

# Systemes Digitaux

Cours 2

# Au dernier cours

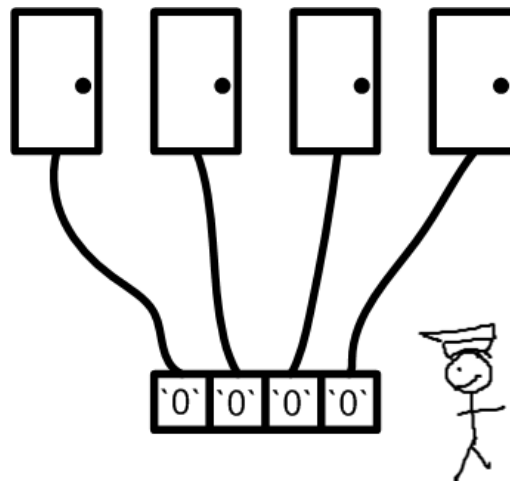
- On a vu les différentes bases et la conversion
  - Décimales
  - Binaires
  - Octales
  - Hexadécimales
- On a parlé des nombres signés:
  - Signe-magnitude
  - Complément à 1
  - Complément à 2

# Au dernier cours

- On a considéré que les nombres binaires étaient des chiffres...
  - Dans certains cas, les nombres binaires représentent autre chose que des chiffres
  - On peut toujours les LIRE comme étant des chiffres, mais ils ne les représentent pas toujours
- Voyons un exemple...

# Binaire: pas toujours un nombre!

- Ex: Un agent de sécurité qui contrôle l'accès à 4 portes:
  - Dans sa machine il peut contrôler le verrouillage de 4 portes différentes
  - Pour débarrer une porte, il doit mettre '1' tandis que pour la barrer, il doit mettre '0'



Il y a 4 bits, mais ça ne représente pas un nombre

# Binaire: pas toujours un nombre!

- Dans le cas précédent, ce n'est pas un nombre:
  - Les opérations comme l'addition et la soustraction ne sont pas très intéressantes
  - Cependant, d'autres opérations pourraient être pertinentes
- Ces opérations sont des opérations LOGIQUES
  - ET, OU, NON, OU-EXCLUSIF, etc.
  - Avant de les voir, on va parler des tables de vérité...

# Table de vérité

- Une table de vérité c'est une table qui:
  - Énumère toutes les entrées possibles en ordre croissant
  - Qui donne la sortie du système dans chaque cas
- Avec N entrées, on aurait  $2^N$  possibilités:
  - On énumère les entrées en leur donnant une valeur qui va de 0 à  $2^N-1$

Avec 2 entrées, on a  $2^2=4$  possibilités

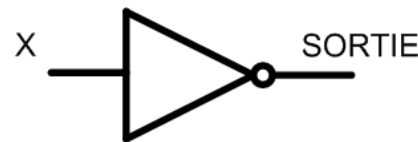
	A	B	SORTIE
0	0	0	1
1	0	1	0
2	1	0	1
3	1	1	1

Allons maintenant examiner les fonctions logiques

# La fonction NON (inversion)

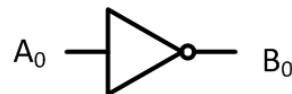
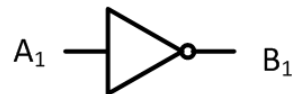
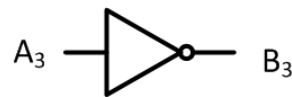
- La fonction NON est une des portes les plus simples et populaires...
  - Son travail est d'inverser ce qu'il y a à l'entrée
  - Ex: "complément à 1"... Chaque bit est inversé

X	SORTIE
0	1
1	0



# Exemple d'application

- Pour effectuer un complément à 1, il faut inverser tous les bits
  - Dans ce cas,  $B_3B_2B_1B_0$  est le complément à 1 de  $A_3A_2A_1A_0$ .
  - Si  $A_3A_2A_1A_0$  était 1101, son complément à 1 serait 0010 (donc -2)





