

# ***RTE-VR5500-CB(32)***

**ユーザース・マニュアル (Rev.2.00)**

***Midas lab***

## 改定履歴

日付			Rev	内容
Y	M	D		
01	5	19	1.00	初版
01	8	7	1.01	SW5-4 の記述を追記 ( 5.7スイッチ 5 ( SW5 ) )
01	11	6	2.00	ボードの改版(Board Rev.1.x -> Rev.2)に伴う変更 -> Borad Rev.1 と Rev.2 の相違点は下表を参照

## Borad Rev.1.x と Rev.2 の相違点

	Rev.1.x	Rev.2 以降
対象 CPU	ES1.1	ES2.0 以降
DRAMC の設定 (RCD, RP, RC, RRC, RAS, LAT, RSC)	必要	不要
ICE の接続時の制限	CPU は ES1.1 に限定	制限なし

## 目次

1.	はじめに.....	1
1.1.	マニュアル表記について.....	1
2.	特徴と機能.....	2
3.	主な特徴.....	3
4.	基本仕様.....	3
5.	ボードの構成.....	4
5.1.	リセット・スイッチ (SW_RESET) .....	4
5.2.	電源コネクタ (JPOWER) .....	4
5.3.	ジャンパ 1 (JP1) .....	4
5.4.	スイッチ 1 (SW1) .....	4
5.5.	スイッチ 2 (SW2) .....	6
5.6.	スイッチ 3 (SW3) .....	6
5.7.	スイッチ 5 (SW5) .....	7
5.8.	7SEG-LED, xxx-LED.....	7
5.9.	クロック・ソケット (OSC1) .....	7
5.10.	ROM ソケット.....	8
5.11.	シリアル・コネクタ (JSIO1,2) .....	8
5.12.	パラレル・コネクタ (JPRT) .....	8
5.13.	ROM エミュレータ用テストピン (JROM-EML) .....	9
5.14.	JGBUS コネクタ (JGBUS) .....	9
5.15.	ICE コネクタ(JDCU).....	10
5.16.	CPU コネクタ(J1,2,3).....	10
6.	ホスト PC との接続.....	12
6.1.	RS-232C 接続.....	12
7.	ハードウェア・リファレンス.....	13
7.1.	リセット.....	13
7.2.	割込み.....	13
7.3.	アドレス・マップ.....	14
7.3.1.	SRAM 空間 (0000-0000H ~ 07FF-FFFFH) .....	14
7.3.2.	DRAM 空間 (0800-0000H ~ 0FFF-FFFFH) .....	14
7.3.3.	GBUS 空間 (1000-0000H ~ 17FF-FFFFH).....	14
7.3.4.	I/O 空間 (1800-0000H ~ 1EFF-FFFFH) .....	15
7.3.5.	ROM 空間 (1F00-0000H ~ 1FFF-FFFFH) .....	15
7.4.	I/O 詳細.....	16
7.4.1.	SRAM コントローラ (SRAMC B800-6000[Read/Write]) .....	16
7.4.2.	DRAM コントローラ (DRAMC B800-5000[Read/Write]) .....	17
7.4.3.	DRAM 初期化ポート (DRAM_INIT B800-4000H[Write Only]) .....	19
7.4.4.	SW1 読み出しポート(SW1 B800-0000H [Read Only]).....	20
7.4.5.	SW2 読み出しポート(SW2 B800-1000H [Read Only]).....	20
7.4.6.	7セグメント LED 表示データ出力ポート(7SEG-LED B800-2000 [Write Only]) .....	20
7.4.7.	シリアル/パラレル I/O (SCC0/1, LPT B900-1000 ~ B900-4000[Read/Write]) .....	20

7.4.8.	タイマ (Timer B900-5000[Read/Write])	21
7.4.9.	割込みコントローラ (PIC B900-0000[Read/Write])	23
7.4.10.	DMA コントローラ	24
8.	バスサイクル・タイミング	28
8.1.	SRAM アクセス	28
8.2.	DRAM アクセス	30
8.3.	ローカルバス・アクセス	33
8.4.	所要クロック一覧	35
8.4.1.	SRAM のアクセス	35
8.4.2.	SDRAM のアクセス	35
8.4.3.	ROM のアクセス	35
8.4.4.	SYSTEM IO のアクセス	36
8.4.5.	GBUS のアクセス	36
9.	ROM 化プログラミング	37
9.1.	初期化	37
9.1.1.	SRAM の初期化	37
9.1.2.	ROM データ配置	37
10.	GBUS 個別仕様	38
10.1.	概要	38
10.2.	バス・サイクル	39
10.3.	チップセレクト	40
11.	APPEDIX.A MULTI モニタ	41
11.1.	ボードの設置	41
11.1.1.	RTE for Win32 のインストール	41
11.1.2.	SW1 の設定	41
11.2.	Multi モニタ	41
11.2.1.	起動時の 7Seg-LED	41
11.2.2.	モニタ・ワーク RAM	42
11.2.3.	割込み	42
11.2.4.	_INIT_SP の設定	42
11.2.5.	タイマ割込み	42
11.2.6.	ハードウェアの初期化	42
11.2.7.	特殊命令	42
11.3.	RTE コマンド	43
11.3.1.	HELP(?)	43
11.3.2.	INIT	43
11.3.3.	VER	43
11.3.4.	CACHEFLUSH	43
11.3.5.	SHOWTLB	44
11.3.6.	IOREAD	44
11.3.7.	IOWRITE	44
12.	APPEDIX.B PARTNER モニタ	45
12.1.	ボードの設置	45

12.1.1.	SWI の設定 .....	45
12.2.	PARTNER モニタ .....	45
12.2.1.	起動時の 7Seg-LED .....	45
12.2.2.	ROM モニタ・ワーク RAM .....	46
12.2.3.	割込み .....	46
12.2.4.	INIT_SP の設定 .....	46
12.2.5.	リモート接続 .....	46
12.2.6.	ハードウェアの初期化 .....	46
12.2.7.	特殊命令 .....	46
13.	APPEDIX.C 割込みプログラミング .....	47
13.1.	割込みライブラリ .....	47
13.1.1.	int InitIrqVect(void) .....	47
13.1.2.	int TermIrqVect(void) .....	47
13.1.3.	int SetIrqVect(int no, int func) .....	47
13.1.4.	int GetIrqVect(int no) .....	47
13.1.5.	void SetIPnBit(int IPn) .....	47
13.1.6.	void ResIPnBit(int IPn) .....	47
13.1.7.	void ei(void) .....	47
13.1.8.	void di(void) .....	47
13.2.	割込みルーチン .....	47
13.3.	サンプルプログラム .....	48
14.	APPEDIX.C GBUS 共通仕様 .....	49
14.1.	用語 .....	49
14.1.1.	CPU ボードとマザー・ボード .....	49
14.1.2.	バス・サイクル、マイクロ・サイクル .....	49
14.2.	信号 .....	49
14.3.	ピン配置 .....	53
14.4.	未使用端子の処理 .....	54
14.5.	GCS-[7:0]の割り付け .....	54
14.6.	バス・サイクル .....	55
14.6.1.	シングル・サイクル .....	55
14.6.2.	バースト・サイクル .....	55
14.6.3.	GWAITF- .....	56
14.6.4.	GBTERM- .....	57
14.7.	タイミング .....	58
14.7.1.	セットアップ・タイム .....	58
14.7.2.	ディレイ・タイム .....	58

## 1. はじめに

「RTE-VR5500-CB(32)」は NEC 製の RISC プロセッサ VR5500 の評価を目的とした CPU ボードで、()内の 32 は CPU のデータバス幅を示しています。

ボードは最高 300MHz で動作する VR5500 CPU とメモリ、シリアル/パラレルインターフェース、拡張用のバスコネクタ等で構成されます。メモリは、高速の PB-SRAM と大容量の SDRAM を標準で実装しています。

これらの機能を使用して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・プログラムの初期段階の開発など幅広くご利用頂けます。

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uggaとしてご使用になれます。ご使用になるデバugga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ugga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。

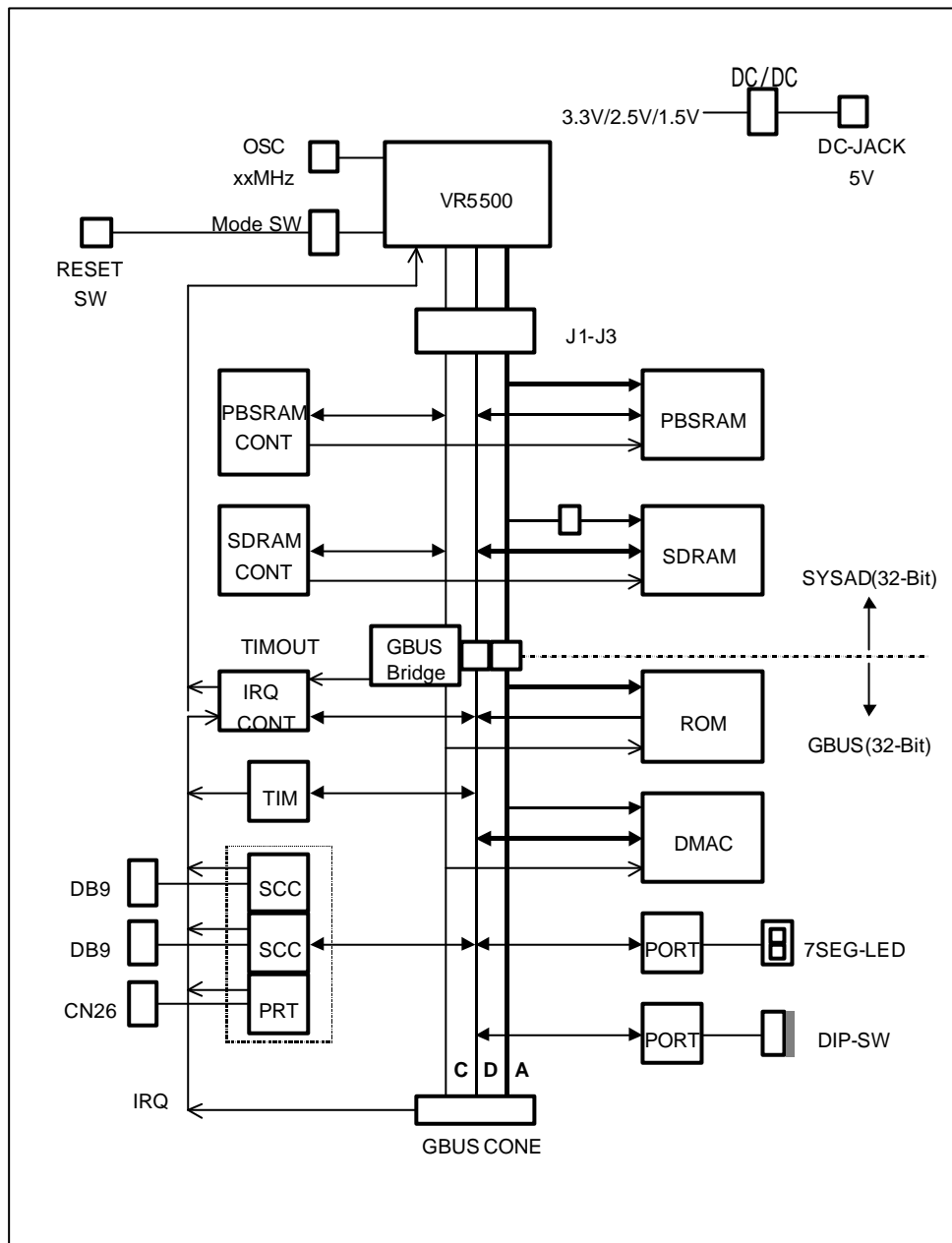
進数	表記規則	例
10 進数	数字のみを示します	“10”は 10 進数の“10”を示します
16 進数	数字の末尾に”H”を記します	“10H”は 10 進数の“16”を示します
2 進数	数字の末尾に”B”を記します	“10B”は 10 進数の“2”を示します

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-VR5500-CB の機能ブロックの概要を下図に示します。



ブロック図

### 3. 主な特徴

- GreenHills 社の MULTI と自社製 PARTNER 用のモニタ ROM を用意しています。
- 高級言語レベルでリアルタイム実行・評価が可能です。
- PB-SRAM を 1M-Byte、SDRAM は 64M-Byte を標準搭載しています。
- シリアル(2ch)、パラレル(IEEE1284)のインタフェースを用意しています。
- ROM エミュレータや N-Wire ツールが接続できます。

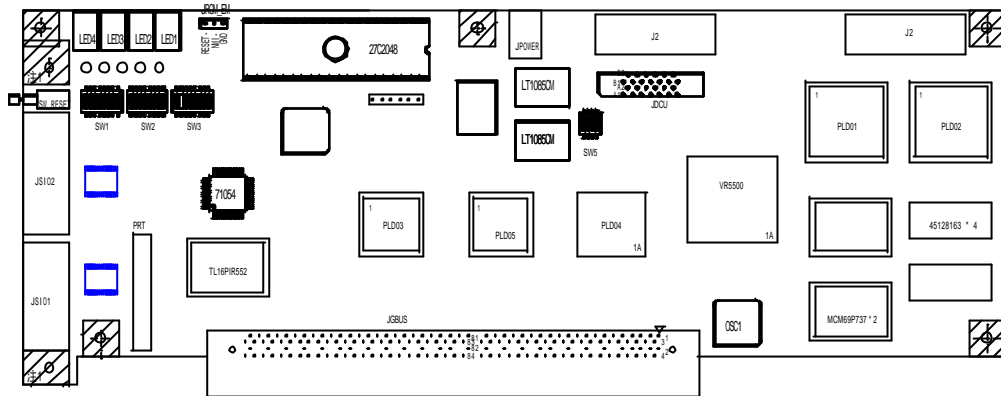
### 4. 基本仕様

プロセッサ	VR5500	
CPU クロック	280MHz (max300MHz)	
バスクロック	80MHz (max)	
バス幅	32-Bit	
電源	+5V , 2.5A (max)	
メモリ		
EPROM	256KB	128K × 16bit (40pin-DIP) × 1(max.512KB)
PB-SRAM	1MB	128K × 36bit × 2 (内 8-bit は Parity-bit)
SDRAM	64MB	2M × 16Bit × 4Bank × 4
I/O		
シリアル I F	NS16550 相当	DB9 コネクタ × 2ch
パラレル I F	IEEE1284 準拠	CON26 コネクタ
タイマ	uPD71054	分解能 500nS
IO ポート	LED(7seg)表示 / スイッチ入力	
その他		
CPU コネクタ	VR5500 全機能ピンを接続した測定用コネクタ	
GBUS コネクタ	RTE-CB 標準 32bit I/F(4GB,32bit バス,DMA 対応)	
リセット・スイッチ	Push 式	



## 5. ボードの構成

下図は RTE-VR5500-CB の主要な部品の物理的な配置です。ここでは、ボード上の主な構成部品について説明します。



ボードの概観図

### 5.1. リセット・スイッチ (SW\_RESET)

SW\_RESET は本ボード全体のリセット・スイッチです。このスイッチを押すと CPU を含む全ての回路がリセットされます。

### 5.2. 電源コネクタ (JPOWER)

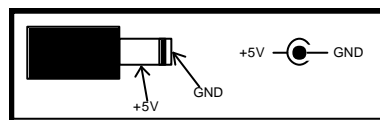
JPOWER コネクタに供給する電源は、以下の通りです。

電圧：5V

電流：2.5A(max)

適合コネクタ：Type A ( 5.5 )

極性：



電源コネクタへは、付属の電源以外を接続しないでください。  
また、JGBUS コネクタから電源を供給する場合は、JPOWER  
に電源を接続しないでください。

### 5.3. ジャンパ 1 (JP1)

JP1 は、出荷時の状態 ( 1-2 ショート ) でご使用ください。(JP1 は Board Rev.1 にはありません)

### 5.4. スイッチ 1 (SW1)

SW1 は汎用の入力ポートのスイッチです。設定状態は、入力ポートから読み出すことが可能です (「7.4.4

SW1 読み出しポート(SW1 B800-0000H [Read Only])」参照)。ポートからの読み出し時、スイッチは、OFF で 1、ON で 0 の値になります。モニタ ROM を使用する場合には、一部を除き、割り当て済みです。モニタ ROM での割り付けは、以下の各章を参照し、使用環境に合わせて設定してご使用ください。

Multi を使用する場合、「11.1.2 SW1 の設定」を参照ください。

PARTNER を使用する場合、「12.1.1 SW1 の設定」を参照ください。

## 5.5. スイッチ 2 (SW2)

SW2 は、本ボードの動作をハード的に切り替えるスイッチです。設定内容は入力ポートから読み出すことが可能です（「7.4.5 SW2 読み出しポート(SW2 B800-1000H [Read Only]）」を参照）。

番号	名称	出荷時の設定	機能
1	FBOOT	OFF	CS0 空間に割り付ける資源を設定します。(通常:OFF) OFF:CS0 の空間はボード上の UV-EPROM が割り付けられず。 ON : CS0 の空間は GBUS の GCS1-空間が割り付けられます。 (「10.3 チップセレクト」参照)。
2	GBUS_OLD	OFF	G-BUS のメモリ,IO マップ(GCS0-7)を設定します。 OFF: 本ボードのマップです。 ON : RTE-VR5432-CB と互換のマップです。
3	BCLK_LOW	OFF	常時、OFF に設定してください
4	GBCLK_LOW	OFF	常時、OFF に設定してください
5	WDTRATE0	ON	Write Data Rate を指定します。この設定に従って、VR5500 の Config レジスタの設定が必要です。 [WDTRATE1, WDTRATE0] : Rate [ OFF , OFF ] : DxxxDxxxDxxxDxxx... [ OFF , ON ] : DxxDxxDxxDxx... [ ON , OFF ] : DxDxDxDxDx... [ ON , ON ] : DDDD...
6	WDTRATE1	ON	
7	未使用	OFF	常時、OFF に設定してください。
8	未使用	OFF	常時、OFF に設定してください。

