Architecture Des Ordinateurs

<<Rappel De La Logique Combinatoire>>

École Supérieure Polytechnique (ESP) de Dakar, 2020

Intervenant: Abdou DIOP

Les circuits combinatoires

Objectifs

- Apprendre la structure de quelques circuits combinatoires souvent utilisés (demi additionneur , additionneur complet,……..).
- Apprendre comment utiliser des circuits combinatoires pour concevoir d'autres circuits plus complexes.

1. Les Circuits combinatoires

- Un circuit combinatoire est un circuit numérique dont les sorties dépendent uniquement des entrées.
- $S_i=F(E_i)$
- $S_i=F(E_1,E_2,\ldots.,E_n)$

Schéma Bloc

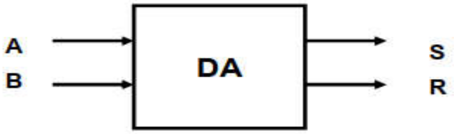
- C'est possible d'utiliser des circuits combinatoires pour réaliser d'autres circuits plus complexes.

Exemple de Circuits combinatoires

1. **Demi Additionneur**
2. **Additionneur complet**
3. **Comparateur**
4. **Multiplexeur**
5. **Demultiplexeur**
6. **Encodeur**
7. **Décodeur**

2. Demi Additionneur

- Le demi additionneur est un circuit combinatoire qui permet de réaliser la somme arithmétique de deux nombres A et B chacun sur un bit.
- A la sotie on va avoir la somme S et la retenu R (Carry).



Pour trouver la structure (le schéma) de ce circuit on doit en premier dresser sa table de vérité

- En binaire l'addition sur un seul bit se fait de la manière suivante:

$$\begin{cases} 0+0 = 00 \\ 0+1 = 01 \\ 1+0 = 01 \\ 1+1 = 10 \end{cases}$$

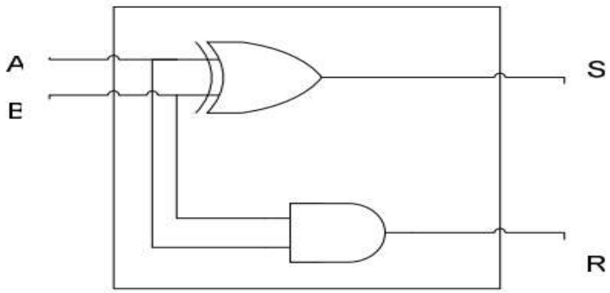
•La table de vérité associée :

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

De la table de vérité on trouve :

$$R = A.B$$
$$S = \overline{A}.B + A.\overline{B} = A \oplus B$$

$$R = A.B$$
$$S = A \oplus B$$



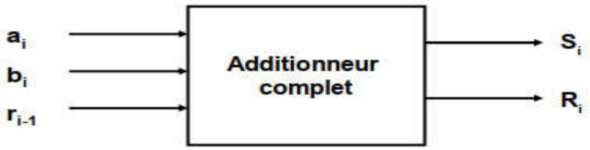
3. L'additionneur complet

- En binaire lorsque on fait une addition il faut tenir en compte de la retenue entrante.

$$\begin{array}{ccccc} r_4 & r_3 & r_2 & r_1 & r_0 = 0 \\ & a_4 & a_3 & a_2 & a_1 \\ + & b_4 & b_3 & b_2 & b_1 \\ \hline r_4 & s_4 & s_3 & s_2 & s_1 \end{array} \qquad \begin{array}{cc} & r_{i-1} \\ & a_i \\ + & b_i \\ \hline r_i & s_i \end{array}$$

3.1 Additionneur complet 1 bit

- L'additionneur complet un bit possède 3 entrées :
  - a_i : le premier nombre sur un bit.
  - b_i : le deuxième nombre sur un bit.
  - r_{i-1} : le retenue entrante sur un bit.
- Il possède deux sorties :
  - S_i : la somme
  - R_i la retenue sortante



