

**RTE-VR4310-PC**

ユーザーズ・マニュアル(Rev. 1.02)

*Midas lab*

## 改定履歴

日付 Y M D	Rev	内容
98 7 4	1.00	暫定初版
98 9 4	1.01	正式初版
99 4 10	1.02	誤記訂正 5.2 割り込み...INT2 <-> INT3 入れ換え

## 目次

<b>1. はじめに.....</b>	<b>1</b>
1.1. マニュアル表記について .....	1
<b>2. 特徴と機能.....</b>	<b>2</b>
<b>3. ボードの構成.....</b>	<b>3</b>
3.1. リセット・スイッチ ( SW RST ) .....	3
3.2. 電源コネクタ ( JPOWER ) .....	3
3.3. LED ランプ .....	4
3.4. DIVMODE スイッチ ( SW1 ) .....	4
3.5. ROM-BANK スイッチ ( SW3 ) .....	4
3.6. 汎用入力スイッチ ( SW4 ) .....	5
3.7. ユーザ制御ピン ( JUSR ) .....	5
3.8. CPU テスト・コネクタ ( JC1,JC2 ) .....	6
3.9. シリアル・コネクタ ( JSIO1,2 ) .....	6
3.10. パラレル・コネクタ ( JPRT ) .....	6
3.11. オシレータ・ソケット(OSC1).....	7
3.12. DRAM-SIMM ソケット .....	7
3.13. ROM ソケット .....	7
<b>4. ホストPCとの接続.....</b>	<b>8</b>
4.1.1. ボード単体で使用する場合 ( RS-232C 接続 ) .....	8
4.1.2. PCI スロットに組み込む場合 ( PCI バス接続 ) .....	8
<b>5. ハードウェア・リファレンス.....</b>	<b>9</b>
5.1. リセット.....	9
5.2. 割り込み.....	9
5.3. アドレス・マップ .....	10
5.3.1. SRAM 空間 ( 0000-0000H ~ 07FF-FFFFH ) .....	10
5.3.2. DRAM 空間 ( 0800-0000H ~ 0FFF-FFFFH, x800-0000H ~ xFFF-FFFFH ) .....	10
5.3.3. MEM-CNT 空間 ( 1000-0000H ~ 17FF-FFFFH ) .....	10
5.3.4. 未使用 ( 1800-0000H ~ 19FF-FFFFH ) .....	11
5.3.5. EXT-BUS 空間 ( 1A00-0000H ~ 1BFF-FFFFH ) .....	11
5.3.6. I/O 空間 ( 1C00-0000H ~ 1EFF-FFFFH ) .....	11
5.3.7. ROM 空間 ( 1F00-0000H ~ 1FFF-FFFFH ) .....	11
5.4. I/O 詳細.....	11
5.4.1. SRAM コントローラ ( SRAMC ) .....	12
5.4.2. DRAM コントローラ ( DRAMC ) .....	12
5.4.3. 汎用入出力ポート ( SWLED ) .....	14
5.4.4. シリアル/パラレルI/O ( SCC0/1, LPT ) .....	15
5.4.5. タイマ ( Timer ) .....	15

5.4.6. 割り込みコントローラ (PIC) .....	17
5.4.7. PCI コントローラ .....	18
<b>6. バスサイクル・タイミング.....</b>	<b>19</b>
6.1. SRAM アクセス .....	19
6.2. DRAM アクセス.....	21
6.3. ローカルバス・アクセス .....	25
6.4. メモリコントローラ・レジスタアクセス.....	29
<b>7. EXT-BUS 仕様.....</b>	<b>30</b>
7.1. コネクタの仕様.....	30
7.2. 信号.....	31
7.3. EXT-BUS タイミング .....	32
<b>8. ソフトウェア.....</b>	<b>33</b>
8.1. 割込みプログラミング .....	33
8.1.1. 割込みライブラリ .....	33
8.1.2. 割込みルーチン .....	34
8.1.3. サンプルプログラム .....	34
8.2. ROM プログラミング .....	35
8.2.1. 初期化 .....	35
8.2.2. 割り込み .....	35
8.2.3. ROM データ配置 .....	36
<b>9. APPENDIX.A MULTI モニタ .....</b>	<b>37</b>
9.1. RTE FOR WIN32 のインストール.....	37
9.2. ボードの設置.....	37
9.2.1. スイッチの設定 .....	37
9.2.2. ホスト PC とボードの接続 .....	38
9.3. MULTI モニタ .....	39
9.3.1. モニタ・ワーク RAM .....	39
9.3.2. 割り込み .....	39
9.3.3. 強制ブレーク用の割り込み .....	39
9.3.4. スタックの設定 .....	39
9.3.5. 特殊命令 .....	39
9.4. RTE コマンド .....	39
9.4.1. HELP(?) .....	40
9.4.2. INIT .....	40
9.4.3. VER .....	40
9.4.4. CACHEFLUSH .....	40
9.4.5. SHOWTLB .....	40
9.4.6. IOREAD .....	40
9.4.7. IOWRITE .....	40

<b>10. APPENDIX.B PARTNERI モニタ .....</b>	<b>41</b>
10.1. ボードの設置.....	41
10.1.1. スイッチの設定.....	41
10.1.2. ホスト PC とボードの接続.....	42
10.2. PARTNER モニタ .....	42
10.2.1. モニタ・ワーク RAM.....	42
10.2.2. 割り込み .....	42
10.2.3. 強制ブレーク用の割り込み .....	42
10.2.4. スタックの設定.....	42
10.2.5. 特殊命令 .....	42
<b>11. APPENDIX.C JC1,JC2 コネクタ .....</b>	<b>43</b>

## 1. はじめに

このマニュアルでは、日本電気製 RISC-CPU VR4310 用の評価ボード「RTE-VR4310-PC」について説明します。本ボードは、最高 166MHz で動作する VR4310 CPU と SRAM, DRAM, ROM とシリアル(2ch) / パラレル(1ch)とタイマなどの I/O で構成されます。

プロセッサの性能評価、デモ、シミュレータの実行エンジン、アプリケーション・プログラムの開発など、幅広くご利用いただけます。

本製品は、開発用のソフトウェアツールとして、GHS 社の Multi と自社製の PARTNER のどちらかをソースレベルデバッガとしてご使用になれます。ご使用になるデバッガによって、ROM に搭載するモニタは異なります。

ROM は、購入時にご指定頂いたモニタが搭載されています。デバッガを同時に購入されていない場合は、それぞれ別売りされていますので、別途お買い求めください。

### 1.1. マニュアル表記について

本書では、数字の表記については下表の表記を用います。16 進数や 2 進数の表記では、桁数が多くて読みにくい場合は、4 桁ごとに“-”（ハイフン）を入れてあります。また、数値で任意を示す場合には、”1FxxH” のように”x” で示します。

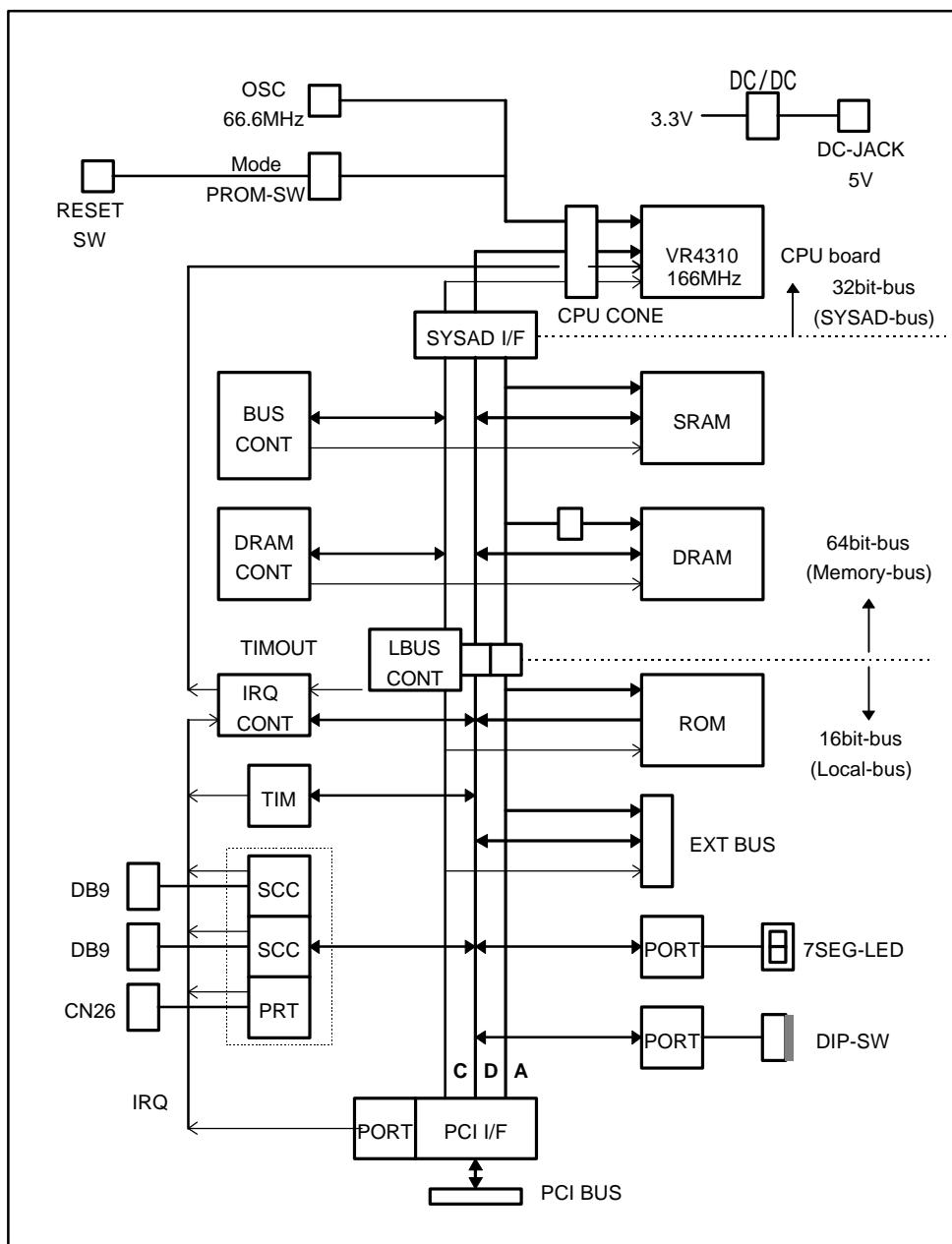
進数	表記規則	例
10 進数	数字のみを示します	“10”は 10 進数の“10”を示します
16 進数	数字の末尾に”H”を記します	“10H”は 10 進数の“16”を示します
2 進数	数字の末尾に”B”を記します	“10B”は 10 進数の“2”を示します

数字表記規則

MULTI は、米国 Green Hills Software 社の商標です。

## 2. 特徴と機能

RTE-VR4310-PC の機能ブロックの概要を下図に示します。



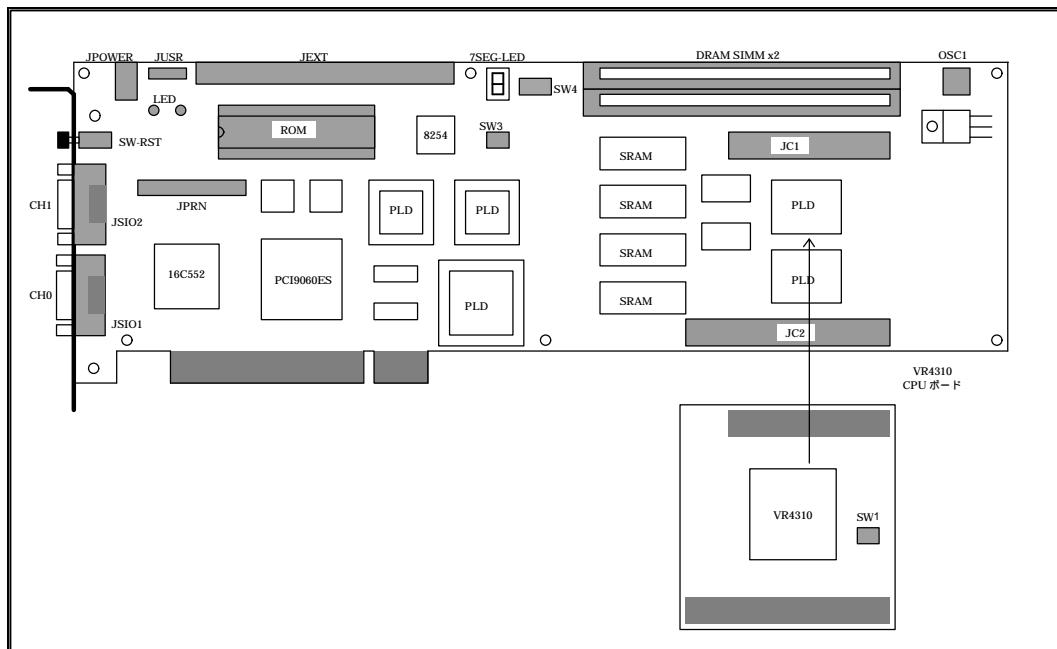
ブロック図

### 特徴

- ROM 256K バイト (128K × 16 ビット × 1)
- SRAM 512K バイト (64K × 16 ビット × 4)
- DRAM SIMM72 ピン・ソケット(×2)により 8M,16M,32M (標準 8M バイト)
- RS-232C ポート × 2ch (D-SUB 9 ピン)
- ROM インサーキット・デバッガ用の接続ピン

### 3. ボードの構成

下図はボード上の主要な部品の物理的な配置です。ここでは、それぞれの部品について説明します。



ボード概観

#### 3.1. リセット・スイッチ(SW RST)

SW RST はリセット・スイッチです。このスイッチを押すとボード全体がリセットされます。

#### 3.2. 電源コネクタ(JPOWER)

外部から電源を供給するための DC ジャックです。

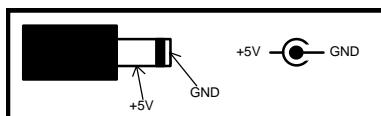
JPOWER に供給する電源は、下記の通りです。

電圧： 5 V

電流： 最大 3.5A

適合コネクタ：Type A ( 5.5 )

