Université Joseph Fourier – Grenoble 1

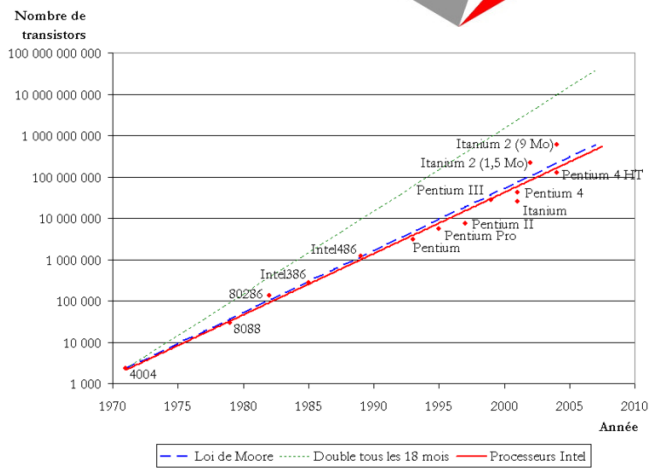
Architecture des ordinateurs

Exercices et problème

Original Juin 2011, revue 2012 pour les illustrations

Exercice. Loi de Moore.

Expliquez, illustrez et commentez cet extrait d'article de « Pour la science » (2001). Illustration wikipedia.



En avril 1965, dans un article publié dans la revue Electronics Magazine, Gordon Moore (qui dirige alors la recherche chez Fairchild Semiconductors et qui cofondera avec Robert Noyce la firme Intel en 1968) remarque que, depuis l'invention du circuit intégré (en 1958), on observe un doublement annuel du nombre de transistors par unité de surface. Il prédit que ce doublement durera encore dix ans.

Sa prévision se réalisera assez bien ; pourtant, en 1975, dans un exposé où on lui demande de proposer un nouveau pronostic, G. Moore, analysant les difficultés technologiques présentes, annonce cette fois un doublement du nombre de transistors par unité de surface tous les deux ans (au lieu de tous les ans).

En prenant en compte l'augmentation de la fréquence d'horloge des microprocesseurs (la vitesse avec laquelle on les fait calculer), ou en faisant la moyenne entre les prédictions de 1965 et celles de 1975, certains des collègues électroniciens de Moore ont reformulé les prédictions et ont parié sur un doublement de performance tous les 18 mois. C'est ce qu'abusivement aujourd'hui nous dénommons la Loi de Moore et que nous dénommerons ici la loi de Moore standard : «les performances informatiques doublent tous les 18 mois».

[…]Parmi les variantes de la loi de Moore, la loi de Rock inquiète de nombreux observateurs. Elle énonce que le coût en investissements double tous les quatre ans. L'industrie du microprocesseur est une industrie de l'ordre de 200 milliards de dollars par an, qui investit 10 pour cent de ses revenus dans la recherche et le développement. La loi de Rock, freinerait peut-être plus les progrès envisageables (les doublements de performances prévus tous les 18 mois) que les difficultés liées à la physique, car rien en économie ne croît aussi vite et, tôt ou tard, il y aura un manque de ressources financières, en tout cas avant que la totalité des richesses économiques terrestres ne soient consacrées à la fabrication de microprocesseurs! La limitation économique imposera sa contrainte à la courbe exprimant la densité de transistors en fonction du temps. Notons, en passant, que la loi de Machrone, variante humoristique de la loi de Moore, est étonnamment exacte : l'ordinateur dont vous rêvez vaut toujours 5 000 dollars (ou euros). […]

Dans le domaine du stockage des données sur disques magnétiques, la loi de Moore s'est appliquée sans fantaisie jusqu'en 1998, mais, depuis, suite à des avancées techniques accélérées (ici, il ne s'agit plus de densité de transistors, mais de maîtrise des constituants magnétiques), l'augmentation de la capacité de stockage est multipliée par 3,5 tous les 18 mois, beaucoup plus que ce que prévoit la loi de Moore standard. Les disques durs d'un teraoctet (10 12 octets, c'est-à-dire mille milliards de caractères) dont nous allons pouvoir disposer d'ici peu permettront de stocker l'équivalent d'un million de livres de 200 pages, ou 400 cassettes vidéo vhs, ou encore toutes les conversations d'une personne durant toute sa vie !

Exercice. Algèbre de Boole, machine de Jevons.

Illustrer, commenter ce texte de Robert Ligonnière tiré de « Préhistoire et histoire des ordinateurs » sur la machine de Jevons, le piano logique dont la première démonstration eut lieu en 1870.

« C'était un piano vertical de 90 centimètres de haut. Sur la face, des ouvertures permettaient de lire des lettres, symbolisation des seize combinaisons possibles à partir de quatre termes et de leurs négatifs. Au bas du piano, les quatre termes étaient représentés deux fois :

- sous leur forme positive et
- sous leur forme négative.

