

H 8 / 3 0 4 8 F
アセンブラプログラム作成時の
周辺回路等 補足説明資料

AKI_H8 付属アセンブラ :

A38H.EXE (H 8 用クロスアセンブラ) の扱いについて、パソコン上にて使用する通常のマクロアセンブラと比較すると、いくつかの制限事項がある。

- (1) アセンブル制御の IFDEF ~ ELSE ~ ENDIF の機能は無い。
- (2) インクルードの機能が無い。
これについては、インクルードとは異なる方法で、複数のソースファイルにて 1 本のオブジェクトを生成する手段がある。 この方法について以下に説明を行う。

複数のソースから 1 本のオブジェクトを生成する方法 :

A38H は、コマンドラインにコンマで区切って複数のソースファイルを並べる事が出来る。

例として、test_a、test_b、test_c の 3 本のソースファイルがあった場合には

```
A38H test_a,test_b,test_c[Enter]
```

の形でコマンドラインに入力する。

これにより単純に test_a と test_b と test_c のファイルを連結したような扱いで読み込まれる。

この場合の各ファイルの注意は、ターゲット CPU の指定は、最初に読み込まれる test_a 内に記述する必要がある。 2 番目以降のファイルには必要ない。あれば 2 重定義エラーとなる。

アセンブラソースファイルの最後を示す .END は最後に読み込まれる test_c の最後の行に記述する。 途中に .END が ある場合はそれ以降のソースは無視される。

ファイルの拡張子は、.mar または .src のいずれかがデフォルトになっている。

通常は、先頭のファイルに 定数定義を行うファイルを宣言する形になると思われる。

```
A38H test_a,test_b,test_c[Enter]
```

を実行した結果は、先頭のファイル名 (この場合 test_a) に拡張子 .obj が付いたファイル名 (test_a.obj) にて生成される。

オブジェクトファイル名を指定する場合は、-OBJECT オプションを指定する。
仮にオブジェクトファイル名を test_x とすると

```
A38H test_a,test_b,test_c -OBJECT=test_x[Enter]
```

の形で指定する。

数値データ表記：

10進数： #10

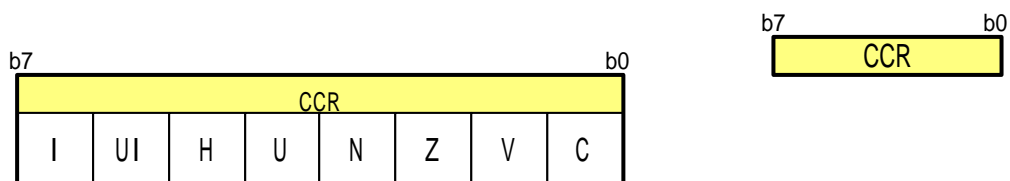
16進数： #H'0A ;
(16進表記の場合最上位桁が A～Fになる場合頭に 0を付ける)

2進数： #B'00001010 ; (8bit長オペランドの場合)

文字定数： #"a" ; シングルコートでない事に注意

内部レジスタ

ER0	ER0		
	E0	R0H	R0L
ER1	ER1		
	E1	R1H	R1L
ER2	ER2		
	E2	R2H	R2L
ER3	ER3		
	E3	R3H	R3L
ER4	ER4		
	E4	R4H	R4L
ER5	ER5		
	E5	R5H	R5L
ER6	ER6		
	E6	R6H	R6L
ER7 (SP)	ER7 (SP)		
	E7	R7H	R7L



CCR各ビットの説明

数値データ表記：

- b7 (I) 割り込みマスクビット：
1にすると NMI以外のハードウェア割り込み（例外処理）が禁止されます。
これをマスクするといいます。
例外処理に入ったときには 1 にセットされます。
- b6 (UI) ユーザビット / 割り込みマスクビット：
ユーザーが自由に使えるフラグビットです。
LDC や ORC など、この命令を直接操作する命令意外では変化しません。
割り込みコントローラの設定で割り込みマスクビットになります。
初期状態では、ユーザービットです。
- b5 (H) ハーフキャリーフラグ：
加算や減算で発生する繰り上がり繰り下がりを示します。
このチェックの対象となるのは演算レジスタのビット 3 です。
ビット 7 の場合は、キャリーフラグ (C) にその結果が現れます。
このフラグは、BCD演算で使います。
- b4 (U) ユーザービット：
ユーザーが自由に使えるフラグビットです。
LDC や ORC など、この命令を直接操作する命令意外では変化しません。
- b3 (N) ネガティブフラグ：
加減算、あるいは転送命令の後で、その結果がマイナスなら 1 になります。
マイナスとは、2 の補数表現での話ですから、最上位ビットが 1 なら
このフラグは 1 になります。
- b2 (Z) ゼロフラグ：
加減算、転送命令で、その結果が 0 なら 1 になります。
- b1 (V) オーバーフローフラグ：
加減算でオーバーフローが発生したら 1 になります。
オーバーフローと キャリーとの違いは
2 の補数表現での桁あふれがオーバーフローです。 プラスの数とプラスの数を
加えてマイナスになるとオーバーフローです。 このときキャリーは
発生しません。 8bitであれば、2 の補数表現でプラス側は最大 127まで、
マイナス側は -128までです。 2 の補数ではなく 0 ~ 255までの値を取ると
考える事も出来ます。 120 + 40を計算するとその結果は 160となり、
0 ~ 255の 8 ビットの範囲に入っていますからキャリーは発生しません。
しかし、2 の補数表現で見たときには最大値 127を越えていますから -32と
なり、プラスとプラスを足してマイナスになっていますからオーバーフロー
フラグが、1 にセットされます。
- b0 (C) キャリーフラグ：
加減算での繰り上がり、繰り下がり で 1 になります。
シフトやローテートでも変化します。

プログラムを作る上で その他に必要なハードウェアの知識

最初に行う事。スタックポインタ (ER7) の設定。

```
MOV.L    #stack, SP      ; スタックポインタの設定
```

PUSH、POPの 動作を C 言語風にかくと

```
プッシュ  * (--ER7) = x;
```

```
ポップ    x = *(ER7++);   となります。
```

このCPUでは、スタックポインタを更新してから保存します。

取り出すときはスタックから取り出してからスタックポインタを更新します。

エンディアン：

上位データが最初のアドレスに来ます。

例えば、H'1234の 16bitデータを H'1000番地に書き込むと

H'1000番地には H'12、 H'1101番地は H'34となります。

16bitのワードデータ、32bitのロングワードデータは先頭アドレスが偶数になるように配置します。

奇数番地に対してワード・ロングワード データをアクセスするとエラーではなく最下位ビットが 0 として扱われます。

(エラーとはならないのでバグの温床となりそうです。気をつけましょう。)

ベクタ：

0 番地から例外処理ベクタテーブルが置かれています。

すぐに使うテーブルは、0 番地から 4 バイトのリセットベクタです。

ここにプログラムの開始番地を設定します。

これを正しく設定しないとプログラムは正常に動作しません。

スタックポインタの値はベクタテーブルにはありませんから、プログラムの先頭でスタックポインタ設定をしなければなりません。

アセンブラの注意事項：

データ領域と align

CPU に対する命令は、必ず偶数番地から始まる必要があります。

ところが、途中で文字列データなどが入ると、そのあとの命令が偶数であるとは限らなくなります。そこを、偶数にしようというのが .align です。

section で指定できる align は、そのアドレスだけで機能します。

そのセクション全体に有効というわけではありません。

そのため、偶数番地から命令を始めようとする、その都度、align=2 とやる必要があります。

.datab が使えませんでした。そのかわりに .res を使いましょう。

文字列データを確保する際の例

プログラムを作る上で その他に必要なハードウェアの知識

文字列データを確保する際の例

```
.CPU      300HA                      ;CPUは H300アドバンスドモード

.section a,data,locate=h'0
bega:     .data.l start                ; スタートベクトル
                                         ; ベクトル領域

.section b,code,locate=h'0400
.org      h'0400

start:                                         ; H'400番地からプログラム
mov.l     @myData, er0
mov.b     @byteD, r0l
bra       start

.section c,data,align=2                    ; 偶数番地からスタート
myData:   .data.l h'12345678
byteD:    .data.b h'a

.align 2                                     ; 偶数番地からスタート
wordD:    .data.w h'b
charD:    .data.b "a","b","c","d","e","f","g"
strD:     .sdata "string data"(h'd)

.align 2                                     ; 偶数番地からスタート
longD:    .data.l h'9abcdef

enda:     .res      1
          .end
```

内臓周辺モジュール

割り込みコントローラ：
割り込みの許可、禁止、優先順位の設定

バスコントローラ：

リフレッシュコントローラ：
DRAMのリフレッシュコントローラ

DMAコントローラ：

I / Oポート

10本の入出力ポートと、1本の入力ポートがあります。

コネクタ：IC信号線 対応表 (1/2)

CN 1

CN1 PIN	IC PIN	信号名
1		GND
2		GND
3	87	P8-0
4	88	P8-1
5	89	P8-2
6	90	P8-3
7	91	P8-4
8	93	PA-0
9	94	PA-1
10	95	PA-2
11	96	PA-3
12	97	PA-4
13	98	PA-5
14	99	PA-6
15	100	PA-7
16	2	PB-0
17	3	PB-1
18	4	PB-2
19	5	PB-3
20	6	PB-4
21	7	PB-5
22	8	PB-6
23	9	PB-7
24	10	RES0
25	12	P9-0
26	13	P9-1
27	14	P9-2
28	15	P9-3
29	16	P9-4
30	17	P9-5
31	18	P4-0
32	19	P4-1
33	20	P4-2
34	21	P4-3
35		5[V]
36		5[V]
37		GND
38		GND
39		POWER
40		POWER

CN 2

CN2 PIN	IC PIN	信号名
1		GND
2		GND
3	62	STBY
4	63	RES
5	64	NMI
6	69	P6-3
7	70	P6-4
8	71	P6-5
9	72	P6-6
10	76	AVCC
11	77	AREF
12	78	P7-0
13	79	P7-1
14	80	P7-2
15	81	P7-3
16	82	P7-4
17	83	P7-5
18	84	P7-6
19	85	P7-7
20	86	AVSS


CN 4


CN4 PIN	IC PIN	信号名
1		POWER
2		GND
3		5[V]
4	U2-8	RX D1
5	U2-13	RX D0
6	U2-7	TX D1
7	U2-14	TX D0
8	10	VPP
9	75	MD2
10	63	RES

CN 5

CN5 PIN	IC PIN	信号名
1		GND
2	73	MD-0
3		GND
4	74	MD-1
5		GND
6	75	MD-2

CN5 (CPU) 設定

 Mode 6 16 [Mbyte]

 Mode 7 1 [Mbyte]

コネクタ：IC信号線 対応表 (2/2)

CN 3

CN3 PIN	IC PIN	信号名
1		GND
2		GND
3	23	P4-4
4	24	P4-5
5	25	P4-6
6	26	P4-7
7	27	P3-0
8	28	P3-1
9	29	P3-2
10	30	P3-3
11	31	P3-4
12	32	P3-5
13	33	P3-6
14	34	P3-7
15	36	P1-0
16	37	P1-1
17	38	P1-2
18	39	P1-3
19	40	P1-4
20	41	P1-5
21	42	P1-6
22	43	P1-7
23	45	P2-0
24	46	P2-1
25	47	P2-2
26	48	P2-3
27	49	P2-4
28	50	P2-5
29	51	P2-6
30	52	P2-7
31	53	P5-0
32	54	P5-1
33	55	P5-2
34	56	P5-3
35	58	P6-0
36	59	P6-1
37	60	P6-2
38	61	CK
39		GND
40		GND

