

RTE-V830-PC

ユーザース・マニュアル

Midas lab

改訂履歴

実施日	Revision	page	内容
1995年11月28日	0.9		暫定版
1995年12月25日	0.91	6,7,... 9,10	SW1 と SW2 の記述が逆であった誤記訂正 SW1(1-2,3-4)の設定表記の誤記訂正
1996年2月5日	0.92	14	6.1章,6.2章のレジスタ表中の誤記訂正
1996年9月12日	0.93	15	PIO Port0 0-2bit のデータ値の誤記訂正 (‘ 0’ -> ‘ 1’) 12章、13章の追加
1996年9月12日	1.00		正式初版
1996年10月8日	1.01	P.19	JEXT コネクタ信号の表中、INT-の機能表記にあった 誤記訂正 (バッファ後の..... -> 割込みベクタ 11)

目次

1. はじめに	4
1.1. マニュアル表記について	4
2. 特徴と機能	5
3. ボードの構成	6
3.1. リセット・スイッチ (SWRESET)	6
3.2. 電源コネクタ (JPOWER)	6
3.3. スイッチ 1 (SW1)	6
3.4. スイッチ 2 (SW2)	7
3.5. LED	7
3.6. テストピン (TP)	7
3.7. シリアル・コネクタ (JSIO)	7
3.8. CPU テストピン (J1)	8
3.9. クロック・ソケット (OSC1)	8
3.10. DRAM-SIMM ソケット	8
3.11. ROM ソケット	8
4. インストールと使用方法	9
4.1. ボードの設定.....	9
4.2. ISA バスに実装する場合	10
4.3. ボード単体で使用する場合.....	10
5. ハードウェア・リファレンス	12
5.1. メモリ・マップ	12
5.2. I Oマップ	13
6. SYSTEM-IO	14
6.1. UART/TIMER(SCC2691)	14
6.2. PIO (UPD71055)	14
6.3. その他のポート	18
7. JEXT バス仕様	19
8. その他のCPU 資源	21
8.1. RESET-	21
8.2. NMI-.....	21
9. MULTI モニタ	22
9.1. モニタ・ワーク RAM	22
9.2. 割り込み.....	22

9.3. _INIT_SP の設定	22
9.4. リモート接続	22
10. RTE コマンド	23
10.1. HELP(?)	23
10.2. INIT	23
10.3. VER	23
10.4. INB,INH,INW	23
10.5. OUTB,OUTH,OUTW	24
10.6. DCTR コマンド	24
10.7. ITCR コマンド	24
10.8. PLLCR コマンド	24
10.9. CMCR コマンド	24
11. DRAM のタイミング	25
11.1. DRAM インターフェース概要	25
11.2. 信号の説明	25
11.3. 32 ビット・バスモード (シングル・リード、ノーマル)	26
11.4. 32 ビット・バスモード (シングル・リード、ヒット)	27
11.5. 32 ビット・バスモード (シングル・リード、ノーヒット)	28
11.6. 32 ビット・バスモード (シングル・ライト、ノーマル)	29
11.7. 32 ビット・バスモード (シングル・ライト、ヒット)	30
11.8. 32 ビット・バスモード (シングル・ライト、ノーヒット)	31
11.9. 32 ビット・バスモード (バースト・リード、インターリーブ)	32
11.10. 32 ビット・バスモード (バースト・ライト、インターリーブ)	33
11.11. 32 ビット・バスモード (バースト・リード、ノン・インターリーブ)	34
11.12. 32 ビット・バスモード (バースト・ライト、ノン・インターリーブ)	35
11.13. 16 ビット・バスモード (シングル・リード)	36
11.14. 16 ビット・バスモード (シングル・ライト)	38
11.15. 16 ビット・バスモード (バースト・リード、インターリーブ)	39
11.16. 16 ビット・バスモード (バースト・ライト、インターリーブ)	40
11.17. 16 ビット・バスモード (バースト・リード、ノン・インターリーブ)	41
11.18. 16 ビット・バスモード (バースト・ライト、ノン・インターリーブ)	42
12. マスカブル割り込みを使用したアプリケーションの開発	43
12.1. 割り込みベクタ	43
12.2. 内蔵 RAM ベクタ	44
12.3. 一般的な制限事項 / 注意事項	44
12.4. ブレーク・ポイント使用に関する制限事項 / 注意事項	45
13. ボード上のタイマ割り込みをアプリケーションで使用する場合	46
13.1. 準備	46
13.2. 制限事項 / 注意事項	46

1. はじめに

このマニュアルでは、日本電気社製 CPU の V830 用の評価ボード「RTE-V830-PC」について説明します。RTE-V830-PC では、GreenHills 社の Multi デバッガを利用してプログラムの開発 / デバッグや CPU のパフォーマンス評価などができます。このデバッガとの通信には、IBM-PC/AT の ISA バスか RS-232C シリアル・インターフェースが使用できます。またボードに設けられたローカル・バス用のコネクタを使用してメモリや I/O を拡張することもできます。

1.1. マニュアル表記について

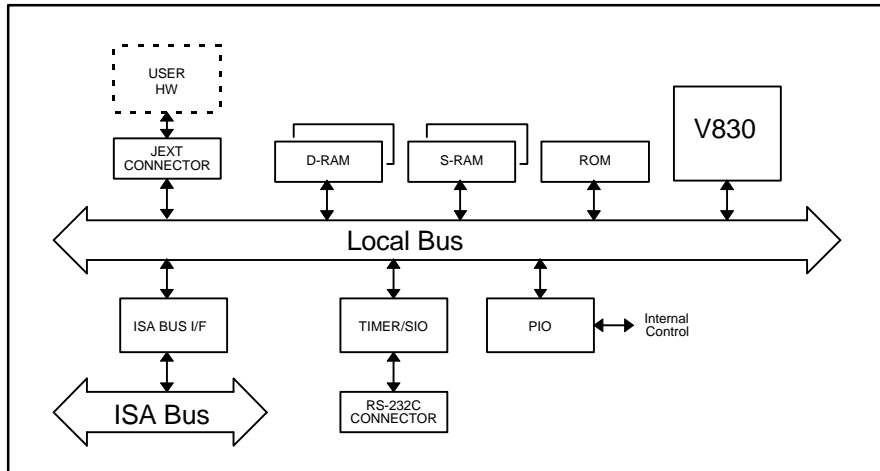
本書では、数字の表記については表の表記を用います。16 進数や 2 進数の表記では、桁数が多くて読みにくい場合は、4 桁ごとに“-”（ハイフン）を入れてあります。

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

2. 特徴と機能

RTE-V830-PC の機能ブロックの概要を図に示します。



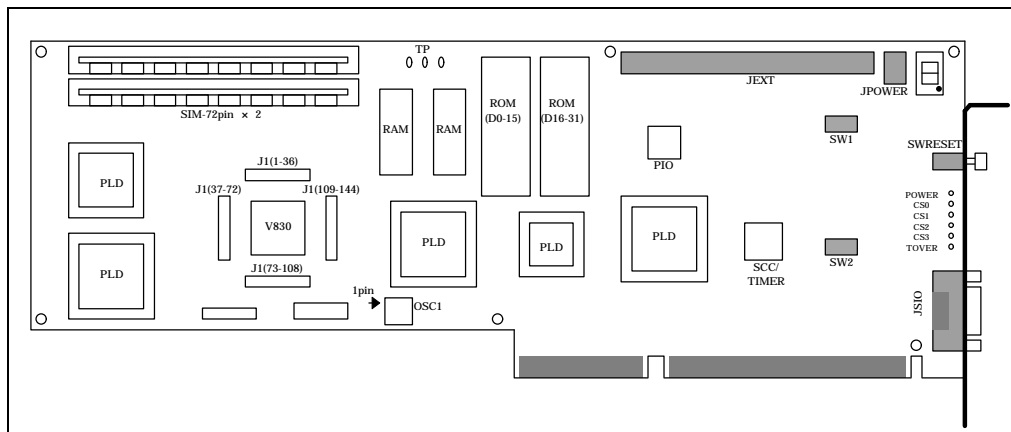
RTE-V830-PC ブロック図

特徴

- ROM 256K バイト (64K × 16 ビット EPROM × 2)
- SRAM 512K バイト (64K × 16 ビット SRAM × 4)
- DRAM SIMM72 ピン・ソケット × 2 枚により 8M, 16M, 32M (標準 8M バイト)
- RS-232C ポート (D-SUB 9 ピン)
- PC/AT 互換機の ISA バスによる通信機能
- ユーザー拡張用のローカル・バスのコネクタ
- CPU の全ての信号を計測できるプロセッサ・ピンコネクタ
- 外部リセット・スイッチをリアパネルに用意
- 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

図は RTE-V830-PC 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。



RTE-V830-PC 概観

3.1. リセット・スイッチ (SWRESET)

SWRESET はリセット・スイッチです。このスイッチを押すと CPU がリセットされます。

3.2. 電源コネクタ (JPOWER)

本ボードを ISA バス・スロットに挿さずに単体で使用する場合に、JPOWER コネクタに外部電源を接続して電源を供給します。

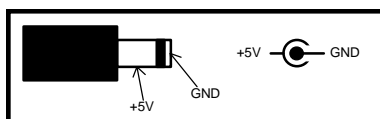
JPOWER コネクタに供給する電源は、下記の通りです。

電圧：5 V

電流：最大 2 A (ただし、JEXT コネクタへの供給分を含まず)

適合コネクタ：Type A (5.5)

極性：



【注意】電源コネクタの極性に注意してください。また、ISA バス・スロットに挿して使用する場合には、JPOWER に電源を接続しないでください。

3.3. スイッチ1 (SW1)

SW1 は、汎用の入力ポートのスイッチです。スイッチは、OFF で 1、ON で 0 の値となります。詳細は、「6.2. PIO (uPD71055)」を参照してください。

3.4. スイッチ2 (SW2)

SW2 は、ISA バスの I/O アドレス選択のスイッチです。スイッチの番号 1 ~ 8 が ISA バスのアドレス A4 ~ A11 に対応しています (A12 ~ A15 は 0 固定)。したがって、I/O アドレスとして 000xH ~ 0FFxH が選択できます。スイッチは、OFF で 1、ON で 0 の値となります。

SW1 番号	1	2	3	4	5	6	7	8
アドレス	A4	A5	A6	A7	A8	A9	A10	A11

SW2 アドレス対応

3.5. LED

LED は、各種ステータスを示しています。表に内容を示します。

名称	内容
POWER	ボードに電源が供給されている時に点灯
CS0	CPU の CS0-端子がアクティブ(Low)時に点灯
CS1	CPU の CS1-端子がアクティブ(Low)時に点灯
CS2	CPU の CS2-端子がアクティブ(Low)時に点灯
CS3	CPU の CS3-端子がアクティブ(Low)時に点灯
TOVER	タイムオーバー発生時に点灯

LED ステータス

3.6. テストピン (TP)

TP は、ROM インサーキット型のデバッグを接続する際に使用するテストピンです。ROM インサーキット・デバッグから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。

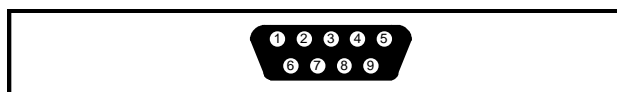
信号名	入出力	機能
RESET-	入力	Low レベル入力により、CPU がリセットされる。ROM インサーキット・デバッグからのリセット要求信号を接続。1K プルアップ。
NMI-	入力	Low レベル入力により、CPU に NMI が入る。ただし、ソフトウェアによってマスクできる。ROM インサーキット・デバッグからの NMI 要求信号 (ブレイク要求) を接続。1K プルアップ。
GND	- - -	GND。ROM インサーキット・デバッグの GND と接続。

TP 端子の機能

3.7. シリアル・コネクタ (JSIO)

JSIO コネクタは、シリアル・コントローラ (SCC2691) 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図と表の通りです。

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)。



JSIO ピン配置

ピン番号	信号名	入出力	ホストの接続ピン番号	
			D-SUB9	D-SUB25
1	NC			
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	NC			

JSIO コネクタ信号

3.8. CPU テストピン (J1)

CPU テストピンは、CPU の端子番号がコネクタのピン番号に対応して、1対1で接続されています。テストピンとして、信号引出用に使用できます。

3.9. 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のためにオシレータを実装します。V830 では、システム・クロックの生成に PLL を使用しています。OSC1 ソケットに実装するオシレータの周波数は SW1-7 の設定により、内部の周波数の 2 分の 1 または、3 分の 1 になります。