備考 : WDTRATE[1..0]の状態に対する VR5500 の Config レジスタの EP フィールドの設定値を以下に示します。キャッシュ領域へのアクセス前に必ず正しく設定してください。

WDTRATE1	WDTRATE0	ConfigReg[27..24]=EP	Rate
OFF	OFF	[1,0,0,0]	DxxxDxxxDxxxDxxxDxxxDxxxDxxxDxxx
OFF	ON	[0,1,1,0]	DxxDxxDxxDxxDxxDxxDxxDxx
ON	OFF	[0,0,1,1]	DxDxDxDxDxDxDxDxDxDx
ON	ON	[[0,0,0,0]	DDDDDDDD

## 5.6. スイッチ 3 (SW3)

SW3 は、ROM ソケットに搭載されている ROM の種類とバンクに関して設定します。

番号	信号名	出荷時の設定	機能
1	ROM_TYPE0	OFF	ROM の種類を設定します。 [ROM_TYPE1,ROM_TYPE0 ] [ OFF , OFF ] : モニタ ROM 使用時 [ OFF , ON ] : 27C4096 使用時 [ ON , OFF ] : 27C2048 使用時 [ ON , ON ] : 27C1024 使用時
2	ROM_TYPE1	OFF	
3	BANK_DIS	OFF	ROM をバンクに分けて使用するかどうかを設定します。 モニタ ROM 使用時は、OFF に設定してください。 OFF : バンクに分けて使用 ON : バンクに分けずに連続した領域として使用
4	未使用	OFF	常時、OFF に設定してください。
5	未使用	OFF	常時、OFF に設定してください。
6	未使用	OFF	常時、OFF に設定してください。
7	MON64	OFF	モニタの種類を指定します。 OFF : 32bit mode ON : 64bit mode
8	LITTLE	OFF	ENDIAN を指定します。 OFF : BIG ON : LITTLE

備考：ROM エミュレータを接続する場合は、BANK\_DIS を ON にして、ROM を連続して使用する設定でご使用ください。

### 5.7. スイッチ 5 (SW5)

SW5 の 1-3 は、CPU の DIVMODE モードを設定するスイッチです。OFF で 1、ON で 0 です。

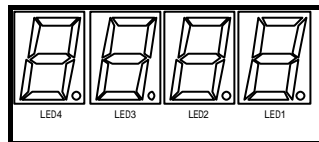
SW5			DivMode[2..0] Ratio	備考 (* は出荷時の設定)
3	2	1		
ON	ON	ON	Divide2	
ON	ON	OFF	Divide2.5	
ON	OFF	ON	Divide3	
ON	OFF	OFF	Divide3.5	*
OFF	ON	ON	Divide4	
OFF	ON	OFF	Divide4.5	
OFF	OFF	ON	Divide5	
OFF	OFF	OFF	Divide5.5	

SW5 の 4 は、SysMode[0] に接続されています。

番号	信号名	出荷時の設定	機能
4	SysMode[0]	OFF	OFF = 1, ON = 0 です。 OFF: Timer Int が有効 ON : Int[5]#が有効

### 5.8. 7SEG-LED, xxx-LED

LED は、各種の状態を示しています。表に内容を示します。4 つの 7SEG-LED は、起動時にモニタが使用しますが、その後、ユーザアプリケーションで自由に使用可能です。



名称	内容
POWER	ボードに電源が供給されている時に点灯
SRAM	PB-SRAM 空間へのアクセス時に点灯します。
DRAM	SDARM 空間へのアクセス時に点灯します。
LOCAL	LOCAL 空間へのアクセス時に点灯します。
GBUS	GBUS 空間へのアクセス時に点灯します。

ボード LED ステータス

### 5.9. クロック・ソケット (OSC1)

OSC1 ソケットには、CPU に供給するクロック用のオシレータを実装します。OSC1 の出力は、3.3V にレベル変換されて、そのままの周波数が CPU に入力されています。(バスクロックに相当します) オシレータは、DIP8 ピンタイプ (ハーフタイプ) のものを使用します。



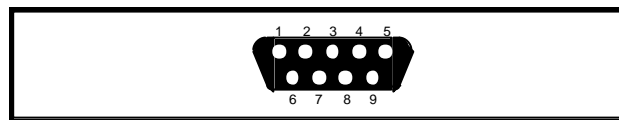
オシレータの足を切って実装する場合、足が短かすぎるとフレーム (外装) 部分がソケットの端子とショートしてしまいますのでご注意ください。

### 5.10. 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 が実装されています。変更する場合は、27C1024,27C2048,27C4096 タイプで、アクセス・タイムが 120ns 以下のものをご使用ください。ROM の種類を変更する場合や、ROM エミュレータを接続する場合は、SW3 の再設定が必要です。(「5.6 スイッチ 3 (SW3)」を参照)

### 5.11. 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 (TL16PIR552) 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、ホストと接続する場合の接続信号について、ホスト側が 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)。



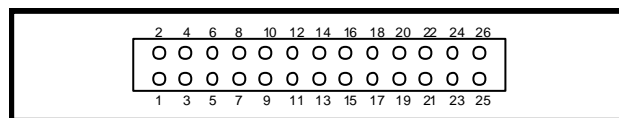
JSIO1,2 ピン配置図

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

### 5.12. 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 (TL16PIR552) 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

### 5.13. ROM エミュレータ用テストピン (JROM-EML)

JROM-EML は、ROM エミュレータを接続する際に使用するテストピンです。下記の制御信号が入力できます。表に信号名と機能を示します。

信号名	入出力	機能
RESET-(1)	入力	Lowレベル入力により、ボード全体がリセットされます。ROMエミュレータからのリセット要求信号を接続します。1K でプルアップされています。
NMI-(2)	入力	Lowレベル入力により、CPUにNMIが入ります。ROMエミュレータからのNMI要求信号を接続します。1K でプルアップされています。
GND(3)	- - -	GND。ROMエミュレータのGNDと接続します。

JROM\_EM 端子の機能

### 5.14. JGBUS コネクタ (JGBUS)

拡張用の 32Bit データ幅のバスコネクタです。詳細は「10 GBUS 個別仕様」、14 APPEDIX.C GBUS 共通仕様」を参照してください。

### 5.15. ICE コネクタ(JDCU)

ICE を接続するためのコネクタで、RTE-TP 系の ICE が接続できます。以下の信号が接続されています。

ピン番号	信号名	ピン番号	信号名
A1	TRCLK	B1	GND
A2	TRCDATA0	B2	GND
A3	TRCDATA1	B3	GND
A4	TRCDATA2	B4	GND
A5	TRCDATA3	B5	GND
A6	TRCEND	B6	GND
A7	DDI	B7	GND
A8	DCK	B8	GND
A9	DMS	B9	GND
A10	DDO	B10	GND
A11	/DRST	B11	NC.
A12	/RMode*/BkTgio*	B12	NC.
A13	NC.	B13	+3.3V

JDCU コネクタ信号

基板側のコネクタ : KEL 社 8830E-026-170S

### 5.16. CPU コネクタ(J1,2,3)

CPU コネクタの信号は、VR5500 と直結した信号です。ロジックアナライザ用の信号観測用のコネクタです。

ピン番号	信号名	ピン番号	信号名
1	NC.	2	NC.
3	GND	4	NC.
5	CPUCLK	6	IU.
7	SASAD15	8	SASAD31
9	SASAD14	10	SASAD30
11	SASAD13	12	SASAD29
13	SASAD12	14	SASAD28
15	SASAD11	16	SASAD27
17	SASAD10	18	SASAD26
19	SASAD9	20	SASAD25
21	SASAD8	22	SASAD24
23	SASAD7	24	SASAD23
25	SASAD6	26	SASAD22
27	SASAD5	28	SASAD21
29	SASAD4	30	SASAD20
31	SASAD3	32	SASAD19
33	SASAD2	34	SASAD18
35	SASAD1	36	SASAD17
37	SASAD0	38	SASAD16

J1 コネクタ信号

ピン番号	信号名	ピン番号	信号名
1	NC.	2	NC.
3	GND	4	NC.
5	/VALIDOUT	6	/VALIDIN
7	SYSCMD8	8	SYSADC7
9	SYSCMD7	10	SYSADC6
11	SYSCMD6	12	SYSADC5
13	SYSCMD5	14	SYSADC4
15	SYSCMD4	16	SYSADC3
17	SYSCMD3	18	SYSADC2
19	SYSCMD2	20	SYSADC1
21	SYSCMD1	22	SYSADCO
23	SYSCMD0	24	IU.
25	/SYSRESET	26	/INT2
27	/COLDRESET	28	/INT1
29	/RRDY	30	/INT0
31	/RDRDY	32	/CS_LBUS
33	/VALIDIN	34	/CS_GBUS
35	/VALIDOUT	36	/CS_DRAM
37	CPUCLK	38	/CS_SRAM

J3 コネクタ信号

NC：基板側で未接続です。

IU.：メーカーチェック用です。

基板側のコネクタ：AMP社 MICTOR38



## 6. ホスト PC との接続

### 6.1. RS-232C 接続

モニタ ROM を使用して、ホスト PC とシリアルで接続する際の手順は以下の通りです。

添付品の RS-232C ケーブルと、電源を用意してください。

ボード上のスイッチの設定と確認を行ってください。SW1 のポーレイトの設定は、必ず実施ください。(「11.1.2 SW1 の設定」、「12.1.1 SW1 の設定」の各項を参照ください。)

JSIO1 コネクタ(CH0)とホストマシンを RS-232C ケーブルで接続し、JPOWER コネクタへ電源を供給してください。ボードの POWER-LED が点灯し、モニタの起動を示す 7seg-LED の表示がなされることを確認してください。



**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 でデバッグを起動し、RS-232C 経由でコネクタします。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 (特にポーレート) の設定等に間違いがないかを確認してください。デバッグの起動方法や手順は各デバッグのマニュアルを参照ください。



**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。**

## 7. ハードウェア・リファレンス

ここでは、RTE-VR5500-CB ボードのハードウェア仕様について説明します。

### 7.1. 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。
- ・ **GBUS・リセット**：JGBUS コネクタからのリセット要求です。
- ・ **JROM\_リセット**：JROM\_EML コネクタの 1 番端子(RESET-)からの入力です (「5.13 ROM エミュレータ用テストピン (JROM-EML)」を参照)。
- ・ **DCU・リセット**：ICE からのリセット要求です。(「5.13 ROM エミュレータ用テストピン (JROM-EML)」参照)

### 7.2. 割り込み

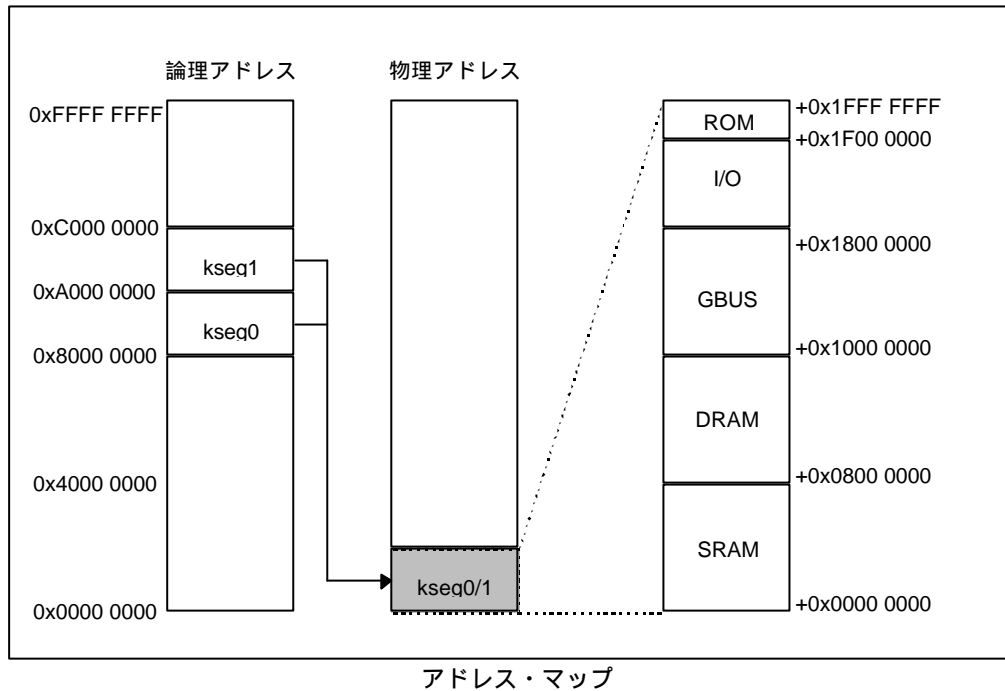
CPU 外部からの割り込み要因は、以下の通りです。

割り込み	要因	参照 (章)
NMI-	JROM_EML(NMI-)	5.13
INT0-	PIC INT0M	7.4.9
INT1-	PIC INT1M	7.4.9
INT2-	GBUS_INT1-	RTE-MB-A のマニュアルを参照。
INT3-	GBUS_INT2-	RTE-MB-A のマニュアルを参照。
INT4-	GBUS_INT3-	RTE-MB-A のマニュアルを参照。

**メモ】** モニタ環境下で割り込みを使用する場合のプログラミング方法は、「13 APPEDIX.C 割り込みプログラミング」を参照してください。

### 7.3. 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。



リモート・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。

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

SRAM 空間には、128K-byte \* 32bit の PB-SRAM を 2 個実装し、パリティ付きで 1M-Byte の容量を搭載しています。最大バスクロックまで 0 ウェイトでアクセス可能ですが、0 ~ 3clk のウェイトを入れてのウェイトエミュレーションも可能です。(「7.4.1 SRAM コントローラ (SRAMC B800-6000[Read/Write])」を参照)。

空間内のアドレス線の上位ビットはデコードしていませんので、1M-Byte おきにイメージが発生します。キャッシュ / 非キャッシュどちらの空間でもアクセスできます。

#### 7.3.2. DRAM 空間 (0800-0000H ~ 0FFF-FFFFH)

DRAM 空間には、4M \* 8bit \* 4bank の SDRAM を 4 個実装し、64M-Byte の容量を搭載しています。使用に先立ち、SDRAM への設定が必要です。(「7.4.2 DRAM コントローラ (DRAMC B800-5000[Read/Write])」を参照)。

空間内のアドレス線の上位ビットはデコードしていません。その結果、64M-Byte 先にイメージが発生します。キャッシュ / 非キャッシュどちらの空間からでもアクセスできます。

#### 7.3.3. GBUS 空間 (1000-0000H ~ 17FF-FFFFH)

GBUS 空間は GBUS へのアクセス空間がマップされています。詳細には、「10 GBUS 個別仕様」を参照してください。本ボードからはキャッシュ / 非キャッシュどちらの空間からでもアク

セスできますがアクセス先の資源によって使い分けください。

#### 7.3.4. I/O 空間 (1800-0000H ~ 1EFF-FFFFH)

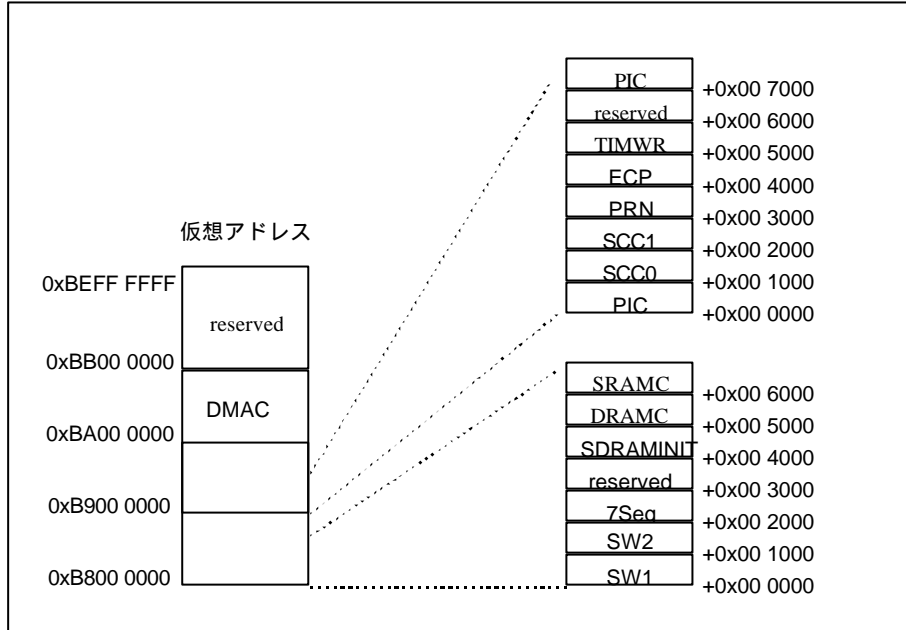
I/O 空間はボード上の各機能を制御するための I/O デバイスが割り付けられている空間 (メモリ・マップド I/O) です。I/O マップなどの詳細については「7.4 I/O 詳細」を参照してください。非キャッシュ空間でアクセスします。

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

ROM 空間は 128K バイト (64KWord × 16Bit)、256K バイト (128Kword × 16Bit)、512K バイト (256KWord × 16Bit) のいずれかで、アクセスタイムを 120nS 以下の UV-EPROM が実装できます。搭載する ROM の種類と使用方法は、SW3 で設定します (「5.6 スイッチ 3 (SW3)」参照)。アドレス線の上位ビットはデコードしていませんので、ROM の容量おきにイメージが現れます。ROM の Wait 数は上限周波数において、120nS 以上のアクセスタイムを確保します。キャッシュ / 非キャッシュどちらの空間からでもアクセスできます。

7.4. 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などがあります。各 I/O の割り付けは以下の通りです。



I/O マップ

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、D0 を LSB とした 32-Bit のバスです。アクセスは 32-Bit バス幅以下でアクセスしてください。また、ワードアクセス(32-Bit)以外のアクセスサイズでアクセスする場合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+3 を、ハーフワードでアクセスする場合には、バイト・オフセットとして+2 をそれぞれのアドレスに加算してアクセスしてください。リトル・エンディアンの場合そのままのアドレスでアクセスできます。

