B 720

PROCESSOR

Burroughs

FIELD ENGINEERING

TECHNICAL MANUAL

Printed in U.S. America

7-30-75

For Form 1077534

Burroughs believes that the information described in this manual is accurate and reliable, and much care has been taken in its preparation. However, no responsibility, financial or otherwise, is accepted for any consequences arising out of the use of this material. The information contained herein is subject to change. Revisions may be issued to advise of such changes and/or additions.

Correspondence regarding this document may be addressed directly to Systems Documentation, Technical Information Organization, Burroughs Corporation, 200 West Lancaster Ave., Wayne, Pa. 19087.

LIST OF EFFECTIVE PAGES

Page	Issue
Title	Original
ii	Original
iii	Original
iv	Blank
v	Original
vi	Blank
1-1 thru 1-20	Original
2-1 thru 2-113	Original
2-114	Blank
4-1 thru 4-3	Original
4-4	Blank

Printed in U.S. America

For Form 1077534

INDEX

Printed in U.S. America 7-30-75 For Form 1077534

B 720
PROCESSOR

SECTION
1

INTRODUCTION
AND
OPERATION

Burroughs

FIELD ENGINEERING

TECHNICAL MANUAL

Printed in U.S. America 7-30-75 For Form 1077534

INTRODUCTION

The B 720 Central Processing Unit (CPU) is the central operating and controlling element (system processor) used in B 720-series systems. Figure I-1 shows the general features of the CPU, and figure I-2 shows the general interface of the CPU in a system.

The CPU is unlike conventional processors in that the basic logic operations are organized into controlled building blocks external to the processor section. The logic in these building blocks is not like the normal hard-wired control logic used in conventional processors. Instead, the control logics are replaced by control signals generated by the firmware store section. This approach adds a great deal of flexibility to the processor and utilizes a minimum amount of hardware.

The CPU is a word-addressable unit with a semiconductor main memory that is expandable, in increments of 8K bytes, up to 96K bytes. The basic system uses a minimum of 32 bytes of main memory and 512 words of nanoprogram memory. There are expansion provisions for an additional 512-nanoword capacity. The main memory is a Random Access Memory (RAM), while the nanomemory is a Read Only Memory (ROM).

The internal clock pulse rate of the CPU is 1 MHz. Certain peripheral controls or subsystems are provided with a 10-MHz clock from the CPU.

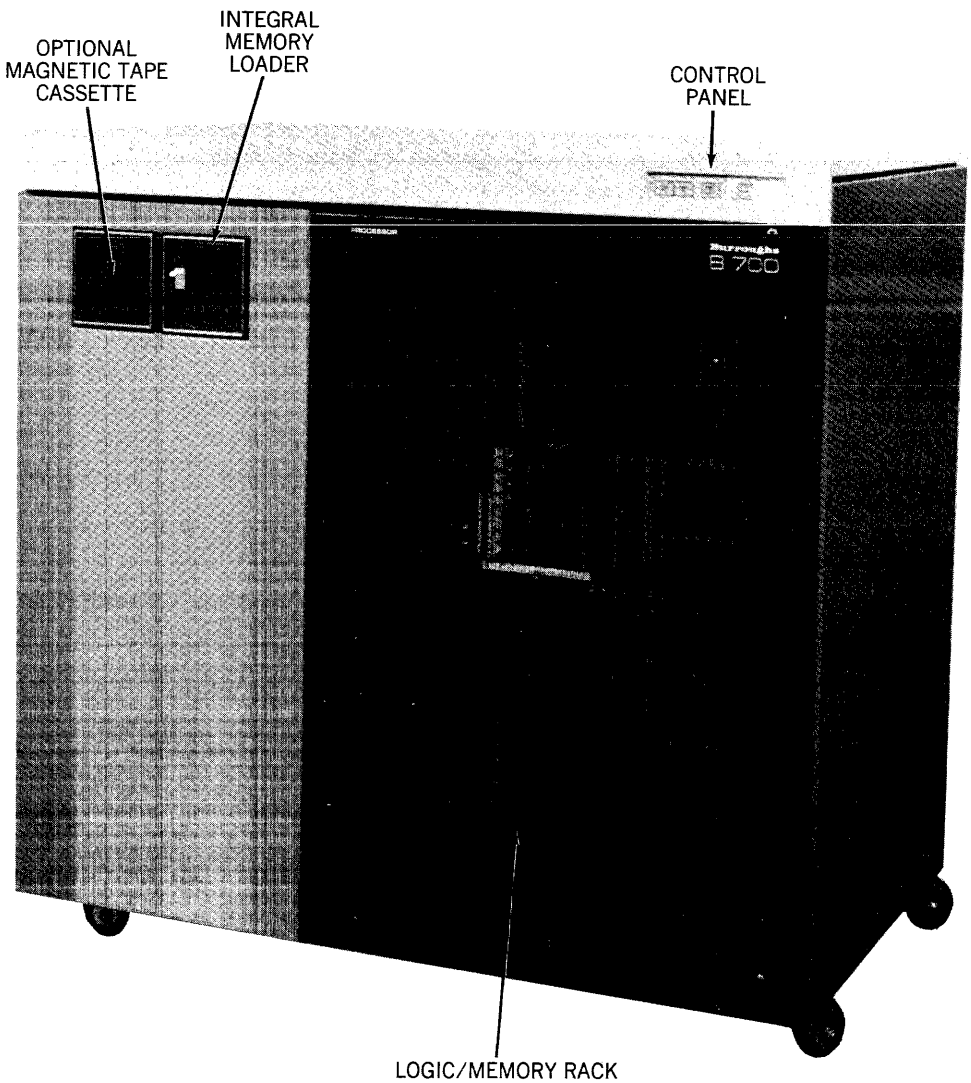


Fig. I-1 B 720 CENTRAL PROCESSING UNIT

INTERFACE CONFIGURATION

The B 720 CPU is a modular CPU in terms of functions and subsystems and can be configured within the minimum and maximum restrictions to fulfill a customer's requirements. The basic (or minimum requirement) System consists of the following:

a. B 720 Central Processing Unit with minimum main-memory capacity of 32K-bytes, photoelectric (paper tape) memory loader, and power supply group.

b. B 9343-22 Console (26 inch form-feed, 64 characters) or B 9343-42 Console (26 inch form-feed, 94 characters).

c. B 346 (64-character) or B 346-1 (94-character) Console Control.

d. B 9480-11/12 (single/dual drive, 100 TPI) or B 9481-11/12 (single/dual drive, 200 TPI) Disk Cartridge Drive.

e. B 489-2 Cartridge Disk Drive Control.

Up to 11 I/O control ports (Device Dependent Ports, DDP's) are available in the CPU to accommodate basic and optional peripheral devices or interfaces. An additional port, DDP2, is reserved for communications processor expansion.

Three of the processor IOC ports (DDP1, DDP4, and DDP12) are dedicated to the data communications subsystem and basic system controls. Three additional controls may be installed in interchangeable ports in the basic system. Up to five additional controls may be installed by use of the optional I/O expansion module, which provides five interchangeable ports. Table I-1 lists the applicability of the basic system I/O controls (IOC's) and those I/O controls available to interface the various optional peripheral devices.

Table I-2 summarizes the DDP configuration and device allocation.

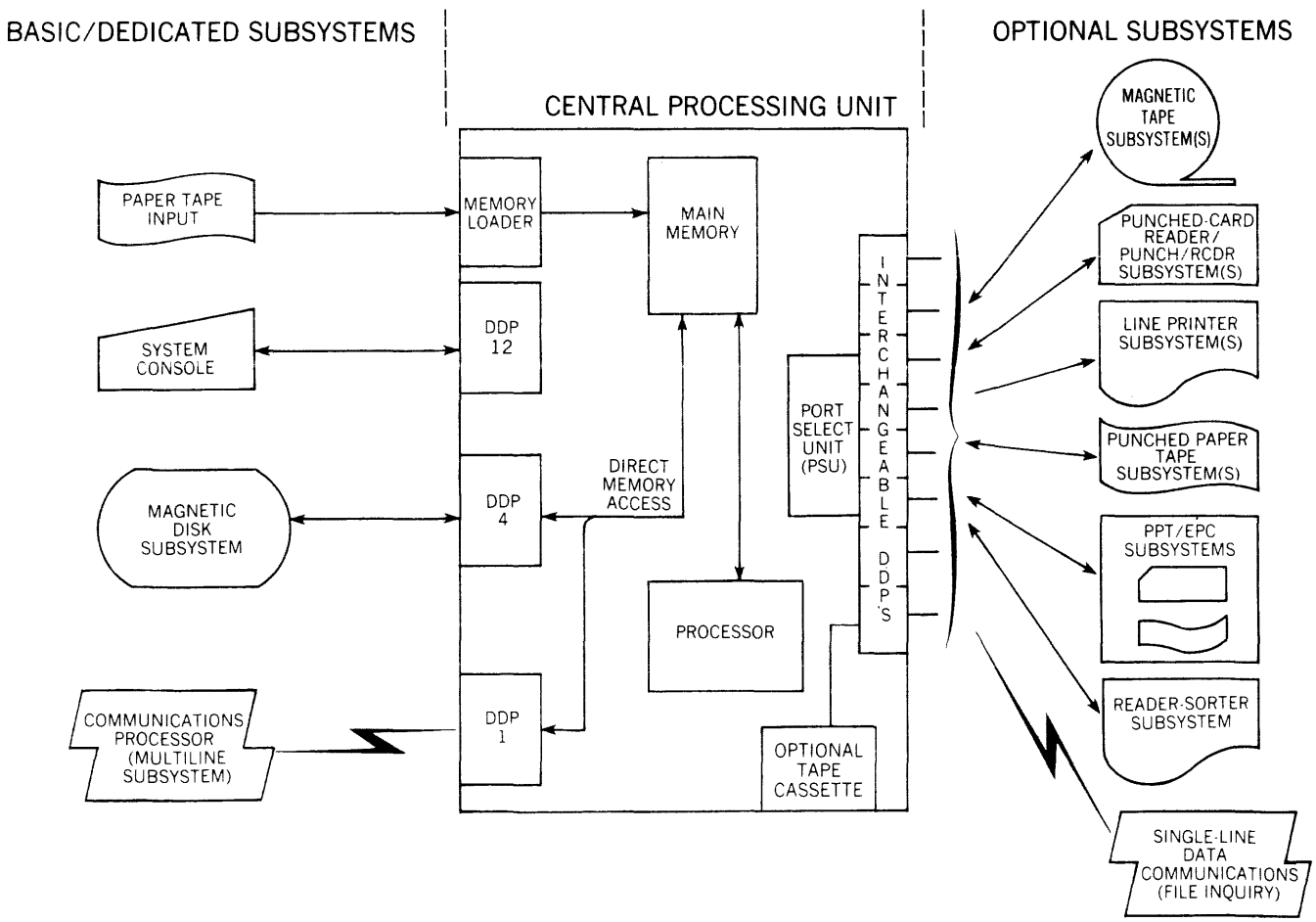


Fig. I-2 CPU-TO-SYSTEM INTERFACE CONFIGURATION

Table I-1. B 720 I/O CONTROLS

I/O Control	Peripheral Interface	Notes
B 346 Console Control (2602 4836)	B 9343-22 Console	Required for 64-character electronic keyboard; dedicated to port 12 of processor.
B 346-1 Console Control (2602 5791)	B 9434-42 Console	Required for 94-character (Katakana) electronic keyboard; dedicated to port 12
B 489-2 Cartridge Disk Drive (2602 4844)	B 9480-11/12 or B 9481-11/12 Disk Cartridge Drive	Dedicated to processor port 4.
B 351/351-1 Single-Line Control (SLC) (2602 7425)	Single-Line Data communications interface (direct or through data set/ modem)	Occupies two interchangeable processor ports (DDP's).
B 352 Communications Processor (2601 9810)	Communications Processor interface (direct of through data sets/modems).	Dedicated to processor port 1; must be installed in modular I/O expansion rack.
B 111 Card Reader Control (1448 0347)	A 9114-1 Card Reader	For 80-column card media; installed in interchangeable processor port (DDP).
B 243 Line Printer Control (1448 0388)	A 9249 Line Printers (1, 2, 3)	Installed in interchangeable processor port (DDP).
B 244 and B 244-X Line Printer Control (1448 0362, 2603 8737)	A 9247 Line Printers (2, 3, 12, 13)	Installed in interchangeable processor port (DDP). Optional EBCDIC and KATAKANA code sets.)
B 391 Magnetic Tape Unit (1448 0420)	A 9491-2 Magnetic Tape Unit	Installed in interchangeable processor port (DDP).
B 392 Magnetic Tape Cassette Control (1448 8654)	A 9490-25 Magnetic Tape Cassette Unit	Used for external tape unit installation; installed in interchangeable processor port (DDP).
B 392-1 Magnetic Tape Cassette Control (2603 6855)	A 9490-21 Magnetic Tape Cassette Unit	Used for processor cabinet (integral) installation of tape unit; installed in interchangeable processor port (DDP).
B 131 Reader-Sorter Control (2601 6600)	A 9135 or B 9136 Reader-Sorter	Installed B 312 I/O expansion module; restricted from installation in basic system interchangeable DDP's.
B 121-1 Paper Tape Reader Control (1448 9181)	A 9122-1 Paper Tape Reader	Installed in interchangeable processor port (DDP).
B 221 Paper Tape Punch Control (1449 0296)	A 9222-1 Paper Tape Punch	Installed in interchangeable processor port (DDP).
B 311 Card Reader/Punch/ Recorder Control (1448 0321)	A 9418 RDR/PNCH/RCDR (80-col), A 9419 RDR/PNCH/RCDR (96 col), or A 9119 Card Reader (96 col)	Handles 80 or 90 column card formats; installed in interchangeable processor port (DDP)

Table I-2. DDP CONFIGURATION AND DEVICE ALLOCATION

DDP	Location		I/O Device
	Logic/Memory Rack	I/O Expansion Rack	
1		X	Communications Processor
2		X	Unassigned
3		X	Optional (standard IOC's)
4	X		Disk IOC
5,6,7 and 8		X	Optional (standard IOC's)
9,10 and 11	X		Optional (standard IOC's)
12	X		System Console IOC

NOTE: The DDP priority scheme is in the increasing order according to the DDP number. (DDP 12 has the highest priority.)

GENERAL FUNCTIONAL ORGANIZATION

Figure I-3 is a general functional block diagram of the CPU, which consists of three major functional sections and two associated sections as follows:

a. The memory section, which consists of a semiconductor main (shared) memory and an integrated-circuit read-only nanoprogram memory (NPM). The main memory provides both data/program memory (DPM) and microprogram memory (MPM) storage. The MPM and NPM comprise the firmware storage area of the CPU.

b. The processor section, which contains the logic and control circuitry used in performing arithmetic, processing, I/O interface, memory control, and system control functions. This section also supplies the basic internal and external timing to the CPU.

c. The I/O section, which consists of the circuitry used in controlling and interfacing I/O subsystem operations. This section also includes the memory loader and load interface elements.

d. The power supply group, which provides and controls regulated operating voltages for CPU circuitry and external devices (as required).

e. The F.E. (Field Engineering) control section, which provides test and monitoring functions for equipment maintenance purposes.

MEMORY SECTION

The memory section of the CPU is divided into three functional memories: the data/program memory (DPM), the microprogram memory (MPM), and the nanoprogram memory (NPM). (See figure I-3.) The DPM and MPM share the semiconductor random access main memory (RAM), while the NPM resides a separate integrated-circuit read-only memory (ROM). Functionally, the MPM and NPM comprise the firmware storage area of the CPU. The MPM stores microinstructions, and the NPM stores nanoinstructions. Figure I-4 shows a typical main-memory allocation. The DPM stores data and user programs.

The main (shared) memory is a modular memory with a capacity of 32 to 96 K-bytes. The minimum 32K capacity may be expanded, in 8K increments, to 96K by installing additional B 31-2 Memory Modules. A total of 12 B 31-2 modules may be used.

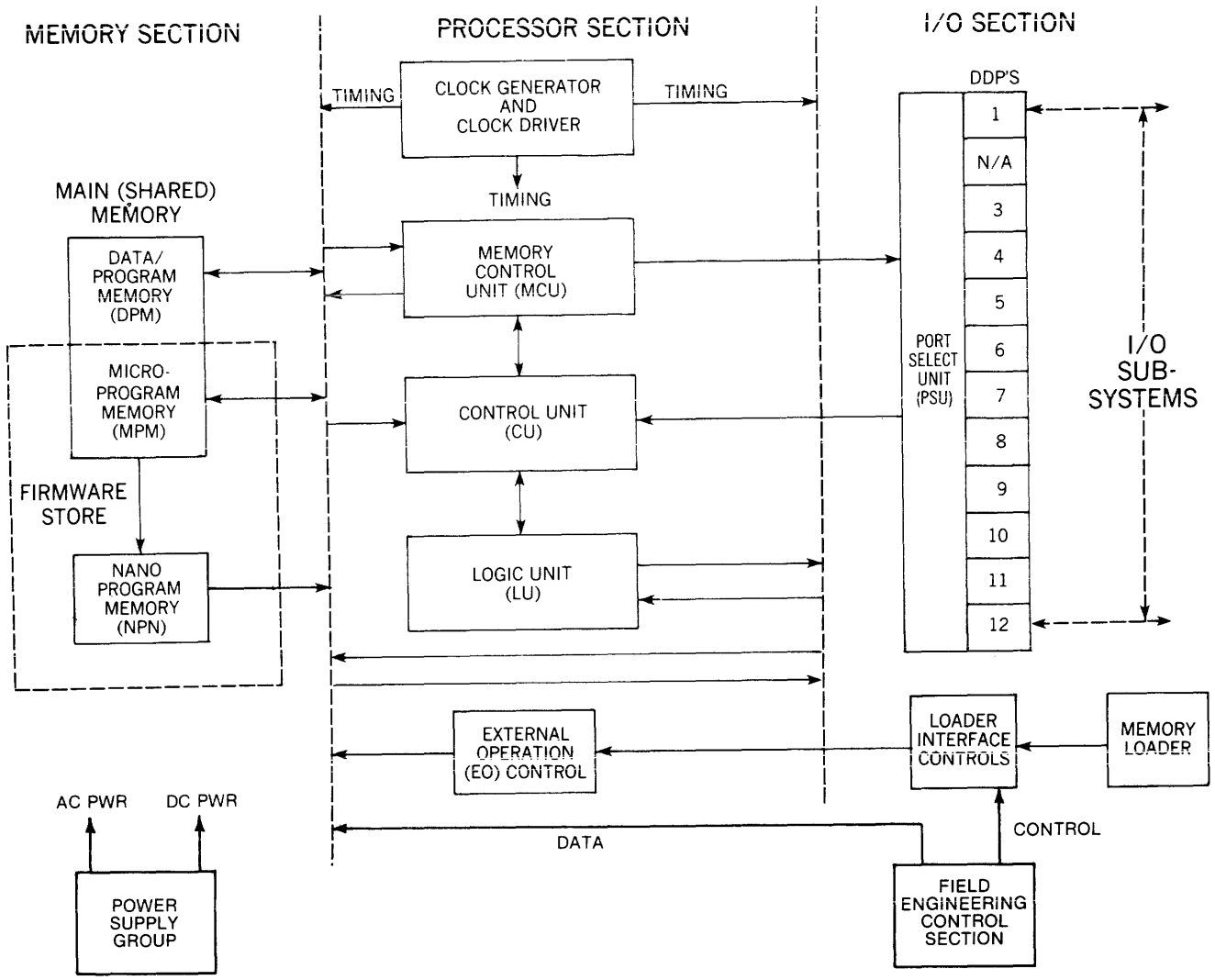


Fig. I-3 CPU GENERAL FUNCTIONAL BLOCK DIAGRAM

FIRMWARE STORE SECTION

The firmware store section, which uses RAM integrated memory for microprogram storage and ROM integrated-circuit memory for nanoprogram storage, contains the systematic controls required by the processor section. Because the main RAM memory is also used for data/program storage, addressing and data control is accomplished by shared memory controls. The microprogram memory outputs are called microprogram codes and are actual 16-bit instructions. These instructions are classified as either type I for nanomemory addressing, or type II for loading specific registers within the processor section.

The nanoprogram memory outputs are called nanocodes and are actual 55-bit instructions. Figure I-5 shows the configuration of microprogram and nanoprogram codes. Descriptions of the functional registers and elements are provided in the processor section description.

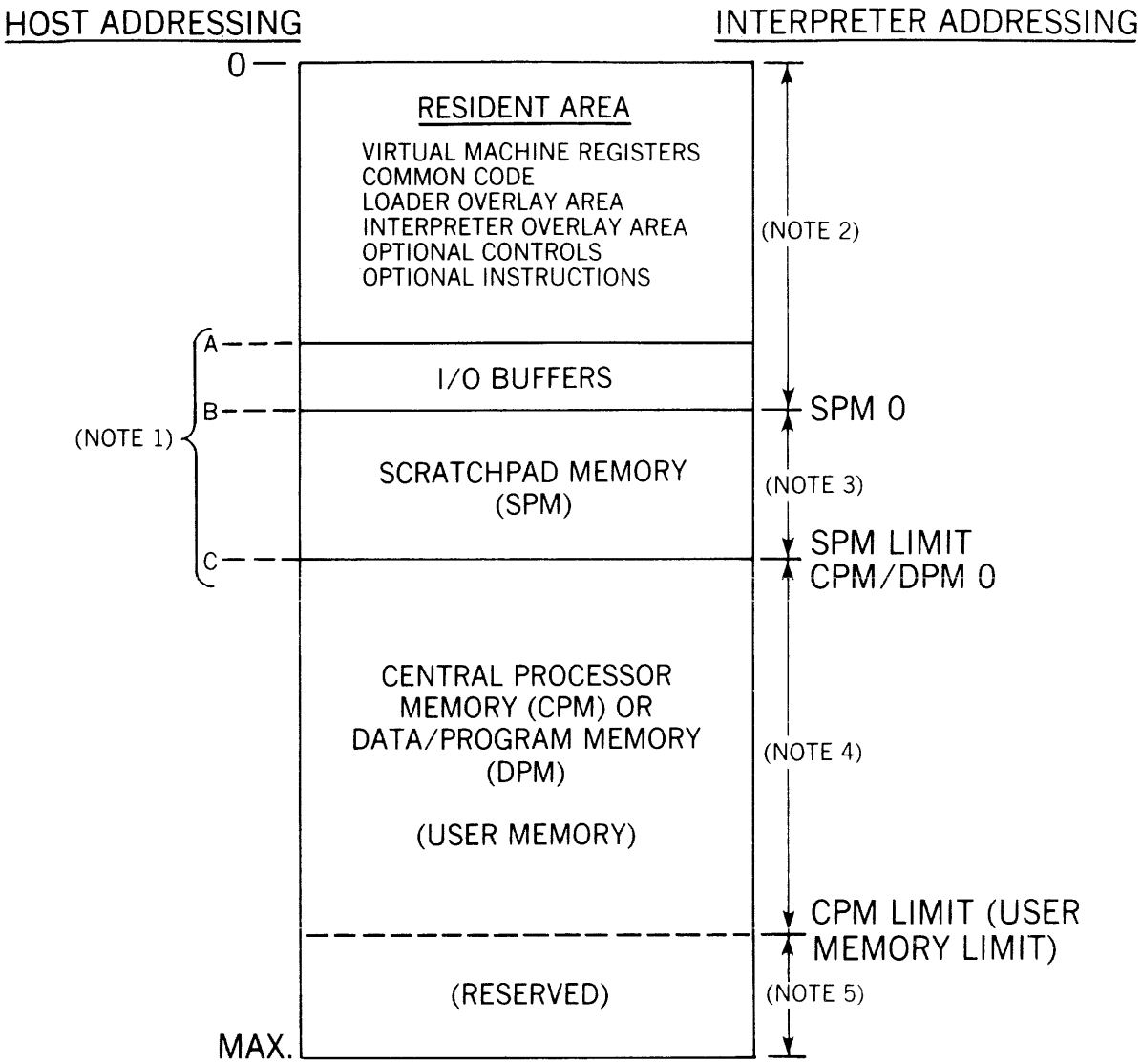
Nanoinstructions have three major classifications:

a. No conditional logic.
b. Conditional logic met.
c. Conditional logic not met.

In MPM storage, correct parity is generated (bit 17) when writing, and checked when reading. Bit 56 in NPM storage is used to check parity when reading nanocodes. Odd parity is used for MPM and NPM validity.

DATA/PROGRAM (USER) MEMORY SECTION

The Data/Program Memory (DPM) section is that portion of main memory which contains user type data and S-level programs and is controlled by shared memory logic in conjunction with external operations.



Notes:
1. Each of these boundaries is variable, depending on implementation and program requirements.
2. Not addressable by interpreter program.
3. Only as large as required.
4. Maximum CPM required for this program.
5. This space **may** be available to the interpreter program through indexing.

Fig. I-4 TYPICAL MEMORY ALLOCATION

MICROPROGRAM CODES

MPM BITS

1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6	1 7	
0 0 ——— AMPCR ———		P
0 1 ——— MPCR & INCR ———		A
1 0 ——— MPCR & INCR ———		R
1 1 0 0 — SAR — ——— LIT ———		I
1 1 0 1 — SAR — 0 0 0 0 0 0 0 0		T
1 1 1 0 0 0 0 0 ——— LIT ———		Y
1 1 1 1 ——— NANOADDRESS ———		B
		I
		T

0 = UNUSED

NANOPROGRAM CODES

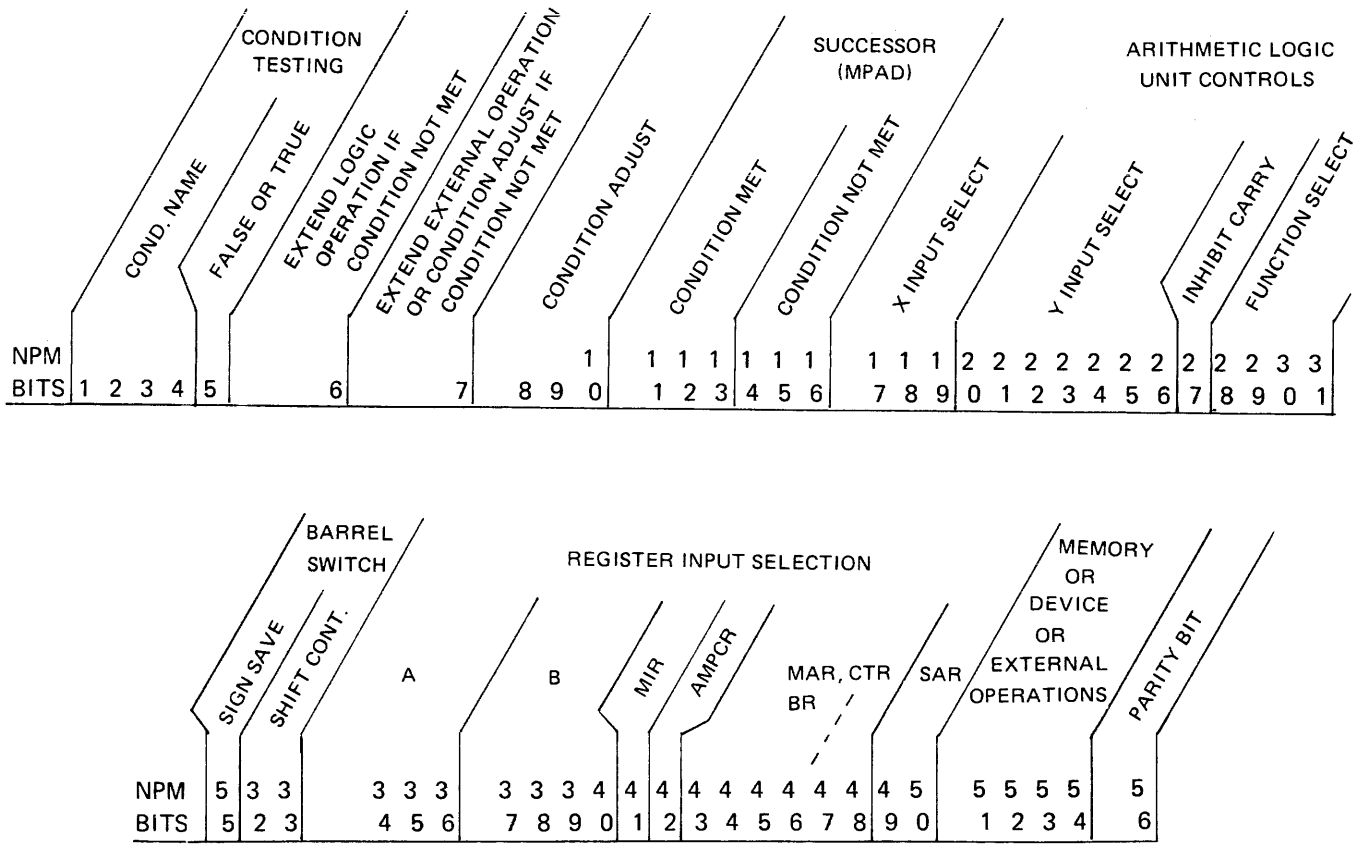


Fig. I-5 MICROPROGRAM AND NANOPROGRAM CODES

SHARED MEMORY CONTROLS

Microprogram memory and S-level memory share the same physical memory in the CPU. The shared memory (SM) controls are used for addressing both segments of memory and for transferring data to and from memory.

The size of shared memory ranges from a minimum of 8,192 words (16,384 bytes) to a maximum of 49,152 words (98,304 bytes) when all memory options are used. Extensions to memory are in physical increments of 4,096 words (8,192 bytes). The specification of a memory address in excess of the address limit results in an error condition.

Because the memory is shared by S-level and MPM, provisions are made for addressing the two segments separately. The memory controls contain selection gating for accepting the memory address from either the base register (BR), the MAR (for S-level memory), or from the incrementer (for MPM memory), as determined by the appropriate memory controls. Up to a maximum of 16,384 words (32,768 bytes) of the shared memory may be reserved for MPM, depending on the requirements of the microprogram. The division between S-level and MPM is soft, allowing the upper limit of MPM to be varied at the discretion of the system software.

An external (EXT) timer, which is a binary counter, sets the EXT flip-flop every 125 milliseconds. EXT is reset by testing and can be used for various applications by the micro-programmer.

S-LEVEL MEMORY OPERATION

Addressing of S-level memory is accomplished by use of an external operation command which specifies a memory read or write operation. In this case the external operation controls enable the appropriate control signals, causing the next memory address to be taken from output select lines 1 through 16. Data being written to S-level memory is transferred through the S-memory (SM) controls from the MIR bus of the processor. Data read from memory goes only to the B register of the processor through the SM controls and the external bus. All data transfers to and from S-level memory are in 16-bit parallel format. Parity is generated for all words written into memory and is checked on all words read from memory.

MEMORY-STANDBY MODE

A standby mode is provided so that, during equipment maintenance operations, power can be maintained to the first 8 K-bytes of memory to facilitate servicing. This segment of memory is used to store maintenance Test Routines (MTR's), which can remain undisturbed during power on-off cycling in diagnostic operations.

PROCESSOR SECTION

The processor section is the major section of the CPU and performs operations defined by the program stored in the firmware store section, under the control of microcodes and nanocodes. The processor section is subdivided into five functional circuit areas:

a. Logic unit (LU).
b. Memory control unit (MCU).
c. Control unit (CU).
d. Clock generator (CG).
e. External operation controls (EO).

LOGIC UNIT

The Logic Unit (LU) performs the shifting, arithmetic, and logic functions and provides a set of scratch-pad registers for temporary storage. The LU also provides the registers for the data interfaces to and from the I/O controls and S-level memory (DPM). The major registers and elements comprising the Logic Unit are the A registers, B register, memory information register, adder, and barrel switch. Figure I-6 shows the functional configuration of the logic unit.

A REGISTERS (A1, A2 AND A3)

The A registers are 16-bit registers used for the temporary storage of data being transferred from the Barrel Switch (BSW) to the adder. Any or all A registers may be selected for input from the BSW. The registers serve as a primary input to the adder and are individually selectable.

B REGISTER

The B register is a 16-bit register which provides the primary interface from both S-level (data/program) memory and the I/O controls. It serves as the secondary input to the adder, and can be used for temporary storage of certain information resulting from arithmetic operations. The only destination for this data is

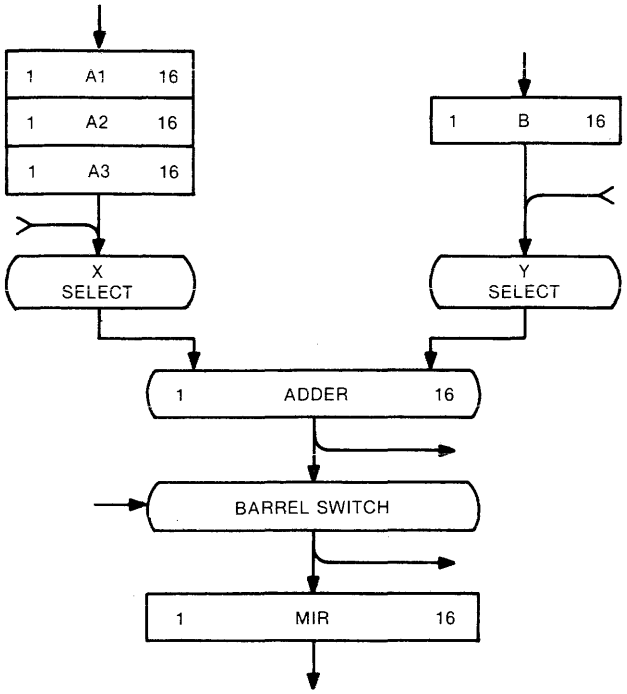


Fig. I-6 LOGIC UNIT

the adder. The nanocode allows the manipulation of the least significant bit, most significant, and/or the 14 central bits of the B-register contents upon transfer to the adder.

ADDER

The adder performs the arithmetic and Boolean operations on data from the B-register, literal/counter register(s), alternate microprogram count register, or base register/memory address register. Output from the adder goes unconditionally to the barrel switch, but may also be sent to the B-register if so specified.

BARREL SWITCH (BSW)

The Barrel Switch (BSW) is a matrix of gates used to shift a parallel input data word a number of places left or right, end-off, or end-around. The shift amount is specified by the contents of the Shift Amount Register (SAR). Data input to the barrel switch is from the adder. Destinations are the A registers, B register, memory information register, alternate microprogram count register, memory address register, either base register, or the shift amount register.

MEMORY INFORMATION REGISTER (MIR)

The MIR is a 16-bit register used primarily to buffer information being written in memory or sent to a device. MIR information is sent to main memory, an I/O control or to the B register.

MEMORY CONTROL UNIT

The Memory Control Unit (MCU) is used primarily for memory and I/O device addressing. The major registers and elements comprising the MCU are described below and shown in figure I-7.

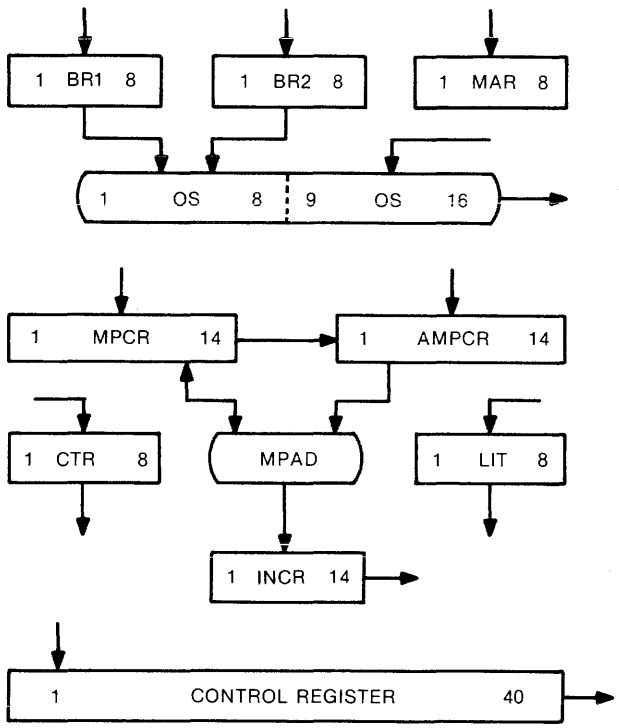


Fig. I-7 MEMORY CONTROL UNIT

BASE REGISTER 1 (BR1)

Base register 1 is an eight-bit register which holds a device address or the base address of a 256-word block of memory data. When used for a memory address, BR1 is concatenated with MAR to form an absolute memory address that is transferred by the output select gates to S-level memory. The concatenated registers may also be sent to the adder.

When used to hold a device address, BR1 is interfaced by the output select gates to the port select unit, where it is used to address the IOC to be used. The input to BR1 is from the most significant byte of BSW. Once placed in the selected mode by a memory/device command, BR1 remains selected until a command is issued to select BR2.

BASE REGISTER 2 (BR2)

Base register 2 functions exactly like BR1, thus providing a second register for holding

memory/device addresses. Once placed in the selected mode by a memory/device command, BR2 remains selected until a command is issued to select BR1.

MEMORY ADDRESS REGISTER (MAR)

The MAR is an eight-bit register which holds the eight LSB of a memory address. MAR is concatenated to either base register 1 or base register 2 to form the absolute address. MAR may be loaded from either the least-significant byte of the barrel switch or from the literal register. The output is to S-level memory. The MAR along with the currently selected concatenated register may also be sent to the adder.

OUTPUT SELECTION GATES (OS)

The output selection gates select the specific source (BR1/MAR or BR2/MAR) of S-level memory or device addresses. A Read/Write "1" command selects BR1/MAR; whereas, Read/Write "2" selects BR2/MAR. If no source is specified, the register selection remains unchanged. OS output is to the port select unit and/or to the adder.

MICROPROGRAM COUNT REGISTER (MPCR)

The MPCR is a 14-bit register which contains the instruction address for the microprogram. MPCR contains the current instruction address except when an "EXECUTE" instruction is performed. MPCR can be loaded by a type II instruction or with the output of the incrementer. MPCR output goes to both the incrementer and the AMPCR.

ALTERNATE MICROPROGRAM COUNT REGISTER (AMPCR)

The AMPCR is a 14-bit register which contains the jump or return address for program jumps and subroutine returns within microprograms. The address is one less than the position to be jumped to and two less than the position to be returned to. AMPCR can be loaded from MPCR, from the 14 LSB of the barrel switch, or by a type II microinstruction fetch from microprogram memory. AMPCR output goes to the incrementer.

MICROPROGRAM ADDRESS CONTROLS (MPAD)

The MPAD controls are used to control the loading of MPCR and AMPCR, the selection of MPCR and AMPCR input to the incrementer, and the selection of the value (0, 1, or 2) to be used in incrementing. The selection of controls and a true or false successor (testing for alternative actions) is done during phase 1 of the microinstruction and involves the results of condition testing and successor determination associated with bits 1 through 16 of the nanoinstruction. If a type II microinstruction is executed, the controls for a step successor are forced.

INCREMENTER

The incrementer adds 0, 1, or 2 to the selected input from either MPCR or AMPCR. The output of the incrementer is the input to the MPCR and also provides an address to the microprogram memory. The MPAD address controls select both the input source and the amount to be incremented.

COUNTER (CTR)

The CTR is an eight-bit counter used for loop control and other counting functions. CTR is loaded through the MAR/CTR selection gates from either the literal register or the least-significant byte of the barrel switch. CTR can be used as an input to the most-significant byte of the adder. A counter overflow results in setting the Counter Overflow flag (COV) in the control unit condition register; COV is reset either by testing the bit or by loading the counter with a new value.

LITERAL REGISTER (LIT)

The LIT is an eight-bit register used as temporary storage for literals in the microprogram. LIT is loaded from microprogram memory using a type II microinstruction. LIT may be used as an input to the least-significant byte of the adder, and/or through the MAR/CTR input selection gates to either the counter register or the MAR.

MAR/CTR INPUT SELECTION GATES

The selection gating consists of eight bits and is used to select an input to the MAR or CTR from either the LIT register or the least-significant eight bits of BSW. These functions are mutually exclusive.

CONTROL REGISTER

The control register is a 40-bit register used to store all control signals from the nanomemory that are not used in phase 1. Certain control signals are decoded before being strobed into the control register while other are decoded on output. Gate delays introduced into the control signal sequence determine whether the coding is done before or after the register. The register includes both MPAD controls and phase 3 controls and thus con-

tains their respective clock gating networks. The control register is physically contained in the MCU, CU, and EO registers and provides nanocontrols to all sections of the processor.

CONTROL UNIT

The Control Unit (CU) performs two main control functions: shift amount control and storage of condition indicators, data and status interrupts, and status indicators. The control register, defined within the MCU, distributes nanocontrols throughout the processor. The elements comprising the control unit are described below and shown in Figure I-8.

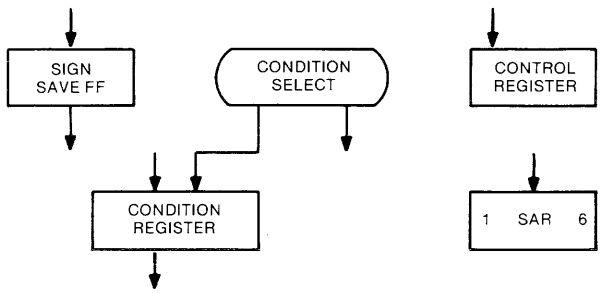


Fig. I-8 CONTROL UNIT

CONDITION REGISTER AND SELECT

The condition register has six of the 16 selectable condition bits, only one of which may be selected and tested by a nanoinstruction. If an attempt is made to reset and set a condition bit at the same time, the set condition is dominant except on counter overflow (COV), in which case the reset condition is dominant.

SIGN-SAVE FLIP-FLOP

The sign-save flip-flop is used to store either adder overflow or barrel switch end-off bits.

SHIFT AMOUNT REGISTER (SAR)

The shift amount register and its associated logic is used to control the loading of shift amounts and the sequencing of shift operations. (Refer to barrel switch description.)

CLOCK GENERATOR AND CLOCK DRIVER

The Clock Generator (CG) contains a 10-MHz oscillator and a divider which produce the 1-MHz system timing clock (S-clock); that is, a 50-nanosecond pulse every microsecond. The S-clock is distributed throughout the processor, except to the logic unit. The logic unit is clocked by the phase-3 clock (3-clock), derived from the S-clock. The main function of the phase-3 clock is to inhibit destination register selection during an extended phase-3 condition; these registers are clocked upon phase-3 completion. A 10-MHz clock pulse is also distributed to peripheral device subsystems that use a miniprocessor. A third set of controls on the CG board is used for generating the clear signals to the processor elements. The Clock Driver (CD) contains drivers for clock pulse buffering and distribution.

EXTERNAL OPERATION CONTROLS (EO)

Memory and I/O operations are mutually exclusive due to the fact that both share common input busses. The selection of either memory or a device is controlled through the decoding of nanobits or from an I/O subsystem that has a direct memory access channel (DMAC). The nanobits, which are sent to a control register in the EO controls, specify whether a memory operation or an I/O operation is to take place. The nanobits also define the function that is to be performed, and select the base register to be used in the addressing operation. If a memory read or write operation is indicated, the command is fully decoded, and the appropriate interface signals are generated to the memory. An I/O subsystem with a direct-memory access channel can "steal" a processor main-memory cycle. (Refer to description of DMAC under I/O control section heading.)

Device read and write operations are decoded before being sent to the port selector, where they are buffered and gated so as to cause the device operations to be terminated at the proper times. However, the Address/Status Request (ASR) signal is decoded directly from the nanobits and sent to the port selector as a separate signal without gating. This is to allow immediate enabling of the status word lines to the external bus since the status word is returned to the processor in the same clock time.

ERROR DETECTION

The processor automatically detects error conditions and initiates the following:

a. System clock is inhibited and the HALT indicator (on EO card edge) is turned on.

b. The shared memory address (to be accessed is forced to 0. (INCR and MPCR are *not* cleared.) Location 0 must contain a CPCR call instruction to direct the firmware to the appropriate error-processing routine and to store the contents of MPCR in AMPCR for reporting to the operator.

c. The four-bit error descriptor code is displayed by the EO2 card error indicators, which are card-edge indicators visible through the front panel of the processor. The error descriptor code is also placed on EXT bus bits 13 through 16, and the exception bit is placed on EXT bus bit 1. These EXT bits are read by the error-processing routine.

d. The HALT indicator is turned off.

e. When the error processing routine is executed, the error indicators and EXT bits are cleared.

f. When the error processing routine is completed, the initial error condition is reset to permit processing any subsequent error condition.

If a second error condition occurs prior to the resetting of the first condition, the system locks in the halt condition (all system clocks are inhibited) and the descriptor code of the second error condition is displayed on the EO2 indicators.

If the error stop switch is set, or the load mode is set, the system locks in the halt condition on the first error.

Table I-3 shows the configuration of error codes presented on the EXT bus and displayed by the EO2 card-edge indicators.

NOTE

EXT bits 13 thru 16 are indicated by EO bits 8, 4, 2, 1, from top to bottom on EO card.

In general, the error indications have the following meanings:

a. Memory loader parity read error detected in the media data during read-in of data from memory loader to microprogram memory.

b. MPM parity error detected in the MPM portion of shared memory.

c. Read-after write MPM parity error detected during load on microprogram memory read-after-write check during loading from console paper tape reader.

d. DPM parity error detected in the DPM portion of shared memory.

e. NPM parity error detected in nanoprogram memory word.

f. Memory address limit error exceeds memory limit register setting.

Table I-3. Error Descriptor Codes

EO Indicators (EXT Bits)					
Halt (1)	8 (13)	4 (14)	2 (15)	1 (16)	Error
0	0	0	0	0	None
1	0	0	0	0	Transient
1	0	0	0	1	Loader Error
1	0	0	1	0	Load MPM Parity
1	0	0	1	1	Load Address
1	0	1	0	0	Nano Parity
1	0	1	0	1	Nano Address
1	0	1	1	0	MPM Parity
1	0	1	1	1	MPM Address
1	1	0	0	0	Not Used
1	1	0	0	1	DPM Address, Write
1	1	0	1	0	DPM Parity
1	1	0	1	1	DPM Address, Read
1	1	1	0	0	Not Used
1	1	1	0	1	Steal Address, Write
1	1	1	1	0	Steal Parity
1	1	1	1	1	Steal Address, Read

I/O CONTROL SECTION

The I/O control section of the CPU is the interface for control communications and data transfer between the processor section and the various I/O and data communications subsystems. The I/O section has four major functional units or areas: the port select unit, the device-dependent ports, the I/O controls, and the loader interface area. Direct-memory access circuitry is also implemented for direct accessing between certain I/O subsystems and the memory section.

The Port Select Unit (PSU) interfaces all peripheral device I/O subsystems and data communications subsystems and the memory load subsystem. The PSU maintains input and output control for the various interfaces and establishes service and interrupt priorities.

The Device-Dependent Ports (DDP's) are the I/O backplane or module areas that accommodate the various I/O controls (IOC's). These ports are physically and functionally categorized into two groups: special (or dedicated) DDP's, and interchangeable (standard) DDP's. A total of 11 DDP's are available if the I/O expansion module and communications subsystem are installed. An IOC is designed to interface a particular device or interface by converting levels, interpreting signals, converting

formats, synchronizing or timing operations, and buffering data. (Refer to Section II for a complete description of the DDP and IOC configuration.)

The allocation of DDP's to IOC's is predicated on a priority service basis, established during system installation for the particular I/O configuration used.

DEVICE ADDRESSING

The PSU receives device addresses from the base registers (BR1 and BR2) in the processor. The firmware selects a device for access by sending to the PSU the device address contained in the base register and pointed to by the read/write command. This binary address, sent through the output select (OS) gates, defines the device IOC to the port selectors, as shown in figure I-9.

The most-significant bit of the address is used by each IOC to specify control or data word format.

NOTE

Control format on a device read operation requests status information from the IOC.

Input information to the addressed IOC comes from the processor section MIR. Output information from the addressed IOC (EXT lines 1 through 16) is made available only to the B register in the processor section.

INTERRUPT HANDLING

Interrupts are always sent from the IOC's to the port selector, even though the control may not be addressed. Interrupts from unaddressed IOC's are in the form of an Input Request (IRQ) to the processor-section control unit from the highest priority device and are generated by either status interrupt or data interrupt signals. Handling of the interrupt condition is the responsibility of the firmware store program. The program must issue an address status request command, placing the reason for interrupt from the IOC on the EXT lines, along with the address of the IOC position from the port selector. Interrupts from addressed IOC's to the processor-section control unit are an unsolicited request (URQ) for a status interrupt and a solicited request (SRQ) for a data interrupt. Handling of these interrupts is also the responsibility of the firmware store program.

MEMORY-LOADING

Direct memory loading is facilitated by an integral photoelectric paper-tape reader mounted in the left-front section of the CPU cabinet (I-1). Loader interface (LI) controls permit data to be transferred directly from the paper tape reader to microprogram memory. Data is written into the MPM one word at a time, including a generated parity bit. These words are written sequentially up

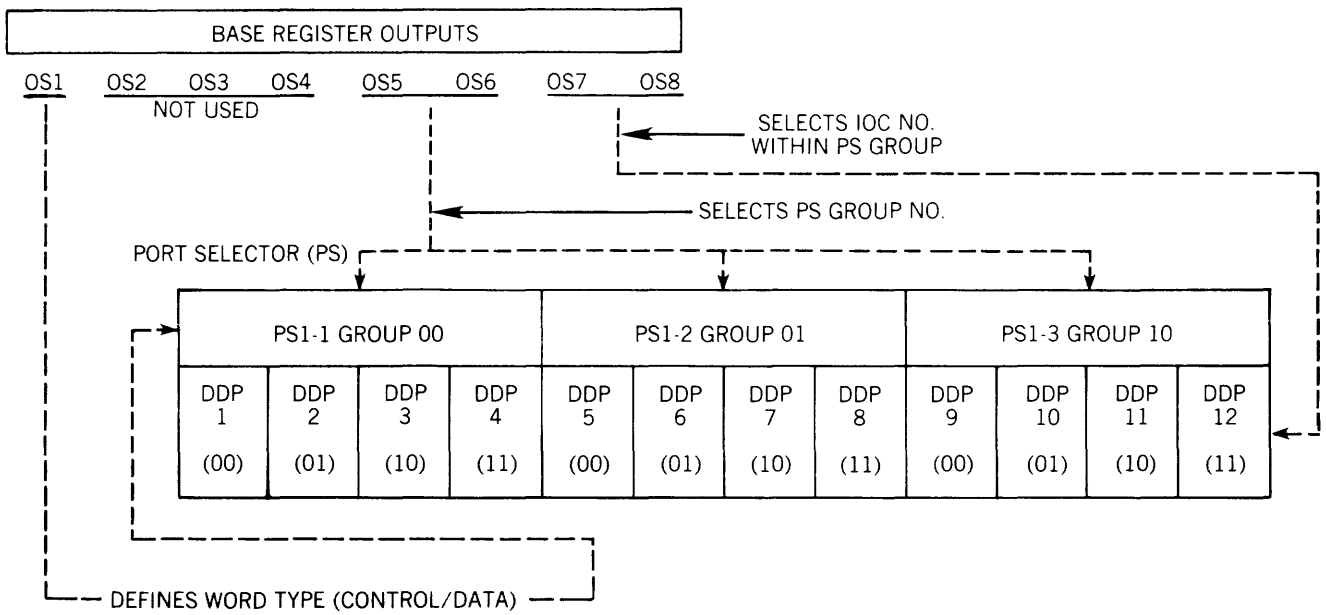


Fig. I-9 I/O DEVICE ADDRESSING

through main memory when the processor LOAD pushbutton is enabled.

DIRECT MEMORY ACCESS

Direct memory access (DMAC) channels are provided for the disk and programmable data communication subsystems (communications processor) to enable transfer of data and control words between main memory and the subsystems. Direct memory access is performed without any processor intervention required and thus frees the processor to perform other tasks.

A process referred to as "stealing" is used in direct memory transfers; that is, the disk IOC or Communications processor "steals" one cycle (clock period) of processor time for each data/control word transfer. This process is controlled for both DMAC channels by circuits in the disk IOC. The steal control circuits receive steal request signals from the disk IOC and the Communications processor and grant memory access on a service priority basis. The disk I/O subsystem has top priority. The steal control circuits also receive MPM data outputs directly from the shared memory control circuits of the processor and apply the data to the common IOC data bus. This bus handles both memory data retrieved through the direct-memory access and control words and/or data sent to an IOC under processor logic control.

SYSTEM POWER SECTION

The CPU receives its input power directly from the facility-supplied power source and, through internal power distribution, the CPU can provide power to connected system peripherals. There is different power distribution circuitry for domestic and international applications to compensate for power supply frequency and voltage variations. The AC controls provide the means for properly sequencing the power on and off functions. Switches are also provided for the memory standby mode of operation.

FIELD ENGINEERING CONTROL SECTION

The Field Engineering cards (FE1 through FE4) are used by the field engineer to isolate system malfunctions. They provide for two modes of operation: (1) a memory test mode, which is used to establish confidence in and/or troubleshoot the MPM and NPM memories, and (2) an MTR mode, which is used to isolate system malfunctions other than in the MPM/NPM memory. Section V describes each mode of operation, the functions provided by the operator controls, and the arrangement of controls and indicators on the FE cards.

Two switches are provided for field engineering use for entry to and exit from the standby power mode. The switches are mounted on an internal frame member in a manner that, when the right side panel is attached, the standby power switches are disabled (normal operation), and, when the side panel is removed, the switches are enabled. The switches must be manually sequenced in conjunction with the power ON and OFF switches to activate and deactivate standby power to the memory. During normal operation the standby power supply is disabled.

Card slots are wired in each CPU to accept the FE cards; however, the cards themselves are not always installed in the unit when shipped. These cards are part of the standard test equipment provided for site installations.

OPERATIONAL DESCRIPTION AND APPLICATION

The following paragraphs describe the operation of the CPU in terms of the hardware being controlled by firmware with user applications and utility software. Each operation and the required flow of operations performed by the CPU are determined by the firmware. The microprogram instructions and the nanoprogram memory bits are the firmware of the CPU.

The firmware is the interpreter of the user program. The user program consists of macroinstructions (S-level) which are interpreted into the required sequences of microprogram steps that control the hardware through the necessary logic and arithmetic functions. In this manner, the macroinstructions of the user program are executed through firmware interpretation.

Figure I-10 is an operational-flow diagram of the CPU. The microinstructions are stored in the microprogram memory (MPM) and are identified as either type I or II instructions. A type I microinstruction is used to address the nanoprogram memory (NPM) and select one nanoinstruction. Type II microinstructions are used to load literal constants and micropro-

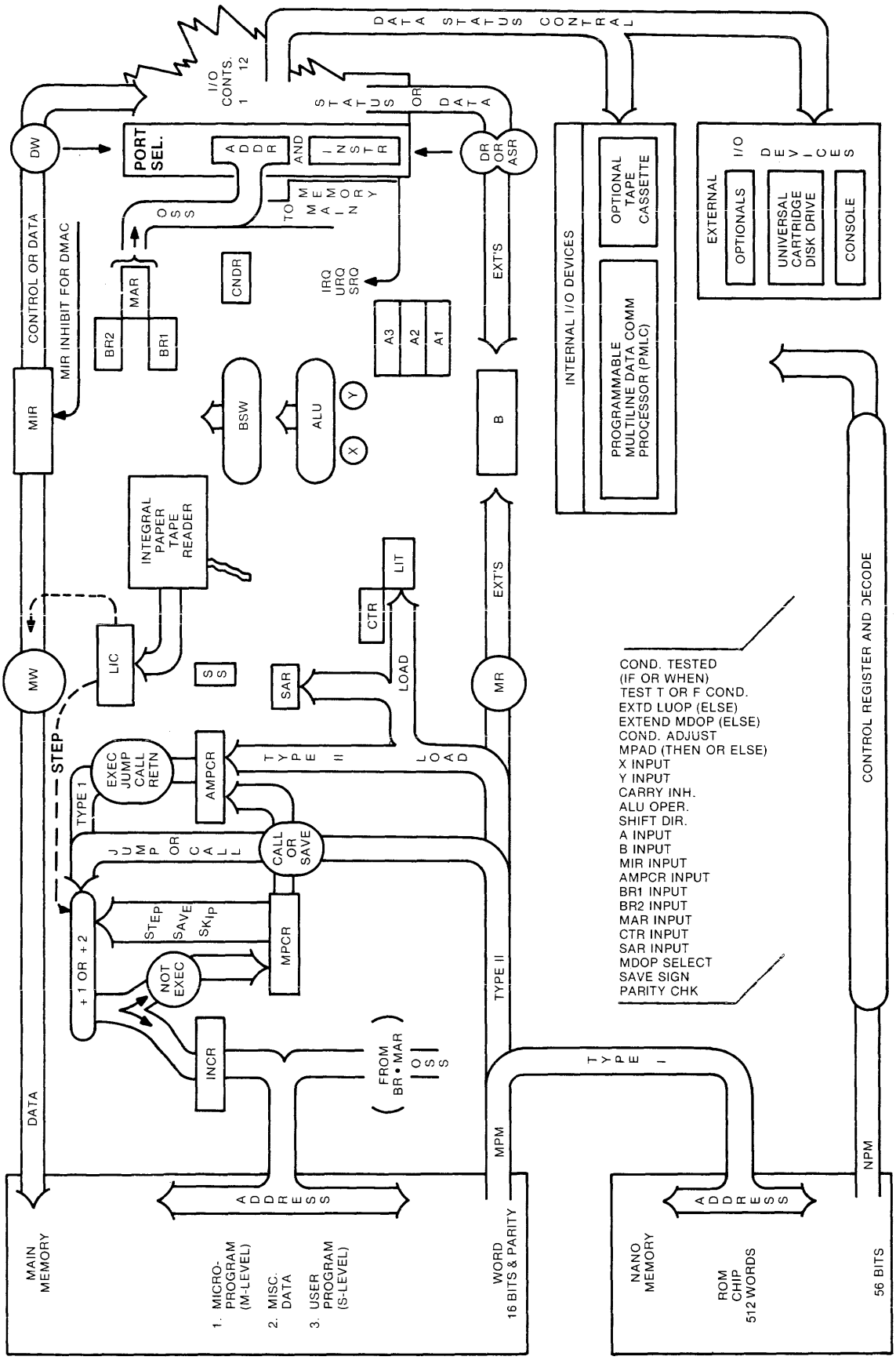


Fig. I-10 OPERATIONAL FLOW

gram address counts into selected registers and counters.

The means of implementing the hardware functions of a CPU is by the microcode and nanocode present at any given clock time. The nanocode has up to 512 different word combinations that are addressable from the microcode. The required flow through a firmware program is accomplished using three registers and incrementing logic. The first register, Microprogram Counter (MPCR) retains the present program address during most operations. The second register, the Alternate Program Counter (AMPCR) retains an alternate address that the program may use at some successive operation. The third register, the Incrementer (INCR), actually addresses memory for the present instruction. The next address that the INCR produces depends on the present program instruction. Possibilities are: MPCR plus 1 or 2, AMPCR plus 1 or 2, and also the address content of MPM plus 1. The nanodecode MPAD equivalents or type II microcode equivalents determine the selection.

Each instruction, as it is addressed by the INCR, is said to be "fetched". The type I instruction cannot be executed until some successive instruction. The type II instruction is however complete on each fetch cycle. Only type I instructions have the ability to produce both fetch and execute cycles. (Basically, a fetch cycle stores nanobits in the control register, and an execute cycle uses them.) This feature makes it possible to establish first the hardware logic function, second the loading of register values to satisfy desired inputs to the function, and third the execution of the logic function. Details of the above process are explained under "phasing" in Section II.

The user requires a microprogram so constructed that it can either interpret some intermediate program language or perform all the duties of a standard communications terminal. In either case, the microprogram must somehow be stored in main memory. The integral paper tape reader and the loader interface control in conjunction with the load switches and cold starting procedures are used to store the microprogram in main memory and make the hardware begin to function.

Figure I-11 is a block diagram of a CPU microprogram flow. The systems microprogrammer constructs B7MPL statements as source information to a medium system computer containing a microprogram language assembler. The assembler produces an output of the microprogram in compact hexadecimal code on either paper tape or cards, along with a listing of the program. The following programs are generated in this manner:

a. Four types of cold start programs identifying a specific I/O device for further cold start data.

b. A disk primer for central processor applications.

c. The interpreter to implement different intermediate languages.

d. Warm start programs for central processing systems that have already had the cold start procedures completed.

e. Various Maintenance Test Routines (MTR's) for diagnosing system faults.

f. Micro-utility programs.

The following operations occur when cold starting the CPU: (1) the cold start program is loaded into main memory through the integral paper tape memory loader interface; (2) the cold start program loads the disk primer and cold start (DPCS) program from any of four different I/O devices and then it transfers control to the disk primer program; (3) the disk primer program initializes the disk cartridge in preparation for more programs and then loads the segments of the interpreter programs on the disk cartridge, including any necessary utility programs to implement loading user programs. Transfer to the interpreter program is accomplished at the end of the last loaded program. At this point, the system has completed the cold start process.

A warm start is accomplished to bring interpreter data from an already cold-started disk back into main memory. (See figure I-12.) As in the cold start procedures, the warm start microprogram is first loaded into main memory through the load interface control in conjunction with the LOAD switch. The warm start microprogram then reads the disk cartridge interpreter data and writes it into main memory. When the writing is completed, the warm start microprogram transfers control to the interpreter microprogram system configuration subroutine.

After the system configuration has been introduced to the system from the console keyboard following a cold or warm start, the

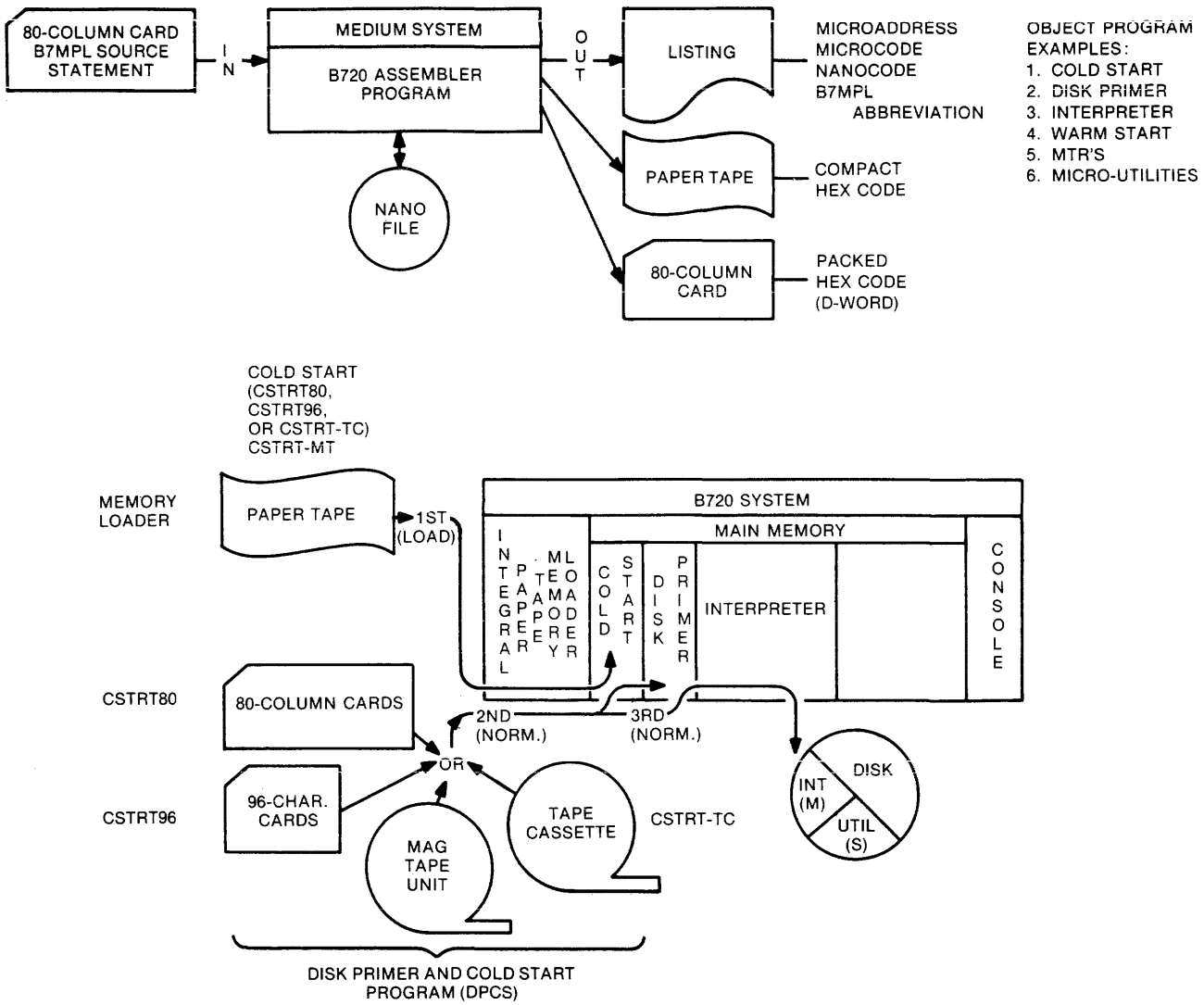


Fig. I-11 MICROPROGRAM FLOW

interpreter program returns to the ready mode. In the ready mode, the interpreter has four basic options: (1) load a new user or utility program from disk to main memory; (2) enter any of the system micro-utilities; (3) enter system date; or (4) implement inquiry option. Each word of memory is written into until the memory limit is detected.

Figure I-12 is a flow diagram of a user or utility program operations. The user or utility programs originate at applications software. The applications programmer constructs COBOL source statements to be compiled by a medium system processor. The COBOL compiler produces an S-level code output with a listing.

Any program that the user desires to execute must first be stored on disk cartridge. One of three different loader utility programs must have been stored on disk at cold start time and can now be loaded into main memory using the load function when the interpreter is in the ready mode. The header of this utility program is examined (figure I-12, step 3) to see if any additional interpreter segments (delayed binding) are needed (step 4) from disk to execute this new program. The S-level utility program now in main memory (step 5) is examined by the interpreter to determine the different steps the interpreter program must take to achieve the results the S-level utility program desires. In the example of figure I-12, step 6, it is desired to load additional user programsfrom the 80-column card I/O device to the disk cartridge. All machine functions are handled through the interpreter microprogram. Any other user or utility program to be executed would follow the same general pattern.

