



Architectures Logicielles et Matérielles

Travaux Dirigés

Circuits



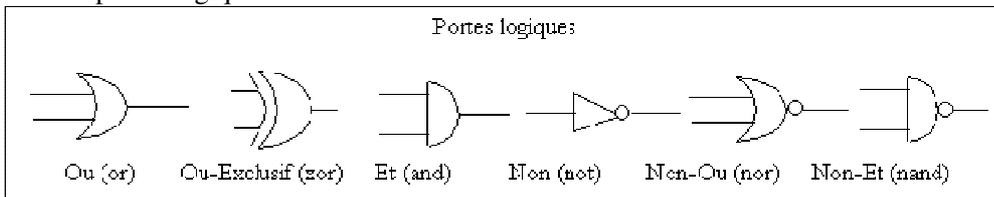
– Attention –

– En cours de rédaction –

– à compléter –

Rappel :

dessins des portes logiques.



Déroulement envisagé :

sur 7 séances

séance 1 : une première séance sur le binaire (représentation de l'information, des nombres et un peu d'arithmétique)

séances 2, 3, 4 : trois séances sur les circuits combinatoires, pour commencer des tables de vérité, de la logique et des circuits de base (séance 2), ensuite une UAL et/ou un circuit à n entrées simple (séance 3) et enfin un circuit plus complexe (récuratif, par ex.) ou un exercice méta

séances 5, 6, 7 : trois séances sur les circuits séquentiels, par exemple un automate (éventuellement la partie combinatoire de l'automate peut avoir été vue dans les séances combinatoires), un circuit à flot de données (séance 6), une pc/po (séance 7).

(en ajoutant une séance de CC ou de correction de CC, et 2 séances sur de la microprogrammation, on obtient 10 séances)

Séance 1.1 Représentation de l'information

Donner le nombre d'éléments différents que l'on peut coder avec 1 bit, 8 bits, 10 bits, 32 bits.

Etablir les tables de correspondances sur 4 bits entre vecteur de bit de nombre pour les représentations courantes (binaire pur, C1, C2, excédent)

Donner des représentations de nombres avec virgules (virgule fixe, virgule flottante)

Séance 1.2 Représentation des entiers

Combien faut-il de bit au minimum pour coder 2011 ? (est-ce pareil pour votre année de naissance ?)

Coder 2011 en binaire pur, et en C2. (pareil pour votre année de naissance) Donner les résultats en hexadécimal également.

Combien faut-il de bit au minimum pour coder -2011 ?

Coder -2011 en C2. (pareil pour l'opposé de votre année de naissance) Donner les résultats en hexadécimal également.

Séance 1.3 Arithmétique sur les entiers

Faire en binaire les opérations suivantes :

2011 + votre année de naissance

2011 - votre année de naissance

$7*31, 4*30, (7*31)+(4*30)+28, [(7*31)+(4*30)+28] / 7$. Qu'obtenez-vous ?

Séance 1.4 Code Barre

Essayer d'intuiter le codage (version simple premiers chiffres de l'ean 13), et après explication du codage des 12 premiers chiffres, trouver une méthode pour en coder un 13 ième.

Séance 1.5-8 Autres codages

Etudier les codages ISBN, BCD, UTF, EAN, Détecteur et correcteur d'erreur, ... (d'après sujets d'examen, par exemple)

Séance 2.1 Table de vérité vers formule logique

Donner la forme normale disjonctive correspondant à cette table de vérité :

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

Séance 2.2 Implication et Modus Ponens

Un implication logique $A \Rightarrow B$ est fausse seulement quand A est vrai et B est faux en même temps.

- Retrouver la table de vérité de l'implication $A \Rightarrow B$.
- Donner la forme normale disjonctive correspondant à cette table de vérité.

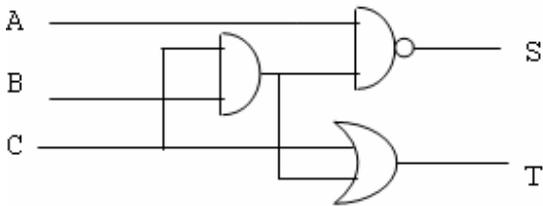
- En déduire un circuit réalisant l'implication.

Le modus ponens utilisé en logique est donnée par la formule $[A \text{ et } (A \Rightarrow B)] \Rightarrow B$.

- Dessiner un circuit correspondant au modus ponens.
- Donner la table de vérité du modus ponens. Qu'observez-vous ? Quelle conclusion en tirer pour le dessin du circuit précédent.

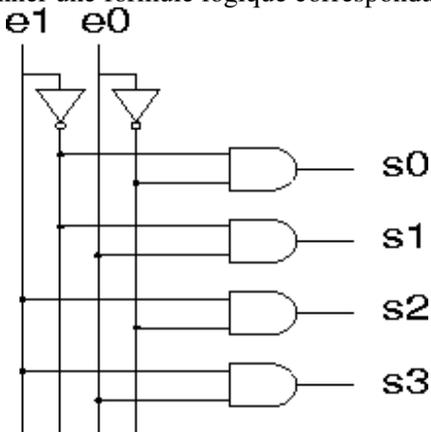
Séance 2.3 Circuit mystère simple

Donner une/des formule/s logique/s correspondant au circuit suivant :