OSC1 ソケットには、D I P 8 ピンタイプ (ハーフタイプ) のオシレータを実装してください。

【注意】 オシレータやクリスタル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願います。

3.10. DRAM-SIMM ソケット

DRAM-SIMM ソケットには、標準で 4M バイトの SIMM が 2 枚実装されています。このソケットには 4M, 8M, 16M バイトの 72 ピン SIMM (DOS/V 機用) が実装できますので、DRAM の容量を増設可能です。別表を参考にして、タイミング上満足するアクセスタイムのものをご使用ください。尚、S I M M は、同種のを必ず 2 枚 1 組でご使用下さい。また、実装されている SIMM の容量は PIO ポートから読み出すことができます (「6.2. PIO (uPD71055)」参照)。

3.11. ROM ソケット

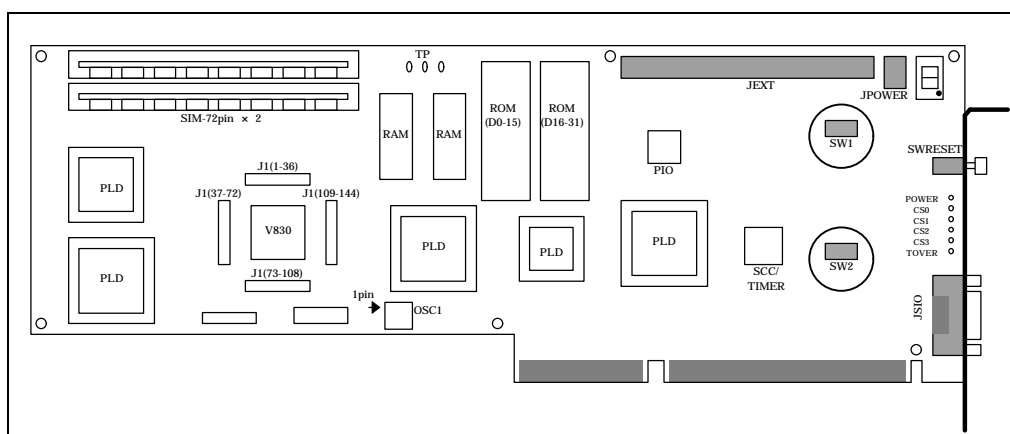
ROM ソケットには、標準で 128K バイト (64K × 16 ビット) の 40 ピン ROM が実装されています。もし、変更する場合はアクセス・タイムが 150ns 以下のものをご使用ください。

4. インストールと使用方法

ボードのインストールは、PC/AT 互換機（以下 PC）の ISA バス・スロットに組み込んで使用することを想定していますが、外部電源を用意して単体で使用することもできます。また、動作確認のテストや 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 Windows）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、「RTE for Windows インストール・マニュアル」を参照してください。

4.1. ボードの設定

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。スイッチの箇所を図に示します。



ボードのスイッチ

SW2 は、ISA バスの I/O アドレス選択のスイッチです。スイッチの番号 1 ~ 8 が ISA バスのアドレス A4 ~ A11 に対応しています（A12 ~ A15 は 0 固定）。したがって、I/O アドレスとして 000xH ~ 0FFxH が選択できます。なおスイッチは、OFF で "1"、ON で "0" の値となります。一般的には、20xH ~ 3FxH の間で設定します。

SW2 番号	1	2	3	4	5	6	7	8	
アドレス	A4	A5	A6	A7	A8	A9	A10	A11	I/O アドレス
ON/ OFF	0	0	0	0	0	1	0	0	020xH (出荷時の設定)

I/O アドレスの対応

SW1 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 用モニタでは、RS-232C のボーレートとプロファイラのタイマ周期の設定に使用しています。

SW1 番号	1	2	ボーレート
設定	ON	ON	未使用
	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 5ms
	ON	OFF	100 Hz 10ms
	OFF	OFF	60 Hz 16.67ms (出荷時の設定)

プロファイラ周期の設定

SW1 の 5-6 は、Multi モニタでは使用していません (常時 OFF)。

SW1 番号	7	CMODE (V830 ピン)
設定	ON	3 倍モード
	OFF	2 倍モード

CMODE の設定

SW1 番号	8	SIZE16 (V830 ピン)
設定	ON	3 2 ビットモード
	OFF	1 6 ビットモード

SIZE16 の設定

4.2. ISA バスに実装する場合

ボードを PC の ISA バス・スロットに実装すると、ISA バスからボードへ電源 (+5V) が供給されます。また、デバッガとの通信に ISA バス経由が使用できるため、プログラムの高速ダウンロードなどが実現できます。

ISA バス・スロットへの実装は、以下の手順となります。

ボード上のディップ・スイッチにより、PC の I/O アドレスを設定します。I/O アドレスは他の I/O と重ならないように注意してください。スイッチの設定については「4.1. ボードの設定」を参照ください。

PC の電源を切って筐体をあけ、ボードを実装する ISA バス・スロットを確認します。実装するスロットにリアパネルが付いている場合は、そのリアパネルを外します。

ボードを ISA バス・スロットに差し込み、ボードが隣接の他のボードなどと接触していないかを確認し、ボードに付いているリアパネルを PC の筐体にネジで固定します。

PC の電源を入れ、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 PC の電源を切り接続を確認してください。**システムが正常に立ち上がらない (デバイス・ドライバの組み込みでエラーが発生するなど) 場合は、設定した I/O アドレスが他の I/O と重なっている可能性があります。PC のマニュアルや他に実装されているボードのマニュアルなどを参照して、ボードの I/O アドレスを再確認してください。

システムが正常と判断できたら、再度 PC の電源を切ってから筐体を元に戻します。

4.3. ボード単体で使用する場合

PC に組み込まず、ボード単体で使用する場合は、外部からの電源供給が必要となります。また、デバッガとの通信も RS-232C 経由のみとなります。PC/AT 互換機以外のホスト・デバッガを使用する場合や、ハードウェアの確認や拡張を目的とする場合に便利です。

ボードを単体で使用する場合は以下の通りです。

ホストと接続するための RS-232C ケーブルと、電源供給のための外部電源 (+5V 1A) 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。また、ボードの 4 隅にスペーサを取り付けるなど、設置場所にも問題がないようにしてください。RS-232C ケーブルの結線は「3.7. シリアル・コネクタ (JSIO)」、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。

ボード上のディップ・スイッチで、RS-232C のボーレートを設定します。スイッチの設定については「4.1. ボードの設定」を参照ください。

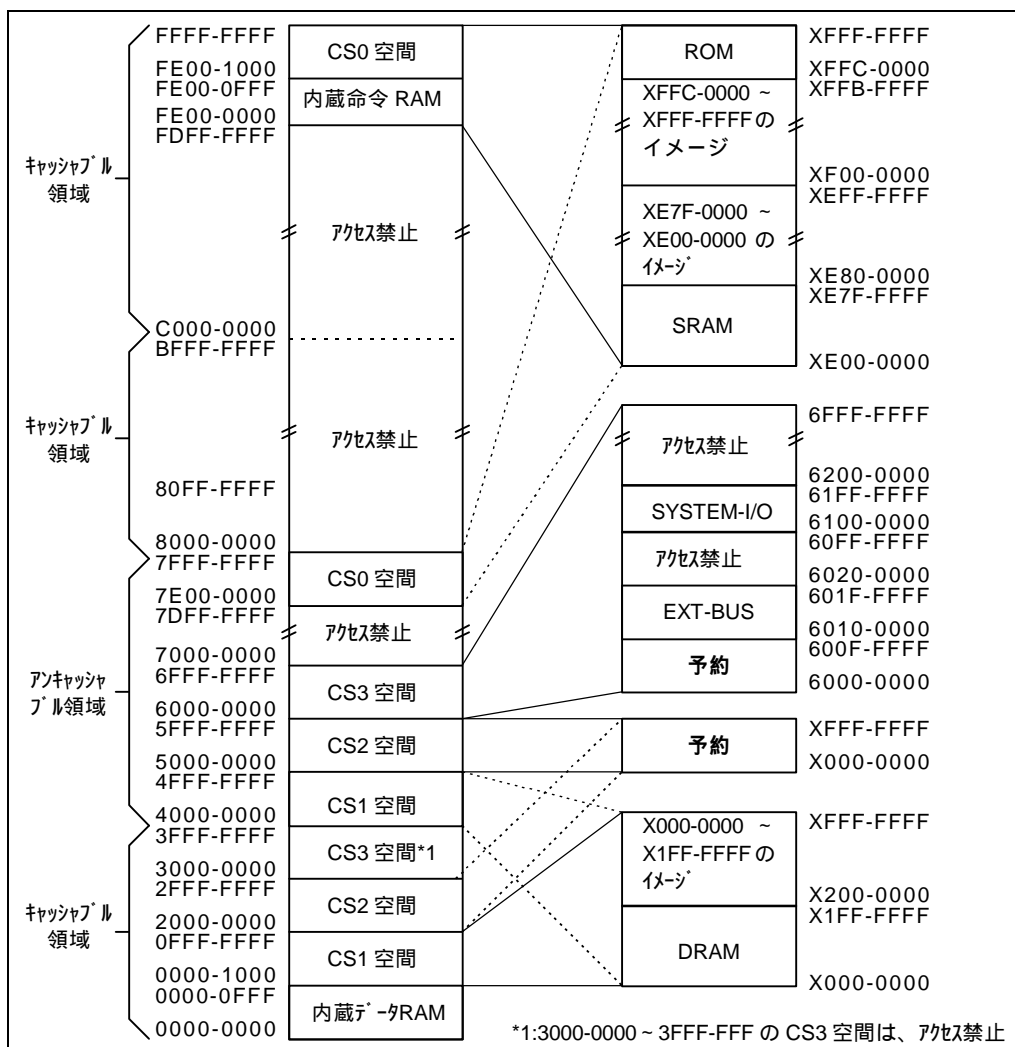
ホストと RS-232C ケーブルで接続して、JPOWER コネクタに電源を接続し、ボードの POWER-LED が点灯することを確認します。LED が点灯しない場合は、すぐに電源を切り接続を確認してください。

5. ハードウェア・リファレンス

ここでは、RTE-V830-PC ボードのハードウェアについて記述します。

5.1. メモリ・マップ

ボードのメモリ割り付けは、以下の通りです。



メモリ・マップ

DRAM 空間 (0000-0000H ~ 01FF-FFFFH, 4000-0000H ~ 41FF-FFFFH,)

ボードに実装する 72 ピンの SIMM の空間です。標準では 4M バイトの SIMM が 2 枚実装されていますが、8M または 16M バイトの SIMM に交換してメモリを拡張できます。RAS, CAS, プリチャージ幅の設定が可能です。(6.2参照)

予約及びアクセス禁止空間

この空間へのアクセスは行なわないでください。

EXT-BUS 空間 (6010-0000H ~ 601F-FFFFH)

ボード上の JEXT コネクタに接続するハードウェア拡張ボードのための空間です。EXT-BUS についての詳細は「7. JEXT」を参照してください。

SYSTEM-I/O 空間 (6100-0000H~61FF-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です。これは、メモリ・マップド I/O となっています。詳細については「6. SYSTEM-IO」を参照してください。

SRAM 空間 (FE00-0000H~FE07-FFFFH,7E00-0000H~7E07-FFFFH)

ボード上に実装された SRAM で 512K バイトの空間があります。SRAM は 0wait でアクセスできます。リード、ライトサイクル別々にウェイトの設定が可能です。(??? ?参照)

ROM 空間 (FFFC-0000H~FFFF-FFFFH,7FFC-0000H~7FFF-FFFFH)

ボード上に実装された ROM で 256K バイトの空間があります。ROM は ready 信号制御によりアクセス・サイクルに 10wait が挿入されています。外部バスクロックが 50MHz の時には、アクセス・タイムが 150nsec 以下の ROM を使用します。

実装されている ROM には、標準で Multi 用のモニタが組み込まれています。

5.2. I Oマップ

V830-CPU の持つ I/O 空間は、ボードでは使用していません。制御用の I/O は、メモリ・マップドされた SYSTEM-I/O 空間に存在します。

6. SYSTEM-IO

SYSTEM-IO は、メモリ空間にマップされた I/O デバイスで UART/TIMER, PIO, ISA バス・インターフェースなどがあります (ISA バス・インターフェースについては、説明を省略します)。

