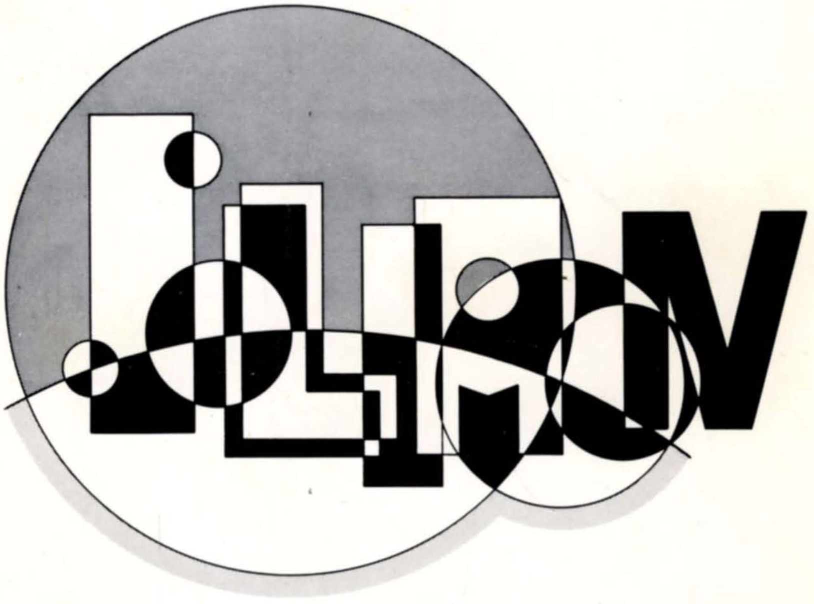
60005

B-176-B

22 APRIL 1968

ILLIAC IV

TECHNICAL SUMMARY



Burroughs Corporation

PROJECT SUPPORTED BY ADVANCED RESEARCH PROJECTS AGENCY AS ADMINISTERED BY UNIVERSITY OF ILLINOIS, URBANA, ILLINOIS AND THE UNITED STATES AIR FORCE ROME AIR DEVELOPMENT CENTER AT GRIFFISS AIR FORCE BASE, N. Y.

CONTENTS

CONTENTS (Cont'd)

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION I

INTRODUCTION

This document introduces the ILLIAC IV – a fourth generation computing system that employs an advanced concept of parallel design to achieve a major increase in processing capacity.

Section II presents a general discussion of the ILLIAC IV system organization and of the major units within this organization. Emphasis is given especially to interactions between the major subsystems – ILLIAC IV array, I/O interface equipment, disk file, and B6500 control computer – and the primary functions that each performs.

Section III treats simulation results obtained to date for ILLIAC IV application. Some additional problems are described to indicate other tasks that appear to be especially suitable for such a highly parallel computer organization.

Section IV describes the hardware characteristics of ILLIAC IV and emphasizes the design features of specific subsystem equipment. The microelectronic technologies used for implementing the logic, thin-film memory, and power system are also detailed.

Section V completes the presentation by discussing the availability of the ILLIAC IV computer system in terms of reliability and maintainability.

SECTION II

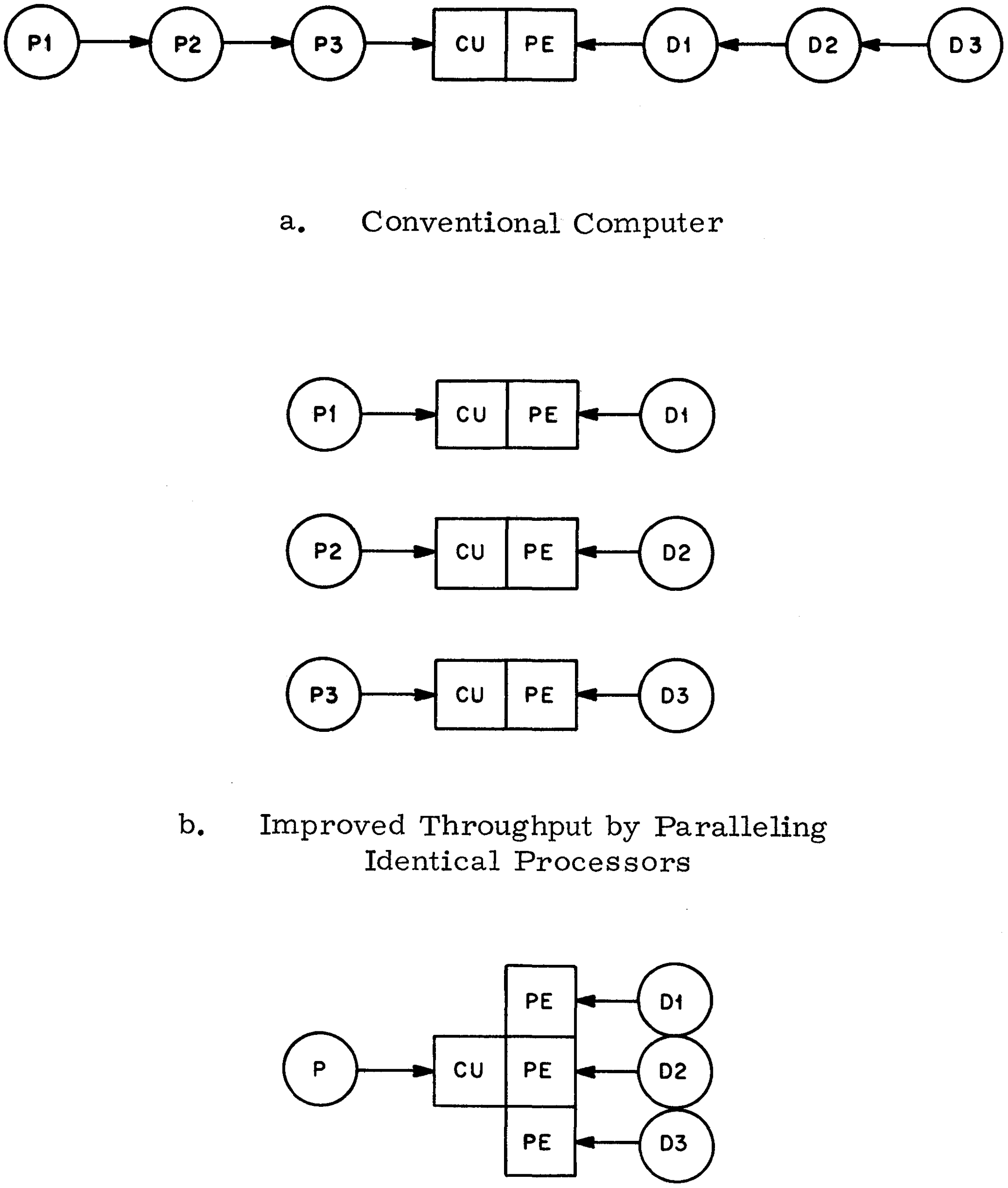
GENERAL FUNCTIONAL DESCRIPTION

SYSTEM

ILLIAC IV is a large digital computing system that provides a level of parallel processing many times that of conventional designs. To achieve this, a new and fundamentally different approach is used. For important classes of problems, many repetitive loops of the same instruction string are executed with different and independent data blocks for each loop. Parallelism may be applied here by using N computers, each executing the identical program concurrently on separate data blocks. This improves execution time by a factor of N for that program. Similarly, since each computer is executing the identical program, much of the control logic of the computers could be made common. This is the fundamental proposition of the ILLIAC IV computer.

Figure 2-1 shows a three-step evolution from conventional design to the ILLIAC IV. The top schematic (Figure 2-1a) shows three identical program loops (P1, P2, P3) operating on three different data blocks (D1, D2, D3) in series. The block element shown is a computer, without input-output or memory, that is functionally separated into a control part (CU) and an execution part (PE). Figure 2-1b shows a simple application of parallelism that produces a threefold increase in throughput. The final schematic in Figure 2-1c shows the ILLIAC IV approach with its simplifications and economies over the above method.

The ILLIAC IV system has a distributed memory system which allows each execution element uninhibited access to an assigned data block within its own memory. If a conventional centralized memory were used, much time would be wasted in routing data to and from such a memory.



a. Conventional Computer

b. Improved Throughput by Paralleling Identical Processors

c. ILLIAC IV Approach, Using Common Control Logic and Parallel Identical Processors

Figure 2-1. Development of Parallelism Toward Improving Program Throughput

SYSTEM ELEMENTS

The four major elements of the ILLIAC IV are the Control Unit (CU), the Processing Element (PE), the Processing Element Memory (PEM), and the Input-Output (I/O) subsystem. The combination of a PE and a PEM is called a Processing Unit (PU). A CU directly governs 64 PUs configured in an array as illustrated in Figure 2-2. In the ILLIAC IV system there are four such identical subarrays called quadrants, making a total of four CUs and 256 PUs. Quadrants may function jointly or separately.

Each PU is labeled with a unique three-digit octal number. The first octal position is the quadrant number and the second two positions are the PU number within a quadrant. The four "nearest neighbor" connections within the array are defined in terms of direct parallel word transfer paths between one PU and others with labels that have values plus or minus eight, or plus or minus one, from the value of the former PU's label (Figure 2-3). Thus for example, PU 33 can transfer directly only to PUs 23, 32, 34 or 43. This connectivity is maintained for both separate and joined quadrants, and enables a variety of physical images to be modeled - for instance, weather maps - by means of a combination of these transfer paths. All CUs have full-word data interconnections for programs that operate in more than one quadrant.

The Burroughs parallel disk file is the principal secondary storage element. Successor to the present head-per-track disk files, this file provides a storage capacity of 161×10^6 bits per storage unit with a transfer rate of 500×10^6 bits per second. Six such storage units are provided for the initial ILLIAC IV system. Data is routed in and out of the disk files through the I/O Controller (IOC), the I/O Switch (IOS), and the Buffer I/O Memory (BIOM).

CONTROL COMPUTER FUNCTIONS

To complete the system, a B6500 computer will serve as the principal managing element. All executive control, facility allocation, peripheral-equipment control, I/O processing and initialization, fault recovery, and program assembly will be done by this subsystem. Figure 2-2 shows a control link between the B6500 system and the Control Units. It is this link that the B6500 uses to set the initial state word in each CU. The state word includes the initial value of the program counter, the control state, and the configuration of the array. The configuration describes which quadrants are working jointly on the same program and which, if any, are operating independently. The B6500 will also institute the necessary disk-to-array memory transfer of program and operands before allowing the CUs to proceed.

The I/O Controller, supplied with start address and word count information by the B6500, provides the necessary intermediate memory address to the CU and the disk file during a transfer. Data transfers are made directly to or from the PEMs. Once the required number of instructions and operands

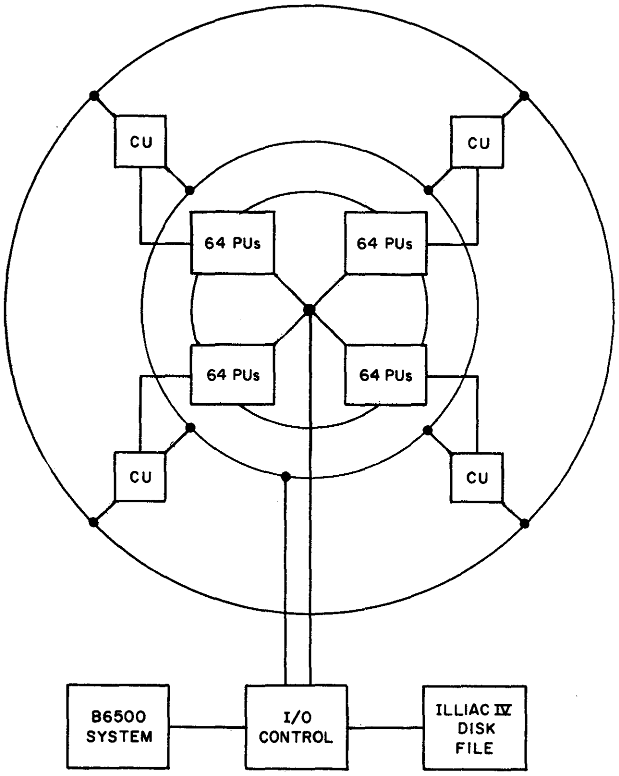


Figure 2-2. ILLIAC IV System Configuration

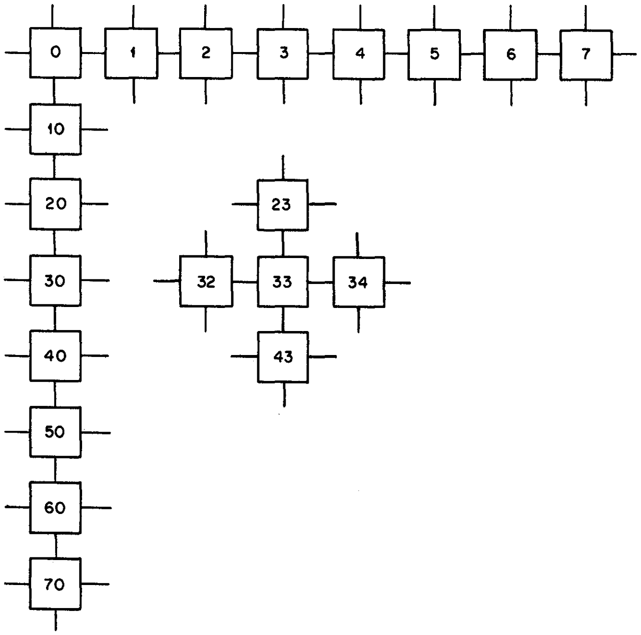


Figure 2-3. Array Connectivity

have been transferred from the disk, the CU will begin with an initial instruction fetch from the PEMs and proceed in the conventional manner of a stored program computer. Instructions as well as operands may be transferred across quadrant boundaries, so they need to be stored only once, regardless of the configuration.

CONTROL UNIT FUNCTIONS

The Control Unit is the part of the computer system that performs all the necessary initial instruction processing up to and including the generation of the instruction microsequences for a step-by-step control of the Processing Element. Figure 2-4 is a block diagram of this single cabinet unit. Contained within the CU are five separate operating elements which perform specialized processing tasks on a semi-independent basis. The instruction look-ahead (ILA) section of the CU fetches instruction words in 8-word blocks from the array memory into a 64-word content-addressable memory used as an instruction stack memory. Individual instruction blocks are located by an associative memory that holds all but the four low-order bits of each instruction address. The value from an instruction counter is sent to the associative memory to locate the proper 8-word group in the instruction stack. Then the four low-order bits are used to select the individual instruction. Program loops of up to 128 instructions can be contained within the instruction stack.

From the instruction stack, instructions are fed in turn to the advanced station (ADVAST), which is the principal housekeeper of the system. Such functions as address arithmetic, loop control, mode control, interrupt processing, and configuration control are performed here. The hardware complement of ADVAST consists of a 64-word operand stack, four full-word accumulators, and a combinatorial logic unit. The logic unit permits functions such as adds, compares, shifts, bit testing, etc. This station provides all those activities generally described as program control to be performed concurrently in advance of, and separately from, the main processing activity.

Instructions fall into two general categories: those executed at ADVAST and those executed at the final station (FINST). Since all instructions are first at ADVAST, those instructions intended for execution at FINST are transferred to FINST through the final queue (FINQ). This element is composed of eight instruction storage positions, which perform a time-smoothing function between ADVAST and FINST. FINST decodes each instruction into control microsequences, which are broadcast to 64 array elements over a common control bus. FINST also broadcasts full-word operands, shift counts, test values, and other common array parameters on a data bus. In actual operation, FINST and the 64 array element sequences are lock-stepped, except for the fixed transmission delay of the intervening control bus.

The memory service unit (MSU) resolves the conflicts of the four users of the array memory: I/O, ADVAST, FINST, and ILA. It also transmits the

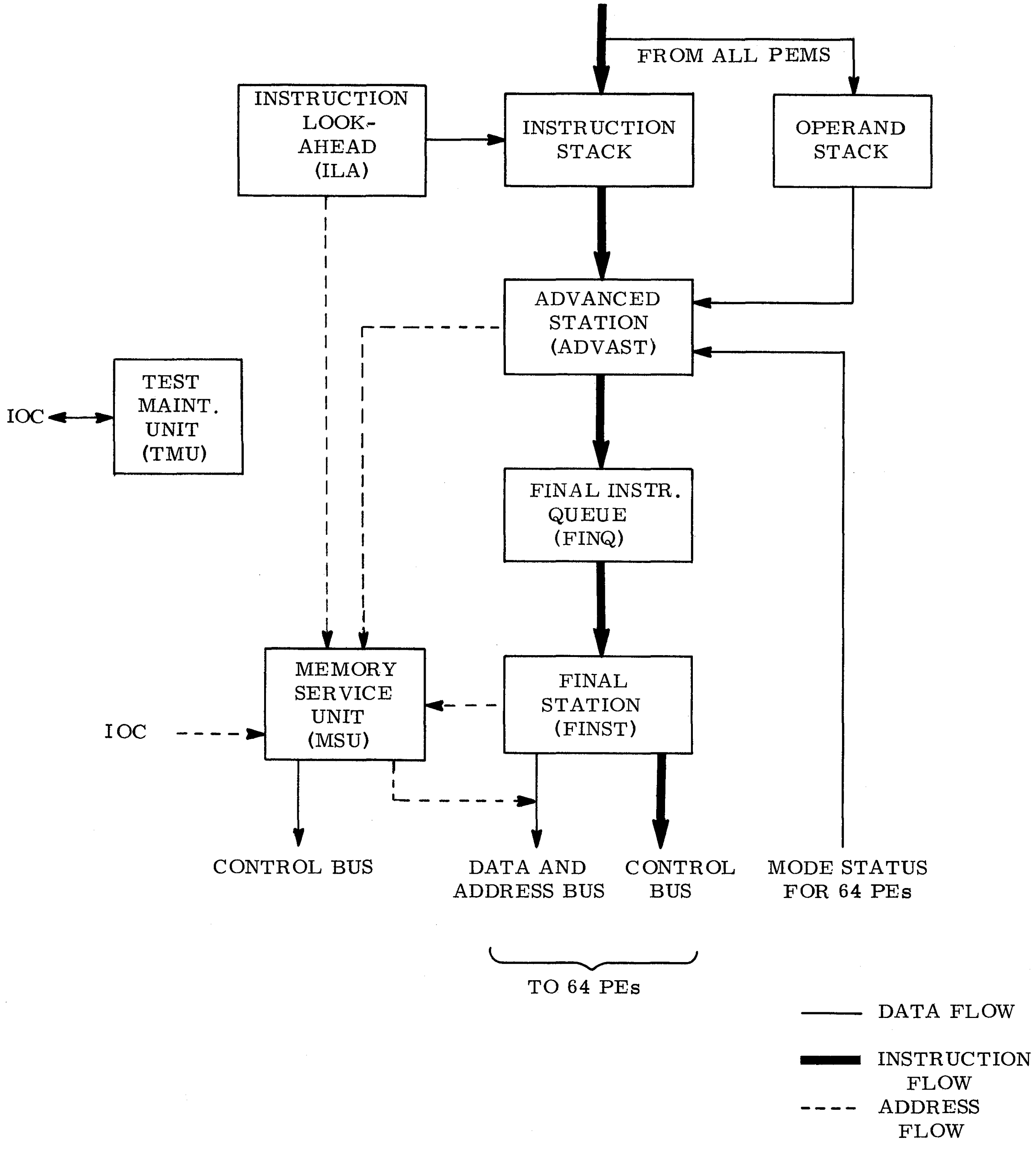


Figure 2-4. Control Unit

appropriate address to memory and exercises control over the memory cycle. As a hardware expedient, the addresses are transmitted over the same common data bus mentioned above.

