Secondo Scritto Parziale di Architettura degli Elaboratori

A.A. 1997/98 28-1-98

Cognome :

Nome :

Login :

Regole Generali

Questo compito é composto da 2 diversi tipi di domande: alcune a risposte multiple, altre in cui dovete dare voi la risposta corretta mediante lo svolgimento di un semplice esercizio. Nel secondo tipo di domande si richiede di riportare solo il risultato finale.
Le risposte vanno riportate sul MODULO DI RISPOSTA distribuito a parte. Si consiglia comunque di riportare le risposte anche all'interno del testo in modo che il candidato possa tenere una copia dopo la consegna per la verifica delle risposte ed il calcolo del punteggio. Il calcolo dei punteggi relativi ai due tipi di domande é spiegato più avanti.
NON USARE FOGLI ADDIZIONALI né per la consegna del compito né per la "brutta"! E' ammesso utilizzare il rovescio dei fogli del testo come "brutta copia" (che in ogni caso non sarà presa in considerazione per la correzione).
Usare solo penne e NON MATITE per la compilazione delle risposte, in quanto NON VERRANNO ACCETTATE CONTESTAZIONI DI NESSUN TIPO riguardo eventuali correzioni o cancellazioni (comprese quelle effettuate col bianchetto).
Non é ammessa nessuna forma di comunicazione tra i candidati. Non sono ammessi "prestiti" di materiale di nessun tipo tra i candidati.
E' possibile assentarsi dall'aula solo previo deposito del testo del compito presso il docente "di guardia". Anche in caso di ritiro anticipato dal compito il testo dell'esercizio NON PUÒ essere portato fuori dall'aula prima dello scadere del termine di consegna.

Compilazione delle risposte

Si consiglia di scrivere le risposte inizialmente sul testo e di utilizzare gli ultimi 5–10 minuti per ricopiare le risposte sull'apposito foglio da consegnare.

Domande a risposte multiple

Rispondere segnando sul testo con una crocetta le risposte che ritenete corrette e lasciando in bianco quelle che ritenete errate. Ad ogni domanda corrisponde almeno una risposta corretta (ci possono essere più risposte corrette per una domanda).
Domande di questo tipo hanno associati due valori di punteggio: uno negativo e uno positivo. Ogni risposta errata segnata con una crocetta comporta la detrazione del valore negativo dal punteggio complessivo accumulato. Il valore positivo corrisponde al punteggio globalmente assegnato a tutte le risposte corrette ad una domanda.
Per esempio, se una domanda etichettata coi valori $(-1, 4)$ prevede 4 risposte possibili di cui **a. d.** errate e **b. c.** corrette, allora se il candidato segna le risposte **b. c.** e non segna le risposte **a. d.** ottiene il massimo punteggio 4, mentre se segna le risposte **a. b.** e non **c. d.** allora ottiene il punteggio $-1+2=1$; nel caso segnasse tutte le risposte otterrebbe in questo esempio il punteggio $-1+2+2-1=2$. Notare

che in generale, CONTRARIAMENTE a quanto succede nell'esempio, la somma algebrica dei punteggi ottenuti segnando tutte le risposte é ≤ 0!

Esercizi

In questo caso é il candidato che deve calcolare e scrivere la risposta finale. Ogni domanda é accompagnata da un solo valore positivo che indica il punteggio massimo assegnato per la risposta corretta a tutti i quesiti. In caso di risposte parziali o parzialmente corrette il punteggio viene ridotto in proporzionalità diretta rispetto al numero di risposte corrette (con troncamento al più grande numero intero minore o uguale nel caso di risultato frazionario).

Tattica suggerita

Come nel caso di altri test da completarsi in tempi predefiniti, risulta normalmente conveniente completare il compito in due passate: in una prima fase dedicare una quantità predefinita e limitata di tempo alla lettura di tutte le domande, svolgendo solo gli esercizi più semplici. In un secondo passo (dopo aver risposto rapidamente a tutte le domande semplici) tornate a considerare quelle più difficili in ordine decrescente di punteggio. In questo modo riuscirete a dedicare la maggior quantità di tempo alla soluzione degli esercizi difficili ma più redditizi. Per le domande a risposte multiple con punteggi negativi evitate di segnare risposte di cui non siete assolutamente certi in quanto i valori dei punteggi sono studiati per penalizzare i "tentativi a caso". Per gli esercizi senza punteggi negativi invece le risposte sbagliate non vi penalizzeranno.

