

RTE-NB85E-CB

ユーザース・マニュアル (Rev. 1.05)

改訂履歴

実施日	Revision	章	内容
1999年08月12日	1.00		暫定初版
1999年09月17日	1.01		正式初版
1999年10月23日	1.02	7.2.1	BCC 初期値変更:3C00 -> 3C40
2000年01月17日	1.03	7.2.1	RFS3 初期値誤記修正 0x0032/0016 -> 0x8032/8016
2000年02月03日	1.04	7.2.1	RFS3 初期値修正 0x8032/8016 -> 0x8017/800F
2000年10月06日	1.05	5.4	CPU のバージョン.4 に対応するに伴う、SW-2 の設定の一部の機能を変更

目次

1. はじめに	1
1.1. マニュアル表記について	1
1.2. 使用上の注意事項	1
2. 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. スイッチ 1 (SW1)	4
5.4. スイッチ 2 (SW2)	5
5.5. スイッチ 3 (SW3)	6
5.6. スイッチ 11 - 15 (SW11 - 15)	7
5.7. スイッチ 16 (SW16)	8
5.8. スイッチ 17 (SW17)	8
5.9. 7SEG-LED,POWER-LED,TOVER-LED.....	9
5.10. ROM エミュレータ用テストピン (JROM-EML)	9
5.11. クロック・ソケット (OSC1)	9
5.12. クリスタル・ソケット(JP1).....	9
5.13. VBCLK パッファ切替えジャンパ(JP2).....	10
5.14. ROM ソケット.....	10
5.15. シリアル・コネクタ (JSIO1,JSIO2)	10
5.16. デバッグ用コネクタ(JDCU).....	11
5.17. JGBUS コネクタ (JGBUS)	11
5.18. CPU コネクタ(CN1 - 4).....	12
6. ホスト PC との接続	16
6.1. RS-232C 接続.....	16
7. ハードウェア・リファレンス	17
7.1. メモリ・I/O のマップ	17
7.2. 推奨設定	19
7.2.1. MEMC レジスタ.....	19
7.3. メモリ資源	20
7.3.1. SDRAM (CS3:1000000 - 1FFFFFF).....	20
7.3.2. SRAM (CS7:3C00000 - 3FFFFFF).....	20
7.3.3. UV-EPROM (CS0 :0000000 - 03FFFFFF).....	20
7.3.4. 内蔵 ROM (0000000 - 00FFFFFF, 0100000 - 01FFFFFF).....	20

7.4.	IO マップ	21
7.5.	IO 一覧	21
7.5.1.	SW1 読出しポート(SW1 3800000H [Read Only])	21
7.5.2.	SW2 読出しポート(SW2 3801000H [Read Only])	21
7.5.3.	SW16 読出しポート(SW16 3809000H [Read Only])	22
7.5.4.	7セグメントLED表示データ出力ポート(7SEG-LED 3802000 [Write Only])	22
7.5.5.	タイムオーバ・レディーLEDクリア・パルス(TOVRDY_LED_CLRPLS 3803000H [Write Only])	22
7.5.6.	割り込みコントローラ(PIC:3804000 - 3804020[Read/Write])	23
7.5.7.	UART (TL16C550C:3807000 - 3807070)	24
7.5.8.	TIC (uPD71054 3808000H ~ 380803FH)	25
8.	ソフトウェア	26
8.1.	初期化	26
8.2.	uPD71054 に対する連続アクセス	26
8.3.	ライブラリ	26
8.4.	タイマの使用例	27
9.	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	28
9.1.	割り込みベクタ	28
9.2.	一般的な制限事項 / 注意事項	30
9.3.	ダウンロード時の代替ベクタ領域書換え方法	30
9.4.	ブレーク・ポイント使用に関する制限事項 / 注意事項	31
10.	CPU 端子接続	32
10.1.	一覧	32
10.2.	RESET-	33
10.3.	MWAIT-	33
10.4.	NMI0,1,2, INT0	34
10.5.	INT10,11,12,13	35
10.6.	RXD/INT45	35
10.7.	TXD/INT46	35
10.8.	PORT0/MPXSCZ,PORT1/DSTBZ,PORT2/RDCYZ,PORT3/BUSST	36
10.9.	DMARQ0/INT32,DMARQ1/INT33,DMARQ2/INT34,DMARQ3/INT35	36
10.10.	DMAAK0/INT36,DMAAK1/INT37,DMAAK2/INT38,DMAAK3/INT39	37
10.11.	TC0/INT40,TC1/INT41,TC2/INT42,TC3/INT43	37
10.12.	その他の信号	37
11.	GBUS 個別仕様	38
11.1.	概要	38
11.2.	バス・サイクル	39
11.3.	チップセレクト	40
12.	APPEDIX.A MULTI モニタ	41

12.1. ボードの設置	41
12.1.1. RTE for Win32 のインストール.....	41
12.1.2. SW1 の設定.....	41
12.1.3. その他のSW の設定.....	41
12.1.4. ボードの接続.....	41
12.2. Multi モニタ.....	42
12.2.1. 起動時の 7Seg-LED.....	42
12.2.2. ROM モニタ・ワーク RAM.....	42
12.2.3. モニタ割り込み.....	42
12.2.4. _INIT_SP の設定.....	42
12.2.5. タイマ割り込み.....	42
12.2.6. ハードウェアの初期化.....	42
12.2.7. 特殊命令.....	42
12.3. RTE コマンド.....	43
12.3.1. HELP(?).....	43
12.3.2. INIT.....	43
12.3.3. VER.....	43
12.3.4. SFR コマンド.....	43
13. APPEDIX.B PARTNER モニタ	44
13.1. ボードの設置	44
13.1.1. SW1 の設定.....	44
13.1.2. その他のSW の設定.....	44
13.1.3. ボードの接続.....	44
13.2. PARTNER モニタ.....	45
13.2.1. 起動時の 7Seg-LED.....	45
13.2.2. ROM モニタ・ワーク RAM.....	45
13.2.3. モニタ割り込み.....	45
13.2.4. SP の設定.....	45
13.2.5. ハードウェアの初期化.....	45
13.2.6. 特殊命令.....	45
14. APPEDIX.C GBUS 共通仕様.....	46
14.1. 用語.....	46
14.1.1. CPU ボードとマザー・ボード.....	46
14.1.2. バス・サイクル、マイクロ・サイクル.....	46
14.2. 信号.....	46
14.3. ピン配置.....	50
14.4. 未使用端子の処理.....	51
14.5. GCS-[7:0]の割付け.....	51
14.6. バス・サイクル.....	52
14.6.1. シングル・サイクル.....	52
14.6.2. バースト・サイクル.....	52
14.6.3. GWAITI.....	53

14.6.4. GBTERM.....	54
14.7. タイミング.....	55
14.7.1. セットアップ・タイム.....	55
14.7.2. デイレイ・タイム.....	55

1. はじめに

「RTE-NB85E-CB」は、NEC製のRISCプロセッサNB85E(V850E core)の評価を目的とした評価ボードです。

ボードは、最高50MHzで動作するNB85Eとメモリ、シリアル・インターフェース、拡張用のバスコネクタで構成されます。メモリは、高速SRAMと大容量のSDRAMを標準で搭載しています。SDRAMの制御は、NB85Eが内蔵するメモリコントローラを使用して行います。

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ッガとしてご使用になれます。ご使用になるデバッガによって、ROMに搭載するモニタは異なります。

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

1.1. マニュアル表記について

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

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

1.2. 使用上の注意事項

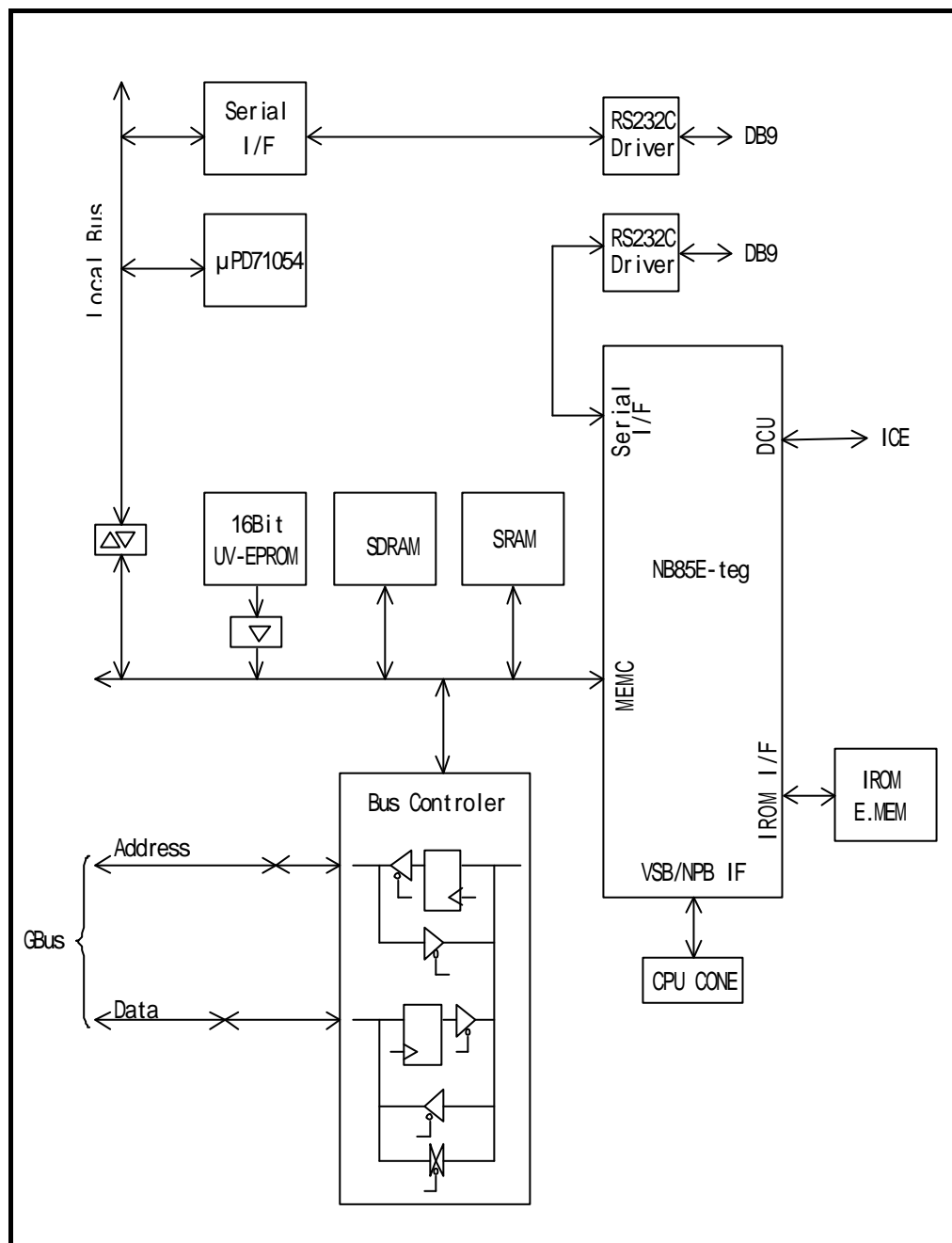
本ボードに実装されるNB85Eチップは評価用サンプルです。評価用以外の用途に使用することはできません。

M883BA01と捺印された基板の仕様は、このマニュアルの内容とは異なりますので、ご注意ください。

Multiは米国Green Hills Software, Incの商標です。

2. 機能

RTE-NB85E-CB の機能ブロックの概要を図に示します。



RTE-NB85E-CB ブロック図

"Local Bus"はMEMCのバスをバッファしたバスで、CPUに同期したバスです。"GBUS"は、CPUバスとは独立したバスで33MHz固定のバスです。

3. 主な特徴

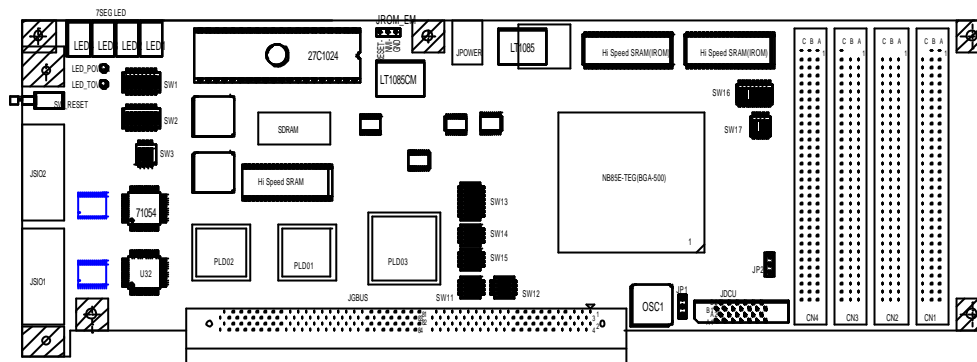
- GreenHills 社の Multi と自社製 PARTNER 用のモニタ ROM を 2 種類用意しています。
- Multi また PARTNER を使用した高級言語レベルでのリアルタイム実行・評価が可能です。
- 内蔵 ROM のエミュレーションメモリを搭載しています。
- 高速 SRAM を 1M-Byte、SDRAM は 16M-Byte(32bit bus access 時)を標準搭載しています。
- シリアル(2ch)のインタフェースを用意しています (1ch は CPU 外部のコントローラ、もう 1ch は CPU 内蔵のコントローラを使用し、モニタは外部コントローラのシリアルを使用)
- タイマ 3ch を搭載しています (モニタで 1ch 使用)
- ROM エミュレータが接続できます。

