
Conception de circuits combinatoires

Daniel Etiemble
de@lri.fr

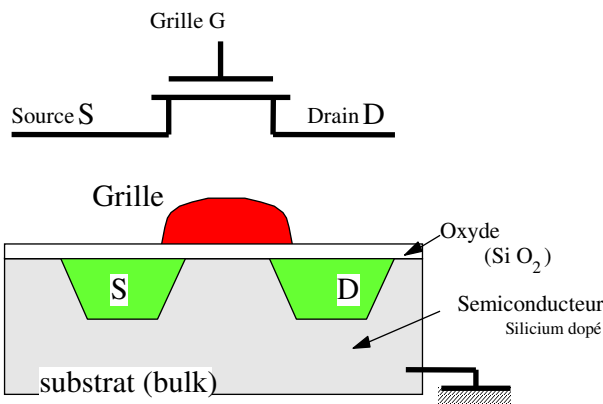
Les différentes approches

- Support : la technologie CMOS
- La logique aléatoires : les portes de base
 - Bibliothèque de portes
 - Bibliothèque de cellules
- La logique régulière
 - multiplexeurs
 - décodeurs
- La logique programmable à deux niveaux
 - PALs
 - PLAs
 - ROMs
- La logique programmable avec mémoires SRAM

Portes et technologies CMOS

- Technologie CMOS et réalisation des fonctions logiques
 - Portes de base
 - Portes complexes
 - Portes de transmission
- Caractéristiques électriques
 - Entrances et sortances
 - Caractéristiques dynamiques

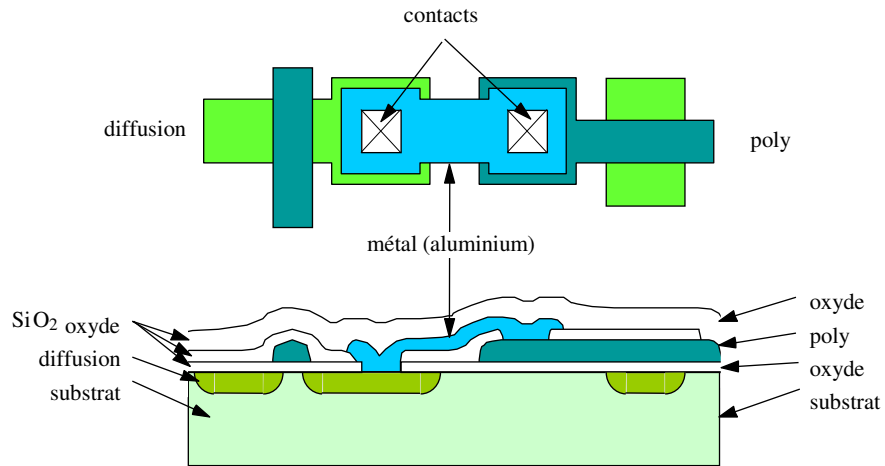
La technologie MOS



- Au début (RCA 1962) la grille était en Aluminium d'où le nom MOS : Métal/Oxyde/Semiconducteur
- Le MOS est parfaitement symétrique et on appelle SOURCE (d'électrons) le côté le plus négatif (*le plus positif pour les PMOS*)
- Le substrat est mis à la masse (à V_{dd} pour les PMOS)

Coupe d'un transistor

Un circuit intégré est une superposition de couches, semi-conductrices, conductrices ou isolante



M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

5

Equations (simplifiées) du transistor MOS

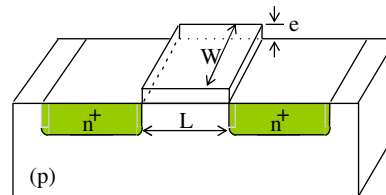
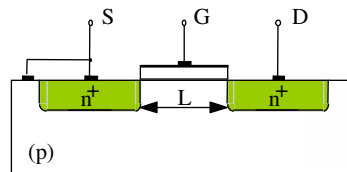
Si $V_{gs} - V_t < 0$, mode bloqué : $I_{ds} = 0$

Si $0 < V_{ds} < V_{gs} - V_t$, mode linéaire

$$I_{ds} = \mu \frac{Cox}{2} \frac{W}{L} (2(V_{gs} - V_t)V_{ds} - V_{ds}^2)$$

Si $0 < V_{gs} - V_t < V_{ds}$, mode saturé

$$I_{ds} = \mu \frac{Cox}{2} \frac{W}{L} (V_{gs} - V_t)^2$$

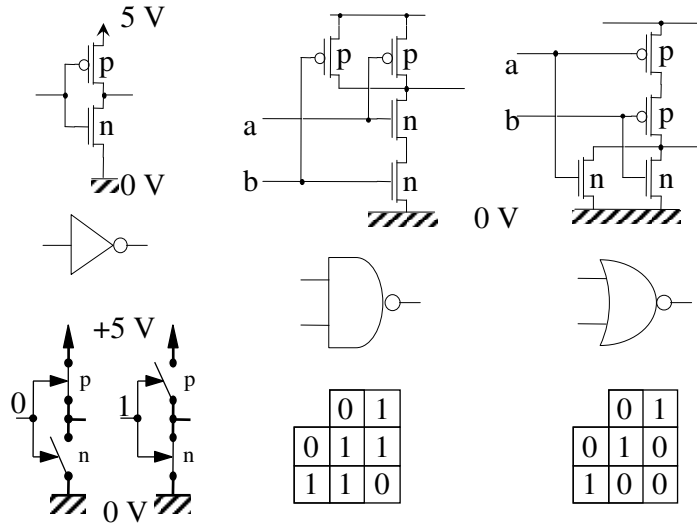


M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

6

Portes élémentaires en CMOS

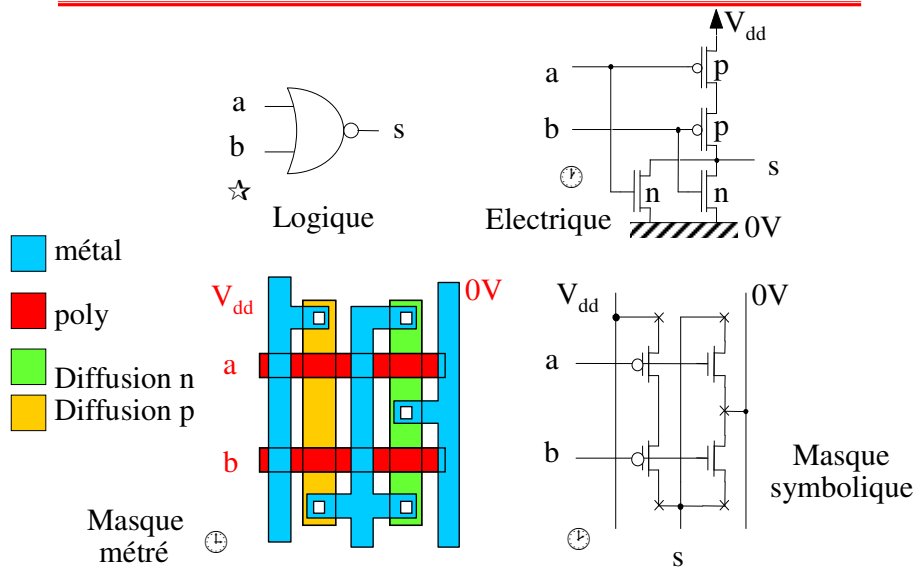


M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

7

4 vues d'une fonction logique

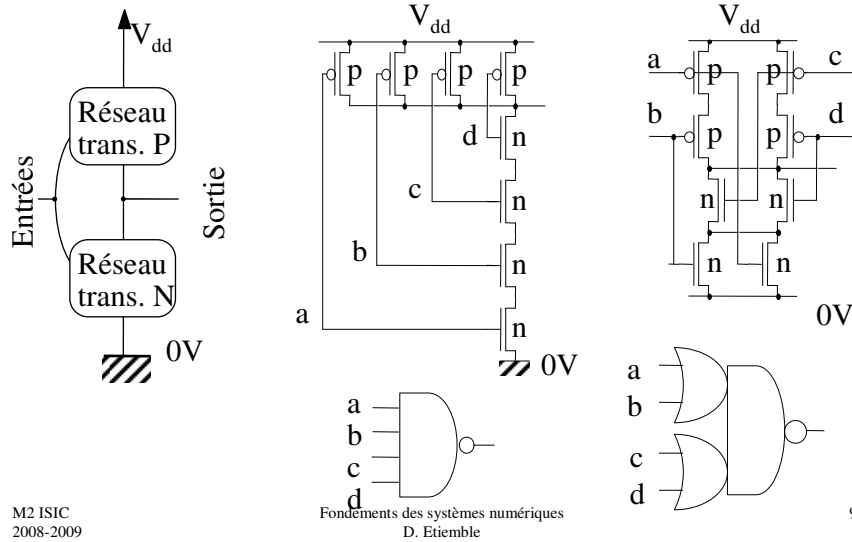


M2 ISIC
2008-2009

S Fondements des systèmes numériques
D. Etiemble

8

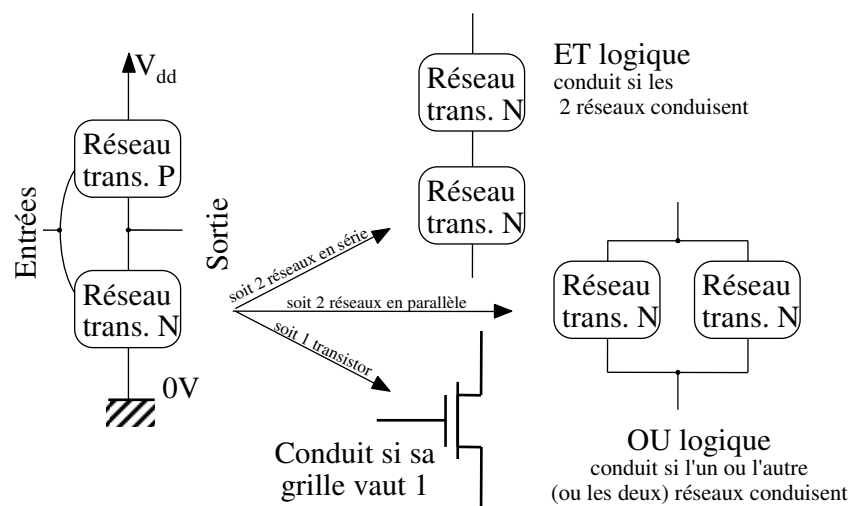
Portes logiques un peu plus complexes



M2 ISIC
2008-2009

9

Réseau de transistors

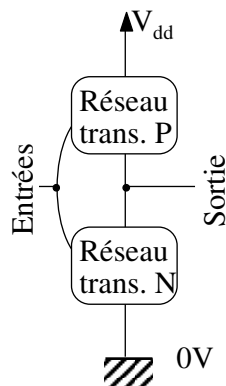


M2 ISIC
2008-2009

10

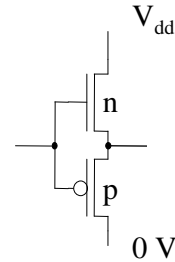
Conception des portes complexes (1)

Equation logique \Rightarrow schéma électrique



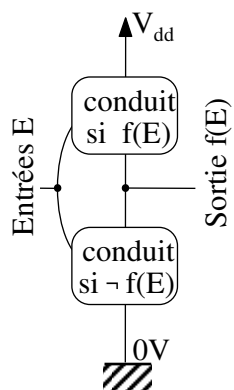
Les transistors P sont utilisés pour tirer à 1 et les transistors N pour tirer à 0. Il n'y a pas de perte de seuil

En conséquence les fonctions réalisables sont des fonctions DECROISSANTES des entrées

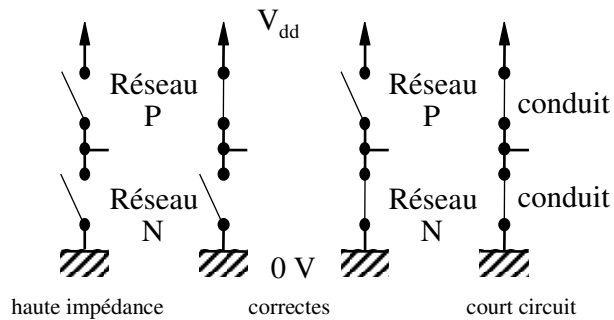


Fonction identité qui ne marche pas

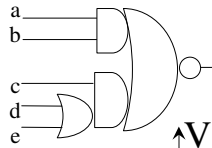
Conception des portes complexes (2)



Un et un seul des deux réseaux N et P conduit à chaque instant. Les réseaux sont logiquement complémentaires. Comme l'un est en transistors P et l'autre en transistors N, les réseaux N et P sont duaux. Les deux réseaux ont les mêmes entrées et le même nombre de transistors.

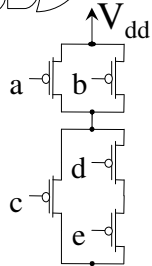


Conception des portes complexes (3)



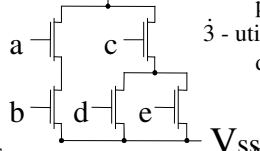
Pour construire une porte complexe, on construit en premier le réseau de transistors N, avec les règles :

ET - réseaux en série
OU - réseaux en parallèle.



