Travail en architecture matérielle (Partie 1 : logique combinatoire)

*Des opérations de base:*

1)Trouver en ligne comment sont fabriqués les portes logiques not or et and.(utilisation de transistors)

2)Donner les tables de vérité de ces 3 fonctions logiques;

3)Donner celles du nor (ou non) nand et xor.

4)Vérifier que a.(b+c)=a.b+a.c et a+(b.c)=(a+b).(a+c) (+ pour or et . pour et.

 mais attention au + ce n'est pas l'addition arithmétique c'est l'addition logique : ou)

5)Vérifier que a+non(a).b=a+b

6)De même on a a+non(a)=1 a.non(a)=0 1+a=1 1.a=a 0+a=a et 0.a=0

Vérifier que non(a+b)=non(a).non(b) et non(a.b)=non(a)+non(b)

On parle d'algèbre deBoole puisque l'ensemble {0;1} avec les 2 opérations . et + (et ou)

a une structure d'algèbre(opération non en plus).

*Simplification de portes logiques:*

7) a b c=f(a,b)

 0 0 1

 0 1 1

 1 0 1

 1 1 0

 Méthode :on regarde les 1 dans c : c==non(a).non(b) + non(a).b +a.non(b)

 et on simplifie par les règles algébriques.

 Quelle fonction retrouve t'on?

8)On peut utiliser la méthode des tableaux de Karnaugh qui utilise le code de Gray ou code binaire réflexif

 a 0 1

 b

 0 1 1

 1 1 0

 on trouve directement non(a)+non(b)

 9) *Multiplexeur:*

 Regarder en quoi consiste un multiplexeur 1 bit (3 entrées x0,x1,Eo et une sortie y)

 puis 2 bits (6 entrées x0 x1 x2 x3 E0 E1 et une sortie y.

 En donner les tables de vérité , simplifier et proposer un circuit qui les réalise.

 On peut aussi utiliser un décodeur d'adresse puis un selecteur d'entrée pour réaliser un multiplexeur.

 10)*Additionneur*

 Quelques propriétés utiles du Xor

 Rappel a xor b = non(a).b +a.non(b)

 (a xor b)xor c = a xor (b xor c)

 non(a xor b)=non(a).non(b)+a.b

 demi-additionneur 1 bit (pas de retenue en entrée une unité

 et une deuzaine(retenue) en sortie:

 Donner la table de vérité des sorties et un circuit qui le

 réalise.

 Comment réaliser un additionneur 1 bit complet à l'aide de 2

 1/2.

 additionneur un bit complet

 Donner la table de vérité des sorties (s pour somme et r

 pour retenue)

 Exprimer s à l'aide de xor.

 Algorithmique

 Ecrire un algorithme qui réalise l'addition de deux entiers

 codés sur n bits chacun en utilisant des booléens.

Pour aller plus loin

 Revenir sur la page précédente et ouvrir le lien

 "additionneur complet".

 La propagation de la retenue dans le cas du premier montage

 qui utilise des additionneurs 1 bit complet ralentit la

 réalisation de l'addition des 2 entiers (il faut attendre la

 retenue précédente).

 L'idée est de faire directement un additionneur 4 bits avec

 9 entrées et 5 sorties (4 bits pour le résultat et un pour

 la retenue r4).

 Du document distribué exprimer les sorties 1 2 3 4 et

 vérifier que ce sont bien les expressions attendues.

 11)*UAL*

 Voir le principe de fonctionnement simplifié:

 Utilise un multiplexeur et des fonctions de base comme l'additionneur.

 Exposés possibles:

 Tableaux de Karnaugh

 Utilisation du logiciel logisim