旧マザーボードI/O接続

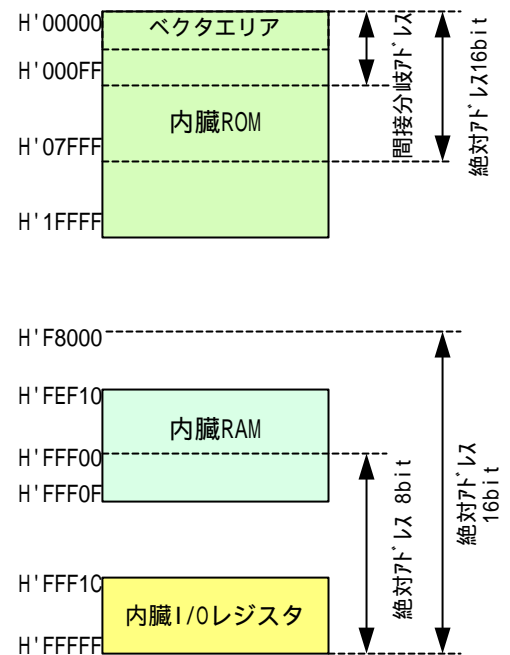
I/O部品	信号名	備考
S1	P4-4	押すと 0、離すと 1
S2	P4-5	"
S3	P4-6	"
S4	P4-7	"
S5-1	P2-0	ONで 0、OFFで 1
S5-2	P2-1	"
S5-3	P2-2	"
S5-4	P2-3	"
S5-5	P2-4	"
S5-6	P2-5	"
S5-7	P2-6	"
S5-8	P2-7	"
LCD_D4	P3-0	データ信号 D4
LCD_D5	P3-1	データ信号 D5
LCD_D6	P3-2	データ信号 D6
LCD_D7	P3-3	データ信号 D7
LCD_RS	P3-4	コマンド / データ選択信号
LCD_E	P3-5	イネーブル信号
LED1	P5-0	1 で点灯、0 で消灯
LED2	P5-1	"
LED3	P5-2	"
Serial_CN5-2	RX_D1	シリアル 1 送信データ
Serial_CN5-3	TX_D1	シリアル 1 受信データ

メモリマップ

CPU MODE 6 (16Mbyte)



CPU MODE 7 (1Mbyte)



CPUのポート

ポート 1	P1_7 / (A7)
	P1_6 / (A6)
	P1_5 / (A5)
	P1_4 / (A4)
	P1_3 / (A3)
	P1_2 / (A2)
	P1_1 / (A1)
	P1_0 / (A0)

ポート 7	P7_7 / (AN7 / DA1)
	P7_6 / AN6 / DA0
	P7_5 / (AN5)
	P7_4 / (AN4)
	P7_3 / (AN3)
	P7_2 / (AN2)
	P7_1 / (AN1)
	P7_0 / AN0

ポート 2	P2_7 / (A15)
	P2_6 / (A14)
	P2_5 / (A13)
	P2_4 / (A12)
	P2_3 / (A11)
	P2_2 / (A10)
	P2_1 / (A9)
	P2_0 / (vA8)

ポート 8	P8_4 / (CS0)
	P8_3 / (IRQ3)
	P8_2 / (IRQ2)
	P8_1 / (IRQ1)
	P8_0 / (IRQ0)

ポート 3	P3_7 / (D15)
	P3_6 / (D14)
	P3_5 / (D13)
	P3_4 / (D12)
	P3_3 / (D11)
	P3_2 / (D10)
	P3_1 / (D 9)
	P3_0 / (D 8)

ポート 9	P9_5 / (SCK1 / IRQ5)
	P9_4 / (SCK0 / IRQ4)
	P9_3 / (RxD1)
	P9_2 / (RxD0)
	P9_1 / (TxD1)
	P9_0 / (TxD0)

ポート 4	P4_7 / (D 7)
	P4_6 / (D 6)
	P4_5 / (D 5)
	P4_4 / (D 4)
	P4_3 / (D 3)
	P4_2 / (D 2)
	P4_1 / (D 1)
	P4_0 / (D 0)

ポート A	PA_7 / (TP7 / TIOCB2 / A20)
	PA_6 / (TP6 / TIOCA2 / A21 /
	SS1)
	PA_5 / (TP5 / TIOCB1 / A22 /
	SS5)
	PA_4 / (TP4 / TIOCA1 / A23 /
	CS6)
ポート A	PA_3 / (TP3 / TIOCB0 / TCLKD)
	PA_2 / (TP2 / TIOCA0 / TCLKC)
	PA_1 / (TP1 / TEND1 / TCLKB)
	PA_0 / (TP0 / TEND0 / TCLKA)

ポート 5	P5_3 / (A19)
	P5_2 / (A18)
	P5_1 / (A17)
	P5_0 / (A16)

ポート B	PB_7 / (TP15 / DREQ1 / ADTRG)
	PB_6 / (TP14 / DREQ0 / CS7)
	PB_5 / (TP13 / TOCXB4)
	PB_4 / (TP12 / TOCXA4)
	PB_3 / (TP11 / TIOCB4)
	PB_2 / (TP10 / TIOCA4)
	PB_1 / (TP9 / TIOCB3)
	PB_0 / (TP8 / TIOCA3)

ポート 6	P6_6 / (LWR)
	P6_5 / (HWR)
	P6_4 / (RD)
	P6_3 / (AS)
	P6_2 / (BACK)
	P6_1 / (BREQ)
	P6_0 / (WAIT)

動作モード 6 デフォルト / ポート設定

ポート 1 Adr Bus 兼用	P1_7(入力)	
	P1_6(入力)	
	P1_5(入力)	
	P1_4(入力)	
	P1_3(入力)	
	P1_2(入力)	
	P1_1(入力)	×
	P1_0(入力)	

ポート 7 アナログ 端子兼用	P7_7(入力)	
	P7_6(入力)	
	P7_5(入力)	
	P7_4(入力)	
	P7_3(入力)	
	P7_2(入力)	
	P7_1(入力)	
	P7_0(入力)	

ポート 2 Adr Bus 兼用	P2_7(入力)	
	P2_6(入力)	
	P2_5(入力)	
	P2_4(入力)	
	P2_3(入力)	
	P2_2(入力)	
	P2_1(入力)	
	P2_0(入力)	

ポート 8	P8_4(入力)	
	P8_3(入力)	
	P8_2(入力)	×
	P8_1(入力)	
	P8_0(入力、出力)	

ポート 3 Data Bus 固定	D15(入出力)	
	D14(入出力)	
	D13(入出力)	
	D12(入出力)	
	D11(入出力)	
	D10(入出力)	
	D 9(入出力)	
	D 8(入出力)	

ポート 9	P9_5 / (SCK1 / <u>IRQ5</u>)	×
	P9_4 / (SCK0 / <u>IRQ4</u>)	
	RxD1(入力)	
	RxD0(入力)	
	TxD1(出力)	
	TxD0(出力)	

ポート 4 Data Bus 兼用	P4_7(入力、出力)	
	P4_6(入力、出力)	
	P4_5(入力、出力)	
	P4_4(入力、出力)	
	P4_3(入力、出力)	
	P4_2(入力、出力)	
	P4_1(入力、出力)	
	P4_0(入力、出力)	

ポート A	A20(出力)	
	PA_6(入力、出力)	
	PA_5(入力、出力)	
	PA_4(入力、出力)	
	PA_3(入力、出力)	
	PA_2(入力、出力)	
	PA_1(入力、出力)	
	PA_0(入力、出力)	

ポート 5 Adr Bus 兼用	P5_3(入力)	
	P5_2(入力)	
	P5_1(入力)	
	P5_0(入力)	

ポート B	PB_7(入力、出力)	×
	PB_6(入力、出力)	
	PB_5(入力、出力)	
	PB_4(入力、出力)	×
	PB_3(入力、出力)	×
	PB_2(入力、出力)	×
	PB_1(入力、出力)	×
	PB_0(入力、出力)	×

ポート 6	<u>LWR</u>	
	<u>HWR</u>	
	<u>RD</u>	
	<u>AS</u>	
	P6_2(入力、出力)	
	P6_1(入力、出力)	
	P6_0(入力、出力)	

動作モード7デフォルト/ポート設定

ポート 1	P1_7(入力、出力)	
	P1_6(入力、出力)	
	P1_5(入力、出力)	
	P1_4(入力、出力)	
	P1_3(入力、出力)	
	P1_2(入力、出力)	
	P1_1(入力、出力)	
	P1_0(入力、出力)	

ポート 7	P7_7(入力)	
	P7_6(入力)	
	P7_5(入力)	
	P7_4(入力)	
	P7_3(入力)	
	P7_2(入力)	
	P7_1(入力)	
	P7_0(入力)	

ポート 2	P2_7(入力、出力)	×
	P2_6(入力、出力)	×
	P2_5(入力、出力)	×
	P2_4(入力、出力)	×
	P2_3(入力、出力)	×
	P2_2(入力、出力)	×
	P2_1(入力、出力)	×
	P2_0(入力、出力)	×

ポート 8	P8_4(入力、出力)	
	P8_3(入力、出力)	
	P8_2(入力、出力)	
	P8_1(入力、出力)	
	P8_0(入力、出力)	

ポート 3	P3_7(入力、出力)	
	P3_6(入力、出力)	
	P3_5(入力、出力)	×
	P3_4(入力、出力)	×
	P3_3(入力、出力)	×
	P3_2(入力、出力)	×
	P3_1(入力、出力)	×
	P3_0(入力、出力)	×

ポート 9	P9_5 / (SCK1 / <u>IRQ5</u>)	
	P9_4 / (SCK0 / <u>IRQ4</u>)	
	RxD1(入力)	
	RxD0(入力)	
	TxD1(出力)	
	TxD0(出力)	

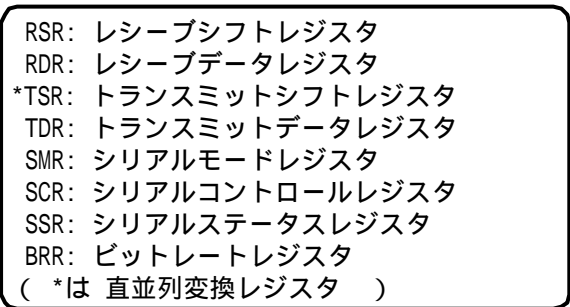
ポート 4	P4_7(入力、出力)	×
	P4_6(入力、出力)	×
	P4_5(入力、出力)	×
	P4_4(入力、出力)	×
	P4_3(入力、出力)	
	P4_2(入力、出力)	
	P4_1(入力、出力)	
	P4_0(入力、出力)	

ポート A	PA_7(入力、出力)	
	PA_6(入力、出力)	
	PA_5(入力、出力)	
	PA_4(入力、出力)	
	PA_3(入力、出力)	
	PA_2(入力、出力)	
	PA_1(入力、出力)	
	PA_0(入力、出力)	

ポート 5	P5_3(入力、出力)	
	P5_2(入力、出力)	
	P5_1(入力、出力)	×
	P5_0(入力、出力)	×

ポート B	PB_7(入力、出力)	
	PB_6(入力、出力)	
	PB_5(入力、出力)	
	PB_4(入力、出力)	
	PB_3(入力、出力)	
	PB_2(入力、出力)	
	PB_1(入力、出力)	
	PB_0(入力、出力)	

ポート 6	P6_6(入力、出力)	
	P6_5(入力、出力)	
	P6_4(入力、出力)	
	P6_3(入力、出力)	
	P6_2(入力、出力)	
	P6_1(入力、出力)	
	P6_0(入力、出力)	



* * * プログラムからアクセスするレジスタ * * *

RDR : レシーブデータレジスタ
受信したシリアルデータを格納するレジスタです。

TDR : トランスミットデータレジスタ
シリアル送信するデータを格納する 8bitのレジスタです。

SMR : シリアルモードレジスタ

SMRは、SCIのシリアル通信フォーマットの設定と、ボーレートジェネレータのクロックソースを選択するためのレジスタです。

SCR : シリアルコントロールレジスタ

SCRは、SCIの送信 / 受信動作、調歩同期式モードでのシリアルクロック出力、割り込み要求の許可 / 禁止、及び送信受信クロックソースの選択を行うレジスタです。

SSR : シリアルステータスレジスタ
SCIの動作状態を示すステータスフラグと、マルチプロセッサビットを内蔵した8bitのレジスタです。

BRR : ビットレートレジスタ
BRRは、SMRの CKS1、CKS0ビットで選択されるボーレートジェネレータの動作クロックとあわせて、シリアル送受信のビットレートを設定する 8bitのレジスタです。

