



HAL
open science

La Saga des machines-langage et -système

François Anceau

► **To cite this version:**

François Anceau. La Saga des machines-langage et -système. Cahiers d'histoire du Cnam, 2017, La recherche sur les systèmes: des pivots dans l'histoire de l'informatique, vol.07 - 08 (2), pp41-52. hal-03027079

HAL Id: hal-03027079

<https://hal.science/hal-03027079>

Submitted on 27 Nov 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

La Saga des machines-langage et -système

François Anceau

Collaborateur bénévole, Lip 6, Sorbonne-Université/UPMC.

Résumé

Parallèlement à l'évolution des machines informatiques standard, la recherche et l'industrie informatique ont développé, dès le début des années 1960, des machines originales fondées sur l'idée de déplacer la frontière entre le logiciel et le matériel au profit de ce dernier. Cette évolution de l'architecture des machines s'appuyait sur l'évolution technologique (rapidité du matériel et possibilité de réaliser des machines complexes). Plusieurs nouvelles classes de machines en ont découlé. Nous retiendrons ici la classe des machines-langage orientées vers l'exécution directe de langages évolués et celle des machines-système possédant des mécanismes systèmes au niveau matériel et surtout accédant à ses divers objets via des descripteurs. Cette évolution promettait une simplification et une accélération des fonctions appartenant à l'origine aux couches logicielles. Ces espoirs n'ont pas été complètement satisfaits. Après un départ glorieux, les machines-langage ont disparu dès les années 2000, et les machines-système ont été simulées sur d'autres machines pour

continuer à assurer leurs services. Seule, la lignée des machines-système x86 et amd64 a continué sa progression grâce à sa prééminence dans les ordinateurs personnels. Depuis, cette lignée se trouve en compétition avec les ARM venus du monde de l'enfouï et des portables.

Mots clés : architecture des ordinateurs, machines spécialisées, machines-langage, machines-système, architecture virtuelle, intérieur-décor.

Introduction

Dès le début des années 1960, les concepteurs de machines informatiques ont essayé de casser le schéma conceptuel dans lequel ils commençaient à s'enfermer. À cette époque, le traitement de l'information se résumait au calcul numérique. Ces calculs se présentaient alors sous deux formes très distinctes :

- les calculs simples se répétant sur un ensemble assez important de données : par exemple le calcul des feuilles de paie, la réalisation de tables (logarithmes, trigonométriques, éphémérides pour la navigation, tables de tir...), relevant du domaine de la mécanographie ;
- les calculs non répétitifs réalisés à la demande (rares et coûteux).

À cette époque, il était beaucoup plus simple de consulter une table, que de calculer un logarithme, ou une fonction trigonométrique, compte tenu de la lenteur, et de la rareté de ces machines à calculer.

L'électronique était alors en pleine mutation grâce à l'arrivée des transistors au silicium dans le monde de l'électronique industrielle en remplacement des tubes à vide. Outre une diminution de taille et de poids, l'arrivée de ces composants allait transformer l'utilisation des machines informatiques par l'augmentation importante de leur fiabilité, faisant passer de la journée à la semaine leur temps moyen de fonction-

nement entre pannes, permettant ainsi les longues exécutions des programmes devenus plus complexes.

Le passage du calcul numérique au traitement de l'information, se déroula pendant les années 1960. Cette extension du domaine d'application de ce genre de machine s'appuyait sur des travaux théoriques montrant la généralité de la notion de calcul :

- la thèse de Church-Turing qui montre que tout calcul peut se faire par la combinaison de quelques opérations simples, ce qui a permis de définir les limites des machines à calculer programmables ;
- la diagonalisation de Gödel montre que toute transformation d'ensembles énumérables peut se coder comme un calcul ;
- le théorème de programmation structurée de Corrado Böhm et Giuseppe Jacopini (1966) a ouvert la porte de la programmation moderne.