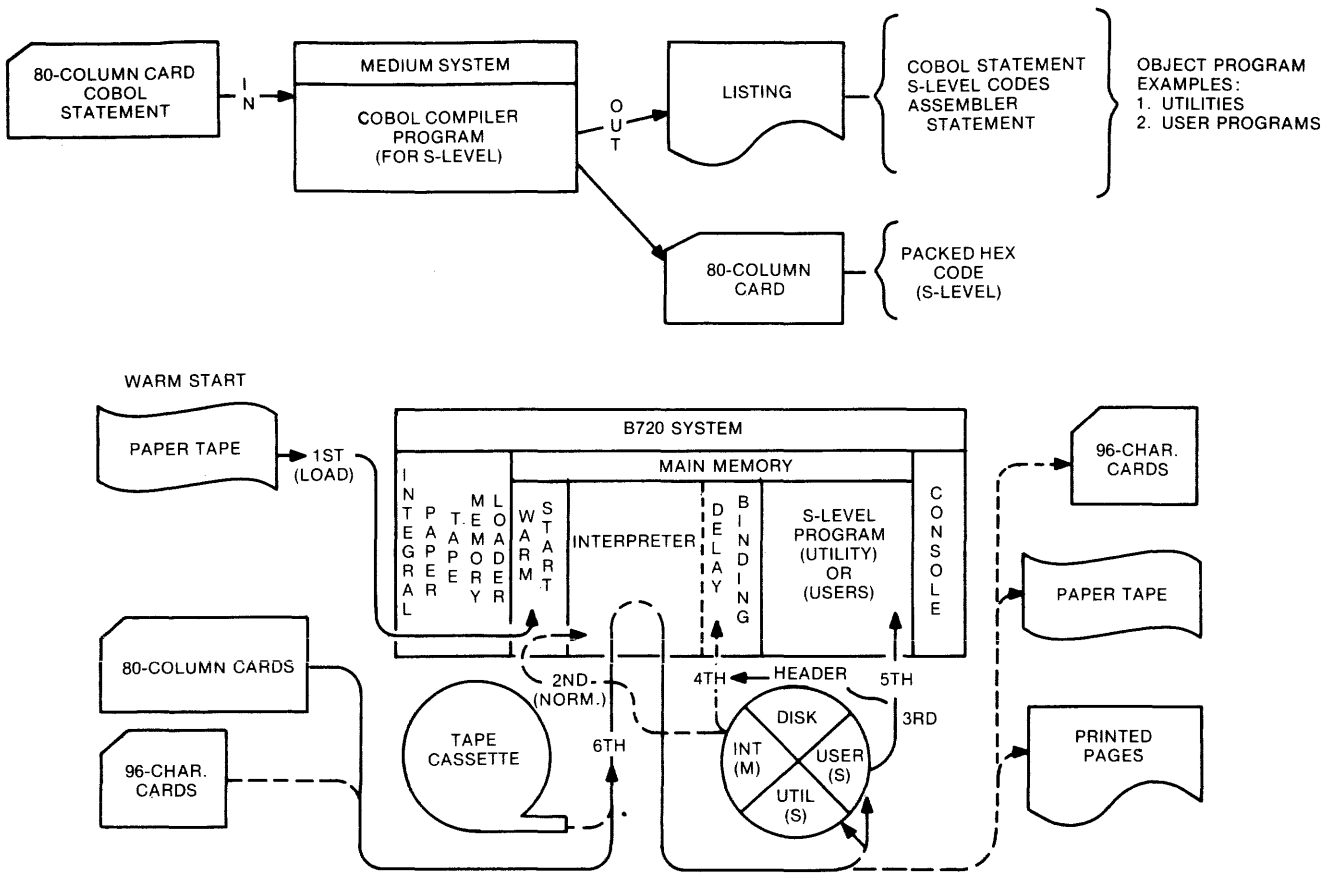


Fig. I-12 USER/UTILITY PROGRAM FLOW

Figure I-13 is a block diagram of loading the MTR program into the CPU. Unlike the users utilization of the processing system, the field engineer uses Maintenance Test Routines (MTR), which are stand-alone microprograms that exercise only the hardware specified within the test routine. No interpreter program is necessary for these applications. All MTR's are loaded to the main memory through the integral memory loader (paper tape reader) and the load interface control in conjunction with the load switch. At the completion of the load, control is turned over to the MTR program.

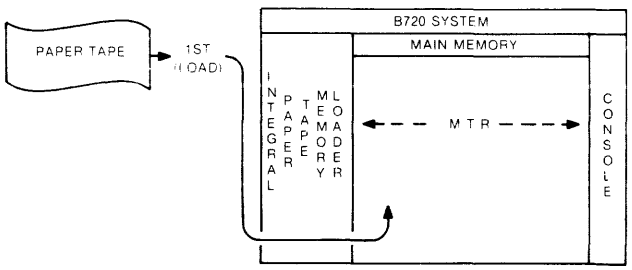


Fig. I-13 MTR FLOW

GLOSSARY OF TERMS AND SYMBOLS

This glossary defines the abbreviations, acronyms, and terms used in the system.

Building block	The primary functional units of an interpreter-based system configuration.
B7MPL Assembler	A program utilizing English written source statements to produce binary patterned code for micromemory and/or nanomemory as its object output.
B7MPL source Language	B 700 Microprogramming Language. A set of abbreviated English statements used as source to the B7MPL Assembler.
Central Processor Unit (CPU)	The central unit in a system (system processor).
COBOL	Common business Oriented Language used as a source language for compiling S-level programs.
Compiler program or assembler program	A manufacturer-furnished program capable of translating a symbolic coded program (such as COBOL or FORTRAN) into machine code (machine executable statements). The input data to a compiler (symbolic coded program) is called the source program (source deck if cards), and the output data is code (machine executable statements) called the object program.
CPM	Synonomous with DPM, Central Processor Memory or Data Program Memory.
CPU	Central Processing Unit.
Data/program memory (DPM) (also "S" memory)	Provides storage for data and conventional program in an emulation application and functions similarly to the main memory of a conventional computer system.
Device	Peripheral equipment such as disk pack drives, magnetic tape units, line printers, card readers, and various sensors usually found in special data processing applications. The function of devices is to provide the unique input/output medium for each system application.
Delayed Binding	The non-commitment of resources until the actual need for these resources arises. Binding occurs at program load (from disk) time.
Device Dependent Port (DDP)	A device dependent port (DDP) and an installed Input/Output Control (IOC) permits a peripheral device to be interfaced with the port selector unit. This is accomplished by providing specific device electrical interfacing such as, logic level conversion, line driver/receiver capability, and timing and synchronization when required (as in the case of disk pack drives and magnetic tape units).
Disk Directory	A table on disk pointing to the location of each file in a disk subsystem or disk pack.
DMAC	Direct memory Access Channel independent from CPU intervention.
End-around shift	A right-shift operation in which the bit or bits which would be shifted out of a register are reinserted into the most-significant end of the request.
End-off shift	A shift operation in either the left or right direction, in which the bit or bits shifted out of the register are lost. Vacated bit positions may be automatically filled with zeros.
Emulate	Imitate another system or machine hardware registers and machine language structure.
Fatal errors	Erros that impare the proper functioning of a system, such as hardware, address-limit, or I/O errors. Fatal errors provide alert alarm, print, display, and program halt indications.
Firmware	Firmware consist of the combination of microinstructions and nanoinstructions which are contained in the firmware store (microprogram memory and nanomemory).
Firmware program	A firmware program refers to one or several microprograms which constitute a fully operation program.
Firmware store	The Microprogram Memory (MPM) and Nanomemory (NPM) which contain the microinstructions and nanoinstructions.
Host system	The primary system in a multi-system or subsystem network.
Input/Output Control (IOC)	Interfacing circuitry between peripheral device and PSU. (Refer to DDP.)
Interpreter	A firmware program capable of emulating another machine instruction set.
Katakana	A Japanese business-oriented character set.
Large Scale Integration (LSI)	The implementation of more than 100 bipolar logical gates in a single integrated circuit chip.
LRC	Longitudinal Redundancy Character.

M-Level Language (M-Language)	Longitudinal Redundancy Character. Microlanguage; a language that is used for developing microprogramings.
Medium Scale Integration (MSI)	The implementation of 20 to 100 bipolar gates in a single integrated circuit chip.
Microinstruction (M-instruction)	A discrete instruction, which is contained in the Microprogram Memory (MPM) of the processor unit that performs a set of basic functions in parallel. A microinstruction is comprised of a unique set of microcodes and may contain a constant (literal) or an address of a nanoinstruction in nanomemory. A Type I mocrocode specifies an address of a nanoinstruction in the Nanomemory. A Type II microcode specifies constant values to be loaded into the Shift Amount Register (SAR), Alternate Microprogram Count Register (AMPCR), Literal register (LIT), or both the SAR and LIT registers, also includes MPCR and INCR.
Microprogram (M-program)	A microprogram is comprised of a set of microinstructions or microsequences.
Microsequence (M-sequence)	A sequence of microinstructions that performs a basic function.
Microprogram Memory (MPM or M memory)	The Microprogram Memory (MPM) is one of the major functional units of the processor unit which stores microinstructions that characterize the actual or virtual machine for a given application. The MPM can be implemented as a read-only memory (ROM) or as a read/write memory (RWM).
MPAD	Microprogram Address Controls.
MTR	Maintenance Test Routine.
Nanocodes	A set of 56 code bits that provide control of the desired logical function within the processor unit.
Nanoinstruction	A single instruction stored in the nanomemory of the firmware store, the contents of which constitute 56 unique nanocode signals for controlling the hardware logic of the processor unit.
Nanomemory	The nanomemory is one of the major functional units of the processor unit that stores 56 specific enable signals (nanobits).
Overlay area	A reserved area in memory for placing additional segments of a program that would normally be too large to be placed in memory all at once.
Program generator	This is a program capable of generating complete programs according to specifications received from the programmer.
Random-Access Memory (RAM)	A memory that stores data that can be read, altered, and rewritten in any location sequence.
Read-Only Memory (ROM)	A memory that sores data not alterable by program instruction.
RPG	Report Program Generator. (Refer to program generator.)
S-Instruction	A primitive instruction which emulates a machine instruction in a conventional computer.
S-Level language (S-language)	A language equivalent to the assembly or object language in a conventional computer, such as: source language, advanced assemblers, basic assemblers, and machine language.
S-Memory	The memory which contains the S-programs that are executed and the data on which these S-program operate. The S-memory is also referred to as SM.
S-Program	A named set of S-instructions.
SPM	Scratch Pad Memory. A variable area in main memory defined by the users program.
SPO	Supervisory Printer Output. (System Console).
Small Scale Integration (SSI)	The implementation of 5 to 20 logical gates in a single integrated circuit chip.
Transistor-Transistor-Logic (TTL)	A family of transistor circuits used to implement digital logic networks, and characterized by its high speed, large capacitance drive capability, and excellent noise immunity.
Virtual Machine	The image, characteristics, and functional results of one specific machine emulated by another.
Z Register section	The Z register section contains a collection of registers and selection gates in the Memory Control Units (MU's) of the processor unit, which include a loadable Counter (CTR) and the Literal register (LIT).

B 720
PROCESSOR

SECTIO
II

FUNCTIONAL
DETAIL

Burroughs

FIELD ENGINEERING

TECHNICAL MANUAL

Printed in U.S. America 7-30-75 For Form 1077534

SECTION II

INTRODUCTION

This section presents the functional detail of the Central Processing Unit (CPU) in the following four subsections to aid in locating specific functional detail descriptions:

a. Processor section.
b. Main memory section.
c. I/O device control section.
d. Power supply section.

Figures II-1 and II-2 are general and detailed functional block diagrams of the CPU.

PROCESSOR SECTION

The processor section is the major building block of the computer system which contains the minimally-committed logic or hardware that is controlled by the program residing in the firmware store. The firmware store is comprised of a Microprogram Memory (MPM) and Nanomemory (NM). The Microprogram Memory (cards MADD-1, MADD-2, MOB, MPA, and MPB) provides the logical sequence of operations in the order that the processor must perform them, as defined by the microinstructions. The Nanomemory (cards NM14, NM15, NM6, and NM7) provides the logical control levels to the processor and to the I/O device control section.

The processor minimally committed circuits which perform operations directed by the microprogram are as follows:

a. Logic Unit (cards LU1, LU2, LU5, LU6, and LU7).
b. Control Unit (cards CU1 and CU2).
c. Memory Control Unit (cards MU2 through MU6).
d. External Operation Control (card EO2).
e. Shared Memory Control (cards SM4 through SM7).
f. Clock Generation and Distribution (cards CG2 and CD1).

These circuits are described in the following paragraphs. (Refer to the Main Memory subsection for a description of shared memory control circuits.)

FIRMWARE STORE (MPM AND NM)

The firmware store consists of the Microprogram Memory (MPM) and Nanomemory (NM). The MPM, which is one of the major functional units of the processor, contains the microinstructions that provide the logical sequence of operations to be performed by the processor. The microinstruction consists of two levels: the microcode and the nanocode. The microcode consists of 16 bits which may contain a nanomemory address for selecting a nanocode, or preset values that are to be loaded into the Literal register (LIT), Shift Amount Register (SAR), Alternate Microprogram Count Register (AMPCR), or the Incrementer (INCR) and Microprogram Count (MPCR) Registers. The Nanomemory provides the logic control levels to the processor and the port selector. The nanocode, which is contained in the nanomemory, consists of 56 bits that are used to enable 56 nanobit control levels in parallel. Figure II-3 shows the input and output control signals of the microprogram memory and nanomemory.

The Incrementer register output signals (INCRX through INCR12), which are generated in the Memory Control Unit (MU6-1 through MU6-4), provide the addressing capability of the Microprogram Memory through shared memory controls ADR1-16. The WRITE signal from the EO2 card and the memory data inputs (DTM01 through DTM17), along with Start Memory Cycle, are inputs because a Read/Write Memory (RWM) configuration is used for the microprogram memory.

The Microprogram Memory output bits (MPM1 through MPM17) are sent to the various control circuits in the processor as follows:

a. MPM bits 1 through 4 are used by the Microprogram Address Controls (MPAD CNTL) on the card MU2.
b. Bits 3 through 16 are inputs to the Alternate Microprogram Count Register (AMPCR) on cards MU6-1 through MU6-4.
c. Bits 9 through 16 are inputs to the Literal register (LIT) on cards MU6-2 and MU6-3.
d. Bits 5 through 8 are inputs to the Shift Amount Register (SAR) input select gates on card CU2.
e. MPM bit 17 is used for parity checking.

The microcode format is shown in figure II-4 for further details about the MPM bits. MPM bits 6 through 16 (least significant 11 bits) are used by the nanomemory as group and address information. Nanomemory consists of two groups of 256 addressable 56-bit nanowords.

Nanomemory is a read-only memory constructed from LSI chips assembled on four individual printed circuit cards. Each card contains seven memory chips and is capable of producing 256 different output combinations. Thus, one card can produce 256 half-nanowords (28 bits). Table II-1 lists all available nanowords. The nanobit output signals (N01 through N55) are used as enable control levels in the processor. The nanoword contents and nanocode format information are provided in figure II-4.

The firmware store (MPM and nanomemory) is characterized by the unique ability of maintaining changeable nanocodes and microinstruction phasing. The microcode controls and nanocode controls, which are generated by microprogramming, are reflected in the contents of either the microprogram memory or nanomemory.

MICROCODE CONTROLS

The microcode controls stored within the Microprogram Memory can perform any one of the following functions:

a. Set the Shift Amount Register (SAR).

b. Set both the Shift Amount Register (SAR) and Literal register (LIT).

c. Set the Literal register (LIT).

d. Specify the microprogram memory address for the Alternate Microprogram Count Register (AMPCR).

e. Specify the microprogram memory address for the Incrementer (INCR) and microprogram count register (MPCR).

f. Specify a nanomemory address of a nanocode.

NANOCODE CONTROLS

The nanocode controls contained in the nanomemory may be constructed in a general or specific manner relative to the functions that the nanocode controls are to perform. A nanocode control may specify any of the following operations:

a. Test specific conditions (Condition Selection).

b. Select successors (Successor Condition).

c. Select Logic Unit conditions (Phase 3 Modifier).

d. Select external operation conditions (Memory/Device/Condition Adjust Modifier).

e. Set specific flag bits (Condition Adjust).

f. Select specific operands upon which logical operations are to be performed (MPAD Controls).

g. Select inputs to the Arithmetic/Logic Unit (X-Selection to Adder and Y-Selection to Adder).

h. Inhibit the bit carries for the allowable Adder operations (Inhibit Carry Between Bytes).

i. Specify any of the allowable Arithmetic/Logic Unit operations (Adder Operation).

j. Shift the output of the Adder (Shift Selection).

k. Select specific destinations which are to receive inputs or remain unchanged during a specific operation (A Register Input Selection, B Register Input Selection, MIR Input Selection, or AMPCR Input Selection).

l. Select input data to the Base Registers (BR1 or BR2), Memory Address Register (MAR), Counter (CTR), and Shift Amount Register (SAR).

m. Read from or write to a memory or external device (memory/Device Operation).

n. Perform the sign save control function.

MICROINSTRUCTION PHASING

The execution of a microinstruction requires one or more sequential time periods designated phase 1, phase 3, or extended phase 3. The interval of time from the trailing edge of one clock pulse to the trailing edge of the next clock pulse is considered as one phase of a microinstruction. (See figure II-5.)

Some microinstructions can occur during phase 1, while others can occur during phase 1 and phase 3 time intervals. Other microinstructions occur during a phase 1, extended phase 3, and phase 3 time interval. Phases of successive microinstructions always overlap one another, so that phase 1 of a current microinstruction can be executed simultaneously with phase 3 of a prior microinstruction. The overlapping of microinstruction phases allows for the start of the execution of a new microinstruction each time a new clock period starts. Therefore, it can be stated that one microinstruction can be executed in one clock time.

A microinstruction may contain either a constant (Type II microinstruction) or the address of a nanoinstruction (Type I microinstruction). For a Type II microinstruction, phase 1 pro-

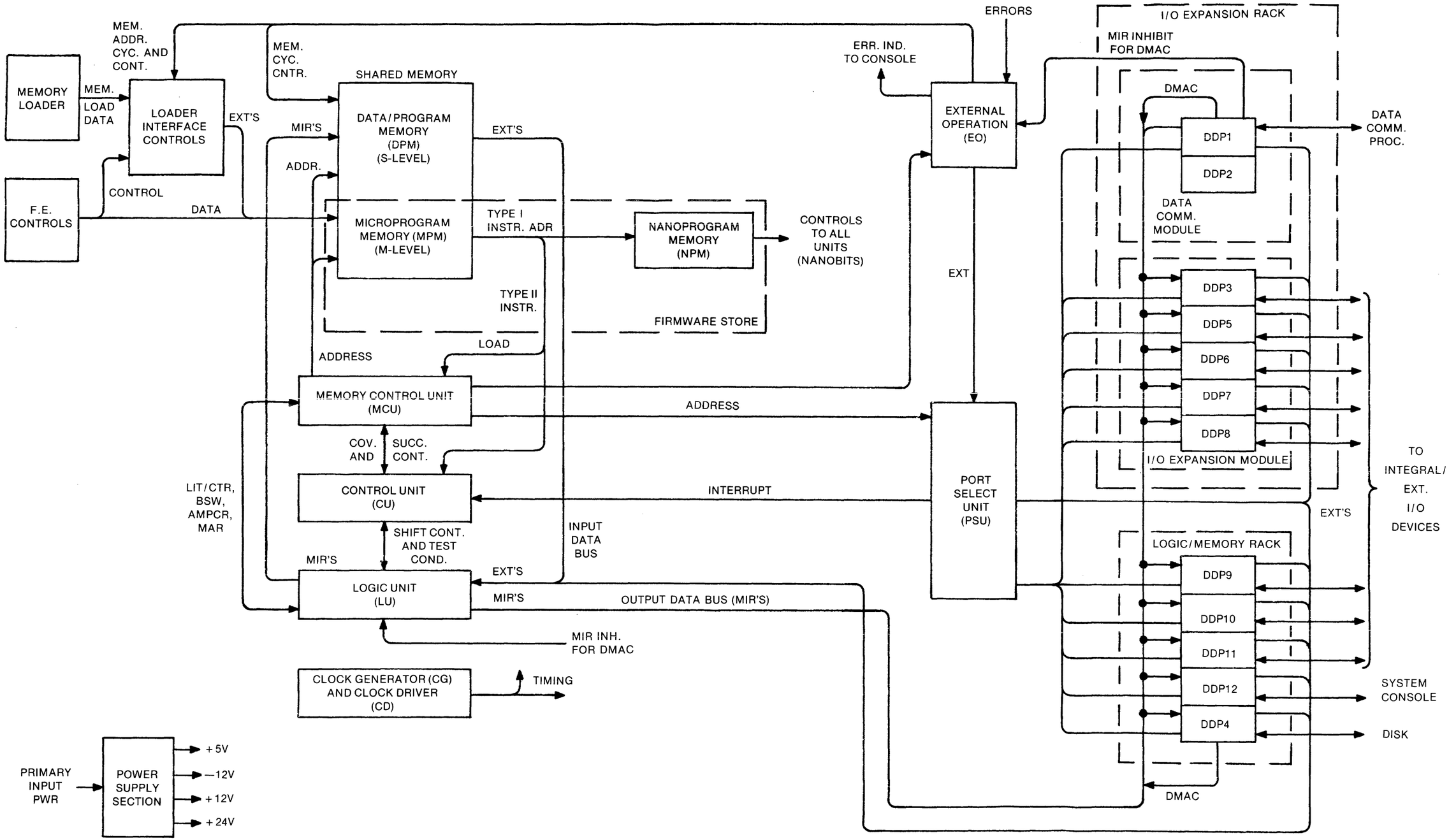


Fig. II-1 PROCESSING SYSTEM, GENERAL FUNCTIONAL BLOCK DIAGRAM

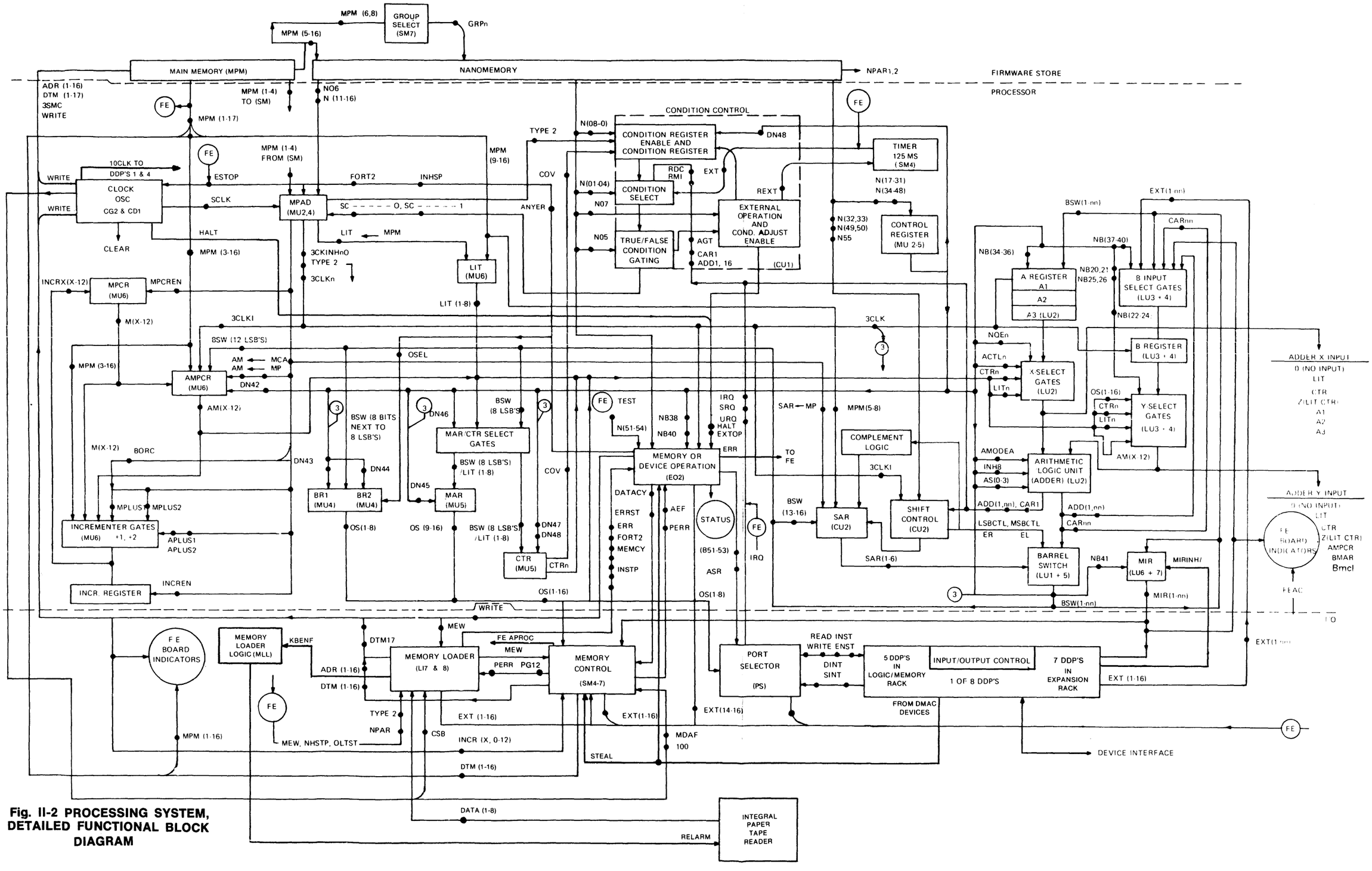


Fig. II-2 PROCESSING SYSTEM, DETAILED FUNCTIONAL BLOCK DIAGRAM

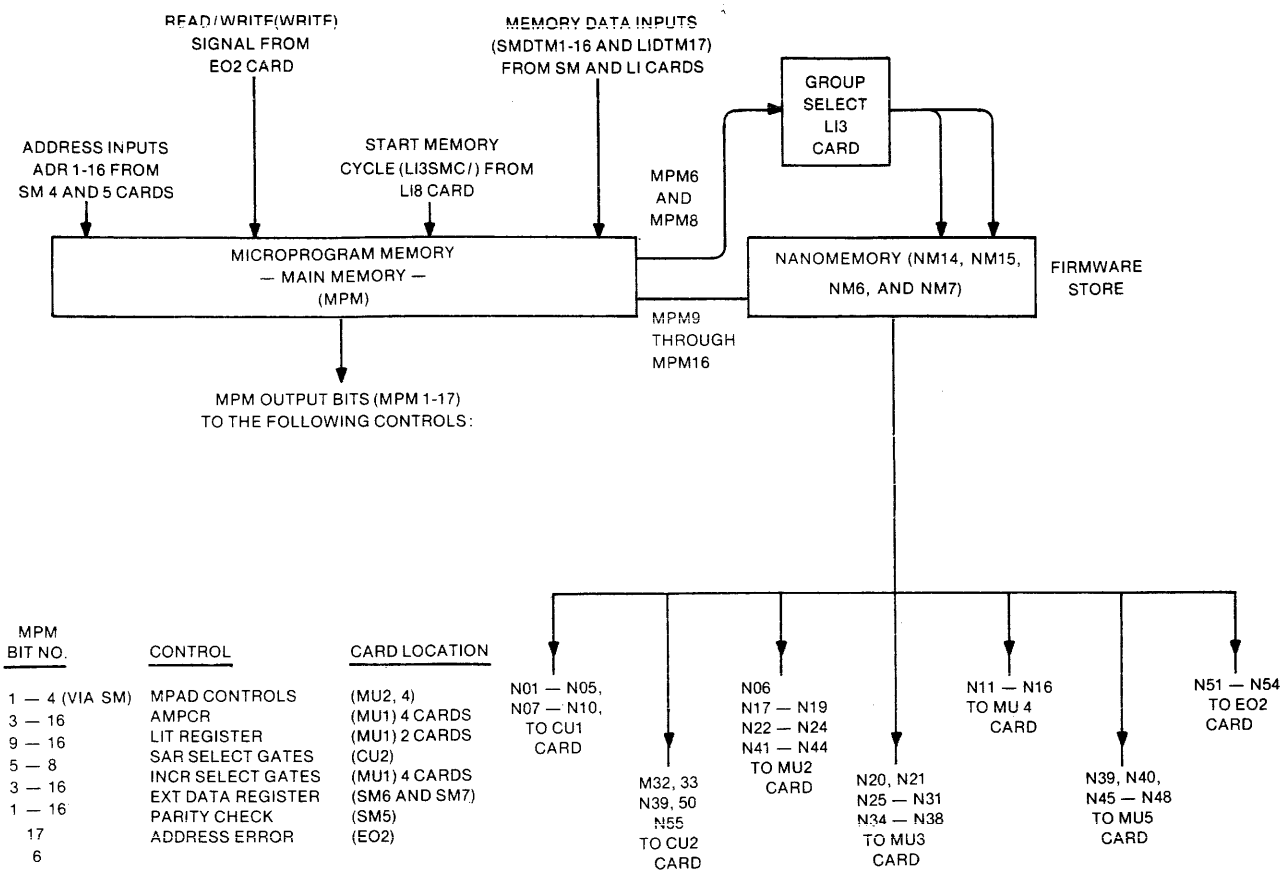


Fig. II-3 MICROPROGRAM MEMORY (MPM) AND NANOMEMORY (NPM), BLOCK DIAGRAM

vides sufficient time to execute the instruction (completion of the STEP successor and literal assignment) and therefore requires no additional phases. For a Type I microinstruction, the following information describes the events that occur in each of the following three phases:

Phase 1: Condition testing and adjusting, and selection of the controls for the next instruction address computation (successor microprogram address control), initiation of memory and device operations, and the setup for Logic Unit operation (phase 3). (See figure II-4.)

Extended Phase 3: Holding phase for Logic Unit operation phase 3 controls.

Phase 3: Initiation and completion phase for performing Logic Unit operations and to change destination registers as specified in the logic operation. (See figure II-4.)

All Type I microinstructions which do not contain a conditional logical operation always have a phase 1 and a phase 3. However, those Type I microinstructions which contain a conditional logical operation can be divided into two categories: those which meet the specified conditions, or those that do not meet the specified conditions. The Type I microinstructions that meet the conditions require a phase 1 and 3 for the execution of the operation to be completed. The Type I microinstructions that do not meet conditions require only a phase 1 for the complete execution of the operation. Phase 1 occurs at each clock time, and phase 3 completes the execution of a Logic Unit operation.

The controls for phase 3, which are contained in the Control Register of the Memory Control units (MU2 through MU5 cards) and the Shift Control Register of the CU2 card, are divided into two parts: the Logic Unit operations and the destination controls. The Logic Unit operations are as follows:

a. X-selection to Adder.
b. Y-selection to Adder.
c. Inhibit carry between bytes.
d. Adder operation.
e. Shift selection.

NANOWORD CONTENTS

1-4	Condition Selection
5	Successor Tested Condition
6	Phase 3 Modifier
7	Memory/Device/Condition Adjust Modifier
8-10	Condition Adjust
11-13	Microprogram Address Controls; SC=1
14-16	Microprogram Address Controls; SC=0
17-19	X - Selection to Adder
20-26	Y - Selection to Adder
27	Inhibit Carry Between Bytes
28-31	Adder Operation
32-33	Shift Selection
34-36	A Register Input Selection
37-40	B Register Input Selection
41	MIR Input Selection
42	AMPCR Input Selection
43	BR1 Input Selection
44	BR2 Input Selection
45-46	MAR Input Selection
46-48	Counter Input Selection
49-50	SAR Input Selection
51-54	Memory/Device Operation
55	Sign Store Control
56	Parity
1-16,51-54	Phase 1 Operations
17-33	Phase 3 Controls
34-50, 55	Destination Controls (Executed at Termination of Phase 3)

NOTES

CONDITION SELECTION MNEMONICS

ABT = ADDER BITS TRUE
AOV = ADDER OVERFLOW
COV = COUNTER OVERFLOW
EXT = EXTERNAL INTERRUPT REQUEST (125 MS TIMER)
GC1 = GLOBAL CONDITION 1
GC2 = GLOBAL CONDITION 2

LC3 = LOCAL CONDITION 3

IRQ = INPUT REQUEST
LC1 = LOCAL CONDITION 1
LC2 = LOCAL CONDITION 2
LST = ADDER LEAST SIGNIFICANT BIT
MST = ADDER MOST SIGNIFICANT BIT
RDC = READ COMPLETE
RMI = READY MIR
SRQ = SOLICITED REQUEST
URQ = UNSOLICITED REQUEST

NANOCODE

1 2 3 4	CONDSEL
0 0 0 0	GC1
0 0 0 1	GC2
0 0 1 0	LC1
0 0 1 1	LC2
0 1 0 0	MST
0 1 0 1	LST
0 1 1 0	ABT
0 1 1 1	AOV
1 0 0 0	COV
1 0 0 1	RMI
1 0 1 0	RDC
1 0 1 1	IRQ
1 1 0 0	EXT
1 1 0 1	LC3
1 1 1 0	SRQ
1 1 1 1	URQ

5 **FT** (Successor Condition)
0 SC = 1 if CONDSEL is FALSE
1 SC = 1 if CONDSEL is TRUE

6 **LUC** (Phase 3 Modifier)
0 Do Phase 3 operations unconditionally
1 Do Phase 3 operations if SC = 1

7 **MDC** (MDOP & CNDADJ Modifier)
0 Do MDOP and CNDADJ unconditionally
1 Do MDOP and CNDADJ if SC = 1

8 9 10	CNDADJ	(Do if MDC = 1)
0 0 0	-------	
0 0 1	Set LC2	
0 1 0	Set GC2	
0 1 1	Reset GC2	
1 0 0	Set LC3	
1 0 1	Reset GC1	
1 1 0	Set GC1	
1 1 1	Set LC1	

MICROCONTROLS

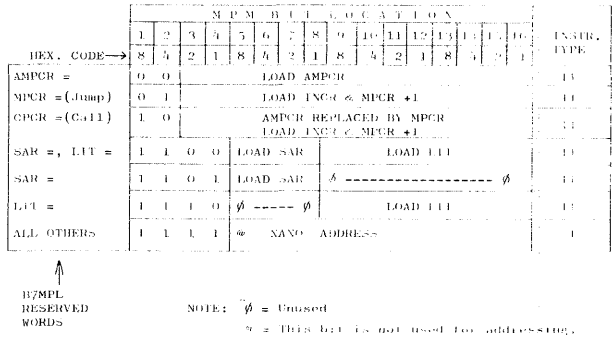


Fig. II-4 MICROCODE FORMAT, NANOWORD CONTENTS, AND NANOCODE FORMAT (SHEET 1 OF 2)

MPAD CONTROLS

Used if SC=1 (11 12 13)			Used if SC=0 (14 15 16)			
0	0	0	0	0	0	WAIT
0	0	1	0	0	1	STEP
0	1	0	0	1	0	SAVE
0	1	1	0	1	1	SKIP
1	0	1	1	0	1	JUMP
1	0	0	1	0	0	EXEC
1	1	0	1	1	0	CALL
1	1	1	1	1	1	RETN

17 18 19	X Selection to Adder
0 0 0	Zero
0 0 1	LIT (To LS Byte)
0 1 1	CTR (To MS Byte)
1 0 0	Z (CTR /LIT)
1 0 1	A1
1 1 0	A2
1 1 1	A3

20 21 22 23 24 25 26	Y Selection to Adder
0 0	Zero in MS Bit
0 1	B in MS Bit
1 0	$\overline{B}$ in MS Bit
1 1	1 in MS Bit
0 0 0	ZERO
0 0 0 0 1 0 1	LIT (To LS Byte)
0 1 0 1 1 0 0	CTR (To MS Byte)
1 0 0	B
0 1 1 0 1 0 1	Z (CTR/LIT)
0 0 1 1 0 0 1	AMPCR
0 1 1 1 1 0 1	BMAR (OS)
0 0	ZERO in LS Bit
0 1	B in LS Bit
1 0	$\overline{B}$ in LS Bit
1 1	1 in LS Bit

27	Inhibit Carry Between 8-bit Bytes
0	Allow
1	Inhibit

28 29 30 31	ADOP	MNEMONIC
0 0 0 0	X + Y	ADD
0 0 0 1	$\overline{X \vee Y}$	NOR
0 0 1 0	$\overline{X} \cdot Y$	NRI
0 0 1 1	X + Y + 1	ADD + 1
0 1 0 0	$\overline{X \cdot Y}$	NAND
0 1 0 1	(X ∨ Y) + X	OAD
0 1 1 0	X ⊕ Y	XOR
0 1 1 1	$X \cdot \overline{Y}$	NIM
1 0 0 0	$\overline{X} \vee Y$	IMP
1 0 0 1	$\overline{X \oplus Y}$	EQV
1 0 1 0	X + (X · Y)	AAD
1 0 1 1	X · Y	AND
1 1 0 0	X - Y - 1	SUB-1
1 1 0 1	$X \vee \overline{Y}$	RIM
1 1 1 0	X ∨ Y	OR
1 1 1 1	X - Y	SUB

32 33	Shift Selection
0 0	–
0 1	R
1 0	L
1 1	C

34 35 36	A Register Input Selection
0	A1 Unchanged
1	BSW → A1
0	A2 Unchanged
1	BSW → A2
0	A3 Unchanged
1	BSW → A3

37 38 39 40	B Register Input Selection
0 0 0 0	None
0 0 0 1	Complement of 4-bit carries (BC4)
1 0 0 0	ADDER (BAD)
1 0 0 1	Complement of 8-bit carries (BC8)
1 0 1 0	BSW ∨ ADDER (BBA)
1 0 1 1	BSW
1 1 0 0	EXT (BEX)
1 1 0 1	MIR (BMI)
1 1 1 0	BSW ∨ EXT (BBE)
1 1 1 1	BSW ∨ MIR (BBI)

41	MIR Input Selection
0	No Change
1	BSW → MIR

42	AMPCR Input Selection
0	No Change
1	BSW Least Bits → AMPCR

43	BR1 Input Selection
0	No Change
1	BSW Next Least Byte → BR1

44	BR2 Input Selection
0	No Change
1	BSW Next Least Byte to BR2

45 46	MAR Input Selection
0	No Change
1 0	LIT to MAR
1 1	BSW Least Byte to MAR

46 47 48	Counter Input Selection
- 0 0	No Change
0 0 1	$\overline{LIT}$ to CTR (LCTR)
1 0 1	$\overline{BSW}$ Least Byte to CTR (CTR)
- 1 0	Increment CTR (MOD 256) (INC)

49 50	SAR Input Selection
0 0	No change
0 1	Complement SAR (CSAR)
1 0	BSW Least Bits → SAR (SAR)

51 52 53 54	MDOP
0 0 0 0	No Change
0 0 0 1	– – –
0 0 1 0	MR1
0 0 1 1	MR2
0 1 0 0	DLD
0 1 0 1	– – –
0 1 1 0	MW1
0 1 1 1	MW2
1 0 0 0	ASR
1 0 0 1	ASE
1 0 1 0	DR1
1 0 1 1	DR2
1 1 0 0	USEL
1 1 0 1	USEL
1 1 1 0	DW1
1 1 1 1	DW2

55	SSC
0	$\overline{SHIFT}$ - NO ACTION/SHIFT - ENDOFF → SSF
1	$\overline{SHIFT}$-AOV → SSF/SHIFT-ENDOFF → SSF → BSW

56	
0	PARITY (ON OR OFF TO MAKE TOTAL BITS ODD)
1	

Fig. II-4 MICROCODE FORMAT, NANOWORD CONTENTS, AND NANOCODE FORMAT (SHEET 2 OF 2)

TABLE II-1. NANO LIST

MICRO CODE	NANO ADDRS		NANO CODE
		FIRST GROUP OF 256	
F000	0000 *	WAIT.	0000 0000 0000 0100
F001	0001	A1.	0009 A000 0000 0100
F002	0002 *	IF GC1 STEP ELSE SKIP.	080B 0000 0000 0100
F003	0003	A1 EQV B.	0009 AC52 0000 0000
F004	0004	A1 EQV B000.	0009 A012 0000 0100
F005	0005 *	IF GC1 STEP ELSE RETN.	080F 0000 0000 0000
F006	0006 *	SET GC1.	0189 0000 0000 0100
F007	0007	A1 – AMPCR.	0009 0640 4000 0100
F008	0008 *	IF GC1 SKIP.	0819 0000 0000 0100
F009	0009	A1 = A1 AND LIT.	0009 A156 4000 0100
F00A	000A	A1 = A1 – LIT.	0009 A15E 4000 0000
F00B	000B	A1 = A1 L.	0009 A001 4000 0100
F00C	000C *	RESET GC1.	0149 0000 0000 0100
F00D	000D	A1 = A1 R.	0009 A000 C000 0100
F00E	000E *	IF NOT GC1 EXEC.	0021 0000 0000 0100
F00F	000F	A1 = A1 – 1.	0009 A0DE 4000 0000
F010	0010	A1 = A1 + B.	0009 AC40 4000 0100
F011	0011	A1 = A1 + LIT.	0009 A140 4000 0000
F012	0012	A1 = A1 + 1.	0009 A0C0 4000 0000
F013	0013	A1 = A2.	0009 C000 4000 0000
F014	0014 *	IF LC1 SKIP..	2819 0000 0000 0000
F015	0015 *	IF LC2 SKIP.	3819 0000 0000 0100
F016	0016	A1 = A3.	0009 E000 4000 0100
F017	0017	A1 = B.	0009 0C40 4000 0100
F018	0018 *	SET LC1.	01C9 0000 0000 0000
F019	0019 *	SET LC2.	0049 0000 0000 0000
F01A	001A *	SET LC3.	0109 0000 0000 0000
F01B	001B	A1 = B L.	0009 0C41 4000 0000
F01C	001C	A1 = BMAR.	0009 0F40 4000 0100
F01D	001D	A1 = LIT.	0009 2000 4000 0100
F01E	001E *	IF LC3 SKIP.	D819 0000 0000 0000
F01F	001F *	RESET GC2.	00C9 0000 0000 0100
F020	0020	A1 = LIT + B.	0009 2C40 4000 0000
F021	0021	A2.	0009 C000 0000 0100
F022	0022	A2 EQV B000.	0009 C012 0000 0100
F023	0023	A2 = AMPCR.	0009 0640 2000 0100
F024	0024	A2 = A1.	0009 A000 2000 0000
F025	0025	A2 = A2 C.	0009 C001 A000 0000
F026	0026	A2 = A2 L.	0009 C001 2000 0100
F027	0027	A2 = A2 R.	0009 C000 A000 0100
F028	0028	A2 = A2 OR B.	0009 CC5C 2000 0000
F029	0029	A2 = A2 + LIT.	0009 C140 2000 0000
F02A	002A	A2 = A2 + 1.	0009 C0C0 2000 0000
F02B	002B	A2 = A2 – 1.	0009 C0DE 2000 0000
F02C	002C	A2 NAN LIT.	0009 C148 0000 0000
F02D	002D	A2 = A2 AND LIT.	0009 C156 2000 0100
F02E	002E *	IF COV SKIP.	8819 0000 0000 0000
F02F	002F	A2 = A2 OR LIT.	0009 C15C 2000 0100

NOTE: * INDICATES ENTRY OUT OF ALPHABETICAL ORDER.

TABLE II-1. NANO LIST (Cont.)

MICRO CODE	NANO ADDRS		NANO CODE
F030	0030 *	INC.	0009 0000 0002 0000
F031	0031	A2 = A2 + B.	0009 CC40 2000 0100
F032	0032	A2 - A3.	0009 E000 2000 0100
F033	0033	A2 = B.	0009 0C40 2000 0100
F034	0034	A2 = B000.	0009 0000 2000 0000
F035	0035	A2 - BMAR.	0009 0F40 2000 0100
F036	0036	A2 = B L.	0009 0C41 2000 0000
F037	0037	A2 = B R.	0009 0C40 A000 0000
F038	0038	A2 = LIT.	0009 2000 2000 0100
F039	0039	A2 = LIT L.	0009 2001 2000 0000
F03A	003A	A2 = LIT + B.	0009 2C40 2000 0000
F03B	003B	A3.	0009 E000 0000 0000
F03C	003C	A3 EQV B.	0009 EC52 0000 0100
F03D	003D	A3 EQV B000.	0009 E012 0000 0000
F03E	003E	A3 EQV LIT.	0009 E152 0000 0000
F03F	003F	A3 EQV Z.	0009 ED52 0000 0000
F040	0040	A3 – B.	0009 EC5E 0000 0100
F041	0041	A3 = AMPCR.	0009 0640 1000 0100
F042	0042	A3 = Al.	0009 A000 1000 0000
F043	0043	A3 = A2.	0009 C000 1000 0000
F044	0044	A3 = A3 C.	0009 E001 9000 0100
F045	0045	A3 = A3 L.	0009 E001 1000 0000
F046	0046	A3 = A3 R.	0009 E000 9000 0000
F047	0047	A3 = A3 AND LIT.	0009 E156 1000 0000
F048	0048	A3 = A3 OR B.	0009 EC5C 1000 0100
F049	0049	A3 = A3 XOR B.	0009 EC4C 1000 0000
F04A	004A	A3 = A3 OR LIT.	0009 E15C 1000 0000
F04B	004B	A3 = A3 – 1.	0009 E0DE 1000 0100
F04C	004C	A3 = A3 + 1.	0009 E0C0 1000 0100
F04D	004D	A3 = A3 + B.	0009 EC40 1000 0000
F04E	004E	A3 = A3 + LIT.	0009 E140 1000 0100
F04F	004F	A3 = A3 – LIT.	0009 E15E 1000 0100
F050	0050 *	IF ABT SET LC1.	6BC9 0000 0000 0000
F051	0051	A3 = B.	0009 0C40 1000 0100
F052	0052	A3 = B L.	0009 0C41 1000 0000
F053	0053	A3 = B R.	0009 0C40 9000 0000
F054	0054	A3 = B000.	0009 0000 1000 0000
F055	0055	A3 = B001.	0009 00C0 1000 0000
F056	0056	A3 = BMAR.	0009 0F40 1000 0100
F057	0057	A3 = LIT L.	0009 2001 1000 0000
F058	0058	A3 = LIT.	0009 2000 1000 0100
F059	0059	A3 = LIT AND B.	0009 2C56 1000 0100
F05A	005A	A3 = Z.	0009 8000 1000 0100
F05B	005B	AMPCR = B.	0009 0C40 0040 0100
F05C	005C	AMPCR = A1.	0009 A000 0040 0000
F05D	005D	ASE.	0009 0000 0000 2500
F05E	005E	ASR.	0009 0000 0000 2000
F05F	005F	B.	0009 0C40 0000 0000
F060	0060	B = AMPCR.	0009 0640 0800 0100
F061	0061	B = Al L.	0009 A001 0B00 0100
F062	0062	B = A1 AND LIT.	0009 A156 0B00 0100

NOTE: * INDICATES ENTRY OUT OF ALPHABETICAL ORDER.

TABLE II-1. NANO LIST (Cont.)

MICRO CODE	NANO ADDRS		NANO CODE
F063	0063	B = A1 + B.	0009 AC40 0B00 0100
F064	0064	B = A1 – B.	0009 AC5E 0B00 0100
F065	0065	B = A1 OR B.	0009 AC5C 0B00 0000
F066	0066	B = A1.	0009 A000 0B00 0000
F067	0067	B = A1 R.	0009 A000 8B00 0100
F068	0068	B = A2.	0009 C000 0B00 0000
F069	0069	B = A2 R.	0009 C000 8B00 0100
F06A	006A	B = A2 L.	0009 C001 0B00 0100
F06B	006B	B = A2 + B.	0009 CC40 0B00 0100
F06C	006C	B = A2 OR B.	0009 CC5C 0B00 0000
F06D	006D	B = A2 AND LIT.	0009 C156 0B00 0100
F06E	006E	B = A3 + B.	0009 EC40 0B00 0000
F06F	006F	B = A3 OR B.	0009 EC5C 0B00 0100
F070	0070	B = A3 – B.	0009 EC5E 0B00 0000
F071	0071	B = A3.	0009 E000 0B00 0100
F072	0072	B = A3 R.	0009 E000 8B00 0000
F073	0073	B = A3 L.	0009 E001 0B00 0000
F074	0074	B = A3 AND LIT.	0009 E156 0B00 0000
F075	0075	B = BTT0.	0009 0C00 0B00 0000
F076	0076	B = B111.	0009 001A 0B00 0100
F077	0077	B = B001.	0009 00C0 0B00 0000
F078	0078	B = B0TT.	0009 0440 0B00 0000
F079	0079	B = B100.	0009 1800 0B00 0000
F07A	007A	B = B1TT.	0009 1C40 0B00 0000
F07B	007B	B = B000.	0009 0000 0B00 0000
F07C	007C	B = B L.	0009 0C41 0B00 0000
F07D	007D	B = B C.	0009 0C41 8B00 0100
F07E	007E	B = B + 1.	0009 0C46 CB00 0100
F07F	007F	B = B R.	0009 0C40 8B00 0000
F080	0080	B = BMAR.	0009 0F40 0B00 0100
F081	0081	B = NOT CTR R.	0009 6010 8B00 0000
F082	0082	B = LIT L.	0009 2001 0B00 0000
F083	0083	B = LIT + B.	0009 2C40 0B00 0000
F084	0084	B = LIT NRI B.	0009 2C44 0B00 0100
F085	0085	B = LIT XOR B.	0009 2C4C 0B00 0000
F086	0086	B = LIT AND B.	0009 2C56 0B00 0100
F087	0087	B = LIT OR B.	0009 2C5C 0B00 0100
F088	0088	B = LIT – B.	0009 2C5E 0B00 0000
F089	0089	B = LIT.	0009 2000 0B00 0100
F08A	008A	B = Z.	0009 8000 0B00 0100
F08B	008B	BBI = B.	0009 0C40 0F00 0000
F08C	008C	BMI, MIR = B.	0009 0C40 0D80 0000
F08D	008D	BMI.	0009 0000 0D00 0000
F08E	008E	BR2 = A2 OR B100.	0009 D81C 0010 0100
F08F	008F	BR2 = LIT L.	0009 2001 0010 0000
F090	0090	BR2 = A2.	0009 C000 0010 0000
F091	0091	CALL.	0036 0000 0000 0100
F092	0092	CSAR.	0009 0000 0000 4000
F093	0093	CSAR, B = B L.	0009 0C41 0B00 4100
F094	0094	CTR = B001 + 1.	0009 00C6 0005 0100
F095	0095	CTR = B.	0009 0C40 0005 0000
F096	0096	DR1 BEX.	0009 0000 0C00 2900
F097	0097	DR2 BEX.	0009 0000 0C00 2C00

TABLE II-1. NANO LIST (Cont.)

MICRO CODE	NANO ADDRS		NANO CODE
F098	0098	DW2.	0009 0000 0000 3D00
F099	0099	DW1.	0009 0000 0000 3800
F09A	009A	EXEC.	0024 0000 0000 0100
F09B	009B	IF ABT LUOP, JUMP.	6C29 0000 0000 0000
F09C	009C	IF ABT JUMP.	6829 0000 0000 0100
F09D	009D	IF ABT SKIP.	6819 0000 0000 0100
F09E	009E	IF AOV SKIP.	7819 0000 0000 0000
F09F	009F	IF COV JUMP.	8829 0000 0000 0000
F0A0	00A0	IF EXT SKIP.	C819 0000 0000 0100
F0A1	00A1	IF GC1 JUMP.	0829 0000 0000 0100
F0A2	00A2	IF GC2 SKIP.	1819 0000 0000 0000
F0A3	00A3	IF IRQ EXEC.	B821 0000 0000 0100
F0A4	00A4	IF IRQ JUMP.	B829 0000 0000 0000
F0A5	00A5	IF LC1 STEP.	2809 0000 0000 0100
F0A6	00A6	IF LC1 JUMP.	2829 0000 0000 0000
F0A7	00A7	IF LC1 SET LC1 ELSE SKIP.	2BC8 0000 0000 0000
F0A8	00A8	IF LC1 SET LC1 SKIP.	2BD9 0000 0000 0000
F0A9	00A9	IF LC2 STEP.	3809 0000 0000 0000
F0AA	00AA	IF LC2 JUMP.	3829 0000 0000 0100
F0AB	00AB	IF LC2 SET LC2 ELSE SKIP.	3A4B 0000 0000 0100
F0AC	00AC	IF LC2 SET LC2 SKIP.	3A59 0000 0000 0100
F0AD	00AD	IF LC3 STEP.	D809 0000 0000 0100
F0AE	00AE	IF LC3 SET LC3 SKIP.	D819 0000 0000 0000
F0AF	00AF	IF LC3 SET LC3 ELSE SKIP.	D808 0000 0000 0000
F0B0	00B0	IF LST SET LC1.	5BC9 0000 0000 0000
F0B1	00B1	IF LST JUMP.	5829 0000 0000 0100
F0B2	00B2	IF LST SKIP.	5819 0000 0000 0100
F0B3	00B3	IF MST SKIP.	4819 0000 0000 0000
F0B4	00B4	IF NOT ABT JUMP.	6029 0000 0000 0000
F0B5	00B5	IF NOT ABT SKIP.	6019 0000 0000 0000
F0B6	00B6	IF NOT AOV JUMP.	7029 0000 0000 0100
F0B7	00B7	IF NOT AOV SKIP.	7019 0000 0000 0100
F0B8	00B8	IF NOT COV SKIP.	8019 0000 0000 0100
F0B9	00B9	IF NOT COV JUMP.	8029 0000 0000 0100
F0BA	00BA	IF NOT EXT SKIP.	C019 0000 0000 0000
F0BB	00BB	IF NOT GC1 SKIP.	0019 0000 0000 0000
F0BC	00BC	IF NOT GC2 SKIP.	1019 0000 0000 0100
F0BD	00BD	IF NOT IRQ SKIP.	B019 0000 0000 0100
F0BE	00BE	IF NOT LC1 JUMP.	2029 0000 0000 0100
F0BF	00BF	IF NOT LC1 SKIP.	2019 0000 0000 0100
F0C0	00C0	IF NOT LC2 SKIP.	3019 0000 0000 0000
F0C1	00C1	IF NOT LC3 SKIP.	D019 0000 0000 0100
F0C2	00C2	IF NOT LST JUMP.	5029 0000 0000 0000
F0C3	00C3	IF NOT LST SKIP.	5019 0000 0000 0000
F0C4	00C4	IF NOT MST SKIP.	4019 0000 0000 0100
F0C5	00C5	IF SRQ THEN DR2 BEX SKIP.	EE19 0000 0C00 2D00
F0C6	00C6	IF SRQ DW2 SKIP.	EA19 0000 0000 3D00
F0C7	00C7	IF URQ SET LC2 ELSE JUMP.	FA4D 0000 0000 0100
F0C8	00C8	INC IF COV SKIP.	8819 0000 0002 0100
F0C9	00C9	JUMP.	002D 0000 0000 0100
F0CA	00CA	LCTR.	0009 0000 0001 0000
F0CB	00CB	LIT EQV B.	0009 2C52 0000 0100

TABLE II-1. NANO LIST (Cont.)

MICRO CODE	NANO ADDRS		NANO CODE
F0CC	00CC	LIT – B.	0009 2C5E 0000 0100
F0CD	00CD	LIT NAN B.	0009 2C48 0000 0000
F0CE	00CE	LMAR.	0009 0000 0008 0000
F0CF	00CF	MAR1 = AMPCR.	0009 0640 002C 0100
F0D0	00D0	MAR1 = A1.	0009 A000 002C 0000
F0D1	00D1	MAR1 = A2.	0009 C000 002C 0000
F0D2	00D2	MAR1 = A3.	0009 E000 002C 0100
F0D3	00D3	MAR1 = A3 + LIT.	0009 E140 002C 0100
F0D4	00D4	MAR1 = A3 + 1.	0009 E000 002C 0100
F0D5	00D5	MAR1 = B.	0009 0C40 002C 0100
F0D6	00D6	MAR1 = B + 1.	0009 0C46 002C 0100
F0D7	00D7	MAR1 = BMAR + 1.	0009 0F46 002C 0100
F0D8	00D8	MAR1 = LIT.	0009 2000 002C 0100
F0D9	00D9	MAR1 = LIT + B.	0009 2C40 002C 0000
F0DA	00DA	MIR = AMPCR.	0009 0640 0080 0100
F0DB	00DB	MIR = A1	0009 A000 0080 0000
F0DC	00DC	MIR = A1 + B.	0009 AC40 0080 0100
F0DD	00DD	MIR = A2.	0009 C000 0080 0000
F0DE	00DE	MIR = A3.	0009 E000 0080 0100
F0DF	00DF	MIR = A3 OR B.	0009 EC5C 0080 0100
F0E0	00E0	MIR = A3 OR LIT.	0009 E15C 0080 0000
F0E1	00E1	MIR = B.	0009 0C40 0080 0100
F0E2	00E2	MIR = B C.	0009 0C41 8080 0100
F0E3	00E3	MIR = B + 1.	0009 0C46 0080 0100
F0E4	00E4	MIR = B L.	0009 0C41 0080 0000
F0E5	00E5	MIR = B R.	0009 0C40 8080 0000
F0E6	00E6	MIR = B000.	0009 0000 0080 0000
F0E7	00E7	MIR = B001.	0009 00C0 0080 0000
F0E8	00E8	MIR = B111.	0009 001A 0080 0100
F0E9	00E9	MIR = BMAR.	0009 0F40 0080 0100
F0EA	00EA	MIR = LIT.	0009 2000 0080 0100
F0EB	00EB	MIR = LIT AND B.	0009 2C56 0080 0100
F0EC	00EC	MIR = LIT OR B.	0009 2C5C 0080 0100
F0ED	00ED	MIR = LIT + B.	0009 2C40 0080 0000
F0EE	00EE	MIR = LIT L.	0009 2001 0080 0000
F0EF	00EF	MIR = Z.	0009 8000 0080 0100
F0F0	00F0	MIR = 0 + Z + 1.	0009 0D46 0080 0000
F0F1	00F1	MR1.	0009 0000 0000 0800
F0F2	00F2	MR1 A1 = A1 + 1.	0009 A0C0 4000 0900
F0F3	00F3	MW1.	0009 0000 0000 1900
F0F4	00F4	SAR = B.	0009 0C40 0000 8100
F0F5	00F5	SAVE.	0012 0000 0000 0100
F0F6	00F6	SET GC2.	0089 0000 0000 0000
F0F7	00F7	SKIP.	001B 0000 0000 0100
F0F8	00F8	WHEN IRQ STEP.	B808 0000 0000 0000
F0F9	00F9	WHEN RDC BEX.	AC08 0000 0C00 0000
F0FA	00FA	WHEN RDC BEX, MAR1 = BMAR + 1.	AC08 0F46 0C2C 0000
F0FB	00FB	WHEN RMI MAR1 = BMAR + 1.	9C08 0F46 002C 0000
F0FC	00FC	WHEN SRQ STEP.	E808 0000 0000 0000
F0FD	00FD	WHEN URQ STEP.	F808 0000 0000 0100
F0FE	00FE	Z EQV B.	0009 8C52 0000 0100
F0FF	00FF	0 EQV B.	0009 0C52 0000 0000

TABLE II-1. NANO LIST (Cont.)

MICRO CODE	NANO ADDRS		NANO CODE
		SECOND GROUP OF 256	
F100	0100	AMPCR = A1 + AMPCR.	0009 A640 0040 0100
F101	0101	AMPCR = A3.	0009 E000 0040 0100
F102	0102	AMPCR = A2.	0009 C000 0040 0000
F103	0103	AMPCR = A2 + AMPCR.	0009 C640 0040 0100
F104	0104	AMPCR = A3 + AMPCR.	0009 E640 0040 0000
F105	0105	AMPCR = AMPCR + 1.	0009 0646 0040 0100
F106	0106	AMPCR = LIT + B.	0009 2C40 0040 0000
F107	0107	ASE, BMI, MAR1 = B, JUMP.	002D 0C40 0D20 2400
F108	0108	ASR BEX.	0009 0000 0C00 2000
F109	0109	A1 – LIT – 1.	0009 A158 0000 0100
F10A	010A	A1 = LIT EQV B.	0009 2C52 4000 0000
F10B	010B	A1 = A1 C.	0009 A001 C000 0000
F10C	010C	A1 = A1 NIM B.	0009 AC4E 4000 0000
F10D	010D	A1 = A1 OR B.	0009 AC5C 4000 0000
F10E	010E	A1 = B + 1.	0009 0C46 4000 0100
F10F	010F	A1 = B000.	0009 0000 4000 0000
F110	0110	A1 = B111 R.	0009 001A C000 0000
F111	0111	A1 EQV LIT.	0009 A152 0000 0100
F112	0112	A1 EQV Z.	0009 AD52 0000 0100
F113	0113	A1 = LIT – B.	0009 2C5E 4000 0000
F114	0114	A1 EQV 0, IF COV SKIP.	8819 A012 0000 0000
F115	0115	A1 = Z.	0009 8000 4000 0100
F116	0116	A2 EQV B.	0009 CC52 0000 0000
F117	0117	A2 EQV LIT.	0009 C152 0000 0100
F118	0118	A2 EQV 0, IF LC2 SET LC2 SKIP.	3A59 C012 0000 0100
F119	0119	A2 – B.	0009 CC5E 0000 0000
F11A	011A	A2 – B – 1.	0009 CC58 0000 0000
F11B	011B	A2 – LIT.	0009 C15E 0000 0100
F11C	011C	A2 RIM LIT.	0009 C15A 0000 0000
F11D	011D	A2 = A2 + AMPCR.	0009 C640 2000 0100
F11E	011E	A2 = A2 – B.	0009 CC5E 2000 0100
F11F	011F	A2 = A2 – LIT.	0009 C15E 2000 0000
F120	0120	A2 = A2 XOR LIT.	0009 C14C 2000 0000
F121	0121	A2 = A2 XOR B111.	0009 C012 2000 0000
F122	0122	A2 = A3 L.	0009 E001 2000 0000
F123	0123	A2 = B C.	0009 0C41 A000 0100
F124	0124	A2 EQV Z.	0009 CD52 0000 0100
F125	0125	A2 = LIT C.	0009 2001 A000 0100
F126	0126	A2 = LIT AND B.	0009 2C56 2000 0100
F127	0127	A2 = LIT – B.	0009 2C5E 2000 0000
F128	0128	A2 = Z.	0009 8000 2000 0100
F129	0129	A2 BC4 = A3 + B.	0009 EC40 2100 0100
F12A	012A	A3 NAN B.	0009 EC48 0000 0000
F12B	012B	A3 – B – 1.	0009 EC58 0000 0100
F12C	012C	A3 – LIT.	0009 E15E 0000 0000
F12D	012D	A3 = A2 C.	0009 C001 9000 0000
F12E	012E	A3 = A2 L.	0009 C001 1000 0100
F12F	012F	A3 = A2 R.	0009 C000 9000 0100
F130	0130	A3 = A3 AND B.	0009 EC56 1000 0100
F131	0131	A3 = A3 – B.	0009 EC5E 1000 0000
F132	0132	A3 = A3 NIM B.	0009 EC4E 1000 0100

MICRO CODE	NANO ADDRS		NANO CODE
F133	0133	A3 = A3 AND B110.	0009 E0CE 1000 0000
F134	0134	A3 = A3 – B C, IF NOT AOV SET LC1, JUMP ELSE JUMP.	73ED EC5F 9000 0100
F135	0135	A3 = A3 – B C, IF AOV SET LC1, JUMP ELSE JUMP.	7BED EC5F 9000 0000
F136	0136	A3 = A3 XOR LIT.	0009 E14C 1000 0100
F137	0137	A3 = B110.	0009 00DA 1000 0100
F138	0138	A3 = B111 L.	0009 001B 1000 0000
F139	0139	A3 = B111 R.	0009 001A 9000 0000
F13A	013A	A3 = B + 1.	0009 0C46 1000 0100
F13B	013B	A3 = LIT NRI B.	0009 2C44 1000 0100
F13C	013C	A3 = LIT + B.	0009 2C40 1000 0000
F13D	013D	A3 = 0 + LIT.	0009 0140 1000 0000
F13E	013E	A3 SSC = A2 L.	0009 C001 1000 0200
F13F	013F	A3 SSC = A2 R.	0009 C000 9000 0200
F140	0140	A3 SSC = LIT + B.	0009 2C40 1000 0300
F141	0141	B = A1 AND B.	0009 AC56 0B00 0000
F142	0142	B = A1 XOR B.	0009 AC4C 0B00 0100
F143	0143	B = A2 AND B.	0009 CC56 0B00 0000
F144	0144	B = A2 NIM B.	0009 CC4E 0B00 0000
F145	0145	B = A2 XOR B.	0009 CC4C 0B00 0100
F146	0146	B = A2 – LIT.	0009 C15E 0B00 0000
F147	0147	B = A2 C.	0009 C001 8B00 0000
F148	0148	B = A3 – 1.	0009 E0DE 0B00 0100
F149	0149	B = A3 + 1.	0009 E0C0 0B00 0100
F14A	014A	B = A3 AND B.	0009 EC56 0B00 0100
F14B	014B	B = A3 C.	0009 E001 8B00 0100
F14C	014C	B = A3 NRI B.	0009 EC44 0B00 0100
F14D	014D	B - A3 NRI B C.	0009 EC45 8B00 0100
F14E	014E	B = A3 XOR B R.	0009 EC4C 8B00 0100
F14F	014F	B = A3 R, IF ABT SKIP.	6819 E000 8B00 0000
F150	0150	B = B000 JUMP.	002D 0000 0B00 0000
F151	0151	B = B001 JUMP.	002D 00C0 0B00 0000
F152	0152	B = B100 JUMP.	002D 1800 0B00 0000
F153	0153	B = B101.	0009 18C0 0B00 0000
F154	0154	B = B111 R.	0009 001A 8B00 0000
F155	0155	B = B01T.	0009 189A 0B00 0000
F156	0156	B = BT00.	0009 0800 0B00 0100
F157	0157	B = BT0T.	0009 0840 0B00 0000
F158	0158	B = BTT1.	0009 0CC0 0B00 0000
F159	0159	B = BFTF.	0009 1480 0B00 0100
F15A	015A	B = CTR.	0009 6000 0B00 0000
F15B	015B	B = LIT – 1.	0009 20DE 0B00 0100
F15C	015C	B = LIT NRI B C.	0009 2C45 8B00 0100
F15D	015D	B = LIT XOR B L.	0009 2C4D 0B00 0100
F15E	015E	B = Z AAD B.	0009 8C54 0B00 0000
F15F	015F	B = Z IMP B.	0009 8C50 0B00 0100
F160	0160	B = Z NAN B.	0009 8C48 0B00 0100
F161	0161	B = Z NOR B.	0009 8C42 0B00 0100
F162	0162	B = 0 – B.	0009 0C5E 0B00 0100
F163	0163	B = 0 – B – 1.	0009 0C58 0B00 0100
F164	0164	B = 0 – LIT.	0009 015E 0B00 0000
F165	0165	B = 0 – LIT – 1.	0009 0158 0B00 0000

TABLE II-1 NANO LIST (Cont.)