8

Table de vérité d'un additionneur complet sur 1 bit

a_i	b_i	r_{i-1}	r_i	s_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$$S_i = \overline{A_i}.\overline{B_i}.R_{i-1} + \overline{A_i}.B_i.\overline{R_{i-1}} + A_i.\overline{B_i}.\overline{R_{i-1}} + A_i.B_i.R_{i-1}$$
$$R_i = \overline{A_i}B_iR_{i-1} + A_i\overline{B_i}R_{i-1} + A_iB_i\overline{R_{i-1}} + A_iB_iR_{i-1}$$

9

Si on veut simplifier les équations on obtient :

$$S_i = \overline{A_i}.\overline{B_i}.R_{i-1} + \overline{A_i}.B_i.\overline{R_{i-1}} + A_i.\overline{B_i}.\overline{R_{i-1}} + A_i.B_i.R_{i-1}$$
$$S_i = \overline{A_i}.(\overline{B_i}.R_{i-1} + B_i.\overline{R_{i-1}}) + A_i.(\overline{B_i}.\overline{R_{i-1}} + B_i.R_{i-1})$$
$$S_i = \overline{A_i}(B_i \oplus R_{i-1}) + A_i.(\overline{B_i \oplus R_{i-1}})$$
$$S_i = A_i \oplus B_i \oplus R_{i-1}$$

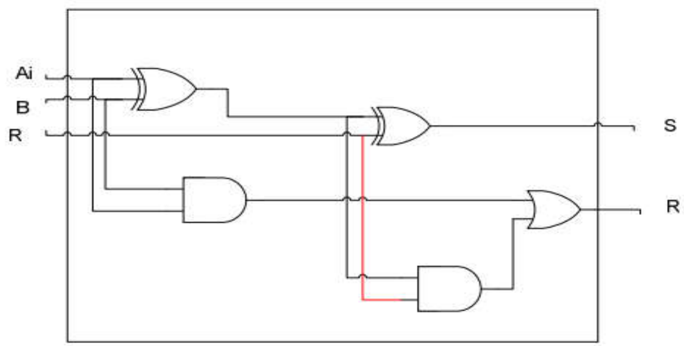
$$R_i = \overline{A_i}B_iR_{i-1} + A_i\overline{B_i}R_{i-1} + A_iB_i\overline{R_{i-1}} + A_iB_iR_{i-1}$$
$$R_i = R_{i-1}.(\overline{A_i}.B_i + A_i.\overline{B_i}) + A_iB_i(\overline{R_{i-1}} +_i R_{i-1})$$
$$R_i = R_{i-1}.(A_i \oplus B_i) + A_iB_i$$

10

3.3 Schéma d'un additionneur complet

$R_i = A_i.B_i + R_{i-1}.(B_i \oplus A_i)$

$S_i = A_i \oplus B_i \oplus R_{i-1}$



11

Rappel Logique Combinatoire

3.4 En utilisant des Demi Additionneurs

$R_i = A_i.B_i + R_{i-1}.(B_i \oplus A_i)$

$S_i = A_i \oplus B_i \oplus R_{i-1}$

Si on pose $X = A_i \oplus B_i$ et $Y = A_i B_i$

On obtient :

$R_i = Y + R_{i-1}.X$

$S_i = X \oplus R_{i-1}$

et si on pose $Z = X \oplus R_{i-1}$ et $T = R_{i-1}.X$

On obtient :

$R_i = Y + T$

$S_i = Z$

- On remarque que X et Y sont les sorties d'un demi additionneur ayant comme entrées A et B
- On remarque que Z et T sont les sorties d'un demi additionneur ayant comme entrées X et R_{i-1}

12

M.DIOP

Rappel Logique Combinatoire

$X = A_i \oplus B_i$

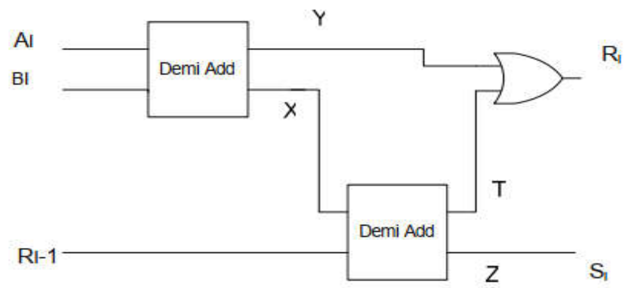
$Y = A_i B_i$

$Z = X \oplus R_{i-1}$

$T = R_{i-1}.X$

$R_i = Y + T$

$S_i = Z$



13

M.DIOP

Rappel Logique Combinatoire

3.4 Additionneur sur 4 bits

- 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 retenu entrante
- En sortie on va avoir le résultat sur 4 bits ainsi que la retenu (5 bits 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 ?

