

BULLETIN D'INFORMATIQUE APPROFONDIE ET APPLICATIONS

SCIENCE DE L'INFORMATION

COMITE SCIENTIFIQUE

N° 53 - JUIN 1999

Patrick Abellard
Françoise Adreit
Jalal Almhana
France Chappaz
M'hamed Charifi
Roger Cusin
Bernard Goossens
Patrick Isoardi
Robert Jacquier
Jean - Philippe Lehmann
Nadia Mesli
Patrick Sanchez
Rolland Stutzmann
André Tricot

DIRECTEUR

Jean - Michel Knippel

REDACTEUR EN CHEF

Edmond Bianco

REDACTEUR ADJOINT

Sami Hilala

REDACTION

Université de Provence
Equipe Hermès. Case 33
3, place Victor Hugo
F - 13331 Marseille Cedex 3
Téléphone: (0)4 91 10 62 30
Télécopie : (0)4 91 50 91 10

DEPOSITAIRE

Université de Provence
Bibliothèque Universitaire
3, place Victor Hugo
F - 13331 Marseille Cedex 3
Téléphone: (0)4 91 62 44 16
Télécopie : (0)4 91 95 75 57

1 EDITORIAL.
Informatique, intelligence,

par Edmond Bianco

5 CHAPITRE 7.
LE PROCESSEUR,

par Edmond Bianco

27 VOZZAVEDIBISAR.
Le symbole,

par Edmond Bianco

D'ici quelque temps le bulletin aura ses informations sur WWW:
<http://scamup.univ-mrs.fr> <http://www.up.univ-mrs.fr>

Publication trimestrielle, gratuite, de l'Université de Provence

Edition 2001

ISSN 0291 - 5413

BULLETIN D'INFORMATIQUE APPROFONDIE ET APPLICATIONS

SCIENCE DE L'INFORMATION

COMITE SCIENTIFIQUE

N° 53 - JUIN 1999

Patrick Abellard
Françoise Adreit
Jalal Almhana
France Chappaz
M'hamed Charifi
Roger Cusin
Bernard Goossens
Patrick Isoardi
Robert Jacquier
Jean - Philippe Lehmann
Nadia Mesli
Patrick Sanchez
Rolland Stutzmann
André Tricot

DIRECTEUR

Jean - Michel Knippel

REDACTEUR EN CHEF

Edmond Bianco

REDACTEUR ADJOINT

Sami Hilala

REDACTION

Université de Provence
Equipe Hermès. Case 33
3, place Victor Hugo
F - 13331 Marseille Cedex 3
Téléphone: (0)4 91 10 62 30
Télécopie : (0)4 91 50 91 10

DEPOSITAIRE

Université de Provence
Bibliothèque Universitaire
3, place Victor Hugo
F - 13331 Marseille Cedex 3
Téléphone: (0)4 91 62 44 16
Télécopie : (0)4 91 95 75 57

1 EDITORIAL.
Informatique, intelligence,

par Edmond Bianco

5 CHAPITRE 7.
LE PROCESSEUR,

par Edmond Bianco

27 VOZZAVEDIBISAR.
Le symbole,

par Edmond Bianco

D'ici quelque temps le bulletin aura ses informations sur WWW:
<http://scamup.univ-mrs.fr> <http://www.up.univ-mrs.fr>

Publication trimestrielle, gratuite, de l'Université de Provence

Edition 2001

ISSN 0291 - 5413

ÉDITORIAL,

Informatique, intelligence.

Après avoir fait l'effort, non négligeable, de se débarrasser des passions et des préjugés du moment, il devient très curieux d'observer vivre nos contemporains. S'il est un sujet, ô combien rebattu, combien important aux regards de personnages eux aussi importants, voire de tout le monde, c'est celui de l'intelligence, de ses manifestations, et surtout de l'art que cette qualité a de briller. Car il s'agit d'une qualité, et jamais d'un défaut.

Partons donc sur la piste, lumineuse comme une traînée de poudre, de l'intelligence, et essayons de la suivre jusqu'à ce qu'elle atteigne ces points d'apothéose dont l'irrésistible explosion éclaire la nuit de notre ignorance comme une magnifique pièce d'artifice multicolore.

Piquons un peu au hasard. Les applications de l'intelligence à des techniques de pointe telles que l'informatique et son support matériel l'électronique, ont permis le remplacement de nombre de personnes vouées à des tâches ingrates, mécaniques, par des automates plus fiables, plus rapides, plus économiques. Voilà donc un effet lumineusement plaisant de l'intelligence. Oui, mais voilà, conséquence de cet apport positif à la civilisation, les gens qu'on avait si aimablement remplacés à des travaux dégradants, répugnants, voire dangereux, se sont retrouvés sans emploi. Et sans emploi, aucune intelligence n'avait prévu qu'on devient en quelque sorte hors la loi, on disparaît de la société, on n'existe plus. Privé de ressources, c'est-à-dire du seul passeport efficace, l'argent, on n'a plus sa place dans le monde. Quelques indemnités ont bien été prévues, ça et là, mais tellement rognées, restreintes, vilipendées, parce qu'abominablement coûteuses et improductives, qu'on hésite à les croire résultat de l'application de la moindre des intelligences.

Supposons tout de même le concept d'intelligence intact malgré ces apparentes insuffisances, et essayons de réfléchir en se plaçant d'un point de vue relatif. L'intelligence a joué parfaitement son rôle, mais il y a ces rejetés de la civilisation.

Quand on fait de la menuiserie, il reste toujours par terre un bon tas de débris divers constitués de sciure, de copeaux et de petites tombées dont la destination sera le poêle ou la décharge. En transposant, doit-on se demander si la construction d'une société bien organisée, intelligemment organisée, implique obligatoirement qu'une partie de la population soit rejetée comme sont rejetés copeaux et sciures?

Cela semble rappeler fâcheusement certain empire qui devait durer mille ans, mais continuons à raisonner...

Dans un pays où les chômeurs se comptent par millions, sans qu'on ne comptabilise pour autant ceux qui n'ont encore jamais pu avoir de travail -pudeur de l'intelligence, sans doute- un jour on inaugura le "Charles de Gaulle", ce "machin" comme aurait dit "Le grand général", lui même. Et que découvrit-on aux yeux des foules éberluées? Un immense voile "bleu blanc rouge" recouvrant l'objet comme un magnifique paquet cadeau que le contribuable s'offrait à lui-même. Aussitôt des esprits chagrins de se demander combien de familles de smicards ou autres "érémistes" on aurait pu nourrir avec l'argent de cette dépense frivole, je ne parle que du voile aux couleurs patriotiques, bien sur, quand au "machin" lui-même, pourra-t-il un jour sortir de sa cale?