SCIのレジスタ構成

端子構成：

SCIは、チャンネル毎に下図に示すシリアル端子をもっています。

チャンネル	名 称	略称	入出力	機 能
0	シリアルクロック端子	SCK0	入出力	SCI0の クロック入出力
	レシーブデータ端子	RxD0	入力	SCI0の受信データ入力
	トランスミットデータ端子	TxD0	出力	SCI0の送信データ出力
1	シリアルクロック端子	SCK1	入出力	SCI1の クロック入出力
	レシーブデータ端子	RxD1	入力	SCI1の受信データ入力
	トランスミットデータ端子	TxD1	出力	SCI1の送信データ出力

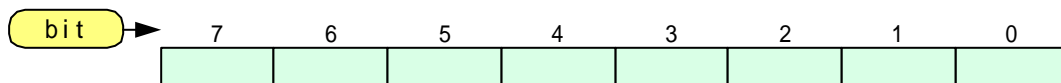
レジスタ構成：

SCIには、下表に示す内部レジスタがあります。 これらのレジスタにより調歩同期式モード/クロック同期式モードの指定、データフォーマットの指定、ビットレートの指定、および送信部、受信部の制御を行う事ができます。

チャンネル	アドレス	名 称	略 称	R/W	初期値
0	H'FFB0	シリアルモードレジスタ	SMR	R/W	H'00
	H'FFB1	ビットレートレジスタ	BRR	R/W	H'FF
	H'FFB2	シリアルコントロールレジスタ	SCR	R/W	H'00
	H'FFB3	トランスミットデータレジスタ	TDR	R/W	H'FF
	H'FFB4	シリアルステータスレジスタ	SSR	R/W*2	H'84
	H'FFB5	レシーブデータレジスタ	RDR	R	H'00
1	H'FFB8	シリアルモードレジスタ	SMR	R/W	H'00
	H'FFB9	ビットレートレジスタ	BRR	R/W	H'FF
	H'FFBA	シリアルコントロールレジスタ	SCR	R/W	H'00
	H'FFBB	トランスミットデータレジスタ	TDR	R/W	H'FF
	H'FFBC	シリアルステータスレジスタ	SSR	R/W*2	H'84
	H'FFBD	レシーブデータレジスタ	RDR	R	H'00

レシーブシフトレジスタ（ RSR ）：

RSRは、シリアルデータを受信するためのレジスタです。



R/W : - - - - - - - -

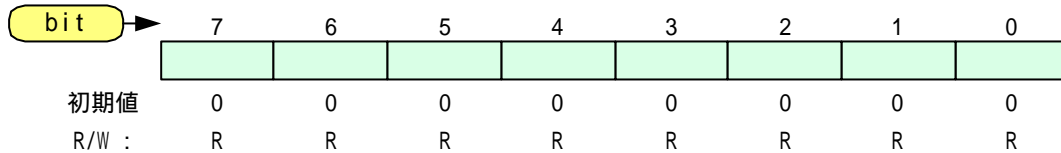
SCIは、RSRに RxD端子から入力されたシリアルデータを LSB（ ビット 0 ）から受信した順にセットしパラレルデータに変換します。 1バイトのデータ受信を終了すると、データは自動的に RDRへ転送されます。

（ CPUから、RSRを直接リード/ライトする事は出来ません。 ）

SCIのレジスタ構成

レシーブデータレジスタ (RDR) :

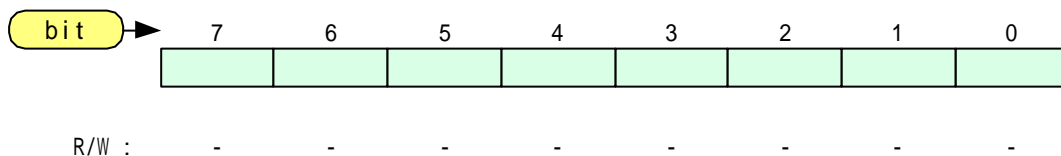
RDRは、受信した シリアルデータを 格納するレジスタです。



SCIは、1 バイトのシリアルデータの受信が終了すると、RSRから RDRへ受信したシリアルデータを転送して格納し受信動作を完了します。 この後、RSRは受信可能になります。このようにRSRと RDRはだぶるバッファになっているため連続した受信動作が可能です。 RDRは、リード専用レジスタで、CPUからライト出来ません。

トランスミットシフトレジスタ (TSR) :

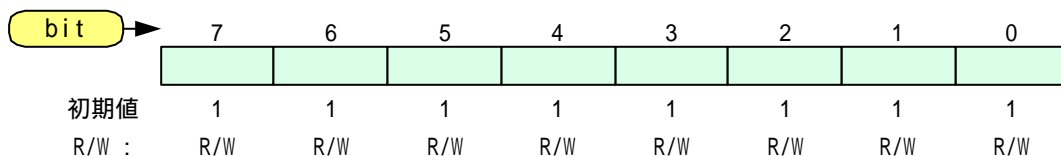
TSRは、シリアルデータを送信するためのレジスタです。



SCIは、TDRから送信データを一旦 TSRに転送し、LSB (bit 0) から順に TxD端子に送り出す事でシリアルデータ送信を行います。1 バイトのデータ送信を終了すると自動的に TDRから、TSRへ次の送信データを転送し、送信を開始します。 但し、SSRの TDREビットが 1 にセットされている場合には、TDRから TSRへのデータ転送は行いません。 CPUから直接 リード/ライトする事は出来ません。

トランスミットデータレジスタ (TDR) :

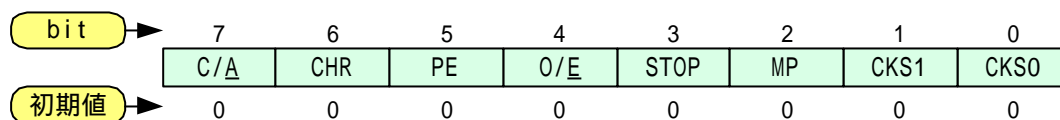
TDRは、シリアル送信するデータを格納する 8 ビットのレジスタです。



SCIは、TSRの空を検出すると、TDRにライトされた送信データをTSRに転送してシリアル送信を開始します。 TSRのシリアル送信中にTDRに次の送信データをライトしておくと、連続シリアル送信が出来ます。TDRは、常に CPUによるリード/ライトが可能です。

SCIの SMRとSCRレジスタ設定

シリアルモードレジスタ (SMR)



C/A	0	調歩同期式モード
	1	クロック同期式モード

STOP	0	1ストップビット
	1	2ストップビット

CHR	0	8 bit (調歩同期)
	1	7 bit (調歩同期)

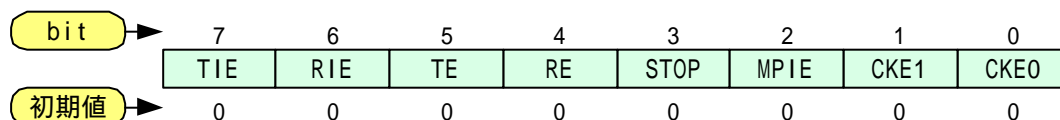
MP	0	マルチプロセッサ禁止
	1	マルチプロセッサ選択

PE	0	パリティ無し
	1	パリティ有り

CKS 1	CKS0	クロックの状態
0	0	クロック (初期値)
	1	/ 4 クロック
1	0	/16 クロック
	1	/64 クロック

O/E	0	Even Parity
	1	Odd Parity

シリアルコントロールレジスタ (SCR)



TIE	0	送信データエンプティ割込み要求禁止
	1	送信データエンプティ割込み要求許可

TE	0	送信動作を禁止
	1	送信動作を許可

RIE	0	受信データフル割込み要求と	禁止
	1	受信エラー割込み要求を	許可

RE	0	受信動作を禁止
	1	受信動作を許可

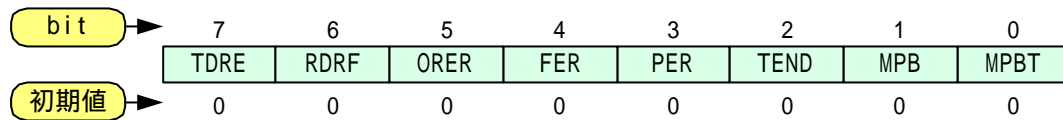
MPIE	0	マルチプロセッサ割り込み禁止
	1	マルチプロセッサ割り込み許可

TEIE	0	送信終了割込み禁止
	1	送信終了割込み許可

CKE1	CKE0	説明 (調歩同期の場合)
0	0	内部クロック / SCK端子は入出力ポート
	1	内部クロック / SCK端子はクロック出力
1	0	外部クロック / SCK端子はクロック入力
	1	外部クロック / SCK端子はクロック入力

SCIの SSRレジスタ内容

シリアルステータスレジスタ(SSR)



TDRE	0	TDRに有効な送信データが 格納されている事を表示 <クリア条件> TDRE=1 を読んだ後、0 を書き込む
	1	TDRに有効な送信データが 無い事を表示 <セット条件> リセットまたはスタンバイモードのとき、または RDRF=1 を読みこんだ後 0 を書き込んだとき

RDRF	0	RDRに受信データが格納されていない事を表示 <クリア条件> リセットまたはスタンバイモードのとき、または RDRF=1 を読んだ後、0 を書き込んだ時
	1	RDRに受信データが格納されている事を表示 <セット条件> シリアル受信が正常終了し、RSRから RDRへ受信データが転送された時

ORER	0	受信中オーバランエラー 無し <クリア条件> リセットまたはスタンバイモードのとき、または ORER=1 を読んだ後、0 を書き込んだ時
	1	受信中オーバランエラー 有り <セット条件> RDRF=1 の状態で次のシリアル受信を完了した時

FER	0	受信中フレーミングエラー 無し <クリア条件> リセットまたはスタンバイモードのとき、または FER=1 を読んだ後、0 を書き込んだ時
	1	受信中フレーミングエラー 有り <セット条件> SICが受信終了時にデータ末尾のストップビット1をチェックしそれが 0で有った時

PER	0	受信中パリティエラー 無し <クリア条件> リセットまたはスタンバイモードのとき、または PER=1 を読んだ後、0 を書き込んだ時
	1	受信中パリティエラー 有り <セット条件> 受信データと パリティビットの 1 の合計が 0/Eの設定と一致しなかった時

TEND	0	送信中である事を表示 <クリア条件> TDRE=1を読んだ後、TDREフラグに 0を書き込んだ時、または、DMACで TDRにデータを書き込んだ時。
	1	送信を終了した事を表示 <セット条件> リセットまたはスタンバイモードの時、または SCRのTEビットが 0 の時、または最後尾ビット送信時にTDRE=1のとき

MBP	0	マルチプロセッサビットが 0 の データを受信した事を表示
	1	マルチプロセッサビットが 1 の データを受信した事を表示

MPBT	0	マルチプロセッサビットが 0 の データを送信
	1	マルチプロセッサビットが 1 の データを送信

SCIの BRRレジスタ内容

ボンドレイトレジスタ(BRR)の設定

ボーレイトの設定は、BRRに 8bitの値 (N) を格納する事で行う。
Nを決める要素としては、ビットレートB、クロック周波数、ボーレイトジェネレータ入力クロック比nなどがあります。

調歩同期式モード、CPUクロック 16[MHz]の場合のNを求める表は以下
のようになります。

B (bps)	CPU 16[MHz]		
	n	N	誤差(%)
110	3	70	0.03
150	2	207	0.16
300	2	103	0.16
600	1	207	0.16
1200	1	103	0.16
2400	0	207	0.16
4800	0	103	0.16
9600	0	51	0.16
19200	0	25	0.16
31250	0	15	0.00
38400	0	12	0.16

↑
BRRに設定する値

n	クロック	SMRの設定値	
		CKS1	CKS0
0		0	0
1	/ 4		1
2	/ 16	1	0
3	/ 64		1

割り込み処理

割り込み例外処理を開始させる要因には、7つの外部割り込み（NMI，IRQ0～IRQ5）と、内部周辺モジュールからの要求による30の内部要因があります。
割り込み要因と要因数を以下に示します。

