

Esercizi sulla rappresentazione dell'informazione

Esercizio 1A

Trovare la rappresentazione binaria di 56,83 in virgola fissa quattro bit di precisione.

Soluzione 1A:

La rappresentazione binaria di 56,83 è 111000,1101

Per ottenere tale rappresentazione si procede separatamente per la parte intera e per la parte frazionaria.

Parte intera:

$$56:2 = 28 \text{ con resto } 0$$

$$28:2 = 14 \text{ con resto } 0$$

$$14:2 = 7 \text{ con resto } 0$$

$$7:2 = 3 \text{ con resto } 1$$

$$3:2 = 1 \text{ con resto } 1$$

$$1:2 = 0 \text{ con resto } 1$$

quindi la rappresentazione binaria di 56 è 111000

Parte frazionaria:

$$0,83 \times 2 = 1,66$$

$$0,66 \times 2 = 1,32$$

$$0,32 \times 2 = 0,64$$

$$0,64 \times 2 = 1,28$$

quindi la rappresentazione binaria di 0,83 è 0,1101

Esercizio 1-B

Trovare la rappresentazione binaria di 47,76 in virgola fissa quattro bit di precisione.

Soluzione 1B

La rappresentazione binaria di 47,76 è 101111,1100

Per ottenere tale rappresentazione si procede separatamente per la parte intera e per la parte frazionaria.

Parte intera:

$$47:2 = 23 \text{ con resto } 1$$

$$23:2 = 11 \text{ con resto } 1$$

$11:2 = 5$ con resto 1
 $5:2 = 2$ con resto 1
 $2:2 = 1$ con resto 0
 $1:2 = 0$ con resto 1

quindi la rappresentazione binaria di 47 è 101111

Parte frazionaria:

$0,76 \times 2 = 1,52$
 $0,52 \times 2 = 1,04$
 $0,4 \times 2 = 0,08$
 $0,08 \times 2 = 0,16$

quindi la rappresentazione binaria di 0,76 è 0,1100

Esercizio 1C

Trovare la rappresentazione binaria di 39,58 in virgola fissa quattro bit di precisione.

Soluzione 1C

La rappresentazione binaria di 39,58 è 100111,1001

Per ottenere tale rappresentazione si procede separatamente per la parte intera e per la parte frazionaria.

Parte intera:

$39:2 = 19$ con resto 1
 $19:2 = 9$ con resto 1
 $9:2 = 4$ con resto 1
 $4:2 = 2$ con resto 0
 $2:2 = 1$ con resto 0
 $1:2 = 0$ con resto 1

quindi la rappresentazione binaria di 39 è 100111

Parte frazionaria:

$0,58 \times 2 = 1,16$
 $0,16 \times 2 = 0,32$
 $0,32 \times 2 = 0,64$
 $0,64 \times 2 = 1,28$

quindi la rappresentazione binaria di 0,58 è 0,1001

Esercizio 1D

Convertire in base 5 con rappresentazione in virgola fissa il numero decimale 214,1362 avendo a disposizione 5 cifre per la parte intera e 6 per la parte decimale. La rappresentazione ottenuta precisa o un approssimazione del numero decimale di partenza?

Soluzione 1D

Conversione della parte intera:

- $214 : 5 = 42$ con resto 4
- $42 : 5 = 8$ con resto 2
- $8 : 5 = 1$ con resto 3
- $1 : 5 = 0$ con resto 1

Conversione della parte frazionaria:

- $0,1362 \times 5 = 0,681$
- $0,681 \times 5 = 3,405$
- $0,405 \times 5 = 2,025$
- $0,025 \times 5 = 0,125$
- $0,125 \times 5 = 0,625$
- $0,625 \times 5 = 3,125$

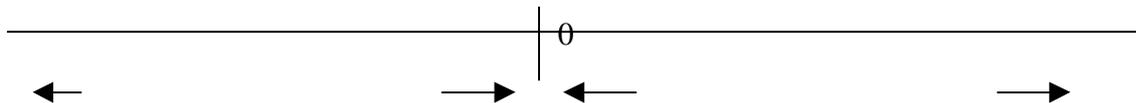
Pertanto la rappresentazione in base 5 è 01324,032003. Non è esatta perché la conversione della parte frazionaria è stata interrotta al 6° passo (come richiesto dalla rappresentazione), mentre la codifica esatta non è ottenibile perché il numero è periodico.

