

実践的 PowerPC 活用テクニック



第9回 MPC5200B での割り込み制御方法

坂井 弘亮

今回は組み込みプログラミングで必須の割り込み処理について解説する。CPU 内蔵の割り込みコントローラの概要や、シリアル・コントローラからの割り込みとデクリメンタ例外を使った割り込みプログラミングについて解説する。
(編集部)

前回(2009年3月号, pp.156-162)はCPU内蔵のシリアル・コントローラを直接操作してターミナルと通信するプログラムを作成しました。今回は、組み込みでは必須といえる割り込みプログラミングについて説明します。

1 PowerPC の外部割り込み

PowerPCでは外部割り込み線は1本しかありません。このため割り込みの際には割り込みハンドラが呼ばれ、ハンドラ内部で割り込みコントローラの持つレジスタを参照して割り込み要因を調べ、それに応じたハンドラをソフトウェア的に呼び出す処理が必要になります。

● MPC5200B の割り込みコントローラ

MPC5200Bは、内部にSIU(System Integration Unit)と呼ばれるユニットを内蔵しています。ここに割り込みコントローラも内蔵しているため、SIUの持つ割り込みコントローラのレジスタを操作すれば、割り込み制御が行えます。割り込みコントローラに関しては参考文献(1)の「7.2 Interrupt Controller」を参照してください。

割り込みコントローラのレジスタ群は、MBAR + 0x0500 ~ 0x0544 のアドレスに配置されています^{注1, 2}。今回利用するのは、表1に示す二つのレジスタです。

ICTL Peripheral Interrupt Mask Registerは、割り込みマスクの設定を行います(図1)。特定のビットをクリアすることで、割り込みを有効化できます。PSC1^{注3}はビット1(C言語でビット指定する場合はビット30)、PSC2はビット2(同じくビット29)なので、0x9FFFFFF0を設定するこ

とでPSC1とPSC2の割り込みを有効化できます(下位8ビットは予約なのでゼロでよい)。これはu-bootではMPC5XXX_ICTL_PER_MASKとして定義されています。

ICTL Peripheral Interrupt Status All Registerは割り込み発生時に参照し、割り込み要因を取得するのに利用します(図2)。PSC1はビット11(C言語でビット指定する場合はビット20)、PSC2はビット12(同じくビット21)なので、このレジスタが0x00100000だったらPSC1、0x00080000だったらPSC2に割り込みが発生したことを示します。これはu-bootではMPC5XXX_ICTL_PER_STSとして定義されています。

なおPSCの割り込みは、受信バッファを参照すること(つまり、受信した文字の取り込み)で割り込みがクリアされるようです。このため割り込み処理の内部でICTL Peripheral Interrupt Status All Registerに対して、割り込みクリア操作をする必要はありません。

● デクリメンタ例外

PowerPCは簡単なタイマとして、デクリメンタ・レジスタ(DEC)を持っています。

DECレジスタは1クロックごとに減少していき、ゼロになったとき(正確には、最上位ビットが0→1に変化したとき)に割り込み(デクリメンタ例外)が発生します。これは簡単なタイマの実装に利用できます。

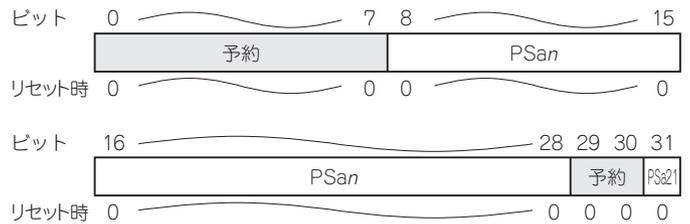
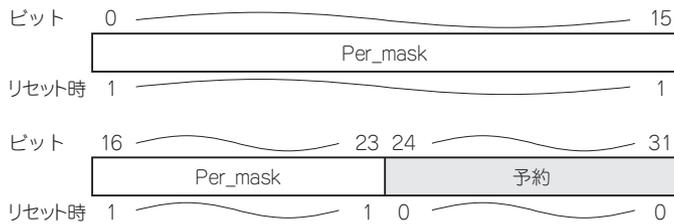
注1: MBARについては前回は参照。

注2: 参考文献(1)の「7.2.4 Interrupt Controller Registers」参照。

注3: MPC5200Bが内蔵するシリアル・コントローラ(Programmable Serial Controller)。詳しくは前回の記事を参照。

表1
割り込み関連レジスタ(一部)