6.1. UART/TIMER(SCC2691)

UART/TIMER として PHILIPS Signetics 製の SCC2691UART レシーバ/トランスミッタ LSI を使用しています。SCC2691 は受信部に 3 キャラクタ分のバッファを内蔵しているため受信時のオーバーラン・エラーの可能性を最小限に押さえられます。また、X1,X2 端子には 3.6864MHz が接続されており、SCC2691 内の 16 ビット・カウンタを使用することにより、約 271nS ~ 17.8mS まで測定できます。

SCC2691 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、SCC2691 のマニュアルを参照ください。

アドレス	読み出し	書き込み
6100-0400h	MR1,MR2	MR1,MR2
6100-0404h	SR	CSR
6100-0408h	Reserved	CR
6100-040Ch	RHR	THR
6100-0410h	Reserved	ACR
6100-0414h	ISR	IMR
6100-0418h	CTU	CTUR
6100-041Ch	(CTL)	CTLR

SCC2691 レジスタ配置

汎用出力端子(MPO)は RTS(RS)として、汎用入力端子(MPI)は CTS(CS)として使用しています。なお DTR(DR),DSR(ER)は PIO によって制御されていますので、「6.2.PIO (uPD71055)」を参照してください。

SCC2691 は、システム・リセット(「8.1.RESET-」参照)によってリセットされます。

6.2. PIO (uPD71055)

PIO は NEC 製の uPD71055 が実装されています。uPD71055 は Intel 製の i8255 と互換であり、3 つの平行ポートを持っています。このポートにより、各種の制御を行なっています。PIO の各レジスタは、表の通りに割りあてられています。

アドレス	読み出し	書き込み
6100-0800h	PORT0	PORT0
6100-0804h	PORT1	PORT1
6100-0808h	PORT2	PORT2
6100-080Ch	-----	COMMAND REG

PIO のレジスタ配置

PIO は、システム・リセットによってリセットされます。その時、各ビットは全て入力になるため、出力として使用しているビットの信号状態はプルアップ抵抗により High レベルとなります。各ポートのビットの使用状況を表に示します。

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORT0	SRAMRD WIDE1	SRAMRD WIDE0	SRAMWR WIDE1	SRAMWR WIDE0	INTERLEA VE	予約'1'		
	出力							
PORT1	PCWIDE1	PCWIDE0	RDCAS WIDE1	RDCAS WIDE0	WRCAS WIDE0	MINRASWIDE[2..0]		
	出力							
PORT2	PD[2..1]		TOVERF-	DSR-	DTR-	NMIMASK	TOVERCLR-	予約'1'
	入力				出力			

PIO ビット割り付け

各ポートの詳細を下記に示します。

Port 0 : 内部コントロールポート(出力)・・・61000800h

P07	P06	P05	P04	P03	P02	P01	P00
SRAMRD WIDE1	SRAMRD WIDE0	SRAMWR WIDE1	SRAMWR WIDE0	INTERLEAV E	予約		

予約1' : システムで予約されているビットです。初期化時に3ビット共"1"に設定した後は、変更しないでください。

INTERLEAVE : DRAMのインターリーブの設定です。1通常モード、0でノンインターリーブエミュレーションモードになります。

SRAMWRWIDE1..0 : SRAMライト時のウェイト設定です。

SRAMWRWIDE1	SRAMWIDE0	機能
0	0	SRAMライト時のウェイトを0に設定
0	1	SRAMライト時のウェイトを1に設定
1	0	SRAMライト時のウェイトを2に設定
1	1	SRAMライト時のウェイトを3に設定

SRAMRDWIDE1..0 : SRAMリード時のウェイト設定

SRAMRDWIDE1	SRAMRDWIDE0	機能
0	0	SRAMリード時のウェイトを0に設定
0	1	SRAMリード時のウェイトを1に設定
1	0	SRAMリード時のウェイトを2に設定
1	1	SRAMリード時のウェイトを3に設定

Port 1: DRAMのアクセス条件の設定出力ポート(入力)・・・61000804h

P17	P16	P15	P14	P13	P12	P11	P10
PRCWIDE	PRCWIDE	RDCASWI	RDCASWI	WRCASWI	MINRAS	MINRAS	MINRAS
1	0	DE1	DE0	DE0	WIDE2	WIDE1	WIDE0

MINRASWIDE2..0: DRAMの最小RAS幅の設定

RASWIDE	RASWIDE	RASWIDE	機能
2	1	0	
0	0	0	設定禁止
0	0	1	設定禁止
0	1	0	最小RAS幅を2CPUバスクロックに設定
0	1	1	最小RAS幅を3CPUバスクロックに設定
1	0	0	最小RAS幅を4CPUバスクロックに設定
1	0	1	設定禁止
1	1	0	設定禁止
1	1	1	設定禁止

WRCASWIDE0: DRAMのライト時のCAS幅の設定

WRCASWIDE0	機能
0	ライト時のCAS幅を1CPUバスクロックに設定
1	ライト時のCAS幅を2CPUバスクロック相当に設定

RDCASWIDE1..0: DRAMのリード時のCAS幅の設定

RDCASWIDE1	RDCASWIDE0	機能
0	0	設定禁止
0	1	リード時のCAS幅を1CPUバスクロックに設定
1	0	リード時のCAS幅を2CPUバスクロックに設定
1	1	リード時のCAS幅を3CPUバスクロックに設定

PRCWIDE1..0: DRAMのプリチャージ幅の設定

PRCWIDE1	PRCWIDE0	機能
0	0	設定禁止
0	1	プリチャージ幅を1CPUバスクロックに設定
1	0	プリチャージ幅を2CPUバスクロックに設定
1	1	プリチャージ幅を3CPUバスクロックに設定

Port 2 : 内部コントロールポート(出力) 61000808h

P23	P22	P21	P20
DTR-	NMIMASK	TOVERCLR-	予約 1

予約1 : システムで予約されているビットです。初期化時に” 1” に設定しています。変更しないでください。

TOVERCLR- : PORT2 の Bit5 にある TOVERF-を” 1” にクリアするための制御ビットです。初期化時と通常は” 1” に設定してください。TOVERF-をクリアする時のみ、” 0” としてから” 1” に戻してください。

NMIMASK : CPU に入る NMI のマスクを制御するビットです。” 1” に設定すると NMI はゲートによりマスクされます。初期化時は” 1” として、NMI を受けられる状態になったら” 0” に設定してください。Multi のモニタでは、1 で初期化しています。

DTR- : JSIO コネクタから出力されている DTR 信号を制御するビットです。このビットの設定状態が反転され RS232C レベルに加工されて JSIO コネクタから出力されます。

Port 2 : 内部コントロールポート(入力) 61000808h

P27	P26	P25	P24
PD2	PD1	TOVERF-	DSR-

DSR- : JSIO コネクタから入力される DSR 信号の状態を示すビットです。このビットの状態は JSIO コネクタの DSR 信号の状態が反転したものです。

TOVERF- : パス・サイクルが 30 サイクル以上となり、タイム・アウトが発生すると” 0” となります。フラグをクリアする(” 1” にもどす)には、PORT2 の Bit1(TOVERCLR-)を使います。

PD[2..1] : ボードに実装されている DRAM (72 ピン SIMM) の PD[2..1]が読みだせます。このビットの状態によって、実装されている DRAM のサイズを確認できます。PD[2..1]と DRAM の容量の関係を表に示します。

PD[2]	PD[1]	DRAM の容量
0	0	4M バイト
0	1	予約
1	0	16M バイト
1	1	8M バイト

PD[2..1]と DRAM の容量

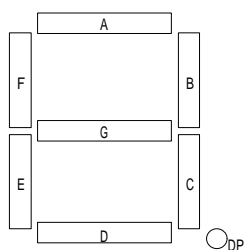
PORT 0 - 2 制御ポート 6100080Ch

PORT 0, 1, 2 は、uPD71055 のポートであり、当該番地への書き込みでこれらのポートの初期設定を実施する。

6.3. その他のポート

Port 3 : 7 seg LEDの表示データ出力ポート(出力)・・・61000C00h

P17	P16	P15	P14	P13	P12	P11	P10
DPseg	Gseg	Fseg	Eseg	Dseg	Cseg	Bseg	Aseg



Port 4 : DIPSW1の状態の読み込みポート(入力)・・・61000C04h

P17	P16	P15	P14	P13	P12	P11	P10
SW1-8	SW1-7	SW1-6	SW1-5	SW1-4	SW1-3	SW1-2	SW1-1
SIZE16B	CMODE	no use	no use	TIM1	TIM0	BPS1	BPS0

SW1-[8..1] : ボード上に実装されている SW1 の状態を読み出せます。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。

BPS1	BPS0	ボーレート
ON	ON	9600bps
ON	OFF	19200bps
OFF	ON	38400bps
OFF	OFF	don' t use

TIM1	TIM0	タイマーレート
ON	ON	タイマを使用しない
ON	OFF	200Hz (5ms)
OFF	ON	100Hz (10ms)
OFF	OFF	60Hz (16.67ms)

CMODE :内部クロックの通倍率(ON で 3 倍、OFF で 2 倍)

SIZE16B :バスサイズの設定 (ON で 3 2 ビット、OFF で 1 6 ビット)

7. JEXT バス仕様

JEXT は、メモリや I/O などを拡張できるように用意されたコネクタです。このコネクタには、本ボードの内部のローカル・バスが接続されています。

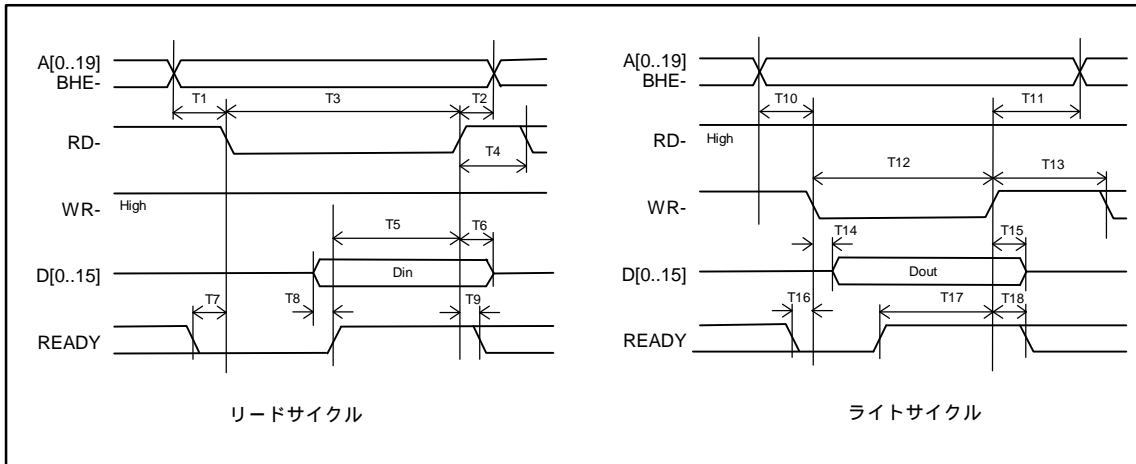
以下に JEXT コネクタのピン配置、信号およびタイミングを示します。

番号	信号名	番号	信号名	番号	信号名	番号	信号名
1	+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	CPUCLK	60	GND

JEXT コネクタピン配置

信号名	入出力	機能
A[0..19]	出力	アドレス・バス信号。CPU のアドレス信号をバッファして接続。
BHE-	出力	バイトハイ・イネーブル信号。CPU の UBE-信号をバッファして接続。
D[0..15]	入出力	データ・バス信号。CPU のデータ・バス信号をバッファして接続。 また、ボード上で 10K プルアップ。
RD-	出力	リード・サイクルのタイミング信号。JEXT 空間のアクセス時のみ、アクティブになる。
WR-	出力	ライト・サイクルのタイミング信号。JEXT 空間のアクセス時のみ、アクティブになる。
READY	入力	サイクルの終了を CPU に通知する信号。JEXT 空間のみで有効。 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。また、ボード上で 10K プルアップ。
INT-	入力	Low アクティブの割り込み要求信号。割り込みベクター 11 を発生します。また、ボード上で 10K 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。
CLK	出力	クロック信号。V830 の CLKOUT 端子がバッファ後、接続されている。

