

Cours de logique combinatoire et séquentielle

Introduction :

On distingue de nombreuses familles de circuits.

La famille en CMOS 4000 fonctionne de 5 à 15V.

Les 74LS utilisaient l'ancienne technologie TTL (issue de la DTL).

Les 74HC utilisent la technologie CMOS (entrées haute impédance).

Les circuits logiques ont deux états : 0 et 1.

Pour un 74HC : 0 = 0V et 1 = 5V (par rapport à la masse), sauf drain ouvert (74HC05).

Les 74LV fonctionnent à partir de 1V.

Pour l'état haut, il faut $V_{out_min} > V_{in_max}$. Pour l'état bas, il faut $V_{out_max} < V_{in_min}$.

Les 74HCT (T = compatible TTL) peuvent servir d'interface 3V3 / 5V.

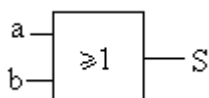
Logique combinatoire :

Les fonctions de base :

Porte NON : $S = a/$



Porte OU : $S = a + b$



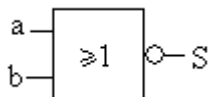
$S = 1$ si $a = 1$ ou $b = 1$ (si le nombre d'entrées à 1 est supérieur ou égal à 1).

Table de vérité :

a	b	S
0	0	0
0	1	1
1	0	1
1	1	1

La valeur numérique de ab ($2a + b$) varie de 0 à 3

Porte NOR : $(a+b)/$



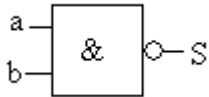
a	b	S
0	0	1
0	1	0
1	0	0
1	1	0

Porte ET : $S = a.b$



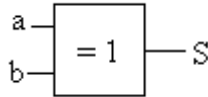
a	b	S
0	0	0
0	1	0
1	0	0
1	1	1

Porte NAND : $S = (ab)'$



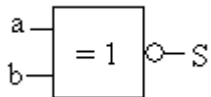
a	b	S
0	0	1
0	1	1
1	0	1
1	1	0

OU-Exclusif : $S = a \oplus b$ (l'un ou l'autre mais pas les deux) $S = a.b' + a'.b$ (les ET sont prioritaires sur le OU)



a	b	S
0	0	0
0	1	1
1	0	1
1	1	0

Identité : $S = (a \oplus b)'$



a	b	S
0	0	1
0	1	0
1	0	0
1	1	1

Propriétés :

Elément neutre : $a + 0 = a$ $a.1 = a$

Elément absorbant : $a + 1 = 1$ $a.0 = 0$

Commutativité : $a + b = b + a$

$a.b = b.a$

Associativité : $a + (b + c) = (a + b) + c$

$a.(b.c) = (a.b).c$

Distributivité : $a + bc = (a + b).(a + c)$

$a.(b + c) = a.b + a.c$

Complémentarité : $a + a' = 1$ $a.a' = 0$

Involution : $a'' = a$

Idempotence : $a + a = a$ $a.a = a$

Absorption 1 : $a + a.b = a$ $a.(a + b) = a$

Absorption 2 : $a + a'.b = a + b$

$a.(a' + b) = a.b$

Consensus : $a.b + a'.c + bc = a.b + a'.c$

$(a + b).(a' + c).(b + c) = (a + b).(a' + c)$

Disjonction : $a.b + a.b' = a$

De Morgan : $(a + b)' = a'.b'$

$(a.b)' = a' + b'$

Application :

Réaliser une porte XOR (OU-Exclusif) avec 4 portes NAND

Simplification d'une équation logique :

Tableau de Karnaugh :

dc\ba	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

Les nombres dans les cases, indiquent les valeurs numériques, correspondant aux entrées d,c,b,a
($N = 8d + 4c + 2b + 1a$).

Les cases sont repérées en code Gray (Un seul bit est changé pour passer d'une combinaison à une autre).
Il suffit de regrouper le maximum de cases adjacentes (ex 1 et 5), en un minimum de paquets.