Il est très amusant également d'entendre Minc argumenter chez Pivot, pour instaurer de la concurrence entre les universités, il ne comprend d'ailleurs pas comment les gens peuvent se refuser à ce petit jeu. Mais, dit-il en substance, les diplômés français vont désormais se trouver en concurrence directe avec les diplômés de toute l'Europe, pourquoi pas donc, en concurrence avec les autres diplômés français. Prolongeons le raisonnement, on peut se demander alors pourquoi les professeurs s'échinent malgré tout à enseigner sciences et connaissances; Apprendre à se servir d'une mitrailleuse, de grenades ou d'un fusil d'assaut serait bien plus utile, dans la compétition effrénée que prône notre ami. D'ailleurs son copain Jean Marie Messier renchérit en déclarant que bien plus que de diplômés il a besoin de gens mus par l'enthousiasme et la volonté de foncer, en quelque sorte, pour lui, Bob Denard serait un meilleur employé de Vivendi que Poincaré ou Einstein. Pourtant, quand on entend ces deux personnages s'exprimer au petit écran, on les croirait vraiment doués d'intelligence.

Tout pousse à la concurrence dans la "nouvelle économie", la concurrence c'est à dire la lutte à mort. Celui qui ne se révèle pas le plus fort au sens de certains critères définis par les Desmaret, les Messier et autres Jaffré, est bon pour le chômage, c'est à dire pour la casse. Étranges exemples d'application de l'intelligence. Surtout quand on en constate les conséquences, personne n'a l'air de se sentir gêné d'affréter de vieux cargos fissurés de partout pour transporter des déchets particulièrement dangereux, et encore moins gênés quand ces produits se répandent ... Bien au contraire, c'est avec beaucoup d'intelligence que les sociétés pétrolières concernées en profitent pour se faire un peu de publicité à la télévision en montrant comment elles viennent sur les plages, parachever l'oeuvre des bénévoles, quand il n'y a plus évidemment qu'à donner le dernier coup de chiffon pour faire briller les cailloux.

Faisons maintenant un peu d'exotisme, il existe sur la planète, un petit coin particulièrement aride mais tout spécialement béni des Dieux, où l'un d'entre eux, justement a parachuté son fils, dit-on, afin qu'il soit torturé à mort pour sauver l'âme de l'espèce humaine, parmi ces divinités, il en est une autre, plus jeune et plus à la mode qui y serait apparue et dont on prétend qu'elle protège aussi ce même territoire. Et une bande de pauvres gens, crève-la-faim pour certains, moins pour d'autres, se battent comme des chiens pour s'arracher ce lopin hautement symbolique, chacun pour honorer son dieu dignement. Et honorer son Dieu dignement consiste à hacher menu les adorateurs du concurrent. Il me semble qu'une intelligence même très moyenne pourrait en conclure soit que les Dieux en question n'existent pas sinon pourraient-ils laisser s'accomplir de telles imbécillités, ou alors les populations en question sont particulièrement endoctrinées, catéchisées, au point d'en devenir complètement stupides. Rien n'empêche logiquement que les deux propositions soient vraies en même temps.

Nous n'irons pas jusqu'à imaginer que les histoires des Dieux, poudre aux yeux pour les naïfs, ne servent qu'à recouvrir des visées d'une autre nature. Mais alors les chemins de "l'intelligence" se révèlent particulièrement tortueux et difficiles à suivre.

Il est d'autres exemples apparemment très différents, mais tout aussi étranges qui "interpellent", pour parler à la mode, quand au véritable sens de cette fugace notion d'intelligence. Je ne parle même pas de ce vieux dictateur, prêt à toutes les singeries noyées dans la vodka, pour masquer ses petites combines, faire réhabiliter un ancien Tsar complètement oublié, ou, plus grave, tenter de faire disparaître les Tchétchènes de la surface de la terre. Non, je parle de quelque chose qui nous touche de plus près, à l'époque où périodiquement se réunit un "sommet" mondial pour la protection de la planète contre la pollution.

Prenons le cas de la distribution de l'énergie. On s'est aperçu que la consommation débridée d'énergie avait une conséquence sur l'équilibre de la planète. On assiste alors à une étrange réaction, alors s'engage un étrange ballet planétaire. Nos constructeurs s'agitent pour vendre des millions de voitures à la Chine, marché vierge et splendide, nos compagnies pétrolières se disputent pour "aider" les pays africains à explorer leurs ressources minières ... et pétrolières. Plus près du consommateur d'énergie, une intelligence tout à fait moyenne pourrait s'étonner de constater comment ce produit est distribué. En effet, quels que soient les tarifs, on commence par payer une somme fixe, non négligeable, dite location du compteur, à laquelle s'ajoute la valeur consommée. Et pour les grandes consommations les tarifs sont dégressifs. En quelque sorte, pour réduire le prix de l'unité il faut consommer beaucoup, étrange manière pour pousser à l'économie ! On peut distinguer alors deux sortes de consommateurs, ceux dont la consommation est intégrée à une activité lucrative, et ce genre de

tarification leur est bénéfique si leur entreprise est en développement, et ceux dont la consommation est à fond perdu, et ceux là paieront toujours très cher l'énergie, de même pour les entreprises qui n'ont pas vocation à consommer toujours plus, seule manière de faire baisser le coût moyen.

Quand à la tarification des télécommunications, elle devient grotesque, on diminue de quelques centimes le prix de la minute de communication internationale à grands renforts de tambour télévisuel, et on double carrément la taxe fixe, ce qui vous met la communication locale au prix de l'or, surtout si vous téléphonez peu.

On pourrait imaginer qu'une politique "de gauche" de la distribution de l'énergie, permettrait à la fois de réhabiliter la notion de service public et d'organiser la réduction du gaspillage. Peut être suffirait-il de retourner la notion de tarification en calculant une tranche gratuite d'énergie pour chaque citoyen, au delà de laquelle le tarif serait progressif par tranches successives, afin de mettre rapidement le gaspillage hors de prix, tout en accordant à chacun un minimum vital digne d'un vrai service public.

On peut toujours rêver.

On peut aussi comprendre que les petites compagnies qui produisaient l'électricité, le gaz, les essences et autres énergies essayaient d'accroître leur puissance à une époque de faible consommation globale, mais qu'aucune "intelligence" issue des "grandes écoles" ou de l'ENA n'ait été capable de se projeter dans un avenir de débordement énergétique polluant, peut paraître surprenant si l'on se fait encore quelques illusions sur les capacités "cognitives" et imaginatives des personnages importants.

Mais on rencontre des choses plus surprenantes encore, inimaginables. Récemment a été supprimé un impôt inique, et qui, de plus avait été détourné de son but apparemment social, qui était de créer un fonds de solidarité à la vieillesse. La vignette, la célèbre vignette a disparu, que le diable la patafiole, eh bien il s'est trouvé des gens pour s'élever contre cette mesure de salubrité publique...

Il est ainsi des jours de déprime où on ressent profondément un certain désespoir à la simple idée de réfléchir à cette sacrée notion d'intelligence...

Edmond Bianco

P.S.. Un dernier exemple de la brillance explosive de l'intelligence: Il concerne les farines animales dont on engraisse les vaches. Avec quelques dizaines d'années de retard on a fini par s'apercevoir du danger de la chose, sans pour autant prendre autre chose que des semblants de mesures. Mais le danger se faisant pressant on calcule que la suppression de ces déchets de putréfaction vont coûter cinq milliards ... A qui?

CHAPITRE 7

LE PROCESSEUR

Edmond Bianco

Nous disposons d'une machine universelle dont nous avons étudié le rôle. Un pas reste à franchir. Si nous voulons disposer d'un automate capable de réaliser n'importe quel calcul pourvu qu'il soit présenté sous forme de programme, codé donc, à partir d'une phrase quelconque du langage de la machine de Nolin, il va nous falloir passer au câblage. Que signifie réellement procéder à du câblage?

