

Corso di Fondamenti di Informatica-Modulo A

Alcune soluzioni al Primo Compitino

10 dicembre 2005

Domanda 1 Considerare un circuito booleano con tre ingressi binari A, B, C e una singola uscita R . L'uscita R è tale per cui la stringa binaria $ABCR$ abbia un numero dispari di zeri. Ad esempio, se $ABC = 000$, allora $R = 1$ in modo tale che $ABCR = 0001$ abbia un numero dispari (in questo caso 3) di zeri. Come altro esempio, se consideriamo $ABC = 011$ allora $R = 1$, in modo tale che $ABCR = 0111$ abbia un numero dispari (in questo caso 1) di zeri.

Si chiede di:

1. Scrivere la tabella di verità del circuito che calcola R ;
2. Disegnare il circuito di cui al punto precedente, utilizzando esclusivamente porte di tipo AND, OR e NOT.

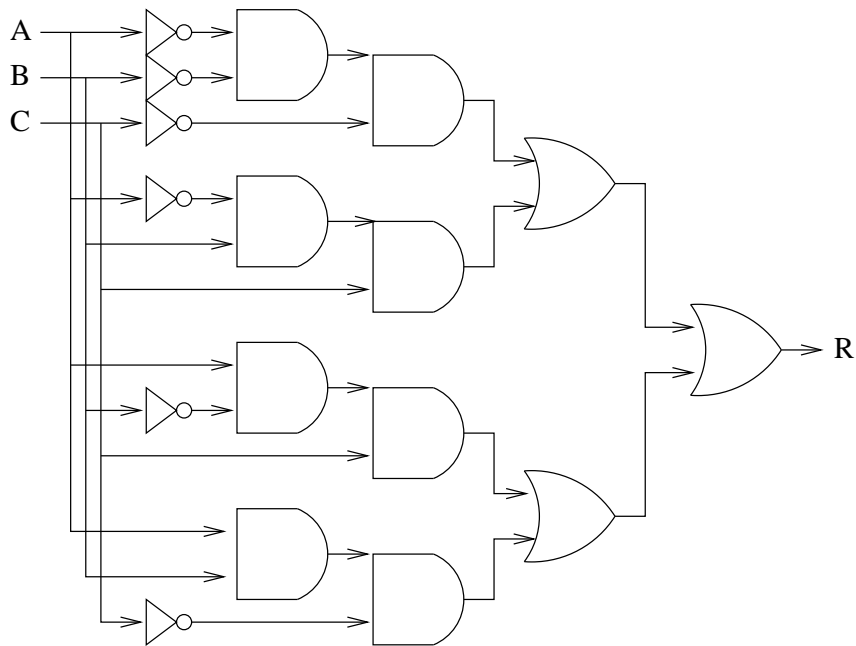
Risposta La tabella di verità del circuito richiesto è la seguente:

A	B	C	R
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

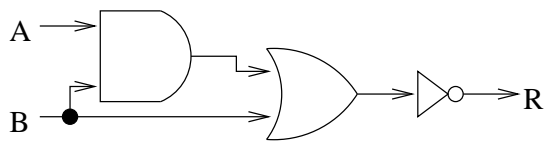
La tabella di verità precedente diventa l'equazione logica:

$((\text{not } A) \text{ and } (\text{not } B) \text{ and } (\text{not } C)) \text{ or}$
 $((\text{not } A) \text{ and } B \text{ and } C) \text{ or}$
 $(A \text{ and } (\text{not } B) \text{ and } C) \text{ or}$
 $(A \text{ and } B \text{ and } (\text{not } C))$

Da cui si ricava il circuito logico seguente:



Domanda 2 Si consideri il circuito logico in figura:



Scrivere tutte le combinazioni degli input A e B che causano un output R uguale a 1.

Risposta La tabella di verità per questo circuito è la seguente:

A	B	R
0	0	1
0	1	0
1	0	1
1	1	0

Pertanto, le combinazioni degli input tali che $R = 1$ sono $A = 0, B = 0$ e $A = 1, B = 1$.