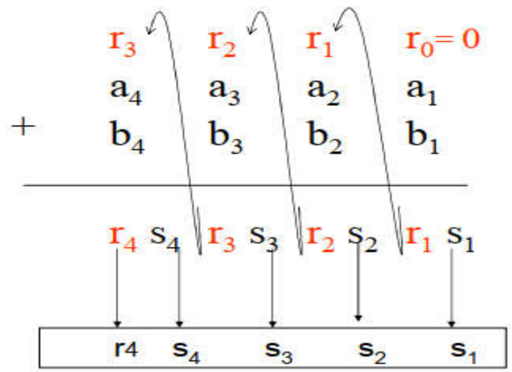
14

M.DIOP

Rappel Logique Combinatoire

- Lorsque on fait l'addition en binaire , on additionne bit par bit en commençant à partir du poids fiable et à chaque fois on propage la retenue sortante au bit du rang supérieur.

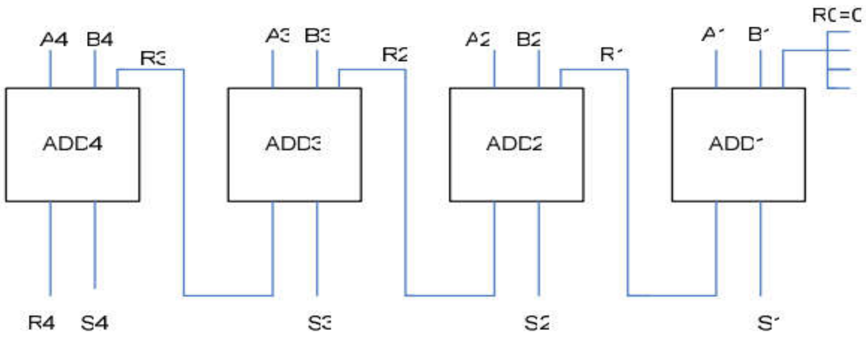
  L'addition sur un bit peut se faire par un additionneur complet sur 1 bits.



Résultat final

M.DIOP

3.4.1 Additionneur 4 bits (schéma)



Rappel Logique Combinatoire

Exercice

- Soit une information binaire sur 5 bits ($i_4i_3i_2i_1i_0$). Donner le circuit qui permet de calculer le nombre de 1 dans l'information en entrée en utilisant uniquement des additionneurs complets sur 1 bit ?
- Exemple :

Si on a en entrée l'information ($i_4i_3i_2i_1i_0$) =(10110) alors en sortie on obtient la valeur 3 en binaire (011) puisque il existe 3 bits qui sont à 1 dans l'information en entrée .

Rappel Logique Combinatoire

4. Le Comparateur

- C'est un circuit combinatoire qui permet de comparer entre deux nombres binaire A et B.
- Il possède 2 entrées :
  - A : sur un bit
  - B : sur un bit
- Il possède 3 sorties
  - fe : égalité (A=B)
  - fi : inférieur (A < B)
  - fs : supérieur (A > B)

Rappel Logique Combinatoire

4.1 Comparateur sur un bit

A	B		fs	fe	fi
0	0		0	1	0
0	1		0	0	1
1	0		1	0	0
1	1		0	1	0

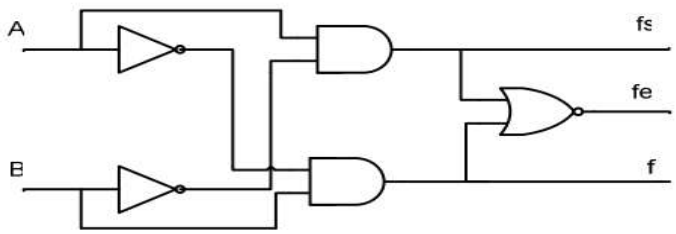
$$fs = A.\overline{B}$$

$$fi = \overline{A}B$$

$$fe = \overline{A}\,\overline{B} + AB = \overline{A \oplus B} = \overline{fs + fi}$$

