

Primo Compitino del 17 aprile 2009

Con soluzioni proposte dal docente

Scrivere nome e cognome su tutti i fogli utilizzati

Gli esercizi contrassegnati con [Extra] verranno valutati solo se si raggiunge la sufficienza con gli altri esercizi, quindi vanno svolti dopo tutti gli altri.

[Esercizio 1]

(1.a) Scrivere la tavola di verità della seguente funzione booleana in tre variabili:

$$F(A, B, C) = ((A \text{ and not } B) \text{ or } C) \text{ xor } (\text{not } B \text{ or } A)$$

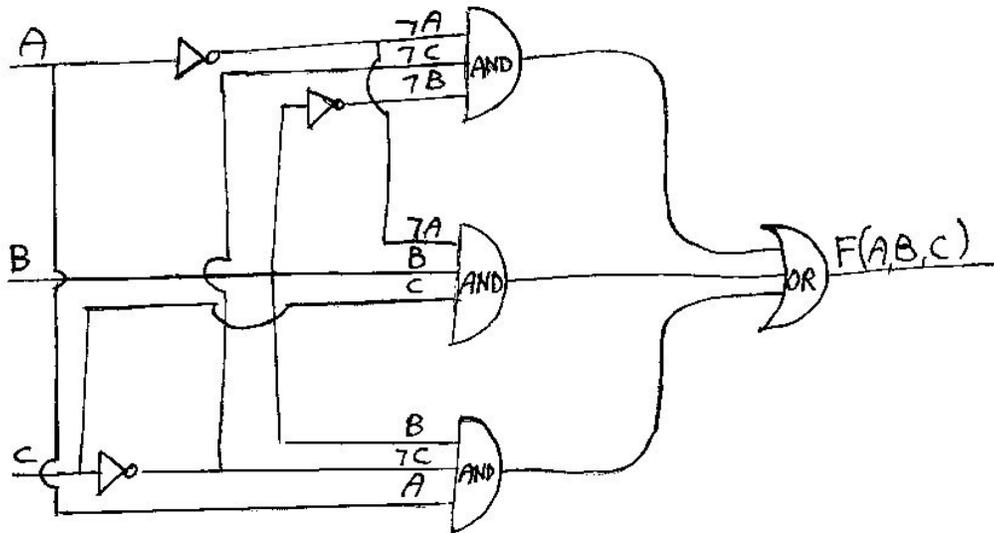
Soluzione: La seguente tabella mostra nell'ultima colonna la tavola di verità della funzione $F(A,B,C)$ [si intende che TRUE=1 e FALSE=0]. Le colonne D, E e F sono dei valori intermedi, utili per il calcolo della funzione.

| A | B | C | D | E | F | F(A,B,C) |
|---|---|---|-------------|--------|------------|----------|
| | | | A and not B | D or C | not B or A | E xor F |
| 0 | 0 | 0 | FALSE | FALSE | TRUE | TRUE |
| 0 | 0 | 1 | FALSE | TRUE | TRUE | FALSE |
| 0 | 1 | 0 | FALSE | FALSE | FALSE | FALSE |
| 0 | 1 | 1 | FALSE | TRUE | FALSE | TRUE |
| 1 | 0 | 0 | TRUE | TRUE | TRUE | FALSE |
| 1 | 0 | 1 | TRUE | TRUE | TRUE | FALSE |
| 1 | 1 | 0 | FALSE | FALSE | TRUE | TRUE |
| 1 | 1 | 1 | FALSE | TRUE | TRUE | FALSE |

(1.b) Sfruttando la tavola di verità, disegnare un circuito combinatorio che calcola la funzione F usando solo porte AND, OR e NOT. Se si desidera, si possono usare porte AND e OR con più di due ingressi.

Soluzione: Il circuito che si richiedeva era quello basato sulla lettura della tavola di verità di $F(A, B, C)$ come una espressione booleana, costituita dall'OR delle righe per cui la funzione vale TRUE, e per ogni riga l'AND delle variabili A, B e C oppure del loro negato. Quindi si chiedeva il circuito che rappresentava $F(A, B, C) = (\neg A \text{ and } \neg B \text{ and } \neg C) \text{ or } (\neg A \text{ and } B \text{ and } C) \text{ or } (A \text{ and } B \text{ and } \neg C)$.

Il circuito che segue usa porte OR e AND con tre ingressi.



[Esercizio 2]

(2.a) Si mostri la rappresentazione in complemento a 2 su otto bit dei numeri -74 e 43.

Soluzione: -74 = 10110110 43 = 00101011

(2.b) Si calcoli la somma binaria delle due rappresentazioni trovate al punto (2.a). Mostrare sia la configurazione di bit risultante che il suo valore in base 10.

Soluzione: La somma è 11100001 Valore in base 10: -31

(2.c) [Extra] Si mostrino l'OR e l'AND delle due rappresentazioni trovate al punto (2.a), e si mostri il corrispondente valore in base 10. E' vero che la somma di questi due valori è uguale a -31, cioè proprio il risultato di $-74 + 43$? Se è vero, pensi che si tratti di un caso o di una proprietà generale? Perché?

