IBM CONFIDENTIAL

Date: May 1, 1967

From (location or U.S. mail address): Advanced Computing Systems
Menlo Park

Dept. & Bldg.: 990/031

Telephone Ext.:

IBM

Subject: A Description of the ILLIAC IV

Reference:

To: File

Enclosed is a factual account of the ILLIAC IV computer, based on information currently available.

I am indebted to many friends within IBM and the University of Illinois for information in this interim report, particularly to Dr. Gene M. Amdahl. Comprehensive papers on the subject will be published in the next few months by the University of Illinois scientists.

T. C. Chen

Tie Chi Chen

TCC:mas

cc: SADL

1. The ILLIAC IV

The ILLIAC is a very large scale (three million circuits) special purpose computer to be supplied to the University of Illinois by Burroughs Corporation and Texas Instruments, Inc. Dr. Daniel Slotnick's group at the University of Illinois is responsible for the inception, negotiation, and development of this machine and is the prime contractor to ARPA which funds the project.

There is now a commitment from Burroughs and TI to produce a system of 64 processor elements (a quadrant of the final machine), using a "hybrid LSI" technology at a reported cost of $7.6 million to be delivered "in 1969". The full machine with 256 processor elements is estimated at $14-$15 million with delivery perhaps before 1972.

Intended as a computer system for problems possessing highly parallel internal structures, Dr. Slotnick makes no pretense of claiming general purpose capability. He is confident, however, that there are many superscale problems, ciritcally important to the society, with highly parallel features; and that optimum utilization of hardware potential can be achieved by manual programming using expert programmers.

Therefore, the following should be borne in mind:

The ILLIAC IV is NOT a general purpose computer (in fact, it is not even a general "parallel-purpose" computer because of its unique design features).

The ILLIAC IV requires hand-honing of programs to avoid alarmingly inefficient use of hardware.

The ILLIAC IV has a minimum systems-programming support (only an assembler).

2. Prehistory: Dr. Slotnick and the SOLOMON Design

While Dr. Slotnick was still with IBM, he and Dr. John Cocke became interested in the parallel computing possibilities. In 1958 they jointly published a brief Research Note (See Reference 1) on evaluating polynomials using parallel hardware. Their study of a parallel hardware design was reported in a document by Manfred Kochen (Reference 2).

After joining Westinghouse, Baltimore, in 1961, Dr. Slotnick began in earnest the design of a collection of computing elements linked in a square array, for the solution of partial differential equations. This idea is traceable, I believe, to Laplace, who thought of employing a rectangular array of clerks, each passing information to his four neighbors, and averaging the numbers received from its neighbors, to approximate the solution of the Laplace equation.

Dr. Slotnick's design consisted of a square array, with 32 processor elements (PE's) on each side. There would be 1024 PE's in all. He called the system SOLOMON, for 1024 approximates the number of Solomon's wives (References 3 and 4).

The SOLOMON contained a number of interesting new features and received wide publicity and academic support. Dr. Slotnick was, however, unable to get financial backing for the actual implementation of the full machine. A 128-PE version was delivered to Rome AFB who sponsored the technical study.

There were a number of technical reasons why the SOLOMON was not a success. To name a few:

A. It represented excessive hardware, both in circuit and memory, than existent technology could bear.

B. There was a packaging problem. Packing memory cells and circuits together to form PE's is not easy to accomplish, at least for the conventional type memories.

C. Each of the SOLOMON PE's was to be a 32-bit fixed-point serial processor. Many potential scientific users had come to demand floating-point arithmetic, and a longer word length. Built-in floating-point would increase hardware, and to simulate floating point efficiently by the synchronous fixed-point serial hardware would be very difficult to do in parallel.

D. The really big, truly fixed-point problems do not possess the square-array topography.

E. Even for parallel problems there is still the "exception-handling" problem. What seems to be trivial fixup for conventional computing may mean heavy loss of efficiency here.

F. In general, precipitous drop of performance can easily result from bad problems and/or bad programming. Heavy rethinking is required even for good programmers on ideally parallel problems.