割り込み	外部割り込み	NMI (1)
		IRQ0 ~ IRQ5 (6)
	内部割り込み	WDT (1) *1
		リフレッシュコントローラ (1) *2
		ITU (15)
		DMAC (4)
		SCI (8)
		A/D変換機 (1)

注： ()内は要因数を示します。

*1 WDTをインターバルタイマとして使用した時は、カウンタがオーバーフローすると割り込み要求を発生します。

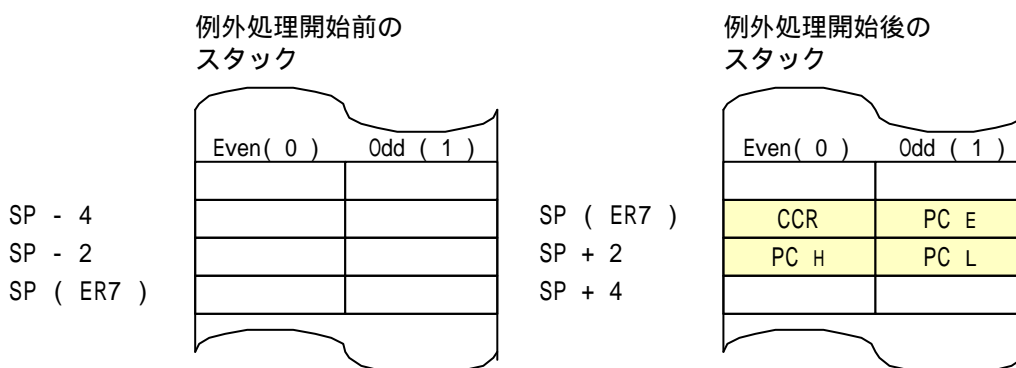
*2 リフレッシュコントローラをインターバルタイマとして使用した時コンペアマッチにより割り込み要求を発生します。

トラップ命令：

トラップ命令例外処理は、TRAPA命令を実行すると例外処理を開始します。
例外処理によって、SYSCRのUEビットが1のときはCCRのIビットが1にセットされ、UEビットが0のときにはCCRのIビット、UIビットがそれぞれ1にセットされます。
TRAPA命令は、命令コード中で指定した0～3のベクタ番号に対応するベクタテーブルからスタートアドレスを取り出します。

例外処理誤のスタックの状態：

トラップ命令、例外処理及び割り込み例外処理後のスタック状態を以下に示します。



CCR : コンディションコードレジスタ
PC E : プログラムカウンタ(PC)の b23 ~ b16
PC H : プログラムカウンタ(PC)の b15 ~ b 8
PC L : プログラムカウンタ(PC)の b 8 ~ b 0
SP : スタックポインタ

PUSH する時、SPをデクリメントしてから レジスタの退避を行う。
POP する時、SPをインクリメントしてからレジスタの復帰を行う。

SP(ER7)の 設定は、必ず偶数アドレスで行う。

割り込みコントローラ

割り込みコントローラには次の特徴があります。

IPRにより、優先順位を設定可能

割り込み優先順位を設定するインタラプトプライオリティレジスタA、B (IPR A,B) を備えており、NMI以外の割り込みを要因ごとまたはモジュールごとに 2 レベルの優先順位を設定できます。

CPUのコンディションコードレジスタ (CCR) の I、IUビットにより、3 レベルの許可 / 禁止状態を設定可能。

独立したベクタアドレス

すべての割り込み要因には独立したベクタアドレスが割り当てられており、割り込み処理ルーチンでの要因を判断する必要がありません。

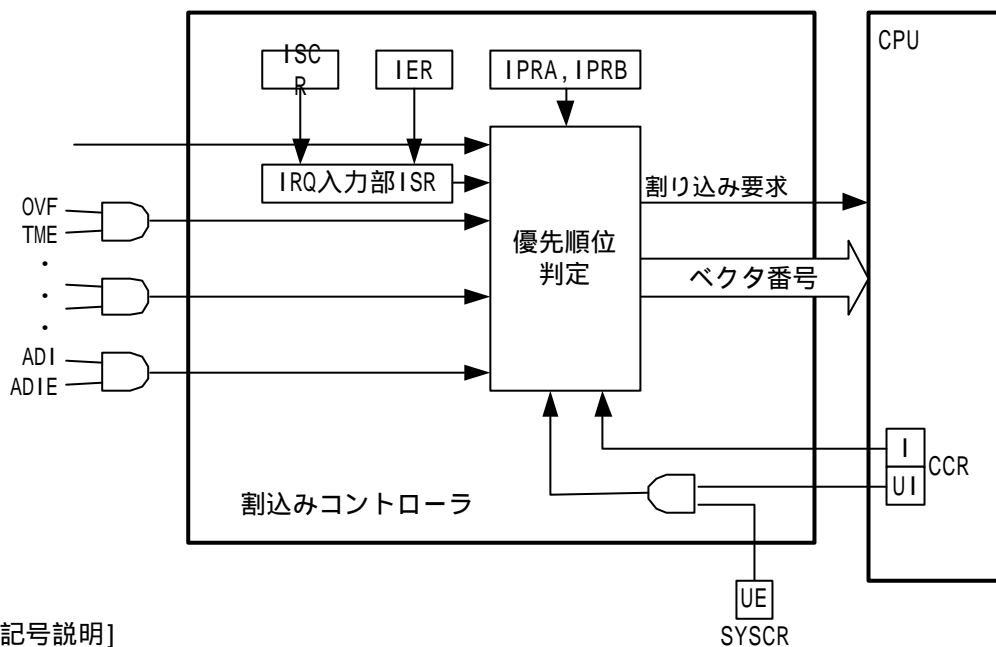
7本の外部割り込み端子

NMIは最優先の割り込みで常に受けつけられます。

NMIは立ち上がりエッジ / 立ち下がりエッジを選択できます。

また、IRQ0 ~ IRQ5は 立下りエッジ / レベルセンスを独立に選択できます。

ブロック図



[記号説明]

ISCR: IRQセンスコントロールレジスタ
 IER: IRQイネーブルレジスタ
 ISR: IRQステータスレジスタ
 IPRA: インタラプトプライオリティレジスタA
 IPRB: インタラプトプライオリティレジスタB
 SYSCR: システムコントロールレジスタ

端子構成：

名称	略称	入出力	機能
ノンマスク割り込み	NMI	入力	マスク不可能な外部割り込み 立ち上がりエッジ / 立下りエッジ選択可能
外部割り込み要求5 ~ 0	IRQ5 ~ IRQ0	入力	マスク可能な外部割り込み 各端子毎 立下りエッジ / レベルセ ンス選択可能

割込みコントローラ

レジスタ構成：

アドレス*1	名 称	略称	R/W	初期値
H'FFF2	システムコントロールレジスタ	SYSCR	R/W	H'0B
H'FFF4	IRQセンスコントロールレジスタ	ISCR	R/W	H'00
H'FFF5	IRQイネーブルレジスタ	IER	R/W	H'00
H'FFF6	IRQステータスレジスタ	ISR	R/[W]*2	H'00
H'FFF8	インタラプトプライオリティレジスタA	IPRA	R/W	H'00
H'FFF9	インタラプトプライオリティレジスタB	IPRB	R/W	H'00

[注] *1 アドレスの下位 16bitを示しています。

*2 フラグをクリアするための 0 書込みのみ可能です。

システムコントロールレジスタ (SYSCR)：

SYSCR、8bitのリード/ライト可能なレジスタで、ソフトウェアスタンバイモードの制御、CCRの UIビットの動作の選択、NMIの検出エッジの選択、及び内臓RAMの有効/無効の選択を行います。

SYSCRは、リセット、またはハードウェアスタンバイモード時に、H'0Bにイニシャライズされます。ソフトウェアスタンバイモードではイニシャライズされません。

bit	7	6	5	4	3	2	1	0
	SSBY	STS2	STS1	STS0	UE	NMIEG	-	RAME
初期値	0	0	0	0	1	0	1	1

(b7)ソフトウェアスタンバイ (SSBY)

ソフトウェアスタンバイモードへの変移を指定します。

なお、外部割込みによりソフトウェアスタンバイモードが解除され、通常動作に変移した時、このビットはセットされたままです。

クリアする場合は、0 をライトして下さい。

SSBY	説 明
0	SLEEP命令実行後、スリープモードに変移
1	SLEEP命令実行後、ソフトウェアスタンバイモードに変移

(b6 ~ b4)スタンバイタイムセレクト (STS2 ~ STS0)

外部割込みによって、ソフトウェアスタンバイモードを解除する場合に、内部クロックが安定するまでCPUと内臓周辺モジュールが待機する時間を指定します。水晶発振の場合、動作周波数に応じて待機時間が 7[ms]以上となるように設定して下さい。

SSBY	SSBY	SSBY	説 明
SSBY	SSBY	SSBY	
0	0	0	待機時間 = 8192ステート
		1	待機時間 = 16384ステート
	1	0	待機時間 = 32768ステート
		1	待機時間 = 65536ステート
1	0	0	待機時間 = 131072ステート
		1	待機時間 = 1024ステート
	1	-	>> 使用禁止 <<

割込みコントローラ

(b3) ユーザイネーブルビット (UE)

CCRの UIビットをユーザビットとして使用するか割込みマスクビットとして使用するかを選択します。

UE	説 明
0	CCRの UIビットを割り込みマスクビットとして使用
1	CCRの UIビットをユーザビットとして使用

(b2) NMIエッジセレクト (NMIEG)

NMI端子の入力エッジ選択を行います。

NMIEG	説 明
0	NMI入力の下向きエッジで割込み要求を発生
1	NMI入力の立ち上がりエッジで割込み要求を発生

(b1) リザーブビット

リザーブビットです。 読み出すと常に 1 です。書き込みは無効です。

(b0) RAMイネーブル (RAME)

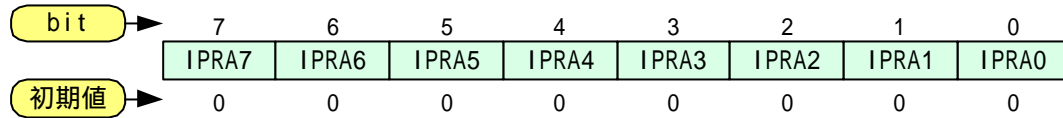
内臓RAMの有効、無効を選択します。 RAMEビットは、RES端子の立ち上がりエッジでイニシャルされます。 ソフトウェアスタンバイモードでは、イニシャライズされません。

RAME	説 明
0	内臓RAM無効
1	内臓RAM有効

割り込みコントローラ

インタラプトプライオリティレジスタ A (IPRA) :

IPRAは各 8bitの Read/Write可能なレジスタで割り込みの優先順位を制御します。



(b7) プライオリティレベル A 7 (IPRA7)

IRQ0の 割り込み要求のプライオリティレベルを設定します。

IPRA7	説 明
0	IRQ0 の 割り込み要求はプライオリティレベル 0 (非優先) (初期値)
1	IRQ0 の 割り込み要求はプライオリティレベル 1 (優先)

(b6) プライオリティレベル A 6 (IPRA6)

IRQ1の 割り込み要求のプライオリティレベルを設定します。

IPRA6	説 明
0	IRQ1 の 割り込み要求はプライオリティレベル 0 (非優先) (初期値)
1	IRQ1 の 割り込み要求はプライオリティレベル 1 (優先)

(b5) プライオリティレベル A 5 (IPRA5)

IRQ2、IRQ3の 割り込み要求のプライオリティレベルを設定します。

IPRA5	説 明
0	IRQ2、IRQ3 の 割り込み要求はプライオリティレベル 0 (非優先) (初期値)
1	IRQ2、IRQ3 の 割り込み要求はプライオリティレベル 1 (優先)

(b4) プライオリティレベル A 4 (IPRA4)

IRQ4、IRQ5の 割り込み要求のプライオリティレベルを設定します。

IPRA4	説 明
0	IRQ4、IRQ5 の 割り込み要求はプライオリティレベル 0 (非優先) (初期値)
1	IRQ4、IRQ5 の 割り込み要求はプライオリティレベル 1 (優先)

