ITI 1500

—

Notes

Chapitre 1: Le système binaire

A) La représentation des nombres

Il y a deux notations possibles pour représenter un nombre dans un système donné:

1. Notation valeur de position : N = (a(n-1), a(n-2)…a1,a0.a-1,a-2…a-m)r
    - . = point de base
    - R = la base
    - N = nombre de chiffre de la partie entière
    - M = le nombre de chiffre de la partie fractionnaire
    - a(n-1) = le chiffre le plus significatif
    - a(-m) = le chiffre le moins significatif

- Système de numération décimal:
    - Le système de numération décimal est un nombre à valeure de position
    - Exemple: $(4621)10 = 4*10\text{^}3 + 6*10\text{^}2 + 2*10\text{^}1 + 1*10\text{^}0$
- Le système binaire:
    - Composé de 0 et de 1
    - Chaque bits représente 2^n

2. Notation polynomiale:

- Même exemple qu'en haut. Les nombre à la base 10 sont mutlipliés par 10^n selon leur position
- Le système binaire: (11010.11)2 = 1*2^4 + 1*2^3+…

3. Conversion décimale vers binaire:

$N = (a(n-1), a(n-2)...a1,a0.a-1,a-2...a-m)_r$

La partie entière se retrouve avant le point de base et la partie fractionnaire après. Elles sont converties de manière différentes.

- Conversion de la partie entière:
  - Continuer à diviser par 2 jusqu'à ce que le quotient est 0
  - Collecter le reste dans l'ordre inverse
  - Exemple:
    - 162\2 = 81 reste 0
    - 81\2 = 40 reste 1
    - 40\2 = 20 reste 0
    - 10\2 = 5 reste 0
    - 5\2 = 2 reste 1
    - 2\2 = 1 reste 0
    - 1\2 = 0 reste 1
    - Alors : $(1010010)_2$
- Conversion de la partie fractionnaire :
  - Continuer à multiplier la partie fractionnaire par 2 jusqu'à obtenir 0. Collecter les parties entières dans l'ordre avant
  - Exemple: $(0.375)_{10}$
    - 0.375 * 2 = 0.750
    - 0.750 * 2= 1.500
    - 0.500 *2 = 1
    - Alors: $(0.375)_{10} = (.011)_2$

4. Conversion binaire vers décimale:

Pour convertir les nombre binaire ou de base 2 vers décimal, on obtient d'abord la notation polynomiale du nombre, ensuite on fait la somme de tous les produits polynomiale

Exemple: $(1101.01)_2$ = (1*2^3) + (1*2^2) + (0*2^1) + (1*2^0) + (1*2^-1) + (1*2^-2) = $(13.25)_{10}$

5. Le système Octal:

Chiffres: {0,1,2,3,4,5,6,7}

Notation polynomiale: $(127.4)_8 = (1*8^2) + (2*8^1) + (7*8^0) + (1*8^{-1})$

Le système comprends des chiffres entre 0 à 7., Puisque $8 = 2^3$, un chiffre octal nécessite 3 positions binaires.

- Conversion Décimal vers Octal:
    - Partie entière: diviser par 8 jusqu'à ce que le quotient est 0. Collecter le reste dans l'ordre inverse
    - Partie fractionnaire: continuer à multiplier la partie fractionnaire par 8 jusqu'à obtenir 0. Collecter les restes dans l'ordre avant.

- Conversion Octal vers Décimale:
    - Pour convertir le nombre octal vers décimal, on obtient d'abord la notation polynomiale du nombre, ensuite, on fait la somme de tous les produits.
    - Exemple: $(127.4)_8 = (1*8^2) + (2*8^1) + (7*8^0) + (1*8^{-1}) = (87.5)_{10}$
- Conversion Binaire vers Octal:
    - Former des groupes de 3 bits à partir du point binaire. Ajouter des zéros à chaque bout si nécessaire. Ensuite, convertir chaque groupe de bits à son équivalent octal.
    - Exemple: $(10110100.001011)_2 = (010\ 110\ 100.\ 001\ 011)_2 = (264.13)_8$
- Conversion Octal vers Binaire:
    - On remplace chaque chiffre par son équivalent de séquence de 3-bits binaire
    - Exemple: $(261.35)_8 = (010\ 110\ 001\ .\ 011\ 101)_2$

6. Le système Hexadécimal:

Chiffre: {0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}

Notation polynomiale: $(B65F)_{16} = (11*16^3) + (6*16^2) + (5*16^1) + (15*16^0)$

Hexadécimal (base 16), chiffre de 0 à F. Puisque $16 = 2^4$, un chiffre hexadécimal est équivalent à 4 positions binaires