Pour construire le réseau de transistors P on peut procéder de 3 façons :

- 1 - croiser les règles ci-dessous
- 2 - utiliser l'algèbre de Boole pour compléter la fonction et procéder comme pour les N
- 3 - utiliser une méthode graphique pour tracer le dual du graphe N



M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

13

Construction du dual

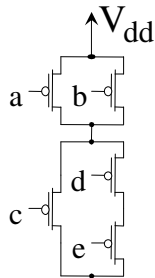
Méthode 1: croiser les règles

Règle pour le réseau N

ET - transistors série
OU - transistors parallèles.

Règle pour le réseau P

ET - transistors parallèle
OU - transistors série



Méthode 2: compléter

Equation pour le réseau N $F = (a \wedge b) \vee c \wedge (d \vee e)$

Equation pour le réseau P $F = (\bar{a} \vee \bar{b}) \wedge (\bar{c} \vee \bar{d} \wedge \bar{e})$

Se souvenir que N tire à la masse, donc l'équation doit être complémentée, et P conduit pour un 0 donc les variables doivent être complémentées

M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

14

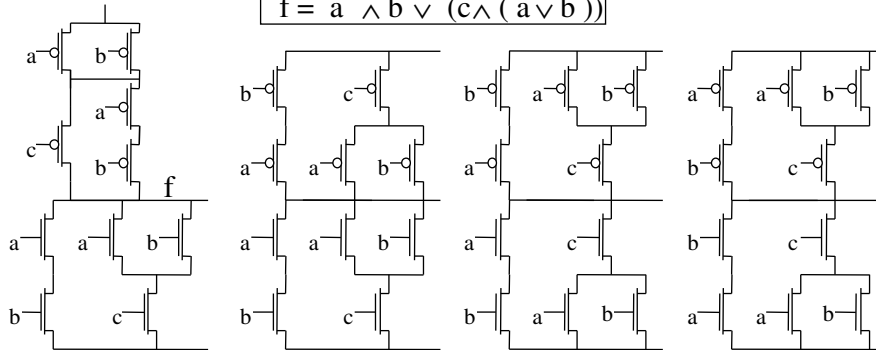
Portes complexes : améliorations électriques

Raccourcir les chemins
entre sortie et alimentation

Minimiser la capacité
parasite de sortie

Mettre plus près de la sortie les
transistors activés le plus tard

$$f = \overline{a \wedge b \vee (c \wedge (a \vee b))}$$

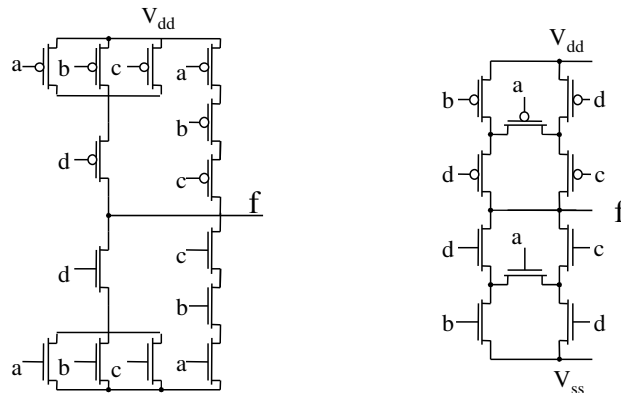


M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

15

Variantes d'une fonction



$$F' = abc + d(a+b+c)$$

$$\text{Si } a = 0, F' = db + dc$$

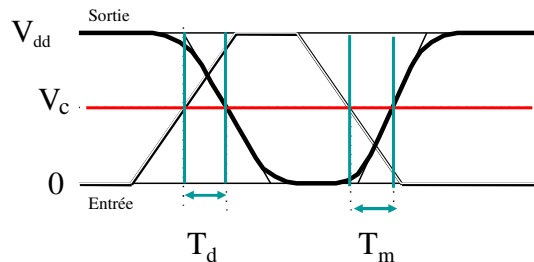
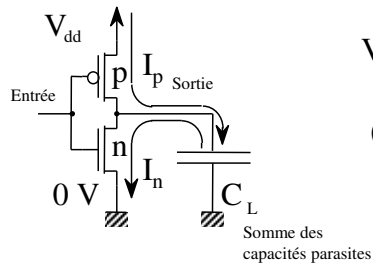
$$\text{Si } a = 1, F' = bc + d = bc + dc + db + dd = (b+d)(c+d)$$

M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

16

Modélisation du comportement dynamique

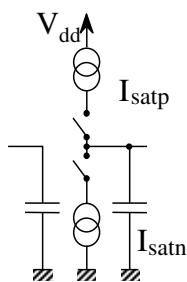


$$T_d = V_c \frac{C_L}{I_{satn}}$$

$$T_m = V_c \frac{C_L}{I_{satp}}$$

L'inverseur est un générateur de courant déclenché par le passage du seuil

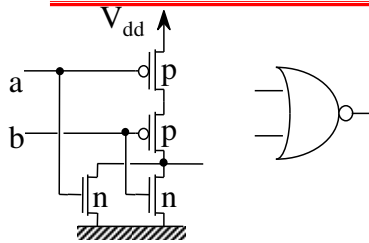
Courants et dimensions



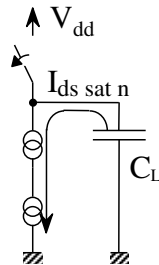
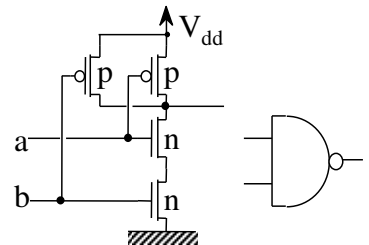
$$I_{satp} = K_p \frac{W_p}{L_p} (V_{dd} - V_{tp} - V_c)^2$$

$$I_{satn} = K_n \frac{W_n}{L_n} (V_{dd} - V_{tn} - V_c)^2$$

Conservation des temps de commutation

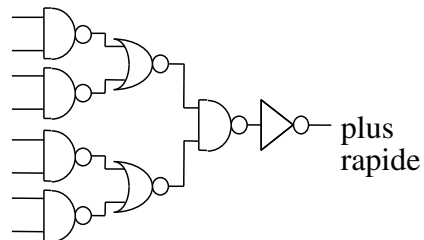
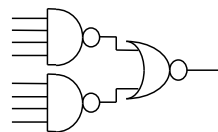
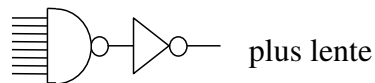


Pour conserver le W/L équivalent, si l'on met n transistors en série, il faut multiplier W par n
Impact sur les capacités d'entrée des portes Nand/Nor à n entrées

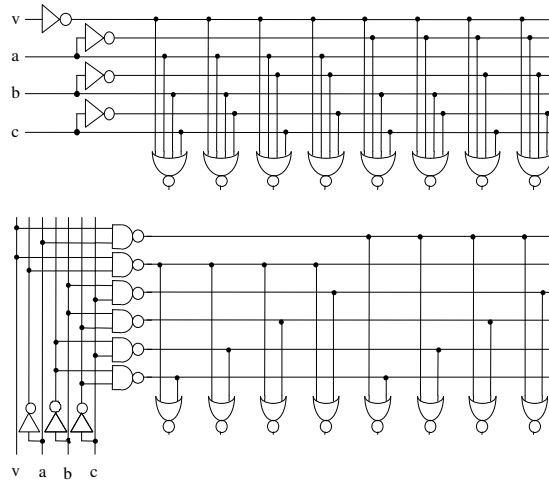


début de décharge

Entrances et temps de retard



Quel est le circuit le plus rapide ?

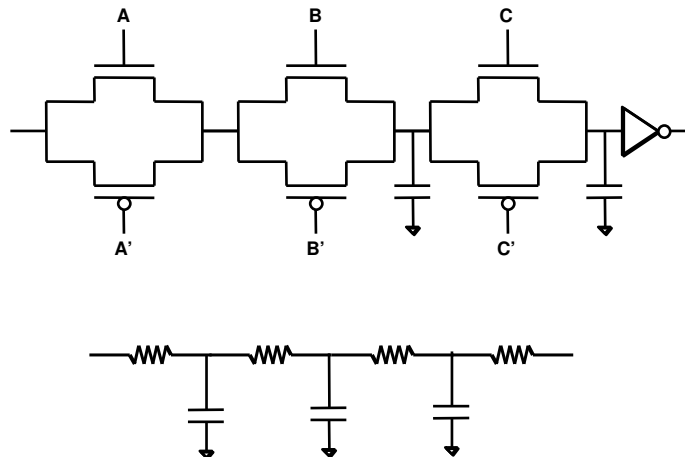


M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

21

Portes de transmission en série



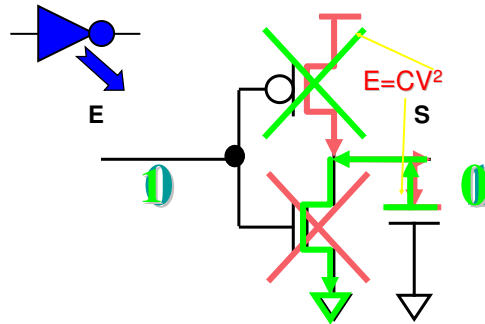
M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

22

Consommation d'énergie

- **Consommation d'énergie**
 - Quand une capacité est chargée ou déchargée
 - Une capacité chargée est un 1 logique, déchargée est un 0 logique



- Les capacités peuvent être les grilles d'autres transistors ou des fils (bus et interconnexion longues)

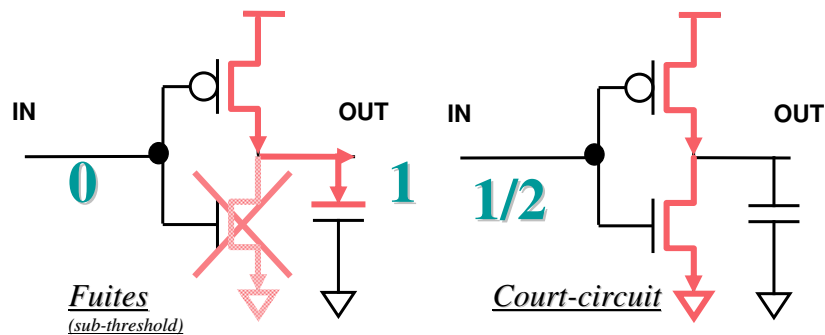
M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

23

Consommation d'énergie

- Des effets secondaires comme les courants de fuite et de court-circuits augmentent avec les technologies modernes



M2 ISIC
2008-2009

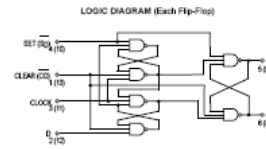
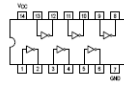
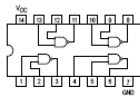
Fondements des systèmes numériques
D. Etiemble

24

La logique aléatoire



- Transistors intégrés dans des portes logiques (années 60)
- Catalogue de portes de base (années 70)
 - Texas Instruments Logic Data Book
 - Tous les boîtiers de base fournis et caractérisés (délai, puissance)
 - Boîtiers typiques
 - CI 14 pattes : 6 inverseurs, 4 Nand à 2 entrées, 4 portes Ou exclusif
- Très peu utilisés aujourd'hui
- Cependant, des bibliothèques utilisant ces "éléments" existent pour la conception de circuits
 - Les concepteurs utilisent des portes logiques déjà caractérisées

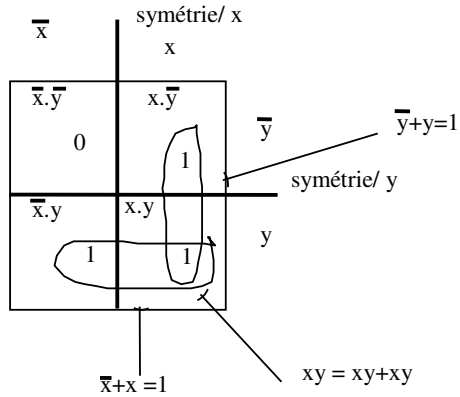


Synthèse en logique aléatoire

- Exprimer une fonction logique (table de vérité) sous forme d'une expression logique
 - Disjonctive normale (Ou de ET)
 - Nand de Nands
 - Conjonctive normale (ET de OU)
 - Nor de Nors
- Simplification
 - Diagrammes de Karnaugh
- Implantation sous forme de portes
 - ET, OU, Inverseurs
 - Nands, Nors