7.4.1. SRAM コントローラ (SRAMC B800-6000[Read/Write])

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
B800-6000H	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	備考
1	0		
0	0	0	推奨値
0	1	1	
1	0	2	
1	1	3	(リセット値)

#### 7.4.2. DRAM コントローラ (DRAMC B800-5000[Read/Write])

DRAMC は SDRAM のアクセス条件設定を行いません。レジスタ割り付けは下表の通りです。Rev.2 以降のボードでは、リセットによって推奨値に設定されますので通常設定は不要ですが、Rev.1 のボードとの互換性を確保する為に推奨値での再設定を推奨します。

論理アドレス	レジスタ	データバス			
		D3	D2	D1	D0
<b>B800-5000H</b>	<b>DRAMC RCD</b>	0	0	RCD1	RCD0
-5010H	DRAMC RP	0	0	RP1	RP0
-5020H	DRAMC RC	0	RC2	RC1	RC0
-5030H	DRAMC RRC	0	RRC2	RRC1	RRC0
-5040H	DRAMC RAS	0	RAS2	RAS1	RAS0
-5050H	DRAMC LAT	0	0	LAT1	LAT0
-5060H	DRAMC RSC	0	0	RSC1	RSC0
-5070H	DRAMC AP	0	0	0	AP

**RCD[1..0]:** SDRAM のパラメータ:tRCD(Delay ACT to READ/WRITE)を設定します。

RCD		tRCD	備考
1	0		
0	0	--	
0	1	--	
1	0	2 SYSCLK	推奨値 (リセット値)
1	1	3 SYSCLK	

**【メモ】** 推奨値でご使用ください。推奨値以外の設定での動作は保証しません。

**RP[1..0]:** SDRAM のパラメータ:tRP(PRE to ACT/REF)を設定します。

RP		tRP	備考
1	0		
0	0	--	
0	1	--	
1	0	2 SYSCLK	推奨値 (リセット値)
1	1	3 SYSCLK	

**【メモ】** 推奨値でご使用ください。推奨値以外の設定での動作は保証しません。

**RC[2..0]:** SDRAM のパラメータ:tRC(ACT to REF/ACT)を設定します。

RC			tRC	備考
2	1	0		
0	0	0	--	
0	0	1	--	
0	1	0	--	
0	1	1	--	
1	0	0	--	
1	0	1	5 SYSCLK	推奨値 (リセット値)
1	1	0	6 SYSCLK	
1	1	1	7 SYSCLK	

**【メモ】** 推奨値でご使用ください。推奨値以外の設定での動作は保証しません。

**RRC[2..0]:** SDRAM のパラメータ:tRRC(REF to REF/ACT)を設定します。

RRC			tRRC	備考
2	1	0		
0	0	0	--	
0	0	1	--	
0	1	0	--	
0	1	1	--	
1	0	0	--	
1	0	1	--	
1	1	0	6 SYSCLK	推奨値 (リセット値)
1	1	1	7 SYSCLK	

**【メモ】** 推奨値でご使用ください。推奨値以外の設定での動作は保証しません。

**RAS[2..0]:** SDRAM のパラメータ:tRC(ACT to PRE)を設定します。

RAS			tRAS	備考
2	1	0		
0	0	0	--	
0	0	1	--	
0	1	0	--	
0	1	1	--	
1	0	0	4SYSCLK	推奨値 (リセット値)
1	0	1	--	
1	1	0	--	
1	1	1	7 SYSCLK	

**【メモ】** 推奨値でご使用ください。推奨値以外での設定での動作は保証しません。

**LAT[1..0]:** SDRAM のパラメータ:tLAT(/CAS latency)を設定します。

LAT		tLAT	備考
1	0		
0	0	--	
0	1	--	
1	0	2 SYSCLK	推奨値
1	1	3 SYSCLK	(リセット値)

**【メモ】** 推奨値でご使用ください。推奨値以外の設定での動作は保証しません。

**RSC[1..0]:** SDRAM のパラメータ:tRSC(Mode Register Set Cycle Time)を設定します。

RSC		tRSC	備考
1	0		
0	0	--	
0	1	--	
1	0	2 SYSCLK	推奨値 (リセット値)
1	1	3 SYSCLK	

**【メモ】** 推奨値でご使用ください。推奨値以外の設定での動作は保証しません。

**AP:** SDRAM のオートプリチャージ機能を使用するかどうかを指定します。

AP	オートプリチャージ	備考
0	使用する。	
1	使用しない	リセット値

#### 7.4.3. DRAM 初期化ポート ( DRAM\_INIT B800-4000H[Write Only] )

DRAMC の設定値に基づき SDRAM を初期化します。データは Don't Care です。出力することで SDARM の Mode Register Set Command を発行します。

このポートはポートへ設定したパラメータを SDRAM に書き込むためのものです。全のパラメータをポートに設定後、最後にこのポートへ出力してください。



#### 7.4.4. SW1 読み出しポート(SW1 B800-0000H [Read Only])

SW1 の状態を読み出すためのポートです。データ・フォーマットを下表に示します。

物理アドレス	データバス								内容
	D7	D6	D5	D4	D3	D2	D1	D0	
B800-0000H 入力	SW1-8	SW1-7	SW1-6	SW1-5	SW1-4	SW1-3	SW1-2	SW1-1	0=ON 1=OFF

SW1-1 が SW1 の”1”のスイッチに、SW1-8 が SW1 の”8”のスイッチに対応しています。また、該当するビットのスイッチが ON で 0 が、OFF で 1 が読み出されます。SW3 は、モニタの動作設定用のスイッチとして使用しています。設定方法は、「11.1.2 SW1 の設定」、「12.1.1 SW1 の設定」の各項を参照してください。

#### 7.4.5. SW2 読み出しポート(SW2 B800-1000H [Read Only])

SW2 の状態を読み出すためのポートです。データ・フォーマットを下表に示します。

物理アドレス	データバス								内容
	D7	D6	D5	D4	D3	D2	D1	D0	
B800-1000H 入力	SW2-8	SW2-7	SW2-6	SW2-5	SW2-4	SW2-3	SW2-2	SW2-1	0=ON 1=OFF

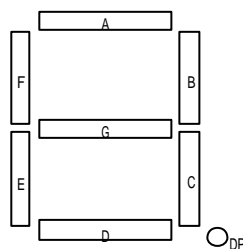
SW2-1 が SW2 の”1”のスイッチに、SW2-8 が SW2 の”8”のスイッチに対応しています。また、該当するビットのスイッチが ON で 0 が、OFF で 1 が読み出されます。SW2 は、ハードウェアの動作を切り替えます。それぞれのスイッチの機能は、「5.5 スイッチ 2 (SW2)」を参照してください。

#### 7.4.6. 7セグメント LED 表示データ出力ポート(7SEG-LED B800-2000 [Write Only])

4 つの 7 セグメント LED に表示するデータを設定します。データ・フォーマットを下表に示します。該当するビットに 0 を設定すると対応するセグメントが点灯します。

論理アドレス	データ								内容
	D7..31	D6..30	D5..29	D4..28	D3..27	D2..26	D1..25	D0..24	
B800-2000H 出力	LED1 -DP	LED1 -G	LED1 -F	LED1 -E	LED1 -D	LED1 -C	LED1 -B	LED1 -A	0=点灯 1=消灯
B800-2001H 出力	LED2 -DP	LED2 -G	LED2 -F	LED2 -E	LED2 -D	LED2 -C	LED2 -B	LED2 -A	
B800-2002H 出力	LED3 -DP	LED3 -G	LED3 -F	LED3 -E	LED3 -D	LED3 -C	LED3 -B	LED3 -A	
B800-2003H 出力	LED4 -DP	LED4 -G	LED4 -F	LED4 -E	LED4 -D	LED4 -C	LED4 -B	LED4 -A	

7 セグメント LED のビット対応は、下図の通りです。



#### 7.4.7. シリアル/パラレル I/O (SCC0/1, LPT B900-1000 ~ B900-4000[Read/Write])

UART/PRINTER として TEXAS INSTRUMENTS 製の TL16PIR552(DUAL UART WITH 1284

PARALLEL PORT)LSI を使用しています。TL16PIR552 は、UART を 2 チャンネル、IEEE1284 準拠の双方向プリンタ・ポートを 1 チャンネル備えており、UART の送受信部には 16 キャラクタ分の FIFO バッファを持ち、RTS/CTS フローを自動的に制御する機能を備えているため、最小限の割込みでオーバーラン・エラーを押さえられます。

TL16PIR552 の各レジスタは、表のように割り付けられています。各レジスタの機能については、TL16PIR552 のマニュアルを参照してください ( TL16PIR552 のマニュアルは米国 TEXAS INSTRUMENTS 社のホームページ( <http://www.ti.com/> )の TI&ME のコーナーで入手可能です )。

アドレス	機能	読み出し	書き込み
B900-1000	UART-CH#0	RBR/DLL	THR/DLL
B900-1010		IER/DLM	IER/DLM
B900-1020		IIR	FCR
B900-1030		LCR	LCR
B900-1040		MCR	MCR
B900-1050		LSR	LSR
B900-1060		MSR	MSR
B900-1070		SCR	SCR
B900-2000	UART-CH#1	RBR/DLL	THR/DLL
B900-2010		IER/DLM	IER/DLM
B900-2020		IIR	FCR
B900-2030		LCR	LCR
B900-2040		MCR	MCR
B900-2050		LSR	LSR
B900-2060		MSR	MSR
B900-2070		SCR	SCR
B900-3000	PRINTER(PCPS-)	DATA	DATA/ECPAFIFO
B900-3010		DSR	----
B900-3020		DCR	DCR
B900-3030		EPPADDR	EPPADDR
B900-3040 ~ B900-3070		EPPDATA	EPPDATA
B900-4000	PRINTER(ECPCS-)	PPDATAFIFO/ TESTFIFO/CNFGA	PPDATAFIFO/ TESTFIFO
B900-4010		CNFGB	----
B900-4020		ECR	ECR

TL16PIR552 レジスタ配置

TL16PIR552 の XIN 入力には 16MHz のクロックが接続されています。

UART-CH#0、UART-CH#1、PRINTER の各割込みは PIC を介して、CPU の割込みに接続することが可能です。

UART-CH#0 はボードのリアパネルにある JSIO1 コネクタに、UART-CH#1 は JSIO2 コネクタ、PRINTER は JPRT に接続しています。また、UART-CH#0 はデバッグをシリアル通信で用いる場合に使用され、その時の割込みは、PIC を通して、INT0 を使用します。

TL16PIR552 は、システム・リセットによってリセットされます。

**注意：モニタを使用中は、UART-CH#0 の内容は変更しないでください。**

#### 7.4.8. タイマ (Timer B900-5000[Read/Write])

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
B900-5000	PCNT0	タイマ 0
B900-5010	PCNT1	タイマ 1
B900-5020	PCNT2	タイマ 2
B900-5030	PCNTL	制御

各レジスタの機能については、uPD71054 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	補足
0	2MHz	2	タイマ割込み 0	モニタで使用
1	2MHz	2	タイマ割込み 1	ユーザで使用可
2	2MHz	2	DRAM リフレッシュ	

**注意：モニタ使用中は、タイマ0の内容は変更しないでください。**

## 7.4.9. 割り込みコントローラ (PIC B900-0000[Read/Write])

PIC は、割り込みの制御を行いません。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
B900-0000	PIC INT0M	IM07	IM06	IM05	IM04	IM03	IM02	IM01	IM00
B900-0010	PIC INT1M	IM17	IM16	IM15	IM14	IM13	IM12	IM11	IM10
B900-0020	PIC INTR	IR7	IR6	IR5	IR4	IR3	IR2	IR1	IR0
B900-0030	PIC INTEN	0	0	0	0	0	0	INT1 EN	INT0 EN

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 (High)
2	GBUS-INT0-	レベル (Low)
3	BUS ERROR	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 (High)
6	パラレル (プリンタ)	エッジ (立ち上がり)
7	DMAC INTREQ-	レベル(Low)

INTEN レジスタは、INT0,INT1 への割り込みの有効 / 禁止を制御します。

**INTEN0:** INT0 を制御します。このビットで INT0 をハード的に禁止することができます。この時、INT0 端子の状態は High レベルとなります。

INTEN0	INT0	補足
0	禁止する	(リセット値)
1	有効にする	

**INTEN1:** INT1 を制御します。このビットで INT1 をハード的に禁止することができます。この時、INT1 端子の状態は High レベルとなります。

INTEN1	INT1	補足
0	禁止する	(リセット値)
1	有効にする	

**注意:** モニタを使用中は、INT0関連の内容は変更しないでください。

#### 7.4.10. DMA コントローラ

GBUS からの DMA 要求に対し、GBUS 資源間で DMA 転送を行うコントローラです。全部で 4 チャンネルの DMA コントローラがあります。

DMA 転送は常に 2 サイクル (読みだしサイクルと書き込みサイクル) で行われます。また、バースト・サイクルはサポートしていません。

##### 7.4.10.1.DMA チャンネル 0 制御ポート (DMACNT0 BA00-0000[Read/Write])

##### 7.4.10.2.DMA チャンネル 1 制御ポート (DMACNT1 BA00-0080[Read/Write])

##### 7.4.10.3.DMA チャンネル 2 制御ポート (DMACNT2 BA00-0100[Read/Write])

##### 7.4.10.4.DMA チャンネル 3 制御ポート (DMACNT3 BA00-0180[Read/Write])

Bit	信号名	RST	機能
0	DMA_EN <sup>*1</sup>	0	0:DMA 停止 1:DMA 実行
1	TC <sup>*2</sup>	0	0:DMA 転送未終了 1:DMA 転送終了
2	DMA_SINGLE <sup>*3</sup>	x	0:連続して GBUS を使用する 1:1 転送ごとに GBUS を一旦手放す
3	SOFT_DREQ <sup>*4</sup>	x	0:ハードウェアの DMA 要求線に従って DMA 転送を行う 1:ハードウェアの DMA 要求線がインアクティブでも DMA 転送を行う
4	DATA_SIZE[0] <sup>*5</sup>	x	DATA_SIZE[1:0]で転送するデータのサイズを設定します
5	DATA_SIZE[1] <sup>*5</sup>	x	
6	未使用	x	
7	未使用	x	
8	DST_CD[0] <sup>*6</sup>	x	DST_CD[1:0]で、転送先アドレスの増減を設定します
9	DST_CD[1] <sup>*6</sup>	x	
10	未使用	x	
11	未使用	x	
12	SRC_CD[0] <sup>*6</sup>	x	SRC_CD[1:0]で、転送元アドレスの増減を設定します。
13	SRC_CD[1] <sup>*6</sup>	x	
14	未使用	x	
15	未使用	x	
16	DST_CS[0] <sup>*7</sup>	x	DST_CS[1:0]で、転送先アドレスのGBUSのチップセレクト空間を指定します。
17	DST_CS[1] <sup>*7</sup>	x	
18	DST_CS[2] <sup>*7</sup>	x	
19	未使用	x	
20	SRC_CS[0] <sup>*7</sup>	x	SRC_CS[1:0]で、転送元アドレスのGBUSのチップセレクト空間を指定します。
21	SRC_CS[1] <sup>*7</sup>	x	
22	SRC_CS[2] <sup>*7</sup>	x	
23	未使用	x	
24	未使用	x	
25	未使用	x	
26	未使用	x	
27	未使用	x	
28	未使用	x	
29	未使用	x	
30	未使用	x	
31	未使用	x	

#### 《注意事項》

- DMA\_EN は、他のビット設定が終わってから”1”に設定してください。他のビットの操作と同時に DMA\_EN を”0”から”1”に設定しないでください。  
また、DMA\_EN が”1”の時は、DMA コントローラ内の他のビットを書き換えしないでください。書き換えた場合の動作は不定です。
- TC は、バイト・カウント・レジスタに設定したバイト数分の転送が終了すると”1”にな

ります。TC をクリアするためには、このビットに”1”を書き込んでください。

また、TC をクリアする前に DMA\_EN を”0”に設定してください。DMA\_EN が”1”の時に TC をクリアすると、クリアした途端に再度 DMA が開始してしまいます。

- 3 . DMA\_SINGLE を”1”に設定すると、DMA コントローラは 1 回の DMA 転送が終わるたびに GBUS の使用権を放棄し、CPU が GBUS を使用できるようにします。DMA\_SINGLE を”0”に設定すると、DMA コントローラは GBUS の使用権を確保したまま DMA を継続し、DMA 転送が終了するまで CPU は GBUS の使用権を取ることができなくなります。詳しくは、『7.4.10.20 シングル転送と非シングル転送』を参照してください。
- 4 . SOFT\_DREQ を”1”に設定すると、ハード的な DMA 要求を必要とせずに DMA 転送が行われます。ただし、SOFT\_DREQ が”1”の場合でも、GBUS の DMAAK-信号が DMA サイクル時にアクティブになります。
- 5 . DATA\_SIZE[1:0]は、次のフォーマットです。
  - [0,0] : 8Bit 転送
  - [0,1] : 16Bit 転送
  - [1,0] : 32Bit 転送
  - [1,1] : 設定禁止
- 6 . DST\_CD[1:0]と SRC\_CD[1:0]は、次のフォーマットです。転送アドレスの増加 / 減少の単位は DATA\_SIZE[1:0]で設定された内容で決定します。
  - [0,0] : 固定
  - [0,1] : 増加
  - [1,0] : 減少
  - [1,1] : 設定禁止
- 7 . DST\_CS[2:0]と SRC\_CS[2:0]は、次のフォーマットです。
  - [0,0,0] : GCS0-空間
  - [0,0,1] : GCS1-空間
  - [0,1,0] : GCS2-空間
  - [0,1,1] : GCS3-空間
  - [1,0,0] : GCS4-空間
  - [1,0,1] : GCS5-空間
  - [1,1,0] : GCS6-空間