- Conversion décimal vers Hexadécimal:
    - Partie entière: diviser par 16 jusqu'à ce que le quotient est 0. Collecter les restes dans l'ordre inverse.
    - Partie fractionnaire: continuer à multiplier la partie fractionnaire jusqu'à obtenir 0. Collecter les restes dans l'ordre avant. (Même principe que pour la conversion binaire)

- Conversion Hexadécimal vers décimal:
    - On obtient d'abord la notation polynomiale de nombre. Ensuite, on fait la somme de tous les produits.
    - Exemple: $(B65F)_{16} = (11*16^3) + (6*16^2) + (5*16^1) + (15*16^0) = (46.687)_{10}$
- Conversion Hexadécimal vers binaire:
    - On remplace chaque chiffre Hexadécimal par son équivalent de séquence binaire 4 bits
    - Exemple: $(261.35)_{16} = (0010\ 0110\ 0001.0011\ 0101)_2$
- Conversion binaire vers Hexadécimal:
    - Former des groupes de 4 bits à partir du point binaire. Ajouter des zéros à chaque bout si nécessaire. Ensuite, convertir chaque groupe de bits en son équivalent hexadécimal.
    - Exemple: $(1011\ 0100\ .\ 0010\ 1100)_2 = (B4.2C)_{16}$

B) Les opérations arithmétiques binaires:

1) Addition binaire:

0+0 = 0 1+0 = 1

0+1 = 1 1+1 = 0 retenue 1

Note: Lorsqu'on fait une addition binaire, on peut convertir de binaire à decimal pour vérifier notre réponse.

2) Soustraction binaire:

0-0 =0 1-0 = 1

0-1 = 1 retenue 1 1-1 =0

3) Compléments dans les systèmes de numérations:

Il y a deux types de compléments pour chaque base «r» d'un système:

- Complément à la base «r»
    - Exemple: Complément à la base 10, à base 2… etc.
- Complément à la base moins un (r-1)
    - Exemple: base 10 à 9, base 2 à 1

4) Compléments à la base

$[N]_r = \{ r^n - (N)_r$ si N est différent de 0, sinon 0

- Complément à deux:
    - Copier chaque bit du nombre à partir du bit le plus à droite (moins significatif) vers le bit le plus à gauche (le bit le plus significatif) jusqu'à ce que le premier 1 est copié.
    - Après que le premier 1 est copié, remplacer chaque bit restant pas un 1 si le bit est 0 ou par 0 si le bit est 1.
    - Exemple: $(1010100)_2 = (0101100)$
- Complément à la base moins un: $[N]_r = (r^n-1)_r - (N)_r$
    - Complément à 9 de $[546700]_9 = 999999 - 546700 = 453299$
    - Complément à un: les nombres sont inversés
- Complément à un:
    - Remplace chaque 0 par 1 et chaque 1 par 0 (les nombres sont inversés)
    - Exemple: $(1011000)_2 = (0100111)_2$

5) Soustraction avec le complément à deux:

Le complément à deux est utilisé pour réduire la soustraction à une addition.

A-B = A + (-B). Donc, nous additionnons le complément à deux de B à A.

Exemple:

A = 1010100 B = 1000011. Faisons maintenant le complément à deux de B pour effectuer l'opération.

$A - B = A+ (-B) = (1010100)_2 - (0111101)_2 = (0010001)_2$

***Il ne faut pas tenir compte de la retenue lors de l'opération.

6) Soustraction avec le complément à un:

A = 1010100 et B=1000011. Faisons maintenant le complément à un pour effectuer l'opération.

$A-B = A+[B] = (1010100)_2 + (0111100)_2 = ()$

*** Lors de l'opération il faut ajouter la retenue (donc faire une autre addition) avec le résultat obtenu.

7) Soustraction avec les compléments 10 et 9:

- Complément à 10:
    - Exemple: $(72)_{10} - (32)_{10} = (40)_{10}$, $[32] = 10^2 - (32)_{10} = (68)_{10}$, $(72)_{10} + (68)_{10} = 1\ (40)_{10}$
    - Lorsqu'on fait l'addition finale, on oublie la retenue!
- Complément à 9:
    - Même exemple mais: $[32] = (10^2-1) - (32)_{10} = (67)_{10} = (72)_{10} + (67)_{10} = (1 +30)_{10} = (40)_{10}$

8) Les nombres binaires signés :

Les systèmes numériques sont construits avec des composants à deux états: 0 et 1. Ce sont les seuls états, c'est-à-dire qu'il n'y a aucun état correspondant à un chiffre négatif. À cause des limitations, l'ordinateur utilise le bit le plus à gauche du nombre pour représenter le signe.