Diagramme de Karnaugh

- Représentation graphique des tables de vérité faisant apparaître les symétries

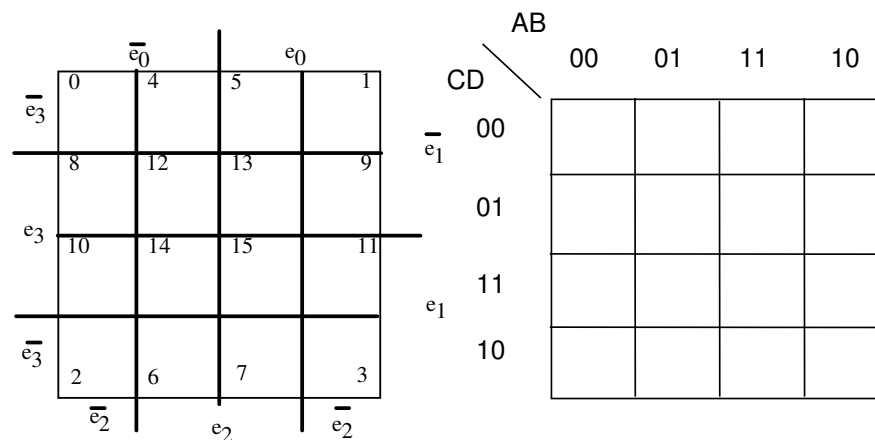


M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

27

Diagrammes de Karnaugh (4 entrées)



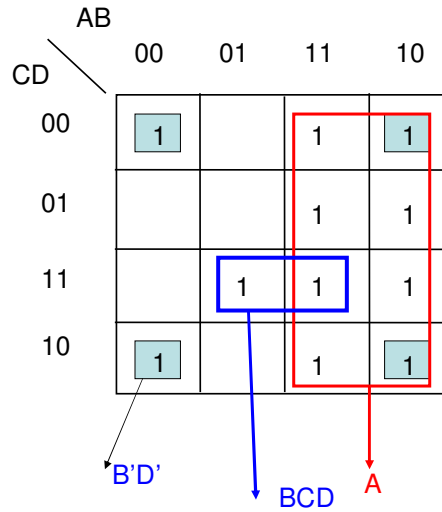
M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

28

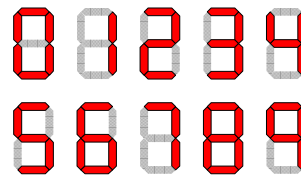
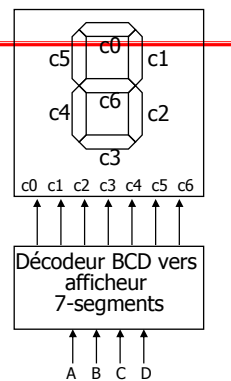
Simplification

- Regroupements de taille maximale
 - 8 supprime 3 variables
 - 4 supprime 2 variables
 - 2 supprime 1 variable
- Tous les 1 doivent être pris une fois
- Pas de regroupements redondants



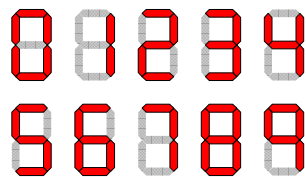
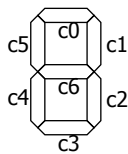
Afficheur 7 segments

- Entrées
 - Chiffre décimal codé sur 4 bits (A, B, C, D)
- Sorties
 - Signaux de contrôle pour l'afficheur (7 sorties C_0 à C_6)



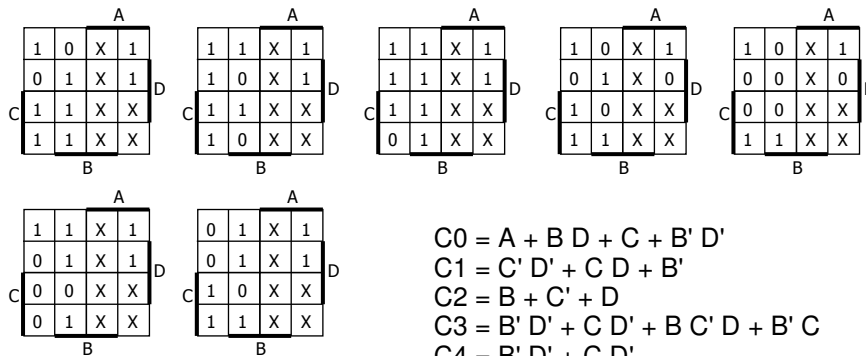
Synthèse avec logique aléatoire

- Table de vérité avec cas indifférents
- Minimisation avec diagramme de Karnaugh



A	B	C	D	C0	C1	C2	C3	C4	C5	C6
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	0	0	1	1
1	0	1	-	-	-	-	-	-	-	-
1	1	-	-	-	-	-	-	-	-	-

Implémentation sous forme somme de produits



$$\begin{aligned}
 C0 &= A + B D + C + B' D' \\
 C1 &= C' D' + C D + B' \\
 C2 &= B + C' + D \\
 C3 &= B' D' + C D' + B C' D + B' C \\
 C4 &= B' D' + C D' \\
 C5 &= A + C' D' + B D' + B C' \\
 C6 &= A + C D' + B C' + B' C
 \end{aligned}$$

Si minimisation indépendante de chaque fonction, 15 termes produit

Implémentation sous forme OU de ET (suite)

- On peut faire mieux
 - 9 termes produit (au lieu de 15)
 - On partage des termes entre les sorties
 - Chaque sortie n'est pas nécessairement sous forme minimale

	A			
C2	1	1	X	1
	1	1	X	1
C	1	1	X	X
0	1	X	X	
	B			

$$\begin{aligned}
 C0 &= A + B D + C + B' D' \\
 C1 &= C' D' + C D + B' \\
 C2 &= B + C' + D \\
 C3 &= B' D' + C D' + B C' D + B' C \\
 C4 &= B' D' + C D' \\
 C5 &= A + C' D' + B D' + B C' \\
 C6 &= A + C D' + B C' + B' C
 \end{aligned}$$

	A			
C2	1	1	X	1
	1	1	X	1
C	1	1	X	X
0	1	X	X	
	B			

$$\begin{aligned}
 C0 &= B C' D + C D + B' D' + B C D' + A \\
 C1 &= B' D + C' D' + C D + B' D' \\
 C2 &= B' D + B C' D + C' D' + C D + B C D' \\
 C3 &= B C' D + B' D + B' D' + B C D' \\
 C4 &= B' D' + B C D' \\
 C5 &= B C' D + C' D' + A + B C D' \\
 C6 &= B' C + B C' + B C D' + A
 \end{aligned}$$

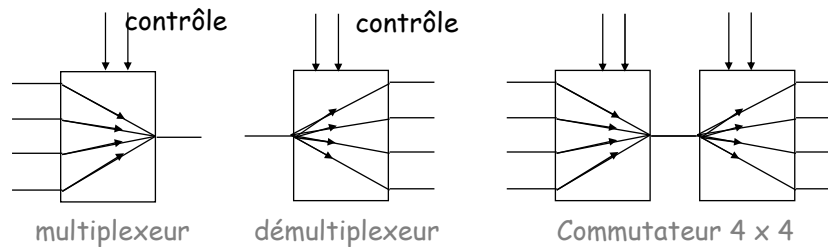
Logique aléatoire

- Difficile de savoir exactement quelles portes utiliser
 - Correspondance logique vers réseau de portes NAND/NOR
 - Déterminer le nombre minimal de boîtiers
 - Des modifications légères de la fonction peuvent diminuer le coût
- Les modifications sont difficiles à réaliser
 - On doit changer les connexions
 - On peut avoir besoin de nouvelles parties
 - Conception avec des portes de secours (quelques inverseurs et portes supplémentaires sur chaque carte)
- La conception doit être plus rapide
- On doit rendre les modifications plus faciles

 Logique régulière

Multiplexeurs - démultiplexeurs

- Connexions directes point à point entre des portes
 - fils
- Une entrée parmi plusieurs vers une sortie --- MUX(multiplexeur)
- Une entrée vers une parmi plusieurs sorties --- DEMUX(démultiplexeur)



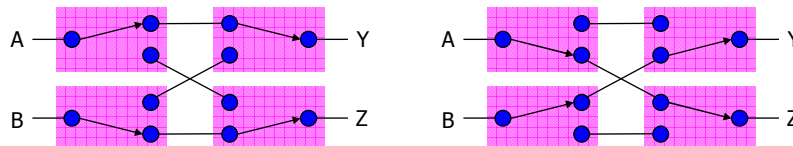
M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

35

Mux et démux

- Implémentation de commutateurs à l'aide de mux et démux.
 - Peuvent être combinés pour créer des réseaux de commutation de taille quelconque
 - Utilisés pour implémenter des interconnexions plusieurs entrées - plusieurs sorties



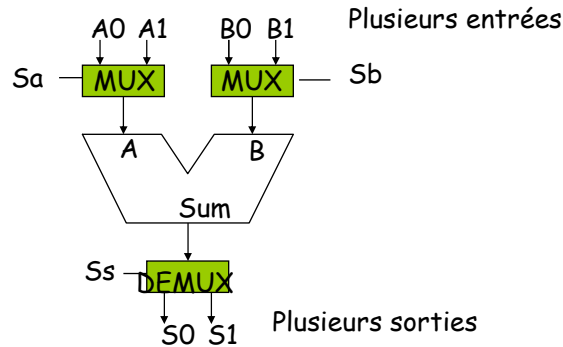
M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

36

Mux et démux (suite)

- Utilisation de multiplexeurs/démultiplexeurs dans des connexions multi-points



M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

37

Multiplexeurs - Sélecteurs

2^n entrées de données, n entrées de contrôle, 1 sortie pour
Connecter 2^n points à un seul point. Le signal de contrôle
est la valeur du numéro de l'entrée connectée à la sortie

$$Z = A' E_0 + A E_1$$

Forme fonctionnelle

Forme logique

Deux manières
d'écrire la table de
vérité d'un MUX 2:1

A	Z
0	E_0
1	E_1

E_0	E_1	A	Z
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

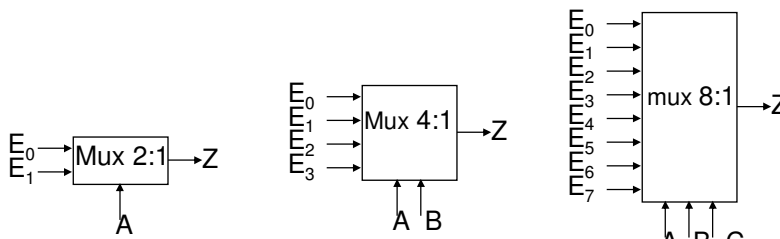
M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

38

Multiplexeurs - sélecteurs (suite)

- Mux 2:1 : $Z = A'E_0 + AE_1$
- Mux 4:1 : $Z = A'B'E_0 + A'BE_1 + AB'E_2 + ABE_3$
- Mux 8:1 : $Z = A'B'C'E_0 + A'B'CE_1 + A'BC'E_2 + A'BCE_3 + AB'C'E_4 + AB'CE_5 + ABC'E_6 + ABCE_7$
- En général : $Z = \sum_{k=0}^{2^n-1} m_k E_k$

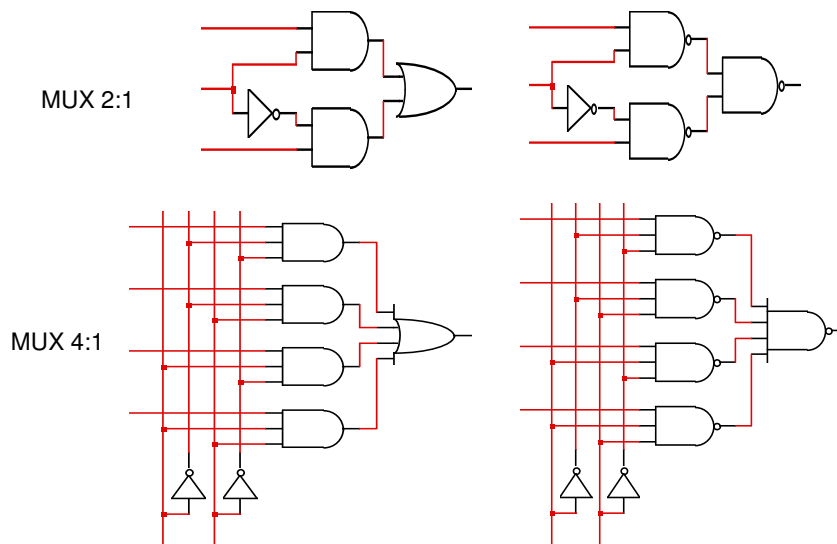


M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

39

Implantation des mux avec des portes



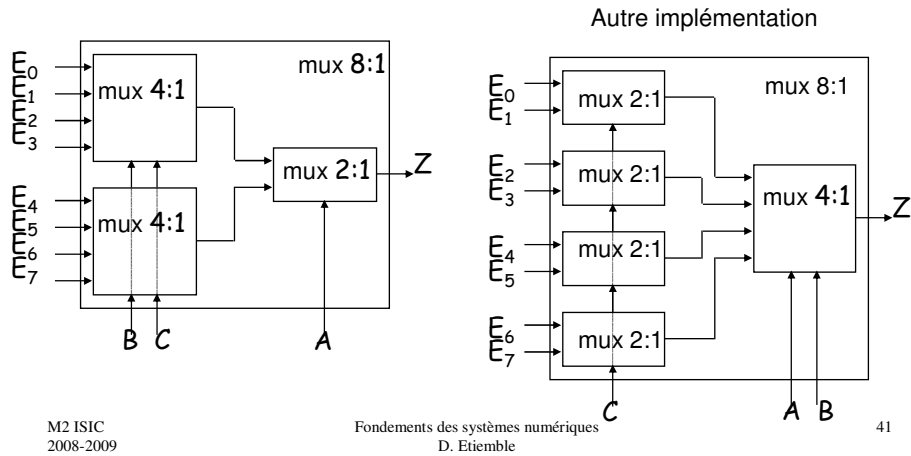
M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

40

Combinaison de multiplexeurs

- De gros multiplexeurs peuvent être réalisés à partir de multiplexeurs plus petits.