#### 7.4.10.5.DMA チャンネル 0 転送元アドレスポート ( DMASRCADDR0 BA00-0010[Read/Write] )

#### 7.4.10.6.DMA チャンネル 1 転送元アドレスポート ( DMASRCADDR1 BA00-0090[Read/Write] )

#### 7.4.10.7.DMA チャンネル 2 転送元アドレスポート ( DMASRCADDR2 BA00-0110[Read/Write] )

#### 7.4.10.8.DMA チャンネル 3 転送元アドレスポート ( DMASRCADDR3 BA00-0190[Read/Write] )

DMA の転送元アドレスを設定するポートです。アドレスは A0 ~ A23 の 24Bit が有効です。転送アドレスは、各チャンネルの DMA 制御ポートに設定した DATA\_SIZE[1:0]のデータ・サイズにアラインされたアドレスでなければなりません。設定アドレスがアラインされていない場合の動作は不定です。

また、DMA 転送中もしくは転送終了後には、このポートからは次に DMA 転送を行うアドレスの内容が読み出せます。

## 7.4.10.9.DMAチャンネル0 転送先アドレスポート (DMADSTADDR0 BA00-0020[Read/Write])

## 7.4.10.10.DMAチャンネル1 転送先アドレスポート (DMADSTADDR1 BA00-00A0[Read/Write])

## 7.4.10.11.DMAチャンネル2 転送先アドレスポート (DMADSTADDR2 BA00-0120[Read/Write])

## 7.4.10.12.DMAチャンネル3 転送先アドレスポート (DMADSTADDR3 BA00-01A0[Read/Write])

DMAの転送先アドレスを設定するポートです。アドレスはA0～A23の24Bitが有効です。転送アドレスは、各チャンネルのDMA制御ポートに設定したDATA\_SIZE[1:0]のデータ・サイズにアラインされたアドレスでなければなりません。設定アドレスがアラインされていない場合の動作は不定です。

また、DMA転送中もしくは転送終了後には、このポートからは次にDMA転送を行うアドレスの内容が読み出せます。

## 7.4.10.13.DMAチャンネル0 バイトカウントポート (DMABYTECNT0 BA00-0030[Read/Write])

## 7.4.10.14.DMAチャンネル1 バイトカウントポート (DMABYTECNT1 BA00-00B0[Read/Write])

## 7.4.10.15.DMAチャンネル2 バイトカウントポート (DMABYTECNT2 BA00-0130[Read/Write])

## 7.4.10.16.DMAチャンネル3 バイトカウントポート (DMABYTECNT3 BA00-01B0[Read/Write])

DMA転送する**バイト数 - 1**を設定するポートです。データはBit0～Bit23の24Bitが有効です。DMAコントローラは、1回のDMA転送が終わるごとに、このポートの内容を各チャンネルのDMA制御ポートに設定したDATA\_SIZE[1:0]のデータ・サイズに従って1,2もしくは4減少させます。この減算の結果キャリーが発生すると、DMAが終了します(TCが"1"になります)。また、DMA転送中もしくは転送終了後には、現在のバイトカウンタの値が読み出せます。

## 7.4.10.17.DMA割り込み制御ポート (DMAINTCNT BA00-0200[Read/Write])

Bit	信号名	RST	機能
0	INT_EN[0]	0	0:チャンネル0のTCビットが"1"の時に割り込みを要求しない。 0:チャンネル0のTCビットが"1"の時に割り込みを要求する。
1	INT_EN[1]	0	0:チャンネル1のTCビットが"1"の時に割り込みを要求しない。 0:チャンネル1のTCビットが"1"の時に割り込みを要求する。
2	INT_EN[3]	0	0:チャンネル2のTCビットが"1"の時に割り込みを要求しない。 0:チャンネル2のTCビットが"1"の時に割り込みを要求する。
3	INT_EN[4]	0	0:チャンネル3のTCビットが"1"の時に割り込みを要求しない。 0:チャンネル3のTCビットが"1"の時に割り込みを要求する。
4	TC[0] <sup>*1</sup>	0	チャンネル0のTCビットの内容の読みだし
5	TC[1] <sup>*1</sup>	0	チャンネル1のTCビットの内容の読みだし
6	TC[2] <sup>*1</sup>	0	チャンネル2のTCビットの内容の読みだし
7	TC[3] <sup>*1</sup>	0	チャンネル3のTCビットの内容の読みだし

## 《注意事項》

- TC[3:0]は、各チャンネルのDMA制御ポートのBit1の内容がそのまま読み出せます。このビットにより、どのチャンネルが割り込み要求をしているかが判断できます。このポートのTC[3:0]は読みだし専用であり、TCをクリアするには各チャンネルの制御ポートのTCを操作する必要があります。

## 7.4.10.18.DMA エンディアン指定ポート (DMAENDIAN BA00-0280[Read/Write])

Bit	信号名	RST	機能
0	BIG-/LITTLE	1	0: BIG 1: LITTLE
1		X	未使用
2		X	未使用
3		X	未使用
4		Xv	未使用
5		X	未使用
6		X	未使用
7		X	未使用

## 《注意事項》

CPU の動作中のエンディアンと同じに設定にしてください。この設定は全 DMA チャンネルに共通です。

## 7.4.10.19.優先順位

各チャンネルは、チャンネル0が最も優先順位が高く、チャンネル3が最も低い優先順位となっています。複数のDMA要求が競合した場合に、どのチャンネルのDMA転送が行われるかは、GBUS使用権を取得後のバス・サイクル開始直前のDMA要求の状態です。

## 7.4.10.20.シングル転送と非シングル転送

DMA制御ポートのDMA\_SINGLEへの設定は、各チャンネルの1サイクル分のDMA転送終了後に、GBUS使用権を手放すかどうかを決定します。各チャンネルのDMA\_SINGLEの設定は、対応するチャンネルのDMA転送サイクル後の状態を決定します。

DMA\_SINGLEを”1”に設定すると、DMAコントローラは1回の転送を終了するたびにGBUSの使用権を放棄し、CPUがGBUSを使用する余地を与えます。引き続きDMA要求がアクティブであれば、DMAコントローラは改めてGBUSの使用権を要求します。

DMA制御ポートのDMA\_SINGLEを”0”に設定すると、DMA要求がアクティブであると(もしくはSOFT\_DREQが”1”の場合)、転送が終了後もGBUSの使用権を手放さずに、続けてDMA転送を行います。ただし、チャンネル間の優先順位の調停は行われるため、同じチャンネルのDMA転送が継続して起きるとは限りません。

## 7.4.10.21.非シングル転送時のDMA要求サンプル・タイミング

非シングル転送時に継続してDMAサイクルを行うかどうかは、転送先への書き込みサイクルがレディーになる(GREADYがLow、GWAITIがHigh)タイミングのDMA要求の状態によります。このタイミングで、DMA要求がアクティブであれば、GBUSの使用権を確保したままで継続的にDMA転送を行います。



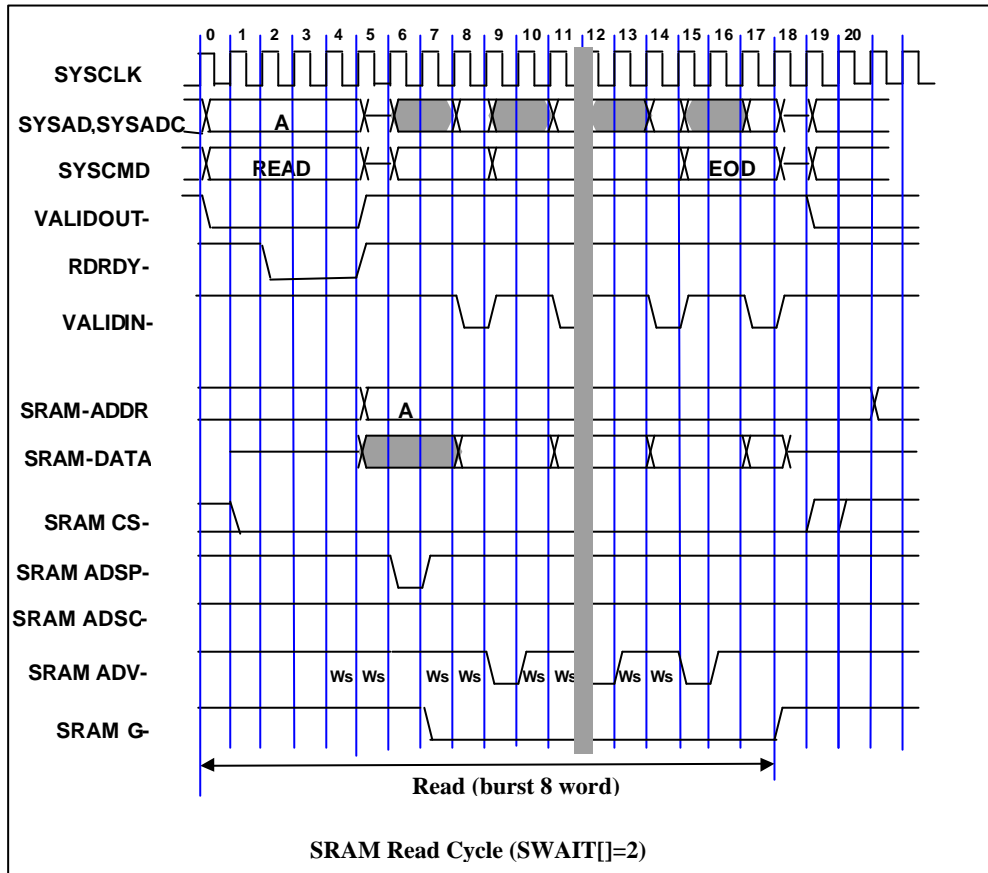
## 8. バスサイクル・タイミング

RTE-VR5500-CB では、SRAM, DRAM, ROM, I/O などの各デバイスに合わせて、SYSAD バスからブリッジを介しアクセスしています。ここでは、各アクセス・サイクルの基本的な制御タイミングについて説明します。

### 8.1. 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 (「7.4.1 SRAM コントローラ (SRAMC B800-6000[Read/Write])」参照)

下図は、2 ウェイトを設定した時のバーストリードの例です。



クロックの番号に添って状態を説明します。

0~1: 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。

2~4: SRAM のリードサイクルが確定し、RDRDY-がアクティブになります。

4~5: 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。

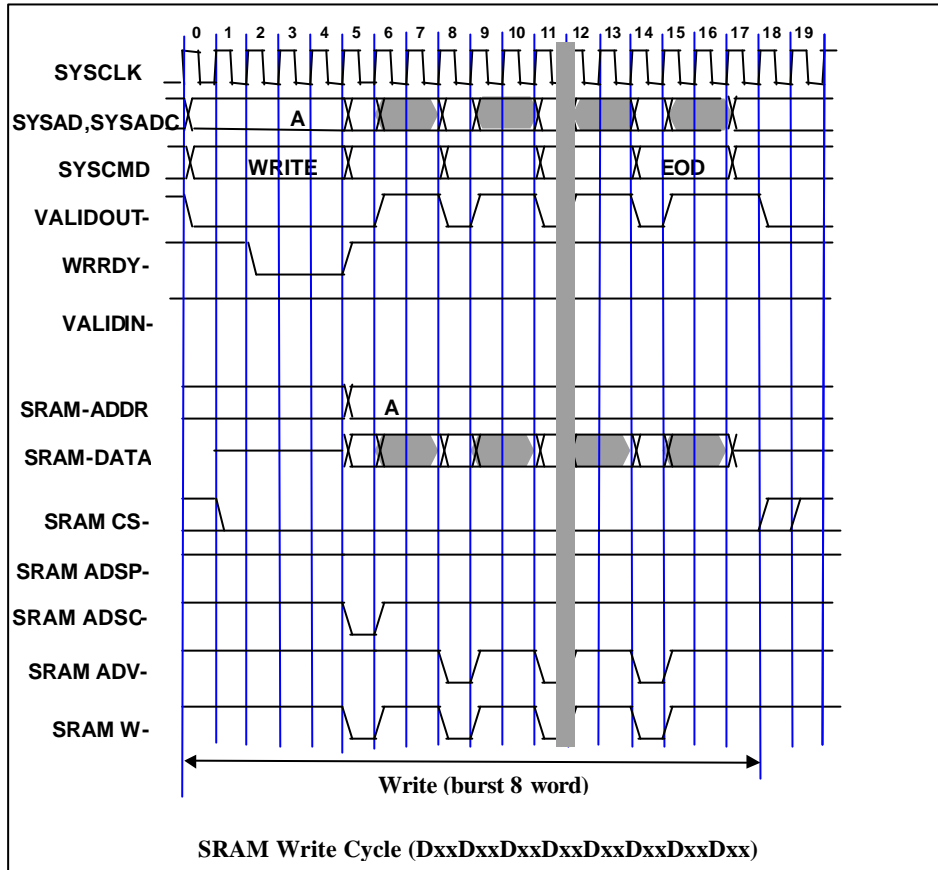
6: PBSRAM へリードコマンドを与えます。

8: PBSRAM からデータが出力されます。これに合わせ EVALID-を返します。

9~17: 最後のデータ(EOD)となるまで計8回サイクルをくり返します。PBSRAM へのアドレスは初回と5回目だけ外部からロードし、それ以外は PBSRAM 内部のアドレスカウンタをインクリメントして使用します。

(7~9): SWAIT で設定したウェイトサイクル (この例では2 ウェイト) が挿入された後、SRAM ADV-をアクティブにし、PBSRAM の内部アドレスを加算させます。

SRAM のライトサイクルは、CPU の出力データレート(DDDDDDDD から DxxDxxDxxDxxDxxxDxxxDxxxDxxx)に従います。SRAM の場合には、アドレス確定から2クロック後に WRRDY-を返してサイクルをスタートさせます。下図は、データレート:DxxDxxDxxDxxDxxDxxDxxDxx のバーストサイクルの例です。



クロックの番号に添って状態を説明します。

0~1: CPU の外部ライトサイクルの始まりです。通常 WRRDY-はインアクティブです。

2~4: SRAM のライトサイクルが確定し、WRRDY-がアクティブになります。

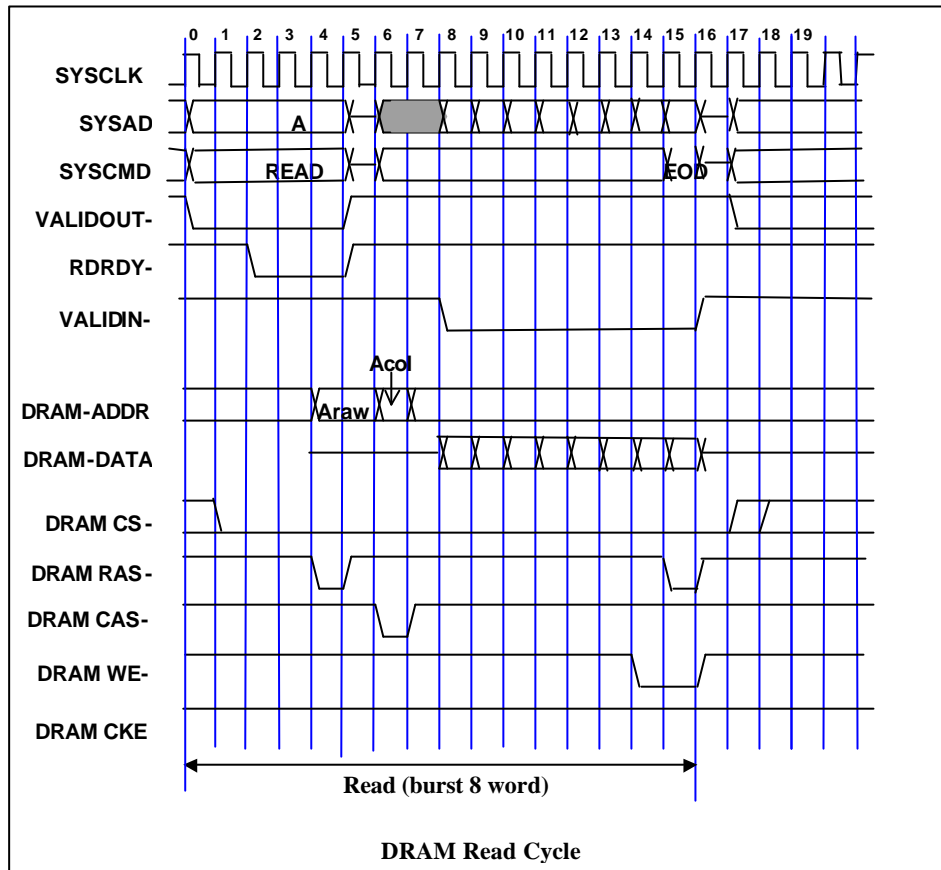
5: 最初のデータ出力です。この時のデータを PBSRAM に直接ラッチさせ、ライト動作を開始します。アドレスは予めラッチしており、この時に同時に与えます。

6~14: ライト動作を計8回繰り返します。PBSRAM へのアドレスは初回と5回目だけ外部からロードし、それ以外は PBSRAM 内部のアドレスカウンタをインクリメントして使用します。SRAM ADV-, SRAM W-は、データレートに合わせてタイミングをとっています。(データレート:DDDDDDDD の時は連続してアクティブになります)

## 8.2. DRAM アクセス

DRAM は、DRAM コントローラの設定が必要です。(「7.4.2 DRAM コントローラ ( DRAMC B800-5000[Read/Write] )」参照)。以下は、全て推奨値に設定した場合のもので、

PAGE=0 ( オートプリチャージ ON ) の場合の DRAM リードサイクルを下図に示します。ただし、これはサイクルの開始にあたって、リフレッシュやプリチャージと重ならなかった場合のもので、重なった場合は、それらのサイクルが終了するまで待たされます。



クロックの番号に添って状態を説明します。

0～1: CPU の外部リードサイクルの始まりです。通常、WRRDY-は、インアクティブです。

2～4: DRAM へのアクセスが確定し、 WRRDY-がアクティブになり、4で SDRAM に対し、アクティブコマンドを発行します。この時 DRAM ADDR には、ロー(Raw)アドレスを与えます。

