# CENTRAL CONTROL
# DESCRIPTION AND THEORY OF OPERATION
# "3B*" 20D MODEL 1 COMPUTER

CONTENTS PAGE



CONTENTS PAGE



* Trademark of Western Electric

CONTENTS PAGE



CONTENTS PAGE



Figures

CONTENTS PAGE



Tables



CONTENTS PAGE



## 1. GENERAL

**1.01** This section provides a physical and functional description and theory of operation of the central control (CC) unit used in the 3B 20D Model 1 computer.

**1.02** This section is reissued to include hardware changes to the CC. These changes are effective with the duplex multi-environment real time (DMERT) operating system generic program PG-4C004 (Generic 2). Revision arrows are used to emphasize the more significant changes. The specific reasons for this reissue are listed below:

(a) Change the circuit pack designation UN28 to UN28/UN28B

(b) Change the circuit pack designation UN44 to UN44/UN135

(c) Change the circuit pack designation UN18 to UN48 (this change is not related to Generic 2)

(d) Change the writable microstore description to include the 4K memory and the 4K memory addressing scheme.

### A. Purpose

**1.03** The CC, located in the control unit (CU) frame (Fig. 1), provides the high-speed CU functions (logic, control, and arithmetic processes) required by

the CU in the 3B 20D Model 1 computer. The CU consists of the following units:

(a) Direct memory access input/output (DMA I/O) unit

(b) CC unit

(c) Main store module (MASM). [The 3B 20D Model 1 computer may consist of one MASM (0) or two MASMs (0 and 1). The MASM is referred to as the main store (MAS).]

(d) Power units:

- 244D dc-to-dc converters (maximum of 4)
- J1C129AE (one ED-4C188 dc-to-dc converter and one 132AJ dc-to-dc converter)

(e) Cooling unit J1C129AF. (Three ED-4C191 fan units are located directly below the MAS.)

(f) Fuse panels.

### B. Configuration

**1.04** The 3B 20 computer may have two configurations:

(a) Simplex—3B 20S (one CU)

(b) Duplex—3B 20D (two CUs).

#### Simplex

**1.05** The 3B 20S computer configuration is used in applications where the computer is not performing service-affecting functions, and when only one CU is required to perform all operational functions.

#### Duplex

**1.06** The 3B 20D (duplicated) computer configuration uses two CUs (on-line and off-line). If a failure occurs in the on-line CU, the on-line CU is switched to off-line, and the off-line CU is switched to on-line. This automatic switching action reduces any interruption to the service. The duplex configuration is used for high reliability and continuity of service.

**1.07** The CC unit is comprised of the following subunits:

(a) Microstore (MIS)

(b) Central processing unit (CPU)

(c) Main store update unit (MASU)—required for direct memory access controller (DMAC) and/or duplex operation

(d) Maintenance channel (MCH)—required for duplex operation and optional for simplex operation

(e) Cache store unit (CSU)—optional; however, if not provided, a UN30 circuit pack must be installed

(f) I/O channels—two channel positions, 0 and 1

(g) Microlevel test set (MLTS) interface position

(h) Utility circuit (UC)—optional.

## 2. PHYSICAL DESCRIPTION

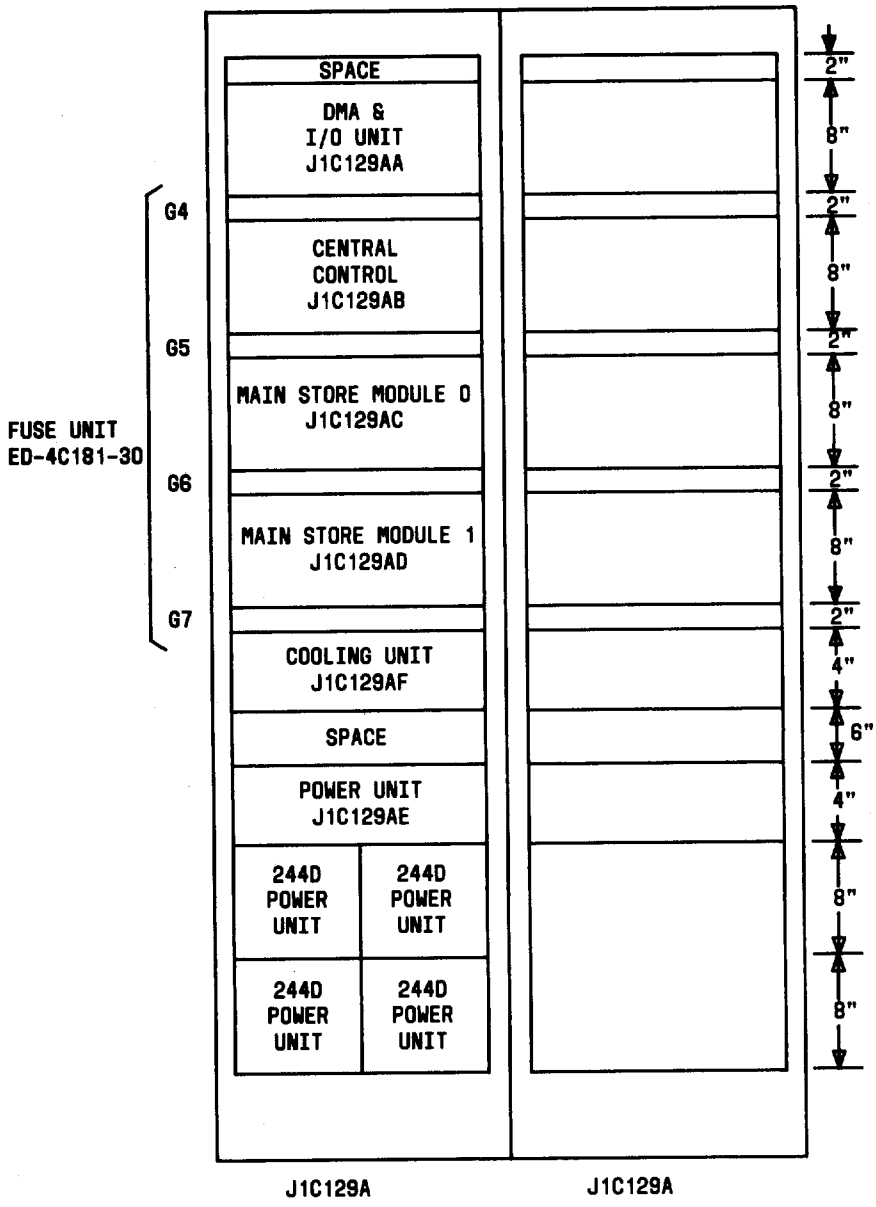
**2.01** The CC unit is 26 inches wide by 8 inches high. Refer to Fig. 2 for the circuit pack locations. The subunits consist of the following circuit packs:

(a) MIS—two to four circuit packs

- ▶UN48◀ (1) writable microstore (WMS)
- UN28 ▶or UN28B effective with Generic 2◀ (1 to 3) read-only memory (quantity used is dependent upon the application) (MIS)

(b) CPU—eight circuit packs

- UN44 ▶or UN135 effective with Generic 2◀ (1) microcontrol (MC)
- UN45 (1) store address translator (SAT)
- UN01 (1) data manipulation unit (DMU 0)
- UN23 (1) data manipulation unit (DMU 1)
- UN06 (1) store data control (SDC)
- UN02 (1) special registers (special register 0)




NOTES:
1. DUPLEX CONFIGURATION SHOWN
2. MINIMUM TWO 244D POWER UNITS REQUIRED
3. MAXIMUM CONTROL UNIT CONFIGURATION
4. COOLING UNIT MOUNTED BENEATH MAIN STORE MODULE 0 WHEN MAIN STORE MODULE 1 IS NOT USED

**Fig. 1—3B 20D Model 1 Computer Control Frame**

- UN03 (1) special registers (special register 1)
- UN43 (1) store address controller (SAC)

(c) MASU—one circuit pack

- UN34

(d) MCH—one circuit pack

- UN22

(e) CSU—three circuit packs

- UN11 (1) cache memory
- UN10 (2) cache controller

(f) I/O channel positions (two positions are provided in the CC unit). Any combination of the following circuit packs may be used:

- UN09 dual serial channel (DSCH)
- UN19 application channel interface (ACHI)
- UN26 serial channel (SCH)

(g) MLTS—one circuit pack

- UN16

(h) UC—one circuit pack

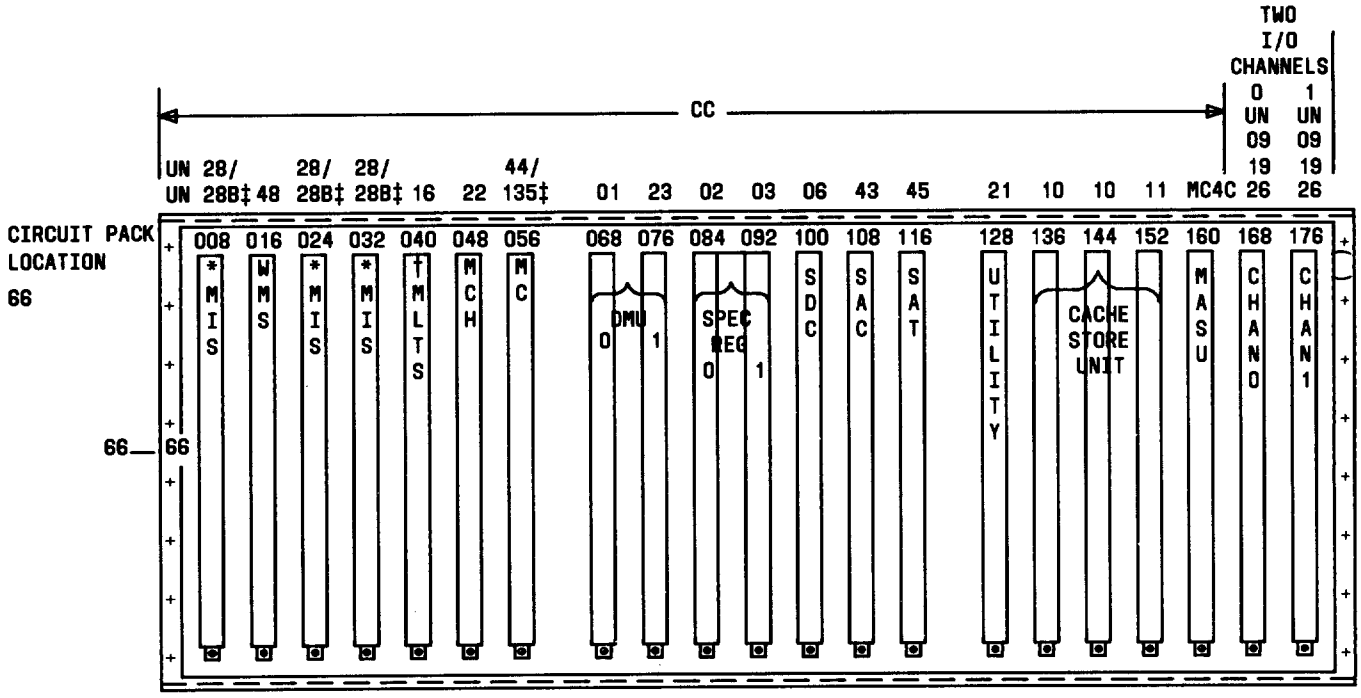
- UN21.