Schéma d'un comparateur dur un bit

$$fs = A.\overline{B}$$
$$fi = \overline{A}B$$
$$fe = \overline{fs + fi}$$



Rappel Logique Combinatoire

4.2 Comparateur 2 bits

- Il permet de faire la comparaison entre deux nombres A (a_2a_1) et B(b_2b_1) chacun sur deux bits.

Rappel Logique Combinatoire

1. A=B si

A2=B2 et A1=B1

$$fe = (\overline{A2 \oplus B2}).(\overline{A1 \oplus B1})$$

2. A>B si

A2 > B2 ou (A2=B2 et A1>B1)

$$fs = A2.\overline{B2} + (\overline{A2 \oplus B2}).(A1.\overline{B1})$$

3. A<B si

A2 < B2 ou (A2=B2 et A1<B1)

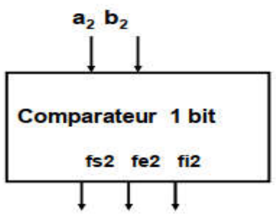
$$fi = \overline{A2}.B2 + (\overline{A2 \oplus B2}).(\overline{A1}.B1)$$

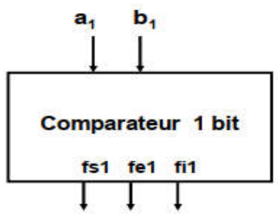
A2	A1	B2	B1	fs	fe	fi
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

Rappel Logique Combinatoire

4.2.2 comparateur 2 bits avec des comparateurs 1 bit

- C'est possible de réaliser un comparateur 2 bits en utilisant des comparateurs 1 bit et des portes logiques.
- Il faut utiliser un comparateur pour comparer les bits du poids faible et un autre pour comparer les bits du poids fort.
- Il faut combiner entre les sorties des deux comparateurs utilisés pour réaliser les sorties du comparateur final.





Rappel Logique Combinatoire

1. A=B si

A2=B2 et A1=B1

$$fe = (\overline{A2 \oplus B2}).(\overline{A1 \oplus B1}) = fe2.fe1$$

2. A>B si

A2 > B2 ou (A2=B2 et A1>B1)

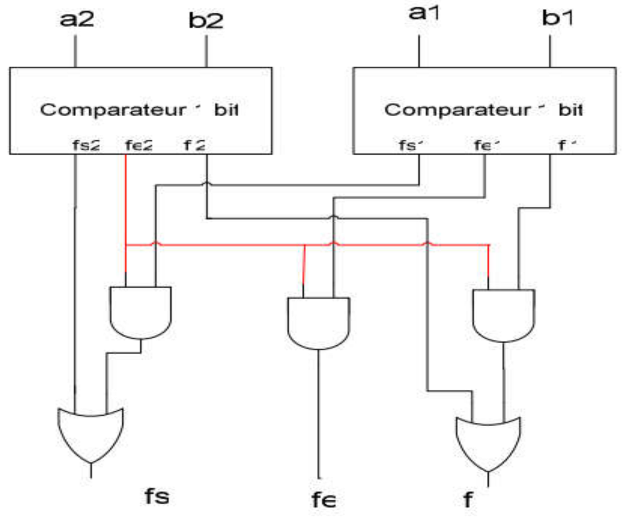
$$fs = A2.\overline{B2} + (\overline{A2 \oplus B2}).(A1.\overline{B1}) = fs2 + fe2.fs1$$

3. A<B si

A2 < B2 ou (A2=B2 et A1<B1)

$$fi = \overline{A2}.B2 + (\overline{A2 \oplus B2}).(\overline{A1}.B1) = fi2 + fe2.fi1$$