The test maintenance unit (TMU) provides the control channel for the B6500 and the manual maintenance panel to the Control Unit.

The array element, the execution portion of the computer shown in Figure 2-1c, is called a Processing Element (PE). This unit is devoid of all independent control with the exception of mode and some data-dependent conditions. Mode permits a PE to accept or ignore a broadcast control sequence from the CU.

Figure 2-5 is a block diagram of a PE. It is essentially a four-register arithmetic unit which performs a full repertoire of instructions on 64-, 32-, and 8-bit operands. Full floating-point operations are included for the 64- and 32-bit words.

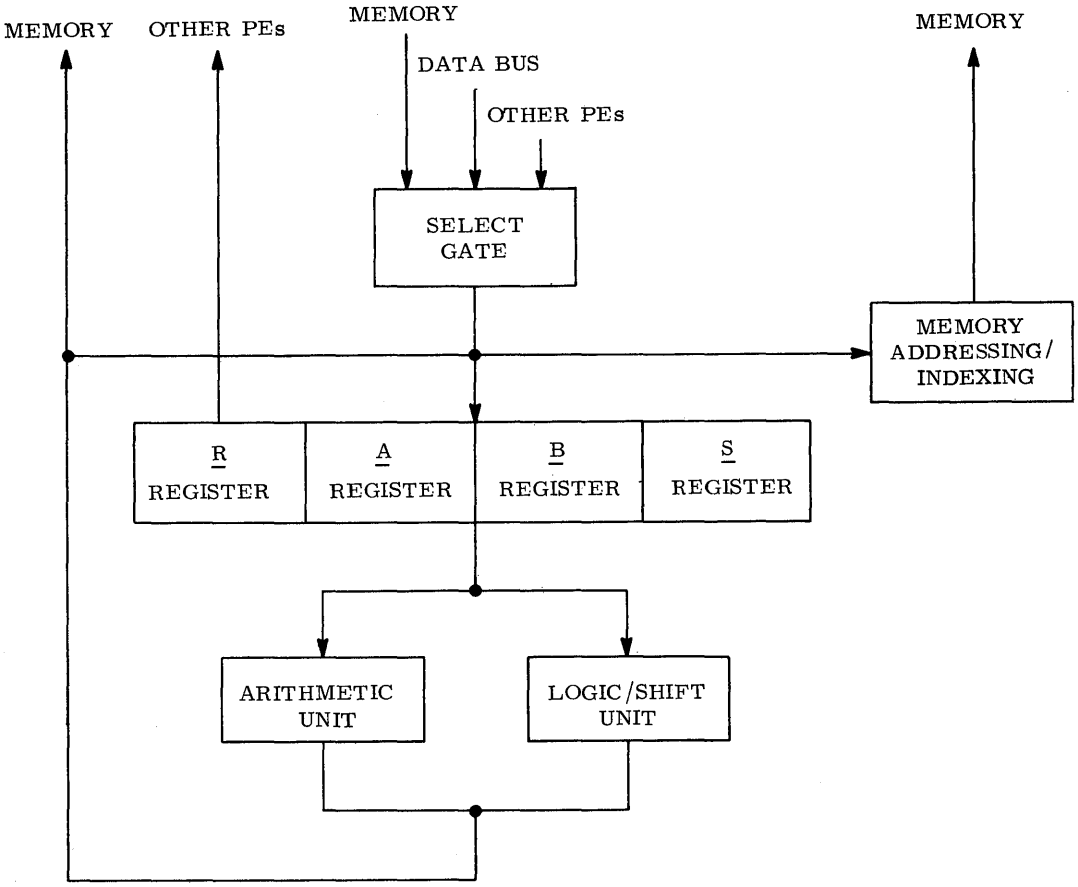


Figure 2-5. Processing Element

An arithmetic unit combines a carry-save adder tree and a parallel adder with carry look-ahead logic to give a floating-point multiply time on the order of 400 nanoseconds and a floating-point add time of 200 nanoseconds. Both times include post normalization. Other logic elements include a barrel switch for rapid data-shifting, a leading-one's detector and a logic unit for Boolean operations. Instruction operands may originate in any of the PE registers, the common data bus, the nearest orthogonal neighboring PEs, or the array memory.

The array memory (or PEM) consists of independent thin-film memory modules with each module collocated and assigned to a PE. Each module has 2048 words of 64 bits. The memory is designed for a 250-nanosecond read-write cycle. The PE memory address register supplies memory addresses. A separate address adder and index register permit independent memory indexing and addressing. Such independence provides important flexibility for addressing data stored in a variety of ordered forms.

CONTROL UNITS

Each Control Unit (CU) directly controls 64 Processing Units (PU) of a four-quadrant array, as was noted in the preceding section. Four identical quadrants comprise the ILLIAC IV system, making a total of four CUs and 256 PUs. Associated with each subarray of 64 PUs are certain common registers and logical elements which can be manipulated by instructions. Decoding of instructions for the Processing Elements (PE) is also common. Both the decoding functions and the common registers and logic are contained within the CU. The CU manipulates two types of instructions in the instruction stream: those instructions which it decodes for specifying commands for the PEs - called PE instructions - and those which control the common registers - called ADVAST instructions. Some of these instructions are used to effect communication between the common registers and the PEs. A detailed block diagram of the CU is shown in Figure 2-6. A general block diagram of the CU showing its five main functional areas appeared previously in Figure 2-4.

In arrays of 128 or 256 PEs, there are two or four CUs operating in parallel. These CUs normally execute identical programs, have identical initializations, and precede data-dependent actions by sharing of data among both or all four CUs. Therefore, the separate CUs will execute identical instructions in parallel, and they will be indistinguishable from one unit with 128 or 256 Processing Element Memory units (PEMs).

The CU shares the same physical memory with the PEs. Addressing of memory uses the PE number as the least significant portion of the memory address. Successive memory addresses therefore progress across individual PEMs such that addresses "n" and "n + 1" are in adjacent but different PEMs.

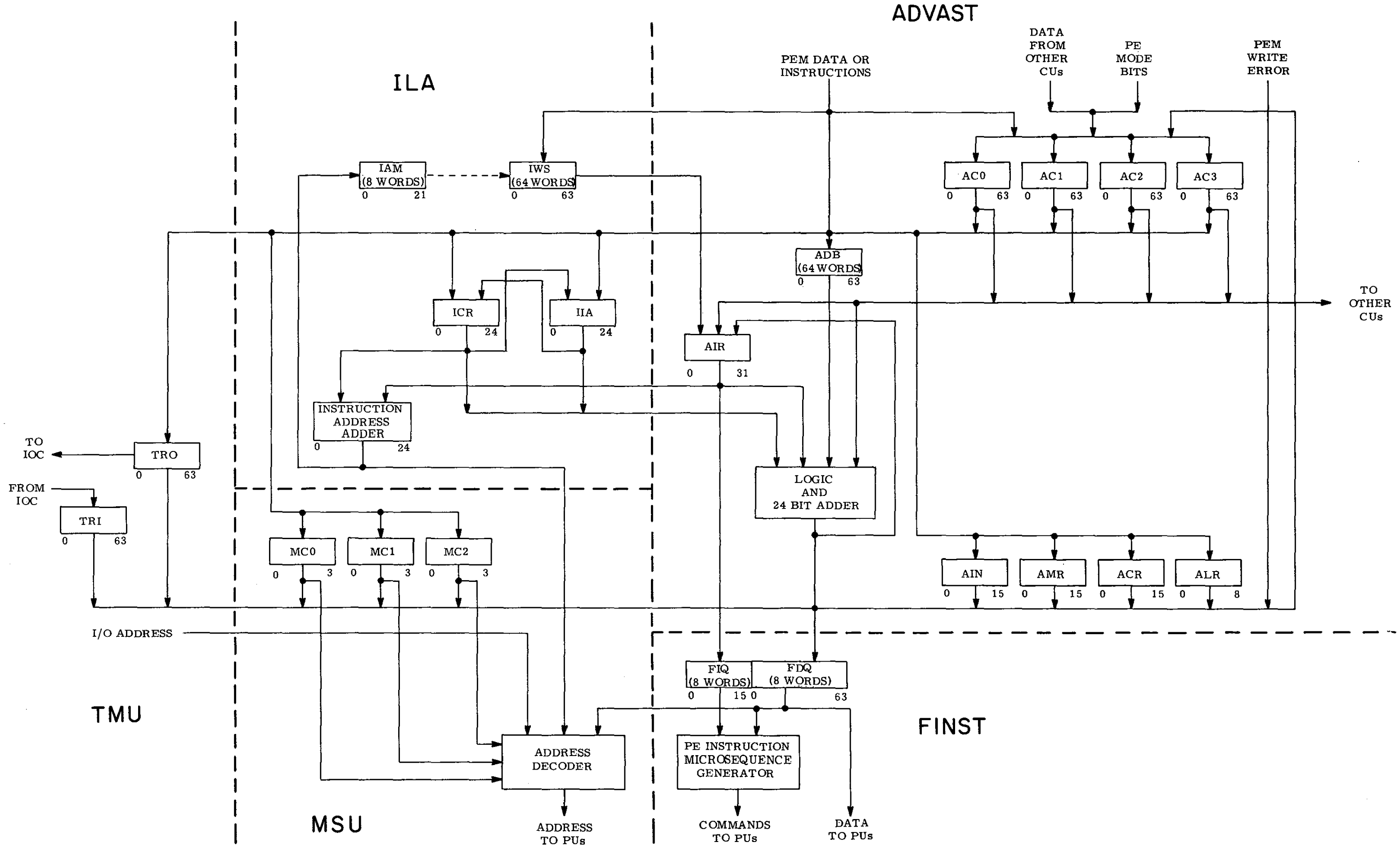


Figure 2-6. Control Unit Block Diagram

Program steps are fetched in blocks from memory, and executed one at a time. Although there is rather extensive machinery in the control unit to reduce the actual number of memory fetches from one fetch per program step, as in conventional machines, to 0.0025 or 0.015 fetch per instruction, this machinery requires no attention on the part of the user programmer.

The registers in the CU are as follows:

AC0, AC1, AC2, AC3 - A set of 4 registers, 64 bits each, general purpose accumulators (ACARs)

ACR - ADVAST control register, which contains CU status information

ADB - A set of 64 registers of 64 bits each, used as a scratchpad memory

AIR - ADVAST instruction register

AIN - ADVAST interrupt register

AMR - ADVAST interrupt mask register

ALR - A register which holds the address of pending memory fetches

MC0, MC1, MC2 - Array configuration control registers

IIA - ILA interrupt storage for ICR

ICR - ILA instruction counter

TRI - TMU input register

TRO - TMU output register

All of the above registers can be manipulated by the program.

CONTROL UNIT STRUCTURE

The order code contains instructions of the common-register manipulating type (ADVAST instructions) and of the PE controlling type (PE instructions). Since the two instruction types do not interact, they can be viewed as two interlaced but distinct instruction streams. The hardware of the CU takes advantage of this partial independence to execute the two streams independently and concurrently with each other. The CU has five main functional areas, as follows:

Instruction Look Ahead (ILA). The instructions are fetched, in large blocks of contiguous code, to a section of the CU called the instruction look-ahead (ILA). An associative memory (IAM) detects which blocks or instructions are currently in ILA storage. ILA also contains the instruction counter (ICR).

Advanced Station (ADVAST). Each instruction is passed in sequence to the instruction register (AIR) of the advanced station (ADVAST). Since most of the common registers make up the ADVAST section, instructions referring only to the common registers are discarded when manipulations are completed.

Final Station (FINST). Instructions from ADVAST enter a section of the CU called the final station (FINST). Outputs from FINST manipulate the Processing Elements. Instructions enter FINST through a final queue (FINQ) so that the instruction execution time at FINST is decoupled from the execution time at ADVAST. Some instructions (e.g., LOAD) are partially executed at ADVAST and partially executed at FINST because of potential interaction between the two sections. In general, the programmer need not be aware of the overlap and asynchronism between the two sections since, under normal conditions, the instructions are properly sequenced by the hardware.

Memory Service Unit (MSU). The memory service unit (MSU) receives requests for memory from three sources: FINST, ILA, and the input-output controller (IOC) of the I/O subsystem. The MSU resolves conflicts among the three sources as well as conflicts concerned with other FINST uses of the common paths from CU to memory.

Test and Maintenance Unit (TMU). The test and maintenance unit (TMU) of the CU contains TRI and TRO (which are addressable by instructions in ADVAST) and provides paths to the maintenance panel, the display, and the B6500. The display will, on external command, indicate the state of any CU register. A portion of TMU serves as a "test instruction" register for diagnostics, testing, and initialization.

TIMING CONSIDERATIONS

Potential program difficulties are introduced by the asynchrony between ADVAST and FINST since ADVAST may be executing instructions which occur later in the instruction stream than those which are in FINQ awaiting execution. In the majority of the cases, the hardware automatically detects the potential problem and introduces the necessary synchronism to prevent its occurrence. All memory referencing instructions, for example, whether LOAD, STORE, LDA, or STA, are properly synchronized with each other. There are some cases where the same bits are accessible from both ADVAST and FINST (or the PEs). For example, changing of the bit in ACR which controls the response to floating-point underflow is not synchronized with arithmetic executions at FINST. If the ACR bit must be altered, instruction FINQ is used. Changing the word size must be done with instruction CHWS. However, there is a variant of the CACRB instruction which changes word size but asynchronously with FINST. The effects of program-caused interrupts are somewhat delayed in reaching AIN; an attempted STORE instruction into a program area may have no affect on the program being executed.

CU WORD FORMATS

The decoding of instructions requires the utilization of two instruction formats: for the ADVAST and PE instruction sets. The former is used for instructions which are executed at ADVAST; the latter is for instructions which pass through ADVAST directly to FINST, with no decoding at ADVAST. The two formats correspond approximately to those instructions which manipulate the common registers and those which are solely concerned with the Processing Elements, respectively.

SEQUENCE OF OPERATION

The operation of the ILLIAC IV system is somewhat complex due to the close coupling of interquadrant operations and the largely decoupled operation of intraquadrant functions. Superimposed on this structure are communications with the B6500 and the IOC, which can be considered as being asynchronous with the ILLIAC IV system itself. The program flow described here traces the actions of the various system components during the execution of a program.

System Start-Up

The B6500 receives a job request and places the program and required data base on the ILLIAC IV disk system. The quadrants of the system which will be used for this program are then selected, and a command is sent to the TMU sections of the selected CUs which causes the CUs to be stopped and initialized. The B6500 then causes the disk-held program and data to be loaded into the appropriate array memory locations by issuing commands to the IOC. When loading has been completed, the B6500 sends commands to the TMUs which cause the instruction counter (ICR) in the ILA section to be set to the first instruction in the program and the system to be started.

Fetching the Program

After initialization, the instruction look-ahead unit (ILA) is set to indicate that there are no instructions in its instruction word storage (IWS). Immediately upon start-up, the ILA will recognize this condition and request a block of instructions — via the MSU — from the PE Memory that contains the instruction addressed by the ICR.

The IWS may be considered as an instruction queue for the ADVAST station. It holds up to 112 instructions which are fetched in blocks of eight words, two instructions per word. IWS contains seven of these blocks. ILA checks IWS whenever the eighth instruction of a block of 16 has been accessed to assure that the next block of 16 instructions (sequentially) is available in IWS. This "look-ahead" provides sufficient time to load the instructions before they could be sequentially addressed by the ICR. The associative memory (IAM) performs this function. If the instruction block is not in IWS, it will be fetched from PE Memory and placed in the next free block in IWS or it will overlay the oldest instruction block in IWS. IAM keeps track of where the blocks are in IWS, and from where in PE Memory they came. Except for initial start-up and for transfers of control to instructions not held in IWS, ADVAST is never delayed awaiting instruction fetches.

ADVAST Processing

The function of the ADVAST station is to handle the housekeeping for the quadrant. From a programming point of view, FINST and the PEs perform the "inter-loops" of a program while ADVAST handles most of the "outer loop" and control functions. Included in its tasks are exception condition processing, interquadrant decision making, and interrupt handling.

When ILA holds the instruction addressed by the ICR, the instruction is sent to the ADVAST instruction register (AIR) which determines whether it is a PE (FINST) type instruction or one that ADVAST can process. In the former situation the instruction will be passed on to the final queue (FINQ) to await execution by FINST and the PEs. ADVAST instructions remain in the AIR while they are being executed.

The ADVAST station AC (or ACAR) registers are primarily index/limit/increment registers that are used to supply addresses for PE instructions. They also can be used for performing logical functions such as decision making and data formatting. The ADVAST data buffer (ADB) is used in conjunction with the ACARs in data formatting and information broadcasting to the PEs. The other registers controlled by ADVAST are manipulated to effect program sequencing and control.