**2.02** The basic circuit pack used in the 3B 20D Model 1 computer measures 8 inches by 13 inches and is equipped with either a 200-pin connector (TN code series) or a 300-pin connector (UN code series).

**2.03** The circuit packages are standard transistor-transistor logic (TTL) circuits and TTL-compatible large scale integration circuits. Low power integrated circuit packs are used whenever possible to minimize power consumption and heat dissipation. High-speed integrated circuit packs are used where higher speed is required.

## 3. INTERFACES

**3.01** The CU interfaces to its periphery via the I/O channels (DMAC, DSCH, ACHI, and SCH) to the other CU via the MCH, and to the MLTS via the MLTS interface circuit (UN16) in the CC (Fig. 3).




* SYSTEM MAY CONTAIN 1-3 UN28s/UN28Bs
† POSITION PROVIDED FOR MICROLEVEL TEST SET UN16 INTERFACE
‡ EFFECTIVE WITH GENERIC 2

Fig. 2—Circuit Pack Locations

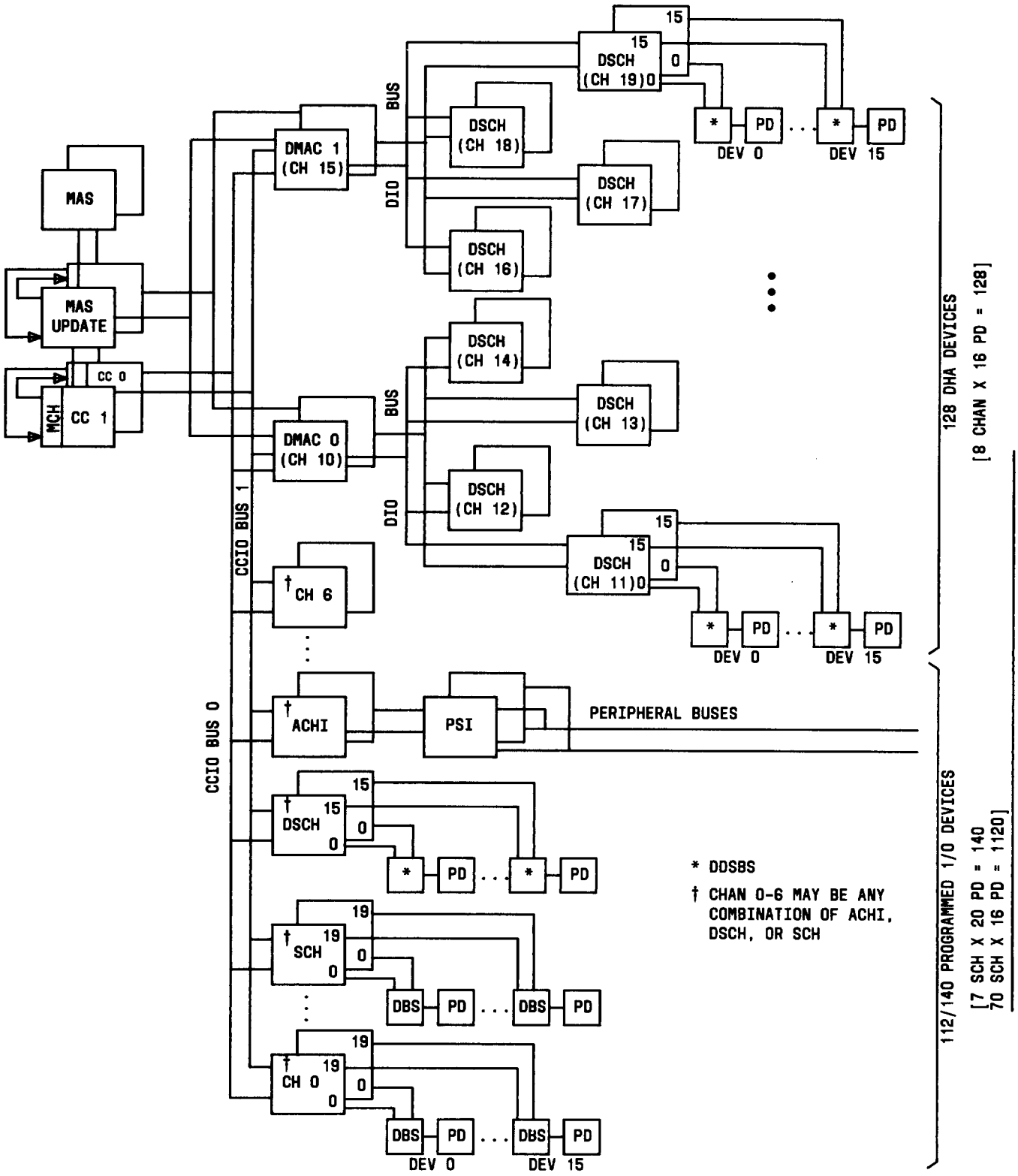



Fig. 3—3B 20D Model 1 Computer I/O Interfaces

**3.02** The CC communicates with the other CUs (I/O channels and the main store [MAS]) via buses. These buses are flat, ribbon-type cables terminated with connectors which plug onto the connector pins on the rear of the backplane. The CC interfaces to the other CC in the duplex configuration via a data link between the MCHs. Also, the CC is interfaced to the MLTS via the MLTS interface (UN16) circuit pack and its associated cables to the MLTS. This circuit pack and associated cables are part of the MLTS.

**3.03** Internal communication between the CC subunits is via internal buses comprised of printed circuit wiring within the backplane.