Rappel Logique Combinatoire



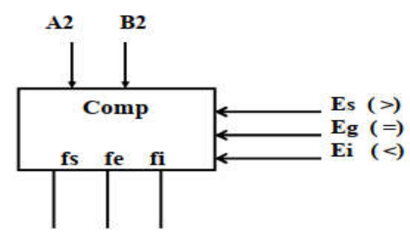
Rappel Logique Combinatoire

4.2.3 Comparateur avec des entrées de mise en cascade

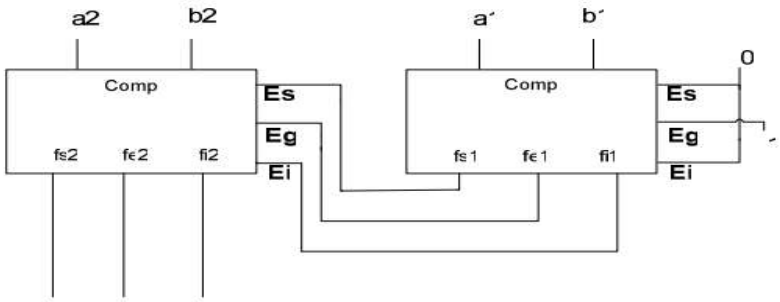
- On remarque que :
  - Si A2 >B2 alors **A > B**
  - Si A2<B2 alors **A < B**
- Par contre si A2=B2 alors il faut tenir en compte du résultat de la comparaison des bits du poids faible.
- Pour cela on rajoute au comparateur des entrées qui nous indiquent le résultat de la comparaison précédente.
- Ces entrées sont appelées des entrées de mise en cascade.

Rappel Logique Combinatoire

A2	B2	Es	Eg	Ei		fs	fe	fs
A2>B2		X	X	X		1	0	0
A2<B2		X	X	X		0	0	1
A2=B1		1	0	0		1	0	0
		0	1	0		0	1	0
		0	0	1		0	0	1



fs= (A2>B2) ou (A2=B2).Es
fi= (A2<B2) ou (A2=B2).Ei
fe=(A2=B2).Eg



Rappel Logique Combinatoire

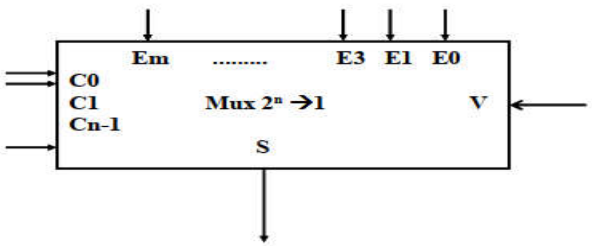
Exercice

- Réaliser un comparateur 4 bits en utilisant des comparateurs 2 bits avec des entrées de mise en cascade?

Rappel Logique Combinatoire

5. Le Multiplexeur

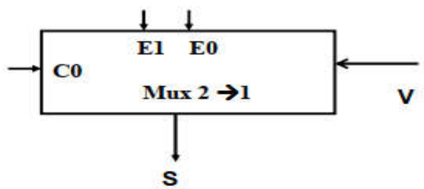
- Un multiplexeur est un circuit combinatoire qui permet de sélectionner une information (1 bit) parmi 2^n valeurs en entrée.
- Il possède :
  - 2^n entrées d'information
  - Une seule sortie
  - N entrées de sélection (commandes)



Rappel Logique Combinatoire

5.1 Multiplexeur 2 →1

V	C_0		S
0	X		0
1	0		E0
1	1		E1

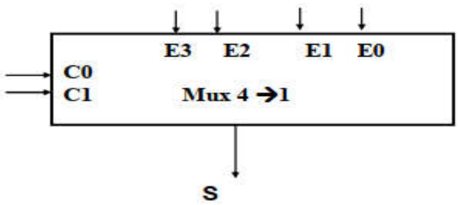


$$S = V.(\overline{C_0}.E0 + C_0.E1)$$

Rappel Logique Combinatoire

5.2 Multiplexeur 4 →1

C1	C0		S
0	0		E0
0	1		E1
1	0		E2
1	1		E3

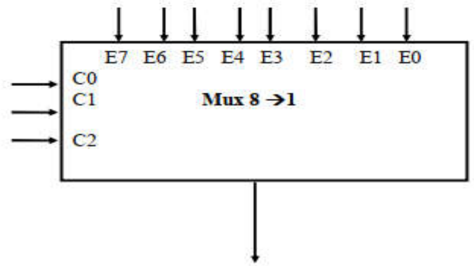


$$S = \overline{C1}.\overline{C0}.(E0) + \overline{C1}.C0.(E1) + C1.\overline{C0}.(E2) + C1.C0.(E3)$$