Final Station Processing

The final station (FINST) accepts PE instructions from the AIR and places them in the final queue (FINQ). FINQ is composed of two sections: the FINST instruction queue (FIQ) and FINST data queue (FDQ). FDQ holds the address values or data required by the instruction in FIQ. There are eight locations in FINQ that are serviced on a first-in first-out basis. It is FINQ that permits the concurrent operation of ADVAST and FINST.

PE instructions are taken from FINQ for execution. The MSU participates in this when a PE instruction requires memory access. The PE instruction, when taken from the queue, is in largely undecoded form. The function of FINST is to decode the instructions from this form into sets of microsequence commands for the array of 64 PUs. In some cases synchronism with other quadrants in an array is required and is also accomplished in this process. The generated microsequences contain the individual enable signals that control the information flow — both in direction (register to register) and in time — within the PUs. The generated microsequences are then broadcast to all of the PUs selected to accomplish the execution of the instruction.

Communication and Input-Output

When the ILLIAC IV has processed a block of data it may require more data and/or program, or the output of the processed data. The system has no input-output commands of its own. Instead, the CU places a request code in its TMU output register (TRO) which then interrupts the B6500. The B6500 reads the request code via the IOC and interprets its meaning. The B6500 will send an "operation complete" code to the appropriate TMU(s) to be accumulated in the TMU input register (TRI) when the requested operation has been performed. The CU can accept this information by periodic sampling of the TRI or on an interrupt basis. The CU will interpret the operation complete code and cause the indicated processing to be performed.

Other CU Functions

Other CU functions are largely ADVAST controlled. Synchronism requirements are delineated in the individual instruction descriptions and are accomplished at either ADVAST or FINST depending on the instruction set. The Configuration Control description details the grouping of quadrants into arrays and the synchronism that this implies. The interrupt system is described in the Operational Control section, which explains in more detail the uses and effects of the associated registers. The content of the control registers is also described separately so that the features for programming utility and service routines are available for the systems programmer.

PROCESSING UNITS

The Processing Unit (PU) functions as a general purpose computer under the direction of an ILLIAC IV Control Unit (CU). All of the 256 Processing Units in the ILLIAC IV system are electrically, mechanically, and functionally identical. Each PU consists of a Processing Element (PE) and a Processing Element Memory (PEM). Data inputs to and outputs from the PE and PEM are shown in Figure 2-7.

For control, the PE and PEM receive enable signals from the CU for sequential enabling of data paths and logic during the execution of instructions and for controlling the reading and writing in the PEM. In addition, the CU monitors the control status of the PE by one input and one output of the PE mode logic, and the memory protect error status of the PEM by one input and one output of the PEM.

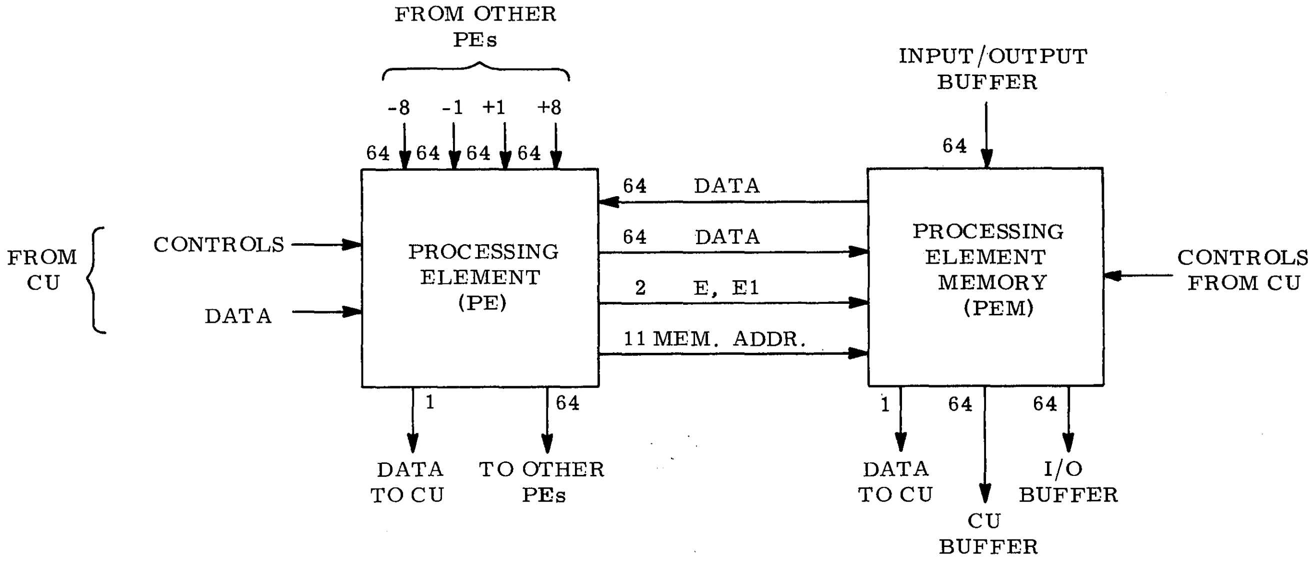


Figure 2-7. Processing Unit Data Inputs and Outputs

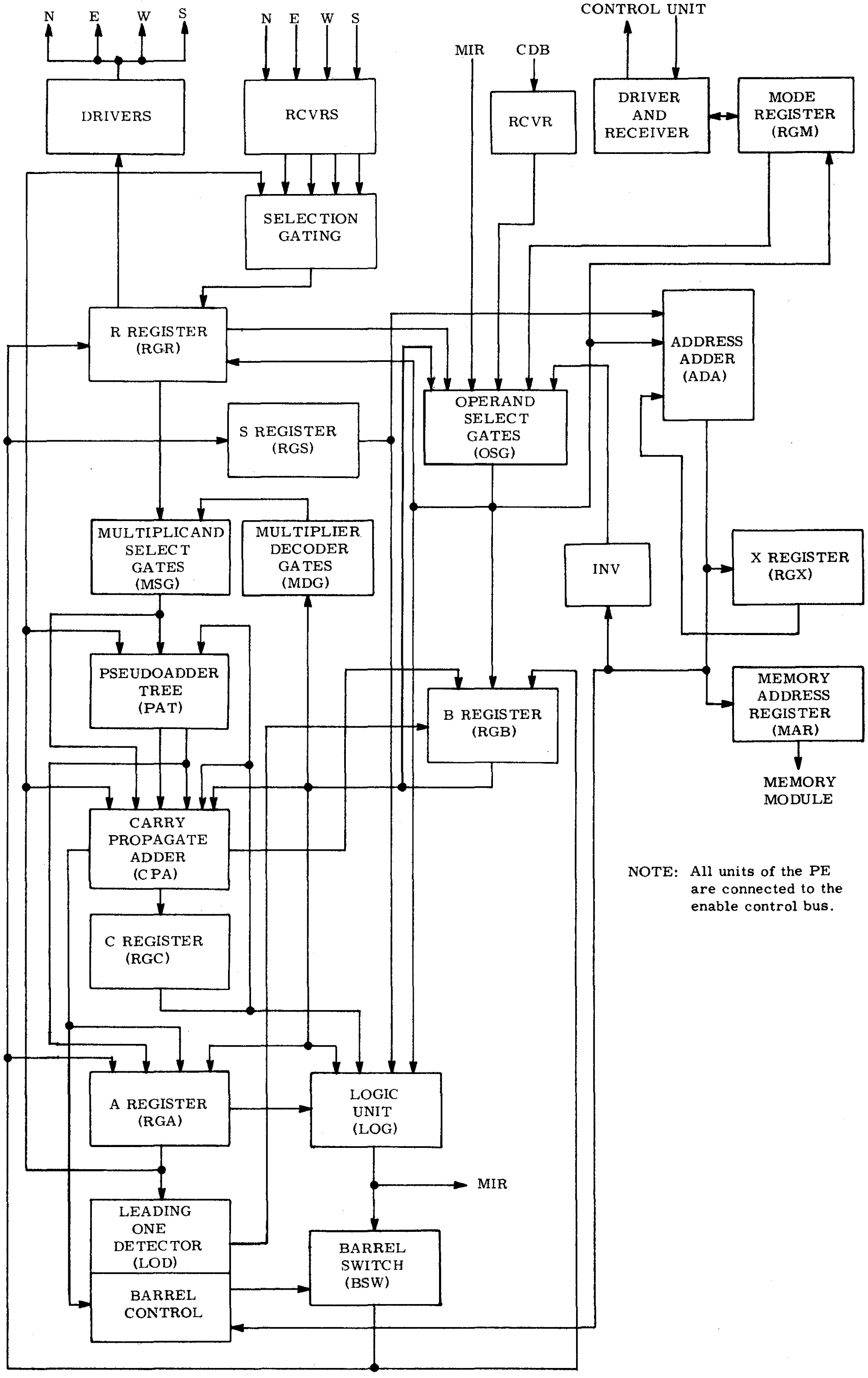
PROCESSING ELEMENT (PE)

The block diagram in Figure 2-8 shows the data manipulation portions of the PE; distribution of controls in the PE is omitted from the diagram. The principal registers within the PE are five 64-bit data registers, one 16-bit index register, and one 16-bit memory address register. Large, parallel logic gating structures are provided for rapid shifting, adding, and multiplying. A full complement of arithmetic and data manipulation instructions can be executed with this equipment. Separate instructions allow use of 64-, 32-, or 8-bit word formats. All operation is fully synchronized in the PE by a clock supplied to it. The externally supplied controls are timed to this clock before being buffered for distribution within the PE. While most controls originate outside the PE, some data dependent controls (such as for normalization and signed arithmetic) are formed outside the PE.

Registers and Logic

Data Registers

The five 64-bit data registers are A, B, C, R, and S. The A register holds one operand and receives the output of the adder and may be considered as the accumulator. The B register holds a second operand and communicates most



NOTE: All units of the PE are connected to the enable control bus.

Figure 2-8. Processing Element Block Diagram

directly with external data. The C register is used in certain instructions to save carries from the adder. The R register is the routing register, used principally for communications with other PEs, and at times for temporary storage of operands. The S register is used for programmatic storage of an operand within the PE. A and S are protected by the enable bits E and E1.

Addressing

Addressing of a PE memory module is accomplished from the 16-bit address adder (ADA) via the memory address register (MAR). Inputs to the adder are from the 16-bit index register (X), the S register, or the operand select gates (OSG). Sums may be sent to X (which is also protected by enable bit E), to the 16-bit memory address register (MAR), and to the barrel switch (BSW) controls. The sum output is also sent to the OSG, but is used only for transfers from X. With these data paths, all shift counts and memory addresses are indexable by either X or S. Comparison tests may be made to either X or S, and X may be modified.

Adding and Multiplying

The requirements for the utmost speed in the addition and multiplication instructions demand a parallel adder capable of extremely rapid operation. The one chosen for a carry propagating adder (CPA) uses three levels of look-ahead to achieve a 64-bit sum in a single clock period. Eight-bit gating allows the interruption of carry propagating for byte operations. For speed in multiplication, the eight bits of the multiplier are decoded for each iteration and the proper multiplies of the multiplicand are generated by the multiplicand select gates (MSG) which are added in a multiple layer of parallel carry-save adders (CSA). This logic accomplishes a single multiplication iteration in one clock time of the multiply instruction.

Shifting

A 64-place, right shifting, end-around barrel is used as the shift network in the PE. With the logic unit to select the input and with full distribution of the output, the barrel allows generalized, one clock period shifting of registers in the PE. Extensive barrel control allows 64- or 32-bit words to be shifted left, right, end-off, or end-around. Inputs to the barrel control include shift amounts calculated by the address adder, fixed amounts required in certain instructions, and variable amounts derived from operands to be normalized or aligned. The normalization amount is generated in the leading one detector (LOD) - a fast, parallel logic network. From the output of the A register, the LOD locates the position of the most significant nonzero bit in the mantissa, 48- or 24-bit, and generates both the shift controls for the barrel switch (BSW) and a binary number to be used for exponent correction.

Mode Register

The mode register contains eight binary storage elements for controlling the operation of the PE and the storage of the PE state. The two E bits determine the enable status of outer (E) and inner (E1) half-words and are used to protect the A, S, and X registers and the memory information register (MIR) in the PEM. The two F bits are used to store faults (underflow, overflow, etc.). The remaining bits, G, H, I, and J, are manipulated in conjunction with the E and F bits and are used primarily for temporary storage of test results. By instruction, a mode bit may be set from the CU or its status may be sent to the CU.

Instructions

The instruction set of the PE is that of a complete, modern, general purpose digital computer. Floating-point arithmetic in both 64- and 32-bit words is provided, with options for rounding and normalization. The arithmetic instruction group permits full-word operations, 8-bit byte operations, operations ignoring exponents or using exponents only, and operations with fixed signs.. A full set of tests is permitted by making all registers addressable and allowing all possible comparisons to be made. Test results are set into a mode latch, which may then be used to programmatically direct the flow of the instructions. Instructions allowing interchanges of portions of 32-bit words, bit manipulation, shifts, and logical operations complete the PE instruction set.

Control

The PE is driven by a CU to execute the instruction string contained in the CU. The PE receives the fully decoded controls for the enabling of every data path and internal control of the PE. While many of these external control inputs are issued directly, some must be modified according to the data in the PE. Modifiers include the mode bits E and E1, the signs of the A and B registers, and the output of the LOD.

There are a few internal control signals of the PE which are generated in conjunction with data dependent operations such as multiplier decoding and mantissa normalization. These will arise in PE gates and are timed to coincide with external controls.

Processor Element Circuit Considerations

The high speed circuit performance necessitated by the Illiac IV system requires the use of circuits with propagation times of approximately 2.5 nanoseconds and capable of driving transmission lines. The Emitter Coupled Logic (ECL) circuit shown in Figure 2-9 has been used as the basic gate for design of the arrays for the system.

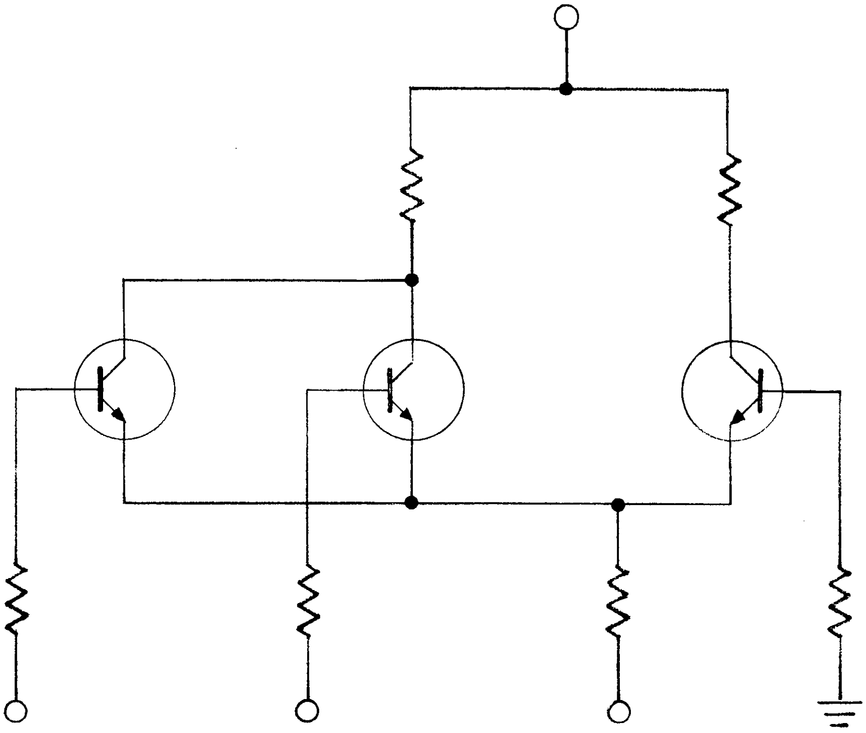


Figure 2-9. Basic Emitter Coupled Logic (ECL) Circuit

PROCESSING ELEMENT MEMORY (PEM)

A PEM provides 2048 words of storage, each word containing 64 bits. The memory operates in destructive readout mode (DRO) with a read/restore cycle time of 250 nanoseconds. The memory plane is organized as 1024 locations each containing 128 bits (two words). The 64-bit word which is not addressed will be read out of the memory plane and restored each memory cycle without being changed in any way. The PEM can accept data from the PE or from the Input-Output Switch (IOS) to be written into the memory plane, and can send data read from the memory plane to the PE, the IOS, or the CU.

A memory cycle is initiated by an initiate pulse from the CU if the PEM is selected. Data to be written into or read from the memory plane is temporarily stored in the memory information register (MIR) and is entered into the MIR, 100 nanoseconds after the initiate pulse. Another signal from the CU specifies whether a read or a write cycle is to take place. The memory data select bits from the CU determine the source of the data if a write is specified, or the destination of the data if a read is specified.

During a memory read, data is read out of the memory plane into the MIR. The data is then enabled out to its destination during the restore portion of the memory cycle. During a memory write, data is gated into the MIR from outside the PEM. During the restore portion of the memory cycle, it is written into the memory plane.

Data from the PE is written into the memory as a function of the E and E1 bits from the PE. Bit E specifies the outer 32 bits of the word (bits 0-7, 40-63) and E1 specifies the inner 32 bits (bits 8-39). All four combinations of E and E1 are permissible. Whenever a portion of the word is disabled by the E or E1 bit, the data in that portion of the addressed memory location is read out and restored unchanged.

Data read out of the memory plane and sent to the CU or IOS is accompanied by a signal which notifies the IOS or CU when data is available. Data to the IOS or CU is enabled out only between 100 and 200 nanoseconds after the initiate pulse. Data read out of the PEM to the PE is available until the next memory operation.

The first 128 words of the PEM may be write-protected, specified by a control signal from the CU. If a memory write is attempted in any of words 0 through 127 when the PEM is protected, the memory cycle will not take place; in this event, a memory protect error flip-flop is set, with a memory protect error signal being sent to the CU. The memory protect error flip-flop is reset by a signal from the CU.