4. 基本仕様

プロセッサ	NB85E	
CPU クロック	50MHz	
バスクロック	50MHz	
消費電力	+5V (2A)	
メモリ		
EPROM	128KB	64K × 16bit (40pin-DIP) × 1(max.512KB)
IROM-RAM	1MB	256K × 8bit × 4
SRAM	1MB	256K × 16bit × 2
SDRAM	16MB	1M × 16Bit × 4Bank × 2
I/O		
シリアル(2ch)	CPU 内蔵 NS16550 相当	DB9 コネクタ DB9 コネクタ
タイマ	i8254 相当	分解能 500nS
IO ポート	LED(7seg) × 4	表示 / スイッチ入力
その他		
CPU コネクタ	NB85E の IROM IF 用ピンを除く全機能ピンを接続したコネクタ	
32bit 標準外部拡張バス リセット・スイッチ	RTE-CB 標準 32bit I/F(4GB,32bit バス,DMA 対応) Push 式	

5. ボードの構成

下図は RTE-NB85E-CB ボード上の主要な部品の物理的な配置です。本章では、それぞれの部品について説明します。



RTE-NB85E-CB の部品配置図

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)

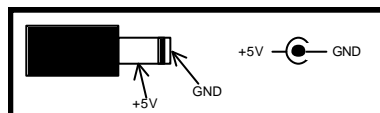
JPOWER コネクタに供給する電源は、以下の通りです。

電圧 : +5V

電流 : 2A(max)

適合コネクタ : Type A (5.5)

極性 :



添付の専用電源 (RTE-PS01)以外の電源を使用しないでください。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. 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 (7.5.1 SW1 読み出しポート(SW1 3800000H [Read Only])を参照)。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、「12.1.2 SW1 の設定」を参照ください。

PARTNERを使用する場合、13.1.1 SW1 の設定」を参照ください。

5.4. スイッチ 2 (SW2)

SW2 は、本ボードの動作を切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 (「7.5.2 SW2 読出しポート(SW2 3801000H [Read Only])」を参照)

番号	信号名	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。 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。 (「7.1 メモリ・I/O のマップ」参照)。
2	TEST	OFF	OFF に設定して下さい。
3	BCLK_LOW	OFF	OSC1 に実装されているオシレータの周波数を設定します。モニタ ROM はこの設定値により ROM や SRAM のウェイト数を変更します。また、ハード的に I/O のウェイト数が切り替わります。 OFF: バスクロックが 33MHz を越える時に設定します。 ON: バスクロックが 33MHz 以下である時に設定します。
4	BSIZE16	OFF	SRAM,SDRAM の H/W 上のバス幅を設定します。 OFF: 32bit 幅に割り付けます。 ON: 16bit 幅に割り付けます。 (この設定に合わせて、MEMC の設定も必要です)
5	NMI/INT0-	ON	モニタが使用する割り込みを指定します。 OFF: NMI を使用します。 ON: INT0 を使用します。
6	CACHE	OFF	キャッシュの初期化時の設定をします。 OFF: 全ての空間を UNCACHE で初期化します。 ON: ボード上のメモリ資源を全て CACHE で初期化します。
7		OFF	モニタ ROM のバージョンによって異なります。
8		OFF	下記の説明参照してください。

モニタ ROM のバージョンが 2.xx(CPU Ver.3 に対応)の場合、以下に従って設定してください。

番号	信号名	出荷時の設定	機能
7	SCPUMODE0	OFF	SW16 で指定したモードに応じて以下の通り設定します。 SW16 の設定 [SCPUMODE1, SCPUMODE0] AUTO [OFF , OFF] <<通常設定
8	SCPUMODE1	OFF	SINGLE MODE0 [OFF , ON] SINGLE MODE1 [ON , OFF] ROMLESS [ON , ON]

注意: Auto 以外は、テスト用ですので、通常は、AUTO の設定でご使用ください。

モニタ ROM のバージョンが 3..00 以上(CPU Ver.4 に対応)の場合、以下に従って設定してください。

番号	信号名	出荷時の設定	機能
7	CACHEMODE 0	ON	SW2-6 が ON(CACHE ON)の時の、モードを指定します。 OFF: ライトスルーモード ON: ライトバックモード
8	CACHEMODE 1	OFF	SW2-7 が ON(ライトバックモード)の時の、ライトアロケートを指定します。 OFF: ライトアロケート禁止 ON: ライトアロケート許可

5.5. 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
2	ROM_TYPE1	OFF	[OFF , ON]: 27C4096 使用時 [ON , OFF]: 27C2048 使用時 [ON , ON]: 27C1024 使用時
3	BANK_DIS	OFF	ROM を前半半分と後半半分のバンクに分けて切替えて使用するかどうかを設定します。モニタ ROM を使用する場合は必ず、OFF にしてください。 OFF: バンクに分けて使用する。 ON: バンクに分けずに連続した領域として使用する。
4	BANK_LOW	OFF	SW3-3 が OFF に設定されている時に、ROM の前半部を有効とするか後半部を有効とするかを設定します。 OFF: 後半部を有効とする ON: 前半部を有効とする。

注意：モニタ ROM には前半部に ROMLESS(及び SINGLE MODE1)で、後半部に SINGLE MODE0 で動作するコードが入っています。SW16(CPU のモード設定)を変更する場合は、それに合わせ、SW3-4 を変更してください。

5.6. スイッチ 11 - 15 (SW11 - 15)

SW11 - 15 は、CPU の端子に接続している本基板内の信号線を物理的にカットするためのスイッチです。出荷時全ての設定は、ON (接続された状態) になっています。外部で使用する場合にのみ、OFF に設定してください。但し、内部で使用している資源が不要な場合に限りです。

備考：以下の表は、CPU 端子と最終的な内部の資源名を書いています。

[SW11]

番号	CPU 端子名	出荷時の設定	内部で使用している資源
1	NMI0	ON	ROM エミュレータからの割り込み
2	NMI1	ON	GBUS の割り込み要求信号 GINT0-
3	NMI2	ON	モニタの割り込み(NMI)
4	INT0	ON	モニタの割り込み(INT0)

[SW12]

番号	CPU 端子名	出荷時の設定	内部で使用している資源
1	PORT0/MPXCSZ	ON	JSIO2 の CTS-
2	PORT1/DSTBZ	ON	JSIO2 の DSR-
3	PORT2/RDCYZ	ON	JSIO2 の RTS-
4	PORT3/BUSST	ON	JSIO2 の DTR-

[SW13]

番号	CPU 端子名	出荷時の設定	内部で使用している資源
1	INT32/DMARQ0/TBO0	ON	GBUS の DMARQ0-
2	INT33/DMARQ1/TBO1	ON	GBUS の DMARQ1-
3	INT34/DMARQ2/TBO2	ON	GBUS の DMARQ2-
4	INT35/DMARQ3/TBO3	ON	GBUS の DMARQ3-
5	INT36/DMAAK0/TBO4	ON	GBUS の DMAAK0-
6	INT37/DMAAK1/TBO5	ON	GBUS の DMAAK1-
7	INT38/DMAAK2/TBO6	ON	GBUS の DMAAK2-
8	INT39/DMAAK3/TBO7	ON	GBUS の DMAAK3-

[SW14]

番号	CPU 端子名	出荷時の設定	内部で使用している資源
1	INT40/TC0/TBO8	ON	GBUS の 129pin
2	INT41/TC1/TBO9	ON	GBUS の 130pin
3	INT42/TC2/TBO10	ON	GBUS の 131pin
4	INT43/TC3/TBO11	ON	GBUS の 132pin
5	INT45/RXD/TBO13	ON	JSIO2 の RXD(受信データ)
6		OFF	未使用
7		OFF	未使用
4		OFF	未使用

[SW15]

番号	CPU 端子名	出荷時の設定	内部で使用している資源
1	INT10/TBI10	ON	GBUS の GINT1-
2	INT11/TBI11	ON	GBUS の GINT2-

3	INT12/TBI12	ON	GBUS の GINT3-
4	INT13/TBI13	ON	TIC の OUT1

5.7. スイッチ 16 (SW16)

SW16 は、CPU の MODE[0..6]端子の状態を設定するスイッチです。OFF で 1、ON で 0 です。

主な設定値を以下に示します。

Initial Bus Wide = 32-bit

番号	CPU 端子名	Single Mode0	Single Mode1	Romless
1	MODE0	ON	OFF	ON
2	MODE1	ON	ON	ON
3	MODE2	ON	ON	ON
4	MODE3	OFF	OFF	ON
5	MODE4	OFF	OFF	OFF
6	MODE5	OFF	OFF	OFF
7	MODE6	OFF	OFF	OFF
8	ROM16(注.2)	OFF	OFF	OFF

Initial Bus Wide = 16-bit

番号	CPU 端子名	Single Mode0	Single Mode1	Romless
1	MODE0	ON	OFF	ON
2	MODE1	ON	ON	OFF
3	MODE2	OFF	OFF	OFF
4	MODE3	OFF	OFF	ON
5	MODE4	OFF	OFF	OFF
6	MODE5	OFF	OFF	OFF
7	MODE6	OFF	OFF	OFF
8	ROM16(注.2)	OFF	OFF	OFF

《注意事項》

1. 出荷時の設定は、Initial Bus Wide = 32-bit, Single Mode0 です。通常この設定でご使用ください。
2. ROM16 は、テスト用です。常に OFF の状態で使用してください。
3. CPU のモードを変更した場合、それにあったモニタの種類の選択が必要です。
SW3-4 を Single Mode0 では OFF、それ以外のモードでは、ON にしてください。
(「5.5 スイッチ 3 (SW3)」を参照)

5.8. スイッチ 17 (SW17)

SW17 は、CPU の状態を設定するスイッチです。OFF で 1、ON で 0 です。

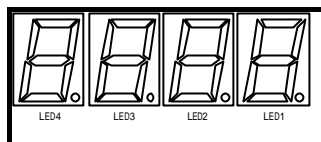
番号	CPU 端子名	出荷時の設定	機能
1	CKSEL0	ON	OSC1 の 2 倍の周波数が CPU のクロックになります。
2	CKSEL1	ON	
3	VOESEL	OFF	リセット直後、VSB Output Signal を有効にします。
4	ROMAA2	OFF	テスト用です。常に OFF で使用してください。

《注意事項》

1. 通常は、全て、出荷時の設定でご使用ください。

5.9. 7SEG-LED,POWER-LED,TOVER-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。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
TOVRDY	タイムオーバ・レディー発生時に点灯、ソフト的にクリアするまで点灯する（「7.5.5タイムオーバ・レディーLED クリア・パルス(TOVRDY_LED_CLRPLS 3803000H [Write Only])」参照）

ボード LED ステータス

5.10. 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、CPU 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（「10.4 NMI」を参照ください。）。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

5.11. 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のオシレータを実装します。

OSC1 は CPU の CPUCLK 端子に 3.3V にレベル変換されて接続されています。

オシレータは、DIP8 ピンタイプ（ハーフタイプ）のものを実装してください。



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願います。

5.12. クリスタル・ソケット (JP1)

JP1 は、CPU に供給するクロックの切替えとクリスタルの実装ソケットの役割を持っています。

OSC1 を CPU のクロックとして使用する場合

JP1 の 1pin と 2pin をショートします。この場合、クリスタルは実装しないでください。

JP1 にクリスタルを実装し、CPU の発振回路を使用する場合

JP1 の 1pin と 3pin の間にクリスタルを実装します。1pin、2pin 間はショートしないでください。

発振が不安定な場合は、C10,C11 を調整してください。（出荷時、10pF が実装されています）

5.13. VBCLK バッファ切替えジャンパ(JP2)

JP2 は、CNx に出力するクロックのバッファの種類を切替えるジャンパです。

1pin - 2pin ショートの場合：74LVTH541 でバッファした信号を出力します。(遅延数 nS)

2pin - 3pin ショートの場合：CY2308 でバッファした信号を出力します。(遅延ゼロ)

備考：遅延が問題になる場合は、CY2308 をご使用ください。但し、PLL を使用していますので、クロックを可変または停止することがある場合には適しません。

5.14. 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。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。

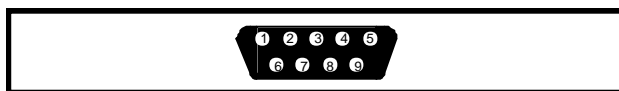
5.15. シリアル・コネクタ (JSIO1,JSIO2)

JSIO1 コネクタは、ボード上のシリアル・コントローラ (TL16C550C) によって制御される RS-232C インターフェース用のコネクタです。

JSIO2 コネクタは、CPU の内蔵シリアル・コントローラによって制御される RS-232C インターフェース用のコネクタです。

コネクタの形状は、PC/AT で用いられる一般的な D-SUB9 ピン (オス) の RS-232C コネクタです。何れも、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。

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,JSIO2 ピン配置 (オス)

JSIO1 ピン番号	JSIO2 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	1 ^{*1}	DCD	入力		
2	2	RxD(RD)	入力	3	2
3	3	TxD(SD)	出力	2	3
4	4	DTR(DR)	出力	1, 6	6, 8
5	5	GND		5	7
6	6	DSR(ER)	入力	4	20
7	7	RTS(RS)	出力	8	5
8	8	CTS(CS)	入力	7	4
9	9 ^{*1}	RI	入力		

JSIO1,2 コネクタ信号

注意事項：

- 1 . JSIO2 の 1 ピン、9 ピンは、ボード内で使用されていません。
- 2 . JSIO2 は、NB85E-TEG の UART と PORT3..0 を使用しています。それぞれの接続状態は、「10.6 RXD/INT45、10.7 TXD/INT46、10.8 PORT0/MPXSCZ,PORT1/DSTBZ,PORT2/RDCYZ,PORT3/BUSST」を参照してください。

5.16. デバッグ用コネクタ(JDCU)

NB85E に内蔵しているデバッグ機能を利用したデバッグ・ツールを接続するためのコネクタです。

ピン番号	信号名	ピン番号	信号名
A1	TRCCLK	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	NC.	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