### A. Central Control Input/Output Bus

**3.04** The central control input/output (CCIO) bus is an 80-lead bus which provides the interface between the CC and the CU I/O channels. Refer to Fig. 4 for the connection of the CC to the I/O channels. The CCIO bus is comprised of the following leads:

(a) 12 control signal leads (CC to I/O devices)

(b) 3 response signal leads (I/O devices to CC)

(c) 36 data leads (32 data and 4 parity, bidirectional)

(d) 6 address leads (CC to I/O devices)

(e) 16 interrupt leads (I/O devices to CC)

(f) 1 channel error lead (I/O devices to CC)

(g) 6 address check leads (I/O devices to CC).

The remaining leads of the CCIO bus are designated as spares.

**3.05** The CU can be equipped with as many as seven programmed I/O channels (SCH, ACHI, DSCH) and a maximum of two DMACs. Programmed I/O channels are directly controlled by CC microcode via the CCIO bus. The DMAC provides the capability for direct memory transfers between the MAS and DMAC-controlled peripheral devices (PDs), reducing the real time for the CC to process I/O requests. The DMA I/O unit can contain one or two DMACs. Each DMAC is capable of autonomous control of a maximum of four DMAC channels (DSCH) via the data input/output (DIO) bus.

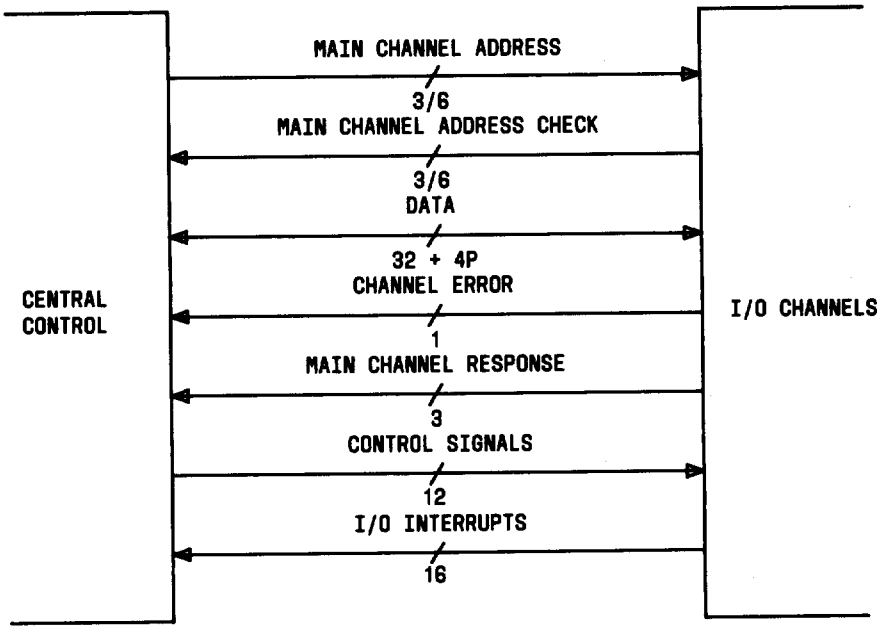



**Fig. 4—3B 20D Model 1 Computer CCIO Bus**

**3.06** The serial channel (SCH) interfaces the CC via the CCIO bus and interfaces its PDs via an ac bus comprised of two 100-ohm serial transformer-coupled coaxial cables. The PDs are low- and medium-speed peripheral units requiring a serial interface with the CC.

**3.07** The dual serial channels (DSCHs) also can be used as a program-controlled channel (not controlled by a DMAC). The DSCH interfaces the CC via the CCIO bus and interfaces the PD via 5-pair private serial data cables bus (private serial point-to-point link).

**3.08** The application channel interface (ACHI) interfaces the CC via the CCIO bus and the application device via differential dc paths comprised of eight 11-pair cables.

**3.09** All signals transmitted between the CC via CCIO bus to the I/O channels are active low (0 volt) using TTL levels. These signals are described in the following paragraphs.

#### Main Channel Address

**3.10** Each CCIO bus position has a unique 3-out-of-6 code (position-dependent, not device- or software-dependent). The CC selects a single I/O channel by transmitting the designated 3-out-of-6 code address on the six main channel address leads.

#### Data and Control

**3.11** Data and control information is transmitted from the CC to the addressed I/O channel via the (32 + 4 parity) data leads. Data and status are transmitted from the addressed I/O channel to the CC via these same leads. Odd parity over each of the data bytes (8 bits) is maintained in both directions, with the entire (32 + 4 parity) data word having even parity.

#### Channel Error

**3.12** The detection of failures in the I/O channel results in the channel error (CER) signal being set. This sets a bit in the CC error register, thereby resulting in an error interrupt.

#### Main Channel Acknowledge

**3.13** The I/O channel selected by the main channel address leads gates its 3-out-of-6 code main channel address onto the six main channel acknowledge leads in response to a control signal. A 3-out-of-6 code check circuit in the CC detects a no response or multiple channel response condition. This results in a bit being set in the CC error register, initiating an error interrupt.

#### Main Channel Responses

**3.14** The channel ready (RDY) signal indicates to the CC when an I/O operation has been completed. The CC transmits a read data control signal (RD) to the channel and tests the RDY signal to determine whether the operation was completed. Requests to read channel data will not initiate a RDY response until the data has been received from the PD, latched into the channel data buffer, and gated onto the CCIO bus. The RDY response to other control signals is determined by the channel addressed. The response is immediate to all other control signals addressed to channels on the CCIO bus. However, if the addressed channel is on the DIO bus (connected to a DMAC), RDY will not be returned at once because the DMAC cannot gate the command to the DSCH immediately. The microcode transfers to an RD loop waiting until the DMAC responds to the command with a RDY.