(b3) プライオリティレベル A 3 (IPRA3)

WDT、リフレッシュコントローラの割り込み要求のプライオリティレベルを設定します。

IPRA3	説 明
0	WDT、リフレッシュコントローラの割り込み要求は プライオリティレベル 0 (非優先) (初期値)
1	WDT、リフレッシュコントローラ の割り込み要求は プライオリティレベル 1 (優先)

(b2) プライオリティレベル A 2 (IPRA2)

ITUチャンネル 0 の割り込み要求のプライオリティレベルを設定します。

IPRA2	説 明
0	ITUチャンネル 0 の割り込み要求はプライオリティレベル 0 (非優先) (初期値)
1	ITUチャンネル 0 の割り込み要求はプライオリティレベル 1 (優先)

割込みコントローラ

(b1) プライオリティレベルA 1 (IPRA1)

ITUチャンネル1の割込み要求のプライオリティレベルを設定します。

IPRA1	説 明
0	ITUチャンネル1の割込み要求はプライオリティレベル0（非優先）（初期値）
1	ITUチャンネル1の割込み要求はプライオリティレベル1（優先）

(b0) プライオリティレベルA 0 (IPRA0)

ITUチャンネル2の割込み要求のプライオリティレベルを設定します。

IPRA0	説 明
0	ITUチャンネル2の割込み要求はプライオリティレベル0（非優先）（初期値）
1	ITUチャンネル2の割込み要求はプライオリティレベル1（優先）

インタラプトプライオリティレジスタB (IPRB) :

IPRBは各 8bitの Read/Write可能なレジスタで割込みの優先順位を制御します。

bit	7	6	5	4	3	2	1	0
	IPRB7	IPRB6	IPRB5	-	IPRB3	IPRB2	IPRB1	-
初期値	0	0	0	0	0	0	0	0

(b7) プライオリティレベルB 7 (IPRB7)

ITUチャンネル3の割込み要求のプライオリティレベルを設定します。

IPRB7	説 明
0	ITUチャンネル3の割込み要求はプライオリティレベル0（非優先）（初期値）
1	ITUチャンネル3の割込み要求はプライオリティレベル1（優先）

(b6) プライオリティレベルB 6 (IPRB6)

ITUチャンネル4の割込み要求のプライオリティレベルを設定します。

IPRB6	説 明
0	ITUチャンネル4の割込み要求はプライオリティレベル0（非優先）（初期値）
1	ITUチャンネル4の割込み要求はプライオリティレベル1（優先）

(b5) プライオリティレベルB 5 (IPRB5)

DMAC (チャンネル 0, 1) の割込み要求のプライオリティレベルを設定します。

IPRB6	説 明
0	DMAC(チャンネル 0, 1)の割込み要求はプライオリティレベル0（非優先）（初期値）
1	DMAC(チャンネル 0, 1)の割込み要求はプライオリティレベル1（優先）

(b4) リザーブビット

リザーブビットです。 Read/Write可能ですが、優先順位に関係ありません。

割込みコントローラ

(b3) プライオリティレベルB 3 (IPRB3)

SCIチャンネル0の割込み要求のプライオリティレベルを設定します。

IPRB3	説 明
0	SCIチャンネル0の割込み要求はプライオリティレベル0（非優先）（初期値）
1	SCIチャンネル0の割込み要求はプライオリティレベル1（優先）

(b2) プライオリティレベルB 2 (IPRB2)

SCIチャンネル1の割込み要求のプライオリティレベルを設定します。

IPRB2	説 明
0	SCIチャンネル1の割込み要求はプライオリティレベル0（非優先）（初期値）
1	SCIチャンネル1の割込み要求はプライオリティレベル1（優先）

(b1) プライオリティレベルB 1 (IPRB1)

A/D変換器の割込み要求のプライオリティレベルを設定します。

IPRB1	説 明
0	A/D変換器の割込み要求はプライオリティレベル0（非優先）（初期値）
1	A/D変換器の割込み要求はプライオリティレベル1（優先）

(b0) リザーブビット

リザーブビットです。 Read/Write可能ですが、優先順位には関係ありません。

IRQステータスレジスタ (ISR) :

ISRは、8bitの Read・Write可能なレジスタで、IRQ0～IRQ5割込み要求のステータスの表示を行います。

bit	7	6	5	4	3	2	1	0
	-	-	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
初期値	0	0	0	0	0	0	0	0

(b7, b6) リザーブビット

リザーブビットです。 Read/Write可能ですが、優先順位には関係ありません。

(b5 ~ b0) IRQ5 ~ IRQ0 フラグ (IRQ5F ~ IRQ0F)

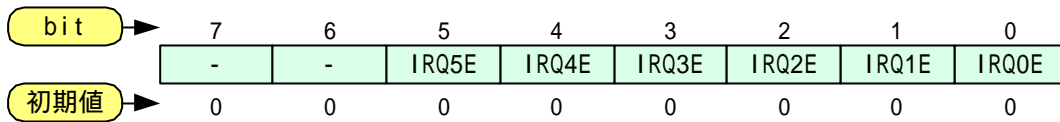
IRQ5 ~ IRQ0 の 割込み要求のステータスの表示を行います。

b5 ~ b0	説 明
0	[クリア条件] (1) IRQnF=1の状態で IRQnFフラグをリードした後、IRQnFフラグを0にライトした時 (2) IRQnSC=0、IRQ入力 Highレベルの状態での割込み例外処理を実行した時 (3) IRQnSC=1の状態でのIRQn割込み例外処理を実行した時
1	[セット条件] (1) IRQnSC=0の状態でのIRQ入力 Lowレベルになった時 (2) IRQnSC=1の状態でのIRQ入力に立下りエッジが発生した時

割込みコントローラ

I R Q イネーブルレジスタ (IER) :

IERは、8bitの Read・Write可能なレジスタで、IRQ 0 ~ IRQ5割込み要求の 許可 / 禁止を制御します。



(b7, b6)リザーブビット

リザーブビットです。

Read/Write可能ですが、割込み要求の 許可 / 禁止には関係ありません。

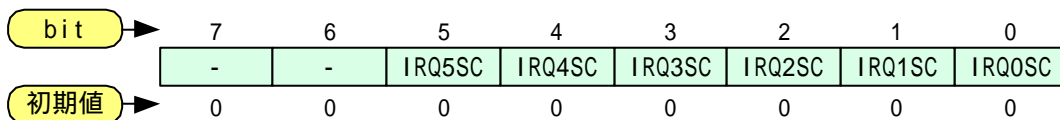
(b5 ~ b0)IRQ5 ~ IRQ0 フラグ (IRQ5F ~ IRQ0F)

IEQ5 ~ IRQ0 の 割込み要求のステータスの表示を行います。

b5 ~ b0	説 明
0	IRQ5 ~ IRQ0 の割込みを禁止 (初期値)
1	IRQ5 ~ IRQ0 の割込みを許可

I R Q センスコントロールレジスタ (ISCR) :

ISCRは、8bitの Read・Write可能なレジスタで、IRQ 0 ~ IRQ5端子の入力レベルセンスまたは立下りエッジを選択します。



(b7, b6)リザーブビット

リザーブビットです。 Read/Write可能ですが、レベルセンス

または立下りエッジの選択には関係ありません。

(b5 ~ b0)IRQ5 ~ IRQ0 センスコントロール (IRQ5SC ~ IRQ0SC)

IEQ5 ~ IRQ0 の 割込みを IEQ5 ~ IRQ0端子のレベルセンスで要求するか、立下りで要求するかを選択します。

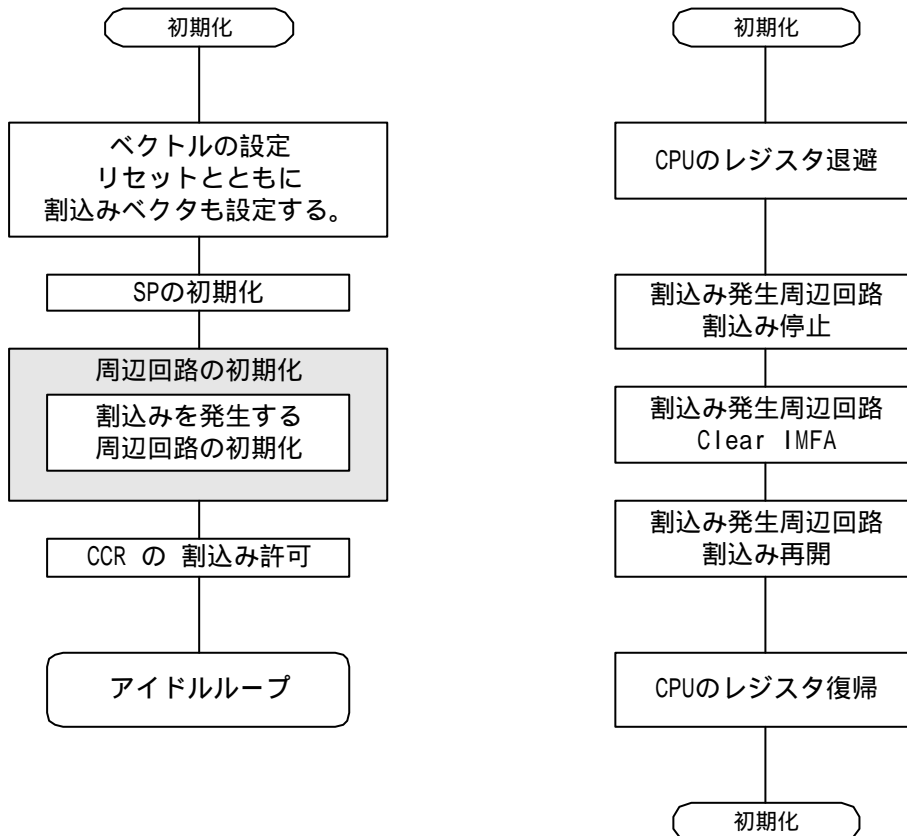
b5 ~ b0	説 明
0	IRQ5 ~ IRQ0 のLowレベルで割込み要求を発生 (初期値)
1	IRQ5 ~ IRQ0 の立下りエッジで割込み要求を発生

↑
↓
低

割り込みベクトルテーブル (2/2)

割り込み要因	要因発生元	ベクタ 番号	ベクタ アドレス	IPR	優先 順位
ERI0 (受信エラー 0) RXI0 (受信データフル 0) TXI0 (送信データエンプティ 0) TEI0 (送信終了 0)	SCIチャネル0	52	H'00D0	IPRB3	高 ↑ ↓ 低
		53	H'00D4		
		54	H'00D8		
		55	H'00DC		
ERI1 (受信エラー 1) RXI1 (受信データフル 1) TXI1 (送信データエンプティ 1) TEI1 (送信終了 1)	SCIチャネル1	56	H'00E0	IPRB3	
		57	H'00E4		
		58	H'00E8		
		59	H'00EC		
ADI (A/D変換 終了)	A/D	60	H'00F0	IPRB1	

割り込み処理のシーケンス (概略) :



AKI_H8用モニタ・デバugg

組み型モニタ概要：

組み型モニタはユーザの実機に組み込まれユーザプログラムのデバuggを行うソフトウェアをいいます。組み型モニタは、ユーザの実機のROMに格納し、ホスト端末よりダウンロードしたユーザプログラムのデバuggを行います。

ホスト端末はパソコンの通信ソフト（ハイパーターミナル）等を使用します。

ユーザプログラムの範囲と制限：

このモニタは、RS-232Cを介してホスト端末とインタフェースを行います。

AKI_H8の SCI-1を占有します。（SCI-0はユーザにて使用可能）

ROM領域はモニタが使用します。

RAM領域の先頭、FEF10 ~ FEFE3 はモニタが使用します。

RAM領域の FF000 ~ FFFF0 は ユーザ用の仮想割り込みベクトル領域になります。

ユーザプログラムは、RAM領域の FF100 ~ FFF00（約 3.5 Kbyteのアドレス空間）で動作します。

スタックポインタは、RAMの後端アドレス FFF10 を 通常 設定します。

FFEF10	モニター使用 (240 byte)
FFF000	仮想割り込みベクトル領域 (256 byte)
FFF100	() ユーザプログラム (3600 byte)
FFFF10	() スタック領域

