

A COMPUTER ARCHITECTURE FOR PARALLEL PROCESSING*

T. N. Mudge
Dept. ECE, Univ. Michigan, Ann Arbor, MI 48109.

SUMMARY

This paper briefly presents some ideas arising out of a study to determine the requirements of a computer architecture that exploits potential parallelism in its source programs.

The architecture is organized to execute a base language which obeys the single assignment (SA) rule (each variable in a program may be assigned at most once). It has been shown in [U] that programs written in an SA language exhibit maximal parallelism during their execution, provided they have sufficient resources. Unfortunately, several obstacles need to be overcome to make SA architectures practical. These are: Implementing iterative procedures (e.g. DO loops) in the face of the SA rule, coping with possible unbounded memory requirements, and handling computed references. A solution to the first of these problems exist if iteration in the base language is represented in a recursive fashion (see [M1] and [U] for details). A solution to the second problem is given in [M1] by associating a read count with each variable (see also [M2] for a proof of the validity of this procedure). Finally, the last of these problems is resolved in [M2] using a technique similar to that proposed in [A] for data flow machines.

To take advantage of the maximal parallelism present in programs written in the base language, the architecture is organized as a high bandwidth memory with multiple instruction-set processors. A result-forwarding scheme between the instruction-set processors is employed to reduce the bandwidth requirements of the memory. Source programs are translated into the SA base language by a translator (written in the base language), then loaded and then run. All the system programs are expressed in the base language, so they may take advantage of an architecture organized to execute the SA base language efficiently. Note that the translator can be regarded as an optimizer, which is aimed at a particular type of machine organization. Further optimization is possible, if potential functional parallelism is exploited.

- [A] Ackerman, W. B., "A Structure Processing Facility for Data Flow Computers", Proc. 1978 Internat. Conf. On Parallel Processing, Aug. 1978.
- [M2] Mudge, T. N., "A Data Driven Computer Architecture", Proc. 1978 Conf. on Information Sciences and Systems, pp. 365-370, March 1978.
- [M2] Mudge, T. N., and W. T. Liu, "Memory Management Procedures for a Single Assignment Programming Language", Systems Eng. Lab. Report, Dept. ECE, Univ. Michigan, (to appear).
- [U] Urschler, G., "The Transformation of Flow Diagrams into Maximally Parallel Form", Sagamore Comp. Conf. on Parallel Processing, pp. 38-46, March 1973.

*This work was supported by the National Science Foundation under grant ENG-78-05779