0 indique un nombre positif tandis que 1 indique un nombre négatif.

- Représentation signe Grandeur:
    - En utilisant 8 bits pour représenter un nombre binaire.
    - Exemple: -3 = 10000011 = 1\(bit signe) 0000011
- Représentation complément à un signé:
    - Exemple: -3 = 10000011 = 1\(bit signe)1111100
- Représentation complément à deux signé:
    - Exemple: -3 = 10000011 = 1\(bit signe)1111101

9) Débordement durant opération d'addition:

Lors de l'addition de nombres positifs, le débordement se produit quand la somme des deux nombres se situe à l'extérieur de la grandeur du registre.

- Débordement - Compléments Signés:
    - Lorsque les nombres sont représentés en compléments signés, une retenue de 1 qui se dégage de l'addition des deux bits les plus significatifs n'est pas un indicateur de débordement.
    - Exemple: 0011 + 11101 = 0000, retenue de 1 (complément à deux)
    - Pour des nombres compléments signés le débordement se produit lorsque l'addition de deux nombre positifs donne un nombre négatif et l'addition de deux nombres négatifs donnent un nombres positif.

- Exemple de débordement: 010001 + 010000 = 100001

10) Les codes binaires:

On utilise souvent le DCB (décimal codé binaire). Il est utilisé pour représenter les chiffres décimaux de 0 à 9 et un registre de 4 bits est utilisé. Les positions sont 8,4,2,1.

Codes:

0000 = 0 0001 = 1 0010 = 2 …

Chapitre 2: Les portes logiques

A) Les opérations logiques:

1) Opération logique -ET:

Cette porte est représentée par un point ou l'absence d'un opérateur. Deux variables xy deviennent vraie lorsque x = 1 et y = 1.

2) Opération logique -OU:

Cette porte est représentée par un signe + . Deux variables xy deviennent fausses si x=0 et y=0.

3) Opération-NON:

Cette porte est représentée par un prime ou une barre. C'est l'inverse de la valeure de la variable.

Note : 1 signifie vrai et 0 signifie faux.

B) Les portes logiques:

1) La portes NON-ET:

C'est une combinaison de la porte ET et de l'inverseur NON

Table de vérité :

A	B	F
0	0	1
0	1	1

1	0	1
1	1	0

F = (AB)’ = A’ + B’

2) La porte NON-OU:

C’est une combinaison de la porte OU et de l’inversue rNON

A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

F= (A+B)’

3) La porte OU exclusif et NON-OU exclusif :

La sortie est 1 uniquement quand l’une des entrées est 1. Le symbole est le plus entouré

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

OU exclusif : F = AB’ + A’B

NON-OU exclusif : F= AB + A’B’

B) Les formes standards:

Deux manière d’écrire la fonction booléenne, soit avec la somme de produits (SDP) ou produits de somme (PDS)

1) Somme de produits:

- Les minterms:
    - C’est le produit logique ET de variable, donc c’est les sorties des variables qui donnent 1 de la table de vérité.
- Sa forme:
    - SDP a la forme: (…) +(…) +(…)+…

2) Produits de somme:

- Les maxterms:
    - Les sorties de la table de vérité qui donnent 0 (donc c’est le contraire des minterms)

C) Les additionneurs:

1) Le demi-additionneur:

Il accepte deux chiffres binaires en entrée et produit deux chiffres binaire en sortie: le bit et la somme retenue.

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

2) L’additionneur complet:

Il accepte en entrée deux bits et un bits retenu, il génère en sortie une somme et une retenue. L’additionneur complet accepte la retenue en entrée.

Chapitre 3: Simplification des fonctions logiques avec le diagramme de Karnaud

A) Représentation:

Chaque carré représente un minterm. Le diagramme est organisé à ce que deux carrés adjacents différents seulement d’une variable. Celui-ci possèdent 2^n carrés. De plus, pour les fonctions

booléennes, les termes produits sont indiqués par des 1 et les termes sommes sont indiqués par des 0 (représenté par les minterms dans le diagramme)

- Les sorties indéterminée (X) peuvent être considérés dans le groupage.