Les techniques de construction des automates ont évolué au cours des âges. Partant du simple piège monté avec des lames, des ressorts et des clefs de déclenchement, et par des combinaisons de plus en plus savantes de métal, de bois, de gaz, on est arrivé à des structures complexes et puissantes. Exemples, la machine à vapeur, à partir du moment où on a imaginé le tiroir de distribution, le moteur à combustion interne, les diverses sortes de turbines, etc.

Mais le niveau de complexité des opérations de choix et de déclenchement demeurait faible. Le va et vient du piston entraînait le va et vient du tiroir qui découvrait successivement les lumières d'admission et d'échappement. Procédé tout simple, repris ultérieurement et récemment à l'usage des suspensions de voitures de type hydropneumatique. Même chose dans le cas du moteur à combustion interne, où le vilebrequin entraîne l'arbre à cames qui commande les soupapes. Dans les moteurs non Diesel, il entraîne en plus la commande de l'allumeur qui distribue au bon moment ses étincelles électriques.

Mais ce sont d'autres mécanismes qui inspirèrent des gens comme Pascal au XVIII^{ème} siècle ou Charles Babbage, au XIX^{ème} siècle. Le premier utilisa les moyens que fournissaient l'horlogerie, l'autre s'inspira des métiers à tisser.

Tout récemment, l'air comprimé et les relais électriques permirent de construire des contrôleurs de processus d'un degré de complexité nettement plus élevé, mais ces mécanismes demeuraient lents, encombrants et coûteux.

Toutefois, en attendant peut-être des solutions techniques encore plus performantes comme la lumière, l'électronique a fourni des moyens déjà raisonnablement efficaces. Surtout sous la forme de circuits intégrés. C'est donc à cette technologie que je vais faire appel pour entreprendre la construction d'un véritable ordinateur.

Il existe différentes catégories de circuits intégrés qui diffèrent essentiellement, non pas tellement par leurs fonctions et leur présentation, mais par des propriétés purement électrique de consommation, de vitesse de réaction, de stabilité etc. Parmi toutes ces catégories, je m'intéresse à deux d'entre elles: la TTL et le MOS.

LANGAGE DE CIRCUITERIE

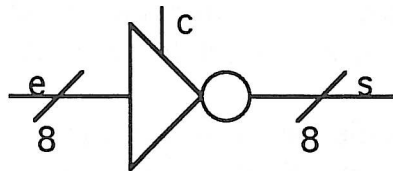
Nous disposons désormais d'une machine universelle valable pour un langage de programmation dont on a montré qu'il nous permet de faire tous les calculs dont on peut avoir besoin, et nous disposons également d'un système susceptible de gérer notre ordinateur pour le cas où nous pourrions le construire.

La définition de ce mot: "ordinateur" coule alors de source, il désigne un mécanisme, d'ordre électronique ou autre, qui contient une mémoire centrale, un algorithme ayant dimension de machine universelle, et s'appliquant sur cette mémoire, un jeu de files externes capable de permettre l'échange d'information avec l'extérieur en quantité suffisante, et un système, programme permanent rendant l'ensemble utilisable.

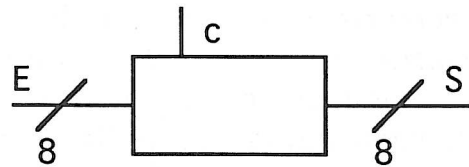
Il reste à montrer qu'on sait construire ce mécanisme. Je l'ai fait au moyen d'un langage de programmation, on va voir que cela est également utile pour aller jusqu'à un câblage matériel. Pour cela je vais définir un langage qui soit commode pour exprimer le phénomène sous une forme proche de ce qui va servir pour matérialiser l'algorithme avec de la circuiterie telle qu'on en trouve dans le commerce.

Je n'ai, en fait besoin que de peu de sortes de circuits, et chacune de ces sortes correspond à un type de circuit intégré des plus courants sur le marché du silicium. Je ne ferai pas la théorie de l'intégration, nombre d'ouvrages font cela très bien. Simplement, plus tard, je donnerai quelques indications pour pouvoir consulter les "data sheets", informations fournies par les fabricants pour la bonne utilisation de leurs circuits.

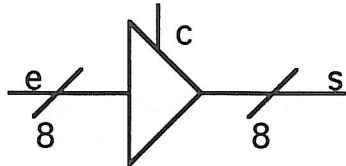
Je désigne chaque type de circuit par son nom en informaticien international, forme sous laquelle on le trouve dans les ouvrages spécialisés. Je vais donc décrire Les buffers, adders, latches, decoders et memories, dont les schémas sont les suivants:



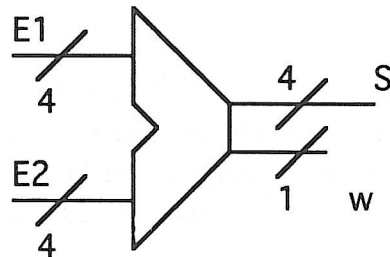
Buffer inverse



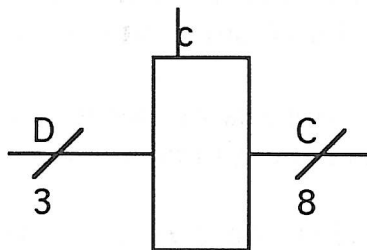
Latch



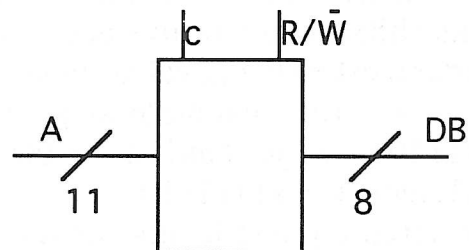
Buffer



Adder



Decoder



Memory

Le buffer Je prends l'exemple de buffers dont le flux est de 8 moments, ce qui signifie qu'il y a 8 fils en entrée et 8 fils en sortie. Ces fils doivent être considérés comme des fils électriques destinés à la propagation d'une certaine énergie sous un certain potentiel. La commande "c" a pour but de servir de robinet, si je lui applique un potentiel maximum (disons 5 volts s'il s'agit de technologie TTL) alors ce circuit fonctionne comme une coupure. Quelle que soit l'information en entrée, en sortie le potentiel est dit "flottant". Au contraire si la commande est au potentiel 0, alors en sortie on retrouve l'information présente en entrée. Par exemple si $e = 10011101$ alors $s = 10011101$.

Le buffer inverse répond de la même façon quant à la commande c, mais lorsque celle-ci est à 0, la sortie est l'inverse de l'entrée, ou encore son complément à 1, ainsi pour $e = 10011101$ on retrouve $s = 01100010$.

Le latch C'est un organe capable de mémoriser 8 bits, dans l'exemple présenté. Ceci se traduit par deux comportements qui dépendent de c. Lorsque $c=1$ alors quel que soit le flux E en entrée, le flux S en sortie demeure inchangé. A partir de l'instant où c passe à zéro, il s'écoule un temps faible mais non nul au bout duquel l'information stabilisée en entrée

se retrouve identique en sortie, et en même temps cette information est mémorisée, de sorte que lorsque c redevient égal à 1, c'est la nouvelle valeur qui est stabilisée en sortie.