G. The lack of plans for a compiler (say FORTRAN).

Many of the above criticisms tend to fade with a brand-new start, based on new knowhow and new technology. A peak performance of one BIPS (billion instructions per second) in floating-point performance can now be hoped to be not only reached but harnessed through good programming on well-suited problems at Illinois.

3. Brief History of the ILLIAC IV

In 1965, Dr. Slotnick joined the University of Illinois and he and his group studied parallel computing applications. In February, 1966, an RFP was sent by the University of Illinois to 17 manufacturers for three study contract awards of $50,000 each. Seven vendors responded favorably and, in July, three of them (Burroughs, RCA, and UNIVAC) were selected for the award.

In January 1967, Burroughs (now allied with Texas Instruments) was chosen for the fabrication and assembly of a pilot system with 64 processor elements, with delivery expected in 1969. The funding is from ARPA, with Rome AFB exercising the detailed supervision and negotiations.

The ILLIAC IV is to be a superscale computer system with 256 processor elements (PE's) each capable of executing 4 million floating point instructions per second. The collection, therefore, can reach 1 BIPS (billion instructions per second). The committed version is a quadrant, one-quarter of the complete system, that is a collection of 64 processor elements with corresponding down-scaling of other hardware. The performance maximum for the quadrant is 256 MIPS.

The hardware count is 10-12 thousand circuits per PE, or about 3 million circuits in all. The quadrant due in 1969 should have about 750 thousand circuits.

Dr. Slotnick spoke of an orderly transition from the initial "hybrid LSI" to the full LSI within the duration of the project. Thus, the quadrant for first delivery is probably based entirely on hybrid LSI circuits. The circuit cycle time is to be 40 nanoseconds. The memory is to be of film type, cycle time 240-250 nanoseconds. The hybrid LSI and full LSI packages are to be "mechanically compatible".

4. The Overall System of 3 Million Circuits

We shall describe the entire anticipated system, with the understanding that only one quadrant has been committed, and the remaining three quadrants have an unfixed schedule, probably a different technology, a matching problem in hardware characteristics, and will probably represent an extension of the current commitment by the funding agency.

A sketch of the full system is given in Fig. 1. It resembles a balloon, with four quadrants surrounded by an I/O Bus, the latter connected to a disk file which in turn is connected to the I/O Processor.

Accepting the low figure of 10K circuits per PE, each quadrant of 64 PE's means at least 640K circuits, and the full system has 2560K circuits, counting PE requirements alone. Each quadrant, in addition, has a control unit with 30-40K circuits, the control lines, I/O Bus, all require circuits. The grand total for the entire system should take 3 million circuits.

Each PE has 2K memory words; this leads to 128K words per quadrant, and 512K words for the entire system, again only counting PE requirements. Since each word has 64 bits, the collection has at least 32,768K bits, or 33.55 million bits.

The IBM 7090 has roughly the circuit count as a PE, and the 32K word memory is roughly 1 million bits. The hardware is, therefore, like 256 7090's in circuitry, and 33 7090's in memory. The peak performance of 1 BIPS is like 5,000 7090's. Performance per "7090 equivalent circuit" is roughly the same as the performance per PE, namely 4 MIPS, which is 20 times that of the 7090.

It is too early to expect complete details of the ILLIAC IV. Apparently a contract has yet to be submitted to Illinois by Burroughs, and complete accord on details has not been reached by all parties concerned.

Papers on the ILLIAC IV are in preparation at Illinois and will appear in a few months. The currently available account appears in a 1967 SJCC Proceedings article (Reference 5).