リセットベクタ (0) 組み型モニターの起動用に使用

ベクタ番号 1 シングルステップの制御として使用

ベクタ番号 2 ブレークポイントの制御として使用

ホスト通信用 RS-232Cの設定

通信速度	9600 bps
データ長	8 bit
パリティ制御	なし
ストップビット	1
フロー制御	X-ON / X-OFF

モニターでデバuggする際のソースの変更：

モニター上でデバuggされるユーザプログラムは、ROMではなく、RAM上に配置される。
この関係で、割り込みベクトル、プログラムのコード領域、プログラムのデータ領域、
計3つのセクションを全てRAM上に配置しなければならない。

通常、メモリ上には

- | | |
|-----------------|--------------|
| 1. 割り込みベクトルテーブル | < ROM領域に配置 > |
| 2. プログラム（コード部分） | < ROM領域に配置 > |
| 3. プログラム（データ部分） | < RAM領域に配置 > |
| 4. スタック領域 | < RAM領域に配置 > |

このようなセクションが配置される。
モニターデバuggを行う場合

1. 割り込みベクトルテーブルは、RAM上の仮想割り込みベクトルテーブルに配置する。
（ FF000 ~ FF0FF ）その場合、リセットベクトルは設定しない。
（ リセットベクトルはモニタが使用しているため。 ）
2. プログラム（コード部分）は、RAMの FF100から配置する。
モニタ上にてユーザプログラムの開始番地は FF100に設定されていると思われる。
3. プログラムのデータ部分は、通常先頭で RAMの開始アドレスを指定するが
モニタ上で使用する場合は、
データ領域先頭のセクション指定はコメントにしておく。
これにより、コード部の 直後にデータ部が配置される形になる。
4. スタックは通常どおり FFF10を 設定する。

懸案事項として、アセンブラにて、複数のソースモジュールを連結して実行形式を作っている場合は、どのように配置されるか確認の必要あり。
C言語ソースの場合は、モニタはマシン語なのでデバuggが厄介と思われる。

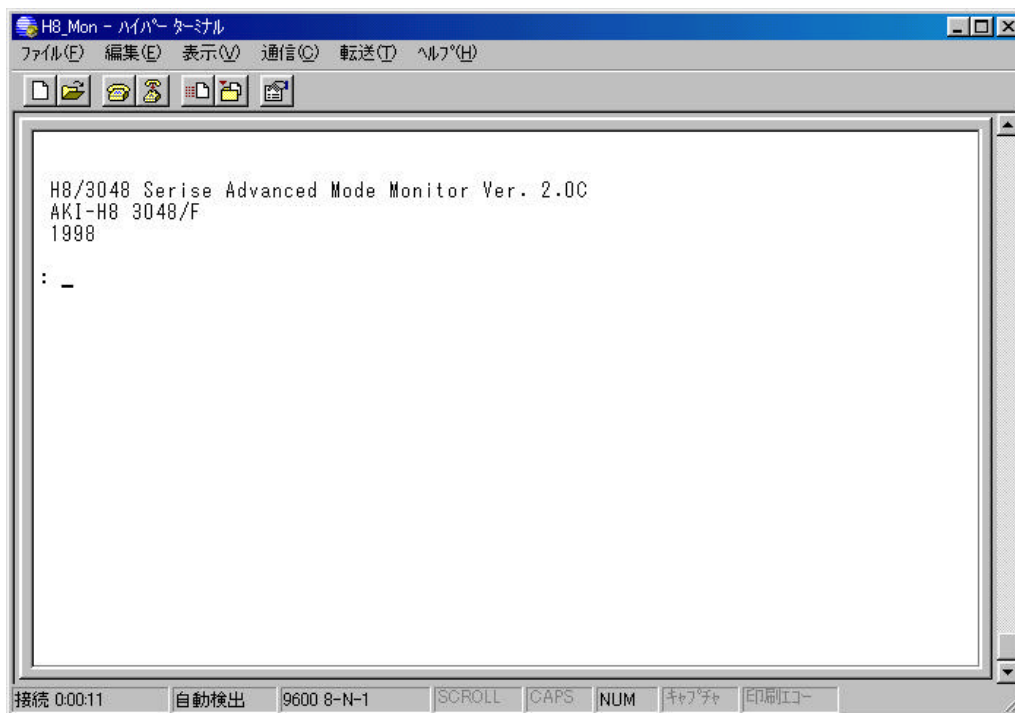
CPU内蔵 RAMは4 Kbyteなのでデバugg出来るプログラムサイズは、最大
3.5 Kbyteぐらいが限度と思われる。
増設 RAMを使用する場合プログラムの配置位置が異なるのでロケーションに
気をつけなければならない。

AKI_H8用モニタ・デバッガ

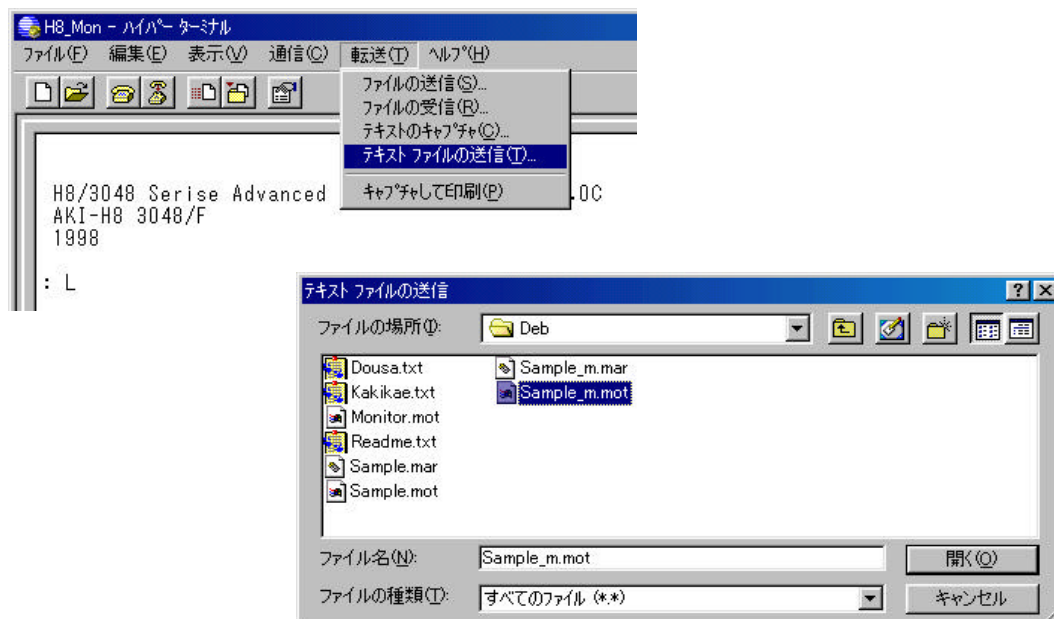
ユーザプログラムを、H 8 ヘダウンロード：

H 8 SC11と、パソコンCOM1を RS-232Cケーブルにて接続する。
Windows環境にて ハイパーターミナルを起動して COM1: 9600,N,8,1 (Xon / Xoff)
に設定する。

次にH 8 の電源を入れる。
ハイパーターミナルの画面に以下のメッセージが表示される。



この状態で、ダウンロードを行うには コマンド L[Enter]を入力する。
次に、ハイパーターミナルメニューの 転送(T) -> テキストファイルの送信(T)をクリックする。 「テキストファイルの送信」のオープンダイアログが表示されるので転送 (デバッグ)する MOTファイルを選択し [OK]をクリックする。
これにより、MOTファイルが転送されRAM上にユーザプログラムが配置される。



AKI_H8用モニタ・デバッガ

ダウンロードが終了すると以下のような表示がなされる。

```
H8/3048 Serise Advanced Mode Monitor Ver. 2.0C
AKI-H8 3048/F
1998

: L
  Top Address=FF070
  End Address=FF1A9
:
```

これによりデバッグの準備が完了といえる。
この後は、ターミナル経由でコマンドを入力し、モニタデバッガを操作する。

モニタのコマンド：

コマンド	機 能
Break point	ブレークポイントの設定、表示、解除
Dump	メモリ内容のダンプ
DisAssemble	逆アセンブル
Fill	データの書き込み
Go	ユーザプログラムの実行
H8 status	内臓周辺機能の状態表示
Load	ユーザプログラムのダウンロード
Memory	メモリ内容の表示、変更
Register	CPUレジスタの一覧表示
Step	シングルステップの実行
.(register)	CPUレジスタの表示、変更
help	コマンドヘルプ
[Enter] key	繰り返し実行
(command).	コマンド履歴
interrupt	内臓周辺機能割込み管理
abort	アボート管理

B (ブレークポイントの設定、表示、解除) :

コマンドフォーマット

- (a) :B <アドレス>[Enter]
- (b) :B -<アドレス>[Enter]
- (c) :B[Enter]

- (a) ブレークポイントの設定
- (b) 設定されているブレークポイントの解除
- (c) 設定されているブレークポイントの表示

ブレークポイントは、RAM領域に配置されているプログラム領域の
OPEコード(命令コード)部分に設定しないと動作は保障されない。
ブレークポイントのアドレス指定は、FFF100 の様に 6 桁で指定して下さい。

D (メモリ内容のダンプ) :

コマンドフォーマット

D <アドレス 1> 「<アドレス 2>」 「;<サイズ>」[Enter]

<アドレス 1>:	ダンプするメモリの先頭アドレス	
<アドレス 2>:	ダンプするメモリの最終アドレス	「省略可」
<サイズ>:	表示サイズの指定	「省略可」
	B	1 バイト単位
	W	2 バイト単位
	L	4 バイト単位
	省略時	1 バイト単位

DA (逆アセンブル) :

コマンドフォーマット

DA <アドレス 1> 「<アドレス 2>」[Enter]

<アドレス 1>	逆アセンブルするメモリの先頭アドレス	
<アドレス 2>	逆アセンブルするメモリの最終アドレス	「省略可」

機能 :

- (a) 指定されたメモリ領域の内容を逆アセンブルします。
<アドレス 2>が省略された場合、16命令分を逆アセンブルします。
- (b) [Enter]のみの入力では前のDAコマンドで表示した次のアドレスから
16命令分を連続的に逆アセンブルします。

F (データの書き込み) :

コマンドフォーマット

F <アドレス 1> <アドレス 2> <書き込みデータ>[Enter]

<アドレス 1>	書き込みするメモリの先頭アドレス
<アドレス 2>	書き込みするメモリの最終アドレス
<書き込みデータ>	1 バイトの書き込みデータ

機能 :

指定されたメモリ領域に <書き込みデータ>で指定されたデータを書き込みます。

G (ユーザプログラムの実行) :

コマンドフォーマット

G 「<アドレス>」 [Enter]

<アドレス> 実行するユーザプログラムの先頭アドレス。

機能 :

現在のプログラムカウンタ値ないしは、<アドレス>で指定したアドレスより
ユーザプログラムを実行します。

H 8 (ユーザプログラムの実行) :

コマンドフォーマット

H8 「<周辺機器名>」 [Enter]

H8 ? 周辺機器のヘルプ表示がなされる

<周辺機器名>

DMAC0	DMAコントローラ 0
DMAC1	DMAコントローラ 1
ITU	16bit Timer / Counter
ITU0	16bit Timer / Counter 0
ITU1	16bit Timer / Counter 1
ITU2	16bit Timer / Counter 2
ITU3	16bit Timer / Counter 3
ITU4	16bit Timer / Counter 4
TPC	プログラマブルタイミングパターンコントローラ
WDT	ウォッチドックタイマー
SCI0	シリアルコントローラインタフェース 0
SCI1	シリアルコントローラインタフェース 1
I/O	I/Oポート
D/A	D/A コンバータ
A/D	A/D コンバータ
INTC	割り込みコントローラ
BSC	BUSコントローラ

L (ユーザプログラムの実行) :