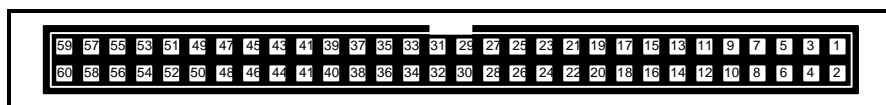
JEXT コネクタ信号



JEXT 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		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	

JEXT バス AC スペック



JEXT のピン配置

8. その他のCPU 資源

8.1. RESET-

CPU へのリセットは下記に示した要因で発生します。このリセットは、CPU のリセットと共に、ボード制御回路のシステム・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 時に発生するリセットです。
- ・**TP からのリセット要求**: TP の RESET-端子からの入力によるリセットです。詳細は『3.6.テストピン (TP)』を参照して下さい。
- ・**SWRESET によるリセット**: ボードのリアパネル部分に用意されているリセット・スイッチ (SWRESET) が押されるとリセットが発生します。
- ・**ホストからのリセット要求**: ISA バス経由でリセットを発生させることができます。

8.2. NMI-

CPU への NMI は下記に示した要因で発生します。

- ・**SCC2691 の割り込み**: SCC2691 の割り込み要求出力 (INTR-) がアクティブになると、NMI が発生します (『6.1.UART/TIMER(SCC2691)』を参照)。
- ・**TP からの NMI 要求**: TP の NMI-端子からの入力によるリセットです。詳細は『3.6.テストピン (TP)』を参照してください。
- ・**ISA バスからの要求**: ISA バス経由の通信制御のために NMI を使用しています。

また、NMI はハード的にマスクすることができます。マスク方法については、『6.2.PIO (uPD71055)』の『NMIMASK』に関する記述を参照してください。NMIMASK による NMI のマスクは、上記の全ての NMI 要求に対して有効になります。

NMI が発生した場合は、下記の手順で処理します。

- PIO の NMIMASK に” 1” を設定して、NMI をハード的にマスクする。
- NMI の要求元 (SCC2691 の ISR、PIO の TOVERF) を検査する。
- 要求元のための NMI 処理を行ない、要求をクリアする。
- PIO の NMIMASK に” 0” を設定して、マスクを解除する。
- NMI 処理から復帰する。

9. Multi モニタ

ボードに実装されている ROM には、Multi 用のモニタが組み込まれています。ホストの Multi サーバと接続して使用する場合の注意点について説明します。

9.1. モニタ・ワーク RAM

モニタでは、SRAM の最上位の 64KB をワーク用の RAM として使用しています。したがって、FE07-0000H ~ FE07-FFFFH はユーザ・プログラムでは使用できません。

9.2. 割り込み

現在 Multi モニタを使用して、ユーザ・プログラムで割り込み処理はできません。

9.3. _INIT_SP の設定

モニタで _INIT_SP (スタック・ポインタの初期値) は、FE06-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)。

9.4. リモート接続

Multi のサーバとの接続は、シリアル接続と ISA バス接続が選択できますが、一度接続した方から他方に切り替える場合には、モニタをリセット (リアパネルのリセット・スイッチを押す) してから RTE for Windows のユーティリティ Check RTE で接続を変更してください。

10. RTE コマンド

サーバと接続すると TARGET ウィンドウが開かれ、ここで RTE コマンドを発行することができます。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
INB, INH, INW	I/O リード
OUTB, OUTH, OUTW	I/O ライト
DCTR,INTR,PLLCR, CMCR	内部レジスタの変更、表示

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

10.1. 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 SFR

SFR コマンドのヘルプを表示します。

10.2. INIT

<書式> INIT

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

10.3. VER

<書式> VER

VER は、RTE 環境のバージョンを表示します。

10.4. INB,INH,INW

<書式> INB [アドレス]

 INH [アドレス]

 INW [アドレス]

INB,INH,INW は、I/O リードを行ないます。INB はバイト、INH はハーフ・ワード、INW はワード単位でアクセスします。アドレスが省略される、前回のアドレスが指定されたものとみなします。

<例> INB 1000

1000H からバイトで I/O リードします。

10.5. OUTB,OUTH,OUTW

<書式> OUTB [[アドレス] データ]
OUTH [[アドレス] データ]
OUTW [[アドレス] データ]

OUTB,OUTH,OUTW は、I/O ライトを行ないます。OUTB はバイト、OUTH はハーフ・ワード、OUTW はワード単位でアクセスします。アドレスとデータが省略される、前回の指定と同じものとみなされます。

<例> OUTH 2000 55AA
2000H 番地にデータ 55AAH をハーフ・ワードで I/O ライトします。

10.6. DCTR コマンド

<書式> DCTR[ALL]

DCTR レジスタを表示します。レジスタは2 5 6 個ありますが、バリッドビットが有効になっているレジスタのみ表示します。ただし、all を指定した場合は、全てのレジスタ値を表示します。DCTR レジスタは、I/O 空間 f2000000h-f2000fffh にマップされています。

10.7. ITCR コマンド

<書式> ITCR[ALL]

ICTR レジスタを表示します。レジスタは1 2 8 個ありますが、バリッドビットが有効になっているレジスタのみ表示します。ただし、all を指定した場合は、全てのレジスタ値を表示します。ICTR レジスタは、I/O 空間 fa000000h-fa000fffh にマップされています。

10.8. PLLCR コマンド

<書式> PLLCR

PLL コントロールレジスタの値を表示します。

10.9. CMCR コマンド

<書式> CMCR [=]VALUE

CMCR(キャッシュメモリコントロールレジスタ) に値を設定します。

11. DRAM のタイミング

11.1. DRAM インターフェース概要

DRAM は 32 ビットのバンクを 2 バンク持っており、通常（インターリーブ・モード）ではその 2 つのバンクを交互にアクセスすることにより、バースト・アクセス時のアクセス・タイムを短縮しています。

また、DRAM へのアクセスはページ・モードを使用し、同じ Row アドレスへのアクセスが続く限り、DRAM への RAS 信号をアクティブに保ち続け、CAS 信号のみを操作してアクセスを行っています。これにより、同じ Row アドレスに連続でアクセスする場合にのアクセス・タイムを短縮しています。

11.2. 信号の説明

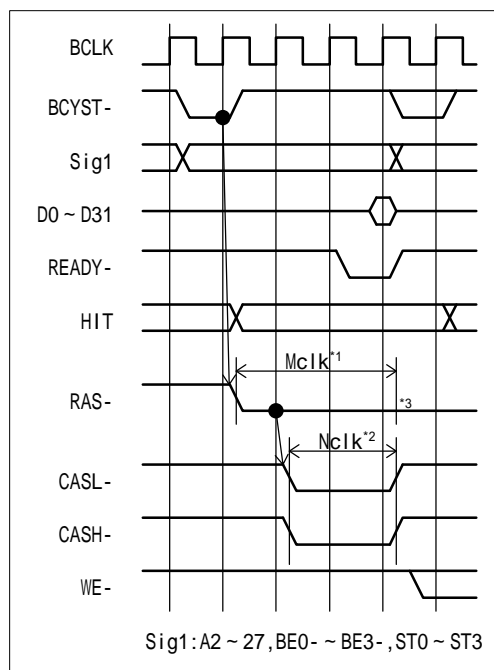
本章で説明する波形に用いられる信号名を下記に説明する。

- BCLK : CPU に入力されるバス・クロック。
- BCYST- : CPU が出力するバス・サイクル・スタート信号。
- READY- : CPU に入力される Ready 信号。
- HIT : DRAM をページモードで使用するために、Row アドレスが前のサイクルと一致しているかどうかを示す信号。DRAM のコントロール回路が生成している。
- RAS- : DRAM に入力される RAS 信号。
- CASL- : DRAM の下位 32 ビット・バンクに入力される CAS 信号。実際には各バイト単位に 1 本ずつ存在する(CASL0- ~ CASL3-)。
- CASH- : DRAM の上位 32 ビット・バンクに入力される CAS 信号。実際には各バイト単位に 1 本ずつ存在する(CASH0- ~ CASH3-)。
- WE- : DRAM に入力される WE 信号。

11.3. 32 ビット・バスモード(シングル・リード、ノーマル)

32 ビット・バスモード時のシングル・リード・サイクルで、その領域にリセット後初めてアクセスする場合や、リフレッシュ・サイクル終了後プリチャージ時間以上経過してからアクセスする場合(ノーマル)の波形を下図に示します。

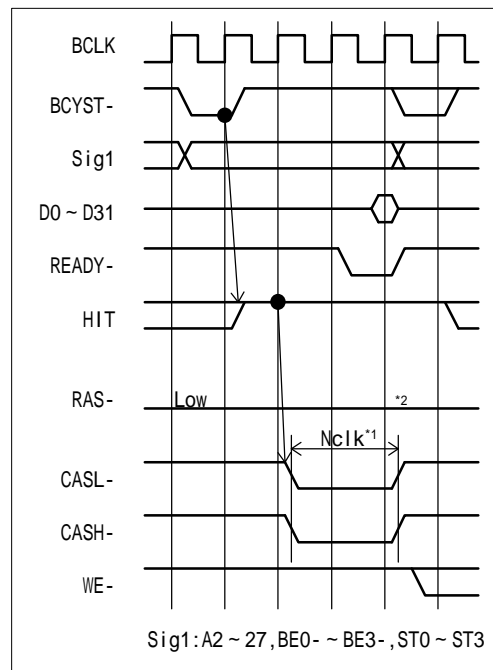
- ✧ 下図に $Mclk/Nclk(*1,*2)$ で示されている波形の幅は、それぞれポートに設定する RAS 幅(2~4) / リード CAS 幅(1~3)のクロック数です。したがって、最短のサイクルは 1Wait です。
- ✧ サイクルが終了後もページ・モード・アクセスのために RAS-信号が Low に保たれています。(*3)。
- ✧ 下位 / 上位のそれぞれ 32 ビット・バンクは同時に読み出され、A2 の状態により適切なバンクのデータが CPU に供給されます。



11.4. 32 ビット・バスモード(シングル・リード、ヒット)

32 ビット・バスモード時のシングル・リード・サイクルで、Row アドレスが前のサイクルと一致した(ヒットした)場合の波形を下図に示します。

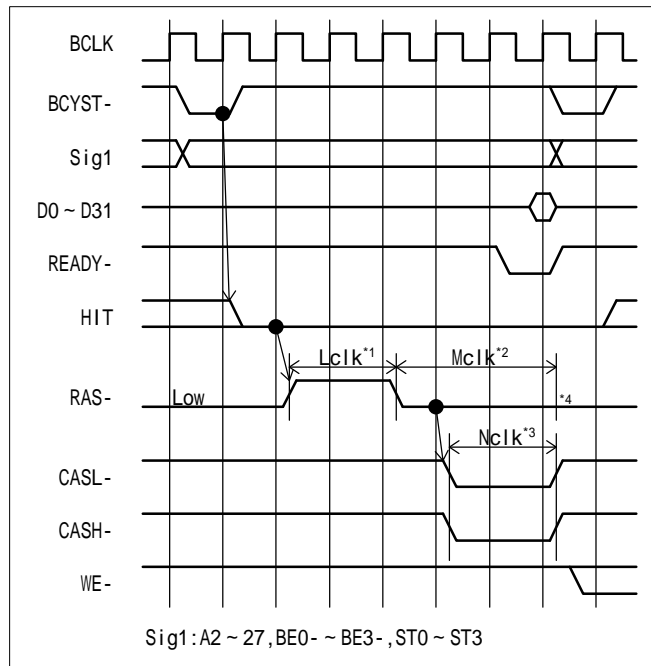
- ◇ 下図に $N_{clk}(*1)$ で示されている波形の幅は、ポートに設定するリード CAS 幅(1~3)のクロック数です。したがって、最短のサイクルは 1Wait です。
- ◇ サイクルが終了後もページ・モード・アクセスのために RAS-信号が Low に保たれています。(*2)
- ◇ 下位 / 上位のそれぞれ 32 ビット・バンクは同時に読み出され、A2 の状態により適切なバンクのデータが CPU に供給されます。



11.5. 32 ビット・バスモード(シングル・リード、ノーヒット)