5.17. 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「11 GBUS 個別仕様」、「14 APPEDIX.C GBUS 共通仕様」を参照してください。

5.18. CPU コネクタ(CN1 - 4)

CPU コネクタの信号は、NB85E と直結した信号です。MEMC バスの多くの信号は、ボード内部で使用していますので、JCPU から信号を引き出す場合は、注意が必要です。尚、信号のレベルは、3.3V です。DIN96 ピンのコネクタが実装できるようになっていますので、用途に応じて、適当なコネクタを実装してご使用ください。(出荷時は未実装です)

	信号名:A 列	信号名:B 列	信号名:C 列
1	+5V	GND	VBD0
2	VBD1	VBD2	VBD3
3	VBD4	VBD5	VBD6
4	VBD7	GND	VBD8
5	VBD9	VBD10	VBD11
6	VBD12	VBD13	VBD14
7	VBD15	GND	VBD16
8	VBD17	VBD18	VBD19
9	VBD20	VBD21	VBD22
10	VBD23	GND	VBD24
11	VBD25	VBD26	VBD27
12	VBD28	VBD29	VBD30
13	VBD31	+5V	GND
14	VBA0	VBA1	VBA2
15	VBA3	VBA4	VBA5
16	VBA6	VBA7	GND
17	VBA8	VBA9	VBA10
18	VBA11	VBA12	VBA13
19	VBA14	VBA15	GND
20	VBA16	VBA17	VBA18
21	VBA19	VBA20	VBA21
22	VBA22	VBA23	VBA24
23	VBA25	VBA26	VBA27
24	GND	(NC)	(NC)
25	VAREQ0	VBWAIT	VBAHLD
26	VBLAST	VBEXDC	VBEXCLK
27	GND	RESET-	VAACK0
28	VBLOCK	VBSTZ	VBWRITE
29	GND	VBCTYP0	VBCTYP1
30	VBCTYP2	VBTTYP0	VBTTYP1
31	GND	VBSTR	VBSEQ0
32	VBSEQ1	VBSEQ2	+3.3V

CN1 コネクタ信号

	信号名:A 列	信号名:B 列	信号名:C 列
1	+5V	GND	VBBENZ0
2	VBBENZ1	VBBENZ2	VBBENZ3
3	VBSIZE0	VBSIZE1	VBDC
4	VDSELPZ	GND	VDCSZ0
5	VDCSZ1	VDCSZ2	VDCSZ3
6	VDCSZ4	VDCSZ5	VDCSZ6
7	VDCSZ7	GND	VBCLK1
8	GND	NMI0	NMI1
9	NMI2	GND	INT0/TBI0
10	INT1/TBI1	INT2/TBI2	INT3/TBI3
11	INT4/TBI4	INT5/TBI5	INT6/TBI6
12	INT7/TBI7	INT8/TBI8	INT9/TBI9
13	INT10/TBI10	+5V	GND
14	INT11/TBI11	INT12/TBI12	INT13/TBI13
15	INT14/TBI14	INT15/TBI15	INT16/TBI16
16	INT17/TBI17	INT18/TBI18	INT19/TBI19
17	INT20/TBI20	INT21/TBI21	INT22/TBI22
18	INT23/TBI23	INT24/TBI24	INT25/TBI25
19	INT26/TBI26	INT27/TBI27	INT28/TBI28
20	INT29/TBI29	INT30/TBI30	INT31/TBI31
21	GND	INT32/DMARQ0/TBO0	INT33/DMARQ1/TBO1
22	INT34/DMARQ2/TBO2	INT35/DMARQ3/TBO3	INT36/DMAAK0/TBO4
23	INT37/DMAAK1/TBO5	INT38/DMAAK2/TBO6	INT39/DMAAK3/TBO7
24	INT40/TC0/TBO8	INT41/TC1/TBO9	INT42/TC2/TBO10
25	INT43/TC3/TBO11	INT44/DMASTP/TBO12	INT45/RXD/TBO13
26	INT46/TXD/TBO14	GND	INT47/TBO15/TAPSM0
27	INT48/TBO16/TAPSM1	INT49/TBO17/TAPSM2	INT50/TBO18/TAPSM3
28	INT51/ELKRT/TBO19/ DDOUT	INT52/EINTLV0/TBO20/ DDOENB	INT53/EINTLV1/TBO21/ DBRESZ
29	INT54/EINTLV2/TBO22 / RESMK	INT55/EINTLV3/TBO23/ MSKSTP	INT56/RINTLV4/TBO24/ MSKNMI0
30	INT57/EINTLV5/TBO25 / MSKNMI1	INT58/EINTLV6/TBO26/ MSKNMI2	INT59/EINTRQ/TBO27/ MSKHRQ
31	INT60/EINTAK/TBO28/ DBRDY	INT61/EASTB/TBO29/ EVASTB	INT62/EDSTB/TBO30/ EVDSTB
32	INT63/ECLRIP/TBO31/ EVCLRIP	GND	+3.3V

CN2 コネクタ信号

	信号名:A 列	信号名:B 列	信号名:C 列
1	+5V	GND	VPD0/EAD0
2	VPD1/EAD1	VPD2/EAD2	VPD3/EAD3
3	VPD4/EAD4	VPD5/EAD5	VPD6/EAD6
4	VPD7/EAD7	GND	VPD8/EAD8
5	VPD9/EAD9	VPD10/EAD10	VPD11/EAD11
6	VPD12/EAD12	VPD13/EAD13	VPD14/EAD14
7	VPD15/EAD15	GND	VPA0
8	VPA1	VPA2	VPA3
9	VPA4	VPA5	VPA6
10	VPA7	VPA8	VPA9
11	VPA10	VPA11	VPA12
12	VPA13	(NC)	(NC)
13	(NC)	+5V	GND
14	VPRETR	VPDACT	VPSTB
15	VPWRITE	VPLOCK	VPUBENZ
16	VPEXRETR	VPEXDACT	GND
17	TBI32	TBI33	TBI34
18	TBI35	TBI36	TBI37
19	TBI38	TBI39	TEST
20	BUNRI	GND	TBO33
21	TBO32	TBO34	VPTCLK
22	PHTEST	TESTN	TBREDZ
23	GND	DBINT	EVTTRG
24	IDBR0	IDBR1	IDBR2
25	(NC)	(NC)	(NC)
26	(NC)	(NC)	(NC)
27	RESET_VB-	(NC)	IORD-
28	IOWR-	HLDK-	HLDREQ-
29	GND	(NC)	(NC)
30	(NC)	STOPZ-	PWM/ASTBZ
31	PORT0/MPXCSZ	PORT1/DSTBZ	PORT2/RDCYZ
32	PORT3/BUSST	GND	+3.3V

CN3 コネクタ信号

	信号名:A 列	信号名:B 列	信号名:C 列
1	+5V	GND	D0
2	D1	D2	D3
3	D4	D5	D6
4	D7	GND	D8
5	D9	D10	D11
6	D12	D13	D14
7	D15	GND	D16
8	D17	D18	D19
9	D20	D21	D22
10	D23	GND	D24
11	D25	D26	D27
12	D28	D29	D30
13	D31	+5V	GND
14	A0/EVAD0	A1/EVAD1	A2/EVAD2
15	A3/EVAD3	A4/EVAD4	A5/EVAD5
16	A6/EVAD6	A7/EVAD7	A8/EVAD8
17	A9/EVAD9	A10/EVAD10	A11/EVAD11
18	A12/EVAD12	A13/EVAD13	A14/EVAD14
19	A15/EVAD15	GND	A16/EVLKRT
20	A17/EVINTLV0	A18/EVINTLV1	A19/EVINTLV2
21	A20/EVINTLV3	A21/EVINTLV4	A22/EVINTLV5
22	A23/EVINTLV6	A24/EVINTRQ	A25/EVINTAK
23	GND	CS0-/RAS0-	CS1-/RAS1-
24	CS2-/RAS2-	CS3-/RAS3-	CS4-/RAS4-
25	CS5-/RAS5-	CS6-/RAS6-	CS7-/RAS7-
26	GND	VBCLK1 ^{*1}	GND
27	BCYST-	RD-	WR0-/CAS0-/DQM0-
28	WR1-/CAS1-/DQM1-	WR2-/CAS2-/DQM2-	WR3-/CAS3-/DQM3-
29	GND	OE-	WE-
30	REFRQ-	SDRAS-	SDCAS-
31	CKE	GND	SDCLK
32	GND	MWAIT-	+3.3V

CN4 コネクタ信号

*1:VBCLK1 は、VBCLK をバッファした信号です。

6. ホスト PC との接続

6.1. RS-232C 接続

モニタ ROM を使用して、ホストマシンとシリアルで接続する際は以下の通りです。

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 のボーレイトの設定は、必ず実施ください。(「12.1.2 SW1 の設定」、「13.1.1 SW1 の設定」の各項を参照ください。)

JSIO1 コネクタとホストマシンを 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。

ホストマシン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、モニタの種類やスイッチ (ボーレイト、CPU モード等) の設定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、デバッグのマニュアルを参照ください。



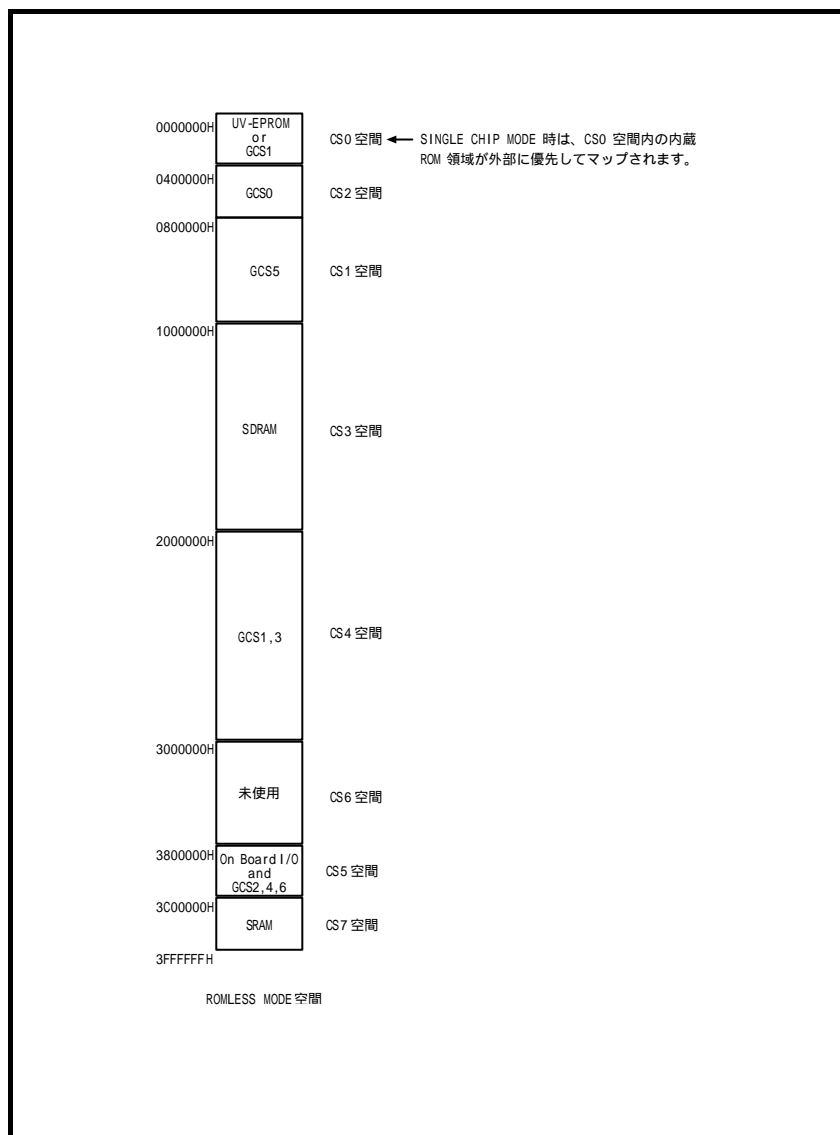
ボードは絶縁物の上に設置してください。通電中に、導電物を基板に接触させると故障の原因になります。

7. ハードウェア・リファレンス

ここでは、RTE-NB85E-CB ボードのハードウェアについて記述します。

7.1. メモリ・I/O のマップ

ボードのメモリと I/O の割付けは、以下の通りです。アドレスモードは、64M モードです。



メモリ・I/O マップ

CS0 空間 (UV-EPROM,GCS1-) :0000000 - 03FFFFFF(4M-byte)

CS0 空間は、SingleChip モード時の内蔵 ROM 空間以外は、UV-EPROM が割付けられるか、または GBUS の GCS1-用の空間として予約されています。この空間は、SW2-2 (FBOOT)が OFF の時は、オンボードの UV-EPROM が割付けられています(「5.4 スイッチ 2 (SW2)」参照)。SW2-2 (FBOOT)が ON の時は、GBUS の GCS1-用の空間として予約されています。GCS1 空間は、CS4 空間からもアクセス可能なため、GBUS の GCS1 空間にフラッシュ ROM を配置することで、フラッシュ ROM の書換えは UV-EPROM のモニタ ROM を用いて行い、その後そのフラッシュ ROM からブートすることが可能になります。

GBUS については「11 GBUS 個別仕様」、「14 APPEDIX.C GBUS 共通仕様」を参照してください。尚、SingleChip モードに設定した場合は、内蔵 ROM の空間(0x0 ~、または、0x100000 ~)にはエミュレーションメモリが優先的に割付けされます。内蔵 ROM の空間は、モニタからのみ書き換えができません。

CS1 空間 (GCS5-) :0800000 - 0FFFFFFF(8M-byte)

CS1 空間は、GBUS の GCS5-用の空間として予約されています。GBUS については「11 GBUS 個別仕様」、「14 APPEDIX.C GBUS 共通仕様」を参照してください。