極性：



**【注意】** 接続にあたっては、極性に十分注意してください。

### 3.3. LED ランプ

LED は、下表のような状態を示します。

LED	名称	意味	備考
LED1	TOVER	バス・タイムアウト発生	ローカルバスのみ
LED2	POWER	電源 ON 時に点灯	
U36	-	7 セグメント LED ( ポート出力 )	

7 セグメント LED は汎用の出力ポートからの制御で点灯 / 消灯させることができます。詳細は、「5.4.3 汎用入出力ポート ( SWLED )」を参照してください。

### 3.4. DIVMODE スイッチ ( SW1 )

SW1 は、CPU の DIVMODE[2..0] ピンのレベルを指定するスイッチです。OFF で 1 、 ON で 1 です。

SW3	SW3	SW3	DIV	DIV	DIV	分周率(Mclk/Pclk)	備考
							( * は出荷時の設定 )
1	2	3	0	1	2		
OFF	OFF	OFF	1	1	1	1:3	
ON	OFF	OFF	0	1	1	1:2	
OFF	ON	OFF	1	0	1	1:1.5	
ON	ON	OFF	0	0	1	1:4	
OFF	OFF	ON	1	1	0	1:3	
ON	OFF	ON	0	1	0	1:2.5	*
OFF	ON	ON	1	0	0	1:6	
ON	ON	ON	0	0	0	1:5	

注意 : PCLK が VR4310 の動作保証周波数以下になる設定でご使用下さい。

### 3.5. ROM-BANK スイッチ ( SW3 )

SW3 は、ROM のバンク設定を行うためのスイッチです。スイッチにより、ROM の最上位アドレス 2 ビットを操作して、最大各 64K バイト空間を 4 つバンクとしてもつことができます。

SW3	32/64 ビット動作(MD32/64)			備考 ( * は出荷時の設定 )
	1			
ON	0: 64 ビットモード			
OFF	1: 32 ビットモード			*

SW3	ROM アドレス			バンク選択 ( * は出荷時の設定 )
	2	3	ROM-A15	ROM-A16
ON	ON	CPU-A16	CPU-A17	バンク無し
ON	OFF	CPU-A16	ENDIAN	エンディアンのみ
OFF	ON	CPU-A16	MD32/64	動作ビットモードのみ
OFF	OFF	ENDIAN	MD32/64	フルバンク *

SW3 4		エンディアンの指定	備考 (*は出荷時の設定)
ON	0: リトル・エンディアン		
OFF	1: ビッグ・エンディアン		*

搭載されているモニタでは、以下の通り使用しています。

ROM アドレス(16bit)	バンク	MULTI モニタの場合(*1)
0x00000-0x07FFF	バンク 0	リトルエンディアン / 64 ビット用
0x08000-0x0FFFF	バンク 1	ビッグエンディアン / 64 ビット用
0x10000-0x17FFF	バンク 2	リトルエンディアン / 32 ビット用
0x18000-0x1FFFFFF	バンク 3	ビッグエンディアン / 32 ビット用

\*1: PARTNER の場合は、Bank2,3 のみ使用しています。

### 3.6. 汎用入力スイッチ(SW4)

SW4 は汎用入力ポートのデータ・スイッチです。スイッチは、OFF で 1, ON で 0 の値となります。詳細は「9.2.1 または、10.1.1 スイッチの設定」および「5.4.3 汎用入出力ポート (SWLED)」を参照してください。

### 3.7. ユーザ制御ピン(JUSR)

JUSR コネクタは主に ROM インサーキット・デバッガを使用する際に接続する制御端子です。ROM インサーキット・デバッガからのリセットや割り込みが入力できます。

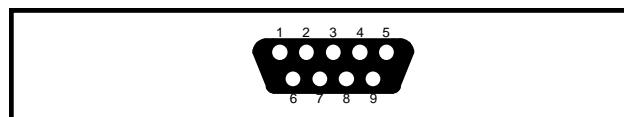
JROMEM	名称	機能	備考
1	RST	RESET-入力 (Low アクティブ)	10K プルアップ
2	NMI	NMI-入力 (Low アクティブ)	10K プルアップ
3	IRQ	INT3-入力 (Low アクティブ)	10K プルアップ
4	GND	接地	

### 3.8. CPU テスト・コネクタ (JC1,JC2)

JC1,JC2 コネクタは CPU 端子が接続されています。信号測定などのテスト用としてや CPU ボードの接続コネクタとして使用します。コネクタのピン配置については、「11 APPENDIX.C JC1,JC2 コネクタ」を参照してください。

### 3.9. シリアル・コネクタ (JSIO1,2)

JSIO1,2 コネクタは、シリアル／パラレル・コントローラ (TL16C552A) によって制御される RS-232C 用のコネクタです。コネクタの形状は、PC/AT で用いられる一般的な D-SUB9 ピンの RS-232C コネクタ（オス）で、全ての信号は RS-232C レベルに変換されています。コネクタのピン番号と内容は以下の通りです。また表には、ホスト PC と接続する場合の接続信号について、ホスト PC 側が D-SUB9 ピンの場合と D-SUB25 ピンの場合の布線をそれぞれ示してあります（一般的なクロスケーブルの布線です）。



JSIO1,2 ピン配置図

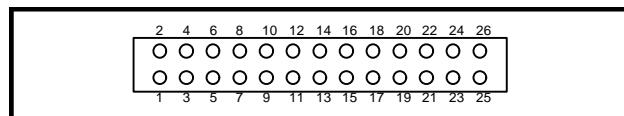
JSIOx ピン	信号名	入出力	ホスト PC の接続ピン番号	
			D-SUB9	D-SUB25
1	DCD	入力		
2	RxD (RD)	入力	3	2
3	TxD (SD)	出力	2	3
4	DTR (DR)	出力	1, 6	6, 8
5	GND		5	7
6	DSR (ER)	入力	4	20
7	RTS (RS)	出力	8	5
8	CTS (CS)	入力	7	4
9	RI	入力		

JSIO1,2 コネクタ信号表

【メモ】パネルには JSIO1 が “CH0”、JSIO2 が “CH1” と記されています。

### 3.10. パラレル・コネクタ (JPRT)

JPRT コネクタは、シリアル／パラレル・コントローラ (TL16C552A) によって制御されるプリンタ用のコネクタです。コネクタは 26 ピンのヘッダピンですので、一般的なプリンタ・コネクタとして使用するには変換ケーブルが必要です。コネクタのピン番号と内容は以下の通りです。



JPRT ピン配置図

JPRT ピン	信号名	入出力	備考
1	STB-	出力	10K プルアップ
2	AUTO_FD-	出力	10K プルアップ
3	D0	出力	10K プルアップ
4	ERROR-	入力	10K プルアップ
5	D1	出力	10K プルアップ
6	INIT-	出力	10K プルアップ
7	D2	出力	10K プルアップ
8	SELECT_IN-	出力	10K プルアップ
9	D3	出力	10K プルアップ
11	D4	出力	10K プルアップ
13	D5	出力	10K プルアップ
15	D6	出力	10K プルアップ
17	D7	出力	10K プルアップ
19	ACK-	入力	10K プルアップ
21	BUSY	入力	10K プルアップ
23	PE	入力	10K プルアップ
25	SELECT	入力	10K プルアップ
26	NC		未使用
10,12,14,16, 18,20,22,24	GND		

JPRT コネクタ信号表

### 3.11. オシレータ・ソケット(OSC1)

オシレータ・ソケットは、CPU に供給するクロックのためのソケットで、出荷時には 66.6MHz のオシレータが実装されています。

【注意】オシレータの足を切って実装する場合、足が短かすぎるとフレーム（外装）部分が、ソケットの端子とショートしてしまいますので注意してください。

### 3.12. DRAM-SIMM ソケット

DRAM-SIMM ソケットには、標準で 4M バイトの SIMM が実装されています。このソケットには 4M, 8M, 16M バイトの 72 ピン SIMM (いわゆる DOS/V 機用と言われるもの) が実装できますので、DRAM の容量の増設が可能です。また、実装されている SIMM の種類は I/O ポートから読み出すことができます (「5.4.2DRAM コントローラ (DRAMC)」参照)。

### 3.13. ROM ソケット

ROM ソケットには、標準で 256K バイト (128K × 16 ビット) の 40 ピン ROM が 1 つ実装されています。アクセス・タイムが 150ns 以下のものをご使用ください。ROM は、エンディアンや動作モードに対応させるため、アドレス切り替えにより最大 4 つのバンクを持たすることができます。バンクの設定に関しては、「3.4DIVMODE スイッチ (SW1)」および「9.2.1 スイッチの設定」を参照してください。

#### 4. ホストPCとの接続

##### 4.1.1. ボード単体で使用する場合(RS-232C接続)

ホストPCと接続する際には、以下の手順で行ないます。

ホストPCと接続するためのRS-232Cケーブルと、電源供給のための外部電源(+5V 4A)を用意してください。特に電源については、電圧とコネクタの極性に注意してください。

RS-232Cケーブルの結線は「3.9シリアル・コネクタ(JSIO1,2)」、電源コネクタについては、「3.2電源コネクタ(JPOWER)」を参照してください。

JSIO1(CH0)コネクタにRS-232CケーブルでホストPCと接続し、JPOWERコネクタから電源を供給して、ボードのPOWER-LEDが点灯することを確認します。**LEDが点灯しない場合は、すぐに電源を切り接続を確認してください**

ホストPCでMULTIデバッガを起動して、RS-232C経由でコネクトします。エラーが発生する場合には、シリアル・ケーブルやスイッチ(特にボーレート)の設定等に間違いがないかを確認してください。

##### 【注意】

1. 電源を入れて動作させるとCPUや近辺の放熱版が熱くなってきます。また、電源を切った後もしばらくの間熱いままですので、手など触れないようご注意ください。
2. ボードは絶縁物の上に設置してください。通電中に導電物を基板に接触させると故障の原因になります。
3. PCIスロットに組込んで使用しない場合は、必ず添付のPCIバスターミネータ基板を接続してください。

##### 4.1.2. PCIスロットに組み込む場合(PCIバス接続)

ホストPCのPCIスロットに組み込んで使用する場合には、以下の手順で行います。

ホストPCの筐体を開き、PCIスロットにボードを取り付けます。しっかりと差し込まれたかを確認して、バックパネルをねじ止めしてください。

ホストPCの電源を入れてボードのPOWER-LEDが点灯することを確認します。**LEDが点灯しない場合は、すぐに電源を切り接続を確認してください**。また、ホストPCが正常に起動するかも確認してください。

ホストPC上でデバッガを起動して、PCIバス経由でコネクトします。エラーが発生する場合には、ボードの実装やソフトウェアのインストールに間違いがないかを確認してください。

## 5. ハードウェア・リファレンス

ここでは、RTE-VR4310-PC ボードのハードウェア機能について説明します。

### 5.1. リセット

リセットは下記に示した要因で発生します。これらは CPU に対してコールドリセットとして扱われ、ボード上の制御回路へのシステム・リセットとなります。

- ・**パワーオン・リセット**：ボードの電源 ON 時に発生します。
- ・**リセット・スイッチ**：ボードに用意されているリセット・スイッチ ( SW RST ) により発生します。
- ・**ホスト・リセット**：ホスト PC から PCI バス・コントローラを通してのリセット要求です。
- ・**JUSR-1**：JUSR コネクタの 1 番端子(RST)からの入力です。

### 5.2. 割り込み

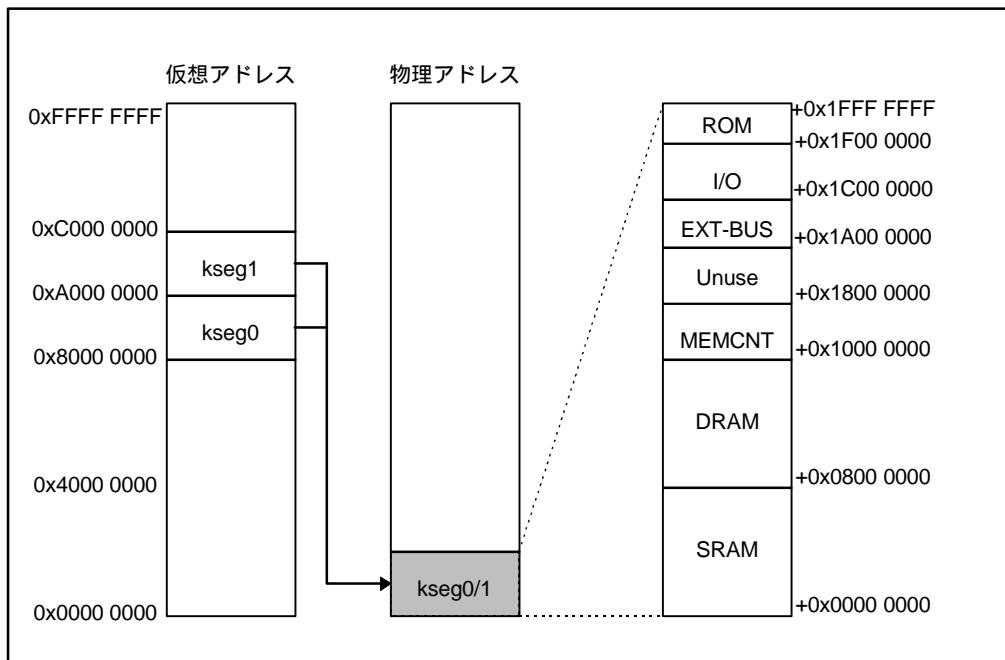
CPU 外部からの割り込み要因は、以下の通りです。

割り込み	要因	参照
NMI-	JUSR-2 (NMI)	3.7 ユーザ制御ピン ( JUSR )
INT0-	割り込みコントローラ INT0M	5.4.6 割り込みコントローラ ( PIC )
INT1-	割り込みコントローラ INT1M	5.4.6 割り込みコントローラ ( PIC )
INT2-	EXT-BUS	7 EXT-BUS 仕様
INT3-	JUSR-3 (IRQ)	3.7 ユーザ制御ピン ( JUSR )
INT4-	未使用 (1 固定)	

