Interruzioni, Display 7 Segmenti e Stack

08/04/2016

Ing. Marco Scialdone

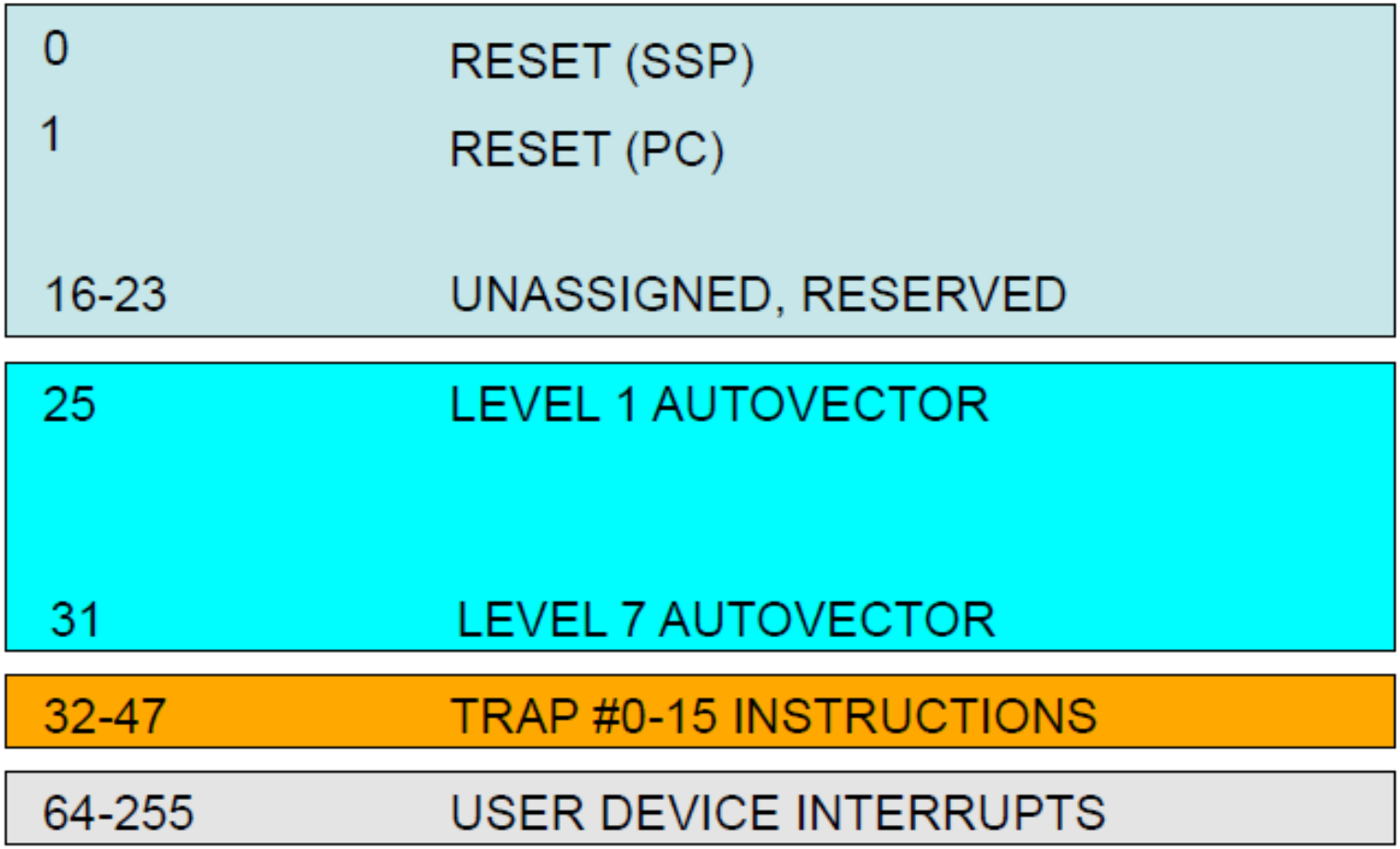
Gruppi di Eccezioni del 68000

- Il 68000 fornisce una divisione delle eccezioni in gruppi.
- Il gruppo 0 provoca una interruzione immediata dell'istruzione corrente.

Gruppo	*Tipo di eccezione*
0	Reset
	Errore sul Bus
	Errore di indirizzo
1	Trace
	Interrupt
	Istruzione illegale
	Violazione di privilegio
2	TRAP
	TRAPV
	CHK
	Divisione per zero

Nel caso di eccezioni multiple, il processore valuta i livelli di priorità, salva internamente la richiesta di interruzione avente priorità più bassa (*eccezione pendente*), esegue la procedura di gestione dell'eccezione avente priorità più alta.