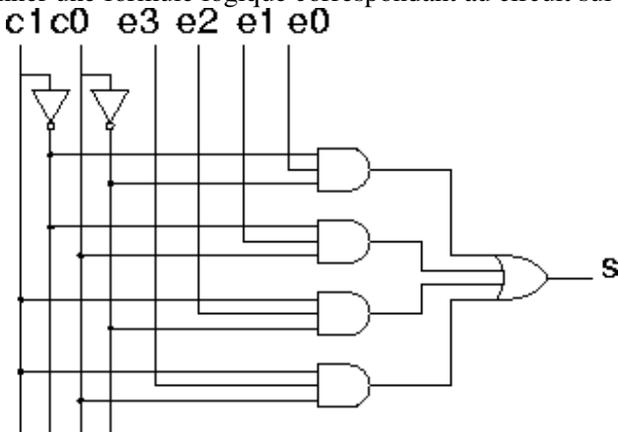
Séance 2.4 Circuit à reconnaître

Donner une formule logique correspondant au circuit suivant :



Séance 2.5 Circuit à reconnaître

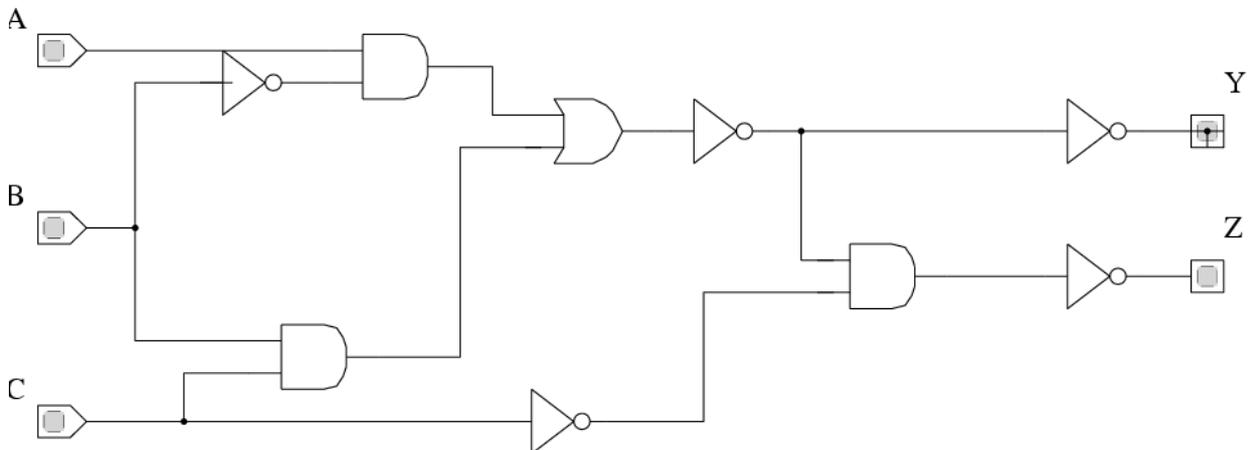
Donner une formule logique correspondant au circuit suivant :



Donner la table de vérité correspondante et reconnaître le circuit.

Séance 2.6 Circuit à simplifier

Donner une formule logique correspondant au circuit suivant :



Séance 2.7 Circuits de base

Pour chacun des circuits dont la définition suit (rédigée pour 3 entrées arbitrairement), le même travail est demandé :

- Donner la table de vérité du circuit.
- En déduire une formule logique correspondant à cette table de vérité.
- Simplifier cette formule si c'est possible.
- Dessiner le circuit représentant cette formule logique.
- Donner le coût de ce circuit (en nombre de portes), et le délai maximum (en nombre de niveaux de portes).

Pour varier, la formule logique peut être :

- Une forme normale disjonctive.
- Une forme normale conjonctive.
- Une forme en Nand de Nand.
- Une forme en Nor et Nor.
- Une forme en Xor.

Pour varier, le dessin peut être :

- Sous forme d'arbre.
- Sous forme de PLA (Programmable Logic Array).

Le choix de la forme logique et du dessin peut se faire en fonction du circuit.

Pour approfondir, certains circuits décrit pour un nombre fixe de bits s'étendent à un nombre plus grand de bits, un nombre puissance de 2 de bits ou un nombre arbitraire de bits, étudier ces possibilités.

Les circuits :

si nécessaire, les trois entrées sont notés E_0, E_1 et E_2 . Lorsque le circuit n'a qu'une sortie, celle-ci est notée S , sinon elles sont notées S_0, S_1, \dots, S_n .

Majorité à 3 entrées : le circuit a une seule sortie ayant la valeur apparaissant majoritairement parmi les entrées.

Un seul Vrai à 3 entrées : le circuit a une seule sortie de valeur vrai si et seulement si une seule de ses entrées est à vrai. Autrement dit : $S=(E_0+E_1+E_2=1)$.

Parité à 3 entrées : le circuit (apparenté au Xor) a une seule sortie de valeur vrai si et seulement si un nombre pair de ses entrées est à vrai. Autrement dit : $S=(E_0+E_1+E_2=0[2])$.

Compteur de bits Vrai à trois entrées : le circuit a deux sorties donnant le nombre d'entrées à vraie. Autrement dit : $(S_1S_0)_2=E_0+E_1+E_2$.

Encodeur à 3 entrées : le circuit a 2 sorties dont la valeur donne l'indice du premier fil d'entrée à vrai, ou si aucun fil n'est à vrai donne la valeur 3.

Discuter du cas où l'on peut garantir qu'il y a toujours une et une seule entrée à vrai.

Décodeur à 3 entrées : le circuit a 8 sorties (2^3) dont une seule est à vrai, celle dont l'indice correspond au nombre binaire donné en entrée sur trois bits. Autrement dit : $S_i=((E_2E_1E_0)_2=i)$.

Multiplexeur à 3 entrées : le circuit a une sortie qui vaut E_0 si E_2 est à 0 et E_1 si E_2 est à 1. Autrement dit : $S=E_{E_2}$.

Demultiplexeur à 3 entrées : le circuit a 4 sorties dont une seule est déterminée, c'est celle dont l'indice correspond au nombre $(E_1E_0)_2$ et sa valeur correspond à E_2 . Autrement dit : $S_{(E_1E_0)_2}=E_2$.