Rappel Logique Combinatoire

5.3 Multiplexeur 8→1

C2	C1	C0		S
0	0	0		E0
0	0	1		E1
0	1	0		E2
0	1	1		E3
1	0	0		E4
1	0	1		E5
1	1	0		E6
1	1	1		E7



$$S = \overline{C2}.\overline{C1}.\overline{C0}.(E0) + \overline{C2}.\overline{C1}.C0(E1) + \overline{C2}.C1.\overline{C0}(E2) + \overline{C2}.C1.C0(E3) + C2.\overline{C1}.\overline{C0}(E4) + C2.\overline{C1}.C0(E5) + C2.C1.\overline{C0}(E6) + C2.C1.C0(E7)$$

Rappel Logique Combinatoire

Exemple : Réalisation d'un additionneur complet avec des multiplexeurs 8→1

•Nous avons besoin d'utiliser deux multiplexeurs :Le premier pour réaliser la fonction de la somme et l'autres pour donner la retenue.

a_i	b_i	r_{i-1}		r_i
0	0	0		0
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		1

a_i	b_i	r_{i-1}		S_i
0	0	0		0
0	0	1		1
0	1	0		1
0	1	1		0
1	0	0		1
1	0	1		0
1	1	0		0
1	1	1		1

Rappel Logique Combinatoire

Réalisation de la fonction de la somme

$$S_i = \overline{A_i}.\overline{B_i}.\overline{R_{i-1}}(0) + \overline{A_i}.\overline{B_i}.R_{i-1}(1) + \overline{A_i}.B_i.\overline{R_{i-1}}(1) + \overline{A_i}.B_i.R_{i-1}(0) + A_i.\overline{B_i}.\overline{R_{i-1}}(1) + A_i.\overline{B_i}.R_{i-1}(0) + A_i.B_i.\overline{R_{i-1}}(0) + A_i.B_i.R_{i-1}(1)$$

$$S = \overline{C2}.\overline{C1}.\overline{C0}.(E0) + \overline{C2}.\overline{C1}.C0(E1) + \overline{C2}.C1.\overline{C0}(E2) + \overline{C2}.C1.C0(E3) + C2.\overline{C1}.\overline{C0}(E4) + C2.\overline{C1}.C0(E5) + C2.C1.\overline{C0}(E6) + C2.C1.C0(E7)$$

On pose :
$C2=A_i$
$C1=B_i$
$C0=R_{i-1}$
E0=0, E1=1, E2=1, E3=0, E4=1, E5=0, E6=0, E7=1

Réalisation de la fonction de la retenue

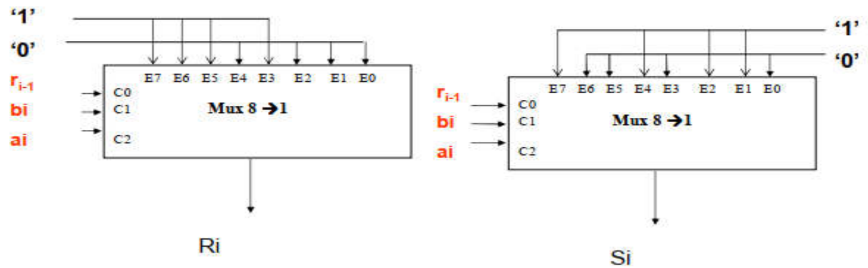
$$R_i = \overline{A_i}\,\overline{B_i}\,\overline{R_{i-1}}.(0) + \overline{A_i}\,\overline{B_i}\,R_{i-1}.(0) + \overline{A_i}\,B_i\,\overline{R_{i-1}}.(0) + \overline{A_i}\,B_i\,R_{i-1}.(1) + A_i\,\overline{B_i}\,\overline{R_{i-1}}.(0) + A_i\,\overline{B_i}\,R_{i-1}.(1)$$
$$+ A_i\,B_i\,\overline{R_{i-1}}.(1) + A_i\,B_i\,R_{i-1}.(1)$$