B) Les impliquants premiers:

Chaque groupe représente un terme produit dans la fonction qui doit contenir un ensemble minimum de termes

Sélectionner les groupes qui contiennent au moins un carré non-couvert par d'autres groupes: Impliquants premiers essentiels

Ne pas sélectionner les groupes dont les carrés sont tous couverts par d'autres groupes: Impliquants premiers optionnels

C) Obtenir le PDS avec un SDP:

1- Avec la Fonction F, écrire la fonction en utilisant les minterms

2- Puis, complémenter F pour obtenir la forme F'

Chapitre 4: Circuits logiques combinatoires

A) Décodeurs:

C'est un circuit logique qui accepte un ensemble d'entrées n représentant un nombre binaire puis active uniquement la sortie

1) Décodeur 2-vers-4:

Le décodeur 2-vers-4 fonctionne selon la table de vérité suivante:

- Les 2 bits en entrée sont appelés S1S0, et les quatres sorties sont de Q0-Q3

S1	S0	Q0	Q1	Q2	Q3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0

1	1	0	0	0	1

- Par exemple, si S1S0 = 10, alors la sortie Q2 sera vraie et toutes les autres seront fausses.
- Fonction: Q0 = S1'S0', Q1 = S1'S0, Q2 = S1S0', Q3 = S1S0
- Entrée Enabler:
    - Elle est utilisée pour activer (1) ou désactiver (0) le dispositif.
    - Quand elle est =1, elle réalise la table ci-dessus. Par contre, lorsqu'elle est initialisés à 0, toutes les sorties du décodeurs sont remises à 0.

2) Décodeurs - Générateurs de minterms:

Pour une somme de produits, on peut utiliser les sorties des minterms du décodeur pour implémenter la fonction.

3) 3-vers-8 Décodeur:

Trois entrées sont décodées en 8 sorties, D0 à D7.

- Implémentation d'un décodeur avec des portes NON - ET:
    - Les décodeurs active high (qui génère des minterms) sont implémentés avec des portes ET
    - Les décodeurs active low (qui génère des maxterms) sont implémentés en utilisant des portes NON-ET, avec EN et les sorties inversés.

4) Réalisation d'une fonction booléenne avec décodeur

Note: Tout circuit combinatoire peut être réalisé avec des décodeurs et une porte OU.

5) Construire un décodeur 3-vers-8 avec deux décodeurs 2-vers-4:

- À partir de la table de vérité du décodeur 3-vers-8, on peut noter que:
    - Quand S2 =0, les sorties Q0-Q3 sont générés comme dans un décodeur 2 vers 4
    - Quand S2=1, les sorties Q4-Q7 sont générés comme dans un décodeur 2 vers 4
    - S2 peut être utilisé comme Enabler qui active ou désactive les deux décodeurs 2 vers 4.
    - On peut utiliser le Enabler pour lier les deux décodeurs ensembles.

B) Les encodeurs:

Ce dispositif possèdent un nombre de lignes en entrée avec seulement une entrée qui est activé à un moment donné. C'est l'opposé du processus du décodeur.

1) Particularités:

- Seul une entrée peut être activé à un moment donné. Si deux entrés sont activé au même moment, les sorties produisent un résultats indéterminé
- Une sortie avec tout zéro peut être générée quand tous les entrés sont à zéro ou quand D0 =1.

C) Les multiplexeurs:

Il sélectionne parmi plusieurs signaux en entrée et le passe vers la sortie.

1) 2-vers-1 multiplexeur:

Lorsque S=0, alors C=A

Lorsque S=1, alors C=B

- Demux, mux 4 vers 1

Chapitre 5: Circuits logiques séquentiels

A) Circuits séquentiels synchrones:

Si on ajoute un signal de synchronisation (impulsion d'horloge) à l'élément de mémoire RS, on obtient un circuit séquentiel synchrone appelé bascule bistable.

Ce type de circuit est plus complexe, mais plus fiables que les circuits séquentiels asynchrone. Les bascules bistables sont: RS, D, T et JK.

1) La bascule SR:

L'Entrée de contrôle est utilisée de deux manière: comme signal ON ou OFF ou comme un signal de synchronisation.

Aussi, il est utile de ne pas permettre le changement d'état. En effet, quand C=0, cela empêche tout changement d'état dans le Latch. Ainsi, le signal de contrôle permet de changer l'état quand C=1.

Table d'état:

R	S	Q_{n+1}
0	0	Q_n (inchangée)

0	1	0 (RESET)
1	0	1 (SET)
1	1	Not allowed

Q_n = état avant le front montant

Q_{n+1} = état après le front montant

- Bascules Synchrones:
    - Définition: C'est un Latch avec une entrée horloge. La sortie du circuit change quand son entrée Horloge détecte un Front. Cette bascule d'appelle clock, définit comme CLK .

2) La bascule JK:

Table d'état:

J	K	Q_{n+1}
0	0	Q_n (inchangée)
0	1	0 (RESET)
1	0	1 (SET)
1	1	Q'_n (complément)

3) La bascule D :

Les sorties changent seulement au niveau de la transition d'horloge.