Soluzione:

10110110 AND 00101011 = 00100010 [= 34 in base 10]

10110110 OR 00101011 = 10111111 [= -65 in base 10]

Si, è vero: 34 - 65 = - 31

Non si tratta di un caso, ma è una proprietà generale.

Infatti la somma di due bit differisce dal loro OR solo quando entrambi valgono 1 ($1 + 1 = 10$, ma $1 \text{ OR } 1 = 1$). Per riottenere la somma, possiamo aggiungere al risultato dell'OR il risultato di $1 \text{ AND } 1 = 1$; questo funziona perché in tutti gli altri casi il risultato dell'AND vale 0. Questo ragionamento si estende facilmente a sequenze di bit.

[Esercizio 3]

(3.a) Si considerino le seguenti configurazioni di bit, che rappresentano numeri in virgola mobile in un formato che prevede **1 bit** per il segno, **4 bit** per l'esponente (in complemento a 2) e **11 bit** per la mantissa.

Quali numeri in base 10 rappresentano?

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Soluzione:

La prima configurazione rappresenta il numero **-16.3125** o, equivalentemente, **-(16+5/16)**.

Infatti il primo bit (**1**) indica che il numero è negativo; l'esponente (**0101**) vale **5** in complemento a due; e spostando la virgola di **5** posizioni verso destra dall'inizio della mantissa si ottiene **10000,010100**. Di questo numero la parte intera vale **16**, mentre la parte decimale vale $1 \cdot 2^{-2} + 1 \cdot 2^{-4} = 0,3125$.

La seconda configurazione rappresenta il numero **0.21875** o, equivalentemente, **7/32**.

Il primo bit (**0**) indica che il numero è positivo; l'esponente (**1110**) vale **-2** in complemento a due; e spostando la virgola di **2** posizioni verso sinistra dall'inizio della mantissa si ottiene **0,0011100000000** = **0,00111**. Di questo numero la parte intera vale 0, mentre la parte decimale vale $1 \cdot 2^{-3} + 1 \cdot 2^{-4} + 1 \cdot 2^{-5} = 0,21875$.

(3.b) [Extra] Mostrare la rappresentazione in virgola mobile (usando lo stesso formato del punto (3.a)) del numero in base dieci 8193,5. Può essere rappresentato in modo esatto oppure c'è un errore di troncamento? Se c'è troncamento, qual'è il numero effettivamente rappresentato?

Soluzione:

La rappresentazione binaria del numero **8193,5** è **10000000000001.1** (come si trova facilmente sfruttando la tabella delle potenze del 2 fornita). Quindi nella notazione esponenziale normalizzata (spostando la virgola prima del primo 1), **8193,5** avrebbe esponente **14** e mantissa **100000000000011**. Il numero **14** in base 2 è **1110**, e **non è rappresentabile in complemento a 2 su quattro bit** (il numero massimo rappresentabile è **0111**, cioè 7). Quindi **8193,5** non è rappresentabile in virgola mobile usando lo stesso formato del punto (3.a).

Se si disponesse di **5** bit per l'esponente e **11** per la mantissa, allora **8193,5** sarebbe rappresentato come

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

In questo caso ci sarebbe un troncamento (si perderebbero gli ultimi due uni) e il numero effettivamente rappresentato sarebbe $0,1 * 2^{14} = 2^{13} = 8192$.

[Esercizio 4]

(4.a) Qual'è la distanza di Hamming tra le configurazioni di 8 bit corrispondenti alle sequenze esadecimali A7 e 8C?

Soluzione:

A7 = 10100111 8C = 10001100

Distanza di Hamming: **4**

(4.b) Esiste una configurazione di 8 bit (oppure di due cifre esadecimali) che abbia distanza di Hamming minore o uguale a 2 da ognuna delle due sequenze del punto (4.a)? In caso positivo, mostrane una. In caso negativo, spiegare perché.

Soluzione:

Esistono diverse configurazioni di bit che hanno distanza di Hamming uguale a due da **A7** e **8C**, per esempio **10100100 = A4**.

[Esercizio 5]

(5.a) Spiegare brevemente a cosa servono i registri speciali PC (Program Counter) e IR (Instruction Register) della CPU.

Soluzione:

Il registro PC contiene in ogni istante l'indirizzo di memoria della prossima istruzione da eseguire. Il registro IR contiene l'istruzione che si sta eseguendo.