外部割り込みはハード的にマスク制御することができます（「5.4.6 割り込みコントローラ ( PIC )」を参照）。

### 5.3. アドレス・マップ

ボードのアドレス割り付けは、以下の通りです。



アドレス・マップ

MULTI モニタを使用してプログラム実行する場合には、TLB マップを使用しないカーネル空間 8000-0000H ~ BFFF-FFFFH で行ないます。キャッシュを使用する場合には 8000-0000H , 使用しない場合には A000-0000H を、物理アドレスに加算して論理アドレスを決定します。

#### 5.3.1. SRAM 空間 (0000-0000H ~ 07FF-FFFFH)

ボード上に実装された SRAM は 512K バイトの実空間があり、他はイメージ空間となります。SRAM のアクセスは、外部バスロックが 40MHz 以上の場合に 1 ウェイト以上を SRAM コントローラ (SRAMC) で設定してください（「5.4.1SRAM コントローラ (SRAMC)」を参照）。キャッシュ / 非キャッシュ空間でアクセスできます。

#### 5.3.2. DRAM 空間 (0800-0000H ~ 0FFF-FFFFH, x800-0000H ~ xFFF-FFFFH)

ボードに実装する 72 ピンの DRAM-SIMM の空間で、標準では 4M バイトの SIMM が実装されていますが、8M や 16M バイトの SIMM に交換することでメモリ容量を拡張できます。また、実装された DRAM-SIMM × 2 の実空間以外にはイメージが発生します。DRAM のアクセスタイミングは、DRAM コントローラ (DRAMC) で設定します（「5.4.2DRAM コントローラ (DRAMC)」を参照）。キャッシュ / 非キャッシュ空間でアクセスできます。

#### 5.3.3. MEM-CNT 空間 (1000-0000H ~ 17FF-FFFFH)

SRAM,DRAM のメモリアクセス制御を行うレジスタがマップされています。詳細については「5.4.1SRAM コントローラ (SRAMC)」と「5.4.2DRAM コントローラ (DRAMC)」を参照してください。非キャッシュ空間でアクセスします。

### 5.3.4. 未使用(1800-0000H ~19FF-FFFFH)

この空間をアクセスすると、タイムオーバー機能が有効な場合には、タイムオーバーレディによりバス・サイクルが強制終了させられて、タイムオーバー割り込みが発生します。

### 5.3.5. EXT-BUS 空間(1A00-0000H ~1BFF-FFFFH)

外部拡張バスである EXT-BUS 用の空間で 1M バイトの実空間となり、他はイメージ空間となります。非キャッシュ空間でアクセスします。

### 5.3.6. I/O 空間(1C00-0000H ~1EFF-FFFFH)

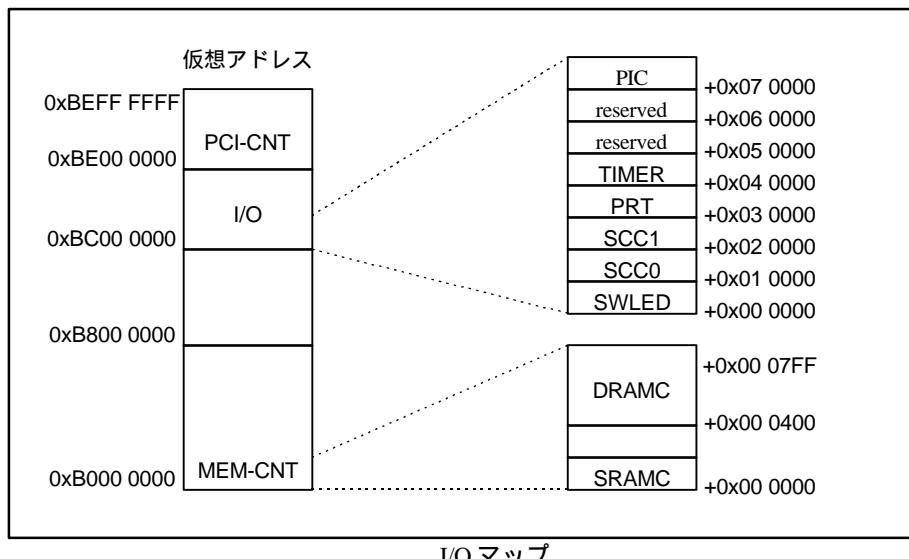
ボード上の各機能を制御するための I/O デバイスが割り付けられている空間(メモリ・マップド I/O)です。I/O マップなどの詳細については「5.4 I/O 詳細」を参照してください。非キャッシュ空間でアクセスします。

### 5.3.7. ROM 空間(1F00-0000H ~1FFF-FFFFH)

ボード上に実装された ROM で 256K バイト(272048)の実空間となり、他はイメージ空間となります。実装されている ROM には、MULTI 用モニタが組み込まれています。非キャッシュ空間でのみアクセスできます。

## 5.4. I/O 詳細

メモリ空間にマップされた I/O デバイスでメモリアクセスコントローラ、DUART/LPT,TIMER、割り込みコントローラ、PCI(通信用 I/O)コントローラなどがあります。ボードの I/O 割り付けは、以下の通りです。



これらの I/O はカーネル非キャッシュ空間でのアクセスを前提としているため、以降の説明では論理アドレスを用います。

**【メモ】** I/O デバイスに接続されているデータバスは特に明記がない場合、D[7..0]と接続されていますので、バイトアクセス時にはエンディアンを考慮する必要があります。ビッグ・エンディアンでバイトアクセスする場合には、バイト・オフセットとして+7 を加算します。(バスは、内部的に 64bit で扱っている為です)

#### 5.4.1. SRAM コントローラ(SRAMC)

SWAIT レジスタは、SRAM 空間のウェイト制御を行ないます。このレジスタによってリードサイクルに 0~3 ウェイトが設定できます。レジスタ割り付けは下表の通りです。

論理アドレス	レジスタ	データバス			
		D3	D2	D1	D0
B000-0000H	SRAMC SWAIT	0	0	SWAIT1	SWAIT0

SWAIT[1..0]: SRAM リード時のウェイト数を設定します。

SWAIT	SRAM リード・ウェイト数			
	1	0	1	2
0	0	0		
0	1	1		
1	0	2		
1	1	3		(リセット値)

**【メモ】** SRAM のウェイト 0 を設定できる条件は、外部バスクロック(SysClock)が 40MHz 以下 (1 クロック幅 25ns 以上) の場合です。これは、以下の計算により求められます。  
 アドレス遅延 + SRAM アクセス時間 + データセットアップ =  
 5ns + 17ns + 3ns = 25ns

#### 5.4.2. DRAM コントローラ(DRAMC)

DRAMC は DRAM のアクセス制御を行ないます。RAS/CAS 幅や各種の動作モードを設定します。レジスタ割り付けは下表の通りです。

論理アドレス	レジスタ	データバス			
		D3	D2	D1	D0
B000-0400H	DRAMC RCAS	0	0	RCAS1	RCAS0
-0500H	DRAMC MRAS	0	MRAS2	MRAS1	MRAS0
-0600H	DRAMC PRAS	0	PRAS2	PRAS1	PRAS0
-0700H	DRAMC DMODE	PD1	PD2	EDOEN	HITEN

RCAS[1..0]: DRAM リード時の CAS サイクルのクロック数を設定します。設定した値 (0 ~ 3) +1 のクロック数となります。

RCAS	CAS リードサイクル数			
	1	0	1	2
0	0	1 SYSCLK		
0	1	2 SYSCLK		
1	0	3 SYSCLK		
1	1	4 SYSCLK		(リセット値)

**【メモ】** CAS のプリチャージ・サイクルは 1 クロックに固定されています。

**MRAS[2..0]:** DRAM の RAS アクセス・サイクル数を設定します。設定した値 (0~7) +1 のクロック数となります。

MRAS			RAS アクセス・サイクル数
2	1	0	
0	0	0	1 SYSCLK
0	0	1	2 SYSCLK
0	1	0	3 SYSCLK
0	1	1	4 SYSCLK
1	0	0	5 SYSCLK
1	0	1	6 SYSCLK
1	1	0	7 SYSCLK
1	1	1	8 SYSCLK (リセット値)

**PRAS[2..0]:** DRAM の RAS プリチャージ・サイクル数を設定します。設定した値 (0~7) +1 のクロック数となります。

PRAS			RAS プリチャージ・サイクル数
2	1	0	
0	0	0	1 SYSCLK
0	0	1	2 SYSCLK
0	1	0	3 SYSCLK
0	1	1	4 SYSCLK
1	0	0	5 SYSCLK
1	0	1	6 SYSCLK
1	1	0	7 SYSCLK
1	1	1	8 SYSCLK (リセット値)

**HITEN:** DRAM コントローラのページヒットアクセス機能を使用するかを設定します。HITEN=1 の時には、DRAM アクセス終了後 RAS をホールドして、次の DRAM アクセス時のロードアドレス (DRAM のページアドレス) と一致した場合に、高速ページアクセスを実現します。ただし、RAS サイクルのホールドはリフレッシュ要求により解除されます。また、ヒットしなかった場合には、プリチャージサイクル後に通常のアクセスが行なわれます。

HITEN	DRAM のページヒット
0	使用しない (リセット値)
1	使用する

**EDOEN:** DRAM の種類を FPM か、EDO タイプかを設定します。EDOEN=1 の場合、バーストストリード時のデータレディが CAS プリチャージ期間となるため、CAS 幅を 1 クロック短くできることがあります。

EDOEN	DRAM のタイプ
0	FPM-DRAM (リセット値)
1	EDO-DRAM

**PD[1..2]: DRAM-SIMM の種別端子 PD[1..2]が読み出せます（リード専用）。**

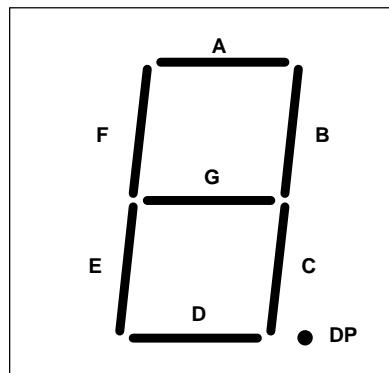
PD1	PD2	DRAM-SIMM の容量	
0	0	4M バイト	(標準)
0	1	16M バイト	
1	0	(予約)	
1	1	8M バイト	

#### 5.4.3. 汎用入出力ポート（SWLED）

汎用入出力ポートは、ボード上のディップ・スイッチ(SW4)の設定値の読みだしと、7セグメント LED の点灯制御を行ないます。

論理アドレス	データバス								内容
	D7	D6	D5	D4	D3	D2	D1	D0	
BC00-0000H 入力	SW4 -8	SW4 -7	SW4 -6	SW4 -5	SW4 -4	SW4 -3	SW4 -2	SW4 -1	0=ON 1=OFF
BC00-0000H 出力	7SEG -DP	7SEG -G	7SEG -F	7SEG -E	7SEG -D	7SEG -C	7SEG -B	7SEG -A	0=点灯 1=消灯

7セグメント LED のビット対応は、下図の通りです。



**【注意】 7セグメント LEDへの出力データをリードすることはできません。**

#### 5.4.4. シリアル／パラレルI/O (SCC0/1, LPT)

シリアル／パラレルI/OとしてTI製のTL16C552Aを使用しています。これは、NS16550互換のシリアルコントローラを2つとPC/AT(PS2)互換のパラレル・ポートを1つ内蔵したデバイスです。TL16C552Aの各レジスタの割り付けは、以下の通りです。

論理アドレス	レジスタ	補足
<b>BC01-0000H</b>	RBR/THR/DLL	SCC0
-0100H	IER/DLM	
-0200H	IIR/FCR	
-0300H	LCR	
-0400H	MCR	
-0500H	LSR	
-0600H	MSR	
-0700H	SCR	
<b>BC02-0000H</b>	RBR/THR/DLL	SCC1
-0100H	IER/DLM	
-0200H	IIR/FCR	
-0300H	LCR	
-0400H	MCR	
-0500H	LSR	
-0600H	MSR	
-0700H	SCR	
<b>BC03-0000H</b>	LPD	LPT
-0100H	LPS	
-0200H	LPC	
-0300H	-	

シリアル・コントローラへの入力クロックは、16MHzとなっています。また、各レジスタの機能については、TL16C552Aのマニュアルを参照ください。

#### 5.4.5. タイマ(Timer)

タイマはNEC製のuPD71054を使用しています。uPD71054はIntel製のi8254と互換であり、3つのタイマカウンタを持っています。これらのタイマにより、各種の制御を行なっています。

uPD71054の各レジスタは、表の通りに割り当てられています。

論理アドレス	レジスタ	補足
<b>BC04-0000H</b>	PCNT0	タイマ0
-0100H	PCNT1	タイマ1
-0200H	PCNT2	タイマ2
-0300H	PCNTL	制御

各レジスタの機能については、uPD71054(i8254)のマニュアルを参照ください。

タイマは以下のように使用されます。

タイマ	クロック	モード	用途	
0	2MHz	2	タイマ割り込み0	モニタで使用
1	2MHz	2	タイマ割り込み1	ユーザで使用可
2	2MHz	2	DRAMリフレッシュ	



#### 5.4.6. 割り込みコントローラ(PIC)

PIC は、主に割り込み関係の制御を行ないます。レジスタ割り付けは下表の通りです。

論理アドレス	レジスタ	データバス								
		D7	D6	D5	D4	D3	D2	D1	D0	
<b>BC07-0000H</b>	<b>PIC INT0M</b>	IM07	IM06	IM05	IM04	IM03	IM02	IM01	IM00	
-0100H	<b>PIC INT1M</b>	IM17	IM16	IM15	IM14	IM13	IM12	IM11	IM10	
-0200H	<b>PIC INTR</b>	IR7	IR6	IR5	IR4	IR3	IR2	IR1	IR0	
-0300H	<b>PIC INTEN</b>	0	0	0	0	TOV EN	0	INT EN	NMIE N	

