



Bases de l'architecture pour la programmation LIFASR 3

Circuit plus complexe

1- Dans cet exercice, on veut réaliser un comparateur de deux nombres binaires x_i et y_i à 1 bit, dont le schéma synoptique est donné par la figure 1.

- 1) Trouvez la table de vérité
- 2) Donnez les expressions logiques des sorties
- 3) Le **logigramme** du comparateur

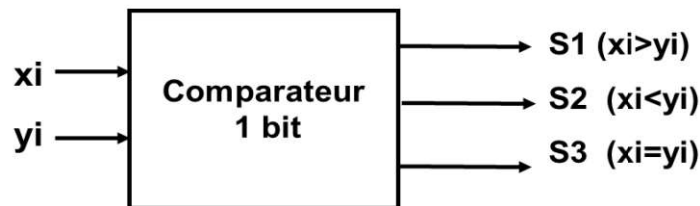


Figure 1

2- On veut maintenant réaliser un comparateur de deux nombres binaires à **deux bits** $X=X_1X_0$ et $Y=Y_1Y_0$, dont le schéma synoptique est donné par la figure 2. On note que X_0 et Y_0 sont les bits de poids les plus faibles.

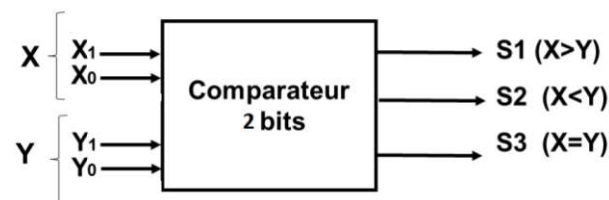


Figure 2

1. Donnez les expressions logiques des sorties S_1 , S_2 et S_3 en fonction des entrées X_i et Y_i avec $i=0, 1$ du comparateur à 1 bit.
2. Tracez le logigramme du comparateur à 2 bits.

3- On veut afficher les sorties du comparateur (S_1 , S_2 , S_3) sur un afficheur 7 segments en utilisant un **transcodeur 3 vers 7**, comme le montre la figure 2, et ce pour obtenir l'affichage donné par la figure 3.

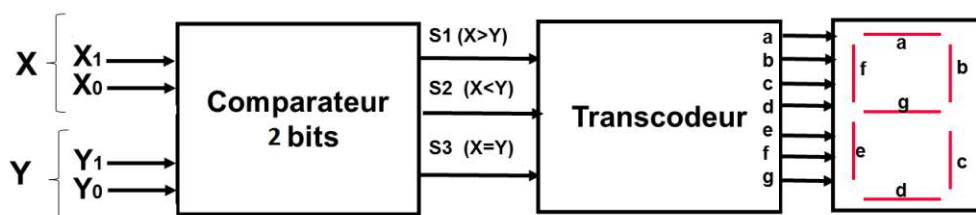


Figure 2



Bases de l'architecture pour la programmation LIFASR 3

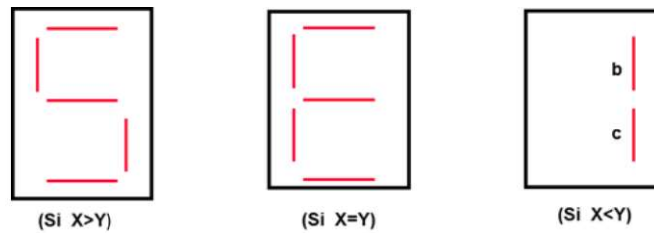


Figure 3

1. Donnez la **table de transcodage** permettant le passage du code S1, S2, S3 au code 7 segments.
2. En déduire le schéma interne (logigramme) du transcodeur.

Correction :

Table de vérité

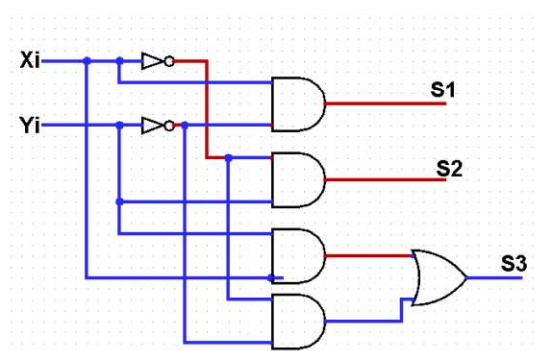
X_i	Y_i		S1	S2	S3
0	0		0	0	1
0	1		0	1	0
1	0		1	0	0
1	1		0	0	1

$$S1 = X_i \overline{Y_i}$$

$$S2 = \overline{X_i} Y_i$$

$$S3 = \overline{X_i} \overline{Y_i} + X_i Y_i = \overline{X_i} \oplus \overline{Y_i}$$

Logigramme



Partie 2 : Pour le comparateur à 2 bits (voir le cours svp)

Partie 3 :

Table de transcodage



Bases de l'architecture pour la programmation LIFASR 3

Affichage	S1	S2	S3		a	b	c	d	e	f	g
S	1	0	0		1	0	1	1	0	1	1
E	0	0	1		1	0	0	1	1	1	1
I	0	1	0		0	1	1	0	0	0	0

Les équations :

$$a = f = g = S_1 \cdot \overline{S_2} \cdot \overline{S_3} + S_3 \cdot \overline{S_1} \cdot \overline{S_2}$$

$$b = \overline{S_1} \cdot \overline{S_3} \cdot S_2$$

$$c = \overline{S_3} (S_2 \oplus S_1)$$

Logigramme :