32 ビット・バスモード時のシングル・リード・サイクルで、Row アドレスが前のサイクルと異なっている場合(ノーヒット)の波形を下図に示します。

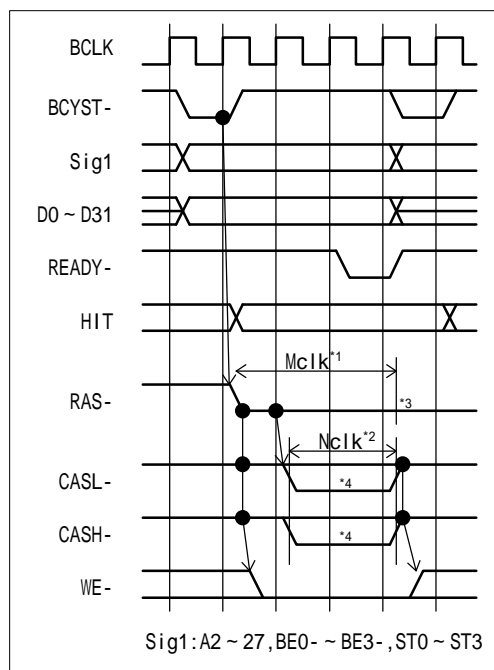
- ◇ 下図に Lclk/Mclk/Nclk(*1,*2,*3)で示されている波形の幅は、それぞれポートに設定するプリチャージ幅(1~3)/RAS 幅(2~4)/リード CAS 幅(1~3)のクロック数です。したがって、最短のサイクルは 3Wait です。
- ◇ サイクルが終了後もページ・モード・アクセスのために RAS-信号が Low に保たれています。(*4)
- ◇ 下位/上位のそれぞれ 32 ビット・バンクは同時に読み出され、A2 の状態により適切なバンクのデータが CPU に供給されます。



11.6. 32ビット・バスモード(シングル・ライト、ノーマル)

32ビット・バスモード時のシングル・ライト・サイクルで、その領域にリセット後初めてアクセスする場合や、リフレッシュ・サイクル終了後プリチャージ時間以上経過してからアクセスする場合(ノーマル)の波形を下図に示します。

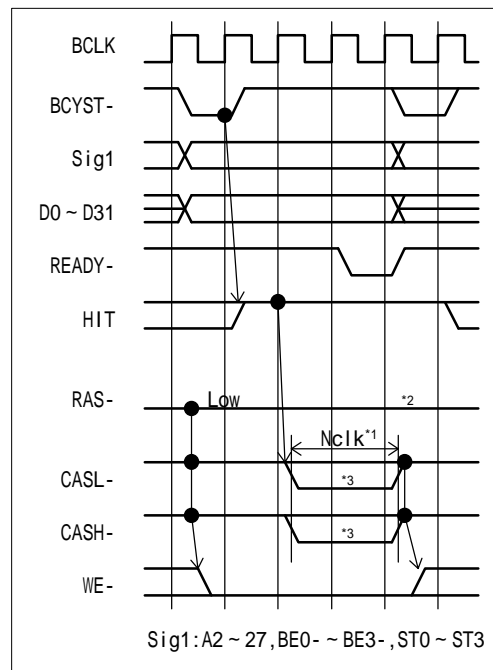
- ◇ 下図に $Mclk/Nclk(*1,*2)$ で示されている波形の幅は、それぞれポートに設定する RAS 幅(2~4)/ライト CAS 幅(1~2)のクロック数です。したがって、最短のサイクルは 1Wait です。
- ◇ サイクルが終了後もページ・モード・アクセスのために RAS-信号が Low に保たれています。(*3)
- ◇ CASL-および CASH-は、A2 およびバスイネーブルの BE0-~BE3-の状態により、適切なバンクの適切なバイト位置の信号のみアクティブになります。(*4)



11.7. 32 ビット・バスモード(シングル・ライト、ヒット)

32 ビット・バスモード時のシングル・ライト・サイクルで、Row アドレスが前のサイクルと一致した(ヒットした)場合の波形を下図に示します。

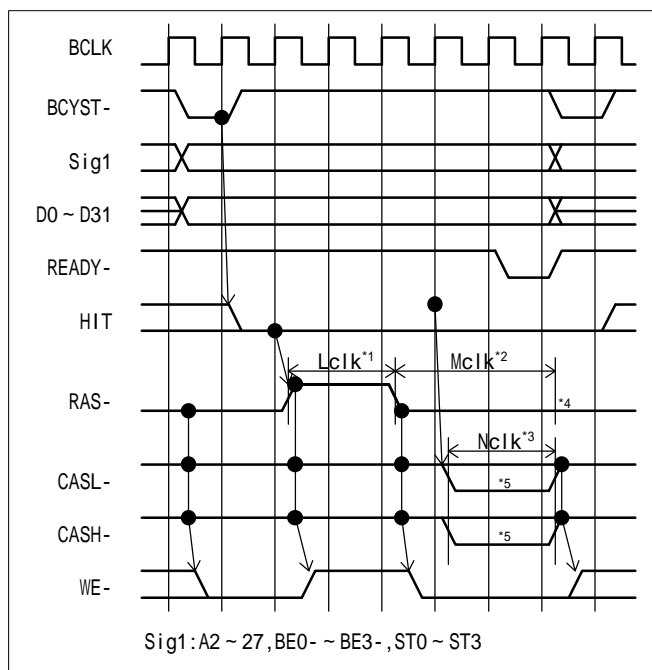
- ◇ 下図に $Nclk(*1)$ で示されている波形の幅は、ポートに設定するライト CAS 幅(1~2)のクロック数です。したがって、最短のサイクルは 1Wait です。
- ◇ サイクルが終了後もページ・モード・アクセスのために RAS-信号が Low に保たれています。(*2)
- ◇ CASL-および CASH-は、アドレスの A2 およびバスイネーブルの BE0-~BE3-の状態により、適切なバンクの適切なバイト位置の信号のみアクティブになります。(*3)



11.8. 32 ビット・バスモード(シングル・ライト、ノーヒット)

32 ビット・バスモード時のシングル・ライト・サイクルで、Row アドレスが前のサイクルと異なっている場合(ノーヒット)の波形を下図に示します。

- ◇ 下図に Lclk/Mclk/Nclk(*1,*2,*3)で示されている波形の幅は、それぞれポートに設定するプリチャージ幅(1~3)/RAS 幅(2~4)/ライト CAS 幅(1~2)のクロック数です。したがって、最短のサイクルは 3Wait です。
- ◇ サイクルが終了後もページ・モード・アクセスのために RAS-信号が Low に保たれています。(*4)
- ◇ CASL-および CASH-は、アドレスの A2 およびバスイネーブルの BE0~BE3-の状態により、適切なバンクの適切なバイト位置の信号のみアクティブになります。(*5)

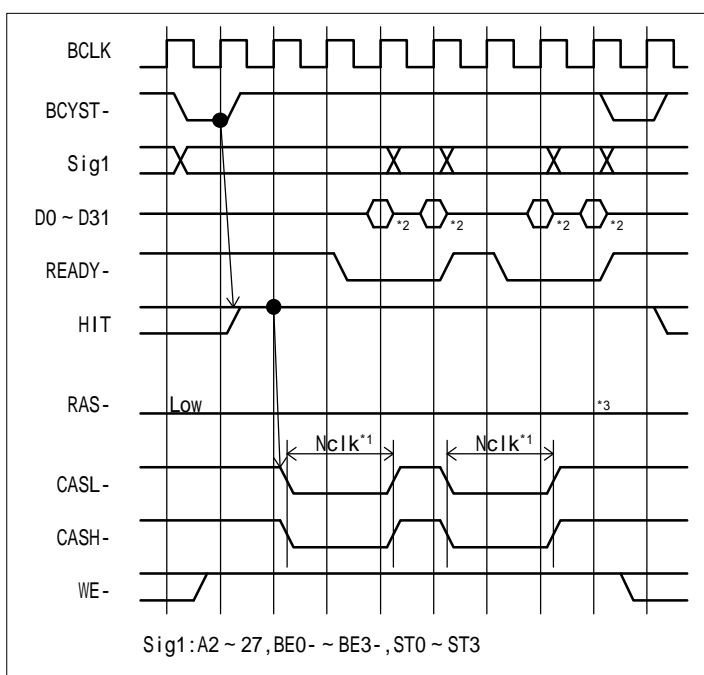


11.9. 32 ビット・バスモード (バースト・リード、インターリーブ)

32 ビット・バスモードで DRAM をインターリーブ・モード、バースト・リード・サイクル、Row アドレスが前のサイクルと一致した (ヒットした) 場合の波形を下図に示します。

Row アドレスが前のサイクルと一致しない場合 (ミスヒット) や、リフレッシュ・サイクル後のサイクルでは、サイクルの開始部分が異なります。この開始サイクルは、シングル・リード・サイクルの時と同じになりますので、そちらを参照して下さい。

- ◇ 下図に $N_{clk}(*1)$ で示されている波形の幅は、ポートに設定するリード CAS 幅(1~3)のクロック数です。したがって、最短のサイクルは 3-1-1-1 形式になります。
- ◇ データは一度に 64 ビット読み出され、下位の 32 ビットが最初のデータ・サンプルで CPU に引き渡され、上位の 32 ビットが次のクロックで CPU に引き渡されます。(*2)
- ◇ サイクルが終了後もページ・モード・アクセスのために RAS-信号が Low に保たれています。(*3)

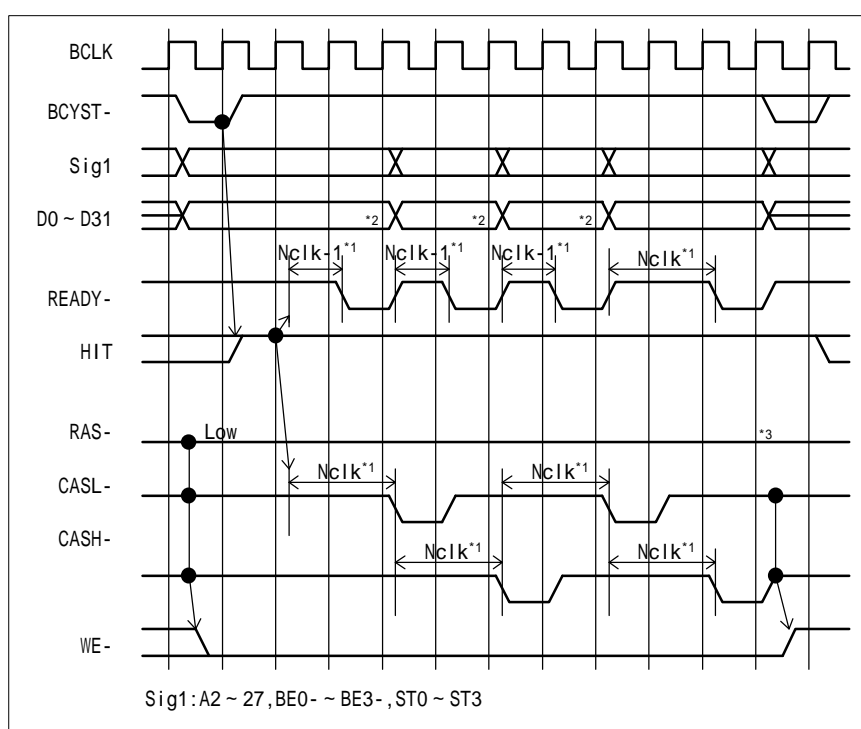


11.10.32 ビット・バスモード(バースト・ライト、インターリーブ)

32 ビット・バスモードで DRAM をインターリーブ・モード、バースト・ライト・サイクル、Row アドレスが前のサイクルと一致した (ヒットした) 場合の波形を下図に示します。

Row アドレスが前のサイクルと一致しない場合 (ミスヒット) や、リフレッシュ・サイクル後のサイクルでは、サイクルの開始部分が異なります。この開始サイクルは、シングル・ライト・サイクルの時と同じになりますので、そちらを参照して下さい。

- ◇ 下図に $N_{clk}(*1)$ で示されている波形の幅は、ポートに設定するライト CAS 幅(1~2)のクロック数です。したがって、最短のサイクルは 3-1-1-2 形式です。
- ◇ ライト・データは、DRAM のコントロール回路がスルー・ラッチしてから DRAM へ与える事により、CAS 信号の立ち下がりへのセットアップ/ホールド・タイムを保証しています。(*2)
- ◇ サイクルが終了後もページ・モード・アクセスのために RAS-信号が Low に保たれています。(*3)