コマンドフォーマット

L [Enter]

機能 :

ホスト端末より、指定されたSタイプフォーマットのロードモジュールを
メモリ上にダウンロードします。

コマンド投入後、ホスト端末よりプログラムファイルを転送する。

M (メモリ内容の表示、変更) :

コマンドフォーマット

M <アドレス> 「; <サイズ>」 [Enter]

<アドレス>	表示変更を行うメモリの先頭アドレス
<サイズ>	表示、変更の単位の指定
B	: 1 バイト単位
W	; 2 バイト単位
L	; 4 バイト単位
省略時	; 1 バイト単位

機能 :

指定されたアドレスのメモリ内容を (サイズ) で指定した単位で表示、変更します。

[Enter] を入力すると次のメモリ内容を表示します。

. [Enter] を入力すると終了します。

操作例) FF131番地の 64を FFに置き換える場合。

```
:M FF12D [Enter]
FF12D 6F ? [Enter]
FF12E 79 ? [Enter]
FF12F 00 ? [Enter]
FF130 00 ? [Enter]
FF131 64 ? FF[Enter]
FF132 68 ? . [Enter]
:
```

R (レジスタの表示) :

コマンドフォーマット

R [Enter]

機能 :

C P Uのレジスタ一覧を表示します。

```
R [Enter]
PC=000000 CCR=FF: IUHUNZVC SP=00FFFF00
ER0=00000000 ER1=00000000 ER2=00000000 ER3=00000000
ER4=00000000 ER5=00000000 ER6=00000000 ER7=00FFFF00
```

S (シングルステップの実行) :

コマンドフォーマット

S 「<実行ステップ数>」 [Enter]

<実行ステップ数> 実行する命令数 (10進 2行で 表現)

機能 :

- (a) 現在のプログラムカウンタが値から指定された命令数分ユーザプログラムを実行します。
- (b) 実行ステップ数が省略されると 1 命令だけ実行します。
- (c) [Enter] のみの入力では前の S コマンドで実行した命令の次の命令から再び指定された命令数分、実行します。

H 8 ポートの設定

概要：

各入出力ポートは、動作するCPUモードにより機能が異なるため注意が必要である。
CPUモードに関わらずポート7は、入力専用ポート（アナログ出力は、2ch可能であるがデジタル出力は出来ない。）

CPUモードによりポートの機能が変則的に変わるため整理する事とした。
今回使用する、モード6、モード7に関して説明をする。

ポート1：

ポート1は、A7～A0のアドレス出力と兼用の8bit入出力ポートです。

	モード5，6	モード7
ポート1	P17（入力） / A7（出力）	P17（入力 / 出力）
	P16（入力） / A6（出力）	P16（入力 / 出力）
	P15（入力） / A5（出力）	P15（入力 / 出力）
	P14（入力） / A4（出力）	P14（入力 / 出力）
	P13（入力） / A3（出力）	P13（入力 / 出力）
	P12（入力） / A2（出力）	P12（入力 / 出力）
	P11（入力） / A1（出力）	P11（入力 / 出力）
	P10（入力） / A0（出力）	P10（入力 / 出力）

ポート1の設定用レジスタ構成

アドレス	名称	略称	R/W	初期値
FFC0	ポート1ディレクションレジスタ	P1DDR	W	00
FFC2	ポート1データレジスタ	P1DR	R/W	00

ポート1データディレクションレジスタ（P1DDR）の設定

モード5，6（内臓ROM有効拡張モード）

P1DDRに1をセットすると対応するポート1の端子は、アドレス出力となり、0をセットすると、入力ポートになります。

モード7（シングルチップモード）

ポート1は入出力ポートとして機能します。P1DDRに1をセットすると対応するポート1の端子は出力端子となり、0をセットすると入力端子となります。

実際のデータ入出力は、P1DRにて行います。

初期値は、00です。

ポート2：

ポート2は、A15～A8のアドレス出力と兼用の8bit入出力ポートです。

	モード5，6	モード7
ポート2	P27（入力） / A15（出力）	P27（入力 / 出力）
	P26（入力） / A14（出力）	P26（入力 / 出力）
	P25（入力） / A13（出力）	P25（入力 / 出力）
	P24（入力） / A12（出力）	P24（入力 / 出力）
	P23（入力） / A11（出力）	P23（入力 / 出力）
	P22（入力） / A10（出力）	P22（入力 / 出力）
	P21（入力） / A9（出力）	P21（入力 / 出力）
	P20（入力） / A8（出力）	P20（入力 / 出力）

H 8 ポートの設定

ポート 2 の 設定用レジスタ構成

アドレス	名 称	略称	R/W	初期値
FFC1	ポート 2 ディレクションレジスタ	P2DDR	W	00
FFC3	ポート 2 データレジスタ	P2DR	R/W	00
FFD8	ポート 2 プルアップMOSレジスタ	P2PCR	R/W	00

ポート 2 データディレクションレジスタ (P2DDR) の設定

モード 5 , 6 (内臓 ROM有効 拡張モード)

(初期値、入力ポート)

P1DDRに 1 をセットすると対応するポート 1 の端子は、アドレス出力となり、0 をセットすると、入力ポートになります。

モード 7 (シングルチップモード)

ポート 1 は入出力ポートとして機能します。 P1DDRに 1 をセットすると対応するポート 1 の端子は出力端子となり、0 をセットすると入力端子となります。

実際のデータ入出力は、P1DRにて行います。

初期値は、00です。

ポート 2 入力プルアップMOSコントロールレジスタ (P2PCR) の設定

P2DDRにて入力に設定したビットに、P2PCRの 対応するビットを 1 にするとMOSによるプルアップが行われます。

ポート 3 :

ポート 3 は、D15 ~ D8のデータバスと兼用の 8bit入出力ポートです。

	モード 6	モード 7
ポート 3	D15 (入出力)	P37 (入力 / 出力)
	D14 (入出力)	P36 (入力 / 出力)
	D13 (入出力)	P35 (入力 / 出力)
	D12 (入出力)	P34 (入力 / 出力)
	D11 (入出力)	P33 (入力 / 出力)
	D10 (入出力)	P32 (入力 / 出力)
	D9 (入出力)	P31 (入力 / 出力)
	D8 (入出力)	P30 (入力 / 出力)

ポート 3 の 設定用レジスタ構成

アドレス	名 称	略称	R/W	初期値
FFC4	ポート 3 ディレクションレジスタ	P3DDR	W	00
FFC6	ポート 3 データレジスタ	P3DR	R/W	00

ポート 2 データディレクションレジスタ (P2DDR) の設定

モード 5 , 6 (内臓 ROM有効 拡張モード)

ポート 3 はデータバスとして機能します。 P3DDRは無効です。

モード 7 (シングルチップモード)

ポート 3 は入出力ポートとして機能します。 P3DDRに 1 をセットすると対応するポート 1 の端子は出力端子となり、0 をセットすると入力端子となります。

H 8 ポートの設定

ポート 4 :

ポート 4 は、D7～D0のデータバスと兼用の 8bit入出力ポートです。
 ポート 4 の設定は、バスコントローラのバス幅コントロールレジスタ（ABWCR）の設定により D7～D0 の データバスになるか、I/Oポートとして使えるかが決まります。

	モード 6	モード 7
ポート 4	D7 (入出力)	P47 (入力 / 出力)
	D6 (入出力)	P46 (入力 / 出力)
	D5 (入出力)	P45 (入力 / 出力)
	D4 (入出力)	P44 (入力 / 出力)
	D3 (入出力)	P43 (入力 / 出力)
	D2 (入出力)	P42 (入力 / 出力)
	D1 (入出力)	P41 (入力 / 出力)
	D0 (入出力)	P40 (入力 / 出力)

バスコントローラのABWCRレジスタ（ アドレス：FFEC ）は、
 CPUモード 6 の場合 初期値（ FF ）
 CPUモード 7 の場合 初期値（ 00 ）
 で、モード 6 の場合、FFであれば、外部データバス幅は 上位（D15～D8）の 8 bit となり、ポート 4 は入出力ポートとして使用可能となる。
 ABWCRのどれか 1 つでも 0 が設定されている場合は、データバス幅は 16bitとなり
 ポート 4 は、データバスの D7～D0の下位 8bit入出力となる。

ポート 4 の 設定用レジスタ構成

アドレス	名 称	略称	R/W	初期値
FFC5	ポート 4 ディレクションレジスタ	P4DDR	W	00
FFC7	ポート 4 データレジスタ	P4DR	R/W	00
FFDA	ポート 4 ブルアップMOSレジスタ	P4PCR	R/W	00

ポート 2 データディレクションレジスタ（ P4DDR ）の設定

モード 5 , 6（ 内臓 ROM有効 拡張モード ）
 バス幅16bitの設定が行われている場合、ポート 4 はデータバスとして機能します。
 バス幅 8bitの設定が行われている場合、P4DDRにて、入出力の設定を行います。 1 が 出力、 0 が 入力です。

モード 7（ シングルチップモード ）
 ポート 4 は入出力ポートとして機能します。 P3DDRに 1 をセットすると対応するポート 1 の端子は出力端子となり、 0 をセットすると入力端子となります。

P4DRは、データポートです。

P4PCRは、入力ポートに設定されている場合に、MOSによるブルアップの設定を行います。
 1 で ブルアップ有り、 0 で ブルアップ無しです。

H 8 ポートの設定

ポート 5 :

ポート 5 は、アドレス出力 (A19 ~ A16) 兼用の 4bit の入出力ポートです。
 ポート 5 は、CPUモードにより端子機能が異なります。

	モード 6	モード 7
ポート 5	P53(入力) / A19(出力)	P53 (入力 / 出力)
	P52(入力) / A18(出力)	P52 (入力 / 出力)
	P51(入力) / A17(出力)	P51 (入力 / 出力)
	P50(入力) / A16(出力)	P50 (入力 / 出力)

ポート 5 の 設定用レジスタ構成

アドレス	名 称	略称	R/W	初期値
FFC8	ポート 5 ディレクションレジスタ	P5DDR	W	F0
FFCA	ポート 5 データレジスタ	P5DR	R/W	F0
FFDB	ポート 5 プルアップMOSレジスタ	P5PCR	R/W	F0

ポート 5 データディレクションレジスタ (P5DDR) の設定
 (設定は下位4bitが有効です。)

モード 5 , 6 (内臓 ROM有効 拡張モード)

ポート 5 はリセットされた直後は 入力ポートになっています。
 P5DDRを 1 にセットすると対応するポート 5 の端子がアドレス出力となり
 0をセットすると入力端子となります。

モード 7 (シングルチップモード)

ポート 5 は入出力ポートとして機能します。 P5DDRに 1 をセットすると
 対応するポート 1 の端子は出力端子となり、 0 をセットすると入力端子と
 なります。

P5DRは、データポートです。

P5PCRは、入力ポートに設定されている場合に、MOSによるプルアップの設定を行います。
 1 で プルアップ有り、 0 で プルアップ無しです。

ポート 6 :

ポート 6 は、バス制御入出力端子と兼用になった、7bitの入出力ポートです。

	モード 6	モード 7
ポート 6	LWR(出力)	P66 (入力 / 出力)
	HWR(出力)	P65 (入力 / 出力)
	RD(出力)	P64 (入力 / 出力)
	AS(出力)	P63 (入力 / 出力)
	P62(入出力) / BACK(出力)	P62 (入力 / 出力)
	P61(入出力) / BREQ(入力)	P61 (入力 / 出力)
	P60(入出力) / WAIT(入力)	P60 (入力 / 出力)

ポート 6 の 設定用レジスタ構成

アドレス	名 称	略称	R/W	初期値
FFC9	ポート 6 ディレクションレジスタ	P6DDR	W	80
FFCB	ポート 6 データレジスタ	P6DR	R/W	80

H 8 ポートの設定

ポート 6 データディレクションレジスタ (P6DDR) の設定
(設定は下位 7bit が有効です。)

モード 5 , 6 (内臓 ROM 有効 拡張モード)