Toutefois l'unification entre le calcul et le traitement de l'information était pressentie depuis longtemps. G. W. Leibniz, avec sa « Caractéristique Universelle », avait déjà compris en 1666 la possibilité d'utiliser le calcul logique pour autre chose que des nombres. Ada Lovelace avait aussi prédit cette évolution en 1842 dans ses commentaires de sa traduction de la description de la machine de Babbage (Tools, 1992). Toutefois, sa mise en œuvre sera progressive pour n'être effective que vers 1965, provo-

quant la naissance conjointe de l'informatique et des ordinateurs (Mounier-Kuhn, 2010 ; Anceau, 2000, p. 89). En effet, beaucoup de ces machines à calculer ne pouvaient traiter que des chiffres par des codages plus ou moins optimisés pour le calcul (par exemple le code bi-quinaire) ce qui rendait la manipulation de lettres ou d'autres items particulièrement difficile. L'arrivée de l'informatique et des ordinateurs est marquée par la possibilité de manipuler individuellement des éléments binaires regroupés dans des mots.

Machines spécialisées

Très tôt, l'idée de spécialiser les machines informatiques afin de les optimiser pour des tâches précises inspira les concepteurs. De nombreuses voies s'ouvraient alors, allant de l'adaptation de la

structure des machines de base à un type d'application, jusqu'au remplacement des langages d'instruction par des langages de programmation préexistants. Dans ce texte, nous ne traiterons que des processeurs et coprocesseurs-langage et des processeurs-système :

- les processeurs et coprocesseurs-langage exécutent directement les langages intermédiaires des langages de programmation ;
- les processeurs-système exécutent directement des fonctions systèmes élaborées, comme l'adressage segmenté, la gestion des tâches, leur synchronisation et l'isolement hiérarchique par des anneaux de protection entre les couches du système et celles des utilisateurs.

La Figure 1 donne un aperçu de la grande variété des processeurs spécialisés.

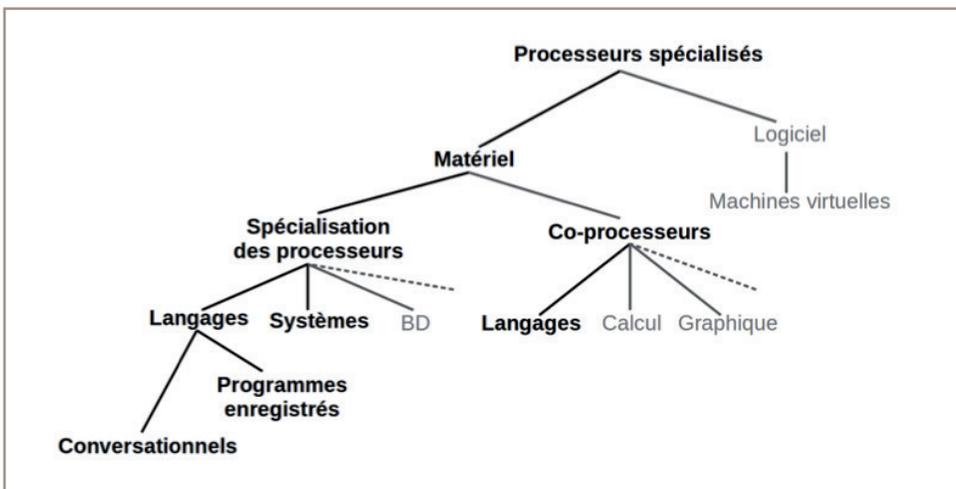


Figure 1 - Famille des processeurs spécialisés

Machines-langage

Le début des années 1960 vit la naissance des premiers langages informatiques capables de décrire des processus de traitement de l'information. Dès le départ, ceux-ci se divisèrent en plusieurs classes :

- les langages pour le calcul répétitif sur des enregistrements (par exemple Cobol) ;
- les langages pour le calcul scientifique (par exemple Fortran) ;
- les langages pour la recherche (par exemple : Lisp, IPL5 (pour la programmation fonctionnelle), Algol (pour un usage général).