INT0M,INT1M レジスタはそれぞれ INT0,INT1 に入力する割り込みをマスクします。IM0x, IM1x のビットが”1”の時にイネーブルとなり、複数ビットを選択した場合にはそれぞれの OR で割り込みがアクティブとなります。

INTR レジスタは割り込みステータスで、割り込み要求がある場合に”1”が読み出せます。これはマスク状態に関係ありません。またエッジ割り込み要求の解除（クリア）には、このレジスタの対応ビットに”1”を書き込みます。

IM0[0..7],IM1[0..7].IR[0..7]の各ビットに割り付けられている割り込み要因は以下の通りです。

IM0,IM1,IR	割り込み要因	要求レベル
0	タイマ0（モード2）	エッジ（立ち上がり）
1	シリアル0	レベル（Low）
2	ホストPC（PCI通信）	レベル（Low）
3	タイムオーバー	レベル（Low）
4	タイマ1（モード2）	エッジ（立ち上がり）
5	シリアル1	レベル（Low）
6	パラレル（プリンタ）	レベル（Low）
7	未使用（0固定）	-

INTEN レジスタは、各割り込み種別のイネーブルを制御します。

NMIEN: ノンマスカブル割り込み（NMI）のマスクを設定します。このビットでマスクすることによって、 NMI をハード的に禁止することができます。この時、 NMI 端子の状態は High レベルとなります。

NMIEN	NMI
0	マスクする (リセット値)
1	マスクしない

INTEN: ボード上で使用している外部割り込み（INT0～3）のマスクを設定します。このビットでマスクすることによって、 INT0～3 をハード的に禁止することができます。この時、 INTx 端子の状態は High レベルとなります。

INTEN	INT0～5
0	マスクする (リセット値)
1	マスクしない

**TOVEN:** タイムオーバー機能の使用を設定します。タイムオーバーは、ローカル・バスのアクセス時に適用され、バスサイクルが約 8 μ秒間続いた場合、強制的にそのサイクルを終了させます。

TOVEN	タイムオーバー
0	使用しない (リセット値)
1	使用する

#### 5.4.7. PCI コントローラ

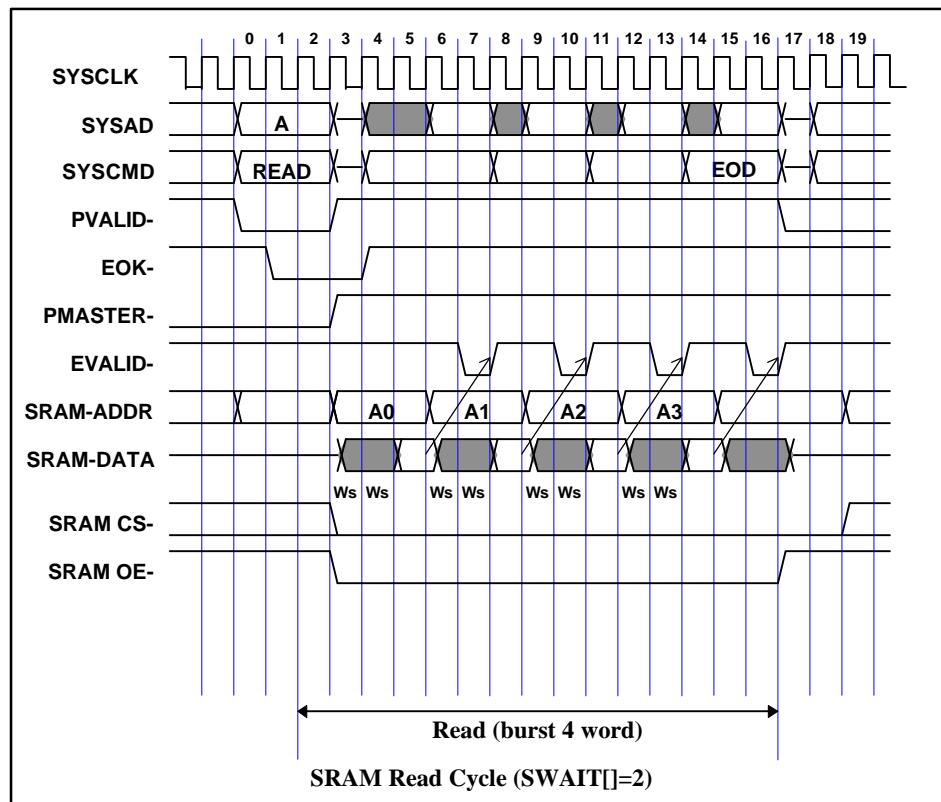
PCI バス通信のために、PLX Technorogy 社製の PCI9060ES を使用しています。PCI9060ES は、エンディアン制御ピンを使用していますので、内部レジスタはビッグ / リトル両エンディアンに対応しています。

## 6. バスサイクル・タイミング

RTE-VR4310-PC では、SRAM, DRAM, ROM, I/O などの各デバイスに合わせてバス・サイクルを制御しています。ここでは、各アクセス・サイクルのタイミングについて説明します。

### 6.1. SRAM アクセス

SRAM リードサイクルは、SRAMC-SWAIT レジスタの設定により 0~3 ウェイトのアクセスとなります（「5.4.1SRAM コントローラ (SRAMC)」参照）。ノーウェイト・アクセスが可能となるのは、バスクロック (SYSCLK) が 40MHz 以下の場合です。



クロックの番号に添って状態を説明します。

0: CPU の外部リードサイクルの始まりです。通常 EOK-はインアクティブです。

1~3: EOK-がアクティブになります。

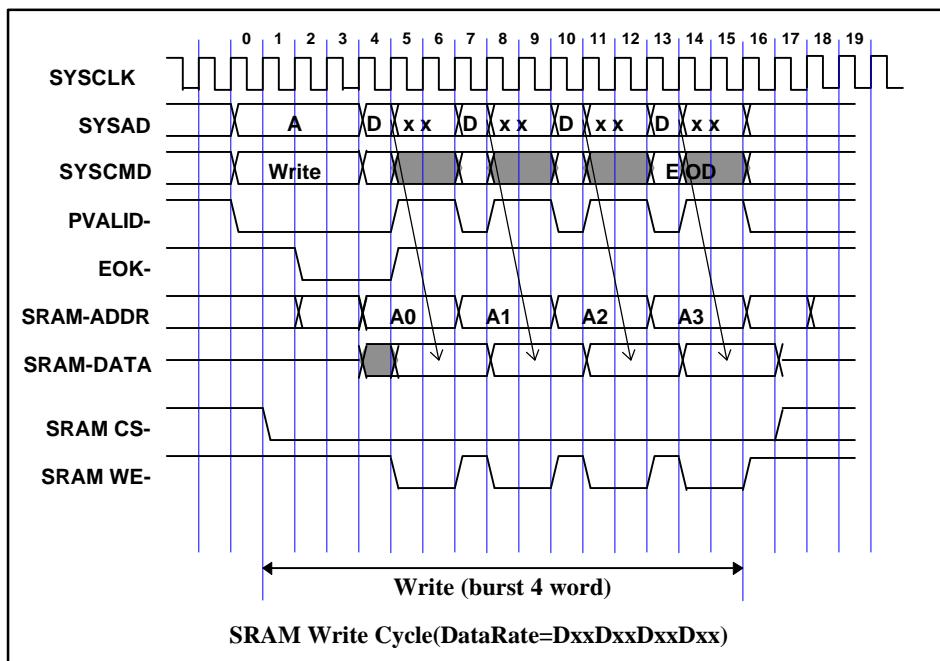
3~4: SWAIT で設定したウェイトサイクル（この例では 2 ウェイト）が挿入されます。

5: SRAM のデータ確定となります。次のサイクルで SRAM アドレスを変化させます（サブアドレス）。

5~6: データをラッチホールドして 2 クロック後に EVALID-を返します。

7~15: 最後のデータ(EOD)となるまで、サイクルをくり返します。

SRAM のライトサイクルは、CPU の出力データレートに従います。SRAM の場合には、アドレス確定から 1 クロック後に EOK-を返してサイクルをスタートさせます。



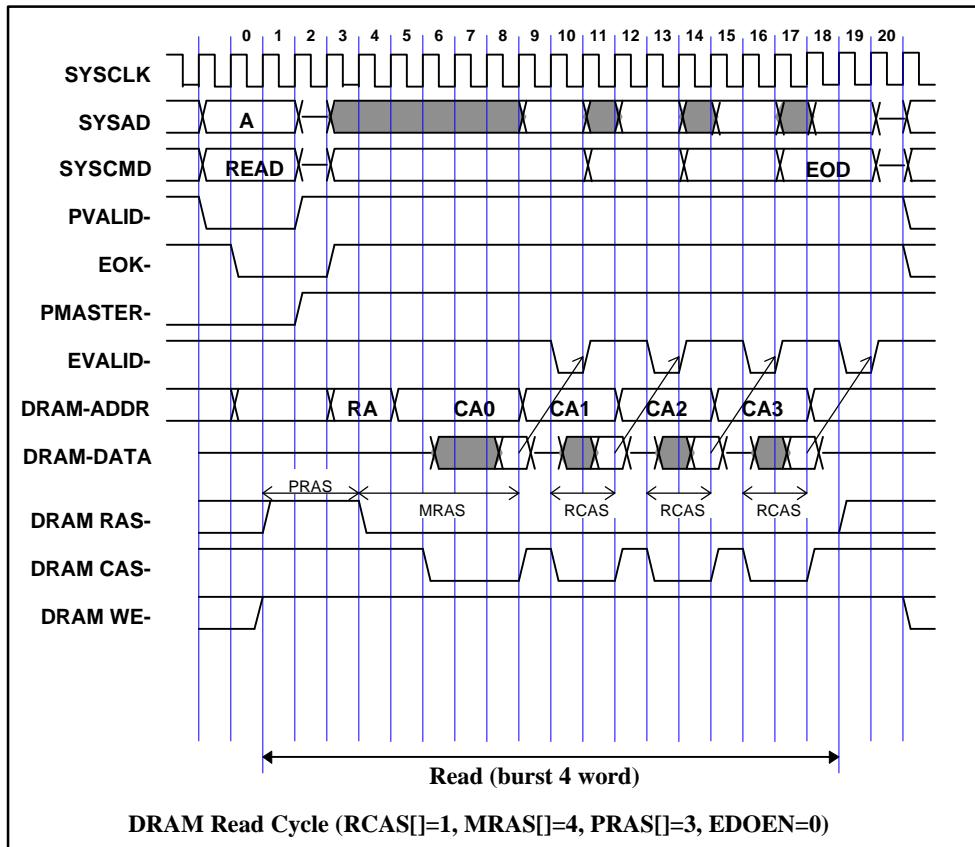
クロックの番号に添って状態を説明します。

- 0~1: 通常 EOK-はインアクティブです。
- 2~3: SRAM のライトサイクルが確定します。
- 4: 最初のデータ出力です。このデータをラッチホールドして、次のサイクルから EOK- をインアクティブとして SRAM の WE をアクティブとします。
- 5~6: SRAM ライトサイクルです (Dxx のデータレート固定ですので、ライトパルス幅は 2 クロックサイクルとなります。次のデータをラッチする前に WE-をインアクティブとして、次のサイクルで SRAM アドレスを変化させます (シーケンシャルアドレス) )
- 7~15: ライトサイクルをくり返します。
- 16: 最後のデータ(EOD)をライトした次のサイクルで RDRDY をアクティブとします。

## 6.2. DRAM アクセス

DRAM は、DRAM コントローラの設定により RAS/CAS 幅の制御ができます（「5.4.2DRAM コントローラ (DRAMC)」参照）。

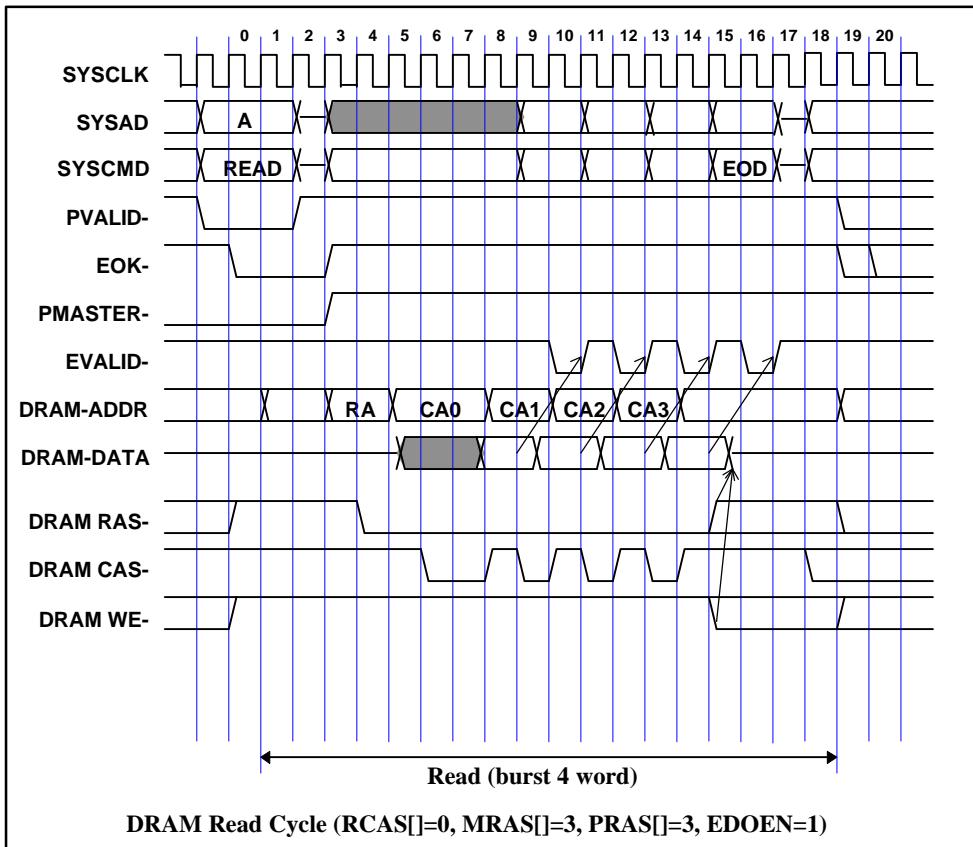
EDOEN=0 の場合 (FPM-DRAM) の DRAM リードサイクルは、下図の様になります。ただしこれはリフレッシュや前の DRAM アクセスの RAS プリチャージと重ならなかった場合で、最速のサイクルとなっています。