# La fonction ET

- La fonction ET prend 2 (ou plus) entrées et donne 1 sortie:
  - Il faut que TOUTES les entrées soient 1 pour que la sortie soit 1
  - Ex: Dans les robots culinaires, pour partir le moteur, il faut peser sur le bouton ET que le contenant soit fermé

A	B	SORTIE
0	0	0
0	1	0
1	0	0
1	1	1

La fonction "ET"



# La fonction OU

- La fonction OU prend 2 (ou plus) entrées pour donner 1 sortie:
  - Il faut avoir au MOINS un '1' pour avoir '1'
  - C'est un OU inclusif: c'est soit l'un, soit l'autre ou les 2 en même temps

M	N	SORTIE
0	0	0
0	1	1
1	0	1
1	1	1

La fonction "OU"

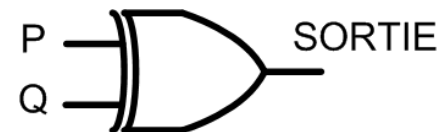


# OU-Exclusif

- La fonction OU-Exclusif prend 2 (ou plus) entrées et donne 1 sortie
  - “C’est soit l’un, soit l’autre mais pas les deux”
  - Un autre terme: “mutuellement exclusif”
- Avec plus de 2 termes, la description change
  - Ça devient “detection d’un nombre impair de 1”

P	Q	SORTIE
0	0	0
0	1	1
1	0	1
1	1	0

La fonction “OU-exclusif”

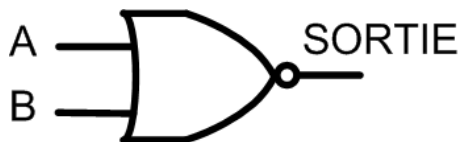


# D'autres fonctions

- Il existe aussi d'autres fonctions:

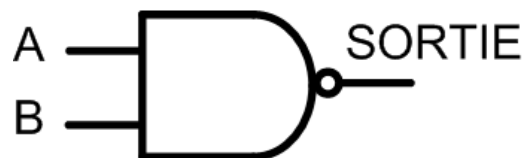
NON-OU

A	B	SORTIE
0	0	1
0	1	0
1	0	0
1	1	0



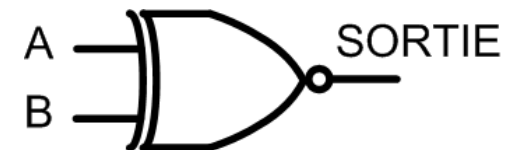
NON-ET

A	B	SORTIE
0	0	1
0	1	1
1	0	1
1	1	0



NON-OU-EXCLUSIF

A	B	SORTIE
0	0	1
0	1	0
1	0	0
1	1	1



# Notation

- Pour écrire les équations logiques, on utilise des notations:
  - ET:  $\bullet$  (parfois, 2 termes collés implique aussi une fonction ET... ABC c'est aussi  $A \bullet B \bullet C$ )
  - OU:  $+$  (Ex:  $A+B$ )
  - NON: (barre en haut)  $\overline{A}$  ou apostrophe  $A'$
  - OU-exclusif:  $\oplus$  Ex:  $A \oplus B$

# Exemple

- Trouvez la table de vérité et dessinez le circuit de la fonction suivante:

$$F = (\bar{A} \bullet B) + (\bar{B} \bullet C)$$

# Exemple

- En connaissant l'équation et les symboles, on peut dessiner le circuit:

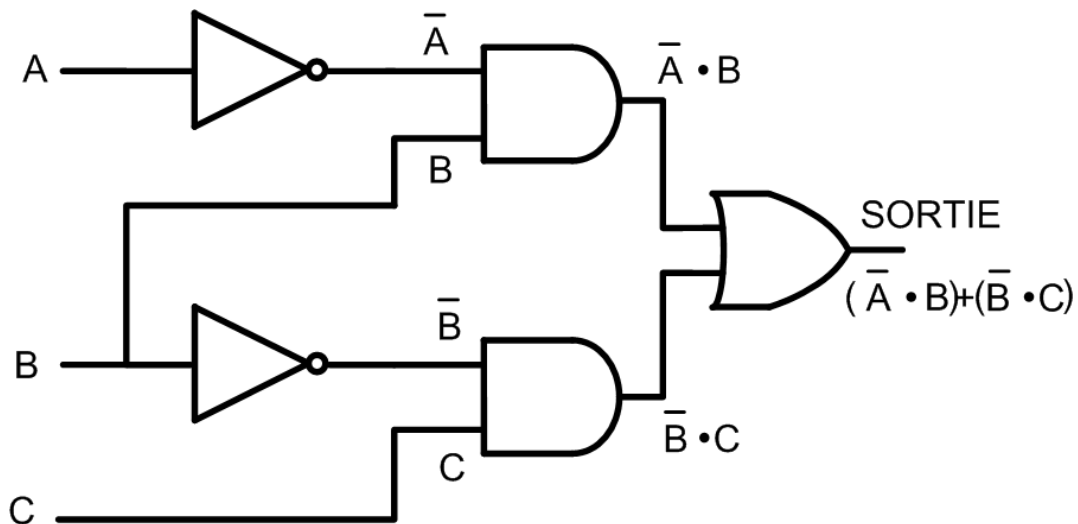
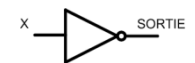
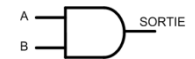
$$F = (\bar{A} \cdot B) + (\bar{B} \cdot C)$$

Rappel

•: ET

+: OU

BARRE: NON



# Exemple

- La table de vérité se trouve progressivement:

$$F = (\bar{A} \bullet B) + (\bar{B} \bullet C)$$

A	B	C	$\bar{A}$	$\bar{B}$	$\bar{A}B$	$\bar{B}C$	F
0	0	0	1	1	0	0	0
0	0	1	1	1	0	1	1
0	1	0	1	0	1	0	1
0	1	1	1	0	1	0	1
1	0	0	0	1	0	0	0
1	0	1	0	1	0	1	1
1	1	0	0	0	0	0	0
1	1	1	0	0	0	0	0



# Exemple (seul)

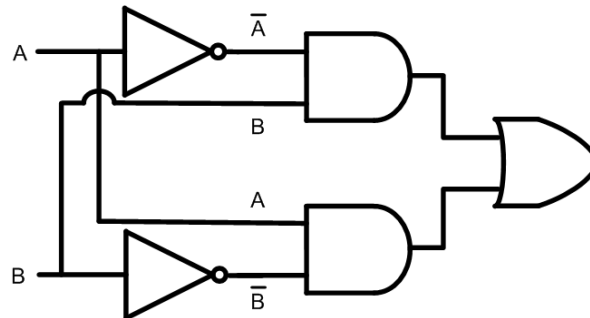
- Dessinez le circuit et trouvez la table de vérité de la fonction suivante:

$$F = (\bar{A} \bullet B) + (\bar{B} \bullet A)$$

# Exemple (seul)

- On y va étape par étape:
  - On sait que A et B seront là
  - On sait que leurs inverses aussi... On met des NON
  - On combine les termes avec ET
  - On combine les termes avec OU

$$F = (\bar{A} \bullet B) + (\bar{B} \bullet A)$$



# Exemple (seul)

- Pour la table de vérité, on met tous les termes
  - A, B, A', B' A'B, AB' et finalement F
  - C'est une suggestion: ce n'est pas obligatoire

$$F = (\bar{A} \bullet B) + (\bar{B} \bullet A)$$

A	B	A'	B'	A'B	AB'	F
0	0	1	1	0	0	0
0	1	1	0	1	0	1
1	0	0	1	0	1	1
1	0	0	0	0	0	0

# Chemin inverse

- On a pris une fonction et on a trouvé sa table de vérité
  - Est-il possible trouver la fonction à partir d'une table de vérité?
  - Parfois c'est plus facile de commencer avec une table de vérité que de commencer avec une fonction
- Reprenons la table de vérité de l'exemple précédent et essayons de trouver la fonction
  - "Faisons le chemin inverse"

# Chemin inverse

- La table de vérité ressemblait à ceci:
  - Comment trouver le circuit qui fera ça...
- On va se poser la question “Quand est-ce que la sortie est ‘1’?”