The PEM is capable of transferring data from the PE through the MIR to the CU without performing a memory cycle. When a PE to CU transfer is initiated by a transfer pulse from the CU, PE data is entered into the MIR as a function of the E and E1 bits. If portions of the word are disabled, zeros will be placed into those portions of the MIR. The data is then enabled out of the MIR to the CU. Memory timing for a PE to CU transfer is identical to that for a memory cycle. A more detailed description of the array memory is presented in Section IV.

I/O SYSTEM

The three major component groups of the I/O system are:

1. A Burroughs B6500 data processing system which, together with its peripherals, performs all the functions of the control computer;
2. A Model II AP disk file subsystem providing approximately one billion bits of storage;
3. An I/O subsystem which interfaces between the above elements and the ILLIAC IV array.

The relationship of these elements to one another and to the array is illustrated in Figure 2-10 and described in the following paragraphs.

B6500 I/O CONTROL COMPUTER

The primary functions of the I/O control computer are to execute the supervisory program for the ILLIAC IV complex and prepare programs for ILLIAC IV. The supervisory program controls the operation of ILLIAC IV; schedules jobs for the array; maintains the Model II AP disks; transmits control words (descriptors) to the I/O Controller, which directs the I/O transactions in and out of the array; responds to interrupt conditions from the array or elsewhere; and communicates with the operator.

The initial B6500 data processing system necessary to run the supervisory program and prepare user programs consists of: one processor, 32 K words of memory, an I/O multiplexer with one peripheral control cabinet, and suitable peripherals including a disk file with 10^7 bytes of storage. Associated with the multiplexer are controller units which interface with the various peripherals. These are Burroughs units for the standard peripherals: magnetic tape, disk file, line printer, card reader, card punch, and console printer/keyboard. The B6500 can be expanded from this initial complement of equipment to include an additional processor and multiplexer as well as additional memory (up to 512 K words). On-line communication may be added by including a Datacom processor, multiline controls, and line adapters.

The interface between the I/O subsystem and the I/O control computer is designed to take advantage of the existing properties of the B6500 and the ILLIAC IV array. Control words are transmitted to the I/O Controller (IOC) through the scan interface provided from the B6500 processor. I/O descriptors are fetched by the IOC over the word interface of the B6500 multiplexer (MPX). There are two data paths between the B6500 system and the I/O subsystem. The data path between the IOC and the B6500 is via the word-wide path provided in the multiplexer. This path bypasses the multiplexer's own internal controls. In effect, it is an entry into the multiplexer's path to B6500 memory during those times that the multiplexer is not using it. The second and main data path involves the Buffer I/O Memory (BIOM), which is attached to the B6500 system as a 2730-word memory module. As shown in Figure 2-10, BIOM is connected to both the processor and multiplexer memory buses. All of the above interfaces between the B6500 system and the I/O subsystem are of 20-bit addresses and 48-bit data words. All of these interfaces utilize bidirectional cables. For a more detailed description of the B6500 refer to Section IV.

ILLIAC IV DISK FILE SUBSYSTEM

The ILLIAC IV disk file subsystem will initially consist of two Model II AP disk files with six storage units each. Each Model II AP disk file is comprised of an electronics unit and Burroughs Model IIA mechanisms, with

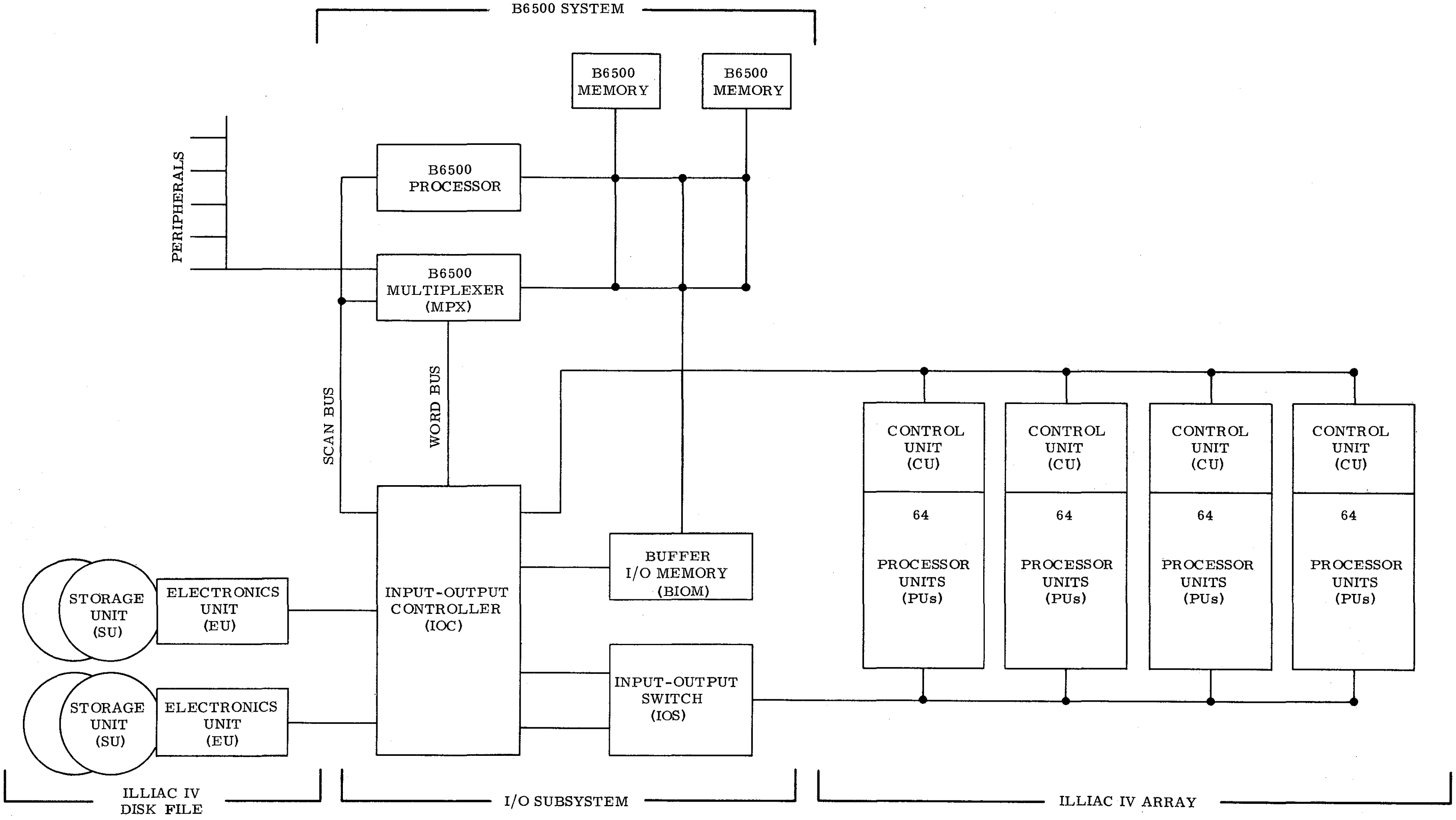


Figure 2-10. ILLIAC IV Interface Diagram

sufficient electronic circuitry for reading or writing simultaneously on 96 tracks of one disk. Each disk has a capacity of 78, 796, 800 bits and a maximum of nine such disks may be connected to an electronics unit. The maximum access time is 40 milliseconds. The electronics unit houses certain common electronics, registers for providing conversion of information from disk-serial to control-unit-parallel form, control logic, power, motor control, and the air pressure system. Approximate transfer rate to and from the Control Unit is 500×10^6 bits per second. The interface between each electronics unit and its controller in the IOC is 288 bidirectional data lines and 20 control-address lines.

The disk-track format in the ILLIAC IV file is an expansion of the format presently used in the B8500 disk system. To retain maximum bit packing density across the disk and therefore maximum storage capacity, the tracks are divided into three frequency zones into which data is written in a frequency ratio of 4:3:2. Sixteen tracks in each of the three zones on either face of a disk, or a total of 96 tracks, are activated simultaneously to provide a bit transfer rate of 288 bits every 570 nanoseconds or the approximate 500×10^6 bit per second system rate.

The track layout consists of 192 active information tracks per disk face, arranged in three zones of 64 tracks each, as shown in Figure 2-11. Within each zone the heads are wired such that 16 of the 64 tracks are selected at a time by one of four center tap drivers. The same center tap driver selects the combined 96 tracks for both disk faces. A clock head is located on each disk face to indicate segment location and provide timing pulses. A disk revolution is divided into 1200 segments and there are four logical tracks on a disk, thus providing 4800 segments in four revolutions. From all 96 tracks, a total of 16, 416 bits are read or written per segment.

ILLIAC IV I/O SUBSYSTEM

The I/O subsystem is shown in Figure 2-10 as consisting of the I/O Controller (IOC), Buffer I/O Memory (BIOM), and I/O Switch (IOS). The functions performed by these elements are briefly described below.

The IOC is comprised of two major functional sections: controller descriptor control (CDC) and disk file controller (DFC). The CDC receives I/O initiate signals from the processor via the scan interface; fetches I/O descriptors from B6500 memory via the MPX word interface; controls execution of these descriptors; and sends back result descriptors to the processor via the scan interface. CDC also executes the descriptors sent between the B6500 and the ILLIAC IV Control Units. The data interface between the CDC and CUs is 48-bit, bidirectional. The DFC consists of two controllers which execute descriptors held in CDC for transfers from disk to/from array, disk to/from BIOM, BIOM to/from array, or real-time link to/from array. All transfers involving the array are via the IOS.

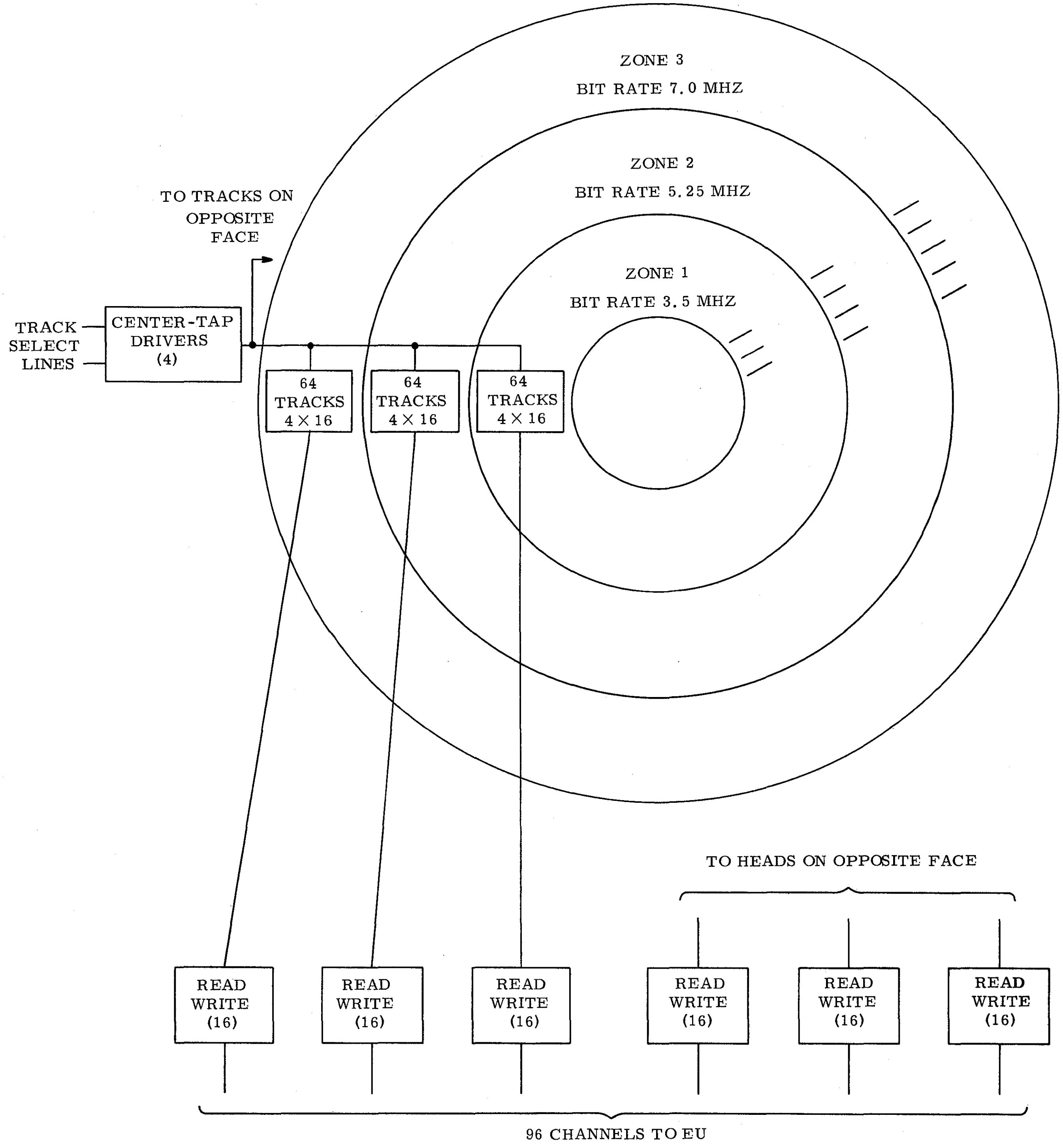


Figure 2-11. Disk-Track Configuration

As previously noted, the BIOM acts as a memory module for the B6500 system. Within the I/O subsystem, the BIOM has a 128-bit bidirectional interface with each of the two DFC units. All transfers through this interface are under the control of DFC descriptors.

The IOS unit buffers and distributes data between the IOC and the ILLIAC IV array. The IOS has a 256-bit bidirectional interface with each of the two DFC units and initially a 1024-bit bidirectional interface with the ILLIAC IV array. The IOS design provides for possible future expansion of the real-time link with the array to 4096 bits.

SECTION III

APPLICATIONS

INTRODUCTION

The ILLIAC IV, with its network of parallel processors, can effectively exploit the parallelism that exists in a large and important class of data processing problems to achieve orders of magnitude increase in speed over existing machines. To realize this increase in speed on any particular application it is necessary that ILLIAC IV be able to partition the data among the Processing Element Memories (PEM) so that the Processing Elements (PE) can be kept busy. The aim is to achieve a storage allocation scheme which provides as uniform as possible distribution of operands among the PE memories (to the end that one PE does not become over burdened by having to perform more computations than the other PE's), while at the same time avoiding the need for very complex indexing schemes which would cause the Control Unit (CU) to become a bottleneck by being excessively involved in purely housekeeping operations.

Table 3-1 presents a list of applications for which detailed analysis and coding have been done for ILLIAC IV operations. These pilot studies indicate that suitable storage allocation schemes can be devised for these problems to realize the potential increase in processing speed offered by the array organization.

Table 3-1. Some ILLIAC IV Applications

Matrix Operations

Matrix Storage Techniques

Inversion, Eigenvalues and Eigenvectors of Matrices

Solution of Linear Systems of Equations

Sparse Matrix Techniques

Linear Programming and Extensions

Partial Differential Equations and Simulation of Physical Systems

General Methods (Successive Overrelaxation, Alternating Direction Implicit, Fourier Analysis Solution of Poisson's Equation)

Numerical Weather Prediction (General Circulation Models)

Nuclear Reactor Calculations (Neutron Diffusion Equations, Neutron Transport Equations)

Weapons' Effects (Hydrodynamics and Photon Transport Equations of Atmospheric Nuclear Blasts; Effect of Nuclear Blast on Underground Structures)

Signal Processing

Phased Array Radar Data Processing

Seismic Array Data Processing

Multichannel Filter Design and Filtering

Convolution, Correlation and Fast Fourier Transform Techniques

Suitability of allocation schemes relating to some of the applications listed in Table 3-1 have been checked by using a timing simulator (implemented on the B5500 computer) which accepts an input program written in the ILLIAC IV Assembly language. The input program is augmented using pseudo opcodes which control program sequencing. These pseudo opcodes are necessary since the simulator does not maintain any updated data sets existing in the ILLIAC IV memory during program execution. As a result, the outcome of comparisons, which normally govern transfers of control, have to be explicitly given by the user (with pseudo opcodes).

The simulator assigns a storage location to each instruction, as does the assembler for ILLIAC IV, and times the fetching and execution of each as the program is run. Records are obtained of any delays encountered in the execution such as the advanced station delayed by no instruction, the final station delayed by no instruction, and the final station delayed by memory in use. A detailed printout can be obtained for each instruction as it is executed. However, this may be supressed in favor of a summary of the total running time, delays and memory usage. A sample summary, reproduced in Table 3-2, shows the results obtained using a General Circulation Model Code. This code is designed to simulate the behavior of the earth's atmosphere and is described later in the section.

Table 3-2. General Circulation Model Code

	Clocks	
Elapsed Time	47, 554	1.981 msecs
ADVAST Delays		
No Instruction	1, 420	59.167 μsec
Full FINQ	34, 721	1.447 msecs
Memory in Use	0	
FINST Delays		
Empty FINQ	125	5.208 μsecs
Memory in Use	2, 155	89.792 μsecs
PES Idle	2, 283	95.125 μsecs
Memory Use		
FINST and PES	15, 408	33.320%
ADVAST	0	0
Instruction Fetch	2, 496	5.236%
Input/Output	0	0

For ILLIAC IV codes run on the simulator it is found that the time required by ADVAST to process all the instructions is less than the time required by FINST. This is a necessary (but not sufficient) condition for the PE array to be busy all the time. It is also found that a FINQ capacity of eight instructions results in the almost complete overlapping of the ADVAST operations by FINST operations. The result of these conditions indicates that the typical usage of the array is very efficient.

TYPE OF PROBLEMS

Matrix Operations

Matrix methods are widely used for analyzing problems in engineering, physics, statistics and economics. The matrix operations of addition, multiplication, inversion and of determining the eigenvalues and eigenvectors are therefore of fundamental importance to a variety of ILLIAC IV users. A few sample applications are multichannel filter design, linear programming, vibration and flutter analysis of engineering structures, and statistical calculations. ILLIAC IV is well suited for carrying out the calculations involved in these operations as is illustrated in the next paragraph for the case of matrix inversion. Linear programming provides an example of the specific application of these operations with the added requirement of sparse matrix manipulation techniques.