Esercizio 1E

Dato un numero $\langle s,m.e \rangle$ espresso in virgola mobile con 8 bit per rappresentare la mantissa normalizzata, e 4 bit per rappresentare l'esponente in complemento a due, quali sono i numeri MINIMO e MASSIMO rappresentabili? Fornire tali limiti in binario ed in decimale.

Soluzione Esercizio 1E

Si vedano per dettagli le esercitazioni on-line sul sito su questo argomento.



I ranges sono: $\langle 0,10000000,1000 \rangle$ e $\langle 0,11111111,0111 \rangle$ per i positivi e $\langle 1,10000000,1000 \rangle$ e $\langle 1,11111111,0111 \rangle$ per i negativi

Esercizio 1F

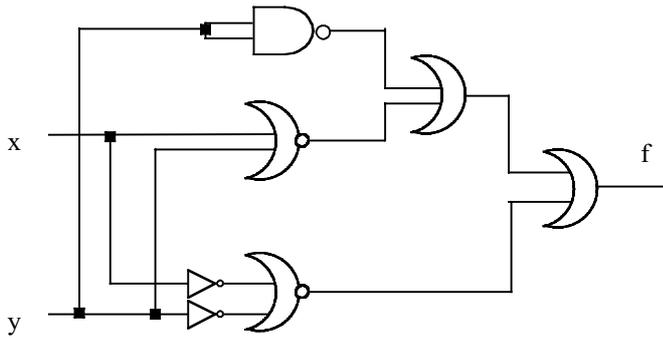
Si esprima nella rappresentazione in complemento a due il numero -4720 rappresentato in base 8. Si supponga di lavorare con interi a 16 bits.

Soluzione 1F

Dovete dapprima convertire in base 10. Il valore in base 10 corrispondente al numero 4720_8 è 2512_{10} .

Il valore assoluto del numero dato rappresentato come numero binario intero positivo di 16 bit è 0000 100 111 010 000. Per ottenere il corrispondente numero negativo in complemento a due, basta complementare e poi sommare 1.

Il complemento a uno è 1111 011 000 101 111 e quindi il complemento a due è (sommando 1) 1111 011 000 110 000



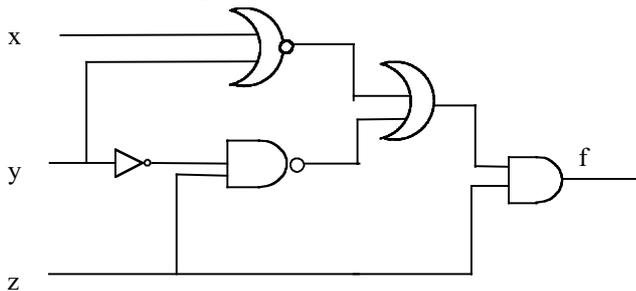
Ricavare dapprima l'Espressione Booleana completa, quindi minimizzare.

Soluzione 2C

$$f = ((\overline{x + y}) + (\overline{y \cdot y})) + (\overline{\overline{x + \overline{y}}}) = ((\overline{x \cdot \overline{y}}) + \overline{y}) + x \cdot y = \overline{y} + xy$$

Esercizio 2D

Analizzare il seguente circuito:



Ricavare dapprima l'Espressione Booleana completa, quindi minimizzare.

Soluzione 2D

$$f = ((\overline{x + y}) + (\overline{y \cdot z})) \cdot z = (\overline{x \cdot \overline{y}} + y + \overline{z}) \cdot z = \overline{xyz} + yz = (\overline{xy} + y) \cdot z = (\overline{xy} + y(1 + \overline{x})) \cdot z = (\overline{x} + y) \cdot z = \overline{xz} + yz$$

Soluzione 2F

$$\begin{array}{cccc} \text{-----} & & \text{-----} & \\ \text{-----} & \text{-----} & \text{-----} & \text{-----} \\ \text{---} & \text{---} & \text{---} & \text{---} \end{array}$$

$$((x \oplus x) \oplus y) \oplus (x \oplus (y \oplus y)) = (x \oplus y) \oplus (x \oplus y) = x \oplus y + x \oplus y = x \oplus y = x \oplus y$$

Esercizi di sintesi

Esercizio 3A

Progettare un circuito che riceve in ingresso la codifica binaria di un valore x , con $0 \leq x \leq 7$, e fornisce in uscita la codifica binaria della somma aritmetica tra x e il suo precedente (il precedente del valore 0 è 7). Attenzione: di quanti segnali di ingresso, e di quanti segnali di uscita ho bisogno?