**3.15** An all-seems-well (ASW) signal is not used by the channel to signal the CC that an I/O operation was terminated because of an error condition in the PD. Data parity errors are detected by the CC and DMAC, but not the other channels. Errors signified by the CCIO bus ASW lead do not initiate hardware checks, but are registered in the condition cone register. A maintenance response signal is returned by the SCH (if a 101 return code was received from the PD). However, the DSCH, ACHI, and DMAC will never activate (set) the maintenance reset (MR) lead.

#### Control Signals

**3.16** The CC controls the operation of the I/O channels by transmitting control signals (pulses) to the channels. Each control signal initiates an operation to be performed by the channel. The 12 control signals are described in the following paragraphs.

**3.17** Write the channel control/address register (WCA) signal is used as the control signal to latch control information and device address data into the control/address register of the addressed I/O channel. The acknowledge (ACK) and the channel RDY response signals are returned to acknowledge receipt of the control signal. The channel reports an error via the channel error (CER) signal if its control/address register is presently full. Once the control information and address data have been successfully loaded into the control/address register, the channel initiates the requested operation (except for the ACHI).

**3.18** Write the channel data buffer (WD) signal is used as the control signal to latch data into the data buffer of the addressed I/O channel. Signals ACK and ASW are transmitted to the CC to acknowledge the receipt of the control signal. If the addressed channel (DSCH) is on the DIO bus, the DMAC gates the data into the DMAC CCIO data buffer register but does not transmit it to the channel until the receipt of a WCA signal. Once the WCA is received by the DMAC, it transmits data contained in the CCIO status buffer and the CCIO data buffer to the addressed channel (specified during the WCA operation).

**3.19** Read the channel data buffer (RD) signal generally follows a series of control pulses which have initiated an action in the channel. The CC utilizes the RD to indicate when the requested action is completed. At this time, the channel returns an RDY upon receipt of the RD signal. If the operation is supposed to return data, the channel data buffer register will send the results to the CC. If no errors were detected during the operation, ASW will also be returned to the CC. However, if a PD error is detected during the I/O operation, the ASW signal is not sent to the CC during the acknowledgment sequence. An MR is set by the SCH if it received an MR signal from the PD. The RDY signal is not transmitted to the CC if the channel data buffer register is not full (this indicates that the I/O operation is incomplete). The DMAC does not transmit the RDY signal for an RD signal to one of the DSCHs until the DSCH status and data have been read into the DMAC status and data buffer registers. All PD-to-DSCH communications initiate a read-the-channel-status-register (RST) signal to that channel by the DMAC before the RDY signal is returned to the CC.

**3.20** Read the channel status register (RST) signal, received from the CC by the addressed channel, results in the gating of the channel status onto the CCIO bus. Also, the channel sends the ACK, ASW, and RDY signals to the CC. If the addressed channel is connected to the DIO bus, the DMAC transmits the data in the CCIO status buffer register to the CC.

**3.21** The idle channel sequencer (IDLE) signal from the CC initializes the addressed channel. However, channel errors are not cleared.

**3.22** The receipt of a read channel service request (RSR) signal causes the addressed channel to gate the contents of its service request receivers onto the CCIO bus. Signals ACK, ASW, and RDY are transmitted to the CC to acknowledge receipt of the RSR signal (RDY is not provided for the DMAC channel). This signal (RSR) is only acknowledged by the DSCHs, ACHI, and DMACs (not by the SCH).

**3.23** The receipt of a read channel interrupt state (RINT) signal from the CC results in the channel gating its interrupt state onto the CCIO bus. The channel also transmits ACK, ASW, and RDY signals to the CC (RDY is not provided for the DMAC channel).

**3.24** When the clear channel errors (CLRER) signal is received from the CC, the channel initializes and clears its channel error registers. Then the channel transmits ACK, ASW, and RDY signals to the CC (RDY is not provided for the DMAC channel).

**3.25** Upon receipt of the I/O interrupt acknowledge (IACK) signal from the CC, each main channel gates its interrupt state onto an assigned data bit lead. Bit 0 is assigned to channel 0, bit 1 to channel 1, etc. Bits 10 through 19 are assigned to the DMAC and the associated I/O channels. Also, the channel responds to IACK by transmitting ACK, ASW, and RDY signals to the CC.

**3.26** Receipt of the channel error acknowledge (EACK) signal from the CC causes the channel to gate its error state onto assigned data leads. Bit 0 is assigned to channel 0, bit 1 is assigned to channel 1, etc. Bits 10 through 19 are assigned to the DMAC and the associated I/O channels. The main channels return ACK, ASW, and RDY signals to the CC.

**3.27** Receipt of the service request acknowledge (SRACK) signal from the CC causes the channel to gate its service request state onto assigned data leads. Bit 0 is assigned to channel 0, bit 1 is assigned to channel 1, etc. There is no response to this signal by the DMAC and the associated I/O channels. The addressed channels return ACK, ASW, and RDY signals to the CC. This signal (SRACK) is not acknowledged by the SCH.

**3.28** The I/O inhibit (INH) signal is generated by the CC in the form of a dc signal which is held active by the on-line CC to disable I/O operations in the off-line CC. To facilitate I/O channel diagnostics, INH can be made inactive on any channel via an override command loaded into the channel control/address buffer register.

### B. Main Store Bus

**3.29** The main store bus (MASB) provides the interconnection between the CC and MASM(s), and the DMAC and the MASM(s) via the main store update unit (MASU). Refer to Fig. 5. The MASB is an 80-lead, flat, ribbon-type cable terminated with connectors that plug onto the connector pins on the rear of the backplane. All signals transmitted between the CC and the MAS via the MASB are active low (0 volts) using TTL levels. Those cables not used for signals are designated as spares.