Pour un tableau avec quatre variables :

- Un paquet de huit cases est défini par une seule variable
- Un paquet de quatre cases est défini par deux variables
- Un paquet de deux cases est défini par trois variables

0, 2, 8 et 10 sont aussi adjacentes et correspondent à $c/.a/$

Si des combinaisons sont inutilisées, on peut considérer que les sorties associées sont 0 ou 1, de manière à agrandir les regroupements, et à diminuer leur nombre (on note X pour inutile).

Réalisation de fonctions logiques :

Portes logiques : Il faut souvent plusieurs circuits intégrés, pour réaliser une fonction.

Multiplexeur :

La sortie d'un multiplexeur est dans le même état que l'entrée sélectionnée, parmi 4 ou 8, grâce à un code de 2 ou 3 variables ($S = E0.b/.a/ + E1.b/.a + E2.b.a/ + E3.b.a$). Il suffit de câbler les entrées à sélectionner à la masse ou au +, en suivant la table de vérité, et d'appliquer la combinaison voulue sur les entrées de sélection (adresse). On peut aussi appliquer des variables sur des entrées à sélectionner :

Ex : Fonction majorité : $F = a(b+c) + bc = b/c(a) + bc/(a) + bc(1)$

On peut aussi brancher des multiplexeurs en cascade...

En principe, il faut de préférence, mettre les variables à compléter en adresse (pour essayer d'économiser des portes 'NON'), et appliquer les variables seules, ou les plus fréquentes (dans les termes de l'équation logique), sur l'adresse du dernier multiplexeur. Ex : $F = a + b/c/ + bcd$

$F = a(1) + a/[b/c/(1) + bc(d)] \quad F = b/c/(1) + b/c(a) + bc/(a) + bc(a+d).$

Démultiplexeur :

$S0 = E.b/.a/ \quad S1 = E.b/.a$
 $S2 = E.b.a/ \quad S3 = E.b.a$

PAL :

C'est un circuit programmable. Exemple : PAL 16L8 (10 à 16 entrées et 2 à 8 sorties).

Contacts :

On les branche en série pour réaliser des fonctions ET, et en parallèle pour réaliser des fonctions OU.
Par exemple, la fonction $ab(e+f) + cd(ae+bf)$, se traduit par $(ab+cd)(ae+bf)$, pour gagner deux contacts...

Additionneur :

$S = a \oplus b \oplus R-1$ (=1 si le nombre d'entrées à 1 est impair) et Retenue $R = R-1.(a+b) + a.b$ (majorité)

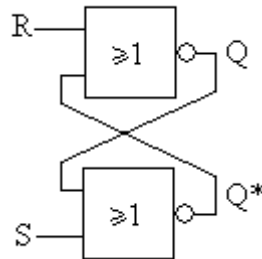
Il peut être réalisé à partir de 2 demi-additionneurs.

Logique séquentielle :

La logique séquentielle se distingue de la logique combinatoire, dans le fait que pour la même combinaison des entrées, il peut exister plusieurs combinaisons différentes des sorties, selon les états précédents. Les cellules de bases sont des mémoires élémentaires appelées Bascules.

Bascule RS :

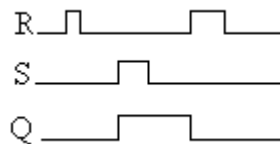
R	S	Q	Q*
0	0	Qp	Qp*
0	1	1	0
1	0	0	1
1	1	0	0



Qp représente l'état précédent de la sortie Q.

R = Reset = Remise à Zéro.

Pour concevoir ou analyser un circuit séquentiel, on utilise le chronogramme :

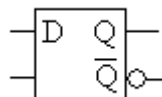


On peut réaliser un séquenceur avec une bascule RS par étape :

- $S_n = Q_{n-1}.T_n$
- $R_{n-1} = Q_n$

T_n étant la transition associé à l'étape n.

Bascule D-Latch :



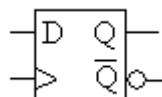
Q = D pour 2ème entrée à 1.