Inversion

An ILLIAC IV code that has been run on the simulator finds the inverse of a matrix by Gauss-Jordan reduction. The matrix to be inverted is stored skewed (with row 1 stored across the array, starting in PE #1 and row 2 stored similarly but starting in PE #2 and so on). The pivoting operation is carried out by routing the pivot row around the array a distance of one PE at a time. In each location the point row is aligned, element for element, with another row of the matrix. It is scaled by the element from the pivot column of this row and subtracted from that row, reducing the element on the pivot column to zero. The required pivot column element is broadcast to the PE's.

In this way the original matrix is reduced to a unity matrix and the same operations performed on a unity matrix produce the inverse of the original matrix. The inverse is stored on the zero columns of the original matrix as they are produced. The time to invert a 500 X 500 matrix using this method is approximately one second which represents a speed up by a factor of about 6000 over the IBM 7094.

Linear Programming

Linear programming problems are an application of the matrix operations performed by ILLIAC IV. A detailed code, in ILLIAC IV assembly language, for solving linear programming problems by the revised simplex method indicates that a problem with 1,600 variables subject to 600 constraints and with a 10 percent sparseness ratio in the matrix of the constraints would require about one millisecond for each iteration on ILLIAC IV. This is about 6000 times faster than the estimated two seconds required on an IBM 7094 for the same operation. In this code the inverse of the basis is stored explicitly in the ILLIAC IV memory and is a dense matrix. The matrix of the constraints is sparse and is stored packed. For large scale linear programming problems the product form of the revised simplex method appears very promising for ILLIAC IV.

Linear programming is the major technique being used to optimize large activities. Some of the many applications are military logistics, resource allocation, economic models, agricultural systems, transportation networks, and production facilities scheduling. The size and speed of ILLIAC IV make possible the complete solution of large problems which previously could only be handled by piecemeal sub-optimization techniques.

Partial Differential Equations (PDE)

An important application for ILLIAC IV is the solution of parallel differential equations by finite difference methods. Typically in such methods the solution is obtained at a net of mesh points defined throughout the region (in space and time) of interest. The basic advantage that ILLIAC IV has when applied to these problems derives from the fact that the calculations of different mesh points are identical and can be carried out in parallel. All PE's can work simultaneously on different mesh points.

Storage

The best method to distribute the mesh points among the PE's depends on the particular solution technique adopted and also on the boundaries of the region being considered. For example, a vertical boundary which requires special treatment could slow down a computation by a factor of two if stored entirely within a single PE. However, this can be handled in a fully parallel way if the mesh is skewed so that the vertical as well as the horizontal boundaries are spread across all PE's.

General Methods

For elliptic PDE's successive overrelaxation and single line overrelaxation methods are straightforward to implement on ILLIAC IV. Particular attention has been given to alternating direction implicit (ADI) methods which work in two and three dimensions and are also available for the parabolic equations governing heat transfer. An example of the use of ADI codes is in neutron diffusion calculations for nuclear reactors. Two ADI codes have been run on the ILLIAC IV timing simulator with most satisfactory results. The mesh is stored skewed to allow scanning by rows and columns. The indicated time for a complete double sweep of a 64 × 64 mesh is 0.85 millisecond on a one-quadrant ILLIAC IV using a 64-bit word length. This corresponds to an average time per floating operation (elapsed time divided by total numbers of floating operations performed) of 8.65 nanoseconds for one quadrant. This time is over 500 times faster than a FORTRAN code for the same problem run on the CDC 6600 for which the elapsed time was 437 milliseconds and the average time per floating operation was 4.4 μsecs.

A code for solving Poisson's Equation by using Fourier Analysis, a fast method due to Hockney*, has also been shown by the simulator to run efficiently on ILLIAC IV. This method is even faster than that of the ADI codes and has applications in the study of electron devices with simple geometries and in atmospheric turbulance studies.

Numerical Weather Prediction

One way to predict large scale phenomena in the earth's atmosphere is to set up the equations governing the atmosphere and to solve the resultant initial value problem. The equations involved are based on the laws of fluid dynamics and thermodynamics as applied to the earth's atmosphere, but treated as a compressible fluid subject to radiation. The initial state may be determined, for example, by radio-sonde ballons equipped with telemetering equipment.

The amount of data and the length of the calculation are such that both the storage capacity and the speed of ILLIAC IV can be exploited. In a particular sample model, outlined by NCAR,** the state of the atmosphere at any time is determined by the values of 10 variables (e.g., components of wind velocity, temperature, pressure ratio, and water-vapor mixing ratio) at each point in a grid of 81,920 mesh points defined throughout the atmosphere. The mesh points are considered at 10 vertical levels and the total number of variables involved is 819,200. These variables may be formatted in 32-bit word lengths so that the whole model (at one time step) could be totally

*R. W. Hockney, "A Fast Direct Solution of Poisson's Equations Using Fourier Analysis," Jan. 1965, J. ACM.

**National Center for Atmospheric Research, Boulder, Colorado contained in the ILLIAC IV memory if four quadrants are assumed. On a single quadrant machine the data would have to be flowed through the memory from disk to disk.

A code written to update an 8-variable, 55,000 mesh point model through a one-time step indicates that two milliseconds are required on a one-quadrant ILLIAC IV to update the data for two of the circles of lattitude (one in the northern hemisphere and one in the southern hemisphere). Table 3-2 is the output of the simulator for the execution of this loop. The results indicate that the usage of the ILLIAC IV is efficient and the percentage of time the PE's are idle due to an empty FINQ is negligible.

A simplified benchmark problem of the same type required 0.404 millisecond to update two circles of lattitude. This represents an increase in speed by a factor of more than 500 over a FORTRAN code for the CDC 6600 for the same problem which requires 223 milliseconds. The average time per floating operation on ILLIAC IV (i.e., one quadrant, 32-bit word length) was 5.80 nanoseconds for this benchmark problem. For the more complex model this average was 8.17 nanoseconds.

Weapons Effects

For hydrodynamics calculations a two-dimensional code for one material has been written and run on the simulator. The method is a continuous flow version of the Particle-in-cell Method (PIC). This two-dimensional code runs efficiently with an indicated FINST-delayed-by-empty-FINQ time of about one percent of the total. The estimated time to treat 256 cells (one four-quadrant machine) is about 160 μsec. This method can be extended to 3-dimensional regions and also to multimaterial problems. This type of code finds application in the analysis of the effects of atmospheric nuclear blasts.

Another method representing a large mesh elastic-plastic two-dimensional underground shock code was run on the simulator with about the same efficiency (one percent PE idle time). Such codes have application in studies of the underground effects of nuclear blasts.

Nuclear Reactors

An important application of elliptic partial differential equations is in multigroup neutron diffusion calculations arising in nuclear reactor design. A double iteration procedure is used comprising an outer (power) iteration and an inner (flux) iteration.

The ADI method, already discussed, is widely used to handle this flux iteration. The computational speed of ILLIAC IV will permit solutions to three-dimentional neutron diffusion problems and to the transport equation.

Signal Processing

Processing the data generated by arrays of sensors is a relatively new computational problem and one which ILLIAC IV is ideally suited to handle. This is due to its ability to process the large data rates involved and to the array organization of its computing section. Examples which bear upon phased array radar applications and in seismic arrays are briefly discussed below.

Seismic Arrays

In a seismic array, the sensors are seismometers, and the purpose is to monitor teleseismic disturbances. A very large array of this type is LASA (Large Aperture Seismic Array) located near Miles City in eastern Montana. This array consists of 525 seismometers arranged in 21 subarrays of 25 each. The ultimate goal of the LASA is the capability to classify small teleseismic disturbances as natural events or man-made explosions.

Two types of processing for handling the analysis of the data are:

1. "Multi-channel filtering" or "filter and sum processing." The weighting applied to each seismometer is a function of the frequency (the weights are actually filters).

2. "Beamforming" or "delay and sum processing" weighted or unweighted. The weights are constants or are unity and represent conventional tapering of the phased array.

These functions are readily implemented on ILLIAC IV and the speed of the machine can be used to achieve greater resolution and surveillance capability from the seismic array.

Radar Data Processing

The computational parallelism and increased computational speed of ILLIAC IV can be applied to handling surveillance data provided by a phased array radar. The major functions of a large phased array radar for urban defense have been programmed for ILLIAC IV to demonstrate how ILLIAC IV can satisfy this type of application. The major functions are:

1. Radar beam forming and control,
2. Scan
3. Designation (or filter targets out of clutter or noise)
4. Tracking.

Functions 1, 3 and 4 have been programmed for a single quadrant ILLIAC IV system. The conclusion is that all functions could be handled efficiently with time available for diagnostic, more radar functions, and more complex tracking functions.

This application is ideal for ILLIAC IV as a large phased array radar is susceptible to the network array computer approach. Also, the large data rates involved can be handled by such a computer whereas they exceed the capabilities of single processor machines.

Fast Fourier Transforms

A code for the Cooley-Tukey algorithm, which has been written in ILLIAC IV assembly language and run on the simulator, represents a fast method for computing Discrete Fourier Transforms (DFT). The running time for N=4096 (where N = number of sample points) is 0.73 millisecond for a 64-bit word length. This compares with 2.95 seconds required by one implementation of the algorithm on the IBM 7094. The running time for this algorithm is proportional to $N \log_2 N$ when N is a power of 2. For ILLIAC IV the value of the constant of proportionality is 14.79 and 9.5 nanoseconds for 64-and 32-bit word lengths, respectively. The corresponding value for the IBM 7094 implementation already mentioned is 60 μsec.

Known applications of the Cooley-Tukey algorithm include computation of power spectrun and autocorrelation functions of sampled data, simulation of filters, and pattern recognition using a two-dimensional form of the DFT.

SECTION IV

HARDWARE DESCRIPTION

GENERAL

The building block structure for the ILLIAC IV system is totally modular to provide for the flexibility required in expanding the system. This same modularity also permits removal of the equipments for minimum system requirements. A typical equipment arrangement plan for the ILLIAC IV system is shown in Figure 4-1.

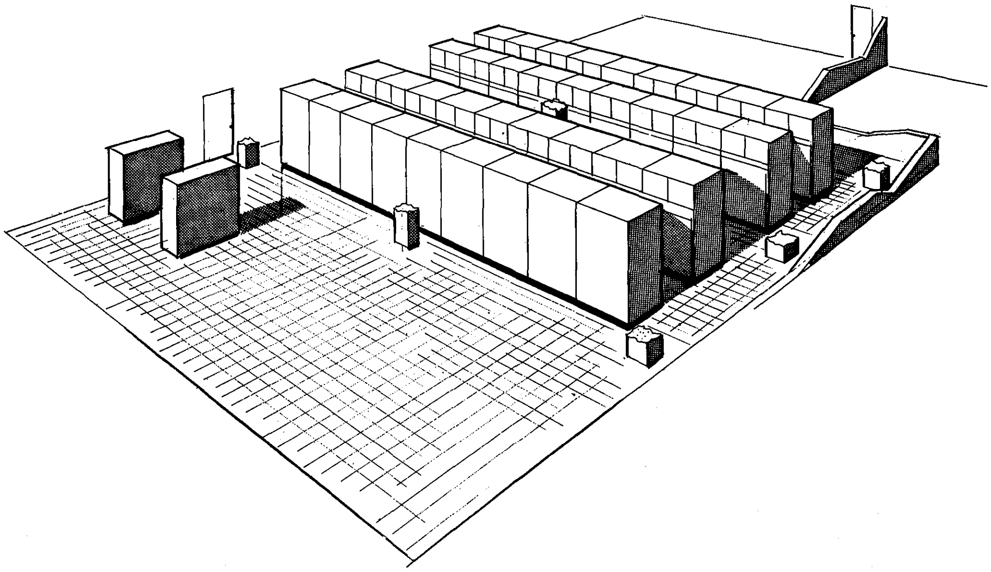


Figure 4-1. Typical ILLIAC IV Equipment Arrangement

This modular design is implemented by packaging eight individual Processing Elements (PE) and eight thin-film memories (PEM) in a single cabinet. Eight of these cabinets are bolted in a row together with a ninth cabinet housing the Control Unit (CU), thus forming a quadrant. Four, nine cabinet rows are distributed about the room in which the system is assembled. The Disk File Subsystem, its buffer, and the B6500 computer are located in the same area.

The B6500 computer, the PEM memory, and the bulk memory system represent adaptations of existing Burroughs equipment for the ILLIAC IV application. All of the logical, control, and memory storage functions are performed using various fabrication configurations such as:

Multi Medium Scale Integrated (MMSI) arrays

Thin film microcomponents

Multilayered printed circuits and printed backplanes

The microelectronic hardware capability of the ILLIAC IV system is described in the following paragraphs.

MICROELECTRONIC HARDWARE

The microelectronic circuit techniques used to implement the ILLIAC IV logic represent the first practical use of Multi Medium Scale Integrated (MMSI) packaging. This innovation in circuit design and fabrication has been made possible by Texas Instruments, Company, a major subcontractor to Burroughs Corporation.

More than 80 percent of the system logic for ILLIAC IV is implemented through the use of MMSI chip arrays. These arrays are used extensively in the Processing Elements (approximately 175 per PE). Their utilization reduces power and space requirements, and increases system speed. Further details in the application of these integrated electronics to the Processing Elements are described in the following paragraphs.

Logic Gate Partitioning

The ECL gates in the Processing Element are partitioned into 64-pin packages in such a manner that judicious use of "internal" gates is made for intra-chip and intra-package connections. These gates consume less power and their speed is enhanced since they are not required to drive an external load such as a transmission line. For example, consider the Carry Propagate Adder (Figure 2-8). By combining the Adder and First Level Look Ahead stages into one package, several inter-package paths are eliminated.

Processor Element Circuit Considerations

The high speed circuit performance necessitated by the ILLIAC IV system requires the use of circuits with propagation times of approximately 2.5 nanoseconds and capable of driving transmission lines. The emitter coupled logic (ECL) circuit shown in Figure 2-9 has been used as the basic gate for design of the arrays for the system.

The design of the basic gate was accomplished using test bars to define the design tradeoffs between speed, power dissipation and logic levels. The circuit dissipates approximately 30 milliwatts of power per gate and is capable of driving 50 ohm transmission lines to reduce system noise. The basic transistor used in the arrays has been designed using emitters with a width of 0.4 mil and only 1.0 mil long. The logical voltage swing is 900 mv and is symetrically centered about ground.

Because the performance of the circuits is related to the electrical and thermal characteristics of the package, an extensive engineering analysis was required in these areas to insure that final circuit design and manufacturing would be compatible with expected system performance objectives.

Processor Element Array Packaging

The advent of the integrated circuit has contributed much toward more efficient, high density packaging. In order to fully utilize the advantages of the ECL high-speed switching circuits, another step beyond the use of the conventional integrated circuit has been taken for the ILLIAC IV Processing Element. A multi-chip array packaging technique is utilized. Three to four monolithic chips, approximately 120 mil square, containing 15 to 20 gates, are alloyed onto a 1" X 1" ceramic substrate as shown in Figure 4-2. Connections between chips and to the multilayer printed circuit board are made by a system of 64 lead wires, silk screen land pattern on the ceramic, and thermocompression bond wires connecting the chip to the land pattern.

As many as 80 gates may be contained in a single 64-pin package. This package, occupying approximately 2 square inches of board area, is equivalent to 20 14-pin dual-in-line packages on a 4 X 5 inch card.

System Packaging

In order to connect the high speed 64-pin package arrays, a multilayer printed circuit board technique utilizing strip-line transmission lines is used. Terminated lines must be used for transmissions over more than a few inches in order to prevent severe ringing and reflections which could otherwise result. 180 64-pin packages will be mounted on four 10" X 20" multilayer boards as shown in Figure 4-3. An effort has been made to place