(5.b) La tabella seguente mostra una porzione della memoria del computer contenente un programma in linguaggio macchina. Si assuma che l'esecuzione cominci con il valore **B0** nel registro PC. Eseguire manualmente le istruzioni finché la macchina non si fermi.

| indirizzo | contenuto |
|-----------|-----------|
| B0 | 15 |
| B1 | BA |
| B2 | 17 |

| | |
|----|----|
| B3 | BA |
| B4 | 56 |
| B5 | 57 |
| B6 | 36 |
| B7 | BA |
| B8 | C0 |
| B9 | 00 |
| BA | 66 |

Soluzione:

Una descrizione soddisfacente dell'esecuzione del programma comprende il contenuto di PC e IR ad ogni passo, e l'informazione su quando e come viene cambiato il contenuto della memoria centrale e/o dei registri generali. Per esempi:

| Contenuto di PC | Contenuto di IR | Note |
|-----------------|-----------------|--|
| B0 | <sconosciuto> | Indirizzo della prima istruzione da eseguire |
| B2 | 15BA | Carico 66 (contenuto di BA) in R5 |
| B4 | 17BA | Carico 66 in R7 |
| B6 | 5657 | Somma di R5 e R6 in R7 (in comp a 2): CC |
| B8 | 36BA | Copio R6 in BA (CC) |
| B10 | C000 | Termina l'esecuzione |

(5.c) Cosa contengono i registri della CPU (solo quelli modificati dal programma) quando la macchina si ferma?

Soluzione: **R5 = 66; R7 = 66; R6 = CC [66 e CC sono in esadecimale!]**

(5.d) Cosa contiene la cella di indirizzo BA quando la macchina si ferma?

Soluzione: **BA** contiene **CC**

(5.e) [Extra] Se all'inizio la cella BA avesse contenuto il valore 42, cosa avrebbe contenuto alla fine?

Descrivere in meno di dieci parole che cosa fa il programma.

Scrivere un programma più corto di quello mostrato che abbia lo stesso effetto sulla

memoria.

Soluzione:

Se **BA** avesse contenuto all'inizio **42**, alla fine avrebbe contenuto **84**.

Il programma raddoppia il contenuto della cella di indirizzo **BA**.

Si può accorciare il programma evitando di caricare due volte **BA**.

| indirizzo | contenuto | |
|-----------|-----------|---------------------------|
| B0 | 15 | |
| B1 | BA | Carico BA in 15 |
| B2 | 56 | |
| B3 | 55 | Metto R5+R5 in R6 |
| B4 | 36 | |
| B5 | BA | Copio R6 in BA |
| B6 | C0 | |
| B7 | 00 | HALT |
| B8 | | B8 e B9 possono contenere |
| B9 | | qualunque valore. |
| BA | 66 | |

[Esercizio 6] Si consideri un computer in cui possono essere eseguiti più processi in time-sharing.

(6.a) Si indichino almeno quattro risorse del computer il cui accesso deve essere coordinato dal sistema operativo.

Soluzione:

Il sistema operativo deve coordinare l'accesso a praticamente tutte le risorse disponibili: CPU, Memoria Centrale, Memoria Secondaria, Periferiche di ogni tipo (tastiera, mouse, stampanti, lettori CD, ...), Accesso alla rete, ...

(6.b) Quali sono i tre stati in cui si può trovare un processo? Descriverne brevemente il significato.

Soluzione:

Un processo può essere **in esecuzione** (la CPU esegue le sue istruzioni, di solito per un quanto di tempo prefissato); **pronto** (tipicamente in una coda di processi pronti, in attesa che venga il proprio turno di utilizzo della CPU); **in attesa** (attende di completare un'operazione "lenta" come un accesso ad una periferica, oppure che si liberi una risorsa al momento occupata; quando l'attesa termina, tornerà nella coda dei processi pronti).

(6.c) Descrivere una situazione in cui un processo non può consumare interamente il suo quanto di tempo.

Soluzione:

Per esempio, quando durante l'esecuzione sulla CPU arriva un interrupt (es: da una periferica); il processo viene interrotto e messo nella coda dei processi **pronti**, mentre viene eseguito il gestore delle interruzioni (che è parte del sistema operativo). Oppure quando durante l'esecuzione il processo cerca di accedere ad una risorsa condivisa che in quel momento non è disponibile, oppure se fa un'operazione che richiede di attendere una periferica: in questi due casi il processo viene messo in **attesa**.

[Esercizio 7]

(7.a) Cos'è un protocollo di rete a cosa serve? Si descriva sommariamente il protocollo CSMA/CD, usato per reti a bus di tipo Ethernet.

Soluzione:

Un protocollo di rete è un insieme di regole che governano la trasmissione di messaggi tra i computer in una rete.

Il protocollo CSMA/CD prevede che un messaggio sul bus venga inoltrato a tutti i computer sulla rete; solo il computer cui è indirizzato lo trattiene, mentre gli altri lo scartano; per inviare un messaggio, un computer aspetta che il bus sia libero, e dopo averlo inviato lo rilegge per sentire se è stato alterato a causa di un conflitto; in quest'ultimo caso lo ritrasmette dopo un intervallo di tempo casuale.

(7.b) Individuare le varie parti che formano la seguente URL, e descriverne il significato:
<http://www.di.unipi.it/~andrea/Didattica/IG09/index.html>

Soluzione:

| | |
|--------------------------|---|
| http : | protocollo |
| www: | host (computer) |
| di.unipi.it: | dominio (it : dominio di primo livello, unipi : di secondo livello...) |
| ~andrea/Didattica/IG09 : | path (cammino che individua una directory) |
| index.html : | file |