Il faut bien remarquer que $c=1$ est la commande de conservation du contenu qui demeure disponible en permanence sur S . Pour changer le contenu du latch, il faut stabiliser la nouvelle valeur en entrée sur E , puis faire $c=0$ pendant une durée suffisante, qui dépend de la technologie du circuit. Quand c remonte le contenu est figé jusqu'à la prochaine variation de c .

L'adder L'adder ne possède pas de commande, il faut stabiliser deux flux en entrée: $E1$ et $E2$, et au bout d'un temps déterminé, en sortie, sur S et w on dispose de la somme arithmétique des entrées, avec le report sur w . Exemple: si $E1=1001$ et $E2=1100$ alors $S=0101$ et $w=1$.

Le decoder Il existe des décodeurs 3×8 comme pour l'exemple donné, ou bien 4×16 . Ceci signifie que si je suppose numéroté de 0 à 7 chacun des 8 fils de C , et qu'en entrée je stabilise sur D une valeur binaire à trois chiffres, je retrouve des 1 partout en sortie sauf un 0 sur le fil dont le numéro est égal à la valeur binaire en entrée.

A titre d'exemple si je stabilise $D=101$, j'obtiens en sortie $C=1111011$. Si je stabilise $D=000$ en sortie $C=01111111$. Et encore si $D=111$, alors $C=11111110$.

Ce circuit possède une commande, ce que je viens de dire est vrai seulement dans le cas où $c=0$, sinon pour $c=1$ quelle que soit la valeur en entrée, en sortie on n'obtient qu'un ensemble de huit 1.

Le circuit 4×16 fonctionne de la même façon. Mais si je numérote les 16 fils de sortie de 0 à 15, quand je stabilise une valeur binaire à 4 chiffres en entrée, c'est le fil dont le numéro est égal à la valeur binaire qui sort un 0, tous les autres sortent des 1.

Memory La mémoire se différencie du latch en ce qu'elle contient un grand nombre de cellules identiques à la cellule unique du latch. En contrepartie il faut fournir au boîtier de cette mémoire une information supplémentaire, destinée à sélectionner une seule d'entre les cellules. Cette information c'est l'adresse. Dans l'exemple, si A comporte un flux à 11 moments, on peut donc choisir entre 2048 cellules différentes.

Le circuit comporte un data bus, c'est-à-dire un jeu de 8 fils destiné à permettre la communication avec le contenu de la cellule sélectionnée. Il existe de tels circuits qui possèdent deux data bus, l'un réservé pour l'introduction et l'autre pour la récupération de l'information. Mais dans l'exemple fourni, le bus unique fonctionne dans les deux sens. Cela oblige à disposer d'une commande supplémentaire généralement dénommée:

R/\overline{W}

notation hybride faite simplement pour rappeler que si ce signal est à 1 la mémoire est en "lecture", on consulte son contenu, et s'il est à 0 elle est en "écriture", on peut modifier le contenu de la cellule "sélectionnée". Il faut ajouter à cela que si $c=1$ le bus est au potentiel flottant, il est donc coupé de son environnement, et $c=0$ rend la communication possible.

Ces quelques cellules de base vont me permettre de constituer ce langage de description des circuits dont j'ai besoin. Pour compléter il me faut quelques règles de connexion.

1) De toute évidence, une sortie de circuit ne peut être connectée qu'à une entrée de circuit, et réciproquement.

2) On verra apparaître des situations qui obligent à connecter des cellules destinées à échanger de l'information entre elles. Pour cela toutes les entrées toutes les sorties se retrouvent connectées sur un même point qu'on appelle alors BUS. Il faut prendre des précautions telles qu'un seul circuit puisse être émetteur même si plusieurs peuvent sans inconvénient être récepteurs.

Il y a exception à cette règle quand plusieurs émetteurs sont reliés au bus par un circuit qui combine les divers flux. C'est le cas d'un adder.

3) Il faut en plus qu'en entrée active on ait toujours un signal "maintenu" et jamais un signal flottant. Faute de quoi on obtient en sortie un signal aléatoire.

LE SCHEMA DU PROCESSEUR

Je propose un schéma qui va me permettre de définir un ensemble de calculs élémentaires à partir desquels on va essayer de construire le processeur.

On observe d'abord que les 6 cellules m_1 , m_2 , m_3 , m_4 , pc et mw sont toutes connectées, entrées et sorties sur le même bus μB . Directement pour leurs entrées, et par l'intermédiaire d'un buffer pour les sorties. Ceci tient au fait que ces six cellules, des latches, émettent en permanence un flux en sortie, alors que leurs entrées ne deviennent perméables que sous l'effet de la commande c .

Pour décrire les calculs susceptibles de se dérouler dans un tel circuit, je vais reprendre une écriture conforme aux habitudes. Ainsi, l'utilisation de la cellule m_1 se fait en désignant son contenu par cm_1 . Un échange entre m_1 et m_2 prend deux formes: $cm_1 := cm_2$ et $cm_2 := cm_1$ qui présentent le sens habituel des affectations. Pour rendre possibles de tels échanges je fais d'abord l'hypothèse que toutes les commandes sont au potentiel 1, sauf les commandes des cellules mises en jeu.

opérations	commandes
cpc := cm1	s1, cpc
cpc := cm2	cpc, s2
cpc := cm3	cpc, s3
cpc := cm4 + κ 1	cpc, sa4, un
cm2 := cpc	c2, spc
cm4 := cpc	c4, cpc
cm1 := cM	c1, cM, $\epsilon\mu$ B, sA
cM := cm1	$\sigma\mu$ B, cM, s1, sA, R/W
cM := cm2	s2, $\sigma\mu$ B, cM, sA, R/W, c2
cM := cw	ew, cw, $\sigma\mu$ B, cM, sA, R/W
cM := 0	$\sigma\mu$ B, cM, zéro, sA
cm2 := cM	c2, $\epsilon\mu$ B, sA, cM
cm3 := cM	c3, $\epsilon\mu$ B, sA, cM
cm4 := cM	c4, $\epsilon\mu$ B, sA, cM
cm1 := cm4 + κ 1	c1, sa4, un, sa, cw
cm2 := cm4 + κ 1	c2, sa4, un, sa, cw
cm2:=cm3+ κ cm4	c2, sa3, sa4, sa, cw
cm2:=cm3- κ cm4	c2, sa3, ss4, sa, cw
cw := cM	ew, cw, $\epsilon\mu$ B, sA, cM

Jeu de commandes du processeur

Plus loin, je détaillerai 5 autres instructions dont j'ai également besoin, mais qui font intervenir le calcul de l'étiquette, qu'on étudiera tout spécialement.

On peut observer que je n'explique pas toutes les combinaisons possibles entre les diverses cellules, car toutes ne me sont pas utiles. L'écriture M signifie que je m'adresse à la mémoire, pour accéder à la case désirée il faut au préalable charger pc avec la bonne valeur d'adresse.

Au moment du départ d'un calcul, cette valeur, imposée de l'extérieur, doit être celle qui permet d'atteindre au code-opérateur de la première instruction à traiter.