Calcolo del punteggio

Il voto del secondo scritto parziale espresso in quindicesimi sarà calcolato applicando la seguente formula, nella quale s rappresenta la somma dei punti totalizzati per le varie domande:

$$\text{Voto} = 6.1666667 + s \times 0.5 - \max(s, 25) \times 0.25 - \max(s, 35) \times 0.0833333$$

Il voto complessivo in trentesimi sarà ottenuto sommando i voti in quindicesimi del primo e del secondo scritto parziale e arrotondando il risultato (a patto che entrambi gli scritti parziali raggiungano almeno il voto di 8/15, ovvero totalizzando un punteggio $s \geq 22$). Voti inferiori a 18/30 verranno considerati insufficienti, e comporteranno la necessità di superare la prova scritta completa in uno dei 4 appelli annuali d'esame. Il voto finale verrà poi stabilito a partire dal voto dello scritto con un riaggiustamento di tale valore fino ad una variazione massima di 4/30 in più o in meno, a seconda dell'esito dell'orale.

Domande

1. (punti: -1, 4)
   La parte di controllo della microarchitettura VM-1 vista a lezione:

   (a) é di tipo "hardwired" perché progettata e realizzata come un circuito sequenziale sincrono;

   (b) é di tipo "microprogrammato" perché progettata e realizzata come una macchina di Von Neumann (semplificata dal fatto di non prevedere ingresso e/o uscita);

   (c) non é di tipo "microprogrammato" perché la Control Store é realizzata come una memoria a sola lettura (ROM), e quindi non permette all'utente di cambiare il microcodice;

   (d) é organizzata in modo "orizzontale" perché il registro MIR ha più bit del registro MPC;

   (e) é organizzata in modo "verticale" perché il registro MIR codifica la connessione dei registri ai bus interni senza utilizzare un bit per l'abilitazione individuale di ciascun registro;

   (f) nessuna delle affermazioni precedenti é corretta.

2. (punti: -1, 4)
   Considerare un insieme di istruzioni di tipo Load/Store di una macchina convenzionale composto da: 2 istruzioni Load e Store a 2 operandi, 8 istruzioni a 2 operandi, 15 istruzioni a 3 operandi, 7 istruzioni di salto in grado di indirizzare qualsiasi locazione di memoria:

   (a) Se il processore é dotato di 64 registri, la memoria viene indirizzata per parole di 3 byte usando un indirizzamento logico su 20 bit, e se si usa l'indirizzamento diretto per accedere alla memoria in fase di esecuzione, allora ogni istruzione può essere codificata insieme a tutti i suoi operandi in una singola parola di memoria;

   (b) Se il processore é dotato di 16 registri, la memoria viene indirizzata per parole di 3 byte usando un indirizzamento logico su 18 bit, e se si usa l'indirizzamento diretto per accedere alla memoria in fase di esecuzione, allora ogni istruzione può essere codificata insieme a tutti i suoi operandi in una singola parola di memoria;

   (c) Se il processore é dotato di 64 registri, la memoria viene indirizzata per parole di 3 byte usando un indirizzamento logico su 20 bit, e se si usa l'indirizzamento indicizzato per accedere allo stack in fase di esecuzione limitatamente ad un intervallo di +/- 32K parole rispetto al contenuto del registro FP, allora ogni istruzione può essere codificata insieme a tutti i suoi operandi in una singola parola di memoria;

   (d) Se il processore é dotato di 16 registri, la memoria viene indirizzata per parole di 3 byte usando un indirizzamento logico su 22 bit, e se si usa l'indirizzamento indiretto mediante registro per accedere alla memoria in fase di esecuzione, allora ogni istruzione può essere codificata insieme a tutti i suoi operandi in una singola parola di memoria;

   (e) Se il processore é dotato di 64 registri, allora la codifica delle istruzioni diverse da load e store richiede 22 bit;

   (f) nessuna delle affermazioni precedenti é corretta.

