Ext IRQ 1	Pull-Down	Active High
397	M18	ext_bus_irq0	pnl_tf12it0nn2	LVTTL	-	Ext IRQ 0	Pull-Down	Active High
398	K23	pnl_go_l01	pnl_go	-	-	VSSIO	-	-
399	L19	ext_bus_clk	pnl_clk0nn8	LVTTL	-	Ext Clock	-	-
400	L20	pnl_vop_l01	pnl_vop	3.3V	-	VDDIO	-	-
401	L21	ext_init_sync_mode0	pnl_tf12it0nn2	LVTTL	-	Ext Init Sync Mode 0	Pull-Down	Active High
402	L22	chip_id2	pnl_tf12it0nn2	LVTTL	-	Chip ID 2	Pull-Down	-
403	L23	pnl_vc_l03	pnl_vc	1.0V	-	VDD	-	-
404	M19	chip_id3	pnl_tf12it0nn2	LVTTL	-	Chip ID 3	Pull-Down	-
405	M20	ext_master_cs_	pnl_tf12it0nn2	LVTTL	-	Ext Master CS	Pull-Up	Active Low
406	M21	pnl_gcs_l04	pnl_gcs	-	-	VSS	-	-
407	M22	pnl_go_bkp_l00	pnl_go	-	-	VSSIO	-	-
408	M23	pwm_in1	pnl_it2nn2	LVTTL	-	PWM Input 1	-	-
409	N18	pwm_in0	pnl_it2nn2	LVTTL	-	PWM Input 0	-	-
410	N19	pnl_vc_bkp_l00	pnl_vc	1.0V	-	VDD	-	-
411	N20	pwm_out5	pnl_tf04it0nn2	LVTTL	-	PWM Output 5	-	-
412	N21	pwm_out4	pnl_tf04it0nn2	LVTTL	-	PWM Output 4	-	-
413	N22	pwm_out3	pnl_tf04it0nn2	LVTTL	-	PWM Output 3	-	-
414	N23	pnl_gcs_bkp_l00	pnl_gcs	-	-	VSS	-	-

Pin	Bump	Pin Name	Master Cell	Level	Dir.	Description	Termination	Polarity
415	N17	pwm_out2	pnl.tf04it0nn2	LVTTL	-	PWM Output 2	-	-
416	N16	pwm_out1	pnl.tf04it0nn2	LVTTL	-	PWM Output 1	-	-
417	P18	pwm_out0	pnl.tf04it0nn2	LVTTL	-	PWM Output 0	-	-
418	P19	pnl.vop_bkp.l00	pnl.vop	3.3V	-	VDDIO	-	-
419	P20	pulse_counter_pz1	pnl.it2nn2	LVTTL	-	Pulse Counter CH1 PZ	-	-
420	P23	pulse_counter_pb1	pnl.it2nn2	LVTTL	-	Pulse Counter CH1 PB	-	-
421	P22	pulse_counter_pa1	pnl.it2nn2	LVTTL	-	Pulse Counter CH1 PA	-	-
422	P21	pnl.vc_bkp.l01	pnl.vc	1.0V	-	VDD	-	-
423	R18	pulse_counter_pz0	pnl.it2nn2	LVTTL	-	Pulse Counter CH0 PZ	-	-
424	R19	pulse_counter_pb0	pnl.it2nn2	LVTTL	-	Pulse Counter CH0 PB	-	-
425	R20	pulse_counter_pa0	pnl.it2nn2	LVTTL	-	Pulse Counter CH0 PA	-	-
426	R23	pnl.gcs_bkp.l01	pnl.gcs	-	-	VSS	-	-
427	R22	rtc_pad_rtc_clk	pnl.clk0nn8	LVTTL	-	RTC Clock	-	-
428	R21	rtc_pad_rtc_reset_	pnl.it2pu8	LVTTL	-	RTC Reset	Pull-Up	Active Low
429	T19	rtc_pad_rtc_hold_	pnl.it2pu8	LVTTL	-	RTC Hold	Pull-Up	Active Low
430	T23	pnl.vc_bkp.l02	pnl.vc	1.0V	-	VDD	-	-
431	T22	clk_iopad_reset_in_	pnl.it2pu2	LVTTL	-	Reset Input	Pull-Up	Active Low
432	T21	clk_iopad_reset_out_	pnl.tfl2it0nn2	LVTTL	-	Reset Output	-	-
433	T20	pnl.go_bkp.l01	pnl.go	-	-	VSSIO	-	-
434	U21	clk_iopad_clk.out	pnl.tfl2it0nn2	LVTTL	-	Clock Output	-	-
435	U23	clk_iopad_FIN	pnl.it2nn2	LVTTL	-	PLL FIN	-	-
436	U22	clk_iopad_FOUT	pnl.tfl2it0nn2	LVTTL	-	PLL FOUT	-	-
437	U20	pnl.vop_bkp.l01	pnl.vop	3.3V	-	VDDIO	-	-
438	T18	clk_iopad_F0	pnl.it2pd2	LVTTL	-	PLL F0	-	-
439	V21	clk_iopad_F1	pnl.it2pu2	LVTTL	-	PLL F1	-	-
440	V22	clk_iopad_F2	pnl.it2pu2	LVTTL	-	PLL F2	-	-
441	V23	pnl.vc_bkp.l03	pnl.vc	1.0V	-	VDD	-	-
442	U19	clk_iopad_F3	pnl.it2pu2	LVTTL	-	PLL F3	-	-
443	P17	clk_iopad_F4	pnl.it2pd2	LVTTL	-	PLL F4	-	-
444	V20	clk_iopad_F5	pnl.it2pd2	LVTTL	-	PLL F5	-	-
445	W21	pnl.gcs_bkp.l02	pnl.gcs	-	-	VSS	-	-
446	W22	clk_iopad_BP	pnl.it2pd2	LVTTL	-	PLL BP	-	-
447	W23	pnl.vc_bkp.l04	pnl.vc	1.0V	-	VDD	-	-
448	R17	clk_iopad_R0	pnl.it2pd2	LVTTL	-	PLL R0	-	-
449	P16	clk_iopad_R1	pnl.it2pu2	LVTTL	-	PLL R1	-	-
450	U18	pnl.gcs_bkp.l03	pnl.gcs	-	-	VSS	-	-
451	W20	clk_iopad_R2	pnl.it2pd2	LVTTL	-	PLL R2	-	-
452	Y22	clk_iopad_R3	pnl.it2pd2	LVTTL	-	PLL R3	-	-
453	Y23	pnl.vop_bkp.l02	pnl.vop	3.3V	-	VDDIO	-	-
454	Y21	clk_iopad_OEB	pnl.it2pd2	LVTTL	-	PLL OEB	-	-
455	V19	pnl.go_bkp.l02	pnl.go	-	-	VSSIO	-	-
456	V18	clk_iopad_OD	pnl.it2pd2	LVTTL	-	PLL OD	-	-
457	T17	pnl.gcs_bkp.l04	pnl.gcs	-	-	VSS	-	-
458	R16	clk_iopad_PD	pnl.it2pd2	LVTTL	-	PLL PD	-	-
459	AA23	clk_iopad_PVSS2P	PVSS2P	-	-	IO Analog VSS	-	-
460	AA22	clk_iopad_PVDD2P	PVDD2P	3.3V	-	IO Analog VDD	-	-
461	Y20	clk_iopad_PVDD1P1	PVDD1P	3.3V	-	PLL Analog VDD	-	-
462	AA21	clk_iopad_PVDD1P0	PVDD1P	3.3V	-	PLL Analog VDD	-	-
463	W19	clk_iopad_PVSS1PC0	PVSS1PC	-	-	PLL Digital VSS	-	-
464	AB23	clk_iopad_PVSS1P1	PVSS1P	-	-	PLL Analog VSS	-	-
465	AB22	clk_iopad_PVSS1P0	PVSS1P	-	-	PLL Analog VSS	-	-
466	AB21	clk_iopad_PVDD1PC0	PVDD1PC	1.0V	-	PLL Digital VDD	-	-

3

Instruction

3.1 サポートする命令

表 3.1: サポートする命令一覧

LB	LH	LW	LBU	LHU			
SB	SH	SW					
LWC1	SWC1						
ADDI	ADDIU	SLTI	SLTIU	ANDI	ORI	XORI	LUI
ADD	ADDU	SUB	SUBU	AND	OR	XOR	NOR
SLL	SRL	SRA	SLLV	SRLV	SRAV		
ADD.fmt	SUB.fmt	MUL.fmt	DIV.fmt	ABS.fmt	MOV.fmt	NEG.fmt	
ROUND.W.fmt	TRUNC.W.fmt	CEIL.W.fmt	FLOOR.W.fmt	CVT.S.fmt	CVT.D.fmt	CVT.W.fmt	C.cond.fmt
J	JAL	JR	JALR				
BEQ	BNE	BLEZ	BGTZ	BLTZ	BGEZ	BLTZAL	BGEZAL
BC1F	BC1T						
SYSCALL	BREAK						
MULT	MULTU	DIV	DIVU				
MTIMM	MFIMM	MTDMM	MFDMM				
MTC0	MFC0	MTC1	MFC1	CTC1	CFC1		
ERET							

3.2 サポートしていない命令

サポートしていない MIPS の命令は以下のとおりである .

表 3.2: サポートしていない命令一覧

LDC1	SDC1	RFE
------	------	-----

3.3 命令セットの符合化

3.3.1 命令の形式

R 形式

31	26	25	21	20	16	15	11	10	6	5	0
op	rs	rt	rd	shamt	func						
6bit	5bit	5bit	5bit	5bit	6bit						

I 形式

31	26	25	21	20	16	15	0
op	rs	rt	address/immediate				
6bit	5bit	5bit	16bit				

J 形式

31	26	25	0
op	target		
6bit	24bit		

3.3.2 Opecode

31	26	25	0
OP			

		OP[28:26]							
		000	001	010	011	100	101	110	111
OP[31:29]	000	SPECIAL	REGIMM	J	JAL	BEQ	BNE	BLEZ	BGTZ
	001	ADDI	ADDIU	SLTI	SLTIU	ANDI	ORI	XORI	LUI
	010	COP0	COP1						
	011								
	100	LB	LH		LW	LBU	LHU		
	101	SB	SH		SW				
	110		LWC1						
	111		SWC1						

3.3.3 Function

31	26	25	6	5	0
SPECIAL			FUNC		

		FUNC[2:0]							
		000	001	010	011	100	101	110	111
FUNC[5:3]	000	SLL		SRL	SRA	SLLV		SRLV	SRAV
	001	JR	JALR			SYSCALL	BREAK		
	010								
	011	ERET							
	100	ADD	ADDU	SUB	SUBU	AND	OR	XOR	NOR
	101			SLT	SLTU				
	110								
	111								

3.3.4 Rt

31	26	25	21	20	16	15	0
REGIMM			RT				

		RT[18:16]							
		000	001	010	011	100	101	110	111
RT[20:19]	00	BLTZ	BGEZ						
	01								
	10	BLTZAL	BGEZAL						
	11								

3.3.5 Rs

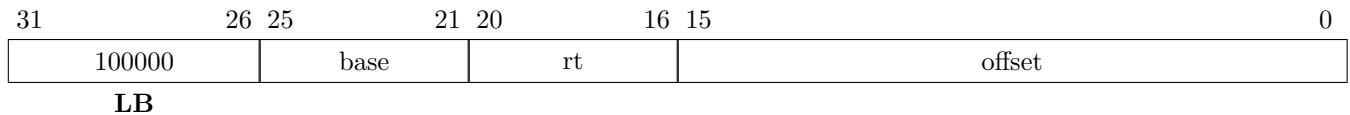
31	26	25	21	20	0
COP		RS			

		RS[23:21]							
		000	001	010	011	100	101	110	111
RS[25:24]	00	MF	CF			MT	CT		
	01	BC							
	10	S	D			W			
	11								

3.4 命令セットの詳細

3.4.1 メモリアクセス命令

LB (Load Byte) : バイトのロード



Format:

LB rt, offset(base)

Description:

$\text{GPR}[\text{rt}] \leftarrow \text{MEMORY}[\text{GPR}[\text{base}] + \text{offset}]$

Exception:

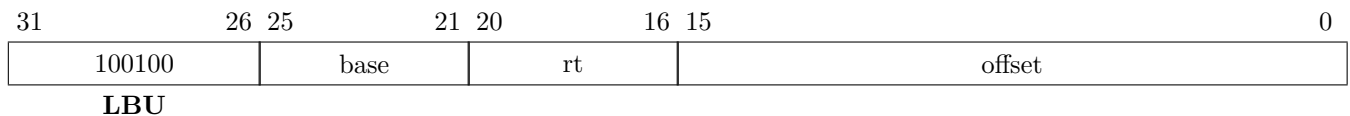
None

Programming Notes:

8bit データをメモリから読み出す .

読み出したデータは 32bit に符号拡張される .

LBU (Load Byte Unsigned) : バイトのロード (符号無し)



Format:

LBU rt, offset(base)

Description:

$\text{GPR}[\text{rt}] \leftarrow \text{MEMORY}[\text{GPR}[\text{base}] + \text{offset}]$

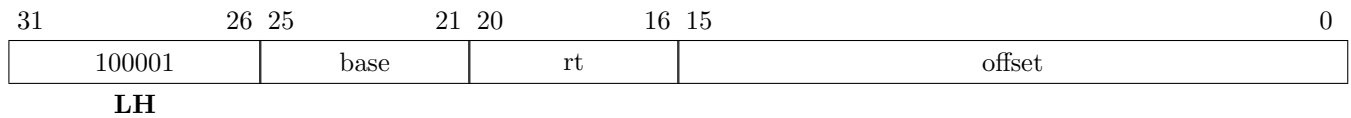
Exception:

None

Programming Notes:

8bit データをメモリから読み出す .

LH (Load Halfword) : ハーフワードのロード



Format:

LH rt, offset(base)

Description:

$\text{GPR}[\text{rt}] \leftarrow \text{MEMORY}[\text{GPR}[\text{base}] + \text{offset}]$

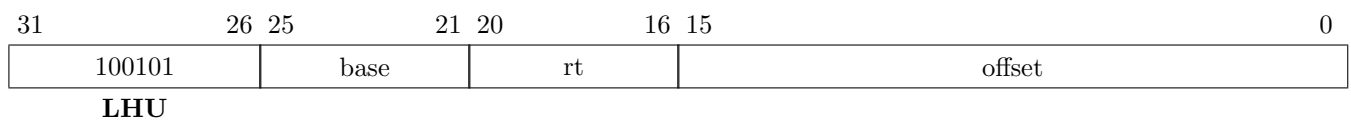
Exception:

Address Miss Aligned

Programming Notes:

16bit データをメモリから読み出す .
読み出したデータは 32bit に符号拡張される .

LHU (Load Halfword Unsigned) : ハーフワードのロード (符号無し)



Format:

LHU rt, offset(base)

Description:

$\text{GPR}[\text{rt}] \leftarrow \text{MEMORY}[\text{GPR}[\text{base}] + \text{offset}]$

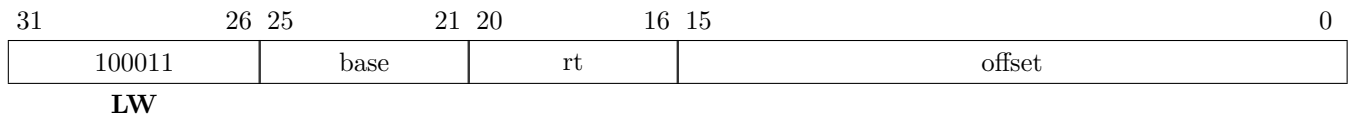
Exception:

Address Miss Aligned

Programming Notes:

16bit データをメモリから読み出す .

LW (Load Word) : ワードのロード



Format:

LW rt, offset(base)

Description:

$GPR[rt] \leftarrow MEMORY[GPR[base]+offset]$

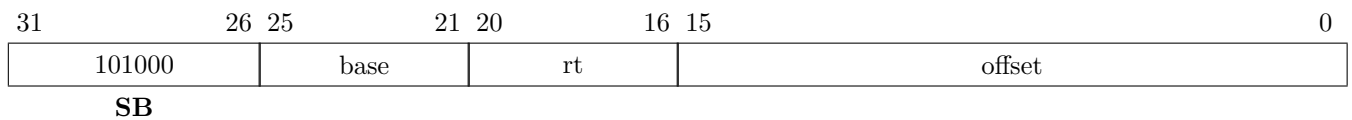
Exception:

Address Miss Aligned

Programming Notes:

32bit データをメモリから読み出す .

SB (Store Byte) : バイトのストア



Format:

SB rt, offset(base)

Description:

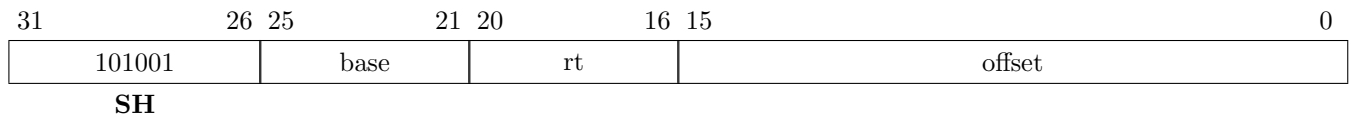
$MEMORY[GPR[base]+offset] \leftarrow GPR[rt]$

Exception:

None

Programming Notes:

8bit データをメモリに書き込む .

SH (Store Halfword) : ハーフワードのストア**Format:**

SH rt, offset(base)

Description:

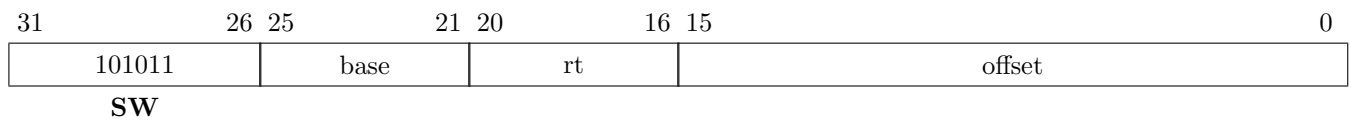
MEMORY[GPR[base]+offset] ← GPR[rt]

Exception:

Address Miss Aligned

Programming Notes:

16bit データをメモリに書き込む .

SW (Store Word) : ワードのストア**Format:**

SW rt, offset(base)

Description:

MEMORY[GPR[base]+offset] ← GPR[rt]

Exception:

Address Miss Aligned

Programming Notes:

32bit データをメモリに書き込む .

LWC1 (Load Word to Floating Point) : 浮動小数点レジスタ用 32bit ロード

31	26	25	21	20	16	15	0
110001	base				rt	offset	

LWC1

Format:

LWC1 ft, offset(base)

Description:

$FPR[rt] \leftarrow MEMORY[GPR[base] + offset]$

Exception:

Address Miss Aligned

Programming Notes:

32bit データをメモリから浮動小数点レジスタへロードする .

SWC1 (Store Word from Floating Point) : 浮動小数点レジスタ用 32bit ストア

31	26	25	21	20	16	15	0
111001	base				rt	offset	

SWC1

Format:

SWC1 ft, offset(base)

Description:

$MEMORY[GPR[base] + offset] \leftarrow FPR[rt]$

Exception:

Address Miss Aligned

Programming Notes:

32bit データを浮動小数点レジスタからメモリへストアする .

3.4.2 即値命令

ADDI (Add Immediate Word) : 即値の加算

31	26	25	21	20	16	15	0
001000	rs	rt	immediate				

ADDI

Format:

ADDI rt, rs, immediate

Description:

$\text{GPR}[\text{rt}] \leftarrow \text{GPR}[\text{rs}] + \text{sign_extention}(\text{immediate})$

Exception:

Arithmetic Overflow

Programming Notes:

32bit 加算を行う .

演算結果がオーバーフローした場合 , オーバーフロー例外が発生する .

ADDIU (Add Immediate Unsigned Word) : 即値の加算 (符号無し)

31	26	25	21	20	16	15	0
001001	rs	rt	immediate				

ADDIU

Format:

ADDIU rt, rs, immediate

Description:

$\text{GPR}[\text{rt}] \leftarrow \text{GPR}[\text{rs}] + \text{sign_extention}(\text{immediate})$

Exception:

None

Programming Notes:

32bit 加算を行う .

SLTI (Set on Less Than Immediate) : 即値未満へのセット

31	26	25	21	20	16	15	0
001010	rs	rt	immediate				

SLTI

Format:

SLTI rt, rs, immediate

Description:

$GPR[rt] \leftarrow (GPR[rs] < sign_extention(immediate))$

Exception:

None

Programming Notes:

32bit の値比較を行う .

rs が immediate より小さい場合 rd に 1 がセットされる .

そうでない場合 0 がセットされる .

SLTIU(Set on Less Than Immediate Unsigned) : 即値未満へのセット (符号無し)

31	26	25	21	20	16	15	0
001011	rs	rt	immediate				

SLTIU

Format:

SLTIU rt, rs, immediate

Description:

$GPR[rt] \leftarrow (GPR[rs] < sign_extention(immediate))$

Exception:

None

Programming Notes:

32bit の値比較を行う .

rs が immediate より小さい場合 rd に 1 がセットされる .

そうでない場合 0 がセットされる . 値は符号無し整数として扱われる .

ANDI (And Immediate) : 即値の論理積

31	26	25	21	20	16	15	0
001100	rs	rt	immediate				

ANDI**Format:**

ANDI rt, rs, immediate

Description: $\text{GPR}[\text{rt}] \leftarrow \text{GPR}[\text{rs}] \text{ AND zero_extention(immediate)}$ **Exception:**

None

Programming Notes:

32bit 論理積演算を行う .

ORI (Or Immediate) : 即値の論理和

31	26	25	21	20	16	15	0
001101	rs	rt	immediate				

ORI**Format:**

ORI rt, rs, immediate

Description: $\text{GPR}[\text{rt}] \leftarrow \text{GPR}[\text{rs}] \text{ OR zero_extention(immediate)}$ **Exception:**

None

Programming Notes:

32bit 論理和演算を行う .

XORI (Exclusive Or Immediate) : 即値の排他的論理和

31	26	25	21	20	16	15	0
001110	rs	rt	immediate				

XORI

Format:

XORI rt, rs, immediate

Description:

$\text{GPR}[\text{rt}] \leftarrow \text{GPR}[\text{rs}] \text{ XOR zero_extention(immediate)}$

Exception:

None

Programming Notes:

32bit 排他的論理和演算を行う .

LUI (Load Upper Immediate) : 即値の上位へのロード

31	26	25	21	20	16	15	0
001111	00000	rt	immediate				

LUI

0

Format:

LUI rt, immediate

Description:

$\text{GPR}[\text{rt}] \leftarrow \{ \text{immediate}, 0^{16} \}$

Exception:

None

Programming Notes:

immediate の値を 32bit レジスタの上位 16bit にロードする .

3.4.3 ALU 命令

ADD (Add Word) : 加算

31	26	25	21	20	16	15	11	10	6	5	0
000000	rs	rt	rd	00000	100000						
SPECIAL				0				ADD			

Format:

ADD rd, rs, rt

Description:

$\text{GPR}[\text{rd}] \leftarrow \text{GPR}[\text{rs}] + \text{GPR}[\text{rt}]$

Exception:

Arithmetic Overflow

Programming Notes:

32bit 加算を行う .

演算結果がオーバーフローした場合 , オーバーフロー例外が発生する .

ADDU (Add Unsigned Word) : 加算 (符号無し)

31	26	25	21	20	16	15	11	10	6	5	0
000000	rs	rt	rd	00000	100001						
SPECIAL				0				ADDU			

Format:

ADDU rd, rs, rt

Description:

$\text{GPR}[\text{rd}] \leftarrow \text{GPR}[\text{rs}] + \text{GPR}[\text{rt}]$

Exception:

None

Programming Notes:

32bit 加算を行う .

SUB (Subtract Word) : 減算

31	26	25	21	20	16	15	11	10	6	5	0
000000	rs	rt	rd	00000	100010						
SPECIAL					0			SUB			

Format:

SUB rd, rs, rt

Description:

$GPR[rd] \leftarrow GPR[rs] - GPR[rt]$

Exception:

Arithmetic Overflow

Programming Notes:

32bit 減算を行う .

演算結果がオーバーフローした場合 , オーバーフロー例外が発生する .

SUBU (Subtract Unsigned Word) : 減算 (符合無し)

31	26	25	21	20	16	15	11	10	6	5	0
000000	rs	rt	rd	00000	100011						
SPECIAL					0			SUBU			

Format:

SUBU rd, rs, rt

Description:

$GPR[rd] \leftarrow GPR[rs] - GPR[rt]$

Exception:

None

Programming Notes:

32bit 減算を行う .

AND (And) : 論理積

31	26	25	21	20	16	15	11	10	6	5	0
000000	rs	rt	rd	00000	100100						
SPECIAL						0			AND		

Format:

AND rd, rs, rt

Description: $GPR[rd] \leftarrow GPR[rs] \text{ AND } GPR[rt]$ **Exception:**

None

Programming Notes:

32bit 論理積演算を行う。

OR (Or) : 論理和

31	26	25	21	20	16	15	11	10	6	5	0
000000	rs	rt	rd	00000	100101						
SPECIAL						0			OR		

Format:

OR rd, rs, rt

Description: $GPR[rd] \leftarrow GPR[rs] \text{ OR } GPR[rt]$ **Exception:**

None

Programming Notes:

32bit 論理和演算を行う。

XOR (Exclusive Or) : 排他的論理和

31	26	25	21	20	16	15	11	10	6	5	0
000000	rs	rt	rd	00000	100110						
SPECIAL					0			XOR			

Format:

XOR rd, rs, rt

Description:

$GPR[rd] \leftarrow GPR[rs] \text{ XOR } GPR[rt]$

Exception:

None

Programming Notes:

32bit 排他的論理和演算を行う .

NOR (Not Or) : 否定論理和

31	26	25	21	20	16	15	11	10	6	5	0
000000	rs	rt	rd	00000	100111						
SPECIAL					0			NOR			

Format:

NOR rd, rs, rt

Description:

$GPR[rd] \leftarrow GPR[rs] \text{ NOR } GPR[rt]$

Exception:

None

Programming Notes:

32bit 否定論理和演算を行う .

SLT (Set on Less Than) : 未満へのセット

31	26	25	21	20	16	15	11	10	6	5	0
000000	rs	rt	rd	00000	101010						
SPECIAL				0				SLT			

Format:

SLT rd, rs, rt

Description:

$\text{GPR}[\text{rd}] \leftarrow (\text{GPR}[\text{rs}] < \text{GPR}[\text{rt}])$

Exception:

None

Programming Notes:

32bit の値比較を行う .

rs が rt より小さい場合 rd に 1 がセットされる .

そうでない場合 0 がセットされる .

SLTU (Set on Less Than Unsigned) : 未満へのセット (符号無し)

31	26	25	21	20	16	15	11	10	6	5	0
000000	rs	rt	rd	00000	101011						
SPECIAL				0				SLTU			

Format:

SLTU rd, rs, rt

Description:

$\text{GPR}[\text{rd}] \leftarrow (\text{GPR}[\text{rs}] < \text{GPR}[\text{rt}])$

Exception:

None

Programming Notes:

32bit の値比較を行う .

rs が rt より小さい場合 rd に 1 がセットされる .

そうでない場合 0 がセットされる . 値は符号無し整数として扱われる .

SLL (Shift Word Left Logical) : 論理左シフト

31	26	25	21	20	16	15	11	10	6	5	0
000000	00000	rt	rd	sa	000000						
SPECIAL			SHIFT			SLL					

Format:

SLL rd, rt, sa

Description:

$\text{GPR}[\text{rd}] \leftarrow \text{GPR}[\text{rt}] \ll \text{sa}$

Exception:

None

Programming Notes:

32bit 論理左シフトを行う .

SRL (Shift Word Right Logical) : 論理右シフト

31	26	25	21	20	16	15	11	10	6	5	0
000000	00000	rt	rd	sa	000010						
SPECIAL			SHIFT			SRL					

Format:

SRL rd, rt, sa

Description:

$\text{GPR}[\text{rd}] \leftarrow \text{GPR}[\text{rt}] \gg \text{sa} \text{ (logical)}$

Exception:

None

Programming Notes:

32bit 論理右シフトを行う .

SRA (Shift Word Right Arithmetic) : 算術右シフト

31	26	25	21	20	16	15	11	10	6	5	0
000000	00000	rt	rd	sa	000011						
SPECIAL			0			SRA					

Format:

SRA rd, rt, sa

Description: $\text{GPR}[\text{rd}] \leftarrow \text{GPR}[\text{rt}] \gg \text{sa} \text{ (arithmetic)}$ **Exception:**

None

Programming Notes:

32bit 算術右シフトを行う。

SLLV (Shift Word Left Logical Variable) : 論理変数左シフト

31	26	25	21	20	16	15	11	10	6	5	0
000000	rs	rt	rd	00000	000100						
SPECIAL						SHIFT			SLLV		

Format:

SLLV rd, rt, rs

Description: $\text{GPR}[\text{rd}] \leftarrow \text{GPR}[\text{rt}] \ll \text{GPR}[\text{rs}]$ **Exception:**

None

Programming Notes:

32bit 論理左シフトを行う。

SRLV (Shift Word Right Logical Variable) : 論理変数右シフト

31	26	25	21	20	16	15	11	10	6	5	0
000000	rs	rt	rd	00000	000110						
SPECIAL				SHIFT				SRLV			

Format:

SRLV rd, rt, rs

Description:

$GPR[rd] \leftarrow GPR[rt] \gg GPR[rs]$ (logical)

Exception:

None

Programming Notes:

32bit 論理右シフトを行う .

SRAV (Shift Word Right Arithmetic Variable) : 算術変数右シフト

31	26	25	21	20	16	15	11	10	6	5	0
000000	rs	rt	rd	00000	000111						
SPECIAL				0				SRAV			

Format:

SRAV rd, rt, rs

Description:

$GPR[rd] \leftarrow GPR[rt] \ggg GPR[rs]$ (arithmetic)

Exception:

None

Programming Notes:

32bit 算術右シフトを行う .

3.4.4 ジャンプ命令

J (Jump) : ジャンプ

31	26	25	0
000010	target		

J

Format:

J target

Description:

$PC \leftarrow PC_{31 \sim 28} \parallel \text{target} \parallel 00$

Exception:

None

Programming Notes:

直接ジャンプを行う。

JAL (Jump and Link) : ジャンプとリンク

31	26	25	0
000011	target		

J

Format:

JAL target

Description:

$GPR[31] \leftarrow pc + 0x04$

$PC \leftarrow PC_{31 \sim 28} \parallel \text{target} \parallel 00$

Exception:

None

Programming Notes:

戻り番地を GPR[31] に保存し、直接ジャンプを行う。

JR (Jump Register) : ジャンプレジスタ

31	26	25	21	20		6	5	0
000000		rs			0000000000000000			001000
SPECIAL			0			JR		

Format:

JR rs

Description:

PC ← target

Exception:

None

Programming Notes:

直接ジャンプを行う。

JALR (Jump and Link Register) : ジャンプレジスタとリンク

31	26	25	21	20	16	15	11	10	6	5	0
000000		rs			00000		rd		00000		001001
SPECIAL			0			0			JALR		

Format:

JR rs (rd = 31 implied)

JR rd, rs

Description:

GPR[rd] ← PC + 0x04

PC ← GPR[rs]

Exception:

None

Programming Notes:

戻り番地を rd に保存し、レジスタ間接ジャンプを行う。

3.4.5 分岐命令

BEQ (Branch on Equal) : 等号での分岐

31	26	25	21	20	16	15	0
000100	rs	rt	offset				

BEQ

Format:

BEQ rs, rt, offset

Description:

if GPR[rs] = GPR[rt] then branch

Exception:

None

Programming Notes:

rs の値が rt と等しい場合，分岐先アドレスへジャンプする．

BNE (Branch on Not Equal) : 不等号での分岐

31	26	25	21	20	16	15	0
000101	rs	rt	offset				

BNE

Format:

BNE rs, rt, offset

Description:

if GPR[rs] $\neq$ GPR[rt] then branch

Exception:

None

Programming Notes:

rs の値が rt と等しくない場合，分岐先アドレスへジャンプする．

BLEZ (Branch on Less Than or Equal to Zero) : ゼロ以下での分岐

31	26	25	21	20	16	15	0
000110		rs		00000		offset	
BLEZ				0			

Format:

BLEZ rs, offset

Description:

if $\text{GPR}[\text{rs}] \leq 0$ then branch

Exception:

None

Programming Notes:

rs の値が 0 以下の場合，分岐先アドレスへジャンプする．

BGTZ(Branch on Greater Than Zero) : ゼロ以上での分岐

31	26	25	21	20	16	15	0
000111		rs		00000		offset	
BGTZ				0			

Format:

BGTZ rs, offset

Description:

if $\text{GPR}[\text{rs}] > 0$ then branch

Exception:

None

Programming Notes:

rs の値が 0 より大きい場合，分岐先アドレスへジャンプする．

BLTZ (Branch on Less Than Zero) : ゼロ未満での分岐

31	26	25	21	20	16	15	0
000001	rs	00000	offset				
REGIMM			BLTZ				

Format:

BLTZ rs, offset

Description:

if $\text{GPR}[\text{rs}] < 0$ then branch

Exception:

None

Programming Notes:

rs の値が 0 より小さい場合 , 分岐先アドレスへジャンプする .

BGEZ (Branch on Greater Than or Equal to Zero) : ゼロ以上での分岐

31	26	25	21	20	16	15	0
000001	rs	00001	offset				
REGIMM			BGEZ				

Format:

BGEZ rs, offset

Description:

if $\text{GPR}[\text{rs}] \geq 0$ then branch

Exception:

None

Programming Notes:

rs の値が 0 以上の場合 , 分岐先アドレスへジャンプする .

BLTZAL (Branch on Less Than Zero and Link) : ゼロ未満での分岐とリンク

31	26	25	21	20	16	15	0
000001	rs	10000	offset				
REGIMM				BLTZAL			

Format:

BLTZAL rs, offset

Description:

if GPR[rs] < 0 then procedure_call

Exception:

None

Programming Notes:

rs の値が 0 より小さい場合，戻り番地を GPR[31] に保存し，分岐先アドレスへジャンプする．

BGEZAL (Branch on Greater Than or Equal to Zero and Link) : ゼロ以上での分岐とリンク

31	26	25	21	20	16	15	0
000001	rs	10001	offset				
REGIMM				BGEZAL			

Format:

BGEZAL rs, offset

Description:

if GPR[rs] ≥ 0 then procedure_call

Exception:

None

Programming Notes:

rs の値が 0 以上の場合，戻り番地を GPR[31] に保存し，分岐先アドレスへジャンプする．

3.4.6 System Call / Break Point 命令

SYSCALL (System Call) : システムコール

31	26	25	6	5	0
000000	00000				001100
SPECIAL				SYSCALL	

Format:
SYSCALL

Description:
signal.exception(System Call)

Exception:
System Call

Programming Notes:
システムコール例外を発生させる .