MICRO CODE	NANO ADDRS		NANO CODE
F166	0166	BAD A3 = A3 + B.	0009 EC40 1800 0100
F167	0167	BBI = B R.	0009 0C40 8F00 0100
F168	0168	BC4.	0009 0000 0100 0000
F169	0169	BC4 CSAR, A3 = A3 – B C.	0009 EC5F 8100 4000
F16A	016A	BC4 SAR = B001.	0009 00C0 0100 8100
F16B	016B	BC8.	0009 0000 0900 0100
F16C	016C	BR1 = LIT L.	0009 2001 0020 0000
F16D	016D	BR2 = A3 L.	0009 E001 0010 0000
F16E	016E	BR2 = B.	0009 0C40 0010 0100
F16F	016F	BR2 = B1TT.	0009 1C40 0010 0000
F170	0170	CSAR A1 = A3 L.	0009 E001 4000 4100
F171	0171	CSAR A3 = B R.	0009 0C40 9000 4100
F172	0172	CSAR, B = B C.	0009 0C41 8B00 4000
F173	0173	CSAR BC4 A3 = A3 + B C.	0009 EC41 9100 4000
F174	0174	CSAR MIR = B000.	0009 0000 0080 4100
F175	0175	CTR = A1.	0009 A000 0005 0100
F176	0176	CTR = A1, IF ABT SKIP.	6819 A000 0005 0100
F177	0177	CTR = A2 R.	0009 C000 8005 0000
F178	0178	CTR = A3.	0009 E000 0005 0000
F179	0179	CTR = BMAR + 1.	0009 0F46 0005 0000
F17A	017A	CTR = CTR + LIT + 1.	0009 6146 0005 0100
F17B	017B	CTR EQV 0.	0009 6012 0000 0100
F17C	017C	CTR = LIT + CTR	0009 2B00 0005 0100
F17D	017D	DLD.	0009 0000 0000 1000
F17E	017E	DR2 BEX, BR2 = B.	0009 0C40 0C10 2C00
F17F	017F	IF ABT LUOP ELSE SKIP.	6C0B 0000 0000 0000
F180	0180	IF ABT RESET GC1.	6B49 0000 0000 0100
F181	0181	IF ABT RESET GC2.	6AC9 0000 0000 0100
F182	0182	IF ABT SET LC2.	6A49 0000 0000 0000
F183	0183	IF AOV JUMP.	7829 0000 0000 0000
F184	0184	IF GC1 SET LC1 SKIP.	0BD9 0000 0000 0100
F185	0185	IF GC1 SET LC2 SKIP.	0A59 0000 0000 0100
F186	0186	IF GC1 STEP ELSE EXEC.	080C 0000 0000 0000
F187	0187	IF GC1 STEP ELSE JUMP.	080D 0000 0000 0100
F188	0188	IF GC1 STEP ELSE SAVE.	080A 0000 0000 0000
F189	0189	IF GC1 STEP ELSE WAIT.	0808 0000 0000 0100
F18A	018A	IF GC1 RETN.	0839 0000 0000 0000
F18B	018B	IF GC1 SAVE.	0811 0000 0000 0000
F18C	018C	IF GC1 WAIT.	0801 0000 0000 0100
F18D	018D	IF IRQ, ASR, BEX, JUMP.	BE29 0000 0C00 2100
F18E	018E	IF IRQ SKIP.	B819 0000 0000 0000
F18F	018F	IF LC1 A2 = A2 + 1.	2C09 C0C0 2000 0100
F190	0190	IF LC1 INC.	2C09 0000 0002 0100
F191	0191	IF LC1 SET LC1 ELSE JUMP.	2BCD 0000 0000 0000
F192	0192	IF LC2 SET LC2 ELSE JUMP.	3A4D 0000 0000 0100
F193	0193	IF LC2 SET LC2 JUMP.	3A69 0000 0000 0100
F194	0194	IF LC3 SET LC3 ELSE JUMP.	DB0D 0000 0000 0000
F195	0195	IF LST CALL.	5831 0000 0000 0100
F196	0196	IF LST SET GC1.	5B89 0000 0000 0100
F197	0197	IF LST SET LC2.	5A49 0000 0000 0000
F198	0198	IF LST SET LC2 SKIP.	5A59 0000 0000 0100
F199	0199	IF NOT ABT RESET GC1.	6349 0000 0000 0000

TABLE II-1. NANO LIST (Cont.)

MICRO CODE	NANO ADDRS		NANO CODE
F19A	019A	IF NOT ABT SET GC1.	6389 0000 0000 0000
F19B	019B	IF NOT ABT SET LC1.	63C9 0000 0000 0100
F19C	019C	IF NOT GC1 JUMP.	0029 0000 0000 0000
F19D	019D	IF NOT GC1 SET GC1 JUMP.	03A9 0000 0000 0100
F19E	019E	IF NOT GC1 STEP ELSE SKIP.	000B 0000 0000 0000
F19F	019F	IF NOT GC2 JUMP.	1029 0000 0000 0100
F1A0	01A0	IF NOT GC2 LUOP SKIP.	1419 0000 0000 0000
F1A1	01A1	IF NOT GC2 SET GC2 JUMP.	12A9 0000 0000 0100
F1A2	01A2	IF NOT LC1 SET LC1.	23C9 0000 0000 0000
F1A3	01A3	IF NOT LC1 SET LC1 ELSE SKIP.	23CB 0000 0000 0100
F1A4	01A4	IF NOT LC1 SET LC1 SKIP.	23D9 0000 0000 0100
F1A5	01A5	IF NOT LC1 SET LC1 JUMP.	23E9 0000 0000 0100
F1A6	01A6	IF NOT LC2 SET LC2 JUMP.	3269 0000 0000 0000
F1A7	01A7	IF NOT LC3 JUMP.	D029 0000 0000 0100
F1A8	01A8	IF NOT LC3 SET LC3 JUMP.	D329 0000 0000 0100
F1A9	01A9	IF NOT LC3 SET LC3 SKIP.	D319 0000 0000 0100
F1AA	01AA	IF NOT LST SET GC2 JUMP.	52A9 0000 0000 0000
F1AB	01AB	IF NOT MST JUMP.	4029 0000 0000 0100
F1AC	01AC	IF NOT IRQ JUMP.	B029 0000 0000 0100
F1AD	01AD	IF NOT SRQ SKIP.	E019 0000 0000 0100
F1AE	01AE	IF NOT URQ SKIP.	F019 0000 0000 0000
F1AF	01AF	IF MST JUMP.	4829 0000 0000 0000
F1B0	01B0	IF MST SET LC1.	4BC9 0000 0000 0100
F1B1	01B1	IF SRQ THEN DW2 JUMP.	EA29 0000 0000 3D00
F1B2	01B2	IF URQ SET GC1 ELSE JUMP.	FB8D 0000 0000 0000
F1B3	01B3	IF URQ SET GC2 ELSE JUMP.	FA8D 0000 0000 0100
F1B4	01B4	IF URQ SET LC2.	FA49 0000 0000 0000
F1B5	01B5	IF URQ SKIP.	F819 0000 0000 0100
F1B6	01B6	INC, B= BMAR.	0009 0F40 0B02 0000
F1B7	01B7	INC IF COV JUMP.	8829 0000 0002 0100
F1B8	01B8	INC, IF NOT COV SKIP ELSE STEP.	8019 0000 0002 0000
F1B9	01B9	INC, SAVE.	0012 0000 0002 0000
F1BA	01BA	LIT EQV 0.	0009 2012 0000 0000
F1BB	01BB	LIT IMP B.	0009 2C50 0000 0000
F1BC	01BC	MAR1 = A1 + LIT.	0009 A140 002C 0000
F1BD	01BD	MAR1 = A2 + 1.	0009 C0C0 002C 0000
F1BE	01BE	MAR1 = A2 + B.	0009 CC40 002C 0100
F1BF	01BF	MAR1 = A2 + LIT.	0009 C140 002C 0000
F1C0	01C0	MAR1 = A3 + AMPCR.	0009 E640 002C 0000
F1C1	01C1	MAR1 = A3 – B.	0009 EC5E 002C 0000
F1C2	01C2	MAR1 = B R. (B711)	0009 0C40 802C 0000
F1C2	01C2	MAR1 = B R, SKIP. (B705)	001B 0C40 802C 0000
F1C3	01C3	MAR1 = B001.	0009 00C0 002C 0000
F1C4	01C4	MAR1 = B001 + 1.	0009 00C6 002C 0000
F1C5	01C5	MAR1 = BMAR + 1 IF NOT COV JUMP ELSE SAVE.	802A 0F46 002C 0100
F1C6	01C6	MAR1 = LIT XOR BMAR.	0009 2F4C 002C 0000
F1C7	01C7	MIR = AMPCR + 1.	0009 0646 0080 0100
F1C8	01C8	MIR = A1 L.	0009 A001 0080 0100
F1C9	01C9	MIR = A2 L.	0009 C001 0080 0100

TABLE II-1. NANO LIST (Cont.)

MICRO CODE	NANO ADDRS		NANO CODE
F1CA	01CA	MIR = A2 + B.	0009 CC40 0080 0100
F1CB	01CB	MIR = A2 OR B.	0009 CC5C 0080 0000
F1CC	01CC	MIR = A2 OR LIT.	0009 C15C 0080 0100
F1CD	01CD	MIR = A3 L	0009 E001 0080 0000
F1CE	01CE	MIR = A3 R.	0009 E000 8080 0000
F1CF	01CF	MIR = A3 AND B.	0009 EC56 0080 0100
F1D0	01D0	MIR = A3 AND LIT.	0009 E156 0080 0000
F1D1	01D1	MIR = A3 AND LIT L.	0009 E157 0080 0100
F1D2	01D2	MIR = A3 +B.	0009 EC40 0080 0000
F1D3	01D3	MIR = A3 – B.	0009 EC5E 0080 0000
F1D4	01D4	MIR = B, SAVE.	0009 0C40 0080 0100
F1D5	01D5	MIR = B001 + 1.	0009 00C6 0080 0000
F1D6	01D6	MIR = NOT CTR R.	0009 6010 8080 0000
F1D7	01D7	MIR = Z + B001.	0009 80C0 0080 0100
F1D8	01D8	MIR = 0 AAD Z.	0009 0D54 0080 0000
F1D9	01D9	MIR = 0 OAD B000.	0009 000A 0080 0000
F1DA	01DA	MIR = 0 OAD Z.	0009 0D4A 0080 0000
F1DB	01DB	MIR = 0 + Z.	0009 0D40 0080 0000
F1DC	01DC	MIR = 0 – Z.	0009 0D5E 0080 0000
F1DD	01DD	MIR = 0 – Z – 1.	0009 0D58 0080 0000
F1DE	01DE	MIR BC8 = A1 + B IC.	0009 AC60 0980 0000
F1DF	01DF	MR1, A3 = B R.	0009 0C40 9000 0900
F1E0	01E0	MR1, CTR = B.	0009 0C40 0005 0900
F1E1	01E1	MR1, IF LC1 STEP.	2819 0000 0000 0800
F1E2	01E2	MR1, INC.	0009 0000 0002 0900
F1E3	01E3	MR1, MIR = B.	0009 0C40 0080 0800
F1E4	01E4	MW1, A2 = A2 + 1.	0009 C0C0 2000 1800
F1E5	01E5	MW1, INC, A3 = A3 XOR B.	0009 EC4C 1002 1900
F1E6	01E6	RESET GC1, SAVE	0152 0000 0000 0100
F1E7	01E7	RETN.	003F 0000 0000 0100
F1E8	01E8	SAR = A2 R.	0009 C000 8000 8100
F1E9	01E9	SAR = A3 R.	0009 E000 8000 8000
F1EA	01EA	SAR = B L.	0009 0C41 0000 8000
F1EB	01EB	SET GC1, JUMP.	01AD 0000 0000 0100
F1EC	01EC	WHEN RDC BEX JUMP.	AC28 0000 0C00 0100
F1ED	01ED	WHEN RDC THEN BEX, A1 = A1 + 1, JUMP	AC28 A0C0 4C00 0000
F1EE	01EE	WHEN RDC THEN BEX, A2 = A2 + 1, JUMP.	AC28 C0C0 2C00 0000
F1EF	01EF	WHEN RDC BEX A3 = B.	AC08 0C40 1C00 0000
F1F0	01F0	WHEN RDC BEX CTR = B.	AC08 0C40 0C05 0100
F1F1	01F1	WHEN RDC BEX, MAR1 = A1.	AC08 A000 0C2C 0100
F1F2	01F2	WHEN RDC BEX, MAR1 = B, JUMP.	AC28 0C40 0C2C 0100
F1F3	01F3	WHEN RDC THEN BEX, MAR1 = BMAR +1, JUMP.	AC28 0F46 0C2C 0100
F1F4	01F4	WHEN RMI A1 = A1 + 1, JUMP.	9C28 A0C0 4000 0000
F1F5	01F5	WHEN RMI A2 = A2 + 1, JUMP.	9C28 C0C0 2000 0000
F1F6	01F6	WHEN RMI THEN MAR1 = BMAR + 1, JUMP.	9C28 0F46 002C 0100
F1F7	01F7	WHEN RMI MAR1 = LIT.	9C08 2000 002C 0000

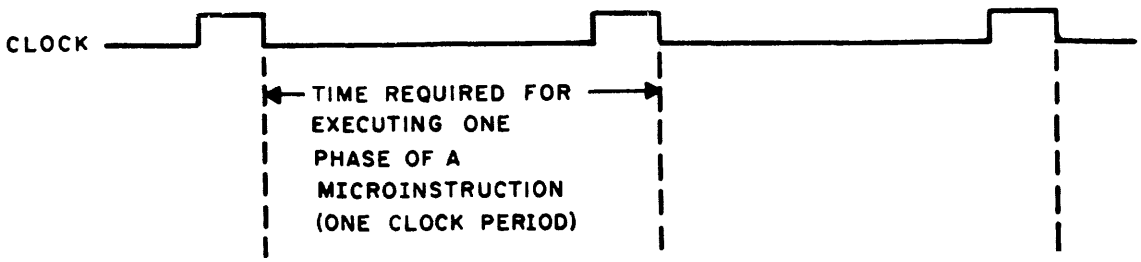


Fig. II-5 CLOCK PULSES AND PHASE RELATIONSHIP

The destination controls are used to perform the following functions:

a. A-register input selection.
b. B-register input selection.
c. Memory Information Register (MIR) input selection
d. Alternate Microprogram Count Register (AMPCR) input selection.
e. Base Register (BR1) input selection.
f. Base Register (BR2) input selection.
g. Memory Address Register (MAR) input selection.
h. Counter (CTR) input selection.
i. Shift Amount Register (SAR) input selection.
j. Shift Store Control (SCC).

See Figure II-4 for nanocode information pertinent to Logic Unit operations and destination controls.

The Logic Unit operation controls (phase 3 controls from decoded nanobits 17 through 33) are provided by the Control Register and are present from the beginning to the end of phase 3. The destination controls, which are executed at the end of phase 3, cause the contents of the destination registers to be changed. However, the destination registers are not changed until the occurrence of the clock pulse which signals the end of phase 3 and simultaneously reloads the Control Register from the execution of a new Logic Unit operation. The destination controls are decoded nanobits 34 through 50 and 55 from the Control Register on the MU and CU cards.

An extended phase 3 interval is the designation given to a phase 3 clock time whose Logic Unit operation is in the process of being performed, but whose destination register change is postponed for one or more clock periods. An explanation of the need for an extended phase 3 interval is presented in the following paragraphs and in Figure II-6.

MICROINSTRUCTION	A	B	C	D	E	F	
TYPE	I	I	I	I	II	I	
LOGIC UNIT OPERATION	P	P	M	P		P	
PHASE 1	A_1	B_1	C_1	D_1	E_1	F_1	
PHASE 3 OR EXTENDED –PHASE 3 (EX-3)	Z_3	A_3	$B_{Ex\text{-}3}$ → B_3	$D_{Ex\text{-}3}$ → D_3		F_3	
CLOCK	0	1	2	3	4	5	6 · · ·

NOTE:
FOR A LOGIC UNIT OPERATION , A "P" INDICATES PRESENT AND AN "M" INDICATES MISSING.

Fig. II-6 TYPICAL EXAMPLE OF PHASED EXECUTION OF MICROINSTRUCTION

A B 700 Processing system processes microinstructions in parallel; therefore, the processor always expects to process phase 1 of one microinstruction and phase 3 of another microinstruction, concurrently. To ensure the existence of a phase 3 interval which occurs concurrently with a phase 1 interval, the phase 3 interval in process is delayed until the occurrence of another phase 3 interval for a subsequent microinstruction.

The external operations (Memory/Device operations, nanobits 51 through 54) occur following phase 1 and are independent of any phase 3 instruction.

The phased execution of a sequence of microinstructions is presented in figure II-6, in which each microinstruction is symbolically represented by letters A through F. The type of operation (Type I or II), the presence or absence of a Logic Unit operation, phase information, and clock intervals are also shown. A subscript indicates the particular phase; an ex-3 subscript indicates an extended phase 3 that was originally a phase 3. A "present" designation opposite Logic Unit operation indicates that the Type I instruction requires the execution of a phase 3 Logic Unit operation. A *missing* designation indicates that the Logic Unit operation was conditional and that the condition required for execution was not met.

The phase 3 or extended phase 3 (EX-3) in process is determined by the presence of a Logic Unit operation. As indicated in figure II-6, the previous phase 3 (Z^3) is completed, and then the next clock initiates the new phase 3 (microinstructions A, B, D, and F) of a Logic Unit operation required for Type I microinstructions. A previous phase 3 is extended if a Logic Unit operation is not required, as indicated for microinstruction C. At this point, microinstruction C creates an extended phase 3 (BEX-3) for the microinstruction at B. Microinstruction D, which is a Type I microinstruction with a Logic Unit operation present, completes the phase 3 for microinstruction B. Microinstruction E which is Type II creates an extended phase 3 (DEX-3) for microinstruction D. The presence of Type II microinstruction E delays the D microinstruction input to the destination registers until the end of phase 1 at microinstruction F time (clock 6); thereby, forcing microinstruction D to use destination register values that occur after clock time 5. Microinstruction E indicates how a Type II microinstruction can affect the results of a previous Type I microinstruction.

A device control having direct memory access (such as the Universal Cartridge Disk Drive Control and the Data Communications Processor) will, asynchronously with microinstruction execution, request and be granted memory access. This is called a *cycle steal* . The cycle steal will cause a memory cycle to occur, thereby forcing an extended phase-3 condition and suspending the instruction fetch and execution processes. (See example E in figure II-7.) Cycle steal requests are given a higher priority than instruction fetching and executing, thus all phase 1 and phase 3 operations are suspended for one clock period when a cycle steal occurs.

Microprogram timing is a critical consideration in the execution of microprogramming in a processing system. The following timing definitions apply to phase information:

a. Phase 1 of a microinstruction is always executed in parallel with phase 3 or extended phase 3 of another microinstruction. (See examples A and B in figure II-7.)

b. An extended phase 3, which is otherwise known as a phase 3 inhibit or delayed phase 3, occurs due to succession by either a Type II or a Type I microinstruction that contains a conditional Logic Unit operation that has not been satisfied. (See example C in figure II-7.)

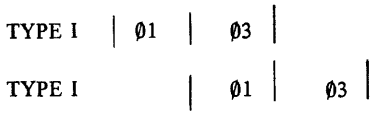
d. Any microinstruction which causes either an extended phase 3 to be initiated or prolongs an existing extended phase 3 is executed between phase 1 and phase 3 of the affected Type I microinstruction. (See example C of figure II-7.)

MICROPROGRAM INSTRUCTION SEQUENCING

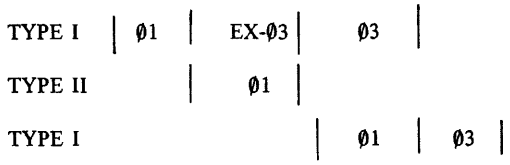
The sequence of logical events that occur in the execution of a microinstruction is presented in figure II-8. The sequence of events (steps 1 through 11) are used in the following text to describe the logical events which occur in the execution of a microinstruction. The first group of steps (sequence of events) that occur are common to both Type I and Type II microinstructions as follows:

1. Development of the Microprogram Address (MPAD). The MPAD Control (MPAD CNTL) is set by the previous microinstruction

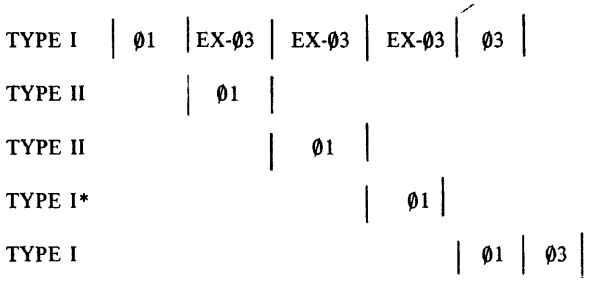
A. Type I Microinstruction followed by a Type I microinstruction for which a Logic Unit operation is required:



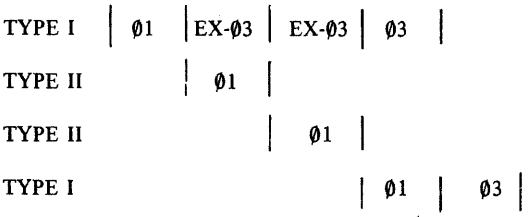
B. Type I microinstruction followed by a Type II microinstruction and a Type I microinstruction for which a Logic Unit operation is required:



C. Type I microinstruction followed by two Type II microinstructions and two Type I microinsructions:



D. Type I microinstruction followed by two Type II microinstruction and a Type I microinstruction for which a Logic Unit operation is required:



NOTE:
The asterisk (*) indicates that this Type I microinstruction consists of only phase 1.

Fig. II-7 PHASING OF TYPE I AND TYPE II MICROINSTRUCTIONS

and, at this time, the MPAD CNTL determines the selection of the Microprogram Count Register (MPCR) or the Alternate Microprogram Count Register (AMPCR).

2. Selection of the correct increment amount (+1 or +2).

3. The microinstruction is read from the Microprogram Memory (MPM).

4. A decode of the MPM word is performed to determine if it is a Type I or II microinstruction.

If the MPM word is a Type II microinstruction, the following sequence of events occur. At the same time that the Type II microinstruction occurs, the phase 3 portion of a previous Type I microinstruction is executed; however, this parallel phase 3 interval is not completed at the end of the clock period (one clock time). Therefore, the destination registers remain unchanged and the phase 3 interval is extended (EX-3) for an additional clock period for the previous Type I microinstruction. The following steps (sequence of events) apply for the Type II microinstruction as shown on figure II-8:

5(A). The low order bits (LSB's) of the microinstruction are used as literal values.

11(a). A STEP successor is sent to the MPAD CNTL.

11(b). The literals are clocked to the specified registers (SAR, LIT, SAR and LIT, or AMPCR).

If the Microprogram word is a Type I microinstruction, the following sequence of events occur. (See figure II-8 and refer to Table II-2.) At the same time that the Type I microinstruction occurs, the phase 3 of a previous Type I microinstruction is executed. However, the completion of this phase 3 portion of the previous Type I microinstruction is dependent upon the requirement of a new phase 3 during the execution of the phase 1 portion of the current Type I microinstruction. The following steps (sequence of events) apply for the Type I microinstruction as shown on figure II-8.

5(b). The low order bits (LSB's) of the microinstruction are used to address the Nanomemory (NM).

6. The nanocode is read from the Nanomemory.

7. The resultant nanocode is decoded.

8. Selection of the condition to be tested (nanobits N1 through N4).

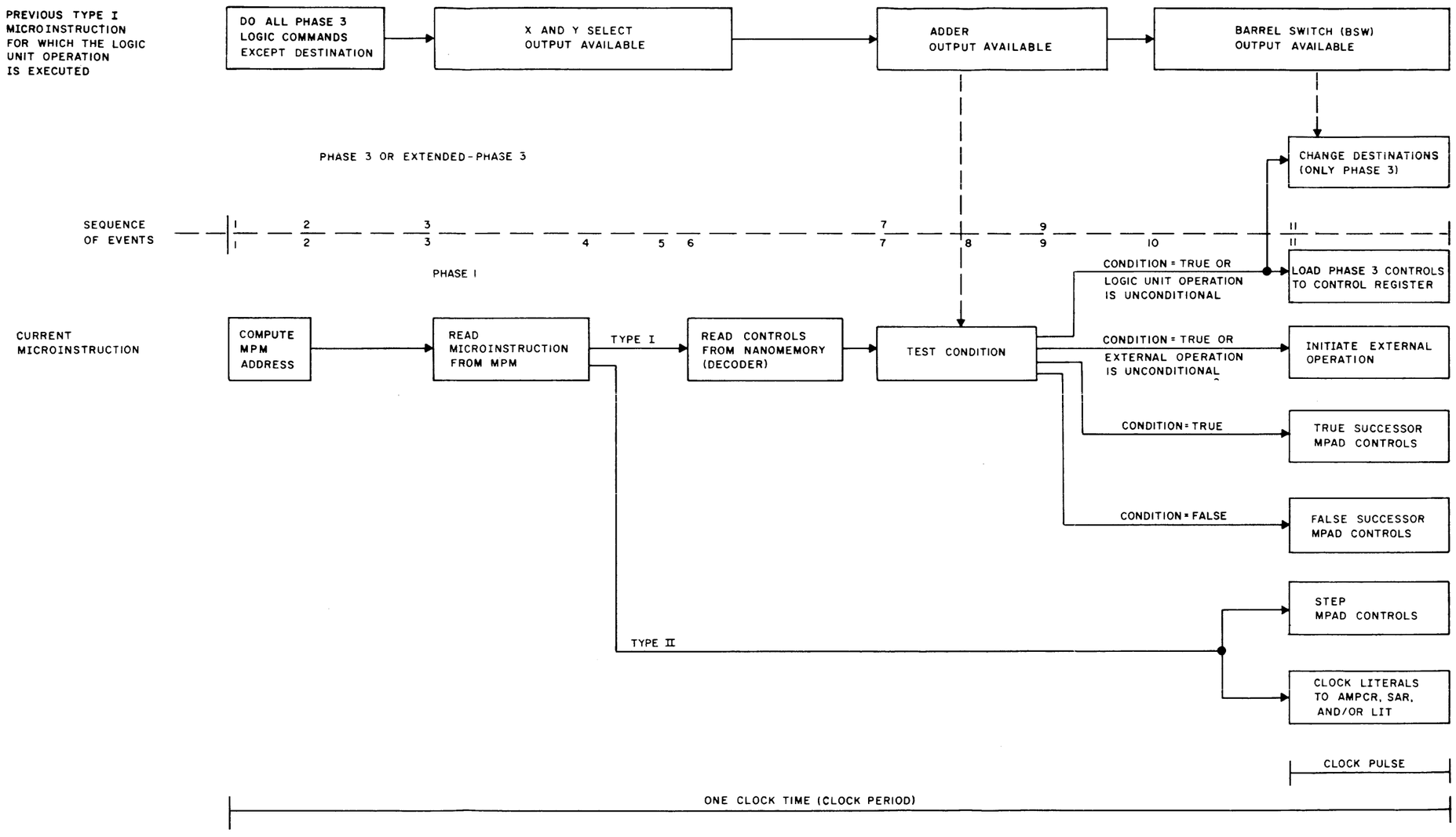


Fig. II-8 MICROINSTRUCTION SEQUENCING DIAGRAM

9. Determination of whether the selected true or complement (false) condition has been met (nanobit N5).

10(a). Determination of whether a new Logic Unit operation (LUOP) is required by this microinstruction (nanobit N6). If the Logic Unit operation (LUOP) to be performed is unconditional or if the test condition was met (condition=true), a new Logic Unit operation (LUOP) is required to complete the phase 3 of the previous microinstruction.

10(b). Determination of whether an external operation (EXTOP) is required by this microinstruction (refer to nanobit N7). An external operation (EXTOP) is required if it is to be performed unconditionally or if the test condition is met (condition=true).

11(c). If an external operation (EXTOP) is to be performed, enable the condition adjust (CNDADJ, nanobits N8 through N10) and memory/device operations (MDOP, nanobits N51 through N54) unless their selection values are ZERO.

11(d). If a new Logic Unit operation (LUOP) is required, the destination portion of the current phase 3 is completed and the new phase 33 controls are decoded and loaded into the Control Register. (Refer to nanobits N17 through N50 and N55.)

11(e). The MPAD CNTL's are loaded with either a true or false successor depending upon which test condition was met; successor conditions=1 (true) or 0 (false). (Refer to nanobits N11 through N16.)

11(f). If applicable, the tested condition is reset.

The following information is applicable for a Type I microinstruction during a phase 3 or extended phase 3 interval. (See figure II-8 and refer to table II-2.)

While phase 1 of the current microinstruction is in process, the phase 3 of a previous microinstruction is executed. If a new phase 3 interval is not required at the end of the current phase 1, the phase 3 is extended (EX-3) during the next phase 1 interval until a new phase 3 is required. The sequence of events (steps) for a Type I microinstruction during a phase 3 or extended phase 3 interval are as follows:

1(a). Selection of X input to Adder. (Refer to nanobits N17 through N19.)

1(b). Selection of Y input to Adder. (Refer to nanobits N20 through N26.)

3(b). Inhibit carries between 8-bit bytes, if required. (Refer to nanobits N28 through N31.)

7. At this time, the dynamic conditions from the Adder are available for testing. (Refer to step 8 of the previous Type I, phase 1 microinstruction description.)

9. Selection of shift direction and performance of shift functions. (Refer to nanobits N32 and N33.)

11(g). The destination registers are changed at this time if the current phase 1 requires a new phase 3; otherwise, the phase 3 interval is extended (EX-3) and re-executed during the next phase 1 interval of a microinstruction. During the sequence of events, any or all of the following registers may be changed (refer to nanobits N34 through N50 and N55):

a. A Registers.
b. B Register.
c. Memory Information Register (MIR).
d. Alternate Microprogram Count Register (AMPCR).
e. Base Register 1 (BR1).
f. Base Register 2 (BR2).
g. Memory Address Register (MAR).
h. Counter (CTR).
i. Shift Amount Register (SAR).
j. Sign Save Control (SSC).

Table II-2 indicates the Nanomemory decoding functions that occur during phase 1, at the end of phase 1, during phase 3, and at the end of phase 3.

MICROPROGRAMMING

Microprogramming is a means of controlling the data and data flow in a sequential manner to perform a given function. Microprogramming is the format that the designer uses to specify the action, function, and the state of each of the B 700 system logic elements at every clock cycle time. In this manner, microprogramming replaces the function of hardware sequential logic that is used in a system configuration to execute an instruction which requires more than one clock time. Therefore, it can be stated that microprogramming is essentially similar to sequential logic design.

The microprogram is written in the B 700 Microprogram Language (B7MPL), which is the source language for a computer program designed to convert English language statements defining the action of the Processor for each clock cycle into binary patterns for the Microprogram Memory (MPM) and Nanomemory (NM). The B7MPL structure consists of a number of reserved words and operators that are used to specify the action of the Proc-

essor for each clock cycle. Refer to table II-3 for the B7MPL reserved words and terminal characters used in the program listings.

A microinstruction or nanoinstruction is a statement that specifies the action of the processor for one clock cycle and requires one word in the Nanomemory (NM) as indicated by the following example:

CSAR, LMAR, A1=A1+B

The value of the Shift Amount Register (SAR) is complemented, the contents of the Literal Register (LIT) are placed in the Memory Address Register (MAR), and the contents of the A1 Register are added to the contents of the B Register and the result is stored in the A1 Register.

B7MPL provides the capability for conditional branching to a subroutine or other instruction strings with reserved words (Successor words) such as STEP, SKIP, CALL, EXEC, WAIT, SAVE, JUMP, and RETN. In the example previously described, there is no Successor word; thus the STEP instruction is implied. This implied STEP instruction causes the Processor to step to the next microinstruction in sequence in the Microprogram. The following is an example of a conditional instruction:

IF LC1 THEN A1=A1+B JUMP ELSE STEP.

If the Local Condition Bit 1 is true, the indicated operations that follow the THEN statement are performed and a JUMP is made to the instruction address that is currently in the Alternate Microprogram Count Register plus 1 (AMPCR+1). If the Local Condition Bit 1 is false, then the action indicated after the ELSE statement occurs; a STEP is made to the next instruction in sequence.

TABLE II-2 NANOMEMORY DECODING

Phase and Control Information	Nanomemory Bit No.	Function
During Phase 1:		
Conditional Controls	1-4	Condition Selection) (CONDSEL)
	5	Successor Condition (True/False)
	6	Logic Unit Condition (Phase 3 Modifier)
	7	Memory/Device/Condition Adjust Modifier (MDOP and CNDADJ Modifier)
At End of Phase 1:		
Successor Determination	11-16	Microprogram Address Controls (MPAD CNTL)
External Operation	8-10	Condition Adjust (CNDADJ)
	51-54	Memory/Device Operation (DOP). Request signals for memory or peripheral device operations.
Phase 3:		
Adder Operation	17-19	X Selection to Adder
Commands	20-26	Y Selection to Adder
	27	Inhibit Carry Between 8-Bit Bytes
	28-31	Adder or Logic Operation
	32 and 33	Shift Selection (right, left, circular determined by amount in SAR)
At End of Phase 3:		
Destination Specifiers	34-36	A Registers (A1, A2, and A3) Input Selection from Barrel Switch (BSW)
	37-40	B Register Input Selection
	41	MIR Input Selection from BSW
	42	AMPCR Input Selection from BSW
	43	Base Register 1 (BR1) Input Selection from BSW
	44	Base Register 2 (BR2) Input Selection from BSW
	45 and 46	MAR Input Selection from BSW or LIT
	46-48	Counter (CTR) Input Selection from LIT, BSW, or Increment CTR
	49-50	SAR Input Selection from Complement SAR (CSAR) or BSW
	55	Sign Save Control (SSC)

TABLE II-3 B7MPL RESERVED WORDS AND OPERATORS

Reserved Word	Function
A1	A1 Register X Select to Adder or destination operator.
A2	A2 Register X Select to Adder or destination operator.
A3	A3 Register X Select to Adder or destination operator.
AAD	ANDed add logic operator: X input plus (X input ANDed with Y input)=X+(XY)
ABT	Adder Bit true Adder output all ONES.
AMPCR	Alternate Microprogram Count Register Y-Select to Adder or destination operator from Barrel Switch.
AND	Logical operator AND: X AND Y = XY.
AOV	Adder overflow, dynamic condition of previous microinstruction which uses the Adder.
ASCII	Identifies a "ASCII" character string.
ASE	Forces concatenation of the BR2 register with MAR for subsequent BMAR.
ASR	Status request for highest priority unselected device; also forces concatenation of the BR1 register with MAR for subsequent BMAR.
B	B Register Y-Select to Adder or destination operator (to B Register from Barrel Switch).
BAD	Destination operator: To B register Input Selection from Adder.
BBA	Destination operator: To B Register Input Selection from Adder ORed with Barrel Switch output.
BBE	Destination operator: To B Register Input Selection from External data bus ORed with Barrel Switch output.
BB1	Destination operator: To B Register Input Selection from prior Memory Information Register (MIR) contents ORed with Barrel Switch output.
BC4	Destination operator: To B Register Input Selection from Adder "not 4-bit carry" (complement of 4-bit carry).
BC8	Destination operator: To B Register Input Selection from Adder "not 8-bit carry" (complement of 8-bit carry).
BEX	Destination operator: To B Register Input Selection from External data bus.
BMAR	Y Selection to Adder of the Base Register 1 (BR1) or Base Register 2 (BR2) concatenated with the Memory Address Register (MAR) output.
BMI	Destination operator: To B Register Input Selection from prior MIR contents.
BOUND	Ensures that a specified area will not cross over a specified SYNC boundary. exam BOUND (nn1) MOD (nn2) nn1 = size in D-words of area placed in bounds. nn2 = SYNC value of area. If condition is not satisfied, the address counter is increased to the next module level.
BR1	Base Register 1
BR2	Base Register 2
C	Circular shift right the entire Adder output; operation takes place in the Barrel Switch.
CALL	Call a procedure: Use AMPCR+1 as address; exchange AMPCR+1 and MPCR.
CAMPCR	Synonomous with CPCR.
COMP	Complement as appropriate for literal destination.
CONT or CONTINUE	Pseudo operator used in the segment statement which causes the next microaddress or nanoaddress assigned to be greater than the greatest address previously used in the program.
COV	Counter overflow condition.
CNST	Pseudo operator that is used to generate a microcode with a specified value.
CPCR	Call Program Counter (TYPE II); MPCR and INCR are replaced by Microcode +1; AMPCR is replaced by prior MPCR.
CSAR	Complement Shift Amount Register (SAR).
CTR	Counter (CTR) Register.
DATE	Constants containing Julian date.
DEFINE	Used for local DEFINE statements.
DEFINITIONS	A reserved label used to identify a group of parameter definitions.
DLD	Used in special system configurations that use an LIC or configuration card, to read system configuration bits into "B" on the next BEX instructions.
DR1	Device Read 1: Read from device specified by Base Register 1 (BR1).
DR2	Device Read 2: Read from device specified by Base Register 2 (BR2).
DW1	Device Write 1: Write to device specified by Base Register 1 (BR1).
DW2	Device Write 2: Write to device specified in Base Register 2 (BR2).

TABLE II-3 B7MPL RESERVED WORDS AND OPERATORS (Cont)

Reserved Word	Function
EBCDIC	Identifies a "EBCDIC" character string.
ELSE	Sequential operator.
EQV	Equivalence logical operator: X EQV Y.
EXEC	Executes out of sequence: Use AMPCR+1 address.
EXT	External condition bit: set externally and reset by testing (real-time clock).
FINISH	Reserved label, which terminates the source input deck.
GC1	Global Condition Bit 1: A set condition causes this bit to be set. Testing this condition bit does not cause resetting of the condition bit.
GC2	Global Condition Bit 2: Refer to explanation for GC1.
IC	Inhibit carries.
IF	Sequential operator which starts the conditional part of an instruction.
IMP	Logical operator Imply: X IMP Y.
INC	Increment Counter; set Counter Overflow (COV) if overflow.
IOL	Initiate memory overlay.
IRQ	Interrupt from unselected device. This interrupt can be either a status or data interrupt.
JUMP	Jump to address in Alternate Microprogram Count Register (AMPCR+1); then, place that address in the Microprogram Count Register (MPCR).
L	Left shift end-off the entire Adder output, right fill with ZEROS. This operation takes place in the Barrel Switch (BSW).
LC1	Local Condition Bit 1: Can be set or tested (reset by test).
LC2	Local Condition Bit 2: Can be set or tested (reset by test).
LC3	Local Condition Bit 3: Can be set or tested (reset by test).
LCTR	One's complement of the Literal Register contents will be placed in the Counter (CTR) and Counter Overflow (COV) will be reset.
LIT	Literal Register: set with a literal and can be an input to the Adder, the Memory Address Register (MAR), or the Counter (CTR).
LMAR	Literal Register contents will be placed in the Memory Address Register (MAR).
LOCN	Pseudo operator used to advance microprogram instruction address to the specified address, or to previously declared label, plus or minus literal offset.
LST	Least significant bit of the Adder output; dynamic condition of the previous microinstruction which uses the Adder.
LUOP	Pseudo operator used to generate a default Logic Unit operation.
MAR	Memory Address Register.
MAR1	Combination parameter specifying Base Register (BR1) and Memory Address Register (MAR).
MAR2	Combination parameter specifying Base Register (BR2) and Memory Address Register (MAR).
MICRO	Pseudo operator that identifies a control parameter that is used to set the microaddress when segmenting.
MIR	Memory Information Register.
MPCR	Mark Program Counter (TYPE II); MPCR and INCR are replaced by Microcode +1; AMPCR remains unchanged.
MR1	Memory Read 1: Read the contents from the memory address specified by Base Register (BR1) and Memory Address Register (MAR).
MR2	Memory Read 2: Read the contents from the memory address specified by Base Register (BR2) and Memory Address Register (MAR).
MST	Most significant bit of the Adder output; dynamic condition from previous microinstruction specifying an Adder operation.
MW1	Memory Write 1: Write the contents of the Memory Information Register (MIR) into the memory address specified by Base Register (BR1) and the Memory Address Register (MAR).
MW2	Memory Write 2: Write the contents of the Memory Information Register (MIR) into the memory address specified by Base Register (BR2) and the Memory Address Register (MAR).
NAN	Logical operator Not AND: X NAN Y.
NIM	Logical operator Not Imply: X NIM Y.
NOR	Logical operator NOR: X NOR Y.
NOT	Complement monadic or condition operator: NOT X
NR1	Logical operator Not Reserve Imply: X NRI Y.
OAD	Logical operator ORed ADD: X input plus (X input ORed with Y input).
OR	Logical operator OR: X OR Y.

TABLE II-3 B7MPL RESERVED WORDS AND OPERATORS (Cont)

Reserved Word	Function
PROGRAM-ID	Bracket word which begins a program.
R	Right shift end-off of the entire Adder output, left fill with ZEROS.
RDC	Read Complete bit: set when external data from External Bus is ready for input to the B Register via the B Input Select gates. This bit is unconditionally ttue.
RELATIVE	Used with SEGMENT.
RENAMES	Pseudo operator used to indicate definition type.
RESET	Reset the specified Condition Bit.
RESV	Pseudo operator that is used to reserve a specified number of data area words or microprogram words.
RETN	Return: use AMPCR+2 as address and as new content of the Microprogram Count Register (MPCR).
RIM	Logical operator Reverse Imply: X RIM Y.
RLOC	Reset location counter to last location previously stored in stack by LOCN.
RMI	Ready MIR bit: set externally when data has been accepted from the Memory Information Register (MIR). This bit is unconditionally true.
SAR	Shift Amount Register.
SAVE	Save the MPCR in the AMPCR: use MPCR+2 as the Microprogram address and as the next MPCR.
SEGMENT	Pseudo used to control program segmentation by the assembler.
SET	Set the Condition Bit specified: either LC1, LC2, INT, GC1, or GC2.
SIZE	Size of specified SEGMENT is stored in MPCR, AMPCR, CPCR, or CNST.
SKIP	Skip the next microinstruction: use MPCR+2 as the Microprogram address and as the next MPCR.
SRQ	Solicited Request bit: set externally and reset appropriate I/O command.
SSC	Shift Store Control: store the first shift-off bit and load the opposite bit or store the Adder Overflow (AOV) condition.
START	Reserved label which indicates the beginning of the microstatements.
STEP	Step to next microinstruction: use MPCR+1 as Microprogram address and as next MPCR.
SYNC	Pseudo operator which causes the microprogram address to be advanced to the next multiple of the specified value.
THEN	Sequential operator for true alternative of a conditional instruction.
URQ	Unsolicited Request bit: set externally and reset by status read.
USEL	Place all devices in unselected mode.
VALUE IS	Pseudo operator used to indicate definition type.
WAIT	Wait Condition bit: Microprogram address is in the MPCR. The MPCR and AMPCR remain unchanged.
WHEN	Same as IF, except that a false-successor of WAIT is produced.
XOR	Logical Operator Exclusive OR: X XOR Y.
Z	Counter (CTR) output in 8 MSB's and Literal register (LIT) outputs in 8 LSB's used as input to the Adder via X-Select and Y-Select gates. In system configurations with word lengths larger than 16 bits, ZOPT inputs from an external source are used for the center bits.
Zero	Hexadecimal@0000@ is placed in the number of memory locations specified by literal.
+	Arithmetic Operator Add: X + Y.
-	Arithmetic Operator Subtract: X - Y.
=	Assignment or set operator.
@	Indicates hexadecimal value.

LOGIC UNIT (LU1, LU2, LU5, LU6, AND LU7)

The Logic Unit (LU1, LU5, LU6, and LU7 card assemblies) is one of the major functional units of the processor. The Logic Unit is used to perform all of the arithmetic, Boolean logic, and shifting operations in the B 700 Processing system. The Logic Unit consists of the following functional circuits.

a. A Register (A1, A2, and A3).
b. X-Select Gates.
c. B Register and Associated Input Selection Gates.
d. Y-Select Gates.
e. Arithmetic/Logic Unit (ALU or Adder) and Group Carry Gates.
f. Barrel Switch (BSW) and Associated Control Circuits.
g. Memory Information Register (MIR).

See figure II-2 for the functional circuits of the Logic Unit and their interconnection with the other functional units of the Processing system. Figure II-9 is a functional block diagram of the Arithmetic/Logic Unit (Adder) and associated input circuits.

The Logic Unit provides for a system word length of 16-bits. Each Logic Unit card assembly (LU2, LU6, and LU7) is designed for a four-bit configuration; therefore, a 16-bit system configuration requires four LU2, two LU6 and two LU7 card assemblies. The following registers within the Logic Unit contain 16-bits in four-bit increments: the A registers (A1, A2, and A3), the B register, and the memory information register (MIR).

A REGISTERS (A1, A2, AND A3)

The three A registers are functionally identical and are used for temporary data storage within the Processing system and serve as a primary input to the Arithmetic/Logic Unit (Adder).

Any or all of the A registers may be loaded with the Barrel Switch (BSW) output during one operation. The presence of nanobit control pulse (NB34) from the Control Register of the Memory Control Units (MU3 card) enables the clock pulse to the A1 register. Nanobit control pulse (NB35) enables the clock pulse to the A2 register and NB36 enables the clock pulse to register A3. The phase 3 clock signal (3CLKB*C) from the clock generator circuits of the CG card is used as the clock pulse to the A registers.

The nanobit control signals (NB34, NB35, and NB36) correspond to the nanobits (N34, N35, and N36) from the Nanomemory. These nanobits are used to determine A register input selection. (See nanobits 34 through 36 on figure II-4.) If nanobit N34 is at a 0 level, the A1 register is unchanged. If N34 is at a 1 level, the Barrel Switch (BSW) output is sent to register A1. Likewise, if nanobit N35 or N36 is at a 0 level, the respective A register (A2 or A3) remains unchanged.

If either of these nanobits is at a 1 level, the appropriate BSW output is sent to the respective A register (A2 or A3).

The outputs of these three A registers are sent to the X-select gates, which select the appropriate input (A1, A2, A3, Z, LIT, CTR, or ZEROS) to the Arithmetic/Logic Unit (Adder).

X-SELECT GATES

The X-select gates are used to select the X input to the Arithmetic/Logic Unit (ALU or Adder). The X selection to the Adder consists of the following inputs:

ZERO, LIT, CTR, Z (CTR/LIT), A1, A2, or A3.

Three signals (NQE1, NQE2, and NQE3) from the Control Register in the Memory Control Units (MU2 card) are used as enable signals for the X-select gates. These three enable signals are used to inhibit unwanted bits during a Literal register (LIT) and Counter (CTR) transfers through the X-select gates. If all three enable signals are false (NQE1, NQE2, and NQE3), a ZERO output is generated from the X-select gates. The NQE1 enable signal is used for the most significant byte (8-bits) during Counter (CTR) transfer through the X-select gates. Only enable signals NQE1 and NQE3 are used in a 16-bit system configuration to transfer Counter (CTR) and Literal register (LIT) inputs through the X-Select gates. The NQE3 enable signal is used for the least significant byte (8-bits) during Literal register (LIT) input transfer through the X-select gates.

Two additional control signals (ACTL1 and ACTL2) from the Control Register of the Memory Control Circuits (MU2 card) are used as controls for the selection of the A Register and Z inputs in the X-select gates. The following list indicates the levels for the ACTL

signals and the functions specified:

ACTL1	ACTL2	Function Specified
0	0	Selection of A3 Register.
0	1	Selection of A2 Register.
1	0	Selection of A1 Register.
1	1	Selection of Z inputs (CTR/LIT).

Nanobits N17 through N19 from the Nanomemory are sent to the Control Register input gates on the MU2 card to generate the aforementioned enable and control signals (NQE and ACTL signals). These nanobit signals (N17 through N19) are used to determine X Selection to the Adder. (See nanobits 17 through 19 in figure II-4.) The following list indicates the nanobit codes for X-selection to the Arithmetic/Logic Unit (ALU or Adder):

Nanobits			
17	18	19	X-Selection To Adder
0	0	0	ZERO
0	0	1	Literal Register (LIT) to most significant byte
0	1	1	Counter (CTR) to least significant byte
1	0	0	Z inputs (CTR/LIT)
1	0	1	A1 Register
1	1	0	A2 Register
1	1	1	A3 Register

The output signals from the X-select gates are sent to the Arithmetic/Logic Unit (ALU or Adder).

B REGISTER AND ASSOCIATED INPUT SELECTION GATES

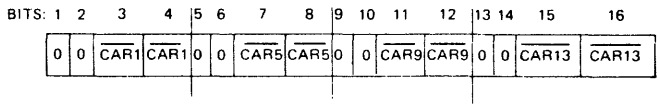
The B register and associated input selection gates are the primary interface between the main memory and the Device Dependent Ports (DDP's). The B register serves as the secondary input to the Arithmetic/Logic Unit (Adder) and can store certain arithmetic operation functions, such as complements of carries (BC4 and BC8). Figure II-10 shows the gating of the complement of group carries. The B register can be loaded via the B input selection gates with any of the following inputs during one operation (one instruction):

a. Barrel Switch outputs (BSWnn signals).

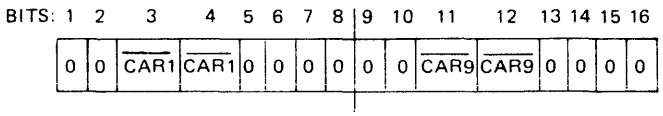
b. Adder output (ADDnn signals).

c. External bus (EXTnn signals) for processor status information, from Data/Program Memory, from the Device Dependent Ports (DDP's), or for system configuration data.

d. The complements of the carries of the four-bit Adder groups (CAR1, 5, 9, and 13 signals) are placed in the two least-significant bits (LSB's) of the corresponding four-bit B register groups, with ZERO's in the remaining bits. The following example is for a four-bit B register groups (BC4):



e. The complements of the carries of the eight-bit Adder groups (CAR01, 09 signals) are placed in the third and fourth bits of each 8-bit group (byte), with ZERO's in the remaining bits. The following example is for an eight-bit B register groups (BC8):



f. Barrel Switch outputs (BSW1-16 signals) ORed with Adder outputs (ADD1-16 signals).

g. Barrel Switch outputs ORed with data from the External Bus (EXT1-16 signals).

h. Contents of the Memory Information Register (MIR1-16 signals).

i. Barrel Switch outputs ORed with the contents of the Memory Information Register (MIR1-16 signals).

Nanobit control signals NB37, NB38, NB39, and NB40 that are not from the Control Register of the Memory Control Unit (MU3 and 5 card) and which correspond to nanobits N37 through N40 of the Nanomemory, are used to control B register input selection. (Refer to nanobits N37 through N40 in figure II-4.) If nanobits N37 through N40 are at a 0 level, there is no change in the B register. The presence of a 0001 in nanobits N37 through N40 indicates the input source selection of the complement of four-bit carries (BC4) from the Adder to the B register. A 1000 in nanobits N37 through N40 indicates the input source selection of the Adder output (BAD) to the B register. Table II-4 lists the remaining nanobit codes for B register input source selection.

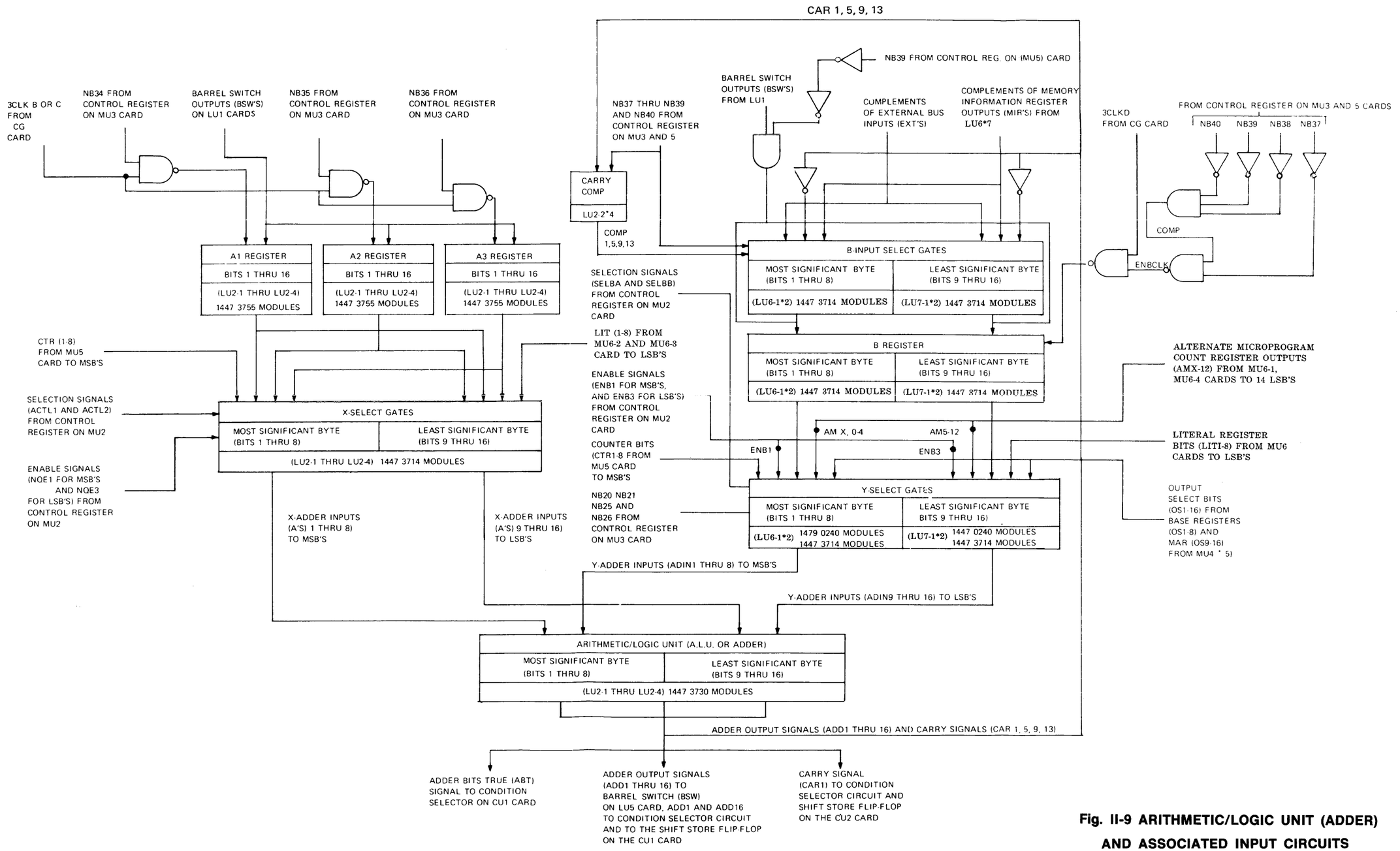


Fig. II-9 ARITHMETIC/LOGIC UNIT (ADDER) AND ASSOCIATED INPUT CIRCUITS FUNCTIONAL BLOCK DIAGRAM

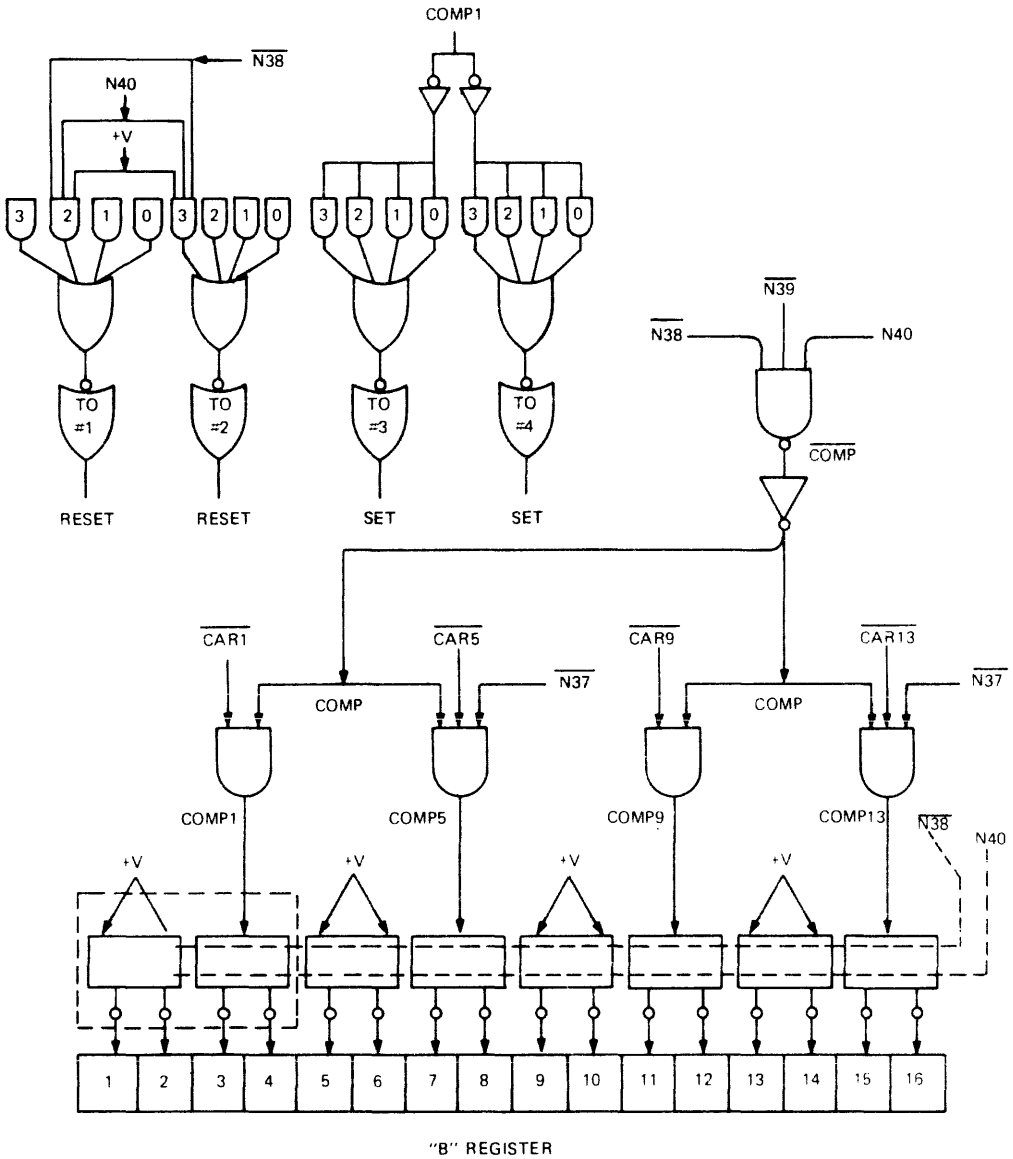
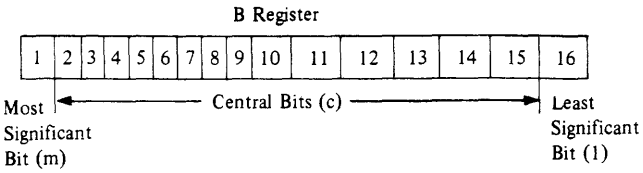


Fig. II-10 GATING OF THE COMPLEMENT OF GROUP CARRIES

TABLE II-4 NANOBIT CODES FOR B REGISTER INPUT SOURCE SELECTION

Nanobits					Reserved
37	38	39	40	B Register Input Source	Word
1	0	0	1	Complement of 8-bit carries from Adder	BC8
1	0	1	0	Barrel Switch outputs ORed with Adder outputs	BBA
1	0	1	1	Barrel Switch (BSW) outputs	B
1	1	0	0	External inputs from External Bus	BEX
1	1	0	1	Contents of the Memory Information Register (MIR)	BMI
1	1	1	0	Barrel Switch outputs ORed with data from the External Bus	BBE
1	1	1	1	Barrel Switch outputs ORed with the data from the Memory Information Register (MIR)	BBI

The B register is 16-bits in length. The word length, minus 2, determines the number of central bits contained in the B register. The number of central bits is 14. The following example presents the 16-bit word, which contains 14 central bits:



In this example, the B register can be represented by B or Bmcl, where B denotes the B register, m denotes the most significant bit (MSB), c denotes the central bits, and the l denotes the least significant bit (LSB).

The output of the B register is sent to true/complement selection gates (Y-select gates) which are divided into three sections: the most significant bit (m), the least significant bit (l) and the remaining central bits (c). Each of these three sections is controlled independently and may contain either 0's, 1's the true contents (T) or the complement (ONE's complement) of the contents (F) of the respective bits in the B register. Although there are no programmatic restrictions on the m, c, and l specifiers, the central bits (c) configuration is restricted, by the hardware, to a 0 or T function. Due to this hardware limitation, when a BxlX or a BxFx is specified, their complements are used (Bx0x or BxTx). The following example specifies the conditions that are represented by the following Bmcl specifiers for a 16-bit system configuration:

Bmcl Specifier (MOST / CENTER / LEAST)	Condition
B 0 – –	ZERO in Bit 1 (MSB) to ALU Y input
B T – –	True value of B reg Bit 1 (MSB) to ALU Y input
B F – –	Complement value of B reg Bit 1 (MSB) to ALU Y input
B 1 – –	ONE in bit 1 (MSB) to ALU Y input
B – 0 –	ZERO's in bits (2 through 15) to ALU Y input
B – T –	True value of B reg bits (2 through 15) to ALU Y input
B – F – *	Complement value of B reg bits (2 through 15) to ALU Y input
B – 1 – *	ONE's in bits (2 through 15) to ALU Y input
B – – 0	ZERO in bit 16 (LSB) to ALU Y input
B – – T	True value of B reg bit 16 (LSB) to ALU Y input
B – – F	Complement value of B reg bit 16 (LSB) to ALU Y input
B – – 1	ONE in bit 16 (LSB) to ALU Y input

NOTE: – indicates insignificant bit.
* requires complement Y input operation.

The actual contents of the B register or a modified configuration of its contents are sent to the Y-Select gates. As stated previously, the B register can be represented as B or Bmcl. The use of B alone implies that the actual contents of the B register, as a unit, are sent to the Y-Select gates. The use of Bmcl enables a reconfiguration of the contents of the B register to be sent to the Y-Select gates, without affecting the actual contents of the B register.

Y-SELECT GATES

The Y-select gates are used to select the Y input to the Arithmetic/Logic Unit (ALU or Adder). The Y selection to the Adder consists of the following inputs: ZERO, B, LIT, CTR, Z(CTR/LIT, AMPCR, or BMAR.

Two enable signals (ENB1 and ENB3) from the Control Register in the Memory Control unit (MU2 card) are two out of eight control signals used by the Y-select gates. These two enable signals are used to inhibit unwanted bits during Literal register (LIT), or Counter (CTR) transfers through the Y-Select gates. If both enable signals are true (ENB1 and ENB3), a ZERO output is generated from the Y-Select gates bits 2 through 15. The ENB1 enable signal is used for the most significant byte (8-bits) transfer of Counter (CTR) inputs through the Y-Select gates. The ENB3 enable signal is used for the least significant byte (8-bits) transfer of Literal register (LIT) inputs through the Y-Select gates.

Two more control signals (SELBA and SELBB) from the Control Register on the MU2 card are used as controls for the selection of the following inputs to the Y-Select gates: B register, Alternate Microprogram Count Register (AMPCR). Base Register 1 or Base Register 2 with Memory Address Register (MAR), or Z inputs (CTR/LIT). The following list indicates the levels for the SELB selection control signals and the function specified at these levels:

SELBA	SELBB	Function Specified
0	0	Selection of B register (bit 1 thru 16)
1	0	Selection of CTR for MS byte and LIT for LS byte
0	1	Selection of AMPCR (bits X, 0 thru 12) for 14 least significant bits.
1	1	Selection of BR1 or BR2 and MAR (OS bits 1 thru 16)

The remaining four control signals used by the Y-Select gates are nanobit control signals NB20, NB21, NB25, and NB26 from the Con-

trol Register of the Memory Control units (MU3 card). Nanobit control signal NB20 corresponds to nanobit N20 from the Nanomemory. This signal, along with NB21, controls the most significant bit (MSB) of the Y-Select gates. Nanobit control signal NB20 is connected to ground on all LU3 and LU4 cards (Y-Select gates) except for the LU3-1 card that contains the most significant bit (Y-MSB Select gate).

Nanobit control signal NB21 corresponds to nanobit N21 from the Nanomemory. This signal, along with NB20, controls the most significant bit (MSB) of the Y-Select gates. The nanobit control signal NB21 input is connected to ground or ENB3 on al LU6 and LU7 cards (Y-Select gates) except for the LU6-1 card that contains the most significant bit (Y-MSB Select gate).

Nanobit control signal NB25 corresponds to nanobit N25 from the Nanomemory. This signal, along with NB26, controls the least significant bit (LSB) of the Y-Select gates. Nanobit control signal NB25 is connected to ground on all LU6 and LU7 cards (Y-Select gates, except for the LU7-2 card that contains the least significant bit (Y-LSB Select gate).

Nanobit control signal NB26 corresponds to nanobit N26 from the Nanomemory. This signal, along with NB25, controls the least significant bit (LSB) of the Y-Select gates. The nanobit control signal NB26 input is connected to ground an ENB1 on all LU6 and LU7 cards (Y-Select gates), except for the LU7-2 card that contains the least significant bit (Y-LSB Select gate).

Nanobits N20 through N26 from the Nanomemory are sent to the Control Register input circuits on the MU2 and MU3 cards to generate the aforementioned enable and selection control signals (ENB1 and ENB3, SELBA and SELBB, NB20, NB21, NB25, and NB26). Nanobit signals N20, N21, N25, and N26 are direct inputs to the Control Register flip-flop on the MU3 card that generate the respective nanobit control signals (NB20, NB21, NB25, and NB26). Nanobit signals N22, N23, and N24 are sent to the Control Register input gates on the MU2 card that are used in the generation of enable signals (ENB1 through ENB3) and selection control signals (SELBA and SELBB). These nanobit signals (N20 through N26) are used to determine Y-Selection to the Adder. (Refer to nanobits N20 through N26 on figure II-4.) Table II-5 lists the nanobits codes for Y-selection to the Arithmetic/Logic Unit (ALU or ADDER).

The output signals from the Y-Select gates (ADIN01 through 16 signals) are sent to the Arithmetic/Logic Unit (ALU or ADDER).

ARITHMETIC/LOGIC UNIT (ALU OR ADDER)

The Arithmetic/Logic Unit (ALU or Adder) is the unit which performs all of the arithmetic and Boolean operations allowed in the Processing system configuration. The inputs to the Arithmetic/Logic Unit can be comprised of various combinations of the following:

a. A Register (A1, A2, or A3).

TABLE II-5 NANOBIT CODES FOR Y-SELECTION TO ALU OR ADDER

Nanobits							
20	21	22	23	24	25	26	Y-Selection to Adder
0	0						ZERO (0) to the most significant bit (MSB) of Adder.
0	1						B register MSB to MSB of Adder.
1	0						Complement of B register LSB to the LSB of the Adder.
1	1						ZERO to Adder.
		0	0	0			ZERO to Adder.
0	0	0	0	1	0	1	Literal Register (LIT) to least significant byte (8-bits) of Adder.
0	1	0	1	1	0	0	Counter (CTR) to most significant byte (8-bits) of Adder.
		1	0	0			B register input to Adder.
0	1	1	0	1	0	1	Z inputs (CTR/LIT) to Adder.
0	0	1	1	0	0	1	Alternate Microprogram Count Register (AMPCR) to 14 LSB's of Adder.
0	1	1	1	1	0	1	Base Register 1 or Base Register 2 and Memory Address Register (MAR) to 16 LSB's of Adder.
					0	0	ZERO (0) to least significant bit (LSB) of Adder.
					0	1	B register LSB to LSB of Adder.
					1	0	Complement of B register LSB to LSB of Adder.
					1	1	ONE (1) to the least significant bit (LSB) of Adder.

b. B Register.
c. Z inputs (CTR/LIT).
d. Alternate Microprogram Count Register (AMPCR).
e. Base Register 1 or Base Register 2 and Memory Address Register (MAR).
f. Literal Register (LIT).
g. Counter (CTR).

The microinstruction of the program specifies two of the various operands upon which the Arithmetic/Logic Unit (ALU or Adder) is to function. Each of the two specified operands (registers) are gated into two gate circuits: X-Select gates and Y-Select gates.

The X-Select gates provide the following inputs to the ALU as the "A" input signals: A1 Register, A2 Register, A3 Register, Z inputs (CTR/LIT), Counter (CTR) to most significant byte (8-bits), and Literal Register (LIT) for least significant byte (8-bits). The X-Select gates function only as a register selector.