Discuter selon la valeur des fils non donnés dans la définition.

Séance 2.8 Prochain nombre premier

Donner un circuit ayant en entrée n bits et n bits en sortie valant 0 si l'entrée correspond à un nombre premier, ou la valeur de l'entier le plus proche de l'entrée, qui lui est supérieur, et qui est premier.

Variante : un reconnaisseur de nombre premier.

Indication : ne pas chercher un circuit général pour n quelconque mais fixer un petit n et faire l'exercice en conséquence.

Séance 2.9 Tables de vérité

Chercher les tables de vérités, ou la forme des tables de vérités des exercices prévus en séance 3.

Séance 2.10 Tables de vérité

Donner la table de vérité des fonctions suivantes :

Next-Prime

lème décimale de Pi

Factorielle

PGCD

...

Séance 2.11 Petites multiplications

Observer comment avec un ou deux décaleurs et un additionneur (éventuellement pour faire une soustraction), on peut faire la plupart des petites multiplications (multiplication par un petit nombre)

Séance 2.12 Circuit logique programmable. (barème indicatif : 5 points)

Décrivez, comparez, commentez, expliquez l'un ou l'autre ou l'ensemble de ces images obtenues sur Internet en réponse à la requête « PLA » (PLA=Programmable logical array)

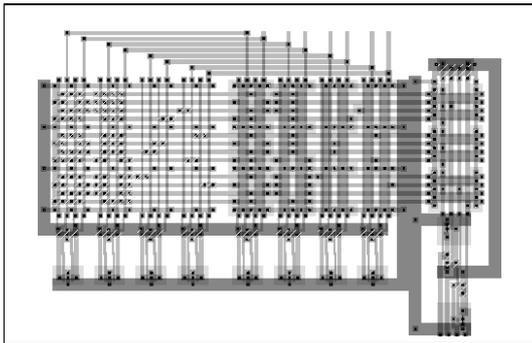


Figure 1

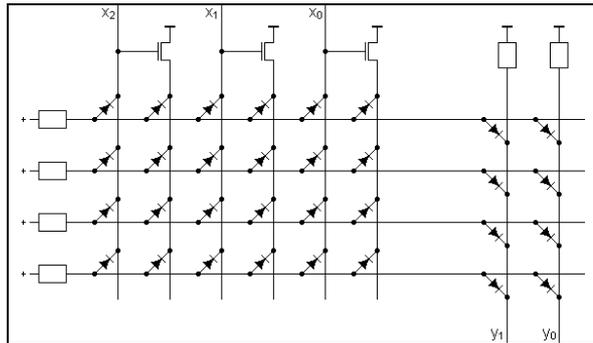


Figure 2

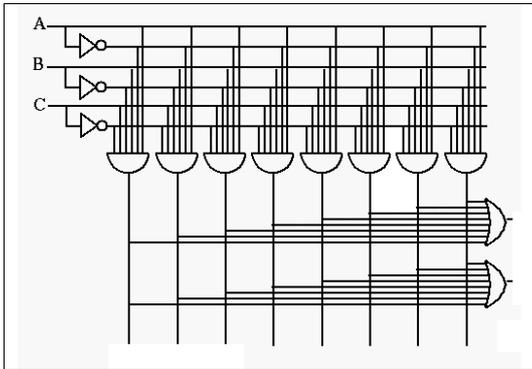


Figure 3

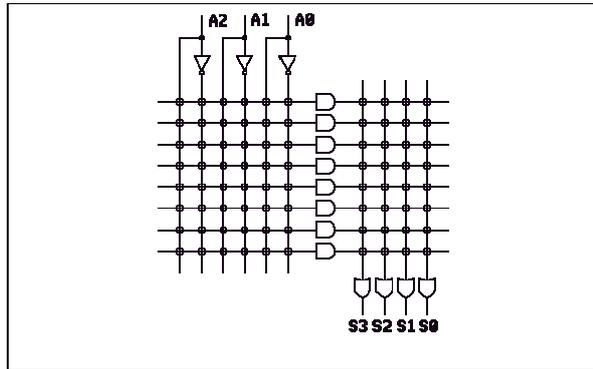


Figure 4

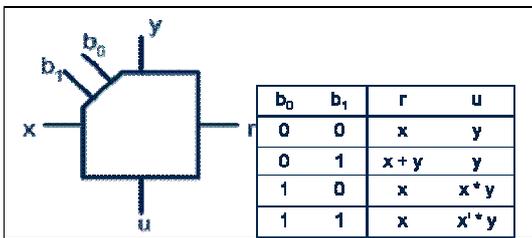


Figure 5

Séance 3.1 Multiple de 21

Donner un circuit ayant en entrée 8 bits et une sortie vraie si et seulement si le nombre en entrée est multiple de 21.

On pourra essayer de chercher d'abord deux circuits pour reconnaître les nombres respectivement multiples de 3 et de 7 puis de les combiner.

Indication : on peut savoir si un nombre est un multiple de 3 ou de 7 comme on le fait pour savoir si un nombre est un multiple de 9 en base 10, il suffit de se placer en base 4 et en base 8 et de faire l'addition de ses chiffres.

Séance 3.2 Incrémenteur n bits.

Il s'agit de faire un circuit à n bits en entrée qui délivre en sortie sur n bits le plus rapidement possible la somme du nombre donnée en entrée et de 1.

Une première solution consiste à faire un additionneur. Regarder alors le temps nécessaire au calcul de l'addition (temps minimum, temps moyen, temps maximum).

Chercher des solutions pour aller plus vite.

Séance 3.3 UAL sur n bits

Commencer par faire un additionneur 1 bit ne prenant que 2 entrées (les deux opérandes).