LE LANGAGE DU PROCESSEUR

Je reprends le code choisi plus haut pour le langage de la machine de Nolin en y rajoutant deux instructions, l'addition et la soustraction. Les algorithmes dont je voudrai obtenir un déroulement, seront codés selon le tableau qui suit, et dûment introduits en mémoire.

06(f) Z : $cm4 := cpc$ Adresse de la partie
suivante du code.

07(g) $cpc := cm4 +_K 1$

08(h) $cm1 := cM$ Appel de la valeur 'a'.

09(i) $cm4 := cpc$ Réserve de l'accès au
code.

0A(j) $cpc := cm1$ Valeur de 'a' en pc.

0B(k) $cM := 0$ Zéro en adresse a.

0C(l) $cpc := cm4 +_K 1$, vers Entrée

0D(m) DPE : $cm4 := cpc$

0E(n) $cpc := cm4 +_K 1$

0F(o) $cm3 := cM$ a en m3.

10(p) $cm4 := cpc$

11(q) $cpc := cm4 +_K 1$

12(r) $cm1 := cM$ b en m1.

13(s) $cm4 := cpc$ Réserve de l'accès au
code.

14(t) $cpc := cm1$ b en pc.

15(u) $cm1 := cM$ Contenu de b en m1.

16(v) $cpc := cm3$ a en pc.

17(w) $cM := cm1$ Contenu de b en a.

18(x) $cpc := cm4 +_K 1$, vers Entrée

19(y) PLUN : $cm4 := cpc$

1A(z) $cpc := cm4 +_K 1$

1B $cm1 := cM$ a en m1.

1C $cm2 := cpc$

1D $cpc := cm1$ a en pc.

1E $cm4 := cM$ contenu de a en m4.

1F $cm1 := cm4 +_K 1$ Plus un modulo K sur ca.

20 $cM := cm1$ Renvoie en a.

21 $cm4 := cm2$

22 $cpc := cm4 +_K 1$, vers Entrée

23 ADD : $cm4 := cpc$

24 $cpc := cm4 +_K 1$

25 $cm1 := cM$ a en m1.

26 $cm4 := cpc$

27 $cpc := cm4 +_K 1$

28	$cm3 := cM$	b en m3.
29	$cm4 := cpc$	
2A	$cpc := cm3$	
2B	$cm3 := cM$	cb en m3.
2C	$cpc := cm4 +_{\mathbb{K}} 1$	
2D	$cm2 := cM$	d en m2.
2E	$cm4 := cpc$	
2F	$cpc := cm2$	
30	$cm2 := cM$	cd en m2.
31	$cpc := cm4 +_{\mathbb{K}} 1$	
32	$cm4 := cm2$	
33	$cm2 := cm3 +_{\mathbb{K}} cm4$	Le résultat: $cb +_{\mathbb{K}} cd$ en m2
34	$cm4 := cpc$	
35	$cpc := cm1$	
36	$cM := cm2$	Le résultat en ca.
37	$cpc := cm4$, <u>vers</u> Entrée	

38	SOUS : $cm4 := cpc$	
39	$cpc := cm4 +_{\mathbb{K}} 1$	
3A	$cm1 := cM$	
3B	$cm4 := cpc$	
3C	$cpc := cm4 +_{\mathbb{K}} 1$	
3D	$cm3 := cM$	
3E	$cm4 := cpc$	
3F	$cpc := cm3$	
40	$cm3 := cM$	
41	$cpc := cm4 +_{\mathbb{K}} 1$	
42	$cm2 := cM$	
43	$cm4 := cpc$	
44	$pc := cm2$	
45	$cm2 := cM$	
46	$cpc := cm4 +_{\mathbb{K}} 1$	
47	$cm4 := cm2$	
48	$cm2 := cm3 -_{\mathbb{K}} cm4$	En m2 le résultat est en complément
49	$cm4 := cm2$	à 1, on ajoute 1 pour l'obtenir en complément à 2.
4A	$cm2 := cm4 +_{\mathbb{K}} 1$	
4B	$cm4 := cpc$	
4C	$cpc := cm1$	
4E	$cM := cm2$	
4F	$cpc := cm4$, <u>vers</u> Entrée	

Il reste encore trois instructions à traiter, les deux comparaisons et la commutation systématique. J'insiste un peu sur la technique de codage de l'étiquette. Je reprends un vieil exemple un peu modifié:

	MU : <u>début</u>		'deb'
R1	c0 := 0		':=0', 0
R2	c0 := c0 +K 1		':=+', 0
R3	E : c8 := c8 +K 1	E	':=+', 8
R4	c0 := c0 +K 1		':=+', 0
R5	<u>si</u> c0 = 0 <u>vers</u> FIN		'si=', 0, code FIN
R6	<u>vers</u> E		'vers', code E
R7	FIN : <u>fin</u>	FIN	'fin'

Je suppose que je place le code de cet algorithme, en mémoire à partir de la case 32:

	'deb'	':=0'	0	':=+'	0	':=+'		8	':=+'	0	'si='	0		5	vers	K-7	'fin'		
	32			35			E			40							45		FIN

Il est facile de constater que le processeur qui rencontre le code 'vers' doit disposer de l'information qui lui permet d'atteindre le code de l'instruction étiquetée E. Il dispose de l'adresse de la case qui contient 'vers'. Si on lui fournit le nombre de cases qui la sépare de celle qui contient le code opérateur repéré par E on compte -7, valeur qui, ajoutée à 44 donne 37 nom de la case qui contient bien le ':=+'. On porte donc K-7 en case 45, valeur équivalente à -7 en arithmétique modulaire.

Pour la condition qui renvoie à FIN, le processeur dispose de l'adresse du code opérateur, ici 41 et il suffit de lui fournir 5 tel que 41+5 lui donne l'adresse du code de l'instruction étiquetée FIN.

Je complète l'algorithme du processeur:

50	VERS: cm4 := cpc	Forme: <u>vers</u> F
51	cpc := cm4 +K 1	
52	cm3 := cM	
53	cpc := cm3 + cm4 , <u>vers</u> Entrée	
54	SEG :cm4 := cpc	
55	cm3 := cpc	Réservation de l'adresse du code.
56	cpc := cm4 +K 1	
57	cm1 := cM	a dans m1.
58	cm2 := cpc	
59	cpc := cm1	
5A	cm1 := cM	ca en m1
5B	cm4 := cm2	

5C	<u>si</u> cm1 = 0 <u>vers</u> SEG1	Comparaison de ca avec 0. Sinon, on passe au code suivant.
5D	cpc := cm4 + κ 1	
5E	cm4 := cpc	
5F	cpc := cm4 + κ 1 , <u>vers</u> Entrée	
60	SEG1 : cpc := cm4 + κ 1	Récupération du code étiquette.
61	cm4 := cM	
62	cpc := cm3 + κ cm4 , <u>vers</u> Entrée	
63	SIOM : cm4 := cpc	Forme: <u>si</u> cw = 0 <u>vers</u> P
64	cm3 := cpc	
65	cpc := cm4 + κ 1	
66	<u>si</u> cw = 0 <u>vers</u> SIO1	
67	cm4 := cpc	
68	cpc := cm4 + κ 1 , <u>vers</u> Entrée	
69	SIO1 : cm4 := cM	
6A	cpc := cm3 + κ cm4 , <u>vers</u> Entrée	
6B	OMA: cm4 := cpc	Forme: cw := ca
6C	cpc := cm4 + κ 1	
6D	cw := cM	
6E	cm4 := cpc	
6F	cpc := cm4 + κ 1 , <u>vers</u> entrée	
70	AOM: cm4 := cpc	Forme: ca := cw
71	cpc := cm4 + κ 1	
72	cM := cw	
73	cm4 := cpc	
74	cpc := cm4 + κ 1 , <u>vers</u> entrée	

"Algorithme" du processeur: lignes 01H à 74H

J'ai besoin de cinq sortes d'instructions élémentaires en plus, dont en particulier trois opérations de commutation qui exigent une circuiterie adaptée: le séquenceur.