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

Quand A=0, B=0 et C=1...

Quand A=0, B=1 et C=0...

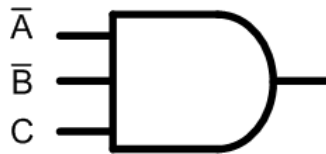
Quand A=0, B=1 et C=1...

ou

Quand A=1, B=0 et C=1...

# Chemin inverse

- Analysons le premier cas:
  - La sortie est 1 quand les entrées sont  $A=0$ ,  $B=0$  et  $C=1$
  - On peut s'exprimer différemment: "Le système donne '1' quand  $A'=1$ ,  $B'=1$  et  $C=1$



- En termes d'équations on peut dire:  $A' \bullet B' \bullet C$
- Il est possible de faire le même exercice pour les autres entrées qui donnent 1...

# Chemin inverse

- Une autre combinaison qui donnerait '1' c'est quand  $A=0$ ,  $B=1$  et  $C=0$ 
  - Ça se traduit en cette equation  $A' \bullet B \bullet C'$
- Les deux autres combinaisons sont  $A' \bullet B \bullet C$  et  $A \bullet B' \bullet C$
- Donc, quand l'une de ces combinaisons donne 1, la fonction donne '1'...
  - Soit la 1<sup>re</sup>, la 2<sup>e</sup>, la 3<sup>e</sup> OU la 4<sup>e</sup>...

$$F = (\bar{A} \bullet \bar{B} \bullet C) + (\bar{A} \bullet B \bullet \bar{C}) + (\bar{A} \bullet B \bullet C) + (A \bullet \bar{B} \bullet C)$$

# Chemin inverse

- Note: l'équation qu'on vient d'avoir est différente de celle qui la précède
  - Or, elles donnent la MÊME table de vérité
  - Elles sont donc équivalentes

$$F = (\bar{A} \bullet B) + (\bar{B} \bullet C) \quad F = (\bar{A} \bullet \bar{B} \bullet C) + (\bar{A} \bullet B \bullet \bar{C}) + (\bar{A} \bullet B \bullet C) + (A \bullet \bar{B} \bullet C)$$

- Il y a plusieurs façons d'arriver à la même table de vérité.
  - On pourra prouver ça avec la logique de Boole (plus tard)

Il y a aussi une autre façon de procéder (beaucoup moins intuitive!)



# Chemin inverse

- On se pose la question: quand est-ce que la sortie est '0'?

A	B	C	F	
0	0	0	0	Quand A=0, B=0 et C=0...
0	0	1	1	
0	1	0	1	
0	1	1	1	
1	0	0	0	Quand A=1, B=0 et C=0...
1	0	1	1	
1	1	0	0	Quand A=1, B=1 et C=0... ou
1	1	1	0	Quand A=1, B=1 et C=1...

Écrivons tout ça sous forme d'équation

# Chemin inverse

- L'équation, une fois écrite, devient:

$$\overline{(\overline{ABC})} + (\overline{ABC}) + (\overline{ABC}) + (ABC)$$

- Par contre, cette équation n'est pas bonne
  - Elle représente les conditions qui donnent 0 à la sortie
  - Pour avoir 1, il faudrait inverser

$$F = \overline{\overline{(\overline{ABC})} + (\overline{ABC}) + (\overline{ABC}) + (ABC)}$$

- On utilise le théorème de DeMorgan (à voir)

# Chemin inverse

- Le théorème de DeMorgan donne l'expression suivante:

$$F = (A + B + C) \cdot (\bar{A} + B + C) \cdot (\bar{A} + \bar{B} + C) \cdot (\bar{A} + \bar{B} + \bar{C})$$

- Sans connaître DeMorgan, il serait possible d'utiliser un truc...