Le langage Cobol fut conçu dans le prolongement des travaux mécanographiques qui effectuaient des traitements simples se répétant sur une multitude d'enregistrements à l'image des cartes perforées. L'importance de traitements plus complexes et non répétitifs (mais comportant des phases itératives !) se fit sentir dans les nouveaux langages comme Fortran, Algol, Lisp, etc. qui jetèrent les bases des langages de programmation modernes.

La programmation au niveau élémentaire d'un ordinateur (langage dit d'assemblage) a d'abord été vue comme l'utilisation de commandes contingentes au matériel utilisé. On a vite compris que l'ordinateur exécutait un algorithme interprétant ses instructions et qu'il fallait relativement peu d'efforts pour qu'il

devienne universel, c'est-à-dire capable d'exécuter un large ensemble d'algorithmes, dont ceux issus des couches basses du logiciel. L'idée est donc venue d'utiliser le matériel d'une machine informatique pour exécuter le plus directement possible les langages évolués dans le but d'en faire les langages de base des nouvelles machines.

À ce niveau, il faut distinguer deux types de langages informatiques :

- les langages conversationnels (par exemple : Lisp, Basic, APL, Python, TRAC...) pour lesquels la machine répond directement, comme une calculatrice, à des commandes frappées sur un clavier ;
- les langages enregistrés (par exemple : Algol, Pascal, C, ADA, Java, Fortran, Cobol...) pour lesquels la machine traite, en plusieurs étapes successives des programmes préparés sous forme de fichiers.

Les langages conversationnels offrent généralement aussi la possibilité de grouper des commandes dans des fichiers. Certains langages conversationnels sont devenus des langages de commande pour l'exécution des travaux sur les ordinateurs (ex : Basic sur les premiers micro-ordinateurs et Python plus récemment).

Les langages enregistrés sont généralement traités en deux temps. Une première phase, appelée compilation transforme les programmes, écrits comme des textes, en des programmes écrits dans

des langages d'instructions spécifiques aux langages évolués, appelés langages intermédiaires par ex : P-Code pour Pascal, Bytecode pour Java... L'exécution de ces langages intermédiaires peut se faire de trois manières :

- par l'exécution du code intermédiaire par un interprète logiciel, généralement appelé « machine virtuelle logicielle » ;
- par la construction matérielle de machines physiques exécutant directement ces codes intermédiaires comme leurs langages d'instruction (voir le paragraphe suivant) ;
- par la traduction du code intermédiaire en instructions d'une machine cible ou dans un autre langage évolué qui devra être lui-même exécuté.

Les langages intermédiaires pour les langages enregistrés sont souvent de la forme post-fixée dans laquelle les opérateurs suivent les données (ex : A+B s'écrit AB+) qui présente la propriété de s'exécuter avec une pile, ou une file qui permet alors une exécution pipe-line (Anceau, Baille, & Schoellkopf, 1977).

Ordinateurs conversationnels

Plusieurs machines ont été construites pour exécuter directement les langages conversationnels. Compte tenu de la faible puissance de calcul demandée, il s'agit souvent de machines classiques munies d'un logiciel adéquat, appelé interprète.

• *Machines Lisp*

Plusieurs laboratoires se sont lancés dans l'étude de machines Lisp, par exemple :

- la machine CADR du MIT (1981) produite et commercialisée par Symbolics (LM-2, 3600...);
- les machines M3L et MAIA de l'Université Paul Sabatier de Toulouse ;
- le coprocesseur COLIBRI de Siemens ;

• *Machine Alvan*

Cette petite machine franco-américaine exécutait un langage voisin de celui du macro-générateur TRAC. Plusieurs machines ont été vendues à des entreprises pour des travaux de gestion.

• *Machines APL (IBM)*

- IBM 5100/5110 (portable).
- Extension microprogrammée de l'IBM 370/145.

• *Machines BASIC*

Pratiquement tous les micro-ordinateurs des années 1970 exécutaient le BASIC (*via* un interprète) à la fois comme langage de programmation et comme langage de commande.