BREAK (Break Point) : ブレークポイント

31	26	25	6	5	0
000000	00000				001101
SPECIAL				BREAK	

Format:
BREAK

Description:
signal.exception(Breakpoint)

Exception:
Break Point

Programming Notes:
ブレークポイント例外を発生させる .

3.4.7 特権命令

MFC0 (Move from Coprocessor0) : コプロセッサ 0 からの転送

31	26	25	21	20	16	15	11	10	0
010000	00000	rt	rd	000000000000					
COP0	MF								0

Format:

MFC0 rt, rd

Description:

GPR[rt] ← CTRL[rd]

Exception:

Coprocessor Unusable

Programming Notes:

特権命令．制御レジスタから値を読み出す．
 ユーザモードで実行した場合，コプロセッサ不使用が発生する．

MTC0 (Move to Coprocessor0) : コプロセッサ 0 への転送

31	26	25	21	20	16	15	11	10	0
010000	00100	rt	rd	000000000000					
COP0	MT								0

Format:

MTC0 rt, rd

Description:

CTRL[rd] ← GPR[rt]

Exception:

Coprocessor Unusable

Programming Notes:

特権命令．制御レジスタに値を書き込む．
 ユーザモードで実行した場合，コプロセッサ不使用が発生する．

MFIMM (Move from Instruction MMU Control Register) : 命令 MMU 制御レジスタリード命令

31	26	25	21	20	16	15	11	10	6	5	0
010000	00000	rt	rd	00000	000010						
COP0	MF				0	IMMU					

Format:

MFIMM rt, rd

Description:

$\text{GPR}[\text{rt}] \leftarrow \text{IMMU}[\text{rd}]$

Exception:

Coprocessor Unusable

Programming Notes:

特権命令．命令 MMU 制御レジスタから値を読み出す．

ユーザモードで実行した場合，コプロセッサ不使用が発生する．

MTIMM (Move to Instruction MMU Control Register) : 命令 MMU 制御レジスタライト命令

31	26	25	21	20	16	15	11	10	6	5	0
010000	00100	rt	rd	00000	000010						
COP0	MT				0	IMMU					

Format:

MTIMM rt, rd

Description:

$\text{IMMU}[\text{rt}] \leftarrow \text{GPR}[\text{rd}]$

Exception:

Coprocessor Unusable

Programming Notes:

特権命令．命令 MMU 制御レジスタに値を書き込む．

ユーザモードで実行した場合，コプロセッサ不使用が発生する．

MFDMM (Move from Data MMU Control Register) : データ MMU 制御レジスタリード命令

31	26	25	21	20	16	15	11	10	6	5	0
010000	00000	rt	rd	00000	000011						
COP0	MF						0				DMMU

Format:

MFDMM rt, rd

Description:

GPR[rt] ← DMMU[rd]

Exception:

Coprocessor Unusable

Programming Notes:

特権命令．データ MMU 制御レジスタから値を読み出す．
 ユーザモードで実行した場合，コプロセッサ不使用が発生する．

MTDMM (Move to Data MMU Control Register) : データ MMU 制御レジスタライト命令

31	26	25	21	20	16	15	11	10	6	5	0
010000	00100	rt	rd	00000	000011						
COP0	MT						0				DMMU

Format:

MTDMM rt, rd

Description:

DMMU[rt] ← GPR[rd]

Exception:

Coprocessor Unusable

Programming Notes:

特権命令．データ MMU 制御レジスタに値を書き込む．
 ユーザモードで実行した場合，コプロセッサ不使用が発生する．

ERET (Exception Return) : 例外からの復帰

31	26	25		6	5		0
010000	00000000000000000000000000000000						011000
COP0			0				ERET

Format:

ERET

Description:

$KU_c \leftarrow KU_p$

$IE_c \leftarrow IE_p$

Exception:

Coprocessor Unusable

Programming Notes:

KU_p と IE_p を KU_c と IE_c に復帰する .

ERET 命令は PC を復元する JR 命令の遅延スロットで実行する .

ユーザモードで実行した場合 , コプロセッサ不使用が発生する .

3.4.8 FPU 命令

MFC1 (Move from Coprocessor1) : コプロセッサ 1 からの転送

31	26	25	21	20	16	15	11	10	0	
010001			00000		rt		fs		000000000000	
COP1			MF						0	

Format:

MFC1 rt, fs

Description:

$GPR[rt] \leftarrow FCR[fs]$

Exception:

Coprocessor Unusable

Programming Notes:

浮動小数点レジスタの値を汎用レジスタに入れる .

MTC1 (Move to Coprocessor1) : コプロセッサ 1 への転送

31	26	25	21	20	16	15	11	10	0
010001	00100	rt	fs	00000000000					
COP1		MT		0					

Format:

MTC0 rt, fs

Description:

FPR[fs] ← GPR[rt]

Exception:

Coprocessor Unusable

Programming Notes:

汎用レジスタの値を浮動小数点レジスタに入れる .

CFC1 (Move Control Word from Coprocessor1) : コプロセッサ 1 からコントロール・ワードの転送

31	26	25	21	20	16	15	11	10	0
010001	00010	rt	fs	00000000000					
COP1		CFC1		0					

Format:

CFC1 rt, fs

Description:

GPR[rt] ← FCR[fs]

Exception:

Coprocessor Unusable

Programming Notes:

FPU 制御レジスタの値を汎用レジスタに入れる .

CTC1 (Move Control Word to Coprocessor1) : コプロセッサ 1 ヘコントロール・ワードの転送

31	26	25	21	20	16	15	11	10	0
010001	00110	rt	fs	00000000000					
COP1		CTC1		0					

Format:

CTC1 rt, fs

Description:

$\text{FCR}[\text{fs}] \leftarrow \text{GPR}[\text{rt}]$

Exception:

Coprocessor Unusable

Programming Notes:

汎用レジスタの値を FPU 制御レジスタに入れる .

ADD.fmt (Floating Point Add) : 浮動小数点加算

31	26	25	21	20	16	15	11	10	6	5	0
010001		fmt		ft		fs		fd		000000	
COP1						FADD					

Format:

ADD.S fd, fs, ft (fmt = 10000)

ADD.D fd, fs, ft (fmt = 10001)

Description:

$\text{FPR}[\text{fd}] \leftarrow \text{FPR}[\text{fs}] + \text{FPR}[\text{ft}]$

Exception:

Floating Point Invalid Operation

Floating Point Inexact

Floating Point Overflow

Floating Point Underflow

Programming Notes:

単精度または倍精度浮動小数点加算を行う .

SUB.fmt (Floating Point Subtract) : 浮動小数点減算

31	26	25	21	20	16	15	11	10	6	5	0		
010001			fmt			fs		ft		fd		000001	
COP1										FSUB			

Format:

SUB.S fd, fs, ft (fmt = 10000)

SUB.D fd, fs, ft (fmt = 10001)

Description:

$\text{FPR}[\text{fd}] \leftarrow \text{FPR}[\text{fs}] - \text{FPR}[\text{ft}]$

Exception:

Floating Point Invalid Operation

Floating Point Inexact

Floating Point Overflow

Floating Point Underflow

Programming Notes:

単精度または倍精度浮動小数点減算を行う。

MUL.fmt (Floating Point Multiply) : 浮動小数点乗算

31	26	25	21	20	16	15	11	10	6	5	0	
010001			fmt		ft		fs		fd		000010	
COP1						FMUL						

Format:

MUL.S fd, fs, ft (fmt = 10000)

MUL.D fd, fs, ft (fmt = 10001)

Description:

$\text{FPR}[\text{fd}] \leftarrow \text{FPR}[\text{fs}] \times \text{FPR}[\text{ft}]$

Exception:

Floating Point Invalid Operation

Floating Point Inexact

Floating Point Overflow

Floating Point Underflow

Programming Notes:

単精度または倍精度浮動小数点乗算を行う。

DIV.fmt (Floating Point Divide) : 浮動小数点除算

31	26	25	21	20	16	15	11	10	6	5	0		
010001			fmt			fs		ft		fd		000011	
COP1										FDIV			

Format:

DIV.S fd, fs, ft (fmt = 10000)

DIV.D fd, fs, ft (fmt = 10001)

Description:

$FPR[fd] \leftarrow FPR[fs] / FPR[ft]$

Exception:

Floating Point Invalid Operation

Floating Point Inexact

Floating Point Overflow

Floating Point Underflow

Floating Point Divide By 0

Programming Notes:

単精度または倍精度浮動小数点除算を行う。

ABS.fmt (Floating Point Absolute Value) : 浮動小数点絶対値

31	26	25	21	20	16	15	11	10	6	5	0		
010001			fmt			fs		ft		fd		000101	
COP1										FABS			

Format:

ABS.S fd, fs (fmt = 10000)

ABS.D fd, fs (fmt = 10001)

Description:

$FPR[fd] \leftarrow \text{abs}(FPR[fs])$

Exception:

Floating Point Invalid Operation

Programming Notes:

単精度または倍精度浮動小数点レジスタの値の絶対値をとる。

MOV.fmt (Floating Point Move) : 浮動小数点移動

31	26	25	21	20	16	15	11	10	6	5	0
010001	fmt	fs	ft	fd	000110						
COP1						FMOV					

Format:

MOV.S fd, fs, ft (fmt = 10000)

MOV.D fd, fs, ft (fmt = 10001)

Description:

FPR[fd] ← FPR[fs]

Exception:

Floating Point Invalid Operation

Programming Notes:

単精度または倍精度浮動小数点レジスタの値をコピーする。

NEG.fmt (Floating Point Negate) : 浮動小数点符号反転

31	26	25	21	20	16	15	11	10	6	5	0
010001	fmt	fs	ft	fd	000111						
COP1						FNEG					

Format:

NEG.S fd, fs (fmt = 10000)

NEG.D fd, fs (fmt = 10001)

Description:

FPR[fd] ← -(FPR[fs])

Exception:

Floating Point Invalid Operation

Programming Notes:

単精度または倍精度浮動小数点レジスタの値を符号反転する。

CVT.S.fmt (Floating Point Convert to Single Floating Point) : 浮動小数点フォーマット変換

31	26	25	21	20	16	15	11	10	6	5	0	
010001			fmt		00000		fs		fd		100000	
COP1				0				CVTS				

Format:

- CVT.S.D fd, fs (fmt = 10001)
- CVT.S.W fd, fs (fmt = 10100)

Description:

FPR[fd] ← convert_and_round(FPR[fs])

Exception:

- Floating Point Invalid Operation
- Floating Point Inexact
- Floating Point Overflow
- Floating Point Underflow

Programming Notes:

レジスタの値を単精度浮動小数点フォーマットに変換する .

CVT.W.fmt (Floating Point Convert to Word Fixed Point) : 浮動小数点フォーマット変換

31	26	25	21	20	16	15	11	10	6	5	0	
010001			fmt		00000		fs		fd		100100	
COP1					0			CVTW				

Format:

CVT.W.S fd, fs (fmt = 10000)

CVT.W.D fd, fs (fmt = 10001)

Description:

FPR[fd] ← convert_and_round(FPR[fs])

Exception:

Floating Point Invalid Operation

Floating Point Inexact

Floating Point Overflow

Floating Point Underflow

Programming Notes:

レジスタの値を整数フォーマットに変換する .

ROUND.W.fmt (Floating Point Round to Word Fixed Point) : 浮動小数点フォーマット変換

31	26	25	21	20	16	15	11	10	6	5	0						
010001			fmt			00000			fs			fd			001100		
COP1						0						ROUNDW					

Format:

ROUND.W.S fd, fs (fmt = 10000)

ROUND.W.D fd, fs (fmt = 10001)

Description:

FPR[fd] ← convert_and_round(FPR[fs])

Exception:

Floating Point Invalid Operation

Floating Point Inexact

Floating Point Overflow

Programming Notes:

レジスタの値を整数フォーマットに変換する .

TRUNC.W.fmt (Floating Point Truncate to Word Fixed Point) : 浮動小数点フォーマット変換

31	26	25	21	20	16	15	11	10	6	5	0	
010001			fmt		00000		fs		fd		001101	
COP1				0				TRUNCW				

Format:

TRUNC.W.S fd, fs (fmt = 10000)

TRUNC.W.D fd, fs (fmt = 10001)

Description:

FPR[fd] ← convert_and_round(FPR[fs])

Exception:

Floating Point Invalid Operation

Floating Point Inexact

Floating Point Overflow

Programming Notes:

レジスタの値を整数フォーマットに変換する .

CEIL.W.fmt (Floating Point Ceiling to Word Fixed Point) : 浮動小数点フォーマット変換

31	26	25	21	20	16	15	11	10	6	5	0	
010001			fmt		00000		fs		fd		001110	
COP1				0				CEILW				

Format:

CEIL.W.S fd, fs (fmt = 10000)

CEIL.W.D fd, fs (fmt = 10001)

Description:

FPR[fd] ← convert_and_round(FPR[fs])

Exception:

Floating Point Invalid Operation

Floating Point Inexact

Floating Point Overflow

Programming Notes:

レジスタの値を整数フォーマットに変換する .

FLOOR.W.fmt (Floating Point Floor Convert to Word Fixed Point) : 浮動小数点フォーマット変換

31	26	25	21	20	16	15	11	10	6	5	0	
010001			fmt		00000		fs		fd		001111	
COP1				0				FLOORW				

Format:

FLOOR.W.S fd, fs (fmt = 10000)

FLOOR.W.D fd, fs (fmt = 10001)

Description:

FPR[fd] ← convert_and_round(FPR[fs])

Exception:

Floating Point Invalid Operation

Floating Point Inexact

Floating Point Overflow

Programming Notes:

レジスタの値を整数フォーマットに変換する .

C.cond.fmt (Floating Point Compare) : 浮動小数点比較

31	26	25	21	20	16	15	11	10	6	5	4	3	0		
010001			fmt			ft		fs		00000			11	cond	
COP1										0			FC		

Format:

C.cond.S fs, ft (fmt = 10000)

C.cond.D fs, ft (fmt = 10001)

Description: $\text{FCR}[31] \leftarrow \text{FPR}[\text{fs}] \text{ compare_cond } \text{FPR}[\text{ft}]$ **Exception:**

Floating Point Invalid Operation

Programming Notes:

浮動小数点の比較を行う。比較結果は浮動小数点制御レジスタのステータスレジスタに格納される。

BC1F (Branch on Floating Point False) : 浮動小数点分岐

31	26	25	21	20	16	15	0
010001	01000		00000		offset		
COP1		BC		BC1F			

Format:

BC1F offset

Description:if $C(\text{FCR}[31]) = 0$ then branch**Exception:**

None

Programming Notes:

FCR[31] の Condition フィールドが 0 の場合，指定されたアドレスに分岐する。

BC1T (Branch on Floating Point True) : 浮動小数点分岐

31	26	25	21	20	16	15	0
010001	01000	00001	offset				
COP1	BC	BC1T					

Format:

BC1T offset

Description:if $C(FCR[31]) = 1$ then branch**Exception:**

None

Programming Notes:

FCR[31] の Condition フィールドが 1 の場合、指定されたアドレスに分岐する。

4

Exception

4.1 発生する例外

表 4.1: 発生する例外と例外コード

例外コード	名称	詳細
00	外部割り込み	外部の IO デバイスからのイベント通知
01	ITLB ミス	ITLB でミスが発生
02	ITLB 保護例外	命令アクセスで保護違反が発生
03	DTLB ミス	DTLB でミスが発生
04	DTLB 保護例外	データアクセスで保護違反が発生
05	アドレスミスアライン	アラインされていないアドレスへのアクセス
06	予約済み	
07	予約済み	
08	システムコール	syscall 命令の実行
09	ブレークポイント	break 命令の実行
10	予約命令	未定義及び未実装命令実行
11	特権違反	実行モードに許可されていない命令の実行
12	算術オーバーフロー	算術演算においてオーバーフローが発生
13	0 割例外	除数 0 で除算命令を実行
14	FPU 例外	浮動小数点演算器で例外が発生
15	No Exp	No Exception

例外が発生した場合 Exception Code レジスタに例外コード, Exception Program Counter レジスタに例外が発生した命令のアドレスを格納し, CPU は割り込み禁止状態及びカーネルモードになり Exception Vector のアドレスに制御が移る.

4.2 割り込み

表 4.2: IRQ の割り当て

IRQ 番号	割り当て
00	Non Maskable Interrupt
01	デコードエラー
02	ウォッチドッグタイマエラー
03	タイマ CH 0
04	タイマ CH 1
05	タイマ CH 2
06	タイマ CH 3
07	UART CH 0
08	UART CH 1
09	GPIO
10	I2C
11	SPI
12	RTC
13	External Bus IRQ 0
14	External Bus IRQ 1
15	PWM Out CH 0
16	PWM Out CH 1
17	PWM Out CH 2
18	PWM Out CH 3
19	PWM Out CH 4
20	PWM Out CH 5
21	PWM In CH 0
22	PWM In CH 1
23	Pulse Counter CH 0
24	Pulse Counter CH 1
25	DMAC CH 0
26	DMAC CH 1
27	<i>Reserved</i>
28	<i>Reserved</i>
29	Responsive Link
30	PCI
31	Software IRQ

外部割り込みが発生した場合，Interrupt Sense レジスタ内の割り込みを発生させたデバイスに対応する bit が 1 にセットされる。

5

CPU Control Register

5.1 アドレスマップ

表 5.1: 制御レジスタのアドレスマップ

レジスタ番号	名称	アクセス	詳細
00	Cause	RW	例外発生時にその原因情報を格納する
01	Status	RW	プロセッサのステータスを格納する
02	Pending	RW	ペンディングされている割り込みに関する情報を格納する
03	Interrupt Mask	RW	割り込みマスクを格納する
04	Exception PC	RW	例外プログラムカウンタ
05	Exception Vector	RW	例外ベクタのアドレス
06	Invalid Address	RW	インバリデーションするアドレスを格納
07	Cache	RW	キャッシュを設定する
08	IRQ Polarity	RW	IRQ の極性を格納する
09	IRQ Trigger	RW	IRQ のトリガ条件を格納する
10	Software IRQ	RW	ソフトウェア IRQ
11	Thread ID	RW	スレッド ID
12	Invalid Global Address	RW	インバリデーションするグローバルアドレスを格納
13 ~ 25	Reserved	NA	Reserved
26	Clock Counter Low	R	クロックカウンタの下位 32 ビット
27	Clock Counter High	R	クロックカウンタの上位 32 ビット
28	Instruction Counter Low	R	命令カウンタの下位 32 ビット
29	Instruction Counter High	R	命令カウンタの上位 32 ビット
30	Global Address	R	グローバルアドレス
31	CPU Information	R	CPU のバージョン情報

5.2 ビットマップ

Cause

Register Number: 0

[illegible]

31	Delay Bit (D)	Default: 0x0	
ディレイスロットで例外が発生した場合，1 がセットされる．			
3-0	Exception Code (EXP_CODE)	Default: 0x0	
例外が発生した場合，例外コードがセットされる．			

Status

Register Number: 1

31 30 29																		9 8 7 6 5 4 3 2 1 0				
ICE	DCE	Reserved															MODEo	IEo	MODEp	IEp	MODEc	IEc

31	I-Cache Enable (ICE)	Default: 0x1
このビットを 0 にした場合、I-Cache がオンになる。		
30	D-Cache Enable (DCE)	Default: 0x1
このビットを 0 にした場合、D-Cache がオンになる。		
8-7	Execution Mode Old (MODEo)	Default: 0x0
保存した実行モード (2)。 例外が発生した場合、MODEp の値がこのレジスタに格納される。		
6	Interrupt Enable Old (IEo)	Default: 0x0
保存した割り込み有効ビット (2)。 例外が発生した場合、IEp の値がこのレジスタに格納される。		
5-4	Execution Mode Previous (MODEp)	Default: 0x0
保存した実行モード (3)。 例外が発生した場合、MODEc の値がこのレジスタに格納される。 ERET 命令を実行した場合、MODEo の値がこのレジスタに格納される。		
3	Interrupt Enable Previous (IEp)	Default: 0x0
保存した割り込み有効ビット (3)。 例外が発生した場合、IEc の値がこのレジスタに格納される。 ERET 命令を実行した場合、IEo の値がこのレジスタに格納される。		
2-1	Execution Mode Current (MODEc)	Default: 0x0
現在の実行モード。 例外が発生した場合、実行モードは Kernel Mode にセットされる。 Kernel Mode : 2'b00, Supervisor Mode : 2'b01, User Mode : 2'b10, Reserved : 2'b11		
0	Interrupt Enable Current (IEc)	Default: 0x0
現在の割り込み有効ビット。 このビットが 1 の場合、割り込みが発生する。各 IRQ の設定は割り込みコントローラ (後述) の項を参照。 例外が発生した場合、割り込み有効ビットは 0 がセットされる。		

Pending

Register Number: 2

31	0
Pending IRQ	

31:0	Pending IRQ	Default: 0x0
------	-------------	--------------

このレジスタの N ビット目が IRQ の N チャンネルに対応する .
このレジスタを読みだした場合 , IRQ の情報が読み出せる . IRQ がペンディングされている場合は 1 が読み出せる .
このレジスタへ 1 を書き込んだ場合 , 対応するビットの IRQ をクリアする .

Interrupt Mask

Register Number: 3

31	1	0
INT_MASK		0

7:1	Interrupt Mask (INT_MASK)	Default: 0x7F
-----	---------------------------	---------------

このレジスタの N ビット目が IRQ の N チャンネルに対応する .
対応するビットが 1 の場合 , 割り込みがマスクされる .

0	Interrupt Mask (INT_MASK)	Default: 0x0
---	---------------------------	--------------

IRQ0 番はマスク出来ない . (Non Maskable Interrupt : NMI)

Exception Program Counter

Register Number: 4

31	2	1	0
EPC			00

31:2	Exception Program Counter (EPC)	Default: 0x0
------	---------------------------------	--------------

例外が発生した命令のアドレスが格納される .

Exception Vector

Register Number: 5

31	2	1	0
EXP_VECTOR			00

31:2	Exception Vector (EXP_VECTOR)	Default: 0x0
------	-------------------------------	--------------

例外が発生した場合 , このレジスタにセットされたアドレスに制御が移る .

Invalid Address

Register Number: 6

31	0
INV_ADDR	

31:0	Invalid Address (INV_ADDR)	Default: 0x0
------	----------------------------	--------------

キャッシュを無効化するアドレスを格納する。

Cache Control

Register Number: 7

31						5	4	3	2	1	0
<i>Reserved</i>						I_INIT	I_INV	D_INIT	D_INV	D_SYNC	

4	I-Cache Initialize (I_INIT)	Default: 0x1
---	-----------------------------	--------------

このビットを 0 にした場合、I-Cache が初期化される。I-Cache を使うには、このビットを 0 にしてから Status Register の I-Cache Enable のビットを 0 にする。

3	I-Cache Invalid (I_INV)	Default: 0x1
---	-------------------------	--------------

このビットを 0 にした場合、I-Cache が無効化される。

2	D-Cache Initialize (D_INIT)	Default: 0x1
---	-----------------------------	--------------

このビットを 0 にした場合、D-Cache が初期化される。D-Cache を使うには、このビットを 0 にしてから Status Register の D-Cache Enable のビットを 0 にする。

1	D-Cache Invalid (D_INV)	Default: 0x1
---	-------------------------	--------------

このビットを 0 にした場合、D-Cache が無効化される。

0	D-Cache Sync (D_SYNC)	Default: 0x1
---	-----------------------	--------------

このビットを 0 にした場合、D-Cache が同期される。

IRQ Polarity

Register Number: 8

31	0
IRQ_POLARITY	

31:0	IRQ Polarity (IRQ_POLARITY)	Default: 0xffffffff
------	-----------------------------	---------------------

このレジスタの N ビット目が IRQ の N チャンネルに対応する。

対応するビットに 0 がセットされている場合、その IRQ が Low または立ち下がった場合 (IRQ Trigger の設定による)、割り込みを検知する。

対応するビットに 1 がセットされている場合、その IRQ が High または立ち上がった場合割り込みを検知する。

IRQ Trigger

Register Number: 9

31

0

IRQ_TRIGGER

31:0	IRQ Trigger (IRQ_TRIGGER)	Default: 0x0
------	---------------------------	--------------

このレジスタの N ビット目が IRQ の N チャンネルに対応する。
 対応するビットに 0 がセットされている場合、その IRQ はレベルセンシティブになる。
 対応するビットに 1 がセットされている場合、その IRQ はエッジセンシティブになる。

Software IRQ

Register Number: 10

31

0

SW_IRQ

31:0	Software IRQ (SW_IRQ)	Default: 0x0
------	-----------------------	--------------

このレジスタの N ビット目が SW IRQ の N チャンネルに対応する。
 対応するビットに 1 がセットされている場合、ソフトウェア割り込みが発生する。

Thread ID

Register Number: 11

31

0

TH_ID

31:0	Thread ID (TH_ID)	Default: 0x0
------	-------------------	--------------

現在実行しているスレッドの ID を格納する。

Invalid Global Address

Register Number: 12

31

0

INV_GADDR

31:0	Invalid Global Address (INV_GADDR)	Default: 0x0
------	------------------------------------	--------------

キャッシュを無効化するグローバルアドレスを格納する。

Clock Counter

Register Number: 26, 27

31	0
CLK_CNT	

31:0	Clock Counter (CLK_CNT)	Default: 0x0
------	-------------------------	--------------

リセットしてから現在までのクロック経過数が格納される。
レジスタ 26 番に下位 32 ビット、レジスタ 27 番に上位 32 ビットが格納される。

Instruction Counter

Register Number: 28, 29

31	0
INSN_CNT	

31:0	Instruction Counter (INSN_CNT)	Default: 0x0
------	--------------------------------	--------------

リセットしてから現在までに実行した命令数が格納される。レジスタ 28 番に下位 32 ビット、レジスタ 29 番に上位 32 ビットが格納される。

Global Address

Register Number: 30

31	24	23	8	7	0
PRIORITY		CHIP_ID			NODE_ID

31:24	Priority (PRIORITY)	Default: 0x0
-------	---------------------	--------------

自ノードの優先度。

23:8	Chip ID (CHIP_ID)	Default: 0x1
------	-------------------	--------------

自身のチップ ID。

7:0	Node ID (NODE_ID)	Default: 0x0
-----	-------------------	--------------

自身のノード ID。

CPU Information																Register Number: 31															
31				24				23				16				15				8				7				0			
Year								Month								Version								Revision							
31:24				Year								Default: NaN																			
リリースされた年を表す．このレジスタ値に 2000 を足したものが西暦となる．																															
23:16				Month								Default: 1 ~ 12																			
リリースされた月を表す．																															
15:8				Version								Default: NaN																			
リリースされたバージョンを表す．																															
7:0				Revision								Default: NaN																			
リリースされたリビジョンを表す．																															

6

FPU

6.1 制御レジスタ

6.1.1 アドレスマップ

表 6.1: 制御レジスタのアドレスマップ

レジスタ番号	名称	アクセス	詳細
0 ~ 30	Reserved	NA	Reserved
31	Status	RW	FPU のステータスを格納する

6.1.2 ビットマップ

Status

Register Number: 31

312423221716121176210

ReservedCReservedCAUSEENFLAGRM

23

Condition (C)

Default: 0x0

比較命令の結果がセットされる。

16-12

Exception Cause (CAUSE)

Default: 0x00

例外が発生した場合，例外コードがセットされる．対応する例外のビットマップは次節に示す．

11-7

Exception Enable (EN)

Default: 0x00

対応するビットが 1 となっている例外の発生を有効にする．対応する例外のビットマップは次節に示す．

6-2

Exception Flags (FLAG)

Default: 0x00

発生した例外に対応するビットがセットされる．対応するビットが 1 となっている例外の発生は禁止される．対応する例外のビットマップは次節に示す．

1-0

Round Mode (RM)

Default: 0x0

丸めモードを指定する．

設定値丸めモード

2'b00Round to Nearest

2'b01Round to Zero

2'b10Round Ceiling

2'b11Round Floor

6.2 例外

浮動小数点命令で発生する例外は下記のとおりである . 下記のビットマップは Status レジスタ内の例外に関連するフィールドに対応している .

			4	3	2	1	0
			v	z	o	u	i
4	Invalid Operation (V)	Default: 0x0					
命令のオペランドが無効							
3	Divide By 0 (Z)	Default: 0x0					
除算実行時に除数が 0 かつ被除数が 0 以外							
2	Overflow (O)	Default: 0x0					
オーバーフロー例外							
1	Underflow (U)	Default: 0x0					
アンダーフロー例外							

0	Inexact (I)	Default: 0x0
丸めによる不正確な演算結果		

7

Memory Management Unit

7.1 MMU の概要

IO Companion Chip に搭載されているプロセッサでは、命令・データそれぞれに用意された 16 エントリの TLB によってメモリ管理を行う。TLB に格納するページテーブルのフォーマットは、以下の様になっている。

Entry High

Register Number: 1

31

0

TH_ID

31-0	Thread ID (TH_ID)	Default: 0x0
------	-------------------	--------------

スレッド ID

本レジスタにセットされている ID のスレッドのアドレスを変換する。Entry Middle レジスタにあるグローバルビットがセットされている場合、スレッド ID に関係なくアドレス変換を行うため、本フィールドは意味をなさない。

Entry Middle

Register Number: 2

31	29	28					9	8		6	5		3	2	1	0
Reserved			VPN						PSIZE		PROT		G	UC	V	

28-9	Virtual Page Number (VPN)	Default: 0x0
仮想ページ番号 アクセスするアドレスと本フィールドの仮想ページ番号がマッチした場合、アドレス変換を行う。デフォルトでは仮想ページ番号は 20 ビットであるが、ページサイズフィールドの設定により、比較するビット幅が変化する。		

8-6	Page Size (PSIZE)	Default: 0x0																											
ページサイズの設定 <table border="1"> <thead> <tr> <th>設定値</th><th>ページサイズ</th><th>比較するアドレスフィールド</th></tr> </thead> <tbody> <tr><td>3'b000</td><td>4KB</td><td>BYTE_ADDRESS[31:12]</td></tr> <tr><td>3'b001</td><td>64KB</td><td>BYTE_ADDRESS[31:16]</td></tr> <tr><td>3'b010</td><td>1MB</td><td>BYTE_ADDRESS[31:20]</td></tr> <tr><td>3'b011</td><td>16MB</td><td>BYTE_ADDRESS[31:24]</td></tr> <tr><td>3'b100</td><td>128MB</td><td>BYTE_ADDRESS[31:27]</td></tr> <tr><td>3'b101</td><td>256MB</td><td>BYTE_ADDRESS[31:28]</td></tr> <tr><td>3'b110</td><td>512MB</td><td>BYTE_ADDRESS[31:29]</td></tr> <tr><td>3'b111</td><td>1GB</td><td>BYTE_ADDRESS[31:30]</td></tr> </tbody> </table>			設定値	ページサイズ	比較するアドレスフィールド	3'b000	4KB	BYTE_ADDRESS[31:12]	3'b001	64KB	BYTE_ADDRESS[31:16]	3'b010	1MB	BYTE_ADDRESS[31:20]	3'b011	16MB	BYTE_ADDRESS[31:24]	3'b100	128MB	BYTE_ADDRESS[31:27]	3'b101	256MB	BYTE_ADDRESS[31:28]	3'b110	512MB	BYTE_ADDRESS[31:29]	3'b111	1GB	BYTE_ADDRESS[31:30]
設定値	ページサイズ	比較するアドレスフィールド																											
3'b000	4KB	BYTE_ADDRESS[31:12]																											
3'b001	64KB	BYTE_ADDRESS[31:16]																											
3'b010	1MB	BYTE_ADDRESS[31:20]																											
3'b011	16MB	BYTE_ADDRESS[31:24]																											
3'b100	128MB	BYTE_ADDRESS[31:27]																											
3'b101	256MB	BYTE_ADDRESS[31:28]																											
3'b110	512MB	BYTE_ADDRESS[31:29]																											
3'b111	1GB	BYTE_ADDRESS[31:30]																											

5-3	Protection Mode (PROT)	Default: 0x0																											
保護モードの設定 <table border="1"> <thead> <tr> <th>設定値</th><th>保護モード</th><th>意味</th></tr> </thead> <tbody> <tr><td>3'b111</td><td>All Read Only</td><td>全てのモードで読み出し可能。</td></tr> <tr><td>3'b110</td><td>All Read Write</td><td>全てのモードで読み書き可能。</td></tr> <tr><td>3'b101</td><td>User Read Write</td><td>ユーザモード以上で読み書き可能。</td></tr> <tr><td>3'b100</td><td>User Read Only</td><td>ユーザモードで読み出し可能。スーパーバイザモード以上で読み書き可能。</td></tr> <tr><td>3'b011</td><td>Supervisor Read Write</td><td>スーパーバイザモード以上で読み書き可能。</td></tr> <tr><td>3'b010</td><td>Supervisor Read Only</td><td>スーパーバイザモードで読み出し可能。カーネルモードで読み書き可能。</td></tr> <tr><td>3'b001</td><td>Kernel Read Write</td><td>カーネルモードで読み書き可能。</td></tr> <tr><td>3'b000</td><td>Kernel Read Only</td><td>カーネルモードで読み出し可能。</td></tr> </tbody> </table>			設定値	保護モード	意味	3'b111	All Read Only	全てのモードで読み出し可能。	3'b110	All Read Write	全てのモードで読み書き可能。	3'b101	User Read Write	ユーザモード以上で読み書き可能。	3'b100	User Read Only	ユーザモードで読み出し可能。スーパーバイザモード以上で読み書き可能。	3'b011	Supervisor Read Write	スーパーバイザモード以上で読み書き可能。	3'b010	Supervisor Read Only	スーパーバイザモードで読み出し可能。カーネルモードで読み書き可能。	3'b001	Kernel Read Write	カーネルモードで読み書き可能。	3'b000	Kernel Read Only	カーネルモードで読み出し可能。
設定値	保護モード	意味																											
3'b111	All Read Only	全てのモードで読み出し可能。																											
3'b110	All Read Write	全てのモードで読み書き可能。																											
3'b101	User Read Write	ユーザモード以上で読み書き可能。																											
3'b100	User Read Only	ユーザモードで読み出し可能。スーパーバイザモード以上で読み書き可能。																											
3'b011	Supervisor Read Write	スーパーバイザモード以上で読み書き可能。																											
3'b010	Supervisor Read Only	スーパーバイザモードで読み出し可能。カーネルモードで読み書き可能。																											
3'b001	Kernel Read Write	カーネルモードで読み書き可能。																											
3'b000	Kernel Read Only	カーネルモードで読み出し可能。																											

2	Global Bit (G)	Default: 0x0
グローバルビット 本ビットが設定されている場合、スレッド ID に関係なくアドレスの比較を行う。		
1	Uncache (UC)	Default: 0x0
アンキャッシュビット 本ビットが設定されている場合、該当ページはキャッシュされない。		
0	Valid Bit (V)	Default: 0x0
有効ビット 本ビットが設定されている場合、TLB のエントリが有効である。		