3. (punti: -1, 4)
   La definizione di macchine convenzionali con esecuzione delle istruzioni in "pipeline":

   (a) permette di aumentare la frequenza di completamento delle istruzioni di una macchina convenzionale fino ad un massimo teorico pari alla frequenza di clock del processore, anche se le singole istruzioni richiedono più di un ciclo di clock per il loro trattamento;

   (b) permette di ridurre il tempo di completamento delle istruzioni di una macchina convenzionale fino ad un massimo prestabilito e pari al livello di superscalarità (di solito limitato a valori 2, 3, 4);

   (c) richiede l'introduzione del concetto di "delay slot" per poter realizzare correttamente le istruzioni di salto di programma;

   (d) richiede l'individuazione di condizioni di indipendenza tra istruzioni consecutive, del tipo di quelle proposte da Bernstein per tener conto della condivisione di registri.

   (e) richiede la decomposizione di ogni singola istruzione in un numero prefissato di "stadi', soggetti al vincolo di richiedere un tempo di completamento il più uniforme possibile per raggiungere un elevato grado di efficienza e giustificarne quindi l'introduzione;

   (f) nessuna delle affermazioni precedenti é corretta.

4. (punti: -1, 4)
Un bus di tipo sincrono:

(a) a differenza di un bus asincrono non richiede arbitrazione in quanto sia il Master che lo Slave si sincronizzano rispetto ad un unico segnale di Clock globale;

(b) a differenza di un bus asincrono non richiede l'uso di dispositivi a tre stati di uscita per la connessione dei fili "dati";

(c) viene normalmente realizzato mediante un protocollo sincrono di "handshaking" a due variabili;

(d) richiede la definizione di un protocollo che specifica quali azioni sono consentite al Master ed allo Slave in diverse fasi di tempo, scandite dal segnale di Clock;

(e) consente comunque la segnalazione di interruzioni asincrone mediante l'adozione di una connessione detta "wired or" che permette a più dispositivi di mandare la loro richiesta di interruzione indipendentemente sugli stessi fili;

(f) nessuna delle affermazioni precedenti é corretta.

5. (punti: -1, 4)
Una memoria cache da 256 KByte inserita tra la CPU ed il dispositivo MMU in un sistema caratterizzato da un indirizzamento della memoria a livello di singoli Byte con indirizzi virtuali espressi su 30 bit:

(a) richiede la conoscenza della dimensione della RAM per dimensionare il campo "tag";

(b) richiede l'uso di 12 bit per il campo "tag";

(c) richiede 13 bit per il campo "tag" se organizzata con livello di associatività 2;

(d) deve interrogare l'unità MMU prima di poter decidere se l'indirizzo richiesto dall CPU genera "hit" oppure "miss" nella cache;

(e) interagisce con l'unità MMU per poter accedere alla RAM in caso "miss";

(f) nessuna delle affermazioni precedenti é corretta.

6. (punti: -1, 4)
Un sistema per la protezione dall'esecuzione di programmi contenenti errori o "virus" basato sul meccanismo della memoria virtuale a segmentazione e della definizione di due modi diversi di esecuzione delle istruzioni da parte del processore (modo "utente" e modo "sistema"):

(a) richiede la definizione di istruzioni "privilegiate" eseguibili solo in modo sistema ed in grado di indirizzare la memoria con indirizzo fisico anziché logico;

(b) richiede la definizione di istruzioni "privilegiate" eseguibili solo in modo utente ed in grado di indirizzare la memoria con indirizzo logico anziché fisico;

(c) richiede l'uso di Trap per passare dal modo sistema al modo utente, mentre il passaggio inverso può essere effettuato senza problemi limitandosi a settare il bit di mascheramento delle interruzioni nel registro di stato;

(d) richiede l'uso di Trap per passare dal modo utente al modo sistema senza compromettere la sicurezza, mentre il passaggio inverso può essere effettuato senza problemi eseguendo una istruzione privilegiata di cambiamento di modo di esecuzione;

(e) richiede l'uso di Interrupt per passare dal modo utente al modo sistema, mentre il passaggio inverso può essere effettuato senza problemi limitandosi a settare il bit di mascheramento delle interruzioni nel registro di stato;

(f) nessuna delle affermazioni precedenti é corretta.