Considérant que l'addition se fait souvent avec une retenue, faire un additionneur 1 bit prenant 3 entrées (les deux opérandes et la retenue entrante) et fournissant 2 sorties (le résultat et la retenue sortante). On appellera cet additionneur, l'additionneur complet. Remarquer qu'il y a deux solutions, celle associant deux additionneurs tels que faits à la question précédente, et celle repartant de zéro.

En associant plusieurs additionneurs complet, faire un additionneur n bits ($2n+1$ entrées).

Continuer par un soustracteur n bits. Indication : comparer la table de vérité de $A+B+C_{in}=(C_{out}Res)$ avec celle de (qui correspond à la soustraction, A, B C_{in} sont les entrées, $C_{out} Res$ sont les sorties telles que la soustraction tombe juste) et celle de $(\overline{C_{out}}A)-\overline{B}-\overline{C_{in}}=Res$. Faire le lien avec la représentation des nombres en compléments à 2 et la méthode pour obtenir le complément $-B$ à partir de B , i.e. $-B=\overline{B}+1$ (car on a $B+\overline{B}=-1[2^n]$).

Combiner dans un même circuit un additionneur et un soustracteur. Ajouter une entrée pour choisir entre l'additionneur et le soustracteur.

Finir par une UAL qui fasse 4 opérations : Addition, Soustraction, Et et Premier opérande.

Profiter pour faire le lien avec les 4 drapeaux Z (zéro), N (négatif), C (retenue sortante), V (dépassement).

Variante : les 4 opérations peuvent aussi être complément du premier opérande, incrément du premier opérande, double du premier opérande, OuExclusif, NonEt.

Variante : on peut faire 8 opérations.

Séance 3.4 Conversion Binaire DCB

En DCB (Décimal codé binaire), les nombres sont codés à partir de leur représentation décimale, chaque chiffre en décimal est codé par un quartet de bit. Exemple : 36 est codé par un octet (0011 0110).

Définir un circuit qui réalise la conversion Binaire (9 bits) => DCB (10 bits).

Séance 3.5 Comparaison sur n bits

Donner un circuit ayant de entrées sur n bits et une seule sortie sur un bit vraie si la première entrée est supérieure à la seconde, fausse sinon.

Discuter selon le codage des entrées.

Donner une solution avec le circuit d'addition/soustraction de l'UAL.

Séance 3.6 Décaleur (n,d)

Il s'agit de faire un circuit à $n+d$ bits en entrée qui délivre en sortie sur n bits l'entrée décalée de K bits où K est l'entier donnée par les d bits supplémentaires.

Commencer par le décaleur (n,1).

Généraliser par le décaleur (n,d).

Chercher une solution par association de décaleur (n,1).

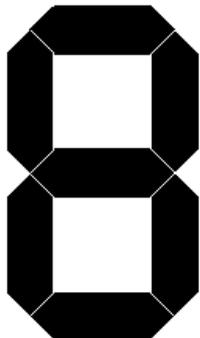
Chercher une autre solution en bouclant sur un décaleur (n,1).

Séance 4.1 Afficheur 7 segments

Définir des représentations distinctes par l'afficheur 7 segments des 10 chiffres 0, 1, ..., 9 et des 6 lettres A, B, ..., F.

Définir un circuit prenant en entrée un symbole parmi 0, 1, ..., F et en sortie les 7 fils de commande pour allumer ou éteindre les 7 segments de la représentation du symbole.

Indication : On pourra travailler segment par segment.



Définir un circuit permettant de faire dérouler l'affichage selon un cycle croissant ou décroissant.

Discuter des circuits combinatoires selon la représentation des symboles d'entrées.

Discuter des circuits séquentiels selon l'objet sur lequel porte le déroulement (symbole d'entrée ou commandes d'affichages)

Variante : On peut réduire le travail à l'affichage des seuls symboles numériques, mais est-ce plus simple ?

Séance 4.2 Combinatoire des circuits et connexité

Rappeler combien il y a de circuits à n entrées et 1 sortie réalisant des fonctions logiques différentes.

Donner pour $n=0$ les circuits possibles.

Donner pour $n=1$ les circuits possibles. Pour les circuits connus donner le nom et le dessin officiel.

Donner pour $n=2$ les circuits possibles. Pour les circuits connus donner le nom et le dessin officiel.

Les circuits dont la zone formée par les valeurs à vrai de la table de vérité est connexe sont dit connexes.

Pour $n=2$ combien de circuits sont connexes, combien ne le sont pas ? Pour ceux qui ne le sont pas, donner un circuit équivalent construit à partir de circuits connexes ayant un minimum d'étage.

Donner pour $n=3$ une estimation du nombre de circuits non connexes. Donner un exemple. Pour cet exemple, donner un circuit équivalent construit à partir de circuits connexes ayant un minimum d'étage. Chercher le cas le pire.

variante : on peut considérer la connexité sur la table de vérité définie sur un tore.

Séance 4.3 Multiplieur sur n bits

Commencer par faire un additionneur n bits.

Continuer en associant des additionneurs.

Chercher une autre solution en bouclant sur un additionneur.

Séance 4.4 Addition récursive de Von Neumann

L'addition récursive de Von Neumann repose sur 2 principes : 1 - anticiper les retenues possibles lors des calculs d'une addition en effectuant 2 fois plus de calculs à un moment donnée (un calcul avec une hypothétique retenue entrante à 0 et un calcul avec une hypothétique retenue entrante à 1) ; 2 - décomposer les calculs de manière dichotomique.

Q1. Cas de base. Pour la somme de 2 nombres (A, B) codés sur 1 bit, donner la table de vérité du circuit de base qui effectue les calculs en doubles : avec une hypothétique retenue entrante à 0 et avec une hypothétique retenue entrante à 1. En entrée, il doit y avoir 2 nombres (A, B) codés sur 1 bit (A_0, B_0), en sortie, il doit y avoir 2 résultats (R, S), codés sur 2 bits ($R_1 R_0, S_1 S_0$), et l'on doit avoir $R=A+B, S=A+B+1$.