ILLIAC IV OVERALL SYSTEM

Funding: ARPA for pilot system

Prime Contractor: University of Illinois (Dr. D. L. Slotnick)

Subcontractor: Burroughs Corporation
Texas Instruments, Inc. (subcontractor to Burroughs)

Total System has 3 million circuits

1 I/O processor (S/360 Mod 44-50 class)

1 Disk file (ten billion bits, each disk with 384 million bits per second rate)

1 I/O Bus (width 4096 bits or 64 words)

4 Quadrants each with 700K circuits, each perhaps with backup memory

Promised Pilot System to be delivered in 1969:

1 QUAD (with no backup memory)

1 I/O Bus (reduced width?)

Most circuits by TI (hybrid LSI)

Film memory by Burroughs

Hardware associated with 1969 System

1 I/O Processor. To be selected.

1 Disk File. To be selected.

1 Backup Memory?

5. The Quadrant: 64 PE's and a Control Unit

The ILLIAC IV has 4 quadrants, "QUAD's", each with one control unit, 64 processor elements, a common data bus, control lines, and (eventually) a backup memory (BUM) mentioned in the previous section. A diagram of a QUAD is shown in Fig. 2.

The control unit apparently has not been completely designed; it is expected to have 30-40K circuits. It will have a 64-word instruction buffer. With no genuine memory otherwise, it gets instructions from the PE memory, from a special bus capable of transporting 8 words (from 8 consecutive PE's) at a time.

The main purpose of the control is to send control signals to the 64 PE's under its command, and sometimes to send data in a "broadcast". Each control unit can handle a different instruction stream; each PE in the same QUAD handles the same task with little flexibility beyond

(a) conditional nonexecute based on mode selection,
(b) local indexing.

With the ILLIAC IV at any given time, there can be

(a) 4 separate instruction streams each by one QUAD control unit, or
(b) 2 separate instruction streams, each over 2 QUAD's, or
(c) 1 instruction stream over all 4 QUAD's.

It is not allowed to have one instruction stream over 2 QUAD's, and at the same time 2 streams, one on each of the remaining QUAD's.

It seems possible for the control to sample numbers from the PE's to decide what to do next. Since, to supply instructions, 8 words from 8 adjacent PE's can be transported into the control unit at one time, this mechanism can be used for the sampling purpose.

There is a one word wide (64 bits) common data bus shared by the control unit with all PE's. This may be the vehicle for broadcast and sampling can certainly be done here as well.

The control unit, probably through the IO Processor, controls I/O flow for the QUAD, but data transmission is with the 64 PE's directly. There is no apparent data path from I/O Bus to QUAD control.

QUAD Summary

4 QUAD's in full ILLIAC IV
1 QUAD in pilot system (1969 delivery)

Each Quad has
1 QUAD control unit (30-40K circuits)
1 common data bus (width 64 bits)
1 instruction supply bus (width may be 8 words or 512 bits)
1 set of control lines into the 64 PE's
1 backup memory (BUM) (late delivery?)
64 PE's (each with 10-12K circuits)
Interface with I/O bus

6. Processor Element Characteristics

The PE is the basic computing element in the ILLIAC IV system. There are 64 PE's in a QUAD, under the same control unit. In the complete system there would be 256 PE's under 4 different control units.

Each PE has its own memory: 2048 words of film with a cycle time of 1/4 microsecond, presumably to be supplied by Burroughs. Most of the memory is for data, although part of it is used to house instructions for the benefit of the QUAD control unit.

The arithmetic ability of a PE is high. With circuit speed 6 times that of memory (40 nanoseconds vs. 240-250 nanoseconds), and with the decoding overlap problem nonexistent in the ILLIAC design, each PE can execute about 4 million instructions per second - roughly 4 to 5 times as fast as the S/360 Model 75 and 40-60% the speed of the Model 91. The collection of 64 PE's would give a maximum performance of 256 MIPS, and in the full machine of 4 QUAD's, 1024 MIPS. The maximum performance is, however, not easy to realize, being highly dependent on the nature of the problem, the chosen problemsolving technique, and detail programming.

The circuitry in a PE is "hybrid LSI" at least for the first delivery. Although full LSI is the aim, Dr. Slotnick spoke of a transition from one technology to the other within the building period of the ILLIAC IV. This probably means the hardware QUAD for 1969 delivery will consist mostly of the hybrid variety. There will be problems matching two kinds of technology together; especially within the same QUAD where time synchronization is of the essence. The hybrid LSI packaging is said to be "mechanically compatible" with full LSI. The PE circuits, indeed the entire PE's except the memory, are to be supplied by TI.