Entry Low

Register Number: 3

31	20	19	0
Reserved			PPN

19-0	Physical Page Number (PPN)	Default: 0x0
------	----------------------------	--------------

物理ページ番号

TLB がヒットした場合、仮想ページ番号を物理ページ番号に変換する。

Entry Global

Register Number: 8

31	24	23	8	7	0
PRIORITY			CHIP_ID		NODE_ID

31-24	Priority (PRIORITY)	Default: 0x0
-------	---------------------	--------------

対象のノードの優先度

23-8	Chip ID (CHIP_ID)	Default: 0x0
------	-------------------	--------------

チップ ID

本レジスタにセットされているチップ ID を持つチップ内ノードのアドレスを変換する。自チップとは異なるチップ ID のアドレスにアクセスする場合、外部バスのアドレスにアクセスする。

7-0	Node ID (NODE_ID)	Default: 0x0
-----	-------------------	--------------

ノード ID

本レジスタにセットされているノード ID を持つノードのアドレスを変換する。

7.2 MMU の制御

MMU の制御は以下の命令を使用して行う。

mfimm	I-MMU の制御レジスタの読み出し
mtimm	I-MMU の制御レジスタへの書き込み
mfdmm	D-MMU の制御レジスタの読み出し
mtdmm	D-MMU の制御レジスタへの書き込み

7.3 制御レジスタ

7.3.1 アドレスマップ

表 7.1: 制御レジスタのアドレスマップ

名称	オフセット	アクセス	詳細
Enable	00	R/W	TLB 有効
Entry High	01	R/W	エントリー High
Entry Middle	02	R/W	エントリー Middle
Entry Low	03	R/W	エントリー Low
Index	04	R/W	インデックス
Lock	05	R/W	ロック
Exception Code	06	R/W	例外コード
Exception Address	07	R/W	例外アドレス
Entry Global	08	R/W	エントリー Global
<i>Reserved</i>	9~15	NA	Reserved
Read Index	16	R/W	インデックスによる読み出し
Write Index	17	R/W	インデックスによる書き込み
Write Random	18	R/W	ランダム書き込み
TLB Probe	19	R/W	TLB のプローブ
<i>Reserved</i>	20~31	NA	Reserved

7.3.2 ビットマップ

EnableRegister Number: 0

31	1	0
Reserved		E

31	Enable (E)	Default: 0x0
このビットに 1 をセットした場合，MMU がオンになる．		

Entry HighRegister Number: 1

31	0
TH_ID	

31-0	Thread ID (TH_ID)	Default: 0x0
スレッド ID		

Entry Middle

Register Number: 2

31	29	28									9	8		6	5		3	2	1	0
<i>Reserved</i>		VPN											PSIZE		PROT		G	_{uc}	V	

28-9	Virtual Page Number (VPN)	Default: 0x0
仮想ページ番号		
8-6	Page Size (PSIZE)	Default: 0x0
ページサイズ		
5-3	Protection Mode (PROT)	Default: 0x0
保護モード		
2	Global Bit (G)	Default: 0x0
グローバルビット		
1	Uncache (UC)	Default: 0x0
アンキャッシュビット		
0	Valid Bit (V)	Default: 0x0
有効ビット		

Entry Low

Register Number: 3

31	20	19	0
<i>Reserved</i>	PPN		

19-0	Physical Page Number (PPN)	Default: 0x0
物理ページ番号		

Index

Register Number: 4

	31	30	4	3	0
	<i>ReservedP</i>				INDEX

31	Probe Hit (P)	Default: 0x0
TLB のプローブを行い，ヒットした場合は 1 が書き込まれ，ミスした場合は 0 が書き込まれる．		
3-0	Index (Index)	Default: 0x0
TLB のインデックスを格納する． インデックスによる書き込みを行う場合，このレジスタに格納されている番号の TLB エントリに書き込みが行われる． TLB のプローブを行い，ヒットした場合はヒットした TLB のエントリ番号がこのレジスタに書き込まれる．		

Lock

Register Number: 5

31	4	3	0
Reserved			LOCK

3-0	Lock Index (LOCK)	Default: 0x0
-----	-------------------	--------------

このレジスタで指定した値未満の TLB エントリは、ランダム置換の対象にならない。

Exception Code

Register Number: 6

30	3	2	0
Reserved			EXP_CODE

3-0	Exception Code (EXP_CODE)	Default: 0x0
-----	---------------------------	--------------

TLB で保護例外が発生した場合、TLB の保護例外コードがセットされる。
CPU で TLB 保護例外が発生した場合、この保護例外コードを読み出し、詳細な情報を得ることが出来る。

Exception Address

Register Number: 7

31	0
EXP_ADDR	

31-0	Exception Address (EXP_ADDR)	Default: 0x0
------	------------------------------	--------------

TLB で保護例外が発生した場合、その際にアクセスしたアドレスが格納される。

Entry Global

Register Number: 1

31	24	23	8	7	0
PRIORITY		CHIP_ID			NODE_ID

31-24	Priority (PRIORITY)	Default: 0x0
-------	---------------------	--------------

優先度

23-8	Chip ID (CHIP_ID)	Default: 0x0
------	-------------------	--------------

チップ ID

7-0	Node ID (NODE_ID)	Default: 0x0
-----	-------------------	--------------

ノード ID

Read Index

Register Number: 16

31	0
READ_INDEX	

31-0	Read Index (READ_INDEX)	Default: 0x0
------	-------------------------	--------------

このレジスタに何か値を書き込んだ場合、TLB エントリの読み出しを行う。
読み出すエントリの番号は Index レジスタで指定する。
読み出した TLB のデータは、TLB Entry レジスタ (High, Middle, Low) に格納される。

Write Index

Register Number: 17

31	0
WRITE_INDEX	

31-0	Write Index (WRITE_INDEX)	Default: 0x0
------	---------------------------	--------------

このレジスタに何か値を書き込んだ場合、TLB エントリへ書き込みを行う。
書き込むエントリの番号は Index レジスタで指定する。
書き込むデータは、TLB Entry レジスタ (High, Middle, Low) に格納されているデータを用いる。

Write Random

Register Number: 18

31	0
WRITE_RANDOM	

31-0	Write Random (WRITE_RANDOM)	Default: 0x0
------	-----------------------------	--------------

このレジスタに何か値を書き込んだ場合、TLB エントリへ書き込みを行う。
書き込むエントリの番号はランダムに選択される。無効な TLB エントリ (有効ビットが 0) が優先的に書き込まれ、ロックされているエントリは除外される。(ただし、無効な TLB エントリの検索には最大でエントリ数分のクロックサイクルを要するため、連続して書き込む場合は注意が必要である)
書き込むデータは、TLB Entry レジスタ (High, Middle, Low) に格納されているデータを用いる。

TLB Probe		Register Number: 19
31	0	
TLB_PROB		
31-0	TLB Probe (TLB_Probe)	Default: 0x0
このレジスタに何か値を書き込んだ場合，TLB エントリのプローブ (検索) を行う。 検索に用いるデータは，TLB Entry レジスタ (High, Middle) に格納されているデータ (スレッド ID と VPN) を用いる。		

8

Bus

8.1 仕様

表 8.1: バスの仕様

バス幅	32bit
バス形態	共有バス（マルチプレックス）
転送方式	クロック同期式
アービトレーション方式	ラウンドロビン
マスターのチャンネル数	16ch
スレーブのチャンネル数	32ch

8.2 バスマスタのマップ

バスマスタは 16 チャンネル接続可能である．以下にバスマスタのマップを示す．

表 8.2: バスマスタのマップ

チャンネル	割り当て
CH 00	CPU 0 (命令フェッチ)
CH 01	CPU 0 (データアクセス)
CH 02	DMA Controller 0
CH 03	DMA Controller 1
CH 04	<i>Reserved</i>
CH 05	<i>Reserved</i>
CH 06	External Bus 0
CH 07	<i>Reserved</i>
CH 08	<i>Reserved</i>
CH 09	<i>Reserved</i>
CH 10	PCI Controller
CH 11	<i>Reserved</i>
CH 12	<i>Reserved</i>
CH 13	<i>Reserved</i>
CH 14	<i>Reserved</i>
CH 15	On-Chip Emulator

8.3 バススレーブのアドレスマップ

バススレーブは 32 チャンネル接続可能である．以下にバススレーブのマップを示す．

表 8.3: バススレーブのマップ

チャンネル	デフォルトのアドレス	割り当て
CH 00	0x0000_0000 ~ 0x1FFF_FFFF	Extenal Bus
CH 01	-	<i>Reserved</i>
CH 02	-	<i>Reserved</i>
CH 03	-	<i>Reserved</i>
CH 04	0x2000_0000 ~ 0x27FF_FFFF	タイマ
CH 05	0x2800_0000 ~ 0x2FFF_FFFF	クロックモジュール
CH 06	0x3000_0000 ~ 0x37FF_FFFF	Legacy IO
CH 07	0x3800_0000 ~ 0x3FFF_FFFF	Control IO
CH 08	0x4000_0000 ~ 0x47FF_FFFF	SRAM
CH 09	0x4800_0000 ~ 0x4FFF_FFFF	DMA Controller
CH 10	0x5000_0000 ~ 0x57FF_FFFF	PCI Controller
CH 11	-	<i>Reserved</i>
CH 12	-	<i>Reserved</i>
CH 13	-	<i>Reserved</i>
CH 14	-	<i>Reserved</i>
CH 15	0x7800_0000 ~ 0x7FFF_FFFF	On-Chip Emulator
CH 16	0x8000_0000 ~ 0xA7FF_FFFF	Responsive Link
CH 17	-	<i>Reserved</i>
CH 18	-	<i>Reserved</i>
CH 19	-	<i>Reserved</i>
CH 20	-	<i>Reserved</i>
CH 21	-	<i>Reserved</i>
CH 22	-	<i>Reserved</i>
CH 23	-	<i>Reserved</i>
CH 24	-	<i>Reserved</i>
CH 25	-	<i>Reserved</i>
CH 26	-	<i>Reserved</i>
CH 27	-	<i>Reserved</i>
CH 28	-	<i>Reserved</i>
CH 29	0xE800_0000 ~ 0xEFFF_FFFF	Link SDRAM DQS Delay
CH 30	0xF000_0000 ~ 0xF7FF_FFFF	External Bus Controller
CH 31	0xF800_0000 ~ 0xFFFF_FFFF	Internal Bus Controller

表 8.4: Legacy IO のマップ

チャンネル	デフォルトのアドレス	割り当て
CH 06-00	0x3000_0000 ~ 0x30FF_FFFF	UART
CH 06-01	0x3100_0000 ~ 0x31FF_FFFF	GPIO
CH 06-02	0x3200_0000 ~ 0x32FF_FFFF	SPI
CH 06-03	0x3300_0000 ~ 0x33FF_FFFF	I2C
CH 06-04	0x3400_0000 ~ 0x34FF_FFFF	<i>Reserved</i>
CH 06-05	0x3500_0000 ~ 0x35FF_FFFF	<i>Reserved</i>
CH 06-06	0x3600_0000 ~ 0x36FF_FFFF	<i>Reserved</i>
CH 06-07	0x3700_0000 ~ 0x37FF_FFFF	<i>Reserved</i>

表 8.5: Control IO のマップ

チャンネル	デフォルトのアドレス	割り当て
CH 07-00	0x3800_0000 ~ 0x38FF_FFFF	PWM Generator
CH 07-01	0x3900_0000 ~ 0x39FF_FFFF	PWM Input
CH 07-02	0x3A00_0000 ~ 0x3AFF_FFFF	Pulse Counter
CH 07-03	0x3B00_0000 ~ 0x3BFF_FFFF	Real-Time Clock
CH 07-04	0x3C00_0000 ~ 0x3CFF_FFFF	<i>Reserved</i>
CH 07-05	0x3D00_0000 ~ 0x3DFF_FFFF	<i>Reserved</i>
CH 07-06	0x3E00_0000 ~ 0x3EFF_FFFF	<i>Reserved</i>
CH 07-07	0x3F00_0000 ~ 0x3FFF_FFFF	<i>Reserved</i>

8.4 バスコントローラ

8.4.1 アドレスマップ

ベースアドレス : 0xf8000000

表 8.6: 制御レジスタのアドレスマップ

オフセット	名称	アクセス	詳細
0x0000	Status	RW	バスのステータスレジスタ
0x0004	Arbitration Policy	RW	バス調停のポリシ
0x0008	Watch Dog Timer Enable	RW	ウォッチドッグタイマの有効化
0x000C	Watch Dog Timer Expire Value	RW	ウォッチドッグタイマの設定値
0x0010	Last Access Address	RW	最後にアクセスしたアドレス
0x0014	Error Address	RW	エラー発生時のアドレス
0x4000 ~ 0x7FFC	Bus Master Priority	RW	バスマスタの調停優先度
0x8000 ~ 0xBFFC	Bus Slave Address Map	RW	バススレーブのアドレスマップ

8.4.2 ビットマップ

Status

offset: 0x0000

[illegible]

1	Watch Dog Timer Expired (WD)	Default: 0x0
ウォッチドッグタイマが満了した場合にセットされる。		
0	Decode Error (DEC)	Default: 0x0
デコードエラーが発生した場合にセットされる。		

Arbitration Policy

offset: 0x0004

31	1	0
<i>Reserved</i>		P

0	Arbitration Policy (P)	Default: 0x0
バス調停のポリシーを設定する．0 をセットした場合，ラウンドロビンで調停を行う．1 をセットした場合，固定優先度で調停を行う．		

Watch Dog Timer Enable

offset: 0x0008

31		1	0
Reserved			WDR

0	Watch Dog Timer Enable (WDE)	Default: 0x0
本ビットをセットするとウォッチドッグタイマが有効になる。		

Watch Dog Timer Expire Value

offset: 0x000C

31	WD_EXPR_VAL	0
----	-------------	---

31:0	Watch Dog Timer Expire Value (WD_EXPR_VAL)	Default: 0x0
------	---	---------------------

ウォッチドッグタイマの満了値をセットする。バスアクセス後に、本レジスタにセットしたクロックサイクル以内にレディが帰ってこなかった場合、ウォッチドッグタイマエラーが発生する。

Last Access Address	offset: 0x0010
---------------------	----------------

offset: 0x0010

0

LA_ADDR

31:0	Last Access Address (LA_ADDRESS)	Default: 0x0
------	----------------------------------	---------------------

最後にアクセスしたアドレスが記録される。

Error Address	offset: 0x0014
---------------	----------------

offset: 0x0014

0

ERROR_ADDR

31:0	Error Address (ERROR_ADDRESS)	Default: 0x0
------	-------------------------------	---------------------

ウォッチドッグタイマエラーが発生したアドレスが記録される。

Bus Master Arbitration Priority_{offset: 0x4000~0x7FFC}

offset: 0x4000~0x7FFC

0

<i>Reserved</i>	PRIORITY
-----------------	----------

7:0	Arbitration Priority (PRIORITY)	Default: 0x0
-----	---------------------------------	---------------------

固定優先度でバス調停を行う場合の優先度を設定する．

Bus Slave Address Map

offset: 0x8000~0xBFEC

0

ADDR_MAP

7:0	Address Map (ADDR_MAP)	Default: 0x0
-----	------------------------	---------------------

バススレーブのアドレスマップを設定する。

9

External Bus

9.1 仕様

表 9.1: 外部バスの仕様

バス幅	32bit
バス形態	共有バス（マルチプレックス）
転送方式	クロック同期式（チップ内クロック同期，外部クロック同期），非同期
チャンネル数	4ch

9.1.1 外部バスのチャンネル切り替え

外部バスには 4 チャンネルの空間が接続されており，接続するチャンネルは外部バスコントローラの External Channel レジスタで指定する．各チャンネルのアドレスは共通であり，External Channel レジスタの値によって切り替わる．

9.2 外部バスコントローラ

9.2.1 アドレスマップ

ベースアドレス：0xf0000000

表 9.2: 制御レジスタのアドレスマップ

オフセット	名称	アクセス	詳細
0x0	Status	RW	外部バスのステータスレジスタ
0x4	Auto Ready Time	RW	オートレディの間隔
0x8	External Channel	RW	外部バスのチャンネル

9.2.2 ビットマップ

Status offset: 0x0

31											9	8	7	6	5	4	3	2	1	0
Reserved											BR	TGL	ARE	SYN	CNCT	END	SIZE			
8	Burst (BR)										Default: 0x1									
このビットが 0 の場合，バーストを有効にする．																				
7	Chip Select Toggle (TGL)										Default: 0x1									
このビットが 0 の場合，チャンネル 0 とチャンネル 1 の空間が入れ替わる．																				
6	Auto Ready Enable (ARE)										Default: 0x1									
このビットが 0 の場合，オートレディが有効になる．																				
5	Sync (SYN)										Default: 0x0									
外部バスのクロック同期方式を指定する． このビットが 0 の場合，チップ内クロック同期になる．1 の場合，外部クロック同期になる．																				
4:3	Connect (CNCT)										Default: 0x0									
バスサイジングを有効にする際に，データ内で有効となるバイトを指定する． 下記の表はビッグエンディアンの場合に，32bit のデータ data[31:0] に対して有効となるバイトの範囲を示す．																				
設定値	16bit 幅										8bit 幅									
2'b00	data[31:16]										data[31:24]									
2'b01	data[31:16]										data[23:16]									
2'b10	data[15:0]										data[15:8]									
2'b11	data[15:0]										data[7:0]									
2	Endian (END)										Default: 0x0									
外部バスアクセス時のエンディアンを指定する． このビットが 0 の場合，ビッグエンディアンになる．1 の場合，リトルエンディアンになる．																				
1:0	Size (SIZE)										Default: 0x0									
バスサイジングのビット幅を指定する．																				
設定値	ビット幅																			
2'b00	32bit																			
2'b01	16bit																			
2'b10	8bit																			

Auto Ready Time

offset: 0x4

31	0
AR_TIME	

31:0	Auto Ready Time (AR_TIME)	Default: 0x20
------	---------------------------	---------------

オートレディの間隔を設定する．単位はクロックサイクル．

External Channel

offset: 0x8

31	2	1	0
Reserved			EXT_CH

1:0	External Channel (EXT_CH)	Default: 0x0
-----	---------------------------	--------------

接続する外部バスのチャンネルを指定する．外部バスのアドレス空間にアクセスした場合，このレジスタで指定したチャンネルの空間にアクセスする．

10

Timer

10.1 概要

タイマは一定時間毎に割り込みを発生させるユニットである．タイマは CPU と同じクロックで動作する．ワード長以外でのアクセスはサポートしない．

10.2 制御レジスタ

10.2.1 アドレスマップ

ベースアドレス CH 0: 0x2000_0000

ベースアドレス CH 1: 0x2000_0010

ベースアドレス CH 2: 0x2000_0020

ベースアドレス CH 3: 0x2000_0030

表 10.1: 制御レジスタのアドレスマップ

名称	オフセット	アクセス	詳細
Control	0x0	R/W	コントロール
Interrupt	0x4	R/W	割り込み
Expiration	0x8	R/W	満了値
Counter	0xC	R/W	カウンタ

10.2.2 ビットマップ

Control : コントロール

Access : Read / Write

Offset: 0x0

31	2	1	0
<i>Reserved</i>			S

1	Periodic (P)	Default: 0x0	
この bit が 0 の場合タイマはワンショットタイマとして動作し、この bit が 1 の場合タイマはピリオディックタイマとして動作する。			
0	Start (S)	Default: 0x0	
この bit に 1 がセットされた場合タイマが動作する。			

Interrupt : 割り込み

Access : Read / Write

Offset: 0x4

31	1	0
<i>Reserved</i>		I

2	Interrupt (I)	Default: 0x0
タイマが満了すると自動的にセットされる．この bit が 1 の場合割り込みが発生する．		

Expiration : 満了値

Access : Read / Write

Offset: 0x8

31	0
EXPR	

31:0	Expiration (EXPR)	Default: 0x0
------	-------------------	---------------------

カウンタがこの値に達した場合に割り込みを発生させる。

Counter : カウンタ

Access : Read / Write

Offset: 0xC

31		0
COUNTER		

8	Counter (COUNTER)	Default: 0x0
---	-------------------	--------------

カウンタが満了値に達した場合に割り込みを発生させる。

11

Clock Generator

11.1 概要

クロックジェネレータは各モジュールに分配されるクロックの分周比を変更する．また，クロックゲーティングおよびソフトウェアリセットを行う．

11.2 制御レジスタ

クロックジェネレータの初期ベースアドレスは 0x28000000 である．

各モジュールのクロック分周比，クロックゲーティング，およびソフトウェアリセットを操作するには，そのモジュールの属するクロックラインのオフセットを指定する必要がある．各クロックラインのオフセットを表 11.1 に示す．

クロック分周を行うには，制御レジスタ Clock Divider のオフセット 0x00 をクロックラインのオフセットに足し，分周比を指定する．すなわち，クロック分周はクロックライン単位で行われる．

クロックゲーティングを行うには，制御レジスタ Clock Gating のオフセット 0x04 をクロックラインのオフセットに足し，さらにモジュール#に対応するビットに 1 をセットする．すなわち，クロックゲーティングはモジュール単位で行われる．モジュールの名称とその#の対応は表 11.1 の通りである．例えば，PWM OUT 0 をクロックゲーティングしたい場合は， $0x28000000 + 0x600 + 0x04 = 0x28000604$ に 0x2 を書き込む．

ソフトウェアリセットを行うには，対象となるモジュールの制御レジスタ Reset のオフセットをクロックラインのオフセットに足す．すなわち，ソフトウェアリセットはモジュール単位で行われる．対象とするモジュールの#とそのオフセットの対応は表 11.2 の通りである．例えば，PWM OUT 3 にソフトウェアリセットをかける場合は， $0x28000000 + 0x600 + 0x90 = 0x28000690$ の 17 ビットに 0 を書き込む (Active-Low) ．

11.2.1 アドレスマップ

表 11.1: 各クロックラインのオフセット

クロックライン		モジュール	
名称	オフセット	#	名称

表 11.1: 各クロックラインのオフセット

Main	0x000	0	CPU
		1	Bus
		2	SRAM
		3	Link Bus
		4	DMAC 0
		5	DMAC 1
		6	DMAC 2
		7	DMAC 3
Timer	0x100	0	Timer Bus
		1	Timer 0
		2	Timer 1
		3	Timer 2
		4	Timer 3
UART	0x200	0	UART Bus
		1	UART 0
		2	UART 1
GPIO	0x300	0	GPIO Bus
		1	GPIO
SPI	0x400	0	SPI Bus
		1	SPI
I2C	0x500	0	I2C Bus
		1	I2C
PWM OUT	0x600	0	PWM OUT Bus
		1	PWM OUT 0
		2	PWM OUT 1
		3	PWM OUT 2
		4	PWM OUT 3
		5	PWM OUT 4
		6	PWM OUT 5
PWM IN	0x700	0	PWM IN Bus
		1	PWM IN 0
		2	PWM IN 1
Pulse Counter	0x800	0	Pulse Counter Bus
		1	Pulse Counter 0
		2	Pulse Counter 1
PCI	0x900	0	PCI Bus
		1	PCI
Link IO Clock	0xa00	0	Link IO
Link COM Clock	0xb00	0	Link COM
Link SDRAM	0xc00		Link SDRAM 1X
		0	Link SDRAM X1X

表 11.1: 各クロックラインのオフセット

		Link SDRAM 2X
		Link SDRAM X2X

表 11.2: リセットにおけるモジュールのオフセット

#	オフセット
0	0x80
1	0x84
2	0x88
3	0x8c
4	0x90
5	0x94
6	0x98
7	0x9c

11.2.2 Clock Divider

Access : Read / Write

Offset: 0x00

31	17 16 15	0
Reserved		T Divider Ratio

16	Through Mode (T)	Default: 0x0
このビットに 1 がセットされた場合, 分周せずに PLL のクロックをスルーする (1/1 分周) .		
15:0	Divider Ratio	Default: 0x0
n の値を書き込むと, PLL に対して $2 \times (n + 1)$ 分周される .		

11.2.3 Clock Gating

Access : Read / Write

Offset: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G31	G30	G29	G28	G27	G26	G25	G24	G23	G22	G21	G20	G19	G18	G17	G16	G15	G14	G13	G12	G11	G10	G9	G8	G7	G6	G5	G4	G3	G2	G1	G0

n	Gating # n (G_n)	Default: 0x0
n ビットに 1 がセットされた場合, # n のモジュールへのクロックの供給が停止される. n は表 11.1 の # の列の数字である .		

11.2.4 Reset

Access : Read / Write										Offset: 0x80-0x9c									
31										0									
Reserved										Auto Reset Timer									
17										Default: 0x1									
Reset (R_)																			
このビットに 0 がセットされた場合、モジュールにリセットをかける。1 がセットされた場合、リセットを解除する。																			
16										Default: 0x1									
Auto Reset (A_)																			
このビットに 0 がセットされた場合、オートリセットを有効にする。1 がセットされた場合、オートリセットを解除する。																			
15:0										Default: 0x0									
Auto Reset Timer																			
オートリセットが有効である場合、この値の間だけリセットがかかる。																			

12

Universal Asynchronous Receiver/Transmitter

初期アドレス: Channel0:0x30000000 Channel1:0x30000020

12.1 アドレスマップ

offset	31	24 23	16 15	8 7	0
0x0000	RB				
0x0000	THR				
0x0000	DL1				
0x0004	IER				
0x0004	DL2				
0x0008	IIR				
0x0008	FCR				
0x000c	LCR				
0x0010	MCR				
0x0014	LSR				
0x0018	MSR				

12.1.1 Receiver Buffer (RB) / Transmitter Holding Register (THR)

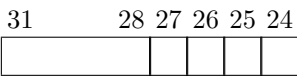
オフセット: 0x0000



bit 名	機能
31-24	送信 FIFO の入力および受信 FIFO の出力 .

12.1.2 Interrupt Enable Register (IER)

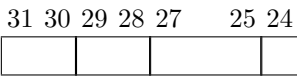
オフセット: 0x0004



bit 名	機能
24	Received Data availble interrupt. ‘ 0 ’ - Disabled. ‘ 1 ’ - Enabled.
25	Transmitter Holding Register empty interrupt. ‘ 0 ’ - Disabled. ‘ 1 ’ - Enabled.
26	Receiver Line Status Interrupt. ‘ 0 ’ - Disabled. ‘ 1 ’ - Enabled.
27	Modem Status Interrupt. ‘ 0 ’ - Disabled. ‘ 1 ’ - Enabled.
31-28	Reserved. Should be logic ‘ 0 ’.

12.1.3 Interrupt Identification Register (IIR)

オフセット: 0x0008



bit 名	機能																														
24	When this is ‘ 0 ’, an interrupt is pending. When this is ‘ 1 ’, no interrupt is pending.																														
27-25	The following table displays the list of possible interrupts along with the bits they enable, priority, and their source and reset control. <table><tr><td></td><td>Prio- rity</td><td>Interrupt Type</td><td>Interrupt Source</td><td>Interrupt Reset Control</td></tr><tr><td>011</td><td>1th</td><td>Receiver Line Status</td><td>Parity, Overrun or Framing errors or Break Interrupt</td><td>Reading the Line Status Register</td></tr><tr><td>010</td><td>2nd</td><td>Receiver Data available</td><td>FIFO trigger level reached</td><td>FIFO drops below trigger level</td></tr><tr><td>110</td><td>2nd</td><td>Timeout Indication</td><td>There’s at least 1 character in the FIFO but no character has been input to the FIFO or read from it for the last 4 char times.</td><td>Reading from the FIFO (Receiver Buffer Register)</td></tr><tr><td>001</td><td>3rd</td><td>Transmitter Holding Register empty</td><td>Transmitter Holding Register Empty</td><td>Writing to the Transmitter Holding Register or reading the IIR</td></tr><tr><td>000</td><td>4th</td><td>Modem Status</td><td>CTS, DSR, RI or DCD</td><td>Reading the Modem Status Register</td></tr></table>		Prio- rity	Interrupt Type	Interrupt Source	Interrupt Reset Control	011	1th	Receiver Line Status	Parity, Overrun or Framing errors or Break Interrupt	Reading the Line Status Register	010	2nd	Receiver Data available	FIFO trigger level reached	FIFO drops below trigger level	110	2nd	Timeout Indication	There’s at least 1 character in the FIFO but no character has been input to the FIFO or read from it for the last 4 char times.	Reading from the FIFO (Receiver Buffer Register)	001	3rd	Transmitter Holding Register empty	Transmitter Holding Register Empty	Writing to the Transmitter Holding Register or reading the IIR	000	4th	Modem Status	CTS, DSR, RI or DCD	Reading the Modem Status Register
	Prio- rity	Interrupt Type	Interrupt Source	Interrupt Reset Control																											
011	1th	Receiver Line Status	Parity, Overrun or Framing errors or Break Interrupt	Reading the Line Status Register																											
010	2nd	Receiver Data available	FIFO trigger level reached	FIFO drops below trigger level																											
110	2nd	Timeout Indication	There’s at least 1 character in the FIFO but no character has been input to the FIFO or read from it for the last 4 char times.	Reading from the FIFO (Receiver Buffer Register)																											
001	3rd	Transmitter Holding Register empty	Transmitter Holding Register Empty	Writing to the Transmitter Holding Register or reading the IIR																											
000	4th	Modem Status	CTS, DSR, RI or DCD	Reading the Modem Status Register																											
29-28	Reserved. Should be logic ‘ 0 ’.																														
31-30	Reserved. Should be logic ‘ 1 ’ for compatibility reason.																														

12.1.4 FIFO Control Register (FCR)

オフセット: 0x0008

31	30	29	27	26	25	24

bit 名	機能
24	Ignored(Used to enable FIFOs in NS16550D). Since this UART only supports FIFO mode, this bit is ignored.
25	Writing a ' 1 ' to bit 1 clears the Receiver FIFO and resets its logic. But it doesn 't clear the shift register, i.e. receiving of the current character continues.
26	Writing a ' 1 ' to bit 2 clears the Transmitter FIFO and resets its logic. The shift register is not cleared, i.e. transmitting of the current character continues.
29-27	Ignored.
31-30	7-6 Define the Receiver FIFO Interrupt trigger level. ' 00 ' - 1 bytes ' 01 ' - 4 bytes ' 10 ' - 8 bytes ' 11 ' - 16 bytes

12.1.5 Line Control Register (LCR)

オフセット: 0x000c

31 30 29 28 27 26 25 24

bit 名	機能
25-24	Select number of bits in each character. ' 00 ' - 5 bits ' 01 ' - 6 bits ' 10 ' - 7 bits ' 11 ' - 8 bits
26	Specify the number of generated stop bits. ' 0 ' - 1 stop bit. ' 1 ' - 1.5 stop bits when 5-bit character length selected and 2 bits otherwise. Note that the receiver always checks the first stop bit only.
27	Parity Enable. ' 0 ' - No parity ' 1 ' - Parity bit is generated on each outgoing character and is checked on each incoming one.
28	Even Parity select. ' 0 ' - Odd number of ' 1 ' is transmitted and checked in each word (data and parity combined). In other words, if the data has an even number of ' 1 ' in it, then the parity bit is ' 1 '. ' 1 ' - Even number of ' 1 ' is transmitted in each word.
29	Stick Parity bit. ' 0 ' - Stick Parity disabled. ' 1 ' - If bits 3 and 4 are logic ' 1 ', the parity bit is transmitted and checked as logic ' 0 '. If bit 3 is ' 1 ' and bit 4 is ' 0 ' then the parity bit is transmitted and checked as ' 1 '.
30	Break Control bit. ' 1 ' - The serial out is forced into logic ' 0 ' (break state). ' 0 ' - Break is disabled.
31	Divisor Latch Access bit. ' 1 ' - The divisor latches can be accessed. ' 0 ' - The normal registers are accessed.

12.1.6 Modem Control Register (MCR)

オフセット: 0x0010

31 29 28 27 26 25 24

--	--	--	--	--	--	--

bit 名	機能
24	Data Terminal Ready (DTR) signal control. ‘ 0 ’ - DTR is ‘ 1 ’ ‘ 1 ’ - DTR is ‘ 0 ’
25	Request To Send (RTS) signal control ‘ 0 ’ - RTS is ‘ 1 ’ ‘ 1 ’ - RTS is ‘ 0 ’
26	Out1. In loopback mode, connected Ring Indicator (RI) signal input.
27	Out2. In loopback mode, connected to Data Carrier Detect (DCD) input.
28	Loopback mode. ‘ 0 ’ - normal operation. ‘ 1 ’ - loopback mode. When in loopback mode, the Serial Output Signal (STX_PAD_O) is set to logic ‘ 1 ’ . The signal of the transmitter shift register is internally connected to the input of the receiver shift register. The following connections are made: DTR → DSR RTS → CTS Out1 → RI Out2 → DCD
31-29	Ignored.

12.1.7 Line Status Register (LSR)

オフセット: 0x0014

31 30 29 28 27 26 25 24

bit 名	機能
24	Data Ready (DR) indicator. ' 0 ' - No characters in the FIFO. ' 1 ' - At least one character has been received and is in the FIFO.
25	Overrun Error (OE) INDICATOR. ' 1 ' - If the FIFO is full and another character has been received in the receiver shift register. If another character is starting to arrive, it will overwrite the data in the shift register but the FIFO will remain intact. The bit is cleared upon reading from the register. Generates Receiver Line Status interrupt. ' 0 ' - No overrun state.
26	Parity Error (PE) indicator. ' 1 ' - The character that is currently at the top of the FIFO has been received with parity error. The bit is cleared upon reading from the register. Generate Receiver Line Status interrupt. ' 0 ' - No parity error in the current character.
27	Framing Error (FE) indicator. ' 1 ' - The received character at the top of the FIFO did not have a valid stop bit. The UART core tries re-synchronizing by assuming that the bit received was a start bit. Of course, generally, it might be that all the following data is corrupt. The bit is cleared upon reading from the register. Generates Receiver Line Status interrupt. ' 0 ' - No framing error in the current character.
28	Break Interrupt (BI) indicator. ' 1 ' - A break condition has been reached in the current character. The break occurs when the line is held in logic 0 for a time of one character (start bit + data + parity + stop bit). In that case, one zero character enters the FIFO and the UART waits for a valid start bit to receive next character. The bit is cleared upon reading from the register. Generates Receiver Line Status interrupt. ' 0 ' - No break condition in the current character.
29	Transmit FIFO is empty. ' 1 ' - The transmitter FIFO is empty. Generates Transmitter Holding Register Empty interrupt. The bit is cleared in the following cases: The LSR has been read, the IIR has been read or data has been written to the transmitter FIFO. ' 0 ' - Otherwise.
30	Transmitter Empty indicator. ' 1 ' - Both the transmitter FIFO and transmitter shift register are empty. The bit is cleared upon reading from the register or upon writing data to the transmit FIFO. ' 0 ' - Otherwise.
31	' 1 ' - At least one parity error, framing error or break indications have been received and are inside the FIFO. The bit is cleared upon reading from the register. ' 0 ' - Otherwise.