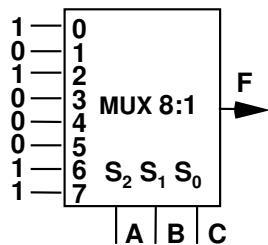


Multiplexeur/Sélecteur : bloc logique d'usage général

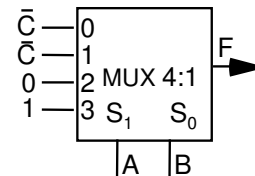
Un multiplexeur $2^{n-1}:1$ peut implanter toute fonction de n variables. $n-1$ variables de contrôle, la variable restant est une donnée d'entrée du multiplexeur

Exemple:

$$\begin{aligned}
 F(A,B,C) &= m_0 + m_2 + m_6 + m_7 \\
 &= A'B'C' + A'BC' + ABC' + ABC \\
 &= A'B'(C') + A'B(C') + AB'(0) + AB(1)
 \end{aligned}$$



A	B	C	F
0	0	0	1 C'
0	0	1	0
0	1	0	1 C
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1



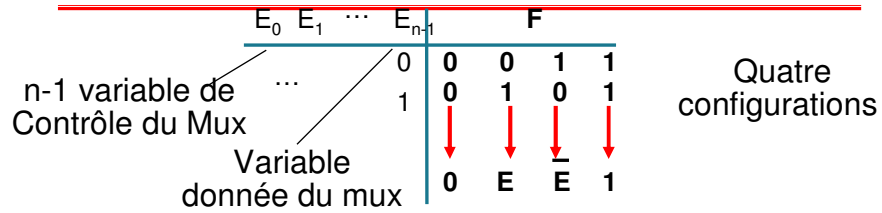
"LUT"

M2 ISIC
2008-2009

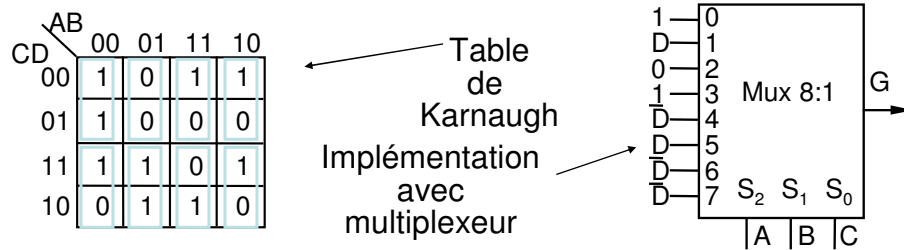
Fondements des systèmes numériques
D. Etiemble

42

Multiplexeur/Sélecteur : généralisation



Exemple:
 $G(A,B,C,D)$ peut être implémenté par un Mux 8:1:



M2 ISIC
 2008-2009

Fondements des systèmes numériques
 D. Etiemble

43

Exemple

- Réaliser la fonction $F = B'CD' + ABC'$ avec un multiplexeur 4:1 et un minimum d'autres portes
- Implémenter la fonction logique
 $f(A,B,C,D) = m(0,1,5,7,11,13,14,15)$
 - (a) Avec un Mux 4:1
 - (b) Avec un Mux 8:1

M2 ISIC
 2008-2009

Fondements des systèmes numériques
 D. Etiemble

44

Décodeurs/Démultiplexeurs

Décodeur : une entrée de données, n entrées de contrôle, 2^n sorties
 Les entrées de contrôles (S) représentent en binaire le numéro de la sortie à laquelle l'entrée est connectée.

L'entrée de donnée est habituellement appelée "enable" (G)

Décodeur 1:2 :

$$O_0 = G \cdot \overline{S}; \quad O_1 = G \cdot S$$

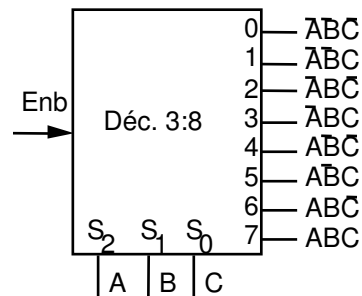
Décodeur 2:4 :

$$\begin{aligned} O_0 &= G \cdot \overline{S_0} \cdot \overline{S_1} \\ O_1 &= G \cdot \overline{S_0} \cdot S_1 \\ O_2 &= G \cdot S_0 \cdot \overline{S_1} \\ O_3 &= G \cdot S_0 \cdot S_1 \end{aligned}$$

Décodeur 3:8 :

$$\begin{aligned} O_0 &= G \cdot \overline{S_0} \cdot \overline{S_1} \cdot \overline{S_2} \\ O_1 &= G \cdot \overline{S_0} \cdot \overline{S_1} \cdot S_2 \\ O_2 &= G \cdot \overline{S_0} \cdot S_1 \cdot \overline{S_2} \\ O_3 &= G \cdot \overline{S_0} \cdot S_1 \cdot S_2 \\ O_4 &= G \cdot S_0 \cdot \overline{S_1} \cdot \overline{S_2} \\ O_5 &= G \cdot S_0 \cdot \overline{S_1} \cdot S_2 \\ O_6 &= G \cdot S_0 \cdot S_1 \cdot \overline{S_2} \\ O_7 &= G \cdot S_0 \cdot S_1 \cdot S_2 \end{aligned}$$

Décodeur/Démultiplexeur : bloc logique de base



Le décodeur génère les termes produit appropriés en fonction des signaux de contrôle

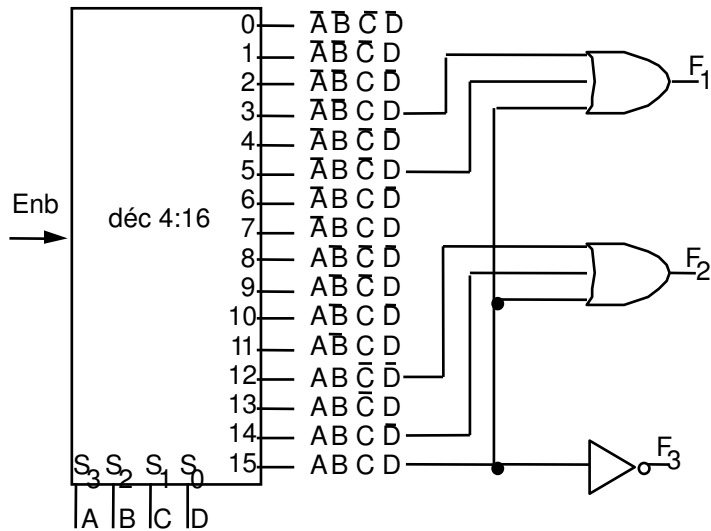
Exemples:

$$F1 = A' B C' D + A' B' C D + A B C D$$

$$F2 = A B C' D' + A B C$$

$$F3 = (A' + B' + C' + D')$$

Décodeur/Démultiplexeur

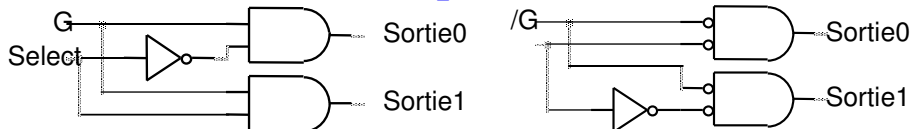


M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

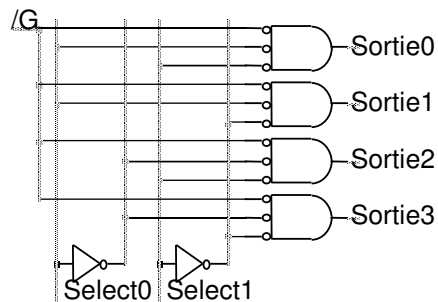
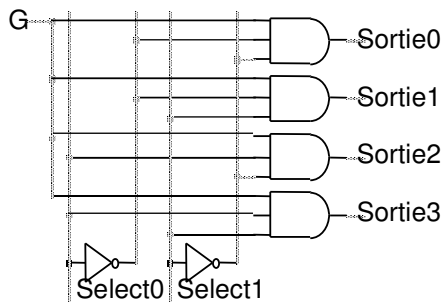
47

Décodeurs/Démultiplexeurs : Autre implémentation



1:2 Décodeur, Enable actif haut

1:2 Décodeur, Enable actif bas



2:4 Décodeur, Enable actif haut

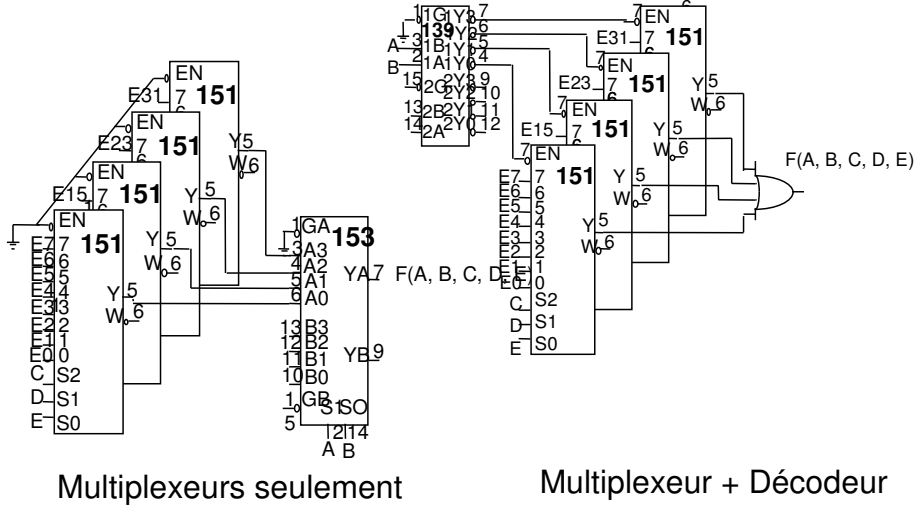
2:4 Décodeur, Enable actif bas

M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

48

Multiplexeurs/Décodeurs : Mux 32:1

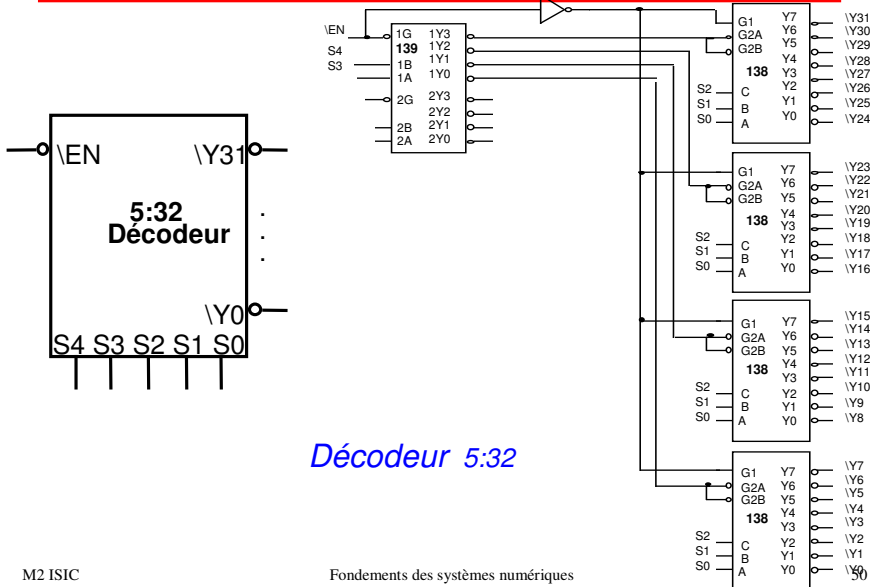


M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

49

Multiplexeurs/Décodeurs

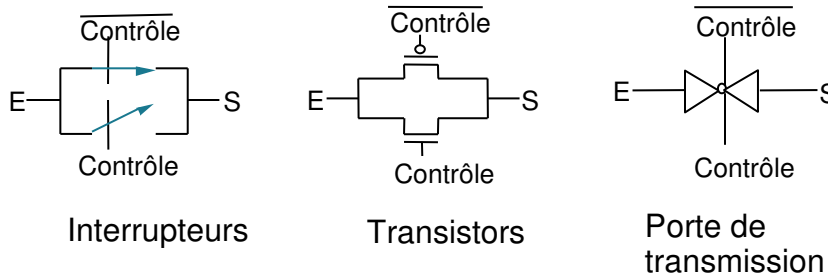


M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

Logique à portes de transmission CMOS

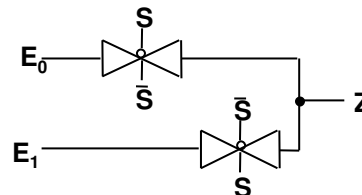
Les transistors nMOS passent bien les 0, mal les 1
 Les transistors pMOS passent bien les 1, mal les 0
 Les portes de transmission "parfaites" les placent en parallèle