There are 10-12K circuits in each PE. According to Dr. Slotnick, the price will soon be $10K each.

A PE has a self-contained floating point arithmetic unit. In order to obtain the high synchronism among PE's in the same QUAD (every active PE has to do the same instruction at the same time), each instruction should have a fixed timing. For floating point arithmetic, this calls for a fixed time for shifting of fractions. In the ILLIAC IV PE there is a one-cycle shifter capable of shifts up to 48 positions. The rest of the PE arithmetic hardware consists mainly of three 64-bit registers (A,B,S), high speed carry-save adders, and an 8-bit wide logical unit which also handles exponents of floating point numbers.

Each PE has an index register (width 16 bits) to afford a degree of flexibility in accessing operands from memory. Actually, 12 bits would suffice; the extra 4 bits are for compatibility and future expansion. Each operation involving memory can thus use an effective address which is the given address plus the current contents of the index register. To make the process meaningful, the effective address must refer to an address within the 2048-word local memory. An address outside of the 2K addressing space would call for special, probably non-parallel, measures.

A mode register (8 bits) is another feature of the PE. It allows the partitioning of the PE's into 256 subsets, and an instruction may specify which combination of subsets is to be active. Full specification may take 256 bits, and is probably not possible; instead the mode register may actually be two sets of 4 bits each.

In addition, each PE has data paths connecting to the outside. These include word-wide linkages to the four (E,W,S,N) neighbors, with the common data bus shared by all PE's and the control unit, with the I/O bus, and directly with the outside world. The use of backup memory may call for other connections.

The control of the PE's is supplied by control lines from the control unit.

The PE data format will be hexadecimal floating point, like System/360. This is almost a necessity to permit 32-bit (halfword) floating point quantities. Fixed point arithmetic will be based on the fraction field of floating point numbers. There also may be byte-processing based on the exponent-handling hardware, but employing all bytes in a word.

The length of the fraction in full word floating point format is 48 bits to conserve hardware circuitry. The full-word (64-bits) floating point word thus has room for a 16-bit exponent.

Burroughs has a tradition of using pushdown accumulators and "syllabic" instructions. The PE design, however, is described as "standard AC-MQ" with A for AC, B for MQ, and S for temporary storage. There will probably be a rich set of inter-accumulator instructions than standard machines. The fact that instructions will be pre-decoded by the QUAD control unit already will call for drastic revamping of any existent instruction design. Thus, the following features must be installed:

Conditional execution based on mode assignment
Neighbor communications
Broadcasting from control to all PE's in QUAD
Mode reassignment, etc.

and an instruction may require quite a few bits.

The design calls for

Load, stores	240-250 microseconds (memory speed)
Floating add	240 microseconds maximum
Floating multiply	400 microseconds maximum

With a very small set of accumulators (3) and limited freedom to use them, corresponding to each arithmetic instruction there is roughly a memory operation, and even if shorter instructions may exist (say 32 bit floating add), the average is still bound by memory cycle time which is 4 million accesses per second. On a conventional design, the memory access time of perhaps 0.125 microseconds is <u>added</u> to the arithmetic time, and the average rate is something like 2-3 MIPS.

A design based on overlap can overlap memory operations with execution. Then the execution time is not the sum of memory access and arithmetic time, but more or less the maximum between the memory cycle time and arithmetic time. This means roughly that the PE with overlap can execute 7090 type instructions at 4 MIPS.

It is to be noted that 7090 type instructions expanded to accomodate 3 registers are still not as powerful as S/360 instructions, or instructions based on multi-accumulator designs.

An interesting feature to improve performance is dual arithmetic, where each word holds a pair of (short format) floating point numbers. Two such pairs interact to give a pair of results. By altering the long-word arithmetic hardware somewhat, dual arithmetic can be comparable in speed with long word arithmetic, and thus the number-crunching rate is doubled when the algorithm permits this manner of processing. (Dual arithmetic was planned for the IBM 7034 computer, which was never built, and existed in fixed point form for SAGE. ILLIAC IV is probably the first announced machine with the dual floating-point feature, however.)