クロックの番号に添って状態を説明します。

- 0: EOK-をアクティブとします。
- 1: DRAM のリードサイクルの始まりです。
- 2~3: ロー(Row)アドレスの確定に伴って、RAS をアクティブにします。
- 4~5: カラム(Column)アドレスに切り替えて、RAS から 2 クロック後に CAS をアクティブとします。
- 7: 最初のリードデータ確定です。この時 DRAMC-MRAS+1 のサイクルまで、CAS はホールドされます。
- 8~9: データをラッチホールドして 2 クロック後に EVALID-を返します。
- 10~18: 最後のデータ(EOD)となるまで、サイクルをくり返します。HITEN=1 の場合には、RAS をアクティブのままホールドします。

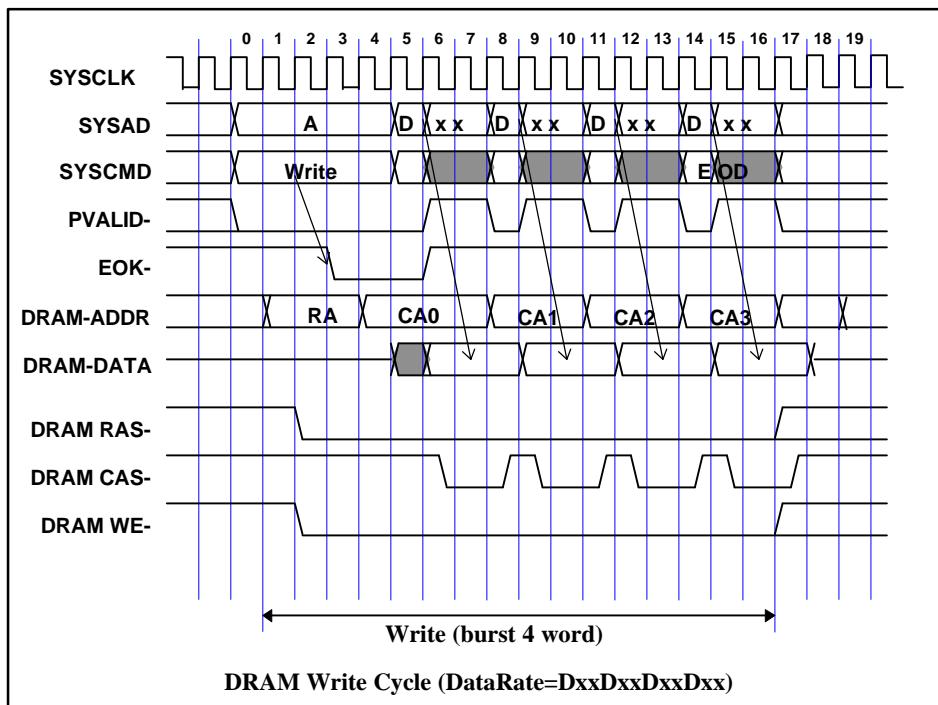
EDOEN=1 の場合は、リードデータの確定を 1 クロック遅らせて CAS プリチャージ・サイクルとなります。この場合、(SYSCLK の幅により)DRAMC-MRAS と DRAMC-RCAS の値を 1 つ少なくすることができます。EDO-DRAM では、RAS がインアクティブとなるまで、データをホールドするため、最後のサイクルでは WE-信号によりデータ出力をオフとする制御を行なっています。



クロックの番号に添って状態を説明します。

- 0: EOK-をアクティブとします。
- 1: DRAM のリードサイクルの始まりです。
- 2~3: 口ウ(Row)アドレスの確定に伴って、RAS をアクティブにします。
- 4~5: カラム(Column)アドレスに切り替えて、RAS から 2 クロック後に CAS をアクティブとします。
- 6~7: 最初のリードデータ確定です。この時 DRAMC-MRAS+1 のサイクルまで、CAS はホールドされます。
- 8~9: データをラッチホールドして 2 クロック後に VALIDIN を返します。この 2 クロックの期間で SYSADC バス用にデータのパリティ計算を行います。
- 10~15: 最後のデータ(EOD)となるまで、サイクルをくり返します。14 で HITEN=1 の場合には RAS をアクティブのままホールドし、WE-をアクティブにすることでデータをハイインピーダンスにします。

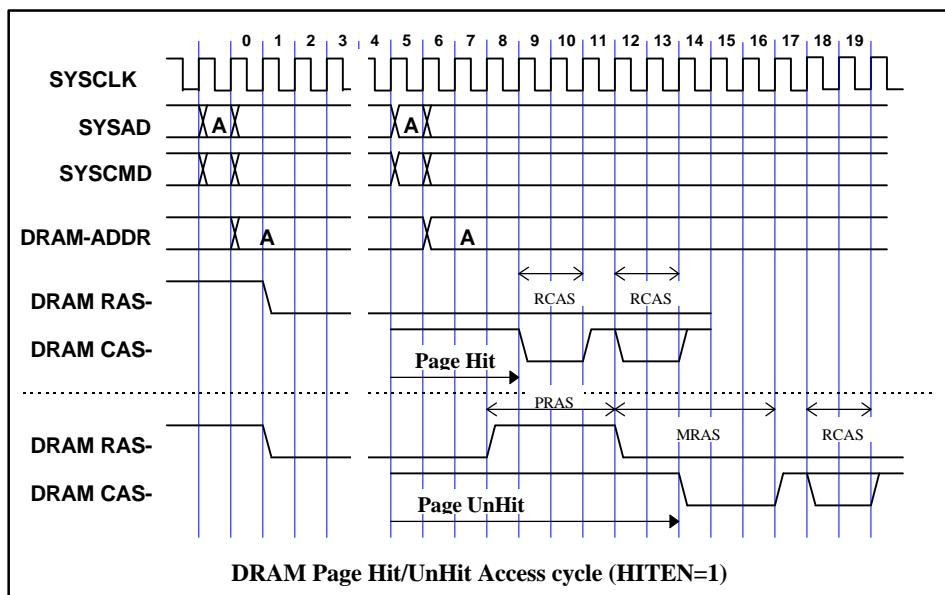
DRAM のライトサイクルは、CPU の出力データレートに従います。DRAM の場合には、RAS アクティブから 1 クロック後に EOK-を返してサイクルをスタートさせます。DRAM のライトにはアーリーライトサイクルを使用しています。



クロックの番号に添って状態を説明します。

- 0: 通常 EOK-はインアクティブとなっています。
- 1~2: DRAM のライトサイクルが確定し、ロウ(Row)アドレスと RAS がアクティブとなります。
- 3~4: RAS のアクティブから 1 クロック後に EOK-をアクティブとし、EOK-から 2 クロック後に CPU の外部ライトサイクルが始まります。
- 5: 最初のデータ出力です。このデータをラッチホールドして、次のサイクルから EOK-をインアクティブとします。
- 7~8: DRAM の CAS ライトサイクルです (Dxx のデータレート固定ですので、CAS ライトパルス幅は 2 クロックサイクルとなります)。次のデータをラッチする前に WE をインアクティブとして、次のサイクルで DRAM アドレスを変化させます (シーケンシャルアドレス)。
- 9~16: ライトサイクルをくり返します。15 で最後のデータをライトした後、2 クロック後に RAS と WE がインアクティブとなります。HITEN=1 の場合には、RAS はアクティブのままホールドされます。

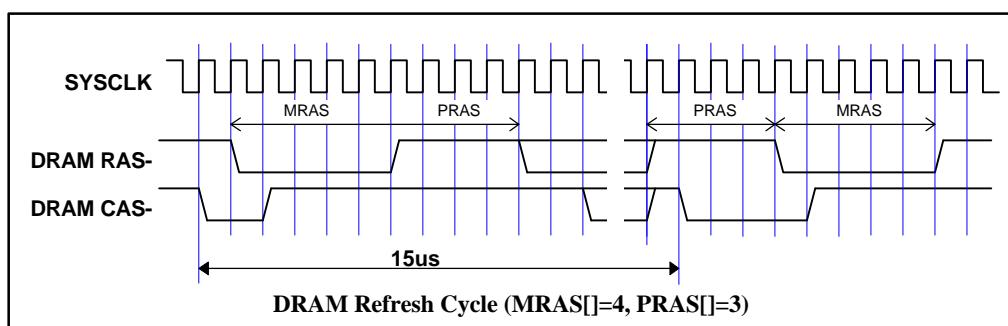
HITEN=1 の場合には、DRAM アクセス後に RAS をホールドして次のアクセスでロウ・アドレスが一致した場合（ヒット）は、すぐに CAS サイクルが実行されるためアクセス・サイクル数が減少します。ただし、これはリードサイクル時の効果であり、ライトサイクルでは一致した場合でも、アドレス比較のため EOK-が 1 クロック遅れてしまいます。また、不一致の場合（アンヒット）には、RAS プリチャージを行った後、通常の RAS/CAS サイクルでのアクセスとなるため、オーバーヘッドが多くなります。



クロックの番号に添って状態を説明します。

- 0~4: 通常の DRAM アクセスサイクルです。
- 5: 次の DRAM リードアクセスの開始です。
- 6: 前回のロウアドレスと比較します。
- 7~8: 一致した場合は、次のサイクルからリード CAS をアクティブとします。不一致の場合は、RAS をインアクティブとして RAS プリチャージを行います。
- 9: 一致した場合のリード CAS は、最初のサイクルからページモード・アクセスとなります。
- 14: 不一致の場合のリード CAS は、通常の RAS アクセス・サイクルとなります。

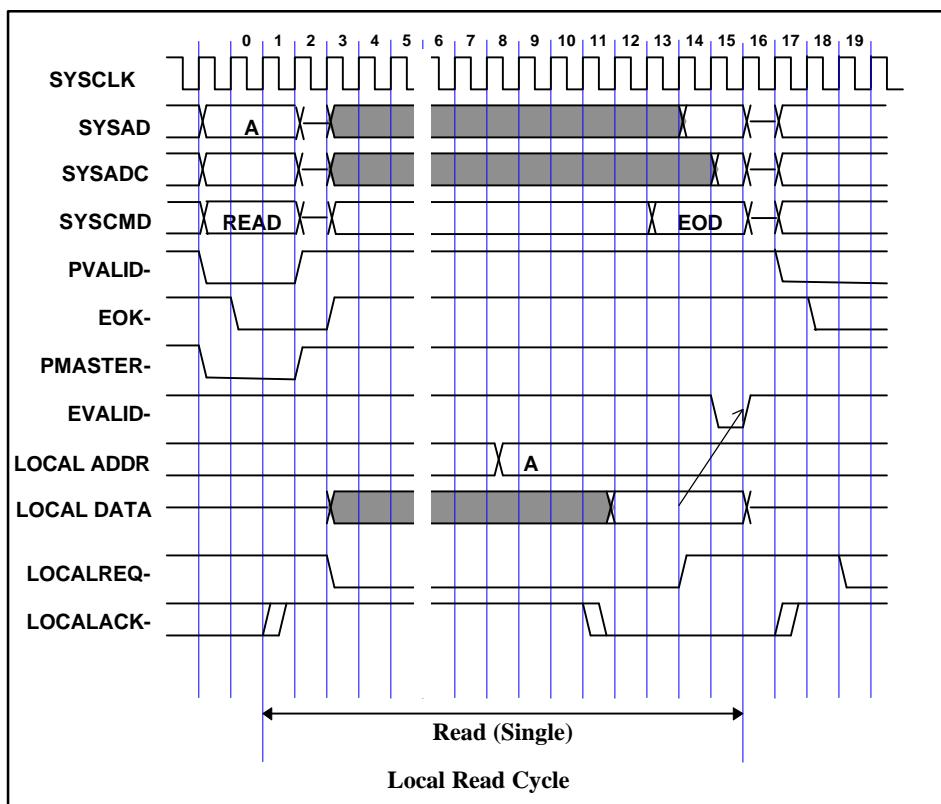
DRAM のリフレッシュは、CAS ピフォア RAS リフレッシュ・サイクルを（タイマ 2 の設定により）約 15 μ秒ごとに行ないます。



### 6.3. ローカルバス・アクセス

ローカルバスは 16 ビットのデータ幅を持つバスで、CPU とは非同期のクロック（32MHz 固定）からローカルバス・コントローラによってサイクルが生成されます。このローカルバスは、シングルアクセスのみが許されており（バーストサイクルは、リード時にはバスエラー、ライト時には無視されます）、CPU の制御回路からローカルバス・コントローラに対してリクエストが出され、コントローラから返されるアクノーレッジでバスの調停を行います。