Opérations

cm3 := cpc
si cw = 0 vers SIO1
cpc := cm3 + κ cm4
si cm1 = 0 vers SEG1
Aig(m1){. . . }

Commandes

c3 , spc
cmpw
cpc , sa3 , sa4
s1 , cmpa
s1 , aig

LANGAGE INTERNE

Après avoir construit l'algorithme interne au processeur, celui qui décrit son fonctionnement, il faut montrer comment il est enregistré en mémoire. L'observation du schéma PROC 1 montre que les cellules de travail, images des circuits intégrés, ont chacune, au moins une commande du genre: cpc, sA, s1, c1, un, zéro etc.

Je fais la convention que c'est la valeur "0" de la commande qui rend la cellule active, et la valeur "1" qui la rend inactive. Ce choix correspond à ce qui se pratique généralement en TTL, et moins en C-MOS. Peu importe, dans le cas d'une convention contraire, il suffit de renverser les valeurs dans les exemples que je montre.

De cette façon, dire que $c1=0$ signifie que l'information stabilisée sur le bus sm1 s'enregistre dans la cellule m1. Dire que $sA=0$ implique que le contenu de pc, considéré comme une adresse, attaque la mémoire "memory", en sélectionnant une case.

A titre d'exemple, l'instruction:

$cm1 := cM,$

qui indique que la cellule de la mémoire, sélectionnée par l'adresse issue de pc, transmet son contenu à la cellule m1, via le data-bus DB, se représente par les valeurs:

$c1=0, sA=0, cM=0, \epsilon\mu B=0$, et toutes les autres commandes sont à 1.

J'indique le code des vingt six premières instructions du programme interne, dans un tableau qui rassemble la liste de toutes les commandes. Je donne cette liste dans l'ordre et le tableau ensuite:

- cpc, sA, Spc, z7 : cellule pc.
- sw, cw, smw, ew: report.
- sa, un, sa3, sa4, ss4 : adder.
- c1, s1, c2, s2, c3, s3, c4, s4 : cellules m1, m2, m3, m4.
- zéro, $\sigma\mu B$, $\epsilon\mu B$: data-bus, micro-bus.
- cM, R/W : mémoire.
- Aig, smw, cmpw, cmpa, SEG1 : le séquenceur que nous allons construire plus loin.

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
cpc	1		0	0	0	0	0		0	0		0	0	0		0	0		0	0		0		0	0	0
sa	0							0		0					0			0		0		0				
spc	1																									
z7	1																									
sw	1																									
cw	1																									
smw	1																									
ew	1																									
sa	1																									
un	1			0			0				0		0			0								0		0
sa3	1																									
sa4	1			0			0				0		0			0								0		0
ss4	1																									
c1	0							0									0			0						
s1	1	0								0										0			0			
c2	1																									
s2	1																									
c3	1														0											
s3	1																					0				
c4	1		0		0	0		0				0				0				0						0
s4	1																									
zéro	1									0																
$\sigma\mu B$	1									0													0			
$\epsilon\mu B$	0							0							0		0			0						
cM	0							0		0					0		0			0		0				
R/W	1																						0			
Aig	1	0																								
smw	1																									
cmpw	1																									
cmpa	1																									

Tableau: codes des 26 premières instructions du programme interne

Dans ce tableau, chaque colonne donne le code interne d'une instruction du processeur. Par exemple, la colonne notée: a , qui correspond à l'instruction cm1 := cM , s'écrit en hexadécimal; en lisant du haut vers le bas:

BFFBFE73,

la seconde s'écrit:

FFFDFFD3.

Les vingt six premières lignes de ce programme occupent donc les vingt six premières cellules, numérotées 00 à 1A en hexadécimal, des quatre PROM PS1, PS2, PS3 et PS4 du schéma PROC-2.

LA FONCTION DE SEQUENCEMENT

Dans son fonctionnement normal le séquenceur fait deux choses à la fois, il émet une adresse à la sortie de L2 qui attaque la batterie de PROMs PS1, PS2, PS3 et PS4. En même temps il enregistre en L1 la valeur de l'adresse suivante.

L'adresse de quoi? Est-on raisonnablement fondé à se demander. La partie calcul du processeur fonctionne pendant une tranche de temps sous l'action d'un jeu de commandes. Ainsi quand on lui fournit, stabilisés pendant une durée suffisante, je dirai une période, un ensemble convenable des valeurs des commandes, le processeur réalise un élément de calcul: une instruction élémentaire. Un calcul complet se fera avec une suite de telles instructions élémentaires. C'est ce qui a été exposé au chapitre précédent.

Le séquenceur est l'organe qui est capable d'émettre la suite des jeux de valeurs des commandes qui sont les images de ces instructions. Il existe en tout 30 commandes. Il faut donc pouvoir à chaque instant fournir ces 30 bits. C'est une batterie de mémoires définitives, des PROM qui émettent l'information. Ce sont ces PROM, désignées par PS1, PS2, PS3 et PS4 dans lesquelles on enregistre au préalable la suite des instructions décrites plus haut et numérotées en hexadécimal de 01 à 74.

L'adresse enregistrée en L1 peut avoir deux origines, l'une, quand on déroule des suites d'instructions sans commutation, est fournie par la PROM PAS, le buffer X est passant, l'autre origine lorsqu'il y a commutation, donc, adresse suivante calculée, l'adresse est injectée sur le bus ASC, ce qui implique que le buffer X soit bloqué.

Le tableau des commandes donne pour chaque instruction la liste des valeurs de chacune d'entre elles. Pour éclaircir le tableau je n'ai pas porté toutes les valeurs "1". Ainsi, chaque case vide est considérée contenir un "1". Je reporte en ligne ces valeurs mais en les écrivant en hexadécimal. Je rajoute le contenu de PAS qui représente la ligne suivante. Quand L2 émet un numéro de ligne, par exemple 3, la quantité 7F AF FF F3 est émise sur le bus des commandes et 4, émis par PAS attaque l'entrée de L1. C'est ainsi qu'au basculement du coup d'horloge suivant, le contenu de L1 remplace celui de L2 qui se met à émettre la nouvelle valeur: 4 qui va permettre de dérouler la commande 7F FF EF F3, tandis que PAS émet 0 pour L1.