The Y-Select gates provide the following inputs to the ALU as ADIN1 through 16 signals: B register, Z inputs (CTR/LIT), Counter (CTR) to the most significant byte (8-bits) of the ALU, Literal Register (LIT) to the least significant byte (8-bits) of the ALU, Alternate Microprogram Count Register (AMPCR) to the 14 least significant bits of the ALU, and Base Register 1 or Base Register 2 along with Memory Address Register (MAR) to all 16-bits of the ALU. The Y-Select gates permit the masking of the B register to the ALU without affecting the original values contained in the B register. The Y-Select gates also function as a register selector.

The four Adder Selection control signals (AS0 through AS3) are generated by the Control Register of the Memory Control Units (MU3 card). These Adder selection control signals are used with the AMODE signal to select the proper arithmetic or logic operation of the Arithmetic/Logic Unit MSI module. The AMODE (Adder mode) signal is generated by the Control Register of the Memory Control Units (MU3 card). When the AMODE signal is true, this indicates that an arithmetic operation is to be performed by the Arithmetic/Logic Unit. When the AMODE signal is false, a logical operation is to be performed by the Arithmetic/Logic Unit. Table II-6 lists the levels of the AMODE and AS0 through AS3 signals and the selected operation to be performed formed by the Arithmetic/Logic Unit.

Nanobits N27 through N31 from the Nanomemory are sent to the Control Register flip-flops of the Memory Control Units (MU3 card). The outputs from these Control Register

TABLE II-6 AMODE AND ASO THROUGH AS3 OPERATIONS

NOTE

In the following list, a + symbol represents a plus (add function), a – symbol represents a minus (subtract function), a v symbol represents an OR function, a $\overline{v}$ symbol represents an exclusive-OR function, and a ● symbol represents an AND function.

AMODE	AS3	AS2	AS1	AS0	Specified Operation	Reserved Word
1	1	0	0	1	$\overline{X+Y}$	+ (add)
0	0	0	0	1	$\overline{XvY}$	NOR
0	0	0	0	0	$\overline{x}$●y	NIR
1	1	0	0	1	x+y+1	+ +1 (add +1)
0	0	1	0	0	$\overline{X●Y}$	NAN
1	1	1	0	1	(XvY)+X	OAD
0	0	1	1	0	C$\overline{v}$Y	XOR
0	0	1	1	1	X●$\overline{Y}$	NIM
0	1	0	0	0	$\overline{X}$vY	IMP
0	1	0	0	1	$\overline{X\overline{v}Y}$	EQV
1	1	0	0	0	X+(X●Y)	AAD
1	1	0	0	0	X+(X●Y)	AAD
0	1	0	1	1	X●Y	AND
1	0	1	1	0	X-Y-1	- -1 (subtract-1)
0	1	1	0	1	XvY	RIM
0	1	1	1	0	XvY	OR
1	0	1	1	0	X-Y	- (subtract)

flip-flops (NB27 through NB31) are sent to the various Control Register output gates, which are used to generate the following control signals: AMODE, AS0 through AS3, CIN, and INH8. These control signals are used by the Arithmetic/Logic Unit to perform various arithmetic and logic functions.

The Adder Mode (AMODE) and Adder Selection control signals (AS0 through As3) are the same as those described in the previous text. The Carry Input (CIN) signal is sent to the least-significant bit (LSB) of the Arithmetic/Logic Unit on the LU2-4 card. The Carry Input signal is used for the following arithmetic operations: X+Y+1 and X–Y. The Inhibit 8-bit carry signal (INH8/) from the output gate of Control Register flip-flop NB27 corresponds to nanobit N27 from the Nanomemory. When the Inhibit 8-bit NOT carry signal is false, it inhibits the carryout of the eight-bit groups of the Adder (Arithmetic/Logic Unit). This signal is gated with the Carry (CAR09) from least Byte, which is generated on the LU2-3 card, and is sent to the LU2-2 card which contains the least bit of eight most significant bits (eight MSB's) of the Adder. Nanobit signal N27 is used to determine whether a carry between eight-bit bytes is allowed or inhibited. (Refer to nanobit N27 in figure II-4.) The nanobit code for the inhibiting of carries between eight-bit bytes is as follows:

Nanobit 27	Function Specified
0	Allow carry between 8-bit bytes.
1	Inhibit carry between 8-bit bytes.

Nanobit signals N28 through N31 are used to determine Arithmetic/Logic Unit operation. Reference should be made to the previous list describing Adder Selection control signals (AS0 through AS3). Adder Mode signal (AMODE), and the operation specified by the designated logic levels for those control signals. Because nanobits N28 through N31 are used to generate the aforementioned signals, the test in table II-7 should also be used in determining Arithmetic/Logic Unit operation.

The Adder output signals (ADD01 through 16) from the Arithmetic/Logic Unit are sent to the Barrel Switch (BSW) on the LU5 card and are also used as inputs to the B register via the B input select gates on the LU6 and LU7 cards. The most significant bit (MSB) and least significant bit (LSB) of the Adder output signals are sent to the Condition Selector circuits of the Control Unit (CU1 card) for condition testing. The MSB and LSB of the Adder

TABLE II-7 NANOBIT N28 THROUGH N31 OPERATIONS

NOTE

In the following list, a + symbol represents an add function, a – symbol represents a subtract function, a v symbol represents an OR function, a ⊻ symbol represents an exclusive-OR function, and a ● symbol represents an AND function.

Nanobits 28	29	30	31	Specified Operation	Reserved Word
0	0	0	0	$\underline{X+Y}$	+ (add)
0	0	0	1	$\overline{X \vee Y}$	NOR
0	0	1	0	$X \bullet \overline{Y}$	NR1
0	0	1	1	X+Y+1	+ +1 (add +1)
0	1	0	0	$\overline{X \bullet Y}$	NAN
0	1	0	1	(XvY)+X	OAD
0	1	1	0	$X \veebar \underline{Y}$	XOR
0	1	1	1	$X \bullet \overline{Y}$	NIM
1	0	0	0	$\overline{X} \vee \underline{Y}$	IMP
1	0	0	1	$\overline{X \veebar Y}$	EQV
1	0	1	0	X+(X●Y)	AAD
1	0	1	1	X●Y	AND
1	1	0	0	X–y–1	– –1 (subtract –1)
1	1	0	1	$X \vee \underline{Y}$	RIM
1	1	1	0	$X \vee \overline{Y}$	OR
1	1	1	1	X–Y	– (subtract)

output signals are also sent to the Shift Store control gates of Shift Control Register on the CU2 card via the CU1 card.

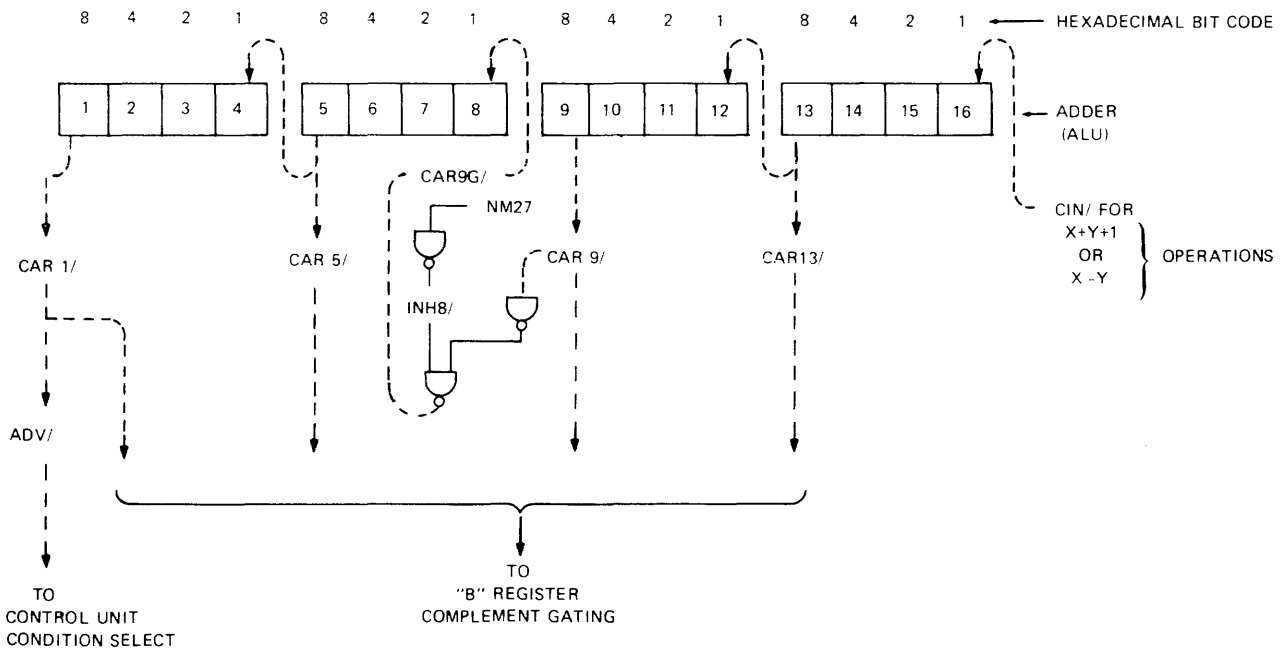
The complement four-bit and eight-bit carry signals (CAR1, 5, 9 and 13), which are also generated by the ALU, are sent to the respective B input select Complement Control gates on the LU2-1 through LU2-4 cards. Refer to the B register and associated input selection gates, along with figure II-11, for information concerning the four-bit and eight-bit carry signals.

The carry signals (CAR1, 5, 9 and 13) are sent to the next higher-significant Arithmetic/Logic Unit module on the LU2-1 through LU2-4 cards. The carry signal indicates a carry out of bit 1, 5, 9, or 13 of the ALU's. The respective carry signal is used as an input to bit "nn-1" of the ALU module. For example, the CAR13 signal is sent to the ALU module that contains Adder bit-12 (nn-1) and CAR9 carry signal is sent to the CN input of the ALU module that contains Adder bit-9. The most significant carry bit signal (CAR1) is sent to the Condition Selection circuits of the Control Unit (CU1 card) for Adder overflow (AOV) test purposes.

BARREL SWITCH (BSW) AND ASSOCIATED CONTROL CIRCUITS

The Barrel Switch (BSW) is a matrix of gates used to shift a parallel input data word a number of places to the left or right, either end-off or end-around. See figures II-2 and II-3 for a functional block diagram of the Barrel Switch (BSW) and shift control circuits.

The Barrel Switch (BSW) Enable Gating circuits consist of the BSW Control gate modules (1447 3722) and their associated input gate circuits. The BSW enable gates generate two sets of internal Enable Shift signals. These Enable Shift signals are used to control shifting operations in the first and second level of gating of the Barrel Switch (BSW). The following list indicates which internal enable shift signals are



TYPICAL CARRY EXAMPLE:

ADDER	X INPUT	= 7	C	5	B	HEXADECIMAL CODED DIGITS
ADDER	Y INPUT	0	3	E	4	
		+ ← OPERATION = ADD				
ADDER	OUTPUT	8	0	3	F	
CARRY	OUTPUT	0	1	1	0	

Fig. II-11 ADDER CARRIES

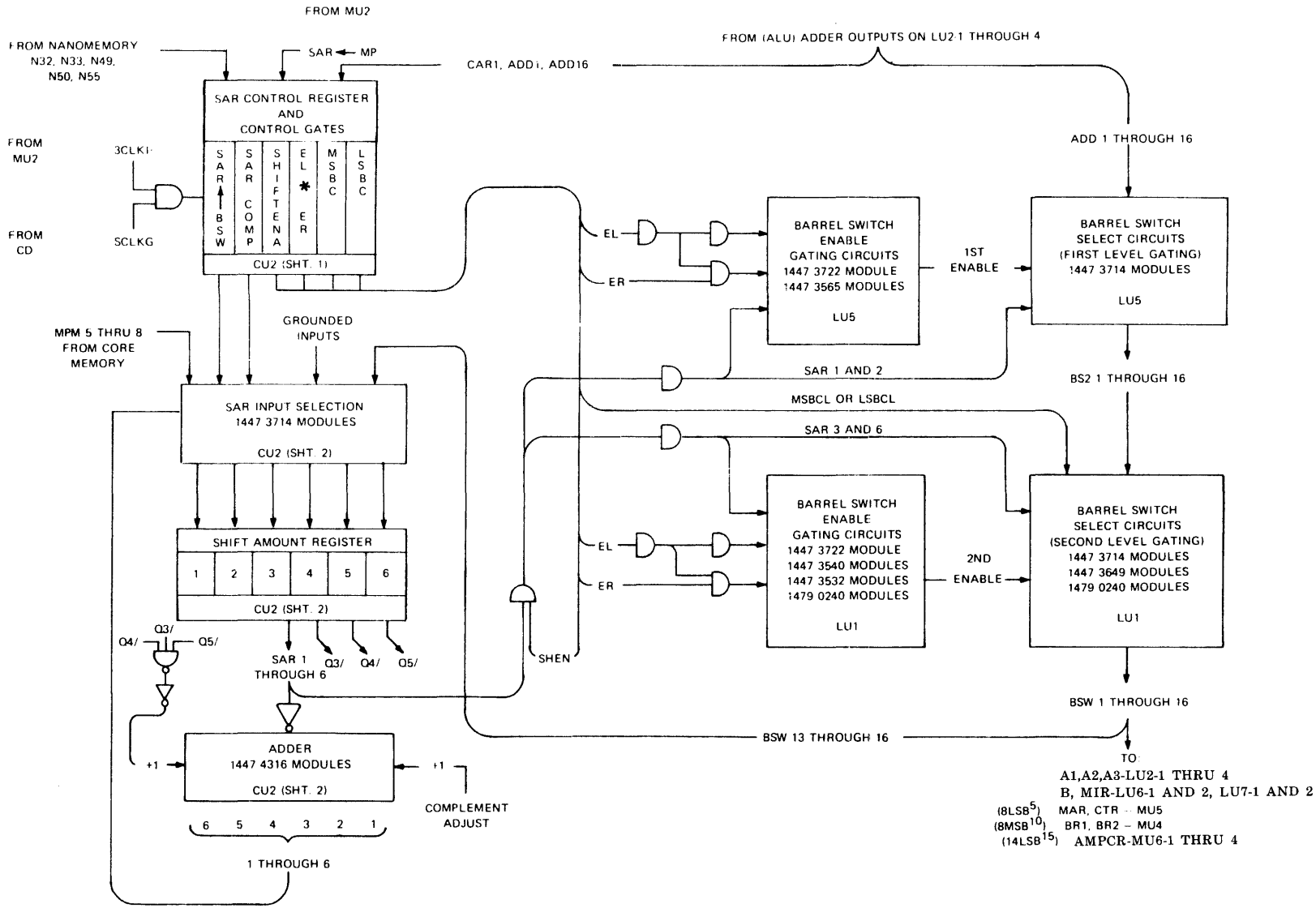


Fig. II-12 BARREL SWITCH (BSW) AND SHIFT REGISTER CONTROL CIRCUITS, BLOCK DIAGRAM

used by the Barrel Switch (BSW) first and second level gates:

Internal Enable Shift Signals	BSW Outputs Enabled
First stage most significant four-bit group	BS201, BS202, BS203, BS204
First stage second most significant four-bit group	BS205, BS206, BS207, BS208
First stage third most significant four-bit group	BS209, BS210, BS211, BS212
Second stage most significant bit	BSW01
Second stage second most significant bit	BSW02
Second stage third most significant bit	BSW03
Second stage least significant bit	BSW16
Second stage second least significant bit	BSW15
Second stage third least significant bit	BSW14
Second stage fourth least significant bit	BSW13

These enable shift signals are generated from the SAR signals of the output gates of the Shift Amount Register (SAR) on the Control Unit (CU2 card) and the Enable Left shift and Enable Right shift control signals (EL and ER) of the shift control circuits, which are also located on the CU2 card. The Enable Left shift signal is true (EL) for both left shifts and right end around (circular) shifts. The Enable Right shift signal (ER) is true at all times except during a left shift. The Enable Left shift and Enable Right shift signals are controlled by the "Shift Selection" nanobits (N32 and N33) from the Nanomemory. Table II-8 lists the nanobit codes for Shift Selection, the Shift operation performed, and the values for ER, EL, and SAR.

For information concerning the use of the Enable Right shift (ER) and Enable Left shift (EL) signals, refer to the Shift Operations information contained under the Shift Amount Register (SAR) heading and the Shift Control Register heading of the CU2 card.

Left shift Barrel Switch operations are actually right shifts using the 2's complement of the desired shift amount value; therefore, all left shifts must be preceded by a Complement SAR Function on the desired shift amount or the 2's Complement as a predetermined constant loaded from MPM to SAR.

The SAR signals CSAR1 and CSAR2 from the output gates of the Shift Amount Register (SAR) control the first level of gating of the Barrel Switch. Signals CSAR3 and CSAR6 control Barrel Switch, Signals CSAR3 and CSAR6 control the second level of gating of the Barrel Switch (BSW). Information concerning the SAR signals and their "complements" for system configurations with word lengths of 16-bits is provided in the Shift Operations functional description under the Shift Amount Register (SAR) heading of the CU2 card.

When the internal Enable Shift signals are at a high level, these signals inhibit that particular controlled Barrel Switch bit or group of Bits by forcing it to ZERO. When an Enable Shift signal is at a low level, the particular Barrel Switch bits associated with the signals are enabled.

The BSW enable gate modules (1447 3722), which generate the internal enable shift signals use two select inputs (A and B) for SAR inputs 2 and 1, or SAR3 and 6, respectively to direct data input or complement data input through to one of four outputs on either 1Y or 2Y gates. 1Y uses NORMAL data input and a strobe gate, while 2Y uses complement data input and a strobe gate. The data and strobe gate inputs are from Enable Left (EL) and the

TABLE II-8 NANOBIT CODES N32 AND N33, SHIFT OPERATIONS

Nanobits N32	Nanobits N33	Shift Operation	ER	EL	SAR Lines
0	0	No shift	1	0	All ZEROS
0	1	Shift right end-off	1	0	SAR value
1	0	Shift left end-off	0	1	SAR value
1	1	Circular shift (right end around)	1	1	SAR value

Enable Right (ER) shift signals. Table II-9 lists various combinations of EL = ER, and SAR the various combinations of EL, ER, and SAR signals to the modules.

The 1Y or 2Y outputs for the first stage are sent to the gating module (1447 3563) to develop only three group enabling signals. When any of these three signals is high, the four BS2nn first stage outputs are disabled at one time.

The second stage BSW enable gating functions just as the first stage, except that SAR3 and SAR6 are used for selecting. The final outputs of the enable gates are used to disable seven different BSW output signals.

The functional equations for the internal enable shift signals are as follows:

The Shift Amount Register (SAR) value (0 through 15) and the shifting of Adder output signals ADD1 through ADD16 during a left shift (L) or right shift (R) are shown in figure II-13. Figure II-13 indicates the Barrel Switch first level of gating. The Barrel Switch interconnection signals BS21 through BS216 are used between the first and second levels of gating of the Barrel Switch. For a ZERO shift, BS201 through BS216 output signals are the equivalent of the ADD01 through ADD16 input signals.

Figure II-13 indicates the Barrel Switch second level of gating. Barrel Switch output signals BSW1 through BSW16 use the corresponding BS2nn outputs from the first level of

ENABLE 1st STAGE MOST SIG. 4-BITS = $ER \cdot EL + \overline{ER} \cdot EL(SAR1 + SAR2) + ER \cdot \overline{EL}(\overline{SAR1 + SAR2})$

ENABLE 1st STAGE 2nd MOST SIG. 4-BITS = $ER \cdot EL + \overline{ER} \cdot EL \cdot SAR1 + ER \cdot \overline{EL} \cdot \overline{SAR1}$

ENABLE 1st STAGE 3rd MOST SIG. 4-BITS = $ER \cdot EL + \overline{ER} \cdot EL(SAR1 \cdot SAR2) + ER \cdot \overline{EL}(\overline{SAR1 \cdot SAR2})$

ENABLE 2nd STAGE MOST SIG. BIT = $EL + ER \cdot \overline{EL}(\overline{SAR3 + SAR6})$

ENABLE 2nd STAGE 2nd MOST SIG. BIT = $EL + ER \cdot \overline{EL} \cdot \overline{SAR3}$

ENABLE 2nd STAGE 3rd MOST SIG. BIT = $EL + ER \cdot \overline{EL}(\overline{SAR3 \cdot SAR6})$

ENABLE 2nd STAGE LEAST SIG. BIT = ER

ENABLE 2nd STAGE 2nd LEAST SIG. BIT = $ER + \overline{ER} \cdot EL(SAR3 \cdot SAR6)$

ENABLE 2nd STAGE 3rd LEAST SIG. BIT = $ER + \overline{ER} \cdot EL \cdot SAR3$

ENABLE 2nd STAGE 4th LEAST SIG. BIT = $ER + \overline{ER} \cdot EL(SAR3 + SAR6)$

TABLE II-9 COMBINATIONS OF SAR, EL, AND ER SIGNALS

	Select B and A SAR		Strobe Data		Low Outputs	Low Outputs	
1st Stage	1	2	EL	ER	1Y	2Y	
2nd Stage	3	6					
	0	0	0	0	None	0	High = Enable
	0	0	1	0	0	None	
	0	0	1	1	None	None	
	0	1	0	1	None	1	
	0	1	1	0	1	None	Low = Disable for 1st or 2nd Stage BSW Output Control
	0	1	1	1	None	None	
	1	0	0	1	None	2	
	1	0	1	0	2	None	
	1	0	1	1	None	None	
	1	1	0	1	None	3	
	1	1	1	0	3	None	
	1	1	1	1	None	None	

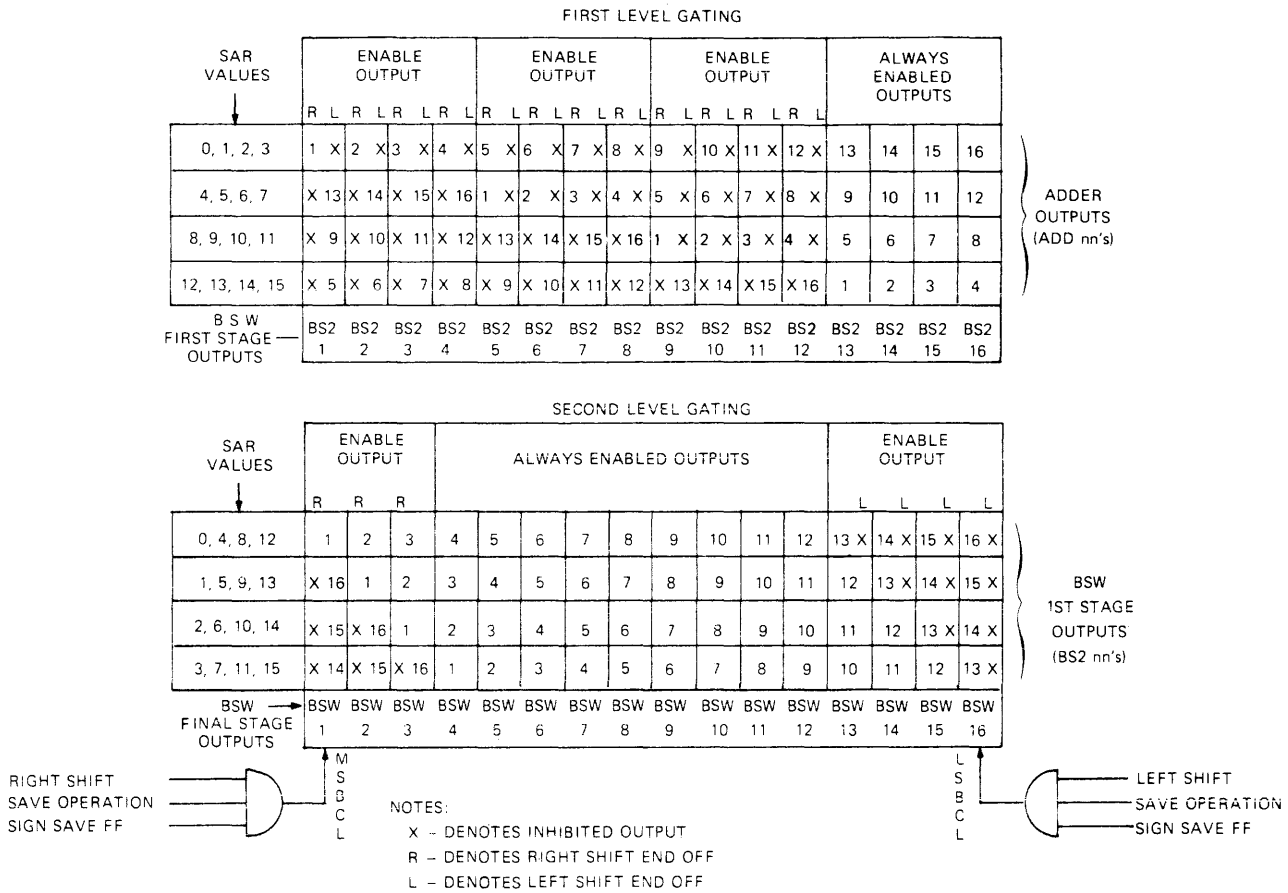


Fig. II-13 BARREL SWITCH (BSW) GATING

BSW gating which had been shifted only for shifts in increments of 4 in (SAR), and accomplishes the shifts required for increments of 1, 2, or 3 (the balance of the shifting not done in the first stage).

Barrel Switch output signals BSW1 through BSW16 are used as the major data outputs of the Logic Unit (LU1 card) that are sent to the following Processor cards:

BSW Destinations	Card Location
A Registers (A1, A2, and A3)	Logic Unit (LU2-1 thru 4)
B-Input Select Gates	Logic Unit (LU6-1 and 2, LU7-1 and 2)
Memory Information Register (MIR)	Logic Unit (LU6-1 and 2, LU7-1 and 2)
Shift Amount Register (SAR), four LSB's of BSW	Control Unit (CU2)
Memory Address Register (MAR), eight LSB's of BSW	Memory Control Unit (MU5)
Counter (CTR), eight LSB's of BSW	Memory Control Unit (MU5)
Base Registers (BR1 and BR2), eight MSB's of BSW	Memory Control Unit (MU4)
Alternate Microprogram Count Register (AMPCR), LSB's of BSW	Memory Control Unit (MU6-1 through 4)

MEMORY INFORMATION REGISTER (MIR)

The Memory Information Register (MIR), which is located on the LU6-1 and LU6-2, LU7-1 and LU7-2 card assemblies, is used to buffer output information that is written into Data/Program Memory (DPM), or sent to a peripheral device. See figure II-2 for the MIR interface within the Processing system. The MIR is also used as input to the "B" Register for processor manipulation of data.

Barrel Switch outputs BSW01 through BSW16 are applied to the respective MIR flip-flops. Control signal DN41 from the Control Register of the Memory Control Unit (MU2 card), which corresponds to nanobit N41 from the Nanomemory, enables the clock pulse to the Memory Information Register. The DN41 signal and the phase 3 clock signal (3CLKD), are applied to NAND gate and are used to clock the MIR flip-flops. Nanobit signal N41 from the Nanomemory is used to determine

MIR input selection, as follows:

Nanobit 41	MIR Input Selection
0	No change in the contents of the MIR.
1	Barrel Switch (BSW) outputs to the MIR.

The Clear signal (CLEAR) is used to reset the MIR. MIR outputs (MIR01 through MIR16 signals), which supply output data from the processor, are sent to the Data/Program Memory (DPM), the Device Dependent Port (DDP) for a peripheral device, or can be used as inputs to the B input select gates of the B register. The MIR BUS from the LU6 and LU7 cards goes to buffer card MIRD1, the output of which is the MIRD bus. This bus goes to the Shared Memory Unit and the DDP's not in the logic/memory rack (DDP1, 2, 3, 5, 6, 7 and 8).

Figure II-14 is a functional block diagram of the Processing system output area, which includes the input and output logic circuits associated with the MIR. This diagram presents the interface between the Output Select circuits (Base Register 1, Base Register 2, and Memory Address Register), MIR, Port Selector (PS1 card), "S" Memory (DPM), and Device Dependent Ports (DDP's).

CONTROL UNIT (CU1 AND CU2)

The Control Unit (CU1 and CU2) card assemblies can be divided into the following three functional areas: the Shift Amount Register (SAR) and associated control circuits, the Condition Register, and the Shift Control Register and associated control circuits. (See figure II-2.) Figures II-12 and II-15 are functional block diagrams of the shift control circuits and show the Barrel Switch interface and Condition Register and control circuits.

SHIFT AMOUNT REGISTER (SAR)

The functions of the SAR and its associated logic circuits are:

a. To load shift amounts from the Microprogram Memory (MPM) or Barrel Switch (BSW) into the SAR for use in shifting operations.

b. To generate the required controls for the Barrel Switch (BSW) to perform the shift operation indicated by the control from the Nanomemory (NM).

c. To generate the complement of the SAR contents, where the complement is defined as the count that will restore the bits of a word to their original position after an end-around shift of "N".

A literal is loaded into the Shift Amount Register (SAR) from the Microprogram Memory (MPM) by a Type II instruction. The transfer of MPM data to the Shift Amount Register signal (SAR-MPM) is generated by the Microprogram Address (MPAD) controls on MU2. This signal is true when a Microprogram Memory word contains a 1100 or 1101 in the first four bits of the word. Refer to the microcode information for a Type II instruction under microinstruction format heading presented under the Microprogram Memory (MPM) at the beginning of Section II in this manual. (See figure II-4.)

The bit generated from the Shift Control Register that corresponds to bit-49 of the nanocode (N49) will load the four least significant bits of the Barrel Switch (BSW), which is located on the Logic Unit (LU1) card assembly, into the Shift Amount Register (SAR).

The Shift Amount Register (SAR) and associated logic circuits control the Barrel Switch shift operations, the number of bits to be shifted, and allows either right or left end-off shifts or end-around shifts (circular shifts).

The shift control of the Barrel Switch is accomplished by the Shift Selection bits (N32 and N33) of the Nanomemory (NM). The following list provides the nanobit code for Shift Selection operation:

Nanobits 32	33	Shift Selection Operation	Word
0	0	No Shift	
0	1	Right shift end-off the entire Adder output, left fill with ZERO's.	R
1	0	Left shift end-off the entire Adder output, right fill with ZERO's.	L
1	1	Circular shift right the entire Adder output (shift end-around).	C

The shift amounts are stored in bits 1, 2, 3 and 6 of the SAR. The SAR gated output signals SAR1 and SAR2 control the first section (level of gating) of the Barrel Switch. Signals SAR3 and SAR6 control the second section of the Barrel Switch. Figure II-16 specifies the contents of the SAR (six bits) for shift amounts of 1 through 15 and their complements for a 16-bit configuration.

The no-shift operation requires the Barrel Switch to allow the Arithmetic/Logic Unit (Adder) output to gate through the Barrel

FROM CONTROL REGISTER ON MU2
FROM MPAD CONTROLS ON MU2
OUTPUT SELECT SIGNAL FROM CONTROL REGISTER ON EO
FROM BARREL SWITCH ON LU1
STROBE ENABLE SIGNALS FROM CONTROL REGISTER ON MU5*
FROM LITERAL REGISTER ON MU6-2*3
DN45
DN46
LIT(1-8)
BSW (8 LSB'S)
3CLK1'
SCLK2
(CG)
3CLKA
SOURCE SELECT GATES MU5
LIT(1-8)'
BSW (8 LSB'S)
(CG)
3CLKD
BSW (1-16)
DN41
MAR
8 7 6 5 4 3 2 1
MU5
MIR
MSB
LSB
LU6'S and LU7'S
OS(9-16)
MIR (1-16)
OS (1-16)
(OUTPUT DATA)
SCLKA FROM CD
MEMORY ENABLE PARITY BIT FROM LI8
35MC
DTM17
MEMORY ENABLE FROM CG
MEMORY CONTROL SIGNALS FROM EO
SHARED MEMORY CONTROL
"S" MEMORY
CGWRITE
WRITE
ERRST
DATACY
STSTEAL
MICRO-PROGRAM ADDR FROM MU6
INCR
MPM
DTMnn
ADRnn
SM 1 THRU 3
MEMORY
"M" MEMORY
OS5/
OS6/
OS(7-8)
DECODER PSI (SHT. 1)
(CD)
SCLKB
PORT SELECT (SHT. 1)
0 1 2 3
UNSELECTED
SELECTED
B51 THRU B53 FROM EO
READ (1-4)
WRITE (1-4)
DINT (1-4)
STINT (1-4)
I 1 AND 2
I 3 AND 4
GATE CIRCUITS PSI (SHT. 1)
GATE CIRCUITS PSI (SHT. 1)
GATE CIRCUITS PSI (SHT. 1)
IRQ
URQ
SRQ
DEVICE DEPENDENT PORT
PERIPHERAL DEVICE
3CLKA
BSW(8MSB'S)
DN43
DN44
OSEL
BRI
8 7 6 5 4 3 2 1
MU4 (SHT. 1)
BRI
8 7 6 5 4 3 2 1
MU4 (SHT. 1)
LIT(1-8) / BSW(8 LSB'S)
OUTPUT SELECT GATES MU4 (SHT. 1)
OS(1-8)
OS1
PRIORITY SELECTION PSI (SHT. 2)
PRIORITY ENABLE SIGNALS FROM OTHER PSI'S
ENAB(1-7)
PREWIRED ADDRESS INFORMATION
ASR
ADDRESS AND STATUS REQUEST SIGNALS FROM
GATE CIRCUITS PSI (SHT. 2)
ENST (1-4)
EXT (5-8)
INST
(DDP)
PORT SELECT OR (PSI)-1 FOR FIRST 4 DDP'S
(PSI-2 FOR SECOND 4 DDP'S)
(PS1-3 FOR THIRD 4 DDP'S)
TO COUNTER (CTR) ON MU5
INPUT REQUEST (IRQ), UNSOLICITED REQUEST (URQ), AND SOLICITED REQUEST (SRQ) FROM PSI
ENAB1
EXT (1-16) (INPUT DATA TO B-INPUT SELECT GATES ON LU3'S AND 4'S)

Fig. II-14 OUTPUT AREA, BLOCK DIAGRAM

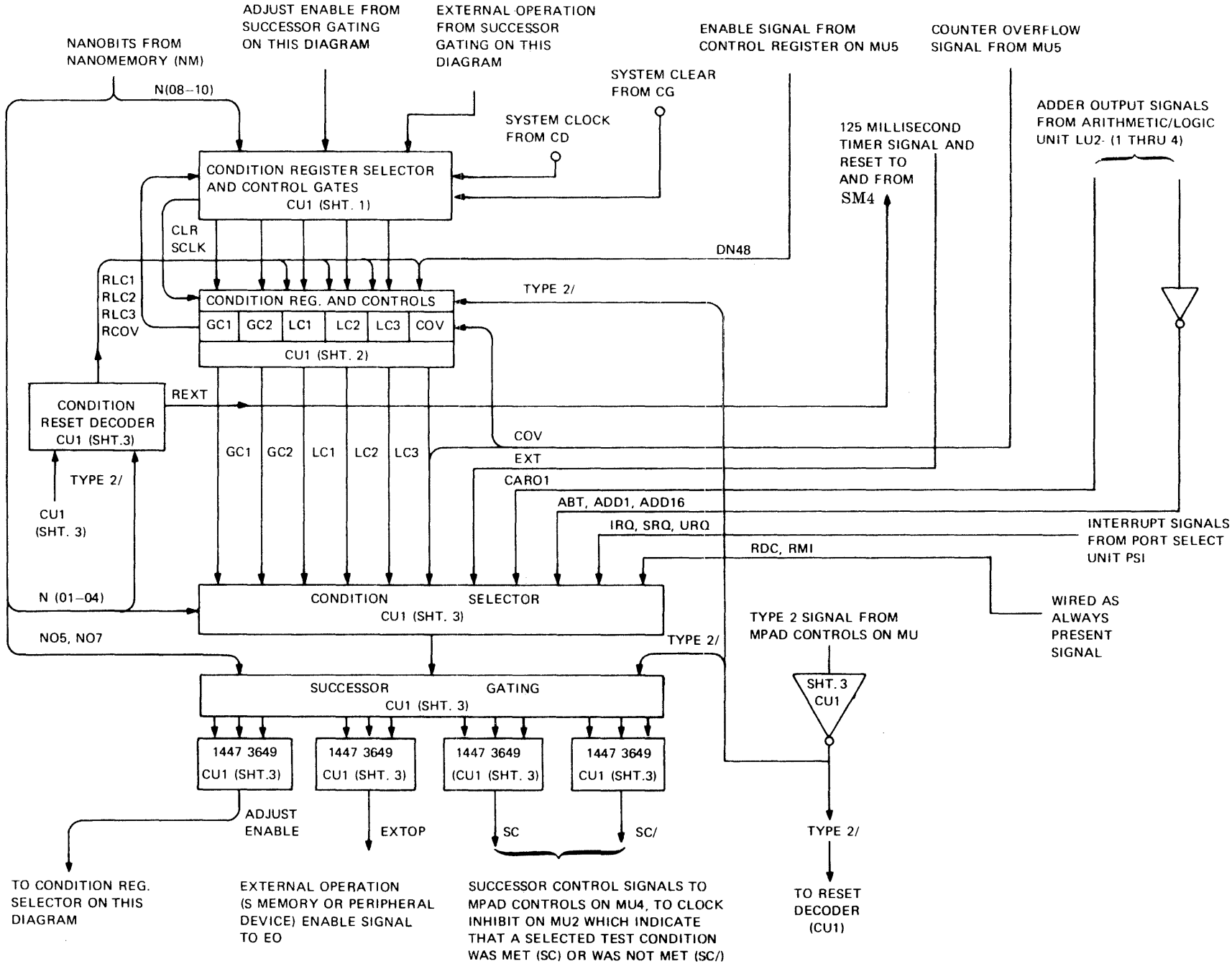


Fig. II-15 CONDITION REGISTER AND CONTROL CIRCUITS, BLOCK DIAGRAM

Logic Unit Width 16 Bits

Bin.	SAR						Complement					
	1	2	3	4	5	6						
0	0	0	0	X	X	0	0	0	0	X	X	0
1	0	0	0	X	X	1	1	1	1	X	X	1
2	0	0	1	X	X	0	1	1	1	X	X	0
3	0	0	1	X	X	1	1	1	0	X	X	1
4	0	1	0	X	X	0	1	1	0	X	X	0
5	0	1	0	X	X	1	1	0	1	X	X	1
6	0	1	1	X	X	0	1	0	1	X	X	0
7	0	1	1	X	X	1	1	0	0	X	X	1
8	1	0	0	X	X	0	1	0	0	X	X	0
9	1	0	0	X	X	1	0	1	1	X	X	1
10	1	0	1	X	X	0	0	1	1	X	X	0
11	1	0	1	X	X	1	0	1	0	X	X	1
12	1	1	0	X	X	0	0	1	0	X	X	0
13	1	1	0	X	X	1	0	0	1	X	X	1
14	1	1	1	X	X	0	0	0	1	X	X	0
15	1	1	1	X	X	1	0	0	0	X	X	1

Notes:
1. The complements shown are not true complements.
2. An X indicates unused bit position.
3. A Bin. indicates binary equivalent.

Fig. II-16 SHIFT AMOUNTS AND THEIR COMPLEMENTS

Switch with a ZERO shift. Because the SAR may have a shift amount stored in it, the shift amount lines to the Barrel Switch must be forced to ZERO. The enable control lines ER and EL may only be in the equivalent of right shift end-off state at this time.

The SAR control logic controls the Barrel Switch with two enable shift control signals (ER and EL) for end-off or end-around operations. The "enable for right shift" (ER signal) controls the data that is being shifted to the right. The "enable for left shift" (EL signal) controls data that is shifted end-around into the left side of the Barrel Switch. The truth table configuration for these two shift enable signals and the function performed is as follows:

ER	EL	Function
0	0	None (can not occur)
0	1	Enable shift left end-off
1	0	Enable shift right end-off
1	1	Enable shift right end-around (circular shift)

The following truth table configuration relates the SAR controls to the Barrel Switch (BSW) controls:

SAR Controls			BSW Controls		
N32	N33	SAR Contents	ER	EL	Shift Amount to BSW (SAR)
0	0	Non-zero	1	0	All zeros
0	1	Non-zero	1	0	Contents of SAR
1	0	Non-zero	0	1	Contents of SAR
1	1	Non-zero	1	1	Contents of SAR
0	0	All zeros	1	0	All zeros
0	1	All zeros	1	0	All zeros
1	0	All zeros	0	1	All zeros
1	1	All zeros	1	0	All zeros

SAR COMPLEMENTING

The bit generated from the shift control register that corresponds to bit 50 of the nanocode (N50) causes the complement of the SAR to be loaded into SAR. Figure II-16 indicates the required complement of the SAR contents for this 16-bit word system configuration. The shift amount is in binary form, while the complement (COMP) is the 2's complement of the shift amount. A ground is wired in the X position for the BSW and MPM inputs to the SAR input select circuits. Only SAR bits 1, 2, 3, and 6 are used at the BSW.

CONDITION REGISTER AND ASSOCIATED LOGIC CIRCUITS

The Condition Register and associated logic circuits of the Control Unit (CU) perform the following four major functions (figure II-15): 15.)

a. Selects one of 16 condition bits for use in performing conditional operations.

b. Stores six resettable Condition bits in the Condition register.

c. Decodes nanobits from nanomemory (NM) for resetting, setting, or to request the setting of certain bits in the Condition register.

CONDITION SELECTION (CONDSEL) AND CONDITION ADJUST (CNDADJ)

The 16 condition bits of the Condition Register and associated logic circuits act as error indicators, interrupts, status indicators, and lockout indicators Six of the condition bits are generated by the Condition Register; six condition bits are from external sources; the four remaining condition bits are generated during one clock time in the Logic Unit. Only one of these 16 conditions can be selected and tested in each microinstruction. Each condition can be tested for either a ZERO or a ONE and is used in conditional, memory conditional, Logic

Unit condition, and in Type I microinstructions which determine the Successor Control (SC) selection. The six condition bits generated by the Condition Register and control circuits are:

a. First Global Condition (GC1).
b. Second Global Condition (GC2).
c. First Local Condition (LC1).
d. Second Local Condition (LC2).
e. Third Local Condition (LC3).
f. Counter Overflow (COV).

The first Global Condition bit (GC1) is set and reset locally (refer to Condition Adjust) within the Processor as a programable flag bit. This Global Condition bit cannot be reset by testing. The functional description for the second Global Condition (GC2) is similar to that described for the first Global Condition (GC1).

The first Local Condition (LC1) bit is used for temporary storage of Boolean conditions (as a flag bit) within the Processor. This Local Condition bit is set locally (refer to Condition Adjust) by the Processor and reset locally by testing. The functional descriptions for the second and third Local Condition (LC2) and (LC3) are similar to that described for the first Local Condition (LC1).

The Counter Overflow (COV) condition bit is used to record overflows from the Counter Register (CTR), which is located in the Memory Control units (MU5 card). This condition bit is set whenever an increment counter operation (INC) causes the Counter (CTR) to overflow. This condition bit is reset when tested or whenever the Counter (CTR) is loaded from either the Barrel Switch (BSW) or the Literal register (LIT). When an attempt is made to set and reset this condition bit at the same clock time, the reset condition is dominant. Any test on this condition bit actually tests the "logical OR" of this bit with the true overflow from the Counter (CTR). The COV condition bit is set if the Counter (CTR) is all ONEs and an increment operation (INC) was in the Type I instruction whose phase 3 is currently being executed. Signal DN48 from the Control Register of the Memory Control units (MU5 card) is used as an input to a gate circuit, along with an output of the Condition Reset Decode module to reset the Counter Overflow flip-flop of the Condition Register. The DN48 signal corresponds to the complement of nanobit N48 from the Nanomemory.

The Condition Selection for Global Conditions, Local Conditions, and Counter Overflow are controlled by nanobits N01 through N04 from the Nanomemory which specify the particular condition to be tested. Table II-10 lists the nanobit code for Condition Selection of the aforementioned conditions, how the condition is set and reset, and the resultant condition (dominant function) when the condition bit is both set and tested in the same microinstruction.

TABLE II-10 CONDITION SELECTION NANOBIT CODES

Nanobits					How Condition		Dominant
1	2	3	4	Condition Selection	Set or	Reset	Function
0	0	0	0	Global Condition 1 (GC1)	CAJ	CAJ	Set
0	0	0	1	Global Condition 2 (GC2)	CAJ	CAJ	Set
0	0	1	0	Local Condition 1 (LC1)	CAJ	Test	Set
0	0	1	1	Local Condition 2 (LC2)	CAJ	Test	Set
1	1	0	1	Local Condition 3 (LC3)	CAJ	Test	Set
1	0	0	0	Counter Overflow (COV)	Overflow Test		Reset

CAJ CONDITION ADJUST NANOBITS N08 THROUGH N10

The Condition Adjust provides the means by which condition bits can be set to ONE or reset to ZERO. The Condition Adjust operation takes place at the end of a phase 1 operation. The Condition Adjust for the Global Conditions and Local Conditions are controlled by nanobits N08, N09, and N10 from the Nanomemory which specify the particular condition to be adjusted. The following list provides the nanobit code for Condition Adjust of the aforementioned conditions:

Nanobits			Condition Adjust Function
8	9	10	
0	0	0	No change
0	1	1	Set Local Condition 2 (LC2)
0	1	0	Set Global Condition 2 (GC2)
0	1	0	Set Global Condition 2 (GC2)
0	1	1	Reset Global Condition 2 (GC2)
1	0	0	Set Local Condition 3 (LC3)
1	0	1	Reset Global Condition 1 (GC1)
1	1	0	Set Global Condition 1 (GC1)
1	1	1	Set Local Condition 1 (LC1)

The six condition bits generated by external sources are as follows:

a. Ready Memory Information Register (RMI).
b. Read Complete (RDC).
c. Input Request (IRQ).
d. External Interrupt (EXT).
e. Solicited Request (SRQ).
f. Unsolicited Request (URQ).

The Ready Memory Information Register (RMI signal) is always present (low active) on the Condition Select module of the Condition Control circuits on the CU1 card.

The Read Complete (RDC signal) is always present (low active) on the Condition Select module of the Condition Control circuits on the CU1 card. Because the Read Cycle is completed in one clock time, RDC always indicates that data is available from a memory read operation (MR1 or MR2).

The Input Request (IRQ signal) from the Port Selector is sent to the Condition Select module of the Condition Control circuits on the CU1 card. When this signal is at a low level, it indicates that a DDP which is unselected and is generating a Status Interrupt (STINT) or Data Interrupt (DINT).

The External Interrupt (EXT signal) from the Shared Memory Control (SM4) is sent to the Condition Select module of the Condition Control circuits on the CU1 card. When this signal is at a low level, it indicates that a real time Counter is set equal to 125 milliseconds.

The Solicited Request (SRQ signal) from the Port Selector is also sent to the Condition Select module. When this signal is at a low level, it indicates that the selected DDP is generating a Data Interrupt (DINT).

The Unsolicited Request (URQ signal) from the Port Selector is also sent to the Condition Select module. When this signal is at a low level, it indicates that the selected DDP is generating a Status Interrupt (STINT).

The Condition Selection for the Ready MIR (RMI), the Read Complete (RDC), Input Request (IRQ), External Interrupt (EXT), Solicited Request (SRQ), and Unsolicited Request (URQ) conditions are controlled by nanobits N01 through N04 from the Nanomemory. The following list provides the nanobit code for Condition Selection of the aforementioned conditions:

Nanobits				Condition Selection
1	2	3	4	
1	0	0	1	Ready Memory Information Register (MIR)
1	0	1	0	Read Complete (RDC)
1	0	1	1	Input Request (IRQ)
1	1	0	0	External Interrupt (EXT)
1	1	1	0	Solicited Request (SRQ)
1	1	1	1	Unsolicited Request (URQ)

Nanobit signals N01 through N04 are applied to the Condition Reset Decode module and the Condition Selection module of the Condition Control circuits to control Condition Selection for the six condition bits generated by external sources.

In addition to the aforementioned 12 condition bits, the four remaining condition bits are generated in the Logic Unit (LU2 (1 through 4) cards) during a phase 3 operation for the previous fetched instruction. The four condition bits from the Logic Unit are as follows:

a. Most Significant Bit Condition (MST).
b. Least Significant Bit Condition (LST).
c. Adder Bits True (ABT).
d. Adder Overflow (AOV).

The Most Significant Bit condition (MST) is the state of the most significant bit (MSB) of the Adder output during the preceding microinstruction (independent of shifting). The MSB from the Adder, signal ADD1, is sent to the Condition Selection module of the Condition Control circuits from the MSB Arithmetic/Logic Unit (ALU) module on the LU2-1 card.

The Least Significant Bit condition (LST) is the state of the least significant bit (LSB) of the Adder output during the preceding microinstruction (independent of shifting). The LSB from the Adder, signal ADD16, is sent to the Condition Selection module of the Condition Control circuits from the LSB Arithmetic/ Logic Unit (ALU) module on the LU2-4 card.

The Adder Bits True (ABT signal) from the Arithmetic/Logic Unit (ALU) modules on the LU2 (1 through 4) cards are sent to the Condition Selection module of the Condition Control circuits. This signal indicates that all of the outputs from the Adder are all ONES during the phase 3 portion of the previously fetched microinstruction. Adder output signals ADD1 through ADD16 are all ONES for this condition.

The Adder Overflow (CAR1 signal), which is the most significant bit carry from the LU2-1 card, is sent to the Condition Select module of the Condition Control circuits. The AOV condition is the carry out of the most significant bit of the Adder for the previous fetches microinstruction.

The Condition Selection for the Most Significant Bit condition (MST), the Least Significant Bit condition (LST), Adder Bits True (ABT), and Adder Overflow (AOV) conditions are controlled by nanobits N01 through N04 from the Nanomemory which specify the particular condition to be tested.The following list provides the nanobit code for Condition Selection of the aforementioned conditions:

Nanobits 1 2 3 4	Condition Selection	Reset by Testing
0 1 0 0	Most Significant Bit of Adder (MST)	No
0 1 0 1	Least Significant Bit of Adder (LST)	No
0 1 1 0	Adder Bits True (ABT)	No
0 1 1 1	Adder overflow (AOV)	No

SUCCESSOR CONTROLS (SC)

The output signals from the Successor Control (SC) modules of the Condition Control circuits are used in the conditional part of the microinstruction to specify the true-successor (SC = 1) and the false successor (SC = 0). Successor Control signal SC=1 indicates that the selected condition test was met. Successor Control signal SC = 0 indicates that the selected condition test was not met. The Successor Control output signals are sent to the Microprogram Address (MPAD) controls of the Memory Control Units (MU4 card).

Nanobit Signal N05, the Condition tested equal true or false (true), and the Type 2 Signal inverted are applied through successor gating to the 1447 3649 gating modules producing logical outputs as illustrated in the following equations:

SC = (N05/+CONDITION)·(N05+CONDITION/) ·TYPE 2/

SC/ = (N05/+CONDITION/)·(N05+CONDITION) TYPE2/

Nanobit Signal N05 selects the level of the Condition being tested for successor determination. When the Condition level being tested is met (true), the true-successor is performed (SC = 1). A true-successor must appear with each condition test. If the condition being tested is not met, or is false, the false-successor is performed (SC=0). A false-successor is optional; therefore, if a false-successor is not present, the successor STEP causes the next microinstruction in sequence to be performed. The successor STEP is performed by the Microprogram Address (MAPD) controls by incrementing the contents of the Microprogram Count Register (MPCR) by +1. The Successor Condition (SC) is controlled by nanobit N05, as follows:

Nanobit 5	Function Specified
0	Test for false state of specified condition. SC=1 if Condition Selection (CONDSEL) is false.
1	Test for true state of specified condition. SC=1 if Condition Selection (CONDSEL) is true.

Refer to the Microprogram Address (MPAD) controls information contained under the Memory Control unit heading for information concerning the Successor Controls (SC) and their operation with the MPAD controls.

EXTERNAL OPERATION

Nanobit signal N07 from nanomemory is applied along with nanobit N05, the Condition being tested equal to true or false (true/), and the type 2 signal inverted, to gating modules 1447 3649. The following output signals as illustrated in the following logical equations:

EXTOP = (N05+CONDITION/+N07/) · (N05+CONDITION+N07/) · TYPE2/

ADJUST ENABLE = (N05 · CONDITION · TYPE2/) +(N05/ · CONDITION/ · TYPE2/)+(N07/ · TYPE2/)

The external operation (EXTOP) output signal is sent to the external operations Control (EO card) to Control Memory or Device functions, and the adjust enable signal along with (EXTOP) signal is sent to the Condition Adjust Circuits to control the setting and resetting of global and local flags.

The (EXTOP) signal used to enable Memory or Device operations (MDOP) and the Condition Adjust (CNDADJ) enable signal is controlled by nanobit N07 as follows:

Nanobit	Function Specified
7	
0	Perform Memory or Device Operation (MDOP) and Condition Adjust (CNDADJ) unconditionally.
1	Perform MDOP and CNDADJ if Successor Control SC=1.

SHIFT CONTROL REGISTER AND ASSOCIATED CIRCUITS

The Shift Control Register (SCR) and associated circuits are comprised of six flip-flops and associated input and output gate circuits. (See figure II-12.) The following six flip-flops in SCR are used for Shift Amount Register (SAR) input selection, shift selection, and Save Sign Control (SSC):

a. BSW → SAR.
b. Complement SAR.
c. Shift Right.
d. Shift Left.
e. Saved Sign Bit.
f. Save Sign Control.

SHIFT AMOUNT REGISTER (SAR) INPUT SELECTION

The SAR input selection is controlled by the Shift Control Register flip-flops: BSW SAR and Complement SAR. The input to these flip-flops is controlled by nanobit N49 or N50 from the Nanomemory. The gated output signal from the BSW SAR flip-flop is sent to the SAR input select circuits of the CU2 card (sheet 2 of logic diagram). The gated output signal from the Complement SAR flip-flop is also sent to the SAR input select circuits.

When the Shift Amount Register is used (SAR BSW), the four least significant bits of the Barrel Switch (BSW) are placed in the SAR (1, 2, 3, and 6) flip-flops. When the complement SAR (CSAR) is used, the contents of the SAR are complemented. (See figure II-16.)

Nanobit signals N49 and N50 from the Nanomemory control Shift Amount Register (SAR) input selection as follows:

Nanobits		Function Specified
49	50	
0	0	No change in the contents of the Shift amount register (SAR).
0	1	The contents of the SAR are complemented (CSAR).
1	0	The four least significant bits of the Barrel Switch (BSW) are placed in the SAR.

SHIFT SELECTION

The Shift selection is controlled by the Shift Control Register flip-flops: Shift Right and Shift Left. The input to these flip-flops are controlled by nanobits N33 and N32. The gated output signals (ER and EL) are sent to the Barrel Switch Shift Control circuits of the Logic Unit (LU1 and LU5 cards). The gated output signal (SHEN) is sent to the SAR output select circuits for shift control. This signal indicates that a shift condition is present.

The Enabled Left shift control signal (EL) is sent directly to the BSW shift control circuits. This signal is true (high level), for both left shift and right end-around (circular) shifts. The Enable Right shift control signal (ER) is also sent to the BSW shift control circuits. This signal should be true (high level) at all times except during a shift-left operation. The number of bits to be shifted is obtained from the current contents of the Shift Amount Register (SAR) at the start of the phase 3 clock which executes the shift instruction. Nanobits N32 and N33 are used to generate the Shift Selection control signals as indicated in the following list:

Nanobits		Function Specified
32	33	
0	0	No shift.
0	1	Indicates a right end-off shift with left ZERO fill. The shift amount is contained in the SAR, which determines the actual number of bits to be shifted to the right.
1	0	Indicates a left end-off shift with right ZERO fill. The shift amount is located in the SAR, which contains the complement of the number of bits to be shifted to the left. The complement must be stored in the SAR because all shifting operations are performed as right shifts.
1	1	Indicates a right end-around (circular) shift. The shift amount is located in the SAR, which contains the actual number of bits to be shifted right.

Refer to the shift operations description under the Shift Amount Register (SAR) heading for additional information.

SAVE SIGN CONTROL

The Save Sign Control (SSC) is comprised of

the Shift Control Register flip-flops: Save Sign Control and Saved Sign Bit. The Save Sign Control flip-flop input is provided by nanobit N55 from the Nanomemory. The Input Data (D) to the Saved Sign bit flip-flop is provided by an AND-OR-INVERT gate circuit. The inputs to this gate circuit are as follows: the LSB output of the Adder (ADD16/) and the set output of the Shift Right flip-flop; the MSB output of the Adder (ADD1/) and the set output of the Shift Left flip-flop; and the MSB carry signal (CAR1/), SHEN/, and the set output of the Save Sign Control flip-flop.

When the Save Sign Control (SSC) operation is used and a shift operation is being executed, the first end bit shifted is stored in the Saved Sign Bit flip-flop (SSF). Either the MSB or LSB of the Adder output (ADDnn01/ or 16/) is stored in the SSF, depending upon the shift operation (shift right or shift left). Simultaneously, the prior bit stored in the Saved Sign Bit flip-flop (SSF) is placed in the opposite end bit position of the Barrel Switch (BSW) by either signal MSBCL or LSBCL. Signal LSBCL is used to force a bit into the least significant bit (LSB) of the Barrel Switch second-level gating in the Logic Unit (LU1 card). Signal MSBCL is used to force a bit into the most significant bit (MSB) of the Barrel Switch second level of gating in the Logic Unit (LU1 card).

When the Save Sign Control (SSC) is not used and a shift operation (shift right on shift left) is executed, the first end-off bit is stored in the Saved Sign Bit flip-flop (SSF) and no other action occurs. (Signal's LSBCL or MSBCL are not generated.) When SSC is used and shift operation is not being executed (signal SHEN is not present), the Adder Overflow (AOV), which is indicated by the presence of the CAR1 signal, is stored in Saved Sign Bit flip-flop (SSF). No other action occurs at this time. When SSC is not used and a shift operation is not being executed, no change takes place in the Save Sign Control circuits.

Nanobit N55 is used to determine the Shift Store Control function as follows:

Nanobit 55	With Shift Operation (SHIFT)	Without Shift Operation (SHIFT)
0	First end-off bit of Adder to Saved Sign Bit flip-flop (SSF).	No change SHIFT: NO ACTION
1	SHIFT:END-OFF ` SSF First end-off bits of the Adder to the Saved Sign Bit flip-flop prior (SSF) to the opposite end-bit of the Barrel Switch (BSW).	Adder Overflow (AOV) to the Saved Sign Bit flip-flop (SSF) SHIFT:AOV

SHIFT:END-OFF ⟶ SSF → BSW

MEMORY CONTROL UNIT (MU1 THROUGH MU5)

The Memory Control Units (MU-6 through MU6-4, MU2, MU3, MU4 and MU5 card assemblies) can be divided into the following four functional areas: the Microprogram Address (MPAD) section, the Memory/Device address section, the Z Register section and the Control Register section. Figure II-17 is a functional block diagram of the Microprogram Count Register (MPCR), Alternate Microprogram Count Register (AMPCR), Incrementer (INCR), and the Microprogram Address (MPAD) controls. The Base Registers (BR1 and BR2), the Memory Address Register (MAR), and associated output select gates are shown in figure II-14. Figure II-18 is a functional block diagram of the Literal Register (LIT) and Counter Register (CTR) with associated input/output circuits.

MICROPROGRAM ADDRESS (MPAD) SECTION

The Microprogram Address (MPAD) section consists of the Microprogram Count Register (MPCR), Alternate Microprogram Count Register (AMPCR), the Incrementer (INCR), the MPAD Controls, and the associated control logic circuits. The MPAD section is used to address the Microprogram Memory (MPM) and is also used to determine the sequencing of microinstructions in the MPM. The MPAD controls of the MPAD section are used to control the loading of the MPCR, loading of the AMPCR, selection of inputs to the Incrementer (INCR), loading of the Shift Amount Register (SAR) from the MPM, loading of the Literal register (LIT) from the MPM, and the generation of the Type II instruction control signal and the phase 3 clock inhibit signal.

MICROPROGRAM COUNT REGISTER (MPCR)

The Microprogram Count Register (MPCR) is a 14-bit register which contains the microinstruction address counter for the Microprogram Memory (MPM). The MPCR contains the current microinstruction address for all suc-

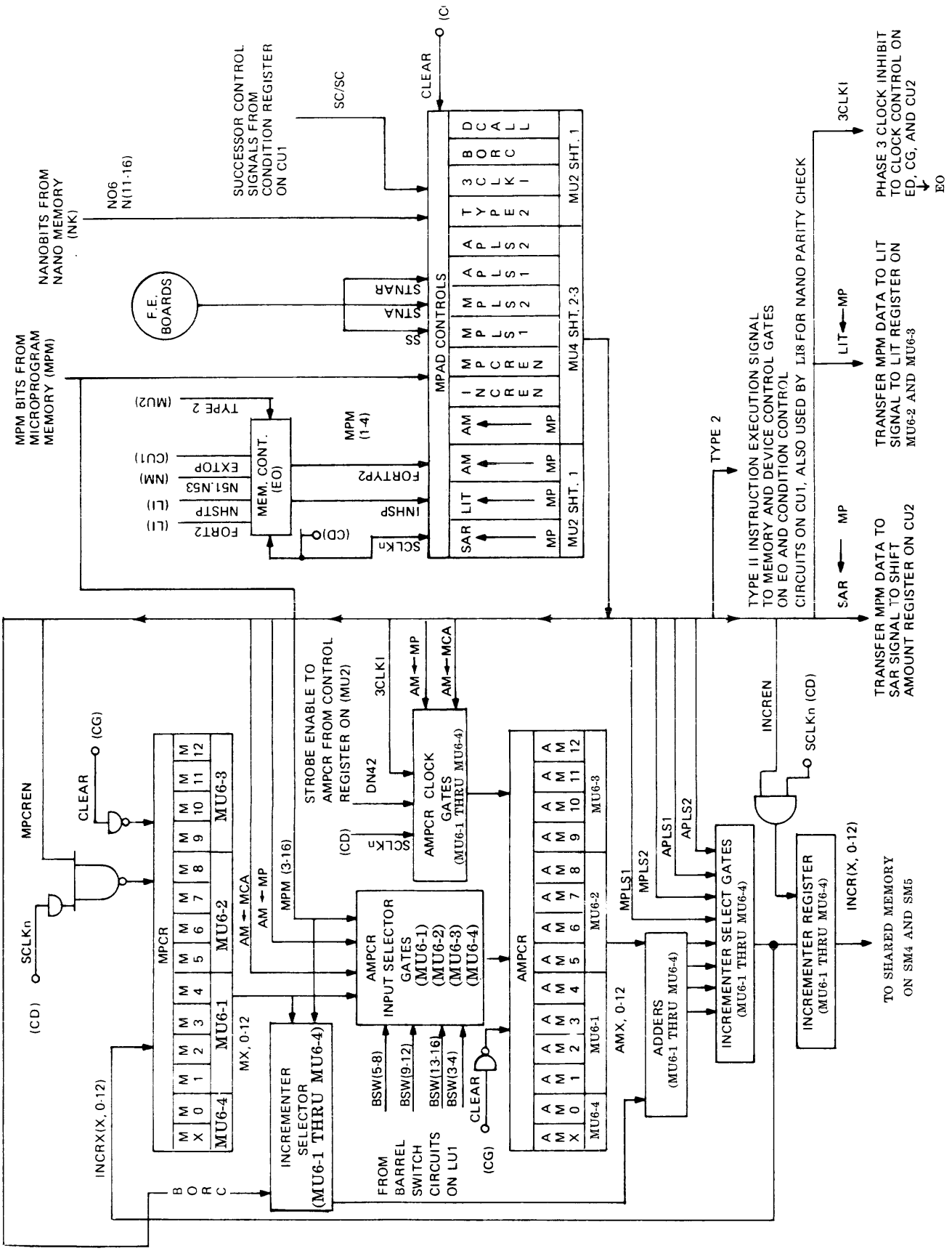


Fig. II-17 MICROPROGRAM COUNT REGISTER (MPCR), ALTERNATE MICROPROGRAM COUNT REGISTER (AMPCR), AND MICROPROGRAM ADDRESS (MPAD) CONTROLS, FUNCTIONAL BLOCK DIAGRAM

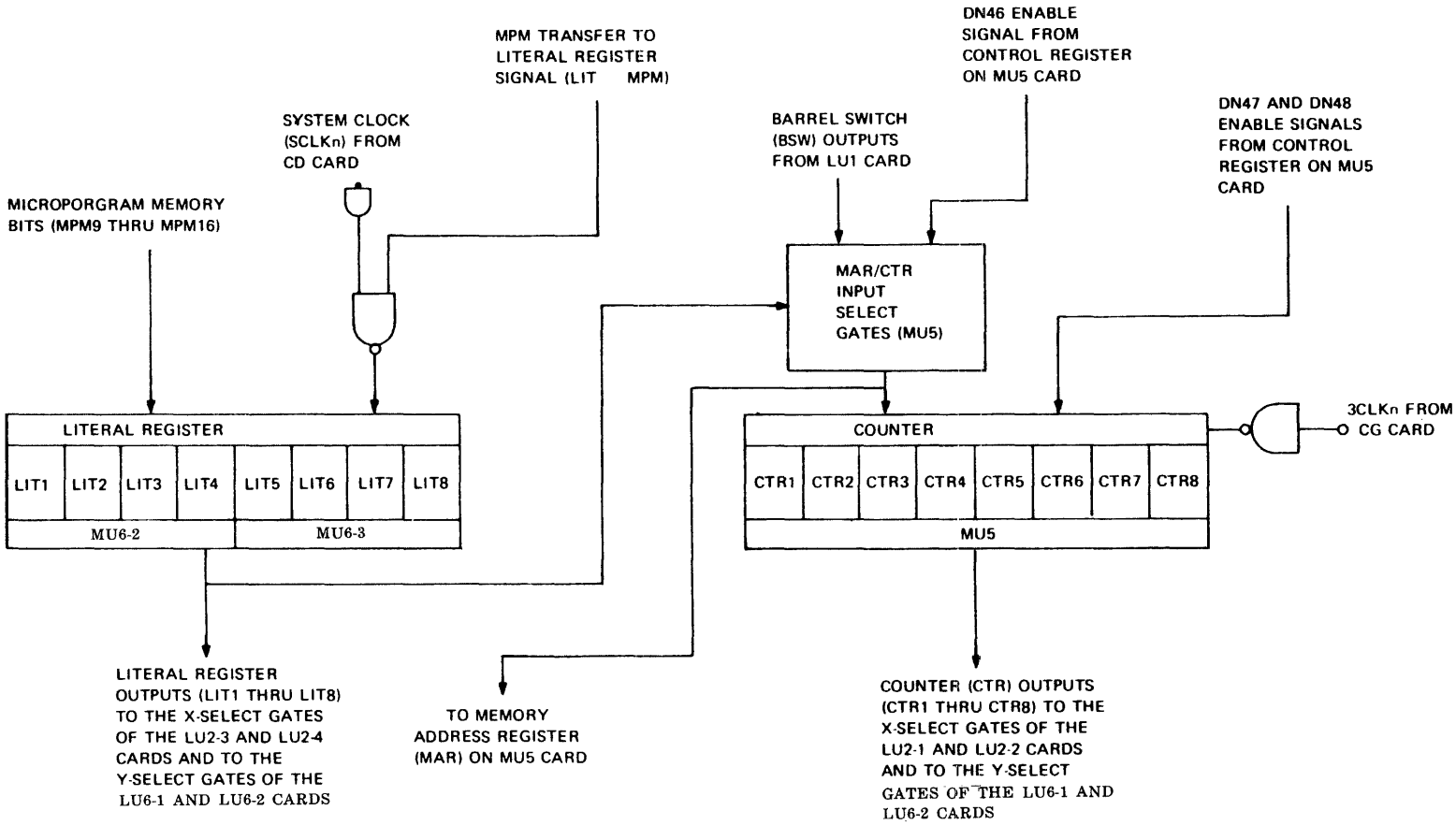


Fig. II-18 COUNTER (CTR) AND LITERAL REGISTER (LIT), FUNCTIONAL BLOCK DIAGRAM

cessors with the exception of Execute (EXEC). For further information concerning successors, refer to the information provided under the MPAD Control heading. The MPCR is located on Memory Control unit card assemblies MU6-1, MU6-2, MU6-3 and MU6-4. The MPCR can only be loaded with output data from the Incrementer Register select gates, which are also located on the MU6-1 through MU6-4 cards. The output signals from the Incrementer select gates (INCRXX through INCRX120) are applied directly to the D input of the respective MPCR flip-flop (MX through M12). The clock inputs (C) to the MPCR flip-flops are provided by NAND gates with the following inputs: SCLKn and MPCREN. Signal SCLKn is the system clock from the clock circuits of the Clock Generator (CG card). Signal MPCREN is the input enable signal from the MPAD Controls in the MU4 card. This signal enables the strobe to the MPCR which causes the MPCR to store Incrementer Register select gate outputs (INCRXX through INCRX12).