À partir de cette table de vérité, on suppose la définition d'un circuit $C_1 : (A_0, B_0) \Rightarrow (R_1 R_0, S_1 S_0)$. Donner le circuit correspondant.

Q2. Premier niveau. A l'aide de multiplexeurs, assembler 2 circuits C_1 obtenus à la question précédente pour obtenir un circuit C_2 effectuant la somme de 2 nombres (A, B) codés sur 2 bits, en effectuant les calculs en doubles : avec une hypothétique retenue entrante à 0 et avec une hypothétique retenue entrante à 1. En entrée il doit y avoir 2 nombres (A, B) codés sur 2 bits ($A_1 A_0, B_1 B_0$), en sortie, il doit y avoir 2 résultats (R, S), chacun sur 3 bits ($R_2 R_1 R_0, S_2 S_1 S_0$), et l'on doit toujours avoir $R=A+B, S=A+B+1$.

Indications : Pour obtenir cet assemblage, utiliser un premier circuit C_1 avec (A_0, B_0), et un second circuit C_1 avec (A_1, B_1). Combiner les résultats de ces circuits avec des multiplexeurs pour obtenir ($R_2 R_1 R_0, S_2 S_1 S_0$). $R_0 S_0$ est obtenu à partir du premier circuit C_1 . Les deux autres sorties de ce premier circuit C_1 servent à commander les multiplexeurs (ce sont les retenues entrantes effectives du second circuit, elles peuvent valoir toutes les deux 0, ou toutes les deux 1, ou être différentes). En entrées de ces multiplexeurs on trouve également les sorties du second circuit C_1 . (Pas de table de vérité à cette question, juste le circuit)

Q3. Généralisation. Comme dans la question précédente, à l'aide de multiplexeurs et de 2 circuits C_n effectuant la somme de 2 nombres (A, B) codés sur n bits et donnant 2 résultats (R,S), chacun sur n+1 bits tels que $R=A+B, S=A+B+1$, obtenir un circuit C_{2n} effectuant la somme de 2 nombres (A, B) codés sur 2n bits et donnant 2 résultats (R,S), chacun sur 2n+1 bits tels que $R=A+B, S=A+B+1$.

Q4. Complexité. D'après votre table de vérité (Q1), évaluer la complexité du circuit C_1 . À partir de Q2, évaluer la complexité du circuit C_2 . Enfin, évaluer la complexité du circuit C_{32} et la comparer avec la complexité du circuit d'addition classique.

Séance 4.5 Comparaison sur 8 bits

Q1. Donner la table de vérité de la fonction binaire de comparaison ($\text{Sup}_{\geq}(A,B)$) pour des entrées A, B sur 1 bit : $\text{Sup}_{\geq}(A,B) = A \geq B$.

Q2. A partir de ce qui précède, en supposant que vous disposez de circuits $\text{Sup}_{\geq,1\text{bit}}$ et en s'inspirant de l'un des algorithmes ci-après proposer un circuit combinatoire qui réalise la comparaison entre 2 nombres.

Algorithme de comparaison : pour A, B deux entiers représentés sur n=7 bits (A_n, \dots, A_0), (B_n, \dots, B_0)

Formulation récursive :

$\text{Sup}_{\geq}(A,B,n)$: si $n=0$ alors retourne $A_0 \geq B_0$
sinon si $A_n = B_n$ alors retourne $\text{Sup}_{\geq}(A,B,n-1)$
sinon retourne $A_n \geq B_n$.

Appel : $\text{Sup}_{\geq}(A,B,8)$.

Formulation itérative :

$\text{Sup}_{\geq}(A,B)$: $i=7$; tant que $i>0$ et $A_i = B_i$ faire : $i \leftarrow i-1$; fin tant que ; retourne $A_i \geq B_i$.

Séance 4.6 Circuits réversibles. (barème indicatif : 7 points)

Dans ses leçons sur l'informatique, Richard Feynman, prix Nobel de physique en 1965, s'intéresse à des circuits « exotiques » : « Les opérations AND et NAND (ainsi que OR et XOR) sont des opérations *irréversibles*. J'entends par là qu'il est impossible de récupérer le signal d'entrée à partir du signal de sortie : l'information initiale est perdue et l'est de manière irréversible. Une sortie de porte AND à quatre entrées qui

vaut 0 peut provenir de quinze combinaisons des signaux d'entrée, et rien ne nous permet de deviner laquelle (en revanche, si la sortie vaut 1, nous pouvons en déduire quelle est la combinaison à l'entrée !). Je voudrais introduire le concept d'opération réversible comme d'une opération dont le signal de sortie recèle suffisamment d'informations pour permettre d'en déduire l'entrée.»

QII-1. Le texte de R. Feynman comporte quelques implicites. Quelles sont les quinze + 1 combinaisons dont il est question ?

QII-2. Montrez que la porte XOR est irréversible.

QII-3. Qu'en est-il de la porte NON ? Est-elle réversible ? Pourquoi ?

Plus loin dans son texte, R. Feynman donne l'exemple d'une porte réversible avec la porte CN (Controlled Not) ayant deux entrées (A, B) et deux sorties (A', B') :

A	B	A'	B'
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

QII-4. Réalisez cette porte à l'aide d'un circuit combinatoire simple (comportant seulement NOT, AND et OR). Donnez sa complexité.

QII-5. Quelle est la table de vérité du circuit comportant deux portes CN, les sorties de la première servant d'entrée pour la seconde.

