Scritto di Architettura prof. Chiola

5/6/1996

Cognome :

Nome :

Matricola :

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 dello scritto espresso in trentesimi sarà calcolato applicando la seguente formula, nella quale s rappresenta la somma dei punti totalizzati per le varie domande:

$$\text{Voto} = \text{troncamento}\,(c1 + \min(s, c2) \times c3 + (s - c2) \times c4)$$

I coefficienti $c1$, $c2$, $c3$ e $c4$ sono valori costanti che vengono determinati tenendo conto della statistica dei valori s ottenuti dai candidati che hanno consegnato compiti precedenti in modo da ottenere una distribuzione equa dei voti tra 18/30 e 30/30 per coloro che hanno risposto correttamente ad una porzione sufficiente di domande. I valori delle costanti per questo compito sono:

$$\text{Voto} = \text{troncamento}\,(3.8 + \min(s, 27) \times 0.6 + (s - 27) \times 0.4)$$

in modo da raggiungere la sufficienza (18/30) totalizzando almeno $s = 25$ punti ed il massimo (30/30) totalizzando almeno $s = 52$ punti (su un massimo di 67). Il voto finale verrà 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 della prova orale. Voti superiori a 30 verranno poi considerati equivalenti alla "lode".

Domande

1. (punti: -1, 4)
   L'organizzazione della memoria virtuale con "segmentazione e paginazione":

   (a) Consente di gestire in modo efficiente (senza rilocazione di programmi già in esecuzione) segmenti di dimensione arbitraria;

   (b) Consente di gestire in modo efficiente (senza rilocazione di programmi già in esecuzione) l'allocazione dinamica di memoria ad uno o più segmenti;

   (c) Consente di definire pagine segmentate in sottoinsiemi di celle con associati diversi privilegi di accesso;

   (d) Consente di definire segmenti composti da pagine fisiche contigue con associati diversi privilegi di accesso;

   (e) Consente di definire segmenti composti da pagine logiche contigue con associati gli stessi privilegi di accesso;

   (f) nessuna delle affermazioni precedenti é corretta.

2. (punti: -1, 4)
   Una codifica binaria in "virgola fissa" per la rappresentazione di soli numeri positivi compresi tra 0 e 3.5:

   (a) richiede complessivamente almeno 3 bit di rappresentazione;

   (b) richiede almeno 3 bit di rappresentazione prima della virgola;

   (c) richiede almeno 5 bit dopo la virgola per codificare correttamente il numero 1.2;

   (d) richiede complessivamente almeno 5 bit per codificare correttamente il numero 1.375;

   (e) utilizzando complessivamente 8 bit permette di codificare correttamente il numero 1.1;

   (f) nessuna delle affermazioni precedenti é corretta.

3. (punti: -1, 4)
   Un Bus di tipo asincrono:

   (a) é basato sull'uso di un suo clock proprio per la temporizzazione dei segnali sui fili, sganciato dal clock degli altri dispositivi;

   (b) sfrutta un protocollo di "handshaking" per consentire la comunicazione anche tra dispositivi funzionanti con velocità diversa;

   (c) Richiede l'uso di un "arbitro" per la designazione di un master nel caso si vogliano connettere dispositivi con possibilità di accesso in DMA oltre all'unità centrale di elaborazione;

   (d) Non permette la connessione di più di una unità CPU a causa della mancanza di un unico segnale di clock per sincronizzarne il funzionamento;

   (e) permette la connessione tra due o più dispositivi master, grazie alla mancanza di un clock ed un controllo centralizzato;

   (f) nessuna delle affermazioni precedenti é corretta.

4. (punti: -1, 4)
I registri in una CPU di tipo RISC:

(a) sono introdotti in numero elevato per sostituire la cache dati di primo livello;

(b) permettono di realizzare i modi di indirizzamento indiretto mediante registro e autoincremento, non normalmente presenti in macchine CISC;

(c) essendo in numero elevato rispetto alle architetture CISC rendono praticabile e conveniente l'organizzazione detta Load/Store dell'insieme di istruzioni;

(d) richiedono l'uso di tecniche sofisticate di ottimizzazione da parte del compilatore per essere usati in modo efficiente;

(e) consentono di stabilire una comunicazione diretta dei dati e dei risultati tra la CPU e l'unità Floating Point (normalmente separata), senza il passaggio in memoria centrale;

(f) nessuna delle affermazioni precedenti é corretta.