Soluzione 3A

Per rappresentare l'input sono necessari 3 bit, ma per l'uscita del circuito sono necessari 4 bit, pertanto la tabella di verità è la seguente:

x2	x1	x0	y3	y2	y1	y0
0	0	0	0	1	1	1
0	0	1	0	0	0	1
0	1	0	0	0	1	1
0	1	1	0	1	0	1
1	0	0	0	1	1	1
1	0	1	1	0	0	1
1	1	0	1	0	1	1
1	1	1	1	1	0	1

Osservando la tabella si ricava subito: $y_0 = 1$ e $y_1 = \text{not}(x_0)$

Minimizzando con Karnaugh si ottengono le seguenti espressioni:

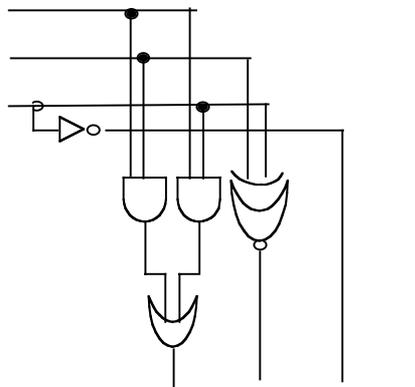
\ x0	0	1
x2x1 \		
00	1	
01		1
11		1
10	1	

$$y_2 = \overline{x_1} \overline{x_0} + x_1 x_0 = x_1 \text{ XNOR } x_0$$

	x_0	0	1
x_2x_1			
00			
01			
11	1	1	
10			1

$$y_3 = x_2 x_1 + x_2 x_0$$

La realizzazione con porte logiche :



Esercizio 3B

Progettare un circuito che riceve in ingresso la codifica binaria di un valore x , con $0 \leq x \leq 7$, e fornisce in uscita la codifica del doppio di $x + 2 \bmod 16$. (cioè, $y=2(x+2)$ e quando y supera 1111 (cioè la codifica di 15) il risultato è 0000). Attenzione: quanti bit servono per rappresentare l'input, e quanti per rappresentare l'output?

Soluzione 3B

Per rappresentare l'input servono 3 bit, per l'uscita del circuito sono necessari 4 bit, pertanto la tabella di verità è la seguente:

x_2	x_1	x_0	y_3	y_2	y_1	y_0	$2(x+2)$
0	0	0	0	1	0	0	4
0	0	1	0	1	1	0	6
0	1	0	1	0	0	0	8
0	1	1	1	0	1	0	10
1	0	0	1	1	0	0	12
1	0	1	1	1	1	0	14
1	1	0	0	0	0	0	16
1	1	1	0	0	0	0	18

Andava ugualmente bene (il testo era effettivamente ambiguo) produrre nelle ultime righe i valori 0000 e 0010 (rispettivamente il modulo 16 di 16 10000 e 18 10010).

Il corrispondente circuito si ricava facilmente.

Esercizio 3C

Date cinque variabili booleane x_0, x_1, x_2, x_3 e x_4 , nonché due variabili booleane di controllo y_0 e y_1 , progettare un circuito con una uscita z tale che, se $y_1 y_0 = 0 0$ $z = x_0 \text{ AND } x_4$, se $y_1 y_0 = 0 1$ $z = x_1 \text{ AND } x_4$, se $y_1 y_0 = 1 0$ $z = x_2 \text{ AND } x_4$, se $y_1 y_0 = 1 1$ $z = x_3 \text{ AND } x_4$. Realizzare il circuito usando porte logiche.

NOTA: questo esercizio si risolve molto più facilmente scrivendo le specifiche direttamente mediante una espressione booleana SOP!! Ovvero: traducete la frase che fornisce la specifica in una corrispondente espressione logica nelle variabili $y_1 y_0 x_0, x_1, x_2, x_3$ e x_4 . Potere arrivare alla soluzione anche con una tabella di verità, ma, con 7 variabili di ingresso, la cosa risulterebbe considerevolmente più onerosa.

