

1121



PASC-HLL : définition et réalisation
d'un calculateur "pipe-line" adapté
au traitement des langages évolués.

J.P. SCHOELLKOPF, G. BAILLE

SI 04

Avril 77

RESUME

La conception de la machine PASC-HLL (a Pipelined Architecture bit Slice Computer for High Level Language) présente plusieurs aspects qui sont:

- la définition d'un code machine adapté à la compilation et à l'exécution des langages de haut niveau à structure de blocs (et spécialement PASCAL)
- la définition d'un mécanisme d'exécution en "pipe-line" du code machine de type post-fixé, utilisant une file d'attente plutôt qu'une pile ;
- l'insertion d'une telle unité de traitement spécialisée dans un système d'exploitation gérant les entrées/sorties du programme exécuté (par exemple connexion de PASC-HLL à la mémoire centrale d'un IRIS 80) ;
- la réalisation à l'aide de microprocesseurs "en tranches" (bit slice) très rapides (temps de cycle = 100 nano secondes) des cinq processeurs micro-programmés qui fonctionnant en parallèle, constituent l'architecture "pipe-line" de la machine.

1.- DEFINITION D'UN CODE MACHINE ADAPTE A LA COMPILATION ET A L'EXECUTION DES LANGAGES DE HAUT NIVEAU A STRUCTURE DE BLOCS (ET SPECIALEMENT PASCAL).

Un ordinateur est défini, en grande partie, par son code machine dont les caractéristiques influent grandement sur les performances des programmes (facilité d'écriture, concision des programmes, encombrement des programmes exécutables, fiabilité software, rapidité d'exécution). Dans le cas précis qui nous concerne d'une machine langage, le concepteur a l'avantage de connaître le langage de haut niveau qu'il aura à compiler et peut choisir un code machine adapté à la compilation puis à l'exécution des programmes compilés. Les critères de choix de ce code machine sont complexes et font appel à deux notions principales qui s'avèrent être étroitement liées entre elles :

- la première notion est celle de compaction du code généré. En effet, un programme "réside" sur des supports d'information généralement coûteux (disques, bandes ...) et ne passe qu'une infime partie de sa "vie" en mémoire centrale. Il a donc tout intérêt à obtenir des programmes exécutables de plus faible encombrement possible, pour réduire les coûts de "résidence" en mémoire secondaire et les temps de transfert en mémoire centrale.

Comment peut-on compacter un programme?

Une instruction étant constituée par un code opération et des opérandes, la taille d'une instruction pourra être optimisée en réduisant la taille des opérandes et leur nombre. Sachant que les instructions les plus fréquentes référencent une (ou plusieurs) variable(s), on cherche à définir un espace d'adressage minimum pour ces variables.

Concrètement, dans le cas d'un langage à structure de blocs, des mesures sur des programmes ont montré qu'il était suffisant de pouvoir référencer jusqu'à 256 variables dans le bloc principal, et jusqu'à 64 variables et 64 paramètres dans les blocs (ou procédure) intermédiaires. D'autre part, en ce qui concerne le langage PASCAL, une imbrication statique des procédures pouvant aller jusqu'à 6 niveaux est suffisante. En conséquence, l'espace d'adressage des variables d'un programme peut être limité à 10 bits, en spécifiant un niveau lexicographique et un déplacement de la manière suivante :