Fonctions multiplexeur/démultiplexeur

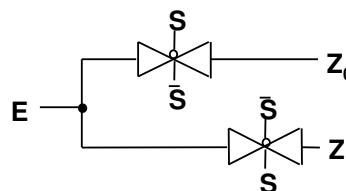
Multiplexeur

Choisit E_0 si $S = 0$
 Choisit E_1 si $S = 1$



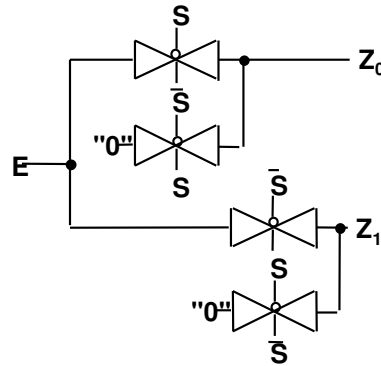
Démultiplexeur

E vers Z_0 si $S = 0$
 E vers Z_1 si $S = 1$



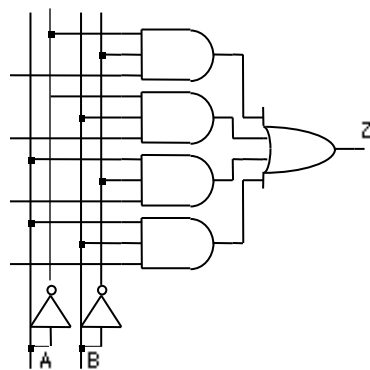
Portes de transmission : sorties non fixées

Problème avec le démultiplexeur : plusieurs sorties, mais une seule connectée à l'entrée

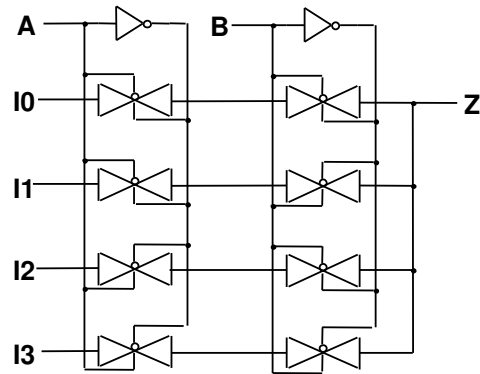


Logique supplémentaire pour que chaque sortie soit fixée
Pas de sorties "flottantes"

Multiplexeurs/Sélecteurs

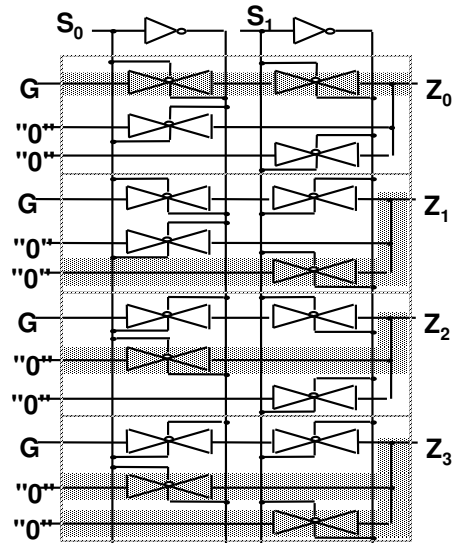


Mux 4:1 avec portes
36 transistors



Mux 4:1 avec portes de transmission
20 transistors

Décodeurs/Démultiplexeurs



Décodeur 4:2

$S_0 = 0, S_1 = 0$
 1 chemin direct
 3 chemins "diagonaux"

État « haute impédance » (Tri-state)

États logiques : "0", "1", État indifférent, inconnu : "X"

3^{ème} état : "Z" ? **Haute impédance, résistance infinie, pas de connexion**

Portes tri-state: les valeurs de sortie sont "0", "1", et "Z"
 entrée supplémentaire: output enable (OE)

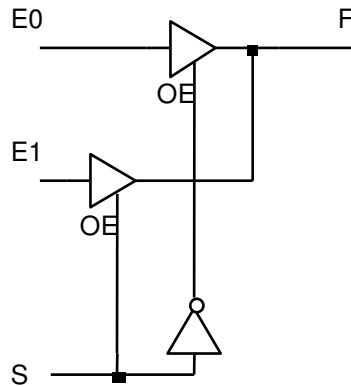
A	OE	F
X	0	Z
0	1	0
1	1	1

Quand OE est bas, la porte est "déconnectée" de la sortie

État « haute impédance » (Tri-state)

Utilisation des portes trois états pour les connexions bus

L'état haute impédance permet de connecter plusieurs portes à la même sortie, à condition qu'une seule porte soit active (OE actif) *en même temps*



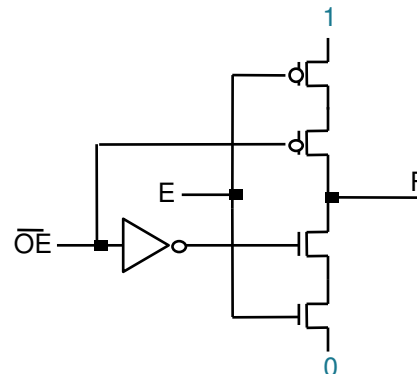
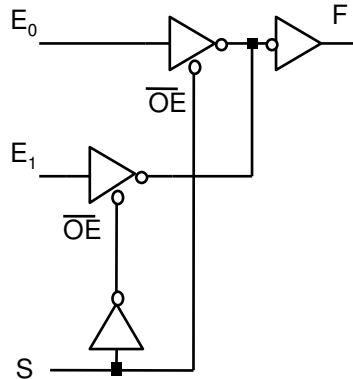
M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

57

État « haute impédance » (Tri-state)

OE actif à l'état bas + buffer
Inverseur 3 états



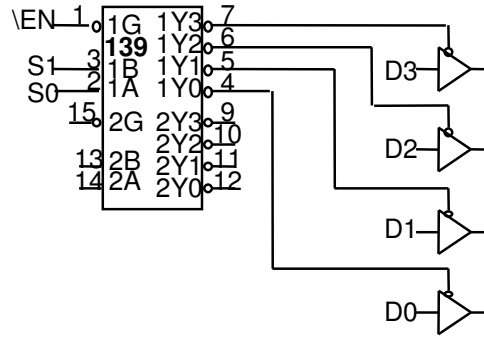
Implémentation de l'inverseur 3-états

M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

58

État « haute impédance » (Tri-state)



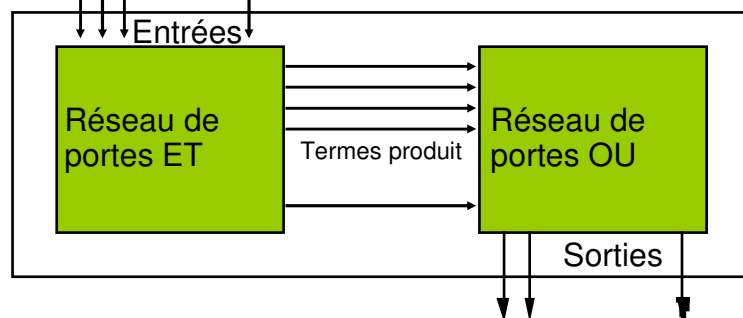
Autre implémentation du Mux 4:1

Décodeur + 4 portes trois états

Logique à deux niveaux : PALs et PLAs

Bloc de base préfabriqué avec beaucoup de portes ET/OU (ou NOR/NAND).
Personnalisé par création ou destruction de connexions entre les portes

Bloc logique programmable pour la forme somme de produits



PALs et PLAs : le partage de termes produit

Équations

Exemple:

$$F_0 = A + B' C'$$

$$F_1 = A C' + A B$$

$$F_2 = B' C' + A B$$

$$F_3 = B' C + A$$

Personnalisation

Terme produit	Entrées			Sorties			
	A	B	C	F ₀	F ₁	F ₂	F ₃
AB	1	1	-	0	1	1	0
BC	-	0	1	0	0	0	1
AC	1	-	0	0	1	0	0
BC	-	0	0	1	0	0	0
A	1	-	-	1	0	0	1

Pour les entrées

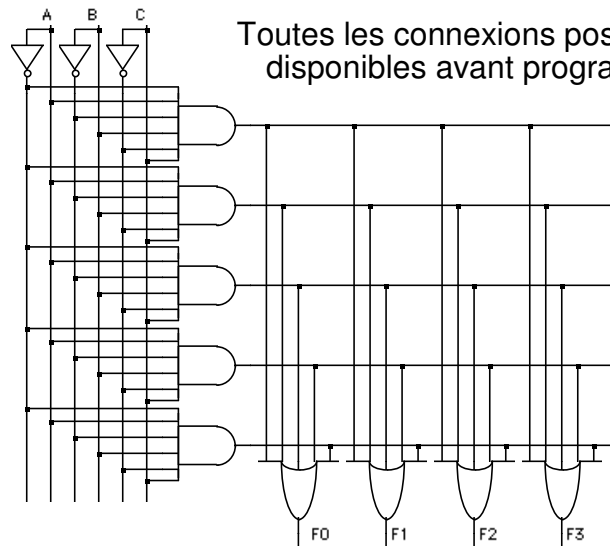
1 = terme non complémenté
 0 = terme complémenté
 - = indifférent

Pour les sorties

1 = connecté à la sortie
 0 = non connecté

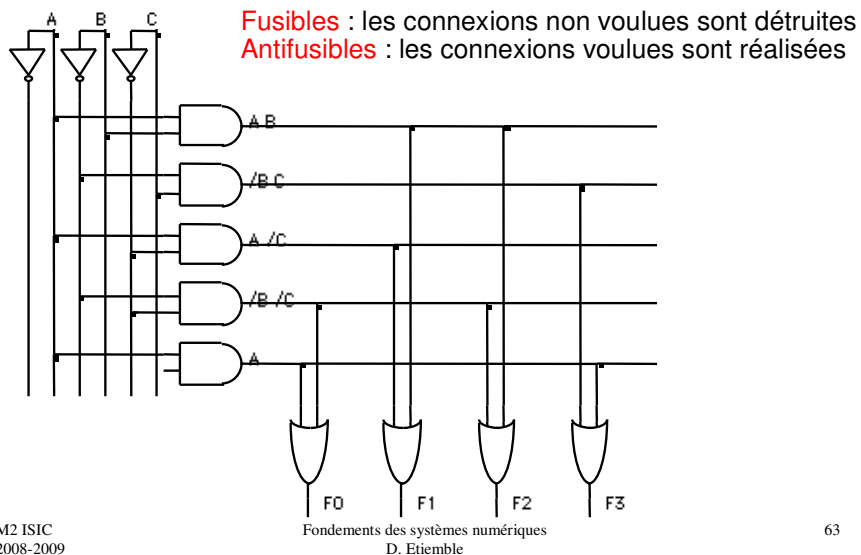
Réutilisation de termes

PALs and PLAs : suite de l'exemple



Toutes les connexions possibles sont disponibles avant programmation

PALs and PLAs

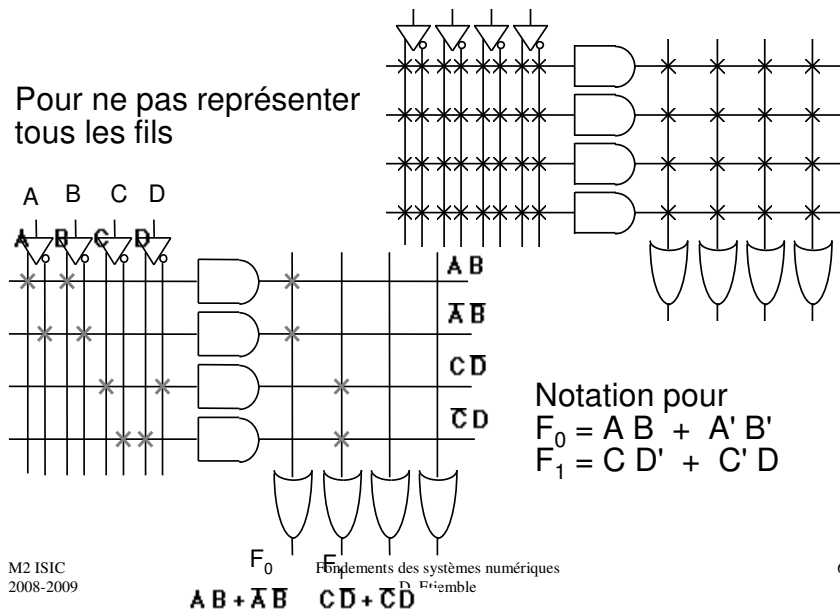


M2 ISIC
2008-2009

63

PALs et PLAs : autre représentation

Pour ne pas représenter tous les fils



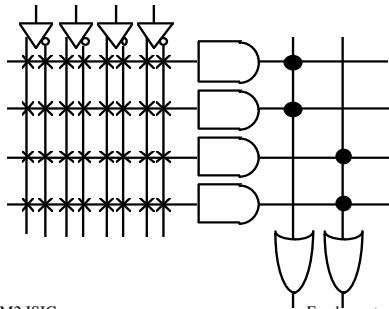
M2 ISIC
2008-2009

64

PALs versus PLAs

PAL : la topologie du réseau OU est fixée

- groupe fixé de lignes ET
- rapide, peu cher
- l'entrée est limitée



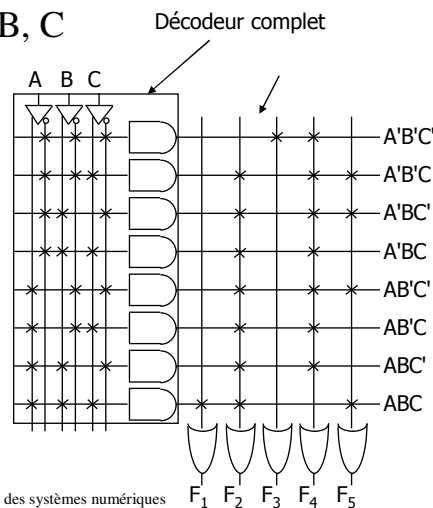
Une colonne donnée du réseau OU a accès seulement à un sous ensemble des termes produit possibles