Il existait aussi cinq touches de fonction :

- deux Ou inclusifs ;
- une touche égale ;
- une touche point pour marquer que l'introduction d'une proposition était terminée ;
- une touche fin pour remettre la machine à zéro.

Faire fonctionner le piano était très simple ; on frappait les touches exactement dans l'ordre des termes de l'équation logique. En fin de travail, la machine affichait la proposition vraie. »

Exercice. Circuits réversibles de R. Feynman.

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.»

Q1. Le texte de R. Feynman comporte quelques implicites. Quelles sont les quinze + 1 combinaisons dont il est question ?

Q2. Montrez que la porte XOR est irréversible.

Q3. 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

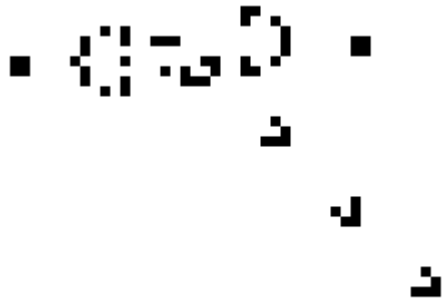
Q4. Réalisez cette porte à l'aide d'un circuit combinatoire simple (comportant seulement NOT, AND et OR). Donnez sa complexité.

Q5. Quelle est la table de vérité du circuit comportant deux portes CN, les sorties de la première servant d'entrée pour la seconde.

Q6. 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.

Q7. A partir du circuit obtenu à la question précédente donnez le circuit d'un multiplexeur 8→1 (Mux 8→1) réversibles.

Exercice. Jeu de la vie de Conway pour une seule dimension



L'authentique jeu de la vie est une simulation imaginée en 1970 par John Horton Conway qui se déroule sur une grille à deux dimensions où les cases représentent des cellules. L'évolution d'une cellule est déterminée par l'état de ses huit voisines : Une cellule morte possédant exactement trois voisines vivantes devient vivante. Une cellule vivante possédant deux ou trois voisines vivantes le reste, sinon elle meurt.

Pour simplifier l'exercice, nous travaillerons avec des cellules sur une seule ligne (jeu de la vie à une seule dimension). L'évolution d'une cellule sera déterminée par son état antérieur et l'état de ses deux voisines, celle immédiatement à gauche et celle immédiatement à droite :

Une cellule morte possédant deux voisines vivantes devient vivante.

Une cellule qui vient de mourir et qui possède au moins une voisine vivante redevient vivante.

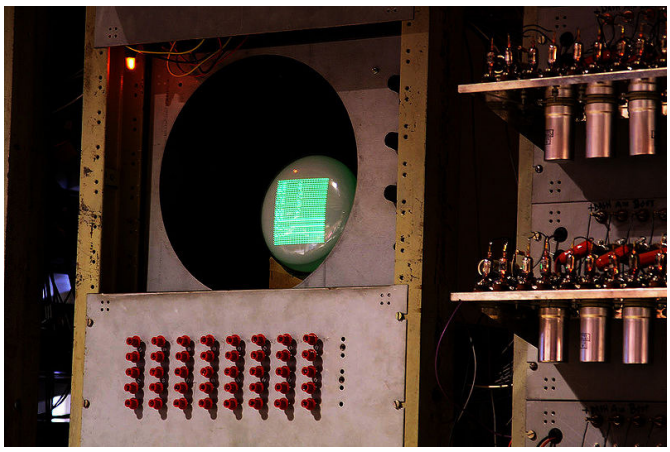
Une cellule vivante qui possède une et une seule voisine vivante ou qui était déjà vivante à l'étape précédente reste vivante sinon elle meurt.

Q1. Donner la table de vérité du jeu de la vie à une dimension, (en entrée : l'état courant d'une cellule et de ces 2 voisines, l'état précédent de la cellule, en sortie : le prochain état de la cellule). Dessiner un circuit réalisant cette fonction.

Q2. Associer 8 circuits définis à la question précédente pour constituer une ligne de 8 cellules. En bout de ligne, on considérera que les cellules sont voisines (la ligne est, en fait, circulaire comme en géométrie Riemannienne ou géométrie elliptique). Donner la complexité de ce circuit.

Q3. On souhaite évaluer le comportement de 8 cellules après plusieurs évolutions. Proposer deux circuits pour le faire. II-3a. Le premier circuit (combinatoire) étudiera l'évolution de 8 cellules après 8 évolutions et sera constituer de 8 circuits définis à la question précédente. II-3b. Le second (séquentiel) n'utilisera qu'un seul circuit tel que défini à la question précédente, mais utilisera en plus une horloge, et des éléments mémoire.

