

- ✓ *Additionneur 4 bits.*
- ✓ *Additionneur –soustracteur.*
- ✓ *Compareteur.*

1. Additionneur 4 bits:

1-1.introduction:

- Un additionneur sur 4 bits est un circuit qui permet de faire l'addition de deux nombres A et B de 4 bits chacun – A ($a_3a_2a_1a_0$) – B ($b_3b_2b_1b_0$)
 - En plus il tient en compte de la retenue entrante
 - En sortie on va avoir le résultat sur 4 bits ainsi que la retenue (5bits en sortie)
 - Donc au total le circuit possède 9 entrées et 5 sorties.
 - Avec 9 entrées on a $2^9=512$ combinaisons !!!!! Comment faire pour représenter la table de vérité ?????
 - Il faut trouver une solution plus facile et plus efficace pour concevoir ce circuit ?
 - Lorsque on fait l'addition en binaire, on additionne **bit par bit** en commençant à partir du poids faible et à chaque fois on **propage** la Retenue sortante au bit du rang supérieur.
- L'addition sur un bit peut se faire par un additionneur complet sur 1 bit.

Solution:

Pour effectuer l'addition de deux nombres de 4 bits, il suffit de chaîner entre eux par 4 additionneurs 1-bit complets. La retenue est ainsi propagée d'un additionneur à l'autre. Un tel additionneur est appelé un additionneur série.

1-2. Demi-additionneur :

Ce circuit, qui permettrait d'effectuer l'addition des deux bits de plus bas poids est appelé demi-additionneur (Half-Adder).
Ecrivons la table de vérité de celui-ci :

1-2-1. Table de vérité/équations/logigramme:

A	B	C	D
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Table 1

Si nous écrivons ces deux fonctions sous leur forme canonique il vient :

$$\begin{cases} D = \bar{A} B + A \bar{B} \\ C = A B \end{cases}$$

Nous reconnaissons pour la sortie D une fonction OU exclusif, donc :

$$\begin{cases} D = A \oplus B \\ C = A B \end{cases}$$

Ce qui peut être réalisé par le circuit schématisé sur le logigramme de la figure 3.

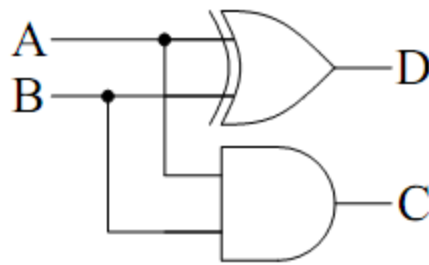
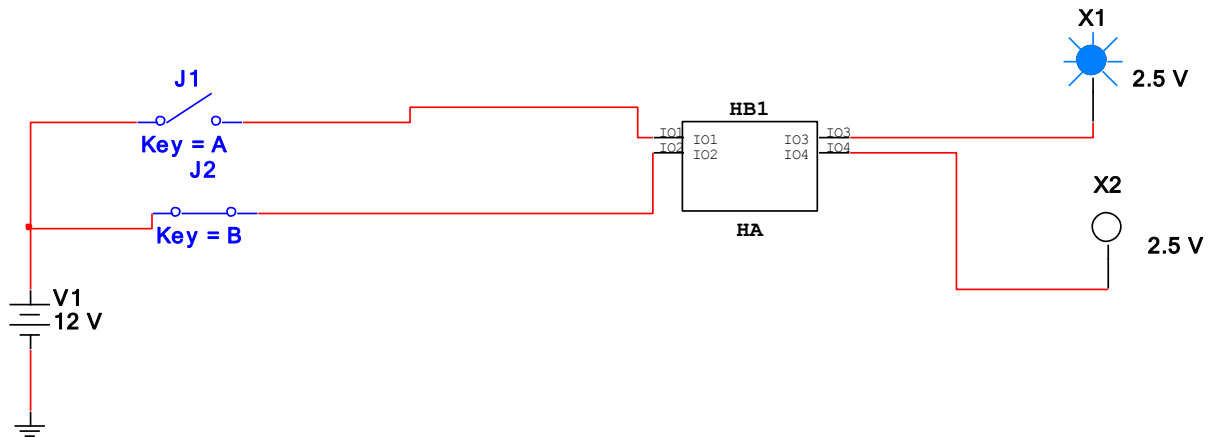