Exemple PAL - PLA

- Plusieurs fonctions de A, B, C

- $F_1 = A B C$
- $F_2 = A + B + C$
- $F_3 = A' B' C'$
- $F_4 = A' + B' + C'$
- $F_5 = A \text{ xor } B \text{ xor } C$

A	B	C	F_1	F_2	F_3	F_4	F_5
0	0	0	0	0	1	1	0
0	0	1	0	1	0	1	1
0	1	0	0	1	0	1	1
0	1	1	0	1	0	1	0
1	0	0	0	1	0	1	1
1	0	1	0	1	0	1	0
1	1	0	0	1	0	1	0
1	1	1	1	1	0	0	1



Exemple PLA/PAL: convertisseur BCD vers Gray

Table

A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	1	1	1	0
0	1	1	0	1	0	1	0
0	1	1	1	1	0	1	1
1	0	0	0	1	0	0	1
1	0	0	1	1	0	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

AB \ CD	00	01	11	10
00	0	0	X	1
01	0	1	X	1
11	0	1	X	X
10	0	1	X	X

W

AB \ CD	00	01	11	10
00	0	1	X	0
01	0	1	X	0
11	0	0	X	X
10	0	0	X	X

X

AB \ CD	00	01	11	10
00	0	1	X	0
01	0	1	X	0
11	1	1	X	X
10	1	1	X	X

Y

AB \ CD	00	01	11	10
00	0	0	X	1
01	1	0	X	0
11	0	1	X	X
10	1	0	X	X

Z

M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

67

PALs and PLAs

Fonctions minimisées :

$$W = A + B D + B C$$

$$X = B C'$$

$$Y = B + C$$

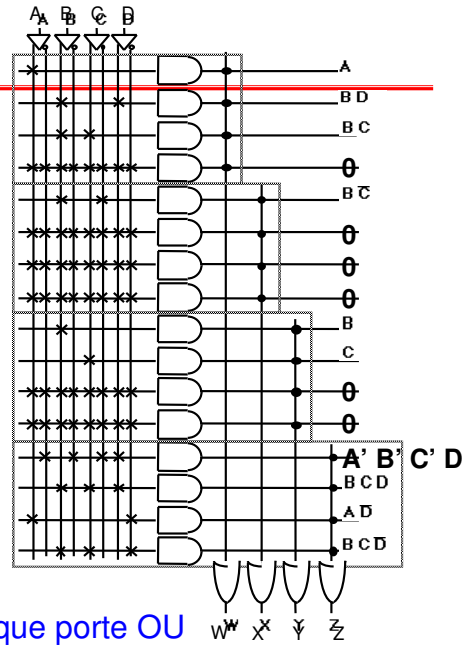
$$Z = A'B'C'D + B C D + A D' + B' C D'$$

Pas de termes partagés



PAL:

4 termes produit pour chaque porte OU

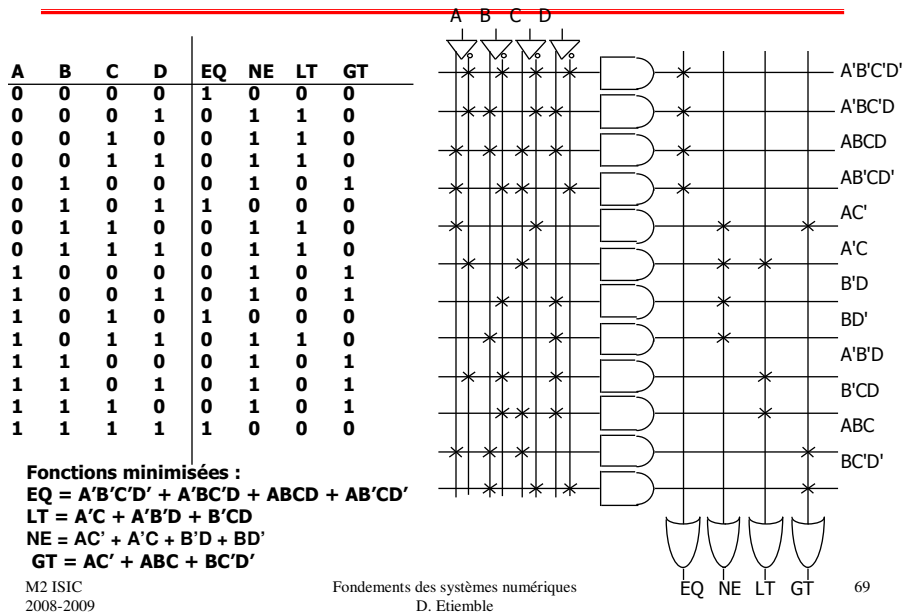


M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

68

PALs et PLAs : comparateur (autre exemple)



Exemple PAL

- Contrainte PAL : Entrance des portes OU

$$S = A'B'R_e + A'BR_e' + AB'R_e' + ABR_e$$

$$R_s = AB + BR_e + AR_e$$

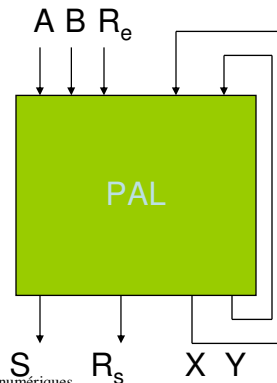
→ Une porte OU à 4 entrées est nécessaire

$$X = AB' + A'B$$

$$Y = AB$$

$$S = XR_e' + X'R_e$$

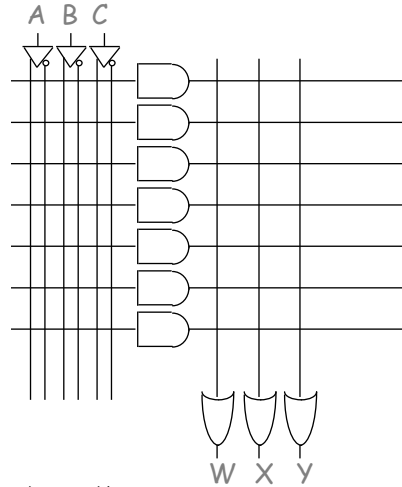
$$R_s = XR_e + Y$$



Exemple

- Implanter les fonctions suivantes sur le PLA:

- $W = AB + A'C' + BC'$
- $X = ABC + AB' + A'B$
- $Y = ABC' + BC + B'C'$



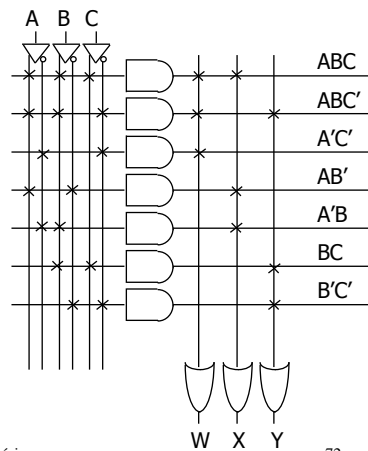
M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

71

Exemple (suite)

- 9 termes et PLA 7 termes
 - On peut appliquer le théorème du consensus pour simplifier W
 $W = AB + A'C'$
- 8 termes et PLA 7 termes
 - $AB = ABC + ABC'$
 - On écrit W pour réutiliser des termes :
 $W = ABC + ABC' + A'C'$
- Maintenant
 - $W = ABC + ABC' + A'C'$
 - $X = ABC + AB' + A'B$
 - $Y = ABC' + BC + B'C'$
- “**technology mapping**”
 - Manipuler les fonctions logiques pour qu'elles puissent utiliser les ressources disponibles



M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

72

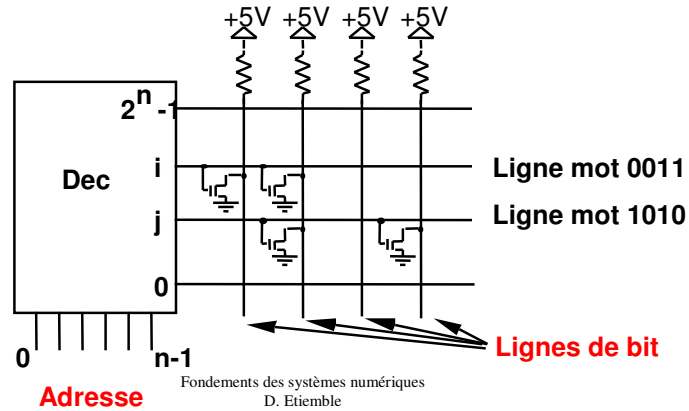
Les mémoires ROM (read only memory)

ROM : Réseau 2D de 1 et de 0.

Une ligne est un mot. Entrées du décodeur : adresse

Largeur d'une ligne : taille du mot

L'adresse est l'entrée ; le mot sélectionné est la sortie



M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

73

ROM et logique combinatoire

Implémentation de logique combinatoire (forme canonique à deux niveaux) avec une ROM

$$F_0 = A' B' C + A B' C' + A B' C$$

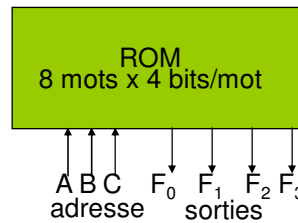
$$F_1 = A' B' C + A' B C' + A B C$$

$$F_2 = A' B' C' + A' B' C + A B' C'$$

$$F_3 = A' B C + A B' C' + A B C'$$

A	B	C	F ₀	F ₁	F ₂	F ₃
0	0	0	0	0	1	0
0	0	1	1	1	1	0
0	1	0	0	1	0	0
0	1	1	0	0	0	1
1	0	0	1	0	0	1
1	0	1	1	0	0	0
1	1	0	0	0	0	1
1	1	1	0	1	0	0

Table



Schéma

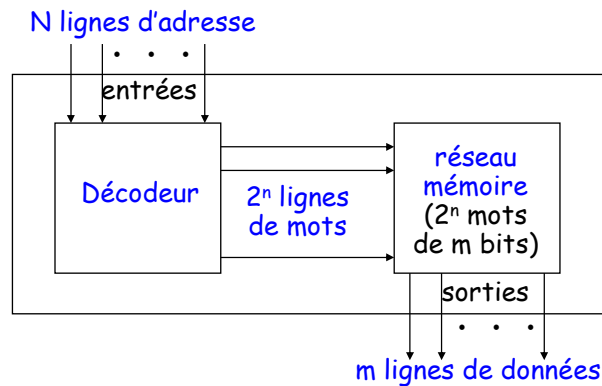
M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

74

Structure ROM

- Comme un PLA, mais avec un réseau ET complet (décodeur)
- Réseau OU complètement flexible (contrairement aux PALs)



ROMs et PLAs

- PLA : topologies généralisées avec plans ET et OU
 - Compact : terme ET partagé
 - A besoin d'un programme CAO : forme somme de produits optimisée
 - Plus lent que les PAL
- ROM : Le plan ET est prédéterminé
 - Implémentation sous forme canonique
 - Le plan ET est aussi appelé décodeur d'adresses
 - Satisfaisant lorsque
 - La plupart des combinaisons d'entrée sont utilisées
 - Peu de partage de termes produit
 - Pas d'outils de CAO
 - La taille double par entrée
 - Pas de termes indifférents
 - Le plus lent

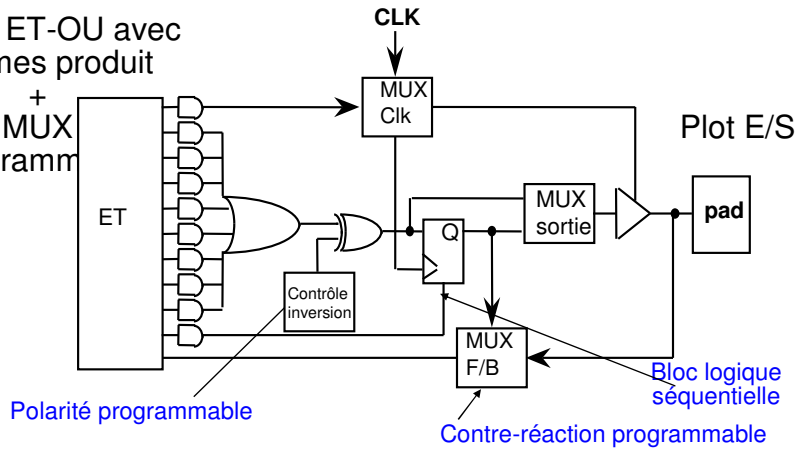
CPLD/EPLD

Altera EPLD (Erasable Programmable Logic Devices)