QII-6. Rappelez la table de vérité du multiplexeur 2→1 (Mux 2→1) et un circuit réalisant cette fonction. Ajoutez, si nécessaire, entrée(s), sortie(s) et porte(s) logique(s) pour obtenir un circuit réversible.

QII-7. A partir du circuit obtenu à la question précédente donnez le circuit d'un multiplexeur 8→1 (Mux 8→1) réversibles.

19 Combinatoire ou séquentiel ?

Boucler la retenue sortante d'un additionneur 1 bit sur la retenue entrante.
Étudier le résultat.

20 Automate reconnaisseur de séquence

Pour une expression régulière de votre choix (par exemple $B(L|A)^*P|$) décrire un circuit reconnaisseur.

21 Un circuit séquentiel d'accumulation

On dispose de registres de toutes tailles et d'un circuit d'addition 8 bits pour faire un accumulateur 16 bits. ($Accu \leq Accu + Entree$).

Proposer un circuit séquentiel qui fasse l'accumulation en deux étapes poids faible (f) puis poids fort (F) avec gestion de la retenue intermédiaire : $Accu_f + Retenu \leq Accu_f + Entree_f$ puis $Accu_F \leq Accu_F + Entree_F + Retenu$.

22 Suite de Syracuse sur n bits

Définir un circuit qui produise la suite de Syracuse sur n bits.

Rappel : la suite de Syracuse à partir de U est donnée par $U_0=U$ et $U_{n+1}=U_n/2$ si U_n est pair, $3U_n+1$ sinon.

23 Chronomètre octal

Dessiner un circuit réalisant un décompte de 0 à 7. C'est à dire un circuit avec une horloge en entrée, et une sortie qui change de valeur à chaque top d'horloge et vaut respectivement 0, 1, ..., 7 puis recommence avec le même cycle 0, 1, ..., 7 indéfiniment.

Discuter en fonction du codage des sorties 0, 1, ..., 7.

Transformer ce chronomètre octal sur un chiffre en un chronomètre octal sur n chiffres.

Discuter de l'intérêt d'avoir un additionneur binaire.

Variante : Chronomètre binaire, ou décimal, ou hexadécimal, ou ...

24 Compteur sur n bits

Dessiner un circuit ayant deux entrées dont une horloge et n sorties donnant le nombre de valeurs en entrée ayant valeur Vrai au top de l'horloge.

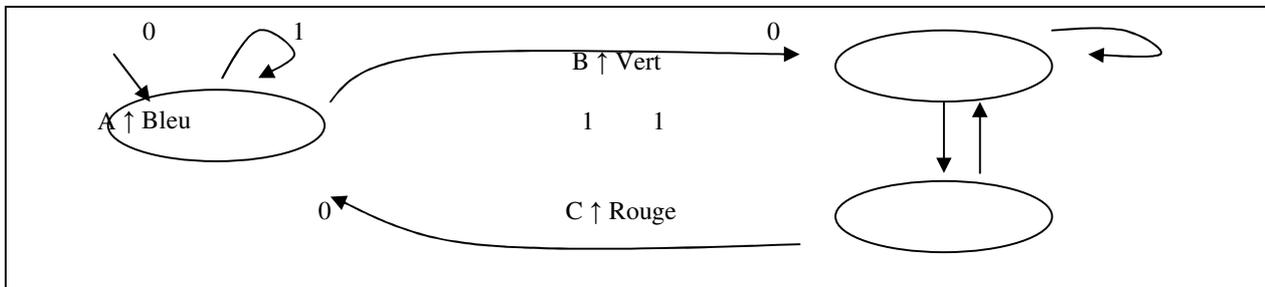
Discuter de l'intérêt d'avoir en sortie un signal de dépassement de capacité.

Variante : Ajouter en entrée un signal reset, et/ou un signal -1.

Discuter de l'intérêt d'avoir un additionneur binaire.

Petit Automate. (barème indicatif : 6 points)

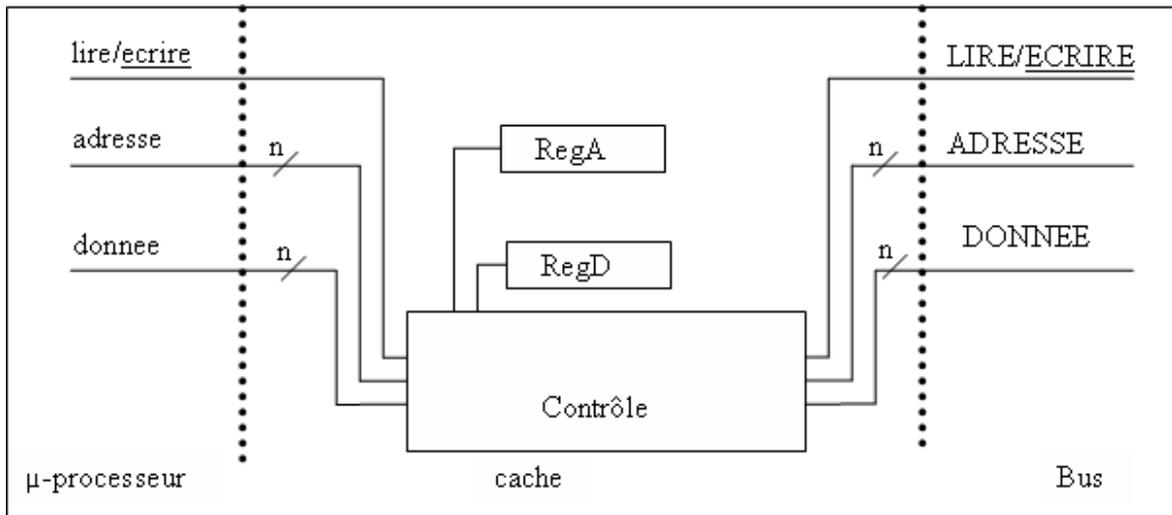
En procédant par étapes, donnez un circuit réalisant l'automate suivant (expliquez votre démarche, vos choix, vos constructions) :