$$S = \overline{C2}.\overline{C1}.\overline{C0}.(E0) + \overline{C2}.\overline{C1}.C0(E1) + \overline{C2}.C1.\overline{C0}(E2) + \overline{C2}.C1.C0(E3) +$$
$$C2.\overline{C1}.\overline{C0}(E4) + C2.\overline{C1}.C0(E5) + C2.C1.\overline{C0}(E6) + C2.C1.C0(E7)$$

On pose :
$C2=A_i$
$C1=B_i$
$C0=R_{i-1}$
E0=0, E1=0, E2=0, E3=1, E4=0, E5=1, E6=1, E7=1

Rappel Logique Combinatoire

Réalisation d'un additionneur complet avec des multiplexeurs 8→1



Rappel Logique Combinatoire

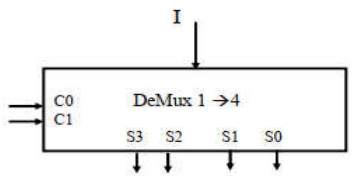
Exercice

- Réaliser le circuit qui permet de trouver le maximum entre deux nombres A et B sur un Bit en utilisant le minimum de portes logiques et de circuits combinatoires?

Rappel Logique Combinatoire

6. Demultiplexeurs

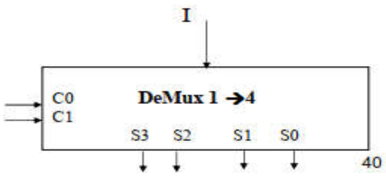
- Il joue le rôle inverse d'un multiplexeurs, il permet de faire passer une information dans l'une des sorties selon les valeurs des entrées de commandes.
- Il possède :
  - une seule entrée
  - 2^n sorties
  - N entrées de sélection (commandes)



6.1 Demultiplexeur 1→4

C1	C0		S3	S2	S1	S0
0	0		0	0	0	i
0	1		0	0	i	0
1	0		0	i	0	0
1	1		i	0	0	0

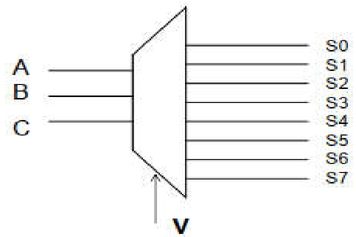
$$S0 = \overline{C1}.\overline{C0}.(I)$$
$$S1 = \overline{C1}.C0.(I)$$
$$S2 = C1.\overline{C0}.(I)$$
$$S3 = C1.C0.(I)$$



Rappel Logique Combinatoire

7. Le décodeur binaire

- C'est un circuit combinatoire qui est constitué de :
  - N : entrées de données
  - 2^n sorties
  - Pour chaque combinaison en entrée une seule sortie est active à la fois

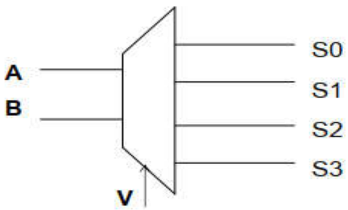


Un décodeur 3→8

Rappel Logique Combinatoire

Décodeur 2→4

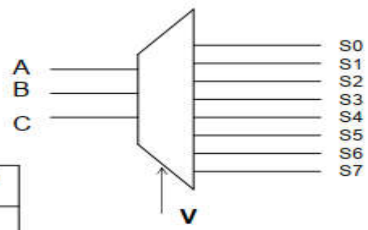
V	A	B		S0	S1	S2	S3
0	X	X		0	0	0	0
1	0	0		1	0	0	0
1	0	1		0	1	0	0
1	1	0		0	0	1	0
1	1	1		0	0	0	1



$$S_0 = (\overline{A}.\overline{B}).V$$
$$S_1 = (\overline{A}.B).V$$
$$S_2 = (A.\overline{B}).V$$
$$S_3 = (A.B).V$$