Although one speaks of the four (E, W, S, N) neighbors, the eastmost PE still has an east neighbor, which is the westmost PE one level below. Corresponding situations occur at all boundaries. It is easiest to visualize the PE's to be arranged on a helix with cross-linkages; and the helix is bent into a doughnut.

The helix is always 8 units in circumference. The length of the helix before bending is 64, 128 or 256 dependent on whether the system is to be employed in the 4 instruction stream mode, the 2-instruction stream mode, or the "united mode" with one instruction stream. The PE-linkages lead to topographies of rectangular arrays: 8 x 8, 8 x 16, and 8 x 32, and there is no 16 x 16 square array provision.

The reason for the helical-doughnut linkage was based on the new conviction that the ILLIAC IV should be used most of the time as a vector machine along, say, the EW direction, with short cut (SN) paths, but not as an array machine *per se*. The writer shares this view, and feels that a vector machine of 256 PE is too long and the system will probably be used usually as 4 separate smaller (64-elements) vector machines.

PE SUMMARY

64 PE's in a QUAD, 256 PE's in a complete system
Each PE has 10-12K circuits at 40 nanoseconds/cycle
(TI: hybrid LSI for 1969)

2K Memory words (1/4 microsecond cycle time)(1word=64 bits)
3 Registers (64 bits wide) A, B, S; serving as AC, MQ, backup
1 Index register (16 bits)
1 Mode register (8 bits)
Hardware to do highspeed floating point arithmetic & indexing
data links to four neighbors
data link with common data bus
data link to I/O bus
data link directly to I/O devices
control lines from QUAD control unit.
instruction supply to QUAD control unit

Philosophy of design: Extended AC, MQ.

Formats: 32 bit floating point like S/360, with 8 bit hex sign-exponent and 24 bit fraction.
64 bit floating point: 48 bit fraction. 16 bit exponent?
fixed point: based on floating point fractions
byte: 8 bit logic. 1 word has 8 bytes.

Performance rating: 4 million instructions per second if memory access is overlapped with computing.
2 - 3 million instructions per second if no overlap.
The above rates are <u>doubled</u> if dual arithmetic is applicable (such as processing two hemispheres in parallel in weather calculations).

Instruction power: somewhat better than 7090
less than S/360
less than multi-accumulator machine instructions.

7. I/O and Back Up Memory

Aside from the four quadrants, the full system has an I/O processor, a disk file, and an I/O Bus. Back-up memory to supply 16 billion bits per second is also being discussed.

The requirement on the I/O processor apparently is slight. It is often said that a Burroughs 6500, an IBM Model 44-50 or an SDS Sigma 7 will do. The I/O processor should handle most of the standard I/O where volume input/output is not required to control the disk file directly, also to deal with the control units of the four quadrants. It further has a word -wide connection with the I/O Bus, and the control of back-up memories, if any.

The I/O processor is expected to continuously monitor the entire system to detect unusual events, and to handle all compiling, and tasks related to an operation system.

The disk file has not been chosen. The requirement is 10 billion bits (0.16 billion words), with a transport rate of 400 to 1000 million bits per second (6-16 million words per second), expected to be achieved using multi-head disks. The access time is not important to Illinois.

The I/O Bus need not be more powerful than the expected maximum traffic requirements. Some requirements are listed below:

a. To saturate PE memory bandwidth,
   1 billion words per second (64 billion bits per second) calling for 40-word wide bus at 40 ns rate.

b. To saturate 64 PE's in one QUAD,
   250 million words per second(16 billion bits per second) calling for 10-word wide bus at 40 ns.

c. To deal with disk file, 16 million words per second (1 billion bits/sec.) calling for 1-word wide bus at 40 ns. (this give 1.6 billion bits/sec.)

It seems, for the 1969 pilot QUAD system at least, a one-word wide I/O bus at 40 ns/cycle is adequate. Wider bus would be needed if there are back-up memories of high bandwidth.