7. (punti: 5)
Scrivere una sequenza di istruzioni (possibilmente di lunghezza minima) in assembler SPARC per caricare nel registro G3 il contenuto della parola di memoria (32 bit) di indirizzo virtuale 4294967188:

8.
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 4 segmenti. Ciascun segmento può essere costituito da un massimo di 256 pagine. Ciascuna pagina é costituita da 2048 parole di 64 bit. L'indirizzamento é espresso a livello di byte. La memoria fisica viene indirizzata mediante 24 bit.

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

(b) (punti: -1, 1)
Quante pagine fisiche potrà contenere al massimo la memoria RAM?

(c) (punti: 2)
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:

$$v1 = 2050 * C1 + 64 * N1$$

$$v2 = 256 * LL$$

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 128. (Esempio: Giovanni Chiola, login 1977s313, $C1 = 3$, $N1 = 7$, $LL = 57$).

(d) (punti: 4)

Date le seguenti tabelle di traduzione:

tab. segmenti

n. segm	n. pag.	tab. pag.
0	4	tab0
1	3	tab1
2	1	tab2
3	2	tab3

tab0

n.pag.log.	n.pag.fis
0	3
1	5
2	29
3	4

tab1

n.pag.log.	n.pag.fis
0	7
1	1
2	6

tab2

n.pag.log.	n.pag.fis
0	11

tab3

n.pag.log.	n.pag.fis
0	2
1	27

Interpretando $v1$ e $v2$ determinati nella domanda (c) come indirizzi virtuali, tradurli nei corrispondenti indirizzi fisici $f1$ e $f2$.

9. (punti: 20)
Considerare la seguente procedura scritta in Assembler SPARC dalla quale sono state cancellate 5 istruzioni. Aggiungere le 5 istruzioni mancanti nei punti individuati dalle etichette MANCA1: ... MANCA5:

```
ppp:    save %sp,-112,%sp
        mov %i0,%o0
        mov %i1,%o4
        ldub [%o0],%i1
        cmp %i1,0
        be MANCA5
        mov 0,%o7
        mov 1,%g4
        sethi %hi(co),%g2
        or %g2,%lo(co),%g1
MANCA1:
        or %g2,%lo(nn),%o1
        add %o1,1,%o3
        sethi %hi(cc),%g2
        or %g2,%lo(cc),%o2
        mov %i1,%g3
.LL27:  ldub [%o0+%g4],%i0
        cmp %i0,0
        be MANCA5
MANCA2:
        ldub [%g1+2],%g2
        cmp %g2,0
        be .LL9
        mov 0,%i4
        mov %g3,%i3
        mov %i0,%i2
        sethi %hi(cc+1),%g2
        or %g2,%lo(cc+1),%i5
        mov %g1,%i0
.LL11:  ldub [%i0+1],%g2
        cmp %g2,0
        bne .LL12
        ldub [%i0],%g3
        ldub [%g3+%o1],%g2
        cmp %g2,%i3
        bne,a .LL26
        add %i0,4,%i0
        b .LL25
        ldub [%g3+%o3],%g2
.LL12:  ldub [%g3+%o2],%g2
        cmp %g2,%i3
        bne,a .LL26
        add %i0,4,%i0
        ldub [%g3+%i5],%g2
.LL25:  cmp %g2,%i2
```

```
        be .LL14
        add %i0,4,%i0
.LL26:  ldub [%i0+2],%g2
        cmp %g2,0
        bne .LL11
        add %i4,1,%i4
.LL9:   sll %i4,2,%g3
        add %g4,-1,%g2
        stb %g2,[%g3+%g1]
        add %g3,%g1,%g3
        stb %o4,[%g3+1]
        mov 2,%g2
        stb %g2,[%g3+2]
        mov 1,%g2
        stb %g2,[%g3+3]
        add %i4,1,%g2
        sll %g2,2,%g2
        add %g2,%g1,%g2
        stb %g0,[%g2+3]
        b .LL3
        add %o7,1,%o7
.LL14:  sll %i4,2,%g3
        add %g3,%g1,%g3
        ldub [%g3+3],%g2
        addcc %i4,-1,%i3
        add %g2,1,%g2
MANCA3:
        bneg .LL3
        mov %g2,%g3
        mov %g3,i5
        sll %i3,2,%g2
        add %g2,%g1,%i2
.LL21:  ldub [%i2+3],%g2
        cmp %g2,%i5
        bgeu .LL3
        sll %i4,2,%g2
        ld [%i2],%i0
        mov %i3,%i4
        ld [%g2+%g1],%g3
        addcc %i3,-1,%i3
MANCA4:
        st %i0,[%g2+%g1]
        bpos .LL21
        add %i2,-4,%i2
.LL3:   add %g4,1,%g4
        cmp %g4,4
        ble .LL27
        mov %i1,%g3
MANCA5:
        restore %g0,%o7,%o0
```