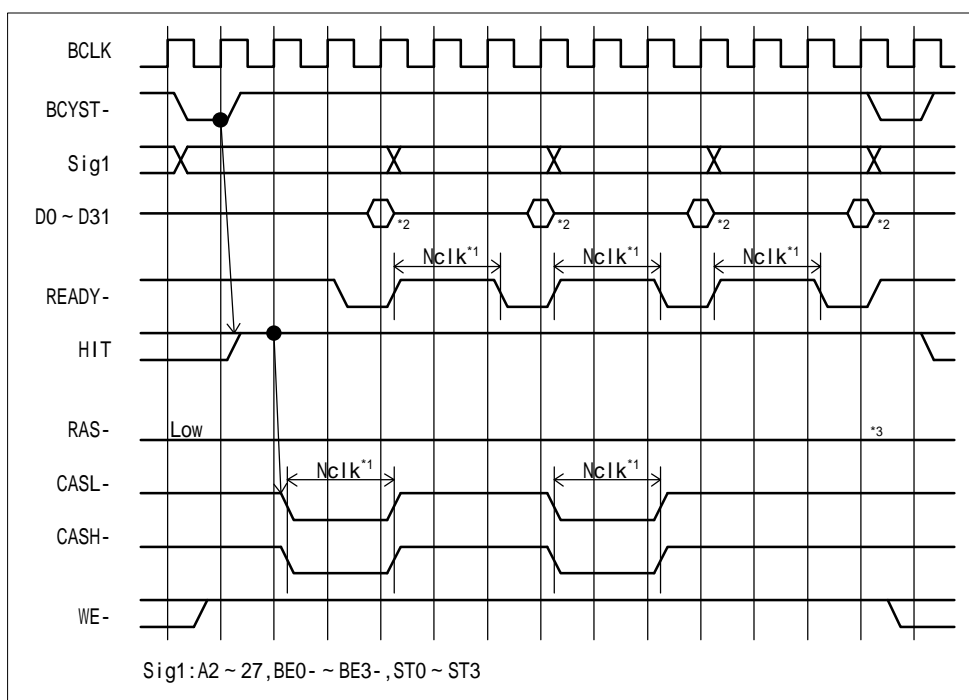
11.11.32 ビット・バスモード(バースト・リード、ノン・インターリーブ)

32 ビット・バスモードで DRAM をノン・インターリーブ・モード、バースト・リード・サイクル、Row アドレスが前のサイクルと一致した(ヒットした)場合の波形を下図に示します。

ノン・インターリーブ・モードは、インターリーブ用の回路を READY 制御して、ノン・インターリーブと同等のサイクルをシミュレートしているだけです。したがって、実際のサイクルとは異なる場合があります。

Row アドレスが前のサイクルと一致しない場合(ミスヒット)や、リフレッシュ・サイクル後のサイクルでは、サイクルの開始部分が異なります。この開始サイクルは、シングル・リード・サイクルの時と同じになりますので、そちらを参照して下さい。

- ◇ 下図に Nclk(*1)で示されている波形の幅は、ポートに設定するリード CAS 幅(1~3)のクロック数です。したがって、最短のサイクルは 3-2-2-2 形式です。
- ◇ データは一度に 64 ビット読み出され、下位の 32 ビットが最初のデータ・サンプルで CPU に引き渡され、上位の 32 ビットが次のクロックで CPU に引き渡されます。(*2)
- ◇ サイクルが終了後もページ・モード・アクセスのために RAS-信号が Low に保たれています。(*3)



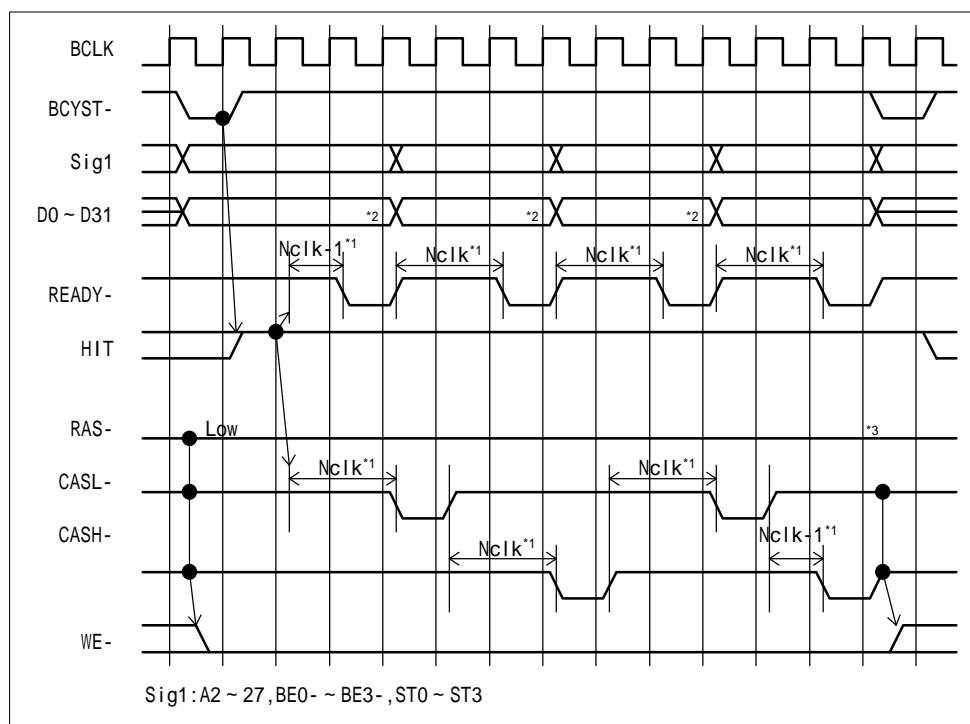
11.12.32 ビット・バスモード(バースト・ライト、ノン・インターリーブ)

32 ビット・バスモードで DRAM をノン・インターリーブ・モード、バースト・ライト・サイクル、Row アドレスが前のサイクルと一致した(ヒットした)場合の波形を下図に示します。

ノン・インターリーブ・モードは、インターリーブ用の回路を READY 制御して、ノン・インターリーブと同等のサイクルをシミュレートしているだけです。したがって、実際のサイクルとは異なる場合があります。

Row アドレスが前のサイクルと一致しない場合(ミスヒット)や、リフレッシュ・サイクル後のサイクルでは、サイクルの開始部分が異なります。この開始サイクルは、シングル・ライト・サイクルの時と同じになりますので、そちらを参照して下さい。

- ◇ 下図に Nclk(*1)で示されている波形の幅は、ポートに設定するライト CAS 幅(1~2)のクロック数です。したがって、最短のサイクルは 3-2-2-2 形式です。
- ◇ ライト・データは、DRAM のコントロール回路がスルー・ラッチしてから DRAM へ与える事により、CAS 信号の立ち下がりへのセットアップ/ホールド・タイムを保証しています。(*2)
- ◇ サイクルが終了後もページ・モード・アクセスのために RAS-信号が Low に保たれています。(*3)



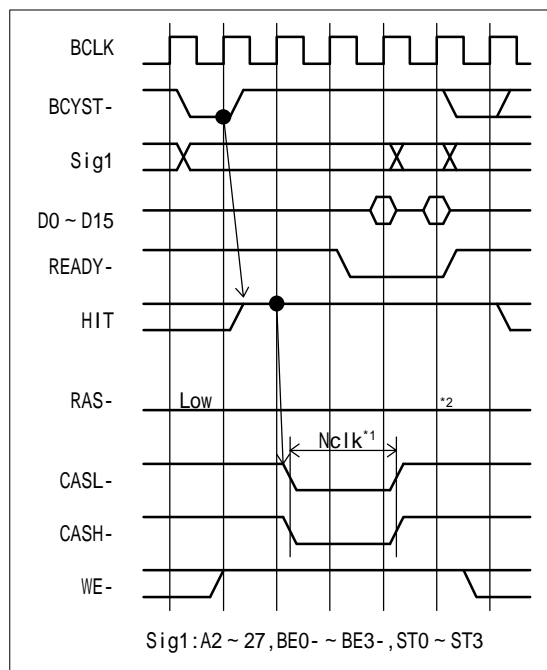
11.13.16 ビット・バスモード (シングル・リード)

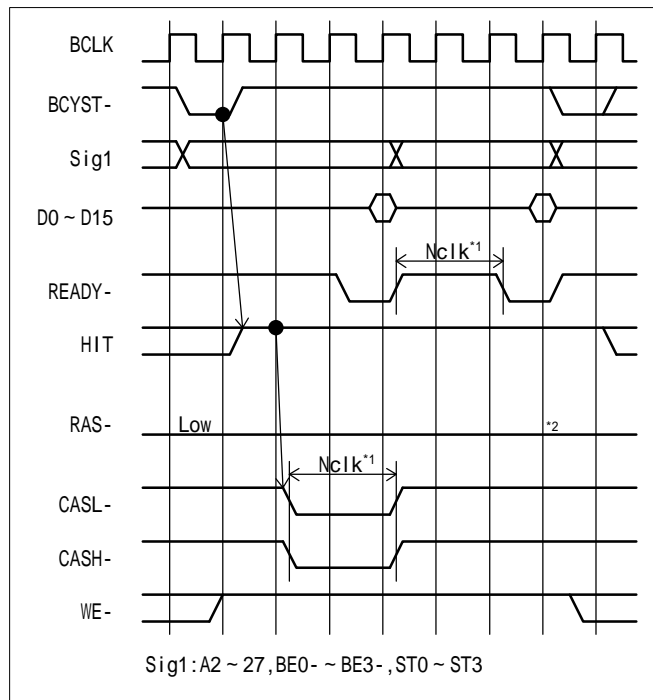
16 ビット・バスモード時のバイトおよびハーフ・ワード・サイズのシングル・リード・アクセスは、32 ビット・バスモード時と同じです。ワード・サイズでのシングル・リードに限りバースト・サイクルと同じように振る舞うため、32 ビット・バスモード時とは異なります。

16 ビット・バスモード時のワード・サイズのシングル・リード・サイクルで、Row アドレスが前のサイクルと一致した (ヒットした) 場合の波形を下図に示します。最初の図はインターリーブ・モード、2 番目の図はノン・インターリーブ・モードの時を示しています。

Row アドレスが前のサイクルと一致しない場合 (ミスヒット) や、リフレッシュ・サイクル後のサイクルでは、32 ビット時のシングル・リード・サイクルの時と同じになりますので、そちらを参照して下さい。

- ◇ 下図に $N_{clk}(*1)$ で示されている波形の幅は、ポートに設定するリード CAS 幅 (1~3) のクロック数です。したがって、最短のサイクルはインターリーブ・モード時で 3-1 形式、ノン・インターリーブ・モード時で 3-2 形式です。
- ◇ サイクルが終了後もページ・モード・アクセスのために RAS-信号が Low に保たれています。(*2)
- ◇ 下位 / 上位のそれぞれ 32 ビット・バンクは同時に読み出され、アドレスの A1 の状態により適切なバンクのデータが下位 16 ビット、上位 16 ビットの順で CPU に供給されます。





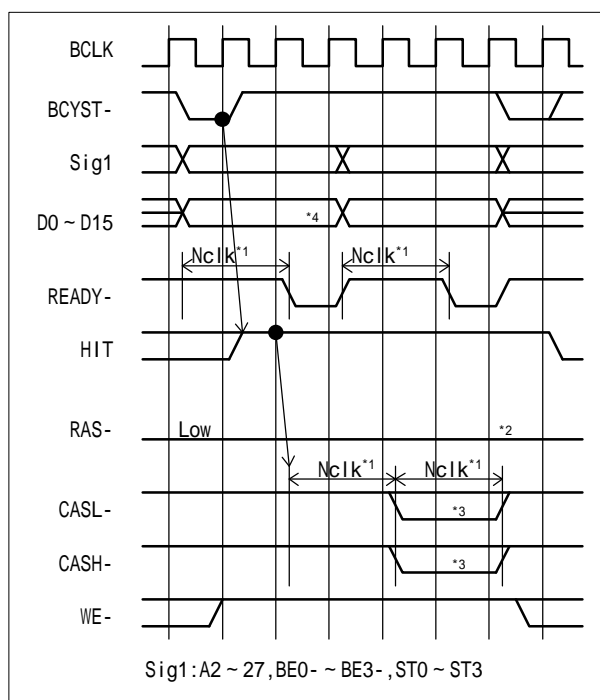
11.14.16 ビット・バスモード(シングル・ライト)