Table d'état:

D	Q_{n+1}
0	0
1	1

4) La bascule T:

Table d'état:

T	Q	Q’
0	Q_n	Q'_n
1	Complément	

B) Les entrées asynchrones:

Jusqu’à présent nous avons considéré les entrées synchrones uniquement, c’est-à-dire que leur effet sur la sortie est synchronisé avec l’horloge.

Les bascules possèdent des entrées asynchrones. Elles opèrent indépendamment des entrées synchrones et de l’horloge. Elles sont utilisés pour mettre la bascule à 1 ou 0 à tout instant.

1) Les tables d’états :

Les séquences de sorties, entrées, et états des bascules sont listées dans la tables des états

État présent: indique la valeur actuelle de la bascule

État suivant: indique la valeur des bascules après la prochaine impulsion

La sortie représente la valeur après l’impulsion actuelle de l’horloge.

2) Le diagramme des états:

- Les cercles indique l’état actuel, les flèches pointent vers l’état suivant
- Pour x/y, x l’entrée et y la sortie

3) Réduction des états:

Il faut obtenir la table des états à partir du diagramme des états

Si nous trouvons deux états identiques, nous pouvons enlever un et en garder seulement un. Cette méthode permet de réduire le diagramme d’état et permet une lecture plus facile.

Chapitre 6: Registres et compteurs

1) La bascule JK:

present	next	J K
0	0	0 X
0	1	1 X
1	0	X 1
1	1	X 0

2) Les compteurs:

Nous avons deux types de compteurs: les compteur asynchrone et synchrone

Les compteurs asynchrones: Les sorties de la bascule servent d'horloge pour les bascules qui leur sont connectés

Les compteurs synchrones: toutes les bascules sont synchronisés avec la même horloge en même temps. (les plus utilisés)

1) Compteur binaire asynchrone:

Le signal reset: remet à 0 les bascules. Dans ce genre de circuit, pas toutes les bascules change d'états en même temps.

2) Compteurs synchrones:

Dans ce genre de circuit, toutes les bascules reçoivent le même signal en même temps. Aussi, elle changent en même temps.

- Compteurs binaires: Modulo
    - Définition: le nombre d'état que traverse un compteur
    - Par exemple, un compteur modulo 6 possède 6 états, de 0 à 5.
    - Un compteur séquentiel représente un comptage naturel binaire, Il peut être croissant ou décroissant.
- Diagramme d'états:
    - Les états du compteur sont séquentiels. Chaque état suit l'autre dans une séquence. Ce diagramme est utilisé pour représenter ces séquences.
- Table d'états:
    - Cette table est aussi un moyen de représenter ces séquences d'états
    - Comme pour le diagramme d'états, la table représente l'état suivant basé sur l'état présent.
- Les séquences du compteur:

    - Compteur complet de séquences: toutes les séquences du compteur binaire sont traversées. Son modulo est le même que son modulo maximum
    - Compteur à séquences tronquées: c'est le cas où le modulo du compteur est moins que le modulo maximum. Le compteur passe par moins d'états qu'il est possible.
- Procédure de conception d'un compteur synchrone:
    - À partir de la table d'état, obtenir le diagramme d'états ainsi que la table de transition pour le compteur
    - Fournir les entrées pour chaque bascule en utilisant les sorties des autres bascules
    - Utiliser les portes logiques externes pour détecter les séquences spécifiques pour chauques bascules de manière à obtenir la séquence suivante des entrées
    - Utiliser le diagramme de K pour simplifier les sorties disponibles et créer les entrées
    - Réaliser le circuit
- Table de transitions:
    - On doit établir comment les bascules réagissent aux différentes données en entrées, en fonction de leur état présent.
    - Construction:
        - Obtenu en utilisant la table de fonction des bascules
        - Quand la table de fonction indique Q, Q' ou pas de changement, nous indiquons les valeurs binaires

3) Table de transition avec la bascule D

present	next	Input
0	0	0
0	1	1
1	0	0
1	1	1

- Construction de diagramme de K avec la bascule D:
    - Nous avons besoin de:
        - Diagramme d'états pour les séquences en sorties
        - La table des états

        - La table de transition des bascules
    - Étape 1: Établir la table d'état et le diagramme d'état pour les séquences en compteur
    - Étape 2: choisir la bascule et sa table de transition
    - Étape 3: construire la table des états pour les bascules
    - Étape 4: faire une table de K pour chaque sortie
    - Étape 5: Déterminer la somme de produits simplifiées pour les entrées
    - Étape 6: Construire le circuit

** Voir l'exemple avec la bascule T (DCB)