CS2 空間 (GCS0-) :0400000 - 07FFFFFF(4M-byte)

CS2 空間は、GBUS の GCS0-用の空間として予約されています。GBUS については「11 GBUS 個別仕様」、「14 APPEDIX.C GBUS 共通仕様」を参照してください。

CS3 空間 (SDRAM) :1000000 - 1FFFFFFF(16M-byte)

CS3 空間には、SDRAM が割付けられています。SW2-4(BSIZE)の指定により、16bit と 32bit 両方のバスサイズに設定可能です。モニタ ROM が実装されている場合は、モニタ ROM が SDRAM の割付けと初期化を行います。

CS4 空間 (GCS1-,GCS3-) :2000000 - 2FFFFFFF(16M-byte)

CS4 空間は、GBUS の GCS1-,GCS3-用の空間として予約されています。GBUS については「11 GBUS 個別仕様」、「14 APPEDIX.C GBUS 共通仕様」を参照してください。

CS5 空間 (I/O,GCS2-,GCS4-,GCS6-) 3800000 - 3BFFFFFF(4M-byte)

CS5 空間は、I/O 空間として使用します。本ボード内の I/O と、GBUS の GCS2-,GCS4-,GCS6-用の予約空間があります。I/O マップについては「7.4 IO マップ」を、GBUS については「11 GBUS 個別仕様」、「14 APPEDIX.C GBUS 共通仕様」を参照してください。

CS6 空間 (未使用) 3000000 - 37FFFFFF(8M-byte)

CS6 空間は、本ボード上では使用していません。

CS7 空間 (SRAM) 3C00000 - 3FFFFFFF(4M-byte)

CS7 空間は、SRAM が割付けられています。SW2-4(BSIZE)の指定により、16bit と 32bit 両方のバスサイズに設定可能です。モニタ ROM が実装されている場合は、モニタ ROM が指定のサイズに初期化を行います。尚、SRAM 後半の 32KB は、モニタがワークで使用していますので、ユーザプログラムでは使用できません。(「12.2.3 モニタ割り込み13.2.2,ROM モニタ・ワーク RAM」を参照してください。)

7.2. 推奨設定

ここでは、メモリおよび I/O 資源のアクセスに関する各レジスタの推奨設定値を示します。

7.2.1. MEMC レジスタ

システム・バスに関する設定は、下表のように設定してください。一部の設定は、SW2-4(BSIZE)や SW2-3(BCLK_LOW)の設定状態により異なりますので、注意してください。

レジスタ名	アドレス	設定値	備考
BCT0	0xFFFF480	0xB888	CS0-2:SRAM/IO,CS3:SDARM
BCT1	0xFFFF482	0x8888	CS4-7:SRAM/IO
DWC0(BCLK_LOW=OFF)	0xFFFF484	0x1111	CS0-3:1wait
DWC0(BCLK_LOW=ON)		0x0000	CS0-3:0wait
DWC1(BCLK_LOW=OFF)	0xFFFF486	0x1711	CS4-5,CS7:1wait,CS6:7wait
DWC1(BCLK_LOW=ON)		0x0700	CS4-5,CS7:0wait,CS6:7wait
VSWC(BCLK_LOW=OFF)	0xFFFF06E	0x77	現在は、BCLK_LOW の状態に関係なくデフォルト値:0x77)
VSWC(BCLK_LOW=ON)			
BCC	0xFFFF488	0x3C40	CS0,1,2,4,7:0clk,CS3:1clk,CS5,CS6:3clk
ASC	0xFFFF48A	0x5555	All 1 addr wait
BCP	0xFFFF48C	0x00	Normal bus cycle
CSC0	0xFFFF060	0xCCC3	(Chip Select Control Register0)
CSC1	0xFFFF062	0xCCC3	(Chip Select Control Register1)
BSC(BSIZE=OFF)	0xFFFF066	0xAAAA	All 32-bit
BSC(BSIZE=ON)		0x6A6A	Only CS3,CS7 16bit
BEC	0xFFFF068	0x0000	All little endian
SCR3(BSIZE=OFF)	0xFFFF4AC	0x20A4	SDRAM
SCR3(BSIZE=ON)		0x2094	(LTM=2,BCW=2,SSO=2,RAW=12,SAW=0) SDRAM (LTM=2,BCW=2,SSO=1,RAW=12,SAW=0)
RFS3(BCLK_LOW=OFF)	0xFFFF4AE	0x8017	50MHz:15.4uS
RFS3(BCLK_LOW=ON)		0x800F	33MHz:15.5uS
ISS	0xFFFF7FA	0x4F	ISS0=1,ISS1=1,ISS2=1,ISS3=1,ISS6=1
RSZ	0xFFFF7FC	0x40	IRAM=60KB
DCC	0xFFFF078	0x0C00	DC05:04=WriteBack(WriteAllocate Enable)
CSZ	0xFFFF7FE	0x10	DTYP=1(SDRAM)
BHC(CACHE=OFF)	0xFFFF06A	0x0000	All uncached
BHC(CACHE=ON)		0xC0C0	CS7,CS3 i/d cacheable

CACHE=OFF/ON は、SW2-6 を操作します。

注意：SDRAM 関連レジスタ(SCR3, SFR3)の設定手順等について、CPU のマニュアルを参照して行ってください。

7.3. メモリ資源

RTE-NB85E-CB では、オンボードのメモリ資源として、SDRAM、SRAM、UV-EPROM があります。ここでは、それらのメモリ・デバイスについて説明します。

7.3.1. SDRAM (CS3:1000000 - 1FFFFFFF)

SDRAM は、1M-Word×16Bit×4Bank の SDRAM (μPD4564163G5) デバイスを 2 つ実装し、16M バイトの容量を搭載しています。データ・バス幅は、SW2-4(BSIZE)の設定値に従い、16Bit と 32bit で使用可能です。但し、16bit のバス幅で使用する場合、メモリの容量は半分の 8M バイトになります。

7.3.2. SRAM (CS7:3C00000 -3FFFFFFF)

SRAM は、256K-Word×16bit:15nS の高速 SRAM を 2 つ実装し、1M バイトの容量を搭載しています。バスクロックが 33MHz を越える時は 1wait、それ以下では 0 ウェイトでアクセスできます。データ・バス幅は、SW2-4(BSIZE)の設定値に従い、16Bit と 32bit で使用可能です。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 バイトおきにイメージが現れます。

尚、SRAM 後半の 32KB は、モニタがワークで使用していますので、ユーザプログラムでは使用できません。(「12.2.3 モニタ割り込み13.2.2,ROM モニタ・ワーク RAM」を参照)

7.3.3. UV-EPROM (CS0 :0000000 - 03FFFFFF)

UV-EPROM 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 ROM が搭載できます。搭載する ROM の種類と使用状態は、SW3 で設定します (「5.5 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 は、InitialBusSize 32 ビットと 16 ビットの両方に対応しています。32 ビットバスに対しては、内部的に ROM に対し 2 回のアクセスを発生させ、16bit の ROM 1 個で賄っています。

ROM の Wait 数は、SW2-3(BCLK_LOW)の設定により、以下の通りハードウェアで強制的に挿入されます。32bit バスでアクセスした場合は、ROM に対するアクセスは 2 回になりますので、この 2 倍のウェイト数が必要です。

SW2-3(BCLK_LOW):OFF = 8-wait

SW2-3(BCLK_LOW):ON = 5-wait

7.3.4. 内蔵 ROM (0000000 - 00FFFFFF, 0100000 - 01FFFFFF)

内蔵 ROM 領域は、内蔵 ROM と同じアクセスタイムでフェッチできるエミュレーションメモリがマップされ、内蔵 ROM に配置したプログラムをそのままのアドレスで実行することができます。

エミュレーションメモリの容量は 1M バイト、通常はリードオンリーです。デバッガからのダウンロードでのみ書き込み可能です。(但し、割り込みベクタの書き換え等の目的でこの領域の書き換えが必要な場合は、「9.1 割り込みベクタ」を参照して実施してください。)

7.4. IO マップ

RTE-NB85E-CB では、オンボード I/O として、シリアル・コントローラ (TL16C550C)、タイマ (μPD71054)、LED、スイッチ等があります。また、I/O 空間の一部として、GBUS のチップ・セレクト空間を予約しています。

ここでは、オンボード I/O のマップと各 I/O デバイスについて説明します。

7.5. IO 一覧

IO の一覧を下表に示します。Wait 数は SW2-3 (BCLK_LOW) の設定状態で切り替わります。

アドレス	用途	Wait 数	
		BCLK_LOW OFF	BCLK_LOW ON
3800000	SW1	10	7
3801000	SW2	10	7
3802000	7SEGLED	10	7
3803000	TOVRDY_LED_CLRPLS	10	7
3804000 - 3804020	PIC	10	7
3807000 - 3807070	UART (TL16C550C)	10	7
3808000 - 3808030	TIC (μPD71054)	10	7
3809000	SW16	10	7

7.5.1. SW1 読出しポート (SW1 3800000H [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	
3800000H 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、「12.1.2 SW1 の設定」、「13.1.1 SW1 の設定」の各項を参照してください。

7.5.2. SW2 読出しポート (SW2 3801000H [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	
3801000H 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替えます。それぞれのスイッチの機能は、「5.4 スイッチ 2 (SW2)」を参照してください。

7.5.3. SW16 読出しポート(SW16 3809000H [Read Only])

SW16 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	
3809000H 入力	SW16-8	SW16-7	SW16-6	SW16-5	SW16-4	SW16-3	SW16-2	SW16-1	0=ON 1=OFF

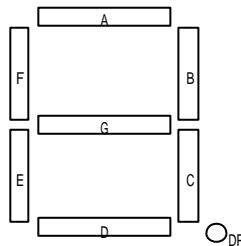
SW16-1 が SW16 の "1" のスイッチに、SW16-8 が SW16 の "8" のスイッチに対応しています。また、該当するビットのスイッチが ON で 0 が、OFF で 1 が読み出されます。SW16 は、CPU の動作モードを切替えます。それぞれのスイッチの機能は、「エラー! 参照元が見つかりません。 エラー! 参照元が見つかりません。」を参照してください。

7.5.4. 7セグメント LED 表示データ出力ポート(7SEG-LED 3802000 [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..D31	D6..D30	D5..D29	D4..D28	D3..D27	D2..D26	D1..D25	D0..D24	
3802000H 出力	LED1 -DP	LED1 -G	LED1 -F	LED1 -E	LED1 -D	LED1 -C	LED1 -B	LED1 -A	0=点灯 1=消灯
3802001H 出力	LED2 -DP	LED2 -G	LED2 -F	LED2 -E	LED2 -D	LED2 -C	LED2 -B	LED2 -A	
3802002H 出力	LED3 -DP	LED3 -G	LED3 -F	LED3 -E	LED3 -D	LED3 -C	LED3 -B	LED3 -A	
3802003H 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。



7.5.5. タイムオーバ・レディーLEDクリア・パルス(TOVRDY_LED_CLRPLS 3803000H [Write Only])

本ポートに書き込みを行うと、ボード上のタイムオーバ・レディー発生時に点灯する TOV_RDY LED が消灯します。書き込まれたデータは無視されます。TOV_RDY LED は一旦点灯すると、本ポートに書き込むか、ボードをリセットするまで点灯し続けます。

7.5.6. 割り込みコントローラ(PIC:3804000 - 3804020[Read/Write])

PIC は、Multi および PARTNER のモニタ・プログラムの動作に必要な割り込みをサポートします。接続可能な割り込みは以下の通りです。

- 1)RS232C デバイス (UART,TL16C550C) からの通信割り込み
- 2)タイマ(TIC, μ PD71054)の TOUT0 によるタイマ割り込み要求
- 3)タイムオーバー・レディの発生

論理アドレス	レジスタ	データバス								
		D7	D6	D5	D4	D3	D2	D1	D0	
3804000H	PIC INT-MASK	x	x	x	x	0	IM2	IM1	IM0	
3804010H	PIC INT-STATUS	x	x	x	x	0	IR2	IR1	IR0	
3804020H	PIC INTENA	x	x	x	x	0	0	INT0/ NMI-	INT EN	

INT-MASK レジスタはそれぞれ入力する割り込みをマスクします。INT_MASK 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。

【注意】 INT-MASK の bit3 は、常に 0 を書き込んでください。

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"を書込みます。

IM[0..2], IR[0..2]の各ビットに割付けられている割り込み要因は以下の通りです。

PIC INT-MASK[],STATUS[]	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	タイムオーバー	レベル(High)

INTENA レジスタは、割り込み全体のイネーブル/ディセーブルなどを制御します。

INTEN: 割り込みをハード的に禁止することができます。この時、割り込み端子の状態は Low レベルとなります。

INTEN	NMI2/INT0
0	マスクする (リセット値)
1	マスクしない

INT0/NMI2-:モニタで使用する割り込みを選択します。

INT0/NMI-	モニタ用の割り込み
0	NMI2 を使用 (リセット値)
1	INT0 を使用

INTEN は、複数の割り込み要因が発生している時に、1つの割り込み処理が終了した時点で、CPU への割り込み信号にエッジを生成するために使用します。CPU の割り込み入力にエッジ検出の場合、割り込みハンドラの最後のステップでは、INTEN ビットを一旦"1"に設定後、"0"にする処理を実行してください。これによって、保留中の割り込みが入ります。

注意: モニタ使用中は、PIC へのアクセスは行わないでください。

7.5.7. UART (TL16C550C:3807000 - 3807070)

UART コントローラとして TEXAS INSTRUMENTS 製の TL16C550C LSI を使用しています。TL16C550C は、UART 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、最小限の割り込みで通信のオーバーラン・エラーを押さえられます。