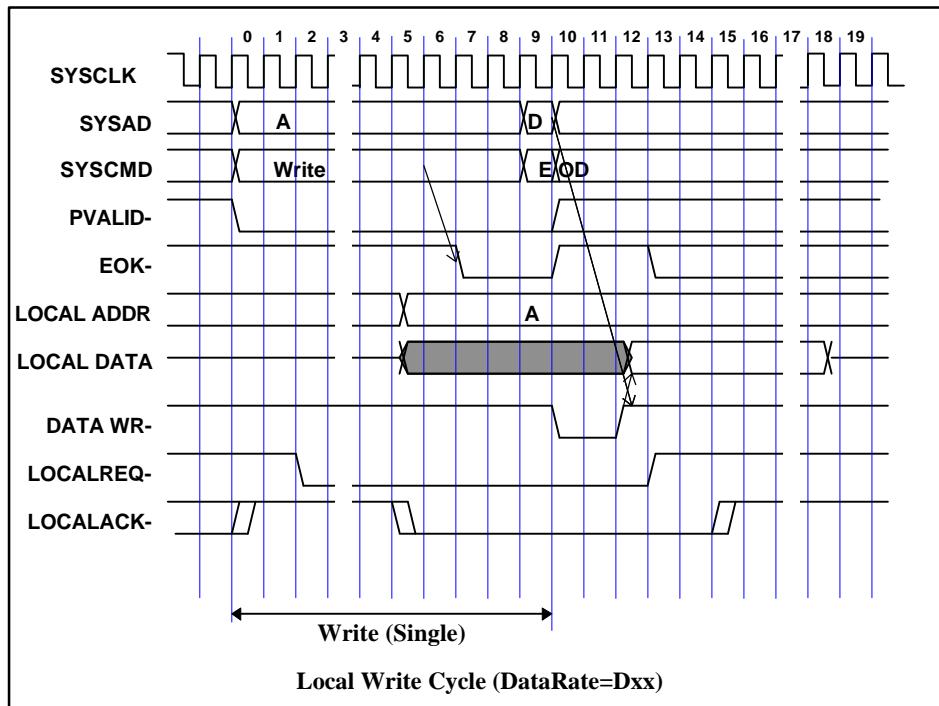
ローカルバスのリードサイクルでは、ローカルバス・コントローラにリクエストが出され、データリード終了後にコントローラ側からアクノーリッジが返されます。



クロックの番号に添って状態を説明します。

- 0~3: ローカルバスへのアクセス開始です。
- 4~: アクノーレッジ(LOCALACK)がインアクティブであることを確認して、ローカルバスに対してアクセス・リクエスト(LOCALREQ)を行います。
- 8: ローカルバス・コントローラはリクエストの発生を認識して、アドレスをラッチしリードサイクルを始めます。
- 11: ローカルバスのリードが終了すると、アクノーレッジを返されます。
- 13: アクノーリッジを 2 クロック同期してリクエストを解除するとともに、データをラッチホールドして 2 クロック後に EVALID-を返します。
- 17: ローカルバス・コントローラは、リクエストの解除を認識してアクノーレッジをインアクティブとします。

ローカルバスへのライトも同様に、リクエスト／アクノーレッジで調停して行われます。ただしローカルバス・コントローラから返されるアクノーレッジは、ライトの許可として扱われ、リクエストが解除された後のアクノーレッジ・サイクルでデータライトが実行されます。この時CPU側はローカルバス以外の次サイクルは実行することができます。

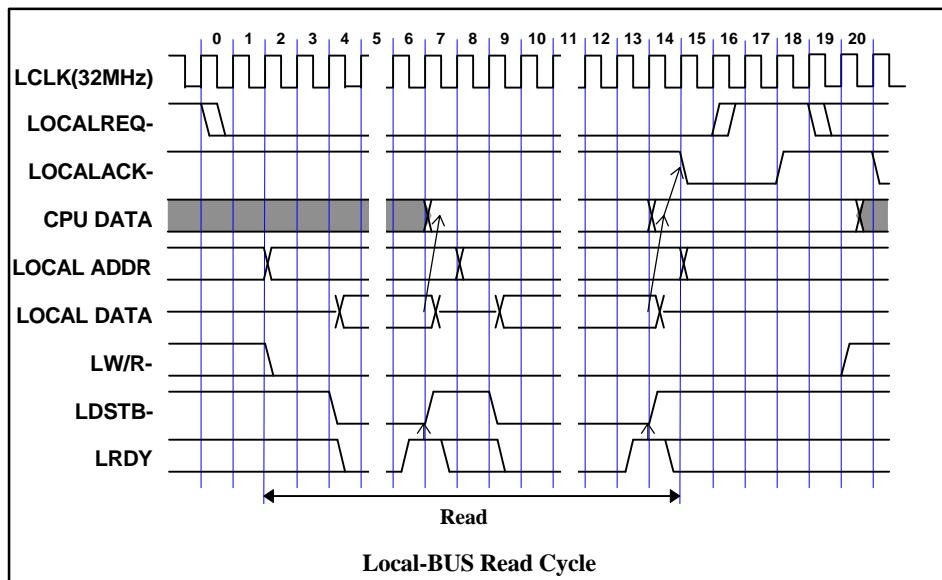


クロックの番号に添って状態を説明します。

- 0~1: ローカルバスへのライトアクセス開始です。
- 2: アクノーレッジ(LOCALACK)がインアクティブであることを確認して、ローカルバスに対してアクセス・リクエスト(LOCALREQ)を行います。
- 5: ローカルバス・コントローラはリクエストの発生を認識して、アドレスをラッチしアクノーレッジを返します。
- 7: アクノーリッジを2クロック同期してCPUにEOK-をアクティブとしライトサイクルを実行します。
- 12~13: ライトデータをローカルバス側にライト後リクエストをインアクティブにします。
- 14~15: ローカルバス・コントローラは、リクエストの解除を認識してからアクノーレッジをインアクティブとし、データライトを実行します。

ローカルバス・コントローラは、CPU からのシングルアクセスによってローカルバスをアクセスします。CPU からのアクセス・データ幅は最大 64 ビットであり、ローカルバスは 16 ビットであることから、バス・サイジングを行い最大 4 回のバスサイクルを生成します。

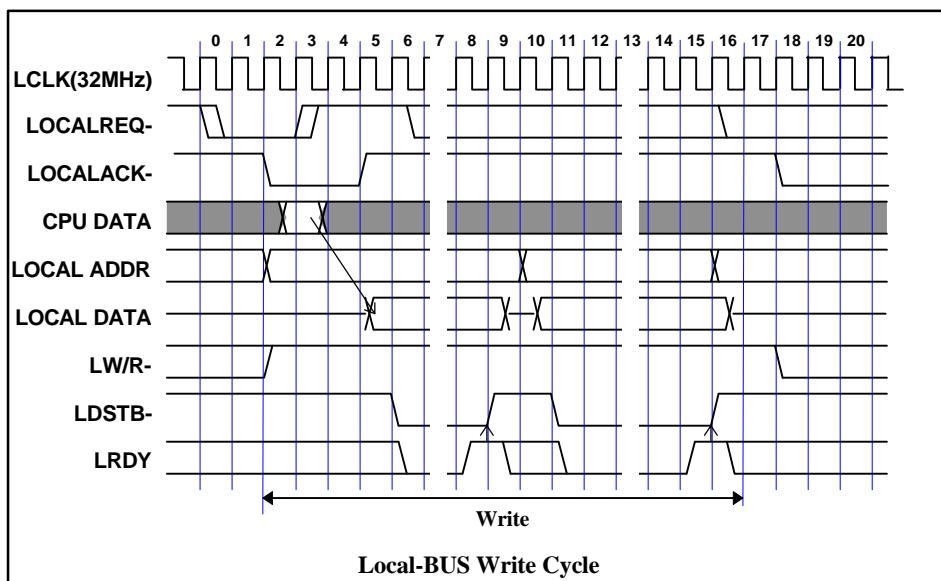
リードサイクルは、以下のようなタイミングとなります。



クロックの番号に添って状態を説明します。

- 0~1: ローカルバスへのリードリクエストの開始です。
- 2: ローカルバス・コントローラは、アドレスとリードステータス(LW/R-)をラッチします。
- 4: アドレスラッチから、2LCLK 後にデータストローブ(LDSTB)をアクティブにしてアクセスを開始します。
- 6~7: ローカルバス・レディー(LRDY)が返されると、データストローブをインアクティブとして CPU データをラッチします。
- 8~14: CPU のデータサイズが 16 ビット以下でない場合、次のアドレスに切り替えてリードサイクルをくり返します。
- 15: すべてのリードサイクルが終了して CPU データが確定すると、アクノーレッジを返します。

ライトサイクルのタイミングは、以下のようにになります。



クロックの番号に添って状態を説明します。

- 0~1: ローカルバスへのライトリクエストの開始です。
- 2: ローカルバス・コントローラは、アクノーレッジを返してアドレスとライトステータス(LW/R-)をラッチします。
- 3~5: リクエストがインアクティブとなってから 2LCLK 後にデータを確定させ、次のサイクルでデータストローブ(LDSTB)をアクティブにしてアクセスを開始します。
- 8~9: ローカルバス・レディ(LRDY)が返されると、データストローブをインアクティブとしてライトサイクルを終了します。
- 10~15: CPU のデータサイズが 16 ビット以下でない場合、次のアドレスとデータに切り替えてライトサイクルをくり返します。
- 17: すべてのライトサイクルを終了すると、次のリクエストを受け付けます。

ローカルバス・レディ(LRDY)は、ローカルバスに割り当てられた空間により異なります。下表にその一覧を示します。

ローカルバス 空間	レディ信号 ローカルバス・クロック 32MHz	備考
ROM	5 LCLK (約 150ns)	固定
I/O *1	7 LCLK (約 210ns)	固定
EXT-BUS	ERDY (EXT-BUS のレディ)	
PCI コントローラ	PCI コントローラのレディ信号	
割り当てなし *2	タイムオーバーレディ (約 8 μs)	固定

\*1 I/Oデバイスでは連続アクセスにおけるRD/RW信号のインアクティブ期間が決められているため、ローカルバス・コントローラではI/Oアクセス後に、7LCLK(約 210ns)のI/Oアクセス禁止時間をハードウェアによってサポートしています。これにより、ソフトウェアでI/Oアクセス後にウェイトサイクルを挿入する必要はありません。

\*2 タイムオーバー機能が有効時(「5.4.6割り込みコントローラ(PIC)」参照)

#### 6.4. メモリコントローラ・レジスタアクセス

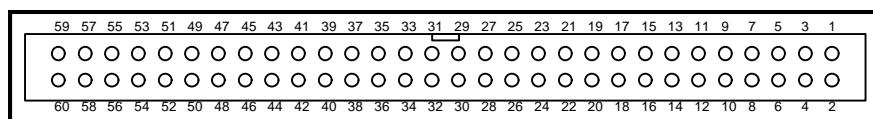
SRAMとDRAMのメモリコントローラ・レジスタのアクセスは、シングルサイクルのみ可能ですが(バーストサイクルは、リード時にはバスエラー、ライト時には無視されます)。この空間のアクセスは、SRAMアクセスの1ウェイト時のタイミングと同じです(「6.1SRAMアクセス」参照)。

## 7. EXT-BUS 仕様

EXT-BUS は、メモリや I/O などを拡張できるように用意されたバスです。JEXT コネクタには、本ボード内部のローカル・バスと接続されています。

### 7.1. コネクタの仕様

以下に JEXT コネクタの形状とピン配置を示します。



JEXT コネクタピン配置図

番号	信号名	番号	信号名	番号	信号名	番号	信号名
1	+5V	2	+5V	31	GND	32	GND
3	D0	4	D1	33	A8	34	A9
5	D2	6	D3	35	A10	36	A11
7	D4	8	D5	37	A12	38	A13
9	D6	10	D7	39	A14	40	A15
11	GND	12	GND	41	+5V	42	+5V
13	D8	14	D9	43	A16	44	A17
15	D10	16	D11	45	A18	46	A19
17	D12	18	D13	47	BHE-	48	GND
19	D14	20	D15	49	GND	50	RD-
21	+5V	22	+5V	51	WR-	52	RESET-
23	A0	24	A1	53	GND	54	GND
25	A2	26	A3	55	READY	56	INT-
27	A4	28	A5	57	GND	58	GND
29	A6	30	A7	59	LBCLK	60	GND

JEXT コネクタ信号配置

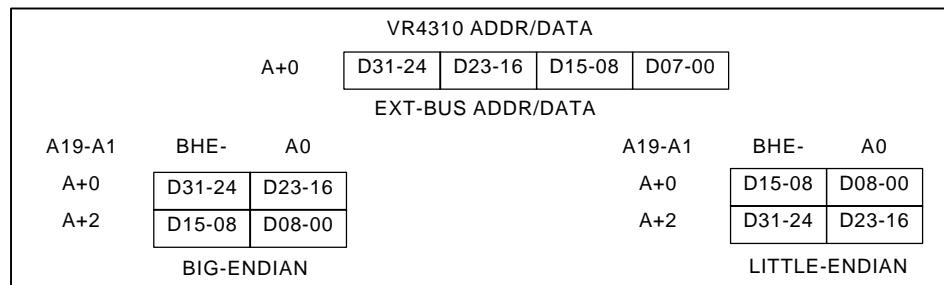
## 7.2. 信号

信号名	入出力	機能	備考
+5V	-	電源 +5V	
GND	-	グランド	
A[1..19]	出力	アドレス・バス信号	
A0	出力	バイトロー・イネーブル信号でこの信号がロー・レベルの時、D[0..7]が有効	
BHE-	出力	バイトハイ・イネーブル信号でこの信号がロー・レベルの時、D[8..15]が有効	
D[0..15]	入出力	データ・バス信号で CPU のデータ・バスをバッファリング	*3
RD-	出力	リード・サイクルのタイミング信号で EXT-BUS 空間のアクセス時のみアクティブ	
WR-	出力	ライト・サイクルのタイミング信号で EXT-BUS 空間のアクセス時のみアクティブ	
READY	入力	サイクルの終了を通知する正論理レディ信号で EXT-BUS 空間のみで有効。ボード上で 1K プルアップ	*1
INT-	入力	Low アクティブの割り込み要求信号でバッファ後 CPU の INT2-端子に接続。ボード上で 1K プルアップ	
RESET-	出力	Low アクティブのシステム・リセット信号	
LCLK	出力	バスクロック信号 (32MHz 固定)	*2