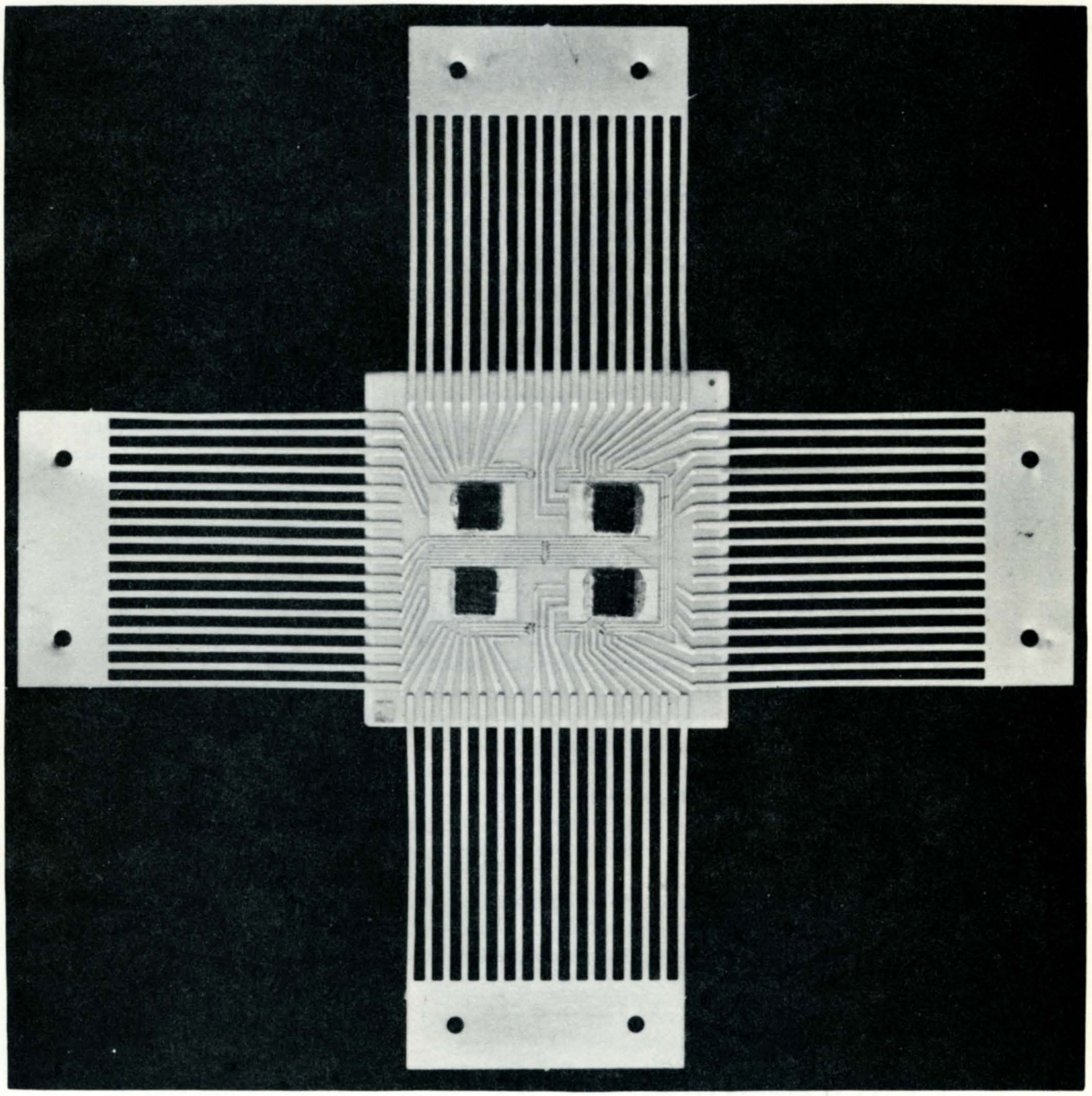


Figure 4-2. 64-Pin MMSI Chip Array

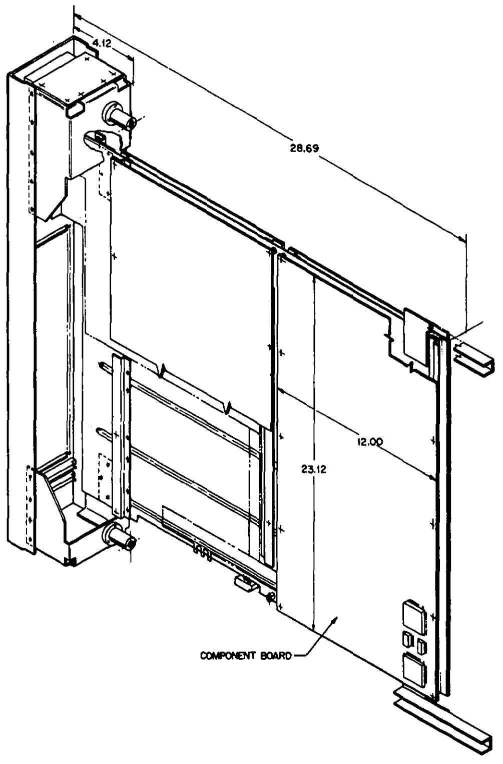


Figure 4-3. Multilayer Circuit Board

all modules associated with a critical algorithm on one board in order to minimize interboard wiring and associated delays. "Stick" boards are used for mounting termination and pull-down resistors. This technique utilizes otherwise wasted space, and for the Processing Element application, the use of discrete resistors rather than resistor modules is more efficient and versatile. The boards also form a cooling channel for the air cooled system. All components are flat mounted such that access.to the rear side of the board is not required for component replacement.

There are four buried signal layers and two surface signal layers in the multilayer printed circuit boards. These signal layers, separated by voltage and ground distribution planes, form 50-ohm transmission lines. The lines are matched or terminated in their characteristic impedance to eliminate reflection.

Processor Element Packaging

The PE occupies 1270 cubic inches. The same system, packaged in a conventional manner with dual-in-line packages (2 to 7 ECL gates per package) would occupy approximately 5200 cubic inches. The use of complex arrays has therefore permitted a 4:1 improvement in volume.

The Processing Element frame is so designed that the components are accessible without separating the board from the frame. When maintenance is required, the Processing Element is removed from the Processor Unit Cabinet and repaired at the bench. To enhance reliability and for conservation of space, the boards are interconnected with soldered wires rather than connectors.

ILLIAC IV THIN MEMORY

Introduction

The ILLIAC IV thin film memory is an amalgam of techniques developed for three separate memory systems. These systems are:

B8500 Memory	–	A 16K-word memory, 52 bits/word, DRO, 500-nanosecond read/restore cycle time.
B6500 Memory	–	A cost-reduced version of the B8500 memory which provides 8K words of 52 bits/word and is otherwise the same as B8500.
High-Speed NDRO	–	A 256-word memory, 200 bits/word, 50-nanosecond read cycle time.

The memory operates in a destructive readout mode (DRO) with a read/restore cycle time of 250 nanoseconds. The organization is linear-select with two 64-bit words per address line. Thus, there is a 32 X 32 address selection matrix and a 128-bit memory information register (MIR). By means of selection gates associated with the MIR, final address decoding is accomplished for selection of the desired 64-bit word from the two words read out. In addition, provision is made for selection of 32-bit words at the MIR when the system is operating in half-word mode. All logic circuits associated with memory operation, as distinguished from memory drivers and sense amplifiers, are implemented in ECL family of logic.

Organization

The thin film memory (Figure 4-4) is organized into two main sections: the memory frame and what is referred to as "associated electronics." The memory frame contains the film storage elements, and also includes the memory electronics. The remaining associated electronics serve the function of address selection, data transfer and the memory timing and control.

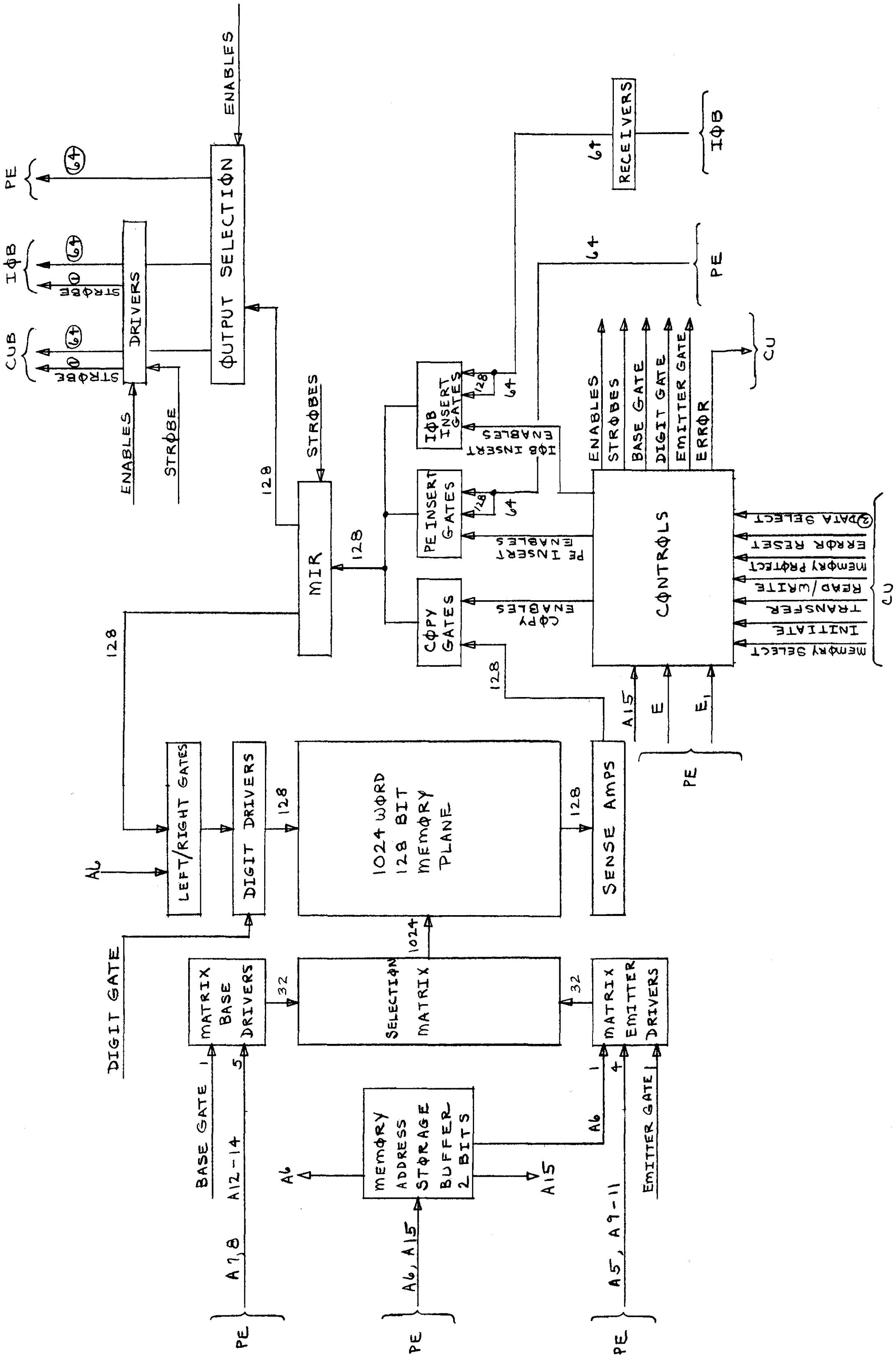


Figure 4-4. ILLIAV IV Memory Block Diagram

Memory Electronics

Word Matrix – Every word on a frame is addressable through a matrix that contains 1024 selection transistors. Each transistor connects to a one-word line matrix. The matrix is arranged into 32 rows and 32 columns and is driven from 32 matrix emitter drivers and 32 matrix base drivers.

Matrix Base Driver – The matrix base drivers connect to 32 transistor bases in the selection matrix.

Matrix Emitter Drivers – The matrix emitter drivers connect to 32 transistor emitters in the selection matrix.

Sense Amplifier – The sense amplifiers amplify the film switching signal obtained during word interrogation. Its output connects to the copy gate network.

Digit Drivers – The digit drivers supply positive or negative polarity information current to the memory frame. Its output comes from the "left/right" gate network via the memory information register. The polarity of information current controls the ONE or ZERO state of the thin film cell at the intersection of the particular digit line and the selected word line.

Associated Electronics

Memory Information Register – The memory information register provides 128 bits of temporary storage for data transfer between the PE and IOS and the memory plane during a write operation, and the memory plane and the PE, IOS or CUB during a read operation. The set input is received from a PE insert gate, an IOS insert gate, or a copy gate. After the read portion of the memory cycle, the data in the MIR is written back into the memory plane during the restore portion of the memory cycle.

Left/Right Gates – The left/right gates determine whether true or complement data is written into the memory, depending upon the stack location of the addressed word. The sense conductors are transposed with respect to the digit lines (for digit write noise cancelling purposes) so that signal reversal exists in only one half of the memory. Therefore, true or complement data is written into the memory enabling true data to be always read out to the MIR.

Output Selection Gates – The addressed 64-bit word is selected from the two words read out of the memory plane by the output selection gates.

Control Section – The timing card contains all the necessary control logic for the associated electronics.

Physical Description

The ILLIAC IV memory, that is to say the Processing Element Memory (PEM), is combined with the Processing Element (PE) to form an integral modular package called the Processing Unit (PU) assembly. The PEM and PE are readily separable as individual subassemblies without the use of a soldering iron. The PEM and PE input/output connector interfaces are distinct and remain with the corresponding assemblies when the units are separated.

The Processing Element (PE) consists of four multilayer printed-circuit cards mounted on an open frame. The boards are mounted on a back-to-back configuration with the circuit components facing to the outside so that they can be replaced without removing the cards from the frame.

Connectors mounted on the frame provide a pluggable interface for the PE and PEM interconnections to facilitate separation of the PE from the PEM. Connection from the multilayer board to the connectors is made via a round wire flat cable, coax and twisted pair as required. The PE is, in itself, a modular assembly suitable for testing as a unit.

The Processing Element Memory (PEM) completes the PU assembly. The memory is constructed using the same fabrication techniques presently being used in the production of the thin-film memories for other Burroughs projects such as B8500. The general physical design of the ILLIAC IV memory varies from the B8500 memory mainly in the areas of circuit design, detail of the artwork, and the number of substrates in the plane. The PEM, like the PE, is a modular assembly complete with its own distinct input/output interfaces.

Frame Assembly

The basic frame assembly (Figure 4-5) is 46" high, 38" deep, and 4-7/8" wide and is formed by 1/8" thick aluminum extrusions held together by corner gusseting and welded joints. The PEM casting, when mounted, forms an integral part of the structure, thus eliminating the need for additional rigidizing members.

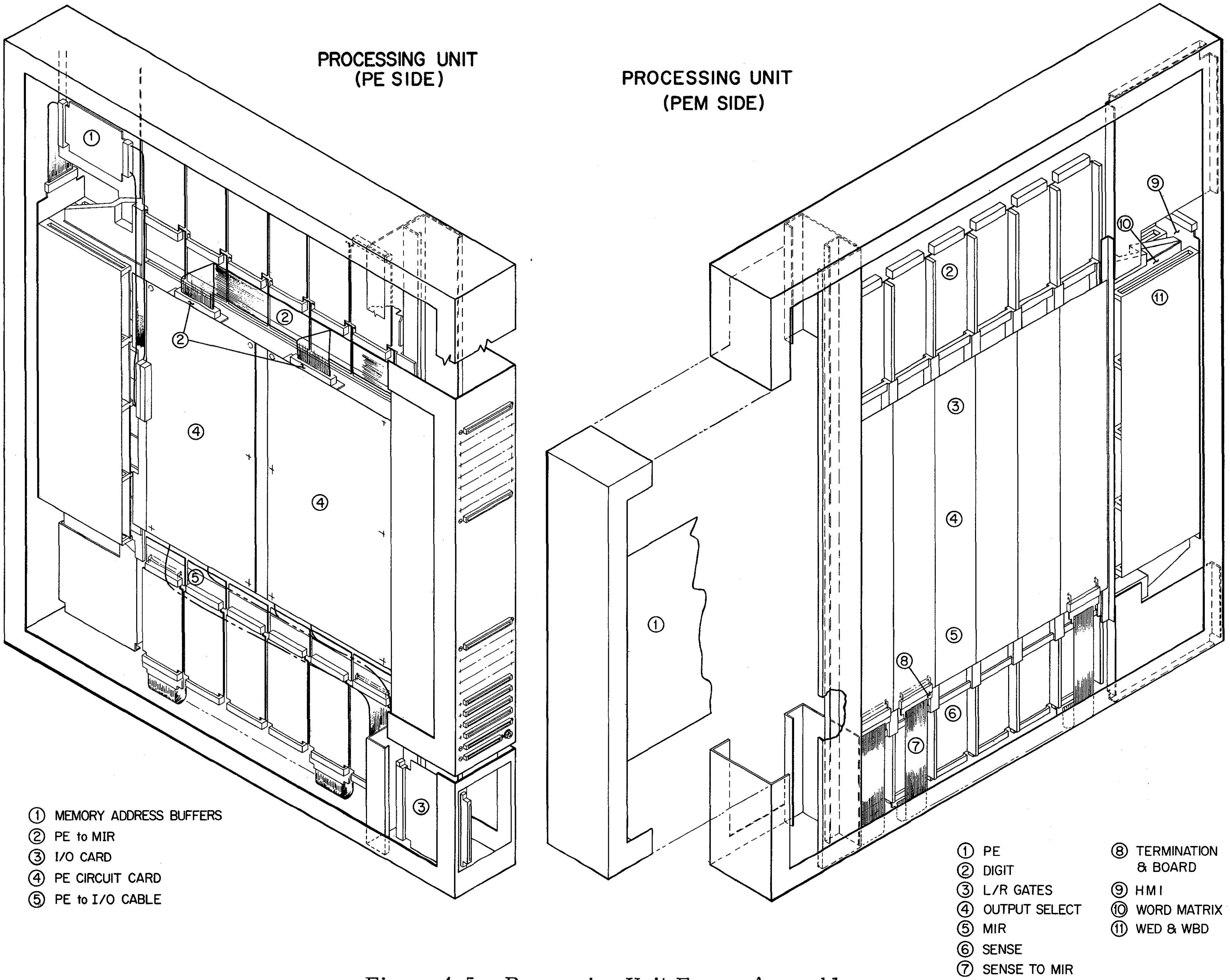


Figure 4-5. Processing Unit Frame Assembly

The PE modular frame, containing its own extrusion for connector mounting, is mounted on one side of the PEM. Power interface is provided at the top, PE interface at the middle, and PEM input and output interface at the bottom. The aluminum extrusions on the top and bottom of the package have cutouts for air circulation and side covers are provided to direct the airflow path through the package. Access holes are also provided in the front extrusion for electrical probing of the PEM matrix interconnect board.

Memory Plane

The construction of a typical memory plane assembly is shown in Figure 4-6. The memory plane is built up on an aluminum casting which controls the flatness, dimensional accuracy, and rigidity of the final assembly. The cross section of the memory plane is shown in Figure 4-7.

Etched word-line tapes and sense-line tapes are assembled into a lattice which is the heart of the memory. The terminal ends of the sense, digit and word lines are solder plated for termination to printed circuits. The lattice is laminated into a permanent assembly using a 1/2-mil sheet of high temperature thermoplastic. Three-mil glass substrates are precisely located and laminated to the lattice using a low-temperature bonding adhesive. Special care is required to keep the magnetic film in intimate and consistent contact with the lattice.

One inside and two outside ground plane assemblies complete the buildup. The outside ground planes are assemblies consisting of a 2-ounce copper sheet laminated to 0.062-inch glass epoxy insulators. The inside ground plane is a 1/6-inch copper sheet with 2-ounce copper flaps soldered at the edges.

The end-around functions are mechanized using an etched section of laminate wrapped around a glass-epoxy backing board. The sense crossover digit-feedthrough functions are accomplished with three printed circuit boards laminated together. Sense and digit lines are terminated in printed circuit boards which provide the connector interface to the drive and sense circuitry.

B6500 INFORMATION PROCESSING SYSTEM

Introduction

The B6500 Information Processing System serves as an external general purpose computer for controlling the 4-quadrant array of the ILLIAC IV system. It sets up the bulk memory for data transfers to and from the ILLIAC IV through a buffer I/O memory, and transfers data to and from the bulk memory and the input/output devices. In addition, it supervises the ILLIAC IV program runs.