Bloc de base Altera = MACROCELL

Réseau ET-OU avec 8 termes produit

+ MUX Programmable



M2 ISIC
2008-2009

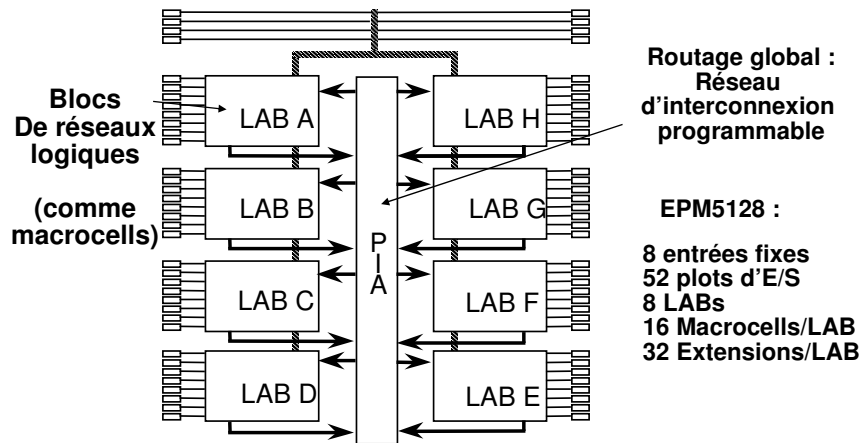
Fondements des systèmes numériques
D. Etiemble

77

Stratégies d'implémentation

Les structures ET-OU sont relativement limitée
Ne peuvent partager les signaux et termes produits entre macrocellules

Solution Altera : matrice de réseaux (MAX)

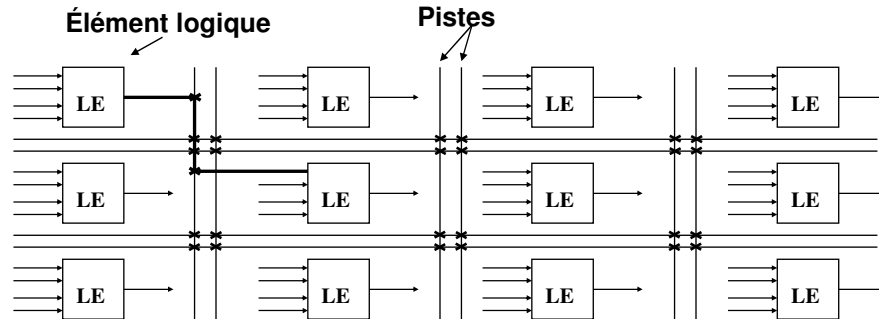


M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

78

Les FPGA



- Chaque élément logique produit une sortie
- Interconnexion programmable entre les éléments
- Pistes d'interconnexion regroupées en canaux.

Élément logique

- LUT (Look-up table)
 - SRAM à 4 ou 5 bits d'adresse et 1 bit de sortie
 - Implante toutes les tables de vérité de 4 ou 5 entrées
- Avantage
 - Programmable (SRAM)
 - Toutes les fonctions sont équivalentes
 - XOR

Adresse	Adresse	Contenu
0	0000	0
1	0001	1
2	0010	1
3	0011	0
4	0100	1
5	0101	0
6	0110	0
7	0111	1
8	1000	1
9	1001	0
10	1010	0
11	1011	1
12	1100	0
13	1101	1
14	1110	1
15	1111	0

LUT dans un bloc logique

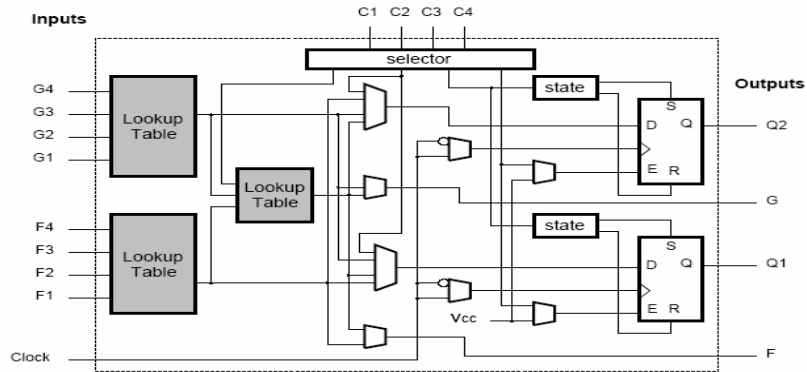
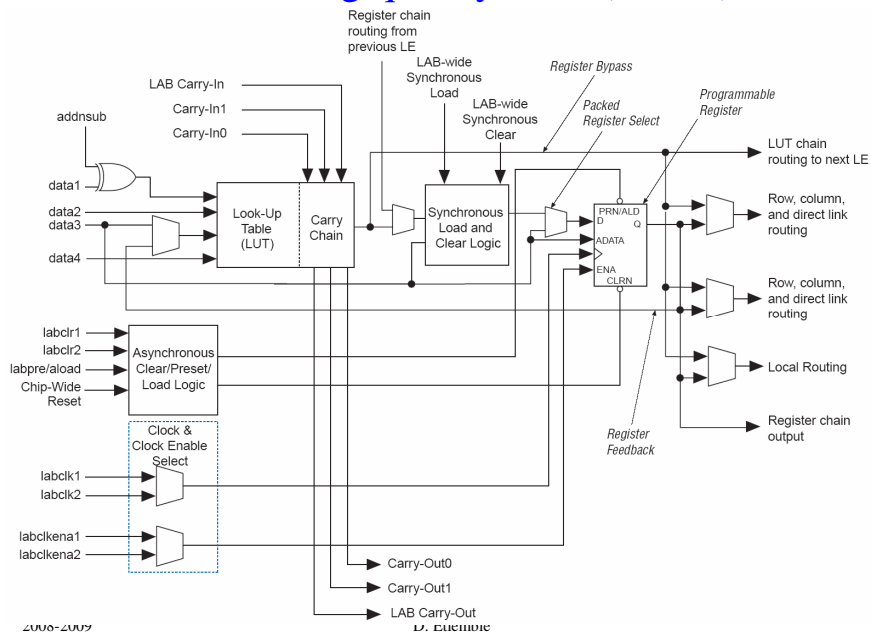
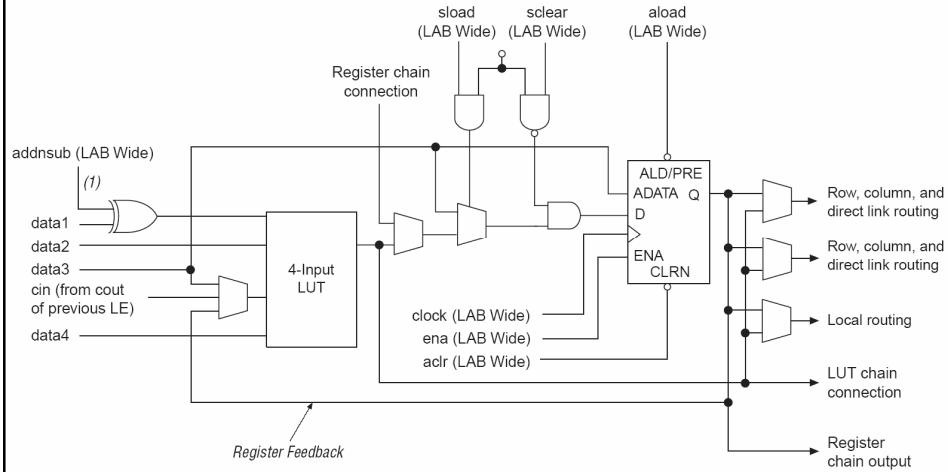


Figure 18 - Xilinx XC4000 Configurable Logic Block (CLB).

Élément logique Cyclone (Altera)



Élément logique – Mode normal

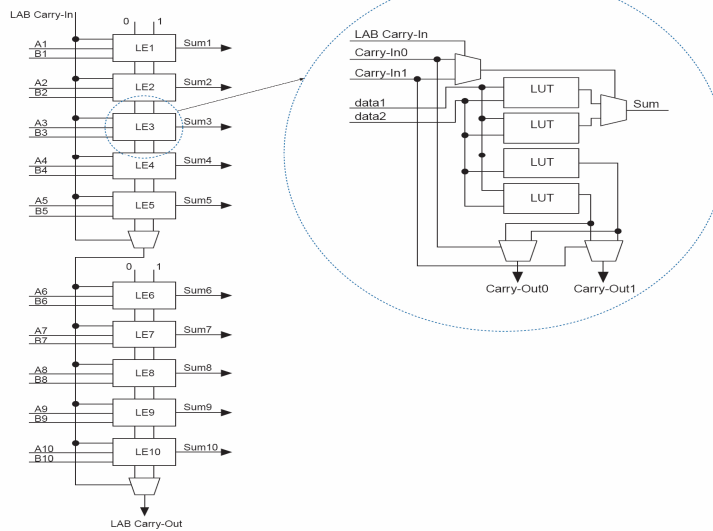


M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

83

Élément logique : retenues rapides



M2 ISIC
2008-2009

FONDEMENTS DES SYSTÈMES NUMÉRIQUES
D. Etiemble

84

Demi - additionneur

A_i	B_i	S	R
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

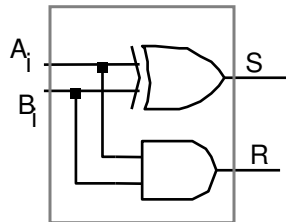
A_i	0	1
B_i	0	1
1	1	0

$$S = \overline{A_i} B_i + A_i \overline{B_i}$$

$$= A_i \oplus B_i$$

A_i	0	1
B_i	0	0
1	0	1

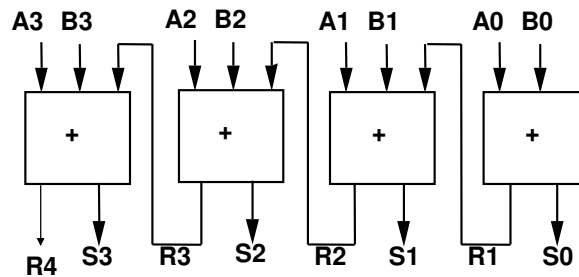
$$R = A_i B_i$$



Demi-additionneur

La brique de base : additionneur 1 bit

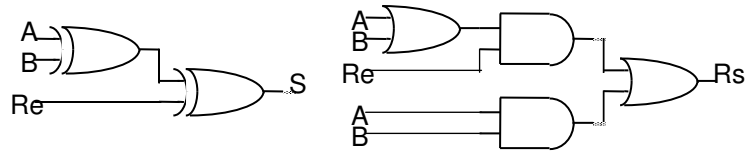
Additionneur 4 bits



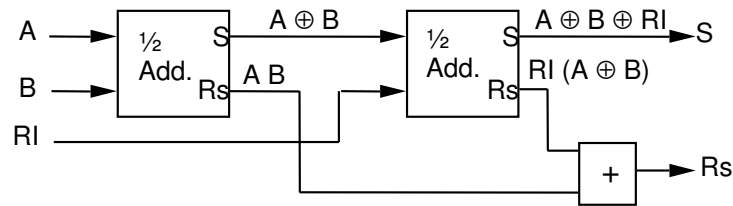
Additionneur 1 bit
+ règles d'assemblage

Additionneurs 1 bit

Approche standard



Autre approche avec demi-additionneurs



$$A B + Re(A \text{ xor } B) = A B + B Re + A Re$$

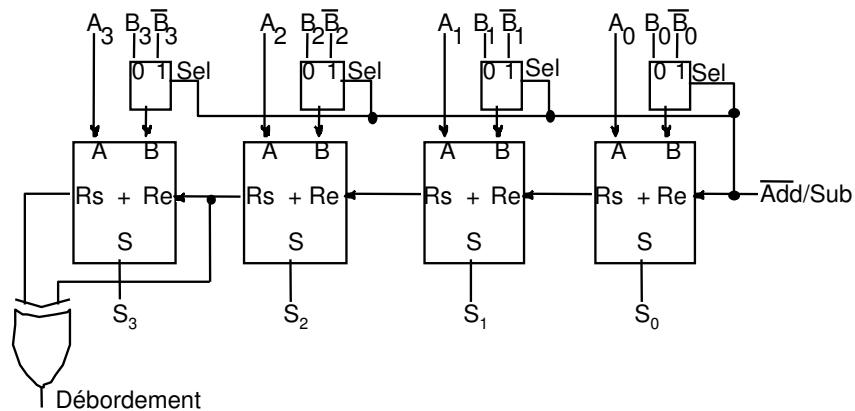
M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