# Chemin inverse

- Recette magique:
  - Regarder toutes les lignes qui donnent 0
  - Chaque variable à 0 donne la variable elle-même
  - Chaque variable à 1 donne la variable inversée
  - Les variables sont connectées par des OU
  - Connecter toutes les lignes avec un ET

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

$$F = (A + B + C) \cdot (\bar{A} + B + C) \cdot (\bar{A} + \bar{B} + C) \cdot (\bar{A} + \bar{B} + \bar{C})$$

# Literal et Terme

- Un **literal** est une variable inversée ou non
- Dans un système où les entrées sont A et B, un literal peut être A, B, A' ou B'
- On combine des littéraux pour former des **termes**: AB', par exemple, est un terme
- Ex: Dans l'équation  $AB'C + A'B'C$ , il y a 2 termes et 6 littéraux

# Minterm

- Un minterm c'est un terme où:
  - **TOUTES** les variables apparaissent exactement 1 fois sous forme normale ou inversée
  - Ces littéraux sont combinés par des ET ("produits")
  - Les minterms sont combinés par des OU ("somme")

$$F = (\bar{A} \cdot \bar{B} \cdot C) + (\bar{A} \cdot B \cdot \bar{C}) + (\bar{A} \cdot B \cdot C) + (A \cdot \bar{B} \cdot C)$$

Il y a 4 minterms

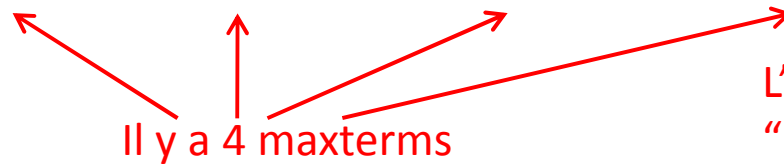
Ce genre d'expression s'appelle parfois "somme de produits"

$$F = \sum (1,2,3,5)$$

# Maxterm

- Un **maxterm** est un terme où
  - **Tous** les variables apparaissent exactement 1 fois
  - Chaque variable peut être lui-même ou inversé
  - Les variables sont séparés par les OU (“somme”)
  - Les maxterms sont connectés par des ET (“produit”)

$$H = (\bar{A} + \bar{B} + \bar{C}) \bullet (A + \bar{B} + \bar{C}) \bullet (A + B + \bar{C}) \bullet (A + B + C)$$



L'expression s'appelle parfois  
“produit des sommes”

$$H = \prod (0,4,6,7)$$

# Définitions

- **Forme canonique:**
  - La forme canonique est une fonction définie avec une somme de minterms ou un produit de maxterms
  - C'est la forme la plus facile à sortir mais est non-optimale (contient toujours un maximum de littéral)
- **Forme standard:**
  - Peut être somme de produits ou produit de sommes
  - Elle peut contenir n'importe quel nombre de littéral



# Définir une fonction

- Pour spécifier une fonction, on a 2 options:
  - La spécifier en termes en minterms
  - La spécifier en termes de maxterms
- Les minterms sont connectés par +:
  - 1 seul '1' suffit pour que la sortie soit '1'
- Les maxterms sont connectés par ●:
  - 1 seul '0' suffit pour que la sortie soit '0'
- Donc: maxterm  $\Rightarrow$  '0' et minterm  $\Rightarrow$  '1'

# Définir une fonction

- Prenons le cas de tantôt...
  - On peut définir la fonction avec des minterms (les '1')
  - On fait ressortir tous les termes qui donnent '1'

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

$$F = (\overline{001} \cdot \overline{010} \cdot \overline{011} \cdot \overline{101}) + (\overline{A} \cdot \overline{B} \cdot C) + (\overline{A} \cdot B \cdot \overline{C}) + (\overline{A} \cdot B \cdot C) + (A \cdot \overline{B} \cdot C)$$

Certains aiment définir les minterms avec des chiffres (décimales): 1,2,3,5...

# Définir une fonction

- Prenons le cas de tantôt...
  - On peut définir la fonction avec des maxterms (les '0')
  - Truc: Inversez les A, B C et mettez des + entre...