12.1.8 Modem Status Register (MSR)

オフセット: 0x0018

31 30 29 28 27 26 25 24

--	--	--	--	--	--	--	--

bit 名	機能
24	Delta Clear To Send (DCTS) indicator. ‘ 1 ’ - The CTS line has changed its state.
25	Delta Data Set Ready (DDSR) indicator. ‘ 1 ’ - The DSR line has changed its state.
26	Trailing Edge of Ring Indicator (TERI) detector. The RI line has changed its state from low to high state.
27	Delta Data Carrier Detect (DDCD) indicator. ‘ 1 ’ - The DCD line has changed its state.
28	Complement of the CTS input or equals to RTS in loopback mode.
29	Complement of the DSR input or equals to DTR in loopback mode.
30	Complement of the RI input or equals to Out1 in loopback mode.
31	Complement of the DCD input or equals to Out2 in loopback mode.

12.1.9 Divisor Latches (DL)

オフセット: 0x0000(DL1), 0x0004(DL2)

The divisor latches can be accessed by setting the 7th bit of LCR to ‘ 1 ’. You should restore this bit to ‘ 0 ’ after setting the divisor latches in order to restore access to the other registers that occupy the same addresses.

31 24

DL1

31 24

DL2

bit 名	機能
DL1, DL2	The 2 bytes form one 16-bit register, which is internally accessed as a single number. You should therefore set all 2 bytes of the register to ensure normal operation. The register is set to the default value of 0 on reset, which disables all serial I/O operations in order to ensure explicit setup of the register in the software. The value set should be equal to (system clock speed) / (16 times desired baud rate). The internal counter starts to work when the LSB of DL is written, so when setting the divisor, write the MSB first and the LSB last.

12.2 動作/使用方法

This UART core is very similar in operation to the standard 16550 UART chip with the main exception being that only the FIFO mode is supported. The scratch register is removed, as it serves no purpose.

12.2.1 Initialization

Upon reset the core performs the following tasks:

- The receiver and transmitter FIFOs are cleared.
- The receiver and transmitter shift registers are cleared.
- The Divisor Latch register is set to 0.
- The Line Control Register is set to communication of 8 bits of data, no parity, 1 stop bit.
- All interrupts are disabled in the Interrupt Enable Register.

For proper operation, perform the following:

- Set the Line Control Register to the desired line control parameters. Set bit 7 to ' 1 ' to allow access to the Divisor Latches.
- Set the Divisor Latches, MSB first, LSB next.
- Set bit 7 of LCR to 0 to disable access to Divisor Latches. At this time the transmission engine starts working and data can be sent and received.
- Set the FIFO trigger level. Generally, higher trigger level values produce less interrupt to the system, so setting it to 14 bytes is recommended if the system responds fast enough.
- Enable desired interrupts by setting appropriate bits in the Interrupt Enable register.

Remember that $(\text{Input Clock Speed})/(\text{Divisor Latch value}) = 16 \times \text{the communication baud rate}$. Since the protocol is asynchronous and the sampling of the bits is performed in the perceived middle of the bit time, it is highly immune to small differences in the clocks of the sending and receiving sides, yet no such assumption should be made calculating the Divisor Latch values.

13

General Purpose I/O Unit

13.1 Outline

General Purpose I/O Unit は 8 bit の入出力を提供する。

13.2 Interface

13.2.1 Address Format

General Purpose I/O Unit の初期ベースアドレスは 0x31000000 である。General Purpose I/O Unit の制御レジスタのアドレスは次のようになる。



Field Name	Range	Description
Offset	4:0	設定する項目を指定する。

13.2.2 Control Register

General Purpose I/O Unit の制御を行う場合、以下に示す Offset をアドレスの Offset に指定することにより、該当設定レジスタにアクセスする。

Data

offset: 0x00

31	8	7	0
Reserved			Data

Field Name	Range	Description
Data	7:0	bit が入力の場合は，読むことにより外部からの入力を得る．bit が出力の場合は，書き込むことにより外部に出力を与える．

Direction

offset: 0x04

31	8	7	0
Reserved			Direction

Field Name	Range	Description
Direction	7:0	0 の場合は入力．1 の場合は出力となる．

Interrupt Enable

offset: 0x08

31	8	7	0
Reserved			Interrupt Enable

Field Name	Range	Description
Interrupt Enable	7:0	1 の場合，Data レジスタの対応する値が変化した時に割り込みを発生させる．割り込み発生条件は Interrupt Upedge レジスタ，Interrupt Downedge レジスタで設定する．

Interrupt Sense

offset: 0x0c

31	8	7	0
Reserved			Interrupt Sense

Field Name	Range	Description
Interrupt Sense	7:0	割り込みの発生要因になった bit に 1 がセットされる．このレジスタに 0 を書き込むと，対応する bit の割り込みがクリアされる．

Interrupt Upedge

offset: 0x10

31	8	7	0
Reserved			Interrupt Upedge

Field Name	Range	Description
Interrupt Upedge	7:0	1 の場合、データが 0 から 1 に変化した時に割り込みを発生させる。

Interrupt Downedge

offset: 0x14

31	8	7	0
Reserved			Interrupt Downedge

Field Name	Range	Description
Interrupt Downedge	7:0	1 の場合、データが 1 から 0 に変化した時に割り込みを発生させる。

Configuration

offset: 0x18 (Read Only)

31	2	1	0
Reserved			BW

Field Name	Range	Description
Bit Width (BW)	1:0	General Purpose I/O の Bit Width を示す。 0x1 : 8 Bit 0x2 : 16 Bit 0x3 : 32 Bit

13.3 Operation

General Purpose I/O は 8 bit の幅を持ち、bit ごとに入出力の方向を設定することができる。設定は Direction レジスタで行う。

入力の場合、General Purpose I/O に入力されるクロックにより、I/O ピンのデータが Data レジスタにラッチされる。出力の場合、Data レジスタの値が I/O ピンに出力される。

各 bit は指定した条件により割り込みを発生させることができる。割り込みを発生させるためには、Interrupt Enable レジスタの対応する bit を 1 にセットする。また、割り込み発生条件により、Interrupt Upedge レジスタ、Interrupt Downedge レジスタの対応する bit を 1 にセットする。両方 1 にセットした場合、値が変化するたびに割り込みが発生する。

14

Serial Peripheral Interface Unit

14.1 Outline

Serial Peripheral Interface Unit はクロック同期式のシリアルインターフェースであり，SPI の規格に準拠した周辺機器を制御する．

14.2 Interface

14.2.1 Address Format

Serial Peripheral Interface Unit の初期ベースアドレスは 0x32000000 である．SPI Unit の制御レジスタのアドレスは次のようになる．



Field Name	Range	Description
Offset	5:0	設定する項目を指定する．

14.2.2 Control Register

Serial Peripheral Interface Unit の制御を行う場合，以下に示す Offset をアドレスの Offset に指定することにより，該当設定レジスタにアクセスする．

Slave Control

offset: 0x00

Slave の設定を行う .

31	5	4	1	0
Reserved				SS_
				A

Field Name	Range	Description
Auto (A)	0	0 に設定すると自動的にデータ転送を繰り返す . 1 を設定すると , マニュアルで SPI の転送を行う .
Slave Select_ (SS_)	4:1	MSB が slave0 に , LSB が slave3 に対応する . Auto bit が 1 の場合は , アクセスする Slave を指定する . 複数の bit が 0 の場合 , 下位の bit が優先される . Auto bit が 0 の場合は , 自動でアクセス Slave を指定する . 複数の bit が 0 の場合 , 下位の bit から順番にアクセスを行う .

FIFO Control

offset: 0x04

FIFO の設定を行う .

31	10	9	8	7	4	3	0
Reserved						CLR	DREQ
						INTR	

Field Name	Range	Description
Interrupt (INTR)	3:0	1 を設定した場合 , 要因により割り込みを発生させる . 0: 受信 FIFO にデータが半分たまと割り込みを発生させる . 1: 受信 FIFO がいっぱいになると割り込みを発生させる . 2: 送信 FIFO のデータが半分未満になると割り込みを発生させる . 3: 送信 FIFO が空になると割り込みを発生させる .
DMA Request (DREQ)	7:4	1 を設定した場合 , 要因により DMA Request を発生させる . 4: 受信 FIFO にデータが半分たまと DMA Request を発生させる . 5: 受信 FIFO がいっぱいになると DMA Request を発生させる . 6: 送信 FIFO のデータが半分より少くなると DMA Request を発生させる . 7: 送信 FIFO がからになると DMA Request を発生させる .
Clear (CLR)	9:8	1 にすると FIFO をクリアする . この bit は自動的に 0 になる . 8: 受信 FIFO をクリアする . 9: 送信 FIFO をクリアする .

31							6	5	4	3	2	1	0
Reserved								TF	TH	TE	RF	RH	RE

Field Name	Range	Description
Rx Empty (RE)	0	受信 FIFO が空の場合 1 になる。
Rx Half (RH)	1	受信 FIFO に半分以上データがたまっている場合 1 になる。
Rx Full (RF)	2	受信 FIFO がいっぱいの場合 1 になる。
Tx Empty (RE)	3	送信 FIFO が空の場合 1 になる。
Tx Half (RH)	4	送信 FIFO に半分以上データがたまっている場合 1 になる。
Tx Full (RF)	5	送信 FIFO がいっぱいの場合 1 になる。

31	0
FIFO	

Field Name	Range	Description
FIFO	31:0	書き込みの場合は送信 FIFO に値が書き込まれる．Slave Control レジスタの Auto bit が 0 の場合，値を書き込むことにより，データの転送を開始する．読み込みの場合は受信 FIFO から値が読み出される．

割り込みの要因を示す，1を書き込むことにより，その bit をクリアする．

31		4	3	2	1	0
	Reserved		TE	TH	RF	RH

Field Name	Range	Description
Rx Half (RH)	0	受信 FIFO に半分以上データがたまっている．
Rx Full (RF)	1	受信 FIFO がいっぱいである．
Tx Half (RH)	2	送信 FIFO のデータが半分未満になった．
Tx Empty (RE)	3	送信 FIFO が空である．

Interval		offset: 0x14
31		0
Interval		

Field Name	Range	Description
Interval	31:0	Slave Control レジスタの Auto bit が 0 の場合，Slave に対する一連のアクセスが終わった後の待ち時間を指定する．

Mode3										offset: 0x20																
Slave Select3 用の設定を行なう。																										
31			30			29			28			24			23			22			21			0		
W_		R_		L		Size						OL		HA		Clock Ratio										

Field Name	Range	Description
Clock Ratio	21:0	同期クロックで出力するクロックの分周率を指定する．実際には指定した数値 × 2 で SPI の内部クロックを分周し，出力する．0 を指定した場合は $2^{22} \times 2$ 分周される．デフォルトは 0 ．
HA, OL	23:22	SPI の動作モードを指定する． 0x0 同期クロックは正極性．立ち上りでデータを受け取る． 0x1 同期クロックは正極性．立ち下りでデータを受け取る． 0x2 同期クロックは負極性．立ち下りでデータを受け取る． 0x3 同期クロックは負極性．立ち上りでデータを受け取る．
Size	28:24	データの転送サイズ．指定した値 + 1 bit を転送する．
LSB (L)	29	1 を指定すると LSB から転送を開始する．0 の場合は MSB から転送する．
Read Enable (R ₋)	30	0 を指定すると外部からの入力を読み込み，受信 FIFO に値を格納する．1 の場合は外部からのデータを読み込まない．
Write Enable (W ₋)	31	0 を指定すると送信 FIFO のデータを外部に転送する．1 の場合はデータを送信しない．

Mode2

offset: 0x24

Slave Select2 用の設定を行なう．

31 30 29 28 24 23 22 21

0

W_	R_	L	Size	OL	HA	Clock Ratio
----	----	---	------	----	----	-------------

Field Name	Range	Description
Clock Ratio	21:0	同期クロックで出力するクロックの分周率を指定する．実際には指定した数値 $\times 2$ で SPI の内部クロックを分周し，出力する．0 を指定した場合は $2^{22} \times 2$ 分周される．デフォルトは 0．
HA, OL	23:22	SPI の動作モードを指定する． 0x0 同期クロックは正極性．立ち上りでデータを受け取る． 0x1 同期クロックは正極性．立ち下りでデータを受け取る． 0x2 同期クロックは負極性．立ち下りでデータを受け取る． 0x3 同期クロックは負極性．立ち上りでデータを受け取る．
Size	28:24	データの転送サイズ．指定した値 + 1 bit を転送する．
LSB (L)	29	1 を指定すると LSB から転送を開始する．0 の場合は MSB から転送する．
Read Enable (R_)	30	0 を指定すると外部からの入力を読み込み，受信 FIFO に値を格納する．1 の場合は外部からのデータを読み込まない．
Write Enable (W_)	31	0 を指定すると送信 FIFO のデータを外部に転送する．1 の場合はデータを送信しない．

Mode1

offset: 0x28

Slave Select1 用の設定を行なう .

31 30 29 28

24 23 22 21

0

W_	R_	L	Size	OL	HA	Clock Ratio
----	----	---	------	----	----	-------------

Field Name	Range	Description
Clock Ratio	21:0	同期クロックで出力するクロックの分周率を指定する．実際には指定した数値 × 2 で SPI の内部クロックを分周し，出力する．0 を指定した場合は $2^{22} \times 2$ 分周される．デフォルトは 0 ．
HA, OL	23:22	SPI の動作モードを指定する． 0x0 同期クロックは正極性．立ち上りでデータを受け取る． 0x1 同期クロックは正極性．立ち下りでデータを受け取る． 0x2 同期クロックは負極性．立ち下りでデータを受け取る． 0x3 同期クロックは負極性．立ち上りでデータを受け取る．
Size	28:24	データの転送サイズ．指定した値 + 1 bit を転送する．
LSB (L)	29	1 を指定すると LSB から転送を開始する．0 の場合は MSB から転送する．
Read Enable (R_)	30	0 を指定すると外部からの入力を読み込み，受信 FIFO に値を格納する．1 の場合は外部からのデータを読み込まない．
Write Enable (W_)	31	0 を指定すると送信 FIFO のデータを外部に転送する．1 の場合はデータを送信しない．

Mode0

offset: 0x2c

Slave Select0 用の設定を行なう。

31 30 29 28

24 23 22 21

0

W ₋	R ₋	L	Size	OL	HA	Clock Ratio
----------------	----------------	---	------	----	----	-------------

Field Name	Range	Description
Clock Ratio	21:0	同期クロックで出力するクロックの分周率を指定する．実際には指定した数値 × 2 で SPI の内部クロックを分周し，出力する．0 を指定した場合は $2^{22} \times 2$ 分周される．デフォルトは 0．
HA, OL	23:22	SPI の動作モードを指定する． 0x0 同期クロックは正極性．立ち上りでデータを受け取る． 0x1 同期クロックは正極性．立ち下りでデータを受け取る． 0x2 同期クロックは負極性．立ち下りでデータを受け取る． 0x3 同期クロックは負極性．立ち上りでデータを受け取る．
Size	28:24	データの転送サイズ．指定した値 + 1 bit を転送する．
LSB (L)	29	1 を指定すると LSB から転送を開始する．0 の場合は MSB から転送する．
Read Enable (R ₋)	30	0 を指定すると外部からの入力を読み込み，受信 FIFO に値を格納する．1 の場合は外部からのデータを読み込まない．
Write Enable (W ₋)	31	0 を指定すると送信 FIFO のデータを外部に転送する．1 の場合はデータを送信しない．

Configuration

offset: 0x30 (Read Only)

31	2	1	0
Reserved			FS

Field Name	Range	Description
FIFO Size (FS)	1:0	FIFO のサイズを示す． 0x1 : 8 Entry 0x2 : 16 Entry 0x3 : 32 Entry

14.3 Operation

本 SPI Unit は 4 本の Slave Select を持ち，各 Slave に対して個別の設定を行うことができる．設定は各 Slave の Mode レジスタで行う．

本 SPI Unit は Slave へのアクセスの方法として，自動で継続的に Slave から値を読み込むモードと，1 つ 1 つ送受信を行うモードがある．

14.3.1 Manual Mode

1 つ 1 つ送受信を行う場合、Slave Control レジスタの Auto bit を 1 にし、アクセスする Slave を Slave Control レジスタの Slave Select_ bit で指定する。

値を送信したい場合は、各 Slave の Mode レジスタの W_ bit を 0 にする。送信したい値を FIFO に書き込むことにより、インターフェースから値が送信される。

値を受信したい場合は、各 Slave の Mode レジスタの R_ bit を 0 にする。FIFO に値を書き込むことによりインターフェースが動作し、Slave から値を受信する。受信した値は受信 FIFO に書き込まれる。

SPI は送受信を同時に行うことができる。各 Slave の Mode レジスタの W_ bit と R_ bit を両方 0 にすることにより、送受信を同時に行う。

14.3.2 Auto Mode

Auto Mode は Slave から自動で継続的に値を読み出す場合に使用する。Slave Control レジスタの Auto bit を 0 にすることにより Auto Mode として動作する。値を読み出す Slave は Slave Control レジスタの Slave Select_ bit で指定する。

各 Slave に対するアクセスは各 Slave の Mode レジスタの設定による。そのため、Mode レジスタの R_ bit と W_ bit を 0 にする必要がある。

Auto Mode では、SPI Unit は Slave Control レジスタの Slave Select_ bit が 0 になっている Slave に対して下位側 (3 番) から順番にアクセスを行う。指定された全ての Slave に対してアクセスを行った後、Interval レジスタで指定されているサイクル数だけ待った後、指定された Slave に対してアクセスを開始する。

15

I2C Master Controller

15.1 Outline

I2C is a two-wire, bi-directional serial bus that provides a simple and efficient method of data exchange between devices. It is most suitable for applications requiring occasional communication over a short distance between many devices. The I2C standard is a true multi-master bus including collision detection and arbitration that prevents data corruption if two or more masters attempt to control the bus simultaneously.

15.2 Interface

15.2.1 Address Format

I2C Master Controller の初期ベースアドレスは 0x33000000 である . I2C Master Controller の制御レジスタのアドレスは次のようになる .



Field Name	Range	Description
Offset	4:0	設定する項目を指定する .

15.2.2 Control Register

I2C Master Controller の制御を行う場合 , 以下に示す Offset をアドレスの Offset に指定することにより , 該当設定レジスタにアクセスする .

Clock Prescale (lo-byte) offset: 0x00									
31							8	7	0
Reserved								scale	

Clock Prescale (hi-byte)										offset: 0x04					
31										8		7		0	
Reserved										scale					

Field Name	Range	Description
scale	7:0	This register is used to prescale the SCL clock line. Due to the structure of the I2C interface, the core uses a 5*SCL clock internally. The prescale register must be programmed to this 5*SCL frequency (minus 1). Change the value of the prescale register only when the 'EN' bit is cleared.

Example: `wb.clk_i = 32MHz`, desired SCL = 100KHz

$$prescale = \frac{32MHz}{5 * 100KHz} - 1 = 63(dec) = 3F(hex)$$

Control										offset: 0x08					
31								8	7	6	5				0
Reserved									EN	EN	Reserved				

Field Name	Range	Description
I2C core enable bit (EN)	7	When set to '1', the core is enabled. When set to '0', the core is disable.
I2C core interrput enable bit (IEN)	6	When set to '1', interrupt is enabled. When set to '0', interrupt is disable.

Transmit

offset: 0x0c (Write Only)

31	8	7	1	0
<i>Reserved</i>			NB	X

Field Name	Range	Description
Next byte (NB)	7:0	Next byte to transmit via I2C.
X	0	In case of a data tranfer this bit represents the data's LSB. In case of a slave address transfer this bit represents the RW bit. '1' = reading from slave. '0' = writing to slave.

Receive

offset: 0x0c (Read Only)

31	8	7	0
<i>Reserved</i>			LB

Field Name	Range	Description
Last byte (LB)	7:0	Last byte received via I2C.

Command

offset: 0x10 (Write Only)

31								8	7	6	5	4	3	2	1	0		
Reserved												STA	STO	R	W	A	Reserved	I

Field Name	Range	Description
Start (STA)	7	Generate (repeated) start condition.
Stop (STO)	6	Generate stop condition.
Read (R)	5	Read from slave.
Write (W)	4	Write to slave.
ACK (A)	3	When a receiver, sent ACK (ACK = '0') or NACK (ACK = '1').
Interrupt ACK (I)	1	Interrupt acknowledge. When set, clears a pending interrupt.

The STA, STO, R, W, and I bits are cleared automatically. These bits are always read as zeros.

Status										offset: 0x10 (Read Only)									
31										8 7 6 5 4 2 1 0									
Reserved										R	B	A	Reserved	T	I				

Field Name	Range	Description
Received acknowledge from slave (R)	7	This flag represents acknowledge from the addressed slave. '1' = No acknowledge received. '0' = Acknowledge received.
I2C bus busy (B)	6	'1' after START signal detect. '0' after STOP signal detect.
Arbitration lost (A)	5	This bit is set when the core lost arbitration. Arbitration is lost when: - a STOP signal is detected, but non requested. - The master drives SDA high, but SDA is low. See bus-arbitration section for more information.
Transfer in progress (T)	1	'1' when transferring data. '0' when transfer complete.
Interrupt Flag (I)	0	This bit is set when an interrupt is pending, which will cause a processor interrupt request if the IEN bit is set. The Interrupt Flag is set when: - one byte transfer has been completed. - arbitration is lost.

15.3 Operation

15.3.1 System Configuration

I2C system uses a serial data line (SDA) and a serial clock line (SCL) for data transfers. All devices connected to these two signals must have open drain or open collector outputs. The logic AND function is exercised on both lines with external pull-up resistors.

Data is transferred between a Master and a Slave synchronously to SCL on the SDA line on a byte-by-byte basis. Each data byte is 8 bits long. There is one SCL clock pulse for each data bit with the MSB being transmitted first. An acknowledge bit follows each transferred byte. Each bit is sampled during the high period of SCL; therefore, the SDA line may be changed only during the low period of SCL and must be held stable during the high period of SCL. A transition on the SDA line while SCL is high is interpreted as a command (see START and STOP signals).

15.3.2 I2C Protocol

Normally, a standard communication consists of four parts:

1. START signal generation
2. Slave address transfer
3. Data transfer
4. STOP signal generation

START signal

When the bus is free/idle, meaning no master device is engaging the bus (both SCL and SDA lines are high), a master can initiate a transfer by sending a START signal. A START signal, usually referred to as the S-bit, is defined as a high-to-low transition of SDA while SCL is high. The START signal denotes the beginning of a new data transfer. A Repeated START is a START signal without first generating a STOP signal. The master uses this method to communicate with another slave or the same slave in a different transfer direction (e.g. from writing to a device to reading from a device) without releasing the bus.

The core generates a START signal when the STA-bit in the Command Register is set and the RD or WR bits are set. Depending on the current status of the SCL line, a START or Repeated START is generated.

Slave Address Transfer

The first byte of data transferred by the master immediately after the START signal is the slave address. This is a seven-bits calling address followed by a RW bit. The RW bit signals the slave the data transfer direction. No two slaves in the system can have the same address. Only the slave with an address that matches the one transmitted by the master will respond by returning an acknowledge bit by pulling the SDA low at the 9th SCL clock cycle.

Note: The core supports 10bit slave addresses by generating two address transfers. See the Philips I2C specifications for more details.

The core treats a Slave Address Transfer as any other write action. Store the slave device's address in the Transmit Register and set the WR bit. The core will then transfer the slave address on the bus.

Data Transfer

Once successful slave addressing has been achieved, the data transfer can proceed on a byte-by-byte basis in the direction specified by the RW bit sent by the master. Each transferred byte is followed by an acknowledge bit on the 9th SCL clock cycle. If the slave signals a No Acknowledge, the master can generate a STOP signal to abort the data transfer or generate a Repeated START signal and start a new transfer cycle.

If the master, as the receiving device, does not acknowledge the slave, the slave releases the SDA line for the master to generate a STOP or Repeated START signal.

To write data to a slave, store the data to be transmitted in the Transmit Register and set the WR bit. To read data from a slave, set the RD bit. During a transfer the core set the TIP flag, indicating that a Transfer is In Progress. When the transfer is done the TIP flag is reset, the IF flag set and, when enabled, an interrupt generated. The Receive Register contains valid data after the IF flag has been set. The user may issue a new write or read command when the TIP flag is reset.

STOP signal

The master can terminate the communication by generating a STOP signal. A STOP signal, usually referred to as the P-bit, is defined as a low-to-high transition of SDA while SCL is at logical '1'.

15.3.3 Arbitration Procedure

The I2C bus is a true multimaster bus that allows more than one master to be connected on it. If two or more masters simultaneously try to control the bus, a clock synchronization procedure determines the bus clock. Because of the wired-AND connection of the I2C signals a high to low transition affects all devices connected to the bus. Therefore a high to low transition on the SCL line causes all concerned devices to count off their low period. Once a device clock has gone low it will hold the SCL line in that state until the clock high state is reached. Due to the wired-AND connection the SCL line will therefore be held low by the device with the longest low period, and held high by the device with the shortest high period.

15.3.4 Clock Stretching

Slave devices can use the clock synchronization mechanism to slow down the transfer bit rate. After the master has driven SCL low, the slave can drive SCL low for the required period and then release it. If the slave's SCL low period is greater than the master's SCL low period, the resulting SCL bus signal low period is stretched, thus inserting wait-states.

16

PWM Generator

16.1 PWM 発生器概要

- PWM 出力：内部レジスタの設定によってデューティ比の異なる矩形波を出力
- Bit 幅：32bit
- ノコギリ波を用いて PWM を発生するノコギリ波モードと三角波を用いた三角波モード
- デッドタイムの設定
- 正論理，負論理の設定
- PWM チャンネルをグルーピングしてグループ毎に同期可能
- PWM の周期毎の割り込み可能
- 汎用出力としても利用可能
- チャンネル数：6

PWM 発生器の初期ベースアドレスは 0x38000000 である。

図 16.1 にノコギリ波モード，図 16.2 に三角波モードの PWM 波形を示す。

デッドタイム付反転出力は，隣の PWM 発生器から出力できるようにサイクリックにカスケード接続されている。具体的には，PWM 発生器 N のデッドタイム付反転出力は，REV bit を立てることにより，PWM 発生器 N+1 で利用することができる。

また，カウンタのスタートが同期した PWM のグループを作ることができる。

16.2 PWM コントロールレジスタ

アドレス	CTRL レジスタ
0x00	PWMCTRL[0]
0x20	PWMCTRL[1]
0x40	PWMCTRL[2]
0x60	PWMCTRL[3]
0x80	PWMCTRL[4]
0xA0	PWMCTRL[5]

リード/ライト

31	10	9	8	7	6	5	4	3	2	1	0
0	INT	SYN	INV	M	REV	DEN	D	P	CLR	CEN	

bit 名	機能
INT	Invert: Default 0 0: 割り込みを発生させない 1: 割り込みを発生させる 割り込みは、周期の開始時に発生される。
SYN	Invert: Default 0 0: カウントの開始を同期しない（CEN のみを使用する） 1: カウントの開始を同期する（本 PWM 発生器より一つ若い番号の PWM 発生器で生成されたスタート信号を使用する） 本 PWM のカウンタのスタート信号は、CEN が 1 になった時、または本 SYN が 1 かつ本 PWM 発生器より一つ若い番号のスタート信号が入力されたときにアクティブとなる。したがって、ある PWM 発生器の CEN を、引き続く複数の PWM 発生器で使うことができ、カウンタのスタートが同期した PWM のグループを作ることができる。
INV	Invert: Default 0 0: PWM 波をそのまま出力する 1: PWM 波を出力する最終段において PWM 波を反転して出力する 最も優先度が高い
M	Mode: Default 0 0: ノコギリ波で PWM を生成するノコギリ波モード 1: 三角波で PWM を生成する三角波モード
REV	Reverse mode enable: Default 0 0: 本 PWM 発生器で生成された PWM 波を出力する通常モード 1: 本 PWM 発生器より一つ若い番号の PWM 発生器で生成された PWM 波のデッドタイム付反転出力を出力するモード（本 PWM 発生器内のカウンタは使用しない） DEN より優先度が高い
DEN	Data Enable: Default 0 0: 生成した PWM 波を出力する 1: D bit に設定された値（一定値）を出力する REV より優先度が低い
D	Data: Default 0 DEN が 1 の時、本 D bit に設定された値（一定値）を出力する
P	Positive: Default 0 PWM 波の論理を決定する（図 16.1, 16.2 参照）。 0: 負論理 1: 正論理
CLR	Counter clear: Default 0 0: 通常動作 1: カウンタをクリアする
CEN	Count Enable: Default 0 0: カウンタを停止する 1: カウンタを起動する

16.3 PWM 周期制御レジスタ

アドレス	PWM 正転制御レジスタ
0x04	FWCNT[0]
0x24	FWCNT[1]
0x44	FWCNT[2]
0x64	FWCNT[3]
0x84	FWCNT[4]
0xA4	FWCNT[5]

リード/ライト	
31	0
FWCNT<31:0>	
bit 名	機能
FWCNT	Forward Counter: Default 0 PWM の周期を決定するカウンタレジスタである . Mode が 0(ノコギリ波モード) の時には , PWM の周期を決定する . PWM 用カウンタが 0 から FWCNT までカウントアップすると , 次のクロックで 0 に戻るようなノコギリ波を生成する (図 16.1 参照) . Mode が 1(三角波モード) の時には , PWM の半周期を決定する . PWM 用カウンタが 0 から FWCNT までカウントアップすると , 次のクロックから 0 にカウントダウンするような三角波を生成する (図 16.2 参照) .

16.4 PWM 反転制御レジスタ

アドレス	PWM 反転制御レジスタ
0x08	REVCNT[0]
0x28	REVCNT[1]
0x48	REVCNT[2]
0x68	REVCNT[3]
0x88	REVCNT[4]
0xA8	REVCNT[5]

リード/ライト

31	0
REVCNT<31:0>	

bit 名	機能
REVCNT	Reverse Counter: Default 0 PWM 出力を反転する時間を決定するレジスタである。カウンタ値が本レジスタ値と同じになったら PWM 出力は反転する（図 16.1, 16.2 参照）。

16.5 デッドタイムレジスタ

アドレス	デッドタイムレジスタ
0x0C	DT[0]
0x2C	DT[1]
0x4C	DT[2]
0x6C	DT[3]
0x8C	DT[4]
0xAC	DT[5]

リード/ライト

31	16	15	0
0		DT<15:0>	

bit 名	機能
DT<15:0>	Reverse Counter :Default 0 デッドタイムを指定するレジスタである。カウンタ値が本レジスタ値と同じになったら PWM 出力は反転する（図 16.1, 16.2 参照）。

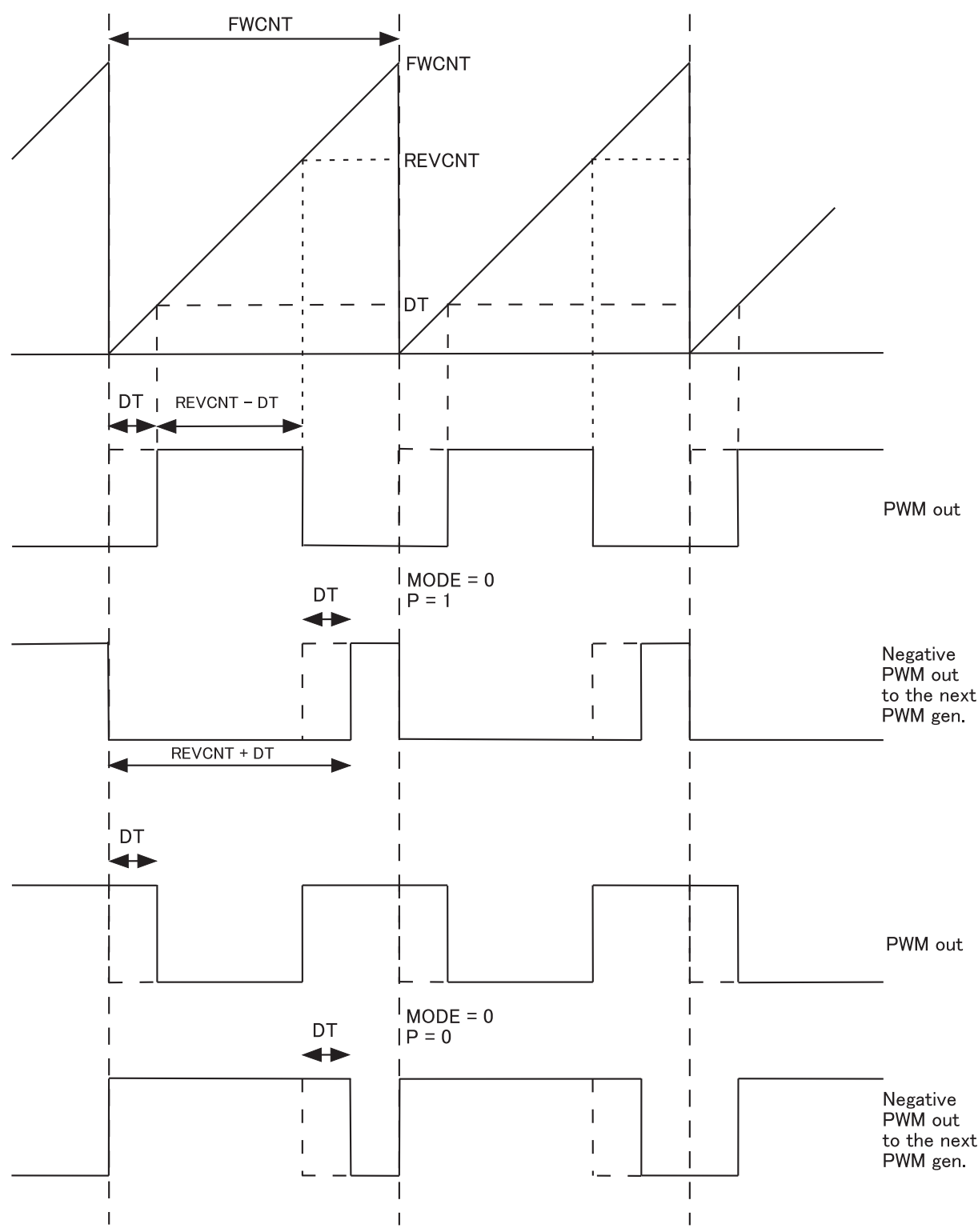


図 16.1: ノコギリ波モード

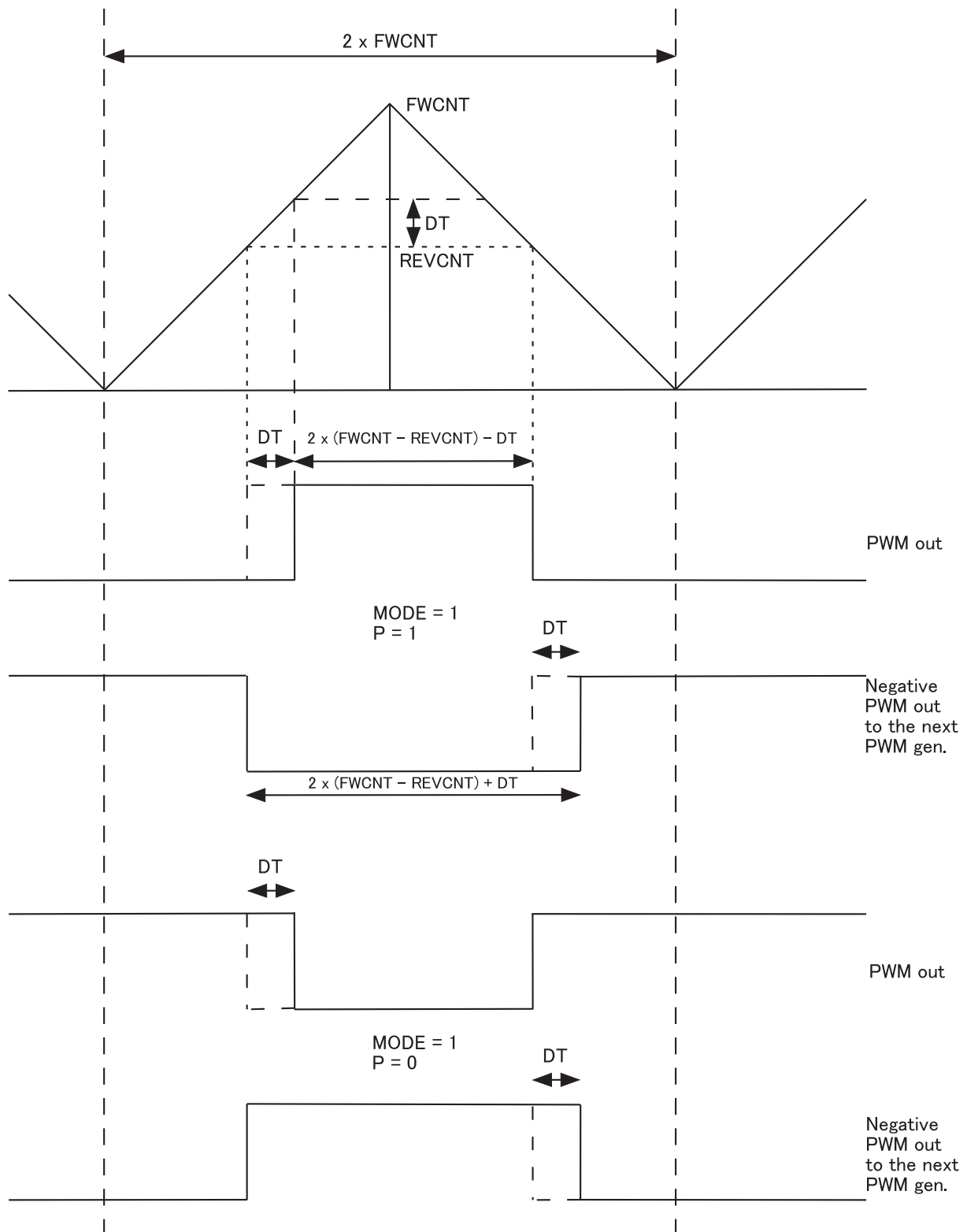


図 16.2: 三角波モード

17

PWM Input

17.1 PWM 入力器概要

- PWM 入力のデューティ比を High カウンタと Low カウンタの比に数値化
- Bit 幅：32bit
- チャンネル数：2
- クロックジェネレータ（11 章参照）で生成した基準クロックによってカウント
- 複数周期のデューティ比を求めて平均化する機構
- 割り込み発生機能

PWM 入力器の初期ベースアドレスは 0x39000000 である。

17.2 PWMIN コントロールレジスタ

アドレス	PWMIN コントロールレジスタ
0xFFFF7400	PWMINCTRL[0]
0xFFFF7420	PWMINCTRL[1]

リード/ライト

31	10	9	6	5	2	1	0
-				LP	LPO	CLR	IEN

bit 名	機能
IEN	Interrupt Enable :Default 0 r/w 0: 割り込みを発生しない . 1: 設定した周期分のデューティ比をカウント後に毎回割り込みを発生させる .
CLR	Interrupt Clear :Default 0 r/w 0:割り込みをクリアしない . 1:割り込みをクリアする . 割り込みクリア完了後に 0 にリセットされる .
LPO	Loop Original :Default 1 r/w 何周期分のデューティ比を平均化するか , その周期を設定する (1 から 15 まで , 0 は禁止) .
LP	Loop :Default 1 ro 現在実行している周期を示す .