Figure 3

1-2-2.Circuit d'après ms11 (ADD 1 bit) :



1-3. Additionneur complet 1bit:

1-3-1. Table de vérité:

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

Table 2

Expressions de s & C :

$$\begin{cases} S = \bar{A} \bar{B} R + \bar{A} B \bar{R} + A \bar{B} \bar{R} + A B R \\ C = \bar{A} B R + A \bar{B} R + A B \bar{R} + A B R \end{cases}$$

D'après table de karnaugh on obtient :

$$C = A B + A R + B R$$

$$S = (A + B + R) \bar{C} + A B R$$

1-3-2.Circuit intégrer add 1 bit:

D'après les expressions S & C on obtiendra le logigramme suivant :

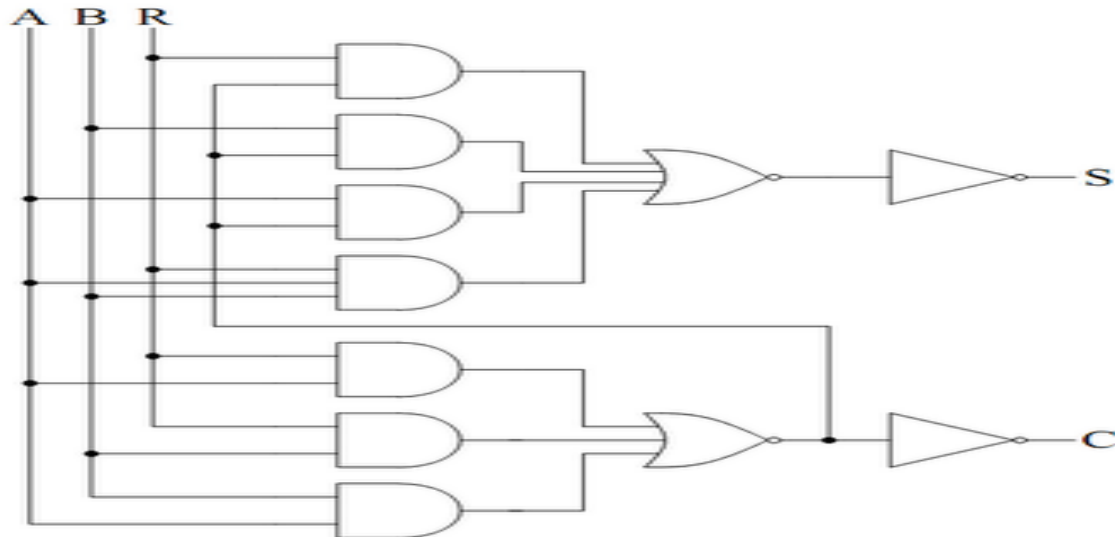
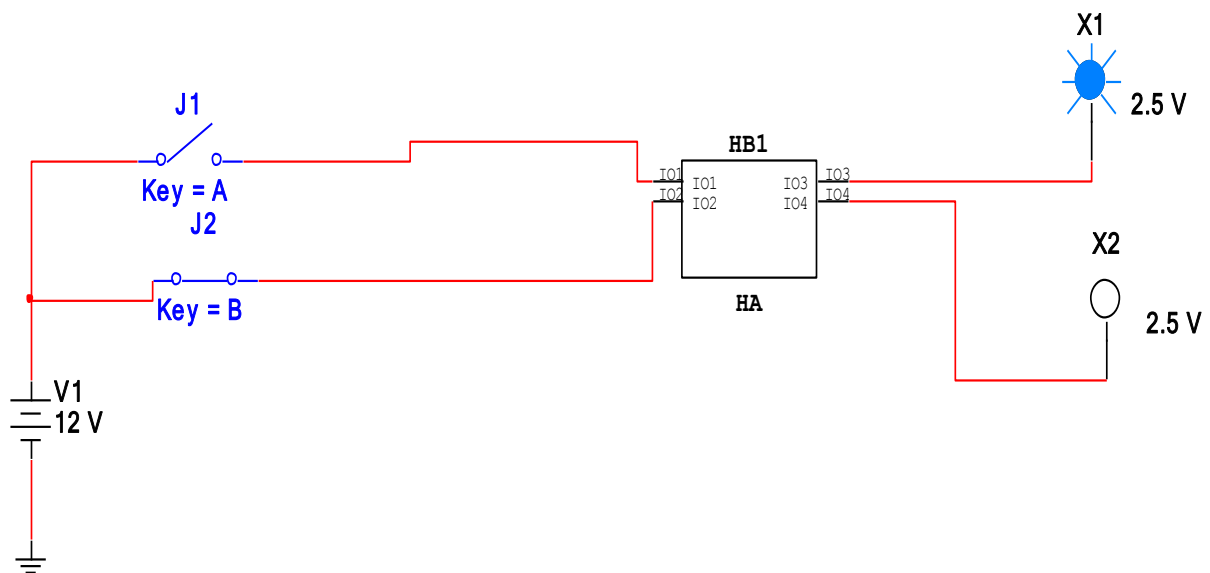