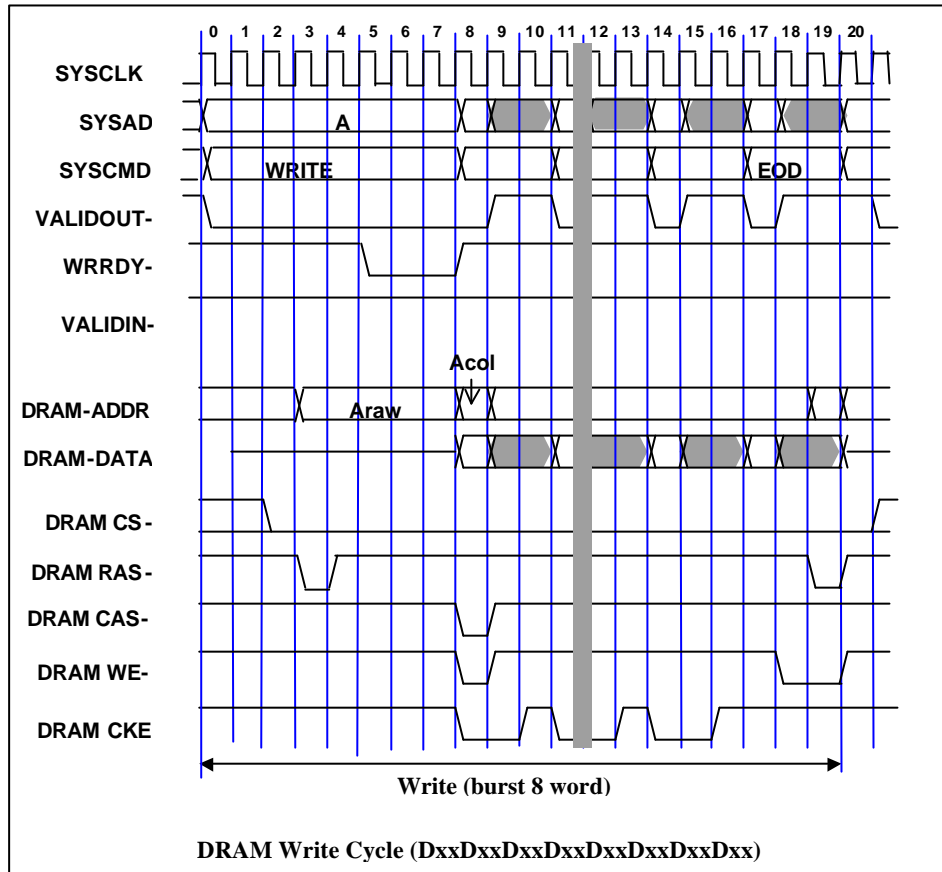
5～6: tRCD を待つ、SDRAM に対し、6で Read コマンドを与えます。この時、DRAM ADDR には、コラム(Column)アドレスを与えます。

8: 最初のリードデータ確定です。VALIDIN-をアクティブにして、このデータを直接 CPU に読み込ませます。

9～15: 連続するバーストで SDRAM からデータを読み出します。この間、CPU に対し VALIDIN-をアクティブにします。SDRAM に対しては、14でバーストストップコマンドを与え、15でプリチャージコマンドを与えます。

補足 : AP=1( オートプリチャージ OFF ) の場合は、15でプリチャージコマンドは発行しません。またこの場合、アクセスの開始に際し Raw アドレスがヒットした場合は、4～5のアクティブコマンドが除かれます。

DRAM のライトサイクルは、CPU の出力データレート (DDDDDDDD から DxxxDxxxDxxxDxxxDxxxDxxxDxxx)に従います。DRAM の場合には、アドレス確定から 3 クロック後にアクティブコマンドを発行し、その 2 クロック後に WRRDY-を返してサイクルをスタートさせます。下図は、データレート:DxxDxxDxxDxxDxxDxxDxxDxx のバーストサイクルの例です。



クロックの番号に添って状態を説明します。

- 0~1: CPU の外部ライトサイクルの始まりです。通常 EOK-はインアクティブとなっています。
- 2~3: DRAM のライトサイクルが確定し、3 で SDRAM に対しアクティブコマンドを発行します。この時、DRAM ADDR には、ロウ(Row)アドレスを与えます。
- 4~7: アクティブコマンドから 2 クロック後に WRRDY-をアクティブとし、WRRDY-から 3 クロック後に CPU の外部ライトサイクルが始まります。
- 8: 最初のデータ出力です。SDRAM に対し、このデータを直接書き込むためにライトコマンドを発行します。この時、DRAM ADDR には、カラム (Column) アドレスを与えます。
- 9~17: 継続するバーストデータのライト動作を計 8 回繰り返します。2 番目以降のライトはデータレートに合わせるために DRAM\_CKE でタイミングを取っています。
- 18: バーストストップコマンドを SDRAM に対し、発行します。
- 19: プリチャージコマンドを SDRAM に対し、発行します。

補足 : AP=1 (オートプリチャージ OFF) の場合は、18 でプリチャージコマンドは発行しません。またこの場合、アクセスの開始に際し Row アドレスがヒットした場合は、3~4 のアクティブコマンドが除かれます。

### オートプリチャージに関して

AP=1 (オートプリチャージ OFF) の場合には、プリチャージが発行される迄の間、SDRAM の内部でローアドレスが一時的にホールドされます。この状態で同一のロウ・アドレス空間へアクセスした場合 (ヒット) 、アクティブコマンドが省略されますので、リードサイクル、ライトサイクル共に、2クロックサイクル分少ないクロックで、SDRAM のアクセスが開始します。但し、不一致の場合 (アンヒット) には、目的のアクセスに先立ち、プリチャージを行う必要がありますので、オーバーヘッドが多くなります。

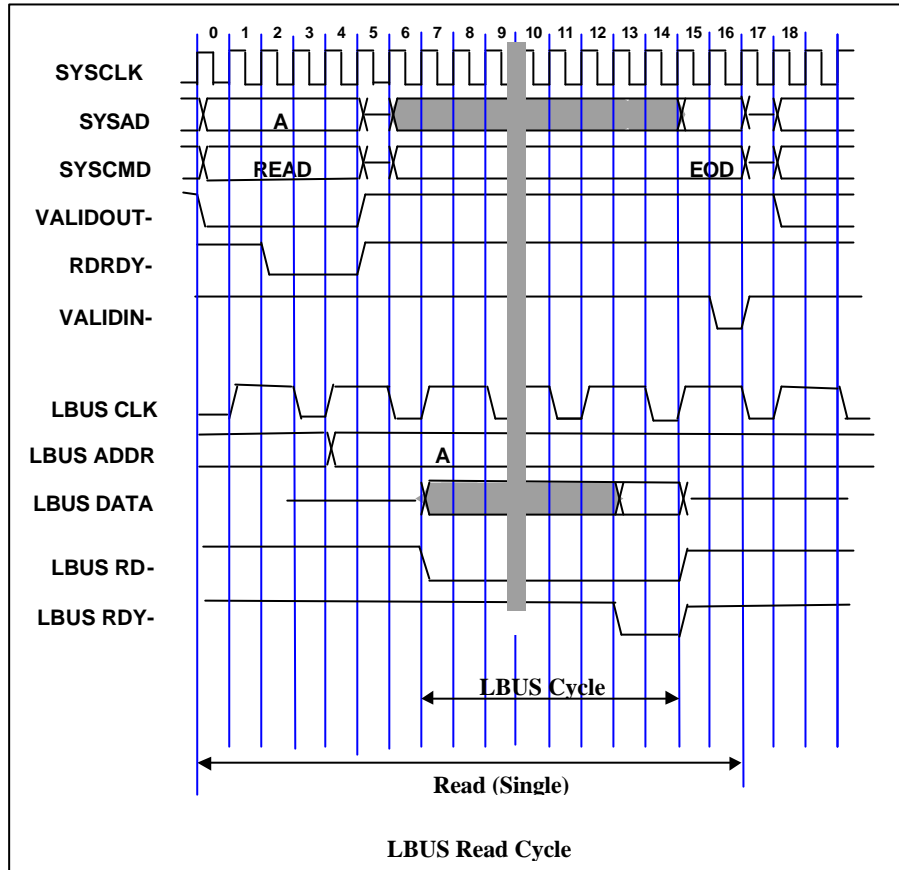
### リフレッシュに関して

SDRAM のリフレッシュは、CBR(auto)リフレッシュコマンドを使用し、タイマ 2 の設定により約 15  $\mu$  秒ごとに行っています。

### 8.3. ローカルバス・アクセス

ローカルバスは SYSCLK の 1/3 で動作する 32 ビットのデータ幅を持つ同期バスです。

ローカルバスのリードサイクルを下図に示します。



クロックの番号に添って状態を説明します。

0~1: CPU の外部リードサイクルの始まりです。通常 RDRDY-はインアクティブです。

2~4: LBUS へのリードサイクルが確定し、RDRDY-がアクティブになります。

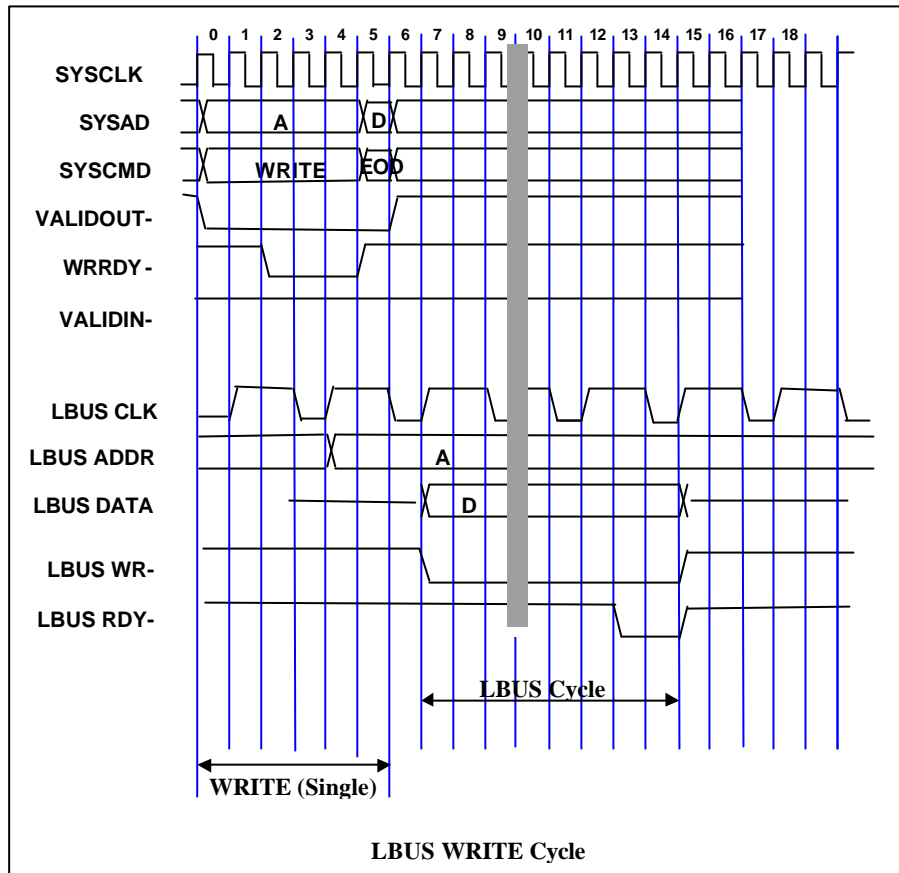
7: LBUS へのリードサイクルが開始します。

9~10: アクセスする資源によってこの間のクロックは異なります。

12~14: LBUS RDY-がアクティブになることでLBUS 上にデータが用意されたことを示します。

14~16: LBUS からのデータは、SYSCLK で同期を取って、SYSAD へ出力します。14 で LBUS RDY-を検出、15 でデータをラッチし、16 で CPU へ VALIDIN-を返し、バスを終了します。

ローカルバスのライトサイクルを下図に示します。



クロックの番号に添って状態を説明します。

0~1: CPU の外部ライトサイクルの始まりです。通常 WRRDY-はインアクティブです。

2~4: LBUS のライトサイクルが確定し、WRRDY-がアクティブになります。

5: データ出力です。LBUS コントローラは、この時のデータをラッチします。CPU のサイクルはこれで終了しますが、実際のライトサイクルは、これ以降に行われます。

7: LBUS へのライトサイクルが開始します。

9~10: アクセスする資源によってこの間のクロックは異なります。

13~14: LBUS CLK の立ち上がりで LBUS RDY-がアクティブになることで LBUS のライトサイクルが終了します。(14)

8.4. 所要クローカー一覧

メモリアクセスの所要クローカー数を以下に示します。

8.4.1. SRAM のアクセス

サイクル種別	1st	2nd	7th	8th	total	備考
SingleRead	7 + n				7 + n	SWAIT[] = n
SingleWrite	6				6	
8 BurstRead	7 + n	1 + n	1 + n	1 + n	10 + 8*n	SWAIT[] = n
8 BurstWrite	6	1 + n	1 + n	1 + n	9 + 7 *n	Data Rate に依存(n=0 ~ 3)

- 補足：1.リードサイクルでは、n = SWAIT[]。  
 2.ライトサイクルでは、DDDDDDDD の時 n=0、  
 DxxxDxxxDxxxDxxxDxxxDxxxDxxxDxxx の時 n=3。

8.4.2. SDRAM のアクセス

サイクル種別	1st	2nd	7th	8th	total	備考
SingleRead	9(7)				9(7)	
SingleWrite	9(7)				9(7)	
8 BurstRead	9(7)	1	1	1	16(10)	
8 BurstWrite	9(7)	1 + n	1 + n	1 + n	12(10) + 7 *n	Data Rate に依存(n=0 ~ 3)

- 補足：1.ライトサイクルでは、DDDDDDDD の時 n=0、  
 DxxxDxxxDxxxDxxxDxxxDxxxDxxxDxxx の時 n=3。  
 2.オートプリチャージが ON の場合、最終サイクル後プリチャージ時間が加算されますが、最終サイクルから次サイクルの開始までに 2CLK 以上のアイドルサイクルがある場合、ペナルティはありません。  
 3.()内の数値はオートプリチャージの設定が OFF で、Raw アドレスがヒットした場合のもので、1<sup>st</sup>サイクルが 2clk 短縮されます。但し、ヒットしなかった場合、1<sup>st</sup>サイクルの直前にプリチャージサイクルが入りますので、2clk 加算されます。

8.4.3. ROM のアクセス

サイクル種別	#1	1st	2nd	7th	8th	#2	total	備考
SingleRead (16-Bit)	4 ~ 6	3+ 4*3				2	21 ~ 23	
SingleRead (32-Bit)	4 ~ 6	3+ 8*3				2	33 ~ 35	
SingleRead (48-Bit)								SingleRead (32-Bit) + SingleRead (16-Bit)
SingleRead (64-Bit)								SingleRead (32-Bit) + SingleRead (32-Bit)
4 BurstRead	4 ~ 6	6+ 16*3	6+ 16*3	6+ 16*3	6+ 16*3	2	222 ~ 224	

- 補足：1.#1 はローカルバスの開始に際し要す CPU のサイクルとブリッジのオーバヘッドです。  
 2.#2 はローカルバス上のデータを CPU に引き渡すサイクルです。



## 8.4.4. SYSTEM IO のアクセス

サイクル種別	#1	1st	2nd	7th	8th	#2	total	備考
SingleRead (8-Bit)	4~6	3+ 3*3				2	18~20	
SingleWrite (8-Bit)	6	3+ 3*3					6(18)	()は Lbus の終了までのサイクルの総 clk 数

補足：1.#1 はローカルバスの開始に際し要す CPU のサイクルとブリッジのオーバーヘッドです。

2.#2 はローカルバス上のデータを CPU に引き渡すサイクルです。

3.ライト時、CPU のサイクルは 6-clk で終了します。但し、ローカルバスアクセス中に再度ローカルバスまたは、G-Bus にアクセスした場合、そのサイクルは前のサイクルの終了まで待たされます。

## 8.4.5. GBUS のアクセス

サイクル種別	#1	1st	2nd	7th	8th	#2	total	備考
SingleRead (32-Bit)	4~6	3+ n*3				2	7~9 + n*3	
SingleRead (64-Bit)								SingleRead (32-Bit) + SingleRead (32-Bit)
8 BurstRead	4~6	6+ 2n*3	6+ 2n*3	6+ 2n*3	6+ 2n*3	2	54~56 + 16n*3	
SingleWrite (32-Bit)	6	3+ n*3					6 (9 + n*3)	
SingleWrite (64-Bit)								SingleWrite(32-Bit) + SingleWrite (32-Bit)
8 BurstWrite	6 + 7m	3+ n*3	3+ n*3	3+ n*3	3+ n*3		6 + 7m (30 + 7m + 8n*3)	m はライトデータレート DDDD..=1 ~ DxxxDxxx...=4 n は、GBUS サイクルの Wait 数

補足：1.#1 は GBUS の開始に際し要す CPU のサイクルとブリッジのオーバーヘッドです。

2.#2 は GBUS 上のデータを CPU に引き渡すサイクルです。

3.ライト時、CPU のサイクルは GBSU に先立ち終了します。total 中の数値は上段が CPU の所要クロック数、下段()内の数値が CPU のバスの開始から GBUS が終了するまでのクロック数です。GBUS のアクセスが終わる前に再度ローカルバスまたは、GBUS にアクセスした場合、そのサイクルは前のサイクルの終了まで待たされます。

## 9. ROM化プログラミング

本ボード用にROM化するプログラムを作成する場合、以下を参考にしてください。

### 9.1. 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とDRAMのリフレッシュ用のタイマ2を設定してください。

以下では、ボード固有の事項に関してのみ記述します。その他に関しては、CPUのマニュアル等を参照してください。

<プロセッサの内部初期化>	
Config Reg	... SW2の説明を参照
<SRAM ウェイトの初期化>	
[0xB8006000+offset].b <= 0	... SRAM Wait >> 0 ウェイト
<SDRAM コントローラの初期化>	
[0xB8005000+offset].b <= 2	... SDRAM tRCD >> 2 クロック
[0xB8005010+offset].b <= 2	... SDRAM tRP >> 2 クロック
[0xB8005020+offset].b <= 5	... SDRAM tRC >> 5 クロック
[0xB8005030+offset].b <= 6	... SDRAM tRRC >> 6 クロック
[0xB8005040+offset].b <= 4	... SDRAM tRAS >> 4 クロック
[0xB8005050+offset].b <= 2	... SDRAM tLAT >> 2 クロック
[0xB8005060+offset].b <= 2	... SDRAM tRSC >> 2 クロック
[0xB8005070+offset].b <= 1	... SDRAM AP >> オートリチャージ :OFF(任意)
[0xB8004000+offset].b <= 0	... SDRAM Mode Reg Set
<リフレッシュ用タイマの初期化>	
[0xB9005030+offset].b <= 0xb4	... タイマ2モード2(約15μ秒周期に設定)
[0xB9005020+offset].b <= 0x1f	... タイマ2下位カウンタ
[0xB9005020+offset].b <= 0x00	... タイマ2上位カウンタ