17.3 PWMIN HIGH レジスタ

アドレス	PWMIN HIGH レジスタ
0x04	HIGH[0]
0x24	HIGH[1]

リード

31	0
HIGH<31:0>	

bit 名	機能
HIGH<31:0>	High :Default X 指定した PWM 周期分合計の High の期間 . 単位は , クロックジェネレータでプログラマブルに設定した PWMIN 用クロックのサイクル数 .

17.4 PWMIN LOW レジスタ

アドレス	PWMIN LOW レジスタ
0x08	LOW[0]
0x28	LOW[1]

リード / ライト

31	0
LOW<31:0>	

bit 名	機能
LOW<31:0>	Low :Default X 指定した PWM 周期分合計の Low の期間 . 単位は , クロックジェネレータでプログラマブルに設定した PWMIN 用クロックのサイクル数 .

bit 名	機能
INT	Interrupt :Default 0 ro 0:割り込みの発生なし．1:割り込みが発生している．割り込みはパルスカウンタ割り込み，タイマ割り込み，Z 相割り込みのいずれかの要因で発生する．本レジスタをリードすると，パルスカウンタ割り込みとタイマ割り込みがクリアされる．
IPCE	Int Pulse Counter Enable :Default 0 0：パルスカウンタによる割り込みを発生させない．1：パルスカウンタによる割り込みを発生させる．カウンタ値がコンペアデータレジスタの値と等しくなると割り込みを発生する．
IZE	Int Z Enable :Default 0 0：Z 相入力があった際に，割り込みを発生させない．1：Z 相入力があった際に，割り込みを発生させる．
ZF	Z Flag :Default 0 0：現状態は Z 相ではない．1：Z 相入力があった際に 1 に設定される．クリアする際には 0 を書く．Z 相割り込み (IZE) を有効にしている場合，0 を書くと Z 相割り込みをクリアする．
RFZ	Reset Flag by phaze Z :Default 0 0：Z 相入力によるカウンタのリセットを行わない．1：Z 相入力によるカウンタのリセットを行う．
ST	START :Default 0 0：内部タイマをリセットして，停止させる．1：内部タイマを起動させる．
TI	Timer Interrupt :Default 0 0：内部タイマによる周期割り込みを発生させない．1：内部タイマによる周期割り込みを発生させる．
SEL	Select :Default 0 カウンタのラッチの動作モードの選択を行う．0：カウンタ値のラッチを行わない．1：内部タイマにより設定された値によって，周期的にカウンタ値をラッチする．
MD<4:3>	Mode :Default 0 00：1 通倍でカウントアップする．01：2 通倍でカウントアップする．10,11：4 通倍でカウントアップする．
IE	Interrupt Enable :Default 0 0:割り込み禁止 1:割り込み許可
CLR	counter CLear :Default 1 0:カウンタをクリアする 1:don 't care
CE	Count Enable :Default 0 0:カウンタを停止する 1:カウンタを起動する

18.2.2 コンペアデータレジスタ

アドレス	コンペアデータレジスタ
0x04	CMP[0]
0x24	CMP[1]

リード/ライト時

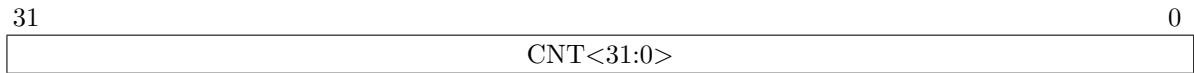
31	0
CMP<31:0>	

bit 名	機能
CMP<31:0>	Compare Data :Default X カウンタ値と比較す比較データを格納する．SEL bit が 0 の場合，カウンタがこの値と等しくなると割り込みを発生する．

18.2.3 カウンタレジスタ

アドレス	カウンタレジスタ
0x08	CNT[0]
0x28	CNT[1]

リード時

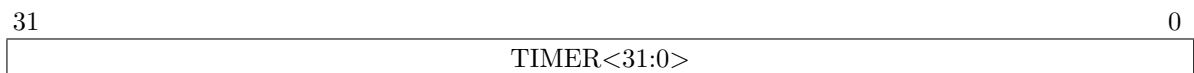


bit 名	機能
CNT<31:0>	Count Data :Default X ラッチパルスが入力された時にカウンタの値が本レジスタにラッチされる。

18.2.4 タイマレジスタ

アドレス	タイマレジスタ
0x0C	TIMER[0]
0x2C	TIMER[1]

リード/ライト時



bit 名	機能
TIMER<31:0>	Timer Data :Default X 周期割り込みに使用するタイマ値を設定する。カウンタクロックをカウントし本タイマ値と等しくなると、SEL bit が 1 の場合、割り込みを発生させる。

19

Real Time Clock

19.1 Outline

Real Time Clock Unit は外部クロックを用いてカレンダー機能を提供するユニットである．外部から供給されたクロックから 1 秒を計測し，各カウンタを制御する．カレンダー機能では年 (下位 2 ケタ)，月，日，曜日，時，分，秒を計測する．指定した日にち，時刻に割り込みを発生させるアラーム機能を持つ．また，うるう年に対応している．

さらに 1 秒単位で設定可能なタイマ機能を持つ．

クロックは外部ピン `rtc_clk` から入力する．デフォルトでは 32.768kHz を入力することにより 1 秒を計測するが，他の周波数であっても Clock Compare レジスタの値を変更することにより対応可能である．

外部ピン `rtc_hold` をアサートすることにより，プロセッサ内部バスからの信号を受け付けなくなる．これにより，プロセッサの他の部分の電力をカットし，バスの信号が不定になった場合でも Real Time Clock Unit は外部からの信号に影響されずに，正しく時間を計測し続けることが可能となる．

Real Time Clock の初期ベースアドレスは 0x3B000000 である．

19.2 Interface

19.2.1 Address Map

Second		offset: 0x00	
31		8 7 6	0
Reserved		U	Second

Field Name	Range	Description
Update (U)	7	前回値をリードしてから値が更新された場合に 1 がセットされる．リードすることにより 0 にクリアされる．
Second	6:0	秒がセットされる．値は BCD コードで表す．

Minute

offset: 0x04

31	8	7	6	0
Reserved				UMinute

Field Name	Range	Description
Update (U)	7	前回値をリードしてから値が更新された場合に 1 がセットされる．リードすることにより 0 にクリアされる．
Minute	6:0	分がセットされる．値は BCD コードで表す．

Hour

offset: 0x08

31	8	7	6	5	0
Reserved					U0Hour

Field Name	Range	Description
Update (U)	7	前回値をリードしてから値が更新された場合に 1 がセットされる．リードすることにより 0 にクリアされる．
Hour	5:0	時がセットされる．値は BCD コードで表す．

Week

offset: 0x0c

31	8	7	6	0
Reserved				UWeek

Field Name	Range	Description
Update (U)	7	前回値をリードしてから値が更新された場合に 1 がセットされる．リードすることにより 0 にクリアされる．
Week	6:0	曜日がセットされる．値は 0 ビット目から日曜日 6 ビット目が土曜日を示す．

Day

offset: 0x10

31	8	7	6	5	0
Reserved				U	0
				Day	

Field Name	Range	Description
Update (U)	7	前回値をリードしてから値が更新された場合に 1 がセットされる．リードすることにより 0 にクリアされる．
Day	5:0	日がセットされる．値は BCD コードで表す．

Month

offset: 0x14

31	8	7	6	5	4	0
Reserved				U	00	Month

Field Name	Range	Description
Update (U)	7	前回値をリードしてから値が更新された場合に 1 がセットされる．リードすることにより 0 にクリアされる．
Month	4:0	月がセットされる．値は BCD コードで表す．

Year

offset: 0x18

31	8	7	0
Reserved			Year

Field Name	Range	Description
Year	7:0	年の下 2 けたがセットされる．値は BCD コードで表す．

Second Alarm

offset: 0x20

31	8	7	6	0
Reserved				D Second

Field Name	Range	Description
Don't Care (D)	7	1 をセットすると、秒に関する条件を無視する。0 の場合、Second に指定した条件にマッチした場合にアラームの割り込みが発生する。
Second	6:0	アラームの秒を指定する。値は BCD コードで表す。

Minute Alarm

offset: 0x24

31	8	7	6	0
Reserved				D Minute

Field Name	Range	Description
Don't Care (D)	7	1 をセットすると、分に関する条件を無視する。0 の場合、Minute に指定した条件にマッチした場合にアラームの割り込みが発生する。
Minute	6:0	アラームの分を指定する。値は BCD コードで表す。

Hour Alarm

offset: 0x28

31	8	7	6	5	0
Reserved					D 0 Hour

Field Name	Range	Description
Don't Care (D)	7	1 をセットすると、時に関する条件を無視する。0 の場合、Hour に指定した条件にマッチした場合にアラームの割り込みが発生する。
Hour	5:0	アラームの時を指定する。値は BCD コードで表す。

Week Alarm

offset: 0x2c

31	8	7	6	0
Reserved			D	Week

Field Name	Range	Description
Don't Care (D)	7	1 をセットすると、曜日に関する条件を無視する。0 の場合、Week に指定した条件にマッチした場合にアラームの割り込みが発生する。
Week	6:0	アラームの曜日を指定する。1 を設定した曜日にアラーム割り込みを発生させる。

Day Alarm

offset: 0x30

31	8	7	6	5	0
Reserved			D	0	Day

Field Name	Range	Description
Don't Care (D)	7	1 をセットすると、日に関する条件を無視する。0 の場合、Day に指定した条件にマッチした場合にアラームの割り込みが発生する。
Day	5:0	アラームの日を指定する。値は BCD コードで表す。

Month Alarm

offset: 0x34

31	8	7	6	5	4	0
Reserved			D	00	Month	

Field Name	Range	Description
Don't Care (D)	7	1 をセットすると、月に関する条件を無視する。0 の場合、Month に指定した条件にマッチした場合にアラームの割り込みが発生する。
Month	4:0	アラームの月を指定する。値は BCD コードで表す。

Time

offset: 0x38 (Read Only)

31	24	23	16	15	8	7	0
Reserved		Hour		Minute		Second	

Field Name	Range	Description
Hour	23:16	Hour レジスタの内容 .
Minute	15:8	Minute レジスタの内容 .
Second	7:0	Second レジスタの内容 .

Date

offset: 0x3c (Read Only)

31	24	23	16	15	8	7	0
Week		Year		Month		Day	

Field Name	Range	Description
Week	31:24	Week レジスタの内容 .
Year	23:16	Year レジスタの内容 .
Month	15:8	Month レジスタの内容 .
Day	7:0	Day レジスタの内容 .

Mode

offset: 0x40

31	5	4	3	2	1	0
Reserved						TMPTTEAEN

Field Name	Range	Description
Test Mode (TM)	4	テストモード . 0 に設定すること .
Periodic Timer (PT)	3	1 を設定すると Periodic Timer , 0 を設定すると One Shot Timer になる .
Timer Enable (TE)	2	1 をセットするとタイマが作動する . タイマが Expire した場合 , One Shot Timer ならば自動的に 0 にクリアされる .
Alarm Enable (AE)	1	1 をセットするとアラームが動作する . アラームに指定した時間がくると割り込みが発生する .
Enable (EN)	0	1 をセットすると Real Time Clock が動作する .

Sense

offset: 0x44

31	2	1	0
Reserved			TI AI

Field Name	Range	Description
Timer Interrupt (TI)	1	タイマ割り込みが発生した場合に 1 がセットされる。1 を書き込むことによりクリアされる。
Alarm Interrupt (AI)	0	アラーム割り込みが発生した場合に 1 がセットされる。1 を書き込むことによりクリアされる。

Timer Compare

offset: 0x48

31	0
Timer Compare	

Field Name	Range	Description
Timer Compare	31:0	タイマ割り込みをかける秒数を指定する。

Timer Count

offset: 0x4c (Read Only)

31	0
Timer Count	

Field Name	Range	Description
Timer Count	31:0	現在のタイマのカウント数。この値が Timer Setup になるとタイマ割り込みが発生する。

Clock Compare

offset: 0x50

31	0
Clock Compare	

Field Name	Range	Description
Clock Compare	31:0	本モジュールに入力されているクロックの周波数を指定する。Clock Count がこのレジスタの値と等しくなった場合に 1 秒経過したと判定される。

Clock Count

offset: 0x54 (Read Only)

31	0
Clock Count	

Field Name	Range	Description
Clock Count	31:0	クロック毎にカウントアップされ , 1 秒を計測する . このレジスタの値が Clock Compare と等しくなった場合 , 1 秒経過したと判定される .

20

DMA Controller

- 32/16/8 bit I/F
- 入力チャネル：4
- 優先順位：固定優先度及びラウンドロビン
- Memory to memory 転送機能
- Bus sizing 機能 (8, 16bit I/O 用)
- Bus swapping 機能 (8, 16bit I/O 用)

20.1 レジスタマップ

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

20.1.1 DMA 制御レジスタ

ライト/リード

オフセット: 0x800	
31	1 0
-	
PRI	
bit 名	機能
PRI	PRiority :Default 0 本ビットは DMA チャンネルのプライオリティを示す 0:プライオリティはラウンドロビン 1:プライオリティは $ch0 > ch1 > ch2 > ch3$

20.1.2 DMA 割り込みクリアレジスタ

オフセット: 0x804	
31	1 0
-	
IC	
bit 名	機能
IC	Interrupt Clear 本ビットは DMA 割り込みのクリアを行う . 0:割り込みクリア

20.1.3 ポート / ソースアドレスレジスタ

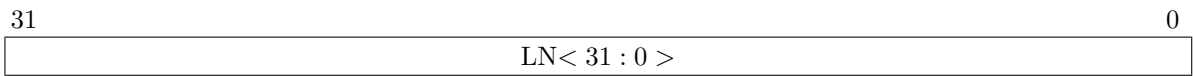
オフセット: $0x40 \cdot (x) + 0x04$	
31	0
PSA< 31 : 0 >	
bit 名	機能
PSA< 31 : 0 >	Port/Source Address :Default X チャンネル x の DMA に対し , 本ビットはメモリから I/O への転送の時 (MODE レジスタの MTM ビットが 0) ポートアドレスを示し , メモリからメモリへの転送の時 (MODE レジスタの MTM ビットが 1) ソースアドレスを示す .

20.1.4 メモリ / デスティネーションアドレスレジスタ

オフセット: $0x40 \cdot (x) + 0x08$	
31	0
MDA< 31 : 0 >	
bit 名	機能
MDA< 31 : 0 >	Memory/Destination Address :Default X チャンネル x の DMA に対し , 本ビットはメモリから I/O への転送の時 (MODE レジスタの MTM ビットが 0) メモリアドレスを示し , メモリからメモリへの転送の時 (MODE レジスタの MTM ビットが 1) デスティネーションアドレスを示す .

20.1.5 転送レングスレジスタ

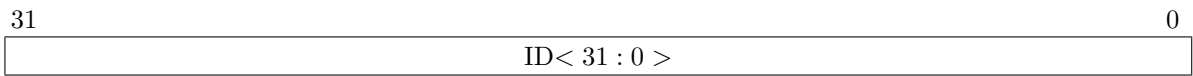
オフセット: $0x40 \times (x) + 0x18$



bit 名	機能
LN< 31 : 0 >	transfer LeNgtH :Default X チャンネル x の DMA に対し，本レジスタは転送レングスを示す．単位はバイトである．

20.1.6 データバッファレジスタ

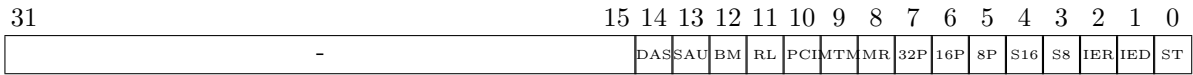
オフセット: $0x40 \times (x) + 0x0c$



bit 名	機能
ID< 31 : 0 >	Internal Data :Default X チャンネル x の DMA に対し，DMA 転送時一度内部のデータバッファにてメモリにライトするデータを組み立てるが，そのデータバッファの値が読める．どのロケーションに有効なデータがあるかはステータスレジスタを見る必要がある．

20.1.7 転送モード制御レジスタ

オフセット: $0x40 \times (x) + 0x10$



ライト/リード

bit 名	機能																
DAS	Destination Address Update :Default X 0:メモリアドレスレジスタで設定したアドレスが次の転送にも使用される． 1:メモリアドレスレジスタの値は，最後に転送を行ったアドレスより 1 ワード先を示す．																
SAU	Source Address Update :Default X 0:ポートアドレスレジスタで設定したアドレスが次の転送にも使用される． 1:ポートアドレスレジスタの値は，最後に転送を行ったアドレスより 1 ワード先を示す．																
BM	Burst Mode :Default X 0:バースト転送しない． 1:バースト転送する．																
RL	Responsive Link :Default X 1:レスポンスリンク用 DPM に対する DMA 転送を行う．																
PCI	PCI :Default X 1:PCI に対して DMA 転送を行う．																
MTM	Memory To Memory transfer :Default X 0:ポートアドレスレジスタで設定した I/O とメモリ間の DMA 転送であることを示す．転送方向は MR ビットにて指定する． 1:ソースアドレスからデスティネーションアドレスへ，レンジレジスタで設定したバイト数のデータを DMA 転送を行う．アドレスカウンタは UP 方向のみのカウントとする．また，4 バイトバウンダリでない転送領域及びレンジの DMA 転送は Memory To Memory ではサポートしない．																
MR	MR Memory Read :Default X 0:I/O からメモリへの転送であることを示す． 1:メモリから I/O への転送であることを示す．																
32P	32bit I/O Port :Default X 0:don't care 1:MTM ビットが 0 の時 32bit の I/O ポートとの転送であることを示す．この時，ポートアドレスのビット 1, 0 は無視される．																
16P	16P 16bit I/O Port :Default X 0:don't care 1:MTM ビットが 0 の時 16bit の I/O ポートとの転送であることを示す．この時，ポートアドレスのビット 0 は無視され，ビット 1 によりどのデータバスに接続されるか (D31-16 or D15-0) を示す．																
8P	8P 8bit I/O Port :Default X 0:don't care 1:MTM ビットが 0 の時 8bit の I/O ポートとの転送であることを示す．この時，ポートアドレスのビット 1, 0 によりどのデータバスに接続されるか (D31-24 or D23-16 or D15-8 or D7-0) を示す．																
S16	Swap at 16bit :Default X 0:don't care 1: 16bit 単位でデータのスワップを行う． 31 <table border="1"><tr><td>A</td><td>B</td><td>C</td><td>D</td></tr></table> 0 → 31 <table border="1"><tr><td>C</td><td>D</td><td>A</td><td>B</td></tr></table> 0	A	B	C	D	C	D	A	B								
A	B	C	D														
C	D	A	B														
S8	Swap at 8bit :Default X 0:don't care 1: 8bit 単位でデータのスワップを行う． 31 <table border="1"><tr><td>A</td><td>B</td><td>C</td><td>D</td></tr></table> 0 → 31 <table border="1"><tr><td>B</td><td>A</td><td>D</td><td>C</td></tr></table> 0 S16=1,S8=1 をセットすると以下のようにスワップされる． 31 <table border="1"><tr><td>A</td><td>B</td><td>C</td><td>D</td></tr></table> 0 → 31 <table border="1"><tr><td>D</td><td>C</td><td>B</td><td>A</td></tr></table> 0	A	B	C	D	B	A	D	C	A	B	C	D	D	C	B	A
A	B	C	D														
B	A	D	C														
A	B	C	D														
D	C	B	A														
IER	Interrupt enable of ER-bit :Default 0 0:割込みを発生しない． 1:割込み発生を許可する．																
IED	Interrupt enable of ED-bit :Default 0 0:割込みを発生しない． 1:割込み発生を許可する．																
ST	Start :Default 0 0:DMA 転送を停止させる． 0 をライト後 DMAC は初期化される． 1:DMA 転送を起動する．																

本レジスタで設定できるモードは次ページの通りであり，それ以外の設定では動作の保証はしない．

転送モード		スワップなし (S16=0,S8=0)	スワップあり (S16=0,S8=1)	スワップあり (S16=1,S8=0)	リトルエンディアン (S16=1,S8=1)
メモリ (32bit)	メモリ (32bit)	○	×	×	○
メモリ (32bit)	I/O 32bit(D31-0)	○	○	○	○
	I/O 16bit(D31-16)	○	×	×	○
	I/O 16bit(D15-0)	○	×	×	○
	I/O 8bit(D31-24)	○	×	×	○
	I/O 8bit(D23-16)	○	×	×	○
	I/O 8bit(D15-8)	○	×	×	○
	I/O 8bit(D7-0)	○	×	×	○
メモリ (D31-16)	I/O 8bit(D31-24)	○	×	×	○

20.1.8 ステータスレジスタ

オフセット: $0x40 \times (x) + 0x14$ ライト/リード

31	30	29	28	27											2	1	0
L0	L1	L2	L3		-										ER	ED	

bit 名	機能
L0	Location 0 :Default 0 0:内部データレジスタの D31-24 は有効なデータでない . 1:内部データレジスタの D31-24 は有効なデータである .
L1	Location 1 :Default 0 0:内部データレジスタの D23-16 は有効なデータでない . 1:内部データレジスタの D23-16 は有効なデータである .
L2	Location 2 :Default 0 0:内部データレジスタの D15-8 は有効なデータでない . 1:内部データレジスタの D15-8 は有効なデータである .
L3	Location 3 :Default 0 0:内部データレジスタの D7-0 は有効なデータでない . 1:内部データレジスタの D7-0 は有効なデータである .
ER	Error :Default 0 0:don 't care 1:DMA 転送中にエラーが発生して DMA 転送が停止したことを示す . 本ビットは 0 をライトするとクリアされる .
ED	END :Default 0 0:don 't care 1:DMA 転送が終了すると 1 に設定される . 本ビットは 0 をライトするとクリアされる .

21

PCI Host Controller

21.1 Outline

Control Register Base Address:0x5000_0000

Initiator Memory Base Address:0x5400_0000

21.2 Bridge Control Register Block

The bridge control register block consists of the 64-byte PCI configuration space register block and a bridge control registers. The registers form a single block which is accessible both from the PCI bus and the host processor bus. In the PCI space the block is mapped in the configuration space. In the host bus the block is mapped as a memory block. The host bus interface implementation depends on the host bus type.

offset	31	24	23	16	15	8	7	0
0x00	Device ID				Vendor ID			
0x04	Status Register				Command Register			
0x08	Class Code						Revision ID	
0x0C	BIST		Header Type		Latency Timer		Cache Line Size	
0x10	Base Address Register 0							
0x14	Base Address Register 1							
0x18	Base Address Register 2							
0x1C	Base Address Register 3							
0x20	Base Address Register 4							
0x24	Base Address Register 5							
0x28	Card Bus CIS Pointer							
0x2C	Subsystem ID				Subsystem Vendor ID			
0x30	Expansion ROM Base Address Register							
0x34	Reserved							
0x38	Reserved							
0x3C	Maximum Latency		Minimum Grant		Intterupt Pin		Interrupt Line	
0x40	Bridge ID Register							
0x44	Bridge Status Register				Bridge Control Register			
0x48	PCI Address Pointer				PCI Transfer Counter			
0x4C	PCI Address Pointer							
0x50	PCI Transfer Counter							
0x54	Reserved		Reserved		Reserved		PCI Command Register	
0x58	Initiator Dual-port Memory Data Pointer							

21.2.1 PCI Configuration Space

Vendor ID This is a 16-bit read-only register that identifies the manufacturer of the device. The value of this register is assigned by the PCI SIG.

Device ID This is a 16-bit read-only register that identifies the device type.

Command Register This is a 16-bit read/write register that provides basic control of the PCI function to respond to the PCI bus and/or access it. See Table 2-3 for a detailed description.

Status Register This is a 16-bit register that provides the status of bus-related events. Read transactions from the status register behave normally. However, write transactions are different from typical write transactions because bits in the status register can be cleared but not set. A bit in the status register is cleared by writing a logic one to that bit. For a detailed description see Table 2-4.

Revision ID Register This is an 8-bit read-only register that identifies the revision number of the device. The value of this register is assigned by the manufacturer (designer). Revision ID is defined in the pci_hb_params by the constant REV_ID.

Class Code Register This is a 24-bit read-only register divided into three sub-registers: base class, sub- class, and programming interface. Class code is defined in the pci_hb_params by the constant CLASS_ID.

Cache Line Size Register This specifies the system cache line size in DWORDS. This read/write register is initialized by system software at power-up.

Latency Timer Register This is an 8-bit register. The register defines the maximum amount of time, in PCI bus clock cycles, that the PCI function can retain ownership of the PCI bus.

Header Type Register This is an 8-bit read-only register that identifies the PCI function as a single- function device. The PCI-HB Core supports only header type 00.

Base Address Registers - The PCI function supports up to six BARs. Each base address register (BAR_n) has identical attributes. The BAR is formatted per the PCI Local Bus Specification, Revision 2.2. Bit 0 of each BAR is read only and it is used to indicate whether the reserved address space is memory or I/O. BARs that map to

memory space must hardwire bit 0 to 0, and BARs that map to I/O space must hardwire bit 0 to 1. The format of the BAR changes depending on the value of bit 0.

Interrupt Line Register This is an 8-bit register that defines to which system interrupt request line the INTA# output is routed. The Interrupt Line Register value is defined in the pci_hb_params by the constant INT_LINE. The Host Bridge does not use the PCI interrupt line, thus the value is set to zero.

Interrupt Pin Register This is an 8-bit read-only register that defines the PCI function PCI bus interrupt request line to be INTAn. The Interrupt Pin Register value is defined in the pci_hb_params by the constant INT_PIN.

Minimum Grant Register This is an 8-bit read-only register that defines the length of time the function would like to retain mastership of the PCI bus. The value set in this register indicates the required burst period length in 250-ns increments. The minimum Grant value is defined in the pci_hb_params by the constant MIN_GNT.

Maximum Latency Register This is an 8-bit read-only register that defines the frequency in which the function would like to gain access to the PCI bus. Maximum Latency value is defined in the pci_hb_params by the constant MAX_LAT.

Command Register

offset: 0x04

Field Name	Range	Description
IO_EN	0	I/O access enable. Controls a device' s response to I/O accesses. When high, it lets the device respond to the PCI bus I/O accesses as a target.
MEM_EN	1	Memory access enable. Controls a device' s response to Memory space accesses. When high, it lets the device respond to the PCI bus memory accesses as a target.
MASTER_EN	2	Master enable
SPEC_CYC	3	Special Cycle Enable. Controls a device' s action on the Special Cycle operation. The current version of the PCI Interface Core doesn' t support this feature, however it is possible to add Special Cycle support by the user.
MWLEN	4	Memory write and invalidate enable.
VGA_PS	5	VGA Palette Snoop. Currently unused.
PERR_EN	6	Parity error enable. When high, it enables the function to report parity errors via the PERR# output.
STEP_EN	7	Stepping Enable. The current version of the PCI Core doesn' t support address/data stepping.
STERR_EN	8	System error enable. When high, it allows the function to report address parity errors via the SERR# output.

Status Register

offset: 0x06

Field Name	Range	Description
	3:0	Reserved.
CAP_EN	4	Read Capabilities list enable. When set, this bit enables the capabilities list pointer register at offset 34h.
PCI66M	5	Read PCI 66-MHz capable. When set, it indicates that the PCI device is capable of running at 66 MHz.
	7:6	Reserved.
MDPERR_DET	8	Master Data Parity Error. When high, it indicates that during a read transaction the function asserted the PERRn output as a master device, or that during a write transaction the PERRn output was asserted as a target device.
DEVSEL_TIM	10:9	Device select timing. The devsel_tim bits indicate target access timing of the function via the DEVSELn output. The PCI-HB Core is designed to be medium speed target device ("01" b)
TABORT_SIG	11	Signaled target abort. This bit is set when a local peripheral device terminates a transaction. Currently not used.
TABORT_DET	12	Detected Target Abort. When high, it indicates that the function in master mode has detected a target abort from the current target device
MABORT_SIG	13	Master abort. When high, MABORT_SIG indicates that the function in master mode has terminated the current transaction with a master abort.
SERR_SIG	14	Signaled system error. When high, SERR_SIG indicates that the function drove the SERR# output active, i.e., an address phase parity error has occurred. The function signals a system error only if an address phase parity error was detected and SERR_EN was set.
PERR_DET	15	Detected parity error. When high, PERR_DET indicates that the function detected either an address or data parity error. Even if parity error reporting is disabled (via perr_ena), the function sets the PERR_DET bit.

21.2.2 Bridge ID Register (0x40)

Bridge ID register uniquely identifies type of the bridge. The ID has encoded the host bus type and core revision.

Bridge ID Register

offset: 0x40

Field Name	Range	Description
version	7:0	PCI-Host bridge version
subtype	11:8	Bridge subtype
type	15:12	Bridge type. This defines a bridge type 0 generic bridge 1 PCI-AMBA bridge 2 PCI-CoreConnect bridge 3 PCI-OCF bridge

21.2.3 Bridge Control Register (0x44)**Bridge Control Register**

offset: 0x44

Field Name	Range	Description
genpcirst	0	PCI reset. This bit generates a PCI bus reset.
initena	1	Enable initiator transfer.

21.2.4 Bridge status register (0x46)

offset:

Field Name	Range	Description
pciready	0	PCI ready. This bit indicates that a PCI bus reset was completed. The bit is clear when the PCI bus reset is in progress.
reserved	4:1	
ptxdone	5	PCI transfer done.
ptxrerr	6	PCI transfer error.

21.2.5 Interrupt Status Register (0x4A)

Interrupt Status Register

offset: 0x4A

Field Name	Range	Description
serrsts	0	SERR status. This bit indicates that a system error was reported on the PCI bus. The bit is set by SERR# line assertion. The bit is cleared by writing a ' 1 ' value to this bit.
intasts	1	INTA# status. This bit indicates an interrupt request on INTA# interrupt line. The bit is equal to ' 1 ' when the line is asserted.
intbsts	2	INTB# status. This bit indicates an interrupt request on INTB# interrupt line. The bit is equal to ' 1 ' when the line is asserted.
intcsts	3	INTC# status. This bit indicates an interrupt request on INTC# interrupt line. The bit is equal to ' 1 ' when the line is asserted.
intdsts	4	INTD# status. This bit indicates an interrupt request on INTD# interrupt line. The bit is equal to ' 1 ' when the line is asserted.

21.2.6 Interrupt Mask Register (0x48)

Interrupt Mask Register

offset: 0x48

Field Name	Range	Description
serrimask	0	System error interrupt mask.
intaimask	1	INTA# interrupt mask.
intbimask	2	INTB# interrupt mask.
intcimask	3	INTC# interrupt mask.
intdimask	4	INTD# interrupt mask.
txtimask	5	PCI transfer finished interrupt mask.
txterrimask	6	PCI transfer error interrupt mask.

