004133317526 030276124540 014426113420 021734510013 027145310015 002153407251 010426174342

Secondo Scritto Parziale di Architettura degli Elaboratori

A.A. 1998/99, 5-2-99

Cognome: .... Nome: .... Login: 199...s.........

Domande

1. (punti: -1, 4)
   Consideriamo la tecnica di arbitraggio "daisy chain" per un bus sincrono, così come descritta sulle dispense del corso. Supponiamo di connettere 4 dispositivi master chiamati nell'ordine M3, M2, M1 e M0:

   (a) ottiene sempre il bus il dispositivo master che lo ha chiesto per primo;

   (b) se M1 ed M3 chiedono contemporaneamente il bus (e nessun altro lo chiede) non si può dire a chi venga assegnato dall'arbitro perchè non abbiamo specificato le priorità relative tra M1 ed M3;

   (c) se M1 ed M3 chiedono contemporaneamente il bus (e nessun altro lo chiede) allora l'arbitro assegna prima il bus ad M3;

   (d) se M1 ed M3 chiedono contemporaneamente il bus allora il bus viene assegnato a M1 che può immediatamente cominciare ad usarlo;

   (e) se M1 ed M3 chiedono contemporaneamente il bus allora l'arbitro decide immediatamente a quale dei due può essere assegnato prima, senza bisogno di controllare se il bus é attualmente libero oppure usato da un terzo master;

   (f) nessuna delle affermazioni precedenti é corretta.

2. (punti: -1, 4)
   Nella microarchitettura VM-1 vista a lezione, il registro MAR:

   (a) serve per scambiare dati tra la CPU e la RAM;

   (b) serve per anagrammare gli indirizzi delle celle di RAM cui il programma deve accedere;

   (c) serve per memorizzare il campo indirizzo dell'istruzione prelevata dalla RAM durante l'ultima fase di fetch;

   (d) contiene l'indirizzo della istruzione che verrà prelevata dalla RAM nella fase di fetch successiva al completamento dell'istruzione in corso;

   (e) può essere usato per memorizzare il valore in uscita dall'ALU, quando questo sia interpretato come un indirizzo;

   (f) nessuna delle affermazioni precedenti é corretta.

3. (punti: -1, 4)
   L'introduzione di tecniche di caching per la realizzazione di una gerarchia di memoria :

   (a) può prevedere anche l'adozione della memoria virtuale con paginazione a richiesta al fine di usare unità a disco come memorie di grandi dimensioni e alta latenza;

   (b) richiede necessariamente l'adozione di una unità MMU;

   (c) richiede necessariamente l'adozione di un controllore di Cache di secondo livello;

(d) richiede necessariamente l'adozione di opportuni protocolli di consistenza per gestire correttamente le operazioni di scrittura;

(e) richiede l'adozione di un meccanismo di tipo "write-through" per il controllo della cache di primo livello, a meno che questa non sia partizionata in due sottoinsiemi disgiunti destinati rispettivamente al codice ed ai dati;

(f) nessuna delle affermazioni precedenti é corretta.

4. (punti: -1, 4)
Il sistema di gestione delle interruzioni di una macchina convenzionale:

(a) viene realizzato a livello fisico mediante la definizione di un insieme di fili di segnalazione a diversi livelli di priorità a cui corrisponde un vettore di interruzione che consente al microcodice di gestire più in fretta le interruzioni più importanti;

(b) viene realizzato a livello di microarchitettura in modo da poter interrompere in qualsiasi momento l'esecuzione di una istruzione per rispondere sempre in modo asincrono ed immediato a tutte le segnalazioni;

(c) viene realizzato a livello di macchina convenzionale eseguendo in modo interrompibile le apposite procedure di gestione delle attività dei dispositivi che possono richiedere interruzioni;

(d) viene realizzato agendo a più livelli di astrazione, e, tra l'altro, definendo una "maschera" a livello di microarchitettura che può rendere sensibile o meno la CPU a richieste di interruzione, anche a seguito dell'esecuzione di istruzioni di livello L2;

(e) viene realizzato agendo a più livelli di astrazione, e, tra l'altro, definendo un meccanismo fisico di segnalazione basato sull'adozione della connessione "wired-OR" per convogliare su un singolo filo richieste provenienti da più dispositivi (allo stesso livello di priorità);

(f) nessuna delle affermazioni precedenti é corretta.