Rappel Logique Combinatoire

Décodeur 3→8



A	B	C		S0	S1	S2	S3	S4	S5	S6	S7
0	0	0		1	0	0	0	0	0	0	0
0	0	1		0	1	0	0	0	0	0	0
0	1	0		0	0	1	0	0	0	0	0
0	1	1		0	0	0	1	0	0	0	0
1	0	0		0	0	0	0	1	0	0	0
1	0	1		0	0	0	0	0	1	0	0
1	1	0		0	0	0	0	0	0	1	0
1	1	1		0	0	0	0	0	0	0	1

$$S_0 = \overline{A}.\overline{B}.\overline{C}$$
$$S_1 = \overline{A}.\overline{B}.C$$
$$S_2 = \overline{A}.B.\overline{C}$$
$$S_3 = \overline{A}.B.C$$
$$S_4 = A.\overline{B}.\overline{C}$$
$$S_5 = A.\overline{B}.C$$
$$S_6 = A.B.\overline{C}$$
$$S_7 = A.B.C$$

Réalisation d'un additionneur complet avec des décodeurs binaire 3→8

$$S_i = \overline{A_i}.\overline{B_i}.R_{i-1} + \overline{A_i}.B_i.\overline{R_{i-1}} + A_i.\overline{B_i}.\overline{R_{i-1}} + A_i.B_i.R_{i-1}$$

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

$$R_i = \overline{A_i}B_iR_{i-1} + A_i\overline{B_i}R_{i-1} + A_iB_i\overline{R_{i-1}}. + A_iB_iR_{i-1}$$

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

On pose $A=A_i$, $B=B_i$, $C=R_{i-1}$

$$S_0 = \overline{A}.\overline{B}.\overline{C}, S_1 = \overline{A}.\overline{B}.C, S_2 = \overline{A}.B.\overline{C}, S_3 = \overline{A}.B.C,$$

$$S_4 = A.\overline{B}.\overline{C}, S_5 = A.\overline{B}.C, S_6 = A.B.\overline{C}, S_7 = A.B.C$$

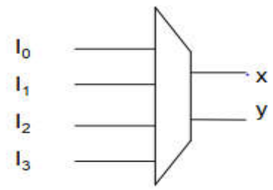
$$R_i = S3 + S5 + S6 + S7$$

$$S_i = S1 + S2 + S4 + S7$$

8. L'encodeur binaire

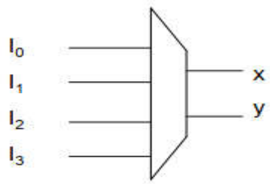
- Il joue le rôle inverse d'un décodeur
  - Il possède 2^n entrées
  - N sortie
  - Pour chaque combinaison en entrée on va avoir sont numéro (en binaire) à la sortie.

Encodeur 4→2



L'encodeur binaire (4→2)

I_0	I_1	I_2	I_3		x	y
0	0	0	0		0	0
1	x	x	x		0	0
0	1	x	x		0	1
0	0	1	x		1	0
0	0	0	1		1	1

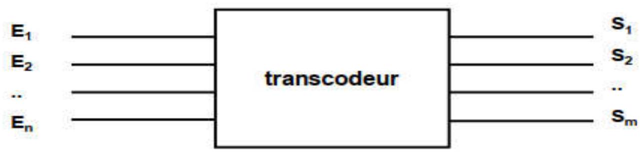


$$X = \overline{I0}.\overline{I1}.(I2 + I3)$$

$$Y = \overline{I0}.(I1 + \overline{I2}.I3)$$

9. Le transcodeur

- C'est un circuit combinatoire qui permet de transformer un code X (sur n bits) en entrée en un code Y (sur m bits) en sortie.



Rappel Logique Combinatoire

Exemple : Transcodeur BCD/EXESS3

A	B	C	D	X	Y	Z	T
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	x	x	x	x
1	0	1	1	x	x	x	x
1	1	0	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	x	x	x	x
1	1	1	1	x	x	x	x

48