Ordinateurs pour langages enregistrés

Plusieurs machines de puissance moyenne, ou forte, exécutaient directement le langage intermédiaire issu de la compilation d'un, ou de plusieurs, langages de programmation enregistrés. Il faut remarquer que cette approche suppose que l'ensemble du logiciel de base de ces machines doit être écrit dans le langage évolué (le système d'exploitation et le compilateur lui-même).

- *Machine Cobol*

Un prototype de machine Cobol a été développé à l'IRIA au début des années 1970 par R. J. Chevance (1974).

- *Machines Algol*

La société Burroughs Corporation a développé dans les années 1960 une ligne de machines puissantes (B5000 à B7700) (multiprocesseurs), exécutant un langage intermédiaire pour Algol / Cobol / Fortran (Organik, 1977). Ces machines ont connu une diffusion mondiale, avant de devoir s'effacer devant l'évolution des machines standards au début des années 1970.

- *Machines Pascal*

La société Western Digital a développé une série d'ordinateurs personnels appelés Pascaline (WD-9000/900/90, SB-1600) bâtis en utilisant un ensemble de circuits intégrés MCP-1600 constituant une sorte de mécano permettant

la réalisation de cœurs de processeurs génériques.

Un prototype de machine Pascal, appelée PascHLL, a été étudié à l'Université de Grenoble à la fin des années 1970 pour démontrer l'intérêt de l'exécution sur file (Schoellkopf, 1977 ; Anceau, Baille, & Schoellkopf, 1977).

- *Machine ADA*

La société Intel développa en 1981 un processeur bi-boitier appelé Intel iAPX 432 adapté à l'exécution du langage ADA et comportant des mécanismes de gestion mémoire du type de ceux existant dans les machines-système, ce qui classe ce processeur dans les deux familles. Très apprécié des universitaires, mais trop lent, il fut rapidement abandonné.

- *Machines JAVA*

Dans les années 1990, la société Sun a développé des microprocesseurs Java appelés Pico-Java (O'Connor & Tremblay, 1997) dans l'espoir du développement de stations Internet (*net-computers*). Malheureusement, l'évolution de la micro-informatique fut différente et ces microprocesseurs furent abandonnés.

Dans la première moitié des années 2000, un prototype de machine Java appelé JMQ pour Java Machine on Queue, de haute performance à exécution sur file a été étudié à l'UPMC (Palus, 2006).

Dans les années 2000, la société ARM a développé une extension du code de ses processeurs, appelée Jazelle, adaptée à l'exécution du Byte-code sur ses microprocesseurs.

Machines-système

La complexification des machines informatiques à la fin des années 1960 a permis d'effectuer un saut dans la nature des systèmes d'exploitation. De nouvelles fonctions ont été ajoutées dans de nombreux processeurs :

- la pagination a résolu le problème de la gestion automatique des mémoires principales (machine ATLAS de l'Université de Manchester en 1962) ;
- l'évolution de la vitesse des processeurs par l'utilisation d'architectures internes complexes (*pipe-line*, exécution dirigée par les données...) ;
- la transposition matérielle ou micro-programmée de fonctions systèmes (gestion de la mémoire segmentée, gestion des processus et des synchronisations, multitraitement, virtualisation, amélioration de la sécurité par des anneaux de protection), ce qui nécessite l'adjonction de nouvelles instructions très spécifiques.

La caractéristique discriminante des machines-système est l'utilisation de descripteurs utilisés par le matériel pour accéder aux différents objets manipulés

(segments (données, programmes ou bibliothèques), tâches, entrées-sorties...). Ces descripteurs sont constitués de mots ou de double-mots qui contiennent généralement :

- la localisation des objets,
- leur nature (segment mémoire, tâche, entrées-sortie...),
- leur état (dans le cas de tâches : active, en attente de synchronisation...),
- leur propriétaire,
- leur niveau de protection (anneaux),
- leur type d'accès (dans le cas de segments mémoire : lecture, lecture-écriture),
- leur disponibilité (en mémoire vive ou secondaire),
- etc.

