Def Sci J, Vol 35, No 3, July 1985, pp 375–382

Parallel Computers in Signal Processing

NARSINGH DEO

Indian Institute of Science, Bangalore-560012

Abstract. Signal processing often requires a great deal of raw computing power for which it is important to take a look at parallel computers. The paper reviews various types of parallel computer architectures from the viewpoint of signal and image processing.

1. Historical Review

Speed is one of the most important characteristic of a computer. Inspite of the impressive gains made in enhancing the speed of the computer during the past three decades, the quest for computers with higher speed goes on relentlessly. This is because of the need for a higher speed computer for solving many compute-bound problems of today.

In this context it is interesting to compare UNIVAC-I computer with CRAY-1. The UNIVAC-I was the first commercially produced computer, and its first unit was delivered to the U.S. Census Bureau in 1951. Twenty-five years later, in 1976 the first CRAY-1 computer was delivered (also to an agency of the U.S. government). The CRAY-1 is the fastest commercially manufactured computer today which is 100,000 times faster than the UNIVAC-I.

There are two points to note : First, the demand for a higher-speed computer is as great in the post CRAY-1 period as it was after the UNIVAC-I. There are many applications where CRAY-1 is not fast enough. Machines of two to three orders of faster magnitude are required. These applications include numerical weather forecasting, analysis of seismic data, A.I. applications etc. In fact, it is expected that the fifth generation knowledge-base machines will have a number-crunching capability of at least 1000 megaflops—about two orders of magnitude more powerful than the CRAY-1[1].

The second point is that the first CRAY-1 was delivered seven years ago, and it is still the fastest commercial machine. Its clock rate of 12.5 nsec remains unbeaten. This is because it is extremely difficult to push the device technology toward higher speed any further. The momentum of earlier three decades of building faster and faster computer components cannot be maintained. It has run into important limitations imposed by the laws of physics, viz. speed of light and dissipation of heat.

It appears that any substantial gain in computational speed has to come through architectural innovations not through the device technology[2]. In fact, a closer examination of UNIVAC-I and CRAY-1 computers shows that the improvement in the device technology accounts for speed gain by a factor of 1000. During the twenty five years between 1951 and 1976, a number of evolutionary improvements in the computer architecture were also made. They account for the rest of the speed gain -- by about a factor of 100 - from UNIVAC-I to CRAY-1.

Although during the past three decades there have been small evolutionary improvements in computer architecture, there was no revolutionary change in the device technology. Computer architecture has largely remained static. We have continued to use von Neumann architecture, with a single computer incorporating a single processor, and a linearly organized memory, performing sequential computations.

1.1 *Architectural Revolution*

Now the computer architecture is in a revolutionary phase. It is about to take a quantum jump and move away from the classic von Neumann design. There are three primary factors responsible for this change :

(i) The ever-increasing demand for faster, more-reliable and less expensive main-frame computers,

(ii) The inability of the component technology to maintain its earlier record of speed gain, and

(iii) The maturing of the VLSI technology.

The fifth generation computers are going to be parallel rather than sequential computers. In contrast to the low-level, very selective parallelism that has increasingly been incorporated into the machines of the second, third and fourth generations, the fifth-generation machines will have massive parallelism, and this implies a qualitative difference π a great leap forward.

1.2 *Early Failures*

The essential idea of putting 10, 20, or 100 processors in a box and interconnecting them for obtaining a faster computer is not new. In fact, the idea of parallel computers is as old as the computer itself. But because of some very difficult problems in implementation, general-purpose parallel computers existed only in theory with one or two exceptions. The best-known and the most ambitious project undertaken to build a parallel computer was ILLIAC IV, but the machine was a failure[3]. It failed not because the ideas were flawed, but because the technology of the late sixties and early seventies was inadequate to implement the parallel architecture. This is why ILLIAC IV was delivered long after its due date and became operational only in 1975—9 years after its construction was started. Its cost exceeded by a factor of four over the amount of the original contract; instead of 256 processing elements, it had only 64; instead of the originally specified clock rate of 40 nsecs, it could realize a clock rate of only 80 nsecs, and the maximum throughput rate, which originally was supposed to be one Gflops, could not even reach 50 Mflops. Thus, in spite of the massive support from the U.S. government, the machine fell far short of its target—primarily because the idea was ahead of its time.

A few other prototypes of general purpose parallel computers have been built since ILLIAC IV. The best known of these are cmmp and cm* at Carnegie-Mellon University, and the DAP machine by ICL[4,5]. None of these can be termed a commercial success. The only truly general-purpose, stand-alone, commercially available parallel computer tody, as far as I know, is the HEP (heterogeneous element processor) built by Denelcor Corporation of Denver, Colorado, and delivered last year. It allows a maximum of eight processors to be connected together.