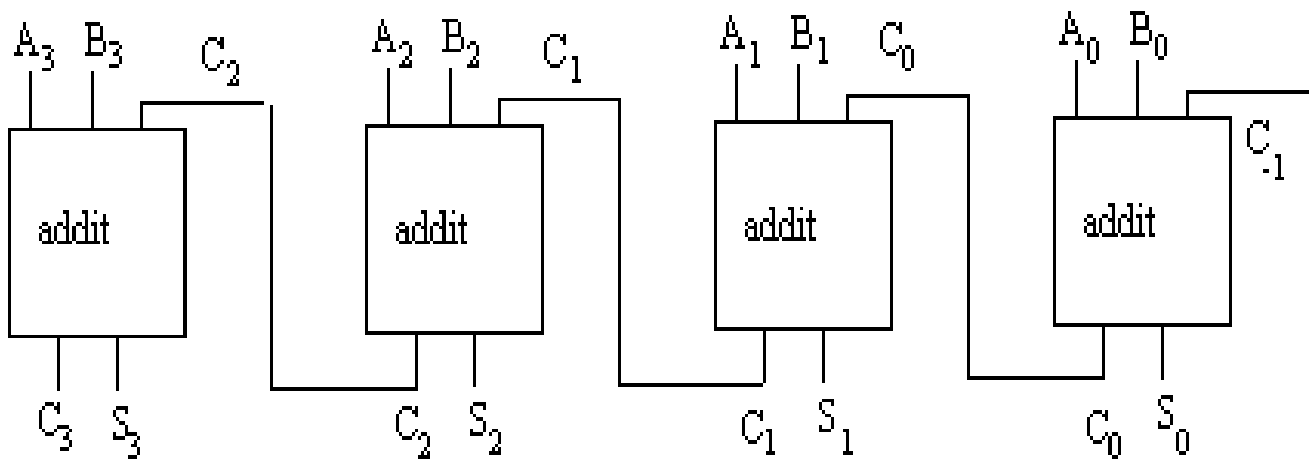
Figure 7

1-3-3.Circuit d'après ms11 (ADD 1 bit) :