Machine Multics

En 1964, un consortium constitué du MIT, des laboratoires BELL, et de General Electric décida d'étudier un nouveau système d'exploitation, appelé Multics (Organik, 1972) fondé sur les nouveaux concepts qui émergeaient des universités et des constructeurs. Ce système est surtout connu pour :

- la gestion segmentée de la mémoire : les segments représentent une division fonctionnelle de l'espace de travail ;
- la gestion des processus et des synchronisations (en logiciel sur Multics) ;
- son écriture en langage évolué (PL1) ;

- l'utilisation semi-matérielle d'anneaux de protection ;
- la pagination (matérielle) servit à construire une énorme mémoire virtuelle englobant tous les fichiers et les accès aux périphériques.

Pour que les temps d'exécution de ces fonctions restent raisonnables, des extensions du matériel furent réalisées, surtout pour la segmentation. L'ordinateur utilisé, un GE635 de General Electric fut modifié par le MIT et rebaptisé GE645. Il servit de modèle à cette nouvelle lignée.

La complexité de Multics incita à la création d'un système plus simple appelé Unix qui fonctionnait sur des machines standards, en particulier les Vax de Digital Equipment et les machines SUN, qui n'avaient pas les caractéristiques des machines-système. Une version d'Unix appelée Linux a été adaptée aux processeurs x86-amd64 des ordinateurs personnels avec un grand succès commercial. Toutefois cette adaptation n'utilise pas la mémoire segmentée du x86. Elle la recouvre par des segments géants pour la transformer en mémoire plate c'est-à-dire non structurée.

Les prototypes

Les ordinateurs de la fin des années 1960 et du début des années 1970 bénéficièrent d'évolutions technologiques qui leur permirent d'accroître fortement leur complexité (circuits intégrés de faible complexité, microprogrammation et mé-

moires à tores). Ceci provoqua l'arrivée d'une floraison de machines-système prototypes issues de l'université et de l'industrie :

- le GE645 du projet Multics que nous avons déjà cité ;
- le MU5, de l'université de Manchester¹ ;
- le Level 64 d'Honeywell-Bull ;
- etc.

À la fin des années 1970 de nouveaux projets furent lancés :

- le Solar 16 de Télémécanique pour le temps réel qui possédait des instructions machines spécifiques pour la synchronisation des tâches ;
- l'Intel iAPX 432 bi-circuits, (déjà cité dans les machines-langage) ;
- l'Intel 286, qui fut une première tentative d'évolution de la gamme x86 vers les machines-système.

Les machines de série

Plusieurs de ces prototypes débouchèrent sur des séries :

- le GE 645 donna naissance à l'Honeywell 8180, puis au Bull DPS 8000 Multics ;
- le MU5 donna naissance à la ligne des ICL 2900 ;

¹ « MU5 », Manchester University [URL : <http://www.cs.manchester.ac.uk/about-us/history/mu5/>].

- le Level 64 donna naissance aux lignes des Bull-GE GECOS 7, Bull DPS 7/7000² et à celle des NEC ACOS 4 ;
- Télémécanique commercialisa la ligne des Solar 16 eut un grand succès commercial.

Cas des x86

Le microprocesseur Intel 386 et ses descendants sont l'exemple le plus remarquable de l'évolution d'une lignée de machines-système. L'Intel 286 ne fut utilisé que comme un super 8086 à cause d'erreurs de définitions qui empêchèrent de l'utiliser comme une machine-système. Par contre, l'Intel 386 fut un succès qui permit de développer des systèmes d'exploitation modernes comme les Windows, OS2 et de supporter Unix, Linux et Mac-OS. Le passage à 64 bit proposé par AMD supprime de fait la segmentation. Il est devenu le standard amd64 utilisé par tous les constructeurs de ce type de circuit.