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

$$F = (A + B + C) \cdot (\bar{A} + B + C) \cdot (\bar{A} + \bar{B} + C) \cdot (\bar{A} + \bar{B} + \bar{C})$$

000                      100                      110                      111

↓  
Les 3 littéraux doivent être '0', pour que la sortie soit '0'

Il suffit qu'un maxterm soit '0' pour que F soit '0'

On peut aussi définir les maxterms avec des chiffres: 0,4,6,7

# Exemple (seul)

- Vous voulez concevoir un système qui a les spécifications suivantes:
  - Il reçoit 3 bits en entrée: A, B et C
  - Il génère '1' si au moins 2 des entrées est '1'
- Trouvez la table de vérité
- Écrivez l'équation canonique en termes de minterms

# Exemple (seul)

- On écrit toutes les possibilités en entrée:
  - Quand on a au moins 2 '1', la sortie est '1'
  - Sinon, c'est '0'

A	B	C	F
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

$$F = (\overset{011}{\bar{A} \cdot B \cdot C}) + (\overset{101}{A \cdot \bar{B} \cdot C}) + (\overset{110}{A \cdot B \cdot \bar{C}}) + (\overset{111}{A \cdot B \cdot C})$$

# Logique de Boole

- En arithmétique, il y a des règles qu'on prend pour acquis...
- Considérons par exemple:  $3^{12}+3 \times 10^7$ :
  - Les exposants sont évalués en premier
  - La multiplication vient après
  - Finalement, c'est l'addition
- En logique, il existe aussi des règles
  - On va passer quelques diapos à en parler.

# Logique de Boole: ordre

- En logique de Boole, l'ordre est le suivant:
  - 1) Les inversions sont considérés en premiers
  - 2) Par la suite, ce sont les ET
  - 3) Finalement, ce sont les OU
- S'il y avait des parenthèse, ce sont ces expressions qui auront priorité...
- Ce sont des choses qui deviendront naturelles avec l'usage

Compliquons les choses avec des postulats...

# Logique de Boole: Postulats

- Il existe plusieurs postulats qui définissent la logique de Boole:
  - 1) Le système est fermé
  - 2) Il existe des éléments “identité”
  - 3) Les opérations sont commutatives
  - 4) Les opérations sont distributives
  - 5) Il existe un complément
  - 6) Il doit exister au moins 2 éléments différents

Passons au travers rapidement



# Logique de Boole: Postulats

- Le systeme est fermé:
  - Une opération de ET et de OU n'introduiraient pas des éléments hors de notre systeme binaire
  - Pensez, par contraste, à la division de nombres entiers:  $2/3=0.666..$  Le résultat n'est plus un entier
- Il existe des éléments "identité":
  - Des éléments identités sont des éléments qui ne font rien et qui laissent le nombre inchangé
  - Pour une variable X:  $X + 0 = X$  et  $X \bullet 1 = X$
  - Les éléments identité sont '0' pour + et '1' pour •

# Logique de Boole: Postulats

- ET et OU sont commutatifs:
  - “L’ordre n’a pas d’importance”
  - On a donc:  $X + Y = Y + X$
  - Et:  $X \bullet Y = Y \bullet X$
- ET et OU sont distributifs:
  - $X \bullet (Y + Z) = X \bullet Y + X \bullet Z$  (comme en arithmétique)
  - $X + (Y \bullet Z) = (X + Y) \bullet (X + Z)$  (PAS comme en arithmétique!)

# Logique de Boole: Postulats

- Il existe un complément pour tous les termes:
  - Un complément est un élément qui “annule”
  - En arithmétique, par exemple: multiplication de X avec  $1/X$  et addition de X avec  $-X$
  - Avec ET:  $X \bullet X' = 0$
  - Avec OU:  $X + X' = 1$
- Il existe au moins 2 éléments différents:
  - Il existe '0' et il existe '1'... Ça fait 2.
  - Pas très intéressant comme propriété

Ces propriétés deviendront évidentes à force de les utiliser...