### EXT-BUS 信号

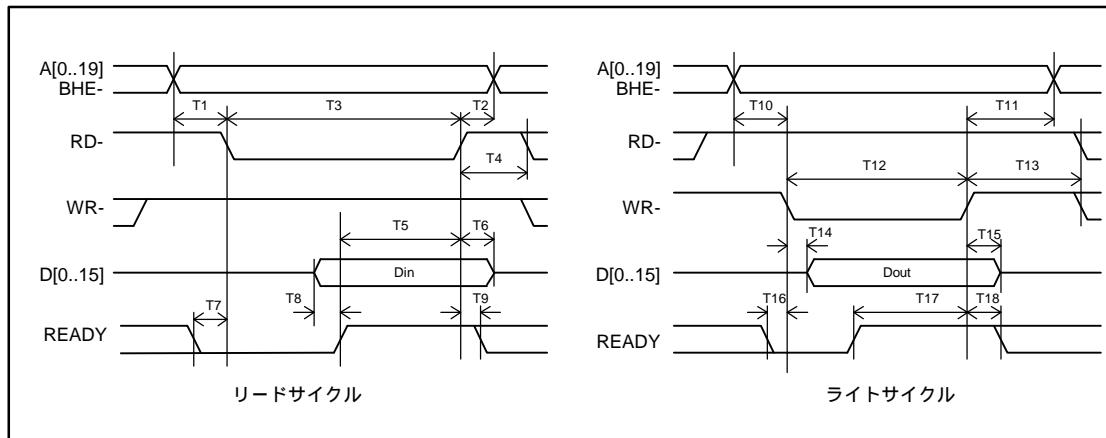
#### 注意事項

- \*1: 確実に CPU に READY を認識させるためには、RD-もしくは WR-がインアクティブになるまで READY をアクティブに保つことが必要になります。
- \*2: このクロック信号は他の RTE シリーズの評価ボードでは異なります。汎用性を考慮するならば、このクロック信号は使用しないで回路設計することをお薦めします。
- \*3: VR4310 のエンディアンのモードにより、アドレス / データバスの対応が以下の様に異なる点に注意してください。



### 7.3. EXT-BUS タイミング

以下に EXT-BUS のタイミングを示します。



EXT-BUS バス・サイクル

記号	内容	MIN(ns)	MAX(ns)
T1	RD アドレス セットアップ時間	0	
T2	RD アドレス ホールド時間	0	
T3	RD サイクル時間	50	
T4	RD サイクル間隔	20	
T5	RD データ セットアップ時間	15	
T6	RD データ ホールド時間	0	
T7	RD READY WAIT セットアップ時間	0	
T8	RD READY セットアップ時間	0	
T9	RD READY ホールド時間	0	
T10	WR アドレス セットアップ時間	0	
T11	WR アドレス ホールド時間	20	
T12	WR サイクル時間	50	
T13	WR サイクル間隔	20	
T14	WR データ 遅延時間	0	20
T15	WR データ ホールド時間	20	
T16	WR READY WAIT セットアップ時間	0	
T17	WR READY セットアップ時間	0	
T18	WR READY ホールド時間	0	

EXT-BUS AC スペック

## 8. ソフトウェア

### 8.1. 割込みプログラミング

モニタを使用してユーザプログラムで割込み処理を行なう場合について説明します。

#### 8.1.1. 割込みライブラリ

VR4310 では、一般例外割込みのベクタがまとめて割当てられており、モニタでも割込みを使用していることから、このベクタを共有することとなります。そこで、割込みライブラリ（添付のサンプルプログラム：Intvect.c, IntHdr.s/mip）が用意されています。

ライブラリは、ベクタをフックし各例外ごとにテーブル管理して、例外処理ルーチンを実行させます。以下に用意されている 8 つのライブラリ関数について説明します。

**int InitIrqVect(void) :**

割込みベクタのフックとテーブルの初期化を行ないます。正常終了の場合には 0、エラーが発生した場合には、負のエラー番号を返します。

**int TermIrqVect(void) :**

割込みベクタを元に戻します。正常終了の場合には 0、エラーが発生した場合には、負のエラー番号を返します。

**int SetIrqVect(int no, int func) :**

引数 no で指定される割込み処理テーブルに、関数 func を(int 型でキャストして)登録します。no には intvect.h ファイルに”INT\_xxx” の形式でマクロ化されているものを使用します。

また func に 0 を指定すると、前に設定されていた処理ルーチンを無効にできます。正常終了の場合には 0、エラーが発生した場合には、負のエラー番号を返します。

**int GetIrqVect(int no) :**

引数 no で指定される割込み処理テーブルに登録されている関数アドレスを返します。0 の場合には登録されている関数が無いことを示します。エラーが発生した場合には、負のエラー番号（奇数）を返します。

**void SetIPnBit(int IPn) :**

引数 IPn で指定される割込みを許可します(ステータスレジスタの該当ビットを 1 にします)。

**void ResIPnBit(int IPn)**

引数 IPn で指定される割込みを禁止します(ステータスレジスタの該当ビットを 0 にします)。

**void ei(void) :**

すべての割込みを許可します。

**void di(void) :**

NMI を除くすべての割込みを禁止します。

### 8.1.2. 割込みルーチン

ユーザの記述する割込み処理ルーチンは int 型の C 関数（\_\_interrupt 修飾子をつけない）を想定しています。また、割込み処理ルーチンを正しくデバッグするために、割込み処理ルーチンの先頭で該当する割込みを禁止、終了する前に割込みを許可してください。また、処理を完結する時には 0、モニタに戻す場合には 0 以外の値でリターンします。

### 8.1.3. サンプルプログラム

ユーザ制御ピンの IRQ(INT-3)を使用して割込み処理を行う場合のサンプルプログラムを以下に示します。

```
#include "intvect.h"

int main( void )
{
    .
    .
    /* --- can not Break & Step --- */
    di();                                /* disable all interrupt */
    InitIrqVect();                        /* save original vector code & set new vector */
    SetIrqVect( VECT_IP4, (int)IrqJusr ); /* set function vector table */
    SetIPnBit( SR_IP4 );                /* set IP4 bit */
    ei();                                /* enable all interrupt */
    /* --- --- */
    .
    .
    .
    /* process */
    .
    .
    /* --- can not Break & Step --- */
    di();                                /* disable all interrupt */
    ResIPnBit( SR_IP4 );                /* reset IP4 bit */
    SetIrqVect( VECT_IP4, 0 );           /* remove vector */
    TermIrqVect();                       /* restore original vector code */
    ei();                                /* enable all interrupt */
    /* --- --- */
    .
    .
    exit;
}

/*** irqJusr ***/
int IrqJusr( struct _irq_stack *istack )
{
    int to_monitor;                      /* flag */
    /* --- can not Break & Step --- */
    ResIPnBit( SR_IP4 );                /* Disable JUSR-IRQ interrupt */
    ei();                                /* Enable multiple interrupt */
    /* --- --- */
    .
    .
    /* interrupt operation */
    .
    /* --- can not Break & Step --- */
    di();                                /* Enable JUSR-IRQ interrupt */
    SetIPnBit( SR_IP4 );
    if( to_monitor != 0 ) {
        return 1;                         /* request monitor handler */
    }
    return 0;                            /* normal end */
}
```

## 8.2. ROM プログラミング

本ボードの ROM にユーザ自身でプログラムを作成する時に参考にしてください。

### 8.2.1. 初期化

リセット処理ルーチンではプロセッサ・レジスタの初期化を行った後、メモリアクセスコントローラに必要なウェイトを設定します。また DRAM を使用する場合には、タイマ 2 を設定してリフレッシュを行うようにします。

<プロセッサの内部初期化>	
( offset はリトルエンディアンでは 0、ビッグエンディアンでは 7 とする )	
[0xB8000000+offset].b <= 1	... SRAM 1 ウェイト
[0xB8000400+offset].b <= 1	... DRAM CAS 幅 2 クロック
[0xB8000500+offset].b <= 4	... DRAM RAS アクセス 5 クロック
[0xB8000600+offset].b <= 3	... DRAM RAS プリチャージ 4 クロック
[0xB8000700+offset].b <= 0	... EDO/HIT ディープル
[0xBC040300+offset].b <= 0xb4	... タイマ 2 モード 2 ( 約 15 μ秒周期に設定 )
[0xBC040200+offset].b <= 0x1f	... タイマ 2 下位カウント
[0xBC040200+offset].b <= 0x00	... タイマ 2 上位カウント

### 8.2.2. 割り込み

割り込みを使用する場合には周辺 I/O の初期化を行ったあと、割り込みマスクレジスタで必要な割り込みを割り付けます。また、INT/NMI 全体のイネーブルも行います。

タイマ割り込み 1 使用例 :

<プロセッサの割り込みディープル>	
( offset はリトル・エンディアンでは 0、ビッグ・エンディアンでは 7 とする )	
[0xBC040300+offset].b <= 0x74	... タイマ 1 モード 2 ( 10 ミリ秒周期に設定 )
[0xBC040200+offset].b <= 0x20	... タイマ 1 下位カウント
[0xBC040200+offset].b <= 0x4e	... タイマ 1 上位カウント
[0xBC070100+offset].b <= 0x10	... INT1M タイマ 1 イネーブル
[0xBC070200+offset].b <= 0x10	... INTR タイマ 1 割り込みクリア
[0xBC070300+offset].b <= 0x02	... INT イネーブル

<プロセッサの割り込みイネーブル>

【メモ】ROM エミュレータを接続して、その制御（ブレーク）に NMI を使用するにはエミュレータのモニタ・プログラムで NMI をイネーブルにしておくことが必要です。

### 8.2.3. ROM データ配置

ROM データの書き込みでは、ROM のアドレス・バンクとデータバスの対応を考慮する必要があります。標準で 272048 タイプ (128Kx16bit) では、スイッチにより 64K バイトごとにバンクが発生します。バンクを使用しない場合には、SW3 の 2,3 を共に ON とします。ROM のバンクについては、「3.4 DIVMODE スイッチ (SW1)」を参照してください。

また 16 ビット・データバスの ROM の書き込みの際には、エンディアンによりデータのバイト並びが ROM ライタの形式と異なる場合があります（たとえば、プログラム・コードはビッグ・エンディアンで ROM ライタでのサポートがリトル・エンディアン形式）。このような場合 16 ビットデータの上位 / 下位 8 ビットのスワップをサポートしている ROM ライタで、その機能を使用してデータ書き込みを行います。

**【メモ】**ROM エミュレータを使用する場合には、ROM のバンクを禁止しないとエミュレータのモニタ・プログラムが正常に動作できない場合があります。

## 9. APPENDIX.A MULTI モニタ

MULTI用のモニタ ROM を使用して、ホスト PC の MULTI デバッガと接続して使用する場合の設置方法と使用上の注意事項について説明します。

### 9.1. RTE for Win32 のインストール

MULTI デバッガを使用する場合は、ホスト PC に通信用のソフトウェア (RTE for Win32) をインストールする必要があります。このソフトウェアのインストールとテストの方法については、添付の「RTE for Win32 インストール・マニュアル」を参照してください。

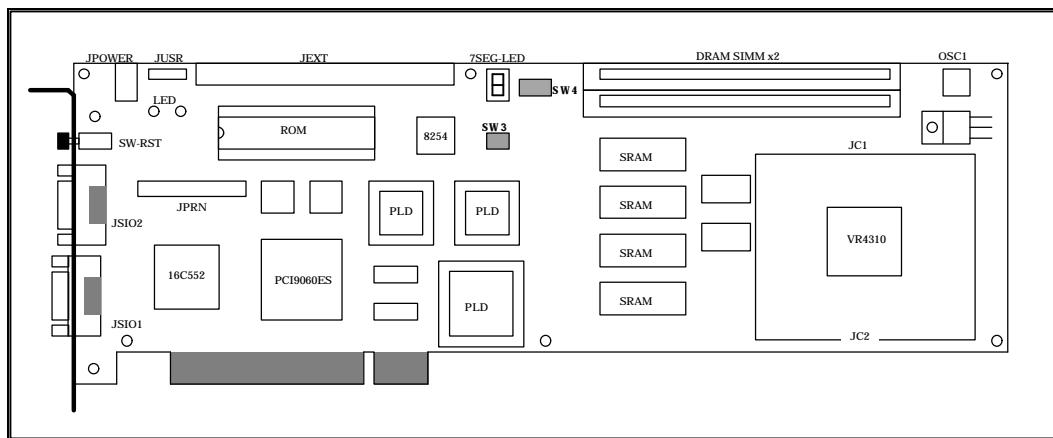
また、MULTI デバッガ本体については、デバッガに付属のマニュアルを参照して、インストールしてください。

### 9.2. ボードの設置

ボード上のスイッチの設定等について説明します。

#### 9.2.1. スイッチの設定

ボードには、いくつかの設定用ディップ・スイッチが設けられています。スイッチの位置を図に示します。



ボードのスイッチ

SW3 の-1 と-4 で、モニタの種類を選択します。

SW3 1	32/64 ピット動作(MD32/64)	備考 (*は出荷時の設定)
ON	0: 64 ピットモード	
OFF	1: 32 ピットモード	*

SW3 4	エンディアンの指定	備考 (*は出荷時の設定)
ON	0: リトル・エンディアン	
OFF	1: ビッグ・エンディアン	*

SW3 の-2 と-3 は、常に OFF にして下さい。

SW3		ROM アドレス		バンク選択 (*は出荷時の設定)
2	3	ROM-A15	ROM-A16	
OFF	OFF	ENDIAN	MD32/64	フルバンク *

SW4 は、汎用の入力ポートのスイッチです。実装されている ROM の MULTI モニタでは、RS-232C のボーレートやプロファイラのタイマ周期の設定などに使用しています。

SW4		ボーレート(JSIO1)	備考 (*は出荷時の設定)
1	2		
ON	ON	0: 115200 bps	
OFF	ON	1: 38400 bps	
ON	OFF	2: 19200 bps	
OFF	OFF	3: 9600 bps	*

**【メモ】** 他の通信設定は、データ 8 ビット、parity なし、ストップ 1 ビット固定です。

SW4		MULTI プロファイラ周期 (タイマ 0)	備考 (*は出荷時の設定)
3	4		
ON	ON	0: プロファイラを使用しない	タイマ割り込みなし
OFF	ON	1: 200 Hz 5.0 ms	
ON	OFF	2: 100 Hz 10.0 ms	
OFF	OFF	3: 60 Hz 16.67 ms	*