2. Special-purpose Computers

Attempts at building special-purpose parallel computers have been somewhat more successful. A good example, is PEPE, whish was built in 1971 for tracking a number of missile positions. The parallelism was of a very elementary kind[6]. Each object in the sky was to be tracked by an independent processing element. A more recent example of a special-purpose parallel computer is MPP (massively parallel processor). It is a 16-processor SIMD computer, built by Goodyear Aerospace to perform real-time image processing for NASA. It can perform almost 300 million multiplications of 32-bit floating point numbers per second[7,8].

If the problem has a well-defined structure, such as picture processing, then it is easy to build a special-purpose parallel computer to solve that problem. In fact, such high-performance computers, are relatively easy and economical to build. The following are four examples of special-purpose parallel computers built for real-time image processing in Japan[9].

(a) IPAS : Interactive Image Processing and Analysis System of Nippon Electric Co., dedicated to multi-spectral data analysis.

(b) PPP : Parallel Pattern Processor of Toshiba. It consists of a number of interconnected special processors, each dedicated to unique tasks (convolution, FFT, Affine Transformation, etc.)

(c) IP : Image Processor of Hitachi, which consists of an array (4 by 4) of identical processing elements whose tasks are assigned by a central control system.

(d) POPS : Poly-Processor System of Electro-technical laboratory. A multi-microprocessor structure connected through a single common bus and a 3-port shared memory.

A number of other parallel computers for image processing are under development and construction at various places such as PUMPS at Purdue University, ZMOB at University of Maryland, Cytocomputer at Environmental Research Institute of

Michigan, GOP at Linkoping University of Sweden, and FLIP at Karlsruhe, West Germany.

3. Types of Parallel Computers

Unlike the case of the uniprocessor system, for which standard models of computation exist (such as RAM and RASP) there are many contending models for parallel computation. A parallel computer means different things to different people. The number of different architectures that have been proposed for parallel machines must literally be in hundreds[2,10-13].

Several attempts have been made at classification of parallel computers, but there are paper machines that defy all of the classifications. The simplest and the best-known classification is by Flynn[5,11,13]. He classified machines into four types by the number of data and instruction streams. Thus, the standard uniprocessor is a single-instruction stream, single-data stream (SISD) machine. An array processor, such as ILLIAC IV, where a set of identical, processors perform the same operation in a lockstep fashion on different data, is called a single-instruction stream, multiple-data stream (SIMD) machine. Although the processing elements execute each statement in parallel, individual units may be programmed to ignore any particular instruction. This ability to mask out processing elements allows synchronization to be maintained through various paths of control structures, such as in the clauses of an if-*then-else* statement. In the literature, often the terms SIMD and array processors are used synonymously However, the term array processors has also been used for *algorithmic array processor* or *peripheral arrayprocessors*[12], such as Datawest 400, or Floating Point System AP120B, which are not SIMD machines. SIMD models can further be classified on the basis of their interconnection, whether or not they have a shared memory, and if the number of processors is fixed or unbounded[2,10,13].

A third type of computer is the multiple-instruction stream single-data stream (MISD) machine. A pipeline architecture falls in this category.

Finally, a multiple-processor system consisting of a number of fully programmable, independent processors, each capable of executing an instruction different from others, is called an MIMD (multiple-instruction stream, multiple-data stream) computer. An MIMD machine can be further classified by the degree of coupling—tightly coupled or loosely coupled, or by memory-processor communication scheme—crossbar switch, a bus, or a mixture, whether or not there is a master-slave relationship among the processors, and so forth.

Flynn's classification is crude, and several finer classification schemes for parallel computers have subsequently been proposed by Enslow[11], Higbie, Hobbs *et al.*, Murtha and Beadles, Hockney[5], and others.

3.1 *General vs. Special*

From the users's view point it makes more sense to classify parallel computers according to the flexibility of their use i.e. how general purpose they are, and also how

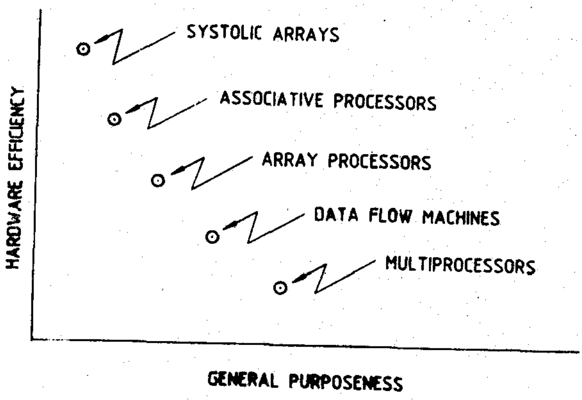

Figure 1. Classification of parallel computers.

cost-effective the hardware utilization is. Using these two attributes as othogonal axes, various types of parallel computer architecture can be classified as shown in Fig. 1.

