Interrupcions en el DSP TMS320c5x

Una avantatge d'un DSP enfront d'altres processadors, és l'habilitat que tenen aquests per rebre i donar servei a varies interrupcions molt ràpidament. Aquestes interrupcions poden venir de Timers, conversors A/D, coprocessadors, etc. Quan s'activa una interrupció s'executa una ISR (interrupt service routine) que emmagatzema el contingut dels registres crítics, processa el treball que li han demanat i finalment restaura el contingut dels registres. Pel que fa al DSP que a nosaltres ens interessa, TMS320c5x, disposa de les següents fonts d'interrupció:

- Reset
- Pins externs d'interrupció
- Perifèrics on-chip
- DMA on-chip
- Traps (interrupcions per soft)

L'usuari pot definir exactament com volem que es comporti el DSP enfront d'una interrupció, és qui ha de configurar i inicialitzar alguns registres i qui especifica on es troben localitzades a memòria. Tot això es pot fer, o bé amb llenguatge ensamblador o amb llenguatge d'alt nivell.

El control de les interrupcions del DSP, TMS320c5x es realitza a través de tres registres. El IMR, el IFR i el PMST, a més del bit INTM.
La funció de cada un d'aquests registres s'explica en els següents apartats.

El registre IMR es troba a l'adreça 4 de la memòria de dades i s'utilitza com a màscara per les interrupcions:
Registre IMR (Interrupt Mask Register):

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							**INT4**	**TXNT**	**TRNT**	**TXINT**	**RINT**	**TINT**	**INT3**	**INT2**	**INT1**

Un 1 en els bits de 8 a 0, indica l'habilitació de la interrupció corresponent, sempre i quan el bit INTM = 0. Observem que tant la interrupció RS com NMI no es troben en el registre IMR.

El registre IFR es troba a l'adreça 6 de la memòria de dades i pot llegir-se per identificar les interrupcions actives o escriure-hi per netejar-les.
Registre IFR (Interrupt Flag Register):

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							**INT4**	**TXNT**	**TRNT**	**TXINT**	**RINT**	**TINT**	**INT3**	**INT2**	**INT1**

El registre PMST es troba a l'adreça 7 de la memòria de dades i conté els camps que es mostren a continuació. El camp IPTR és el que afecta al tema d'interrupcions:
Registre PMST (Processor Mode Status Register):

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPTR					**0**	**0**	**0**	**AVIS**	**0**	**OVLY**	**RAM**	**MP/MC**	**NDX**	**TRM**	**BRAF**

Bit INTM:

S'Utilitza com a habilitació global de les interrupcions. Només es pot accedir a aquest bit mitjançant les instruccions SETC o CLRC. Es troba en el novè bit del Status Register 0.

```
	void main(void)
	{
		asm("\t CLRC \t INTM"); /* Enable interrupts globally */
		.
}
```

Interrupt Service Routine

Les ISR (interrupt service routines) tenen una nomenclatura especial que és la següent:

```
void c_intn(void)
{
      ...
      ...
}
```

encara que es poden renombrar per a una millor comprensió de la següent manera:

```
#define serialportreceiveISR  c_intn
void serialportreceiveISR (void)
{
      ...
      ...
}
```

En el cas d'un DSP de coma fixa, com el nostre, podem canviar *n* per un nombre de 0 a 9.
Dues coses a tenir en compte, quan implementem una ISR són: No es poden passar paràmetres a una ISR i una ISR no pot retornar mai cap valor. Això implica la utilització de variables globals dins d'aquestes rutines.

Activació dels vectors d'interrupció

Després de crear la ISR, aquesta s'ha de carregar correctament a memòria. Normalment, es troben a partir de l'adreça 0x0 però cada DSP pot variar la localització d'aquest vector. En el cas del nostre DSP, TMS320c5x, el reset sempre es troba a la posició 0x0 però la resta de vectors d'interrupció es poden col·locar a qualsevol 2k-word de la pàgina de memòria de programa. La posició exacte dins d'aquest espai de memòria s'especifica mitjançant els bits IPTR del registre PMST.

El registre PMST es troba a l'adreça 7 de la memòria de dades i conté els camps que es mostren a continuació:
Registre PMST (Processor Mode Status Register):

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPTR					**0**	**0**	**0**	**AVIS**	**0**	**OVLY**	**RAM**	**MP/MC**	**NDX**	**TRM**	**BRAF**

TMS320C5x	**Reset**	0x0	
	Interrupt Vectors	Any 2K-word page	Related to value of IPTR bits of the PMST register

La forma més directa de col·locar el vector d'interrupcions dins la posició exacta del mapa de memòria és mitjançant seccions en ensamblador (ASM). Per exemple, el següent tall de codi mostra com es faria amb el DSP TMS320c5x, mitjançant un mòdul ensamblador anomenat *vecs.asm:*

```
.ref _cint0, _c_int9  ;reference interrupt vectors defined externally
.sect "vectors"       ;declare a named section "vectors"

RS: b _c_int0         ;branch to reset vector
I1: b _c_int9         ;branch to interrupt vector 1
```

```
      I2: b _c_int9         ;branch to interrupt vector 2
```

Modificacions a l'hora de linkar

Un cop creada la nova secció vectors en ASM, l'haurem de linkar, amb les opcions de MEMORY i SECTIONS adequades. Per exemple, el següent fitxer "Command Linker File", .cmd mostra com es col·locaria la secció vectors a la posició 0x40.

```
–c
.
vecs.obj                        /* (1) Link in the vector table  */
main.obj
.
–l rts50.lib
MEMORY
{
      PAGE0:VECTORS:      origin = 0000h, length = 0003fh
                                /* (2) Declare mem for vectors   */
      ROM:                origin = 0040h, length = 007cfh
      .
}
SECTIONS
{
      ″vectors″:    {} > VECTORS /* (3) Place vector table        */
      .text:        {} > ROM
      .
}
```

En el cas d'utilitzar el bootloader (com fem nosaltres), el punt d'entrada al codi d'execució és l'adreça destí del primer word que hem enviat a través del on-chip bootloader. Per tant, quan utilitzem aquest mode no tenim ISR per **reset** i per contra, cada vegada que es fa un reset provoquem l'execució del bootloader.

Finalment, com a resum expliquem breument tot el procés:

1) Creació de les diferents ISR (Interrupt Service Routines).
2) Configuració dels registres IMR, IFR i PMST, i del bit INTM. (Interrupt Mask Register, Interrupt Flag Register, Processor Mode Status Register i el bit d'habilitació global d'interrupcions INTM).
3) Creació d'una nova secció amb ensamblador on hi posarem la taula amb els vectors d'interrupció.
4) Linkatge de tots els mòduls.
5) Generació del fitxer .hex que carregarem a la EPROM.

Quan es produeix una interrupció, s'activa un bit del IFR, i tot seguit es va a buscar a memòria (dins la taula amb els vectors d'interrupció) l'adreça on es troba la rutina de servei que s'ha d'executar. Utilitzant les funcions de la *runtime-support library* I$$AVE i I$$REST, es guarden o es restauren els registres que calgui.