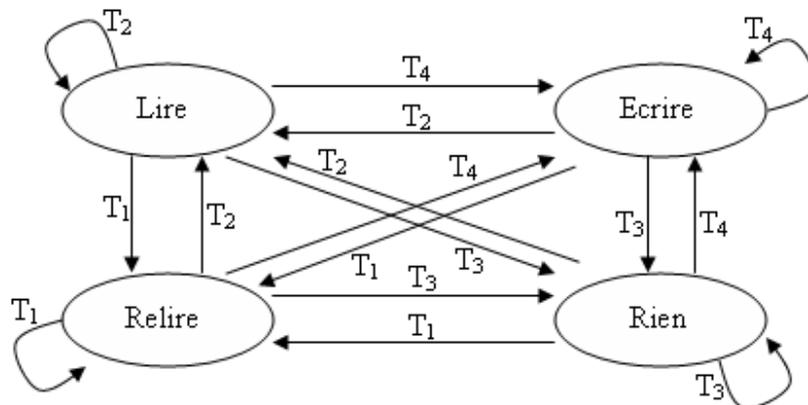
(barème indicatif, présentation de l'ensemble de la copie 2 points)

25 Circuit de cache mémoire

Réaliser le circuit de cache mémoire placé entre le microprocesseur et le bus comportant deux registres, permettant la sauvegarde d'un couple <adresse,donnée>, et un circuit de contrôle :



Le contrôle est décrit par l'automate de Moore :



Les transitions T1 correspondent aux entrées : *lire* et ($adresse=Reg_A$).

Les transitions T2 correspondent aux entrées : *lire* et ($adresse \neq Reg_A$).

Les transitions T3 correspondent aux entrées : *ecrire* et ($adresse=Reg_A$) et ($donnee=Reg_D$).

Les transitions T4 correspondent aux entrées : *ecrire* et ($(donnee \neq Reg_D)$ ou ($adresse \neq Reg_A$)).

L'état Lire correspond aux commandes en sortie : $ADRESSE \leq adresse \parallel Reg_A \leq adresse$, puis $LIRE$ puis $donnee \leq DONNEE \parallel Reg_D \leq DONNEE$.

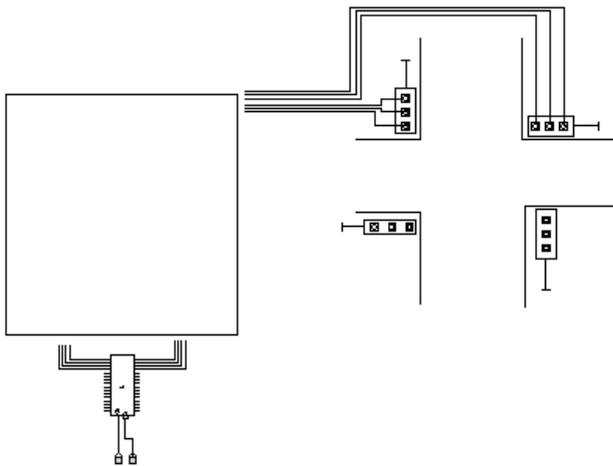
L'état Relire correspond aux commandes en sortie : $donnee \leq Reg_D$.

L'état Ecrire correspond aux commandes en sortie : $ADRESSE \leq adresse \parallel Reg_A \leq adresse \parallel DONNEE \leq donnee \parallel Reg_D \leq donnee$, puis $ECRIRE$

L'état Rien n'a pas de commandes en sortie

26 Feu tricolore

Réaliser un circuit de commande d'un feu tricolore qui reste 2 temps dans le vert, 1 temps dans l'orange et 3 temps dans le rouge. On pourra commencer par dessiner le chronogramme de ce circuit.



Compléter le circuit pour qu'il puisse commander les 2 séries de feu d'un carrefour et des feux piétons.

Ajouter un mode clignotant Orange.

Modifier le circuit pour que le feu puisse prendre en compte la présence d'une voiture.

Variante : le feu tricolore britannique, Vert, puis Orange, puis Rouge, puis Rouge et Orange, puis à nouveau Vert.

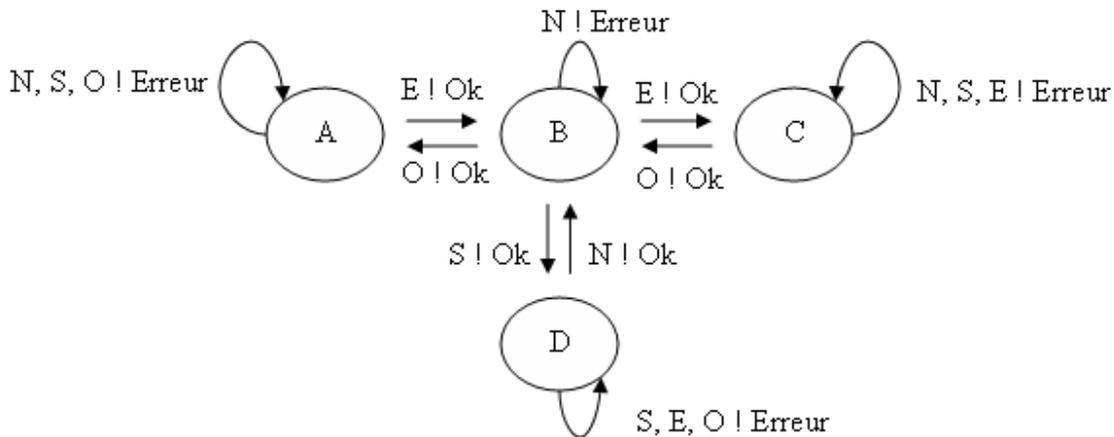
27 Machine à café

Imaginer une machine à café, et définir le circuit correspondant.