The outputs from the MPCR flip-flops (MX through M12- - - -1 signals) are sent to the incrementer Selector and are also used as inputs to the AMPCR input selector gates. MPCR output signals MX and M0 are generated on the MU6-4 card, signals M1 through M4 are generated on the MU6-1 card, signals M5 through M8 are generated on the MU6-2 card, and signals M9 through M12 are generated on the MU6-3 card. These MPCR output signals are sent to the respective Incrementer input gate circuits (M+1 Adder or M+2 Adder) and AMPCR input select gates.

ALTERNATE MICROPROGRAM COUNT REGISTER (AMPCR)

The Alternate Microprogram Count Register (AMPCR) is a 14-bit register which contains the jump, execute, call, and return addresses for instruction flow of routines and subroutines within the microprogram that is contained in the Microprogram Memory (MPM). For further information concerning the successors JUMP, CALL, EXEC (execute), and RETN (return), refer to the information provided under the MPAD Controls heading. The address contained in the Alternate Microprogram Count Register (AMPCR) is usually one less than the address to be jumped to or returned to. The (AMPCR) may also be used as temporary storage of microprogram data.

The AMPCR is located on Memory Control Unit card assemblies MU6-1 through MU6-4. The AMPCR can be loaded from the MPCR, the Microprogram Memory (MPM), or the 14 least significant bits (LSB's) from the Barrel Switch (BSW) which is located in the Logic Unit (LU1 card).

The inputs to the AMPCR input select gates are the MX through M12 signals from the MPCR, the MPM3 through MPM16 signals from the Microprogram Memory, and the 14 LSB's of the Barrel Switch signals BSW03 through BSWnn16. Input transfer signals AM◄–MCA and AM◄–MP are sent to the AMPCR input select gates and also to the AMPCR clock gates from the MPAD Controls of the MU2 and MU4 cards. Signal AM◄–MCA is used to transfer MPCR outputs (MX through M12 signals) to the AMPCR. Signal AM◄–MP is used to transfer Microprogram Memory outputs (MPM3 through MPM16 signals) to the AMPCR.

The data inputs (D) of the AMPCR flip-flops (AMX through AM12) are provided from the respective outputs from the AMPCR input select gates. The clock inputs (C) to the AMX through AM12 flip-flops are provided by the outputs from AND-OR-INVERT gates with the following inputs: SCLKn, AM◄–MCA, AM◄–MP, phase 3 clock inhibit (3CLKI), and DN42. DN42 is from the Control Register on the MU2 card and corresponds to nanobit N42 from the Nanomemory. Signal SCLKn is the system clock signal which is present for phase 1 and phase 3 operations from the clock circuits on the CD card. The phase 3 operation is inhibited during Type II microinstructions or when conditional Logic Unit microinstructions are not executed (condition not met). Signal 3CLKI is the phase 3 clock inhibit signal from the MPAD Controls on the MU2 card. When this signal is at a low level, the generation of phase 3 clock operation is inhibited.

Nanobit N42, which corresponds to the DN42 output signal from the Control Register on the MU2 card, is used to control AMPCR input selection as follows:

Nanobit 42	AMPCR Function Specified
0	No change in AMPCR.
1	Twelve least significant bits (14 LSB's) of the Barrel Switch (BSW) entered in the AMPCR.

When loading data into the AMPCR from the Barrel Switch (BSW), the storing of BSW data does not take place until the phase 3 clock time of the next Type I microinstruction occurs. However, the storing of the AMPCR contents into a register does occur within the clock time for that microinstruction in which the transfer appears.

INCREMENTER REGISTER (INCR)

The Incrementer Register (INCR) and associated gate circuits add a one (+1) or two (+2) to the selected Incrementer Register input from the Microprogram Count Register (MPCR), the Type II Branch or Call Microcode (MPM) or Alternate Microprogram Count Register (AMPCR). The Incrementer Register (INCR) is a 14-bit register which increments by a +1 or +2 the address of the next microinstruction to be executed by the microprogram.

The Incrementer Adder gates (M+1, M+2, AM+1, and AM+2 Adder gate circuits) receive input signals MX through M12 or MPM3 through MPM16 from the MPCR/MPM Selector and input signals AMX through AM12 from the AMPCR. The outputs from these gates are sent to the respective Incrementer register AND-OR-INVERT input gate, along with one of the following control signals from the MPAD controls of the MU4 card: MPLUS1, MPLUS2, APLUS1, and APLUS2. Signal MPLUS1 and MPLUS2 are the transfer MPCR signal to the Incrementer Register select gates. When these signals are at a high level, they indicate an increment of +1 or +2 for the respective signal. Signals APLUS1 and APLUS2 are the transfer AMPCR signals to the Incrementer Register input gates. When these signals are at a high level, they indicate an increment of +1 or +2 for the respective signal.

The clock input (C) of the Incrementer Register is provided by two-input AND gates with the following input signals: INCREN and SCLKn. Signal INCREN is the Incrementer Register (INCR) input enable signal from the MPAD Controls of the MU4 card. This signal enables the strobe to the INCR which causes the INCR to store the Incrementer Register select gate outputs (INCRXX through INCRX12). Signal SCLKn is the system clock from the clock circuits on the CD card.

The carry output signals from the M+1 and M+2 Adder gate circuits of the Incrementer are as follows: M1C1, M2C1, M1C5, M2C5, M1C9, and M2C9. These signals are the MPCR+1 and MPCR+2 Adder carries from the respective bit-1, bit-5 or bit-9 of the M+1 and M+2 Adder gate circuits (C4 output of Adder circuit). These carry signals are used as inputs to the CO inputs of the other M+1 and M+2 Adder gate circuits of the Incrementer.

The carry output signals from the AM+1 and AM+2 Adder gate circuits of the Incrementer are as follows: A1C1, A2C1, A2C5, A2C5, A1C9, and A2C9. These signals are the AMPCR+1 and AMPCR+2 Adder carries from the respective bit-1, bit-5, or bit-9 of the AM+1 and AM+2 Adder gate circuits (C4 output of Adder circuit). These carry signals are used as inputs to the CO inputs of the other AM+1 and AM+2 Adder gate circuits of the Incrementer.

The output signals (INCRX through INCR12) from the Incrementer Register (INCR) are used to provide address information in the Microprogram Memory. The output signals (INCRXX through INCRX12) from the Incrementer Register select gates are sent to the respective data (D) input of the Microprogram Count Register (MPCR) flip-flops.

MICROPROGRAM ADDRESS CONTROL (MPAD CNTL)

The Microprogram Address Control (MPAD CNTL) circuits are used to control the loading of the Microprogram Count Register (MPCR), the loading of the Alternate Microprogram Count Registers (AMPCR) from the MPCR, the selection of inputs used by the incrementer circuits, and also the selection of values to be used by the incrementer circuits. The MPAD Controls are also used to decode control bits from the Microprogram Memory to determine the destination for the contents of the remaining bits in an MPM word.

The MPAD Controls are used to generate the controls that are required to perform the following eight Successor commands: WAIT, STEP, SAVE, SKIP, JUMP, EXEC, CALL, and RETN. The controls for these eight Successor functions (commands) are determined by nanobit inputs from the Nanomemory and the true or false Successor Control (SC) functions from the Successor Control circuits of the Control (SC) functions from the Successor Control circuits of the Control Unit (CU1 card). The MPAD Controls are also used to decode control bits from the Microprogram Memory (MPM) to determine the destinations for the contents of the remaining bits in an MPM word for Type

I and Type II microinstructions.

The MPAD Control gate circuits that are used to generate the control signals for loading the MPCR (MPCREN signal), loading the AMPCR from the MPCR (AM←MCA signal), and selecting the inputs to the Incrementer Register are controlled by nanobits N11 through N16 and Successor Control signals SC/ and SC. The following list provides the nanobit code, the Successor Control (SC) used, and the Successor function for the MPAD Controls:

Nanobits Used If SC=1:				Nanobits Used If SC=0:		
11	12	13	Successor Function	14	15	16
0	0	0	WAIT	0	0	0
0	0	1	STEP	0	0	1
0	1	0	SAVE	0	1	0
0	1	1	SKIP	0	1	1
1	0	0	EXEC	1	0	0
1	0	1	JUMP	1	0	1
1	1	0	CALL	1	1	0
1	1	1	RETN	1	1	1

Table II-11 lists the contents of the Microprogram Count Register (MPCR), Alternate Microprogram Count Register (AMPCR), and Incrementer Register (INCR) when the specified Sucessor function is executed. The logic level of the various control signals is also specified to perform the applicable Successor function and Type II microinstruction.

NOTE

An A in any column of Table II-11 indicates the AMPCR. An I indicates the INCR, an M indicates the MPCR, and an X indicates an insignificant condition.

The following logical equations are given for the MPAD Control functions listed in Table II-11:

$$MPLUS1 = [SC \cdot \overline{N11}(N12 + \overline{N13})] + [\overline{SC} \cdot \overline{N14}(\overline{N15} + \overline{N16})] + TYPE2$$

$$APLUS1 = [SC \cdot N11(\overline{N12} + \overline{N13})] + [\overline{SC} \cdot N14(\overline{N15} + \overline{N16})]$$

$$MPLUS2 = [(SC \cdot \overline{N11} \cdot N12 \cdot N13)] + [(\overline{SC} \cdot \overline{N14} \cdot N15 \cdot N16)]$$

$$APLUS2 = [(SC \cdot N11 \cdot N12 \cdot N13)] + [(\overline{SC} \cdot N14 \cdot N15 \cdot N16)]$$

$$MPCREN = [SC \cdot (N12 + N13)] + [\overline{SC} \cdot (N15 + N16) \cdot \overline{INHSP}]$$

$$INCREN = [SC \cdot (N11 + N12 + N13) + \overline{SC} \cdot (N14 + N15 + N16)] \cdot \overline{INHSP}$$

$$AM \leftarrow MCA = [SC \cdot (N12 \cdot \overline{N13})] + [\overline{SC} \cdot (N15 \cdot \overline{N16})] + DCALL$$

$$3CLKI = TYPE2 + \overline{SC} \cdot N06$$

$$TYPE2 = \overline{MPM1} + \overline{MPM2} + \overline{MPM3} + \overline{MPM4} + FORT2 + SS + FORCE\ STEP + CLEAR$$

$$BORC = DCALL + (\overline{MPM1} \cdot MPM2 \cdot \overline{FORT2} \cdot \overline{SS})$$

$$DCALL = MPM1 \cdot \overline{MPM2} \cdot \overline{FORT2} \cdot \overline{SS}$$

MPLUS1 and MPLUS2 are the transfer MPCR to the Incrementer signals. When at a high level, the MPLUS1 or MPLUS2 signal indicates an increment of +1 or +2. APLUS1 and APLUS2 transfer the AMPCR to the Incrementer. When at a high level, the APLUS1 or APLUS2 signals indicate an increment of +1 or +2. MPCREN is the MPCR strobe enable signal, which causes the MPCR to store Incrementer outputs. INCREN is the Incre-

TABLE II-11 MPAD CONTROL FUNCTIONS

Successor Function and Type II Microinstruction

Register or Control	WAIT	STEP	SAVE	SKIP	JUMP	EXEC	CALL	RETN	Type II STEP	Type II JUMP	Type II CALL
MPCR	M	M+1	M+1	M+2	A+1	M	A+1	A+2	M+1	MPM+1	MPM+1
AMPCR	A	A	M	A	A	A	M	A	A	A	M
INCR	I	M+1	M+1	M+2	A+1	A+1	A+1	A+2	M+1	MPM+1	MPM+1
MPLUS1	X	I	I	0	0	0	0	0	1	1	1
APLUS1	X	0	0	0	1	1	1	0	0	0	0
MPLUS2	X	0	0	1	0	0	0	0	0	0	0
APLUS2	X	0	0	0	0	0	0	1	0	0	0
MPCREN	0	1	1	1	1	0	1	1	1	1	1
INCREN	0	1	1	1	1	1	1	1	1	1	1
AM MC	0	0	1	0	0	1	0	0	0	0	1
3SLKI	0	0	0	0	0	0	0	0	1	1	1
BORC	0	0	0	0	0	0	0	0	0	1	1
DCALL	0	0	0	0	0	0	0	0	0	0	1

menter Register (INCR) strobe enable signal, which causes the INCR to store inputs from the Incrementer select gates. The input INHSP inhibit step holds the Incrementer Register and MPCR during DPM operations, during MEM Read cycles for Load (LI) operations and during steal cycle. AM←MCA is the transfer MPCR-to-AMPCR signal, which is sent to the AMPCR Input Select gates and clock gates inputs of the AMPCR. 3CLKI is the phase 3 clock inhibit signal which is used by the external memory control on the EO card, the phase 3 clock circuits on the CG card, the AMPCR clock gates on the MU6-1 through MU6-4 cards, and the Shift Control Register input circuits of the CU2 card. The inputs used to generate the phase 3 clock inhibit signal (3CLKI) are: SC.N06 + TYPE2. Signal SC/ is the Successor Control signal from the Condition Control circuits on the CU1 card which indicates that the selected condition was not met. Signal N06 is a nanobit from the Nanomemory that is used to control the phase 3 modifier functions as follows:

Nanobit N06	Phase 3 Modified (Logic Unit Condition)
0	Perform phase 3 operations unconditionally.
1	Perform phase 3 operations if SC=1.

The TYPE2 signal is used to indicate that a Type II microinstruction is executed. The TYPE2 signal is sent to the Condition Control circuit of the CU1 card, the Memory or Device control circuit of the EO card, the MPLUS1 circuit of the MU4 card, and the Nano Parity Check circuit on the LI8 card. BORC is the Type II branch or call signal used to select MPM as an input to the M+1 incrementer. When at a high level, MPM+1 is sent to the Incrementer Register. DCALL is the Type II direct call signal used to allow the AMPCR to be replaced by MPCR.

There are eight Successor functions, and any one of these functions can be used whenever a Successor is specified in the microinstruction. (Refer to table II-4.) A WAIT Successor causes the present instruction, whose address is in the Microprogram Count Register (MPCR), to be repeated. The contents of the MPCR, the Alternate Microprogram Count Register (AMPCR), and the Incrementer Register (INCR) are not changed at this time.

A STEP Successor causes the next instruction in sequence to be executed. The contents of the MPCR are incremented by 1 (MPCR+1), but the contents of the AMPCR remain unchanged. The contents of the INCR contain MPCR+1. The absence of a Successor in a microinstruction causes a STEP to be performed.

A SAVE Successor causes the next instruction in sequence to be executed and the current Microprogram Memory (MPM) address to be saved in the AMPCR. This Successor is used to SAVE an address.

A SKIP Successor causes the next instruction in sequence to be skipped and the next instruction following it to be executed. The contents of the MPCR are incremented by 2 (MPCR+2), and the contents of the AMPCR are not changed. The contents of the INCR contain MPCR+2.

A JUMP Successor causes a branch in the sequence of instructions by transferring control to the AMPCR. The next instruction to be executed is the instruction whose address is indicated by the contents of the AMPCR+1. The contents of the MPCR are replaced with the contents of the AMPCR+1. The AMPCR remains unchanged, and the contents of the INCR contain AMPCR+1.

An EXEC Successor causes the execution of one non-sequential instruction whose address is AMPCR+1. The contents of both the MPCR and AMPCR are not changed, but control for the sequencing of instructions remains with the MPCR. The contents of INCR contain AMPCR+1.

A CALL Successor causes a branch in the sequencing of instructions by transferring control to the AMPCR. The next instruction to be executed is the instruction whose address is indicated by the contents of AMPCR+1. The contents of the MPCR are replaced with the contents of AMPCR+1. The contents of the AMPCR are replaced with the contents of the MPCR, and the contents of the INCR are AMPCR+1.

A RETN Successor causes a branch in the sequencing of instructions by transferring control to the AMPCR. The next instruction to be executed is the instruction whose address is indicated by the contents of AMPCR+2. The contents of the MPCR are replaced with the contents of AMPCR+2. The AMPCR remains unchanged, and the contents of the INCR contain AMPCR+2.

For a TYPE II STEP instruction, the contents of the MPCR and INCR contain

MPCR+1, and the contents of the AMPCR remain unchanged.

A TYPE II JUMP instruction causes a branch in the sequence of instructions by transferring control to the Type II microcode least significant 14-bits. The next instruction to be executed in the instruction whose address is indicated by the contents of MPM (3 through 16) plus 1. The contents of MPCR are replaced with the contents of MPM (3 through 16) plus 1. The AMPCR remains unchanged, and the contents of the INCR contain MPM (3 through 16) plus 1.

A TYPE II CALL instruction causes a branch in the sequencing of instructions by transferring control to the Type II microcode 14 least significant bits. The next instruction to be executed is the instruction whose address is indicated by the contents of MPM (3 through 16) plus 1. The contents of MPCR are replaced with the contents MPM (3 through 16) plus 1. The contents of AMPCR are replaced with the contents of MPCR, and the contents of INCR are MPM (3 thru 16) plus 1.

Functionally, the Microprogram Address (MPAD) is determined by what is contained in the MPAD Controls at the start of phase 1. Refer to the information contained under the Microinstruction Phasing and Microprogram Instruction Sequencing headings for detailed information pertinent to phase 1. Either the true successor (SC=1) with the values contained in nanobits N11, N12, and N13, or the false successor (SC=0) with the values contained in nanobits N14, N15, and N16, is loaded into the MPAD Controls. This is determined by the value of the Successor Control (SC) functions from the Condition Control circuits of the CU1 card during phase 1.

The MPAD Controls are also used to decode control bits MPM1 through MPM4 from the Microprogram Memory (MPM). These MPM control bits are used to determine the destination register or registers for the remaining bits in an MPM word for either Type I or Type II instructions. The destination register can be the Shift Amount Register (SAR). Literal register (LIT). Alternate Microprogram Count Register (AMPCR). Microprogram Count Reg-

MICROCONTROLS

	MPM BIT LOCATION																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	INSTR. TYPE
HEX. CODE→	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	
AMPCR =	0	0	LOAD AMPCR														II
MPCR =(Jump)	0	1	LOAD INCR & MPCR +1														II
CPCR =(Call)	1	0	AMPCR REPLACED BY MPCR LOAD INCR & MPCR +1														II
SAR =, LIT =	1	1	0	0	LOAD SAR				LOAD LIT								II
SAR =	1	1	0	1	LOAD SAR				ø ---------- ø								II
LIT =	1	1	1	0	ø ----- ø				LOAD LIT								II
ALL OTHERS	1	1	1	1	@ NANO ADDRESS												I

B7MPL
RESERVED
WORDS

NOTE: ø = Unused
@ = This bit is not used for addressing.

Fig. II-19 MPM CONTROLS

ister (MPCR), Incrementer Register (INCR) or both the SAR and LIT for a Type II instruction. See figure II-19 for information concerning MPM control bits. MPM destination bits, Type of microinstruction, and functional operation.

The SAR←MP signal transfers Microprogram Memory (MPM) data to the Shift Amount Register (SAR) of the SAR Control circuits on the CU2 card. This signal is true when an MPM word contains a "1100" or "1101" in the first four MPM control bits.

The following list specifies the logical equations that are used to generate the various MPAD Control functions which require MPM inputs:

$SAR \leftarrow MP = MPM1 \cdot MPM2 \cdot \overline{MPM3}$

$AM \leftarrow MP = MPM1 \cdot \overline{MPM2}$

$LIT \leftarrow MP = MPM4 \cdot MPM1 \cdot MPM2$

$TYPE2 = \overline{MPM1} + \overline{MPM2} + \overline{MPM3} + \overline{MPM4} + 3SS + FORT2 + FORCE\ STEP$

$BORC = DCALL + (\overline{MPM1} \cdot MPM2 \cdot \overline{FTYP2})$

$DCALL = MPM1 \cdot \overline{MPM2} \cdot \overline{FTYP2}$

The AM←MP signal transfers MPM data into the Alternate Microprogram Count Register (AMPCR) through the AMPCR Input Select gates. This signal is sent to the AMPCR clock gates on the MU6-1 through MU6-4 cards. This signal is true when an MPM word contains 00 in the first two MPM control bits.

The LIT←MP signal transfers MPM data into the Literal Register (LIT). This signal is sent to the LIT strobe enable gates on the MU6-2 and MU6-3 cards. This signal is true when an MPM word contains 1110 in the first four MPM control bits or a 1100 in the first four MPM control bits.

The TYPE2 signal indicates that a Type II microinstruction is to be executed. The Type 2 signal is sent to the Condition Control circuit of the CU1 card and the Memory or Device control circuits of the EO card. This signal is true when an MPM word contains a 0 in any one of the first four MPM control bits or when the Sequential Mode signal (SS), the Force Type II operation signal (FORT2), or the output from the Force Step flip-flop is present. The Sequential Mode signal (SS) is from the MTR/MEM switch on Field Engineering card FE3. When at a low level, this signal causes all Microprogram instructions to be executed as Type II microinstructions to force sequential stepping through the entire Microprogram Memory (MPM). The Force Type II operation signal (FORT2) is from the data cycle of the external operation card of which also receives a force step signal from the load interface cards. A Load Interface card is used when the Microprogram Memory (MPM) is a read/write memory (RWM). The Load Interface card, acts as the interface between an input device, and the read/write memory (RWM) is used when loading memory. The Force Step flip-flop output is generated from the FORCE STEP push button inputs on FE3. When signal STNAR goes low, a single Force Type II pulse is generated. When signal STNA goes low, the circuits are reset to permit the generation of another Force Step pulse. The BORC (branch or call) signal selects MPM as input to incrementer. DCALL gates MPCR to AMPCR. FTYP2 is generated as follows: FTYP2 = FORT2.SS.

MEMORY/DEVICE ADDRESS SECTION

The memory/device address section of the Memory Control Unit is comprised of the following registers and associated gate circuits: the Memory Address Register (MAR), Base Register 1 (BR1), Base Register 2 (BR2), and Output Select gates. MAR, BR1, and BR2 are located on the MU4 and MU5 cards. (See figure II-14.)

The Memory Address Register is an eight-bit register which contains the least significant eight-bits of the Memory/Device address. The outputs from this register are concatenated onto the least significant end of either the outputs from Base Register 1 or Base Register 2 to form the complete memory/device address.

MAR may be loaded from either the least significant byte (eight LSB's) of the Barrel Switch (BSW) on the LU1 card or the Literal register (LIT) on the MU6-2 and MU6-3 cards. The outputs from BSW or LIT are sent to the MAR/CTR input select gates.

The eight LSB's from the Barrel Switch are enabled as inputs to the MAR by enable control signal DN46. When this signal is false, the Literal Register inputs (LIT1 through LIT8) are enabled as inputs to the MAR. The enable control signal DN46, which is generated by the Control Register on the MU5 card, corresponds to nanobit N46 from the Nanomemory. The strobe enable signal DN45, which is also generated by the Control Register on the MU5 card, is used as an input to a NAND gate circuit along with the phase 3 clock signal (3CLKA). The output of this NAND gate circuit is applied to the clock (CP) inputs of the

MAR flip-flops (MAR1 through MAR8). Signal DN45 corresponds to nanobit N45 from the Nanomemory.

Nanobits N45 and N46 are used to control Memory Address Register input selection as follows:

Nanobits 45	46	MAR Input Selection
0	-	No change in the MAR.
1	0	Literal register (LIT) inputs (LIT1 through LIT8) are sent to the MAR (LMAR).
1	1	Eight least significant bits (least significant byte) of the Barrel Switch output are sent to the MAR.

The Output Select signals (OS9 through OS16) from the Memory Address Register (MAR) are used as inputs to the least significant bits of the Y-Select gates on the LU6 and LU7 cards. OS9 through OS16 are concatenated with OS1 through OS8 of the Base Register (BR1 or BR2) and are used as inputs to all 16-bits of the Y-Select gates on the LU6 and LU7 cards. The BRn and MAR Output Select signals (OS1 through OS16) also provide address information to Data/Program Memory through the Shared Memory cards.

Base Register 1 is an eight-bit register which contains the base address of a 256-word block of data or a device address. The output from BR1 is applied to the Output Select gates, which generate Output Select signals OS1 through OS8. These Output Select bits are concatenated with the Output Select bits from the Memory Address Register (MAR) to form an absolute memory or device address that is sent to the 16-bits of the Y-Select gates on the LU6 and LU7 cards. Output Select signals from BR1 are also sent to the Port Selector (PS1 cards). (See figure II-13.) The inputs to BR1 are from the most significant eight-bits of the Barrel Switch (BSW) output.

These Barrel Switch (BSW) outputs are applied directly to the parallel data (P) inputs of the BR1 flip-flops and BR2 flip-flops. The strobe enable signal DN43, which is generated by the Control Register of the Memory Control Unit on the MU2 card, is an input to a NAND gate circuit along with the phase 3 clock signal (3CLKA). The output of this NAND gate is sent to the clock (CP) inputs of the BR1 flip-flops. Strobe enable signal DN43 corresponds to nanobit N43 from the Nanomemory. Nanobit N43 is used as the control for Base Register 1 input selection as follows:

Nanobit N43	Base Register 1 Input Selection
0	No change in the contents of Base Register 1.
1	Next eight LSB's to the least significant byte of the Barrel Switch (BSW) are sent to BR1.

Base Register 2 (BR2) is an eight-bit register which usually contains the base address of a 256-word block of data or a device address. The output from the BR2 is applied to the Output Select gates, which generate Output Select signals (OS1 through OS8). These Output Select bits are concatenated with the Output Select bits from the Memory Address Register (MAR) to form an absolute memory or device address that is sent to the 16-bits of the Y-Select gates on the LU6 and LU7 cards. Output Select signals from BR2 are also sent to the Port Selector (See figure II-14.) Inputs to BR2 are from the eight-bits of the most significant byte of the Barrel Switch output.

These Barrel Switch outputs are applied directly to the Parallel data (P) inputs of the BR2 and BR1 flip-flops. The strobe enable signal DN44, which is generated by the Control Register of the Memory Control unit on the MU2 card, is an input to a NAND gate circuit along with the phase 3 clock signal (3CLKA). The output of this NAND gate is sent to the clock (CP) inputs of the BR2 flip-flops. Strobe enable signal DN44 corresponds to nanobit N44 from the Nanomemory. Nanobit N44 is used as the control for Base Register 2 input selection as follows:

Nanobit N44	Base Register 2 Input Selection
0	No change in the contents of Base Register 2.
1	The eight-bits of the most significant byte of the Barrel Switch (BSW) are sent to BR2.

The Output Select (OS gates, which are located on the MU4 and MU5 cards, are used for selecting data memory addresses or device addresses from either Base Register 1 or Base Register 2. The outputs from these Output Select gates are concatenated with the Output Select bits from the Memory Address Register to form an absolute data memory (16-bit) or device (8-bit) address that can also be sent to the Y-Select gates on the LU6 and LU7 cards. The output from BR1 and MAR can be repre-

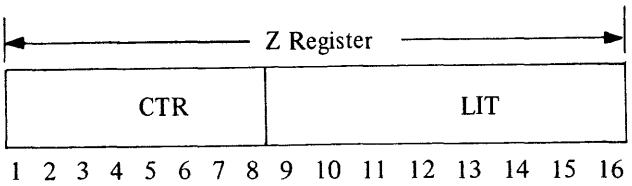
sented as BR1/MAR or MAR1. Likewise, the output from BR2 and MAR can be represented as BR2/MAR or MAR2. The destinations of the Output Select bits that have been previously described for the Memory Address Register and Base Registers (BR1 and BR2) have the memory or device formats shown in figure II-20.

The Output Select signal (OSEL), which is generated by the Control Register of the External Operation Control (EO card), is used to transfer Base Register 2 outputs through the Output Select gates. When this signal is at a low level, Base Register 1 outputs are transferred through the Output Select gates.

Z REGISTER SECTION

The Z register section is comprised of the Counter (CTR), and the Literal register (LIT). CTR and LIT are registers located in the Memory Control Unit.

CTR is located on the MU5 card and LIT is located on the MU6-2 and MU6-3 cards. (See figure II-17.) The eight-bits of the Counter (CTR provide the most-significant byte of the Z Register and the eight-bits from the Literal Register (LIT) provide the least significant byte of the Z Register as indicated in the following format:



The Counter (CTR), which is eight-bits in length, is used for loop control and various count functions. CTR is comprised of two UP Counter modules (1447 3771) and can be loaded from either the Literal register or the least significant byte of the Barrel Switch by means of the MAR/CTR input select gate circuits. The CTR output signals (CTR1 through CTR8) are used as inputs to the most significant bits (MSB's) of the X-Select gates on the LU2 cards and the Y-Select gates on the LU6 and LU7 cards. The CTR can also be incremented by ONE. When the CTR overflows (CTR is incremented to an all ONES condition, which causes an all ZERO condition at the next clock pulse), the Counter Overflow bit (COV) is set in the Condition Register on the CU1 card. The Counter Overflow (COV) bit is reset by either testing or loading the CTR.

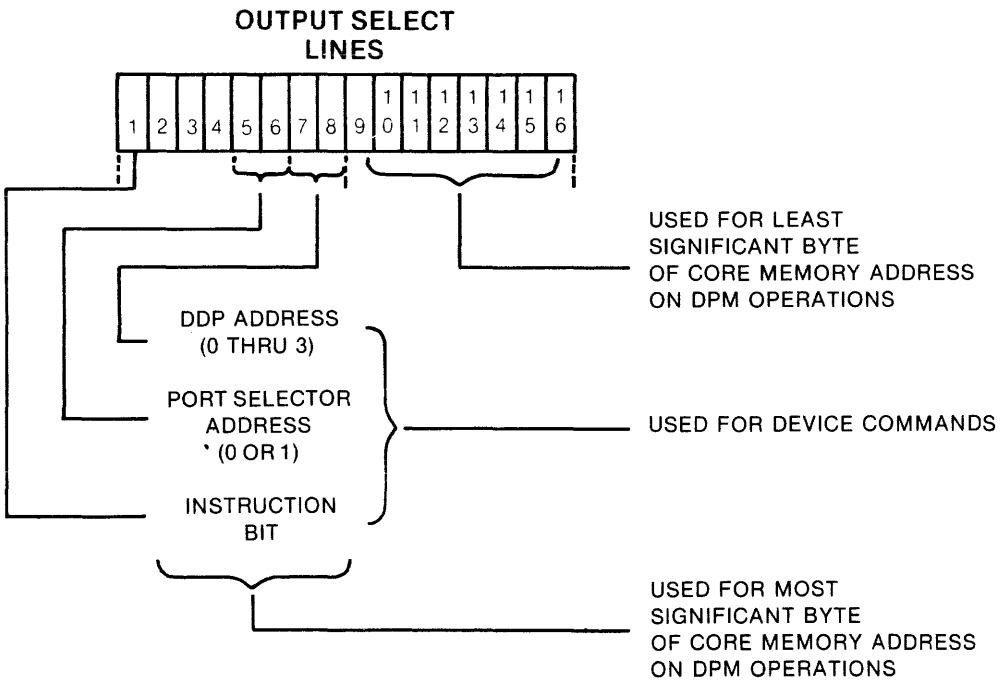


Fig. II-20 OUTPUT SELECT MEMORY OR DEVICE FORMATS

Enable signals DN47 and DN48 are used as control signals for the CTR most-significant bit and least-significant bit modules. Signal DN47 enables the CTR to increment if signal DN48 is false. DN48 enables the transfer of data to the CTR and resets the Counter Overflow (COV) flip-flop in the Condition Register of the CU1 card. The transfer of data (LIT or BSW outputs) to CTR will override increment CTR controls (signal DN47), if these conditions occur simultaneously. Enable signal DN46 is used to allow the Barrel Switch inputs or Literal Register inputs to be transferred through the MAR/CTR input select gates. When this signal is at a high level, it enables the transfer of the Barrel Switch inputs through the select gates. Likewise, when this signal is at a low level, the transfer of Literal Register inputs through the MAR/CTR input select gates occurs.

The DN46, DN47, and DN48 signals, which are outputs from the Control Register of the MU5 card, correspond to nanobit signals N46, N47, and N48 from the Nanomemory. N46, N47, and N48 are used to control CTR input selection as follows:

Nanobit			Counter (CTR) Input Selection
46	47	48	
-	0	0	No change in the contents of the CTR.
0	0	1	Literal register outputs (LIT) are sent to the Counter (LCTR).
1	0	1	Least significant byte of Barrel Switch outputs (BSW) are sent to the Counter (CTR).
-	1	0	Increment Counter (Modulo 256), (INC).

The Literal register (LIT) is an eight-bit register which receives data inputs from the Microprogram Memory (MPM). MPM outputs MPM9 through MPM16 are sent to the parallel (P) inputs of the Literal register flip-flops. The clock (CP) input of the LIT flip-flops are enabled by the output from a two-input NAND gate. These inputs are the system clock (SCLKF) and the transfer signal (LIT MP). LIT MP is the transfer MPM data-to-LIT signal, which is generated by the MPAD Controls on the MU2 card. This signal enables the strobe pulse to the LIT register.

CONTROL REGISTER SECTION

The Control Register, which is located in the Memory Control Unit on the MU2, MU3, MU4 and MU5 cards, is used to store the nanomemory control signals that are not used in phase 1 operations. Figure II-21 is a functional block diagram of the Control Register and associated logic circuits of the Memory Control Unit cards.

Some of the nanobit control signals from the Nanomemory are decoded before being strobed into the Control Register. Nanobit signals N17 through N19 and N22 are applied to the inputs of decoded gate circuits instead of being applied directly to the data inputs of the applicable Control Register flip-flops. The following nanobit signals are applied directly to the data input of the corresponding Control Register flip-flops: N20 through N31 and N34 through N48.

Nanobit control signals N17 through N19 are used in the generation of the ACTL1, ACTL2, NQE1, NQE2, and NQE3 signals. The ACTL signals are used to select the inputs to the X-Select gates on the LU2 cards. The NQE signals are used as enable signals to the respective portion of the X-Select gates. NQE1 is used to enable inputs to the most-significant bit portion of the X-Select gates, and the NQE3 signal is used to enable inputs to the least-significant bit portion of the X-Select gates. The NQE signals inhibit unwanted bits during Literal register output. Counter output, or External (EXT) bit transfer through the X-Select gates of the LU2 cards.

Nanobit signals N20 and N21 are used by the Control Register to generate nanobit decode signals NB20 and NB21. NB20 and NB21 controls are sent to the most significant bit of the Y-Select gates of the LU6 and LU7 cards. Nanobit control signal NB20- - -1 is connected to ground on all LU6 and LU7 cards, except the LU6-1 card that contains the MSB of the Y-Select gates. The nanobit control signal NB21 is also connected to the LU6-1 card that contains the MSB to the Y-Select gates.

Nanobit signals N22 through N24 are used in the generation of the ENB1, ENB2, ENB3, SELBA, and SELBB control signals. ENB enable signals are used to select inputs to the Y-Select gates on the LU6 and LU7 cards. ENB1 controls the most-significant bits, and ENB3 controls the least-significant bits. SELB control signals are used to select various inputs to the Y-Select gates on the LU6 and LU7 cards.

Nanobit signals N25 and N26 are used in the generation of decoded nanobit control signals NB25 and NB26. Nanobit control signal NB25, which is generated by the Control Register on the MU3 card, controls the least-significant bit of the Y-Select gates on the LU6 and LU7

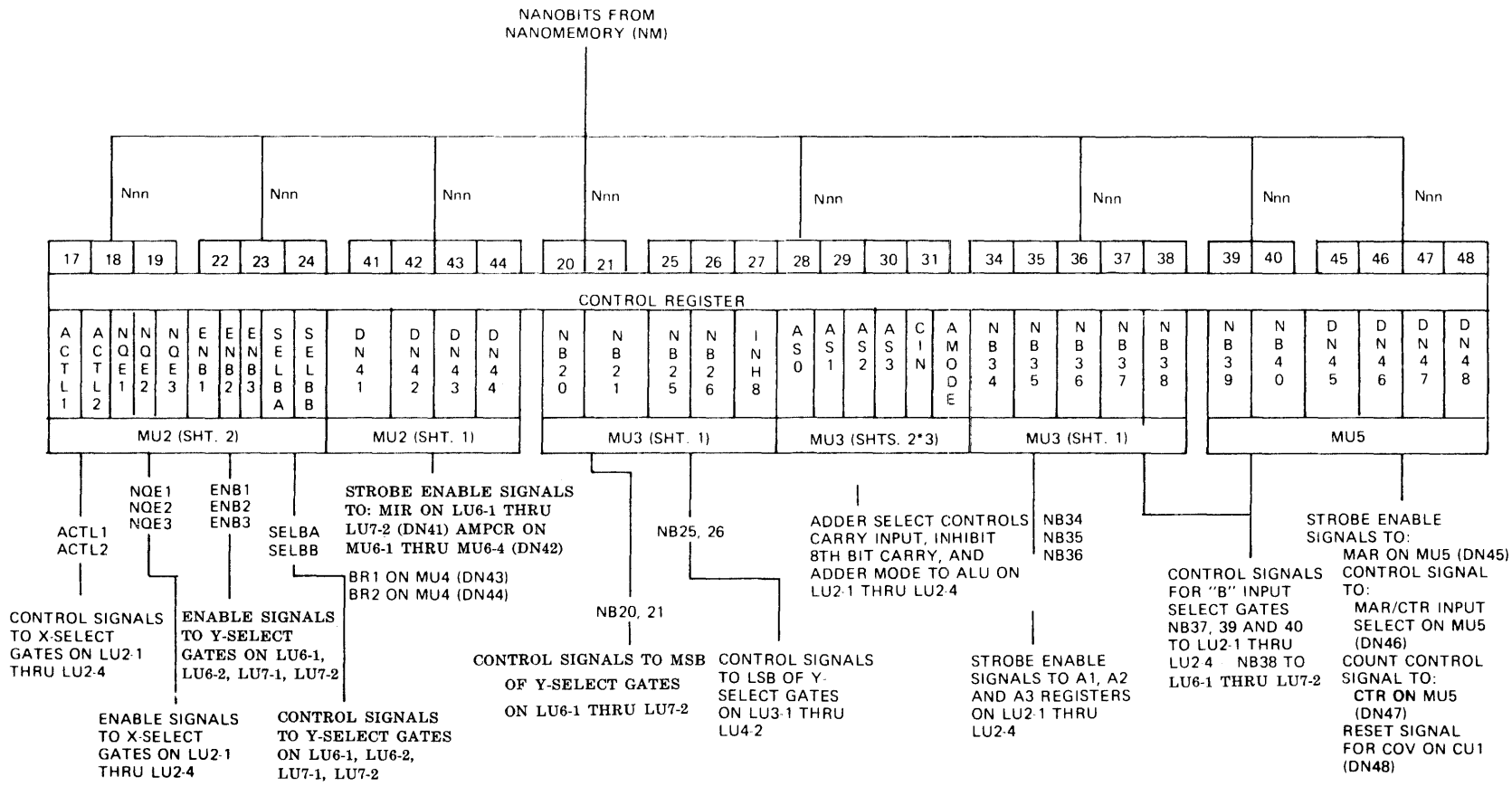


Fig. II-21 CONTROL REGISTER AND ASSOCIATED LOGIC OF MEMORY CONTROL UNIT, BLOCK DIAGRAM

cards. This signal is connected to ground on all LU6 and LU7 cards, except the LU7-2 card that contains the LSB of the Y-Select gates. Nanobit control signal NB26, which is also generated by the Control Register on the MU3 card, is also used to control the LSB of the Y-Select gates on the LU7-2 card.

Nanobit signals N27 through N31 are sent to the data input of the respective Control Register flip-flops NB27 through NB31 on the MU3 card. The outputs from these flip-flops are applied to decode output gates, which are used to generate the Adder select control signals (AS0, AS1, AS2, and AS3), Carry Input signal (CIN), Inhibit 8-bit Carry signal (INH8), and Adder Mode signal (AMODE) that are sent to the Arithmetic/Logic Unit on the LU2-1 through LU2-4 cards. The four Adder control signals (AS0 through AS3) are used with the AMODE signal to select the proper arithmetic or logic operation of the Arithmetic/Logic Unit modules (1447 3730) on the LU2 cards. Carry Input signal (CIN) is sent to the least significant bit of the Adder for arithmetic operations "X+Y+1 or X-Y". The Inhibit 8-bit carry signal (INH8), when false, is used to inhibit the carry out of the least-significant eight-bit group of the Adder.

Nanobit signals N34 through N36 are sent to the data input of the respective Control Register flip-flops NB34 through NB36 on the MU3 card. The outputs from these flip-flops are sent to the A Register on the LU2 cards. Signal N34 is used to enable the strobe pulse to the A1 Register. Signal N35 is used to enable the strobe pulse to the A2 Register, and signal N36 is used to enable the strobe pulse to the A3 Register on the LU2 cards.

Nanobit signals N41 through N48 are sent to the data (P) input, of the respective Control Register flip-flops on the MU2 and MU5 cards. The outputs from these flip-flops (DN41 through DN48) are used to enable the strobe pulses to various registers to control input selection. Signal DN41 is used to enable the strobe pulse to the Memory Information Register on the LU6-1 through LU7-2 cards for the MIR input selection. Signal DN42 is used to enable the strobe pulse to the Alternate Microprogram Count Register (AMPCR) on the MU6-1 through MU6-4 cards for AMPCR input selection. Signal DN43 is used to enable the strobe pulse to Base Register 1 on the MU4 card for BR1 input selection. Signal DN44 is used to enable the strobe pulse to Base Register 2 on the MU4 card for BR2 input selection.

Nanobit control signal DN45 is used to enable the strobe pulse to the Memory Address Register on the MU5 card for MAR input selection. Nanobit control signal DN46 is used to enable Barrel Switch outputs to transfer through the MAR/CTR selection gates. When this signal is false, Literal Register outputs are permitted to transfer through the MAR/CTR selection gates on the MU5 card. Nanobit control signal NB47 is used to enable CTR to increment if signal DN48 is false. Signal DN48 is used to enable the transfer of data to CTR on the MU5 card and also to reset the Counter Overflow (COV) flip-flop in the Condition Register of the Condition Control circuits on the CU1 card. Signals DN45 and DN46 are used to control MAR input selection, and signals DN46, DN47, and DN48 are used to control Counter (CTR) input selection.

There are five remaining Control Register flip-flops on the Control Unit card (CU2) and four control flip-flops on the External Operation Control card (EO). Refer to the functional description of the Shift Control Register and associated control circuits on the CU2 card, or the Control Register and associated control circuits of the EO card.

EXTERNAL OPERATION CONTROL (EO2)

The External Operation (EO) Control section consists of the memory and device control and the memory hardware error control circuits. (See figure II-22.)

MEMORY AND DEVICE CONTROL

External operations to the memory and DDP are enabled by NANO memory bits 51 through 54. The bit functions are listed in table II-12.

As shown in figure II-22, NMN52 sets a D-type flip-flop if the clock input condition is met (CUEXTOP and NMN53 both high). (There is another logical condition, where CUEXTOP and NMN51 are both high, but, as table II-12 shows, NMN53 is always on with NMN51 so that the first condition is always met.)

NMN52 sets the D-type flip-flop and enables signal EOB52 if signal STSTEAL/ is high. A low STSTEAL/ means a DDP with direct memory access is accessing memory. Because the MIR BUS is required, a low STSTEAL/ inhibits EOB52 from enabling the PSU. The NMN52 flip-flop stays set until one of the clock input conditions exists. These conditions and: CUEXTOP high with either NMN53 or

TABLE II-12 NANO MEMORY EO CONTROLS

Nanobits 51	52	53	54	Memory/Device
0	0	0	0	No change
0	0	0	1	Not applicable
0	0	1	0	Memory Read 1 (MR1): read contents from memory at memory address specified by Base Register 1 and Memory Address Register (BR1/MAR).
0	0	1	1	Memory Read 2 (MR2): read contents from memory at memory address specified by Base Register 2 and Memory Address Register (BR2/MAR).
0	1	0	0	Device Load (DLD): used in system configurations that use a CON card. Only affects control register 51.
0	1	1	0	Memory Write 1 (MW1): write contents of Memory Information Register into memory address specified by Base Register 1 and Memory Address Register (BR1/MAR).
0	1	1	1	Memory Write 2 (MW2): write contents of Memory Information Register into memory address specified by Base Register 2 and Memory Address Register (BR2/MAR).
1	0	0	0	ASR: status request for highest priority unselected device (indicates no change in port selector address). Select BR1 for device or memory address lines.
1	0	1	0	Device Read 1 (DR1): read from device specified by device address in Base Register 1.
1	0	1	1	Device Read 2 (DR2): read from device specified by device address in Base Register 2.
1	1	1	0	Device Write 1 (DW1): write contents of Memory Information Register to the device specified by device address in Base Register 1.
1	1	1	1	Device Write 2 (DW2): write contents of Memory Information Register to the device specified by device address in Base Register 2.

NMN51 high, or both NMN52 and NMN53 high.

Figure II-22 shows the logic associated with NANO bits 51, 52, 53, and 54 to perform device operations.

NMN51

NMN51 is used only for device operations. When NMN51 and CUEXTOP are both high

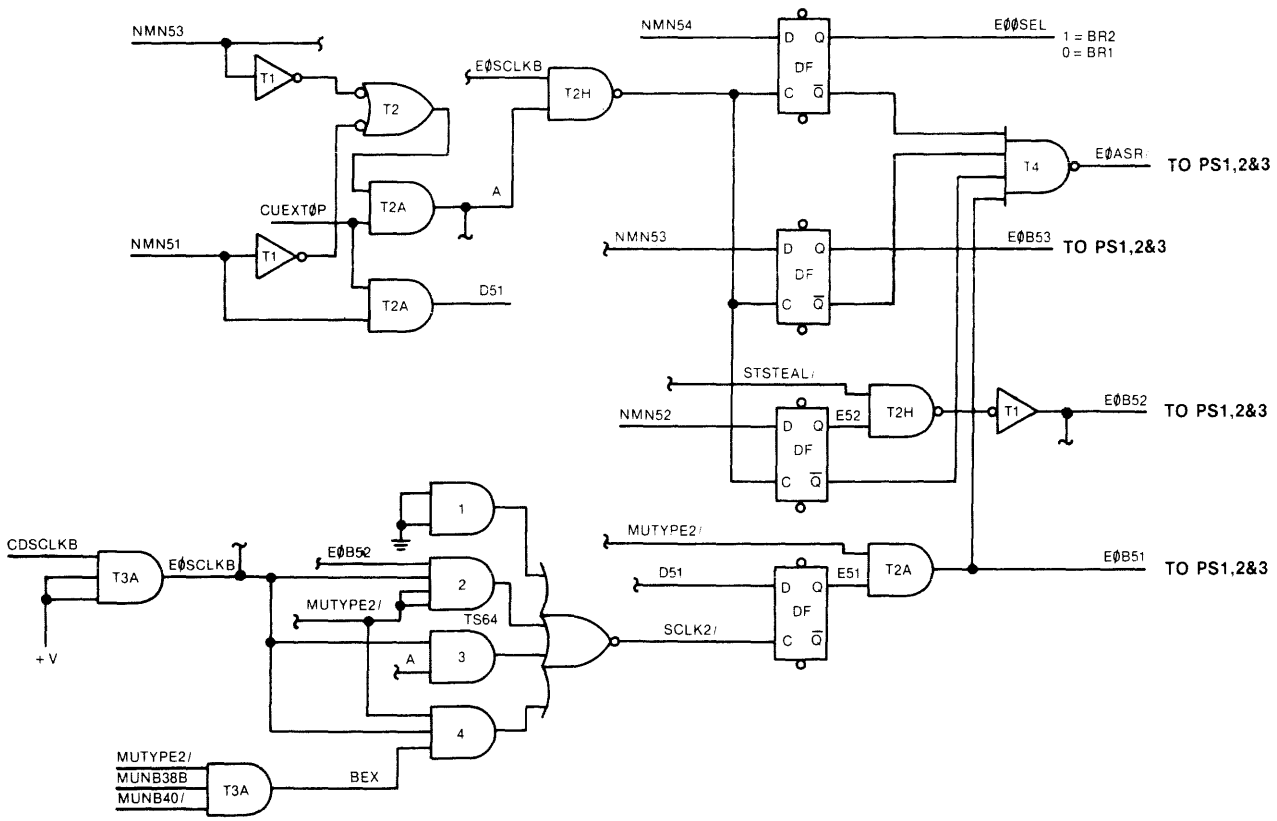


Fig. II-22 MEMORY AND DEVICE CONTROL LOGIC

(figure II-22), they produce signal D51. D51 sets a D-type flip-flop which is clocked by gate 3 because signal A is high at this time. The output of this flip-flop is ANDed with MUTYPE2/ so that signal EOB51 will only be enabled during execution of a type I instruction. EOB51 goes to the three port-select cards to enable a device operation. This flip-flop stays set until one of the three gate conditions is met.

NMN52

NMN52 is used to enable write operations (device and memory).

NMN53

NMN53 is used to enable all device or memory read/write operations. NMN53 is not used in ASR or ASE operations. (See figure II-22.) NMN53 sets a D-type flip-flop if the clock input conditions are met (CUEXTOP high). When NMN53 sets the flip-flop, signal EOB53 is enabled and is used to enable device operations at the port select unit.

NMN54

NMN54 is used to specify BR1 or BR2 register as inputs to OS lines 1 through 8. NMN54 sets a D-type flip-flop if the clock input conditions are met (CUEXTOP high and either NMN53 or NMN51 high, or both NMN53 and NMN51 high).

When NMN54 sets the flip-flop, EOSEL is enabled. A high EOSEL goes to the MU4 card and enables BR2 to OS lines 1 through 8. A low EOSEL enables BR1 to OS lines 1 through 8.

ASR

ASR is a status request operation. EOASR/ is enabled when NMN54, NMN53, and NMN52 are low and NMN51 is high (figure II-22).

A low EOASR/ enables the status word onto the EXT bus from the device which has the highest priority and is generating an interrupt.

Because NMN54 is low, BR1 is selected as an input to OS lines 1 through 8. This operation ensures that BR1 is selected for some future operation.

ASE

ASE is used to select the BR2 register as inputs to OS lines 1 through 8. This operation ensures that BR2 is selected for a future operation.

ASE does not perform any external operations.

MEMORY OPERATIONS

Simultaneous access to both MPM and S-level memory is not allowed because they share the same physical memory. The EO2 memory control logic enables the incrementer or the OS lines to address memory. The memory control logic is shown in figure II-23.

EODATCY

EODATCY, in normal MPM read operations, is low because of the setting of the EODATCY flip-flop by a high signal MOP/. When a data memory read or write operation is to be performed, the EODATCY flip-flop is reset and signal EODATCY goes high. EODATCY resets because signal MOP/ goes low. MOP/ goes low because NMN53 is high, and causes NMN53/ to go low.

The output of a NAND is ANDED with CUEXTOP and causes signal A to go high. A, NANDed with a high NMN51/ (NMN51 is low for a data memory OP), causes signal MOP/ to go low and reset the EODATCY flip-flop. Signal EODATCY goes to cards SM6 and SM7 to clock the MPM lines to storage registers to be enabled to the EXT BUS if the data memory operation is a read operation. EODATCY also goes to cards SM4 and SM5 to select the OS lines, rather than the incrementer, as inputs to the memory address lines (SMADR01/ to SMADR16/).

A low EODATCY/ causes signals EOFORT2/ and EOINHSP/ to go low. EOFORT2/ goes to MU2 and inhibits phase 3 clock (MU3CLKI/) and delays phase-3 operations from taking place. Type II instructions are inhibited by SMMPM1 through SMMPM4, when they are forced to ONES by signal EODATCY on the SM4 cards. Thus, at this time the processor is being inhibited by forcing type I and type II instructions, both of which are inhibited. The incrementer is also inhibited from stepping by a low signal EOINHSP/. This entire operation allows the common memory to be accessed by an external operation rather than the processor memory control unit.

EOWRITE

EOWRITE enables the memory write logic. A memory write operation can occur two ways:

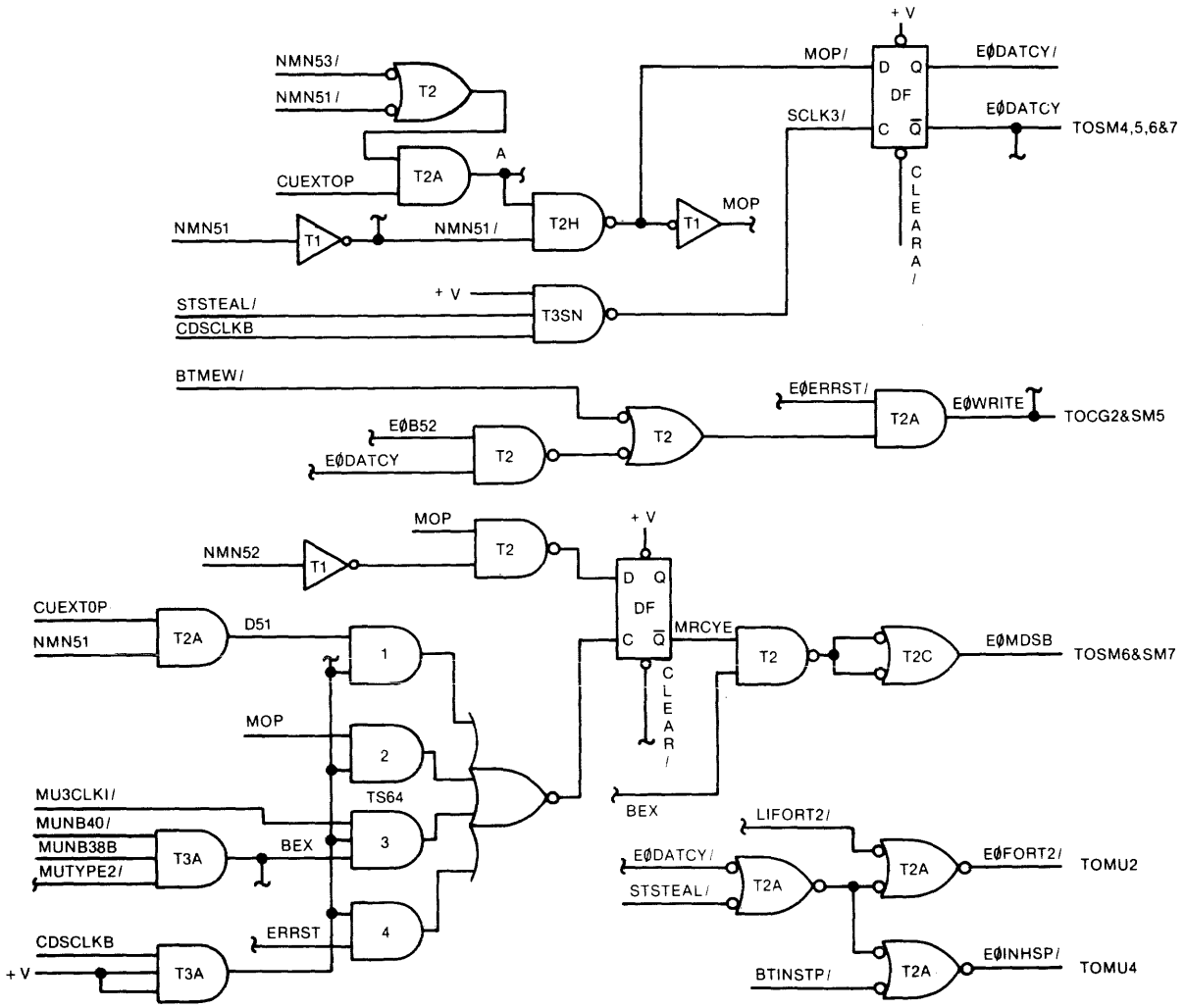


Fig. II-23 MEMORY CONTROL LOGIC

through the loader interface cards during a memory load operation,or during a data memory write operation.

As shown in figure II-23, EOWRITE goes high if BTMEW/ is low. If both EODATCY and EOB52 are high, BTMEW/ goes low during a load memory operation and selects the EXT BUS as input to the data to memory lines (SM6 and SM7). In a data memory write operation, NMN52 and EODATCY are high as described previously. EOWRITE goes to the CG2 card to enable CGWRITE. EOWRITE also goes to card SM5 to disable the parity check logic during the write operation. The MIRO BUS is enabled as input to the data to memory lines because BTMEW/ is high (SM6 and SM7).

EOMDSB

EOMDSB goes high to enable the MPM data (stored on cards SM6 and SM7) to the EXT BUS after a data memory read operation.

EOMDSB goes high when the current NANO word contains a data memory read operation. This means that NMN51 and NMN52 are low and NMN53 is high. A low NMN52, with a high MOP, causes the input to the MRCYE flip-flop to go low. A high MOP will also enable gate 2 (figure II-23) to allow SCLK to reset the MRCYE flip-flop and produce a high MRCYE.

The next clock pulse enables the MPM lines into the SM6 and SM7 storage registers. Also, at this clock pulse time, the incrementer is again selected to address memory. This instruction should contain a BEX because, to use the data just read, it must be placed into the B register over the EXT BUS.

The NANO bits for a BEX go high during this clock period, and the next three CLK sig-

nals set their corresponding flip-flops. BEX is now high and causes EOMDSB to go high. The data present in the SM6 and SM7 storage register to the EXT BUS. The next three CLK/SCLK signals enable the EXT BUS into the B register and complete the data memory read operation by resetting the MRCYE flip-flop.

Timing diagram II-24 shows the major signal timing during data memory operations.

MEMORY HARDWARE ERROR

The memory hardware error detection circuits on the EO2 card detect and report 14 distinct errors. Table II-3 lists the types of errors and the corresponding EO2 indicators and EXT bits.

TABLE II-13 MEMORY HARDWARE ERRORS

EXT Bits (and EO2 Indicators)					
1	13	14	15	16	Error
(Halt)	(8)	(4)	(2)	(1)	
0	0	0	0	0	None
1	0	0	0	0	Transient
1	0	0	0	1	Loader error
1	0	0	1	0	Load MPM parity
1	0	0	1	1	Load address
1	0	1	0	0	Nano parity
1	0	1	0	1	Nano address
1	0	1	1	0	MPM parity
1	0	1	1	1	MPM address
1	1	0	0	0	Not used
1	1	0	0	1	DPM address, write
1	1	0	1	0	DPM parity
1	1	0	1	1	DPM address, read
1	1	1	0	0	Not used
1	1	1	0	1	Steal address, write
1	1	1	1	0	Steal parity
1	1	1	1	1	Steal address, read

The following list describes the various types of errors:

LOADER ERROR

A loader error occurs if the light check fails or an LRC error is detected while paper tape is being loaded.

LOAD MPM PARITY ERROR

A load MPM parity error occurs if, after a paper tape character has been written into memory, the concurrent memory read detects a parity error.

LOAD ADDRESS ERROR

A load address error occurs if the current shared memory address exceeds the bounds of memory during a paper tape load operation.

NANO PARITY ERROR

A nano parity error occurs if a parity error is detected during a nano memory access.

NANO ADDRESS ERROR

A nano address error occurs if a nano memory parity error is detected and MPM06 is high. (MPM06 binary weight is 512.) The reason for the parity error is because a nano memory address overlimit error is not a true parity error.

MPM PARITY ERROR

An MPM parity error occurs if a memory parity error is detected while MPM is being addressed by the processor incrementer register.

MPM ADDRESS ERROR

An MPM address error occurs if the memory address, which is under control of the processor incrementer register, exceeds the hardware memory address limit.

DPM ADDRESS WRITE ERROR

A DPM address write error occurs if the memory address, which is under control of the MAR register, exceeds the hardware memory address limit during an MW operation (data memory write).

DPM PARITY ERROR

A DPM parity error occurs if a memory parity error is detected during an MR operation (data memory read).

DPM ADDRESS, READ ERROR

A DPM address read error occurs if the memory address, which is under control of the MAR register, exceeds the hardware memory address limit during an MR operation (data memory read).

STEAL ADDRESS WRITE ERROR

A steal address write error occurs if the memory address, which is under control of the device stealing, exceeds the hardware memory address limit during a steal write to memory operation.

STEAL PARITY ERROR

A steal parity error occurs if a memory parity error is detected during a device steal memory cycle.

STEAL ADDRESS READ ERROR

A steal address read error occurs if the memory address, which is under control of the de-

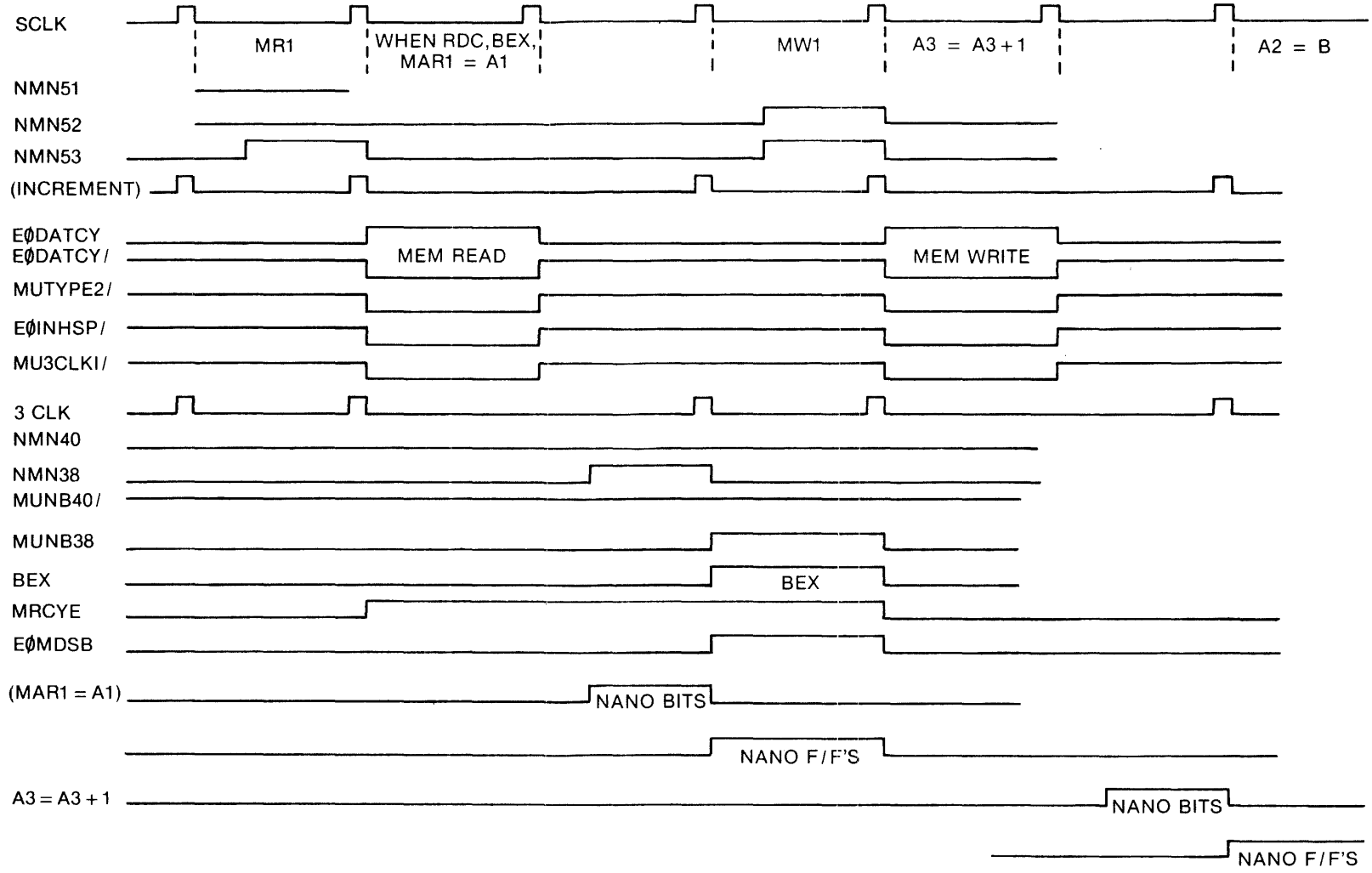


Fig. II-24 DATA MEMORY OPERATION TIMING

vice stealing, exceeds the hardware memory address limit during a steal read from memory operation.

All of the types of errors are produced from four basic error signals:

a. Loader Interface Error (LIERR).
b. Shared Memory Parity Error (SMPERR).
c. Shared Memory Address Error (SMAEF).
d. Nano Parity Error (LINPE).

Figure II-25 shows the error logic on the EO2 card, where the four basic error signals are applied. If any of these signals is low, signals EOERR and EOANYER go high. Note that when the INH/NORM switch is in position INH (inhibit), signal EOANYER is inhibited from going high.

Error signals SMAEF/, SMPERR/, and LINPE/ are clocked into the four D-type flip-flops contained on the DR4 I.C. chip. The clock input for this chip comes from the TS64 chip. The clock for the first error is provided by gate 4 of the TS64 chip. Inputs to this chip are signals CGERCLK (occurs approximately 100 nanoseconds before SCLK/) and EOERR. Thus, if an error exists, CGERCLK clocks the error signal into the D-type flip-flops.