**【メモ】** バイトアクセスにおける Big Endian 時の offset は、+3 です。

#### 9.1.1. SRAMの初期化

SRAM はパリティビットを有していますので、起動後リードアクセスを行う前にパリティビットの初期化が必要です。パリティビットの初期化はCPUから通常に書き込みを行うことで自動的に行われますので、リード操作を行う前に全領域に対し一度書き込み操作を行ってください。

#### 9.1.2. ROMデータ配置

リモートモニタのROMは、4つのバンクに分けて使用していますが、通常は、SW3-3をONにして、連続した空間としてご使用ください。

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ビットのスイッチ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は、必ずROMのバンクを禁止(SW3-3-ON)してご使用ください。

## 10. GBUS 個別仕様

本章では、RTE-VR5500-CB での GBUS の使用状況を説明します。GBUS の一般的な仕様については「14 APPENDIX.C GBUS 共通仕様」を参照してください。

### 10.1. 概要

RTE-VR5500-CB での GBUS の信号線使用状況の概要を下表に示します。

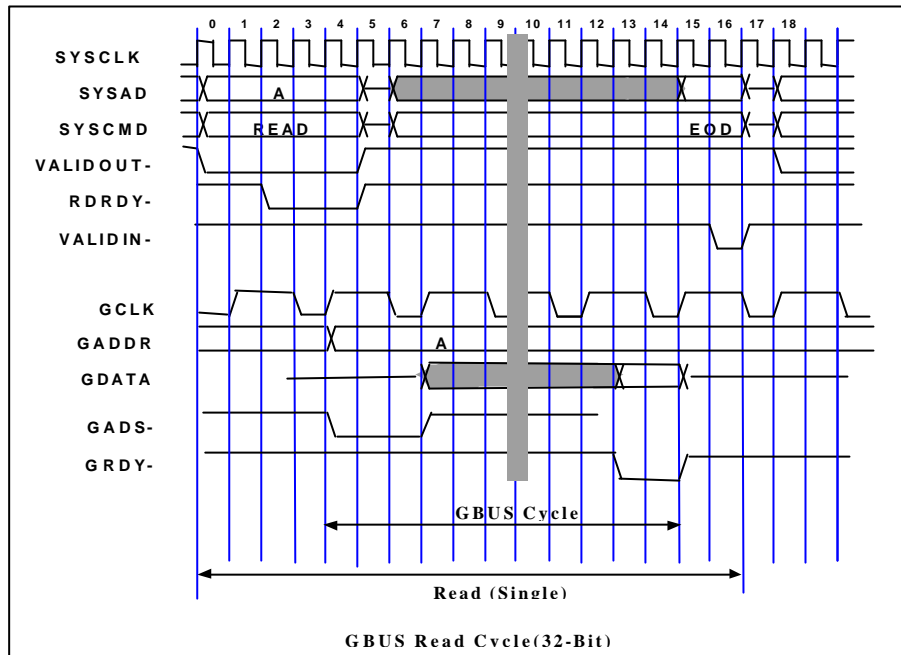
GBUS 信号名	機能	参照
GADDR[31:2]	アドレス線として使用。GADDR[26:31]は未接続。	
GDATA[31:0]	データ線として使用。リード・サイクル時は、SYSCLK で一度ラッチされたものが CPU に供給される。	
GCS-[6:0]	チップ・セレクト線として使用。	10.3
GCLK	CPU の SYSCLK を 2 分周したクロックを接続。	
GRESETI-	本ボード上で発生したリセット要求を出力。	7.1
GRESETO-	ボード上のリセット信号とオアして ColdReset*に接続。	7.1
GADS-, GREADY-, GBLAST-, GW/R-	バス制御信号として使用。	
GWAITI-	バス制御信号として使用。	
GBTERM-	バス制御信号として使用。	
GRD-, GWR-	GBUS の制御信号から生成した、RD-,WR-信号を接続。	
GHOLD-, GH LDA-	未接続	
GBREQ-	未接続	
GDMARQ-[3:0]	DMA リクエスト信号として使用。	7.4.10
GDMAAK-[3:0]	DMA アクノリッジ信号として使用。	7.4.10
GINTO-[3:0]	割込み要求信号として使用。	7.2
GINTI-[1:0]	GINTI0 と GINTI1 へは、それぞれ TIC (μPD71054) の OUT0 と OUT1 を接続。	
GETC[7:0]	未接続。	
GAHI_EN-	未接続	
GMOTHER_DETECT-	JGBUS 未接続時に GBUS へアクセスした場合のレディー生成に使用。	
GUSE_DIRECT_ACC-	未接続	
GCLK_LOW-	SW2-8 (BGCLK_LOW-)が、OFF の時 High、ON の時 Low。	5.5
GLOCK-[1:0]	未接続	

10.2. バス・サイクル

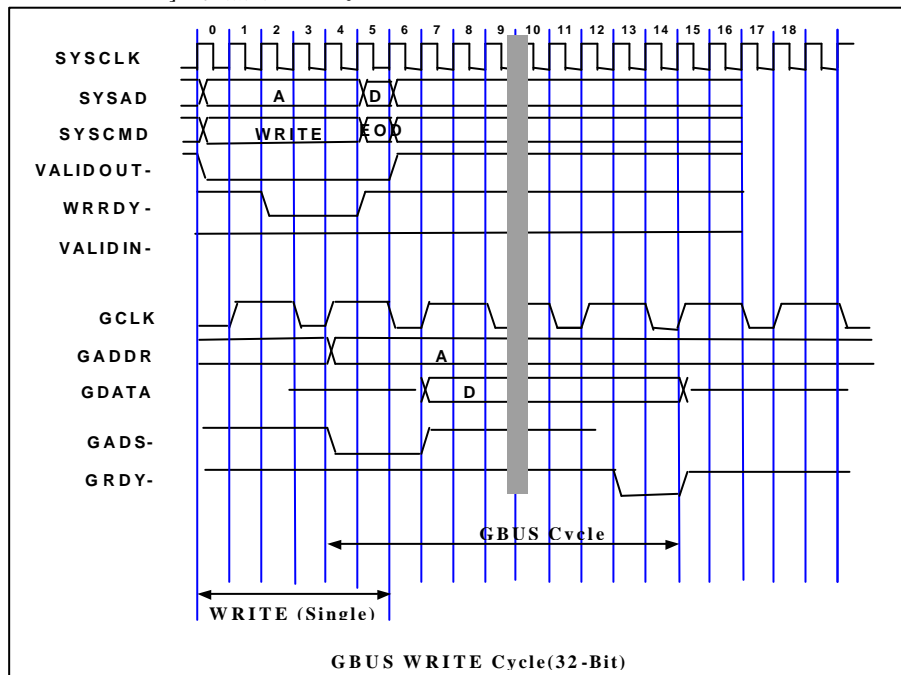
GBUS の GCLK へは、CPU のバス CLK を 3 分周したクロックを使用しています。また、GAHL\_EN-を未接続にしているため、GADDR[26:31]は未接続になっています。

GBUS へのアクセスは、シングルサイクル、バーストサイクルの両方に対応しています。

リード・サイクル (32Bit シングル) の様子を下図に示します。バーストサイクルは、[0 バースト・サイクル]を参照ください。



ライト・サイクル (32-Bit シングル) の様子を下図に示します。バーストサイクルは、[0 バースト・サイクル]を参照ください。



### 10.3. チップセレクト

本ボードの GBUS の各チップセレクトは SW2-2 の状態により 2 種の異なる空間が割り当てられています。以下では物理アドレスはアンキャッシュ領域を記しています。キャッシュ領域としてアクセスする場合は、0x20000000 をマイナスしてください。尚、本ボードはこれらのいずれの資源に対してもキャッシュ領域としてバーストアクセスすることが可能ですが、IO 空間や共有メモリ空間はアンキャッシュでご使用ください。

#### SW2-2 = OFF

GBUS 信号名	論理アドレス範囲	物理アドレス空間	備考(RTE-MB-A の資源)
GCS0-	0xB1000000 - 0xB1FFFFFF	0x11000000 - 0x11FFFFFF	(MB-SRAM)
GCS1-	0xB2000000 - 0xB2FFFFFF	0x12000000 - 0x12FFFFFF	FBOOT=OFF (MB-FLASH)
	0xBF000000 - 0xBFFFFFFF	0x1F000000 - 0x1FFFFFFF	FBOOT=ON (MB-FLASH)
GCS2-	0xB0400000 - 0xB07FFFFF	0x10400000 - 0x107FFFFF	(MB-Peri)
GCS3-	0xB3000000 - 0xB3FFFFFF	0x13000000 - 0x13FFFFFF	(MB-EXT-MEM)
GCS4-	0xB0C00000 - 0xB0FFFFFF	0x10C00000 - 0x10FFFFFF	(MB-EXT-IO)
GCS5-	0xB4000000 - 0xB7FFFFFF	0x14000000 - 0x17FFFFFF	(MB-PCI Bus-MEM)
GCS6-	0xB0800000 - 0xB0BFFFFF	0x10800000 - 0x10BFFFFF	(MB-PCI9080 Reg)
GCS7-	0xB0000000 - 0xB03FFFFF	0x10000000 - 0x103FFFFF	(MB-PCI Bus-I/O)

#### SW2-2 = ON : RTE-VR5432-CB 互換モードのマッピングです。

GBUS 信号名	論理アドレス範囲	物理アドレス空間	備考(RTE-MB-A の資源)
GCS0-	0xB0000000 - 0xB0FFFFFF	0x10000000 - 0x10FFFFFF	(MB-SRAM)
GCS1-	0xB1000000 - 0xB1FFFFFF	0x11000000 - 0x11FFFFFF	FBOOT=OFF (MB-FLASH)
	0xBF000000 - 0xBFFFFFFF	0x1F000000 - 0x1FFFFFFF	FBOOT=ON (MB-FLASH)
GCS2-	0xB2000000 - 0xB2FFFFFF	0x12000000 - 0x12FFFFFF	(MB-Peri)
GCS3-	0xB3000000 - 0xB3FFFFFF	0x13000000 - 0x13FFFFFF	(MB-EXT-MEM)
GCS4-	0xB4000000 - 0xB4FFFFFF	0x14000000 - 0x14FFFFFF	(MB-EXT-IO)
GCS5-	0xB5000000 - 0xB5FFFFFF	0x15000000 - 0x15FFFFFF	(MB-PCI Busp-MEM/IO)
GCS6-	0xB6000000 - 0xB6FFFFFF	0x16000000 - 0x16FFFFFF	(MB-PCI9080 Reg)

## 11. APPEDIX.A MULTI モニタ

MULTI用のモニタ ROM を使用して、ホストの 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。

### 11.1. ボードの設置

#### 11.1.1. RTE for Win32 のインストール

MULTI デバッガを使用する場合には、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。ソフトウェアのインストールとテストについては、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。

#### 11.1.2. SW1 の設定

SW1 は、汎用の入力ポートのスイッチですが、実装されている MULTI 用のモニタでは、以下の通り使用しています。正しく設定してご使用ください。

SW1 番号	1	2	ボーレート
設定	ON	ON	115200 baud
	OFF	ON	38400 baud
	ON	OFF	19200 baud
	OFF	OFF	9600 baud (出荷時の設定)

ボーレートの設定

SW1 番号	3	4	プロファイラ周期
設定	ON	ON	タイマを使用しない
	OFF	ON	200 Hz 5 ms
	ON	OFF	100 Hz 10 ms
	OFF	OFF	60 Hz 16.67ms (出荷時の設定)

プロファイラ周期の設定

SW1 番号	8	デバッグモード
設定	ON	7segLED をモニタが使用
	OFF	通常の使用状態 (出荷時の設定)

デバッグモードの設定

SW1-5~7 は、MULTI モニタでは使用していません。

### 11.2. Multi モニタ

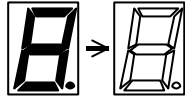
MULTI 用の ROM モニタは、32Bit BigEndian/LittleEndian、64Bit BigEndian/LittleEndian の 4 つ動作モードで実行可能です。モニタの動作モードは SW3-7:8 で設定します。

#### 11.2.1. 起動時の 7Seg-LED

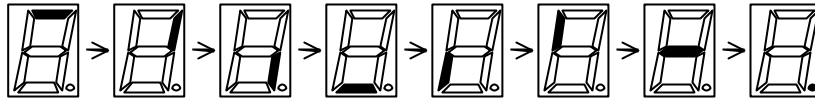
MULTI 用の ROM モニタは、ボードの電源を入れると 7Seg-LED が次のように動きます (黒い部分が点灯部分)。

1)7Seg-LED のチェック動作 (下図参照)

SW1-8 が OFF の場合 :



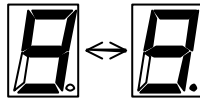
SW1-8 が ON 場合 :



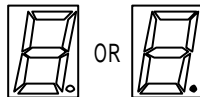
2)SRAM の簡易メモリチェックによる数字のカウンタ

SW1-8 が OFF の場合は行いません。

3)接続待ち状態 (プロファイラのタイマを停止している場合は、ドットの点滅なし)



4)接続状態 (ドットの点灯は、接続した時のドットの点灯状態が保持される)



### 11.2.2. モニタ・ワーク RAM

モニタでは、SRAM(Uncache)の先頭から 10000H ( 64KB ) をワーク用の RAM として使用 ( 予約 ) していますので、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、このイメージ領域も同様です。

### 11.2.3. 割込み

モニタでは INT0 割込みを使用しています。この割込みに対するベクタは、ブートストラップ (BEV=1)の設定で ROM の BFC0-0380H です。したがって、ベクタを書き換えることができません。そこでモニタでは、SRAM(cache)の 8000-0180H に確保した代替ベクタに 1 度分岐する方法で書き換えを可能にしています。

### 11.2.4. \_INIT\_SP の設定

モニタで \_INIT\_SP (スタック・ポインタの初期値) は、A003-FFFCH (SRAM の最上位) に設定されています (MULTI の環境で \_INIT\_SP を変更することもできます)。モニタでは、ユーザ・プログラムで設定したスタック領域を 32 バイト使用します。

### 11.2.5. タイマ割込み

タイマ割込みを禁止しますと、Multi のプロファイラ機能が使用できません (タイマ割込みの設定については『11.1.2 SW1 の設定』を参照)。

### 11.2.6. ハードウェアの初期化

ROM モニタでは、ボード上の資源に対し、直にアクセスできるように初期化を行っています。

### 11.2.7. 特殊命令

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コール機能で使用しています。

BREAK 命令 ( 0xxxxxxdH )

ユーザプログラム内では、ブレーク命令と解釈されるコードは使用しないでください。

### 11.3. RTE コマンド

MULTI デバッガでモニタと MIDAS サーバ (RTESERV) と接続すると 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

#### 11.3.1. 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。

#### 11.3.2. 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。

#### 11.3.3. VER

<書式>    VER

VER は、RTE 環境のバージョンを表示します。

#### 11.3.4. CACHEFLUSH

<書式>    CACHEFLUSH

CACHEFLUSH は、CPU のキャッシュ内容をフラッシュします。



### 11.3.5. SHOWTLB

<書式> SHOWTLB

SHOWTLB は、CPU の TLB 内容の一覧を表示します。

### 11.3.6. 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 0xBC000100  
BC000100: 1A

### 11.3.7. IOWRITE

<書式> OWRITE [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> OWRITE SHORT 0x30F0 0xBC00F000

## 12. APPEDIX.B PARTNER モニタ

PARTNER 用のモニタ ROM を使用して、ホストの PARTNER 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。

### 12.1. ボードの設置

#### 12.1.1. SW1 の設定

SW1 は、汎用の入力ポートのスイッチですが、実装されている PARTNER 用のモニタでは、以下の通り使用しています。正しく設定してご使用ください。

SW1 番号	1	2	ボーレート
設定	ON	ON	115200 baud
	OFF	ON	38400 baud
	ON	OFF	19200 baud
	OFF	OFF	9600 baud (出荷時の設定)

ボーレートの設定

SW1 番号	3	4	タイマ
設定	ON	ON	常時この状態でご使用ください。

SW1 番号	8	デバッグモード
設定	ON	7segLED をモニタが使用
	OFF	通常の使用状態 (出荷時の設定)

デバッグモードの設定

SW1-5 ~ 7 は、PARTNER では使用していません。

### 12.2. PARTNER モニタ

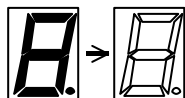
Partner 用の ROM モニタは、32Bit BigEndian/LittleEndian の 2 つ動作モードで実行可能です。モニタの動作モードは SW3-7:8 で設定します。

#### 12.2.1. 起動時の 7Seg-LED

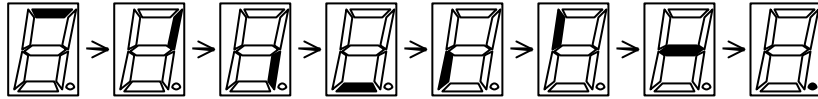
Partner 用の ROM モニタが実装されている場合、ボードの電源を入れると 7Seg-LED が次のように動きます (黒い部分が点灯部分)。

1) 7Seg-LED のチェック動作 (下図参照)

SW1-8 が OFF の場合 :



SW1-8 が ON 場合 :