3.2 *Systolic Arrays*

A systolic array is an array of synchronized, special-purpose, rudimentary processors with a fixed interconnection network. The function of the processors and the type of interconnection scheme depend upon the problem being solved. A processor typically has 2, 4, or 6 neighbours, as shown in Fig. 2.

Systolic arrays have been around for a long time as cellular processors, but they were primarily of theoretical interest. They have been made popular more recently by Kung[14] and some prototypes are being fabricated into special-purpose VLSI chips,

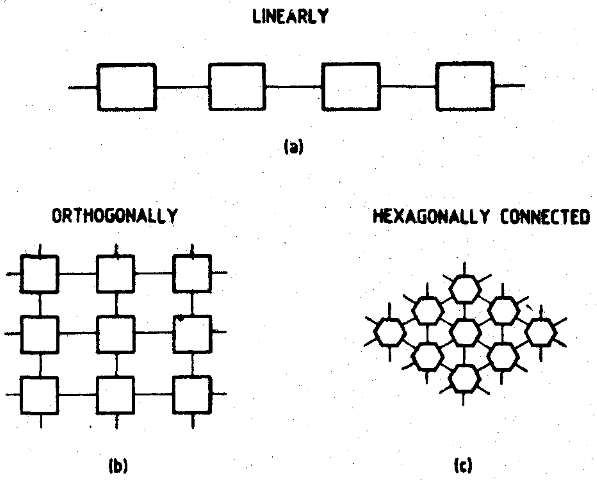

Figure 2. Systolic array configurations.

for pattern matching 1 and multi-dimensional convolutions, finite impulse response (FIR) filters, and discrete Fourier transforms. Since these tasks are computationally demanding, but at the same time the computation is highly regular, they lend themselves nicely to systolic array implementations.

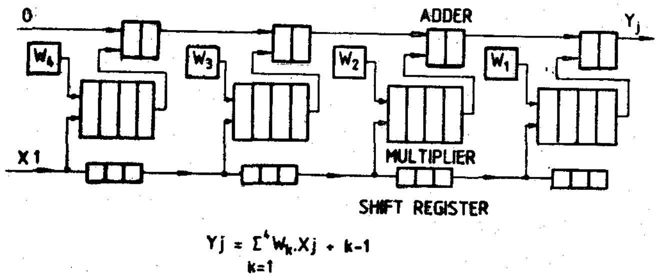

Figure 3. One dimensional convolution array.

The systolic array configurations for multiplication of a vector by a one-dimensional convolution and a band matrix are shown in Fig. 3 and Fig. 4 respectively. A careful look at the first figure will show that by converting an n-point discrete Fourier transform into a matrix-vector multiplication form, the n-point DFT can be computed in 0(n) units of time, rather than 0(n log n) as required by the sequential DFT algorithm. The number of elementary processing elements required to achieve the log n speedup is n. The TRW's Advanced Processor Technical Laboratory in San Jose, has just fabricated systolic processor for signal/image processing, 2-D convolution and resampling FFT operations. It is systolic pipeline driven by a VAX 11/780 host.

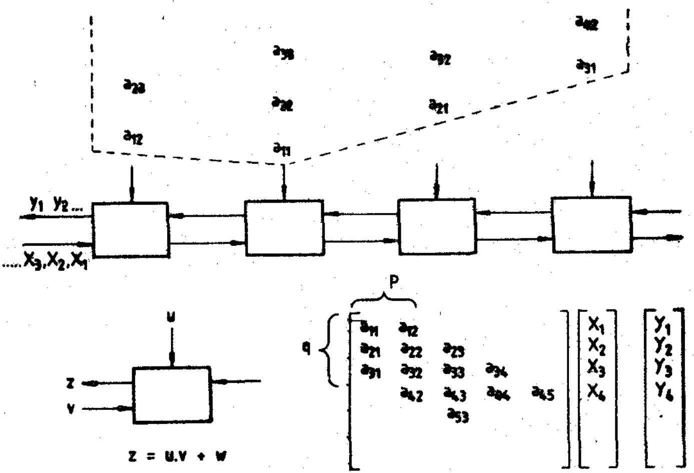

Figure 4. Multiplication of a vector by a band-matrix with $p = 2$ and $q = 3$.

4. Associative Processors

An associative processor is also a special-purpose computer with two distinguishing features (a) an associative memory from which data can be retrieved or manipulated associatively, i.e. by content instead of address, and (b) operations can be performed on many sets of data with a single instruction i.e. in parallel. Associative processors are in fact SIMD system (similar to the ILLIAC IV) except that they access their memories through association instead of vector addressing, and are thus arranged differently.