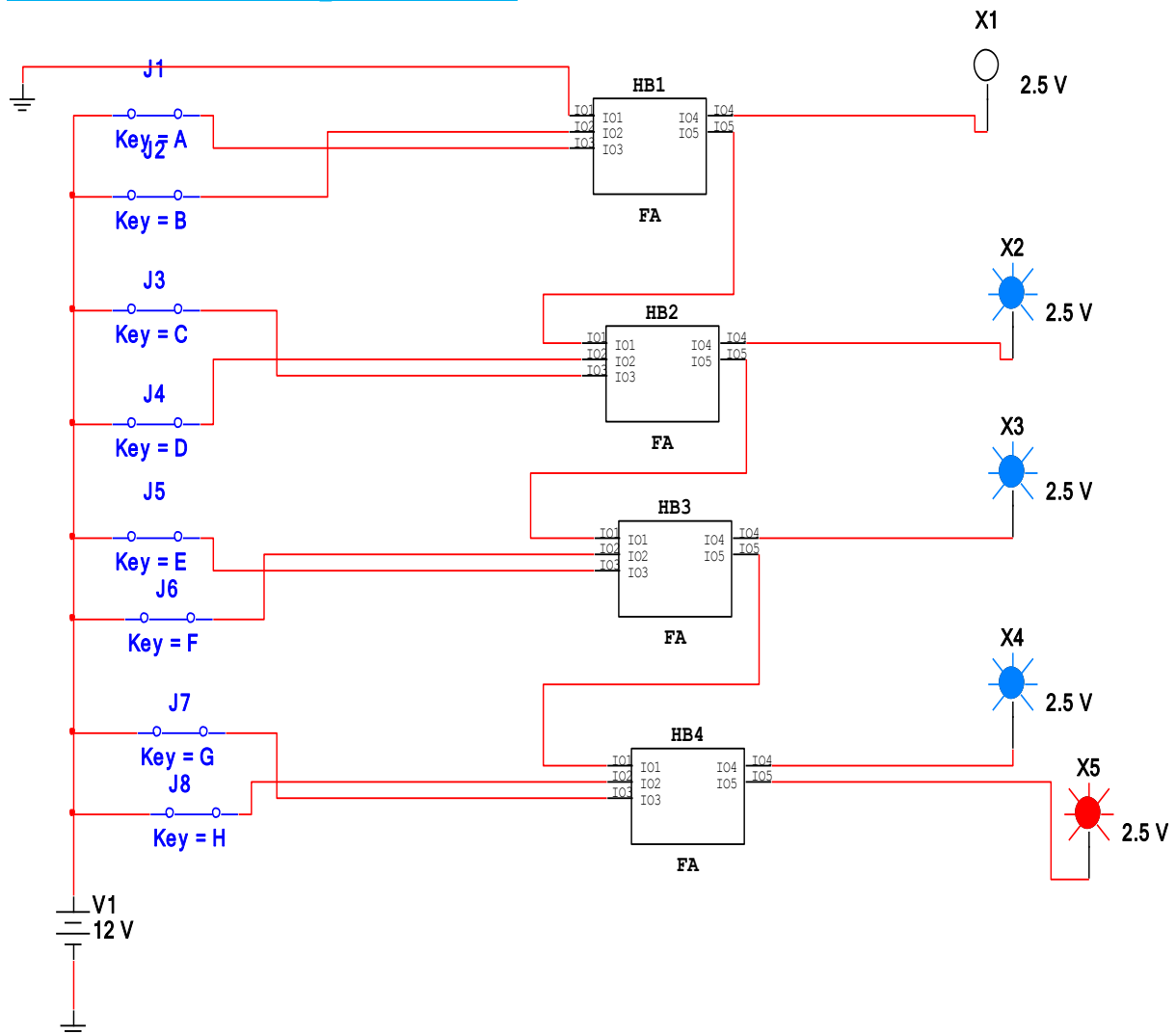


1-4. Additionneur 4 bit :

1-4-1. Association de 4 add 1 bit :



1-4-2. Circuit d'après ms11:



2-Additionneur – Soustracteur:

2-1.Introduction:

Il n'y a pas de circuit soustracteur dans un processeur parce que l'on peut implémenter la soustraction à l'aide de l'additionneur avec des modifications mineures. Pour ce faire, on exploite les propriétés du complément à 2 et le fait que le bit de poids faible de l'additionneur n'a pas de retenue d'entrée. En effet, effectuer $X - Y$ en complément à 2, est équivalent à $X + Y' + 1$. Pour effectuer la deuxième addition (+1), il suffit d'injecter un 1 en guise de retenue dans l'additionneur de poids faible. On peut donc supposer que l'on dispose d'un signal de contrôle c qui vaut 0 lorsque l'on veut faire une addition, et 1 lorsque l'on veut faire une soustraction. On utilise ce signal c comme retenue du bit de poids faible de l'additionneur. Enfin, pour obtenir Y' , il suffit de rajouter un inverseur (une porte XOR) en entrée de chacun des additionneurs 1-bit : $y_i \oplus c$; lorsque c vaut 0, la valeur d'entrée de l'additionneur i est y_i , et lorsque c vaut 1, la valeur d'entrée est y_i' . Donc, lorsque c vaut 0, l'opération effectuée par le circuit est $X + Y$, et lorsque c vaut 1, l'opération effectuée est $X + Y' + 1$.

