

Exercice 1 Représentation des nombres (15 minutes)

A l'intérieur de cet exercice, les questions sont indépendantes.

1. Combien de bits faut-il pour représenter 65 563 entiers différents en binaire naturel ? En complément à 2 ?
2. Soit un ordinateur dont les mots mémoire sont composés de 32 bits. Cet ordinateur dispose de 4 Mio de mémoire. Un entier étant codé sur un mot, combien de mots cet ordinateur peut-il mémoriser simultanément ? Quelle est la plus grande valeur entière (décimale) que cet ordinateur peut mémoriser, cette valeur étant représentée par son codage binaire pur (on ne cherchera pas à calculer les grandes puissances de 2) ?
3. Combien d'entiers positifs peut-on coder en binaire sur un centième de kilo-octet (on ne cherchera pas à calculer les grandes puissances de 2) ?
4. En virgule fixe (non-signé), décoder le nombre binaire 11.011 en décimal puis coder en binaire le décimal 11.625.

Exercice 2 Algèbre de Boole (20 minutes)

A l'intérieur de cet exercice, les questions sont indépendantes.

1. Relier, en justifiant soigneusement, chaque phrases suivantes (1-2-3-4) à la proposition logique associée (a-b-c-d). 1 : toutes les variables A, B, C et D sont nulles. 2 : au moins l'une des variables A, B, C et D est égale à un. 3 : au moins l'une des variables A, B, C et D est égale à zéro. 4 : toutes les variables A, B, C et D sont égales à un. a : $A.B.C.D = 0$. b : $\overline{A} + \overline{B} + \overline{C} + \overline{D} = 0$. c : $\overline{A}.\overline{B}.\overline{C}.\overline{D} = 0$. d : $\overline{A}.\overline{D} + \overline{B}.\overline{C} = 1$.
2. En utilisant les règles de l'algèbre de Boole, simplifier les équations logiques $F_1(A, B, C) = \overline{A}.B.C + A.\overline{B}.\overline{C} + \overline{A}.\overline{B}.C + A.B.C$; $F_2(A, B, C, D) = (A.\overline{B}.(C + B.D) + \overline{A}.\overline{B}).C$ et $F_3(A, B, C) = A.B.C.(A.B + \overline{C}.(B.C + A.C))$.
3. Représenter sous forme de mintermes l'équation logique $F_4(A, B, C, D) = \overline{A}.\overline{B} + A.B.\overline{C}.D$
4. Simplifier à l'aide de tableau de Karnaugh, l'équation logique suivante : $F_5(A, B, C, D) = \overline{A}.\overline{B}.\overline{C}.\overline{D} + \overline{A}.\overline{B}.C.\overline{D} + A.\overline{B}.\overline{D} + A.B.\overline{C}.\overline{D} + A.B.C.\overline{D}$.
5. Dans une tableau de Karnaugh à 5 variables (A, B, C, D, E) donner les cases adjacentes à la case $(0, 0, 0, 0, 0)$.

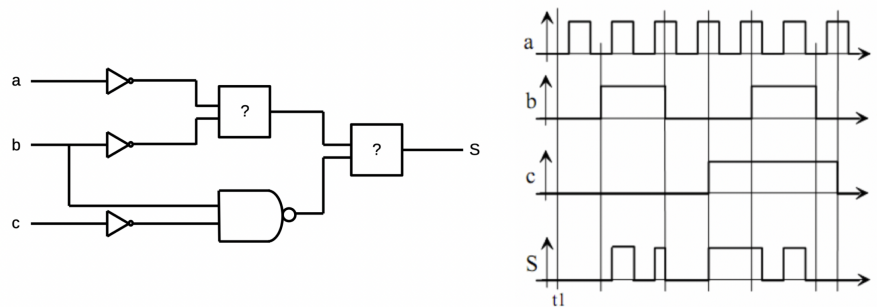
Exercice 3 Soustracteur binaire (30 minutes)

Pour fabriquer un soustracteur entre deux nombres binaires de même taille, il existe plusieurs méthodes. On peut considérer, par exemple, un additionneur classique mais avec un des deux nombres codé en complément à 2. Ici on va suivre la méthode (plus directe) qui a été proposée pour l'additionneur classique, à savoir, l'étude d'un demi-soustracteur (soustraction sur un bit) puis, par un raisonnement itératif, l'extension au cas de deux nombres de N bits chacun.

1. Coder en complément à 2 (sur 2 bits) les chiffres décimaux signés suivants : $(-1)_{10s}$, $(0)_{10s}$ et $(1)_{10s}$. En déduire la table de vérité du soustracteur $\alpha_0 - \beta_0$ où α_0 et β_0 sont deux « mots » de un bit chacun. Simplifier les équations des sorties et donner le circuit logique du demi-soustracteur.
2. Calculer avec cette méthode (en passant par le binaire naturel) $(17)_{10} - (3)_{10}$ puis $(17)_{10} - (20)_{10}$.
3. Par un raisonnement itératif similaire à la conception d'un circuit additionneur, proposer un schéma détaillé pour le soustracteur complet sur N bits $A - B$ où on posera $A = (\alpha_{N-1}\alpha_{N-2} \cdots \alpha_1\alpha_0)_2$ et $B = (\beta_{N-1}\beta_{N-2} \cdots \beta_1\beta_0)_2$.