# Logique de Boole: Théorèmes

- En plus des postulats, il existe aussi des théorèmes utiles:
  - Associativité de ET et OU:  $X + (Y+Z) = (X+Y) + Z$  et la même chose avec la fonction ET
  - DeMorgan:  $(X+Y)' = X' \bullet Y'$  et  $(X \bullet Y)' = X' + Y'$
  - Absorption:  $X + X \bullet Y = X$  et  $X \bullet (X + Y) = X$
  - Autre théorème:  $X+X=X$  et  $X \bullet X=X$
  - Autre théorème #2 :  $X + 1 = 1$  et  $X \bullet 0 = 0$

Encore une fois, ces propriétés deviendront évidentes à force de les utiliser...

## Exemple (p.46 Ex. 2.1 #1)

- En utilisant les propriétés et les théorèmes, simplifiez l'expression suivante:  $X \bullet (X' + Y)$

# Exemple (p.46 Ex. 2.1 #1)

- On peut commencer avec la distributivité...
  - On peut distribuer  $X$  dans la parenthèse...
- $X \bullet (X' + Y)$  peut s'écrire comme:  $X \bullet X' + X \bullet Y$
- La partie de gauche devient 0
  - $0 + X \bullet Y$
- On sait que  $0 + A = A$  et donc, on obtient  $X \bullet Y$

## Exemple (seul) (p.47 Ex. 2.1 #3)

- En utilisant les propriétés et les théorèmes, simplifiez l'expression suivante:

$$(X + Y) \bullet (X + Y')$$

# Exemple (seul) (p.47 Ex. 2.1 #3)

- On commence par distribuer:  $(X + Y) \bullet (X + Y')$ 
  - C'est comme une multiplication de polynôme
  - On obtient  $X \bullet X + X \bullet Y' + Y \bullet X + Y \bullet Y'$
- On examine partie par partie:
  - $X \bullet X = X$
  - $X \bullet Y'$ ... Pas grand chose à faire
  - $Y \bullet X$ ... Non plus
  - $Y \bullet Y' = 0$
  - On obtient donc:  $X + X \bullet Y' + Y \bullet X + 0$

Le zéro de droite disparaît...



## Exemple (seul) (p.47 Ex. 2.1 #3)

- On recommence donc avec:  $X + X \bullet Y' + Y \bullet X$
- On sait que  $A + A \bullet B = A$  (absorption), donc les 2 premiers termes deviennent X...
  - $X + Y \bullet X$
- Il reste une expression qui peut encore être simplifiée avec l'absorption:
  - La réponse finale est donc égale à X

# Complications et techniques

- La manipulation de termes n'est pas évidente
  - Il n'y a pas de technique systématique pour toujours arriver au meilleur résultat
  - Ça vient avec l'expérience
- Pour rendre le travail plus facile, on présente 2 techniques:
  - Introduire des termes neutres
  - Dualité

# Techniques

- Les termes neutres: on introduit des termes qui ne changent pas l'expression logique
  - Ex: Faire un ET avec 1 ou faire un OU avec 0
  - On pourra par la suite substituer 1 et 0 par des expressions
- Dualité: Une relation avec ET/OU peut être remplacée par une relation OU/ET
  - Il faut seulement changer les 0 par des 1 et vice versa
  - Ex:  $X + 1 = 1$  deviendrait  $X \bullet 0 = 0$

## Exemple (p.47 Ex. 2.1 #4)

- Simplifiez:  $X \bullet Y + X' \bullet Z + Y \bullet Z$

# Exemple (p.47 Ex. 2.1 #4)

- On raisonne:
  - On ne peut pas simplifier parce que les termes contiennent des littéraux différents
  - Pour simplifier, on va faire qu'ils aient les mêmes littéraux
- Pour ce faire, on peut introduire un terme neutre
  - $X \bullet Y + X' \bullet Z + Y \bullet Z \bullet 1$
  - $X \bullet Y + X' \bullet Z + Y \bullet Z \bullet (X + X')$

## Exemple (p.47 Ex. 2.1 #4)

- On distribue les termes:
  - $X \bullet Y + X' \bullet Z + Y \bullet Z \bullet X + Y \bullet Z \bullet X'$
- On voit que les termes de même couleur se simplifient (absorption):
  - $X \bullet Y + X' \bullet Z + Y \bullet Z \bullet X + Y \bullet Z \bullet X'$
  - Ça devient:  $X \bullet Y + X' \bullet Z$
- On se retrouve avec une expression beaucoup plus petite.