2)RAM の簡易メモリチェックによる数字のカウント

SW1-8 が OFF の場合は行いません。

3)接続待ち状態



4)接続状態



### 12.2.2. ROM モニタ・ワーク RAM

ROM 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) していますので、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、このイメージ領域も同様です。

### 12.2.3. 割込み

モニタの通信および、強制ブレーク(ESC ボタン)で使用する割込みは、INT0 を使用します。この割込みに対するベクタは、ブートストラップ(BEV=1)の設定で ROM の BFC0-0380H です。したがって、ベクタを書き換えることができません。そこでモニタでは、SRAM(cache)の 8000-0180H に確保した代替ベクタに 1 度分岐する方法で書き換えを可能にしています。

### 12.2.4. INIT\_SP の設定

モニタのスタック・ポインタの初期値は、A003-FFF0H (SRAM の最上位) に設定されています。この値は、PARTNER のコンフィギュレーションファイルで定義する INIT\_SP コマンドで変更できます。モニタでは、ユーザ・プログラムで設定したスタック領域を 32 バイト使用します。

### 12.2.5. リモート接続

デバッガとの接続はシリアルです。SW1 の通信速度の設定と、RPTSETUP で通信路と通信速度の設定を正しく行ってください。

### 12.2.6. ハードウェアの初期化

ROM モニタでは、ボード上の資源に対し、直にアクセスできるように初期化を行っています。

### 12.2.7. 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コール機能で使用しています。

BREAK 命令 (0xxxxxxdH)

ユーザプログラム内では、ブレーク命令と解釈されるコードは使用しないでください。

## 13. APPEDIX.C 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。

### 13.1. 割込みライブラリ

VR5500 では、一般例外割込みをまとめてベクタが割当てられており、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になります。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をして、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。

#### 13.1.1. int InitlrqVect(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。

#### 13.1.2. int TermlrqVect(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。

#### 13.1.3. int SetlrqVect(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"VECT\_IPn"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。

#### 13.1.4. int GetlrqVect(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ないことを示します。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。

#### 13.1.5. 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)。

#### 13.1.6. 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)。

#### 13.1.7. void ei(void)

すべての割込みを許可します。

#### 13.1.8. void di(void)

NMI を除くすべての割込みを禁止します。

### 13.2. 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。

### 13.3. サンプルプログラム

ユーザ割り込みINT1(IP3)を使用して割り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_IP3, (int)IrqHusr ); /* set function vector table */
    SetIPnBit( SR_IP3 ); /* set IP3 bit */
    ei(); /* enable all interrupt */
    /* --- --- */
    .
    . /* process */
    .
    /* --- can not Break & Step --- */
    di(); /* disable all interrupt */
    ResIPnBit( SR_IP3 ); /* reset IP3 bit */
    SetIrqVect( VECT_IP3, 0); /* remove vector */
    TermIrqVect(); /* restore original vector code */
    ei(); /* enable all interrupt */
    /* --- --- */
    .
    .
    exit;
}