TL16C550C 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、TL16C550C のマニュアルを参照してください (TL16C550C 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	読出し	書込み
3807000H	RBR/DLL	THR/DLL
3807010H	IER/DLM	IER/DLM
3807020H	IIR	FCR
3807030H	LCR	LCR
3807040H	MCR	MCR
3807050H	LSR	LSR
3807060H	MSR	MSR
3807070H	SCR	SCR

TL16C550C レジスタ配置

TL16C550C の XIN 入力には 16MHz のクロックが接続されています。

UART はボードの JIS01 コネクタに接続しています。また、UART はリモートデバッグを用いる場合、ホストとの通信に使用されます。

TL16C550C は、システム・リセットによってリセットされます。

注意：モニタ使用中は、UART へのアクセスは行わないでください。

7.5.8. TIC (uPD71054 3808000H ~ 380803FH)

TIC は NEC 社製の uPD71054 が実装されています。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/カウンタにより、モニタのタイマ割り込みの生成を行っています。

TIC の各レジスタは、表の通りに割り当てられています。

アドレス	読出し	書込み
3808000H	COUNTER#0	COUNTER#0
3808010H	COUNTER#1	COUNTER#1
3808020H	COUNTER#2	COUNTER#2
3808030H	-----	Control Word

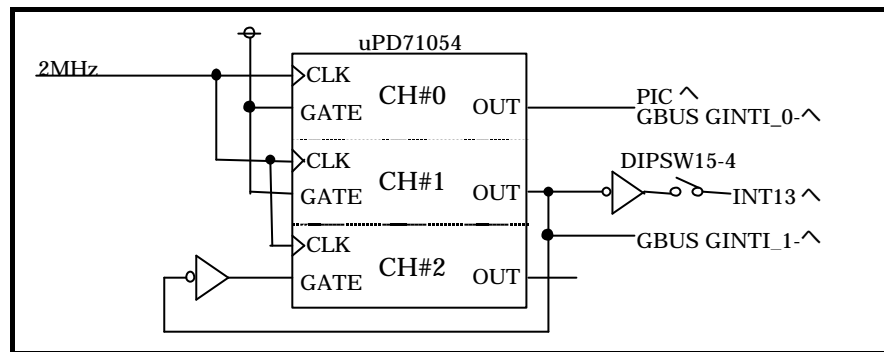
TIC のレジスタ配置

TIC の各チャンネルは下図のように接続されています。

チャンネル0 は、Multi の ROM モニタ・プログラムのインターバル・タイマとして使用されます。

チャンネル1, 2 は、ユーザのプログラムで自由に使用することができます。

チャンネル2 は、チャンネル1 とカスケード接続されています。



使用モード例

- CH#0 : モード2 (レートジェネレータ)
- CH#1 : モード2 (レートジェネレータ)
- CH#2 : モード0 (ダウンカウンタ)

注意 : モニタ使用中は、CH0 へのアクセスは行わないでください。

8. ソフトウェア

RTE-NB85E-CB ボードのハードウェアの初期化と周辺デバイスの使用方法について説明します。

8.1. 初期化

モニタを使用しないで、ROM からブートするプログラムを開発する場合は、最初のルーチンで NB85E の内部バスコントローラを初期化しなければなりません。初期化時に設定する値については、「7.2 推奨設定」を参照してください。

8.2. uPD71054 に対する連続アクセス

uPD71054 に対し連続してアクセスする場合は、1 回目のアクセスと 2 回目のアクセスの間に他の空間へのアクセスを 1 回以上入れてください。これは、uPD71054 に対するリカバリ・タイムを確保するためです。

リカバリ・タイムは、uPD71054 以外の資源（例えば、ROM）をダミーで一度リードすることで確保できます。

8.3. ライブラリ

C コンパイラでプログラムする時に必要となる I/O アクセスなどのライブラリです。ただし、これらの記述やパラメータ受け渡し方法などは、Multi 環境でのものです。他のコンパイラ等を使用する場合には、変更が必要となる場合があります。

```

/* I/O 入出力ライブラリ */
/* GHS V850 コンパイラ パラメータ受け渡し */
/* arg0 : r6, arg1 : r7, arg2 : r8, return : r10 */

inb(int addr)                /* バイト (8 ビット) 入力 */
{
    __asm(" ld.b 0[r6], r10");
}

inh(int addr)                /* ハーフワード (16 ビット) 入力 */
{
    __asm(" ld.h 0[r6], r10");
}

inw(int addr)                /* ワード (32 ビット) 入力 */
{
    __asm(" ld.w 0[r6], r10");
}

outb(int addr, int data)     /* バイト (8 ビット) 出力 */
{
    __asm(" st.b r7, 0[r6]");
}

outh(int addr, int data)     /* ハーフワード (16 ビット) 出力 */
{
    __asm(" st.h r7, 0[r6]");
}

outw(int addr, int data)     /* ワード (32 ビット) 出力 */
{
    __asm(" st.w r7, 0[r6]");
}

```


8.4. タイマの使用例

ボード上の外部タイマ (uPD71054) でカスケード接続されたタイマ1とタイマ2を使用した時間計測のサンプルを示します。タイマ1はインターバルカウンタ (モード2)、タイマ2は、ダウンカウンタ (モード0) として初期化して、時間計測するルーチンの前後でカウンタ値を求めておくことで実行時間が算出できます。ただし、タイマのカウント値はどちらもダウンカウンタとなることに注意してください。また、外部タイマの連続アクセスではコマンドリカバリ (例: SW1のダミーリード) が必要となります。

```

/* タイマによる実行時間計測サンプル */

#define TIMERCLK      2000000          /* 2MHz */
#define INTERVAL     (TIMERCLK * 10 / 1000) /* 10ms (1/100) */
#define IOWAIT()     (*(char *) 0x3800000) /* I/O コマンドリカバリ用 */

InitTimer() /* タイマ初期化 */
{
    outb(0x3808030, 0x74);          IOWAIT(); /* タイマ1 モード2 */
    outb(0x3808010, INTERVAL);     IOWAIT(); /* タイマ1 下位カウンタ */
    outb(0x3808010, INTERVAL / 256); IOWAIT(); /* タイマ1 上位カウンタ */
    outb(0x3808030, 0xB0);          IOWAIT(); /* タイマ2 モード0 */
    outb(0x3808020, 0xFF);          IOWAIT(); /* タイマ2 下位カウンタ */
    outb(0x3808020, 0xFF);          IOWAIT(); /* タイマ2 上位カウンタ */
    return 0;
}

LatchTimer() /* カウントラッチ */
{
    int count1, count2, counts;

    outb(0x3808030, 0xDC);          IOWAIT(); /* タイマ1/2 マルチプルラッチ */
    count1 = inb(0x3808010);         IOWAIT();
    count1 += inb(0x3808010) * 256;   IOWAIT(); /* タイマ1 カウンタ */
    count2 = inb(0x3808020);         IOWAIT();
    count2 += inb(0x3808020) * 256;   IOWAIT(); /* タイマ2 カウンタ */
    counts = INTERVAL * (0xFFFF - count2)
        + (INTERVAL - count1);
    return counts;
}

double total_time;

main()
{
    int start_count, stop_count;

    InitTimer();
    start_count = LatchTimer(); /* スタートカウンタ値 */
    func();
    stop_count = LatchTimer(); /* ストップカウンタ値 */
    total_time = (double)(stop_count - start_count)
        / (double)TIMERCLK; /* 秒数 */
    return 0;
}

#include <time.h>

func() /* 時間計測ルーチン */
{
    ....
}

```

9. 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

本章では、RTE-NB85E-CB 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。

9.1. 割り込みベクタ

NB85E の割り込みベクタ領域である 0000000H ~ 00007FFH 番地は、Singlemode1 及び RomlessMode では ROM により固定されていて書換えることができません。そこでモニタでは、SRAM 上に以下に示す 2 つのベクタ領域を用意しています。

代替ベクタ領域：

ユーザ・プログラムによって書換えが可能なベクタ領域で、割り込みベクタ領域から相対ジャンプが可能な場合に使用します。この場合には、ベクタ領域に相対ジャンプによる分岐命令が置かれます。

中継ベクタ領域：

モニタが使用するベクタ領域で、割り込みベクタ領域から相対ジャンプが不可能な場合に使用します。この場合には、割り込みベクタ領域にレジスタを保存する命令と絶対ジャンプによる分岐命令が、このベクタ領域にはレジスタを復旧する命令と代替ベクタ領域に相対ジャンプする分岐命令がそれぞれ置かれます。

以下に中継ベクタと代替ベクタのそれぞれの領域を示します。

Mode	代替ベクタ領域	中継ベクタ領域
SINGLEMODE0	3CF8000H ~ 3CF87FFH	3CF8800H ~ 3CF8FFFH
SINGLE MODE1	3CF8000H ~ 3CF87FFH	3CF8800H ~ 3CF8FFFH
ROMLESS	3CF8000H ~ 3CF87FFH	3CF8800H ~ 3CF8FFFH

中継ベクタ領域はユーザ・プログラムで使用可能

例えば、例外コードが 0080H の割り込みが発生すると、CPU の割り込み機能により 0000080H 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 0080H 番地への分岐命令があります。ユーザ・プログラムでは、分岐先の代替ベクタ領域を書換えることにより、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できます。

したがって、例外コード 0080H 割り込みの場合、SRAM 上の代替ベクタ領域 3CF8080H 番地に目的の割り込み処理に分岐する命令を書込みます。

通常の NB85E のプログラムと異なるのは、ベクタ領域は ROM 化の時点で固定されており、プログラムで設定する（書換える）必要はありません。しかし、RTE-NB85E-CB 上でモニタを使用したプログラムの場合、プログラムでベクタを書換えてから、割り込みを許可する必要があります。

以下に代替ベクタを書換えるためのプログラム例を示します（割り込み処理ルーチンから代替ベクタ領域への相対アドレスが 22Bit 以内の場合）。

```
#define IROM_WRENA    1
#define IROM_WRDIS    0

void di() /* Disable interrupt */
{
    __asm( "di" );
}
void ei() /* Enable interrupt */
{
    __asm( "ei" );
}
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 = 0x07800000 /* 'jr' opcode */ | (offset & 0x003ffffe);

    *((UINT16 *) (addr + 0)) = (inst >> 16) & 0xffff ;
    *((UINT16 *) (addr + 2)) = (inst << 16) & 0xffff ;
}
.....
void __interrupt IntEntry() /* 割り込み処理ルーチン */
{
    .....
}
.....
main()
{
    .....
    SetAJump((int)(0x080 + 0x3CF8000) ,(int)IntEntry) ;
    /*          目的の割り込みの例外コード */
}
```

9.2. 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以下に示します。

- 1) 代替ベクタの設定前に割り込みが発生した場合や、代替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のプログラム位置でブレークします。これは、代替ベクタの初期値がモニタのブレーク処理ルーチンへの分岐命令になっているためです。
- 2) 代替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 22Bit 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) 代替ベクタ領域及び、SingleMode0 のベクタ領域は、プログラムのダウンロード時に書換えることができます（「9.3 ダウンロード時の代替ベクタ領域書換え方法」を参照）。但し、プログラムのダウンロード時の書換えは、使用する割り込み以外を書換えないようにしてください。
- 4) 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、一度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。ペリフェラルを使用するプログラムの場合、再度プログラムを始めから動作させる場合には、以下の手順に従ってください。
 - (1) モニタをディスコネクトします。
 - (2) RTE-NB85E-CB のリセット・スイッチを押しボードをリセットします。
 - (3) モニタにコネクトします。
 - (4) プログラムをロードして実行します。
- 5) プログラムの先頭で 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ください。
- 6) ブレーク中に割り込みの DI(割り込み禁止) / EI(割り込み許可)状態をデバッガの I/O(レジスタ)操作機能で行う場合は、割り込みマスクレジスタ(IMRn)の該当ビットで行ってください。ブレーク中に割り込み制御レジスタ(PICn or PnnICn)をデバッガの I/O(レジスタ)操作機能で操作した場合は、割り込み動作が正しく機能しなくなることがありますので、割り込み制御レジスタの操作は行わないでください。

9.3. ダウンロード時の代替ベクタ領域書換え方法

プログラムのダウンロード時にベクタを書換える方法としては、色々な方法が考えられます。ここでは、GHS 社の Multi 環境を基に例を示します。また、この方法は実際に ROM 化するプログラムに類似した方法とも言えます。尚、前述のプログラム例と合わせて参照してください。

- 1) 割り込みベクタ書換え用のプログラムを定義 (ASM 言語)

割り込みベクタに置く分岐命令だけのプログラムを以下のように定義します。記述方法の詳細については、言語処理系のマニュアルを参照してください。

```

.section      "intvct", .text /* Defined section name */
.align      4
.globl      _Int80
_Int80:
    jr _IntEntry /* jump to handler */
    nop
    nop

```

但し、1つの割り込みに対してベクタの境界を超えて定義することはできませんので注意してください。

2) セクション・マップの定義

リンク時に使用するセクション・マップを以下のように定義します。記述方法の詳細については、言語処理系のマニュアルを参照してください。

```
{
    .intvct      0x3FE8080  :
    .text
    .data      0x3E00000  :
               align(0x10)  :
    .
    .
}
```

ベクタに置くプログラムのセクションを最初に定義します。

複数の割り込みを使用する場合には、連続したベクタであれば1つのセクションを定義することで対処できます（割り込みベクタの境界を合わせる必要があります）。不連続の場合は、割り込み毎にセクションを定義し、セクション・マップに全てのセクションを指定することが必要です。

この方法でダウンロード時に代替ベクタ領域の所定の場所が書換えられます。また、この方法を用いることで内部 ROM 領域で動作するプログラムでも割り込みベクタの書換えが容易に行えます。更には、割り込みベクタを書換えるためのコードを記述する手間を省くことができます。