# Circuits intégrés

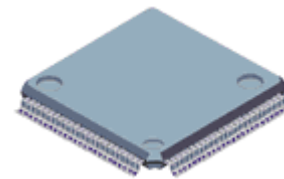
- Les transistors deviennent de plus en plus petits
  - On est capable d'en mettre de plus en plus dans une surface plus petite
  - Il est possible de classier les niveaux d'intégration (selon la taille et la quantité de transistors)
- Les boîtiers différents donnent une IDÉE de quel niveau d'intégration (pas vrai à 100%)



Un seul transistor



Vieux boîtier... "petite"  
quantite de transistors



Boîtier plus recent  
Plus de transistors

# Circuits integres

- Le niveau d'integration le plus faible c'est le SSI (Small scale integration): Quelques portes
- Quand on a 10-1000 portes logiques, on parle de MSI (Medium scale integration)
- Avec des dizaines de milliers de portes, on parle de LSI (Large scale integration)
- Avec des centaines de milliers (ou plus) on parle de VLSI (very large scale integration)

Certains parlent de ULSI (ultra large scale integration) mais ca devient ridicule...  
On va arreter les termes a VLSI...



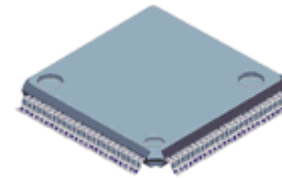
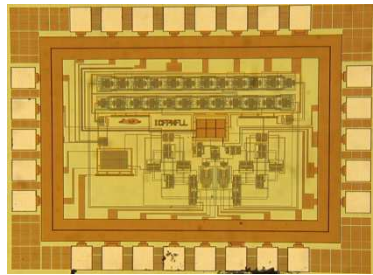
# Methodes d'implémentation

- Il existe plusieurs facons d'implementer des circuits logiques:
  - Composants discrets
  - Logique programmable
  - Circuit dedie
- L'usage des composants discrets est la methode la plus commune:
  - On les connecte les puces ET/OU/... sur une plaquette

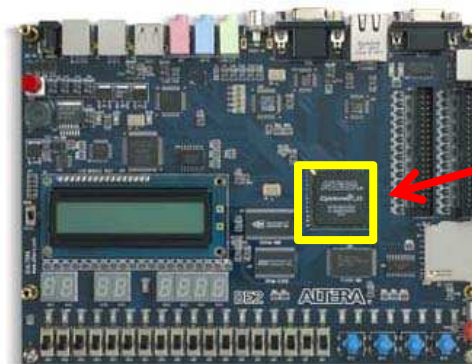


# Methodes d'implémentation

- On pourrait aussi concevoir le circuit et envoyer faire fabriquer une puce dédiée



- Il y a des puces qui ont déjà des portes
  - On peut définir comment ils sont connectés à l'interne



FPGA

# Methodes d'implémentation

- Dans ce cours on va se concentrer sur la logique programmable
  - Les puces utilisées sont les FPGA: field programmable gate arrays
  - Il y a plusieurs façons de spécifier le genre de circuit qu'on veut avoir: en dessin ou en programmation
  - On va apprendre à utiliser les 2 méthodes dans les laboratoires

# Methodes d'implémentation

- On va passer rapidement par les étapes nécessaires pour implémenter en FPGA
  - 1) On dit à l'outil QUEL FPGA on utilise
  - 2) On spécifie le circuit (dessin ou programme)
  - 3) On simule pour voir si ça fonctionne bien (sinon, on retourne à 2)
  - 4) Pour programmer le FPGA, on spécifie les entrées et les sorties
  - 5) Finalement, on programme le FPGA