5. (punti: -1, 4)
Un sistema di memoria virtuale può essere usato per garantire la sicurezza del sistema contro "attacchi" provenienti da virus:

(a) solo se si conosce il "tipo di virus" e come questo intende abusare della memoria di sistema;

(b) solo se a ciascun programma viene impedito di accedere a dati non propri mediante opportuna progettazione del sistema operativo e uso di apposite funzionalità hardware;

(c) solo se la macchina convenzionale non prevede istruzioni con indirizzamento fisico della RAM;

(d) purché la macchina convenzionale sia progettata in modo da non eseguire istruzioni con indirizzamento fisico della RAM in modalità di esecuzione "utente";

(e) solo se l'MMU segnala trap in caso di violazione dei segmenti e il vettore delle interruzioni contenente l'indicazione del gestore di trap non é modificabile dal programma stesso;

(f) nessuna delle affermazioni precedenti é corretta.

6. (punti: -1, 4)
Una macchina convenzionale con microarchitettura superscalare di livello K:

(a) esegue da un minimo di 2 ad un massimo di K istruzioni simultaneamente;

(b) esegue K istruzioni simultaneamente, consentendo quindi di eseguire fino a K programmi diversi, purchè questi siano tutti caricati in RAM;

(c) esegue la stessa istruzione su K registri simultaneamente, applicando quindi la stessa funzione di manipolazione a K dati diversi nello stesso ciclo di clock;

(d) esegue simultaneamente K istruzioni di indirizzo consecutivo, purchè queste non condividano dati di ingresso secondo la definizione di Bernstein e siano tutte diverse l'una dall'altra;

(e) esegue simultaneamente K istruzioni di indirizzo consecutivo, purchè ciascuna di queste riceva come dato di ingresso il dato in uscita della precedente;

(f) nessuna delle affermazioni precedenti é corretta.

7. Considerare un sistema di traduzione da indirizzamento virtuale a indirizzamento fisico realizzato mediante la tecnica di paginazione e segmentazione. Ogni indirizzo logico può far esplicitamente riferimento ad un massimo di 3 segmenti con il metodo della segmentazione "implicita". Ciascun segmento può essere costituito da un massimo di 4096 pagine, e ciascuna pagina é costituita da 1024 parole da 32 bit. L'indirizzamento logico é espresso a livello di byte. Il massimo numero di pagine fisiche coincide con il massimo numero di pagine logiche di un segmento.

(a) (punti: -1, 1)
Quanti bit dovranno essere usati per la rappresentazione degli indirizzi virtuali?

(b) (punti: -1, 1)
Quanti bit dovranno essere usati per rappresentare il numero di pagina fisica?

(c) (punti: 1)
Calcolare i valori $v1$ e $v2$ come specificato qui di seguito e rappresentarli in forma binaria sul numero di bit determinato nella risposta (a) per gli indirizzi virtuali:

$$v1 = 4096 + 1025 * C1 + 16 * N1$$

$$v2 = 64 * LL + 8$$

dove $C1$ e $N1$ rappresentano rispettivamente i numeri d'ordine della prima lettera del cognome e del nome (secondo l'alfabeto inglese) modulo 16, mentre LL rappresenta il numero su tre cifre decimali che conclude il nome del vostro login di laboratorio modulo 64. (Esempio: Giovanni Chiola, login 1977s313, $C1 = 3$, $N1 = 7$, $LL = 57$). Si consiglia di scrivere i valori $v1 =$ .... e $v2 =$ .... in notazione esadecimale.

(d) (punti: 3)

Data la seguente tabella delle pagine per un segmento:

n.pag.log.	n.pag.fis
0	2
1	1
2	0
3	4
4	5
5	8
6	7
7	6

Interpretando $v1$ e $v2$ determinati nella domanda (c) come indirizzi virtuali, tradurli nei corrispondenti indirizzi fisici $f1$ e $f2$ appartenenti a tale segmento. Si consiglia di scrivere i valori $f1 =$ .... e $f2 =$ .... in notazione esadecimale.

8. Considerare una microarchitettura organizzata in pipeline a 5 stadi, di cui i primi 2 stadi realizzano il fetch (indirizzando la memoria in due passi, come nell'esempio VM-1), il terzo realizza la decodifica, e gli ultimi due l'esecuzione. Tale microarchitettura viene fatta operare all frequenza di clock di 100 MHz.