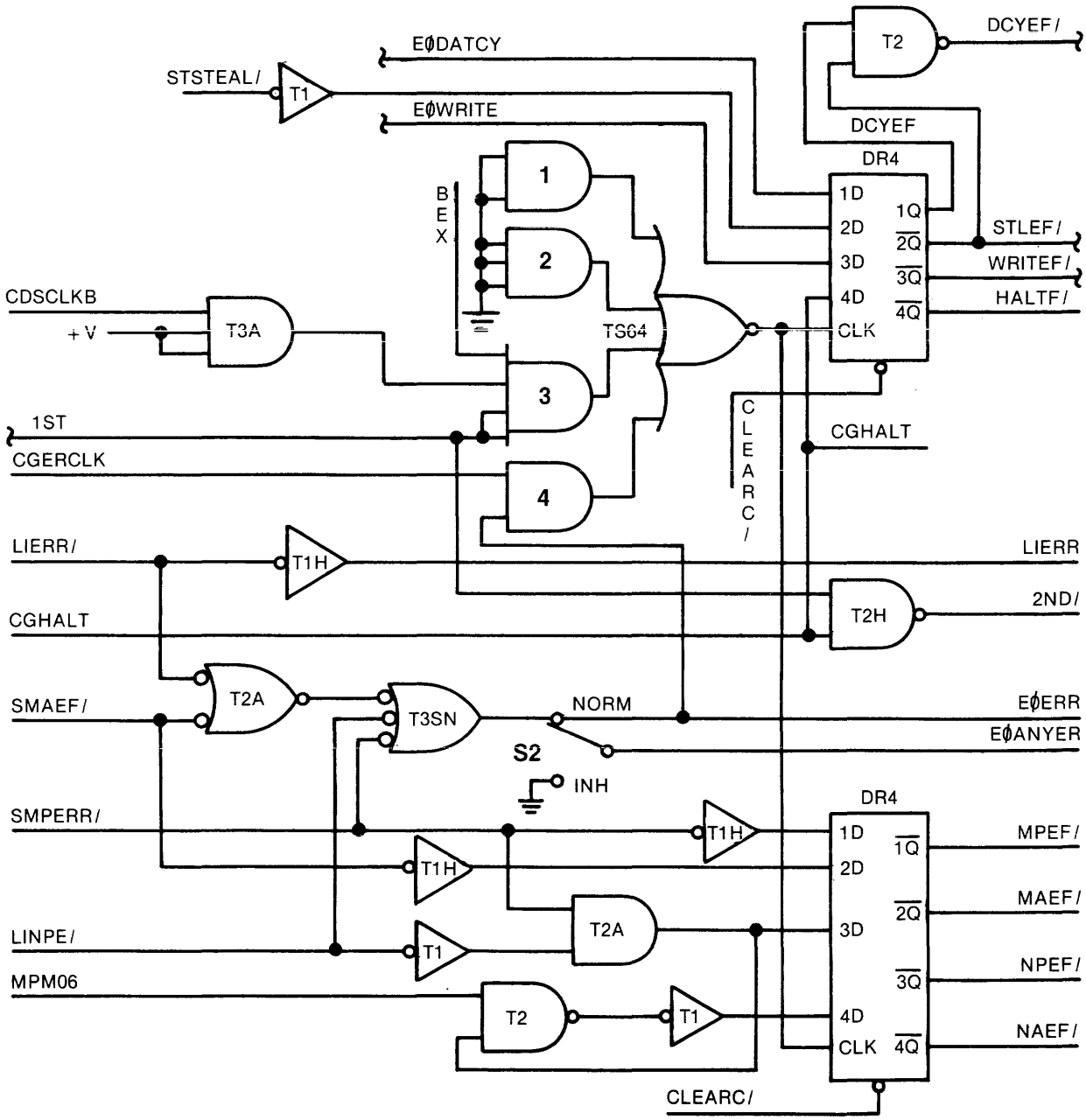


Fig. II-25 MEMORY HARDWARE ERROR LOGIC

The other DR4 chip shown in figure II-25 is used to store CGHALT and the mode of operation in effect when the error occurred. (By knowing the mode of operation, it can be determined if a memory error occurred in MPM or DPM, if the operation was a read or write operation, and if the error occurred during a data cycle or a steal cycle.)

The outputs of the two DR4 chips go to the decoder circuit shown in figure II-26. The decoder circuit decodes the type of error and the mode of operation and enables the four error indicators to display the type of error (table II-13). EXT bits 16, 15, 14, 13, and 1 are also enabled by the decoder circuit to allow the software error routine to determine the type of error.

Figure II-27 shows the halt circuit on the CG2 card. Signal CGHALT goes high if signal EOANYER (figure II-25) is high when signal ERCLK/ (clock error) goes low. If EOANYER is high, signal HALT/ goes low and inhibits SCLK. CGHALT goes to the EO2 card halt circuit (figure II-28) and sets the ERRST flip-flop when the CGSCLKR pulse occurs. CGSCLKR is not inhibited during an error condition and occurs at the same time as SCLK.

CGHALT goes low when signal TC goes high and signal EOASPRST is high. TC occurs 700 nanoseconds after ERCLK to allow the next SCLK to occur (if a second error does not exist) and start the error routine.

The error routine is enabled by causing the shared-memory address lines to address word 0. This is accomplished by a low signal EOERRST/, which disables all inputs to the shared memory address lines on cards SM4 and SM5. The error routine will execute a BEX instruction to read the error currently on the EO2 EXT BUS. A DLD instruction will also be executed to reset the first flip-flop shown in figure II-28.

If a second error occurs before the DLD instruction resets the first flip-flop, signal EOASPRST goes low as shown in figure II-29. This inhibits TC from resetting the second

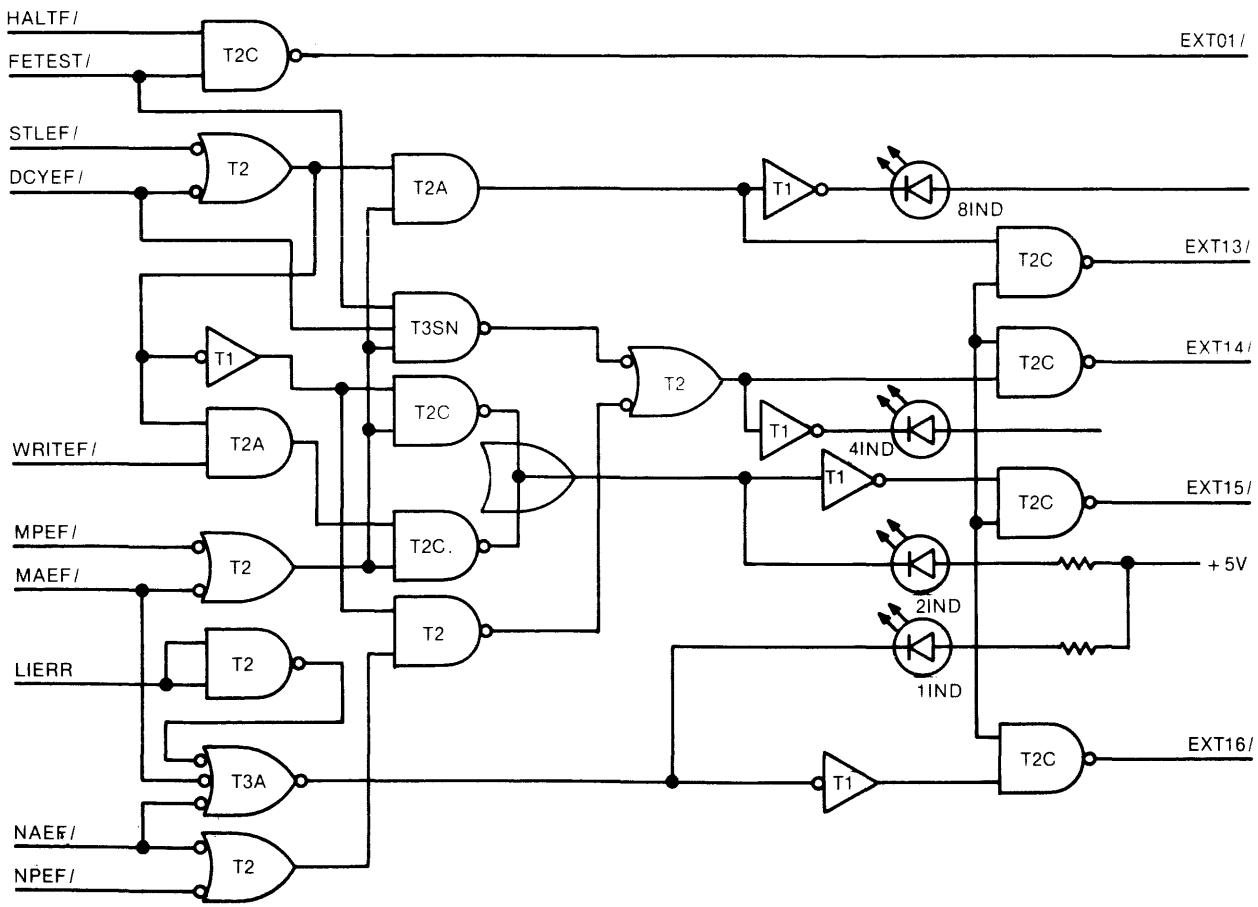


Fig. II-26 MEMORY HARDWARE ERROR DECODE LOGIC

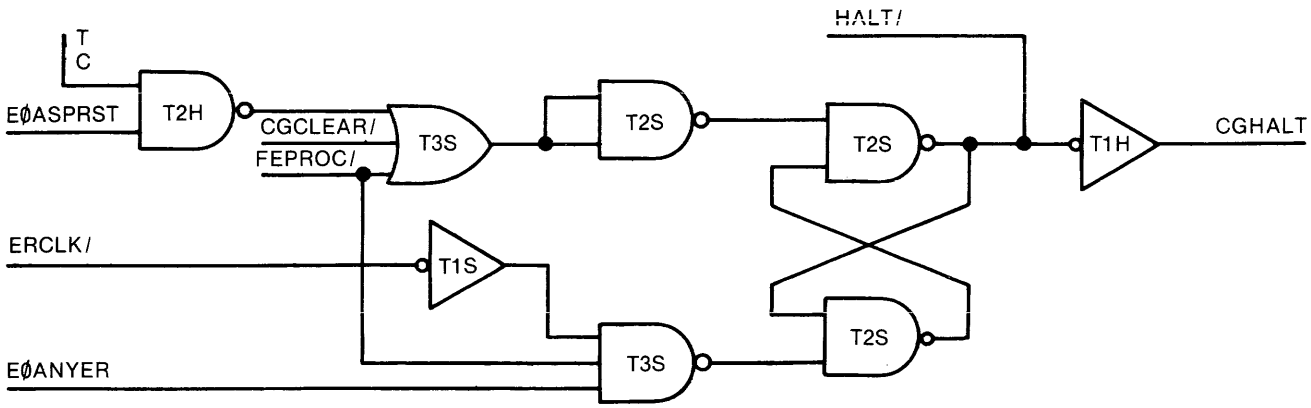


Fig. II-27 CG2 CARD HALT CIRCUIT

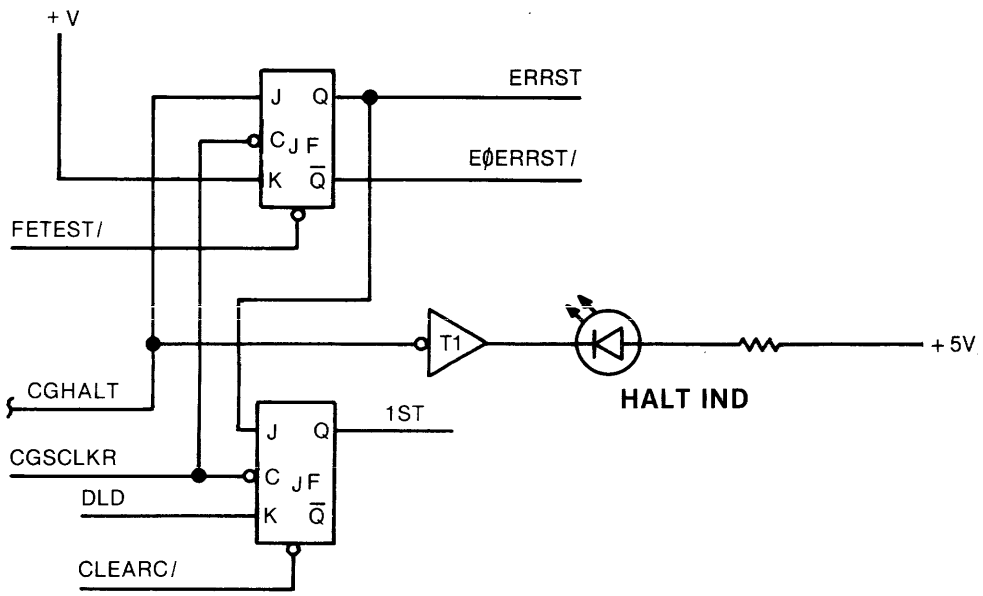


Fig. II-28 EO CARD HALT CIRCUIT

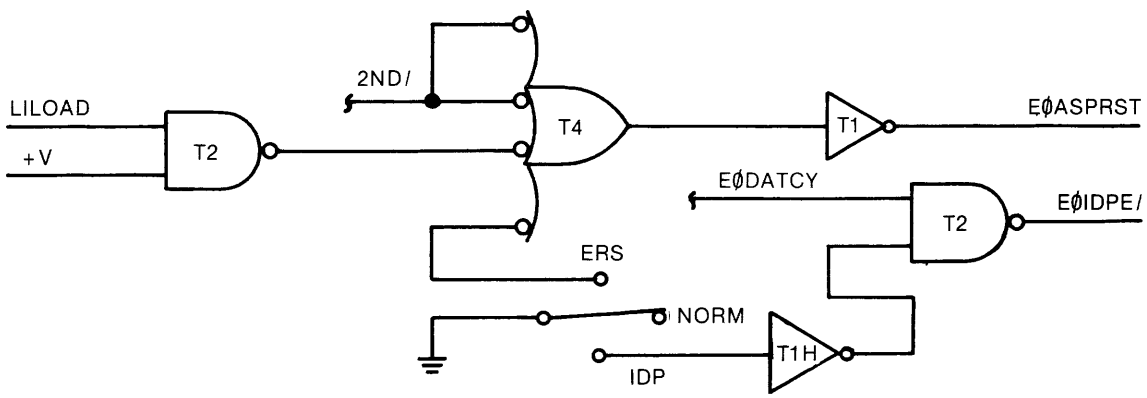


Fig. II-29 SECOND ERROR DETECTION LOGIC

CGHALT and thus inhibits SCLK from occurring until the processor CLEAR pushbutton is pressed. As shown in figure II-27, EOASPRST inhibits CGHALT from being reset if the processor is in the load mode, when a second error occurs, or the ERS-NORM-IDP switch (figure II-29) is in position ERS. This allows the hardware to halt the processor when any error is detected.

If the ERS-NORM-IDP switch is in position IDP and a memory data cycle is in process, (EODATCY goes high) signal EOIDPE/ goes low and inhibits all data memory parity errors detected on card SM5.

Timing diagram figure II-30 summarizes the major operations for one and two detected errors.

CLOCK GENERATION AND DISTRIBUTION

The clock and clear generation circuits are contained on card CG2. The clock distribution circuits are contained on cards CD1 and CD2. CD1 distributes clock pulses in the main processor backplane, and CD2 distributes clock pulses in the optional backplanes (one CD2 card for each backplane).

CLOCK GENERATION

A representation of the main logic that generates clock outputs is shown in figure II-31. Figure II-32 is a timing diagram of the clock generation and distribution operations. The CB counter (figure II-31) counts the 10-MHZ clock pulses clock pulses, and the count is distributed to the various clock circuits. The counter is preset to a count of 6 and then is allowed to count to 15 (TC). Each count of the counter is approximately 100 nanoseconds. Table II-14 lists all clock signals, their distribution destinations, and their purposes.

CLEAR GENERATION

A clear signal is provided when power is turned on, the processor CLEAR pushbutton is pressed, an error occurs, or during an F.E. memory test. Figure II-33 shows the clear circuits provided on the CG2 card.

It should be noted that, when an error occurs, EOERRST/ goes low and enables CGCLEARA and CGCLEARB to go high, but does not affect CGCLEARC. (During an error, there are some circuits that are not to be cleared.) Table II-15 lists the clear signals and their distribution.

TABLE II-14 CLOCK SIGNALS

Signal	Distribution	Purpose
CGSCLK 1	CD1	Distributed to main backplane.
CGSCLK 2	EJ11 (I/O 3)	To CD2 in I/O backplane.
CGSCLK 3	EJ10 (I/O 2)	To CD2 in data communication backplane.
CGSCLK 4	EJ10 (I/O 2)	To CD2 in future expansion backplane.
CGFECLK	EJ14 (MON 1)	
CGFECLK/	FE3	Used for processor pushbutton and single pulse pushbutton circuits.
CGSCLKR	FE4, EO2	Used during in error operations.
CGERCLK	EO2	To set error flip-flop and inhibit SCLK if error occurs.
CGTIME1/	DMD	Clock input for mini-processor.
CGTIME2/	DMD	Clock input for mini-processor.
CG10CLK1	I/O 2	10-MHZ clock to communications processor.
CG10CLK2	I/O 2	10-MHZ clock for future expansion backplane.
CG3CLKA,	MU3, MU4, MU5	To enable phase-3 operations.
CG3CLKB,	LU2-3, LU2-4	
CG3CLKC,	LU2-1, LU2-2	
CG3CLKD	LU7-1, LU7-2, LU6-1, LU6-2	

TABLE II-15 CLEAR SIGNALS

Signal	Distribution
CGCLEARA	EJ9 (I/O 1) (Optional backplanes)
CGCLEARB	IOC12, 11, 10, 9, 4, PS1-1, PS1-3, EO2, SM6, SM5, CON1
CGCLEARC	CU1, SM4, MU6-1,-2,-3,-4, FE3, EO2

As shown in figure II-33, it can be seen that any low input to the three B4 gates will cause the clear signals to go high.

POWER ON CLEAR

When the processor power ON pushbutton is pressed, signal ACPONCL/ goes low and causes the D-type flip-flop to reset on the leading edge of signal A/. When the flip-flop resets, the first JK flip-flop resets, and the next TC pulse resets the second JK flip-flop and produce clear signals A, B, and C. These signals clear the processor and the IOC's. CGPCLR/ goes low and clears various LI7 and LI8 circuits.

CLEAR PUSHBUTTON

When the CLEAR pushbutton is pressed, the two JK flip-flops reset on the next two successive TC pulses and cause signals A, B, and C to go high to clear the processor and the IOC's. CGCLLR/ goes low and clears various LI7 and LI8 circuits.

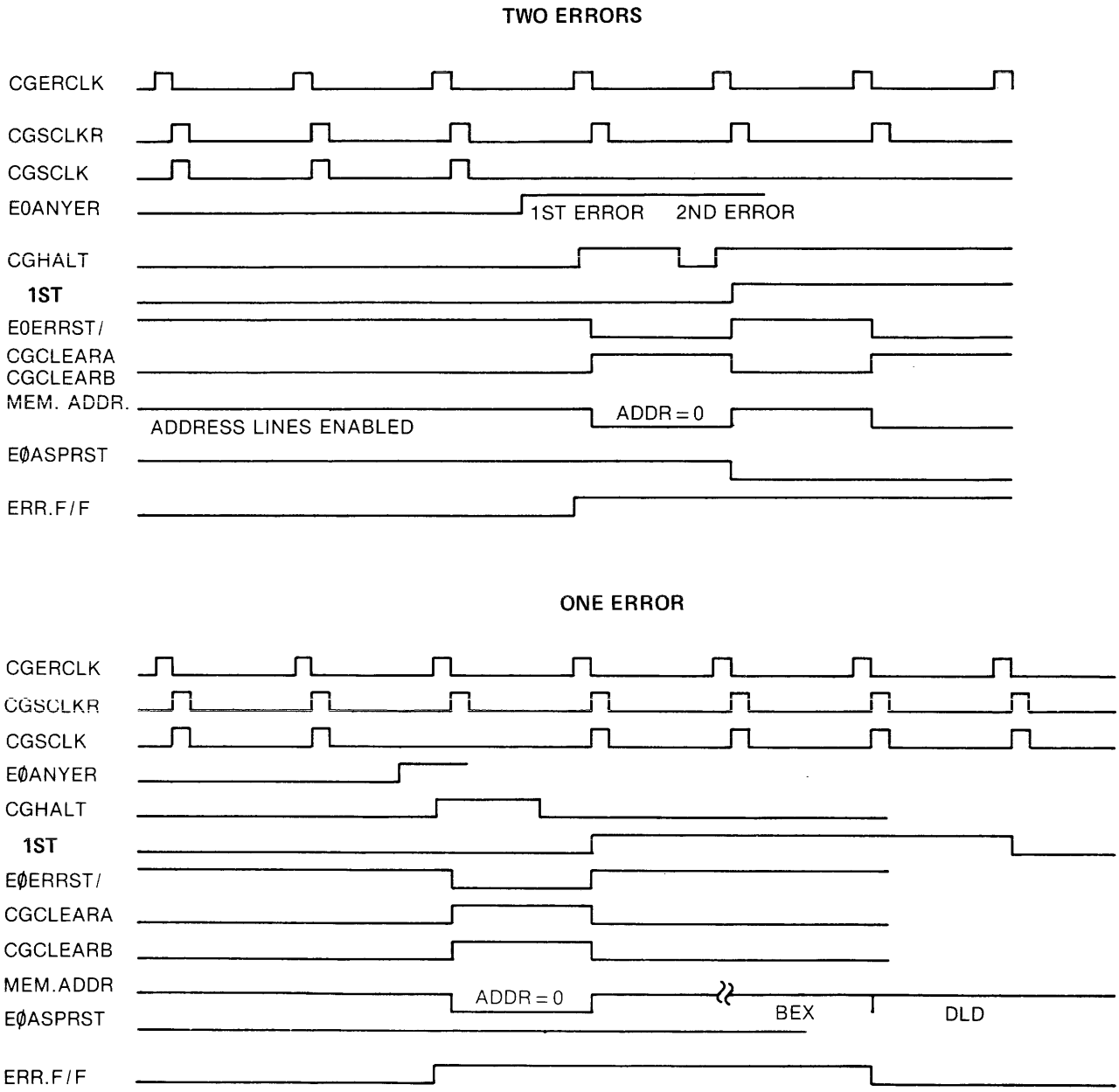


Fig. II-30 MEMORY HARDWARE ERROR TIMING DIAGRAM

ERROR CLEAR

When an error occurs, signal EOERRST/ goes low for one clock period and causes clear signals A and B to clear the IOC's and various processor circuits. CGCLEARC is not affected, because the error logic must not be cleared at this time.

CLOCK DISTRIBUTION

Cards CD1 and CD2 distribute CGSCLK to various backplane areas. Table II-16 lists the various clock outputs and their distribution.

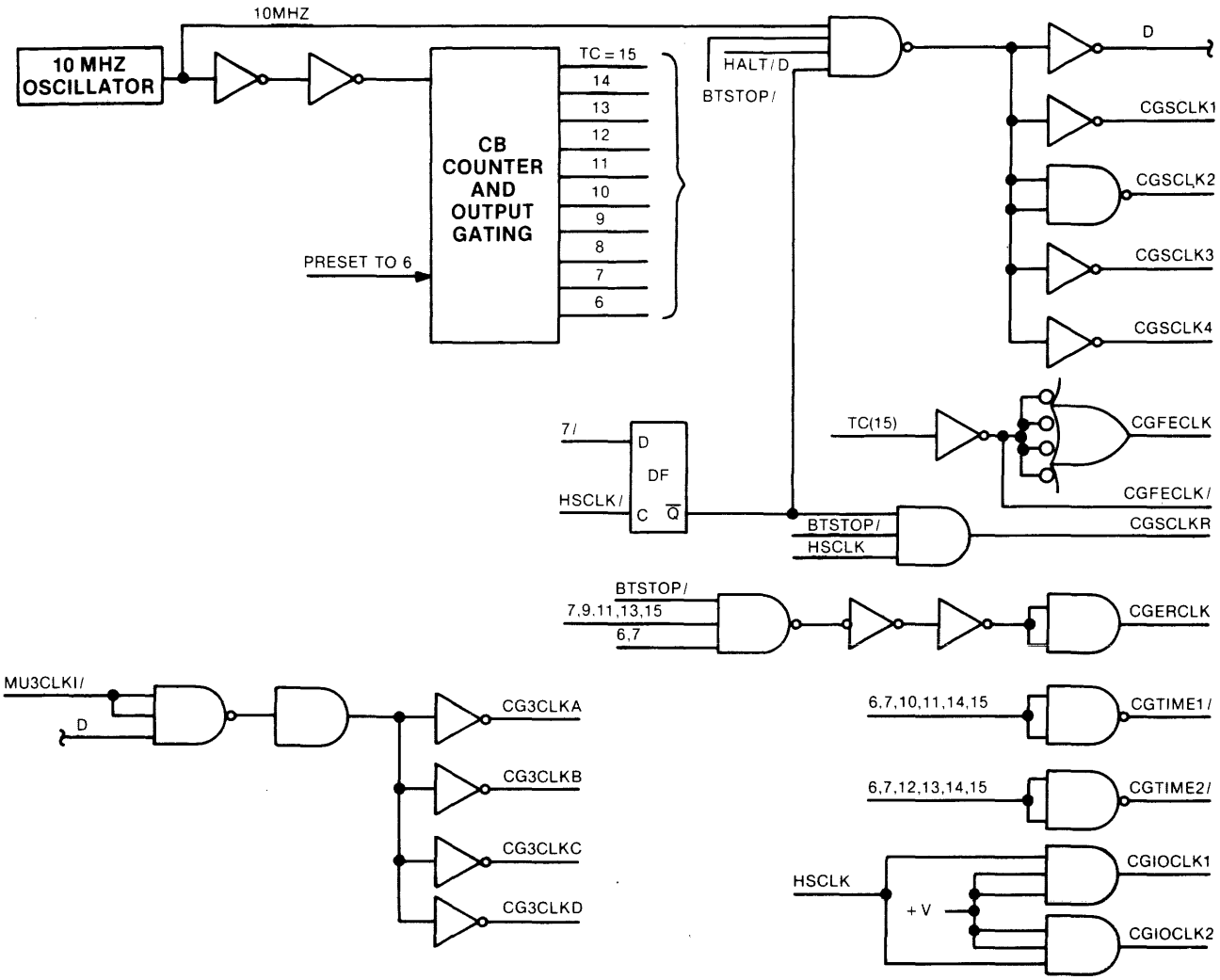
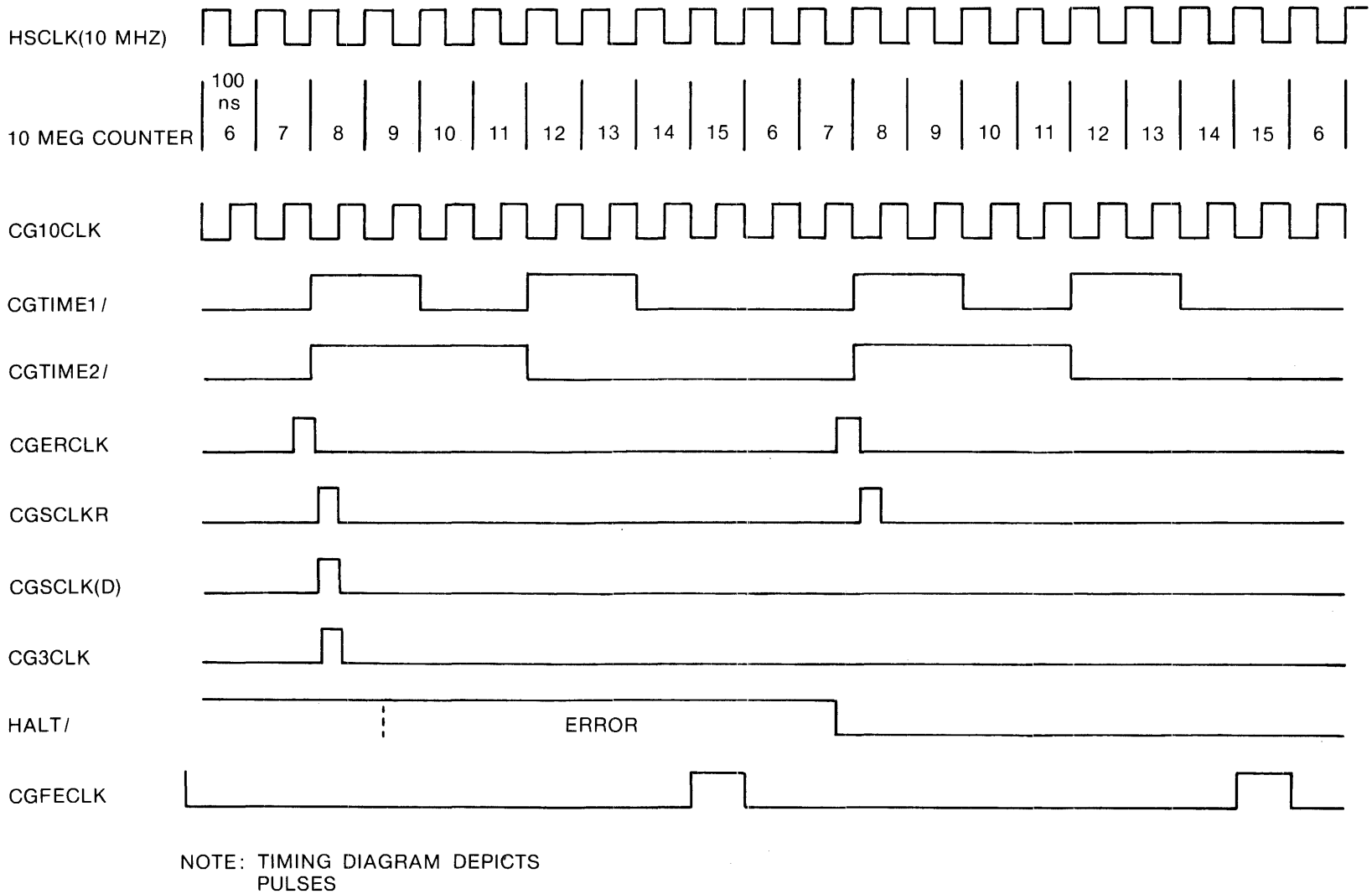


Fig. II-31 CG-2 CLOCK GENERATION CIRCUITS

TABLE II-16 CD1 AND CD2 CLOCK OUTPUTS

Card	Signal In	Signal Out	Distribution
CD1	CGSCLK1	CDSCLK A	SM4, SM6, SM7
		CDSCLK B	PS1-1, EO2, PS1, 3 IOC1
		CDSCLK D	IOC12
		CDSCLK E	LI 7, LI 8
		CDSCLK F	MU6-1, 2, 3, 4
		CDSCLK G	MU2, CU1, CU2
		CDSCLK J	FE1, 2, 3
		CDSCLK K	IOC 4 (DMD)
		CDSCLK PA	IOC 10
		CDSCLK PB	IOC 11
		CDSCLK PC	IOC 12
		CDSCLK P4	IOC 4
		CDSCLK P9	IOC 9
CD2	CGSCLK2 (I/O Backplane)	CDSCLKA	PS1-2
		CDSCLKP3	IOC3
		CDSCLKP5	IOC5
		CDSCLKP6	IOC6
		CDSCLKP7	IOC7
		CDSCLKP8	IOC8



NOTE: TIMING DIAGRAM DEPICTS PULSES

Fig. II-32 CLOCK GENERATION TIMING

Printed in U.S. America 7-30-75
For Form 1077534

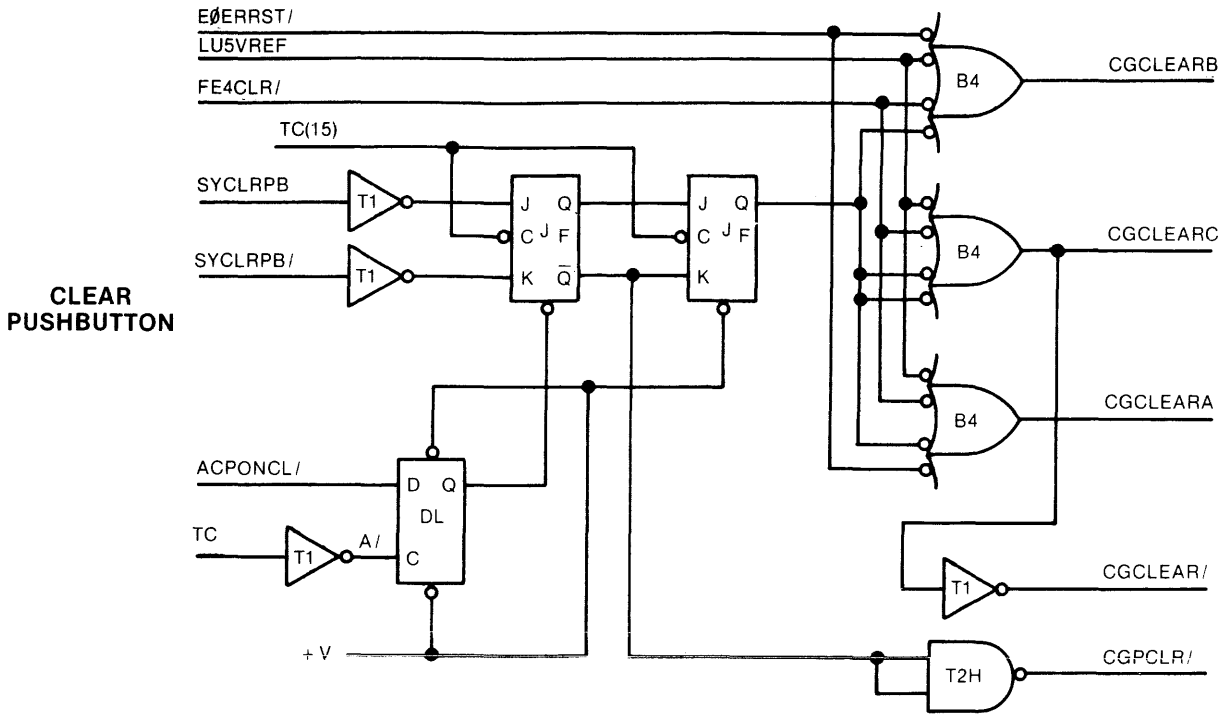


Fig. II-33 CG2 CLEAR CIRCUITS

MEMORY SECTION

Figure II-34 is a block diagram of the CPU memory section. Memory addresses and data to memory are provided by the shared memory unit, and interfacing is provided by cards MADD-1 and MADD-2. These cards transfer data and addresses to memory storage cards MPA-1 through MPA-12 and MPB-1 through MPB-12. The memory storage cards store the data and, when enabled, read the data to the processor section by interfacing with the MOB card.

MEMORY STORAGE DEVICE

Memory storage is accomplished by the use of an integrated-circuit storage device which is a static, random-access memory (RAM) configured as 1024 one-bit words and packaged in a 16 pin dual-in-line package. This integrated circuit is directly TTL-compatible on all inputs and outputs and uses a single 5-volt supply. Figure II-35 shows the functional organization of the device.

When the CHIP ENABLE (CE) input (pin 13) is in the logic high state read and write operations are inhibited, the DATA OUT output (pin 12) is forced to the tri-state condition, .thus providing the wired OR-tie of similar devices. When CE is in the logic low state, the mode of operation is determined by the read/write signal (R/W) at pin 3.

Data is non-destructively read out at pin 12 (DATA OUT) of the selected storage cell when R/W is in the logic high state. Selection of a storage cell for reading or writing is determined by the binary value of the ten address inputs, A0 through A9. Data present at pin 11 (DATA IN) is written into the selected storage cell when R/W is the logic low state and appears noninverted at pin 12.

MEMORY ADDRESS AND DATA DRIVER CARDS

Memory address and data driver cards MADD-1 and MADD-2 interface the memory address and data from the processor to the memory storage modules. Figure II-36 shows the MADD-1 and MADD-2 circuits.

Data to memory is transferred from the shared memory unit over lines SMDT01 through SMDT16. Bit 17 comes from the loader interface as signal LIDTM17. Data to memory is gated directly to the memory modules through MADD-1 and MADD-2 on signal lines MADI01 through MADI17. The memory

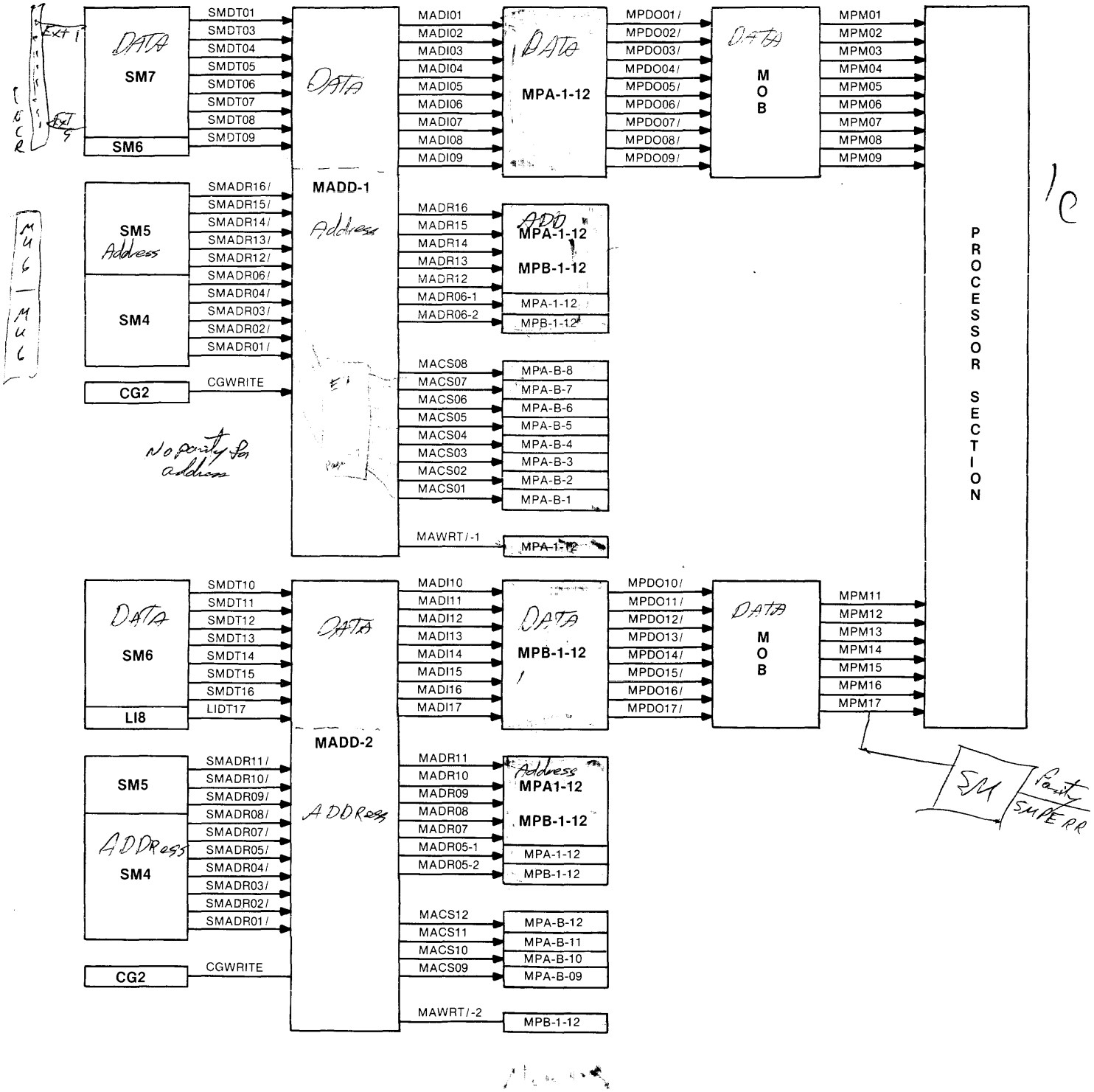


Fig. II-34 MEMORY SECTION BLOCK DIAGRAM

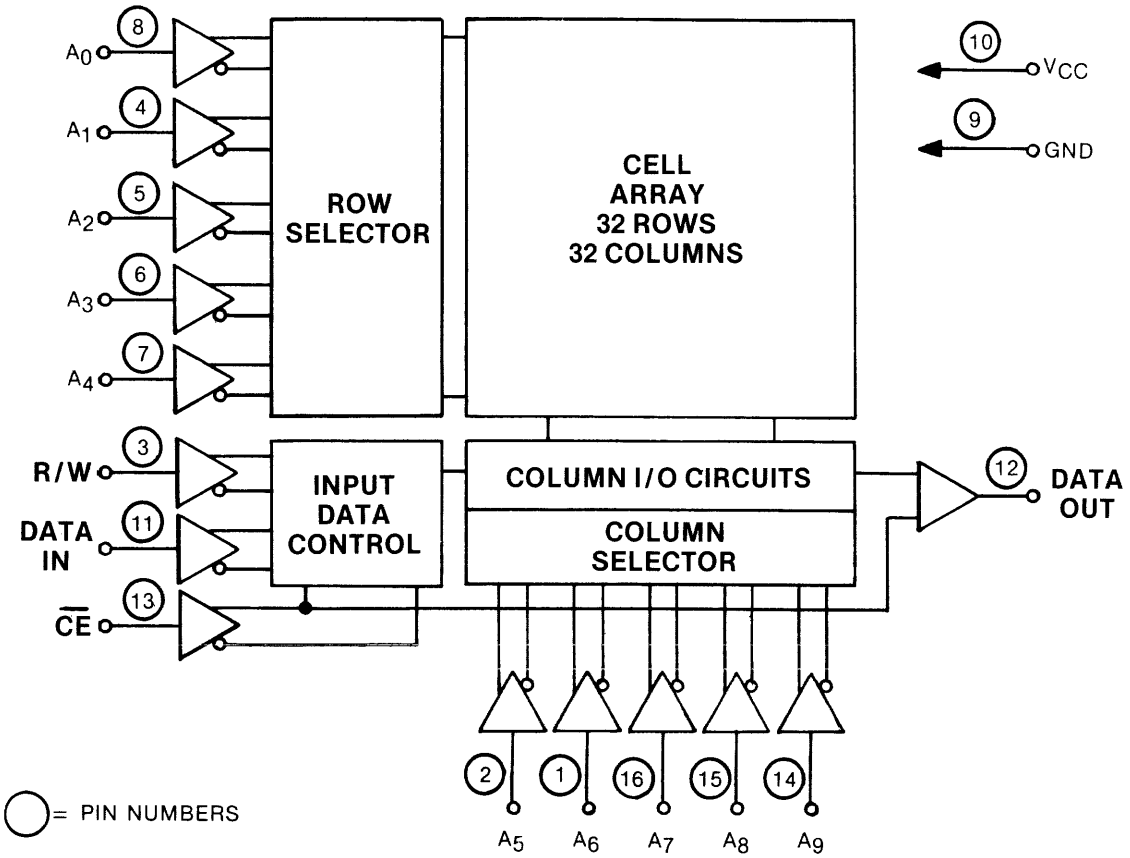


Fig. II-35 MEMORY STORAGE DEVICE BLOCK DIAGRAM

address also comes from the shared memory unit on signal lines SMADR01/ through SMADR16/. Memory address lines SMADR16/ through SMADR05/ are inverted on the MADD cards, and outputs MADR16 through MADR05 are sent to the memory modules.

Memory address lines SMADR04/ through SMADR01/ are decoded on the MADD cards to select 1 of 12 memory modules to be read from or written into. This is accomplished by using two, 3-line to 8-line decoder integrated circuits. (E7 is the 3-to-8 decoder on MADD-1 and MADD-2.)

The 3-to-8 decoder on MADD-1 decodes the binary input on signals SMADR04/, SMADR03/ and SMADR02/ when SMADR01/ is high. The G2A and G2B inputs are grounded; thus E7 is enabled only when G1 is high. The outputs of E7 are inverted to produce signals MACS01 through MACS08. These signals, when high, enable memory modules 1 through 8.

The 3-to-8 decoder on MADD-2 decodes the binary input on signals SMADR04/, SMADR03/ , and SMADR02/ when SMADR01/ is low. Input G1 is always high because there is no connection to it. Because G1 is always high, the 3-to-8 decoder is only enabled when SMADR01/ is low. The outputs of this line decoder are inverted to produce signals MACS09 through MACS12, which when high, enable memory modules 9 through 12.

MEMORY STORAGE MODULE

A 4K memory storage module consists of two circuit cards, MPA and MPB. MPA stores data bits 01 through 09, and MPB stores data bits 10 through 17. MPA and MPB cards contain four 1-by-1024 read-write storage integrated circuits for each bit, plus associated logic. Figure II-37 shows the logic for one bit on the MPA card and represents all bits on MPA and MPB.

MADR05-1 and MADR06-1 are decoded to select one of four 1-by-1024 read/write circuits for each of the 17 bits. Address lines MADR16 through MADR07 address 1-of-1024 addresses of the circuit that is enabled. To enable data to be read from the module, signal MACS (1 through 12) must be high. For data to be writ-

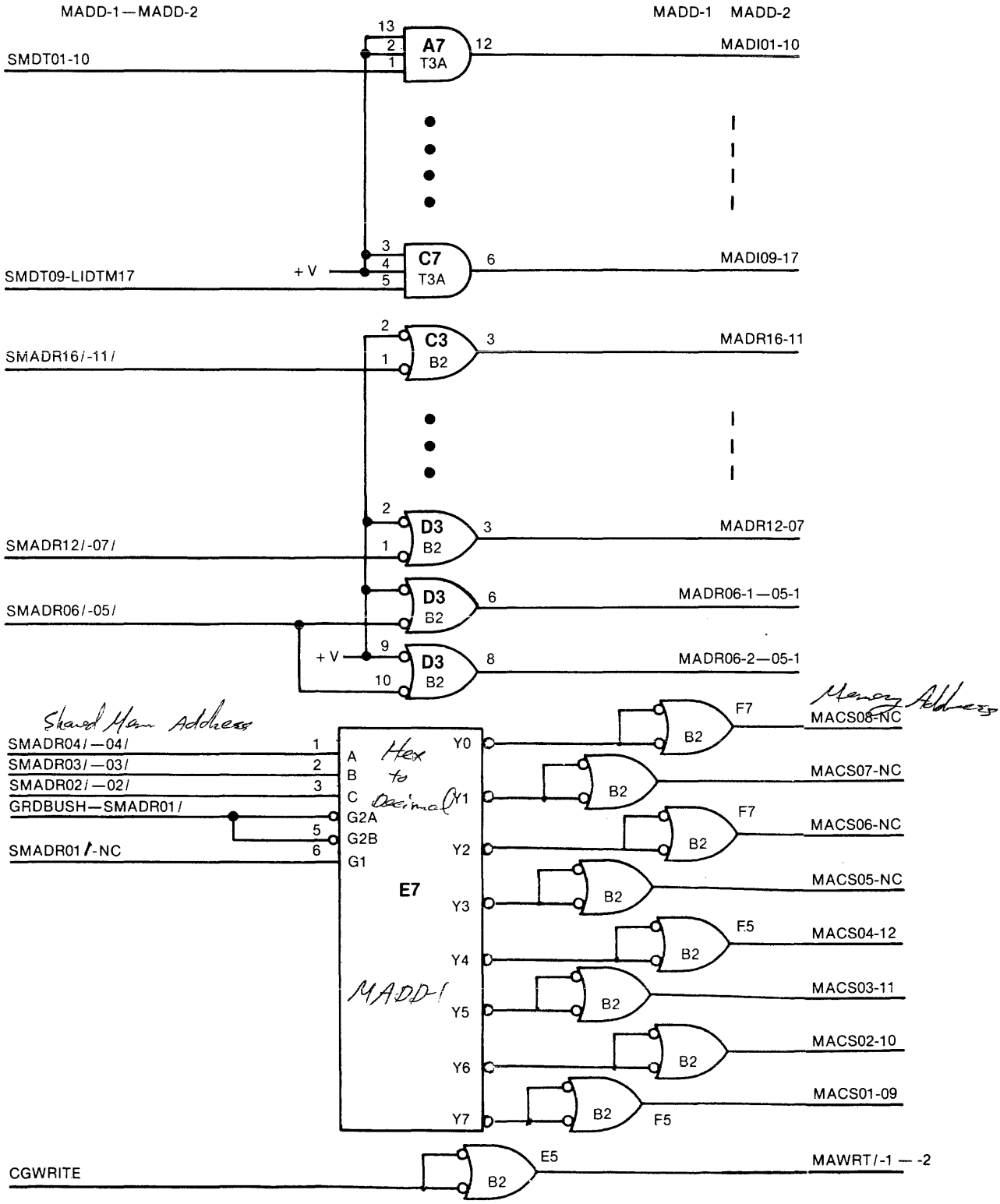


Fig. II-36 MADD-1 AND MADD-2 CIRCUITS

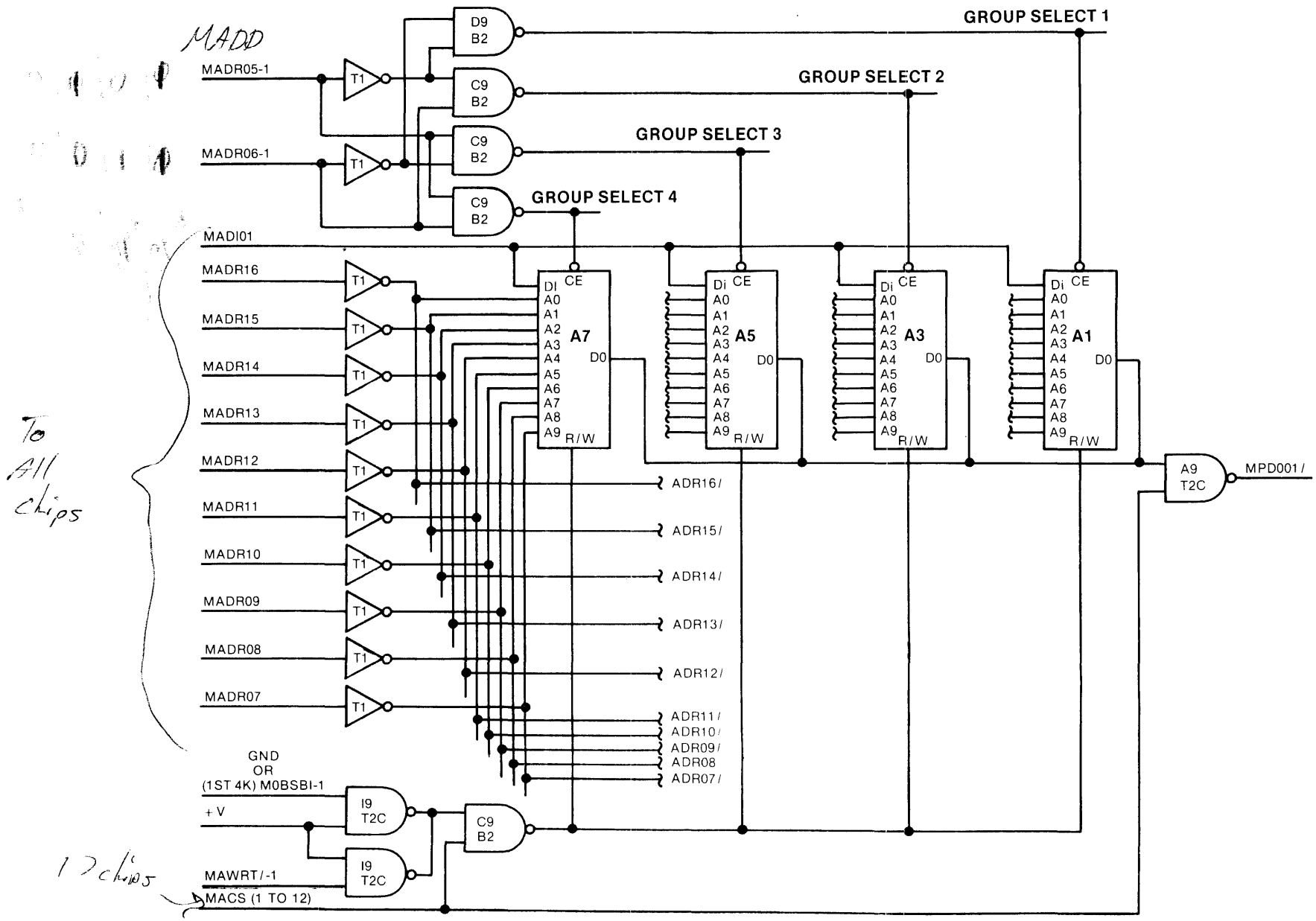


Fig. II-37 MEMORY STORAGE MODULE LOGIC ON MPA AND MPB (ONE BIT)

ten into the circuit signal MACS must be high, along with a low signal MAWRT/1. This enables the circuit to write, into the selected address, a 1 or 0, depending on the state of input DI.

Data out from the MPA and MPB cards goes to memory output buffer card MOB.

MEMORY OUTPUT BUFFER CARD

MPA and MPB card output data goes to the MOB card, which inverts the MPD001/ through MPD001/17 signals and buffers them to produce MPM01 through MPM17. MOB also produces signal MOBSBI-1, which is used only on the first 4K memory module to inhibit writing during power turnon or turnoff when in the standby mode. Standby power is used by the F.E. to save the MTR that is loaded into the first 4K of memory when it is necessary to turn off power when removing a circuit card.

DATA PARITY GENERATION AND CHECKING

PARITY GENERATION FOR MEMORY WRITE OPERATION

As shown in figure II-38, odd parity is generated on the shared memory boards in the processor. Parity is generated from the SMDTM01/ through SMDTM16/ data signals while data is being transferred to memory.

SMDTM01/ through SMDTM08/ are gated by exclusive OR gates A through G to make up signal SMPGEN1. SMDTM09/ through SMDTM16/ are gated by exclusive OR gates H through P to produce signal SMPGEN2. SMPGEN1 and SMPGEN2 are inputs to exclusive OR gate R, whose output is inverted through NAND gate S on an LI8 board. The resultant output is LIDTM17, which is the parity bit data bit-17 sent to memory.

If the number of 1-bits is even, both SMPGEN1 and SMPGEN2 are either high or low. Identical inputs to exclusive OR gate R produce a low output. This low output is inverted by NAND gate S to make LIDTM17 high and add a 1-bit to the data. Thus, the total number of 1-bits is odd.

If the number of 1-bits is odd, SMPGEN1 and SMPGEN2 are opposite in state (one is high and the other is low). Opposite inputs to exclusive OR gate R produce a high output. This output is inverted by NAND gate S and makes LIDTM17 low, leaving the total number of 1-bits odd.

PARITY CHECKING FOR MEMORY READ OPERATION

As shown in figure II-39, parity is checked on the shared memory boards in the processor while data is being transferred from memory into the processor. Parity is checked by generating new parity from MPM lines 01 through 16 in the same manner as the parity bit is generated for memory write operation. The new

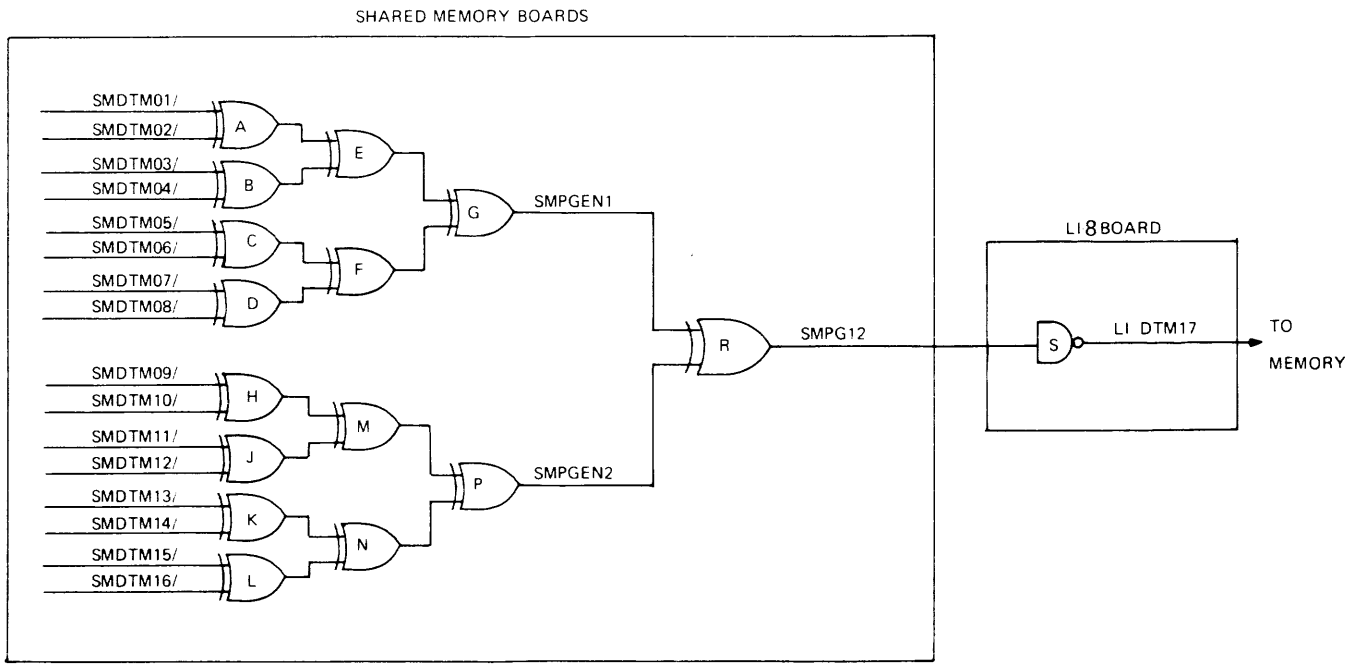


Fig. II-38 DATA PARITY GENERATION

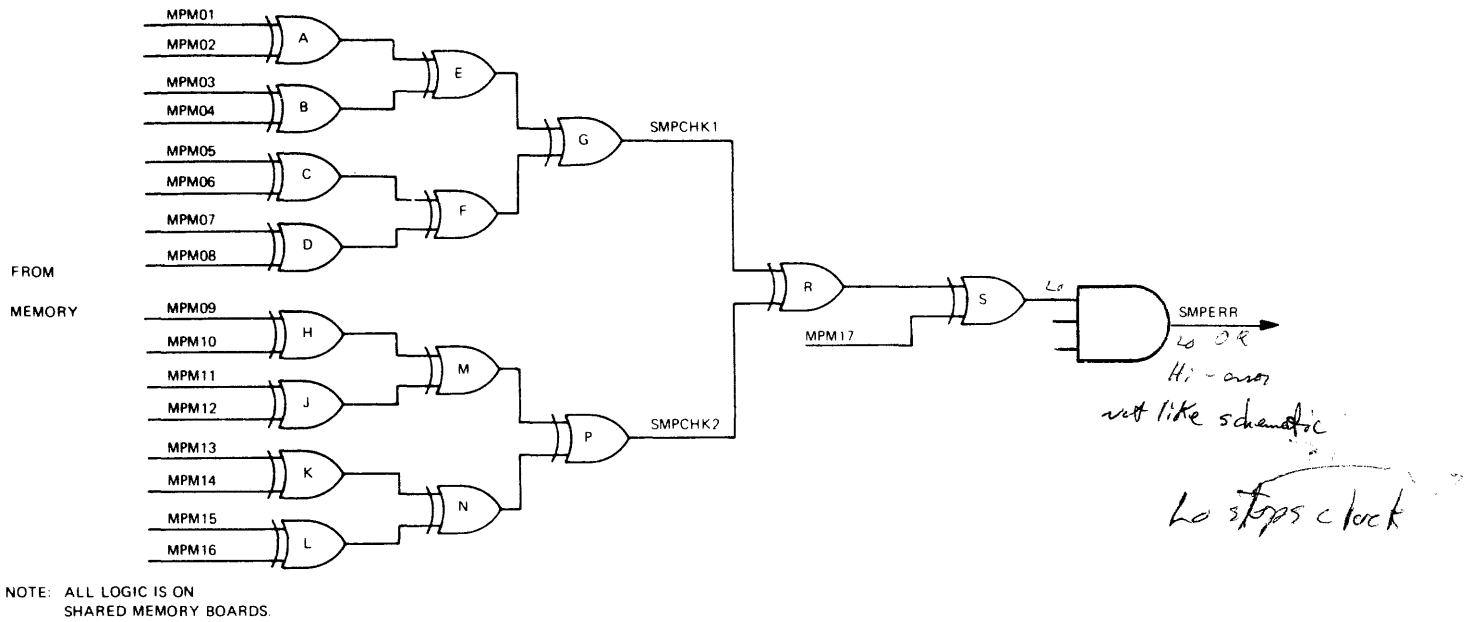


Fig. II-39 PARITY ERROR DETECTION

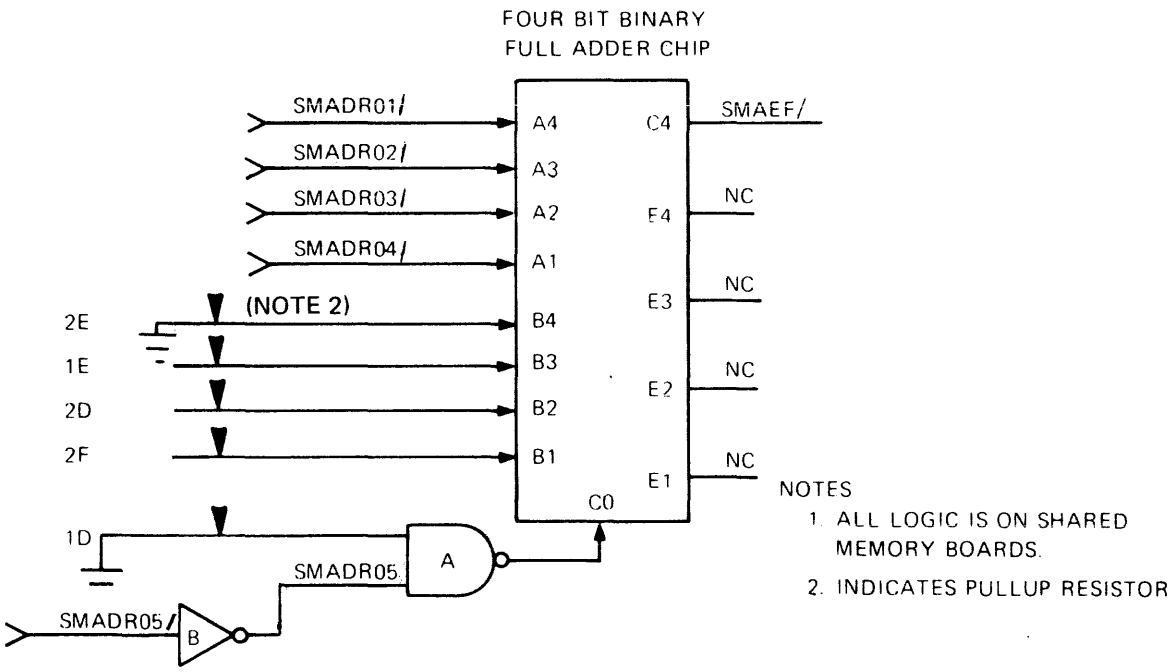


Fig. II-40 ADDRESS ERROR DETECTION

parity is compared to the parity bit read from memory to determine if it is correct.

MPM lines 01 through 08 are gated by exclusive OR gates A through G to produce signal SMPCHK1. MPM09 through MPM16 are gated by exclusive OR gates H through P to produce signal SMPCHK2. SMPCHK1 and SMPCHK2 are inputs to exclusive OR gate R, whose output is gated by exclusive OR gate S with the parity bit from memory (MPM17).

The output of exclusive OR gate S is inverted and is one input to a three-input NAND gate.

When there is no parity error, the output of gate S is high which will cause the output of the NAND gate to be high; thus signal SMPERR will go low. The other two inputs to the NAND gate are enabled signals. These two inputs must be high to detect a parity error. The following are conditions that inhibit parity detection:

a. Memory Write Operation. During a memory write operation, parity is not to be detected. Thus, signal EO write is high, causing the NAND output to go high, no matter what the output of gate S is.

b. Memory Load Operation. During a memory load operation, parity is not to be detected until a memory cycle takes place or during Load Clear (LIPECLR/). When a memory cycle takes place, the parity detector is enabled.

c. Inhibit Data Memory Parity Error. If EO2 switch S1 is in position IDP, data memory parity errors are disabled by signal EOIDPE/ being low.

If MPM01 through MPM16 are an even number of 1-bits, SMPCHK1 and SMPCHK2 are both high, or both low. Identical inputs to exclusive OR gate R produces a low output. If the parity is correct, MPM17 should be high (1-bit). The high and low through exclusive OR gate S produce a high output causing SMPERR to stay low. A low input for producing a parity error (SMPERR is true) can be produced in two ways. First, if the parity bit was bad (MPM17 is low), the two low inputs into exclusive OR gate S result in a low output. Thus, signal SMPERR goes true. Second, if an odd number of bits were picked up or dropped on MPM lines 01 through 16, SMPCHK1 and SMPCHK2 would be opposite inputs to exclusive OR gate R and produce a high output. This high is gated into exclusive OR gate S with the high on MPM17 and produces a low output from gate S thus, signal SMPERR goes true.

If MPM lines 01 through 16 have an odd number of 1-bits, SMPCHK1 and SMPCHK2 are opposites. The output of exclusive OR gate R is high, and MPM17 is low (0-bit) if parity is correct. The low and high input to exclusive OR gate S produces a high which keeps SMPERR high.

A parity error occurs if the parity bit (MPM17) is incorrect. (MPM17 should be low, but is high.) The two highs into exclusive OR gate S produce a low output to make SMPERR true. Also, if an odd number of bits were picked up or dropped, SMPCKH1 and SMPCHK2 would be alike. The output of exclusive OR gate R is low, and because MPM17 is low into exclusive OR gate S, a low output is produced to again make SMPERR true.

MEMORY ADDRESS ERROR DETECTION

Address errors are checked for on the processor shared memory boards. Address errors occur when the address exceeds the amount of memory installed in the system. The address lines are compared to a hard-wired backplane address limit. This limit is wired during installation of the processor and must be changed whenever a memory module is added or removed. (Refer to Section VI.)

Figure II-40 shows the address error detection scheme. The comparison is done by a four-bit binary full adder chip, which is capable of adding two hexadecimal digits and handling all internal carries during an add operation. Shared memory address lines 01/ thru 04/ comprise 1-digit input to the adder chip on pins A4, A3, A2, and A1. The other digit input on B4, B3, B2, and B1 is the result of the address limit installed on the backplane. C0 is the carry input to the chip and is controlled by shared memory address line 05/ and the backplane wired limit. E4, E3, E2, and E1 (binary bit outputs) are not used. C4 is the most significant carry output from the addition of the two digits. If a carry is produced from the addition of the address lines and the wired limit, C4 is high, and the shared memory address error signal (SMAEF/) is high, indicating no error.

As shown in the following example, the backplane is wired for 32 K-bytes, or a 16 K-word capacity, and has a maximum hexadecimal address of 3FFF. An example of

addition using the maximum address is as follows:

EXAMPLE 1

		1 1 1 1	Intrnl Carries
A4,A3,A2,A1	=	1 0 0 0	28K WDS
B4,B3,B3,B1	=	0 1 1 1	Bckpln (32K WDS)
		1	C0 is high.
	C4 = 1	0 0 0 0	

In this example, the comparison of the maximum address with the backplane limit by addition does not generate an address error because a C4 is 1.

An example of addition using an error address is as follows:

EXAMPLE 2

		1 1 1	Intrnl Carries
A4,A3,A2,A1	=	1 0 1 1	33K WDS
B4,B3,B2,B1	=	0 1 1 1	Bckpln (32K WDS)
		1	C0 is high.
	C4 = 0	0 1 1 1	

In this example, the address exceeded the backplane limit and, as a result of the addition, C4 equals a 0. Thus SMAEF/ goes low, indicating an error.

MEMORY LOADER SYSTEM

The memory loader system consist of a photoelectric paper tape reader, a memory loader logic card, and loader interface cards. Figure II-41 is a block diagram of the memory loader system. Data flows from the reader through the memory loader logic card to the loader interface cards, where the data is accumulated to be written into memory.