Problème : la Small-Scale Experimental Machine



« La Small-Scale Experimental Machine (SSEM) (« machine expérimentale à petite échelle »), surnommée Baby (« Bébé »), était la première machine à architecture de von Neumann du monde. Construite à l'université Victoria de Manchester par Frederic Calland Williams, Tom Kilburn et Geoff Tootill, elle exécuta son premier programme le 21 juin 1948. Cette machine ne fut pas construite pour son utilité pratique en tant qu'ordinateur, mais comme un banc de test pour le tube de Williams, une forme primitive de mémoire d'ordinateur. Bien que considérée comme « petite et primitive » selon les standards de son époque, elle fut la première machine fonctionnelle contenant tous les éléments essentiels d'un ordinateur électronique moderne. Dès que la SSEM eut démontré la faisabilité de sa conception, un projet fut lancé à l'université de Manchester pour la développer afin d'en faire un ordinateur plus utilisable, le Manchester Mark I. Le Mark I, à son tour, devint rapidement le prototype du Ferranti Mark I, le premier ordinateur généraliste commercialisé. La SSEM utilisait des mots de 32 bits et une mémoire de 32 mots. Comme elle était conçue pour être l'ordinateur à programme enregistré en mémoire le plus simple possible, les seule opérations arithmétiques implantées dans le matériel étaient la soustraction binaire et le moins unaire; les autres opérations arithmétiques étaient réalisées par des programmes. Le premier des trois programmes écrits pour la machine trouva le plus grand diviseur propre de 218 (262 144), un calcul dont on savait qu'il prendrait un temps important pour s'exécuter, en testant chaque entier de 218-1 à 1 dans l'ordre décroissant, car il fallait réaliser les divisions par des soustractions itérées du diviseur. Le programme consistait en 17 instructions et fonctionna pendant 52 minutes avant d'atteindre la réponse correcte, 131 072, après que la SSEM ait effectué 3,5 millions d'opérations, ce qui donne une vitesse de 1,1 kIPS. »

Extrait de l'article « Small Scale Experimental Machine » sur wikipedia.

Soustractions

En utilisant la même méthode qui a permis de construire une ALU avec additionneur, cet exercice cherche à produire une ALU pour la SSEM basée sur un soustracteur.

Q1.1 : Dans un premier temps, donner la table de vérité d'une cellule de base d'un soustracteur binaire (comme pour la cellule d'additionneur binaire, cette cellule comporte trois entrées et deux sorties : en entrée, les deux opérandes et un éventuel emprunt entrant venant de la cellule précédente ; en sortie, le résultat et un éventuel emprunt sortant pour la cellule suivante)
Q1.2 : Donner le dessin logique de cette cellule.
Q1.3 : Donner le dessin du circuit de soustraction binaire 32 bits obtenu à partir des cellules précédentes.
Q1.4 : Proposer le circuit d'une UAL qui puisse faire soustraction binaire, moins unaire, ou rien comme l'UAL de la machine SSEM.

Circuits séquentiels

Pour trouver le plus grand diviseur propre (PGDP) d'un entier N, la SSEM a utilisé un algorithme du type :
En entrée, N : entier ; en sortie PGDP : entier ; Candidat, Reste sont des variables locales, de type entier.

```
Debut :
  Candidat ←N-1
  Tant que Candidat > 0 :
  | Reste ← N
  | Tant que Reste > 0 :
  | | Reste ←Reste - Candidat
  | FinTantQue
  | Si Reste = 0 :
  | | PGDP ← Candidat
  | | Candidat ← 0
  | Sinon
  | | Candidat ← Candidat -1
  | FinSi
  FinTantQue
Fin.
```

Q2.1 : Proposer un circuit séquentiel ou une association PC/PO réalisant cet algorithme.

Automate de contrôle

Les instructions disponibles sur la SSEM étaient peu nombreuses :

Code	Mnémonic	Opération
000	Jump S	Saut à l'instruction à l'adresse obtenue à l'adresse S
100	JumpRel S	Saut à l'instruction à l'adresse correspondant au compteur programme plus la valeur relative obtenue à l'adresse S
010	LoadNeg S	Charge Acc avec l'opposée de la valeur à l'adresse S
110	Store S	Charge la valeur à l'adresse S avec Acc
001 ou 101	Sub S	Soustrait la valeur à l'adresse S à Acc, enregistre le résultat dans Acc
011	Comp	Saute l'instruction suivante si Acc est négatif
111	Stop	Stop

Q3.1 : En faisant les hypothèses nécessaires sur la partie opérative de la machine SSEM, proposer un graphe de contrôle pour interpréter le langage machine proposé.

Discussion

Pour trouver le plus grand diviseur propre de 218, la SSEM n'utilisa pas un circuit séquentiel comme vous en avez proposé un dans la partie II, mais un programme basé sur les instructions données en partie III et un circuit PC/PO (la SSEM elle-même) capable d'exécuter et d'interpréter ce programme.

Q4.1 : Comparer ces deux approches.