**3.30** The MASB is comprised of the following signal leads:

(a) 36 data leads (32 data bits + 4 parity bits - bidirectional)

(b) 5 control leads (CC to MAS)

(c) 1 control parity lead (CC to MAS)

(d) 27 address leads (24 address + 3 parity, CC to MAS)

(e) 4 error leads (MAS to CC)

(f) 1 store go (SGO) lead (CC to MAS)

(g) 1 store complete (SC) lead (MAS to CC).

**3.31** The data leads of the MASB are bidirectional and are used to gate data between the CC and the MAS. The address leads are used to gate the address of the specific location of data in the MAS. The control leads and the control parity lead signify to the MAS the type of operation to be performed (read/write, clear, byte, store half, or maintenance). The four error leads gate error data from the MAS to the CC. These four error leads are as follows:

(a) ***SERA (My Store Error):*** Hardware error detected in the MAS. When SERA is active, the CC performs a stop-and-switch.

(b) ***SERB:*** When active, SERB signals the CC of an inaccurate store address.

(c) ***SERC:*** When active, SERC signals the CC that, during a store read, byte write, or halfword write, the MAS detected a double or multiple error.

(d) ***SERD:*** This error signal may be active during a read, byte, or halfword operation. When active, it signals the CC that the data was correct before gating it to the CC or prior to byte or halfword write operations; but it is not correct now (the MAS may correct one bad data bit). Also, this error signal may be active during a refresh cycle when data is detected with an error condition.

**3.32** The SGO lead is active to initiate the MAS operations as designated by the CC. The SC lead is active, signaling the CC that the MAS cycle has been completed.

**3.33** The five control signals and their functions are as follows:

(a) Read/Write is active when the CC requests a read or write operation by the MAS.

(b) Byte is activated by the CC when a byte is to be written or during a read operation when the byte is to be cleared after the read operation is completed.

(c) Store Half is active when a halfword is to be written into the MAS or when a halfword is to be cleared after being read.

(d) Clear is active when a read and clear operation is requested by the CC in the MAS.

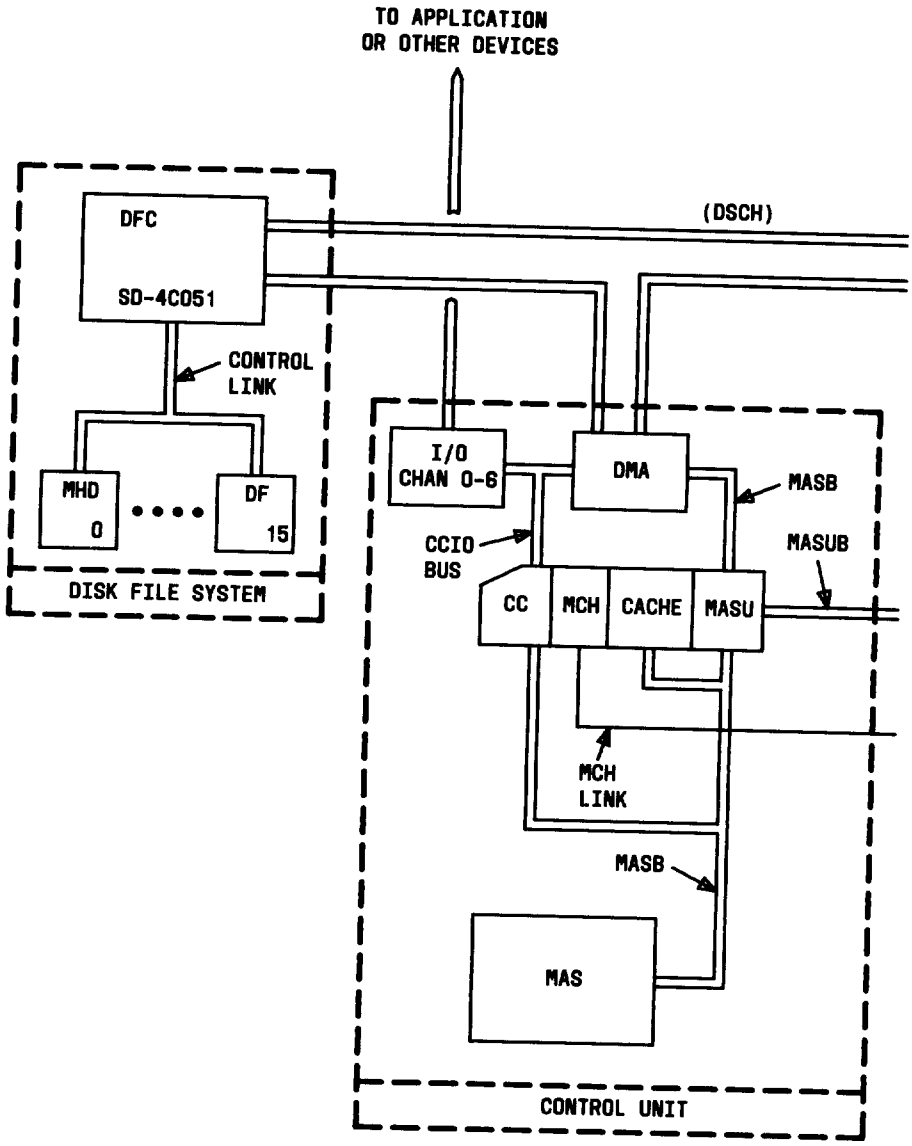



Fig. 5—3B 20D Model 1 Computer—Functional Overview (Sheet 1 of 2)