Soluzione 3C

Notate che la frase: “se $y_1 y_0 = 0 0$ $z = x_0 \text{ AND } x_4$, se $y_1 y_0 = 0 1$ $z = x_1 \text{ AND } x_4$, se $y_1 y_0 = 1 0$ $z = x_2 \text{ AND } x_4$, se $y_1 y_0 = 1 1$ $z = x_3 \text{ AND } x_4$ ” elenca i casi che portano ad uno (o meglio, che rendono “vera”) la funzione booleana di uscita z .

La condizione: “se $y_1 y_0 = 0 0$ $z = x_0 \text{ AND } x_4$ “
 si traduce in: $(\overline{y_1 y_0})(x_0 x_4)$

Analogamente, le altre tre condizioni si traducono rispettivamente nelle:

$$(\overline{y_1 y_0})(x_1 x_4)$$

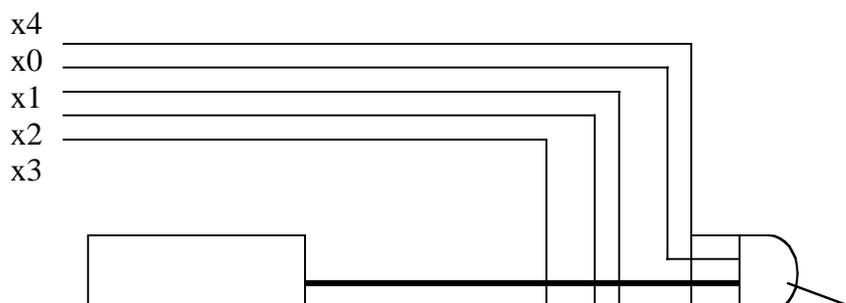
$$(\overline{y_1 y_0})(x_2 x_4)$$

$$(\overline{y_1 y_0})(x_3 x_4)$$

Quindi, la disgiunzione (OR) di queste quattro condizioni di verità è data dalla:

$$z = (\overline{y_1 y_0})(x_0 x_4) + (\overline{y_1 y_0})(x_1 x_4) + (\overline{y_1 y_0})(x_2 x_4) + (\overline{y_1 y_0})(x_3 x_4)$$

osservate che le combinazioni condizionanti di y_1 e y_0 sono ottenute mediante un decodificatore 2×4 .



Esercizio 3D

Una banca è dotata di una doppia porta d'ingresso: la prima collega la strada con il vano per il controllo degli oggetti metallici, la seconda permette di passare dal vano all'interno della banca. Nel vano una telecamera identifica automaticamente il numero di persone presenti. Le porte rispettano le seguenti regole:

- Se almeno una persona si trova davanti alla prima porta e nessuno si trova nel vano, allora la prima porta si apre;
- Se una o due persone si trovano nel vano, la seconda porta rimane aperta e la prima chiusa;
- Se più di due persone si trovano nel vano, la prima porta rimane aperta, la seconda chiusa e un messaggio avverte i clienti di uscire dal vano.

Progettare il circuito delle due porte codificando come linee d'ingresso la presenza di una persona davanti alla prima porta e il numero di persone nel vano (suggerimento: il numero può assumere uno dei valori { 0, 1, 2, 3 }) e come linee d'uscita i tre eventi: "la prima porta è aperta" (A_1), la "seconda porta è aperta" (A_2) e "un messaggio avverte i clienti di uscire dal vano" (W).

Soluzione 3D

Per rappresentare l'input servono 3 bit, il primo (S) indica se è presente una persona in strada davanti la prima porta, gli altri due (V_1 e V_2) codificano il numero di persone presenti nel vano. La tabella di verità è la seguente:

S	V ₁	V ₂	A ₁	A ₂	W
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	1	0	1
1	0	0	1	0	0
1	0	1	0	1	0
1	1	0	0	1	0
1	1	1	1	0	1

Osservando la tabella si ricava subito: $W = V_1 V_2$.

Minimizzando con Karnaugh si ottengono le seguenti espressioni:

$V_1 V_2$	00	01	11	10
S				
0	0	0	1	0
1	1	0	1	0

$$A_1 = V_1 V_2 + \overline{S V_1 V_2}$$

$V_1 V_2$	00	01	11	10
S				
0	0	1	0	1
1	0	1	0	1

$$A_2 = \overline{V_1} V_2 + V_1 \overline{V_2} = V_1 \text{ XOR } V_2$$

Il corrispondente circuito si ricava facilmente.