MEMORY LOADER LOGIC

Memory loader logic card MLL interfaces the reader with loader interface cards LI7 and LI8. Figure II-42 shows the basic logic of the MLL card. Signal KBENF/ goes low when the processor NORMAL-LOAD pushbutton is switched to LOAD and the processor CLEAR pushbutton is pressed. When KBENF/ goes low, signal RELARM goes high and enables the reader motor to drive the tape through the cartridge. A low KBENF/ also sets the ENA latch on the first strobe; the latch remains set until KBENF/ goes high. The ENA latch enables the outputs of the data latches to the loader interface.

Signal TTRAK is the output of the photocell under the feed hold of the paper tape. This signal is used to strobe data into the loader interface and also to reset the data latches before each character is loaded into the data latches. As shown in timing diagram figure II-43 (and figure II-42), when signal TTRAK goes high, it causes the Darlington driver to turn on. The collector voltage drops from 5 volts to 0 volts. As the 5-volt source is falling, the cutoff value of the M1 inverter at point A (figure II-42) is reached, and the high M1 output causes the output of inverter M1 at point B to go low. The RC network associated with this inverter keeps the input of the M1 inverter at point C at a high level.

When the inverter at point B goes low, the input to the inverter at point C goes low until the capacitor charges. (The RC time constant is approximately 200 microseconds.) This 200-

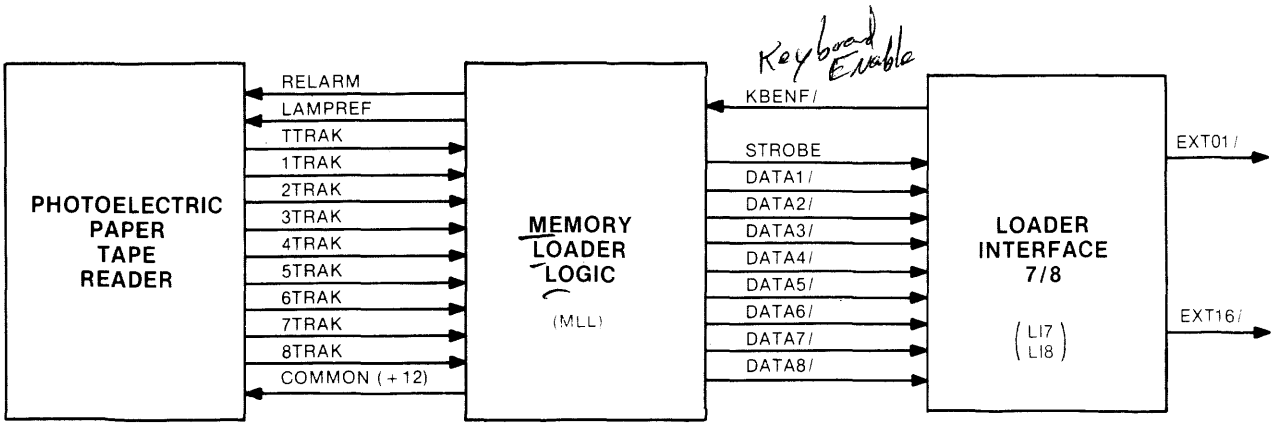


Fig. II-41 MEMORY LOADER SYSTEM BLOCK DIAGRAM

microsecond pulse is inverted and resets the eight data latches. When TTRAK goes low, it turns off the Darlington driver and causes the collector to rise from 0 volts to 5 volts.

When the turnon value of the M1 inverter at point A is reached, its output goes low and causes the input to the M1 inverter at point D to go low until the capacitor charges. (The RC time is approximately 300 microseconds.) The resultant 300-microsecond is inverted and produces a positive pulse to the load interface to strobe in the data that is present on the DATA1/ through DATA8/ lines.

Each of the eight channels on the tape produces an output from the photocell when a hole is detected. This output turns on the driver in the MLL card and sets the data latch.

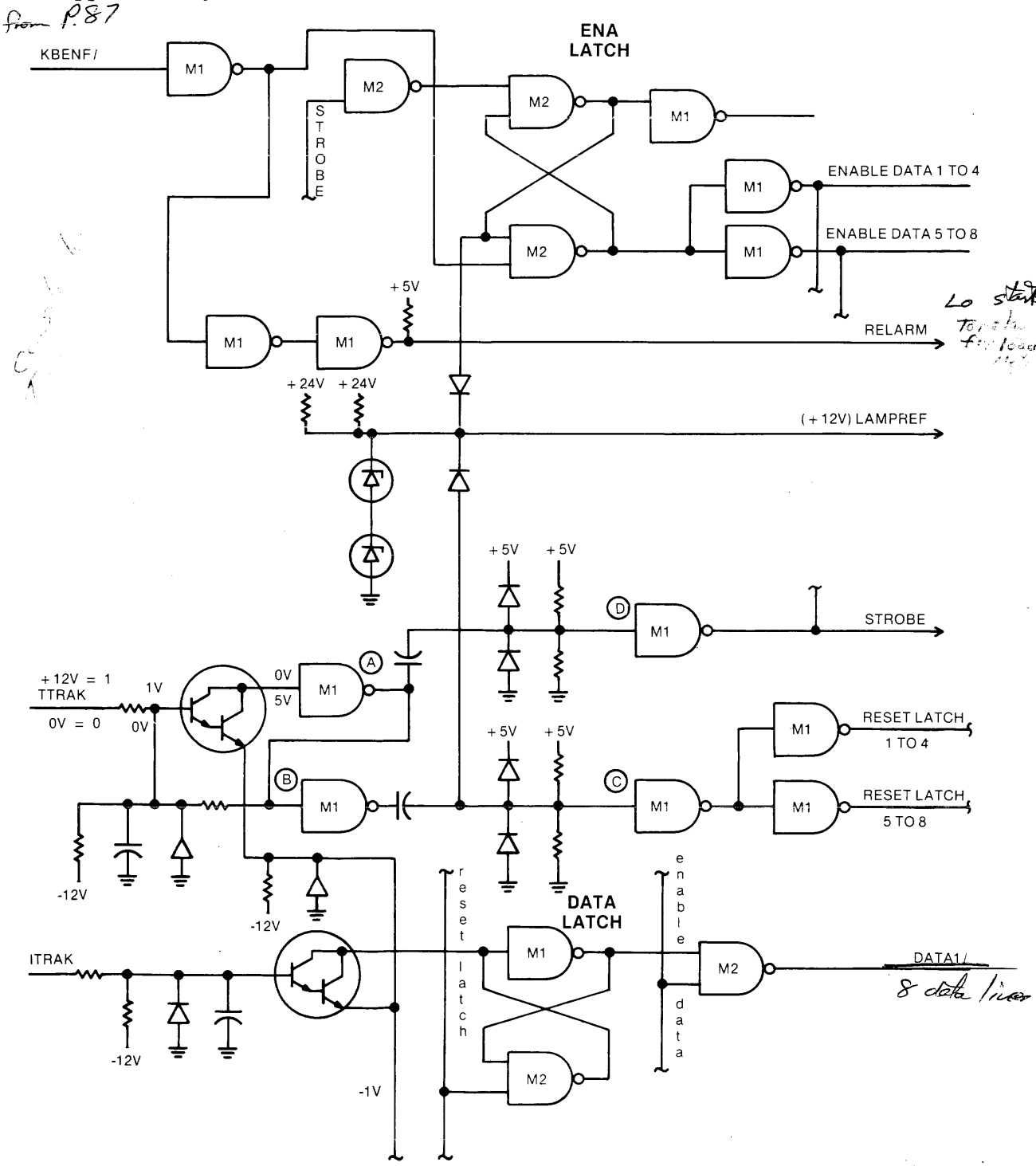


Fig. II-42 MLL CARD LOGIC

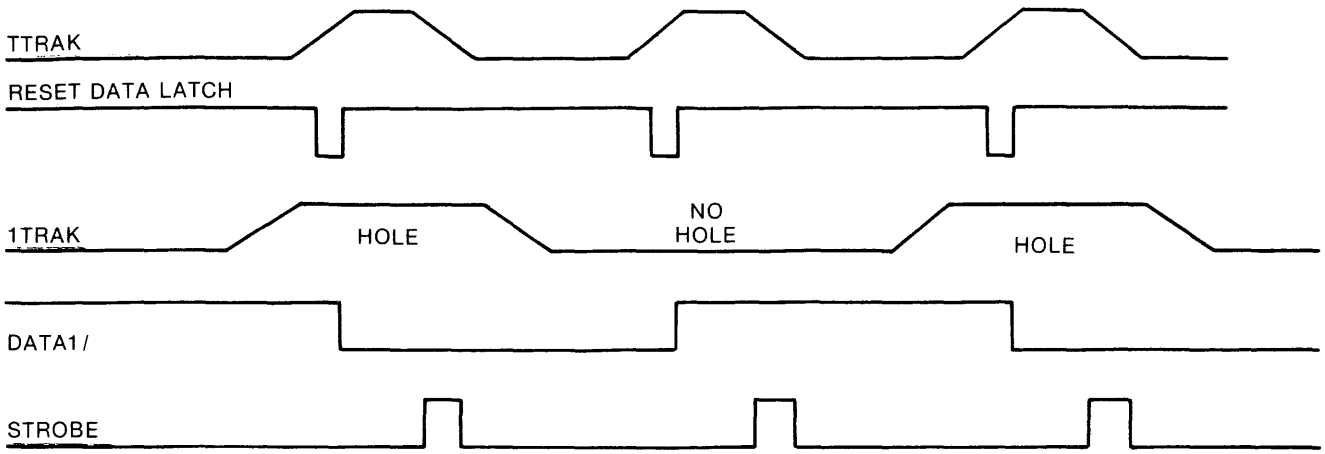


Fig. II-43 MEMORY LOADER TIMING

MEMORY LOADER ASSEMBLY

The memory loader assembly (figure II-44) consists of a removable self-threading tape cartridge, a tape drive, and a reading device. The tape cartridge mechanism feeds a program tape through the memory loader. Drive for tape feeding is supplied by the feed motor. Holes in the program tape are detected by a photocell (light-sensitive transistors) assembly.

TAPE DRIVE AND READER

When memory loader switch S1 is set to the ON position, and the processor LOAD-NORMAL switch is in the load mode, a ground return is supplied to energize relay K1. When K1 is energized, feed motor B1 and lamp DS1 are turned on.

The feed motor turns the gear-driven feed sprocket in the tape cartridge in a clockwise direction. The tape is driven through the tape cartridge at approximately 80 codes (characters) per second.

The light from DS1 is focused through a lens and is deflected 90 degrees by a mirror into the side of the tape cartridge. A photocell (light-sensitive transistor) assembly detects the light directed through the tape cartridge. There is a photocell for each hole position, including the feed hole in the program tape. Figure II-45 is a schematic of the photocell assembly.

TAPE CARTRIDGE (FIGURE II-46)

The gear-driven feed sprocket, rotated by the feed motor, feeds tape through the tape cartridge at a rate of approximately 80 codes per second. Tape inserted in the lower slot is then driven by the feed sprocket out through the upper slot of the tape cartridge.

The tape guide in the bottom of the lower slot positions the tape in the tape cartridge. A hole in the tape guide for each hole position in the tape prevents light from being reflected between photocells. Light entering the side of the tape cartridge is deflected 90 degrees by a prism. The prism directs the light down on the tape, through the holes in the tape and tape guide to the photocell assembly.

MEMORY LOAD OPERATION (LOADER INTERFACE)

The memory load operation involves the reading of paper tape and, after detecting a start code, accumulating two rows of paper tape data (16 bits), and starting a memory-write operation. The LRC of each row of data is also accumulated to be checked with the LRC character read from tape. This process is continued until a stop code is detected or an LRC error is detected. Figure II-47 shows the paper tape format.

An optional light check is available if the light check code (40, FF) is present on the tape. The light check code checks that all photocells will detect a hole. If the check fails, the load operation is stopped.

TAPE LIGHT CODE CHECK

To ensure that all photocells of the memory loader are in proper operating condition, a check is made for a code 40, followed by a code FF on the tape being loaded. A tape not containing code 40 will not enable the circuitry and thus will bypass the check.

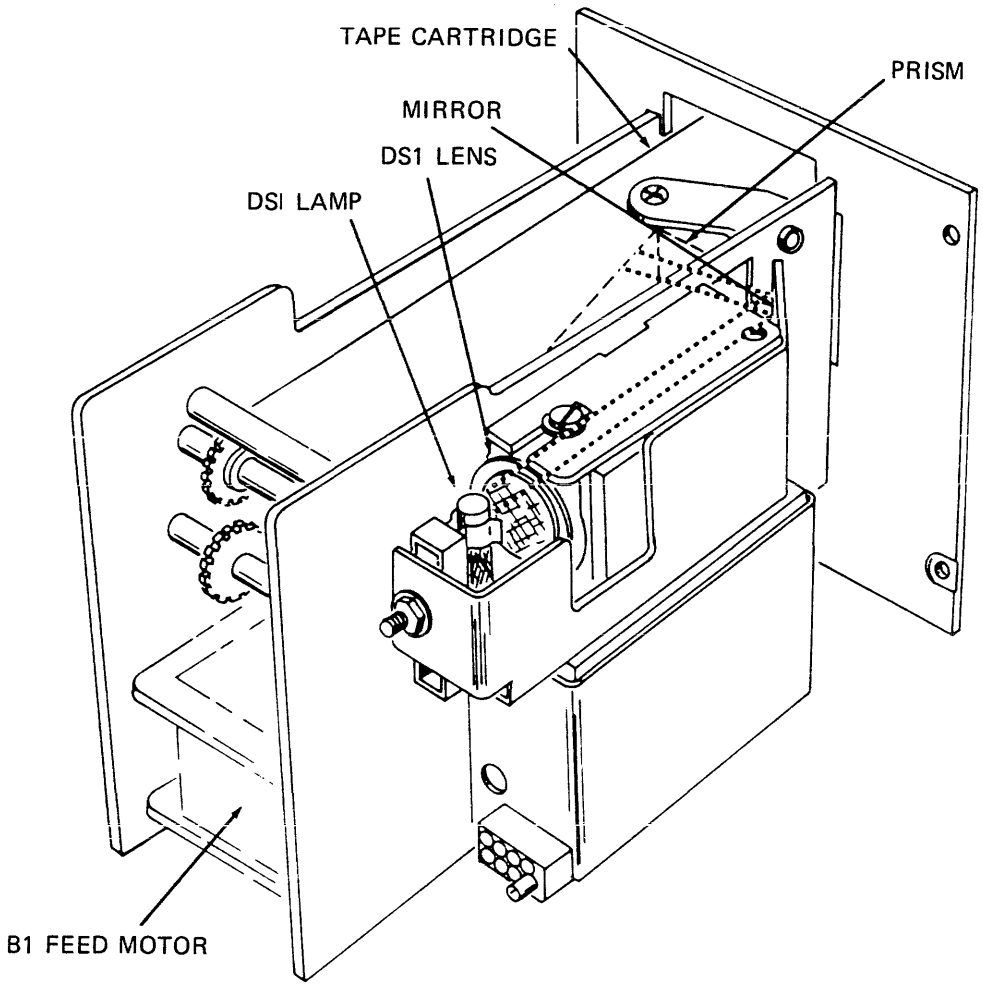


Fig. II-44 MEMORY LOADER

To simplify the description, figure II-48 shows only the logic needed to operate the light code check. Timing diagram figure II-49 includes only those timing signals needed to operate the logic associated with the light code check. Notations 1 to 23 are in reference to signals on the timing diagram. The tape is started through the memory loader by going into the load mode, which causes LIDSE/ (1) to go low, and by pressing the CLEAR pushbutton (2). This causes LILOAD (3) to go high and causes KBENF/ (4) to go low. A low KBENF/ enables MLL signal RELARM to go high and enables the memory loader-on circuitry. As the tape passes over the photocells, each feed hole generates a strobe (6). When the strobe is high, the character timing logic is enabled.

The character timing logic provides the timing to perform the various functions required for each tape character. This logic is configured with four D-type flip-flops (A, B, C and D). By pressing the CLEAR pushbutton, all four flip-flops are reset. The next three clock pulses set, in turn, B, C and D. The timing logic is now present awaiting a strobe. Strobe 6 goes high, sets flip-flop A and enables signals LIENSF (8) and LILSAB (7) to go high. The next clock pulse resets flip-flop B and causes signal LILSAB (7) to go low and signal LISB/ (10) to go high. The next clock pulse resets flip-flop C and causes signal LIENSF (8) to go low and signal LISNCD (9) to go high. The next clock pulse resets flip-flop D and causes signal LISNCD (9) to go low. This completes the character timing.

While the character timing is started by the tape strobe, data associated with the strobe is enabled on the DATAL/ through DATA8/ lines. (See figure II-48.) This data is decoded to produce LIDET4 (19) and LILODET (20). LIDET4 goes high when the upper character contains a 4, and LILODET goes high when the lower character contains a 0.

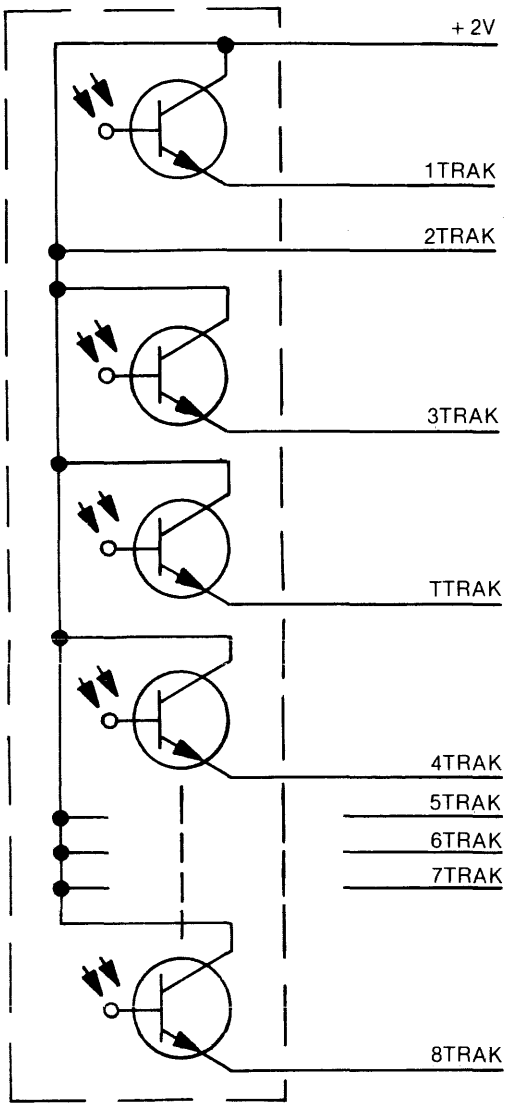


Fig. II-45 PHOTOCELL SCHEMATIC

When LIDET4 and LILODET are both high, the LC JK-type flip-flop is set to enable the light check logic when signal LISNCD goes low. The character timing is preset again when the strobe goes low, awaiting another strobe. The next character moves over the photocells, causing another strobe to start the character timing.

Flip-flop A is set to cause signal LILSAB (7) to go high. The next clock pulse resets flip-flop B and causes signal LILSAB flip-flop to go low (7) and clock the data on the DATA 1 to DATA 8 lines into eight flip-flops. All eight flip-flops should be set, because an "FF" char-

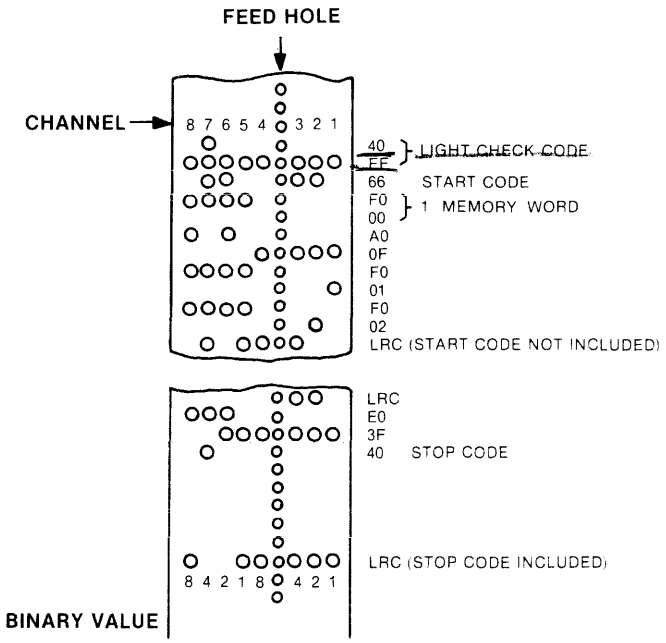


Fig. II-47 TAPE FORMAT

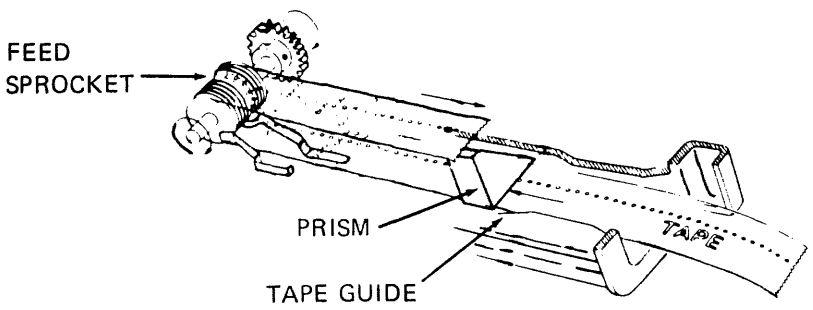


Fig. II-46 TAPE CARTRIDGE

acter was present on the tape. Flip-flop C resets on the next clock pulse, causing signal LISNCD (9) to go high and, on the next clock pulse, flip-flop D resets, causing LISNCD (9) to go low and clock the "LTBD" J type flip-flop. The "LTBD" flip-flop will be set if all of the eight flip-flops are not set, (causing the output of an eight-input NAND gate to go high). If the flip-flop is set, signal LILTBD/ (22) will go low and cause KBENF/ (4) to go high and turn off the memory loader. (See figure II-48.) LILIERR/ (23) will go low and cause a memory loader error to be displayed on the EO-2 card. If the flip-flop does not set, LILTBD stays high and allows the tape to continue.

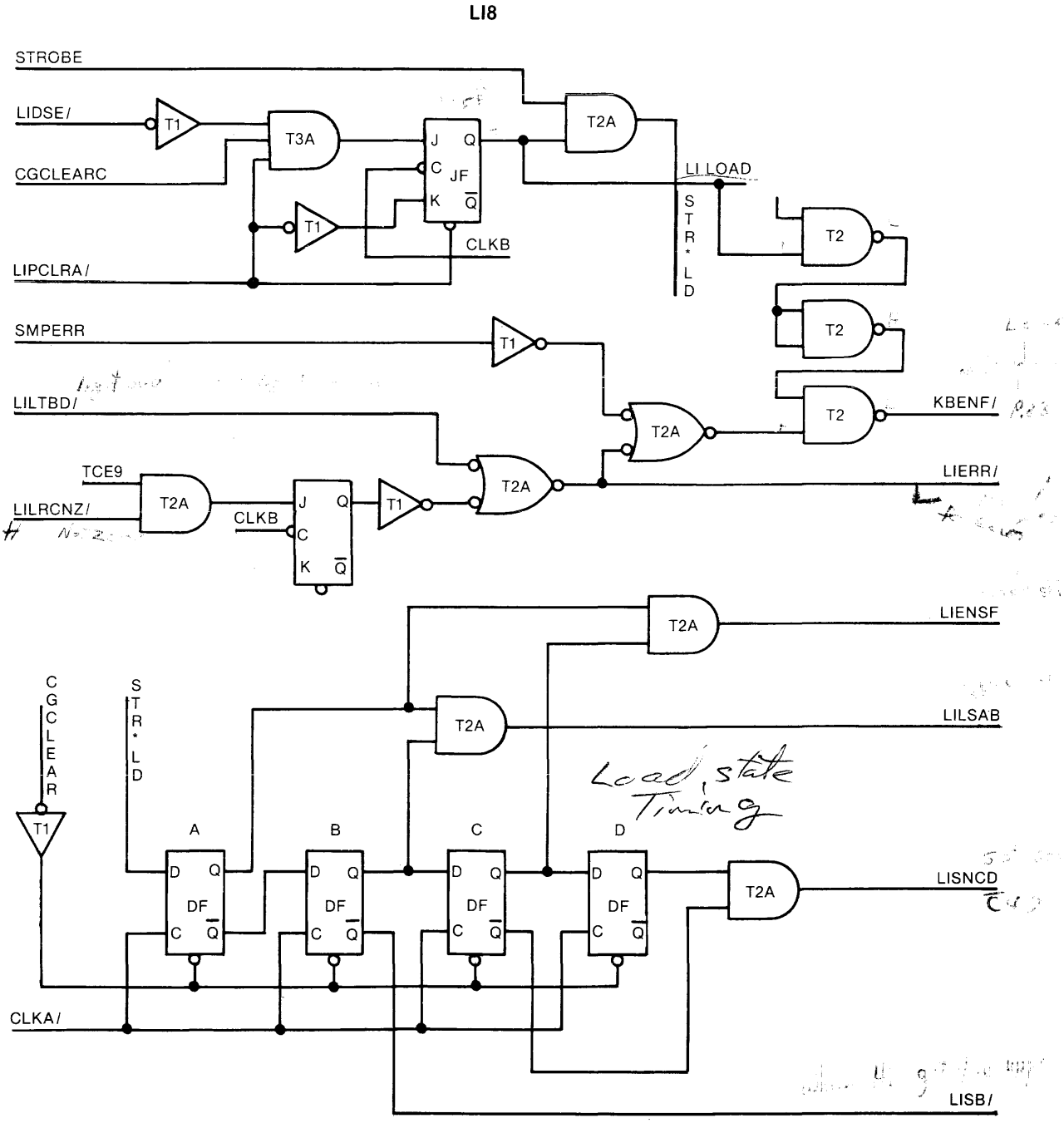


Fig. II-48 LIGHT CODE CHECK LOGIC (SHEET 1 OF 3)

DATA LOAD CIRCUITRY

The objective of the data load circuitry is to detect the tape start code (66), build a 16-bit memory word by reading in two tape characters, write that word to memory, enable the parity check logic, increment the memory address, check the accumulated LRC with the tape LRC character, and detect a stop code (40) to stop memory loading.

START CODE DETECTION

The tape start code is detected by monitoring the data lines from MLL card and decoding a 66. Figure II-50 shows the start code detection circuit.

DATA LOAD

When the start code is detected, signal LICCE66/ goes low and sets a J/K flip-flop on the trailing edge of signal LILSNCD. (See figure II-51.) When this flip-flop sets, signal LIGORST/ goes low and resets the LRC accumulation flip-flops.

(Using timing diagram figure II-52 follow the first data character through the data low circuitry.)

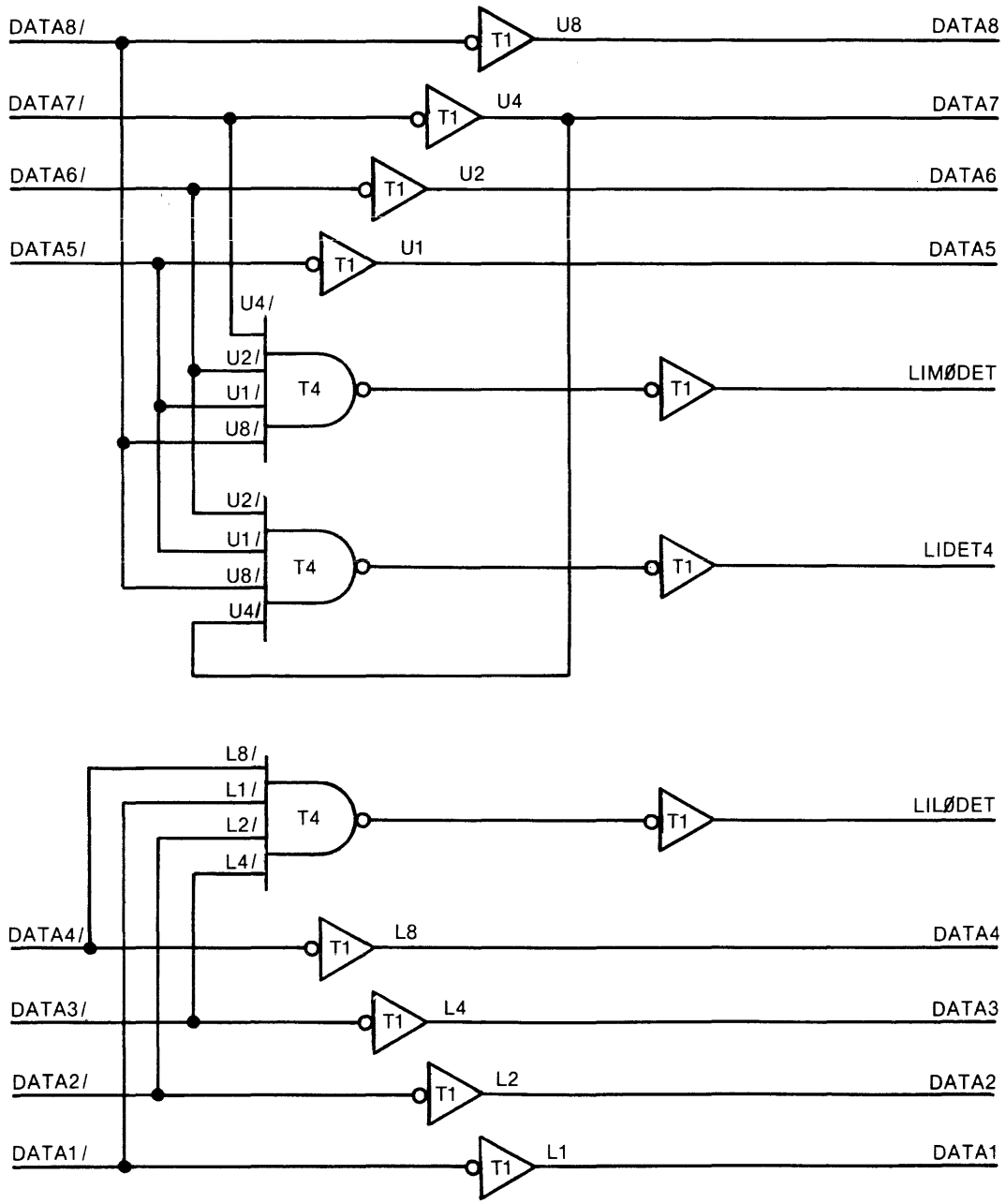


Fig. II-48 LIGHT CODE CHECK LOGIC (SHEET 2 OF 3)

The data coming in on the DATA1/ through DATA8/ lines is "FO". As shown in the timing diagram, the strobe signal start the character load timing circuit previously discussed. High signals LIENSHF and low signal LISB/ will load the upper four bits of the character into the data shift and accumulation register. (This is shown in figures II-51 and II-52.) With signal LIENSHF still high, signal LISB/ goes true and loads the lower four bits of the character into the data shift and accumulation register. During the loading of the lower four bits, the upper four bits previously loaded are shifted to the next position. The character count register is upgraded to "1" (TC-1) by signal LISAB going high, causing LILCC/ to go low. Signal LILCC/ going low will also clock in the data lines into the LRC accumulation register as shown in figure II-51. The first of two characters is now loaded.

The second tape character is now being read in. Following the timing diagram figure II-52, the second character is "A5". As described previously, the character load timing circuit is enabled to load the upper four and lower four bits into the data shift and accumulator register.

Referring to figure II-51 and timing diagram II-52, when the second tape character is being loaded, signal LILSAB going high will cause

LI7

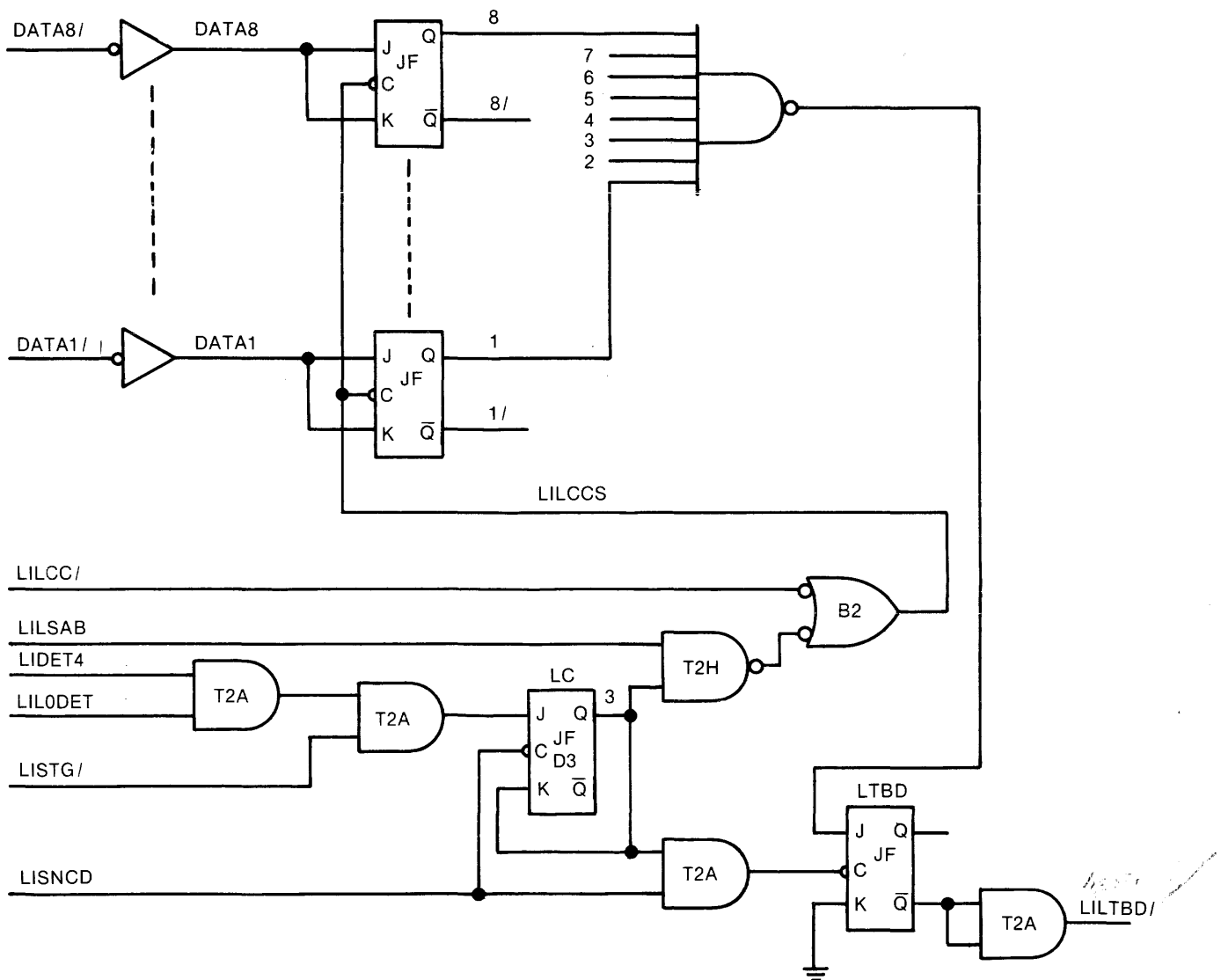


Fig. II-48 LIGHT CODE CHECK LOGIC (SHEET 3 OF 3)

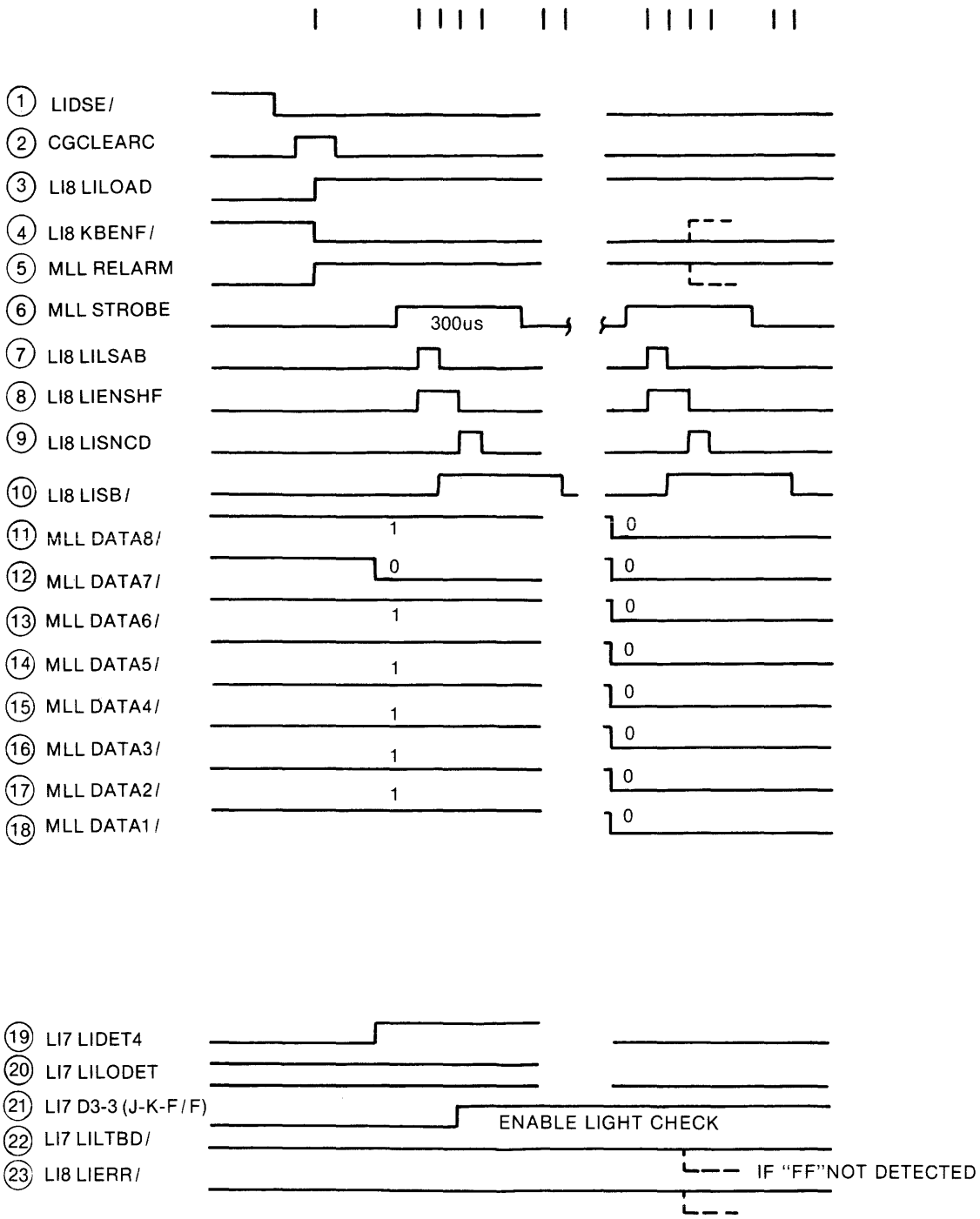


Fig. II-49 TAPE LIGHT CODE CHECK TIMING

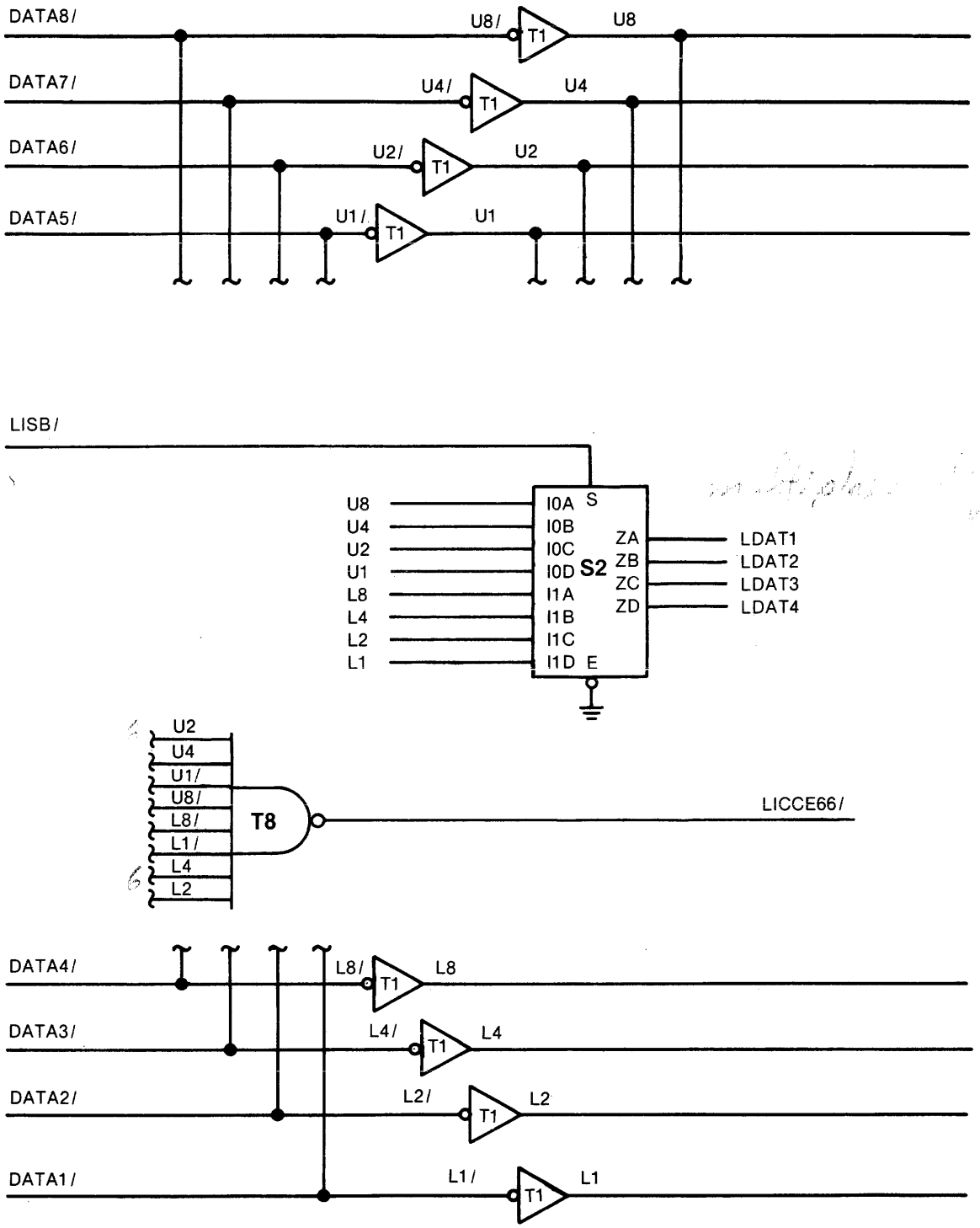


Fig. II-50 START CODE DETECTION LOGIC

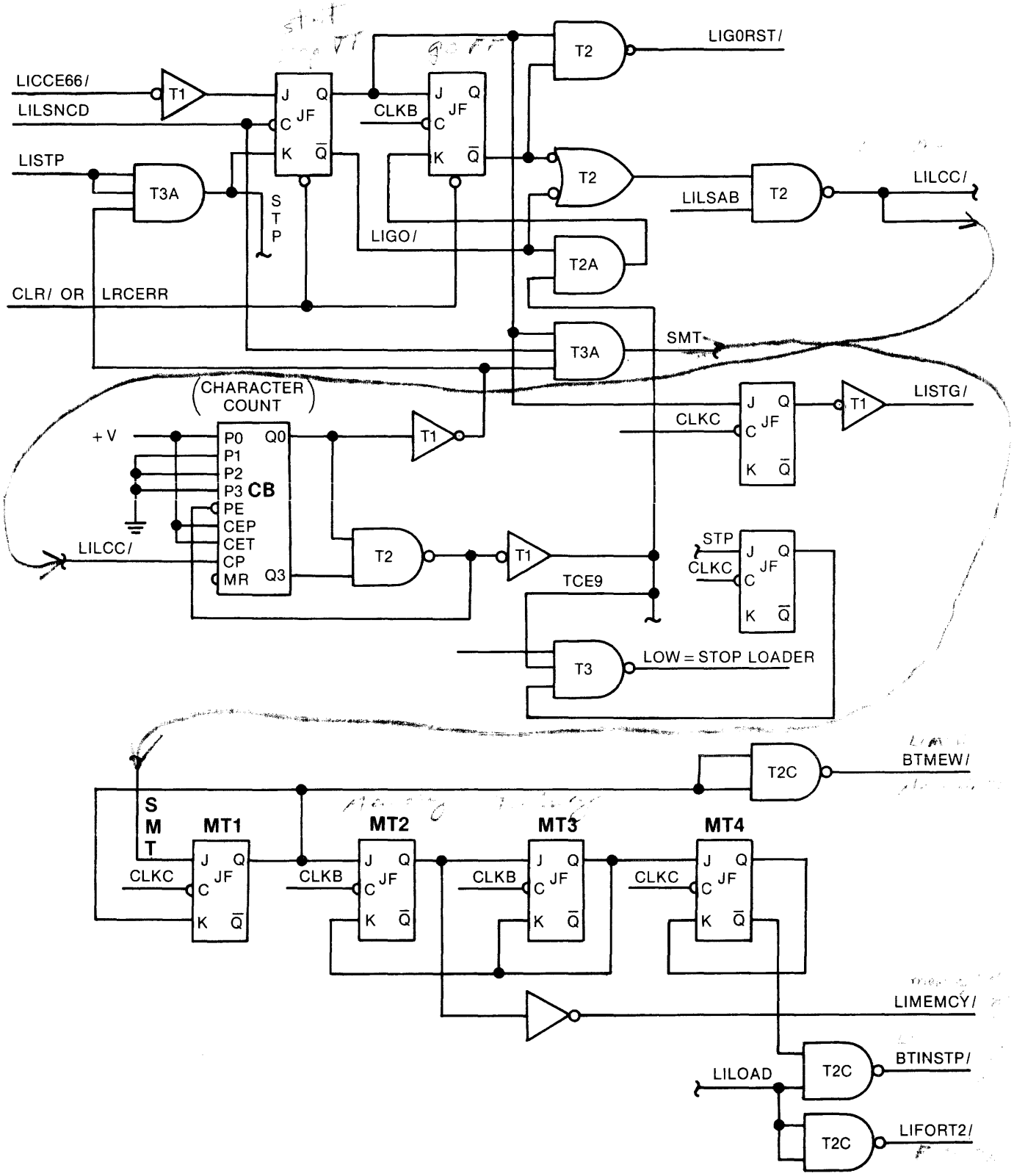


Fig. II-51 DATA LOAD LOGIC (SHEET 1 OF 3)

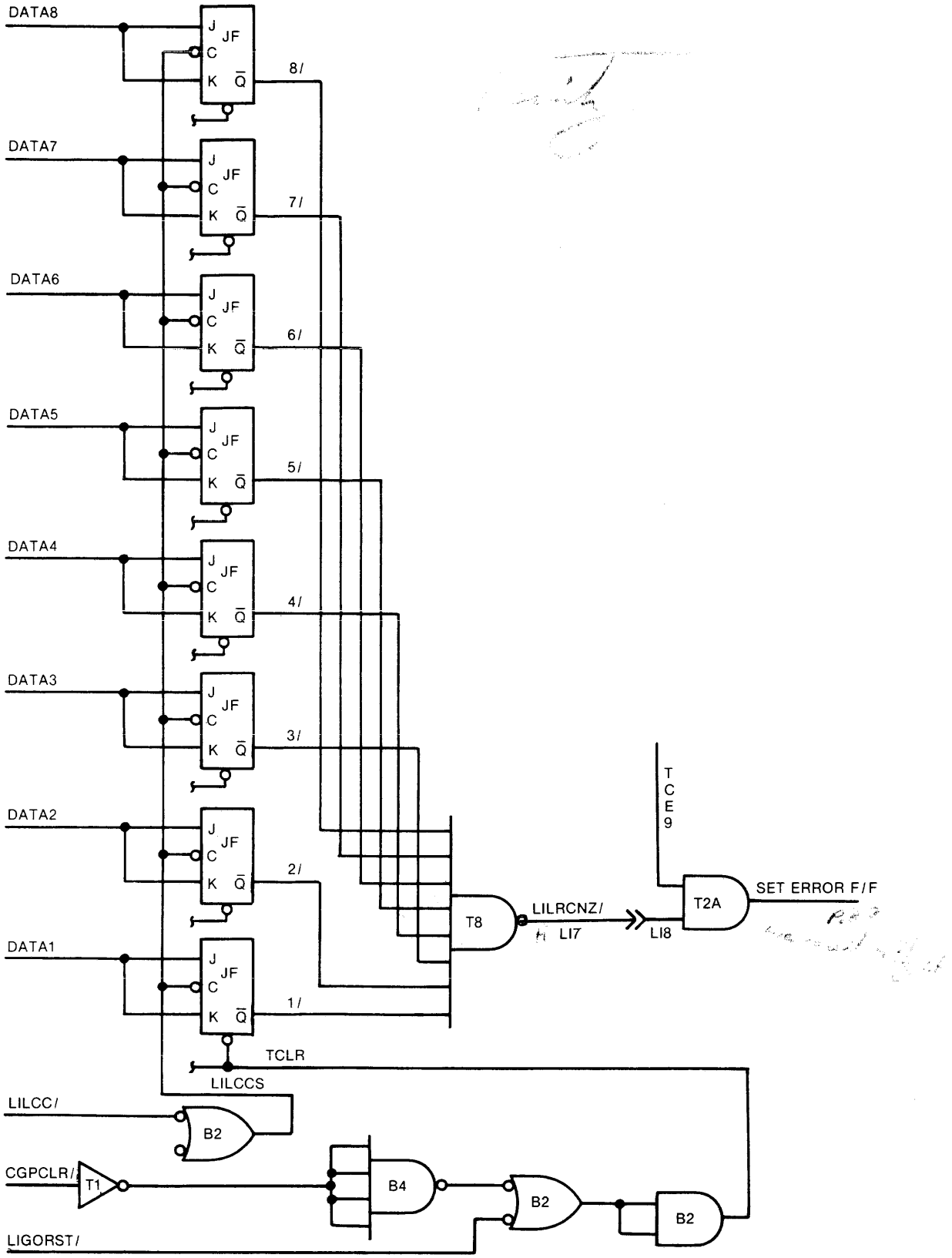


Fig. II-51 DATA LOAD LOGIC (SHEET 2 OF 3)

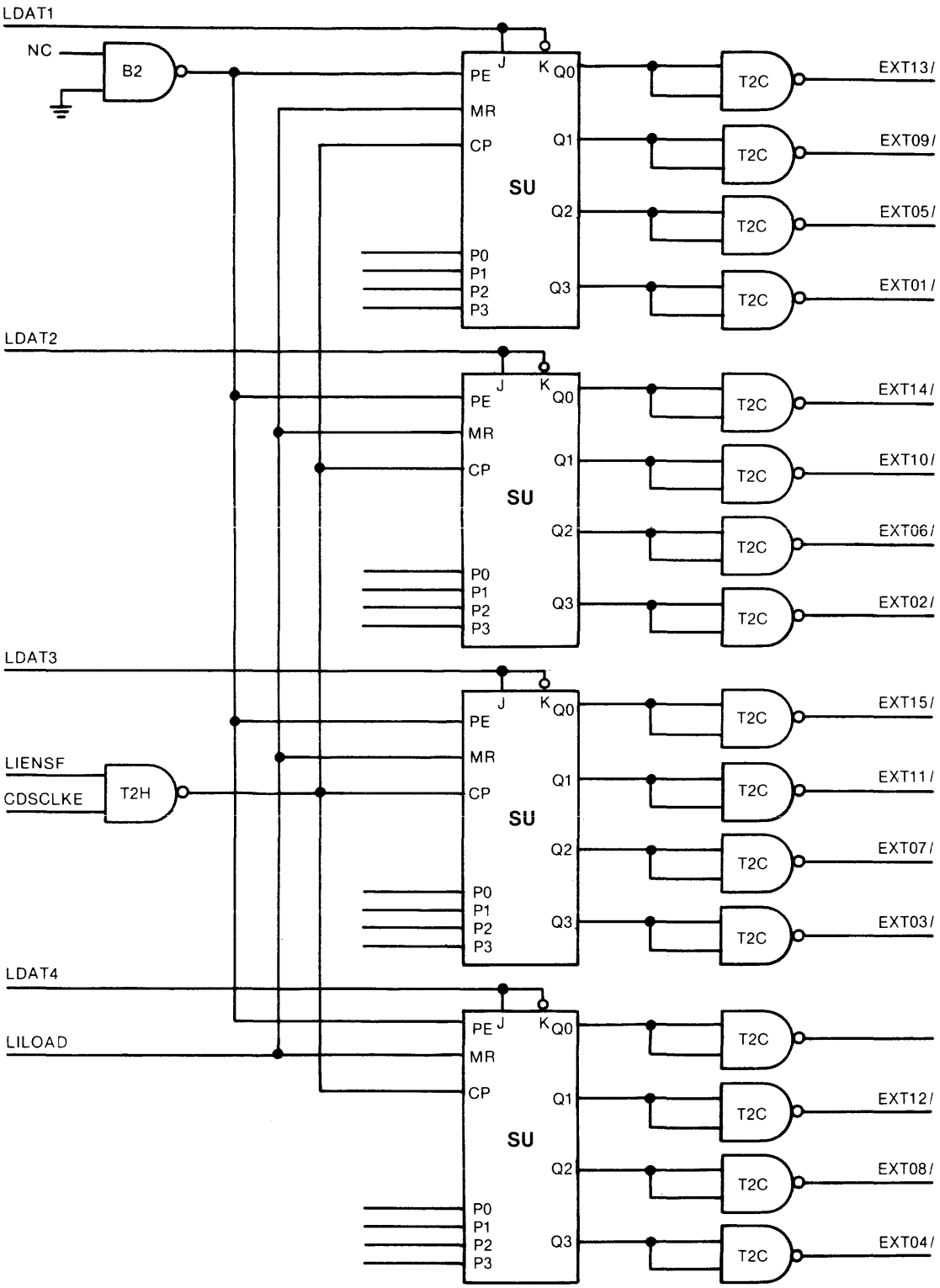


Fig. II-51 DATA LOAD LOGIC (SHEET 3 OF 3)

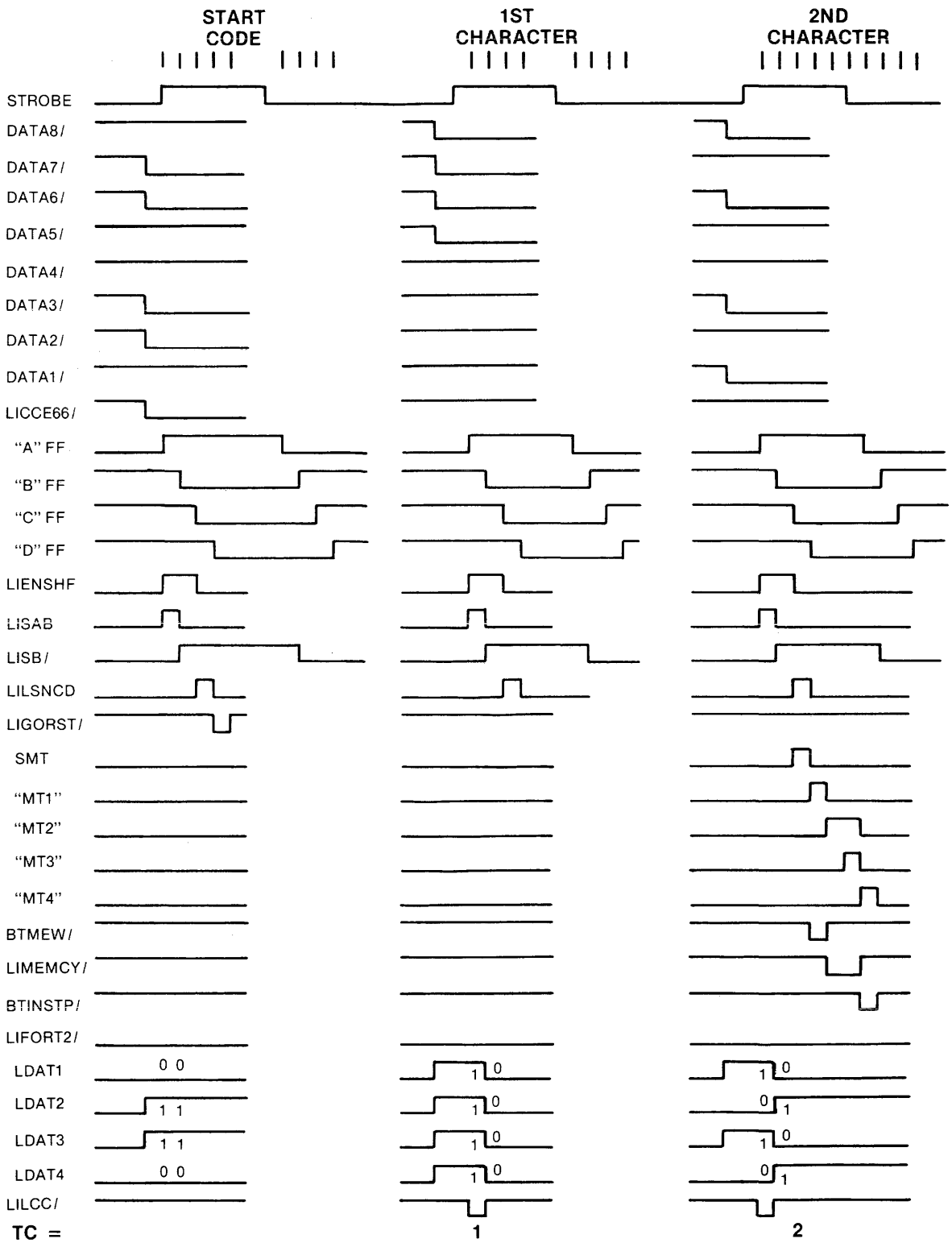


Fig. II-52 DATA LOAD TIMING

signal LILCC/ to low and cause the character count to equal 2. Signal LILSNCD going high causes signal SMT (Start Memory Timing) to go high and enables the memory timing circuit. "SMT" sets the MT1 flip-flop and causes signal BTMEW/ to go low.

A low signal BTMEW/ will write, into memory location 0, the two tape characters that are present on the EXT bus. The EXT bus is enabled to memory through the shared memory unit. The MT1 flip-flop will reset and then set the MT2 flip-flop at the next clock pulse. When MT2 is set, signal LIMEMCY/ goes low and enables the memory parity check circuit on the SM5 card described previously.

On the next clock pulse, MT2 sets the MT3 flip-flop. On the next clock pulse, MT3 resets itself and, with the MT2 set, causes signal BTINSTP/ to go high and allow the incrementer to increment once. Thus the next accumulated data word is enabled to be written into the next memory location.

The next clock pulse will reset flip-flop MT4 and complete the memory load timing. The timing for each tape character is the same as described above until the LRC character is read in. There is one LRC character for each eight data characters. When the character count register as shown in figure II-51 reaches a count of 9, signal TCE9 goes high TCE9 and enables the LRC error circuit if signal LILRCNZ/ is high. As shown in figure II-51, LILRCNZ/ is the output of an eight-input NAND gate whose inputs are the Q side of the "8" LRC accumulation register. The LRC character should reset all of the eight flip-flops. If it does not, there an LRC error, and signal LILRCNZ/ will go high to set the error flip-flop and discontinue the loading process. A correct LRC allows the loading to continue. The next character being loaded will enable TCE9 to reset the character count register to 1 and continue the loading process.

STOP CODE DETECTION

The stop code ("40" - "00") is the last data character and is included in the LRC check. The purpose of the stop code is to inhibit any further writing to memory.

Referring to figure II-48, sheet 2, the stop code detection circuit consists of three four-input NAND gates whose outputs are LIDET4, LILODET, and LIMODET. These outputs from LI7 go the LI8 card.

LIDET4 and LILODET will set a D-type flip-flop (figure II-53) when signal LISB/ goes true (comes from character timing circuit). The output of this flip/flop will set the STP flip-flop if the next character is a 00. Once the STP flip-flop is set, signal LISTP goes true. Referring to figure II-51, a high LISTP resets the LIGO flip-flop, which will inhibit the memory timing from starting. A high LISTP will also set the STPLR flip-flop (stop loader). The output of the STPLR flip-flop will stop the loader when TCE9 is high and PSIRQ/ is low. This function is only used during loader MTR operation.

Normally the loader stops only if an error occurs or the LOAD-NORMAL switch is put in the NORMAL position and the CLEAR pushbutton is pressed after the stop code has been detected. Data continues to be loaded into the LRC register until TCE9 goes high. High signals TCE9 and LIGO/ inhibit signal LILCC/ from going high and loading loading data into the LRC register.

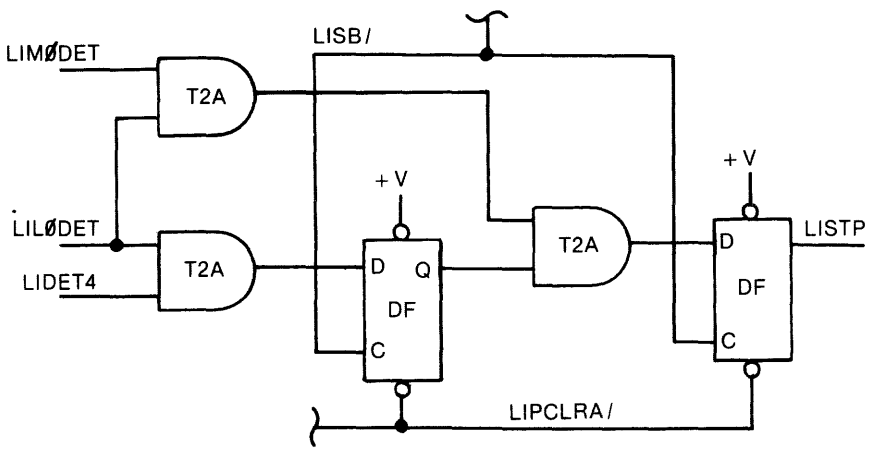


Fig. II-53 STOP CODE DETECTION LOGIC

I/O DEVICE CONTROL SECTION

The Processor controls the I/O devices through the Port Select Unit (PSU), Input/Output Controls (IOC's), and DDP's. Each device is different and requires its own unique I/O Control; however, the Processor addresses and communicates with devices in a similar manner. Figure II-54 shows the I/O device control interface.

The Memory Information Register (MIR) is the output interface register in the Processor and is common to all devices. Also common to all devices is the External Bus (EXT), which is the input interface to the Processor by way of the B Register. Because MIR and EXT lines are shared, it is necessary to select or address each I/O Control separately and only one at a time. Another function of the PSU is to enable the I/O Controls to receive information from the devices (Device Read) or send information to them (Device Write). Handling interrupts and their priority is the last function of the PSU.

PORT SELECTION (FIGURE II-54)

As discussed previously, one function of the PSU is I/O Control addressing. There are 12 possible addresses. The address of the I/O Control is determined by its location in the logic rack. Each specific I/O Control location has a corresponding cable connection for the device. (Refer to Section VI.) The priority of the device is determined by the location of the control and device. The console must be in location (or port) 12. This is fixed because the console has the highest priority and it also has a unique set of I/O Control cards and cables. Most other devices can be utilized from any port, if the hardware configuration defines their locations.

DEVICE ADDRESSING

Device Addressing originates in the Base Registers (BR1 or BR2). (See figure II-55.) It is placed there by the microprogram when it is necessary to use a device. External Operation Bits (EOB) 51 through 53 select device operation and Device Read (DR) or Device Write (DW). Nanomemory bit 54 (NMN54) selects BR1 or BR2 for device addressing. A device becomes addressed when the port is selected by sending the device address in BR1 or BR2 on the Output Select Lines (OS 5/, 6/, 7, and 8) to the PSU. This is done with a Device Read or Device Write nanoinstruction. The PSU is composed of three Port Selector Cards. Port Selector PS1-1 addresses ports 1 through 4, Port Selector PS1-2 addresses ports 5 through 8, and Port Selector PS1-3 addresses ports 9 through 12. Each Port Selector Card is identical. The -1, -2, or -3 designation is determined by the card location in the logic rack. Device addresses, binary 0 through 11, correspond to ports 1 through 12.

Figure II-56 illustrates the execution of a Device Write from BR2 (DW2) to the console. The console address is 12, but since 0 is significant, it is represented as binary 11 in the least-significant bits of the BR. When MUO506 and MUOS05 are high, PS1-3 is selected and controls ports 9 through 12. When MUOS07 and MUOS08 are high, they select the fourth port controlled by the selected Port Selector. The output of the BR, in conjunction with the EOB's, develop one active signal from the PSU. When PSWRITC/ becomes low active, it selects the console I/O Control when DW2 is executed.

SENDING INFORMATION TO AN I/O CONTROL

A device write (DW) nanoinstruction is also used to send a control word or data word to an I/O control. An example of a word format is illustrated in figure II-57. A control word is constructed in MIR by the microprogram. Its purpose is to initiate an I/O operation. For example, if a customer program requires an operator input from the console keyboard, the microprogram builds a control word to enable the logic to accept keyboard inputs. A control word defines the function to be performed. When a control word is sent to the I/O Control, MUOS01 must be set. (See figure II-56.) This is the instruction bit and it is sent to the I/O Control when the Device Address is sent to the PSU. (See figure II-58A.) A DW must be executed to send a control word to the I/O Control. The device address from BR selects the I/O Control through the PSU. The instruction bit indicates that the information in MIR is a control word, as opposed to a data word.