ポート P66 ~ P63 は (バス制御出力端子 (LWR, HWR, RD, AS)) として機能します。 P62 ~ P60 は、入出力ポートとして機能します。
但し、P62 ~ P60 はバスコントローラの設定にて (BACK, BREQ, WAIT) に変わります。 以下のレジスタ設定が 影響します。

1. バスリリースコントロールレジスタ (BRCLR) の BRLE ビットの設定
(BRLE 初期値 : 0)
2. ウェイトステートコントローララインエーブルレジスタ (WCER) の設定
(WCR 初期値 : 11111111)

上記 2 つの設定が初期値のままであれば、P62 ~ P60 は入出力端子として使用可能となります。 P6DDR の b2 ~ b0 に 1 をセットすると対応する対応するポート端子は出力端子となり、0 をセットすると入力端子となります。

モード 7 (シングルチップモード)

ポート 6 は入出力ポートとして機能します。 P6DDR に 1 をセットすると対応するポートの端子は出力端子となり、0 をセットすると入力端子となります。 (最上位 bit は無効)

P6DR は、データポートです。

ポート 7 :

ポート 7 は、8bit の入力専用ポートです。 ポート 7 は、A/D 変換機のアナログ入力端子と、D/A 変換機のアナログ出力端子と兼用になっています。 これらの端子機能は、いずれの動作モードでも共通です。

全モード共通	
ポート 7	P77 (入力) / An7 (入力) / Ao1 (出力)
	P76 (入力) / An6 (入力) / Ao0 (出力)
	P75 (入力) / An5 (入力)
	P74 (入力) / An4 (入力)
	P73 (入力) / An3 (入力)
	P72 (入力) / An2 (入力)
	P71 (入力) / An1 (入力)
	P70 (入力) / An0 (入力)

ポート 7 の 設定用レジスタ構成

アドレス	名 称	略称	R/W	初期値
FFCE	ポート 7 データレジスタ	P7DR	R/W	不定

P7DR のリードを行うと常に端子の状態が読みだされます。

H 8 ポートの設定

ポート 8 :

ポート 8 は、5bit の入出力ポートです。ポート 8 は、CS3 ~ CS0 出力端子、RFSH 出力端子、IRQ3 ~ IRQ0 入力端子と兼用になっています。

	モード 6	モード 7
ポート 8	P84(入力) / CS0(出力)	P84(入力 / 出力)
	P83(入力) / CS1(出力) / IRQ3(入力)	P83(入力 / 出力) / IRQ3(入力)
	P82(入力) / CS2(出力) / IRQ2(入力)	P82(入力 / 出力) / IRQ2(入力)
	P81(入力) / CS3(出力) / IRQ1(入力)	P81(入力 / 出力) / IRQ1(入力)
	P80(入出力) / (出力)RFSH / IRQ0(入力)	P80(入力 / 出力) / IRQ0(入力)

ポート 8 の 設定用レジスタ構成

アドレス	名 称	略称	R/W	初期値
FFCD	ポート 8 データディレクション レジスタ	P8DDR	W	E0
FFCF	ポート 8 データレジスタ	P8DR	R/W	E0

ポート 8 データディレクションレジスタ (P8DDR) の設定
(設定は下位 5bit が有効です。)

モード 6 (内臓 ROM 有効 拡張モード)

ポート P84 ~ P81 は、P8DDR の対応するビットが 1 のとき CS0 ~ CS3 出力端子と 0 のとき入力ポートとなります。P80 は、リフレッシュコントローラをイネーブルにすると強制的に RFSH 出力となり、ディスエーブルのとき入出力となって P8DDR 設定値に従います。

モード 7 (シングルチップモード)

ポート 8 は入出力ポートとして機能します。P8DDR に 1 をセットすると対応するポートの端子は出力端子となり、0 をセットすると入力端子となります。

P8DR は、データポートです。(読み込んだ場合、上位 3bit は常に 1 です。)

ポート 9 :

ポート 9 は、6bit の入出力ポートです。ポート 9 は、シリアルコミュニケーションインタフェースチャネル 0, 1 (SCI0, 1) の入手 T 力端子 (TxD0、TxD1、RxD0、RxD1、SCK0、SCK1) IRQ5、IRQ4 端子と兼用になっています。

	全モード共通
ポート 9	P95(入力/出力) / SCK1(入力/出力) / IRQ5(入力)
	P94(入力/出力) / SCK0(入力/出力) / IRQ4(入力)
	P93(入力/出力) / RxD1(入力)
	P92(入力/出力) / RxD0(入力)
	P91(入力/出力) / TxD1(出力)
	P91(入力/出力) / TxD0(出力)

ポート 9 の 設定用レジスタ構成

アドレス	名 称	略称	R/W	初期値
FFD0	ポート 8 データディレクション レジスタ	P9DDR	W	C0
FFD2	ポート 8 データレジスタ	P9DR	R/W	C0

H 8 ポートの設定

ポート 9 データディレクションレジスタ (P9DDR) の設定
(設定は下位 5bit が有効です。)

機能 :

D9DDR は、データディレクションポートです。

ポート 9 は、P9DDR の対応するビットが 1 のとき 対応するポート 9 の各端子は
出力ポートとなり、0 にすると入力ポートとなります。

P8DR は、データポートです。(読み込んだ場合、上位 3bit は常に 1 です。)

シリアルインタフェースの設定により入出力端子機能がかわります。

P9 5 端子設定

CKE 1	C/A	CKE 0	P9DDR, 5	端子機能
0	0	0	0	入力
		1	1	出力
	1	-	-	SCK1 出力
1	-	-	-	SCK1 入力

P9 4 端子設定

CKE 1	C/A	CKE 0	P9DDR, 4	端子機能
0	0	0	0	入力
		1	1	出力
	1	-	-	SCK0 出力
1	-	-	-	SCK0 入力

P9 3 端子設定

RE	P9DDR, 3	端子機能
0	0	入力
	1	出力
1	-	RxD1 入力

P9 2 端子設定

SMIF E	RE	P9DDR, 2	端子機能
0	0	0	入力
	1	1	出力
1	-	-	RxD0 入力

P9 1 端子設定

TE	P9DDR, 1	端子機能
0	0	入力
	1	出力
1	-	TxD1 出力

P9 0 端子設定

SMIF E	TE	P9DDR, 3	端子機能
0	0	0	入力
	1	1	出力
1	-	-	TxD0 出力

H 8 ポートの設定

ポート A :

ポート A は、8bit の入出力ポートです。ポート A は、プログラマブルタイミングコントローラ (TPC) の出力端子 (TP7 ~ TP0)、16bit インテグレートドタイマユニット (ITU) の入出力端子 (TIOCA2, TIOCB2, TIOCA1, TIOCB1, TIOCA0, TIOCB0)、DMA コントローラ (DMAC) の出力端子 (TEND1, TEND0)、CS4 ~ CS6 出力端子及び、アドレスバス (A20 ~ A23) と兼用になっています。

ポート A は、動作モード 3、4、6 で、A20 が強制的に出力になるのを除き、リセット及びハードウェアスタンバイで入力ポートになっています。TPC、ITU 及び DMAC として端子を使用する場合はそれぞれの説明を参照して下さい。モード 3、4、6 で A23 ~ A21 を出力する場合はバスリリースコントロールレジスタを参照して下さい。モード 1 ~ 6 で CS4 ~ CS6 を出力する場合は、チップセレクト信号を参照して下さい。

モード 6	
ポート A	A20 (出力)
	PA6 (入力/出力) / TP6 (出力) / TIOCA2 (入力/出力) / A21 (出力) / CS4 (出力)
	PA5 (入力/出力) / TP5 (出力) / TIOCB1 (入力/出力) / A22 (出力) / CS5 (出力)
	PA4 (入力/出力) / TP6 (出力) / TIOCA1 (入力/出力) / A23 (出力) / CS6 (出力)
	PA3 (入力/出力) / TP6 (出力) / TIOCB0 (入力/出力) / TCLKD (入力)
	PA2 (入力/出力) / TP6 (出力) / TIOCA0 (入力/出力) / TCLKC (入力)
	PA1 (入力/出力) / TP6 (出力) / TEND1 (入力/出力) / TCLKB (入力)
	PA0 (入力/出力) / TP6 (出力) / TEND0 (入力/出力) / TCLKA (入力)

モード 7	
PA7 (入力/出力) / TP7 (出力) / TIOCB2 (入力/出力)	
PA6 (入力/出力) / TP6 (出力) / TIOCA2 (入力/出力)	
PA5 (入力/出力) / TP5 (出力) / TIOCB1 (入力/出力)	
PA4 (入力/出力) / TP6 (出力) / TIOCA1 (入力/出力)	
PA3 (入力/出力) / TP6 (出力) / TIOCB0 (入力/出力) / TCLKD (入力)	
PA2 (入力/出力) / TP6 (出力) / TIOCA0 (入力/出力) / TCLKC (入力)	
PA1 (入力/出力) / TP6 (出力) / TEND1 (入力/出力) / TCLKB (入力)	
PA0 (入力/出力) / TP6 (出力) / TEND0 (入力/出力) / TCLKA (入力)	

ポート A の 設定用レジスタ構成

アドレス	名 称	略称	R/W	初期値	
				Mode 6	Mode 7
FFD1	ポート 8 データレグション レジスタ	PADDR	W	80	00
FFD3	ポート 8 データレジスタ	PADR	R/W	00	

PADDR は、8bit のライト専用のレジスタで、ポート A 各端子の入出力をビット毎に指定する事ができます。

PADDR を 1 にセットすると対応するポート A の各端子は出力となり、0 にすると入力になります。但し、モード 3、4、6 では PA7 DDR は 1 に限定され、PA7 はアドレス出力として機能します。

ポート A データレジスタ (PADR) は、8bit のリード/ライト可能なレジスタで、ポート A の出力データを格納します。(PA7 は、アドレス出力固定)

H 8 ポートの設定

ポート B :

ポート B は、8bitの入出力ポートです。ポート B は、TPCの出力端子 (TP15 ~ TP8) ITU の入出力端子 (TIOCB4, TIOCB3, TIOCA4, TIOCA3) と出力端子 (TOCXB4, TOCXA4) D M A の入力端子 (DREQD1, REQ0)、A/D変換機のADTRG入力端子、CS7出力端子と兼用になっています。

ポート B は リセットおよびハードウェアスタンバイモードでは入力ポートになっています。

モード 6	
ポート B	PB7(入力/出力) / TP15(出力) / DREQ1(入力) / ADTRG(入力)
	PB6(入力/出力) / TP14(出力) / DREQ0(入力) / CS7(出力)
	PB5(入力/出力) / TP13(出力) / TOCXB4(出力)
	PB4(入力/出力) / TP12(出力) / TOCXA4(出力)
	PB3(入力/出力) / TP11(出力) / TIOCB4(出力)
	PB2(入力/出力) / TP10(出力) / TIOCA4(出力)
	PB1(入力/出力) / TP9(出力) / TIOCB3(出力)
	PB0(入力/出力) / TP8(出力) / TIOCA3(出力)

モード 7
PB7(入力/出力) / TP15(出力) / DREQ1(入力) / ADTRG(入力)
PB6(入力/出力) / TP14(出力) / DREQ0(入力)
PB5(入力/出力) / TP13(出力) / TOCXB4(出力)
PB4(入力/出力) / TP12(出力) / TOCXA4(出力)
PB3(入力/出力) / TP11(出力) / TIOCB4(出力)
PB2(入力/出力) / TP10(出力) / TIOCA4(出力)
PB1(入力/出力) / TP9(出力) / TIOCB3(出力)
PB0(入力/出力) / TP8(出力) / TIOCA3(出力)

ポート B の 設定用レジスタ構成

アドレス	名 称	略称	R/W	初期値
FFD4	ポート B データレジスタ	PBDDR	W	00
FFD6	ポート B データレジスタ	PBDR	R/W	00

PBDDRは、8bitのライト専用のレジスタで、ポート A 各端子の入出力をビット毎に指定する事ができます。

PBDDRを 1 にセットすると対応するポート A の各端子は出力となり、0 にすると入力になります。

ポート B データレジスタ (PBDR) は、8bitのリード / ライト可能なレジスタで、ポート A の出力データを格納します。