   (a) (punti: 3)
   Calcolare il tempo di completamento e la velocità media di esecuzione delle istruzioni nel caso di un programma sequenziale (senza salti di programma) di 10 istruzioni, partendo dalla condizione di pipeline "vuota".
   Risposta: tempo di completamento = ... ns., velocità = ... MIPS

   (b) (punti: 6)
   La realizzazione delle istruzioni di salto (condizionali e non) viene completata nel quarto stadio della pipeline, eccetto che per le istruzioni di salto condizionale qualora la condizione sia valutata da una istruzione precedente di cui non é ancora terminata l'esecuzione (in tal caso il salto avviene nel ciclo immediatamente seguente al completamento dell'istruzione che cambia la condizione). Calcolare il tempo di completamento e la velocità media di esecuzione delle istruzioni del seguente programma VM-2 sempre partendo dalla condizione di pipeline vuota e nell'ipotesi che la microarchitettura non usi le tecniche del "delay slot" o di "branch prediction":

```
50: LOC8 1
51: RSHF
52: JZER 55
53: STOL 2
54: JUMP 57
55: LOC8 1
56: STOL 3
57: LOC8 2
```

   Risposta: tempo di completamento = ... ns., velocità = ... MIPS

9. Consideriamo una parte di memoria Cache di primo livello dedicata a contenere solo il codice del programma in esecuzione. Tale parte consta di 16KByte di memoria organizzata in linee da 2 parole di 32 bit ciascuna. Supponiamo che il processore usi indirizzi espressi su 25 bit per indirizzare le parole di memoria.

   (a) (punti: 3)
   Indicare il numero di bit del campo Tag per la ricerca delle linee in Cache al variare del livello di associatività:
   direct-mapping: .... bit, associatività 2: .... bit, associatività 4: .... bit

   (b) (punti: 3)
   Se la lettura di una parola dalla cache richiede 10 ns, e la lettura di una parola dalla RAM richiede 80 ns, di quanto aumenta la latenza del fetch nel caso la parola richiesta non sia presente in cache (miss) rispetto al caso in cui é già presente (hit)?
   Risposta: la latenza aumenta di .... ns.

10. Consideriamo la realizzazione VM-1 della macchina convenzionale VM-2, e supponiamo di modificare la definizione di quest'ultima, sostituendo una nuova istruzione chiamata "WHAT nn" alla "LOCH nn" (ossia riutilizzando lo stesso codice operativo 11110001 per specificare una serie di azioni diverse). Le nuove microistruzioni che realizzano "WHAT" sono allocate nelle celle della Control Store di indirizzi 96 e 97, e da 66 a 70, e sostituiscono quelle usate per la realizzazione di "LOCH". La nuova istruzione usa anche le microistruzioni "vecchie" agli indirizzi 42, 43 e 51. Riportiamo qui il formato binario di rappresentazione di tutte le microistruzioni (ad eccezione del campo "Int" che comunque "non viene mai usato"):

indir. in CS	A L U	C S	R n W	M A R	M B R	D mpx	A bus	B bus	CA bus	CA en	CD bus	CD en	D bus	m co nd	Addr
96	111	0	0	0	0	11	00	00	00	0	10	1	00	000	1111111
97	000	0	0	0	0	00	00	00	00	0	00	0	00	001	1000010
66	010	0	0	0	0	11	00	00	00	0	11	1	00	000	1000000
67	111	0	0	0	0	01	00	11	00	0	00	0	00	011	1000110
68	001	0	0	0	0	01	00	00	00	0	10	1	10	000	0000000
69	100	0	0	0	0	01	00	00	00	0	10	1	10	000	0000000
70	110	0	0	0	0	00	10	10	11	1	00	0	00	001	0101010
42	000	1	1	1	0	00	11	00	00	0	00	0	00	000	0000000
43	000	0	1	0	1	00	00	00	00	0	00	0	00	001	0110011
51	110	0	0	0	0	10	00	01	00	0	01	1	00	001	0000001

(a) (punti: 7)
Sapendo che i seguenti registri assumono i valori iniziali indicati, scrivere il valore assunto dagli stessi registri al termine dell'esecuzione del microcodice:
IR = 1111 0001 1000 1010
ACC = 13 (decimale)
B = 0
C = 0
FP = 4000 (decimale)
SP = 3900 (decimale)
ADR = 1111 1111 1111

(b) (punti: 2)
Quali modi di indirizzamento usa l'istruzione "WHAT" per individuare i suoi operandi?