9.4. ブレーク・ポイント使用に関する制限事項 / 注意事項

割り込み処理ルーチン内へのブレークポイントの設定や、その後の実行（シングル・ステップ）に関しては、以下の制限事項 / 注意事項がありますのでご注意ください。

- 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。この結果、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、1命令をシングル・ステップする間に割り込み処理に分岐し、割り込み処理を行うことがあります。
したがって、シングル・ステップでも、ブレーク・ポイントに関する注意事項に気をつけなければなりません。
- 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できません（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できません）。同様に、reti 命令のシングルステップもできません。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できません。

10.CPU 端子接続

本章では、RTE-NB85E-CB 内での CPU の各端子の使用状態を説明します。

10.1. 一 覧

下表は主な CPU 端子の使用状態の一覧です。詳細は後続の章で説明します。

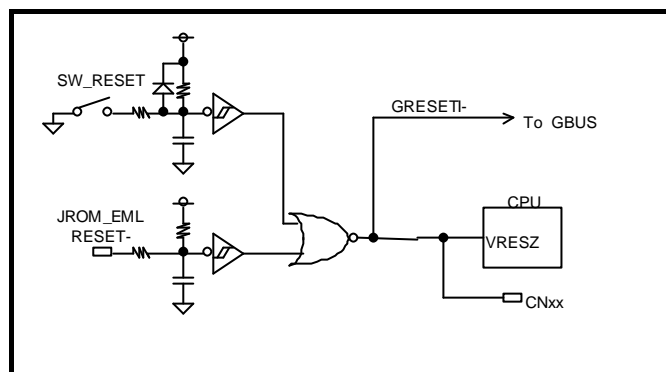
端子名	使用状態	参照章
D0 - D31	システムのデータバスとして使用	
A0 - A25	システムのアドレスバスとして使用	
BCYST- RD-	システムバスの制御信号として使用	
WR0-/CAS0-/DQM0 - WR3-/CAS3-/DQM03	システムバス及び、SDRAMで使用	
OE- WE- SDRAS- SDCAS- CKE SDCLK CS3-/RAS3-	SDRAMで使用	
CS0-/RAS0- CS1-/RAS1- CS2-/RAS2- CS4-/RAS4- CS5-/RAS5- CS7-/RAS7-	システムバスで CSx-として使用	
CS6-/RAS6-	未使用	
HLDREQ-	未使用	
HLDACK-	未使用	
VRESZ	RESETを入力	10.2
MWAIT-	システムバスで使用。但し、CS6 の空間では、Hiz 1K でプルダウン有り	10.3
NMI0 - NMI2, INT0	システムで使用(SW11 で切断可能)	10.4
INT10 - INT13	システムで使用(SW15 で切断可能)	10.5
RXD/IN45	SIO2-RXD として使用(SW14 で切断可能)	10.6
TXD/INT46	SIO2-TXD として使用	10.7
PORT0/MPXSCZ PORT1/DSTBZ PORT2/RDCYZ PORT3/BUSSR	PORT として使用し、SIO2 の制御に使用(SW12 で切断可能)	10.8
DMARQ0/INT32 DMARQ1/INT33 DMARQ2/INT34 DMARQ3/INT35	DMARQ として使用し、GBUS-DMARQ0-3 を接続 (SW13 で切断可能)	10.9
DMAAK0/INT36 DMAAK1/INT37 DMAAK2/INT38 DMAAK3/INT39	DMAAK として使用し、GBUS-DMAAK0-3 に接続 (SW13 で切断可能)	10.10
TC0/INT40 TC1/INT41 TC2/INT42 TC3/INT43	GBUS のリザーブピンに接続(SW14 で切断可能)	10.11
TB136,TB137 STOPZ	10K でプルアップ	10.12
INTxx VB0 - VB31 VPD0 -VPD14 VPRETR,VPDACT TBI32 - TBI34,TBI38,TBI39 TEST,BUNRI VAREQ0,VBWAIT,VBHOLD,VBLAST VBEXDC,VBEXCLK, その他	10K でプルダウン	10.12
	未処理	

10.2. RESET-

CPU へのリセットは以下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、ボード全体のシステム・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。
- **JROMEM コネクタからのリセット要求**：JROMEM コネクタの RESET-端子からの入力によるリセットです（「5.10 ROM エミュレータ用テストピン (JROM-EML)」参照）
- **SW_RESET によるリセット**：リアパネル部分に用意されているリセット・スイッチ(SW_RESET)が押されるとリセットが発生します（「5.1 リセット・スイッチ (SW_RESET)」参照）

RESET の生成ロジックの概要を下图に示します。



10.3. MWAIT-

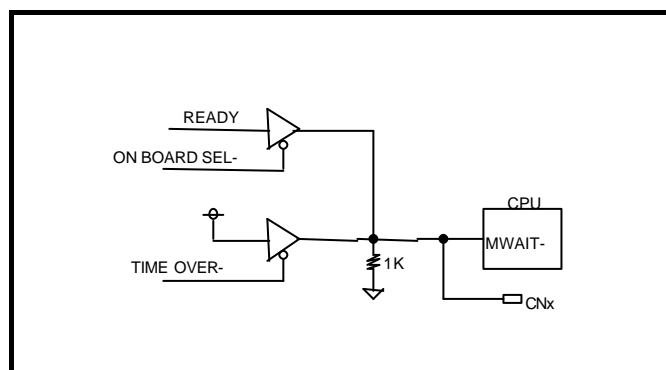
MWAIT 端子は、ボード内の資源にアクセスされている時とタイムオーバ・レディーが発生した場合、レディ信号がドライブされます。

タイムオーバ・レディーは、何等かの障害でバス・サイクルがある一定期間内で終了しない場合、ボード上でそれを検出し、強制的にバス・サイクルを終了させる機能です。

タイムオーバ・レディーが発生した場合、ボード上の TOVER_LED が点灯し、PIC に対し割り込みが入ります。TOVER_LED はタイムオーバ・レディーLED クリア・パルスをソフト的に発生させるか、ボードがリセットされるまで点灯し続けます（「7.5.5タイムオーバ・レディーLED クリア・パルス (TOVRDY_LED_CLRPLS 3803000H [Write Only])」参照）

また、GBUS に割り当てられた空間へのアクセスの場合、GBUS の GMOTHER_DETECT-信号が High (GBUS にボードが接続していない) の場合に限り、タイムオーバ・レディーが発生します。

MWAIT のドライブ部分の構成を下图に示します。



10.4. NMIO,1,2, INTO

NMIO,1,2, INTO は以下の信号が接続されています。

NMIO : ROM-EML(ROM エミュレータ)からの割り込み要求

NMI1 : GBUS-INT0-の割り込み要求

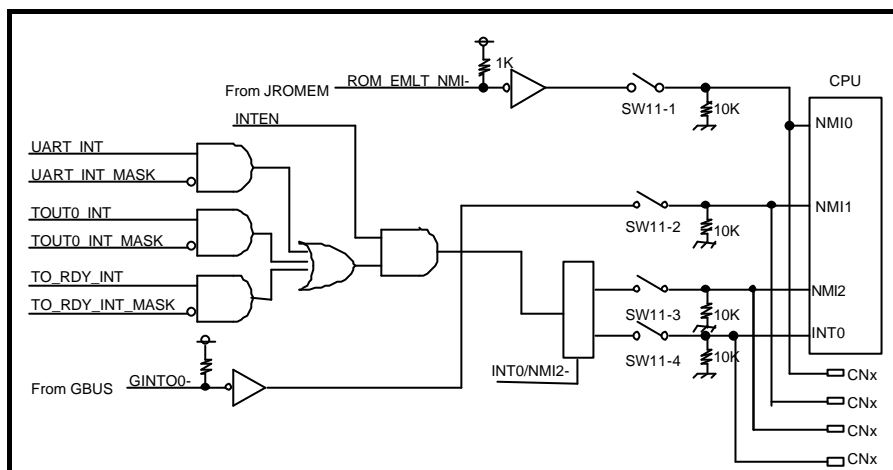
NMI2 : PIC からの割り込み要求

INT0 : PIC からの割り込み要求

NMI2/INT0 は、モニタ用の割り込みで、PIC を介して以下の複数の割り込み要因がハード的に複合されたものが接続されています。割り込みの選択方法については「7.5.6 割り込みコントローラ (PIC:3804000 - 3804020[Read/Write])」を参照してください。

- **UART_INT** : TL16C550C の UART の割り込みです (「7.5.7 UART (TL16C550C:3807000 - 3807070)」参照)
- **TOUT0_INT** : TIC (μ PD71054) の CH#0 の TOUT による割り込みです (「7.5.8 TIC (μ PD71054 3808000H ~ 380803FH)」参照)
- **TO_RDY_INT** : タイムオーバ・レディーの発生による割り込みです (「10.3 MWAIT-」参照)

NMI2 の生成ロジックの概念図を下図に示します。下図の xxx_MASK 信号は、PIC のレジスタへの設定を示します (「7.5.6 割り込みコントローラ (PIC:3804000 - 3804020[Read/Write])」参照)



NMI2 の処理手順を以下に示します。

割り込みイネーブル・レジスタの INTEN に"0"を設定して、NMI2-をハード的に禁止する (割り込みが複数発生している時のために、禁止することで一旦 NMI- 端子がネガティブレベルになる)

NMI2-の要求元を検索する。割り込み要求ステータス・ポートにより調べられる (「7.5.6 割り込みコントローラ (PIC:3804000 - 3804020[Read/Write])」参照)

要求元のための割り込み処理を行ない、要求をクリアする。

割り込みイネーブル・レジスタの INTEN に"1"を設定して、禁止を解除する。

割り込み処理から復帰する。



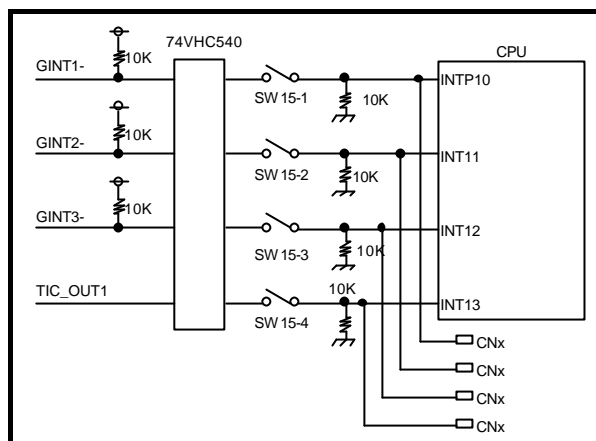
モニタ ROMを使用する場合は、モニタが使用する割り込みは SW2-5 により指定します。NMI を使用する場合は、SW11-3 が ON でなければなりません。

また、ROMエミュレータを使用する場合は、SW11-1 が ON になっていなければなりません。

10.5. INT10,11,12,13

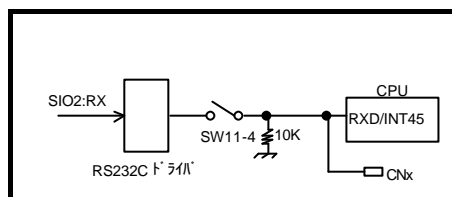
INT10,11,12,13 は、GBUS-INT0,1,2、及び、TIC_OUT1(タイマー-CH1 の出力)がスイッチを経由して接続されています。

それぞれの接続の様子を下図に示します。



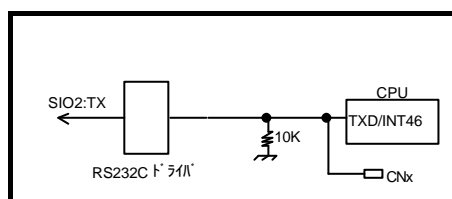
10.6. RXD/INT45

RXD/INT45 端子は、SIO2 の Rx を RS232C 受信ドライバで TTL に変換した信号がスイッチを経由して接続されています。この様子を下図に示します。



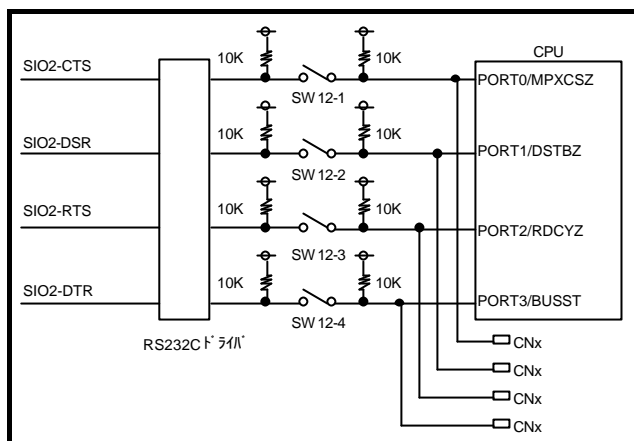
10.7. TXD/INT46

TXD/INT46 端子は、RS232C 送信ドライバを経て、SIO2 の TX に使用されています。この様子を下図に示します。



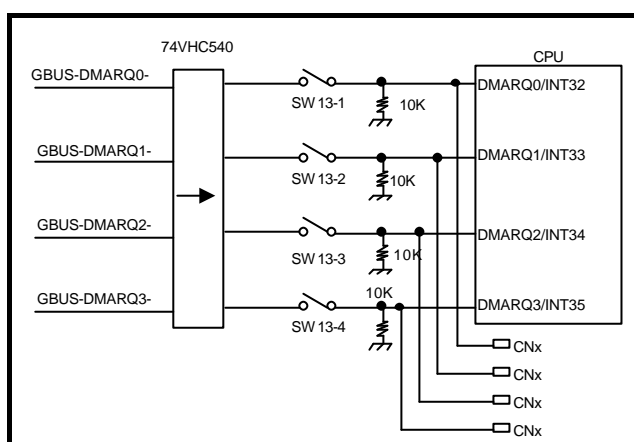
10.8. PORT0/MPXSCZ,PORT1/DSTBZ,PORT2/RDCYZ,PORT3/BUSST