5. (punti: -1, 4)
Una richiesta di interruzione:

(a) proveniente da un dispositivo che richiede un basso tempo di servizio può essere mascherata in modo da farla apparire di priorità superiore rispetto al valore stabilito dalla connessione "daisy chain";

(b) viene inoltrata alla CPU mediante l'esecuzione di una procedura chiamata "interrupt handler" da parte del controller di un dispositivo;

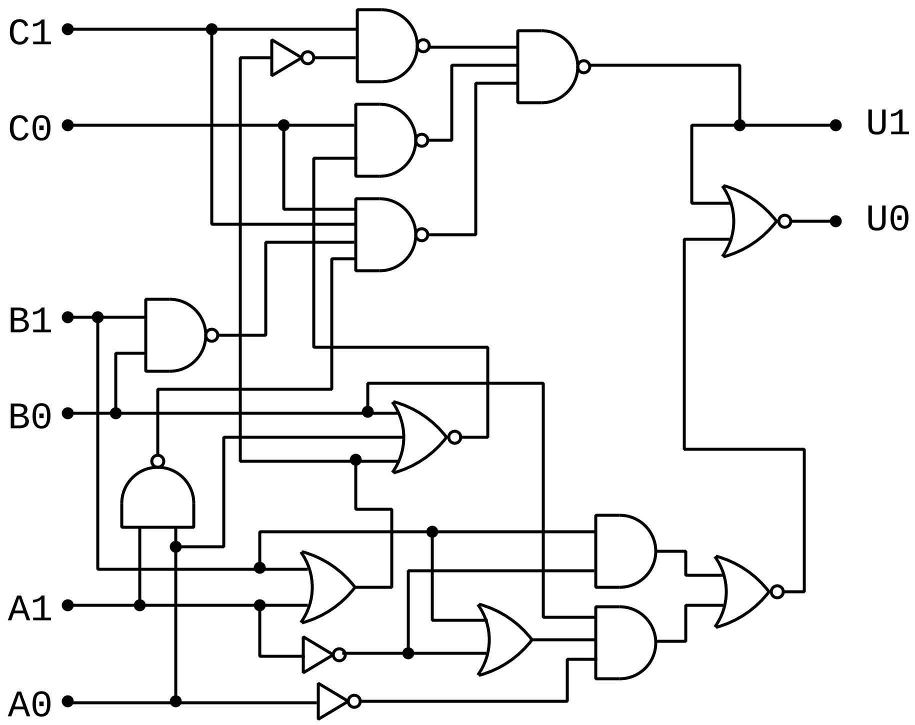
(c) non puo' essere inoltrata in modo asincrono alla CPU se il bus di sistema é di tipo sincrono;

(d) viene riconosciuta dalla CPU quando questa scandisce i registri memory mapped dei dispositivi seguendo l'ordine specificato nel vettore delle interruzioni;

(e) viene sempre gestita durante la fase di fetch della prossima istruzione da parte della CPU;

(f) nessuna delle affermazioni precedenti é corretta.

6. Considerare il seguente circuito con tre insiemi di 2 ingressi booleani e un insieme di 2 uscite booleane:



(a) (punti: -1, 1)

Il circuito é di tipo Combinatorio, Sequenziale Asincrono o Sequenziale Sincrono?

(b) (punti: 8) ricavare parte della tavola di verità:

$A1$	$A0$	$B1$	$B0$	$C1$	$C0$	$U1$	$U0$
0	0	0	0	0	0		
0	0	0	0	0	1		
0	0	0	0	1	0		
0	0	0	0	1	1		
0	0	0	1	0	0		
0	0	0	1	0	1		
0	0	0	1	1	0		
0	0	0	1	1	1		
0	0	1	1	0	0		
0	0	1	1	0	1		
0	0	1	1	1	0		
0	0	1	1	1	1		
1	0	0	0	0	0		
1	0	0	0	0	1		
1	0	0	0	1	0		
1	0	0	0	1	1		

(c) (punti: 3)

Interpretando i tre insiemi di ingressi come codifica di tre numeri binari (senza segno), quale funzione calcola il circuito?