The plans for the ILLIAC IV call for a 4096 bit (64 words) wide bus to operate at 1 billion word per second memory saturation rate. This can be done with a circuit cycle time of 64ns, rather than 40ns as in the PE's.

There is a great deal of talk about a back-up memory (BUM). The desire is to back up each quadrant by a 512K-1024K word memory, (32-64 million bits) with a cycle time of 1-2 microseconds. In one or two microseconds, one word can be delivered to every one of the 64 PE's within the quadrant. With all four quadrants, the total BUM channel requirement is 256 words per 1-2 microseconds, or 8-16 billion bits per second. Studies of linear programming problems have indicated a need for a 24 billion bit rate, and a 20 million word total memory. The complete BUM system is not expected before 1972. It is felt that by that time prices on large memory should come down to less than 1¢ per bit, and each BUM should cost 320K-640K dollars.

There is no firm plan to install BUMs in the 1969 hardware. The Illinois people would like to get one BUM for experimentation. Since each BUM is to attach to an individual QUAD, the BUM channel is not really identical with the I/O Bus, though much sharing can be achieved.

Each PE is expected to be able to connect directly to the external word, and thus operate at a 1-billion word per second transport rate. This possibility is interesting mainly for microsecond real-time situations, and the maximum bandwidth is not expected to be used often.

I/O Requirements Summary

Total system has:

1 - I/O Processor (S/360 Mod 44-50 class)

1 - I/O Bus: 64 word (4096 bit) wide, to deliver 64 billion bits/sec.

1 - Disk file (desired: 10 billion bits at 384 million bits/sec.)

8. Summary

With a 1/4 microsecond memory and 40 nanosecond circuit cycle time, the PE design is well-balanced at 10-12 thousand circuits. The delivered product may have 14 thousand, due to unforeseen requirements or installation of memory fetch/execution overlap. The packaging of PE memory with PE circuits, rather than with other PE memory units, is probably not optimum, but may be demanded by hardware requirements.

The floating point orientation of the PE's is good. It is difficult to justify a hydrodynamic problem-solver without floating point; the days of hand scaling by programmers is over. The fact is, not much hardware can be saved by elimination of floating point arithmetic, and the latter's hardware simulation for a number of synchronous fixed point PE's is very unrewarding. Also for reason of synchronism not only each unit must have floating point, but must complete the same instruction with different operands at the same time (or perhaps sub-instructions already have to be synchronized). This requires some hardware investment, such as a one-cycle full shifter.

It is possible for small subsets of the PE's to pool their resources together to achieve faster computing for less hardware. The memory slowness, however, posts a restriction on speed gains, and the hardware saving is small with the communication and packaging problems worsened. This is not too worthwhile as circuit count is but one of the cost factors. The others being memory cost, packaging-cooling, and powering.

From the point of view of architecture, therefore, the PE's are individually fairly well balanced. Hypothetical small changes of the design itself is not too rewarding. Hypothetical changes on small subsets of the design is again, not too rewarding. Improvements must be sought based on reorganization of large chunks, or better, at the global level. Can the design be at a relative optimum, yet misses the global optimum?