Lignes	PAS	PS1	PS2	PS3	PS4	
0	1	BF	FB	FE	73	
1		FF	FD	FF	D3	Aiguillage
2	3	7F	FF	EF	F3	
3	4	7F	AF	FF	F3	
4	0	7F	FF	EF	F3	
5	0	7F	FF	EF	F3	
6	7	7F	AF	FF	F3	
7	8	BF	FB	FE	73	
8	9	7F	FF	EF	F3	
9	A	7F	FD	FF	F3	
A	B	BF	FF	F9	73	
B	C	7F	AF	FF	F3	
C	D	7F	FF	EF	F3	
D	E	7F	AF	FF	F3	
E	F	BF	FF	BE	73	
F	10	7F	FF	EF	F3	
10	11	7F	AF	FF	F3	
11	12	BF	FB	FE	73	
12	13	7F	FF	EF	F3	
13	14	7F	FD	EF	F3	
14	15	BF	FB	FE	73	
15	16	7F	FF	DF	F3	
16	17	BF	FD	FD	33	
17	18	7F	AF	FF	F3	
18	0	7F	FF	EF	F3	
19	1A	7F	AF	FF	F3	

Contenu de la PROM PAS: suite d'instructions

On voit clairement sur ce tableau que la PROM PAS contient à chaque ligne le numéro de la ligne suivante, par exemple à la ligne 14, PAS contient: 15.

Seules les lignes qui contiennent des instructions de commutation comme les lignes 2, 50 d'étiquette VERS, 54 d'étiquette SEG et 63 d'étiquette SIOM ont un contenu de PAS sans signification, car L1, lors de leur déroulement est chargé par une autre provenance.

COMMUTATION DANS LE PROCESSEUR

Quand on observe le programme interne, celui contenu dans la batterie de PROMs: PS1, PS2, PS3 et PS4, on trouve trois instructions qui ne

représentent pas du calcul au sens de modification de facteurs, mais des ruptures conditionnelles dans le séquençement des instructions, ce sont:

- 1) si $cw = 0$ vers SIO1
- 2) si $cm1 = 0$ vers SEG1
- 3) Aig(m1)

Les deux premières sont faciles à traiter: si la condition n'est pas vérifiée, on déroule après elle l'instruction qui la suit dans l'ordre. Si la condition est vérifiée alors on déroule après elle l'instruction étiquetée SIO1 pour la 1) et SEG1 pour la 2).

Dans le cas de 1), la cellule w , par le canal smw attaque un circuit NOR, c'est alors la commande $cmpw$ qui valide le contenu de w . En effet c'est seulement sur la rencontre de l'instruction 1) que $cmpw$ prend la valeur 0, de telle manière que si $cw = 0$, condition vérifiée, 0 et 0 vont donner 1 en sortie du NOR. Ce 1 attaque le OU qui sort un 1 sur le buffer X et le bloque. Tandis que ce même 1 se transforme en 0 à travers le buffer-inverseur qui attaque le buffer SIO1. Ce buffer SIO1 devient passant et injecte sur le bus ASC la valeur hexadécimale 69, qui va être enregistrée dans L1, et devient donc l'adresse de la prochaine instruction à dérouler.

Si la cellule w contient 1, condition non vérifiée, la commande $cmpw$ étant elle, à 0, 1 ou 0 donnent 0 en sortie du NOR, ce 0 attaque le buffer X en le laissant passant, devient 1 à travers l'inverseur le buffer SIO1 en le verrouillant. Ainsi c'est l'adresse émise par la PROM PAS qui est enregistrée par L1.

REMARQUE

Pour bien saisir ce jeu il faut voir que sauf sur rencontre d'une instruction de commutation, le buffer X est toujours commandé par un 0 qui le rend passant. En effet, hors ces cas, les commandes $cmpa$, Aig et $cmpw$, demeurent par construction maintenues à la valeur 1. Or il suffit que $cmpa$ soit à 1 pour qu'en sortie du NOR qu'il attaque, il y ait un 0, quelle que soit l'autre valeur qui attaque ce NOR. Donc ce 0 sera l'une des entrées du OR qui commande le X. De même, Aig est à 1 et l'inverseur lui fait attaquer le même OU par un 0. La troisième entrée de ce OU est encore à 0, tant que $cmpw$ reste à 1, comme on vient de le voir.

Conséquence, tant que ces commandes: $cmpa$, Aig et $cmpw$ restent à 1, donc en dehors de toute instruction de commutation, le buffer X est commandé par un 0 qui le rend passant, alors que les trois autres buffers demeurent verrouillés.

Dans le cas de 2), il s'agit de vérifier si le contenu $m1$ est nul ou non nul. La cellule $m1$ est reliée par le bus $sm1$ aux huit entrées d'un circuit OR de

sorte qu'à la sortie de ce OR on trouve 0 seulement si sur aucun des huit fils on ne rencontre de 1. Ainsi ce OR émet un 0 si $cm1 = 0$ et émet 1 dans tous les autres cas. La sortie du OR attaque un NOR et ne sera validée que si $cmpa = 0$, ce qui est forcément le cas pour l'instruction 2).

La condition $cm1 = 0$ fait donc apparaître 0 et 0 en attaque du NOR, donc 1 à sa sortie, d'une part ce 1 va bloquer le buffer X, et après inversion débloque le buffer SEG1. C'est donc la valeur hexadécimale 60 qui devient l'adresse de l'instruction suivante en L1. Par contre, le moindre 1 dans le contenu de m1 fait apparaître un 1 à la sortie du OR et le couple 1, 0 donne 0 en sortie du NOR d'où, X reste commandé par un 0, il demeure passant, et c'est un 1 qui commande le buffer SEG1 qui est bloqué. La condition n'est pas vérifiée.

Dans le cas de 3) On voit sur le schéma du séquenceur que le contenu de la cellule m1, par le canal sm1 attaque directement le bus ASC, quand la commande Aig passe à 0. A cet instant le buffer X est bloqué, c'est donc ce qui provient de m1 qui va être enregistré dans L1 comme adresse de l'instruction suivante. Or, m1 contient le "fetch", en l'occurrence le type de l'opérateur de l'instruction interprétée. C'est-à-dire l'un des codes que j'ai désignés par:

':=0' ':=' ':='+ 'si=' 'vers' 'siw' 'wa' 'aw' 'deb' 'fin' '+' '-'

Il est clair que, peu importe la valeur choisie comme code pour ces opérateurs, pourvu qu'elle permette de les distinguer les uns des autres. Pour l'exemple, j'avais choisi les entiers de 0 pour ':=0' à 11 pour '-'.

Mais il est non moins clair, en se référant au schéma du séquenceur, que la valeur à choisir est le numéro de la ligne du programme interne à laquelle l'aiguillage doit renvoyer:

Opérateur	Etiquette	code	Opérateur	Etiquette	code
':=0'	Z	06	'wa'	OMA	6B
':='	DPE	0D	'aw'	AOM	70
':=+'	PLUN	19	'deb'	DEB	03
'si='	SEG	54	'fin'	FIN	05
'vers'	VERS	50	'+'	ADD	23
'siw'	SIOM	63	'-'	SOUS	38

Une dernière remarque concernant le rôle de la PROM PAS qui contient l'adresse de la ligne suivante. A l'exception, comme on vient de le voir des trois instructions de commutation:

si cw =0 vers SIO1
si cm1 = 0 vers SEG1
 Aig(m1)

pour lesquelles l'adresse de la ligne suivante est calculée, PAS contient sur chaque ligne le numéro de la ligne qui suit, sauf pour les dernières lignes de chacun des traitements telles que:

la ligne 4 qui ponctue le traitement de 'deb' et qui contient 0,
traduction du "vers entrée",

la ligne 5 qui ponctue le traitement de 'fin',

la ligne 0C qui achève le traitement de ':=0',

la ligne 18 qui termine l'instruction ':=', etc.