2-2.Demi-soustracteur 1 bits:

2-2-1. Table de vérité/équations/logigramme:

La table de vérité pour un demi-soustracteur (ne tenant pas compte d'une éventuelle retenue provenant des bits de poids inférieurs) est la suivante :

A	B	D	C
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Table 3

Où D représente le résultat de la soustraction $A - B$ et C la retenue.
 Nous en déduisons les expressions logiques définissant D et C :

$$\begin{cases} D = \bar{A} B + A \bar{B} = A \oplus B \\ C = \bar{A} B \end{cases}$$

Et le schéma correspondant :

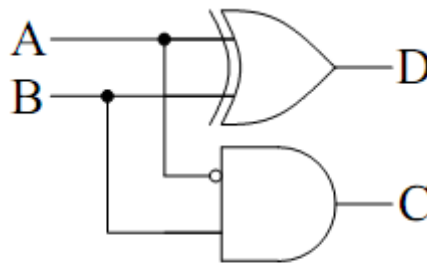
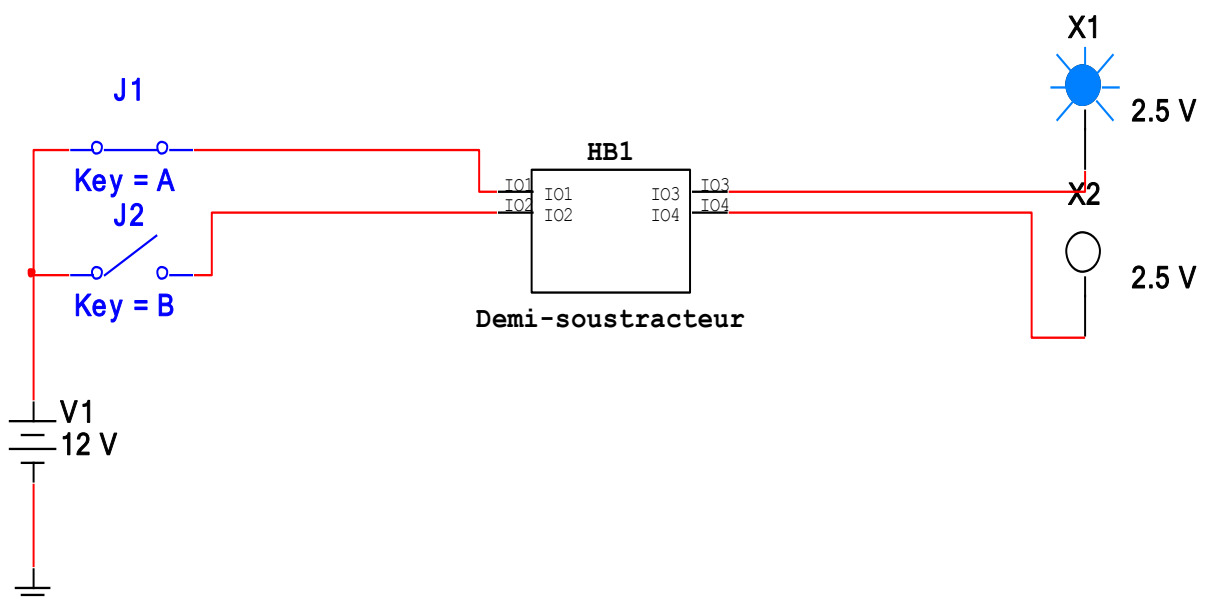


Figure 12

2-2-2.Circuit d'après ms11:

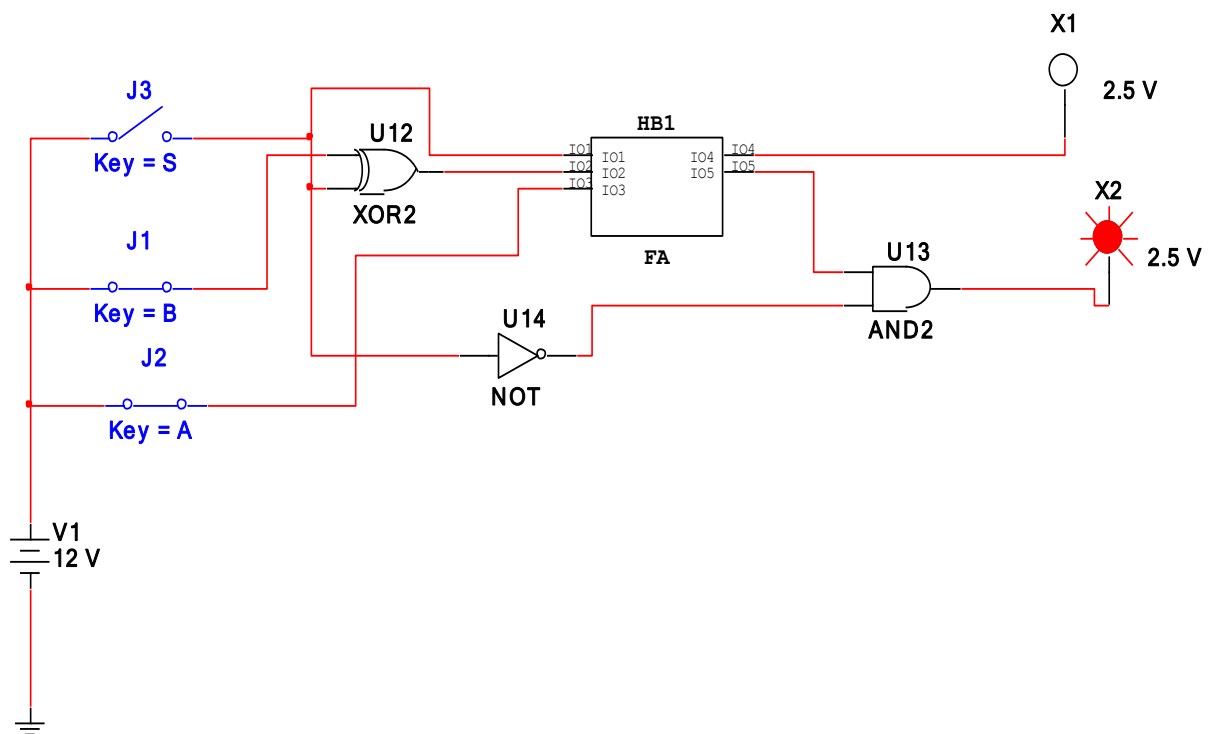


2-3.Additionneur-soustracteur 1bit:

2-3-1.Table de vérité:

	SST	A	b	r ‘	S	R
A+B	0	A	B	0	s	r
A-B	1	A	$\overline{B_1}$	1	s	0

2-3-2.Circuit d’après Ms11:



3-Comparteur:

3-1.Introduction:

Le comparateur est un circuit arithmétique permettant de comparer deux nombres binaires A et B. A et B doivent avoir la même longueur (nombre de bits). On cherche à savoir si $A > B$, $A < B$ ou $A = B$. On comprend donc que le circuit répond à une question à trois choix.

3-2.principe:

Les bits de même rang A_i et B_i des 2 mots à comparer sont analysés par une fonction "OU-exclusif-NON" pour donner en sortie l'indication d'égalité ($e=1$) ou de non égalité ($e=0$). Les 2 mots A et B sont égaux si et seulement si tous leurs bits de même rang A_i et B_i sont égaux. En conséquence pour obtenir $A=B$, il suffit de mettre en condition "ET" les différents résultats.

3-3. comparateur 1bit :

3-3-1. table de vérité / équations / logigramme :

A	B	C ($A > B$)	D ($A < B$)	E ($A = B$)
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

Note :

La fonction C doit être égale à 1 si et seulement si $A > B$, la fonction D si et seulement si $A < B$ et la fonction E si et seulement si $A = B$.

$$\begin{cases} C = A \bar{B} \\ D = \bar{A} B \\ E = \overline{A \oplus B} = \overline{A \bar{B} + \bar{A} B} = \overline{C + D} \end{cases}$$

La figure 14 présente le diagramme d'un bloc logique comparant deux bits A et B.

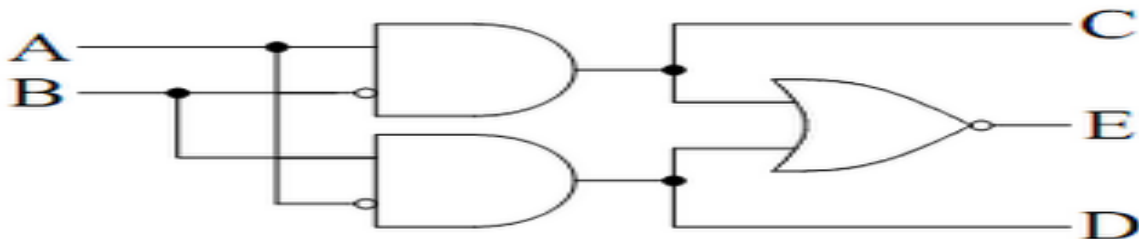


Figure 14

3-3-2. circuit d'après MS 11 :