(c) (punti: 2)
Quale codifica viene usata per specificare l'operando "nn" negli 8 bit meno significativi del codice operativo dell'istruzione?

(d) (punti: 3)
Usando la notazione solita delle dispense per le istruzioni VM-2 (esempio, per "PUSH" la notazione "M[SP]←ACC; SP←SP−1"), scrivere qual é l'effetto dell'istruzione "WHAT nn" a livello di macchine convenzionale. Se necessario aggiungere anche commenti a parole.

11. (punti: 10) Supponiamo di voler realizzare in microcodice una istruzione "NORD X" (di livello 2) il cui effetto sia di inserire in ACC il risultato della funzione NOR tra ACC e RAM[X] (indirizzamento diretto). Completare le microistruzioni e/o i commenti:

indir. in CS	ALU	CS	RnW	MAR	MBR	D mpx	A bus	B bus	CA bus	CA en	CD bus	CD en	D bus	m cond	Addr
k	001	0	0	0	0	10	00	00	00	0	10	1	00	000	0000000
	commento: ...														
k+1															
	commento: ACC ← not ACC														
k+2															
	commento: ...														

Di nuovo abbiamo omesso il campo "Int" del MIR. Eventualmente lasciare in bianco i campi il cui valore é irrilevante (quelli che nelle dispense contengono "XXX").

004133317526 030276124540 014426113420 021734510013 027145310015 002153407251 010426174342

Risposte secondo scritto parziale di Architettura degli Elaboratori, 5/2/1999

Cognome : **Nome**: **Matricola**:

1. (punti: -1, 4) **a** **b** **X c** **d** **X e** **f**
2. (punti: -1, 4) **a** **b** **c** **d** **e** **X f**
3. (punti: -1, 4) **X a** **b** **c** **X d** **e** **f**
4. (punti: -1, 4) **a** **b** **c** **X d** **X e** **f**
5. (punti: -1, 4) **a** **X b** **c** **d** **X e** **f**
6. (punti: -1, 4) **a** **b** **c** **d** **e** **X f**
7.
   (a) (punti: -1, 1) Rappr. ind. virt. = 24 bit
   (b) (punti: -1, 1) Rappr. numero pag. fis. = 12 bit
   (c) (punti: 1)
   Esempio: $C1 = 3$, $N1 = 7$, $LL = 57$, $v1 = 1C73$, $v2 = E48$
   (d) (punti: 3) Esempio: $f1 = 001.C73$, $f2 = 002.E48$
8.
   (a) (punti: 3) Tempo di completamento = 140 ns., Velocità = 71.43 MIPS
   (b) (punti: 6) Tempo di completamento = 140 ns., Velocità = 42.86 MIPS
9.
   (a) (punti: 3)
   direct-mapping: 13 bit, associatività 2: 14 bit, associatività 4: 15 bit
   (b) (punti: 3) La latenza aumenta di 160 ns.
10.
   (a) (punti: 7)
   IR = 1111 0001 1000 1010
   ACC = RAM[ADR]+13 (decimale)
   B = FFF6 (esadecimale)
   C = 0080 (esadecimale)
   FP = 4000 (decimale)
   SP = 3900 (decimale)
   ADR = F96 (esadecimale)
   (b) (punti: 2) Registro e indicizzato rispetto ad FP
   (c) (punti: 2) Numeri relativi in modulo e segno
   (d) (punti: 3) ACC←ACC+M[FP+nn]

11. (punti: 10)

indir. in CS	ALU	CS	RnW	MAR	MBR	D mpx	A bus	B bus	CA bus	CA en	CD bus	CD en	D bus	m cond	Addr
k	001	0	0	0	0	10	00	00	00	0	10	1	00	000	0000000
	commento: B ← not MBR														
k+1	001	0	0	0	0	01	00	00	00	0	01	1	01	000	0000000
	commento: ACC ← not ACC														
k+2	111	0	0	0	0	01	00	01	00	0	01	1	10	001	0000001
	commento: ACC ← ACC and B ; goto 1														