

Compito di Architettura dei Calcolatori - A.A. 2005-06
 Prova di esame del 17 febbraio 2006

COGNOME:

NOME:

MATRICOLA:

Istruzioni: Spiegare chiaramente TUTTE le assunzioni che vengono effettuate per chiarire eventuali punti che si ritengono ambigui o non specificati. Tempo assegnato per lo svolgimento: 90 minuti.

1) [8 punti] Disegnare gli schemi di una periferica di I/O e del suo modulo di controllo, spiegare come sono connessi tra loro e con le altre parti del sistema, descrivere e spiegare quali flussi di dati si scambiano.

SVOLGIMENTO:

Si veda il capitolo 6 del libro di testo e dei lucidi presentati a lezione (in particolare i lucidi 4,5,6,7).

2) [10 punti] Eseguire in binario, utilizzando obbligatoriamente la rappresentazione in complemento a 2 con 4 bit, le seguenti operazioni: A: 4+3; B:5+3; C: -2-4; D: -5-4; E: 5-3; F: 3-5; e discutere il risultato ottenuto. Inoltre, presentare e discutere, per ognuno dei casi, il valore dei *flag* aritmetici CF (*carry flag*) e OF (*overflow flag*)

SVOLGIMENTO:

A: $4d + 3d = 7d$

```

0100b +
0011b =
-----
0111b

```

CF=0, OF=0. Il risultato è correttamente rappresentato.

B: $5d + 3d = 8d$

```

0101b +
0011b =
-----
1000b

```

CF=0, OF=1. Come segnalato da OF=1 il risultato NON può essere rappresentato correttamente con la rappresentazione in complemento a 2 con 4 bit. Se però concateniamo CF con il risultato dell'addizione otteniamo 01000b che rappresenta correttamente 8d usando una rappresentazione in complemento a 2 con 5 bit.

C: $-2d + -4d = -6d$

```

1110b +
1100b =
-----
1010b

```

CF=1, OF=0. Il risultato è correttamente rappresentato. Il valore di CF non deve essere considerato, perché ha senso solo nella rappresentazione di numeri senza segno.

D: $-5d + -4d = -9d$

```

1011b +
1100b =
-----
0111b

```

CF=1, OF=1. Come segnalato da OF=1 il risultato NON può essere rappresentato correttamente con la rappresentazione in complemento a 2 con 4 bit. Se però concateniamo CF con il risultato dell'addizione otteniamo 10111b che rappresenta correttamente -9d usando una rappresentazione in complemento a 2 con 5 bit. Il valore di CF non deve essere considerato, perché ha senso solo nella rappresentazione di numeri senza segno.

E: $5d + -3d = 2d$

```

0101b +
1101b =
-----
0010b

```

CF=1, OF=0. Il risultato è correttamente rappresentato. Il valore di CF non deve essere considerato, perché ha senso solo nella rappresentazione di numeri senza segno.

F: $-5d + 3d = -2d$

```

1011b +
0011b =
-----
1110b

```

CF=0, OF=0. Il risultato è correttamente rappresentato.

3) [12 punti] A partire dallo schema della semplicissima CPU (VS0) sotto riportato disegnare il nuovo schema (utilizzando sempre una struttura interna della CPU basata su comunicazioni dirette e non su bus) per poter gestire un segnale di interruzione INT che arriva all'unità di controllo. Assumere di avere un registro SP (*stack pointer*) per la gestione di una zona di memoria con modalità "a pila" ed un registro GI (*gestione interruzioni*) per l'accesso in modalità indiretta alla routine per la gestione delle interruzioni. Descrivere inoltre, con riferimento a tale nuovo schema, il flusso dei dati tra le varie componenti della CPU dal momento dell'arrivo del segnale di interruzione al momento in cui si inizia ad eseguire la routine per la gestione delle interruzioni.

SVOLGIMENTO:

Il nuovo schema della CPU, con evidenziate le parti aggiunte, è disegnato più sotto. Il flusso dei dati è sostanzialmente quello descritto, in generale e non con specifico riferimento al nuovo schema della CPU, nei lucidi 34 e 35 del capitolo 11. Prima si salva il valore corrente di PC, all'indirizzo di memoria il cui valore è memorizzato in SP (incrementando successivamente SP). Poi si carica in PC l'indirizzo della prima istruzione della routine di gestione delle interruzioni. Tale indirizzo non è contenuto in GI ma è presente in memoria in una locazione specificata da GI (accesso in modalità indiretta). Il microprogramma per la gestione del segnale di interruzione INT è quindi il seguente:

t1	MAR ← SP	C17
	MBR ← PC	C18
t2	(memory) ← MBR	C0, C12, CW
	(SP) + 1	C19, CC, CA
t3	MAR ← GI	C20
	AC ← (ALU)	C9
t4	MBR ← (memory)	C0, C5, CR
	SP ← AC	C21
t5	PC ← MBR	C22

Successivamente si proseguirà normalmente con la fase di fetch che preleva la prima istruzione di tale routine.