Ces processeurs ont traversé plus de 30 ans d'évolutions de l'informatique. Leur passage aux multi-cœurs et l'utilisation de techniques d'exécution très avancées leur donne une position dominante dans tous les domaines de l'informatique, des postes de travail aux superordinateurs.

² Cf. les textes sur le GCOS 7 sur le site de la Fédération des Équipes Bull, FEB-Patrimoine [URL : <http://www.feb-patrimoine.com/projet/gcos7/gcos7.htm>] et [URL : <http://www.feb-patrimoine.com/projet/gcos7/gcos7-system-architecture.htm>].

La saga des RISC

Le déplacement de fonctions du logiciel vers le matériel n'a pas eu que des avantages. D'abord, ces fonctions se sont trouvées de fait figées, tandis que les idées évoluaient rapidement. Ceci a parfois obligé des logiciels plus récents à contourner ces mécanismes pour retrouver leur liberté. Ensuite, les processeurs, surchargés par ces fonctions complexes devenaient plus lents, plus complexes, donc plus difficiles à produire et plus chers.

En 1975, John Cock, chercheur chez IBM, proposa de réaliser des machines plus simples (IBM 801), quasiment réduites à l'essentiel, appelées des RISC (pour Reduced Instruction Set Computers) (Étiemble, 1991). De telles machines se développèrent dans les années 1980. Outre un fort effet de mode, leur vitesse d'exécution s'est trouvée doublée, mais leurs programmes s'étant allongés compte tenu de la faible expressivité du code des premiers RISC compensaient cet avantage...

La mode des RISC rendit obsolète une partie des machines complexes, appelées CISC (pour Complex Instruction Set Computers) qui virent leur marché se réduire. À part les x86-amd64, les autres gammes de machines complexes disparurent ou furent réduites à des logiciels d'émulation sur PC.

Vers le milieu des années 1990 un nouveau mode d'exécution appelé *data-*

flow restraint vint toucher le monde des x86 et quelques RISC « assagis » par l'usage d'instructions complexes (comme les ARM et les IBM Power). Ce mode d'exécution, couplé à l'évolution technologique, a relancé la course à la puissance et n'a été freiné que par la chaleur dégagée. Grâce à un traducteur matériel en ligne, les codes complexes se sont trouvés être transcrits en codes spécifiques, adaptés au matériel, ce qui donne une quasi indépendance entre l'architecture interne de la machine et son code instruction qui devient donc une sorte de langage de programmation de bas niveau n'ayant plus qu'un rapport lointain avec la structure physique du processeur.

Avec ce regain en performance des CISC, les RISC perdirent leurs avantages, et beaucoup d'entre eux rejoignirent le musée des idées de l'informatique. Quelques-uns (ARM et POWER d'IBM) adoptèrent des instructions plus complexes qui leur permirent de devenir aussi efficaces que les CISC.

À titre de conclusion

L'informatique est une pure production humaine ; une myriade d'autres chemins auraient pu être pris pour traiter l'information. Comme d'autres productions humaines, elle est soumise à des effets de mode. Longtemps celle-ci a orienté ses recherches vers l'étude de processeurs qui se voulaient plus pratiques et surtout plus performants. De manière sous-jacente, une âpre lutte s'est développée entre la complexité et la simplicité, toutes deux vues comme le meilleur moyen d'atteindre l'efficacité. En fait, l'expérience a montré que celle-ci se situait souvent dans les choix intermédiaires.

L'évolution des langages et formalismes a structuré la pensée, donc la mode informatique, d'où un effet certain sur l'architecture visible des processeurs qui ont inclus ces mécanismes, ou au moins leurs soubassements, dans leurs architectures. L'histoire aurait pu continuer encore longtemps si deux facteurs n'étaient pas venus modifier son cours.

En premier lieu, la thésaurisation des logiciels est devenue une nécessité vue l'énorme volume de logiciels produits. Celle-ci n'est possible que dans un contexte de stabilité des formalismes d'écriture des programmes, ce qui a poussé fortement les processeurs à être compatibles de manière ascendante (c'est-à-dire capable d'exécuter des programmes écrits pour leurs prédécesseurs), ce qui a fini par bloquer toute évolution des machines virtuelles.