The best-known example of an Associative Processor is STARAN, built by Goodyear Aerospace Co., for NASA. It consists of 32 arrays with 256 processing element in each array, updating a multidimensional access memory. It has been used by NASA ın large area crop inventory experiments (LACIE) and in implementing a digital cartographic system for the Defence Mapping Agency.

Since array processors and multiprocessors have already been discussed briefly in the previous section as SIMD and MIMD machines, respectively, we will now discuss the data flow computer.

5. Data Flow Computers

This is the most controversial and novel architecture, and differs from the current day uniprocessor von Neumann architecture. The execution of a compution in a data flow model is data-driven. An instruction is executed as soon as the required operands become available. Thus, there is no instructions counter in a data flow computer. This model does not have a global updatable memory. The machine deals only with values and not with names (memory addresses) that contain the values. The programme in such a machine is represented as a directed graph, in which the nodes are functions and the arcs are the dependencies between functions. In theory, if sufficient processing elements are available then a data flow machine can exploit maximum parallelism inherent in the algorithm, and at a microscopic level.

In theory, this is a beautiful concept, but in practice there are a number of serious problems that must be solved before one can say that data flow model leads to a viable machine. Currently several groups are working to demonstrate that the data flow concept does work in practice. The best known among these are Jack Dennis's and Arvind's groups at MIT.

6. Conclusion

Signal/image processing involves computation-intensive tasks such as multi-dimensional convolution, logical filtering, histogram computation, affine transform, fast Fousier transform, maximum likelyhood classification etc. Therefore, parallel processing has much to offer. The processing can be performed on a wide variety of parallel computers—from truly general-purpose multiprocessor systems to specially tailored hardware in the form of systolic arrays. Each has its advantages and disadvantages. The special-purpose machine, although most efficient in utilizing the hardware, cannot be used for any other purpose. On the other hand, the most general purpose architecture is the least efficient in hardware utilization, but it can also be used for other jobs.

Both approaches have been employed in signal/image processing applications, depending on the cost speed-requirement, and the volume of the processing to be done. This special vs. general-purpose computer acquisition is a crucial decision and must be made at the outset taking various factors into account.

References

1. Treleaven, P. C., & Lima, I. G., *Computer*, **15** (1982), 79–88.
2. Wallach, Y., 'Alternating Sequential/Parallel Processing', Lecture Notes in Computer Science 127, (Springer-Varlag, Berlin), 1982.
3. Barnes, G. H., *et al.*, *IEEE Trans. Computers*, **C-17** (1968), 746–757.
4. Gehringer, E. F., Jones, A. K. & Segal, Z. Z., *Computer*, **15** (1982), 40–53.
5. Hockney, R. W. & Jesshope, C. R., 'Parallel Computers : Architecture, Programming and Algorithms' (Adam Hilger, Bristol, England), 1981.
6. Evensen & Troy, 'Introduction to the Architecture of a 288-element PEPE, Proc '1973 Sagamore Computer Conference on Parallel Processing, pp. 162–169.
7. Batcher, K. E., *IEEE Trans. Computers*, **C-29** (1980), 836–840
8. Schaefer, D. H. & Fischer, J. R., *IEEE Spectrum*, March (1982), 32–37.
9. Kidode, M., *Computer*, **16** (1983), 68–80.
10, Andrews, G. R. & Schneider, F. B., *ACM Computing Surveys*, **15** (1983), 3–44.
11. Enslow, P. H., Jr., *Computing Surveys*, **9** (1977), 103–129.
12. Karplus, W., (ed) 'Peripheral Array Processors', (APCON, Society for Computer Simulation, La Jolla, California) October 1982.
13. Kuck, D. J., *'Computing Surveys'*. **9** (1977), 29–59.
14. Kung, H. T., *Computer*, **15** (1982), 37-46.
15. Kung, H. T., Ruane, L. M. & Yen, D. W. L., 'A Two-Level Pipelined Systolic Array for Multi-Dimensional Convolution', Carnegie Mellon University, Computer Science Technical Report, November 1982.
16. Mead, C. & Conway, L., 'Introduction to VISI Systems', (Addison-Wesley, Reading, MA), 1980.
17. Stone, H. S., 'Parallel Computers in Introduction to Computer Architecture', (Science Research Associates, Chicago, II) Chapter 8, 1980.
18. Arvind & Gostelow, K. R., *Proc. IFIP Congress*. **77** (1977), 849–853.
19. Dennis, J. B., *Computer*, **13** (1980), 48–56.