RECEIVING A DATA WORD FROM AN I/O CONTROL

Figure II-57 illustrates a data word format. In the case of the console, the keyboard eight-bit key code appears to the I/O Control as a data word. When a key is pressed, this code is sent to the Processor by a Device Read (DR)

DEVICE CONTROL (CONTINUED)

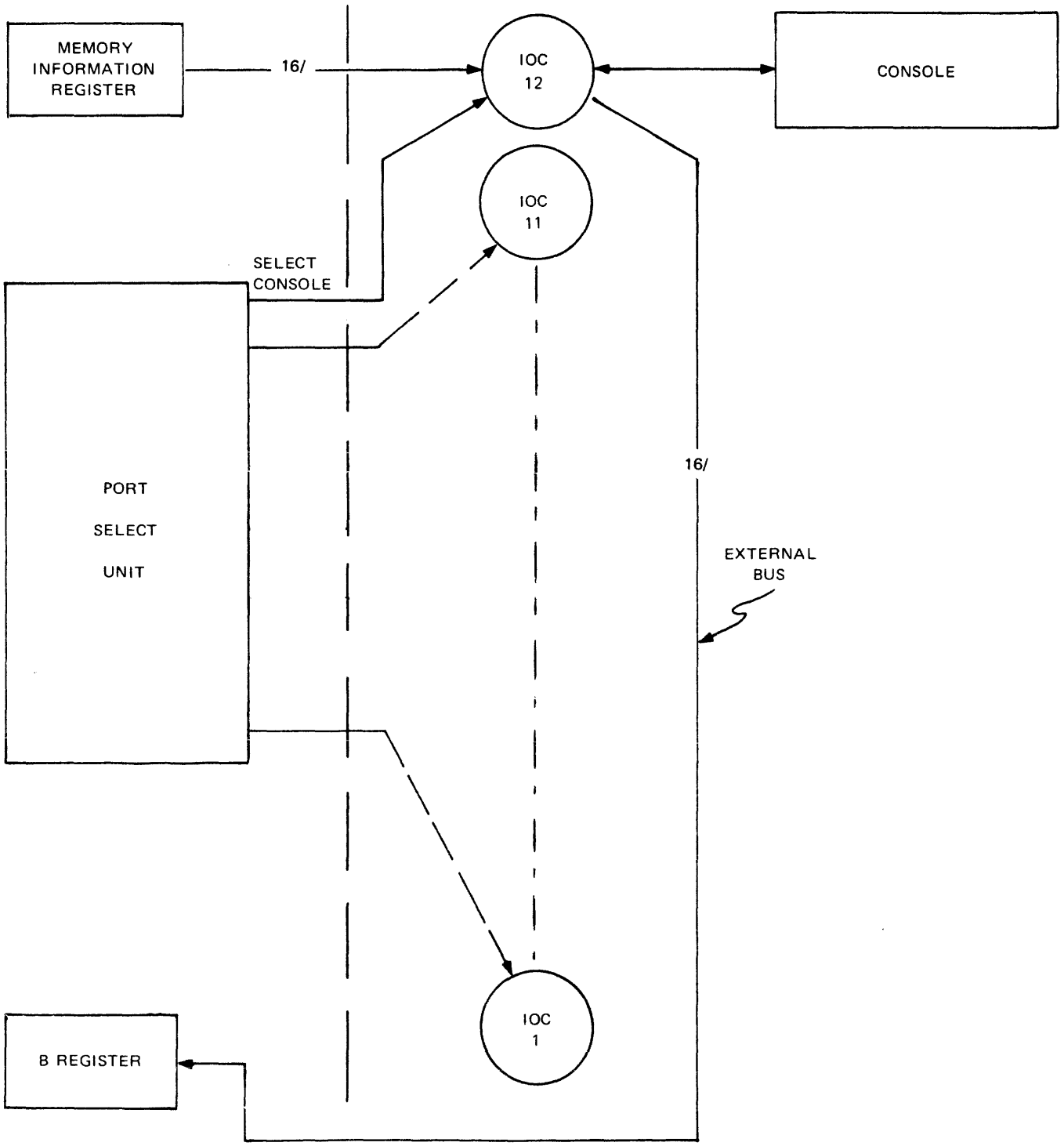


Fig. II-54 I/O DEVICE CONTROL INTERFACE

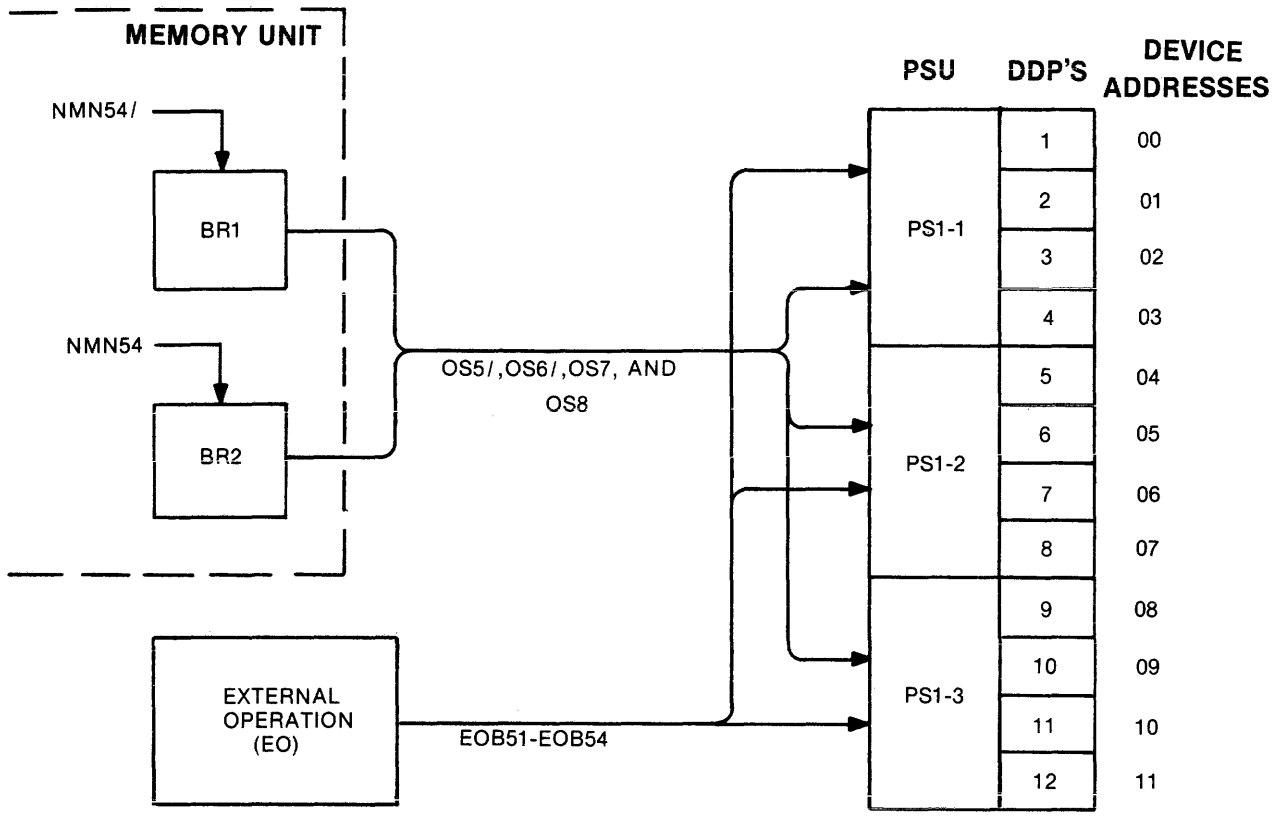


Fig. II-55 PORT SELECTION (DEVICE ADDRESSING)

nanoinstruction. (PSREADC/ is active.) Another requirement to read data, instead of sending a control word, is that the instruction bit must be low. (See figures II-56 and II-58.) The Processor receives an interrupt when the key is pressed, and this is what indicates activity at the keyboard.

INTERRUPTS

It is not until an interrupt is generated from a keyboard that a DR is executed by the Processor. The effect of this is shown in figure II-59. Pressing a key generates a Data Interrupt (DINT) in the I/O Control. Because the console is selected, the PSU sends a Solicited Request (SRQ) to the Condition Register in the Processor. This condition is expected, because the control word enabled the keyboard logic. The microprogram tests for the SRQ by means of a nanoinstruction. When it is active (because a key was depressed), DR is executed. DR not only retrieves data from the I/O Control, but also resets DINT so it can be set by the next key activated.

Figure II-60 illustrates the logic to generate an SRQ. When a key is pressed, DINT12/ from the console I/O Control is low-active into the PSU. At this time, PSWRITC/ is low active; thus Q3/ is low. The outputs at gate pins D1-8, D1-11, and B3-8 are high. Because PSWRITC/ is active, the device is selected. The combined inputs of DINT12/ and PSWRITC/ at PSWRITC/ at chip D7 causes a low from pin 7. PSSRQ/ goes low-active. This is the result of a DINT from a selected device which would be followed by a DR in the case of the console keyboard being enabled.

Another type of interrupt is a Status Interrupt (SINT). (See figure II-57.) The console status word relates the condition or status of the device and can be interrogated by the Processor at any time. Interrupt Not Honored (INH) is active if a key was depressed on the console and the previous key code was not read. INH generates a SINT. Figure II-59 shows SINT as an input to the PSU, just like DINT. The PSU sends an Unsolicited Request (URQ) to the Condition Register in the Processor if the device is selected. The logic for this operation (figure II-60) is similar to the generation of an SRQ. Once again, the microprogram is checking conditions in the Condition

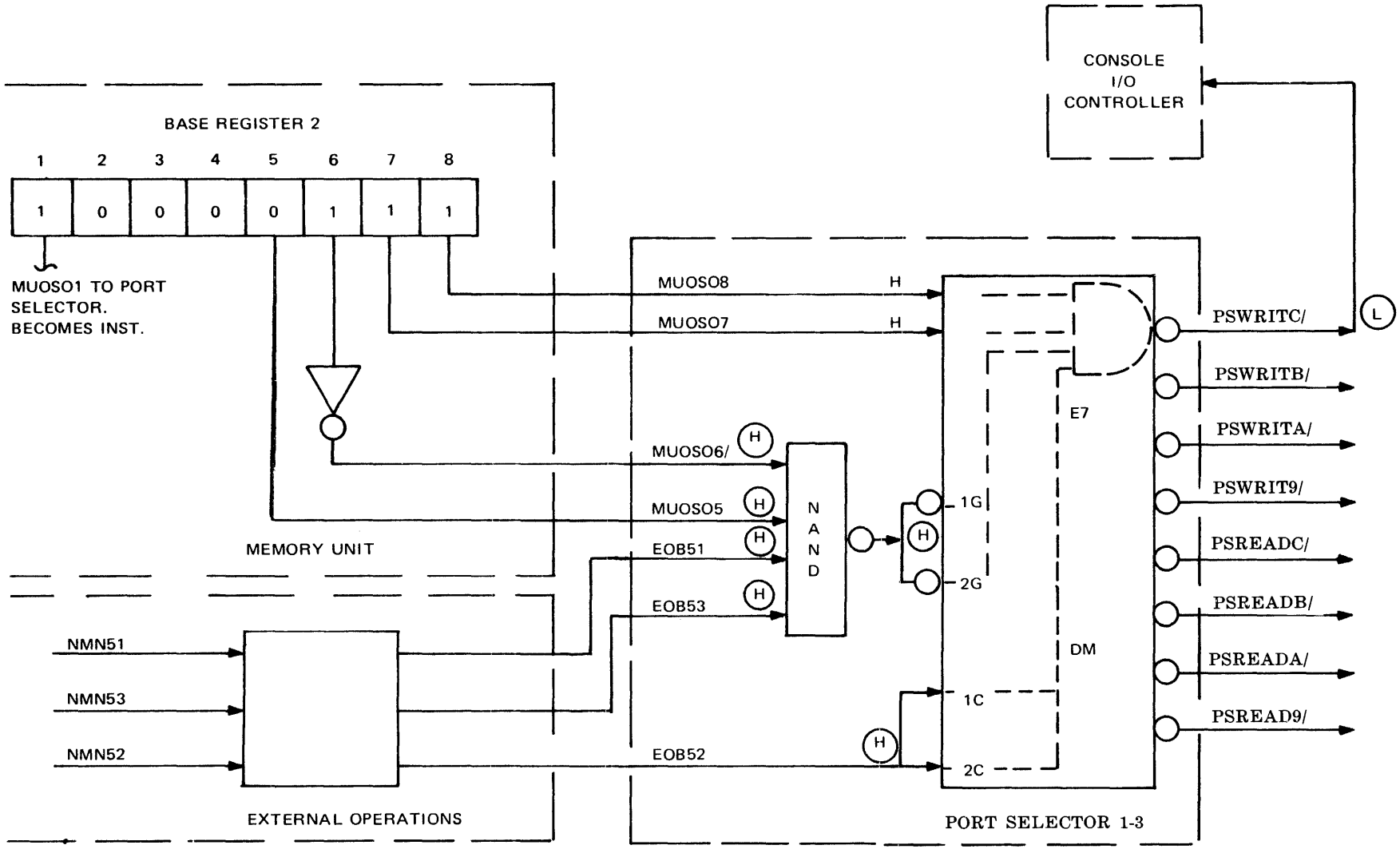
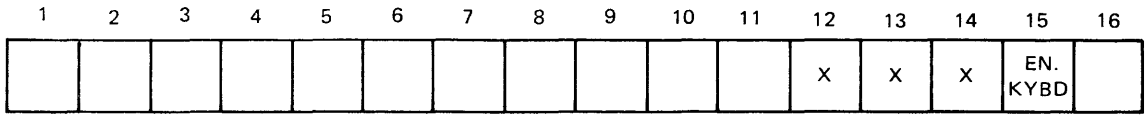
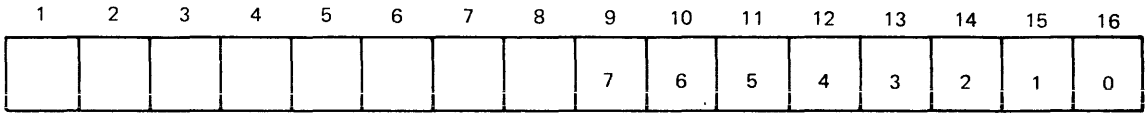


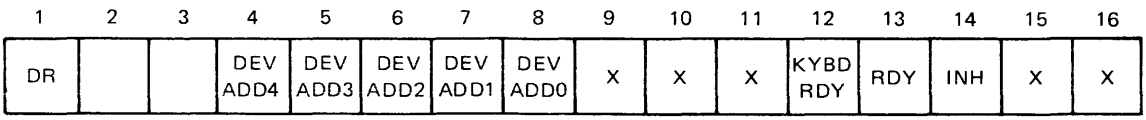
Fig. II-56 DEVICE WRITE FROM BR2 (DW2) TO CONSOLE



CONTROL WORD (MIR)



DATA WORD (B REGISTER)



STATUS WORD (B REGISTER)

NOT USED

X — USED FOR FUNCTIONS OTHER THAN KEYBOARD

Fig. II-57 CONSOLE KEYBOARD WORD FORMATS

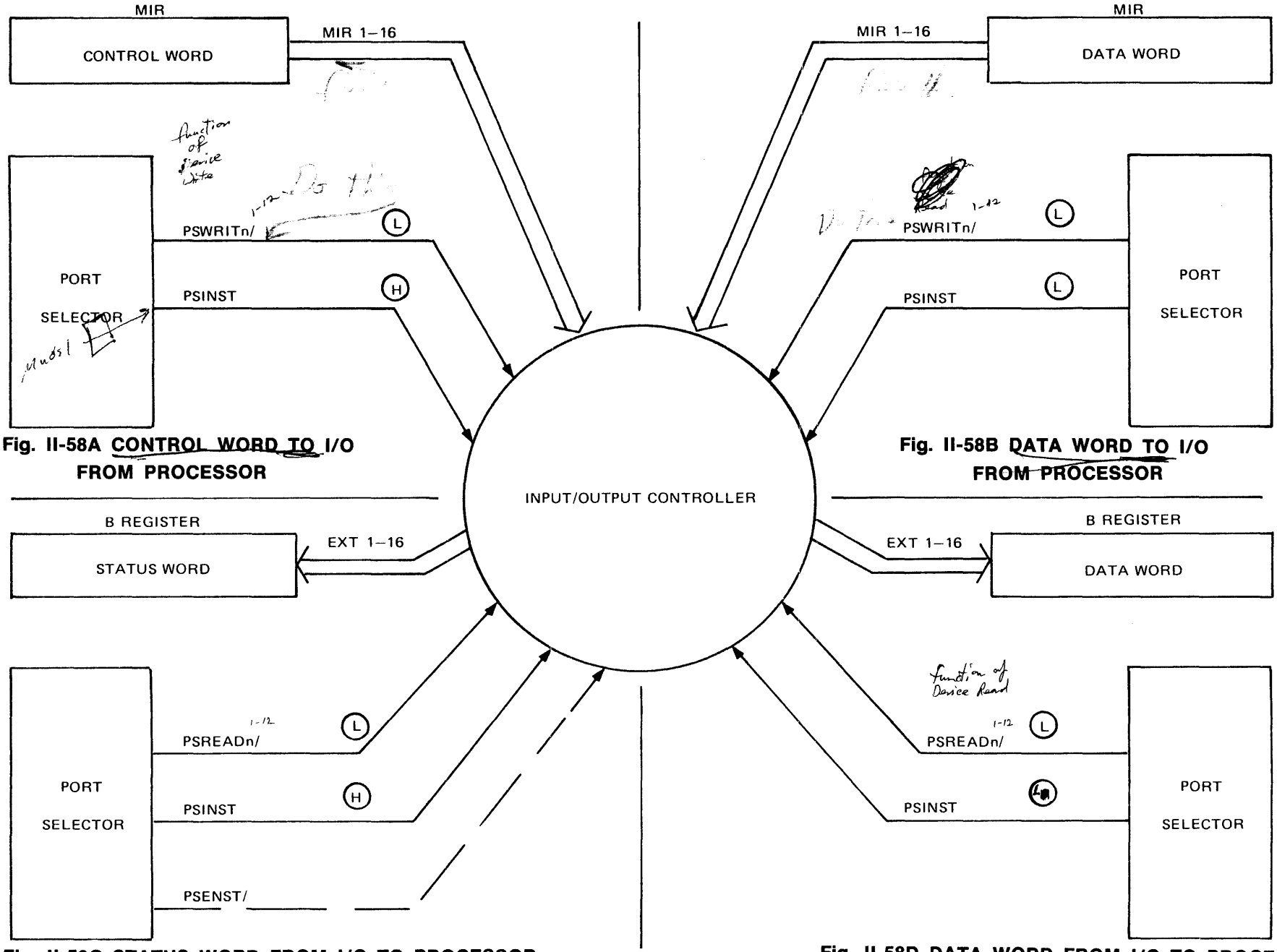


Fig. II-58A CONTROL WORD TO I/O FROM PROCESSOR

Fig. II-58B DATA WORD TO I/O FROM PROCESSOR

Fig. II-58C STATUS WORD FROM I/O TO PROCESSOR

Fig. II-58D DATA WORD FROM I/O TO PROCESSOR

Register. When the URQ is active, the Processor must read status. This is accomplished by executing a DR with the Instruction Bit set. (See figure II-58C.) The status word is sent to the B register; from there the microprogram determines which status bit or bits are active. The result depends on the microprogram.

I/O OPERATIONS

In reference to the example of selecting the console I/O Control, figures II-61 and II-62 show the relative operation flop of sending a control word, generating interrupts, reading data, and returning status words.

Before this point, the console has been considered selected. The next example of communication is initiated by pressing the console Input Request pushbutton when the console is unselected. Assume the Processor is reading 80-column cards and storing the data on disk at the time the Input Request pushbutton is activated.

If the disk was the last device selected and the card reader finished another card at the same time the console Input Request pushbutton was pressed, an Interrupt Request (IRQ) is generated from both devices. An IRQ is an interrupt from an unselected device and it is caused by a DINT or SINT. The microprogram issues an Address and Status Request (ASR) when it finds IRQ active. Only the address of the highest priority device is sent, with the status word, to the Processor. Determining which status word to send is the final function of the PSU. (See PS1-1, PS1-2, and PS1-3 logic diagrams in the FT&R documentation.)

DINT3/ from the card reader and SINTC/ from the console are both low-active in the example; these signals develop I3/ and IR/ low active, respectively. I8/ generates PSENAB2/ low-active, which disables the output from PS1-1. PS1-1 controls ports 1 through 4. Signal I8/, in conjunction with EOASR/ low-active, develops PSENSTC/ low-active. This console I/O Control signal sends the status bits on the EXT lines to the B Register. (See figure II-61C.) I8/ also places the address of the console on the EXT lines by activating EXT05/08/. The status and address are sent to the B Register as a status word simultaneously. (See figure II-57.) EXT13/ is low-active as a result of pressing the Input Request pushbutton. At a convenient time, the microprogram initializes the console (possibly after reading the card reader data), and the system is ready. In any case, the processor services the card reader, because it also generated in IRQ.

MEMORY STEAL OPERATIONS

The purpose of a memory steal operation is to allow an IOC that has direct memory access (DMA) to interrupt the processor and communicate with system memory.

A memory steal operation is controlled by logic contained on the STC card which is included in the universal disk IOC and logic contained in each of the DMA IOC's.

Each DMA IOC has a request-to-steal line going to the STC card. The STC logic enables the steal enable line to the DMA IOC requesting the steal according to a fixed priority. The request-to-steal lines are numbered 1 through 4; line 1 has the highest priority and line 4 has the lowest priority.

When a request-to-steal line is enabled, the STC logic inhibits all lower-priority requests to steal and also inhibits the processor from performing any operation. The processor will be inhibited until all requests to steal are inactive.

The DMA IOC's communicates with system memory over a shared memory address bus which is backplane ORed with the processor shared memory address bus. When the DMA IOC is enabled to steal, the processor SM address bus is disabled. During a memory steal read operation, the processor LUMIR bus is disabled and the MPM data being read is enabled to a LUMIR bus contained on the STC card, allowing the DMA IOC to read the LUMIR bus into its registers.

During a memory steal write operation, the DMA IOC enables its memory write line (BTMEW/) and enables the data to be written to memory onto the EXT BUS.

DEVICE I/O CONTROLS

The hardware configuration and the communication between the Processor and each of the I/O control and device interfaces are described in the respective technical manuals for the I/O controls.

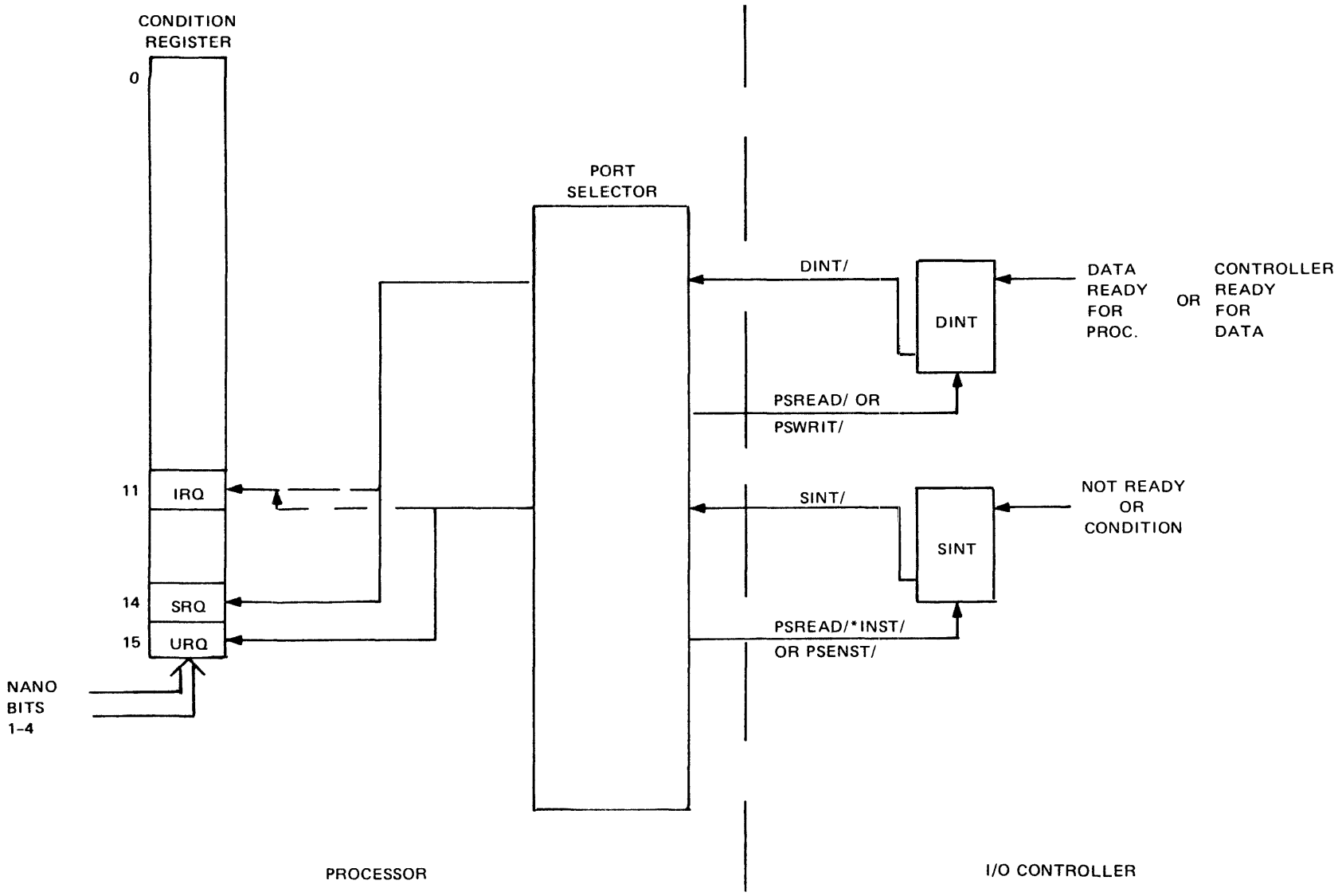


Fig. II-59 INTERRUPT FLOW

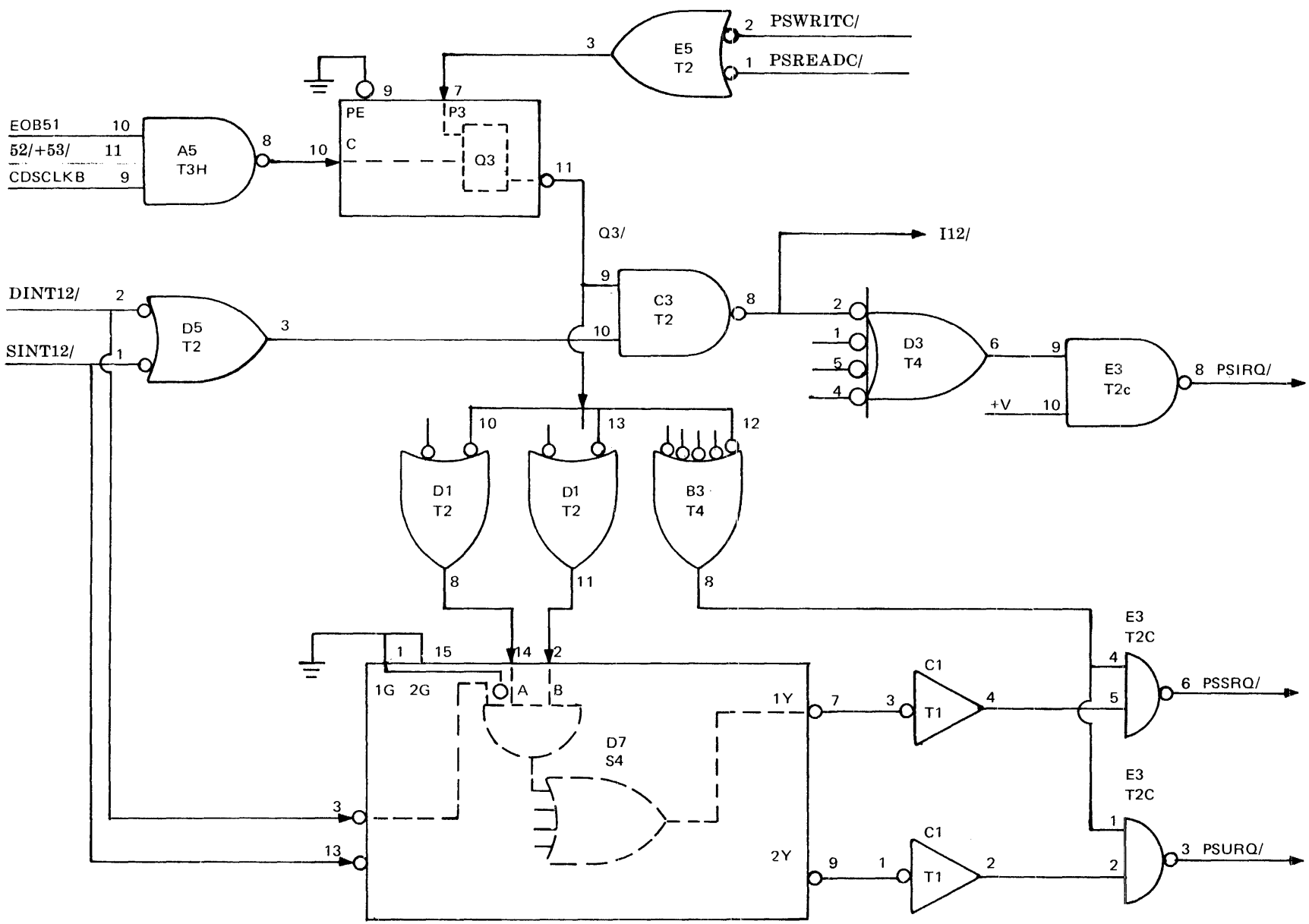


Fig. II-60 GENERATION OF INTERRUPTS (PORT SELECTOR 1-2)

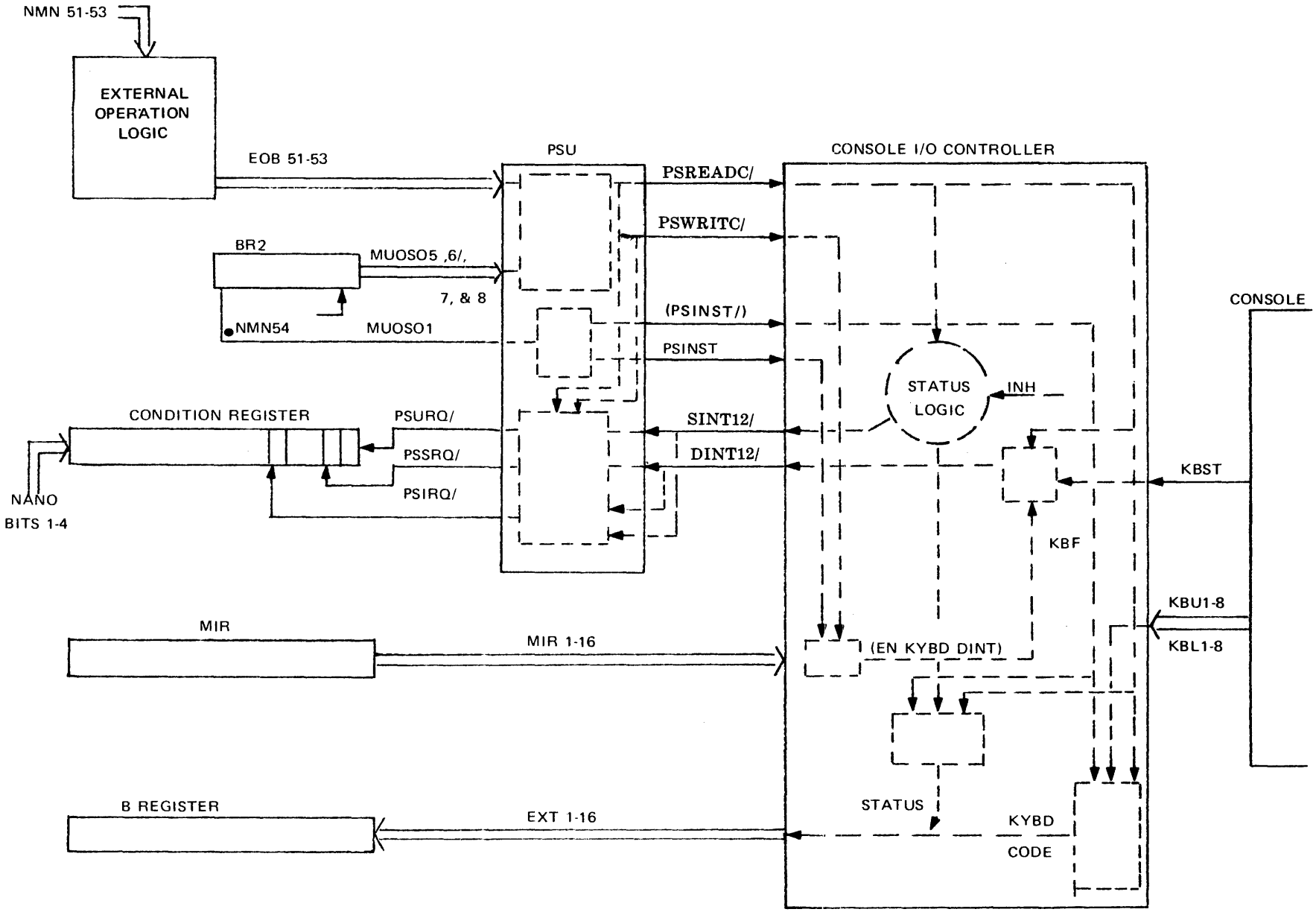


Fig. II-61 CONSOLE KEYBOARD OPERATION

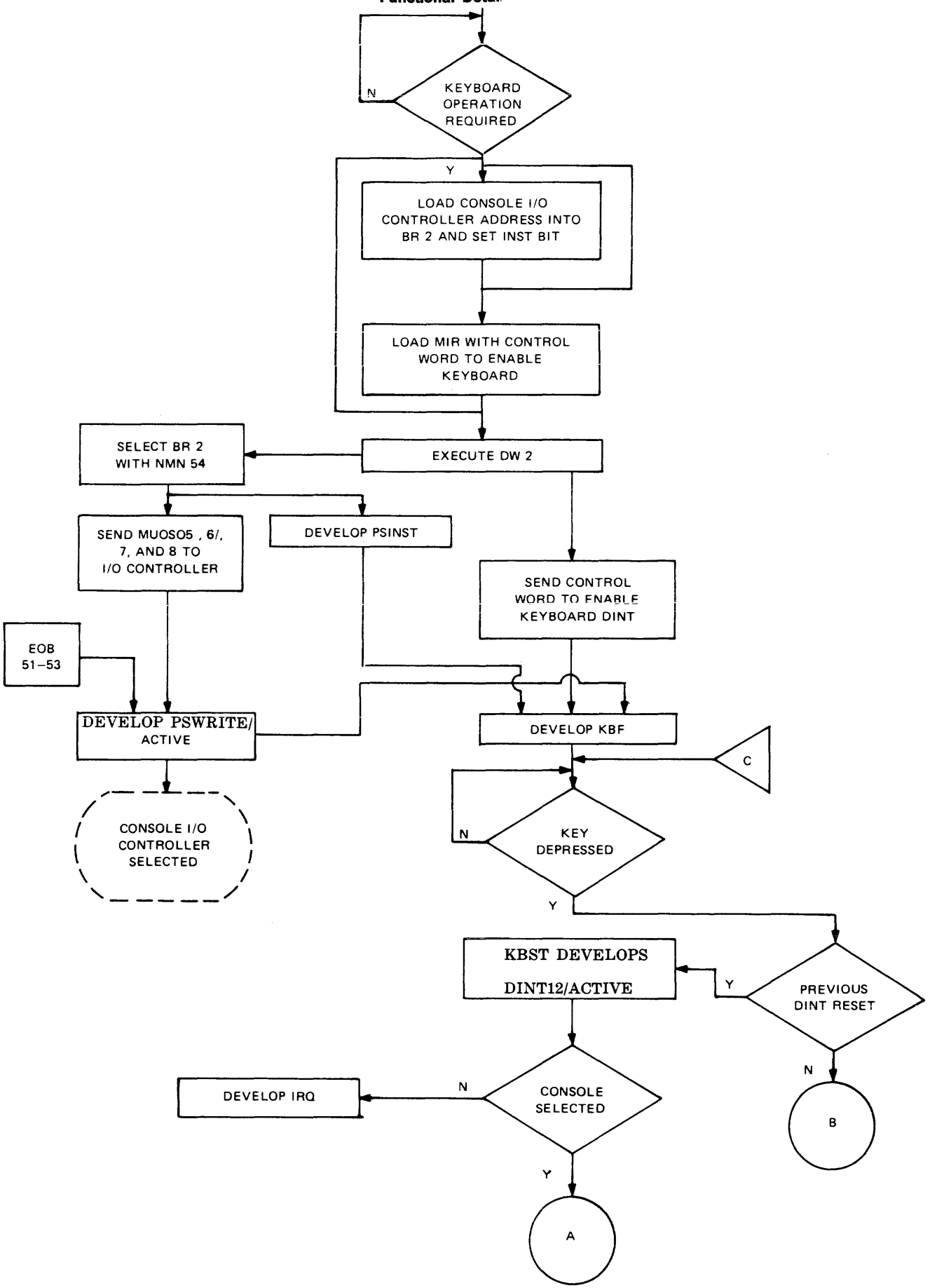


Fig. II-62 CONSOLE KEYBOARD OPERATION (SHEET 1 OF 2)

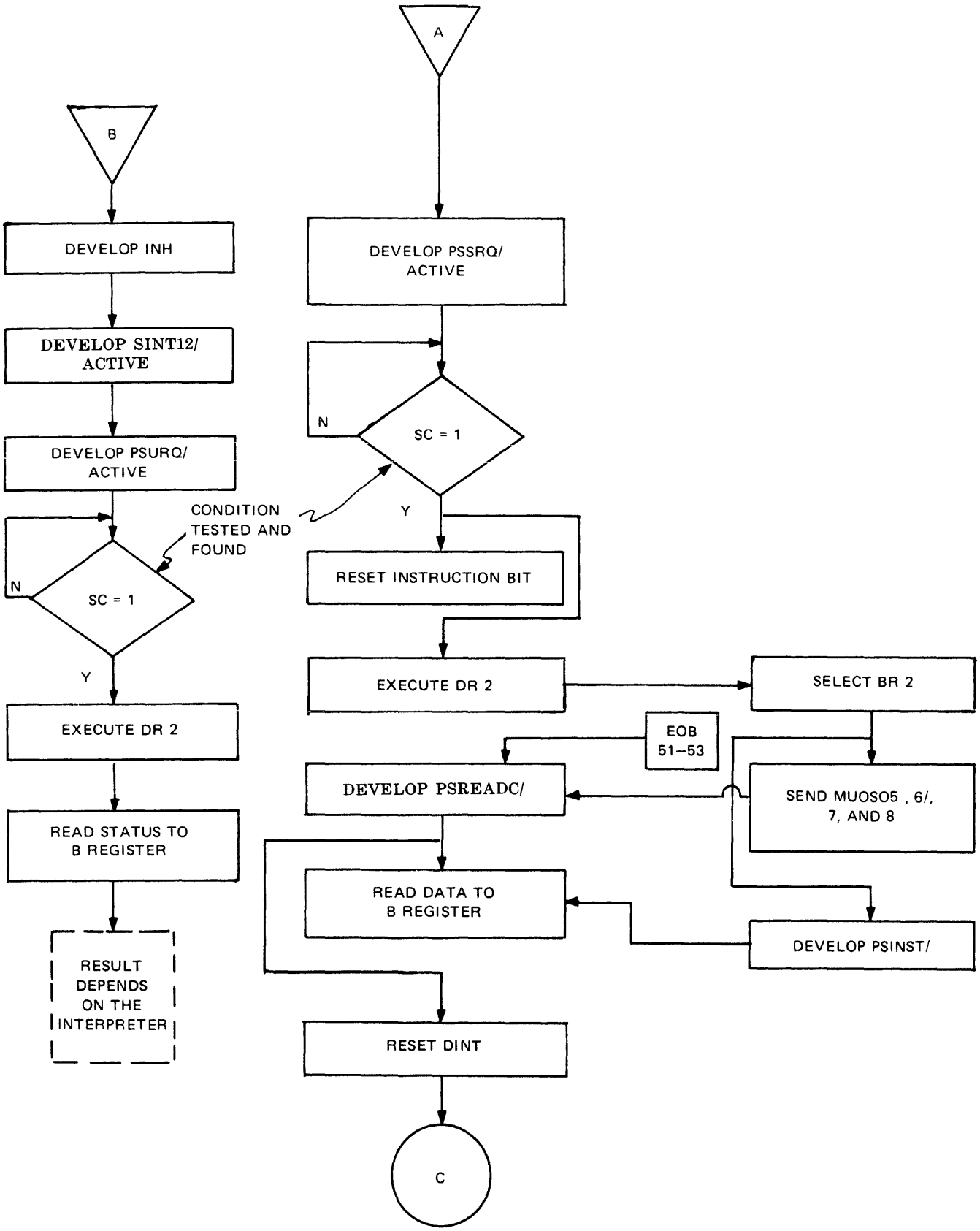


Fig. II-62 CONSOLE KEYBOARD OPERATION (SHEET 2 OF 2)

POWER SUPPLY SECTION

The processor power supply section develops power sources of +5, +12, -12, and +24 volts dc and distributes these voltages to the various processor and I/O circuits. Control card AC3 controls the power-up and power-down sequence of operation and also contains sensing circuits for undervoltage conditions. Figure II-63 is a block diagram of the basic power and control circuits. Figure II-64 is a timing diagram for the power-up and power-down sequences. Figure II-65 is a general block diagram of control card AC3. Figures II-66 and II-67 are flow diagrams of the power-up and power-down sequences, respectively.

POWER-UP SEQUENCE

When CB1 is set ON, the system console emergency power off switch is in position ON, and the processor power ON pushbutton is pressed, relay K1 is energized. Closed contacts A and B of K1 energize the 12 and 24 volt dc power supply. Contact C of K1 enables the dis-

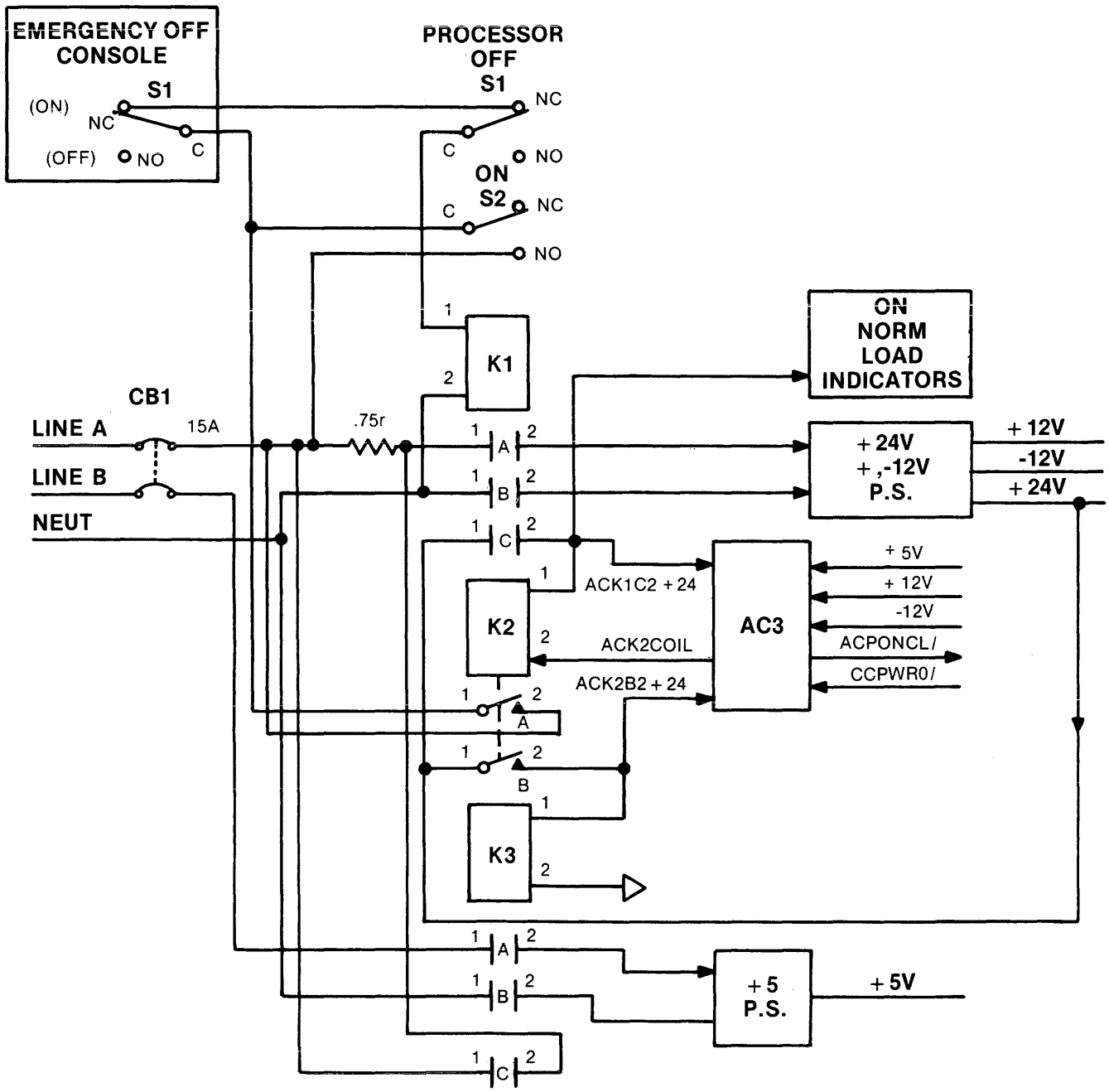


Fig. II-63 BASIC POWER-UP CONTROL CIRCUITS

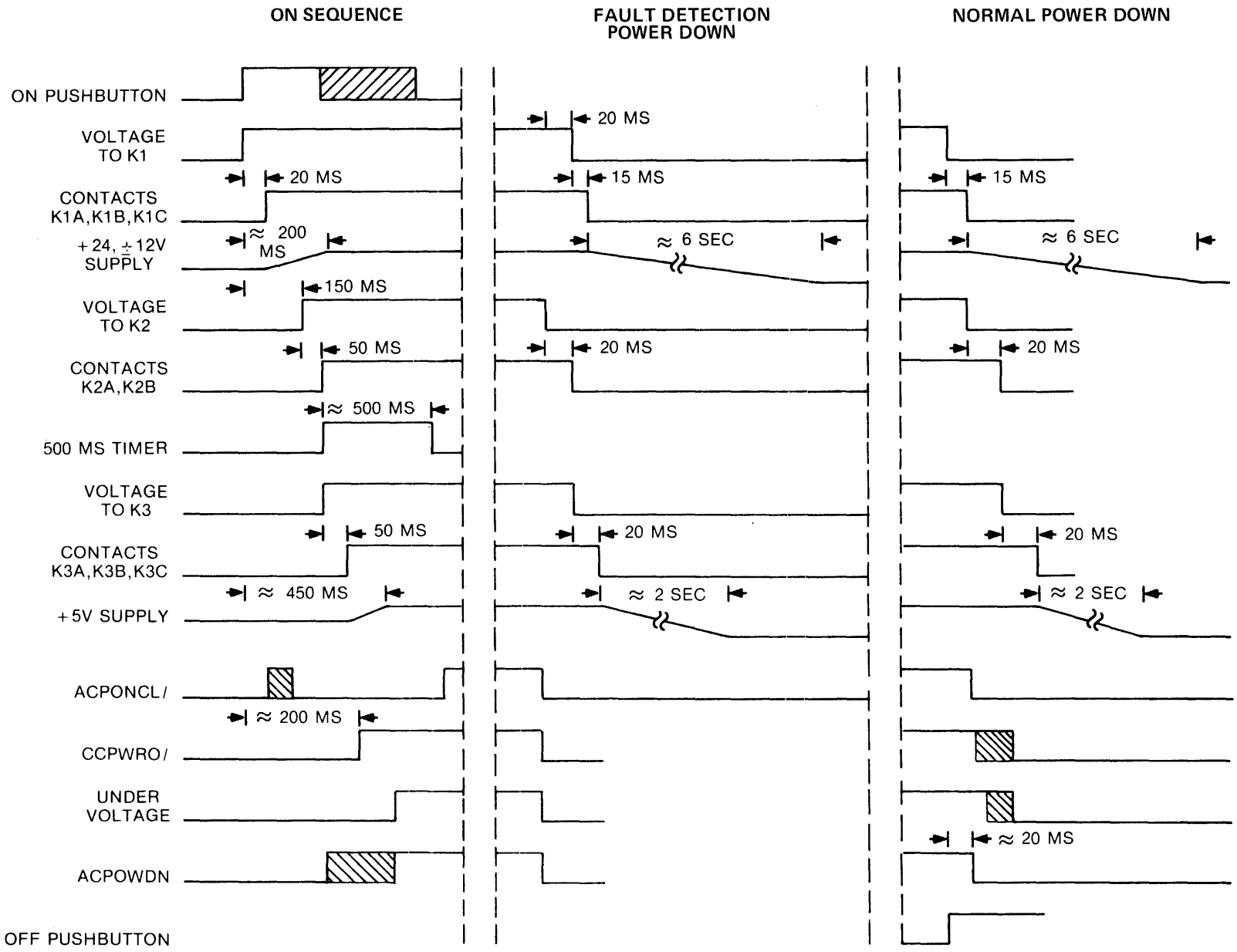


Fig. II-64 POWER-UP AND POWER-DOWN TIMING DIAGRAM

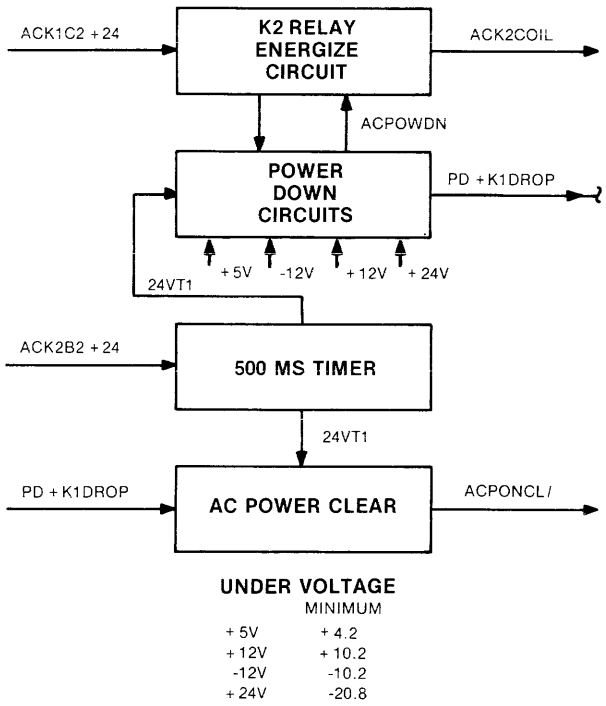


Fig. II-65 AC3 CARD BLOCK DIAGRAM

tribution of +24 volts to card AC3. AC3 contains a circuit which energizes relay K1 when the +24 volt source reaches approximately 17 volts.

When K2 is energized contact K2-A holds K1 energized after the power ON pushbutton is released. Closed K2-B energizes relay K3 and also routes +24 volts to a 500-nanosecond timer on card AC3. This timer is used to delay the power-down circuits from being activated until the power supplies have time to stabilize. The timer also disables the power-on clear signal after the 500-nanosecond delay.

Contacts K3-A and K3-B energize the +5 volt power supply, and contact K3-C bypasses the 0.75-ohm current-limiting resistor used in primary input power line A.

POWER-DOWN SEQUENCE

Processor power can be turned off by one of the following:

a. Processor power OFF pushbutton is pressed (deenergizes relay K1).

b. Console READY pushbutton is pressed during ready-idle state. This programmatically causes K2 to deenergize and subsequently deenergize K1.

c. AC3 undervoltage detection circuits detect an undervoltage condition. This deenergizes K2 and subsequently deenergizes K1.

d. During power-on sequence, +24 volt supply does not reach operational status. This inhibits K2 from energizing and thus deenergizes K1.

e. Any primary power source failure.

Figure II-67 shows the flow of power-down sequence of operations.

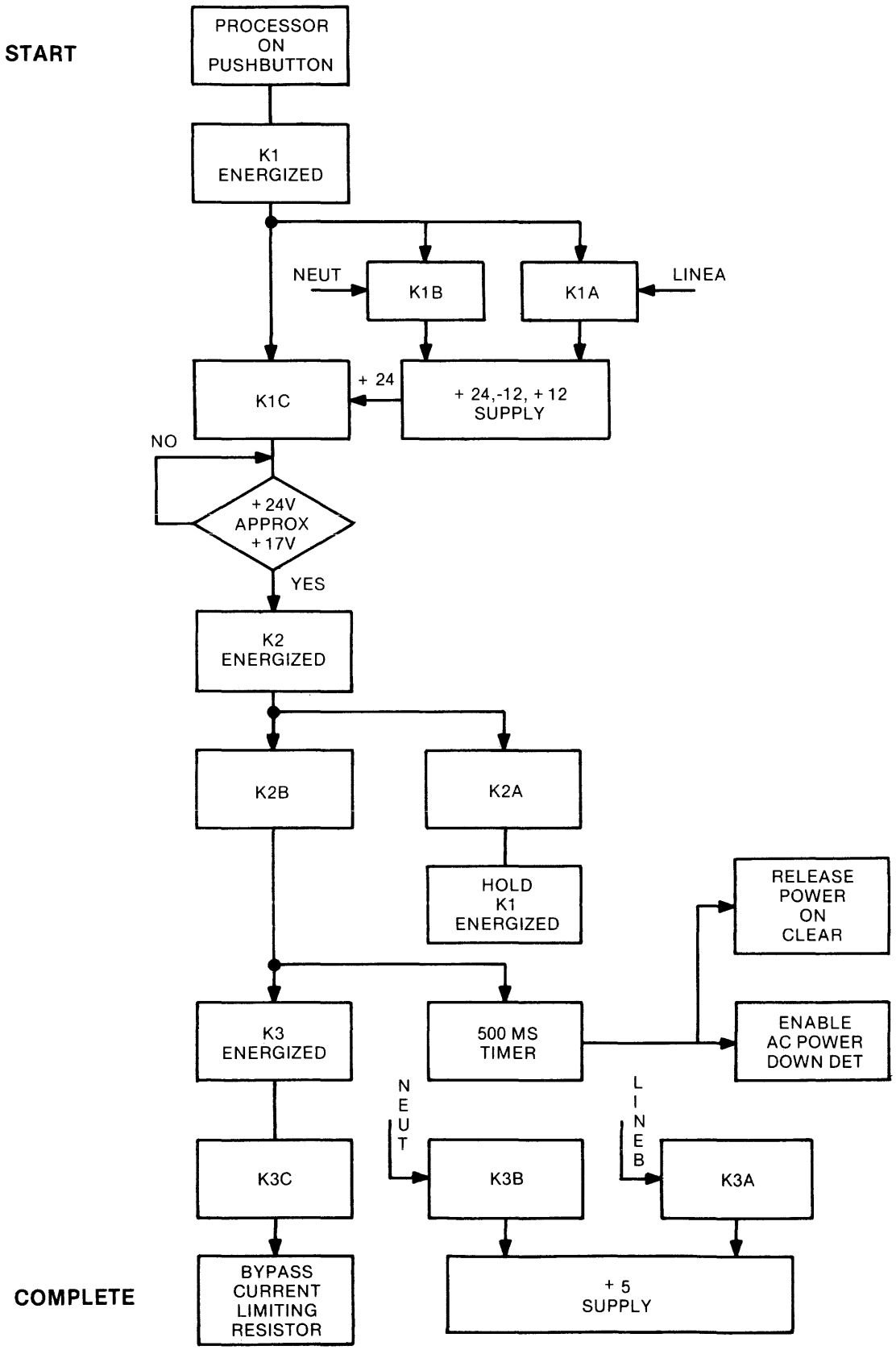


Fig. II-66 POWER-UP SEQUENCE FLOW DIAGRAM

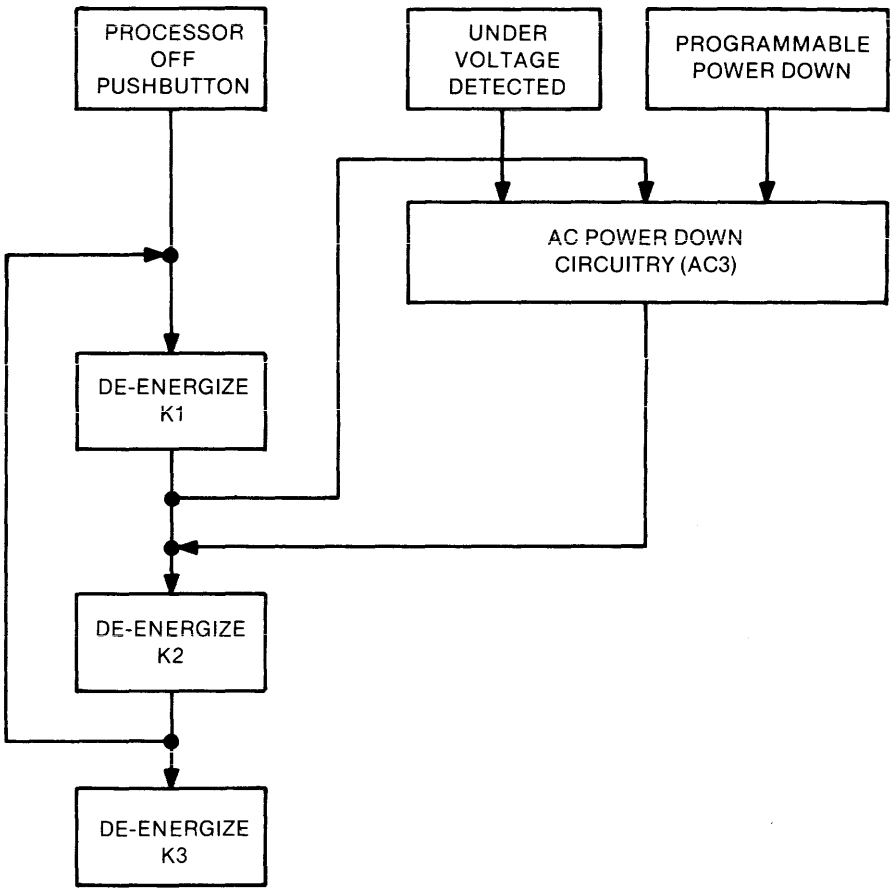


Fig. II-67 POWER-DOWN SEQUENCE FLOW DIAGRAM

B 720
PROCESSOR

SECTION
IV

Burroughs

ADJUSTMENTS

FIELD ENGINEERING

TECHNICAL MANUAL

Printed in U.S. America 7-30-75 For Form 1077534

POWER SUPPLY (5-VOLT) TAP ADJUSTMENT

A tap adjustment is provided for the processor 5-volt power supply to ensure that the 5-volt operating voltage at the processor backplanes is within specified limits.

The tap adjustment involves the positioning of two leads, marked "+" and "–", which are connected to terminal board TB1 from the constant voltage transformer of the 5-volt power supply. (See figure IV-1.) The tap adjustment compensates for manufacturing variations in the power supply components and the power distribution wiring to the backplane, backplane loading, and drift caused by temperature changes. Once the tap adjustment is made, no further adjustment is required unless the nominal backplane voltage limits are not met or there is a load change, a component change, or a defective component. Defective or out-of-tolerance components should be replaced.

ADJUSTMENT PREREQUISITES

The following operating conditions should be established, within the specified limits and as close to the nominal values as possible, before attempting the 5-volt adjustment:

a. Ambient room temperature: 50 to 104 degrees F.

b. Primary (main) input power voltage:

Domestic Units:	107 to 127 volts ac, rms. Measure voltage, under load, between terminals CB1-3 (line) and TB2-A5 (neutral).

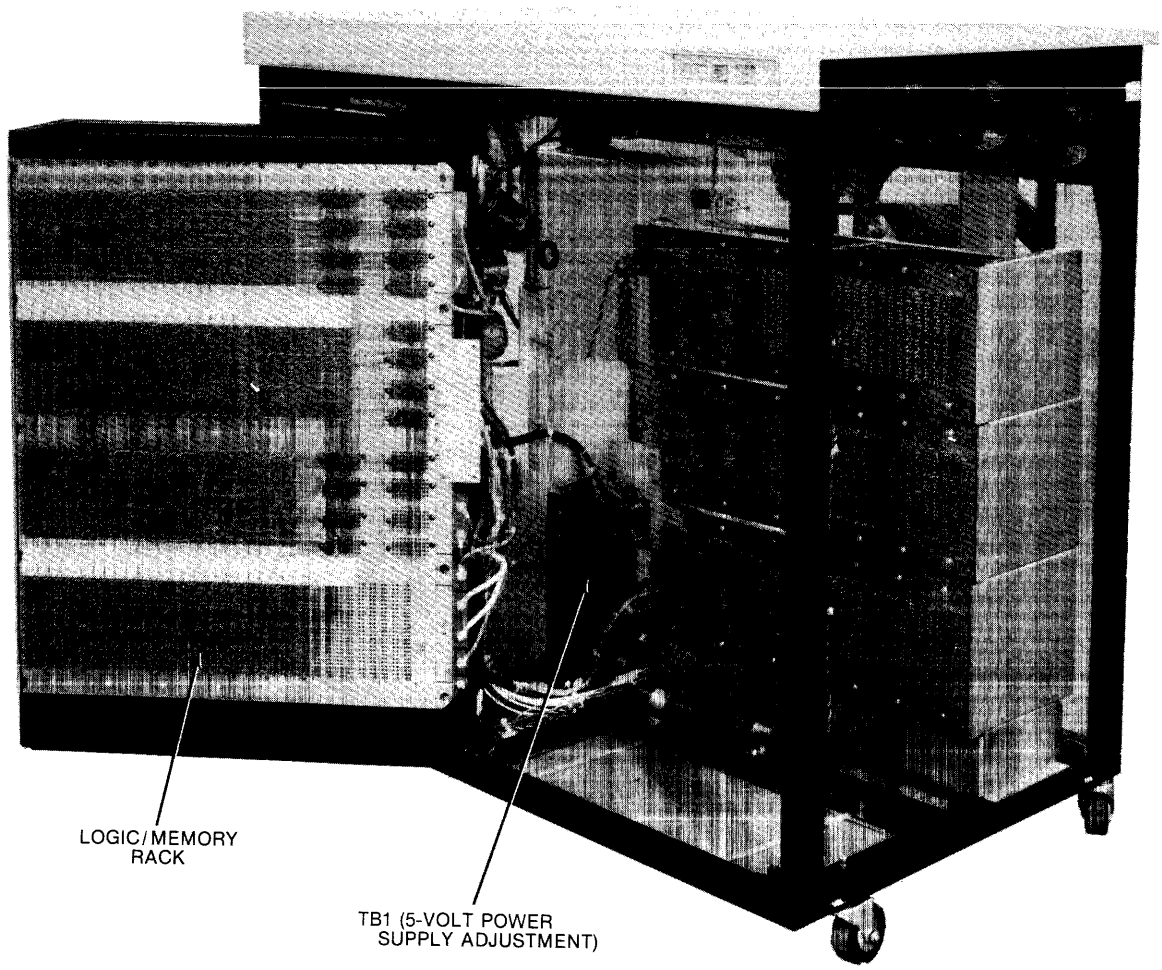


Fig. IV-1 LOCATION OF 5-VOLT POWER SUPPLY ADJUSTMENT

International Units:	+5 to -10 percent of the specified nominal service. Measure voltage, under load, between CB1-1 and CB1-3.

c. Primary (main) input power frequency: ±1 percent of specified value.

d. All printed circuit boards required for the installation should be installed in the processor to obtain normal operation loading.

Turn off primary power before proceeding with adjustment procedure.

ADJUSTMENT PROCEDURE

A dc voltmeter, with an accuracy of 0.2 percent within the 3.5 to 6.0 volt range, is required. Proceed as follows:

1. Count the total number of single L-type and memory printed-circuit cards installed in the processor. (A memory board is equivalent to three L-type cards; use the total equivalent L-type count for this adjustment.)

2. Make certain that power is turned off. Swing open logic/memory rack to gain access to adjustment area. (See figure IV-1.)

3. Preadjust + and - tap lead connections of 5-volt power supply constant-voltage transformer (CVT) to terminal board TB1 in accordance with table IV-1.

4. Turn on primary (main) input power and measure 5-volt supply at pins DD4A and DD6A (ground) on logic/memory backplane. Record this measurement for reference in subsequent steps.

5. Turn off power and adjust CVT tap connections in accordance with table IV-2 and as determined by the measurement obtained at step 4. Note the following:

a. Only one step adjustment is normally required to obtain the nominal 4.85 to 5.05 volt tolerance for a "hot" system, the 4.9 to 5.1 volt tolerance for a warm system, or the 4.95 to 5.15 volt tolerance for a cold system. If two tap steps produce a voltage within the specified tolerance band, select the tap which produces the higher voltage.

b. A single boost (+) step consists of: (1) moving the negative (-) lead to the next lower-numbered terminal if the positive (+) lead is on terminal 1 and the - lead is on a terminal other than 1, or (2) moving the + lead to the next higher-numbered terminal if the positive lead is on terminal 1.

c. A single buck (-) step consists of: (1) moving the + lead to the next lower-numbered terminal if the - lead is on terminal 1 and the + lead is on a terminal other than 1, or (2) moving the - lead to the next higher-numbered terminal is the + lead is on terminal 1.

d. After completing the required tap adjustments above, it is desirable to have at least one lead (+ or -) connected to terminal 1. This represents the preferred base condition from which initial any subsequent field adjustments are made. Other adjustments, where terminal 1 is not connected will work, but it is recommended that the preferred base connection is established as a standard.

6. After tap adjustments are completed, turn on primary power and measure the 5-volt supply at pins DD4A and DD6A (ground). Then measure the 5-volt supply at pins HA4A and HA6A (ground) and at pins HH3A and HH5A (ground). The average of these three backplane measurements shall be within 4.8 to 5.1 volts (hot system), 4.85 to 5.15 volts (warm system), or 4.9 to 5.2 volts (cold system).

7. Repeat tap adjustments, in accordance with step 4 and table IV-2 if the average backplane measurement is not within the specified tolerance band.

NOTE

If the number of installed printed-circuit cards is increased or decreased by more than six single L-type cards or two memory cards, a tap readjustment is required.

Table IV-1. TAP PREADJUSTMENT CONNECTIONS

Card Loading	Transformer to TB1 Terminal Connections	
(Note)	+ Lead	- Lead
50 to 102	1	2
103 to 159	1	1
160 to 210	2	1
211 to 270	3	1

NOTE: Number of equivalent single L-type printed-circuit cards installed. (Each memory card equals three L-type cards.)

Table IV-2. TAP ADJUSTMENTS

Memory Backplane Voltage (VDC)			No. of Tap Steps Required
Cold System (Note 1)	Warm System (Note 2)	Hot System (Note 3)	Buck (-) Boost (+)
4.08 - 4.24	4.03 - 4.19	3.98 - 4.14	+5
4.25 - 4.42	4.20 - 4.37	4.15 - 4.32	+4
4.43 - 4.59	4.38 - 4.54	4.33 - 4.49	+3
4.60 - 4.77	4.55 - 4.72	4.50 - 4.67	+2
4.78 - 4.94	4.73 - 4.89	4.68 - 4.84	+1
4.95 - 5.15	4.90 - 5.10	4.85 - 5.05	NONE
5.16 - 5.32	5.11 - 5.27	5.06 - 5.22	-1
5.33 - 5.50	5.28 - 5.45	5.23 - 5.40	-2
5.51 - 5.67	5.45 - 5.62	5.41 - 5.57	-3
5.68 - 5.85	5.63 - 5.80	5.58 - 5.75	-4
5.86 - 6.02	5.81 - 5.97	5.76 - 5.92	-5

NOTES:

1. A cold system is one which has been powered on for 30 minutes or less and was off for several hours preceding the power turnon.
2. A warm system is one which has been on for less than 2 hours and more than 30 minutes.
3. A hot system is one which has been on for 2 hours or more, or was off for 15 minutes or less after having been on for several hours.

BURROUGHS CORPORATION
DATA PROCESSING PUBLICATIONS
REMARKS FORM

TITLE: B 720 PROCESSOR FIELD ENGINEERING TECHNICAL MANUAL

FORM: 1077534
DATE: 7-30-75

CHECK TYPE OF SUGGESTION:

☐ ADDITION ☐ DELETION ☐ REVISION ☐ ERROR

GENERAL COMMENTS AND/OR SUGGESTIONS FOR IMPROVEMENT OF PUBLICATION:

FROM: NAME _____ DATE _____
TITLE _____
COMPANY _____
ADDRESS _____
_____

cut along ...ted line

STAPLE

FOLD DOWN SECOND FOLD DOWN

BUSINESS REPLY MAIL
First Class Permit No. 46, Wayne, Pennsylvania

Burroughs Corporation
200 West Lancaster Avenue
Wayne, Pennsylvania 19087

Attn: Systems Documentation
TIO EAST

FOLD UP FIRST FOLD UP