En second lieu, l'arrivée des systèmes distribués et à fort parallélisme a reporté l'attention des chercheurs vers les techniques de parallélisation et d'interconnexion des processeurs. Ceux-ci sont devenus de simples moyens technologiques, dont la nature n'a plus vraiment d'importance.

Les machines-langage ont disparu face à la puissance et au coût des machines standards fabriquées en beaucoup plus grand nombre. Les machines-système se sont réduites aux familles x86-amd64. Celles-ci occupent une forte position dans toutes les branches de l'informatique, face aux ARM qui ont presque conquis la totalité du large marché des applications embarquées et portables.

Pour résumer, la mode et les chercheurs se sont désintéressés des processeurs pour se consacrer aux systèmes parallèles en utilisant les processeurs disponibles sans trop se soucier de leurs spécificités. Pour des raisons commerciales, ces survivants se sont réduits à deux voies concurrentes :

- une famille de processeurs-système : les x86 de 32bits qui ont évolué en amd64 de 64 bits, principalement fabriqués par Intel et AMD ;
- deux familles de RISC complexifiés : la famille des ARM issus de l'informatique embarquée et celle des Power développée par IBM.

Tous ces processeurs utilisent des techniques d'exécution parallélisées

qui semblent buter sur une limite difficilement dépassable de 2,5 instructions par cycle d'horloge, laissant la technologie devenir l'unique facteur d'évolution. Seule, la famille des amd64 véhicule encore le concept de machine-système qui s'éteindra si l'évolution s'oriente vers l'autre voie, et ceci dans l'indifférence la plus totale. Les machines-langage et les machines-système constitueront alors une sorte d'antiquité de l'informatique pleine de beaux monuments...

Bibliographie

Anceau F. (2000). « De von Neumann aux super-microprocesseurs ». In Y. Michaud (éd.), *Université de tous les savoirs*. Vol 5, Paris : Odile Jacob.

Anceau F., Baille G. & Schoellkopf J.-P. (1977). « Machine informatique électronique destinée à l'exécution parallèle d'un langage post-fixé ». *Brevet n° 2 325 985*, ANVAR.

Bellec J., Chain T., Humblot D., Joly C. & Lepicard G. (1989). « De Charlie au DPS7000... 15 ans de BULL DPS7 ». Édition semi-interne Bull S.A., « Méthodes et Informatique ».

Böhm C. & Jacopini G. (1966). « Flow Diagrams, Turing Machines and Languages with Only Two Formation Rules ». *Communications of the ACM* 9 (5), pp. 366-371.

Chevance R.J. (1974). « A COBOL Machine ». *ACM Sigplan Notice*, 9/8, New York : ACM, pp. 139-144.

Étiemble D. (1991). *Architecture des processeurs RISC*. Paris : Armand Colin.

O'Connor M. & Tremblay M. (1997). « Picojava-1: The Java Virtual Machine in Hardware ». *IEEE Micro*, 17/2, pp. 45-47.

Organik E.I. (1972). *Multics Systems: An Examination of its structure*. Cambridge : MIT Press.

Organik E.I. (1973). *Computer System Organization*. Academic Press.

Palus M. (2006). « Étude et validation de l'architecture d'une machine Java de hautes performances ». Thèse de Doctorat en informatique, Université Paris 6.

Mounier-Kuhn P. (2010). *L'informatique en France de la seconde guerre mondiale au*

plan calcul, l'émergence d'une science. Paris : PUPS.

Schoellkopf J.-P. (1977). « Machine PASC-HLL : Définition d'une architecture pipeline pour une unité centrale adaptée au langage PASCAL ». Thèse de 3^e cycle en informatique, INPG, Grenoble.

Tools B.A. (1992). *ADA, The enchantress of numbers: A selection from the letters of Lord Byron's daughter and her description of the first computer*. Mill Valley, Ca. : Strawberry Press.