Exercice 4 Rétro-ingénierie¹ d'un système inconnu simple (15 minutes)

Votre employeur vous a demandé d'analyser le circuit intégré d'un produit concurrent afin de savoir comment il fonctionne. Concernant une des fonction logique du circuit, et après étude de la carte, vous avez réussi à tracer un schéma logique partiel (vous n'arrivez pas à identifier deux portes où les références des circuits on été effacé). En testant le circuit avec des entrées types a , b et c , vous avez obtenu le chronogramme suivant pour la sortie S . A l'aide de ces informations et en justifiant soigneusement, identifier les deux portes logiques mystères utilisées dans ce circuit.



Exercice 5 Système ternaire non-équilibré (15 minutes)

On considère le système ternaire² comme étant le système de base 3 avec pour chiffres $\{0, 1, 2\}$.

1. Donner en base 10 le nombre $(102)_3$.
2. Soit un mot de taille T en base ternaire. Quel est la valeur décimale du nombre maximal pouvant être codé ?
3. Soit le nombre $(120)_{10}$, combien faudra t'il de chiffre en base ternaire pour le représenter.
4. A l'aide de vos connaissances, proposer une méthode pour passer d'un nombre décimal (non-signé) à un nombre en base ternaire. Appliquer cette méthode pour écrire $(120)_{10}$ en base ternaire.
5. Effectuer l'addition suivante avec un résultat en base ternaire : $(201)_3 + (12)_3$.

Exercice 6 Incrémenteur-Décrémenteur (25 minutes)

Une grande partie des opérations effectuées par un ordinateur consistent simplement à ajouter ou retirer 1 à une autre valeur binaire x constitué de N bits (on incrémente ou décrémente la valeur x).

1. On souhaite réaliser un incrémenteur $N = 3$ bits avec retenue de sortie. Les entrées seront notées A_0 (BPF) A_1 et A_2 (BPf). Les sorties seront notées I_0 (BPF) I_1 et I_2 (BPf) et R pour la retenue. Donner

1. Il s'agit ici d'un système simple, mais il faut retenir que l'utilisation de signaux tests pour reconstruire la table de vérité d'un circuit intégré est une technique très commune. Plus d'informations sur la rétro-ingénierie des circuits ici <https://connect.ed-diamond.com/misc/mischs-024/introduction-au-reverse-hardware>

2. Il s'agit ici le la base 3 naturelle ou système ternaire non-équilibré. Dans la pratique, on utilise plutôt le système ternaire équilibré où on utilise des chiffres avec les valeurs -1 , 0 , et 1 . Cette combinaison est particulièrement intéressante pour les relations ordinales entre deux valeurs, où les trois relations possibles sont inférieur à, égal à et supérieur à. Le système ternaire équilibré est facilement représenté par les signaux électroniques, comme potentiel pouvant être soit négatif, neutre ou positif. Par conséquent, un fil électrique peut transporter plus d'informations en ternaire (trois états) qu'en binaire (deux états). Ainsi, le système ternaire en électronique permet de réduire le nombre de composants, et donc la consommation électrique. Le ternaire non-équilibré peut être converti en notation ternaire équilibrée en ajoutant $1111 \dots$ avec retenue, puis en soustrayant $1111 \dots$ sans retenue. En 1958 en Union soviétique, l'équipe de Nikolay Brusentsov et Sergei Sobolev (célèbre pour les espaces de Sobolev, des espaces de fonctions caractérisées par des conditions sur leurs transformées de Fourier et pour avoir introduit les premières fonctions généralisées qui deviendront, plus tard, les distributions) à l'université d'État de Moscou a développé un ordinateur ternaire, le Setun, reposant sur l'utilisation de tôles feuilletées miniatures et de diodes utilisables pour créer un système basé sur une logique à trois états. À l'usage, ces éléments se sont révélés plus rapides, plus fiables, plus durables et moins gourmands en énergie que leurs concurrents binaires (du moins avant que l'URSS n'ait accès aux transistors). Plus d'information ici https://fr.wikipedia.org/wiki/Syst%C3%A8me_ternaire

la table de vérité de ce circuit ainsi que son schéma de câblage utilisant le moins de portes logiques possibles.

2. Donner la table de vérité d'un décrémenteur $N = 3$ bits avec 3 entrées (A_0 (BPF) A_1 et A_2 (BPf)) et 3 sorties (D_0 (BPF) D_1 et D_2 (BPf)). Pour l'état $A_0 = A_1 = A_2 = 0$ on choisira la sortie $D_0 = D_1 = D_2 = 0$.

3. Réaliser la fonction D_0 avec un multiplexeur 8 vers 1.

4. Réaliser la fonction D_1 avec un démultiplexeur 1 vers 8 et une porte OU à trois entrées. Indice : on placera 1 en entrée du démultiplexeur et A_0 (BPF) A_1 et A_2 (BPf) aux commandes du démultiplexeur.

5. Réaliser la fonction D_2 avec, entre autre, un multiplexeur 4 vers 1 et un démultiplexeur 1 vers 4 en complétant le schéma ci-dessous