16ビット・バスモード時のバイトおよびハーフ・ワード・サイズのシングル・ライト・アクセスは、32ビット・バスモード時と同じです。ワード・サイズでのシングル・ライトに限りバースト・サイクルと同じように振る舞うため、32ビット・バスモード時と異なります。

16ビット・バスモード時のワード・サイズのシングル・ライト・サイクルで、Rowアドレスが前のサイクルと一致した(ヒットした)場合の波形を下図に示します。このサイクルは、インターリーブ・モード、ノン・インターリーブ・モード共に同じ波形になります。

Rowアドレスが前のサイクルと一致しない場合(ミスヒット)や、リフレッシュ・サイクル後のサイクルでは、32ビット時のシングル・ライト・サイクルの時と同じになりますので、そちらを参照して下さい。

- ◇ 下図に $Nclk^{*1}$ で示されている波形の幅は、ポートに設定するライト CAS 幅(1~2)のクロック数です。したがって、最短のサイクルは 2-2 形式になります。
- ◇ サイクルが終了後もページ・モード・アクセスのために RAS-信号が Low に保たれています。(*2)
- ◇ CASL-および CASH-は、アドレスの A2 およびバスイネーブルの BE0-~BE3-の状態により、適切なバンクの適切なバイト位置の信号のみアクティブになります。(*3)
- ◇ 1回目のサイクルのライト・データは、DRAM のコントロール回路がスルー・ラッチすることにより、1回の CAS 信号で書き込むようにしています。

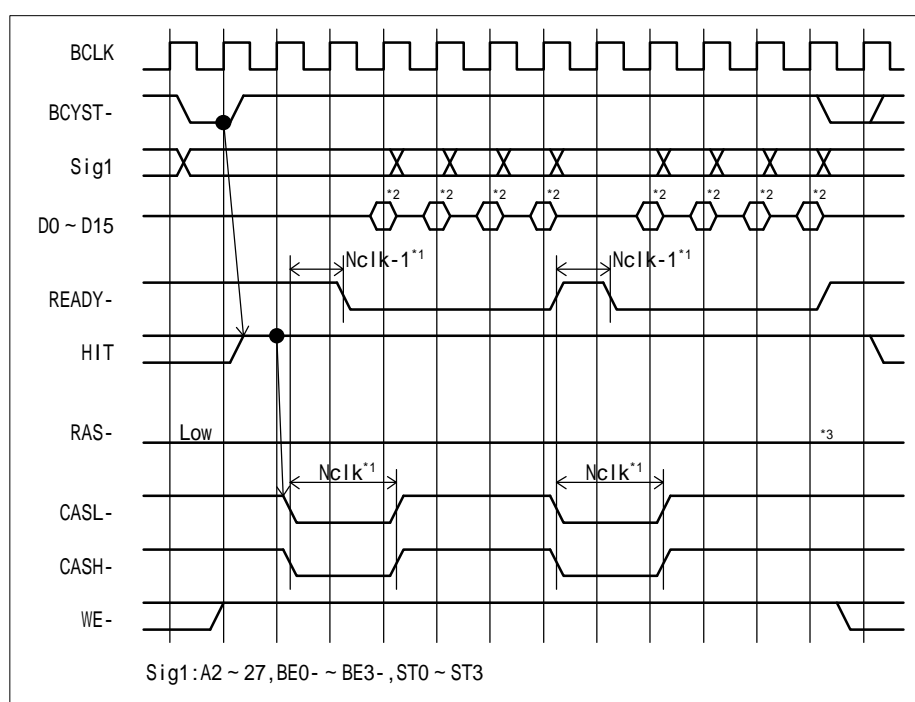


11.15.16 ビット・バスモード(バースト・リード、インターリーブ)

16 ビット・バスモードで DRAM をインターリーブ・モードで用いている時、バースト・リード・サイクルで、Row アドレスが前のサイクルと一致した(ヒットした)場合の波形を下図に示します。

Row アドレスが前のサイクルと一致しない場合(ミスヒット)や、リフレッシュ・サイクル後のサイクルでは、サイクルの開始部分が異なるだけなので、32 ビット・バスモード時のシングル・リード・サイクルを参考して下さい。

- ◇ 下図に $N_{clk}(*1)$ で示されている波形の幅は、ポートに設定するリード CAS 幅(1~3)のクロック数です。したがって、最短のサイクルは 3-1-1-1-1-1-1 形式になります。
- ◇ データは一度に 64 ビット読み出され、下位から順に 16 ビットずつデータが CPU に引き渡されます。(*2)
- ◇ サイクルが終了後もページ・モード・アクセスのために RAS-信号が Low に保たれています。(*3)

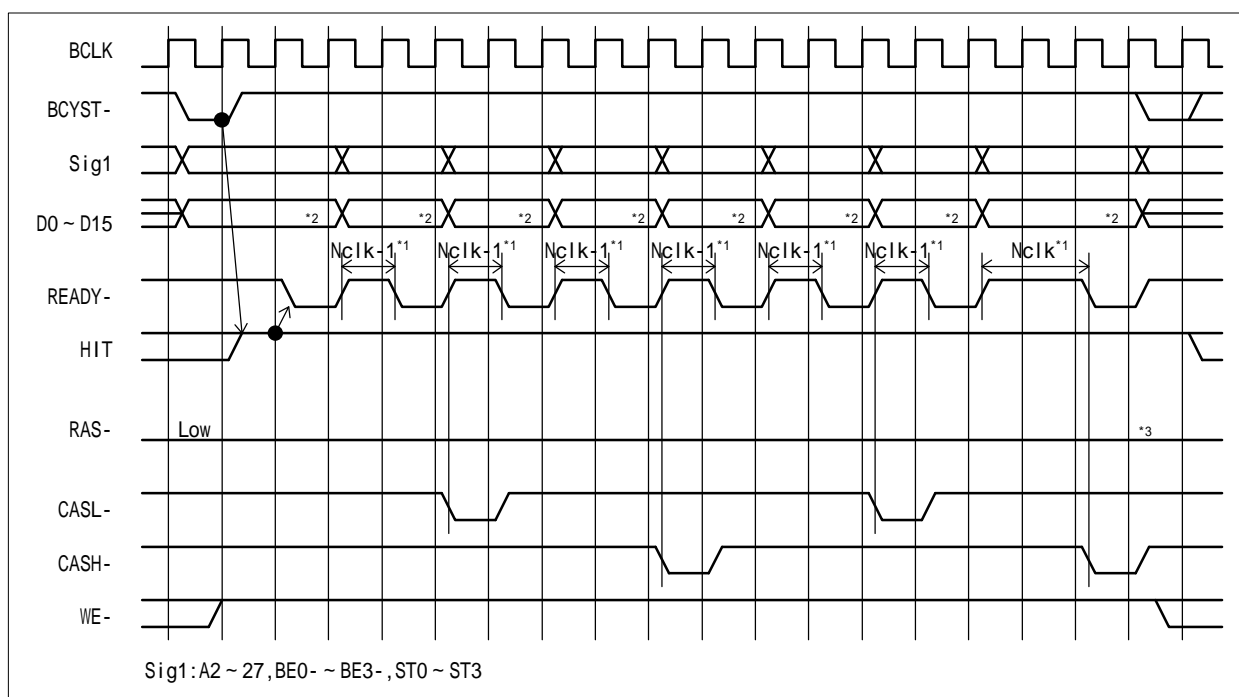


11.16.16 ビット・バスモード(バースト・ライト、インターリーブ)

16 ビット・バスモードで DRAM をインターリーブ・モードで用いている時、バースト・ライト・サイクルで、Row アドレスが前のサイクルと一致した(ヒットした)場合の波形を下図に示します。

Row アドレスが前のサイクルと一致しない場合(ミスヒット)や、リフレッシュ・サイクル後のサイクルでは、サイクルの開始部分が異なるだけなので、32 ビット・バスモード時のシングル・ライト・サイクルを参考して下さい。

- ◇ 下図に Nclk(*1)で示されている波形の幅は、ポートに設定するライト CAS 幅(1~2)のクロック数です。したがって、最短のサイクルは 3-1-1-1-1-1-2 形式になります。
- ◇ ライト・データは、DRAM のコントロール回路がスルー・ラッチしてから DRAM へ与える事により、CAS 信号の立ち下がりへのセットアップ/ホールド・タイムを保証しています。(*2)
- ◇ サイクルが終了後もページ・モード・アクセスのために RAS-信号が Low に保たれています。(*3)



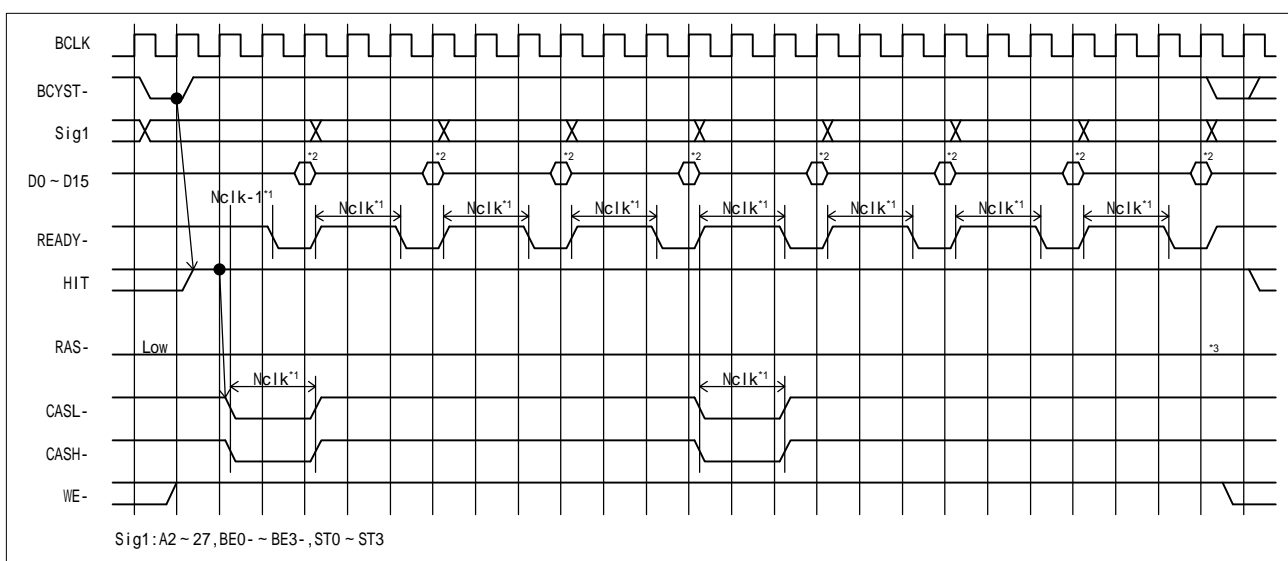
11.17.16 ビット・バスモード(バースト・リード、ノン・インターリーブ)

16 ビット・バスモードで DRAM をノン・インターリーブ・モード、バースト・リード・サイクル、Row アドレスが前のサイクルと一致した(ヒットした)場合の波形を下図に示します。

ノン・インターリーブ・モードは、インターリーブ用の回路を READY 制御して、ノン・インターリーブと同等のサイクルをシミュレートしているだけです。したがって、実際のサイクルとは異なる場合があります。

Row アドレスが前のサイクルと一致しない場合(ミスヒット)や、リフレッシュ・サイクル後のサイクルでは、サイクルの開始部分が異なるだけです。これらのサイクルは、32 ビット・バスモード時のシングル・リード・サイクルを参照して下さい。

- ◇ 下図に $N_{clk}(*1)$ で示されている波形の幅は、ポートに設定するリード CAS 幅(1~3)のクロック数です。したがって、最短のサイクルは 3-2-2-2-2-2-2 形式になります。
- ◇ データは一度に 64 ビット読み出され、下位から順に 16 ビットずつデータが CPU に引き渡されます。(*2)
- ◇ サイクルが終了後もページ・モード・アクセスのために RAS-信号が Low に保たれています。(*3)