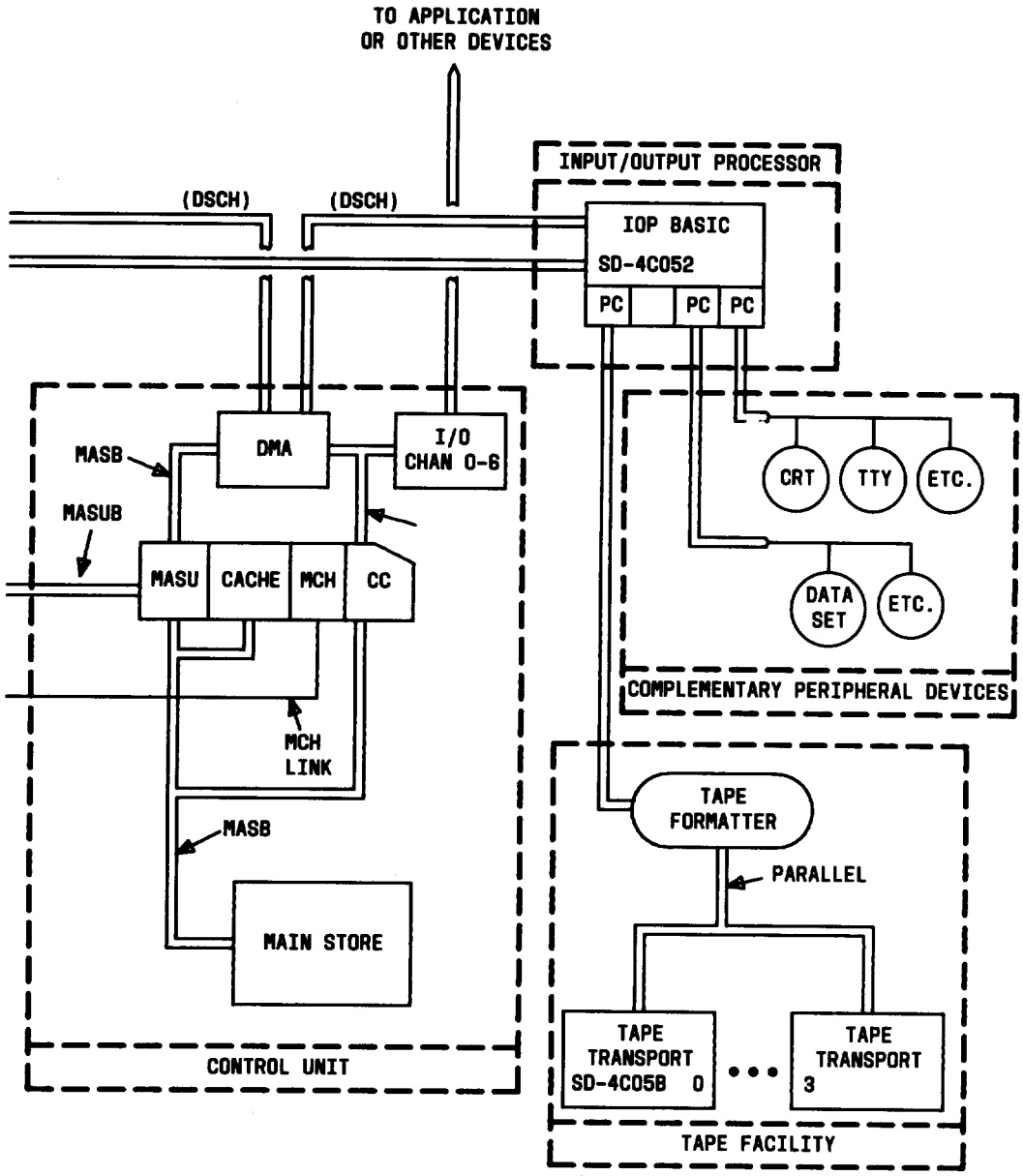



Fig. 5—3B 2D Model 1 Computer—Functional Overview (Sheet 2 of 2)

(e) Maintenance is active when the CC is diagnosing the MAS. The results of these diagnostics are gated to the CC.

### C. Maintenance Channel to Maintenance Channel

**3.34** The maintenance channel (MCH) is equipped with two ports, A and B. Port A interfaces with the CC, while port B interfaces a DSCH in another CU.

**3.35** The MCH (on-line CU) to MCH (off-line CU) communications are via four pairs of leads:

(a) Data high (DAH)—high half of data word

(b) Data low (DAL)—low half of data word

(c) Clock (CLK)—clock signal from transmitting MCH

(d) Request (REQ)—request for stop-and-switch of CUs.

**3.36** The internal connections between the CC and the MCH is via internal buses. These connections are provided by registers having the same names as the buses to which the registers are connected. These buses are as follows:

(a) Microstore address bus (MSA)

(b) Microstore data bus (MSD)

(c) Bidirectional gating bus (BGB)

(d) Maintenance bus (MTCB).

**3.37** The MSA bus provides a data path by which the diagnostics access the microstore. The MSD bus provides a data link by which the diagnostics load a microinstruction into the microstore instruction register (MIR) or read a microinstruction in the microstore. The BGB provides access to the writable microstore. The MTCB provides an access for the diagnostics to read the following:

(a) Source bus (SRC)

(b) Destination bus (DST)

(c) Program status word register (PSW)

(d) Hardware status register (HSR)

(e) System status register (SSR)

(f) Error register (ER).

### D. Central Control to Microlevel Test Set

**3.38** The microlevel test set (MLTS) interface circuit pack is part of the MLTS and must be installed in the CC when the MLTS is used. The MLTS interface provides the interface between the CC and the MLTS for "dead start" maintenance capability. The MLTS interface circuit provides the MLTS with access to the following CC internal buses:

(a) Microstore address bus (MSA)

(b) Microstore data bus (MSD)

(c) Bidirectional gating bus (BGB)

(d) Maintenance bus (MTCB).

