

**Calcolatori Elettronici – II parte (CdL Ingegneria Informatica)**  
**Esame del 22 settembre 2011 – tempo a disposizione: 1 ora e 30 minuti**

**Compito Num. 1** COGNOME:.....NOME:.....

1) (20%) Si vuole realizzare una CPU a 64 bit dotata di due registri *general purpose*, due coppie di registri per il trasferimento di dati e istruzioni da/per la memoria e in grado di eseguire 32 operazioni aritmetiche diverse. La CPU deve avere una architettura parallela, essendo dotata di due pipeline da cinque stadi di cui uno dedicato al pre-fetching delle istruzioni macchina e un altro allo scheduling delle operazioni macchina da eseguire in parallelo (entrambi visti come scatole nere). Tutte le altre specifiche possono essere liberamente scelte.

- A) Disegnare l'architettura generale (in particolare il data path) di tale CPU (comprensiva dei segnali di controllo) secondo i principi RISC e illustrare coincisamente il suo funzionamento.
- B) Descrivere, possibilmente in maniera grafica, cosa succede in cinque cicli di clock consecutivi nei vari stadi delle pipeline.
- C) Dire brevemente come è possibile sfruttare il parallelismo a disposizione della CPU progettata per migliorare le prestazioni della CPU.

2) (20%) Si assuma di avere un microprocessore che lavora a 2Ghz dotato di 10 registri ufficiali (da R0 a R9) e sufficienti registri segreti in grado di avviare 2 istruzioni per ciclo di clock. Tale sistema richiede 1 ciclo di clock per completare gli spostamenti tra registri, 2 cicli di clock per completare somme e differenze e 4 cicli di clock per completare divisioni e prodotti. Si consideri il seguente programma macchina:

(1)  $R2 = R4 * R1$ ; (2)  $R3 = R5$ ; (3)  $R8 = R6 + R3$ ; (4)  $R5 = R7 - R1$ ;

Indicare le istruzioni avviate e ritirate in ogni ciclo di clock (dall'inizio al termine dell'esecuzione del programma) secondo le seguenti strategie di esecuzione: (A) avvio e ritiro in ordine (senza registri segreti); (B) avvio fuori ordine e ritiro in ordine; (C) avvio e ritiro fuori ordine. Fornire per ciascuno dei casi una spiegazione di ciò che succede in ogni ciclo con riferimento ai vincoli tra le istruzioni.

3) (20%) Scrivere un programma in linguaggio assembleativo 8088 che, dato un vettore *vec* N di numeri interi memorizzato in memoria principale: (1) stampa la somma degli elementi del vettore mediante una subroutine *SUM* e (2) verifica, con una subroutine *EQU*, se il vettore N è identico ad un altro vettore M, anch'esso memorizzato in memoria principale. Entrambe le subroutine devono avere come unico argomento il vettore N.

**DOMANDE A RISPOSTA MULTIPLA (40%)** Indicare **in fondo al foglio** se le seguenti affermazioni sono vere (con una croce su [V]) o false (con una croce su [F]).

4) Con riferimento al seguente frammento programma di Assembler, indicare se le affermazioni sono vere o false:

```
.SECT .TEXT !1
    MOV CX,num-vec !2
    SHR CX,1 !3
    MOV BX,vec !4
    MOV SI,0 !5
    MOV AX,(num) !6
1:  CMP (BX)(SI),AX !7
    JG 2f !8
    ADD SI,2 !9
    LOOP 1b !10
    MOV DX,0 !11
    JMP 3f !12
2:  MOV DX,1 !13
3:  .
.SECT .DATA
    vec: .WORD 3,4,7,11,3
    num: .WORD 8
    format: .ASCII "%
```

- A) Il programma verifica che il vettore contenga il numero 8.
- B) Il programma termina dopo aver raggiunto l'ultimo elemento del vettore
- C) Il salto a riga 10 viene effettuato 5 volte
- D) L'istruzione 3 serve a dividere per 2 il contenuto di CX
- E) Se *vec* si trova all'indirizzo 15, *num* si trova all'indirizzo 25
- F) Durante l'esecuzione dell'istruzione 4 il registro CX contiene 5
- G) SI viene usato come registro indice
- H) Il programma stampa 1

5) Con riferimento alle tecniche di esecuzione di istruzioni in un CPU con pipeline, indicare se le seguenti affermazioni sono vere o false.