21.2.7 PCI Address Pointer (0x4C)

The contents of the PCI address pointer register depends on a PCI transaction type. There are four basic types of PCI transaction:

- Memory space access transaction is used for accessing PCI memory space.

31	2	1	0
Address			00

- I/O space access transaction is used for accessing PCI I/O space. The address bits [1:0] must be consistent with byte enables in PCI Command register. For more details about the valid address bit encoding see the Table 2-10.

31	0
Address	

AD[1:0]	Stating byte	Valid BE#[e:0]
00	Byte 0	xxx0 or 1111
01	Byte 1	xx01 or 1111
10	Byte 2	x011 or 1111
11	Byte 3	0111 or 1111

- Configuration space access type 01 transaction is used to access devices that reside on another bus (behind a PCI-to-PCI bridge). A PCI-to-PCI bridge translates the access type 01 to access type 00 on the target bus. The format of the PCI address register for the configuration space access type 01 is described in Figure 2-9 and Table 2-11. The PCI-Host bridge core sends the address register unchanged to the PCI bus.

Field Name	Range	Description
01	1:0	Type 01 identifier
Register number	7:0	This is an encoded value to select a DWORD in the configuration space of a target device.
Function number	10:8	This is a function number used to select a configuration space within a multifunction device.
Device number	15:11	This is a device number used to select a device on a given bus
Bus number	23:16	This is a bus number where the device has to be accessed.

- Configuration space access type 00 transaction is used to select a device on the bus where the transaction is being run. The format of the PCI address register for the configuration space access type 00 is described in Figure 2-10 and Table 2-12. The PCI-Host bridge core converts the address register contents to the PCI bus address as shown in Figure 2-10. The AD[31:11] lines can be used as an IDSEL signal for PCI devices or slots. The bridge is using the AD[31:16] address lines for the IDSEL encoding.

Field Name	Range	Description
00	1:0	Type 00 identifier
Register number	7:0	This is an encoded value to select a DWORD in the configuration space of a target device.
Function number	10:8	This is a function number used to select a configuration space within a multifunction device.
Device number	15:11	This is a device number used to select a device on a given bus

21.2.8 PCI Transfer Counter (0x50)

The PCI transfer counter register defines a number of data transfers to be completed. The number is DWORD aligned however a number of bytes can be reduced by byte enables in the PCI command register. The maximum number of data to be transferred is limited by the size of the dual-port memory.

31	2	1	0
Count			00

21.2.9 PCI Command Register (0x54)

The PCI command register defines a transaction command and a byte enable for the transaction.

31	7	4	3	0
reserved		cmd	ben	

C/BE#[3:0]	Command
0000	Interrupt Acknowledge
0001	Special Cycle
0010	I/O Read
0011	I/O Write
0100	Reserved
0101	Reserved
0110	Memory Read
0111	Memory Write
1000	Reserved
1001	Reserved
1010	Configuration Read
1011	Configuration Write
1100	Memory Read Multiple
1101	Dual Access Cycle
1110	Memory Read Line
1111	Memory Write and Invalidate

21.2.10 Initiator Dual-port Memory Data Pointer (0x58)

31	2	1	0
data pointer			00

22

On-Chip Emulator

22.1 Outline

On-Chip Emulator は SPI 通信でのシングルワードのリード・ライトのエミュレーター機能を提供する。

22.2 Operation

On-Chip Emulator は適当な SPI マスターと 1bit の General Purpose I/O の RELOAD ピンに接続される。

SPI の動作モードはモード 1(同期クロックは正極性で立ち上がりでデータを受け取る) でアクセスし、MSB から転送を開始するように設定し、RELOAD ピンには 1 を入力する。

22.2.1 Single Write

シングルライト転送を行うにはまず、SPI マスターから 8bit のコマンドデータ 0xAB を入力し、転送完了後に RELOAD ピンに 0 を入力する。

次にアドレスデータを MSB から 8bit 入力し、転送完了後に RELOAD ピンに 0 を入力するを 4 回繰り返し、その後同様にライトデータを MSB から 8bit 入力し、転送完了後に RELOAD ピンに 0 を入力するを 4 回繰り返すことで On-Chip Emulator がシングルライト転送を開始する。

22.2.2 Single Read

シングルリード転送を行うにはまず、SPI マスターから 8bit のコマンドデータ 0xAA を入力し、転送完了後に RELOAD ピンに 0 を入力する。

次にアドレスデータを MSB から 8bit 入力し、転送完了後に RELOAD ピンに 0 を入力するを 4 回繰り返すことで On-Chip Emulator がシングルリード転送を開始する。

MISO の I/O ピンから 0 から 1 が出力されるとリードデータの準備が完了する。RELOAD ピンに 0 を入力すると MISO の I/O ピンからリードデータが MSB から 8bit 出力されるのでこれを 4 回繰り返すことでリードデータを出力する。

23

Responsive Link

23.1 概要

Responsive Link は、各種ロボット、自動車、プラント、ホームオートメーション等の種々の分散制御を実現するために必要なハードリアルタイム通信、及び、画像、音声等のマルチメディアデータを滑らかに伝送するために必要なソフトリアルタイム通信の両方を同時に可能にするように設計を行っている。特に、リアルタイムの理論をそのまま応用可能なように、パケットの追い越し機能を実現している。

Responsive Link は柔軟なリアルタイム通信を実現するために、

- 通信パケットに優先度を付け、高い優先度の通信パケットが低い優先度の通信パケットを通信ノード毎に追い越し
- ハードリアルタイム通信（データリンク）とソフトリアルタイム通信（イベントリンク）の分離
- 全く同じネットワークアドレス（送信元アドレス及び送信先アドレス）を持つ通信パケットの経路を優先度によって別の経路に設定することによって専用回線や迂回路を設け実時間通信を制御
- 通信パケットの優先度を通信ノード毎に付け替え可能にすることによってパケットの加減速を分散管理で制御
- ハードウェアによるフレーム単位のエラー訂正

という方法を組み合わせることによって、分散管理を用いて大規模かつ量子時間の小さい実時間通信を実現する。さらに、

- 通信速度を動的に変更可能
- トポロジーフリー
- Hot-Plug&Play

等の様々な機能を実現する。

Responsive Link は国内では情報処理学会試行標準（IPSSJ-TS 2003:0006）として標準化されており、国際的にはでは ISO/IEC JTC1 SC25 WG4 において標準化作業が行われている。

23.2 インタフェース

ソフトリアルタイム通信（以下，単にデータと呼ぶ）のデータサイズ（画像データ，音声データ等）は大きく，それに対してハードリアルタイム通信（以下，単にイベントと呼ぶ）のデータサイズ（制御コマンド，同期信号等）は非常に小さい．従って，従来型の 1 系統の通信路で全ての通信を行う方法では，同時に通信すべき通信データとして，大量のデータパケットと，ごくわずかではあるが分散リアルタイム制御用途には非常に重要なイベントパケットが同一種類のパケットとして存在する．データとイベントを，共有された同一の通信線を通して時分割に通信を行う従来方式ではイベント伝達の時間が正確にバウンドできないので，ハードリアルタイムシステムは実現困難であると考えられる．

また，複数のモジュールでひとつの通信チャネルを共有するシリアルバスでは，同時に何台のモジュールが通信するかによってバンド幅が動的に変化し時間をバウンドすることが困難であり，実効速度も出にくい．

さらに，リアルタイム通信におけるトレードオフとして，ソフトリアルタイム通信は主にバルク的なマルチメディアデータの通信等に用いられ，ハードリアルタイム通信は主に制御等に用いられるので，

- ソフトリアルタイム：バンド幅保証 ⇒
スループットをできるだけ上げたい
- ハードリアルタイム：レイテンシ保証 ⇒
レイテンシをできるだけ小さくしたい

という要求がある．しかしながらパケットサイズを大きくするとスループットは高くなるが，同時にレイテンシも長くなる．逆にパケットサイズを小さくするとレイテンシは短くなるが，オーバーヘッドが大きくなりスループットが低くなる．

従って，*Responsive Link* では，データラインとイベントラインを分離し，かつ各ラインの結合形態を point-to-point の双方向シリアル通信として設計されている（図 23.1 参照）．以下，それぞれをデータリンク，イベントリンクと呼ぶ．データリンクではパケットサイズを固定長かつ大きめにしてソフトリアルタイム通信に使用し，イベントリンクではパケットサイズを固定長かつ小さめにしてハードリアルタイム通信に使用する．

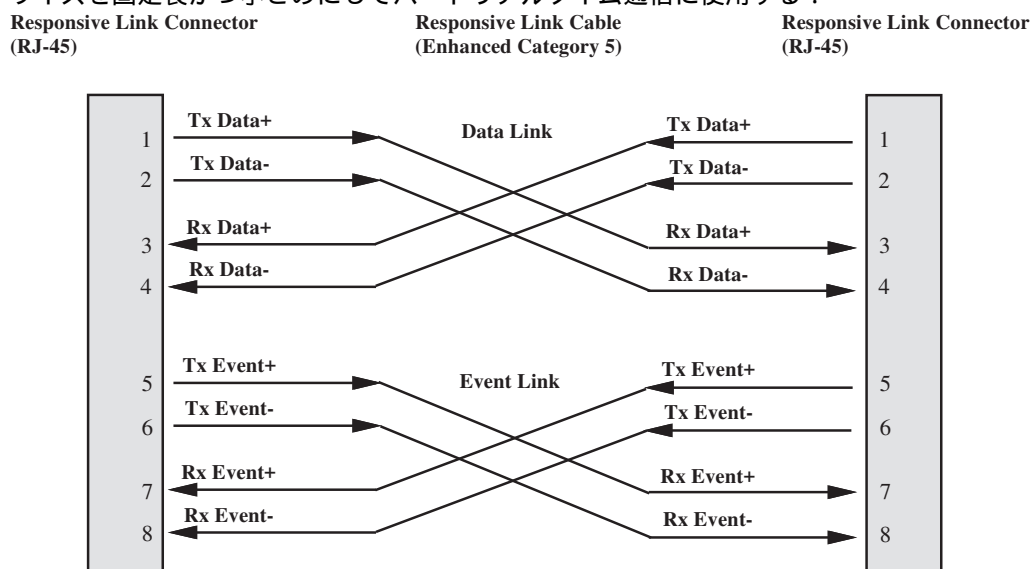


図 23.1: *Responsive Link* インタフェース

図 23.2 の通信パケットのヘッダ部に対して，図 23.3 に示すようにネットワークアドレスに優先度を付加する．256 レベル (8bit) の優先度を有し，優先度は 0 が一番低く，数字が大きくなるにしたがって高くなる．

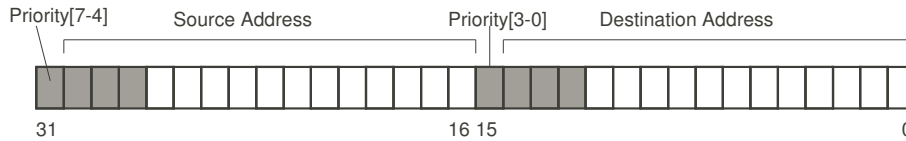


図 23.3: *Responsive Link* のヘッダフォーマット

Responsive Link の最大通信ノード数は，ネットワークアドレス長に制限され，優先度を使用しない場合，理論的には 2^{32} ノードとなる (図 23.3 参照)．*Responsive Link* の規格で推奨している使用法 (ノード毎にノードアドレスを割り当て，12bit の送信元アドレス，12bit の送信先アドレス，8bit の優先度を用いてルーティングを行う) の場合には， $2^{12} = 4096$ ノードとなる．4096 よりノード数が大きなシステムを構築する際には，経路にアドレスを割り当てる (24bit のネットワークアドレスと 8bit の優先度を用いてルーティングを行う) ことにより $2^{24} = 16\text{M}$ ノードまでのノード数をサポートする．

23.3.1 固定長 (64B) のデータパケット

レスポンスリンクのスイッチ部はカットスルー型のスイッチを採用している．データパケットは固定長 (64byte) で，パケットに優先度が付加されている．データパケットはアドレス (ソースとデスティネーション)，ペイロード，ステータスから構成される．カットスルー型のスイッチなので，衝突が起きない限りデータはノードを経由して転送されるが，あるノードで衝突が起こった場合は，優先度の高いパケットが低いパケットを追い越すことができるようになっていいる．この機能によって従来までの集中管理型ではなく分散管理型のリアルタイム通信を実現している．

データパケットは，2byte の送信元アドレス・2byte の送信先アドレス・56byte のペイロード・4byte の制御・状態データの計 64byte より構成される．4byte の制御・状態データは以下のフォーマットをとる．

UD	ユーザ定義フラグ (任意に設定可能)
Full	ペイロード 56byte がすべて有効データで埋められているとき 1，それ以外は 0
Data Length	ペイロードの有効データ長．1 から 56 の値をとる．
Dirty0-15	パケットのどのワード (4byte) にエラーが存在するかを示すビット．パケットの 2 ワード目にエラーがある場合は Dirty1 が 1 となる (ハードウェアによりセットされる)
Start	このパケットがスタートパケットであるとき 1，それ以外は 0
End	このパケットがエンドパケットであるとき 1，それ以外は 0
Int	このパケットを受け取る際に割り込みを生じるときは 1，それ以外は 0
Fatal	このパケットに致命的なエラーが存在するときは 1，それ以外は 0 (ハードウェアによりセットされる)
Correct	このパケットの一部分にエラーが存在し，それが修復されたときは 1，それ以外は 0 (ハードウェアによりセットされる)
Serial Number	パケットのシリアルナンバ．スタートパケットが 0，以降 0 から 7 までを繰り返す．

23.3.2 固定長 (16B) のイベントパケット

イベントパケットも固定長 (16byte) で，送信元アドレス，送信先アドレス，ペイロード，ステータスから構成される．イベントの場合もノードで衝突がない限り，直接ノードを経由してルーティングされるが，衝突が生じた場合はデータの場合と同様に，優先順位に従ってパケットの追い越しを行なう．

4byte の制御・状態データは以下のフォーマットをとる．

UD	ユーザ定義フラグ（任意に設定可能）
Full	ペイロード 8byte がすべて有効データで埋められているとき 1，それ以外は 0
Data Length	ペイロードの有効データ長．1 から 8 の値をとる．
Dirty0-15	パケットのどのバイトにエラーが存在するかを示すビット．パケットの 2 バイト目にエラーがある場合は Dirty1 が 1 となる（ハードウェアによりセットされる）
Start	このパケットがスタートパケットであるとき 1，それ以外は 0
End	このパケットがエンドパケットであるとき 1，それ以外は 0
Int	このパケットを受け取る際に割り込みを生じるときは 1，それ以外は 0
Fatal	このパケットに致命的なエラーが存在するときは 1，それ以外は 0（ハードウェアによりセットされる）
Correct	このパケットの一部分にエラーが存在し，それが修復されたときは 1，それ以外は 0（ハードウェアによりセットされる）
Serial Number	パケットのシリアルナンバ．スタートパケットが 0，以降 0 から 7 までを繰り返す．

23.3.3 優先度による追い越し機構

優先度を用いたパケットの追い越し機構を実現するために，追い越し用バッファと退避用外部記憶を有したネットワークスイッチを搭載している．図 23.4 は 5 入力 5 出力で一つの入力部当たり追い越し用バッファが 4 パケット分あるネットワークスイッチの構成を示している（実際に RMTP に実装されている *Responsive Link* には 8 パケット分の追い越し用バッファが実装されている．）図 23.4 において，最後の数字はポート番号を示している．入力ポート (In0～4) から入力された通信パケットは，通信ノードで衝突しない場合，そのまま出力ポート (Out0～4) へ出力を行う．異なる入力ポートから入力された通信パケットが同じ出力ポートに出力を行なう場合，通信パケットに付加された優先度に従い，低い優先度の通信パケットを追越し用バッファ（意味的には追い越され用バッファ）に貯めて出力を待たし，高い優先度の通信パケットを先に出力させる．高い優先度の通信パケットの出力の後に低い優先度の通信パケットを追越し用バッファから出力ポートに出力し，優先度に従った通信パケットの追い越しを行う．

この際，内部のスイッチングは，ヘッダ部受信のオーバーヘッド及びルーティングテーブルの参照時間を隠蔽するために図 23.4 のように 8bit パラレル（byte 単位）で行うように設計されている．

上記の通信パケットの追い越しを実現するために通信パケットの大きさと等しい追い越し用バッファを 8 本入力ポート側に搭載している．さらに，出力が待たされ続けている時に入力が入り続けバッファが溢れそうになった場合に，追い越し用バッファの内容を一時的に退避するための退避用外部記憶 (DDR SDRAM) を設けることができるようになっている．

図 23.5 は図 23.4 のネットワークスイッチのひとつの入力部の詳細を示している．図 23.5 において，最後の数字はポート番号を示している．通信パケットの追い越しを行うために，まず，入力ポート (In) から入力された通信パケットを，入力ポインタ (In-Pointer) で指し示されている追い越し用バッファ 0 から追い越し用バッファ 3 のうち使用されていない空バッファに書き込む．入力パケットのヘッダ部分は必ず全て受信し追い越し用バッファに書き込み，その受信されたヘッダを元に図 23.6 のようなルーティングテーブルを参照し出力ポート番号と優先度を得る．得られた出力ポート番号は図 23.5 のリンクストローブ (L0～L4) に書き込む．例えば L2 ビットが有効であればその入力パケットの出力先は出力ポート 2 であることを示す．

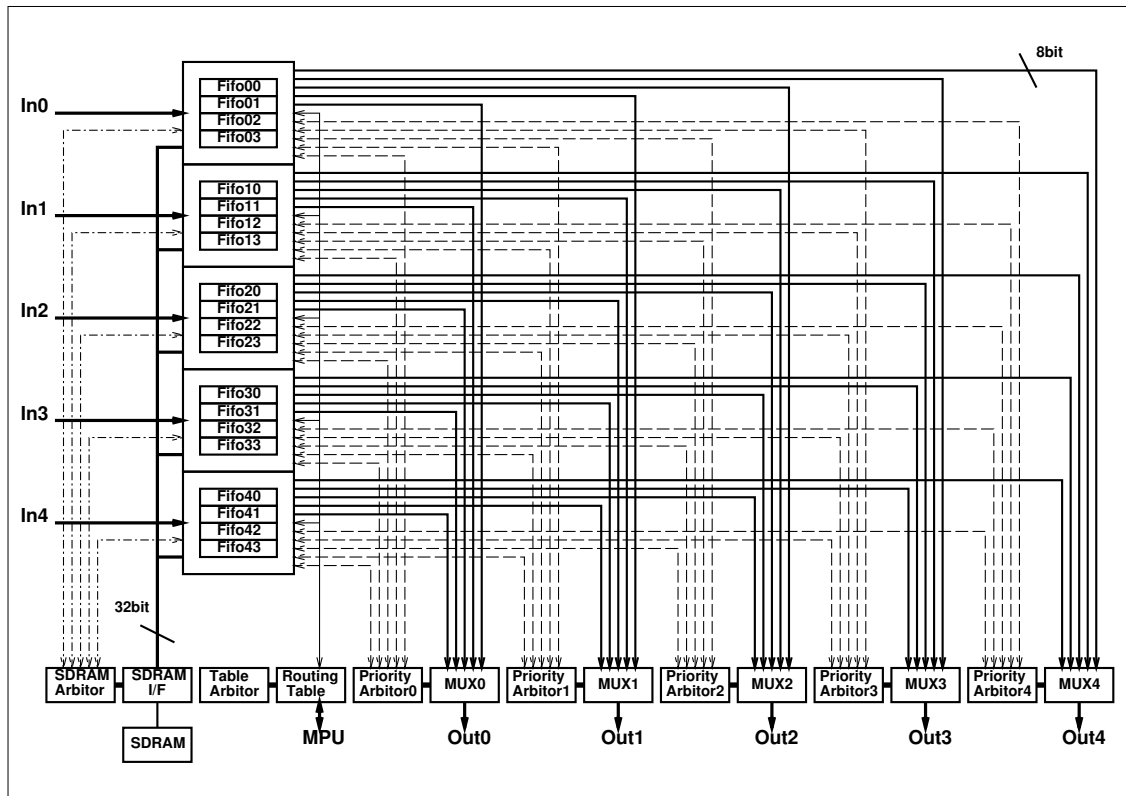


図 23.4: Responsive Link のネットワークスイッチ

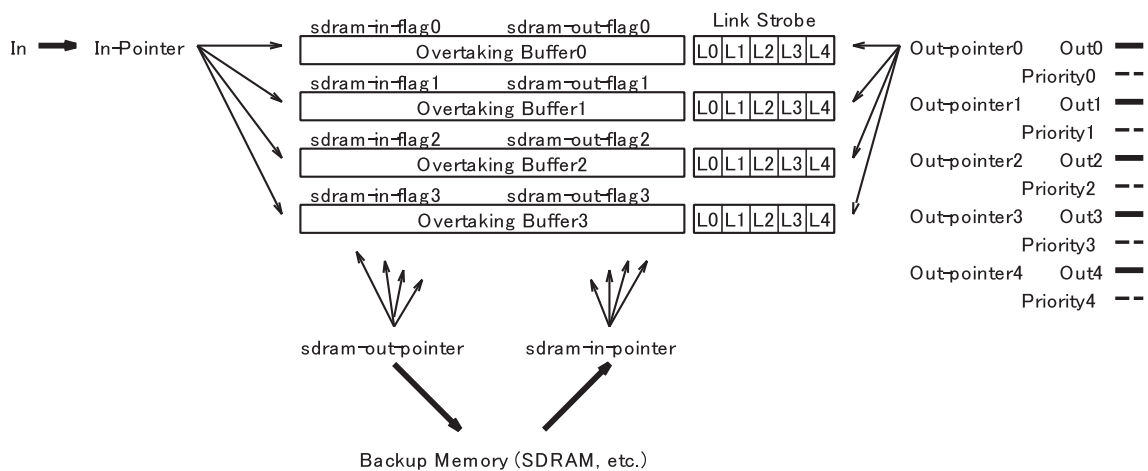


図 23.5: Responsive Link の追い越し用バッファ

図 23.5 において L0 から L4 までの複数ビットが有効であればマルチキャストを意味し、全て有効であればブロードキャストを意味する。入力部の出力側は出力ポート毎 (Out0 ~ Out4) にそれぞれ独立に各追い越し用バッファのリンクストローブを参照し、自出力ポートのリンクストローブが有効な場合、出力側ポート側に配置された当該優先度調停器 (図 23.4 の Priority ArbitratorN) に対して優先度と共に出力要求を行なう。図 23.5 の PriorityN は図 23.4 の Priority ArbitratorN に接続されている。優先度調停器は、出力要求が一つの入力ポートからだけある場合はただちに出力許可を与え、出力要求が複数ある場合は優先度の一番高いものに出力許可を与えるようにする。一番優先度の高い要求が複

数ある場合は、ラウンドロビン方式で出力許可を与える。

通信パケットの衝突がない場合や、衝突があってもその時点での最高優先度の通信パケットの場合は、ヘッダの受信とルーティングテーブル参照の遅延時間後に直ちに出力を開始する。入力部の各出力ポート側ではパケットの送信終了直後に対応するリンクストローブを無効にし、全てのリンクストローブが無効になったらそのバッファが空であることを意味する。

例えば、In-pointer が追い越し用バッファ1 を指している場合、入力ポート In から入力されたパケットは、まずヘッダ部が追い越し用バッファ1 に入る。次にそのヘッダを元にルーティングテーブルを引き、リンクストローブと優先度を得る。例えば、L1 と L3 が有効だった場合、Out-pointer1 と Out-pointer3 は共にその追い越し用バッファ1 を指し、Out1 と Out3 側が出力要求と共にその優先度をそれぞれ Priority1 と Priority3 に出力する。例えば、Out3 にすぐに出力可能であれば、出力許可が Priority Arbitor3 から与えられるので、直ちに追い越し用バッファ1 から Out3 に出力を開始する。出力が終われば、Out3 側が追い越し用バッファ1 の L3 をクリアする。また、Out1 には直ちに出力許可がおりなかったとすると、出力許可が得られるまで出力要求と優先度を Priority1 に出力し続ける。ここで、Out1 への出力待ちの状態で、同じく Out1 へ出力したい高優先度パケットが新たに追い越し用バッファ2 に入ってきた場合、Out-pointer1 はより優先度の高いパケットの入っている追い越し用バッファ2 を指すようになり、その高優先度パケットの出力要求と優先度を Priority1 に出力するようになる。後から到着した高優先度パケットの出力が終わると、他に Out1 に出力したい高優先度パケットがない場合、Out-pointer1 は再び追い越し用バッファ1 を指して、同様に出力を継続しようとする。このように、同一系路上の先行する低優先度パケットが待たされている際にも、後続の高優先度パケットが追い越していくことを可能にする。

図 23.5 において、空バッファが少なくなってい残り 1 本になってしまった場合、次の入力パケットは退避用外部記憶 (DDR SDRAM) に退避を行うようになっている。出力が進んで空バッファの残りが多くなり 2 本以上になると、退避用外部記憶に退避されていた入力パケットを優先度を考慮して追い越し用バッファに書き戻すことにより、出力を継続する。

また、退避用外部記憶が溢れそうになると、そのノードのプロセッシングコアに対して割り込みをかけられるようになっている。退避用外部記憶が溢れる場合は、アドミッションコントロールを行ってパケットの破棄を行ったり、送信元に送信データの一時停止を行うように制御する等の方法が考えられるが、そのプロトコル自身は *Responsive Link* の規格では定めていない。それらは上位のプロトコルで行うことになるので、上記割り込みをかける閾値を設定可能にするように設計している。

リアルタイム通信を実現するために、優先度によるパケットの追い越しをこのように再送を行なわなくてよいように設計されている。

23.4 フレームフォーマット

1byte は、図 23.2 の Frame Format ような冗長ビットを含めたフレームとしてシリアルに送受信される。詳細は低レベル通信の節を参照。

Data bits 8bit のデータ

Redundancy bits byte 毎に Redundancy bits (冗長ビット) を付加することで、CRC 等とは異なり、パケット全てを受信し終わらなくても byte 毎にエラー訂正が可能

23.5 ルーティング・テーブル

Responsive Link の経路制御は、図 23.6 に示すようなルーティングテーブル (経路制御表) を設定することによって行う。ルーティングテーブルは、*Responsive Link* コントローラ内に置き、そのノードのローカルプロセッサから読み

書きできるようになっている．図 23.6 において，Reference 部はパケットのヘッダと同一であり，Referent 部に当該パケットに関する設定を行う．EE ビット及び DE ビットは，それぞれそのラインがイベントリンク用の設定かデータリンク用の設定かを示す．両方とも設定されていれば，両リンクとも同様の設定になる．L[4-0] は，前述のリンクストロービットであり，出力ポート（複数可）を指し示す．

ルーティングテーブルの大きさ（エントリ数）は実装依存で有限となるため，非常に大きな分散システムを構築する際には溢れてしまう可能性がある．ルーティングテーブルに入りきらない大規模なシステムを構築する際には，ローカルノードプロセッサの主記憶上に完全なルーティングテーブルを用意し，Responsive Link コントローラ上のルーティングテーブルはキャッシュとして用いるようにする．つまり，TLB 付きの MMU とページテーブルを用いたメモリ管理と同様な管理手法を行うようにする．

そのために，ルーティングテーブルにヒットしないエントリがあった際には，ローカルノードのプロセッサに対して割り込みをかけると同時に，該当パケットを一時的に前述の退避用外部記憶に退避する．割り込みをかけられたプロセッサは，主記憶上の完全なルーティングテーブルをソフトウェアで検索し，そのエントリを Responsive Link コントローラ上のルーティングテーブルの適切なエントリとスワップするようにする（多くの場合，最近使用されていないエントリとスワップすると考えられるが，それは RT-OS のポリシ依存である．）Responsive Link コントローラ側は，イベントリンクとデータリンクそれぞれについて，LRU エントリアドレスが分かるように設計し，RT-OS に対してヒントを与えるようにする．その後，退避していたパケットを追い越しバッファに書き戻すことによって継続的にルーティングを実現する．

上記のような機構により，大規模な分散リアルタイムシステムが構築可能である．ただし，コントローラ内のルーティングテーブルに収まる範囲の規模でないと，厳密にハードリアルタイム性を維持するのは困難となる．

また，分散リアルタイムシステムの規模が大きくなればなるほど（つまりルーティングテーブルのサイズが大きくなればなるほど）通信のジッタは大きくなり，リアルタイム性の時間粒度も大きくなるが，近傍で激しく通信している経路をキャッシュに置き，そうでないものは主記憶上のルーティングテーブルに置く等の方法をとることにより，運用が可能であると考えられる．

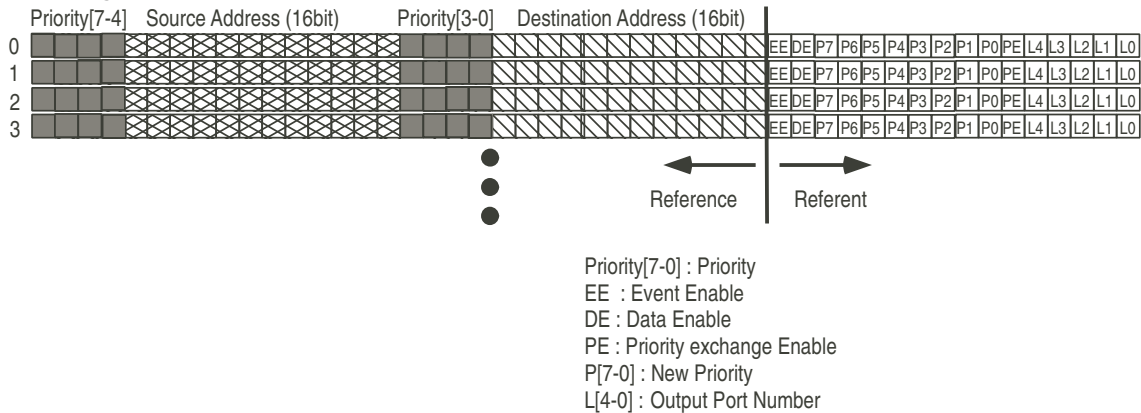


図 23.6: Responsive Link のルーティングテーブル

23.6 パケットの加減速制御

リアルタイム通信パケットの制御を外部から行うことができるようにするために，通信ノード毎にパケットの優先度の付け替えができるようにして，分散管理型でのリアルタイム通信の制御を実現している．

優先度の付け替えは，図 23.6 のルーティングテーブルを用いることによって行なう．図 23.6 において，ネットワークアドレスと優先度を元にルーティングテーブルを参照し出力ポート番号を決定する際に，優先度を付け替えないモード（図 23.6 の優先度付替ビット PE が無効）の場合は優先度はそのままであるが，優先度を付け替えるモード（優先

度付替ビット PE が有効) の場合、出力ポートから出力する際に優先度 (Priority[7-0]) を新優先度 (P7~P0) に置き換える。つまり、現ノードでの通信パケットの優先度は入力パケットのヘッダに付加されている優先度で決定され、その優先度に従って追い抜きやルーティングが決定されるが、次ノード以降での通信パケットの優先度を制御することができる。ルーティングテーブルの設定はソフトウェア (分散リアルタイムオペレーティングシステム等) で行ない、ルーティング (経路制御) 自身はハードウェアで行なうようになっている。

このパケットの加減速制御機構により、例えば、リアルタイム通信の流量やレイテンシを監視するミドルウェアを用いて、リアルタイム通信の制御を可能とする。リアルタイム性の低い通信パケットがバルクの的に流れていて、そのパケットが他のリアルタイム性の高いパケットの通信のリアルタイム性を阻害していたとしたら、通信監視ミドルウェアが当該パケットの優先度を下げることによって、リアルタイム性の制御を行うことができる。あるいは、あるノードでデッドラインミスが発生してしまった場合、その通信パケットの優先度を途中の経路で上げることにより (特にホットスポットで優先度を上げると効果的)、次回からのデッドラインミスを防ぐことが可能となる。

23.7 優先度に従った経路制御

優先度に従って専用回線や迂回路を設けたり、データの流量の制御を行なうことができるように、全く同じネットワークアドレスを持つ通信パケットの経路を優先度によって別の経路に設定することができるようにしている。そのために、基本的にはネットワークアドレスと優先度の組でルーティングテーブルを参照する。

優先度ごとに必ずルーティングテーブルを設定しなければならないと煩雑であるので、デフォルトルートを設定することができる。ネットワークアドレスは同じであるが優先度が一致する組み合わせ (経路) がルーティングテーブル上に無い場合には、最も優先度の低い優先度 0 の経路がデフォルト経路となるようになっている。つまり、

1. ネットワークアドレスと優先度の両方が一致すればその経路が第一優先
2. ネットワークアドレスは一致するが優先度が一致しない場合、優先度 0 の経路

となる。ここで、優先度 0 の経路はデフォルト経路となるので、途中で経路が消滅してしまわないようにルーティングテーブルに必ず登録する必要がある。

図 23.7 は、2 次元格子の交点に通信ノードがあるとし、全く同じ送信元から送信先に対して異なる優先度の通信パケットを同時に通信している状態を示す。例えば、優先度 0 のイベントリンクの経路上は別の通信ノードからの通信パケットも同じ経路を通して送信先に通るように設定しておき、優先度 3 の経路は送信元と送信先の優先度 3 の通信パケットしか通らないように設定しておくことにより、他の通信パケットと衝突が起きない専用回線を実現することができる。*Responsive Link* には優先度による追い越し機構があるが、衝突があると追い越しのために多少のオーバーヘッドが生じてしまうので、このように優先度を用いてパケットの衝突が全くない専用回線を設定することにより、非常にレイテンシ及びジッタが小さいリアルタイム経路の実現を可能とする。また、優先度が異なる経路を複数設定することによってマルチリンクを実現し、バンド幅を広げることも同時に可能とする。