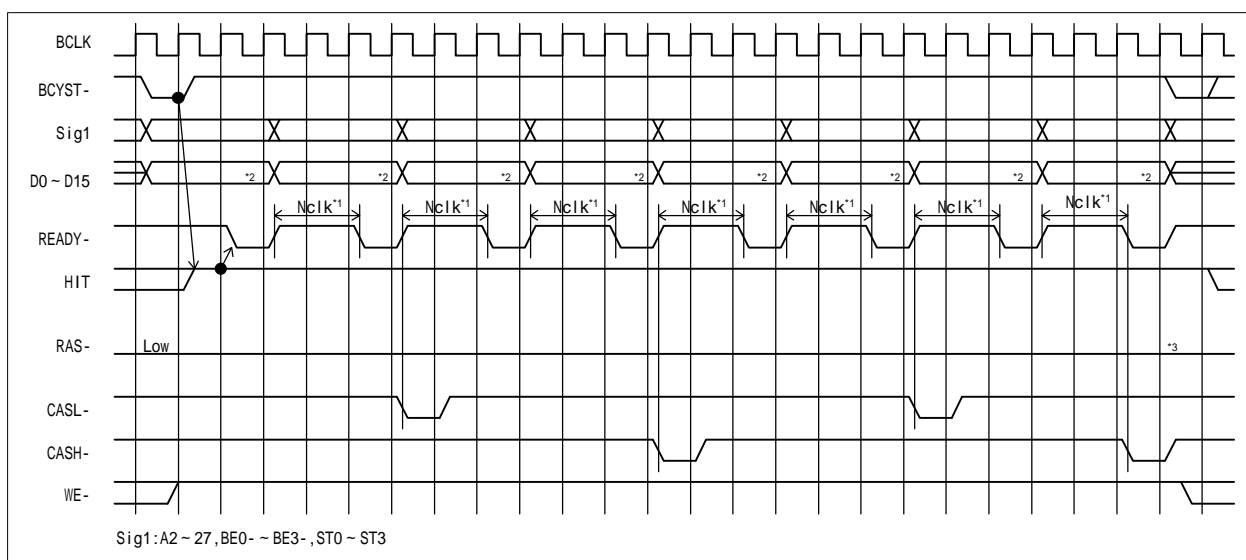


11.18.16 ビット・バスモード(バースト・ライト、ノン・インターリーブ)

16 ビット・バスモードで DRAM をノン・インターリーブ・モード、バースト・ライト・サイクル、Row アドレスが前のサイクルと一致した(ヒットした)場合の波形を下図に示します。ノン・インターリーブ・モードは、インターリーブ用の回路を READY 制御して、ノン・インターリーブと同等のサイクルをシミュレートしているだけです。したがって、実際のサイクルとは異なる場合があります。

Row アドレスが前のサイクルと一致しない場合(ミスヒット)や、リフレッシュ・サイクル後のサイクルでは、サイクルの開始部分が異なるだけです。これらのサイクルは、32 ビット・バスモード時のシングル・ライト・サイクルを参照して下さい。

- ◇ 下図に Nclk(*1)で示されている波形の幅は、ポートに設定するライト CAS 幅(1~2)のクロック数です。したがって、最短のサイクルは 3-2-2-2-2-2-2 形式になります。
- ◇ ライト・データは、DRAM のコントロール回路がスルー・ラッチしてから DRAM へ与える事により、CAS 信号の立ち下がりへのセットアップ/ホールド・タイムを保証しています。(*2)
- ◇ サイクルが終了後もページ・モード・アクセスのために RAS-信号が Low に保たれています。(*3)



12. マスカブル割り込みを使用したアプリケーションの開発

RTE-V830-PC 上でマスカブル割り込みを使用したアプリケーションの開発を行う場合の方法と制限事項について説明します。

12.1. 割り込みベクタ

V830 の割り込みベクタ領域である FFFF-FE00H ~ FFFF-FFFFH 番地は、ROM により固定されていて書き換えることが出来ません。そこで Multi 用モニタ ROM では、SRAM 上に代替のベクタ領域を用意し、FFFF-FE00H ~ FFFF-FFFFH 番地のベクタには、その代替ベクタ領域への分岐命令が置かれています。

例えば、例外コードが FEE0H の割り込みが発生すると、CPU の割り込み機能により FFFF-FEE0H 番地に分岐します。そこには代替ベクタ領域のオフセット FFE0H 番地への分岐命令があります。したがって、ユーザ・プログラムでは、この代替ベクタ領域を本来のベクタ領域と同じように書き換えれば、割り込み発生時にユーザ・プログラムの割り込み処理ルーチンに分岐するように出来ます。

通常の V830 のプログラムと異なるのは、通常はベクタ領域は ROM 化の時点で固定されており、プログラムで設定（書き換える）必需はありません。しかし、RTE-V830-PC 上で Multi を使用したプログラムの場合、プログラムでベクタを書き換えてから、割り込みを許可する必需があります。

代替ベクタ領域は、SRAM 上の FE07-0000H ~ FE07-01FFH にあります（実際には SRAM のイメージが出ますので、他のアドレスからでも参照 / 変更可能ですが、CPU が最初に参照する割り込みベクタが分岐するのは FE07-0000H ~ FE07-01FFH です）。したがって、前述の例外コード FFE0H の割り込みの場合、目的の割り込み処理に分岐する命令を FE07-00E0H 番地に書き込みます。

V830 の場合、CPU がキャッシュ・メモリを内蔵していますので、ベクタを書き換えた後にキャッシュ・フラッシュ操作が必需となります。この操作を忘れると、代替ベクタ書き換え前のベクタが使用されてしまうことがあります。

代替えベクタを書き換えるためのプログラム例を下記に示します（割り込み処理ルーチンから代替えベクタ領域への相対アドレスが 26Bit 以内の場合）。

```

void SetAJump(int addr, int jmpdest) /* ベクタ設定ルーチン */
/* int addr;                      address where we're storing the 'jr' */
/* int jmpdest;                   address where the 'jr' jumps to */
{
    int offset;
    unsigned inst;
    unsigned int *p ;

    offset = jmpdest - addr;
    inst = 0xa8000000 /* 'jr' opcode */ | (offset & 0x3ffffff);
    *((UINT16 *) (addr + 0)) = (inst >> 16) & 0xffff ;
    *((UINT16 *) (addr + 2)) = (inst & 0xffff) ;

    cache_clr() ; /* cache flash ASM で用意する */
}
.....
void __interrupt IntEntry() /* 割り込み処理ルーチン */
{
    .....
}
.....
main()
{
    .....
    SetAJump((int)((0xfffffee0 - 0xfffffe00) + 0xfe070000) ,(int)IntEntry) ;
    /* 目的の割り込みの元のベクタ・アドレス */
}

```

12.2. 内蔵RAM ベクタ

V830 の場合、CPU の機能として、マスカブル割り込みのベクタを内蔵 RAM に配置することが可能です（システム・レジスタ HCCW の IHA ビット）。この機能を使用した場合は、V830 の通常のプログラムと、RTE-V830-PC 上で Multi を使用したプログラムの場合で、ベクタの設定に関して異なることはありません。

内蔵 RAM のベクタの使用方法や、内蔵 RAM の内容の変更方法については、CPU のマニュアルを参照して下さい。

12.3. 一般的な制限事項/注意事項

マスカブル割り込みを使用したアプリケーションをデバッグする上での制限事項と注意事項を下記に示します。

- 1) 代替えベクタの設定の設定前に割り込みが発生した場合や、代替えベクタを正しく設定しないで割り込みが発生した場合には、割り込みの発生時点でのプログラム位置でブレイクします。これは、代替えベクタの初期値がモニタ ROM のブレイク処理ルーチンへの分岐命令になっているためです。ただし、内蔵 RAM に配置したベクタを使用している場合は、内蔵 RAM が初期化されていないため、ベクタの設定を正常に出来なかった場合の結果は予測不能です。
- 2) 代替えベクタ領域から割り込み処理ルーチンまでの相対アドレスが 26Bit を超える場合、割り込み処理ルーチンへの分岐のために、少なくとも 1 つ以上のレジスタの値を壊すか、分岐の中継点を作る必要があります。
- 3) 代替えベクタ領域は、ROM モニタの管理領域として保護されているため、プログラムのダウンロードで書き換えることは出来ません。したがって、ソース・プログラム上

ベクタ領域を独立したセクションとして定義し、リンク時のパラメータによりそのセクションを代替ベクタ領域に割り付ける方法も考えられますが、この方法はダウンロードに失敗してしまうため使用できません。

- 4) 代替ベクタ領域を書き換えた直後に CPU 内蔵のキャッシュ・メモリをフラッシュして下さい。この操作を忘れると、代替ベクタ書き換え前のベクタが使用されてしまうことがあります。
- 5) 割り込み関係を含む全てのペリフェラルは、ボード上のリセット・スイッチによってのみ初期化されます。したがって、一端プログラムを実行した後に、プログラムを再ロードして動作させる場合、前のプログラム実行による影響がペリフェラル上に残ってしまいます。したがって、ペリフェラルを使用するプログラムの場合、一端プログラムを動作させ、再度プログラムを始めから動作させる場合は、rteserv をディスコネクトし、RTE-V830-PC のリセット・ボタンを押し、rteserv を再コネクトする、という手順を踏んで下さい。
- 6) プログラムの先頭で一端 DI (割り込み禁止) 状態にしてから、ペリフェラルやベクタの設定をした後、EI (割り込み許可) 状態にするようにして下さい。

12.4. ブレーク・ポイント使用に関する制限事項 / 注意事項

ブレーク・ポイントを割り込み処理ルーチン内に設定しブレークさせることも出来ますし、その後割り込み処理ルーチン内をシングル・ステップすることも可能です。しかしその場合、下記に示す制限事項 / 注意事項がありますのでご注意下さい。

- 1) ブレーク中は全てのマスカブル割り込みは受け付けません。
- 2) シングル・ステップ機能は、次の命令にテンポラリ・ブレーク・ポイントを設定する方式を取っています。したがって、EI (割り込み許可) 状態のユーザ・プログラムをシングル・ステップする場合、シングル・ステップ中にも割り込みを受け付けます。
- 3) シングルステップによって割り込み処理ルーチンから抜けることは出来ません (具体的には、割り込みルーチンの最後の " }" でのシングルステップが出来ません)。同様に、iret 命令のシングルステップも出来ません。
- 4) デバッガの " Return " 機能で、割り込み処理ルーチンから元のルーチンへ戻ることは出来ません。

13. ボード上のタイマ割り込みをアプリケーションで使用する場合

RTE-V830-PC で、ボード上の SCC2691 のタイマ割り込みをマスカブル割り込みとして使用するアプリケーションの開発を行う場合の方法と制限事項について説明します。

13.1. 準備

SCC2691 のタイマ割り込みをマスカブル割り込みとして使用するには、RTE-V830-PC に下記のものを実装されていなければなりません。

- 1) M493PLD04Z もしくは PLD04Z (ボード上の U17 に実装された PLD の番号)
- 2) Multi 用 ROM モニタ ROM Version 1.xx (ただし Version 1.13 以降)

13.2. 制限事項 / 注意事項

タイマ割り込みを使用するには次の制限事項および注意事項があります。

- 1) ボード上の SW1 の 3 と 4 を ON にして下さい。これにより、Multi 用モニタ ROM はタイマの初期化を行わなくなります。したがって、プロファイル機能を含む Multi の持っている時間計測に関する機能は使用できなくなります。
- 2) RS232C 経由で Multi と接続することは出来ません。ISA バスを使用して接続して下さい。
- 3) Multi のデバッガから RTE-V830-PC に何らかのアクションを起こすと、通信のための NMI 割り込みが RTE-V830-PC 発生します。この割り込みと SCC2691 からの割り込み (タイマおよびシリアル割り込み) が競合した場合、SCC2691 からの割り込みが捨てられてしまうことがあります。
- 4) Multi を使用してマスカブル割り込みを使用したアプリケーションの開発を行う場合は、幾つかの注意事項があります。詳細については第12章を参照して下さい。

- Memo -

RTE-V830-PC ユーザズ・マニュアル
作成 1996年10月8日 Rev.1.01

M471MNL02

Midas lab