REFERENCES

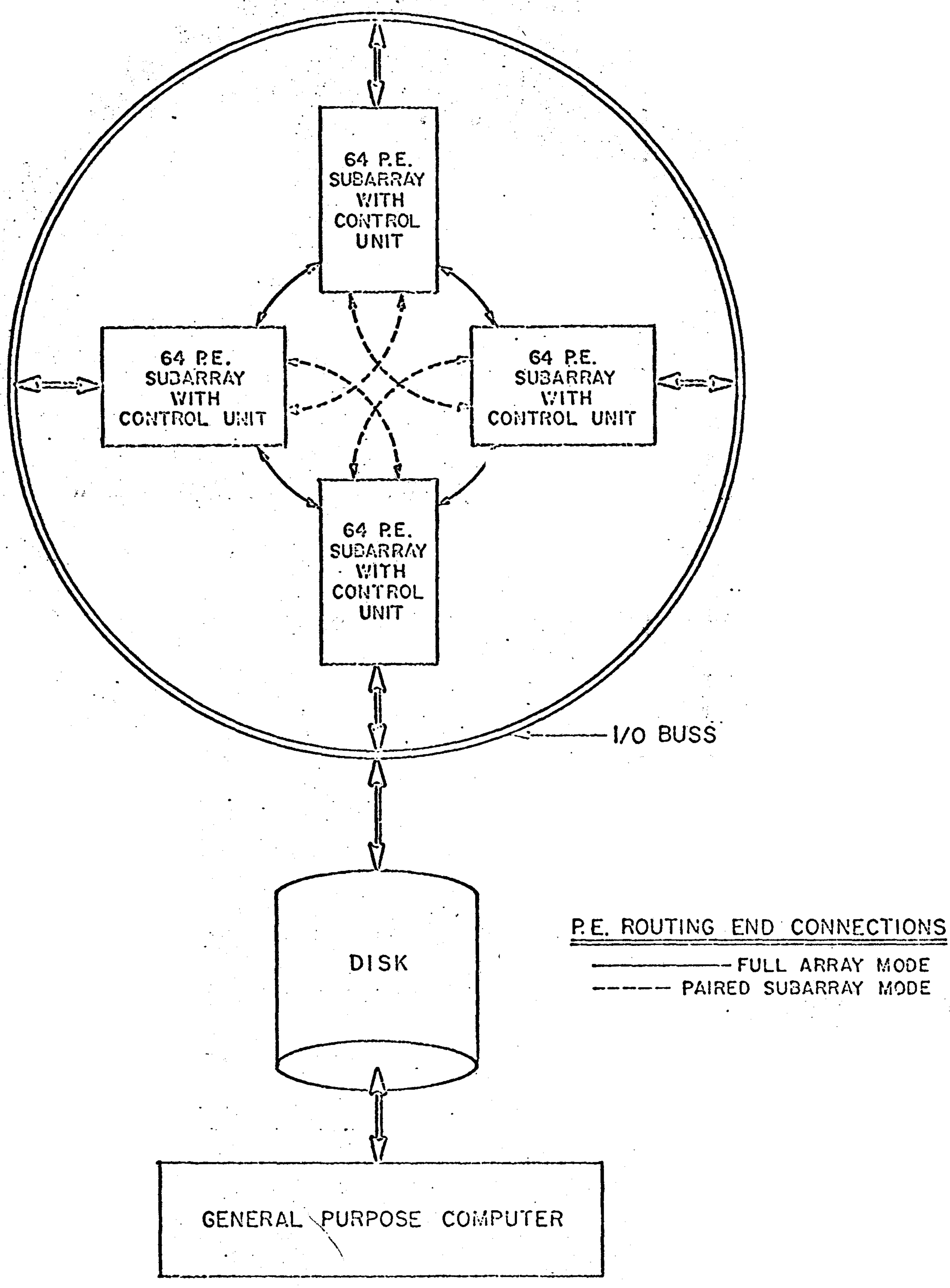
1. John Cocke and Daniel L. Slotnick, "Use of Parallelism in Numberical Calculations," IBM RC-55 (1958).

2. Manfred Kochen, Unpublished report.

3. D. L. Slotnick, W. C. Borck, R. C. McReynolds, "The SOLOMON Computer," 1962, Fall Joint Computer Conference Proceedings, pp. 97-107. Spartan Books (Baltimore, 1963). Also by the same authors, "Comments on H. J. Heijf's Review of "The SOLOMON Computer" IEEE Trans. EC-12, 410 (August 1963).