これらの端子は、RS232C 送受信ドライバを経て、SIO2 の CTS,DTR,RTS,DTR に使用されています。
この様子を下图に示します。



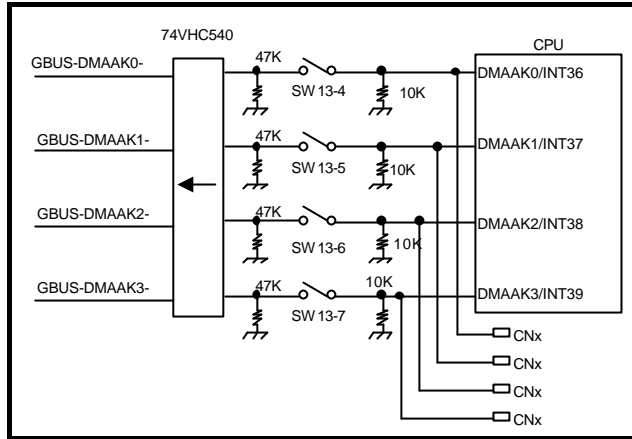
10.9. DMARQ0/INT32,DMARQ1/INT33,DMARQ2/INT34,DMARQ3/INT35

これらの信号ピンには、GBUS からの DMARQ 要求の論理を反転し、スイッチを経由して接続されています。この様子を下图に示します。



10.10.DMAAK0/INT36,DMAAK1/INT37,DMAAK2/INT38,DMAAK3/INT39

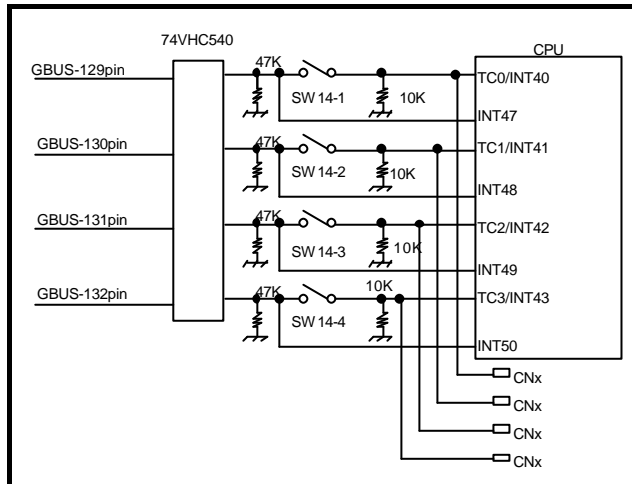
これらの信号ピンは、CPU の出力した信号の論理を反転し、スイッチを経由して GBUS の DMAAK に接続されています。この様子を下図に示します。



10.11.TC0/INT40,TC1/INT41,TC2/INT42,TC3/INT43

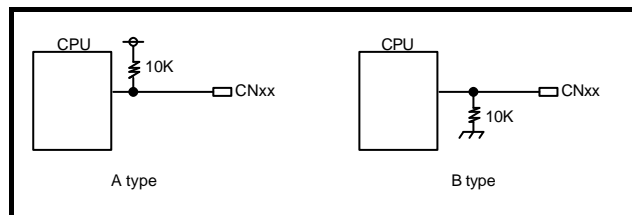
これらの信号ピンは、CPU の INT47 ~ INT50 と、信号の論理を反転し、スイッチを経由して GBUS のリザーブピンに接続されています。この様子を下図に示します。

INT47 ~ INT50 の接続は、DMA の終了割込みを入れる為に接続されています。



10.12.その他の信号

ボード内で使用していない信号は、下図のように JCPU コネクタに接続しています。



11.GBUS 個別仕様

本章では、RTE-NB85E-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。

11.1. 概要

RTE-NB85E-CB での GBUS の信号線使用状況の概要を下表に示します。

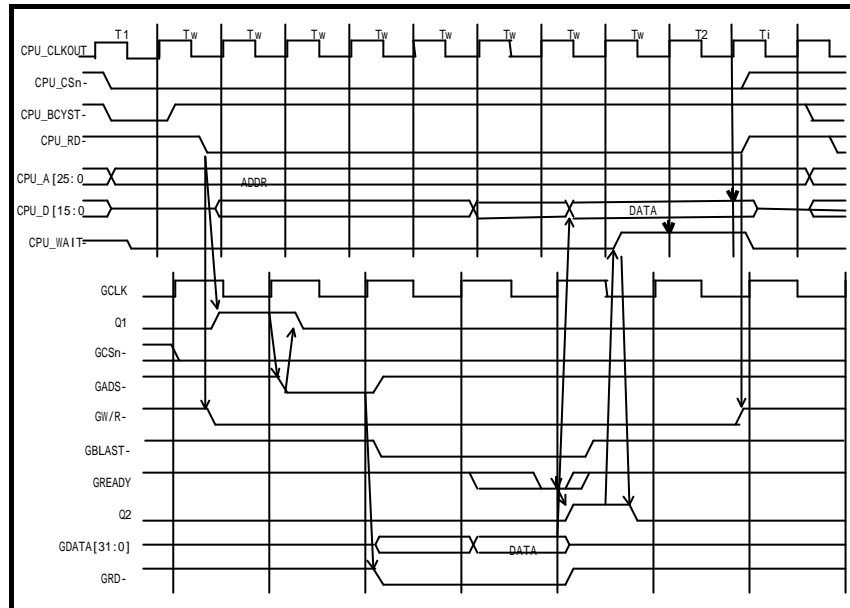
GBUS 信号名	機能	参照
GADDR[31:2]	アドレス線として使用。GADDR[26:31]は未接続。GADDR[25:24]は、Don't care。	
GDATA[31:0]	データ線として使用。リード・サイクル時は、VBCLK の立ち上がりでラッチされたものが CPU に供給される。	
GCS-[6:0]	チップ・セレクト線として使用。	
GCLK	CPU の VBCLKT とは、非同期の 33MHz 固定のクロックを接続。	
GRESETI-	本ボード上で発生したリセット要求を出力。	
GRESETO-	未接続	
GADS-, GREADY-, GBLAST-, GW/R-	バス制御信号として使用。	
GWAITI-	未接続	
GBTERM-	未接続	
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。	10.9
GDMAAK-[3:0]	DMA アクノリッジ信号として使用。	10.10
GINTO-[3:0]	割り込み要求信号として使用。	10.5
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-	タイムオーバ・レディー生成回路で使用。	
GUSE_DIRECT_ACC-	未接続	
GCLK_LOW-	未接続	
GLOCK-[1:0]	未接続	

11.2. バス・サイクル

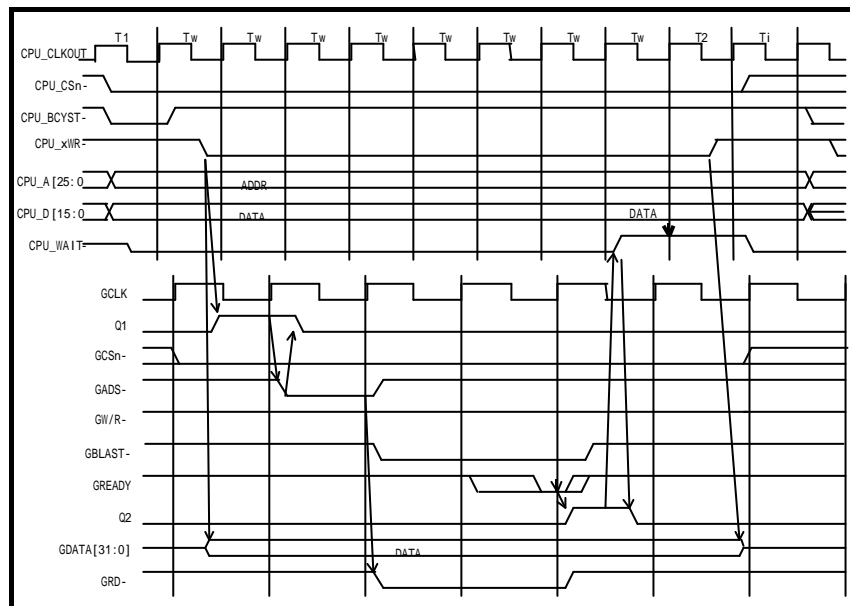
GBUS の GCLK へは、CPU の CLK とは非同期の 33Mhz クロックを接続しています。また、GAHI_EN-を未接続にしているため、GADDR[26:31]は未接続になっています。また、GADDR[24:25]は常に[0,0]です。

GBUS からのリード・サイクルは、GBUS 上で 0Wait 動作が可能です。

以下に示す波形で、CPU_xxx 信号は CPU の信号です。また、Gxxx 信号は GBUS の信号です。
リード・サイクルの様子を下図に示します。



ライト・サイクルの様子を下図に示します。



11.3. チップセレクト

本ボードでは、GBUSの各チップセレクトには、下記の空間が割り当てられています。下記の全ての空間では、CPU内蔵のバス・コンフィグレーション・レジスタの設定は、「7.2.1 MEMC レジスタ」を参照してください。

GBUS 信号名	CPU アドレス空間	物理アドレス範囲	RTE-MB-A の資源
GCS0-	CS2 空間の全て	0400000 - 07FFFFFF	共有 SRAM(2M)
GCS1-	CS4 空間の ADDR[23]=0 の空間と SW2-1(FBOOT)が ON の時は、CS0 空間の全 て	2000000 - 27FFFFFF 0000000 - 03FFFFFF	フラッシュ ROM(8M)
GCS2-	CS5 空間の ADDR[21..19]=[010]の空間	3900000 - 397FFFF	IO レジスタ
GCS3-	CS4 空間の ADDR[23]=1 の空間	2800000 - 2FFFFFF	EXT-bus:メモリ空間
GCS4-	CS5 空間の ADDR[21]=1 の空間	3A00000 - 3BFFFFFF	EXT-Bus:IO 空間
GCS5-	CS1 空間の全て	0800000 - 0FFFFFF	PCI バス空間
GCS6-	CS5 空間の ADDR[21..19]=[011]の空間	3980000 - 398FFFF	PCI-Cont レジスタ

12. 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。

12.1. ボードの設置

12.1.1. 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。

12.1.2. 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。

12.1.3. その他の SW の設定

SW2,SW3 の設定値や CPU の動作モードに応じて、モニタは初期化や動作をかえます。

(5.4 スイッチ 2 (SW2) 5.5 スイッチ 3 (SW3) エラー! 参照元が見つかりません。 エラー! 参照元が見つかりません。を参照)

12.1.4. ボードの接続

「6.ホスト PC との接続」を参照して、シリアルで PC と接続してください。

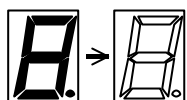
12.2. Multi モニタ

12.2.1. 起動時の 7Seg-LED

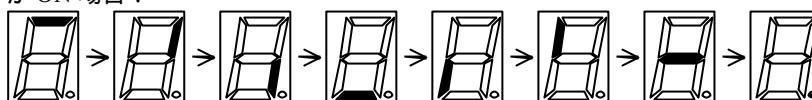
Multi 用の ROM モニタは、ボードの電源を入れると 7Seg-LED が次のように動きます（黒い部分が点灯部分）。

1)7Seg-LED のチェック動作（下図参照）

SW1-8 が OFF の場合：



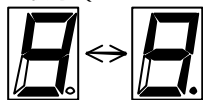
SW1-8 が ON 場合：



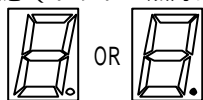
2)SRAM の簡易メモリチェックによる数字のカウンタ

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）



12.2.2. ROM モニタ・ワーク RAM

ROM モニタでは、SRAM の上位の 32KB（3FE8000H - 3FEFFFFH）をワーク用の RAM として使用しています。したがって、この空間とこの空間のイメージ領域は、ユーザ・プログラムで使用できません。

12.2.3. モニタ割り込み

モニタ通信、タイマ及び、強制ブレークで使用する割り込みは、SW2-5 で選択された割り込みを使用します。

12.2.4. _INIT_SP の設定

モニタで_INIT_SP（スタック・ポインタの初期値）は、3FE7FF0H（モニタ・ワーク RAM の直前）に設定されています（Multi の環境で_INIT_SP で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。

12.2.5. タイマ割り込み

タイマ割り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（タイマ割り込みの設定については『12.1.2 SW1 の設定』を参照）。

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

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。

BRKTRAP 命令（0xnn40）

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。

12.3. 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
SFR	内部 I/O 表示 / 設定

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

12.3.1. 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 SFR

SFR コマンドのヘルプを表示します。

12.3.2. INIT

<書式> INIT

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

12.3.3. VER

<書式> VER

VER は、RTE 環境のバージョンを表示します。

12.3.4. SFR コマンド

<書式> SFR [レジスタ名 [=データ]]

レジスタ名を指定してデータを省略した場合は、そのレジスタからリードしたデータを表示します。レジスタ名と“=”の後にデータを指定した場合には、そのレジスタにデータをライトします。データのサイズは、指定したレジスタの有効サイズで自動的に決定されます。内部 I/O レジスタの詳細については、NB85E-CPU のマニュアルを参照してください。

<例 1> SFR

レジスタ一覧を表示します。

<例 2> SFR IMR

レジスタ IMR の内容を表示します。

<例 3> SFR IMR=55AA

レジスタ IMR にデータ 55AAH をライトします。

13.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。

13.1. ボードの設置

13.1.1. 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 モニタでは使用していません。

13.1.2. その他の SW の設定

SW2,SW3 の設定値や CPU の動作モードに応じて、モニタは初期化や動作を変えます。

(5.4スイッチ 2 (SW2)、5.5スイッチ 3 (SW3) エラー! 参照元が見つかりません。 エラー! 参照元が見つかりません。を参照)