(d) (punti: 3)

Disegnare la riduzione del circuito per il calcolo della stessa funzione su due insiemi di 2 ingressi booleani.

(e) (punti: 7)

Data la seguente intestazione di una descrizione VHDL del circuito individuato al punto (d), completare la descrizione del modulo:

```
entity my_entity is
port(A0,A1,B0,B1: in bit; U: out bit);
end my_entity;

library ttl;
use ttl.not1,ttl.or2,ttl.and2;
```

```
architecture structure of my_entity is

        component not1_gate
         port(a: in bit; b:out bit);
        end component;

        for all:not1_gate use entity ttl.not1;

        component or2_gate
         port(a,b: in bit; c:out bit);
        end component;

        for all:or2_gate use entity ttl.or2;

        component and2_gate
         port(a,b: in bit; c:out bit);
        end component;

        for all:and2_gate use entity ttl.and2;

        signal ...                              ; -- DA COMPLETARE


begin -- DA COMPLETARE

    ...




end structure;
```

7. (punti: 10)
   Considerare le seguenti funzione e procedura “funct1” e “proc2” definite in Assembler SPARC:

```
funct1: sra %o0,1,%g2
        andcc %o0,1,%g0
        be,a L2
        add %g2,-1,%g2
L2:     retl
        mov %g2,%o0


proc2:  save %sp,-104,%sp
        ldub [%i0+%i2],%o1
```

```
        mov %i2,%o0
        b L11
        stb %o1,[%i1+%i2]
L7:     stb %o2,[%i1+%o0]
L11:    call funct1
        mov %o0,%i2
        cmp %o0,0
        bge,a L6
        ldub [%i1+%i2],%o2
        ret
        restore
L6:     ldub [%i1+%o0],%o1
        cmp %o2,%o1
        blu,a L7
        stb %o1,[%i1+%i2]
        ret
        restore
```

`proc2` viene chiamata per 10 volte da un programma principale il quale inserisce nel registro `%o0` l'indirizzo di un vettore di 10 caratteri (codificati in ASCII su 8 bit) contenenti le prime cinque lettere del vostro cognome negli indirizzi pari (0, 2, 4, 6, 8) inframmezzate con le prime cinque lettere del vostro nome negli indirizzi dispari (1, 3, 5, 7, 9), e nel registro `%o1` l'indirizzo di un altro vettore di 10 caratteri inizializzato ai valori "X" per tutte le componenti. Nel caso il vostro nome o cognome contenga meno di cinque lettere, inserire i successori dell'ultima lettera fino a completare il vettore. Il registro `%o2` contiene valori interi crescenti tra 0 e 9 (0 la prima volta, 1 la seconda, ecc.).

Scrivere il contenuto del secondo vettore di caratteri al termine delle 10 esecuzioni di `proc2`.

8. (punti: 15)
   Completare la seguente traduzione della procedura `proc2` della domanda precedente in assembler VM-2 con le 5 istruzioni mancanti (ovviamente passando i parametri in ingresso a `proc2` sullo stack):

```
proc2: LODL 1
       ADDL 3
       PSHI
       LODL 1
       ADDL 2
       POPI
       DESP 1
       LODL 1
L1:    CALL funct1
       STOL -2
       JNEG L2
       LODL 1
       ADDL 2
```

```
        PSHI
        LODL -2
I1:
        PSHI
        LODL -4
        SIGN
I2:
        JPOS L2
        LODL 1
        ADDL 2
I3:
        LODL -2
I4:
        POPI
        LODL -2
I5:
        JUMP L1
L2:     RETN
```

Il parametro viene passato alla funzione `parent` inserendolo nel registro accumulatore. La funzione sostituisce il valore del parametro col valore calcolato nel registro accumulatore.