SW4 の 5 ~ 8 は、MULTI モニタでは使用していません（常時 OFF）。

### 9.2.2. ホスト PC とボードの接続

「4 ホスト PC との接続」を参照して、シリアルまたは、PCI バスのどちらかでホスト PC と接続してください。

### 9.3. MULTI モニタ

#### 9.3.1. モニタ・ワーク RAM

モニタでは、SRAM の先頭から 10000H ( 64KB ) をワーク用の RAM として使用 ( 予約 ) しています。すなわち、論理アドレス 8000-0000H ~ 8000-FFFFH , A000-0000H ~ A000-FFFFH は ユーザ・プログラムでは使用できません。また、これらのイメージ領域も同様です。

#### 9.3.2. 割り込み

ユーザ・プログラムで割り込みを使用する場合、「8.1 割込みプログラミング」を参照してください。

#### 9.3.3. 強制ブレーク用の割り込み

モニタでは強制ブレークや通信用の割込みとして、INT0 割り込みを使用しています。

#### 9.3.4. スタックの設定

モニタのスタック・ポインタの初期値は、8007-FFFCH ( SRAM の最上位 ) に設定されています。この値は、MULTI の環境下で、\_INIT\_SP コマンドで変更することもできます。

#### 9.3.5. 特殊命令

モニタでは、以下の命令をシングルステップ、ブレークポイント及び、システムコールの機能で使用しています。

BREAK 命令(Breakpoint)

この命令は、ユーザプログラムで使用することはできません。

### 9.4. RTE コマンド

MULTI デバッガでモニタと MIDAS サーバ ( RTESERV ) を接続すると TARGET ウィンドウが開かれ、ここから RTE コマンドを発行することができます。下表に RTE コマンドの一覧を示します。

コマンド名	内容
HELP, ?	ヘルプ表示
INIT	イニシャライズ
VER	バージョン表示
CACHEFLUSH	キャッシュのフラッシュ
SHOWTLB	TLB 内容表示
IOREAD	I/O リード ( サイズ指定 )
IOWRITE	I/O ライト ( サイズ指定 )

RTE コマンド一覧

各コマンドには、パラメータを必要とするものがあります。アドレスやデータなど、数値のパラメータは、全て 16 進数とみなされます。以下の数値指定は誤りです。

0x1234 1234H \$1234

#### 9.4.1. HELP(?)

<書式> HELP [コマンド名]

HELP は、RTE コマンドの一覧や書式を表示します。また、”HELP” と入力するかわりに”?” としても同様です。コマンド名を省略すると、使用できるコマンド一覧を表示します。

<例> HELP INIT

INIT コマンドのヘルプを表示します。

#### 9.4.2. INIT

<書式> INIT

INIT は、RTE 環境の初期化を行ないます。通常、このコマンドは使用しないでください。

#### 9.4.3. VER

<書式> VER

VER は、RTE 環境のバージョンを表示します。

#### 9.4.4. CACHEFLUSH

<書式> CACHEFLUSH

CACHEFLUSH は、CPU のキャッシュ内容をフラッシュします。

#### 9.4.5. SHOWTLB

<書式> SHOWTLB

SHOWTLB は、CPU の TLB 内容の一覧を表示します。

#### 9.4.6. IOREAD

<書式> IOREAD [BYTE | SHORT | LONG] [アドレス]

IOREAD コマンドは、指定されたサイズでアドレスのメモリをリードし、そのデータを表示します。サイズは BYTE, SHORT, LONG で、8, 16, 32 ビットを指定します。このコマンドは、メモリマップド I/O のアクセスに使用します。

<例> IOREAD BYTE BC000100

BC000100: 1A

#### 9.4.7. IOWRITE

<書式> IOWRITE [BYTE | SHORT | LONG] [データ] [アドレス]

IOWRITE コマンドは、指定されたサイズでアドレスのメモリにデータを書き込みます。サイズは BYTE, SHORT, LONG で、8, 16, 32 ビットを指定します。このコマンドは、メモリマップド I/O のアクセスに使用します。

<例> IOWRITE SHORT 30F0 BC00F000

## 10. APPENDIX.B PARTNERI モニタ

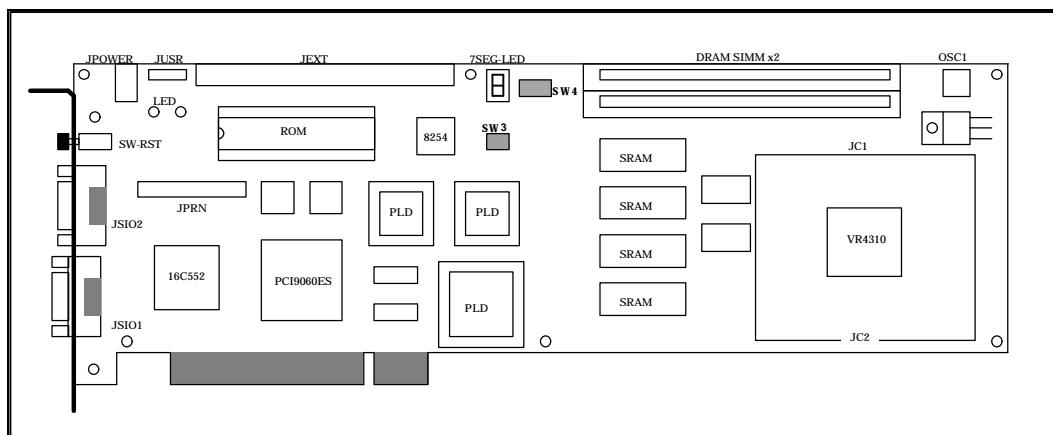
PARTNER 用のモニタ ROM を使用して、ホスト PC の PARTNER デバッガと接続して使用する場合の設置方法と使用上の注意事項について説明します。

### 10.1. ボードの設置

ボード上のスイッチの設定等について説明します。

#### 10.1.1. スイッチの設定

ボードには、いくつかの設定用ディップ・スイッチが設けられています。スイッチの位置を図に示します。



ボードのスイッチ

SW3-4 で ROM モニタの種類（エンディアンの指定のみ）を指定します。

SW3 4	モニタ	備考（*は出荷時の設定）
ON	0: 32bit のリトル・エンディアン用モニタ	
OFF	1: 32bit のビッグ・エンディアン用モニタ	*

SW3-4 以外は、常に以下の状態でご使用ください。

SW3 1	未使用	備考（*は出荷時の設定）
OFF	常にこの状態で使用（32 ビットモード）	*

SW3 2 3	ROM アドレス		バンク選択 (*は出荷時の設定)
	ROM-A15	ROM-A16	
OFF OFF	ENDIAN	MD32/64	フルバンク *

SW4 は、汎用の入力ポートのスイッチです。実装されている ROM モニタでは、RS-232C のボーレートの設定に使用しています。

SW4		ボーレート(JSIO1)	備考 (*は出荷時の設定)
1	2		
ON	ON	0: 115200 bps	
OFF	ON	1: 38400 bps	
ON	OFF	2: 19200 bps	
OFF	OFF	3: 9600 bps	*

**【メモ】** 他の通信設定は、データ 8 ビット、パリティなし、トップ 1 ビット固定です。

SW4			備考 (*は出荷時の設定)
3	4		
ON	ON	常にこの状態で使用	*
OFF	ON	設定禁止	
ON	OFF	設定禁止	
OFF	OFF	設定禁止	

SW4 の 5 ~ 8 は、モニタでは使用していません（常時 OFF）。

### 10.1.2. ホスト PC とボードの接続

「4 ホスト PC との接続」を参照して、シリアルまたは、PCI バスのどちらかでホスト PC と接続してください。

## 10.2. PARTNER モニタ

### 10.2.1. モニタ・ワーク RAM

モニタでは、SRAM の先頭から 10000H ( 64KB ) をワーク用の RAM として使用（予約）しています。すなわち、論理アドレス 8000-0000H ~ 8000-FFFFH , A000-0000H ~ A000-FFFFH は ユーザ・プログラム では使用できません。また、これらのイメージ領域も同様です。

### 10.2.2. 割り込み

ユーザ・プログラムで割り込みを使用する場合、「8.1 割込みプログラミング」を参照してください。

### 10.2.3. 強制ブレーク用の割り込み

モニタでは強制ブレークや通信用の割り込みとして、INT0 割り込みを使用しています。

### 10.2.4. スタックの設定

モニタのスタック・ポインタの初期値は、8007-FFFCH ( SRAM の最上位 ) に設定されています。

### 10.2.5. 特殊命令

モニタでは、以下の命令をシングルステップ、ブレークポイント及び、システムコールの機能で使用しています。

BREAK 命令(Breakpoint)

この命令は、ユーザプログラムで使用することはできません。

## 11. APPENDIX.C JC1,JC2 コネクタ

JC1 ピン	信号名	JC1 ピン	信号名
A1	SYSADC4(N.C)	B1	SYSADC5(N.C)
A2	SYSAD32	B2	SYSAD33
A3	SYSAD34	B3	SYSAD35
A4	SYSAD36	B4	SYSAD37
A5	SYSAD38	B5	SYSAD39
A6	GND	B6	GND
A7	SYSAD40	B7	SYSAD41
A8	SYSAD42	B8	SYSAD43
A9	SYSAD44	B9	SYSAD45
A10	SYSAD46	B10	SYSAD47
A11	+3.3V	B11	+3.3V
A12	SYSADC6(N.C)	B12	SYSADC7(N.C)
A13	SYSAD48	B13	SYSAD49
A14	SYSAD50	B14	SYSAD51
A15	SYSAD52	B15	SYSAD53
A16	SYSAD54	B16	SYSAD55
A17	GND	B17	GND
A18	SYSAD56	B18	SYSAD57
A19	SYSAD58	B19	SYSAD59
A20	SYSAD60	B20	SYSAD61
A21	SYSAD62	B21	SYSAD63
A22	SCVALID(N.C)	B22	SCMATCH(N.C)
A23	GND	B23	GND
A24	SYCLK0	B24	SYCLK1(MODECLK)
A25	GND	B25	GND
A26	/WRRDY(/EOK)	B26	/RDRDY(N.C)
A27	/VALIDOUT(PVALID)	B27	/VALIDIN(EVALID)
A28	/EXTREQ(EREQ)	B28	/RELEASE(PMASTER)
A29	+3.3V	B29	+3.3V
A30	/INT0	B30	/INT1
A31	/INT2	B31	/INT3
A32	/INT4	B32	/INT5(N.C)
A33	/RESET	B33	/NMI
A34	GND	B34	GND
A35	/SCCWE0(N.C)	B35	/SCCWE1(N.C)
A36	/SCDCE0(N.C)	B36	/SCDCE1(N.C)
A37	/SCTCE(N.C)	B37	/SCCLR(N.C)
A38	/SCTDE(N.C)	B38	(N.C)
A39	/SCTOE(N.C)	B39	/SCDOE(N.C)
A40	+5V	B40	+5V

JC1 ピン配置表

JC2 ピン	信号名	JC2 ピン	信号名
A1	(N.C.)	B1	(SCENABLE)
A2	(SCSIZE0)	B2	(SCSIZE1)
A3	(N.C.)	B3	(N.C.)
A4	+3.3V	B4	+3.3V
A5	SYSCMD0	B5	SYSCMD1
A6	SYSCMD2	B6	SYSCMD3
A7	SYSCMD4	B7	SYSCMD5(/LEN)
A8	SYSCMD6(/UEN)	B8	SYSCMD7(N.C.)
A9	SYSCMD8(N.C.)	B9	SYSCMDP
A10	GND	B10	GND
A11	SYSADC0(N.C.)	B11	SYSADC1(N.C.)
A12	SYSAD0	B12	SYSAD1
A13	SYSAD2	B13	SYSAD3
A14	SYSAD4	B14	SYSAD5
A15	SYSAD6	B15	SYSAD7
A16	GND	B16	GND
A17	SYSAD8	B17	SYSAD9
A18	SYSAD10	B18	SYSAD11
A19	SYSAD12	B19	SYSAD13
A20	SYSAD14	B20	SYSAD15
A21	+3.3V	B21	+3.3V
A22	SYSADC2(N.C.)	B22	SYSADC3(N.C.)
A23	SYSAD16	B23	SYSAD17
A24	SYSAD18	B24	SYSAD19
A25	SYSAD20	B25	SYSAD21
A26	SYSAD22	B26	SYSAD23
A27	GND	B27	GND
A28	SYSAD24	B28	SYSAD25
A29	SYSAD26	B29	SYSAD27
A30	SYSAD28	B30	SYSAD29
A31	SYSAD30	B31	SYSAD31
A32	GND	B32	GND
A33	SCWORD0(N.C.)	B33	SCWORD1(N.C.)
A34	SCLINE0(N.C.)	B34	SCLINE1(N.C.)
A35	SCLINE2(N.C.)	B35	SCLINE3(N.C.)
A36	SCLINE4(N.C.)	B36	SCLINE5(N.C.)
A37	SCLINE6(N.C.)	B37	SCLINE7(N.C.)
A38	+3.3V	B38	+3.3V
A39	SCLINE8(N.C.)	B39	SCLINE9(N.C.)
A40	SCLINE10(N.C.)	B40	SCLINE11(N.C.)
A41	SCLINE12(N.C.)	B41	SCLINE13(N.C.)
A42	SCLINE14(N.C.)	B42	SCLINE15(N.C.)
A43	GND	B43	GND
A44	/COLDRESET	B44	VCCOK(N.C.)
A45	MODEIN	B45	MODECLK(SYSCLK1)
A46	(N.C.)	B46	ENDIAN
A47	JTDI	B47	JTDO
A48	JTMS	B48	JTCK
A49	(N.C.)	B49	(N.C.)
A50	+5V	B50	+5V

JC2 ピン配置表

- Memo -

RTE-VR4310-PC ユーザーズ・マニュアル

M762MNL02

*Midas lab*