10. (punti: 12)
Considerare la procedura `ppp` definita in Assembler SPARC nell'esercizio precedente. Supponiamo che venga richiamata una prima volta dopo aver caricato nel registro `%o1` il valore 0 e nel registro `%o0` l'indirizzo del primo elemento del vettore di bytes `nn`, nei quali sono codificati in codice ASCII le prime 5 lettere del vostro nome senza spaziature e con caratteri tutti minuscoli (se il vostro nome contiene meno di 5 lettere, i byte non utilizzati conterranno il valore 0). Supponiamo che venga poi richiamata una seconda volta dopo aver caricato nel registro `%o1` il valore 1 e nel registro `%o0` l'indirizzo del primo elemento del vettore di bytes `cc`, nei quali sono codificati in codice ASCII le prime 5 lettere del vostro cognome senza spaziature e con caratteri tutti minuscoli (se il vostro cognome contiene meno di 5 lettere, i byte non utilizzati conterranno il valore 0). In oltre il byte individuato dall'indirizzo simbolico `co+3` sarà stato inizializzato in modo da contenere 0 prima di richiamare la procedura `ppp` per la prima volta. Scrivere i valori posti dalla procedura `ppp` nel registro `%o0` al ritorno dalla prima e dalla seconda chiamata.

Risposte scritto parziale di Architettura del 28-1-98

Cognome : **Nome:** **Login:**

1. (punti: -1, 4) **a** **b** **c** **d** **e** **X f**
2. (punti: -1, 4) **a** **X b** **X c** **d** **e** **f**
3. (punti: -1, 4) **X a** **b** **c** **d** **X e** **f**
4. (punti: -1, 4) **a** **b** **c** **X d** **X e** **f**
5. (punti: -1, 4) **a** **b** **X c** **d** **X e** **f**
6. (punti: -1, 4) **X a** **b** **c** **X d** **e** **f**
7. (punti: 5)

```
ld [%g0-108], %g3
```

8.

   (a) (punti: -1, 1) bit indirizzi virtuali: 24

   (b) (punti: -1, 1) massimo numero pagine fisiche: 1024

   (c) (punti: 2) Esempio: Giovanni Chiola, login 1977s313
   C1= 3, N1=7, v1= 00-00000000-01100111000.110
   LL= 57, v2= 00-00000000-11100100000.000

   (d) (punti: 4) Esempio: Giovanni Chiola, login 1977s313
   f1= 0000000011-01100111000.110, f2= 0000000011-11100100000.000

9. (punti: 20)

```
MANCA1: sethi %hi(nn),%g2
MANCA2: nop OPPURE mov 0,%o7
MANCA3: stb %g2,[%g3+3]
MANCA4: st %g3,[%i2]
MANCA5: ret
```

10. (punti: 12) inserisce tutte le nuove coppie di caratteri adiacenti nelle due stringhe, mantenendole ordinate per numero decrescente di occorrenze complessive nelle due stringhe. Ritorna il numero di coppie nuove di caratteri effettivamente aggiunte.
    Esempio: “giova” e “chiol” restituisce 4 e 3 (la seconda occorrenza di “io” provoca solo un incremento del conteggio del numero di occorrenze ed un riordinamento).