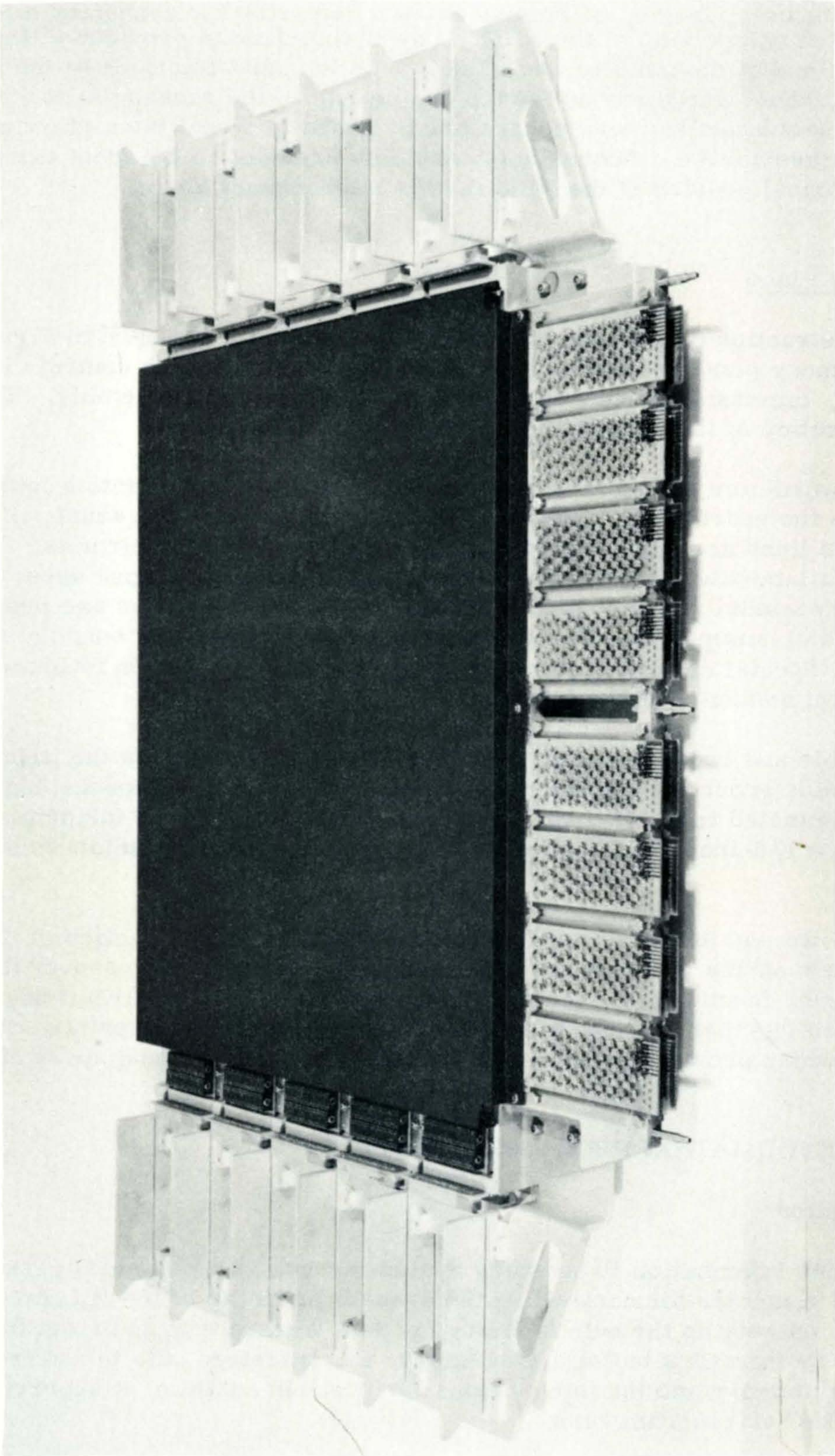


Figure 4-6. Typical Thin Film Memory Plane

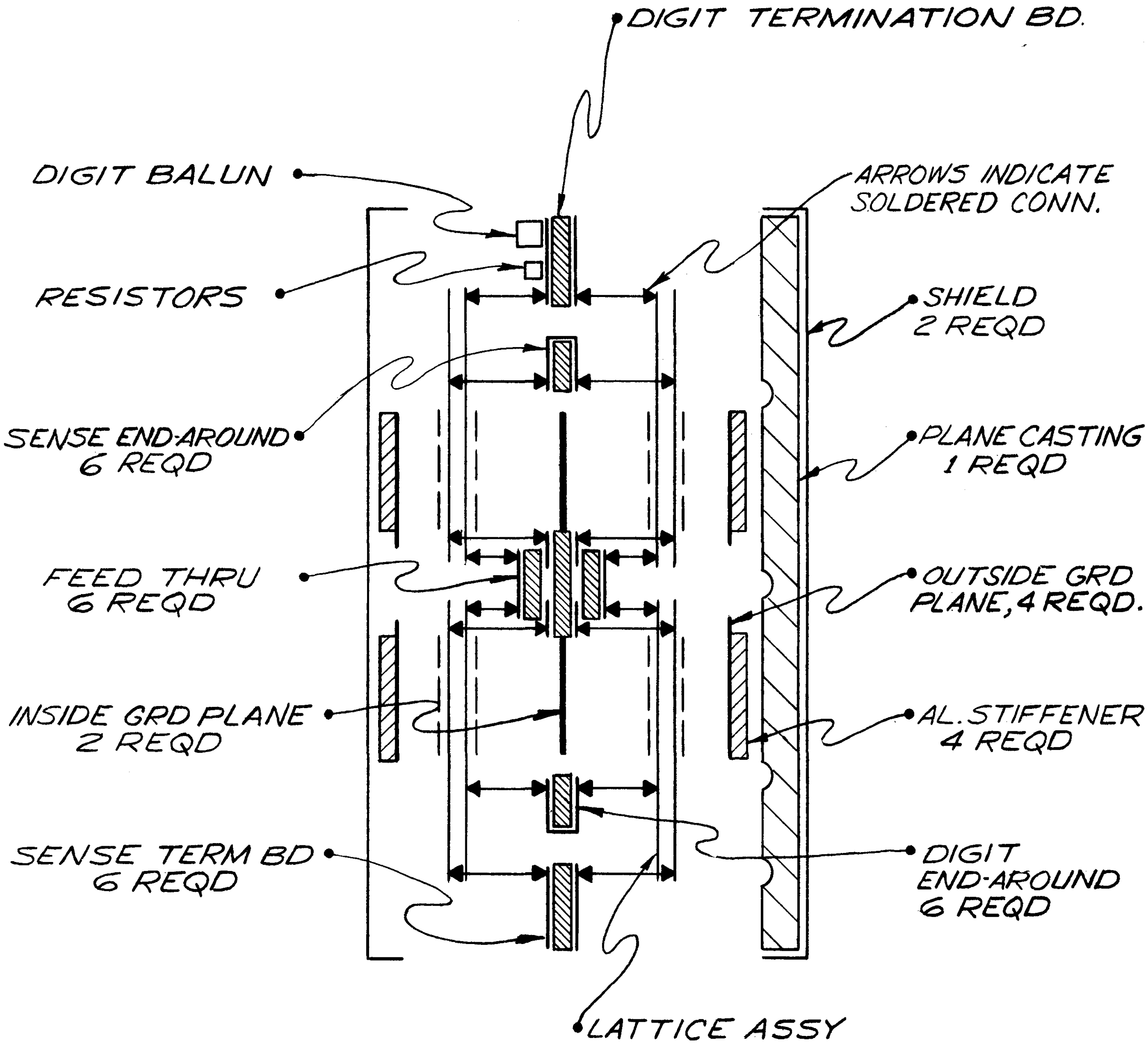


Figure 4-7. Cross Section of a Thin Film Memory Plane

The B6500 basic system consists of a central processor, an input/output multiplexer, and a thin-film memory system. System design is based on program-independent modularity, the ability to process programs on available equipment without reprogramming or recompiling, while, at the same time, making efficient use of that equipment. The primary operating characteristics of the B6500 system are:

- Incorporates monolithic integrated circuitry.
- Controls all input/output operations independently of each other and allows multiple simultaneous read/write complete operations.
- A clock rate of five megacycles.
- A Master Control Program (MCP) which provides for total system management, with direct communication to the operator only when operator action is required.
- Planar thin-film main memory with a 600-nanosecond cycle time per 51-bit word.

Functional Design

The functional design of the B6500 system is shown in Figure 4-8. This hardware design is integrated with the design of key software components, especially the Master Control Program operating system, to provide for optimum execution of an object program for any hardware configuration. Automatic compensation is made for changes in configuration. Therefore, neither computer system expansion nor the loss of components affect the ability of the B6500 to execute programs efficiently. For example, as a B6500 is expanded on the user's site to accommodate mounting workloads, programs are executed by the new configuration at greater speeds. Neither reprogramming nor recompiling is required. The Master Control Program (MCP) recognizes that a larger hardware configuration is available, and fully utilizes the new environment by an immediate re-allocation of memory, processor, I/O and peripheral resources to object programs.

Similarly, components may be removed from the system without destroying its ability to perform. As the schematic diagram shows, the logical functions of the B6500 are not disturbed by the removal of a memory module, a peripheral control channel, a peripheral device or (in dual-processor configurations) a processor, The MCP is able to recognize such omissions and perform its tasks accordingly. Thus, malfunctions do not necessitate complete system shutdown; the concept of "graceful degradation" allows operation to continue while corrections are being made.

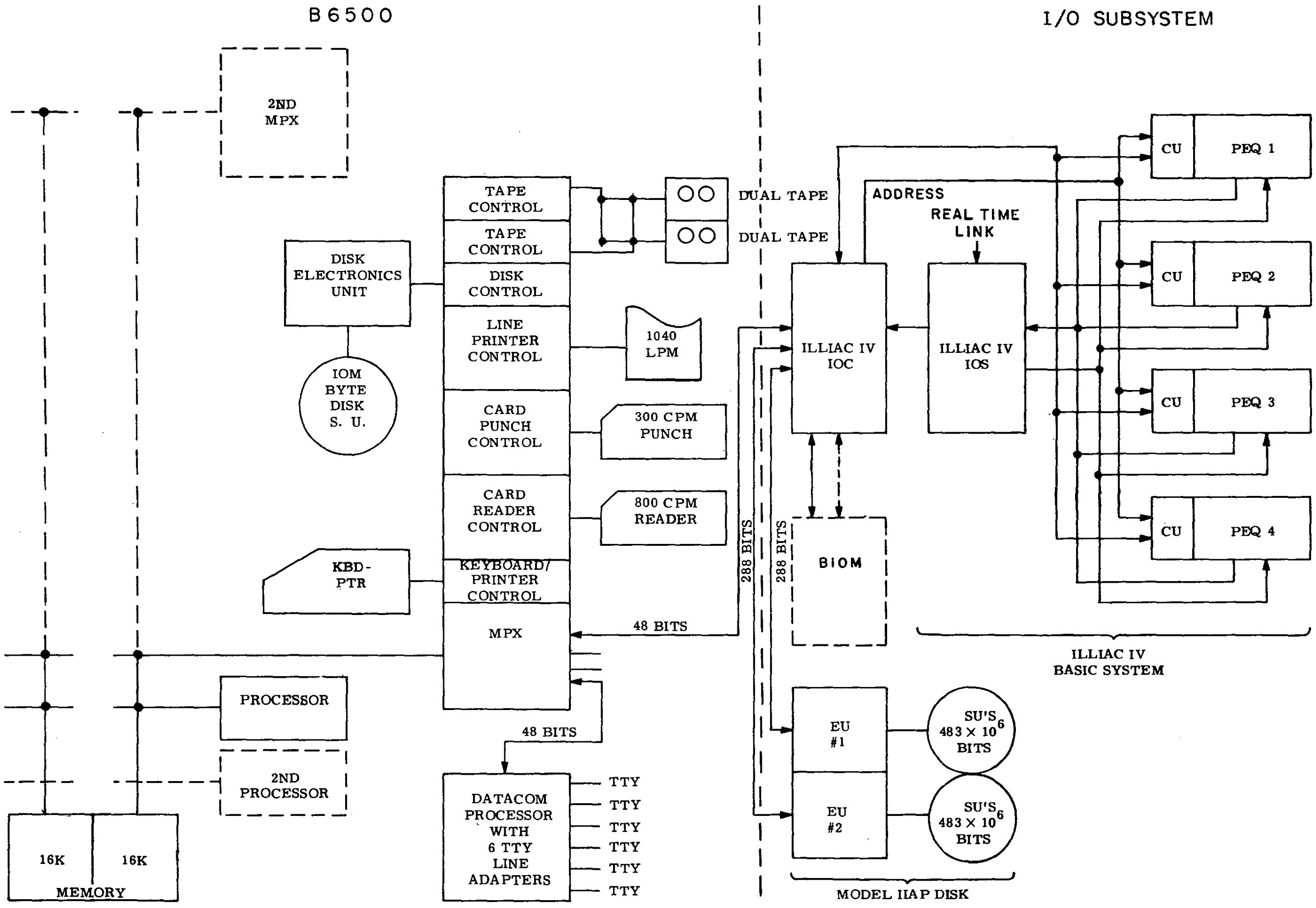


Figure 4-8. B6500 Interface with ILLIAC IV I/O Subsystem

The proprietary information contained in this document is the property of the Burroughs Corporation and should not be released to other than those to whom it is directed, or published, without written authorization of the Burroughs Defense, Space and Special Systems Group, Paoli, Pennsylvania.

Circuit Design

The clock rate of the B6500 is five megacycles, allowing extremely fast operation. Complementary transistor logic is eimployed throughout. This is the newest of several types of monolithic integrated circuits, and is the fastest, least expensive, most flexible type available.

A complete flip-flop is diffused into a piece of single crystal silicon 0. 040 inch across; and is, conservatively, 10 times faster than conventional discrete element circuits. Complementary transistor logic is also significantly less costly than comparably performing discrete circuits.

System Elements

Processors

The B6500 accommodates one or two processors, each of which can access main memory. Expanding from a single-processor to a dual-processor configuration, when a rising workload reaches adequate proportions, results in a significant increase in the computational throughput of a B6500 system at a modest increase in cost. The second processor can be installed on site. No reprogramming is required to take full advantage of the expanded system.

The processor design reflects its purpose to implement higher level languages and to function under MCP control. For example, the major registers and control flip-flops in each of the processors are designed to contribute to the systems multiprocessing capabilities. An automatic hardware stack provides ready access to operands as well as intermediate result storage.

An aggressive hardware method of detecting and servicing system interrupts contributes to the B6500's ability to process a mix of independent programs in an efficient manner. Under the constant, automatic management of the MCP, multiprocessing is the normal mode of operation for the B6500. With one processor in the configuration, multiprogramming is the method used. Dual-processor B6500 systems operate in a multiprogramming manner with either processor, and perform parallel processing when both processors are in operation.

Memory Hierarchy

The modular thin-film main memory of the B6500 has a cycle time of 600 nanoseconds for a 51-bit word, and is expandable to a maximum capacity of 524, 288 words. B6500 words contain 48 information bits, plus a parity bit and special purpose bits. Memory sizes available are in 16, 384 increments up to 524, 288 words.

Second level memory for the B6500 consists of a Burroughs disk file subsystem. The disk file's head-per-track design simplifies the task of large volume storage of both program and data segments, and makes possible the very fast access speeds essential for effective utilization of second level storage techñiques. The B6500 MCP automatically transfers program or data segments to the thin-film main memory as they are needed. Up to four transfer operations to or from the disk file subsystem can occur simultaneously through use of an optional Disk File Exchange.

Input/Output System

A major factor contributing to the B6500's multiprocessing capabilities is the design of the input/output system. The key to this system is the Input/ Output multiplexor. The input/output multiplexor and associated peripheral control modules are used to control transfer of data between the main memory and the peripheral equipment. The multiplexor may contain up to 20 peripheral control channels but can simultaneous execute instructions (received from the processor) for up to 10 peripheral control channels only. The sustained simultaneous operation of up to 10 high speed input and output units plays an important role in the B6500's multiprocessing power.

Connected to the multiplexor are the various peripheral control units, which control the operations of specific input/ output units.

ILLIAC IV DISK FILE SUBSYSTEM

Organization

The Disk File Subsystem (Figure 4-9) which serves as the mass memory for ILLIAC IV is an extremely high speed magnetic disk with a transfer rate of 500 million bits per second – 100 times the speed of the fastest commercially available units. The subsystem consists of an Electronics Unit and Storage Units with sufficient circuitry for reading or writing simultaneously on 96 tracks of one disk. Up to nine disks can be connected to an Electronics Unit. Common electronics such as registers for providing disk serial to control unit parallel conversion of the information, control logic, power, motor control and power are housed in the Electronics Unit.