Esercizio 3E

Si richiede di progettare il circuito per il display di un ascensore in un edificio di 3 piani (più il piano terra). Il circuito riceve in ingresso le informazioni sul piano in cui si trova e sul piano di destinazione scelto da un utente. Le linee d'uscita del circuito devono codificare l'illuminazione o meno dell'indicatore di salita (#) e di quello di discesa (\$) secondo le seguenti regole:

- Se il piano di partenza è più in basso del piano di destinazione, illuminare l'indicatore di salita;
- Se il piano di partenza è più in alto del piano di destinazione, illuminare l'indicatore di discesa;
- Se l'utente seleziona lo stesso piano in cui si trova, non si accende nessun indicatore.

Progettare il circuito del display dell'ascensore. Suggerimento: il circuito avrà bisogno di 4 bit di ingresso, per codificare l'informazione relativa al piano di partenza (P2 e P1) ed al piano di destinazione (D2 e D1). Di quanti segnali di uscita ha bisogno?

Soluzione 3E

La tabella di verità è la seguente:

P_1	P_2	D_1	D_2	S	G
0	0	0	0	0	0
0	0	0	1	1	0
0	0	1	0	1	0
0	0	1	1	1	0
0	1	0	0	0	1
0	1	0	1	0	0
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	0	1
1	0	0	1	0	1
1	0	1	0	0	0

1	0	1	1	1	0
1	1	0	0	0	1
1	1	0	1	0	1
1	1	1	0	0	1
1	1	1	1	0	0

Minimizzando con Karnaugh si ottengono le seguenti espressioni:

	D_1D_2	00	01	11	10
P_1P_2					
00		0	1	1	1
01		0	0	1	1
11		0	0	0	0
10		0	0	1	0

$$S = \overline{P_1}D_1 + \overline{P_1}P_2D_2 + \overline{P_2}D_1D_2$$

	D_1D_2	00	01	11	10
P_1P_2					
00		0	0	0	0
01		1	0	0	0
11		1	1	0	1
10		1	1	0	0

$$G = \overline{D_1}P_1 + \overline{D_1}D_2P_2 + \overline{D_2}P_1P_2$$

Il corrispondente circuito si ricava facilmente.

Esercizio 3F

Progettare il circuito combinatorio che riceve in ingresso quattro bit s x_2 x_1 x_0 che rappresentano un intero con segno (s rappresenta il bit di segno) e produce in uscita quattro bit che codificano lo stesso intero rappresentato in complemento a due.

Soluzione 3F

Un intero rappresentato con il segno destina il bit più significativo all'indicazione del segno, 1 se negativo e 0 se positivo, mentre i restanti $n-1$ bit rappresentano il valore assoluto dell'intero rappresentato. Invece, nella rappresentazione in complemento a due, l'intero rappresentato si ricava dalla formula:

$$A = -c_{n-1} \times 2^{n-1} + \sum_{i=0}^{n-2} c_i \times 2^i$$

La tabella di verità è dunque la seguente (la prima colonna indica l'intero con segno in decimale, le successive 4 rappresentano l'input, cioè la corrispondente stringa binaria in rappresentazione con segno, e le successive 4 l'output, cioè la corrispondente stringa in complemento a due):

	s	x_2	x_1	x_0	y_3	y_2	y_1	y_0
+0	0	0	0	0	0	0	0	0

+1	0	0	0	1	0	0	0	1
+2	0	0	1	0	0	0	1	0
+3	0	0	1	1	0	0	1	1
+4	0	1	0	0	0	1	0	0
+5	0	1	0	1	0	1	0	1
+6	0	1	1	0	0	1	1	0
+7	0	1	1	1	0	1	1	1
-0	1	0	0	0	0	0	0	0
-1	1	0	0	1	1	1	1	1
-2	1	0	1	0	1	1	1	0
-3	1	0	1	1	1	1	0	1
-4	1	1	0	0	1	1	0	0
-5	1	1	0	1	1	0	1	1
-6	1	1	1	0	1	0	1	0
-7	1	1	1	1	1	0	0	1

Notate che la rappresentazione di interi con segno ha due modi di esprimere lo zero: +0 e -0. Invece, in complemento a due, lo zero è rappresentato solo dalla stringa 000..00. Per questo motivo, la configurazione di ingresso 1000 va tradotta in Ca2 ancora con la stringa 0000. la stringa 1000 in Ca2 è interpretata come -8 !!!