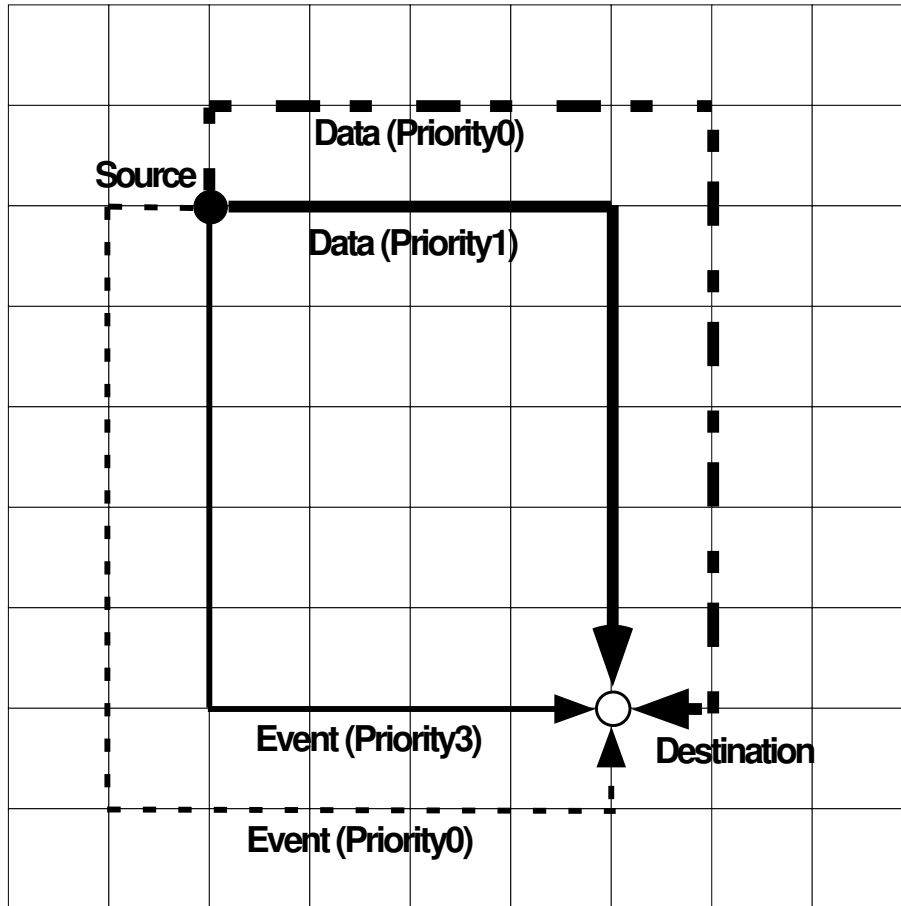
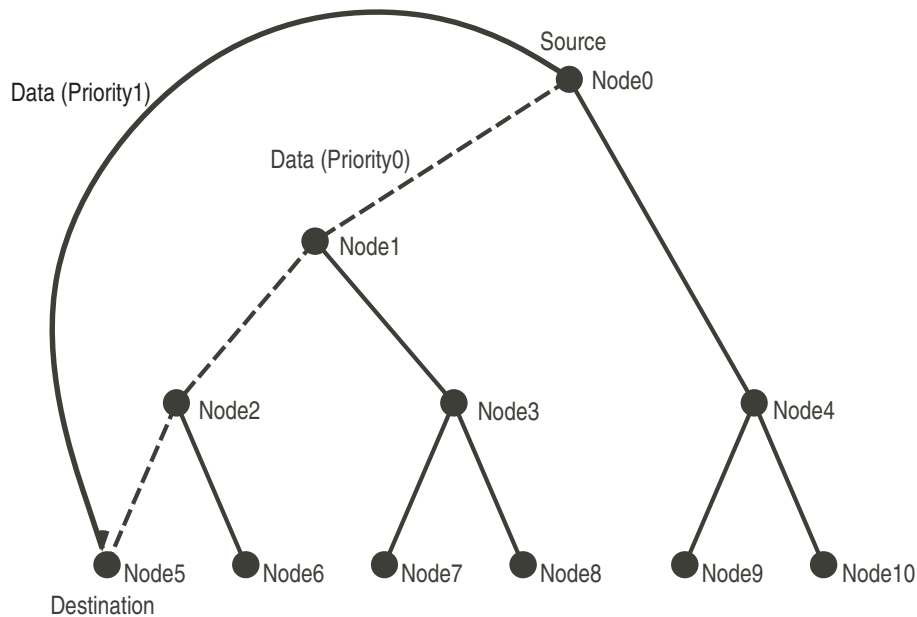


図 23.7: *Responsive Link* の優先度付経路

制御用の分散システムでは図 23.8 のような木構造を採る場合が多い．図 23.8 において通信ノード 0 から通信ノード 5 に通信する場合，優先度 0 の通信パケットは途中に通信ノード 1 と通信ノード 2 という中間ノードを経由して通信を行なうが，優先度 1 の通信パケットは通信ノード 0 から直接通信ノード 5 へ通信を行なうことができる．これは，例えばヒューマノイドロボットを開発した際に，当初は頭モジュール，肩モジュール，肘モジュール，指モジュールと接続しそれらの経路をホップして通信を行っていたが，設計後にどうしても頭モジュールと指モジュール間の通信レイテンシが間に合わないと判明した場合，後付で頭モジュールと指モジュールを直接接続し優先度を変えて通信することにより，容易に通信経路（この場合は専用回線）の増設を可能とする．この機能は，実システムを構築する際に手助けとなる．

図 23.8: *Responsive Link* の優先度付木構造経路

23.8 低レベル通信

Responsive Link は分散制御用途であるので、必ずエラー訂正を行わなければならない。その際、できるだけエラー訂正によってリアルタイム性が損なわれないようにする必要がある。

ここで、パケット単位で CRC を付加しエラー訂正を行う方法では、パケット全体を受信しないとエラー訂正できない。その場合、ホップ毎にレイテンシが積算されていくので、リアルタイム通信用のエラー訂正としては好ましくない。そこで、レスポンスリンクでは 1 ホップごとにフレーム（図 23.2 参照）単位でエラー訂正を行い、1 フレーム（8bit データ+4bit 冗長符号）につき 1bit のエラーであれば、再送することなしにハードウェアで誤り訂正を行うようにする。

23.8.1 CODEC

Responsive Link の CODEC は、8bit の情報ビット列に、誤り訂正用の 4bit の冗長ビット列を加えた 12bit を 1 フレームとして通信を行う。本 CODEC で行われる符合化は、以下のような流れとなる。

1. 巡回組織ハミング符合化（冗長ビット列を加える誤り訂正符合化）
2. Bit Stuffing（連続した 1 の符合に 0 を挿入）
3. NRZI 符合化

以下、各符合化について説明を行う。

23.8.2 巡回組織ハミング符号化

誤り訂正符号として、生成多項式が $x^4 + x + 1$ の巡回組織ハミング符号を採用する。この符号化では、8bit データの下位 (LSB) 側に 4bit の冗長ビット列を付加することで、12bit 中の任意の 1bit の誤りを受信側で訂正することを可能にし、表 23.1 より誤りの位置を特定できる。送信時には、これら 12bit のビット列は、MSB 側から 1bit ずつ送信を行う。

表 23.1: シンドロームとエラーの位置

Syndrome	Error Position (4 redundancy bits)	Meaning
0000	00000000 0000	No error
0001	00000000 0001	Redundancy-bit error
0010	00000000 0010	Redundancy-bit error
0100	00000000 0100	Redundancy-bit error
1000	00000000 1000	Redundancy-bit error
0011	00000001 0000	0bit error
0110	00000010 0000	1bit error
1100	00000100 0000	2bit error
1011	00001000 0000	3bit error
0101	00010000 0000	4bit error
1010	00100000 0000	5bit error
0111	01000000 0000	6bit error
1110	10000000 0000	7bit error

23.8.3 Bit Stuffing

1 が長時間連続することによって引き起こされるリンクへの直流成分が発生や、受信側のビット同期への支障を回避するために、通信データ中に 5 つの連続した 1 が現れた場合には、その後ろに 0 を挿入する。

23.8.4 NRZI 符号化

最終的に送信される際に NRZI(Non Return to Zero Inverted) 符号化を行う。NRZI 符号化は、0 を送る場合にはリンクのデータビットを反転し、1 を送る場合にはデータビットの状態を前のまま保持する。

23.8.5 セットアップパターン

電源投入直後や、予期できないバースト的なリンクエラーなどの後は、送受信インタフェース間でフレーム同期がとれない場合がある。そのような場合、明示的にリンクの初期化を行うようにする。具体的には、以下に示すセットアップパターンを受信側に送信する。

セットアップパターン：000001111110

このパターンは、連続した 1 が 6 個以上は連続しないという bit stuffing の規則に反しているため、いかなる通常のパケットとも区別される。受信側では、このパターンを受信するとその後、最初に認識したフレームを、新しいパケットの第 1 フレームとして解釈する。

表 23.3: 通信速度とケーブル

Speed (Mbaud)	100	200	400
Maximum Length (m)	100	80	60
Recommendable Cable	Cat5e	Cat5e	Cat6

23.8.6 DPLL を用いたビット同期

受信側に DPLL(Digital Phase Lock Loop) 機構を設計し、受信用クロックの立ち上がりエッジに同期して受信信号をサンプリングする。1bit 転送あたりのサンプリング数はソフトウェアの設定によって可変 (4,8,16,32,64,128,256) にする。DPLL では設定された周期ごとに受信用クロックを生成し、受信信号のエッジを検出することにより、信号のエッジ間の中央で受信用クロックが立ち上がるように、受信用クロックの周期を微調整を行う。表 23.2 に DPLL のモードを示す。

モード名	p_mode2	p_mode1	p_mode0	d_clk 周期/1bit 転送
Mode2	1	1	1	2
Mode4	0	0	0	4
Mode8	0	0	1	8
Mode16	0	1	0	16
Mode32	0	1	1	32

表 23.2: DPLL モードの設定

23.8.7 エラーの取扱い

Responsive Link では、誤り訂正符合化によって 1[bit/frame] の誤りまでは自動的にエラー訂正を行うことができる。エラーの箇所を受信側で特定するために、図 23.2 のトレイラ部の Dirty ビットを立てる。具体的には、データリンクの場合ワード (4byte) 単位で、イベントリンクの場合バイト単位で、エラーのあった場所の Dirty ビットを立てる。エラーがハードウェアによって訂正されても、訂正しきれなくても Dirty ビットは立てるようにする。また、そのパケット中に 1 箇所でもエラー訂正が行われハードウェアで訂正しきれた場合、トレイラ部の Correct ビットを立てる。エラー訂正不可能だった場合、Fatal ビットを立てる。受信側のアプリケーションでは、これらを参考にし、例えば、受信データを本当に制御に使用してよいかどうか等を判断することを可能にする。

23.8.8 通信速度

Responsive Link の通信 (変調) 速度は、様々な環境 (コンフィギュレーション, アプリケーション) を想定し、400, 200, 100, 50, 12.5, 6.25 [Mbaud] の範囲で段階的に可変とする。

表 23.3 に、通信速度と最大通信距離、推奨ケーブルの関係を示す。例えば、最大変調速度 400[Mbaud] で通信する場合、ケーブルには Category6 を使用し、最大通信距離は 60[m] 以内である。この場合、DPLL の基準周波数にはデューティ比が 1 対 1 の 800[MHz] のアップダウンエッジを使用し、サンプリング数 4 で DPLL を行うことによって実現する。

レスポンスプロセッサは組み込み用途を想定しているので、消費電力が大きな問題となる。一般に通信速度 (動作周波数) を速くすれば消費電力が大きくなり、遅くすれば小さくなる。通信速度の変更は、受信クロックを変更するのではなく DPLL のサンプリング数を変更することによって行う。従って、通信速度が遅い場合の通信は、通信速度

が遅くなることによる安定性の増加と DPLL のサンプリング数が増加することによる安定性の増加という 2 重の恩恵を受ける。

23.9 メモリマップ

レスポンスブリンク部のアドレスマップは以下の通りである。

デコードアドレス	接続される I/O
0x8000_xxxx	Link SDRAM
0x90xx_xxxx	イベント入力用 DPM (r)
0x94xx_xxxx	イベント出力用 DPM (r/w)
0x98xx_xxxx	データ入力用 DPM (r)
0x9Cxx_xxxx	データ出力用 DPM (r/w)
0xA000_0xxx	レスポンスブリンク内部レジスタ
0xA000_1xxx	レスポンスブリンク用 IRC (r/w)
0xA000_2xxx	ルーティングテーブルアドレス部 (r/w)
0xA000_3xxx	ルーティングテーブルリンク部 (r/w)
0xA800_xxxx	Link SDRAM モードレジスタ

初期アドレス: 0xA000_0000

23.10 レジスタマップ

23.10.1 SDRAM モードレジスタ

オフセット: 0x0000

31		2	1	0
	30'h0			SDMODE

Responsive Link は、パケット追い越し用に外付けの SDRAM を付けることができる。SDMODE(SDram MODE) レジスタは、パケット追い越し用外付け DDR SDRAM の有無と大きさを示す。外付け SDRAM を搭載しない場合は、内蔵の追い越しバッファ（各リンク 8 パケット分）のみで優先度付きパケットの追越を行う。

bit 名	機能
29'h0	0
SDMODE	Default 000 000 : 外付け SDRAM なし 001 : 外付け SDRAM あり, 容量 : 8MB 010 : 外付け SDRAM あり, 容量 : 16MB 011 : 外付け SDRAM あり, 容量 : 32MB 100 : 外付け SDRAM あり, 容量 : 64MB 101 : 外付け SDRAM あり, 容量 : 128MB 110 : 外付け SDRAM あり, 容量 : 256MB 111 : 外付け SDRAM あり, 容量 : 512MB

23.10.2 レスポンシブリンク速度設定レジスタ

オフセット: 0xA000.0004

属性 リード/ライト

31	28	27	25	24	22	21	19	18	16	15	12	11	9	8	6	5	3	2	0
-	Data4	Data3	Data2	Data1	-	Event4	Event3	Event2	Event1										

RSL(*Responsive Link Speed*): Default 000

本レジスタはレスポンシブリンクの変調速度を示す。

- 111 : 800 Mbaud
- 000 : 400 Mbaud
- 001 : 200 Mbaud
- 010 : 100 Mbaud
- 011 : 50 Mbaud

bit 名	機能
Data4	Data Link 4 用 RSL
Data3	Data Link 3 用 RSL
Data2	Data Link 2 用 RSL
Data1	Data Link 1 用 RSL
Event4	Event Link 4 用 RSL
Event3	Event Link 3 用 RSL
Event2	Event Link 2 用 RSL
Event1	Event Link 1 用 RSL

23.10.3 レスポンシブリンク初期化レジスタ

オフセット: 0xA000.0008

属性 リード/ライト

31	29	28	25	24	23	21	20	17	16	15	13	12	9	8	7	5	4	1	0
-	EDINIT	ED	EM	-	EEINIT	EE	ES	-	DDINIT	DD	DM	-	DEINIT	DE	DS				

RLINIT(*Responsive Link INITialization*) レジスタはレスポンシブリンクのスイッチの初期化およびエンコーダ/デコーダ部分、デュアルポートメモリの初期化を行なう。

- 0: 通常動作
- 1: 初期化

bit 名	機能
EDINIT	Event Link のデコーダの初期化 EDINIT[4]: RLINIT[28]: Event link4 の初期化 EDINIT[3]: RLINIT[27]: Event link3 の初期化 EDINIT[2]: RLINIT[26]: Event link2 の初期化 EDINIT[1]: RLINIT[25]: Event link1 の初期化
EMI	Event Link 用のデュアルポートメモリコントローラの初期化（メモリの内容は保持される）
ELINIT	Event link の各エンコードの初期化 EEINIT[4]: RLINIT[20]: Event link4 の初期化 EEINIT[3]: RLINIT[19]: Event link3 の初期化 EEINIT[2]: RLINIT[18]: Event link2 の初期化 EEINIT[1]: RLINIT[17]: Event link1 の初期化
E_s	Event link switch の初期化
DDINIT	Data link の各デコーダの初期化 DDINIT[4]: RLINIT[12]: Data link4 の初期化 DDINIT[3]: RLINIT[11]: Data link3 の初期化 DDINIT[2]: RLINIT[10]: Data link2 の初期化 DDINIT[1]: RLINIT[9]: Data link1 の初期化
DMI	Data Link 用のデュアルポートメモリコントローラの初期化（メモリの内容は保持される）
DEINIT	Data link の各エンコードの初期化 DEINIT[4]: RLINIT[4]: Data link4 の初期化 DEINIT[3]: RLINIT[3]: Data link3 の初期化 DEINIT[2]: RLINIT[2]: Data link2 の初期化 DEINIT[1]: RLINIT[1]: Data link1 の初期化
D_s	Data link switch の初期化

23.10.4 レスポンシブリンク割り込みクリアレジスタ

オフセット: 0xA000.000C 属性 ライト

31	7	6	1	0
-			RLIC	-

本レジスタのオフセットはデコーダリセット割り込みクリアレジスタのオフセットと同じである．*Responsive Link* の IRQ1-6 のいずれかがイネーブルになったときのみ，本レジスタに書き込み可能となる．RLIC(*Responsive Link* Irq Clear) レジスタはイベントリンクの割り込み要求のクリアを行なう．

Default 0

0: 通常動作

1: クリア

bit 名	機能
RLIC[1]	Data-Out EOP(End Of Packet) IRQ Clear: データパケットが DPM の設定した範囲から送信された場合に生じる割り込みのクリア
RLIC[2]	Event-Out EOP IRQ Clear: イベントパケットが DPM の設定した範囲から送信された場合に生じる割り込みのクリア
RLIC[3]	Data-In EOP IRQ Clear: データパケットが DPM の設定した範囲に受信された場合に生じる割り込みのクリア
RLIC[4]	Event-In EOP IRQ Clear: イベントパケットが DPM の設定した範囲に受信された場合に生じる割り込みのクリア
RLIC[5]	Data Packet-In IRQ Clear: 割り込みビットの設定されたデータパケットが到着した場合に生じる割り込みのクリア
RLIC[6]	Event Packet-In IRQ Clear: 割り込みビットの設定されたイベントパケットが到着した場合に生じる割り込みのクリア

23.10.5 デコーダリセット割り込みクリアレジスタ

オフセット: 0xA000_000C 属性 リード/ライト

31	21 20	16 15	5 4	0
-	Event	-	Data	

本レジスタのオフセットはレスポンスリンク割り込みクリアレジスタのオフセットと同じである。*Responsive Link* の IRQ1-6 のすべてがディスエーブルになったときのみ、本レジスタに読み書き可能となる。デコーダリセット割り込み要求のクリアを行なう。

Default 0

0: クリア

1: 割り込み発生（デバッグ用）

bit 名	機能
Event	Event Link のデコーダリセット割り込み Event[4]: Event link4 のデコーダリセット割り込み Event[3]: Event link3 のデコーダリセット割り込み Event[2]: Event link2 のデコーダリセット割り込み Event[1]: Event link1 のデコーダリセット割り込み Event[0]: Event link0 のデコーダリセット割り込み
Data	Data Link のデコーダリセット割り込み Data[4]: Data link4 のデコーダリセット割り込み Data[3]: Data link3 のデコーダリセット割り込み Data[2]: Data link2 のデコーダリセット割り込み Data[1]: Data link1 のデコーダリセット割り込み Data[0]: Data link0 のデコーダリセット割り込み

23.10.6 レスポンシブリンク送信停止割り込みクリアレジスタ

オフセット: 0xA000.0010 属性 リード/ライト

31	21	20	16	15	5	4	0
-		DWIRQC		-		EWIRQC	

Responsive Link はパケット追い越し用 SDRAM を使用している際には追い越し用 SDRAM が溢れそうになると送信停止割り込みを自動生成する。同様に、追い越し用 SDRAM を使用していない際には、追い越し用バッファが溢れそうになると送信停止割り込みを自動生成する。本 WIRQC(Wait IRQ Clear) レジスタはレスポンシブリンク送信停止割り込み要求のクリアを行なう。

Default 0

0: 通常動作

1: クリア

bit 名	機能
DWIRQC	Data link WIRQC DWIRQC[4]: WIRQC[20]: Data link4 DWIRQC[3]: WIRQC[19]: Data link3 DWIRQC[2]: WIRQC[18]: Data link2 DWIRQC[1]: WIRQC[17]: Data link1 DWIRQC[0]: WIRQC[16]: Data link0(CPU)
EWIRQC	Event link WIRQC EWIRQC[4]: WIRQC[4]: Event link4 EWIRQC[3]: WIRQC[3]: Event link3 EWIRQC[2]: WIRQC[2]: Event link2 EWIRQC[1]: WIRQC[1]: Event link1 EWIRQC[0]: WIRQC[0]: Event link0(CPU)

23.10.7 レスポンシブリンク継続割り込みクリアレジスタ

オフセット: 0xA000.0014 属性 リード/ライト

31	21	20	16	15	5	4	0
-		DCIC		-		ECIC	

Responsive Link は、SDRAM に退避されたパケットがスイッチに書き戻された（再度送信された）際にレスポンシブリンク継続割り込み CI(Continuous Irq) を発生する。CIC(Continuous Irq Clear) レジスタはその割り込み要求 CI のクリアを行なう。

Default 0

0: 通常動作

1: クリア

bit 名	機能
DCIC	Data CIC DCIC[4]: CIC[20]: Data link4 DCIC[3]: CIC[19]: Data link3 DCIC[2]: CIC[18]: Data link2 DCIC[1]: CIC[17]: Data link1 DCIC[0]: CIC[16]: Data link0(CPU)
ECIC	Event CIC ECIC[4]: CIC[4]: Event link4 ECIC[3]: CIC[3]: Event link3 ECIC[2]: CIC[2]: Event link2 ECIC[1]: CIC[1]: Event link1 ECIC[0]: CIC[0]: Event link0(CPU)

23.10.8 レスポンシブリンク致命的エラー割り込みクリアレジスタ

オフセット: 0xA000_0018 属性 リード/ライト

31	21 20	16 15	5 4	0
-	DFIC	-	EFIC	

Responsive Link は、各リンクの受信パケットにハードウェアで回復不可能なエラーがあった場合にレスポンシブリンク致命的エラー割り込み FI(Fatal Irq) を発生する。FIC(Fatal Irq Clear) レジスタは、その割り込み要求 FI のクリアを行なう。

Default 0

0: 通常動作

1: クリア

bit 名	機能
DFIC	Data FIC DFIC[4]: FIC[20]: Data link4 DFIC[3]: FIC[19]: Data link3 DFIC[2]: FIC[18]: Data link2 DFIC[1]: FIC[17]: Data link1 DFIC[0]: FIC[16]: Data link0(CPU)
EFIC	Event FIC EFIC[4]: FIC[4]: Event link4 EFIC[3]: FIC[3]: Event link3 EFIC[2]: FIC[2]: Event link2 EFIC[1]: FIC[1]: Event link1 EFIC[0]: FIC[0]: Event link0(CPU)

23.10.9 レスポンシブリンクルーティングテーブル割り込みクリアレジスタ

オフセット: 0xA000.001C 属性 リード/ライト



Responsive Link は、ルーティングテーブルにマッチするエントリが無かった場合にレスポンシブリンクルーティングテーブル割り込み (RTIRQ) を発生する。RTIRQC(Routing Table IRQ Clear) レジスタは、その割り込み要求 RTIRQ のクリアを行なう。

Default 0

- 0: 通常動作 (r) / 割り込みクリア (w)
- 1: 割り込み状態 (r) / 割り込み発生 (デバッグ用) (w)

bit 名	機能
RTIC[0]	Event Routing Table IRQ Clear
RTIC[1]	Data Routing Table IRQ Clear

23.10.10 レスポンシブリンク SDRAM バスリクエストレジスタ

オフセット: 0xA000.0020 属性 リード/ライト

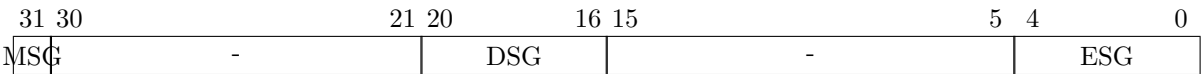


Responsive Link の追い越し用 SDRAM のバスには、Responsive Link とプロセッサバスの 2 つのバスマスタが接続されている。通常、プロセッサ側から追い越し用 SDRAM にアクセスする際には、データのトランザクション毎に、バス権の調停が行われている。プロセッサ側からバースト的に追い越し用 SDRAM をアクセスしたい場合には、本ビットを有効にすることで、追い越し用 SDRAM バスのバス権をプロセッサ側 (プロセッサや DMAC 等) が常に得ることができる (本ビットを設定しなくてもアクセス可能である。) Responsive Link 側が追い越し用 SDRAM バスを参照できなくなる (パケットの退避・復帰ができなくなる) という副作用がある。

bit 名	機能
RLSDBREQ	RLSDBREQ (Responsive Link SDram-Bus REQuest) : Default 1 本ビットはレスポンシブリンクの SDRAM バスへの明示的なバスリクエストを行なう。 0: バスリクエストイネーブル 1: バスリクエストディスエーブル

23.10.11 レスポンシブリンク SDRAM バスグラントレジスタ

オフセット: 0xA000.0024 属性 リード/ライト



RLSDBGRNT(Responsive Link SDram Bus GRaNT) レジスタは、追い越し用 SDRAM バスのバスグラント (どのバスマスタがバス権を有しているか) を示す。

0: バス権獲得

1: バス権開放

bit 名	機能
MSG	Mpu Sdram bus Grant: MPU がバス権を得ている
DSG	Data link Sdram bus Grant: Data Link がバス権を得ている DSG[4]: RLSDBGRNT[20]: Data link4 DSG[3]: RLSDBGRNT[19]: Data link3 DSG[2]: RLSDBGRNT[18]: Data link2 DSG[1]: RLSDBGRNT[17]: Data link1 DSG[0]: RLSDBGRNT[16]: Data link0(CPU)
ES	Event link Sdram bus Grant: Event Link がバス権を得ている ESG[4]: RLSDBGRNT[4]: Event link4 ESG[3]: RLSDBGRNT[3]: Event link3 ESG[2]: RLSDBGRNT[2]: Event link2 ESG[1]: RLSDBGRNT[1]: Event link1 ESG[0]: RLSDBGRNT[0]: Event link0(CPU)

23.10.12 レスポンシブリンクルーティングテーブルバスリクエストレジスタ

オフセット: 0xA000_0028 属性 ライト

31	1	0
		BRQ

Responsive Link のルーティングテーブルのバスには、*Responsive Link* とプロセッサバスの 2 つのバスマスタが接続されているが、デフォルトのバスマスタは *Responsive Link* である。プロセッサ側からルーティングテーブルをアクセスしたい場合には、本ビットを有効にすることで、ルーティングテーブルバスのバス権をプロセッサ側（プロセッサや DMAC 等）が得ることができる。*Responsive Link* 側がルーティングテーブルを参照できなくなる（パケットのルーティングができなくなる）という副作用がある。

bit 名	機能
BRQ	RLTLBREQ (<i>Responsive Link</i> routing TaBLe Bus REQuest): Default 1 本ビットはレスポンシブリンクのルーティングテーブルバスへのバスリクエストを行なう。 0: バスリクエストイネーブル 1: バスリクエストディスエーブル

オフセット: 0xA000_0028 属性 リード

31	30	21	20	16	15	5	4	0
MRR	-	DRR	-	ERR				

本ビットはプロセッサバス側からレスポンシブリンクのルーティングテーブルバスへのバスリクエストを示す。

0: バスリクエスト有

1: バスリクエスト無

bit 名	機能
MRR	Mpu Routing table bus Request
DRR	Data link Routing table bus Request DRR[4]: RLTBLBREQ[20]: Data link4 DRR[3]: RLTBLBREQ[19]: Data link3 DRR[2]: RLTBLBREQ[18]: Data link2 DRR[1]: RLTBLBREQ[17]: Data link1 DRR[0]: RLTBLBREQ[16]: Data link0(CPU)
ER	Event link Routing table bus Request ERR[4]: RLTBLBREQ[4]: Event link4 ERR[3]: RLTBLBREQ[3]: Event link3 ERR[2]: RLTBLBREQ[2]: Event link2 ERR[1]: RLTBLBREQ[1]: Event link1 ERR[0]: RLTBLBREQ[0]: Event link0(CPU)

23.10.13 レスポンシブリンクルーティングテーブルバスグラントレジスタ

オフセット: 0xA000.002C 属性 リード

31 30	21 20	16 15	5 4	0
MRG	-	DRG	-	ERG

RLTBLBGRNT (*Responsive Link* routing TaBLe Bus GRaNT) レジスタは、レスポンシブリンクのルーティングテーブルバスのバスグラント (どのバスマスタがバス権を有しているか) を示す。

0: バス権獲得

1: バス権開放

bit 名	機能
MRG	Mpu Routing table bus Grant: MPU がバス権を得ている
DRG	Data link Routing table bus Grant: Data Link がバス権を得ている DRG[4]: RLTBLBGRNT[20]: Data link4 DRG[3]: RLTBLBGRNT[19]: Data link3 DRG[2]: RLTBLBGRNT[18]: Data link2 DRG[1]: RLTBLBGRNT[17]: Data link1 DRG[0]: RLTBLBGRNT[16]: Data link0(CPU)
ERG	Event link Routing table bus Grant: Event Link がバス権を得ている ERG[4]: RLTBLBGRNT[4]: Event link4 ERG[3]: RLTBLBGRNT[3]: Event link3 ERG[2]: RLTBLBGRNT[2]: Event link2 ERG[1]: RLTBLBGRNT[1]: Event link1 ERG[0]: RLTBLBGRNT[0]: Event link0(CPU)

23.10.14 イベントリンク LRU アドレスレジスタ

オフセット: 0xA000_0030 属性 リード

31	10	9	0
-			ELLRUA

bit 名	機能
ELLRUA	ELLRUA (Event Link LRU Address) レジスタはイベントリンクのルーティングテーブル中で、最も近くに使用されたテーブルの格納されているアドレスを示す。

23.10.15 データリンク LRU アドレスレジスタ

オフセット: 0xA000_0034 属性 リード

31	10	9	0
-			DLLRUA

bit 名	機能
DLLRUA	DLLRUA (Data Link LRU Address) レジスタはデータリンクのルーティングテーブル中で、最も近くに使用されたテーブルの格納されているアドレスを示す。

23.10.16 レスポンシブリンク用割り込みコントローライネーブルレジスタ

オフセット: 0xA000_0038 属性 リード

31	1	0
-		RLICE

bit 名	機能
RLICE	RLICE (<i>Responsive Link</i> Interrupt Controller Enable) レジスタはレスポンシブリンク用割り込みコントローラ RLIRC のイネーブルビットを示す。1 のとき、RLIRC は出力を行っている。

23.10.17 イベントリンク用 SDRAM ループカウントレジスタ

オフセット: 0xA000_0040 属性 リード/ライト

31	8	7	0
-			ELSDCNT

追い越し用 SDRAM に退避されたイベント packets を *Responsive Link* イベントスイッチに再度送信してよいかどうかを調べる間隔を指定する。短すぎると消費電力が大きくなり、長すぎるとリアルタイム性が損なわれる。

bit 名	機能
ELSDCNT	ELSDCNT (Event Link SDram loop CouNTer) レジスタの設定により、追い越し用 SDRAM に退避されているイベントパケットをイベントスイッチに再送しようとするリトライの間隔を 1 パケット分の送信時間を単位として指定する。(1 - 40) Default: 32

23.10.18 データリンク用 SDRAM ループカウントレジスタ

オフセット: 0xA000.0044 属性 リード/ライト

31		4	3	0
-				DLSDCNT

追い越し用 SDRAM に退避されたデータパケットを *Responsive Link* データスイッチに再度送信してよいかどうかを調べる間隔を指定する。短すぎると消費電力が大きくなり、長すぎるとリアルタイム性が損なわれる。

bit 名	機能
DLSDCNT	DLSDCNT (Data Link SDram loop CouNTer) レジスタの設定により、追い越し用 SDRAM に退避されているデータパケットをデータスイッチに再送しようとするリトライの間隔を 1 パケット分の送信時間を単位として指定する。(1 - 95) Default: 4

23.10.19 レスポンシブリンクスイッチモードレジスタ

オフセット: 0xA000.0048 属性 リード/ライト

31		2	1	0
-				RLSM

RLSM(*Responsive Link* Switch Mode) レジスタの設定により、レスポンシブリンクのスイッチの動作を変更する。

- 0: Cut Through Mode レイテンシ的に有利であるがパケットの追い越しをしにくい
- 1: Store and Forward Mode レイテンシ的に不利であるがパケットの追越ししやすい

Default: 0

bit 名	機能
RLSM[0]	Event Link Switch の設定
RLSM[1]	Data Link Switch の設定

23.10.20 レスポンシブリンク用オフラインレジスタ

オフセット: 0xA000.004c 属性 リード

31	21	20	16	15	5	4	0
-			DRLOL	-		ERLOL	

Responsive Link は Plug&Play をサポートするために、リンクアップしていたリンクがリンクダウンするとオフライン割り込みを発生し、リンクダウンしていたリンクがリンクアップするとオンライン割り込みを発生する。

RLOL(*Responsive Link* OffLine) レジスタをリードすることにより、どのリンクがオフライン / オンラインかを調べることができる。

1: Offline

0: Online

bit 名	機能
DRLOL	Data link の RLOL レジスタ DRLOL[4]: RLOL[20]: Data link4 DRLOL[3]: RLOL[19]: Data link3 DRLOL[2]: RLOL[18]: Data link2 DRLOL[1]: RLOL[17]: Data link1 DRLOL[0]: RLOL[16]: Data link0(CPU)
ERLOL	Event link の RLOL レジスタ ERLOL[4]: RLOL[4]: Event link4 ERLOL[3]: RLOL[3]: Event link3 ERLOL[2]: RLOL[2]: Event link2 ERLOL[1]: RLOL[1]: Event link1 ERLOL[0]: RLOL[0]: Event link0(CPU)

オフセット: 0xA000.004c 属性 ライト

31		2	1	0
	-			RLOL

本ビットの設定により、レスポンスリンクのオフライン割り込み及びオンライン割り込みをクリアできる。

1: 割り込みクリアを行わない

0: 割り込みクリア

bit 名	機能
RLOL[0]	<i>Responsive Link</i> Down IRQ Clear: オフライン割り込みのクリア
RLOL[1]	<i>Responsive Link</i> Wakeup IRQ Clear: オンライン割り込みのクリア

23.10.21 パラレルモードレジスタ

オフセット: 0xA000.0050 属性 リード / ライト

31		5	4		1	0
	-			Port		-

Responsive Link の各ポートがパラレル / シリアルモードのどちらで動作するかを設定する。

1: パラレル

0: シリアル

bit 名	機能
Port	<i>Responsive Link</i> の各ポート Port[3]: ポート 4 Port[2]: ポート 3 Port[1]: ポート 2 Port[0]: ポート 1

23.10.22 エラーパケットヘッダレジスタ

オフセット: Event Link : 0xA000_0054 属性 リード

オフセット: Data Link : 0xA000_0058 属性 リード

31	0
Header_Value	

Routing Table に該当しないパケットのヘッダの値が入る .

23.10.23 エラーヘッダポインタレジスタ

オフセット: Event Link : 0xA000_005C 属性 リード

31	3	2	0
-			Err_Header_Ptr

エラーパケットの数を格納する . ただし , 直前のエラーパケット同一のヘッダである場合はインクリメントされない (デバッグ用) .

Default 0

23.10.24 エラーパケットモードレジスタ

オフセット: Event Link : 0xA000_0060 属性 リード / ライト

31	8	7	0
-			Err_Header_Mode_E

オフセット: Data Link : 0xA000_0064 属性 リード / ライト

31	8	7	0
-			Err_Header_Mode_D

本レジスタはエラーヘッダポインタレジスタのために存在する (デバッグ用) .