名称	アドレス	レジスタ長(ビット)	意味
ICTL Peripheral Interrupt Mask Register	MBAR + 0x0500	32	割り込みマスクの設定
ICTL Peripheral Interrupt Status All Register	MBAR + 0x0530	32	割り込み要因の取得



ビット	名称	定義
0	Per_mask	BestComm interrupt source
1	Per_mask	Peripheral 1(PSC1)
2	Per_mask	Peripheral 2(PSC2)
3	Per_mask	Peripheral 3(PSC3)
4	Per_mask	Peripheral 4(PSC6)
5	Per_mask	Peripheral 5(Ethernet)
6	Per_mask	Peripheral 6(USB)
7	Per_mask	Peripheral 7(ATA)
8	Per_mask	Peripheral 8(PCI Control module)
9	Per_mask	Peripheral 9(PCI SC Initiator RX)
10	Per_mask	Peripheral 10(PCI SC Initiator TX)
11	Per_mask	Peripheral 11(PSC4)
12	Per_mask	Peripheral 12(PSC5)
13	Per_mask	Peripheral 13(SPI modf)
14	Per_mask	Peripheral 14(SPI spif)
15	Per_mask	Peripheral 15(I ² C1)
16	Per_mask	Peripheral 16(I ² C2)
17	Per_mask	Peripheral 17(CAN1)
18	Per_mask	Peripheral 18(CAN2)
19	Per_mask	予約
20	Per_mask	予約
21	Per_mask	Peripheral 21(XLB Arbiter)
22	Per_mask	Peripheral 22(BDLC)
23	Per_mask	Peripheral 23(BestComm LocalPlus)

図1 ICTL Peripheral Interrupt Mask Registerのフォーマット

外部割り込みを有効にすることで、デクリメンタ例外も同時に有効になります。今回はデクリメンタ割り込みも受け付けることで、簡単なタイマ動作の実験も行ってみます。

● 割り込みハンドラ

PowerPCでは、割り込みハンドラをゼロ番地から始まるアドレス〔このアドレスはMSR (Machine State Register) レジスタの設定で選択できる品種もある〕に、割り込み種別ごとに256バイト単位で配置します。ここには実行コードをそのまま置きます。つまりベクタ・アドレスではなく、ハンドラをそのまま置くということになります。

BLANCA (組み込み開発評価キットの通称) のPowerPC

注4：割り込みハンドラの延長で割り込み有効化される場合に必要な処理。このため今回のサンプル・プログラムでは不要ののだが、一応退避を行っている。

ビット	名称	定義
8	PSa23	Best Comm Local Plus
9	PSa22	BDLC
10	PSa0	BestComm interrupt source
11	PSa1	PSC1
12	PSa2	PSC2
13	PSa3	PSC3
14	PSa4	PSC6
15	PSa5	Ethernet
16	PSa6	USB
17	PSa7	ATA
18	PSa8	PCI Control module
19	PSa9	PCI SC Initiator Rx
20	PSa10	PCI SC Initiator Tx
21	PSa11	PSC4
22	PSa12	PSC5
23	PSa13	SPI modf
24	PSa14	SPI spif
25	PSa15	I ² C1
26	PSa16	I ² C2
27	PSa17	CAN1
28	PSa18	CAN2
31	PSa21	XLB Arbiter

図2 ICTL Peripheral Interrupt Status All Registerのフォーマット

オプションCPUカードでは、u-bootが割り込みハンドラを用意しています。今回はサンプル・プログラムで割り込みを受け取る必要があるため、独自のハンドラによって上書きします。

● 割り込みハンドラの処理

PowerPCでは、割り込み時にはプログラム・カウンタとMSRレジスタの値がSRR0とSRR1という特殊なレジスタに(自動的に)保存されます。割り込み復帰時〔割り込み復帰命令(rfi)実行時〕には、SRR0の内容がプログラム・カウンタへ、SRR1の内容がMSRにコピーされます。この動作によりプログラム・カウンタの値が復旧し、割り込み発生前の位置から実行が再開されます。

つまり、割り込み処理ではSRR0とSRR1の内容をスタックに退避しておき、復帰時にはスタックからSRR0とSRR1の内容を復旧し、rfi命令の実行で割り込み前の状態に復帰します^{注4}。SRR0とSRR1は直接操作することは