13.1.3. ボードの接続

「6.ホスト PC との接続」を参照して、シリアルで PC と接続してください。

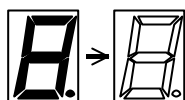
13.2. PARTNER モニタ

13.2.1. 起動時の 7Seg-LED

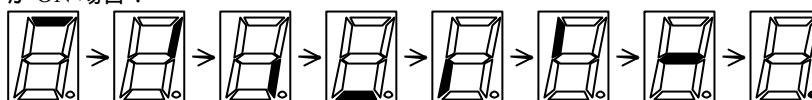
Partner 用の ROM モニタが実装されている場合、ボードの電源を入れると 7Seg-LED が次のように動きます（黒い部分が点灯部分）。

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



13.2.2. ROM モニタ・ワーク RAM

ROM モニタでは、SRAM の上位の 32KB (3FE8000 – 3FEFFFF) をワーク用の RAM として使用しています。したがって、この空間とこの空間のイメージ領域は、ユーザ・プログラムで使用できません。

13.2.3. モニタ割り込み

モニタ通信及び、強制ブレーク（ESC ボタン）で使用する割り込みは、SW2-5 で選択された割り込みを使用します。

13.2.4. SP の設定

モニタのスタック・ポインタの初期値は、3FE-7FF0H（モニタ・ワーク RAM の直前）に設定されています。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。

13.2.5. 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。

13.2.6. 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。

BRKTRAP 命令 (0xn40)

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。

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"> 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に同期して動作する。 マザー・ボード上では、+5V と GND に対してそれぞれ 330 Ω 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。 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。 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。
GRESETI-	入力	<ul style="list-style-type: none"> 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）。
GRESETO-	出力	<ul style="list-style-type: none"> マザー・ボードのリセットが発生した場合、Low になる信号。 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）。

信号名	入出力	機能
GADDR[31:2]	入力/出力	<ul style="list-style-type: none"> 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。 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。 下位アドレスの A1,A0 は、バイト・イネーブル信号を用いる。 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。 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。
GBEN-[3:0]	入力/出力	<ul style="list-style-type: none"> 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。 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。
GDATA[31:0]	双方向	<ul style="list-style-type: none"> GBUSのバス・データ信号。 マザー・ボード上で 10K でプルアップされる。 この信号の方向は、GW/R-により決定する。
GADS-	入力/出力	<ul style="list-style-type: none"> 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。 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。
GREADY-	出力/入力	<ul style="list-style-type: none"> 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。 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オーバー・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。
GWAITI-	入力	<ul style="list-style-type: none"> ウェイト要求信号。GCLK の立ち上がりでサンプルされる。 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 パ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)。 この信号は、CPU ボードがバス・マスタのサイクルのみ有効。
GBLAST-	入力/出力	<ul style="list-style-type: none"> 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。 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。 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。
GBTERM-	出力/入力	<ul style="list-style-type: none"> 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。 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-のアサートと同時でなければならない。 この信号は、アクセスされている側が、パースト・サイクルに対応していなかったり、対応しているパースト回数を越えるパースト・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。
GW/R-	入力/出力	<ul style="list-style-type: none"> 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。 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。
GCS-[7:0]	入力	<ul style="list-style-type: none"> 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。 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。 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 (「14.5 GCS-[7:0]の割付け」参照)。

信号名	入出力	機能
GRD-	入力	<ul style="list-style-type: none"> ・ 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。 ・ この信号はマザー・ボードでは使用しない。 ・ 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。
GWR-	入力	<ul style="list-style-type: none"> ・ 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。 ・ この信号はマザー・ボードでは使用しない。 ・ 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。
GHOLD-	出力	<ul style="list-style-type: none"> ・ バス・ホールド要求信号。 ・ 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。 ・ 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。
GHLDA-	入力	<ul style="list-style-type: none"> ・ バス・ホールド応答信号。 ・ 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。 ・ 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。
GBREQ-	入力	<ul style="list-style-type: none"> ・ バス権返還要求信号。 ・ 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。 ・ 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。 ・ 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。
GDMARQ-[3:0]	出力	<ul style="list-style-type: none"> ・ 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。 ・ 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。 ・ 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。 ・ 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。
GDMAAK-[3:0]	入力	<ul style="list-style-type: none"> ・ DMA 応答信号。 ・ マザー・ボードからの DMA 要求にตอบสนองする場合に Low にアサートする。 ・ 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。 ・ 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。
GINTO-[3:0]	出力	<ul style="list-style-type: none"> ・ 割り込み要求信号。 ・ GINTO0-は、レベル・センシティブとして使用可能。 ・ 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になっている。 ・ Low レベル時、もしくは立ち下がりエッジで割り込み発生を示す。
GINTI-[1:0]	入力	<ul style="list-style-type: none"> ・ 割り込み要求信号 ・ CPU ボード上の割り込みを、他のマザー・ボード上の割り込みと合成して GINTO-[3:0]に戻すために設けられた割り込み信号。 ・ 通常は CPU ボード上の TIC (μ PD71054) の OUT0 と OUT1 が接続される。マザー・ボードは、この割り込み信号に対して、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。

信号名	入出力	機能
GETC[7:0]		<ul style="list-style-type: none"> ・ CPU ボード依存信号。 ・ 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。
GAHI_EN-	入力	<ul style="list-style-type: none"> ・ アドレス上位有効信号。 ・ 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。
GMOTHER_DETECT-	出力	<ul style="list-style-type: none"> ・ マザー・ボード検出信号。 ・ 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。
GUSE_DIRECT_ACC-	入力	<ul style="list-style-type: none"> ・ 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。
GCLK_LOW-	入力	<ul style="list-style-type: none"> ・ 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。 ・ 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。
GBLOCK-[1:0]	入力	<ul style="list-style-type: none"> ・ 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。 ・ 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。 ・ 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。
+5V	出力	<ul style="list-style-type: none"> ・ 電源。+5V±5%をマザー・ボードから CPU ボードへ供給する。
+12V	出力	<ul style="list-style-type: none"> ・ 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。

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、GAHL_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る。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。
GAHL_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]の割付け

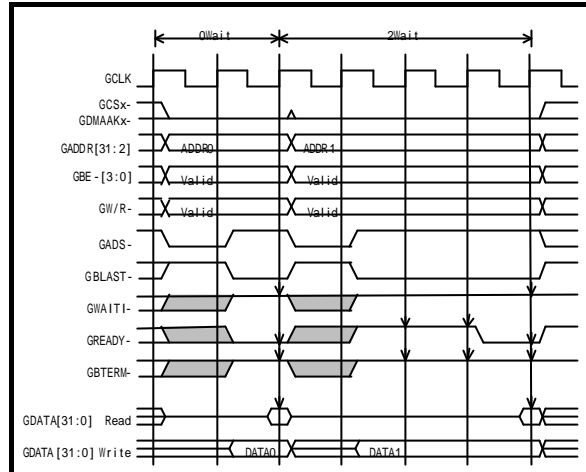
チップセレクト信号 (GCS-[7:0]) の割付けを下表に示します。全ての空間がバースト・サイクルによるアクセスが可能です。下表の推奨空間に I/O と記載されている空間は、CPU に I/O 空間がある場合は、I/O 空間に割付けることを推奨していることを示します。また最少範囲とは、CPU ボードは該当チップセレクトの空間に、最低でも最少範囲が示す領域を割り当てなければならないことを示します。最大範囲に記載がある場合は、CPU ボードのアドレス範囲に余裕がある場合、最大範囲が示す領域まで割り当てることが可能なことを示します。

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。

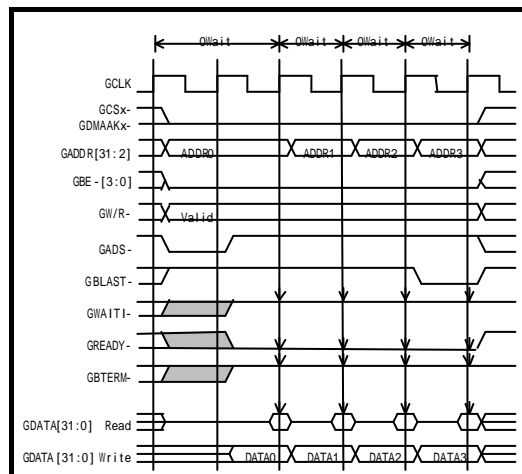


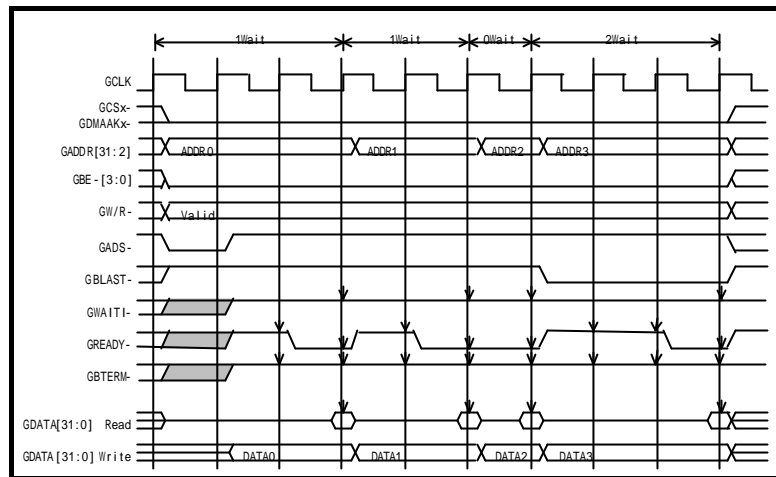
14.6.2. バースト・サイクル

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 (「14.6.4 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。





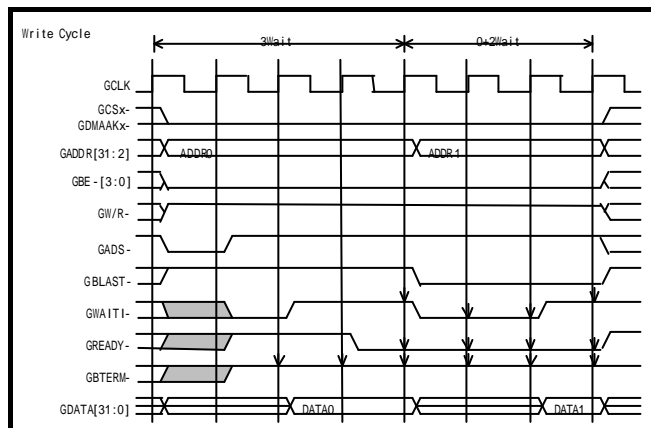
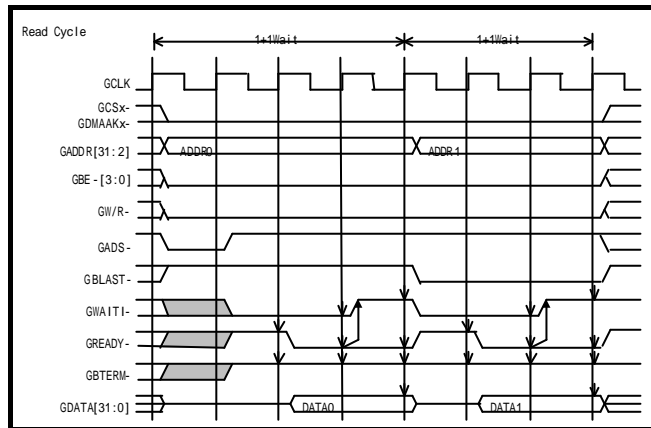
14.6.3. GWAITI-

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の準備ができず、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。

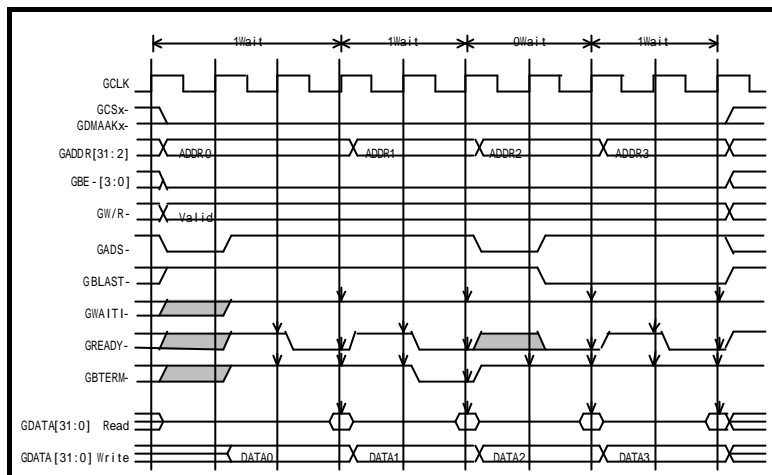


14.6.4. GBTERM-

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。

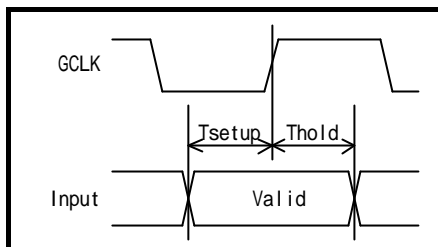
GBTERM-信号によりバースト・サイクルが中断される様子を下図に示します。



14.7. タイミング

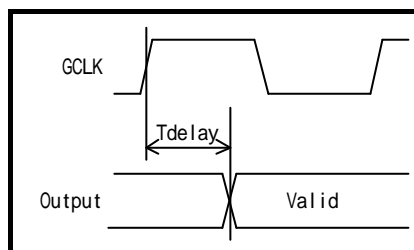
この章では、弊社のマザー・ボードにおけるタイミングについて記述します。CPU ボードはこのタイミングを満たすように設計されています。

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 -