Risposte Scritto di Architettura, 5/6/1996

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

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

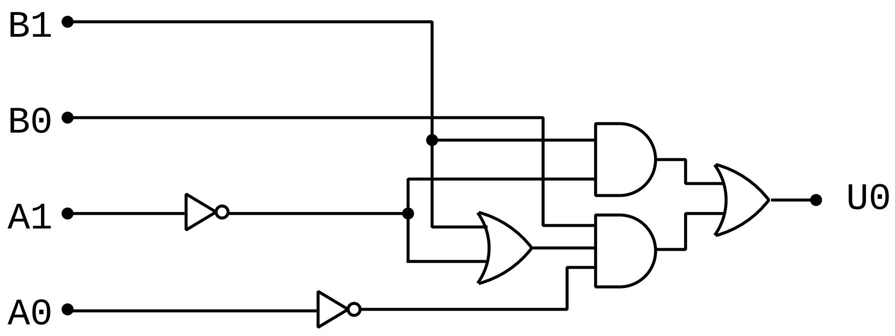
(a) (punti: -1, 1) X Combinatorio Asincrono Sincrono

(b) (punti: 8)

$A1$	$A0$	$B1$	$B0$	$C1$	$C0$	$U1$	$U0$
0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0
0	0	0	0	1	0	1	0
0	0	0	0	1	1	1	0
0	0	0	1	0	0	0	1
0	0	0	1	0	1	0	1
0	0	0	1	1	0	1	0
0	0	0	1	1	1	1	0
0	0	1	1	0	0	0	1
0	0	1	1	0	1	0	1
0	0	1	1	1	0	0	1
0	0	1	1	1	1	0	1
1	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0
1	0	0	0	1	0	0	0
1	0	0	0	1	1	1	0

(c) (punti: 3) individua l'indice dell'insieme che codifica il valore maggiore; in caso di pareggio, da priorità all'insieme di indice inferiore.

(d) (punti: 3):



(e) (punti: 7):

```
        signal notA0,notA1,or_one,and_one,and_two,and_two1:bit;

begin
 not1_gate_one: not1_gate port map(a=>A1,b=>notA1);
```

```
 not1_gate_two: not1_gate port map(a=>A0,b=>notA0);

 or2_gate_one: or2_gate port map(a=>B1,b=>notA1,c=>or_one);

 or2_gate_two: or2_gate port map(a=>and_one,b=>and_two,c=>U);

 and2_gate_one: and2_gate port map(a=>B1,b=>notA1,c=>and_one);

 and2_gate_two1: and2_gate port map(a=>B0,b=>or_one,c=>and_two1);

 and2_gate_two2: and2_gate port map(a=>and_two1,b=>notA0,c=>and_two);
end structure;
```

7. (punti: 10) I dieci caratteri (cinque iniziali del cognome e cinque iniziali del nome) in ordine "alfabetico heap".
   Esempio: Cognome "chiol", Nome "giova", stringa iniziale "cghiioovla", risultato "achigoovli".

8. (punti: 15)

```
       ...
       LODL -2
I1:    ADDL 2
       PSHI
       LODL -4
       SIGN
I2:    ADDL -3
       JPOS L2
       LODL 1
       ADDL 2
I3:    POPI
       LODL -2
I4:    ADDL 2
       POPI
       LODL -2
I5:    STOL 1
       JUMP L1
       ...
```

Risposte Scritto di Architettura, 5/6/1996

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

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

   (a) (punti: -1, 1) Combinatorio Asincrono Sincrono

   (b) (punti: 8)

$A1$	$A0$	$B1$	$B0$	$C1$	$C0$	$U1$	$U0$
0	0	0	0	0	0		
0	0	0	0	0	1		
0	0	0	0	1	0		
0	0	0	0	1	1		
0	0	0	1	0	0		
0	0	0	1	0	1		
0	0	0	1	1	0		
0	0	0	1	1	1		
0	0	1	1	0	0		
0	0	1	1	0	1		
0	0	1	1	1	0		
0	0	1	1	1	1		
1	0	0	0	0	0		
1	0	0	0	0	1		
1	0	0	0	1	0		
1	0	0	0	1	1		

   (c) (punti: 3)

   (d) (punti: 3)

(e) (punti: 7)

```
        signal ...

    begin -- DA COMPLETARE

      ...

    end structure;
```

7. (punti: 10)

8. (punti: 15)

```
        ...
        LODL -2
I1:
        PSHI
        LODL -4
        SIGN
I2:
        JPOS L2
        LODL 1
        ADDL 2
I3:
        LODL -2
I4:
        POPI
        LODL -2
I5:
        JUMP L1
        ...
```