- A) Una predizione dinamica di salto richiede una memoria dedicata.
- B) La predizione dinamica di salti viene gestita dall'hardware.
- C) Nelle CPU moderne vengono caricate e decodificate diverse istruzioni nello stesso ciclo di clock.
- D) Una esecuzione fuori ordine delle istruzioni macchina è in genere più efficiente dell'esecuzione in ordine.
- E) In una esecuzione fuori ordine delle istruzioni macchina la CPU non va mai in stallo.
- F) I vincoli RAW possono essere risolti utilizzando registri segreti.
- G) In una esecuzione fuori ordine delle istruzioni macchina i vincoli WAW possono essere risolti utilizzando registri segreti.
- H) In una esecuzione speculativa alcune delle istruzioni avviate non vengono ritirate.

Risposte domanda 4 = A:[V][F] B:[V][F] C:[V][F] D:[V][F] E:[V][F] F:[V][F] G:[V][F] H:[V][F]

Risposte domanda 5 = A:[V][F] B:[V][F] C:[V][F] D:[V][F] E:[V][F] F:[V][F] G:[V][F] H:[V][F]

**Nota bene: continua sul retro del foglio**

### Compito Num. 1

- 6) Con riferimento ai meccanismi di gestione delle cache, indicare se le seguenti affermazioni sono vere o false.
- A) Il meccanismo di gestione *write through* è più efficiente del meccanismo *write back*.
  - B) È possibile indirizzare da programma macchina una slot di cache.
  - C) Una *cache hit* in scrittura richiede sempre l'accesso alla memoria.
  - D) La lunghezza del campo LINE di un indirizzo non dipende dalla dimensione della cache.
  - E) La politica di rimpiazzamento di linee LRU si applica a cache a mappatura diretta.
  - F) Il numero di collisioni su una slot di cache aumenta all'diminuire delle dimensioni del campo TAG.
  - G) Gli accessi a locazioni contigue della memoria possono risultare inefficienti a causa della presenza di una cache.
  - H) In una cache associativa a più vie una slot contiene una linea di cache.
- 7) Con riferimento ai linguaggi macchina indicare se le seguenti affermazioni sono vere o false.
- A) Nell'UltraSPARC sono disponibili 32 registri *general-purpose* ufficiali.
  - B) Nell'IA-32 sono disponibili 16 registri *general purpose* a 32 bit.
  - C) Con il meccanismo di espansione dei codici, il numero di istruzioni a  $n$  operandi dipende dal numero di istruzioni a  $n - 1$  operandi.
  - D) Nell'IA-32 una operazione aritmetica può indirizzare una locazione di memoria principale.
  - E) Una istruzione che adotta l'indirizzamento a stack non contiene alcun indirizzo.
  - F) Nell'indirizzamento diretto si specifica nell'istruzione l'indirizzo di memoria dell'operando.
  - G) Nell'UltraSPARC solo la LOAD e le STORE indirizzano la memoria.
  - H) L'indirizzamento indiretto a registro richiede più bit dell'indirizzamento diretto.

---

Risposte domanda 6 = A:[V][F] B:[V][F] C:[V][F] D:[V][F] E:[V][F] F:[V][F] G:[V][F] H:[V][F]

Risposte domanda 7 = A:[V][F] B:[V][F] C:[V][F] D:[V][F] E:[V][F] F:[V][F] G:[V][F] H:[V][F]