87

Additionneur/soustracteur

$$A - B = A + (-B) = A + \overline{B} + 1$$



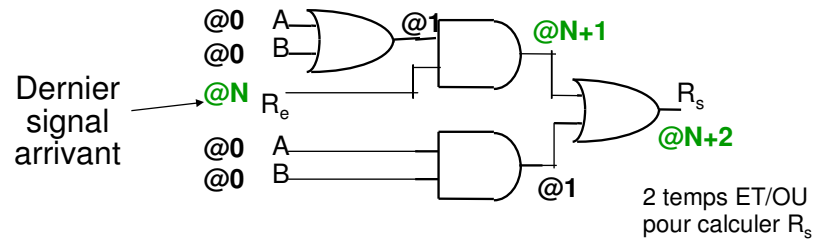
M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

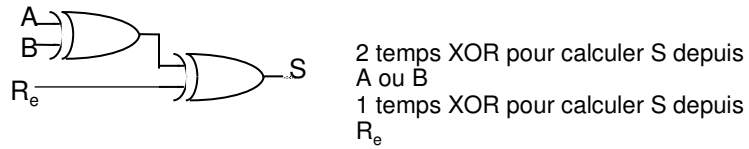
88

Conception d'additionneur

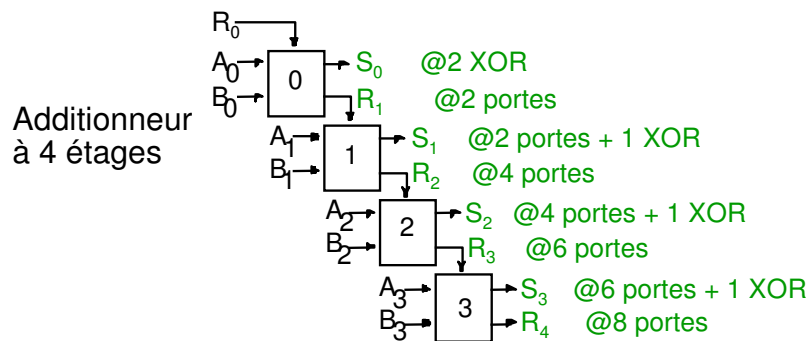
Temps de propagation de la retenue



Temps de propagation de la somme



Additionneur n bits à propagation simple de retenue



Le temps de propagation est proportionnel au nombre d'étages

Additionneur à retenue anticipée

Fonction génération $G_i = A_i B_i$ *doit générer une retenue*

Fonction propagation $P_i = A_i \text{ xor } B_i$ $R_s = R_e$

S et R en terme de fonctions G_i et P_i :

$$S_i = A_i \text{ xor } B_i \text{ xor } R_i = P_i \text{ xor } R_i$$

$$\begin{aligned} R_{i+1} &= A_i B_i + A_i R_i + B_i R_i \\ &= A_i B_i + R_i (A_i + B_i) \\ &= A_i B_i + R_i (A_i \text{ xor } B_i) \\ &= G_i + R_i P_i \end{aligned}$$

Additionneur à retenue anticipée

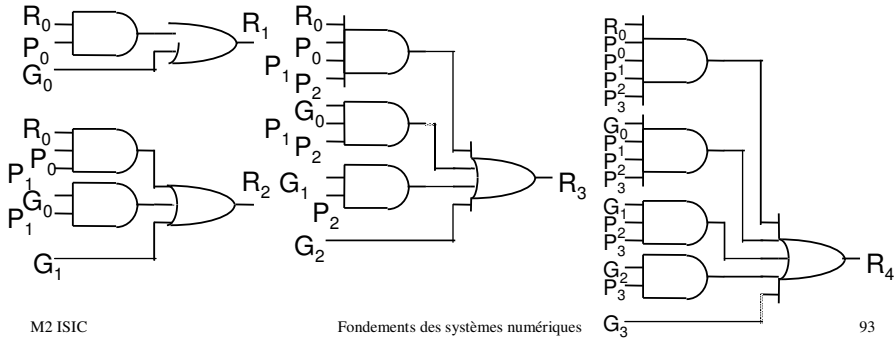
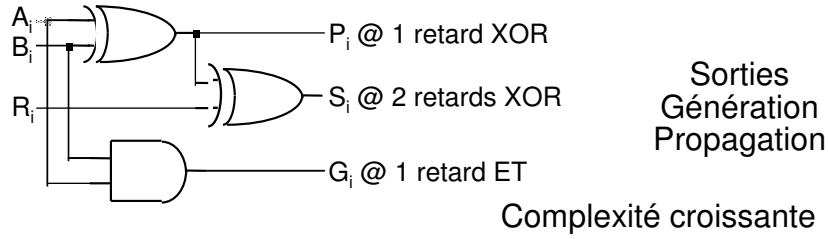
Expressions des retenues

$$\begin{aligned} R_1 &= G_0 + P_0 R_0 \\ R_2 &= G_1 + P_1 R_1 = G_1 + P_1 G_0 + P_1 P_0 R_0 \\ R_3 &= G_2 + P_2 R_2 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 R_0 \\ R_4 &= G_3 + P_3 R_3 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 \\ &\quad + P_3 P_2 P_1 P_0 R_0 \end{aligned}$$

Chaque équation de retenue peut être implantée comme un réseau logique à deux niveaux

Les variables sont les entrées de l'additionneur et la retenue de l'étage 0

Additionneur à retenue anticipée



M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

93

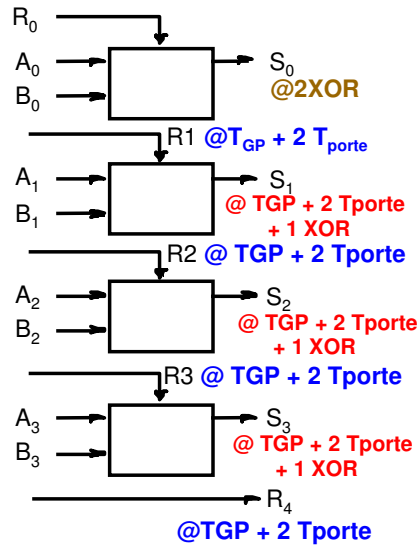
Additionneur à retenue anticipée

Délai critique

La logique de retenues anticipées génère plus vite les retenues
Les sommes sont donc calculées plus vite

$$T_{GP} = \max(tp_{ET}, tp_{XOR})$$

$$T_{porte} = tp(ET, OU)$$

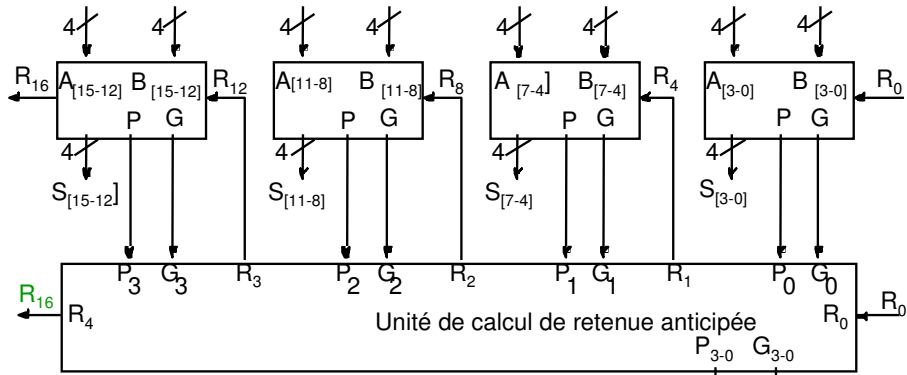


M2 ISIC
2008-2009

Fondements des systèmes numériques
D. Etiemble

94

Additionneur à retenue anticipée : hiérarchisation

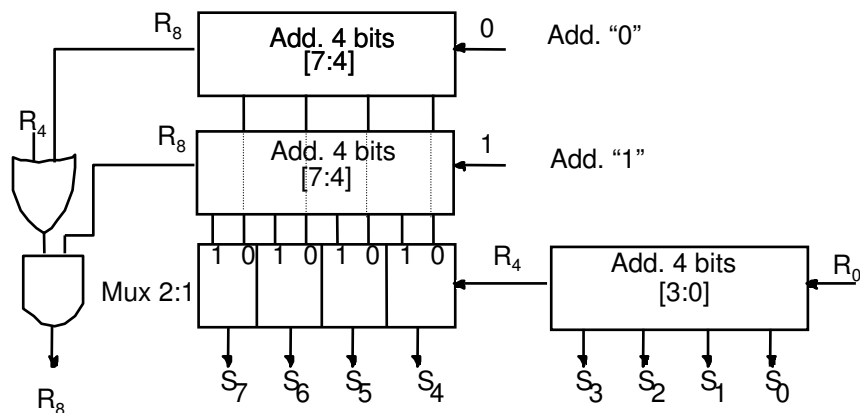


Additionneurs 4 bits avec calcul anticipé de retenue

$$R_4 = G_3 + P_3 R_3 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 R_0$$

Additionneur à sélection de retenue

Matériel supplémentaire pour accélérer le calcul de la retenue



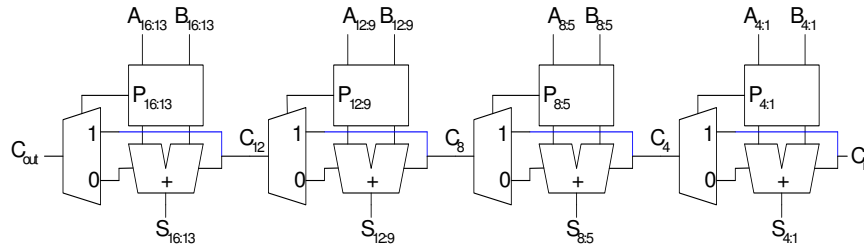
Calcul des sommes de poids fort en parallèle

-Une addition avec retenue entrée = 0

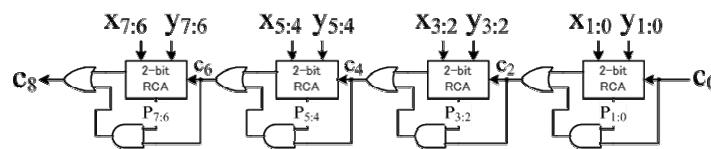
-Une addition avec retenue entrée = 1

Additionneur à saut de retenue

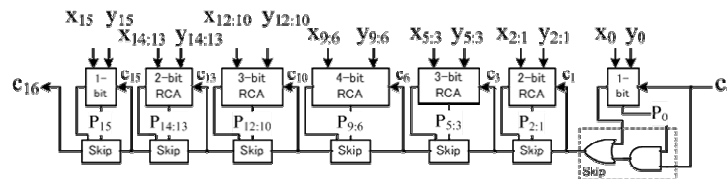
- Saut de retenue par groupe de n bits
 - Calcul de la propagation sur n bits



Exemples d'additionneurs à saut de retenue

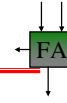


Saut constant



Saut variable

Puissance et délai (*additionneurs*)



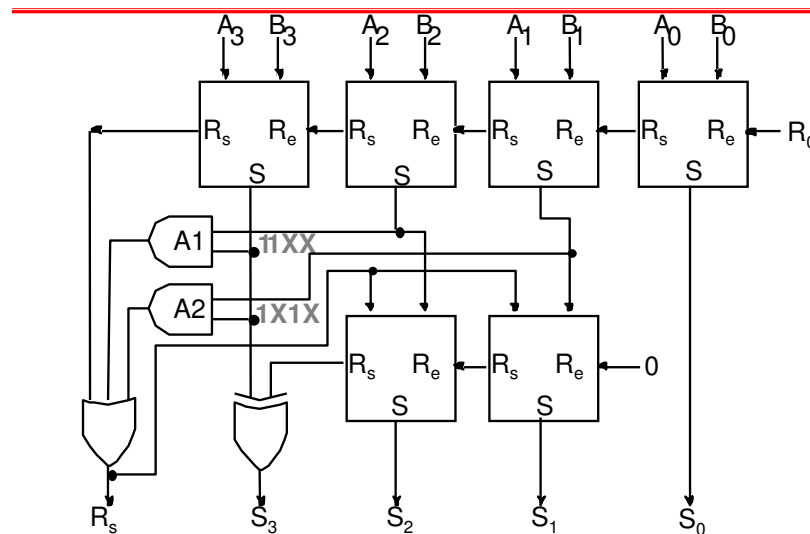
- D'après Callaway et Swartzlander*:

	Energie	
	(pJ)	Délai (ns)
Ripple Carry	117	54.27
Constant Width Carry Skip	109	28.38
Variable Width Carry Skip	126	21.84
Carry Lookahead	171	17.13
Carry Select	216	19.56
Conditional Sum	304	20.05

- En ne choisissant qu'un critère :
 - Puissance – utiliser “constant width carry skip”
 - Délai – utiliser “carry look-ahead”

* “Estimating the power consumption of CMOS adders” - Callaway, T.K.; Swartzlander, E.E., Jr. 11th Symposium on Computer Arithmetic, 1993. Proceedings.

Additionneur BCD



Ajouter 0110 à la somme quand elle dépasse 9 (11XX or 1X1X)