/** IrqHusr */
int IrqHusr( struct _irq_stack *istack )
{
    int to_monitor; /* flag */
    /* --- can not Break & Step --- */
    ResIPnBit( SR_IP3 ); /* Disable INT1 interrupt */
    ei(); /* Enable multiple interrupt */
    /* --- --- */
    .
    . /* interrupt operation */
    .
    /* --- can not Break & Step --- */
    di();
    SetIPnBit( SR_IP3 ); /* Enable INT1 interrupt */
    If( to_monitor != 0 ) {
        return 1; /* request monitor handler */
    }
    return 0; /* normal end */
}
```

## 14. APPEDIX.C GBUS 共通仕様

ここでは、ボードの品種に依存しないGBUSの仕様について説明します。

### 14.1. 用語

この章で用いる用語について説明します。

#### 14.1.1. CPUボードとマザー・ボード

RTE-CBシリーズのボードのことをCPUボード、CPUボードのGBUSに接続する弊社製のボードをマザー・ボードと呼ぶことにします。

#### 14.1.2. バス・サイクル、マイクロ・サイクル

GBUSは、バースト・アクセス可能な一般的なバスです。

バス・サイクルとは、アクセスがバーストした場合も含み、一連のサイクルが終了するまでの区切りのことを示すものとします(1回のGADSのアサートが必要とされる区切り)。

バス・サイクルには、シングル・サイクルとバースト・サイクルがあります。シングル・サイクルとは、1回のデータ転送しか発生しないバス・サイクルのことを示します。バースト・サイクルとは、複数回のデータの転送が発生するバス・サイクルのことを示します。

また、バースト・サイクルのデータ転送1回ごとのサイクルをマイクロ・サイクルと呼ぶことにします。

### 14.2. 信号

GBUSの信号の内容を下表に示します。GBUSの各信号の入出力方向は、マザー・ボード側を基準にして記述します。つまり、「入力」とある場合、CPUボードから出力されマザー・ボードへ入力される信号を示します(信号名にもこの基準が適用されています)。

下表で「双方向」と記載されている信号はバス・サイクルの状態では信号の向きが切り替わることを示します。

また、「入力/出力」と記載されている信号はバス・マスタがCPUボードなのかマザー・ボードなのかによって信号の向きが切り替わる事を示し、前に書かれているのがCPUボードがバス・マスタの時の信号の方向、後に書かれているのがマザー・ボードがバス・マスタの時の信号の方向を示します。

GBUSの信号は、+5VのTTLレベルです。また、マザー・ボードのバスは常にリトル・エンディアンです。

信号名	入出力	機能
GCLK	入力	<ul style="list-style-type: none"> <li>GBUSの同期クロック。最高周波数は33.33MHz。最低周波数は10.0MHz。GBUSはこのクロックの立ち上がりで同期して動作する。</li> <li>マザー・ボード上では、+5VとGNDに対してそれぞれ330Ωでターミネーションされるため、CPUボードの回路はこれをドライブできなければならない。</li> <li>GCLKが16.67MHzより周波数が低い場合、GCLK_LOWをLowにする。これにより、マザー・ボードはウェイト数の調整を行える。</li> <li>この信号は、PLL(Phase Lock Loop)によるゼロ・ディレイ・バッファが使用される場合があるので、GCLKの周波数を変更した場合は、PLLのロックのために周波数変更後最低でも1m秒の間はマザー・ボードにアクセスしてはならない。</li> </ul>
GRESETI-	入力	<ul style="list-style-type: none"> <li>GBUSのリセット信号。CPUボード上でリセットが発生した時に、この信号をLowにする。マザー・ボードはこの信号によってリセットされる(マザー・ボード上の他の要因によってマザー・ボードがリセットされる場合もある)。</li> </ul>
GRESETO-	出力	<ul style="list-style-type: none"> <li>マザー・ボードのリセットが発生した場合、Lowになる信号。</li> <li>マザー・ボードでは、マザー・ボード上で発生したリセットとGRESETI-をORしたものをGRESETO-とする。したがってCPUボードは、GRESETI-とGRESETO-をORした信号で、CPUボード上の回路をリセットする(GRESETI-とGRESETO-をORするのは、マザー・ボードが接続されていない時のため)。</li> </ul>

信号名	入出力	機能
GADDR[31:2]	入力/出力	<ul style="list-style-type: none"> <li>・GBUSのアドレス信号。サイクル中は常に有効な値でドライブされる。</li> <li>・GADDR[31]は、CPUがバス・マスタの場合、マザー・ボード上で無視される。</li> <li>・下位アドレスのA1,A0は、バイト・イネーブル信号を用いる。</li> <li>・GAHL_EN-信号により、CPUボードからのGADDR[31:26]を0として扱うようにできる。</li> <li>・バス・マスタがマザー・ボードの場合、GADDR[25]が0の時、マザー・ボード上の資源が、GADDR[25]が1の時CPUボード上の資源が選択されていることを示します。</li> </ul>
GBEN-[3:0]	入力/出力	<ul style="list-style-type: none"> <li>・GBUSのバイト・イネーブル信号。サイクル中は常に有効な値でドライブされる。</li> <li>・それぞれ、GBEN0-がGDATA[7:0]、GBEN1-がGDATA[15:8]、GBEN2-がGDATA[23:16]、GBEN3-がGDATA[31:24]の各バイト・レーンに対応し、GBENx-がLowの時に対応するバイト・レーンが有効。</li> </ul>
GDATA[31:0]	双方向	<ul style="list-style-type: none"> <li>・GBUSのバス・データ信号。</li> <li>・マザー・ボード上で10KΩでプルアップされる。</li> <li>・この信号の方向は、GW/R-により決定する。</li> </ul>
GADS-	入力/出力	<ul style="list-style-type: none"> <li>・GBUSのアドレス・ストロブ信号。GCLKの立ち上がりでこの信号がLowにサンプルされると、バス・サイクルの開始を示す。</li> <li>・マザー・ボードは、いずれのチップ・セレクト信号(GCS-[7:0])もアクティブでない場合GADS-は無視する。</li> </ul>
GREADY-	出力/入力	<ul style="list-style-type: none"> <li>・GBUSのレディー信号。マイクロ・サイクル中にGCLKの立ち上がりでこの信号がLow、GWAITI-がHighにサンプルされると、マイクロ・サイクルの終了を示す。</li> <li>・CPUボードからマザー・ボードに対するアクセス時のタイムアウト・レディは、マザー・ボードが生成する。これは、GREADY-信号がぶつかってしまうのを回避するためである。</li> </ul>
GWAITI-	入力	<ul style="list-style-type: none"> <li>・ウェイト要求信号。GCLKの立ち上がりでサンプルされる。</li> <li>・CPUボード側の都合で、少ないWait数のサイクルに対応できない場合、CPUボードはGREADY-のサンプル・タイミングで、GWAITI-をLowにサンプルされるようにすることで、仮にそのタイミングでGREADY-がLowであったとしても、それをマザー・ボードにレディーとして扱わせないことができる。通常、CPUボードがゼロWaitバーストに対応できない場合などに使用する(「14.6.3 GWAITI-」参照)。</li> <li>・この信号は、CPUボードがバス・マスタのサイクルのみ有効。</li> </ul>
GBLAST-	入力/出力	<ul style="list-style-type: none"> <li>・バス・サイクル終了通知信号。GCLKの立ち上がりでサンプルされる。</li> <li>・バス・サイクルを終了するマイクロ・サイクルの開始時から、バス・マスタがLowにアサートします。</li> <li>・GBLAST-がLow、GREADY-がLow、GWAITI-がHighがGCLKの立ち上がりでサンプルされると、バス・サイクルが終了します。</li> </ul>
GBTERM-	出力/入力	<ul style="list-style-type: none"> <li>・バス・サイクル終了要求信号。GCLKの立ち上がりでサンプルされる。</li> <li>・アクセスされている側が、バス・サイクルの終了を要求する場合、GREADY-信号と共にGBTERM-信号をLowにする。バス・マスタは、GREADY-がLowとしてサンプルした時、GBTERM-もLowとサンプルした場合、GBLAST-をアサートしていても、バス・サイクルを一旦終了させ、改めてGADS-をアサートしてバス・サイクルを開始しなければならない。GBTERM-のアサートはGREADY-のアサートと同時になければならない。</li> <li>・この信号は、アクセスされている側が、バースト・サイクルに対応していなかったり、対応しているバースト回数を越えるバースト・サイクルを要求された場合に、バス・サイクルを終了させるために使用する。</li> </ul>
GW/R-	入力/出力	<ul style="list-style-type: none"> <li>・Write/Read 信号。データ・バスの方向を示す。バス・サイクル中、常に有効な値でドライブされる。</li> <li>・この信号はバス・マスタにとってのデータ・バスの方向を示す。</li> </ul>
GCS-[7:0]	入力	<ul style="list-style-type: none"> <li>・チップ・セレクト信号。バス・サイクル中、常に有効な値がドライブされる。</li> <li>・CPUボードがバス・マスタの時に、マザー・ボード上の資源を指定するために該当するチップ・セレクト信号をアクティブにする。</li> <li>・各チップ・セレクト信号は、メモリ/I/O空間の別、空間の広さなどに規定がある(「0・GCS-[7:0]の割り付け」参照)。</li> </ul>

信号名	入出力	機能
GRD-	入力	<ul style="list-style-type: none"> <li>・リード・タイミング信号。CPU ボードがバス・マスタの時にアサートされる。</li> <li>・この信号はマザー・ボードでは使用しない。</li> <li>・通常、CPU の RD-コマンド信号がある場合は、その信号が接続される。</li> </ul>
GWR-	入力	<ul style="list-style-type: none"> <li>・ライト・タイミング信号。CPU ボードがバス・マスタの時にアサートされる。</li> <li>・この信号はマザー・ボードでは使用しない。</li> <li>・通常、CPU の WR-コマンド信号がある場合は、その信号が接続される。</li> </ul>
GHOLD-	出力	<ul style="list-style-type: none"> <li>・バス・ホールド要求信号。</li> <li>・マザー・ボードが CPU ボード上の資源にアクセスする場合、Low にアサートしバス権を要求する。</li> <li>・GUSE_DIRECT_ACC-信号が High の場合、CPU ボード側にマザー・ボードからアクセス可能な資源がないことを示し、この場合 CPU ボードは GHOLD-に対応する必要はない。</li> </ul>
GHLDA-	入力	<ul style="list-style-type: none"> <li>・バス・ホールド応答信号。</li> <li>・CPU ボードがマザー・ボードに GBUS のバス権を渡したことを示す信号で、その時 Low にアサートされる。</li> <li>・GUSE_DIRECT_ACC-信号を High にしている CPU ボードは、この信号を未接続にできる。</li> </ul>
GBREQ-	入力	<ul style="list-style-type: none"> <li>・バス権返還要求信号。</li> <li>・GHLDA-が Low にアサートし、マザー・ボードにバス権を渡している間に、CPU ボードがバス権を必要とした場合に GBREQ-を Low にアサートする。</li> <li>・GBREQ-が Low にアサートされた時、マザー・ボードがバス・サイクル中だった場合、次のマイクロ・サイクルで GBLAST-をアサートして、次のマイクロ・サイクルでバス・サイクルを終了し、GHOLD-をデアサートしなければならない。</li> <li>・GBREQ-は、マザー・ボードがバス・マスタのバス・サイクルのバースト回数が多い場合や、リフレッシュ・サイクルなどの優先順位の高いバス・サイクルが CPU ボード上で保留されている場合など、一旦バス権を CPU ボードに返させたい場合に使用する。</li> </ul>
GDMARQ-[3:0]	出力	<ul style="list-style-type: none"> <li>・DMA 要求信号。サポートされる DMA は、2 サイクル DMA のみで、フライバイ DMA はサポートされない。</li> <li>・マザー・ボード上で DMA 要求が発生した場合、Low にアサートする。</li> <li>・CPU ボードは 4 本全ての DMA をサポートしなければならないが、同時に起動できる DMA の数、および GDMAAK-信号が対応できる本数については、CPU ボードに依存する。</li> <li>・CPU ボードは、4 本全ての GDMARQ-に対して GDMAAK-の対応が取れない場合は、DMAAK-[3:2]に優先して DMAAK 信号を割り当てる。</li> </ul>
GDMAAK-[3:0]	入力	<ul style="list-style-type: none"> <li>・DMA 応答信号。</li> <li>・マザー・ボードからの DMA 要求に応答する場合に Low にアサートする。</li> <li>・CPU ボードは、4 本全ての GDMARQ-に対して GDMAAK-の対応が取れない場合は、DMAAK-[3:2]に優先して DMAAK 信号を割り当てる。</li> <li>・マザー・ボードは、GDMAAK-信号がなくとも動作するように設計されている。</li> </ul>
GINTO-[3:0]	出力	<ul style="list-style-type: none"> <li>・割込み要求信号。</li> <li>・GINTO0-は、レベル・センシティブとして使用可能。</li> <li>・GINTO-[3:1]は、レベル・センシティブとエッジ・センシティブのどちらで使用可能かは、CPU ボードに依存する (CPU に直結される場合があるため)。マザー・ボードはどちらにでも対応可能なようになっている。</li> <li>・Low レベル時、もしくは立ち下がりエッジで割込み発生を示す。</li> </ul>
GINTI-[1:0]	入力	<ul style="list-style-type: none"> <li>・割込み要求信号</li> <li>・CPU ボード上の割込みを、他のマザー・ボード上の割込みと合成して GINTO-[3:0]に戻すために設けられた割込み信号。</li> <li>・通常は CPU ボード上の TIC( <math>\mu</math>PD71054 )の OUT0 と OUT1 が接続される。マザー・ボードは、この割込み信号に対して、センシティブの種類やポラリティについて、プログラマブルになっている。</li> </ul>



信号名	入出力	機能
GETC[7:0]		<ul style="list-style-type: none"> <li>・ CPU ボード依存信号。</li> <li>・ 信号の方向や信号の内容まで含めて、GETC[7:0]の内容については CPU ボードが決定する。CPU ボードは特別な目的の信号をマザー・ボードとやり取りする場合は、この信号を用いる。</li> </ul>
GAHI_EN-	入力	<ul style="list-style-type: none"> <li>・ アドレス上位有効信号。</li> <li>・ この信号が Low の時、CPU ボードがバス・マスタの場合、CPU ボードが GADDR[31:26]に有効な値をドライブしていることを示す。この信号が High の場合、CPU ボードが GADDR[31:26]に有効な値をドライブしていないことを示し、マザー・ボード上の回路は、GADDR[31:26]が全て Low として処理する。</li> </ul>
GMOTHER_DETECT-	出力	<ul style="list-style-type: none"> <li>・ マザー・ボード検出信号。</li> <li>・ この信号は、CPU ボード上でプルアップされ、マザー・ボード上で GND に接続される。マザー・ボードが接続されている事を CPU ボード側で判断しなければならない場合にこの信号を使用する。例えば、CPU ボードのタイムオーバ・レディ生成回路。</li> </ul>
GUSE_DIRECT_ACC-	入力	<ul style="list-style-type: none"> <li>・ この信号が Low の時、CPU ボード側にマザー・ボードからアクセス可能な資源が存在することを示す。</li> </ul>
GCLK_LOW-	入力	<ul style="list-style-type: none"> <li>・ この信号が Low の時、GCLK の周波数が 16.67MHz 以下であることを示す。High の場合は、GCLK の周波数が 16.67MHz ~ 33.33MHz であることを示す。</li> <li>・ マザー・ボード上の回路は、この信号を使用して、マザー・ボード上の資源へのアクセスの際のウェイト数を決定する。</li> </ul>
GBLOCK-[1:0]	入力	<ul style="list-style-type: none"> <li>・ バス・ロック信号。バス・サイクル中と、ロックするバス・サイクル間で有効でなければなりません。</li> <li>・ CPU からバス・ロック信号が出力されている場合、この端子を使用してバス・ロック信号をマザー・ボードに接続する。</li> <li>・ GBLOCK0-信号は、GCS0-の空間に対して有効。GBLOCK1-は、GCS5-と GCS7-の空間に有効。</li> </ul>
+5V	出力	<ul style="list-style-type: none"> <li>・ 電源。+5V±5%をマザー・ボードから CPU ボードへ供給する。</li> </ul>
+12V	出力	<ul style="list-style-type: none"> <li>・ 電源。+12V±10%をマザー・ボードから CPU ボードへ供給する。ただし、CPU ボードが+12Vを必要としていなければマザー・ボードは+12Vを供給する必要はない。</li> </ul>

## 14.3. ピン配置

下表に GBUS のピン配置を示します。Reserve は予約済みのピンを、N/C は未接続のピンを示します。

番号	信号名	番号	信号名	番号	信号名	番号	信号名
1	+12V	2	+12V	3	GND	4	+5V
5	GADDR2	6	GADDR3	7	GADDR4	8	GADDR5
9	GADDR6	10	GADDR7	11	GND	12	+5V
13	GADDR8	14	GADDR9	15	GADDR10	16	GADDR11
17	GADDR12	18	GADDR13	19	GADDR14	20	GADDR15
21	GND	22	+5V	23	GADDR16	24	GADDR17
25	GADDR18	26	GADDR19	27	GADDR20	28	GADDR21
29	GADDR22	30	GADDR23	31	GND	32	+5V
33	GADDR24	34	GADDR25	35	GADDR26	36	GADDR27
37	GADDR28	38	GADDR29	39	GADDR30	40	GADDR31
41	GND	42	+5V	43	GBEN3-	44	GBEN2-
45	GBEN1-	46	GBEN0-	47	GND	48	+5V
49	GDATA31	50	GDATA30	51	GDATA29	52	GDATA28
53	GDATA27	54	GDATA26	55	GDATA25	56	GDATA24
57	GND	58	+5V	59	GDATA23	60	GDATA22
61	GDATA21	62	GDATA20	63	GDATA19	64	GDATA18
65	GDATA17	66	GDATA16	67	GND	68	+5V
69	GDATA15	70	GDATA14	71	GDATA13	72	GDATA12
73	GDATA11	74	GDATA10	75	GDATA9	76	GDATA8
77	GND	78	+5V	79	GDATA7	80	GDATA6
81	GDATA5	82	GDATA4	83	GDATA3	84	GDATA2
85	GDATA1	86	GDATA0	87	GND	88	+5V
89	GND	90	GW/R-	91	GBTERM-	92	GREADY-
93	GRESETI-	94	GADS-	95	GBLAST-	96	GWAITI-
97	GND	98	GCLK	99	GND	100	+5V
101	GCS0-	102	GCS1-	103	GCS2-	104	GCS3-
105	GCS4-	106	GCS5-	107	GCS6-	108	GCS7-
109	Reserve	110	Reserve	111	Reserve	112	Reserve
113	GRD-	114	GWR-	115	GND	116	+5V
117	GHOLD-	118	GHLDA-	119	GBREQ-	120	N/C
121	GDMARQ0-	122	GDMARQ1-	123	GDMARQ2-	124	GDMARQ3-
125	GDMAAK0-	126	GDMAAK1-	127	GDMAAK2-	128	GDMAAK3-
129	Reserve	130	Reserve	131	Reserve	132	Reserve
133	GND	134	+5V	135	GINTO0-	136	GINTO1-
137	GINTO2-	138	GINTO3-	139	GINTI0-	140	GINTI1-
141	GETC0	142	GETC1	143	GETC2	144	GETC3
145	GETC4	146	GETC5	147	GETC6	148	GETC7
149	Reserve	150	Reserve	151	GAHI_EN-	152	GMOTHER_DETECT
153	GND	154	+5V	155	GUSE_DIRECT_AC C-	156	GCLK_LOW-
157	GRESETO-	158	GBLOCK0-	159	GBLOCK1-	160	N/C
161	N/C	162	N/C	163	N/C	164	N/C
165	N/C	166	N/C	167	N/C	168	N/C
169	N/C	170	N/C	171	N/C	172	N/C
173	N/C	174	N/C	175	N/C	176	N/C
177	GND	178	+5V	179	+12V	180	+12V

使用するコネクタは下記のものです。

CPU ボード側コネクタ  
マザー・ボード側コネクタ (ストレート)  
マザー・ボード側コネクタ (Lアングル)

ケル株式会社製 8817-180-170L  
ケル株式会社製 8807-180-170S  
ケル株式会社製 8807-180-170L

#### 14.4. 未使用端子の処理

GBUS のマザー・ボードに対しての入力信号で使用しない信号は、マザー・ボード上でプルアップ/ダウンの処理が行われているため、CPU ボード上で未接続にすることができます。未接続にすることが可能な信号と、未接続時のために行われているマザー・ボード上の処理を下表に示します。

信号名	処理内容
GADDR[31:26]	・ GADDR[31:26]を使用しない場合は、GAHI_EN-信号を High もしくは未接続にすることにより、GADDR[31:26]を未接続にすることができます。この場合、CPU がバス・マスタのときマザー・ボード上では GADDR[31:26]は全ビットが 0 として扱われる。
GWAITI-	・ プルアップ処理が行われている。
GBLAST-	・ プルアップ処理が行われている。
GBTERM-	・ プルアップ処理が行われている。
GCS-[7:0]	・ プルアップ処理が行われている。
GHLDA-	・ プルアップ処理が行われている。
GBREQ-	・ プルアップ処理が行われている。
GDMAAK-[3:0]	・ プルアップ処理が行われている。
GINTI-[1:0]	・ プルアップ処理が行われている。
GAHI_EN-	・ プルアップ処理が行われている。
GUSE_DIRECT_ACC-	・ プルアップ処理が行われている。
GCLK_LOW-	・ プルアップ処理が行われている。
GBLOCK-[1:0]	・ プルアップ処理が行われている。

#### 14.5. GCS-[7:0]の割り付け

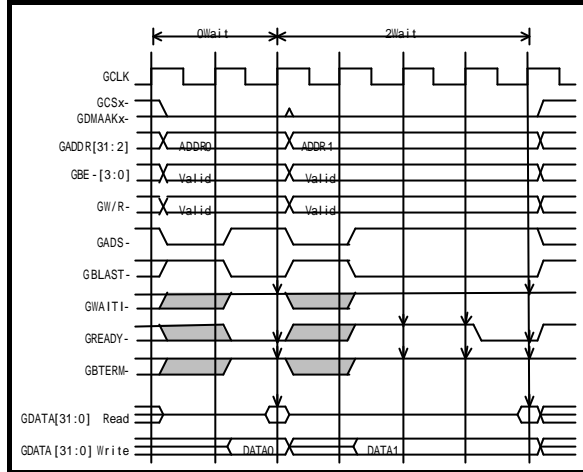
チップセレクト信号 (GCS-[7:0]) の割り付けを下表に示します。全ての空間がバースト・サイクルによるアクセスが可能です。下表の推奨空間に I/O と記載されている空間は、CPU に I/O 空間がある場合は、I/O 空間に割り付けることを推奨していることを示します。また最少範囲とは、CPU ボードは該当チップセレクトの空間に、最低でも最少範囲が示す領域を割り当てなければならないことを示します。最大範囲に記載がある場合は、CPU ボードのアドレス範囲に余裕がある場合、最大範囲が示す領域まで割り当てることが可能なことを示します。

信号名	推奨空間	最少範囲	最大範囲	備考
GCS0-	メモリ	1Mbyte		GLOCK0-によりバス・ロック可能
GCS1-	メモリ	2Mbyte		マザー・ボードでは、この空間にフラッシュ ROM を配置するので、スイッチの切り替えなどで CPU ボード上の UV-EPROM の代わりにこの空間からも Boot できるようにする。
GCS2-	I/O	64Kbyte		
GCS3-	メモリ	64Kbyte	16Mbyte	
GCS4-	I/O	64Kbyte	16Mbyte	
GCS5-	メモリ	1Mbyte	2Gbyte	GLOCK1-によりバス・ロック可能
GCS6-	I/O	512byte		
GCS7-	I/O	64Kbyte	2Gbyte	GLOCK1-によりバス・ロック可能

## 14.6. バス・サイクル

### 14.6.1. シングル・サイクル

GBWAITIおよびGBTERM-が常にインアクティブな場合で、CPU ボードがバス・マスタの場合のシングル・サイクルの様子を下図に示します。マザー・ボードがバス・マスタの場合、GCSx-、GDMAAK-、GWAITI-の各信号がなくなります。



### 14.6.2. バースト・サイクル

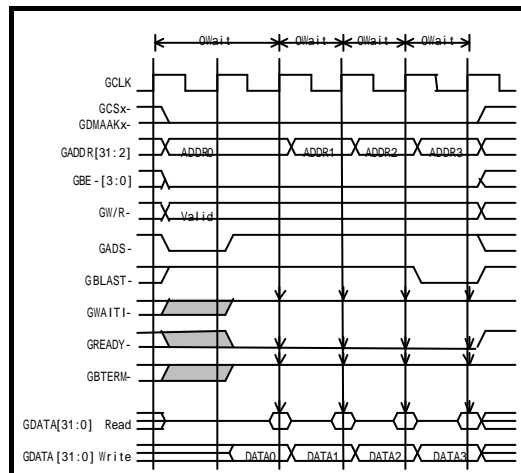
バースト・サイクルでは、次のルールがあります。

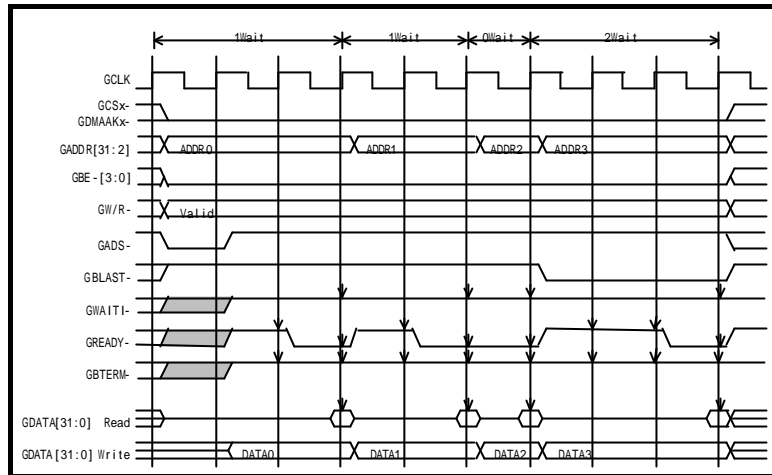
- ・ GBUS のスペックとしては、バースト・サイクル中のアドレスの順番は問いません。ただし、アクセス対象によっては、アドレス順が規定されてしまうことがあります。
- ・ バースト・サイクル中は GBE-[3:0]は全てアクティブでなければなりません。

バースト回数（マイクロ・サイクルの数）に制限はありません。アクセス対象側でバースト回数の制限がある場合は、GBTERM-信号を用いてバーストの中断を要求します（「0

- ・ GBTERM-」参照）。

GBWAITIおよびGBTERM-が常にインアクティブな場合で、CPU ボードがバス・マスタの場合のバースト・サイクルの様子を下図に示します。マザー・ボードがバス・マスタの場合、GCSx-、GDMAAK-、GWAITI-の各信号がなくなります。





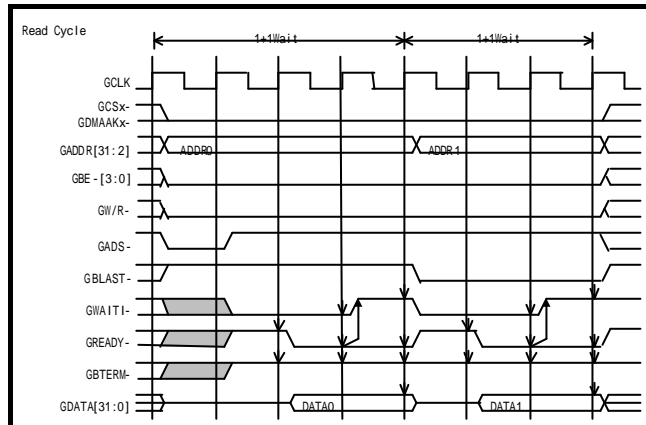
**14.6.3. GWAITI-**

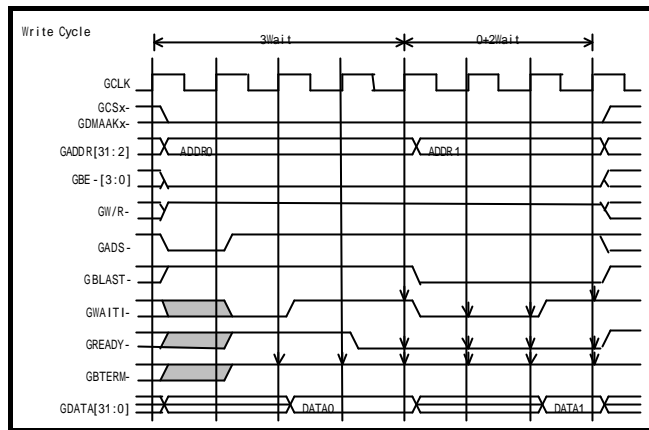
GBWAITI-信号は、CPU ボードがバス・マスタのサイクルで次のような場合に使用できます。

- ・ リード・サイクル時にタイミング的な問題でデータのサンプルができないため、特定クロック数分データのサンプリングを遅らせたい場合。
- ・ ライト・サイクルのバースト・サイクルで、マイクロ・サイクルが終了後すぐに次のマイクロ・サイクルのためのデータ準備ができず、特定クロック数分アクセス対象を待たせたい場合。

言い換えると、リード・サイクルとライト・サイクルで役割は入れ代わりますが、GREADY-とGWAITI-はデータ送信レディーとデータ受信レディーの働きをします。

GWAITI-信号によりウェイトが入っている様子を下図に示します。



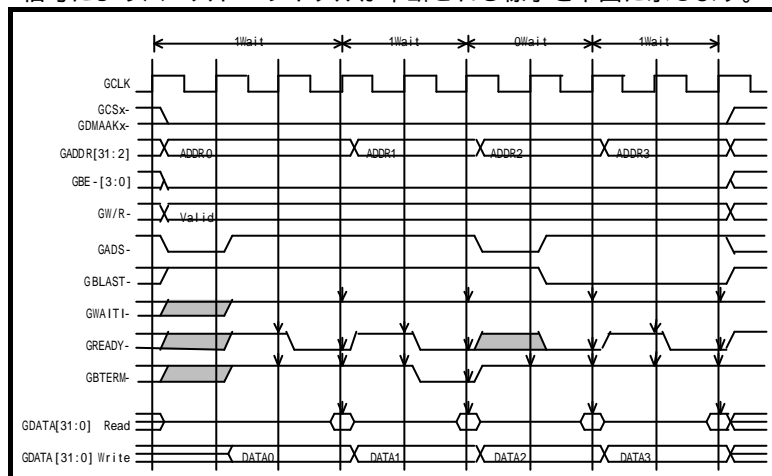


**14.6.4. GBTERM-**

GBTERM-信号が GREADY-信号と共にアクティブになると、バス・マスタは現在のマイクロ・サイクルを最後にバス・サイクルを終了させ、バースト・サイクルの続きは改めて GADS-をアクティブにしてサイクルを始めます。

GBTERM-信号は、アクセス対象がバースト・サイクルに対応していない場合や、対応バースト回数を越えてアクセスされた場合などにアクティブにします。また、GREADY-信号をアクティブにせずに GBTERM-信号のみをアクティブにすることは禁止されています。

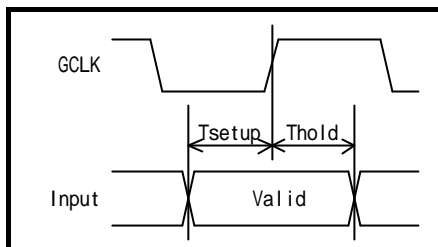
GBTERM-信号によりバースト・サイクルが中断される様子を下図に示します。



## 14.7. タイミング

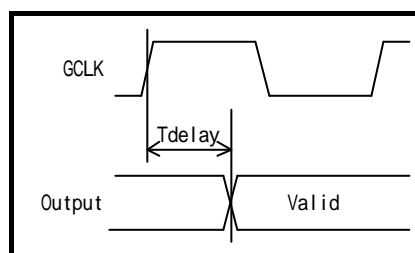
この章では、弊社のマザー・ボードにおけるタイミングについて記述します。CPU ボードはこのタイミングを満たすように設計されています。

### 14.7.1. セットアップ・タイム



信号名	Tsetup Min (nS)	Thold Min (nS)
GADDR[31:2]	12	0
GBEN-[3:0]	8	0
GDATA[31:0]	7	0
GADS-	14	0
GREADY-	9	1
GWAITI-	14	0
GBLAST-	8	0
GBTERM-	8	1
GW/R-	10	0
GCS-[7:0]	14	0
GBREQ-	15	0
GDMAAK-[3:0]	6	0
GLOCK-[1:0]	12	0

### 14.7.2. デイレイ・タイム



信号名	Tdelay MAX(nS)
GADDR[31:2]	21
GBEN-[3:0]	17
GDATA[31:0]	21
GADS-	15
GREADY-	15
GBLAST-	17
GBTERM-	16
GW/R-	15

- Memo -

RTE-VR5500-CB ユーザース・マニュアル

M0A1MNL01

*Midas lab*