Q = Qp pour 2ème entrée à 0.

Elle se fabrique avec une bascule RS :

- $S = M.D$
- $R = M.D/$

Bascule D-Edge :



C'est une bascule D qui mémorise sur front.

Bascule T :

= Bascule D avec $D = Q/$. Attention aux rebonds créés par le bouton poussoir sur l'entrée CLK ($RC = 20ms$).

Elle permet de diviser une fréquence par deux : La sortie Y change d'état à chaque front montant sur l'entrée a. X est une sortie interne. x et y sont les entrées temporellement décalées, correspondant aux sorties X et Y.

Voici la table de fonctionnement :

xy/a	0	1
00	[00]	01
01	11	[01]
11	[11]	10
10	00	[10]

Les sorties entre crochets représentent les états stables.

Les équations sont les suivantes :

- $X = a/.y + a.x + x.y$
- $Y = a/.y + a.x/ + x/.y$

Les termes xy et x/y sont des termes redondants : ils assurent la continuité des signaux.

Bascule JK :

C'est une bascule RS à commande sur front. La table de fonctionnement est la suivante :

JK	Q
00	Qp
01	0
10	1
11	Qp/

La table ci-après indique quels états peuvent être appliqués aux entrées en fonction de l'évolution souhaitée de la sortie :

Qp Q	J K
0 0	0 X
0 1	1 X
1 0	X 1
1 1	X 0

X représente un état indifférent.

Il existe aussi des bascules JK en cascade, appelées JK Maître-Esclave...

Compteurs synchrones :

Les compteurs synchrones intégrés sont réalisés à partir de bascules 'D-Edge' ou 'JK'.

Les entrées des bascules sont définies, par des fonctions logiques de leurs sorties.

'On prépare à l'instant t ce qui arrivera à l'instant $t+1$ '.

Le nombre de bascules correspond au moins, à la puissance de deux supérieure au nombre d'états successifs.

Ex : On détermine D2, D1 et D0 pour cette séquence de haut en bas de Q2, Q1 et Q0 :

Q2	Q1	Q0	D2	D1	D0
1	0	1	1	0	0
1	0	0	1	1	0
1	1	0	0	1	0
0	1	0	0	1	1
0	1	1	0	0	1
0	0	1	1	0	1

Ce qui donne $D2 = Q1/$, $D1 = Q0/$ et $D0 = Q2/$

Pour décompter de 11 à 0, on peut utiliser un décompteur à chargement parallèle :

On fixe les entrées parallèles à 1011 ($8+2+1$).

Transmissions numériques :

Code HDB3 :

Le but est de transmettre données et horloge sur un fil, sans perte de synchronisation, et sans composante continue (transformateurs de signal ou transmission hertzienne).

Les 0 sont codés 0, sauf 0000 :

- codé 000V (V=viol) si dernier V opposé au dernier bit (les V sont toujours alternés).
- codé B00V, B et V de même polarité (B=bourrage), si dernier V comme dernier bit.

Dernier V positif :

- dernier bit positif : -00-
- dernier bit négatif : 000-

Dernier V négatif :

- dernier bit positif : 000+
- dernier bit négatif : +00+

Les 1 (et les B) sont alternés + et -.

Un nombre pair de 1 n'a pas de composante continue : On les ignore.

0000	1	2	3	4	5	6	7	8
+	1 V		1 V		B V		B V	
-		1 V		1 V		1 V		B V

le V est toujours de même signe que le bit précédent.

Toujours un nombre pair de +, suivi d'un nombre pair de - !

Restitution du signal d'horloge :

Il s'agit d'une boucle à verrouillage de phase. Les équations logiques du comparateur de phase sont :

$$\begin{aligned} S &= H \cdot D \\ I+ &= D \cdot Q \\ I- &= H \cdot Q \end{aligned}$$

Voici le courant de charge du condensateur dont la tension sert à définir la fréquence de l'oscillateur :