Domanda 3 Si considero i numeri binari in complemento a due $X = 0100_2$ e $Y = 0011_2$. Eseguire la somma $X + Y$ scrivendo il risultato in complemento a due (con 4 bit) e in decimale. Specificare se si verifica o no overflow.

Risposta I due numeri in complemento a due X e Y risultano positivi, dato che il bit più a sinistra vale 0. Pertanto in decimale risulta $X = 4_{10}, Y = 3_{10}$. La somma in complemento a due da $X + Y = 0111_2 = 7_{10}$. Notare che la somma *non* da overflow, in quanto i segni degli addendi e del risultato sono uguali.

Domanda 4 Si considero i numeri binari in complemento a due $X = 0101_2$ e $Y = 0110_2$. Eseguire la somma $X + Y$ scrivendo il risultato in complemento a due (con 4 bit) e in decimale. Specificare se si verifica o no overflow.

Risposta I due numeri in complemento a due X e Y risultano anche qui positivi; in particolare, si ha $X = 5_{10}, Y = 6_{10}$. Sommando X e Y con 4 bit, si ha $X + Y = 1011_2$. Si noti che $X + Y$ risulta essere un numero *negativo*, perché nella sua rappresentazione in complemento a due il bit più a sinistra vale 1. In particolare, $X + Y = -5_{10}$. In questo caso si è verificato overflow, perché gli addendi hanno lo stesso segno, ma il risultato ha segno opposto.

Domanda 5 Si consideri una immagine bitmap larga 1024 pixel e alta 768 pixel. Supponiamo che il color di ogni pixel possa essere codificato mediante 3 bytes compressivi. Supponiamo inoltre che nessuna forma di compressione venga utilizzata per l'immagine. Quante celle di memoria di dimensione 1 byte sono necessarie per memorizzare l'immagine? (È sufficiente indicare il calcolo senza scrivere analiticamente il risultato).

Risposta In tutto sono necessari $1024 \times 768 \times 3$ byte di memoria per memorizzare l'immagine.

Domanda 6 Supponiamo che un calcolatore abbia un bus indirizzi ampio 10 bit. Quale è la dimensione massima (espressa in numero di bytes) della memoria che può utilizzare? Motivare la risposta.

Risposta Il bus indirizzi serve per rappresentare l'indirizzo di memoria che viene acceduto nelle operazioni di lettura/scrittura della memoria RAM. Con 10 bit è possibile rappresentare tutti i numeri (positivi) compresi tra 0 e $2^{10} - 1$; ciascun numero rappresenta un indirizzo di memoria. Pertanto la dimensione massima della memoria è di 2^{10} bytes.

Domanda 7 Viene fornito di seguito lo schema di un programma in C++. Completare il programma compilando il contenuto della funzione `main()` in modo tale da realizzare la seguente funzionalità: viene chiesto all'utente di inserire tramite tastiera due numeri interi A, B , con $A \leq B$. Se l'utente inserisce per errore due interi che non soddisfano la proprietà precedente (ossia, inserisce per errore due interi tali che $A > B$), il programma deve richiedere A e B finché la condizione $A \leq B$ viene soddisfatta. Il programma stampa la somma di tutti i numeri appartenenti all'intervallo $[A, B]$ (estremi inclusi). Ad esempio, se $A = 3, B = 6$, il programma deve stampare 18 (che è la somma $3 + 4 + 5 + 6$). Se $A = 6, B = 6$, il programma deve stampare 6.

Risposta Una possibile soluzione (non l'unica!) è la seguente.

```
#include <iostream>
using namespace std;

int main( void )
{
    int A;
    int B;
    int somma = 0;
    int i;

    // Si richiedono A e B all'utente; si ripete la richiesta
    // finche' risulta A <= B
    do {
        cin >> A;
        cin >> B;
    } while ( A > B );

    // Calcolo della somma, utilizzando il ciclo for (era possibile
    // anche utilizzare i cicli do-while e while-do)
    for ( i=A; i<=B; i++ ) {
        somma = somma+i;
    }

    cout << ``La somma vale `` << somma << endl;

    return 0;
}
```