Variante : la machine à laver.

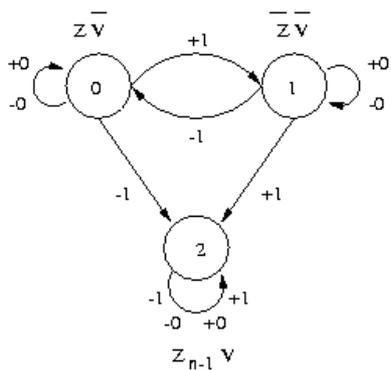
28 Robot

Définir un circuit réalisant l'automate de Mealy suivant simulant les déplacements (N : nord, S : sud ; E : est, O : ouest) d'un robot dans un entrepôt constitué de 4 zones (A, B, C, D). Une transition, notée X, Y ! Z, est une transition sur les entrées X, Y donnant la sortie Z.



29 Automate mystère

À quoi peut servir cet automate ?

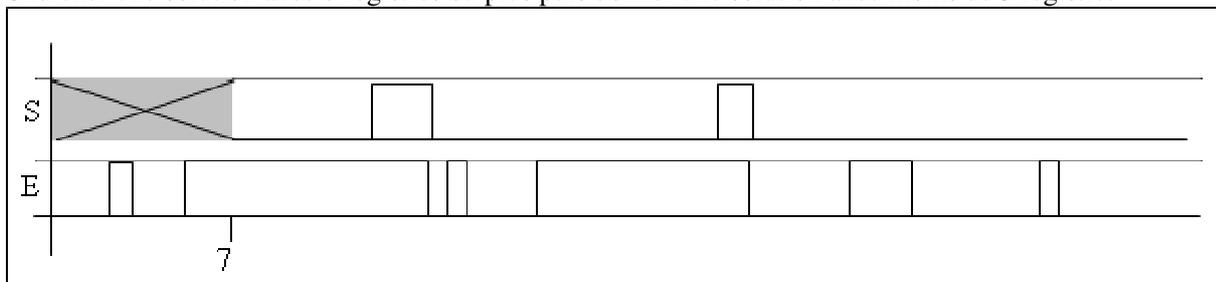


Définir un circuit qui implémente cet automate.

30 7 derniers vrais

Définir un circuit ayant deux entrées (dont une horloge) et une sortie valant vrai si et seulement si les 7 dernières valeurs de l'entrée au front montant de l'horloge ont été vraies.

Chercher une solution avec 7 registres ou plus puis donner une solution avec moins de 5 registre.

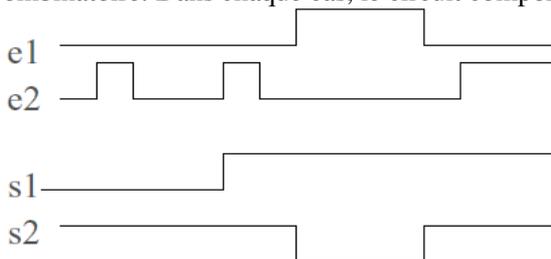


Variante : au lieu de 7 on peut prendre 3 ou 15 ?

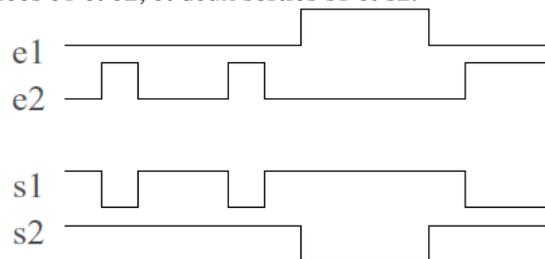
Variante : en sortie on peut avoir le nombre de vrai depuis le dernier faux.

31 Combinatoire ou séquentiel

Discuter pour les deux chronogrammes suivants si le circuit à l'origine de ces chronogrammes est séquentiel ou combinatoire. Dans chaque cas, le circuit comporte deux entrées e1 et e2, et deux sorties s1 et s2.



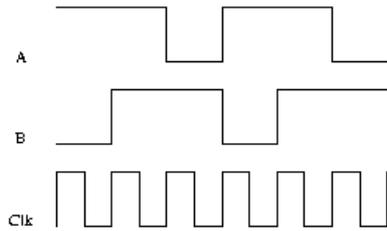
Circuit C1



Circuit C2

32 Chronogramme Mystère

Donner un circuit séquentiel ayant une entrée (clk), deux sorties (A et B) et qui ait le chronogramme suivant :



33 Contrôle de transmission sur ligne n bits

S'inspirer des codages avec bit de parité pour proposer un ensemble de deux circuits réalisant du contrôle de transmission de part et d'autre d'une ligne de transmission comportant n fils pour transmettre les informations et 1 bit pour effectuer le contrôle.

34 Synchronisation de deux automates

Observer les problèmes qui se posent lorsque l'on fait communiquer deux circuits automates comme cela peut se passer lors de la réalisation d'une PC/PO.

En particulier, montrer que si au moins un des deux automates est de type de Moore, il n'y a pas de problèmes, mais que si les deux automates sont de type Mealy, et que au moins l'une des sorties de l'un est une entrée de l'autre et réciproquement alors il y a une boucle dans le circuit (même si les deux automates ne sont pas synchronisés sur la même horloge).

Séance 6 Circuits à flots de données (ou PC/PO)

- PGCD
- Syracuse
- Fibonacci
- Bresenham
- Racine carrée
- Décalage de N bits
- Multiplication à la russe
- Comparaison (=, <, ...)
- Min, Max, Moyenne d'une séquence d'octet (terminée par 0)