COMPLEMENTS

A titre de complément, je donne le détail des autres instructions, avec pour chacune, la listes de ses commandes qui doivent être à zéro, et le code correspondant en hexadécimal, tel qu'il est enregistré dans les PROM du séquenceur.

instructions	commandes à zéro	code en PROM
<u>si</u> cw=0 <u>vers</u> SIO1	cmpw	FFFF FFF1
<u>si</u> cm1=1 <u>vers</u> SEG1	s1, cmpa	FFFD FFF3
cm1 := cm4 + _K 1	c1, sa4, un, cw, sA	FB2B FFF3
cm2 := cm4 + _K 1	c2, sa4, un, cw, sA	FB2E FFF3
cm2 := cm3 - _K cm4	c2, sa3, ss4, cw, sA	FB56 FFF3
cm2 := cm3 + _K cm4	c2, sa3, sa4, cw, sA	FB4E FFF3
cm2 := cM	c2, cM, εμB	FFFE FE73
cm2 := cpc	c2, spc	DFFE FFF3
cm4 := cm2	c4, s2	FFFF 6FF3
cm4 := cM	c4, cM, εμB	FFFF EE73
cpc := cm3 + _K cm4	cpc, sa3, sa4, cw, sA	7B4F FFF3
cpc := cm1	cpc, s1	7FFD FFF3
cpc := cm2	cpc, s2	7FFF 7FF3
cM := cm2	s2, σμB, R/W	FFFF 7DB3
cw := cM	cM, εμB, cw, ew	FCFF FE73
cM := cw	R/W, σμB, z7, sw	E7FF FDB3
cm3 := cpc	c3, spc	DFFF BFF3

Autres instructions: commandes et code hexadécimal

REALISATION

Ce processeur qui a été construit et a parfaitement fonctionné a été inauguré en 1981 à l'Université d'Aix-Marseille II, Marseille-Luminy. C'est le résultat du travail de l'équipe du LITAM que je dirigeais alors. Le marché des circuits intégrés était encore balbutiant, et nous avons choisi la technologie TTL car c'était à cette époque, le plus facile à trouver, et la plus fiable. Le MOS quant à lui était encore très fragile et largement incomplet. Ce premier processeur, décrit dans le numéro Zéro du Bulletin d'Informatique Fondamentale et Applications, constitué, donc de TTL, était vorace en électricité, et nous avons dû fabriquer nous-mêmes nos alimentations en 5 volts régulés. Le laboratoire était alors trop pauvre pour s'offrir un programmeur de PROMs, le générateur de commandes était constitué d'autant de jeux de buffers qu'il y avait de commandes. Et cela ne réduisait pas la consommation.

VOUZZAVEDIBISAR,

Symboles.

J'avais à faire un exposé devant un parterre de doctes personnages. C'était à l'occasion d'une inauguration, je crois. L'inauguration d'un nouvel enseignement. La scène se passe dans cette bonne ville d'Aix. J'avais décidé de parler de la Machine de Nolin. Une occasion d'ouverture des bases de l'informatique à un public de non initiés. Après avoir réussi à endormir ostensiblement une bonne moitié de l'assistance, et plus discrètement l'autre moitié, j'achevais sur une conclusion sonore et dynamique qui réveilla tout le monde.

Comme chaque fois dans ce genre de prestation, les gens se levèrent, heureux de se détendre un peu et se rassemblèrent en petits groupes... pour parler d'autre chose. On a si peu d'occasions pour se retrouver. L'un de mes chers collègues, une sorte d'ami de trente ans, l'organisateur du séminaire, se précipita vers moi pour me remercier de mes efforts. Il en profita pour me faire remarquer combien mon propos était proche de l'intelligence artificielle, et avec un pas de plus, de l'intelligence naturelle, car ma Machine calculait du symbole!

Un peu surpris, je ne pus qu'opiner. Mais cette remarque me travailla longtemps.

A bien y réfléchir le symbole est une des bases solides de notre société, et les concepts de notre culture procèdent par symboles. On pourrait même dire, après avoir pendant des siècles, peiné à dégager des masses de symboles pour représenter les notions courantes du savoir, qu'on ne sait plus distinguer les symboles de ce qu'ils sont censés représenter. Au point qu'ils deviennent des entités à part entière, complètement détachées de leur origine, et à qui on est amenés à vouer une dévotion personnelle. Prenons quelques exemples. Partons d'une notion actuellement très populaire: la démocratie. Tout le monde en parle, beaucoup s'y réfèrent, soit pour en vanter les mérites, soit pour en pourchasser les défauts. Elle a un fondement politique incontestable, c'est le vote. Le citoyen élit des représentants, des gens censés défendre ses intérêts devant les instances nationales, régionales, départementales, communales. Hélas, bien souvent on le dépouille de son scrutin, car il y a loin des promesses électorales à l'exercice du pouvoir. Il y a d'autres domaines où sévit le symbole "démocratie". Je discutais, un jour, avec un éminent chercheur en science,

modestement "directeur général". Plus modestement quand à l'éclat de métal rutilant du symbole, mais pas toujours quant au vrai pouvoir, car certains "PDG" de certains grands groupes industriels sont bien moins connus mais bien plus puissants que nombre de présidents de républiques qui leur doivent trop souvent leur promotion. Question de moyens financiers. Parlons-en des moyens financiers.

Le sang de nos sociétés modernes est représenté par un symbole aux reflets changeants et irisés, aux tons brûlants et glacés à la fois, qui émerge du langage sous une diversité de forme rarement atteinte par d'autres symboles. On dit avoir de "l'argent", mais on possède de "l'or", métal suprême qui vous possède plus que vous ne le possédez. On a aussi du "fer", du "blé", du "flouz", du "fric", du "pognon", des "ronds", de "l'oseille", de la "monnaie", du "liquide", et je laisse de côté les noms des monnaies qui ont cours dans les rues de la terre. Le "fric" c'est comme le "nombre", il y a autant de noms qu'on le désire pour le désigner, et si l'on n'a jamais vu un seul nombre, mais seulement l'un des noms qui le désignent, on n'a jamais vu d'argent autrement que sous forme de nombre, ne serait-ce que celui imprimé sur un morceau de papier plus ou moins spécial. Jugez de la profondeur de l'abstraction. Comment voulez-vous alors qu'on ne se fasse pas largement gruger par des maîtres de l'abstraction aussi dénués de scrupules que le nombre lui-même est dénué de réalité.

Edmond Bianco

A suivre.

**Université de Provence
Atelier de Reprographie
Centre Saint Charles
3, place Victor Hugo
F - 13331 Marseille Cedex 3**