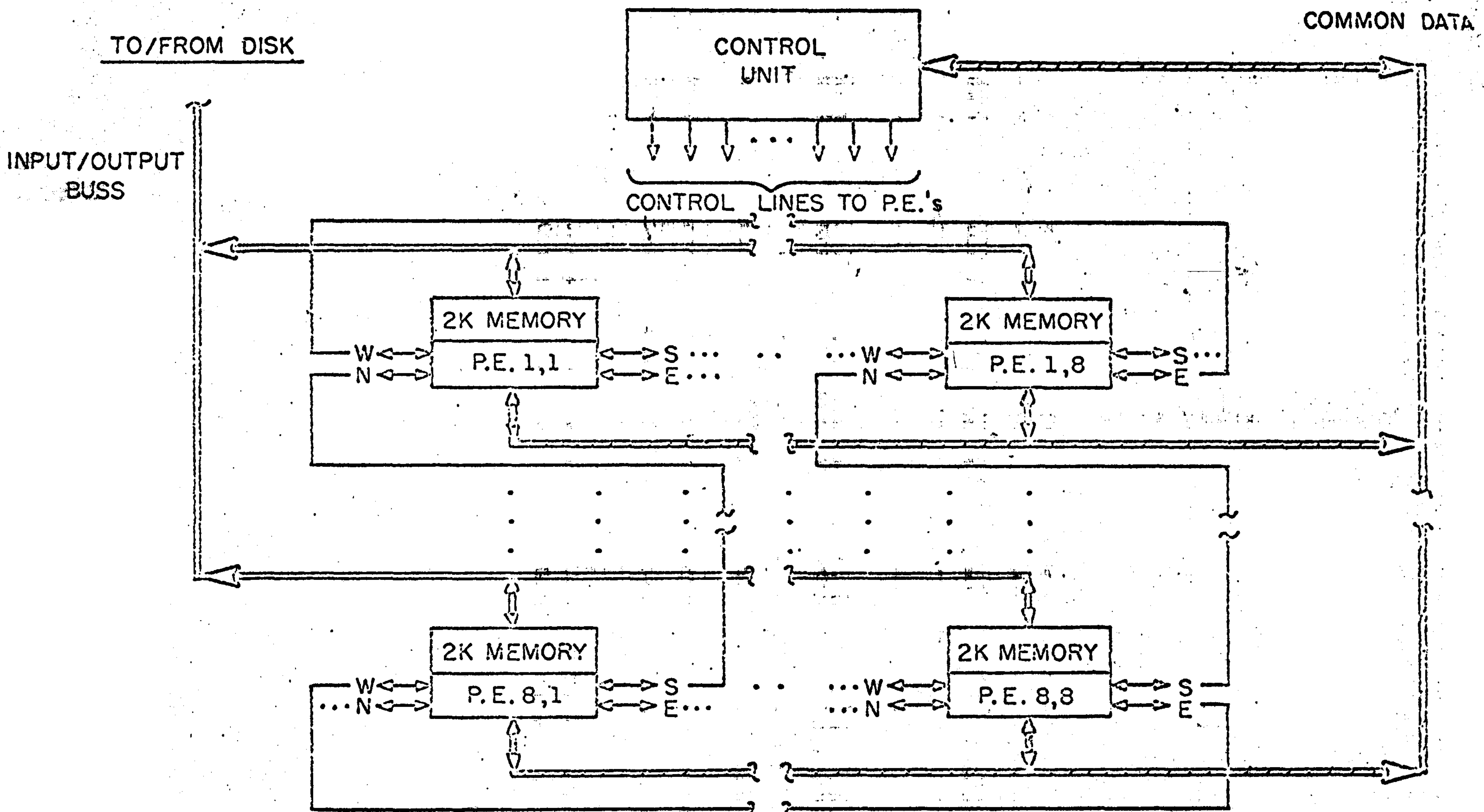
4. J. Gregory and R. McReynolds, "The SOLOMON Computer," IEEE Trans. EC-12, 774-781, (December 1963).

5. Daniel L. Slotnick, "Unconventional Systems," 1967, SJCC Proceedings, pp. 477-481. Also Richard M. Brown, "The Logical Structure of ILLIAC IV," David Kuck, "Systems and Applications Programming of ILLIAC IV," presented at SJCC Session 8, April 18, 1967.



ILLIAC IV GENERAL ORGANIZATION

Figure 1.



ORGANIZATION OF 64 P.E. SUBARRAY

Figure 2.