The leading characteristics of the disk file are listed below:

Storage capacity per disk	78,796,800 bits
Maximum storage per EU (9 disks)	709,171,200 bits

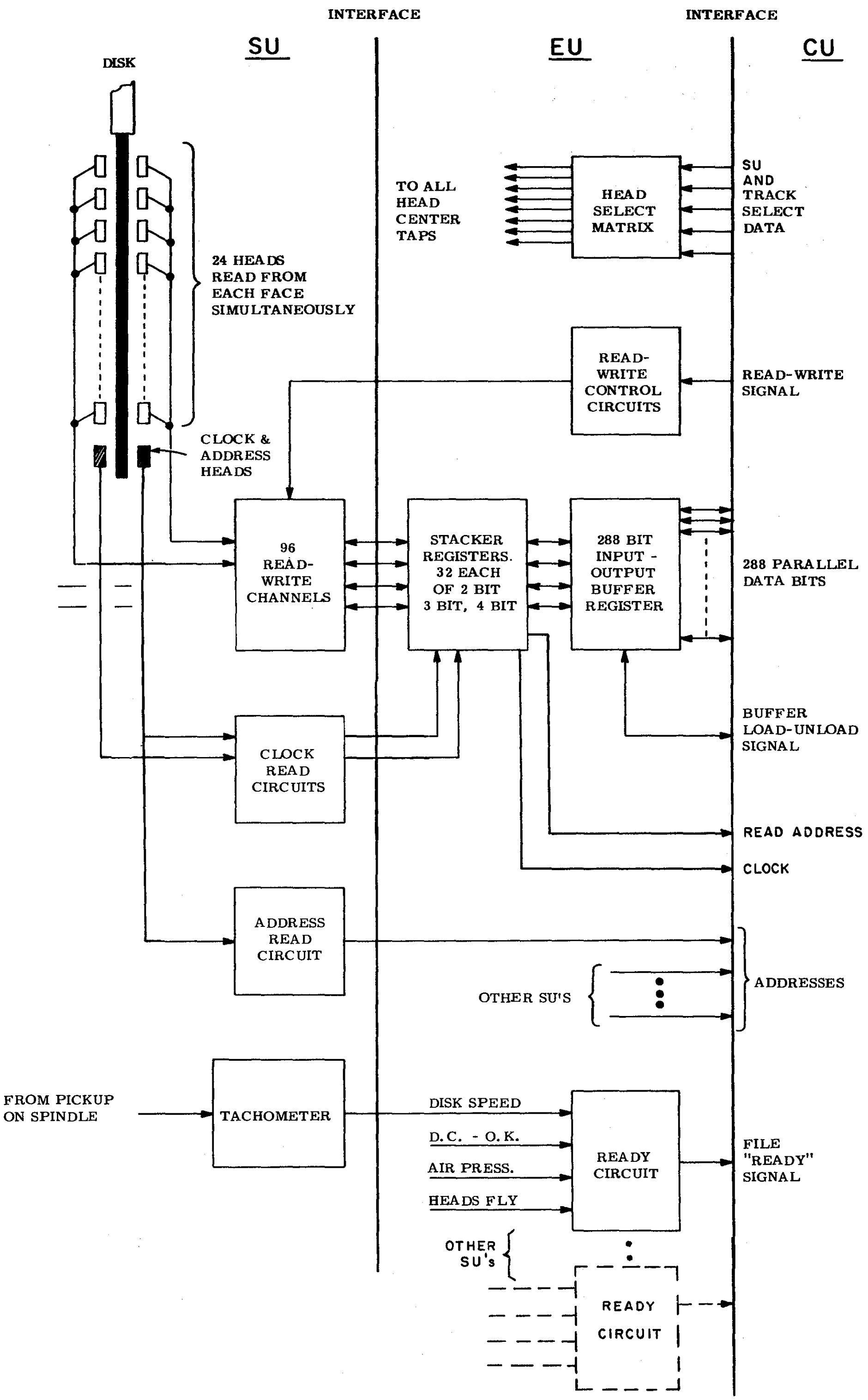


Figure 4-9. ILLIAC IV Disk File Subsystem Block Diagram

RPM	1500
Approximate transfer rate	500×10^6 bits per sec.
Average access time	20 msec
Maximum time to transfer contents of one disk	160 msec
Maximum time to transfer contents of 9 disks	1.76 sec

Functional Description

Disk - Track Layout

The disk - track format in the ILLIAC IV file is an expansion of the format presently used in the B8500 disk system. The tracks are divided into three frequency zones, with data being written into the zones at a frequency ratio of 4:3:2. The track layout consists of 192 active information tracks per disk face which are arranged in three zones of 64 tracks each as shown in Figure 4-10. The zone heads are wired such that 16 of the 64 tracks are selected at a time by one of four center tap drivers. The same center tap driver selects all 96 tracks on the two disk faces. A clock head, also located on each face, indicates segment location and provides timing pulses.

Read - Write Circuitry

The 96 selected tracks connect to 96 read-write amplifiers located at each disk. The read signals are transmitted to the EU for disk selection, amplification, and detection. Coax cables carry the logic levels and write information between the SU and EU.

Logic

Stack-up Registers – Data between the disk and control unit flows through stack-up registers which convert the parallel data from the control unit into serial data at the three different zone frequencies and vice versa. A total of 32 sets of stack-up registers are used to handle the 96 tracks and provide 288 bits in parallel to the control unit.

Clock System. The clock system consists of address and bit clock heads on each disk and associated amplifiers. There are separate clock amplifiers in the EU for each disk, and the addresses read from every disk are transmitted to the CU for queing purposes. During read or write, the

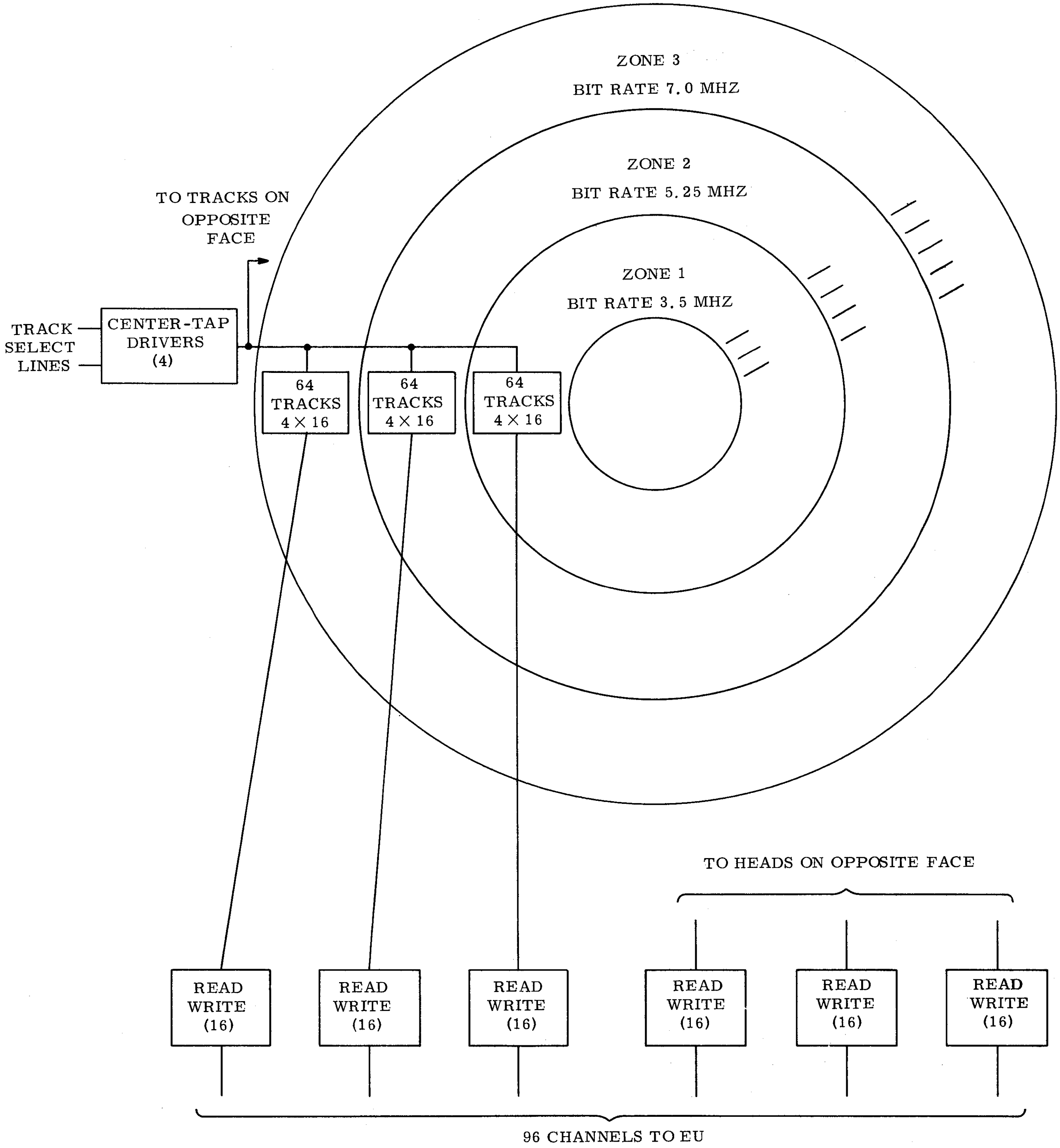


Figure 4-10. Disk-Track Layout

selected disk bit clock amplifier is connected to a clock generator in the EU to generate the three zone clock frequencies.

Control and Index Logic – The control section contains the necessary decoding logic, and during a read or write, determines from the address track where the segments start and end. The control section also contains index logic.

SECTION V
AVAILABILITY

INTRODUCTION

An important parameter in the design and development of the ILLIAC IV system is availability; that is, the percentage of time that a given system is actually available for operating and performing its intended mission. This measure of availability is a function of reliability and maintainability and is expressed in the following equation:

$$A = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}}$$

Where

A = Availability

MTBF = Mean Time between Failures

MTTR = Mean Time to Repair (The term, MTTR, includes the time to diagnose, locate, and correct the failure.)

From the equation it can be seen that the maximum availability is achieved through maximum MTBF and minimum MTTR. This can be also expressed as minimization of hardware failures and minimization of the time to correct a failed portion of the system. It is these two areas, reliability and maintainability, and their interrelationship which have been treated to optimize the ILLIAC IV system availability.

RELIABILITY

ILLIAC IV reliability is achieved through various means, some of which are discussed in the following paragraphs:

Through Integrated Electronics -- The recognition that the ILLIAC IV system would encompass a large count of electronic functions led to early emphasis for achieving minimal failure rates at optimum cost. One of the primary means for attaining this goal is through the application of integrated electronics. More than 80 percent of the system logic for the ILLIAC IV is implemented through the use of Multi Medium Scale Integrated (MMSI) chip arrays. These MMSI arrays are used extensively in the Processing Elements (approximately 175 per PE). Each MMSI array contains between 40 and 80 gates. In addition to the operational, cost, and space advantages, MMSI arrays provide a distinct reliability advantage over the smaller 3- to 5-gate integrated circuit (IC) used in third generation computers. Logic implemented through MMSI arrays will have a reliability improvement factor of at least 2 to 3 over the same logic implemented with IC dual in line packages. In addition, there is a significant reduction in the number of solder joints. Table 5-1 compares the implementation of a 10,000-gate Processing Element by MMSI's and IC's.

Through Component Selection -- Further assurance of system reliability is attained through emphasis on the selection of reliable components, especially when used in extensive quantities. This is borne out in the case of the pull-down resistors in the Processing Elements, where approximately one million individual devices are used. For this purpose, a high reliability metal film unit was chosen.

The connectors used in the ILLIAC IV system are of two basic types: pin and socket and printed circuit card edge connectors. Burroughs has made use of these connector types in military programs and has amassed quantities of information to substantiate their usage in ILLIAC IV. In addition, extensive investigations relating to connector plating finishes, particularly with regard to the card edge connectors, have been conducted to improve the life and reliability of these connectors. In all cases the connectors have been specified for the ILLIAC IV system application and have requirements similar to MIL-C-21097. This assures failure rates of the connectors which are equal to or better than those stated in MIL-HDBK-217A.

While the overall reliability of the ILLIAC IV system will reflect best commercial practices, particularly in the selection of component parts, much of the knowledge gained from the high reliability efforts achieved on Burroughs' military programs is applied to the ILLIAC IV system.

Table 5-1. Comparison of the Implementation of a 10,000-Gate Unit [1] with IC's and MMSI's

Item	14-16 Pin Integrated Circuits	64-80 Pin Multi MSI Chip Arrays
Gates/Package	5	70
Chips/Package	1	3
Bonds/Package	26	130
Pins Used/Package	12	50
Packages/Unit	2,000	145
Chips/Unit	2,000	435
Bonds/Unit	52,000	18,850
Package Solder Joints/Unit	24,000	7,250 [2]
P.C. Boards/Unit	150	4
Inter Board Connections	7,500	1,200 [3]
External Connections	500	500

[1] This unit is not representative of the ILLIAC IV PE but serves as a hypothetical device for comparison of the overall effectiveness of the Multi-Medium Scale Integration (MMSI) arrays. The actual average number of gates per package for the ILLIAC IV MMSI package is about 60 with an average of 3. 5 chips per package.

[2] The ILLIAC IV MMSI arrays also include silicon pull-down resistors within the package, thus further reducing the total number of solder joints and discrete devices per unit.

[3] These interboard connections will be accomplished through soldered flat cable in the actual ILLIAC IV PE.

Through Minimization of Connectors -- More than 90 percent of the electronics in the ILLIAC IV system is contained in the 256 Processing Elements and Processing Element Memory units. To minimize MTTR, these items, when failed, will be removed as a unit and repaired off line. This concept of "pull and replace" of major modules makes possible the minimization of connectors within these major modules. Moreover, connectors are entirely eliminated within the Processing Element and are only used in the Processing Element Memory on those subassemblies which contribute to the majority of the failures.

Through Controlled Environment -- The ILLIAC IV cooling system uses a closed air loop controlled to 25°C and 50 percent relative humidity. This controlled environment assures a more reliable performance of many of the components, especially in the areas of connectors and encapsulated devices. It has often been demonstrated that excessive humidity is a primary factor in degradation of contact resistance and in failure mechanisms involving corrosive actions. While the exit temperature from an individual cabinet may rise as much as 15°C over inlet temperature, the majority of the electronics is exposed to air at 25°C to 30°C. This prevents the occurrence of high thermal stresses and eliminates many of the failure mechanisms which are normally accelerated by high temperature conditions.

MAINTAINABILITY

Maintainability is achieved primarily during the design stage. The philosophy followed to guide the design for maintainability of ILLIAC IV was one of stressing rapid replacement where most failures may occur.

Through Pull and Replacement of Major Modules -- As previously mentioned, a pull and replacement technique is used throughout the ILLIAC IV system. More than 90 percent of the electronics of ILLIAC IV are included in the 256 Processor Units (PU), each of which contains one Processing Element and one Processing Element Memory. Accordingly, emphasis has been placed on the rapid isolation and removal of failed PU modules. This time has been estimated to be 0.29 hour which essentially becomes the MTTR of the entire system. Once a failed PU has been removed, it is immediately replaced and the failed unit is repaired off line. A similar method is employed, where possible, throughout the system for the power supplies, power regulators, and control unit electronic assemblies. This permits more than 90 percent of the system to be maintained on the basis of rapid pull and replacement when failures occur.

Through Efficient Off-Line Maintenance -- Off line repairs, in most cases, may be conducted on site. Test equipment is available for diagnosis of the Processing Element and Processing Element Memory to the card and/or component level. In the case of the Processing Elements the individual

MMSI arrays are replaceable. All components in the PE are immediately accessible without disassembly of the PE configuration. In the Processing Element Memory, those subassemblies having the predominant failure rates are pluggable units. Techniques are available for easy removal of the soldered assemblies. Only in the case of major failures within the memory planes or internal to the multilayer boards does repair of a module require factory personnel and equipment.

SPECIAL SYSTEMS/AVAILABILITY CONSIDERATIONS

The simple availability equation expressed earlier pertains only to the ILLIAC IV system operating all four quadrants available concurrently. The equation, however, does not consider that, during the repair time of a one-quadrant failure, there are three other quadrants available for use. It is possible for system users to prepare and schedule many programs which may be executed with one or two quadrants available. This efficient use of the total system potential results in a new availability expression:

$$A_s = \frac{\sum_{i=1}^{i=3} U_i + (N - r + 1)\,U_A}{\sum_{i=1}^{i=3} U_i + (N - r + 1)\,U_A + \sum_{i=1}^{i=3} \lambda_i + \dfrac{(r + a)\,!\;(\lambda_4 + \lambda_5)^{a+1}}{(r - 1)\,!\;(N - r + 1)^{a}\,U_A^{a}}}$$

Where: A_s = System Availability

λ = Failures per million hours

U = Unavailability of element

This expression is based on the ILLIAC IV system configured as shown in Figure 5-1. The total number of Processing Element quadrants (PEQ) with their associated Control Units is denoted by the letter, N. The required PEQ/CU combinations are denoted by the letter, r. The letter (a) represents the number of spare PEQ/CU combinations and may assume values of 0, 1, 2, and 3. In general, an availability significantly in excess of 90 percent can be expected when N = 4 and a = 1, 2, or 3.

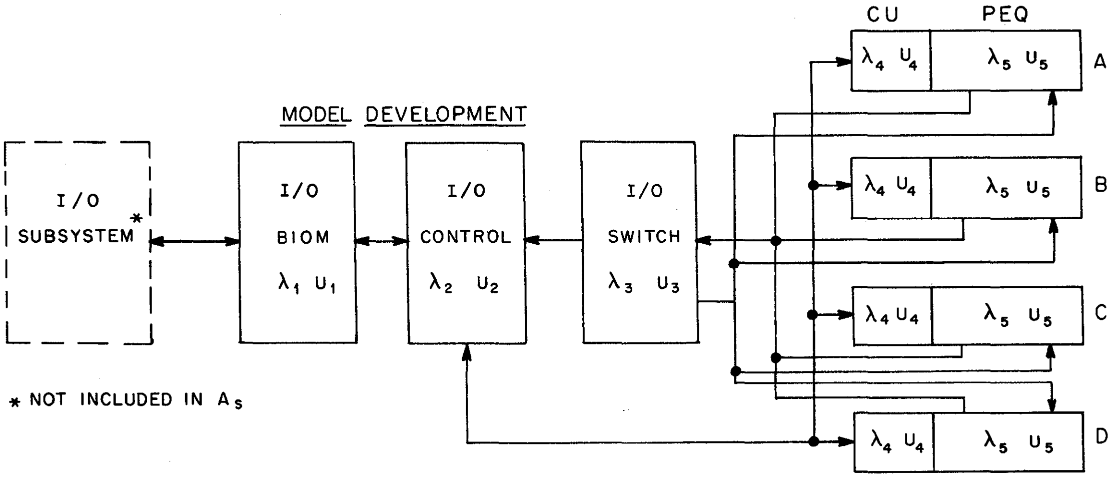


Figure 5-1. Reliability Block Diagram for the Basic ILLIAC IV

Another operational consideration which will optimize the use of the available system is through proper spatial and temporal program segmentation. A program operating when a failure occurs must be restarted and the previous system time is lost even though it was "available". As programs can be compressed or segmented so that they require shorter times for completion, systems use efficiency will increase.

The very high throughput of the ILLIAC IV system and its ability to operate in various combinations of quadrants permit system users to obtain a system usefulness approaching the systems availability.