Exception Vector Table

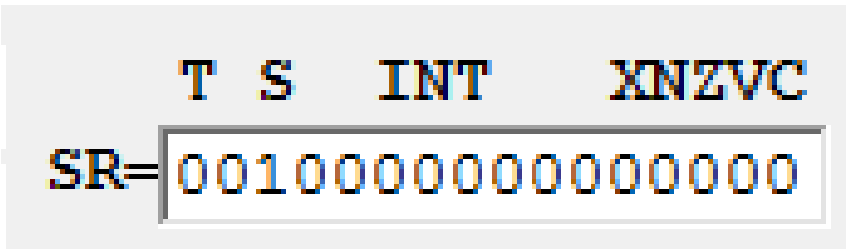


Interrupt autovettorizzate

- Si manda il segnale di Interrupt
- In base alla priorità del segnale di interruzione si accede ad una locazione di memoria prefissata contenente l'indirizzo della ISR
  - Dal vettore delle interruzioni si ricava quindi l'indirizzo di memoria della corretta ISR da eseguire

Vector Number

Vectors Numbers		Address			
Hex	Decimal	Dec	Hex	Space[6]	Assignment
0	0	0	000	SP	Reset: Initial SSP[2]
1	1	4	004	SP	Reset: Initial PC[2]
2	2	8	008	SD	Bus Error
3	3	12	00C	SD	Address Error
4	4	16	010	SD	Illegal Instruction
5	5	20	014	SD	Zero Divide
6	6	24	018	SD	CHK Instruction
7	7	28	01C	SD	TRAPV Instruction
8	8	32	020	SD	Privilege Violation
9	9	36	024	SD	Trace
A	10	40	028	SD	Line 1010 Emulator
B	11	44	02C	SD	Line 1111 Emulator
C	12[1]	48	030	SD	(Unassigned, Reserved)
D	13[1]	52	034	SD	(Unassigned, Reserved)
E	14	56	038	SD	Format Error[5]
F	15	60	03C	SD	Uninitialized Interrupt Vector
10–17	16–23[1]	64	040	SD	(Unassigned, Reserved)
		92	05C		—
18	24	96	060	SD	Spurious Interrupt[3]
19	25	100	064	SD	Level 1 Interrupt Autovector
1A	26	104	068	SD	Level 2 Interrupt Autovector
1B	27	108	06C	SD	Level 3 Interrupt Autovector
1C	28	112	070	SD	Level 4 Interrupt Autovector
1D	29	116	074	SD	Level 5 Interrupt Autovector
1E	30	120	078	SD	Level 6 Interrupt Autovector
1F	31	124	07C	SD	Level 7 Interrupt Autovector
20–2F	32–47	128	080	SD	TRAP Instruction Vectors[4]
		188	0BC		—
30–3F	48–63[1]	192	0C0	SD	(Unassigned, Reserved)
		255	0FF		—
40–FF	64–255	256	100	SD	User Interrupt Vectors
		1020	3FC		—



Interrupt Handler Address

- $64-Level 1 interrupt handler address 60+4
- $68-Level 2 interrupt handler address 60+8
- $6C-Level 3 interrupt handler address 60+C
- $70-Level 4 interrupt handler address 60+10
- $74-Level 5 interrupt handler address 60+14
- $78-Level 6 interrupt handler address 60+18
- $7C-Level 7 interrupt handler address 60+1C

$$\text{Address} = 60_{HEX}+(4*n)_{HEX}$$

Esempio Interruzioni

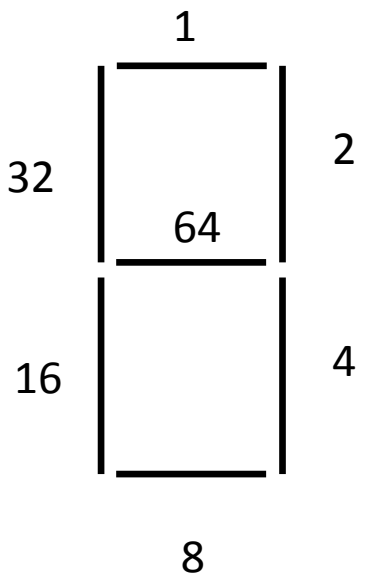
- Se vogliamo gestire l'interruzione di livello 1 dobbiamo:

```
        move.l      #irq1,$64
        ....

irq1    eori.b  #$02,(a1)   ; toggle LED
        rte
```

Carichiamo in memoria l'ISR da eseguire quando viene generato una interruzione di livello 1

Display sette segmenti



- L'indirizzo è: $E00000
- Per rappresentare una cifra bisogna sommare i valori relativi ai segmenti che si vogliono attivare.
  - Es. 3 = 1+2+4+8+64 = 79 = $(4F)_{HEX}$
- Supponiamo di voler rappresentare la cifra 3 alla 4 posizione da sinistra:

```
move.l #$E00000,A0      *carico l'indirizzo in A0
move.b #79,6(A0)        *l'indirizzo delle posizione della cifra discosta di 2
```

Esempio Display 7 Segmenti

- Conviene dichiarare tutte le cifre in questo modo: digits:

```
dc.b  $3F      *digit 0
dc.b  $06      *digit 1
dc.b  $5B      *digit 2
dc.b  $4F      *digit 3
dc.b  $66      *digit 4
dc.b  $6D      *digit 5
dc.b  $7D      *digit 6
dc.b  $07      *digit 7
dc.b  $7F      *digit 8
dc.b  $6F      *digit 9
dc.b  $00      *digit off
dc.b  $40      *dash
```

Esempio: visualizziamo 5 alla terza posizione da sinistra

```
lea      digits,a0
move.l   #$E00000,a1
move.b   #5,d3
move.b   #4,d2
move.b   (a0,d3),(a1,d2)
```

Stack: Esempio di utilizzo

```
move.l   A1,-(A7)
sub.l    #2,A7
jsr      sottprog
move.w   (A7)+,d4
add      #4,A7
```

```
sottprog
move.l   6(a7),d0
…
move.w   d0,4(a7)
rts
```