23.10.25 SDRAM 回復イネーブルレジスタ

オフセット: Event Link : 0xA000_0068 属性 リード / ライト

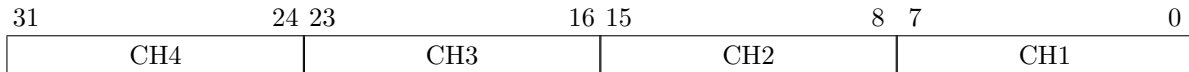


(デバッグ用)

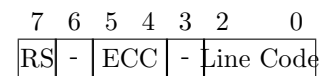
23.10.26 通信コーデック設定レジスタ

オフセット: Event Link : 0xA000_006c 属性 リード/ライト

オフセット: Data Link : 0xA000_0070 属性 リード/ライト



Responsive Link 各チャネルのコーデックを設定する．チャネル毎の設定項目のビットマップを以下に示す．



bit 名	機能
CH*[7]	本ビットをセットすると Reed Solomon 符号によるエラー訂正が有効になる．
CH*[5:4]	バイト毎の ECC を設定する． 00: ECC なし 01: Hamming 符号 10: BCH 符号
CH*[2:0]	伝送路符号を設定する． 001: NRZI+BitStuffing 010: 8B10B 100: 4B10B

23.11 DPM (Dual Port Memory)

Responsive Link とプロセッサは基本的に DPM を介してデータの送受信を行う．DPM はその名の通り 2port を有しており、片方はプロセッサバスに接続され、もう片方は *Responsive Link* の Link0 に接続されている．

Data in/out control register および、Event in/out control register を設定することでパケットの送信/受信方法を決定することができる．イベントパケットの送信および、受信には Event packet in/out 専用の DPM を用い、データパケットの送信および、受信には Data packet in/out 専用の DPM を使用する．

以下に Event in/out , Data in/out それぞれの DPM について説明する．

23.11.1 Event Output

図 23.9 にイベントリンク出力用 DPM の構成を示す．Event out control register (図 23.10 参照) に対して、開始アドレス From_Addr (byte address ではなく word address) と終了アドレス To_Addr (word address) を設定することにより、複数パケットを一度に送信できる．From_Addr と To_Addr は人間に分かりやすいようにこのような名前を付けられているが、実際には全く同じ機能のレジスタが二つ用意されている．From_Addr, To_Addr 共に、設定された

word address - 1 のアドレスに DPM のプロセッサバス側からデータが書かれた瞬間に、DPM から Link0 に対して出力を開始する。

例えば、Mode0 を使用し、From_Addr を 0x00 に設定し To_Addr を 0x07 (byte address: 0x1c) に設定したとする。プロセッサバス側から DMAC もしくはプロセッサによって Payload0, Payload1 の順に DPM にデータが書かれたとすると、DPM のプロセッサ側から 0x06 番地にデータが書かれた瞬間に DPM から *Responsive Link* の Link0 に出力を開始する（この場合、実際には From_Addr には意味がない。）

あるいは、Mode0 を使用し、From_Addr を 0x1f (byte address 0x3c) に設定し To_Addr を 0x2f (byte address: 0x7c) に設定し、さらに DMAC を continuous mode で使用すると、Payload0 ~ 3 の領域と Payload4 ~ 7 の領域を使用して、主記憶等に用意した DPM よりも大きな連続データをハードウェアのみで自動送信することができる（DPM のアドレスデコードの範囲内では、シャドウアドレスでも CS が生成され DPM にアクセスできるように設計しているため。）

offset address
0xC400_00XX

DPM for Event Output

Mode0

0x00	Source Addr.	Destination Addr.
	Payload 0	
	Control & Status	
0x10	Source Addr.	Destination Addr.
	Payload 1	
	Control & Status	
0x20	Source Addr.	Destination Addr.
	Payload 2	
	Control & Status	
0x30	Source Addr.	Destination Addr.
	Payload 3	
	Control & Status	
0x40	Source Addr.	Destination Addr.
	Payload 4	
	Control & Status	
0x50	Source Addr.	Destination Addr.
	Payload 5	
	Control & Status	
0x60	Source Addr.	Destination Addr.
	Payload 6	
	Control & Status	
0x70	Source Addr.	Destination Addr.
	Payload 7	
	Control & Status	

Mode1

0x00	Payload 0	
0x08	Payload 1	
0x10	Payload 2	
0x18	Payload 3	
0x20	Payload 4	
0x28	Payload 5	
0x30	Payload 6	
0x38	Payload 7	
0x40	Payload 8	
0x48	Payload 9	
0x50	Payload 10	
0x58	Payload 11	
0x60	Payload 12	
0x68	Payload 13	
0x70	Payload 14	
0x78	Source Addr.	Destination Addr.
	Control & Status	

図 23.9: DPM for Event Output

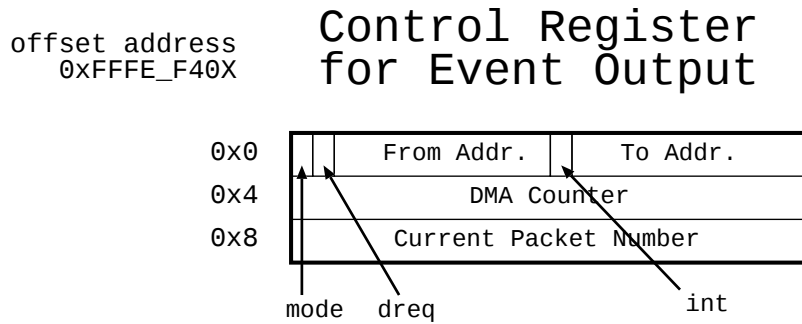


図 23.10: Event Out Control Register

DPM 制御レジスタ

DPM の制御レジスタ (図 23.10 参照) に以下を設定することで、送信の制御を行う。

制御レジスタ (r/w)

- Mode0: mode bit に 0 を設定、すべてのパケットに headr と trailer を付加する。
- Mode1: mode bit に 1 を設定、最後に共通の header と trailer を付加する (すべてのパケットの宛先が同一となる)。
- Int: 本ビットを 1 に設定すると、終了時に EOP(End Of Packet) 割り込みを生成する。
- Dreq: 本ビットを 1 に設定すると、DMA Counter に設定した回数分だけ DMA を行う。
- From_Addr: 設定された word address - 1 のアドレスに DPM のプロセッサバス側からデータが書かれた瞬間に DPM から Link0 に対して出力を開始する。
- To_Addr: 設定された word address - 1 のアドレスに DPM のプロセッサバス側からデータが書かれた瞬間に DPM から Link0 に対して出力を開始する。

DMA Counter (r/w) DMA の回数を指定する

Current Packet Number (r) 現在送信されているパケット番号 (図 23.9 の payload 番号に相当) を示す

23.11.2 Event Input

図 23.11 にイベントリンク入力用 DPM の構成を示す。Event in control register (図 23.12 参照) に対して、開始アドレス From_Addr (byte address ではなく word address) と終了アドレス To_Addr (word address) を設定することにより、複数パケットを一度に受信できる。From_Addr と To_Addr は人間に分かりやすいようにこのような名前を付けられているが、実際には全く同じ機能のレジスタが二つ用意されている。From_Addr, To_Addr 共に、設定された word address - 1 のアドレスに DPM の *Responsive Link* 側からデータが書かれた瞬間に、DPM からプロセッサバス側に対して出力 (DMA 転送) を開始する (dreq bit が設定されている場合)。int bit の割り込みを利用して、ソフトウェアで受信することもできる。

例えば、Mode0 を使用し、From_Addr を 0x00 に設定し To_Addr を 0x07 (byte address: 0x1c) に設定したとする。*Responsive Link* 側から Payload0, Payload1 の順に DPM に受信データが書かれていく。*Responsive Link* 側から DPM

の 0x06 番地にデータが書かれた瞬間に DPM からプロセッサバス側へ出力 (DMA 転送) を開始する (この場合、実際には From_Addr には意味がない。)

あるいは、Mode0 を使用し、From_Addr を 0x1f(byte address 0x3c) に設定し To_Addr を 0x2f(byte address: 0x7c) に設定し、さらに DMAC を continuous mode で使用すると、Payload0~3 の領域と Payload4~7 の領域を使用して、主記憶等に用意した DPM よりも大きなメモリ領域 (サイクリックバッファ等) に対して、受信データをハードウェアのみで連続的に自動受信することができる (DPM のアドレスデコードの範囲内では、シャドウアドレスでも CS が生成され DPM にアクセスできるように設計しているため。)

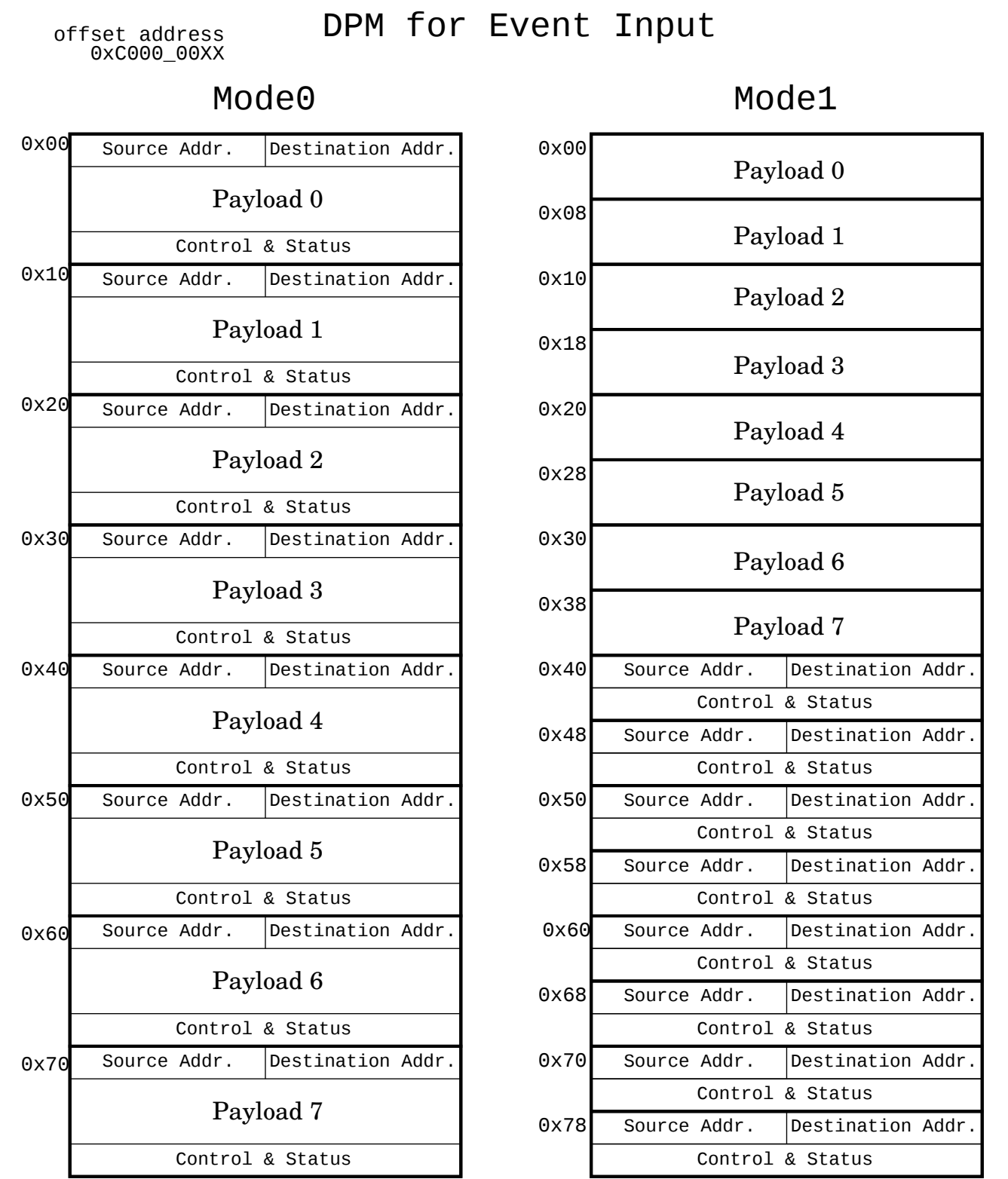


图 23.11: DPM for Event Input

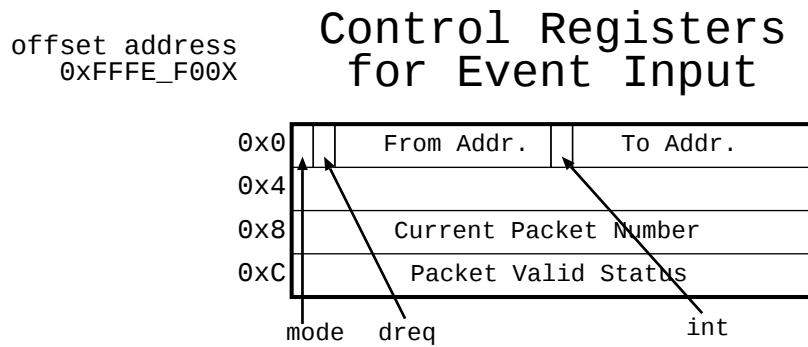


図 23.12: Event in control register

DPM 制御レジスタ

DPM の制御レジスタ（図 23.12 参照）に以下を設定することで、受信の制御を行う。

制御レジスタ (r/w)

- Mode0: mode bit に 0 を設定。すべてのパケットそれぞれに header と trailer が付加された状態で DPM に受信される。
- Mode1: mode bit に 1 を設定。ヘッダとペイロードを図 23.11 のように分離して受信。
- Int: 本ビットを 1 に設定すると、受信終了時にプロセッサに受信完了割り込みを発生する。
- Dreq: 本ビットを 1 に設定すると、From_Addr か To_Addr に設定した word address - 1 にパケットを受信した際に、DMA に対して DREQ を発生する。
- From_Addr: 設定された word address - 1 のアドレスに DPM の *Responsive Link* 側からデータが書かれた瞬間に DPM からプロセッサバス側に対して出力を開始する。
- To_Addr: 設定された word address - 1 のアドレスに DPM の *Responsive Link* 側からデータが書かれた瞬間に DPM からプロセッサバス側に対して出力を開始する。

Current Packet Number (r) 現在送信されているパケット番号（図 23.11 の payload 番号に相当）を示す

Packet Valid Status ハードウェアデバッグ用レジスタ

23.11.3 Data Output

図 23.13 にデータリンク出力用 DPM の構成を示す。Data out control register（図 23.14 参照）に対して、開始アドレス From_Addr (byte address ではなく word address) と終了アドレス To_Addr (word address) を設定することにより、複数パケットを一度に送信できる。From_Addr と To_Addr は人間に分かりやすいようにこのような名前を付けられているが、実際には全く同じ機能のレジスタが二つ用意されている。From_Addr, To_Addr 共に、設定された word address - 1 のアドレスに DPM のプロセッサバス側からデータが書かれた瞬間に、DPM から Link0 に対して出力を開始する。

例えば、Mode0 を使用し、From_Addr を 0x000 に設定し To_Addr を 0x01f (byte address: 0x07c) に設定したとする。プロセッサバス側から DMAC もしくはプロセッサによって Payload0, Payload1 の順に DPM にデータが書かれた

とすると，DPM のプロセッサ側から word address 0x01e 番地にデータが書かれた瞬間に DPM から *Responsive Link* の Link0 に出力を開始する（この場合，実際には From_Addr には意味がない．）

あるいは，Mode0 を使用し，From_Addr を 0x0ff(byte address 0x3fc) に設定し To_Addr を 0x1ff(byte address: 0x7fc) に設定し，さらに DMAC を continuous mode で使用すると，Payload0～15 の領域と Payload16～31 の領域を使用して，主記憶等に用意した DPM よりも大きな連続データをハードウェアのみで自動送信することができる．(DPM のアドレスデコードの範囲内では，シャドウアドレスでも CS が生成され DPM にアクセスできるように設計しているため．)

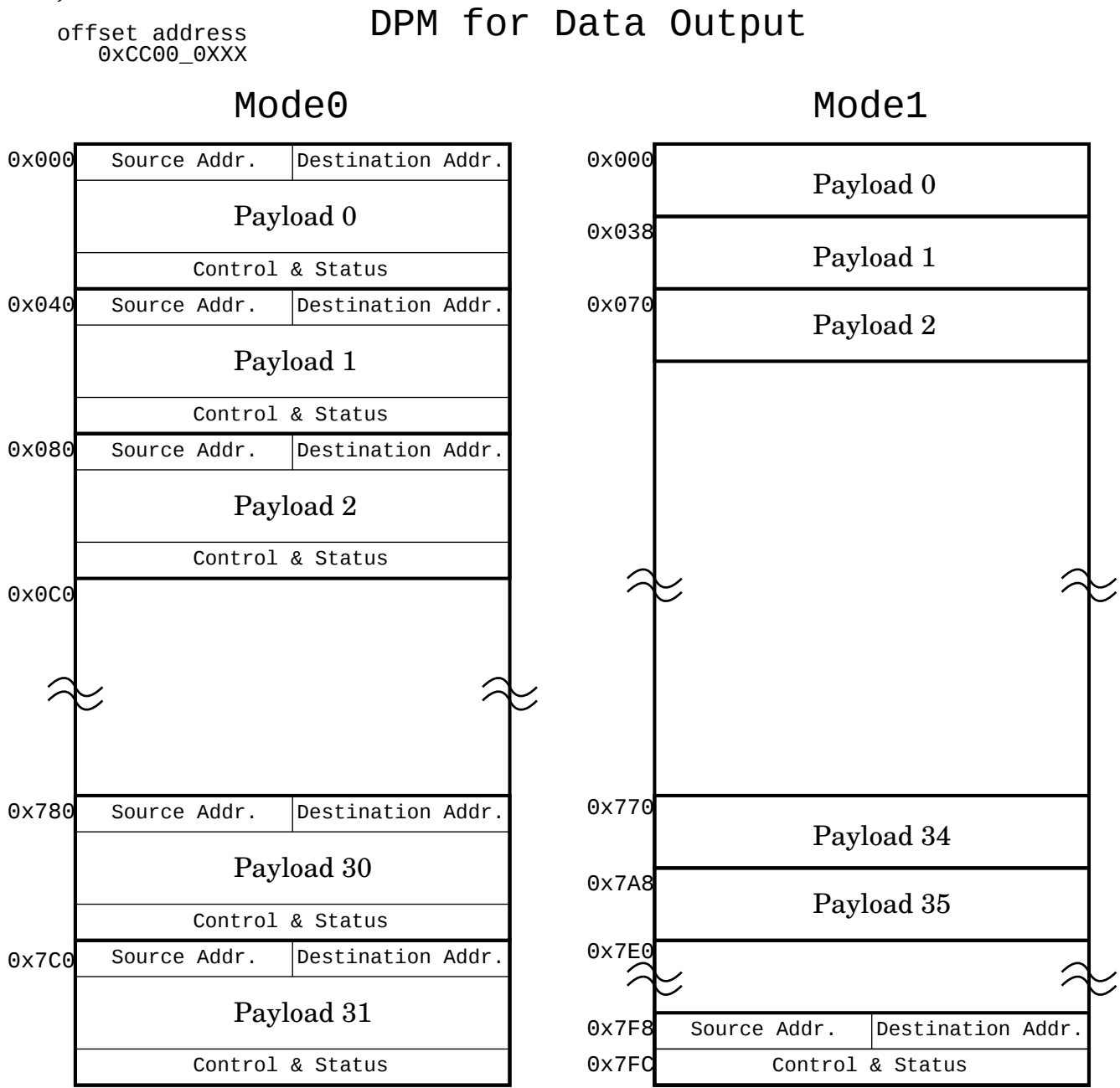


図 23.13: DPM for Data Output

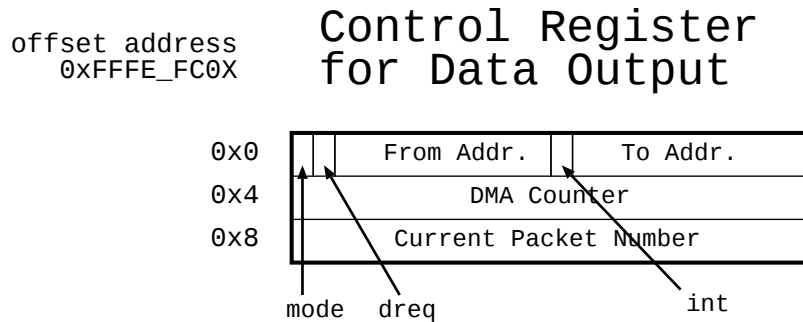


図 23.14: Data Out Control Register

DPM 制御レジスタ

DPM の制御レジスタ（図 23.14 参照）に以下を設定することで，送信の制御を行う．

制御レジスタ (r/w)

- Mode0: (r/w) mode bit に 0 を設定．すべてのパケットに headr と trailer を付加する．
- Mode1: (r/w) mode bit に 1 を設定．最後に共通の header と trailer を付加する（すべてのパケットの宛先が同一となる）．
- Int: (r/w) 本ビットを 1 に設定すると，終了時に EOP(End Of Packet) 割り込みを生成する．
- Dreq: (r/w) 本ビットを 1 に設定すると，DMA Counter に設定した回数だけ DMA を行う．
- From_Addr: (r/w) 設定された word address - 1 のアドレスに DPM のプロセッサバス側からデータが書かれた瞬間に DPM から Link0 に対して出力を開始する．
- To_Addr: (r/w) 設定された word address - 1 のアドレスに DPM のプロセッサバス側からデータが書かれた瞬間に DPM から Link0 に対して出力を開始する．

DMA Counter (r/w) DMA の回数を指定する

Current Packet Number (r) 現在送信されているパケット番号（図 23.13 の payload 番号に相当）を示す

23.11.4 Data Input

図 23.15 にデータリンク入力用 DPM の構成を示す．Data in control register（図 23.16 参照）に対して，開始アドレス From_Addr (byte address ではなく word address) と終了アドレス To_Addr (word address) を設定することにより，複数パケットを一度に受信できる．From_Addr と To_Addr は人間に分かりやすいようにこのような名前を付けられているが，実際には全く同じ機能のレジスタが二つ用意されている．From_Addr, To_Addr 共に，設定された word address - 1 のアドレスに DPM の *Responsive Link* 側からデータが書かれた瞬間に，DPM からプロセッサバス側に対して出力 (DMA 転送) を開始する (dreq bit が設定されている場合)．int bit の割り込みを利用して，ソフトウェアで受信することもできる．

例えば，Mode0 を使用し，From_Addr を 0x000 に設定し To_Addr を 0x01f (byte address: 0x07c) に設定したとする．*Responsive Link* 側から Payload0, Payload1,... の順に DPM に受信データが書かれていく．*Responsive Link* 側が

ら DPM の word address 0x1e 番地にデータが書かれた瞬間に DPM からプロセッサバス側に出力（DMA 転送）を開始する（この場合，実際には From_Addr には意味がない。）

あるいは，Mode0 を使用し，From_Addr を 0x0ff(byte address 0x3fc) に設定し To_Addr を 0x1ff(byte address: 0x7fc) に設定し，さらに DMAC を continuous mode で使用すると，Payload0～15 の領域と Payload16～31 の領域を使用して，主記憶等に用意した DPM よりも大きなメモリ領域（サイクリックバッファ等）に対して，受信データをハードウェアのみで連続的に自動受信することができる（DPM のアドレスデコードの範囲内では，シャドウアドレスでも CS が生成され DPM にアクセスできるように設計しているため。）

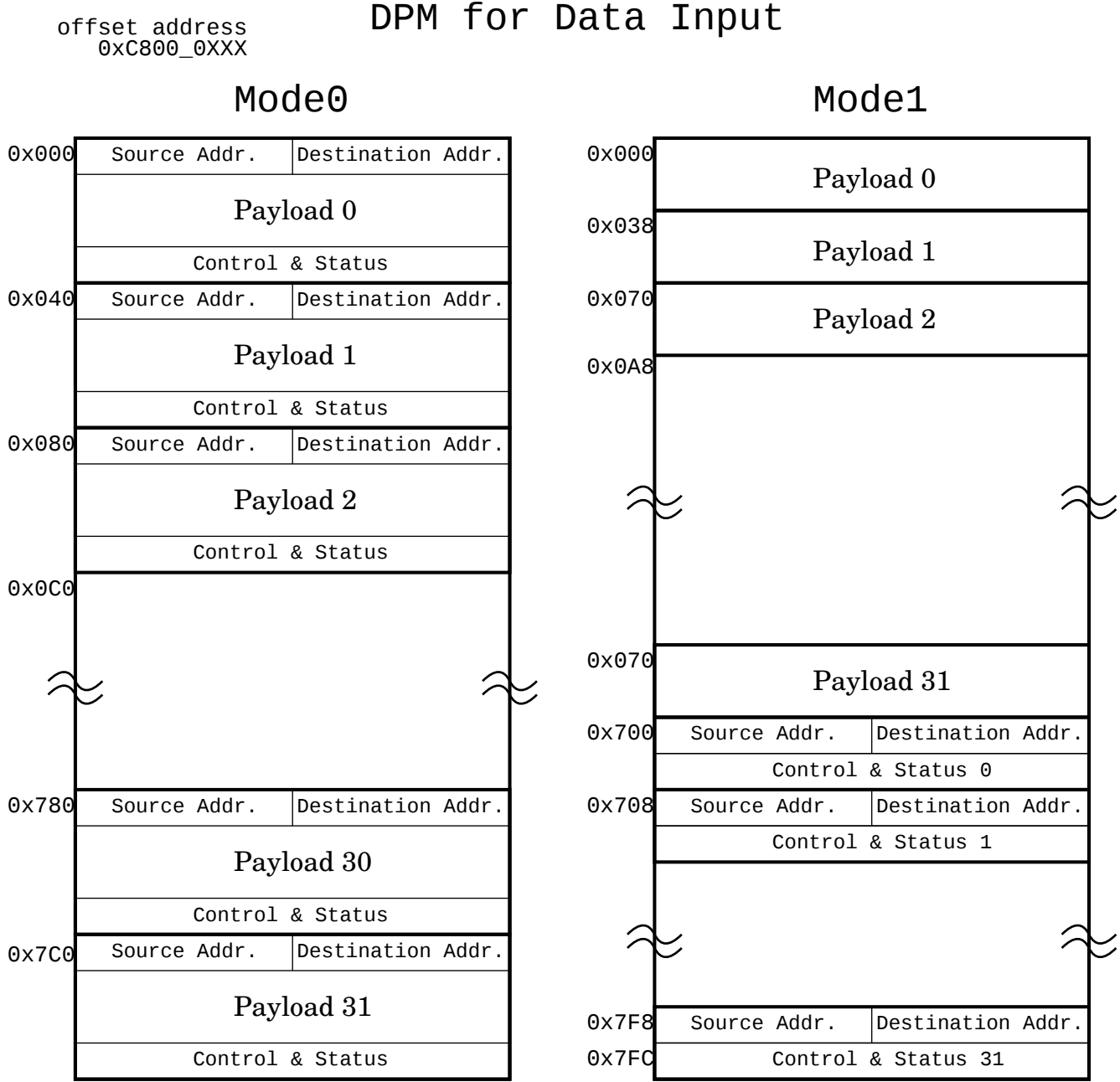
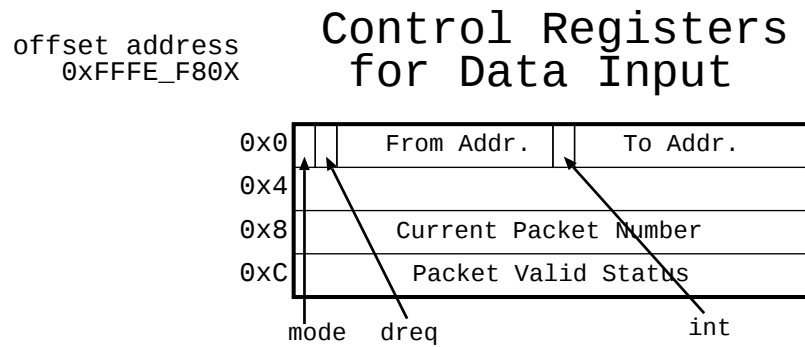


図 23.15: DPM for Data Input



DMA を用いた送信

DPM の容量には当然限界がある．しかし，レスポンスリンクでは，DMA と DPM が協調して動作することで，DPM の容量を越えるような大きなデータを一度に送信することが可能である．その際の手順は以下の通りである．ただし，総データ量は N_{packet} 分であることを仮定する．

DPM の設定

1. N の約数のうち最大のものを f とする．ただし，データリンクでは $f < 36$ ，イベントリンクでは f_{15} であるとする．
2. DPM の DMA Counter を $(N/f)-1$ に設定する．
3. DPM の MODE1_HEADER 及び MODE1_TRAILER に宛先およびパケットの持つべき性質（受信側での割込みなど）を設定する．
4. DPM のコントロールレジスタを mode 1, from(0), to($f \times 0xe + 0xd$), DREQ に設定する．（mode 0 で送信する場合，DMA の転送元にはパケットの形でデータが存在している必要がある．ただし，この場合手順 3 は必要無い．）

DMA の設定

1. DMA の送信元を送信したいデータの格納されているメモリの先頭アドレスに設定する．
2. DMA の送信先を送信用 DPM の先頭のアドレスにする．
3. DMA の送信先を送信用 DPM の先頭のアドレスにする．
4. DMA のコントロールレジスタは SAU, RL, MTM, ST を ON にする．（この ST による起動が DMA Counter の設定時に差し引いた 1 回に相当する）

23.12.2 相互通信の際の注意点

相互通信をする際に注意すべきは，二つのボードをつなげてから，まずそれぞれのボードにおいて「通信速度の設定」を行い，さらに，それぞれのボードにおいて「リンクの初期化」を行う．

当然，通信速度は同じでなければならない．また，リンクの初期化はボードをつなげてから行わないと相互通信ができないので注意すること．その他の設定は個別に各ボードで行う（モニタでの相互通信には，リンクの初期化モジュールを作成し，ボードをつなげたあと，そのモジュールを実行することにより相互通信を行う．）

23.13 *Responsive Link* の割り込みコントローラ

IRQ1~4 は DPM のコントロールレジスタの from addr と to addr までパケットが到達した時の割込みである．また，IRQ1~6 はレスポンスリンク割込みクリアレジスタ (0xa000_000c) に対応している．

レジスタマップにおける Request Sense Register, Request Clear Register, Mask Register は bit と IRQ が対応している．すなわち，各レジスタの i bit 目は IRQ_i に対応する．

初期アドレス: *Responsive Link* IRC: 0xa0001000

23.13.1 レジスタマップ

offset	31	24	23	16	15	8	7	0			
0x00	16'h0			IRQ15	IRQ14	IRQ13	IRQ12	IRQ11	IRQ10	IRQ9	IRQ8
0x04	16'h0			IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	2'h0
0x08	16'h0			Request Sense Register							0
0x0c	16'h0			Request Clear Register							0
0x10	16'h0			Mask Register							0
0x14	26'h0							CL	IRL Latch		
0x18	31'h0										Mode

表 23.4: *Responsive Link* レジスタマップのオフセット

Offset	Name
0x00	RL_IRC_TMR0_OFFSET
0x04	RL_IRC_TMR1_OFFSET
0x08	RL_IRC_RSR_OFFSET
0x0c	RL_IRC_RCR_OFFSET
0x10	RL_IRC_MR_OFFSET
0x14	RL_IRC_ICR_OFFSET
0x18	RL_IRC_MOD_OFFSET

表 23.5: *Responsive Link* IRC 割り込みマップ

IRQ	Name
IRQ31~IRQ14	Reserved
IRQ13	RL_DEC_RESET_IRQ
IRQ12	RL_IRQ_DOWN
IRQ11	RL_IRQ_WAKEUP
IRQ10	RL_IRQ_FATAL
IRQ9	RL_IRQ_TABLE
IRQ8	RL_IRQ_WAIT
IRQ7	RL_IRQ_CONT
IRQ6	RL_IRQ_EVP_IN
IRQ5	RL_IRQ_DAP_IN
IRQ4	RL_IRQ_EV_INEOP
IRQ3	RL_IRQ_DA_INEOP
IRQ2	RL_IRQ_EV_OUTEOP
IRQ1	RL_IRQ_DA_OUTEOP

IRQ	Name	Description
RL_DEC_RESET_IRQ	-	デコーダリセットしたときの割り込み
RL_IRQ_DOWN	-	リンクダウンしたら発生
RL_IRQ_WAKEUP	-	リンクアップしたら発生
RL_IRQ_FATAL	致命的エラー割り込み FI (Fatal IRQ)	受信パケットに回復不可能なエラーが存在する場合に発生
RL_IRQ_TABLE	ルーティングテーブル割り込み RTIRQ (Routing IRQ)	ルーティングテーブルにマッチするエントリが存在しない場合に発生
RL_IRQ_WAIT	送信停止割り込み WIRQ (Wait IRQ)	パケット追い越し用 SDRAM を使用している際に、追い越し用 SDRAM が溢れそうになると送信停止割り込みを自動生成する
RL_IRQ_CONT	レスポンスリンク継続割り込み CI (Continuous IRQ)	SDRAM に退避されたパケットがスイッチに書き戻された（再送信された）際に発生
RL_IRQ_EVP_IN	Event Packet-In IRQ	割り込みフラグ付きの event パケットを受信した際に発生
RL_IRQ_DAP_IN	Data Packet-In IRQ	割り込みフラグ付きの data パケットを受信した際に発生
RL_IRQ_EV_INEOP	Event-In End of Packet	Event-In DPM で設定した場所までパケットを受信した際に発生
RL_IRQ_DA_INEOP	Data-In End of Packet	Data-In DPM で設定した場所までパケットを受信した際に発生
RL_IRQ_EV_OUTEOP	Event-Out End of Packet	Event-Out で DPM からパケットを発射した際に発生
RL_IRQ_DA_OUTEOP	Data-Out End of Packet	Data-Out で DPM からパケットを発射した際に発生

24

更新履歴

Revision	Date	Description
第 1 版	2014 年 5 月 12 日	初版公開 .
第 2 版	2014 年 10 月 7 日	Abstract , Bus , External Bus, Timer, UART, PIO, SPI, Responsive Link の章を更新 .
第 3 版	2014 年 10 月 11 日	Responsive Link の割り込みコントローラの説明を修正 .
第 4 版	2014 年 10 月 14 日	Responsive Link の通信コーデック設定レジスタのオフセットを修正 .
第 5 版	2014 年 10 月 21 日	Responsive Link の割り込みクリアレジスタを修正 , デコーダリセット割り込みクリアレジスタ , パラレルモードレジスタ , エラーパケットヘッダレジスタ , エラーヘッダポインタレジスタ , エラーパケットモードレジスタ , SDRAM 回復イネーブルレジスタを追加 . UART の Line Control Register を修正 .
第 6 版	2016 年 10 月 27 日	ハイパーリンクを追加 .