**3.39** The MSA bus provides a communication path which enables the MLTS to access the microstore. The MSD bus provides a data path which enables the MLTS to load a microinstruction into the microinstruction register (MIR) or read a microinstruction in the microstore. The BGB provides the MLTS access to the writable microstore. The MTCB provides the MLTS with access to read the following:

(a) Source bus (SRC)

(b) Destination bus (DST)

(c) Program status word register (PSW)

(d) Hardware status register (HSR)

(e) System status register (SSR)

(f) Error register (ER).

### E. Central Control Internal Buses

**3.40** Internal communications between CC units are via internal buses which are comprised of printed circuit wiring within the backplane. These buses are as follows:

(a) Bidirectional gating bus (BGB)

(b) Microstore address bus (MSA)

(c) Microstore data bus (MSD)

(d) Source bus (SRC)

(e) Destination bus (DST)

(f) Maintenance bus (MTCB)

(g) Cache data bus

(h) Cache address bus.

**3.41** The internal buses provide the data paths between units in the CC. See Table A.

## 4. FUNCTIONAL DESCRIPTION

**4.01** The CC is comprised of the following subunits:

(a) Microstore (MIS)

(b) Central processing unit (CPU)

(c) Main store update unit (MASU)—DMAC/ duplex operation

(d) Maintenance channel (MCH)—duplex or simplex operation

(e) Cache store unit (CSU)—optional

(f) I/O channels—two positions

(g) Microlevel test set (MLTS) interface position

(h) Utility circuit (UC)—optional.

### A. Microstore

**4.02** The microstore (MIS) unit is comprised of a writable microstore (WMS) ▶UN48◀ and from one to a maximum of three read-only MIS UN28s, ▶or UN28Bs effective with Generic 2,◀ dependent upon the application.

**TABLE A**

**CENTRAL CONTROL INTERNAL BUSES**

| BUS | UNITS CONNECTED BY BUS |
|---|---|
| BGB (36 leads) | Writable microstore, maintenance channel, special registers, and utility circuit |
| MSA (36 leads) | Microstore, microcontral, maintenance channel, and writable microstore |
| MSD (64 leads) | Microstore, microcontrol, maintenance channel, and writable microstore |
| SRC (36 leads) | Data manipulation unit, registers |
| DST (36 leads) | Data manipulation unit, registers |
| MTCB (36 leads) | Maintenance channel, registers, source bus, destination bus, and internal check points |
| Cache Data Bus (64 leads) | Microstore, microcontrol, maintenance channel, writable microstore, cache |
| Cache Address Bus (36 leads) | Store address control, store address register, store address translator, cache |

**Writable Microstore**

**4.03** The writable microstore (WMS) is used to store microinstructions in the form of microsequences, special diagnostic microcodes, and control unit sequences. The WMS contains 1K ▶(4K effective with Generic 2)◀ of writable, random access memory (RAM), which is loaded with microinstructions by either the CC or the MCH via the bidirectional gating register (BGR, UN01 and UN23) and the bidirectional gating bus (BGB). The WMS is addressed by the microcontrol (UN44 ▶or UN135 effective with Generic 2◀) via the microstore address bus (MSA). Data from the WMS is gated to the microinstruction register (MIR) in the microcontrol.

**Read-Only Microstore**

**4.04** Each read-only MIS contains a maximum of 4K of preprogrammed read only memory (ROM) used to store microinstructions in the form of microsequences. The MIS is addressed via the MSA bus by the microcontrol. Microinstructions addressed in the MIS are gated from the MIS to the MIR in the microcontrol via the microstore data (MSD) bus.

**B. Central Processing Unit**

**4.05** The central processing unit (CPU) is comprised of the following circuits:

(a) Microcontrol (MC)—UN44 ▶or UN135 effective with Generic 2◀

(b) Store address translator (SAT)—UN45

(c) Store address control (SAC)—UN43

(d) Store data control (SDC)—UN06

(e) Data manipulation unit (DMU)—UN01 and UN23

(f) Special registers—UN02 and UN03.

The CPU performs arithmetic operations, controls instruction processing, provides timing functions, and provides storage. Figure 6 shows the block diagram of the internal CC architecture.

**4.06** Microcontrol (MC) UN44 ▶or UN135 effective with Generic 2◀ contains a microstore address sequencer, microinstruction register (MIR), destination decoders, and source decoders. Functions provided for by the MC are as follows:

(a) Translates the Opcode of the store instruction into a microstore address

(b) Reads the microstore

(c) Decodes the microinstruction and gates the data onto the designated leads to perform the function of the store instruction

(d) Initiates the decoder outputs for the destination data (gates data from the DST bus to the designated register)

(e) Initiates the decoder outputs for the source data (gates data from the designated register to the SRC bus).

**4.07** The following units of the CPU are considered the store interface circuits:

(a) Store address translator (SAT)—UN45

(b) Store address controller (SAC)—UN43

(c) Store data control (SDC)—UN06.

**4.08** The SAT (UN45) contains two address translation buffers A and B (ATBA and ATBB, respectively) which are addressed by the store address register (SAR) (bits 23 through 11), and parity registers for the interrupt mask (IM) register, channel data register (CDR), pulse point register (PPR), and PSW register. The SAT is used to translate a virtual address in the SAR to a physical address in the main store (MAS).

**4.09** The SAC (UN43) contains the program address register (instruction being used by halfword multiplexer and the instruction buffer), the SAR (address of the next instruction to be used by store instruction register), and the store control register (control bits for operation of the store interface circuits). The SAC provides the store address and control functions for the store interface circuits.

**4.10** The SDC (UN06) contains the store data register (SDR) (data being used), store instruction register (the next instruction to be used), halfword multiplexer (the instruction being used, Opcode), and the instruction buffer (IB) (the instruction being

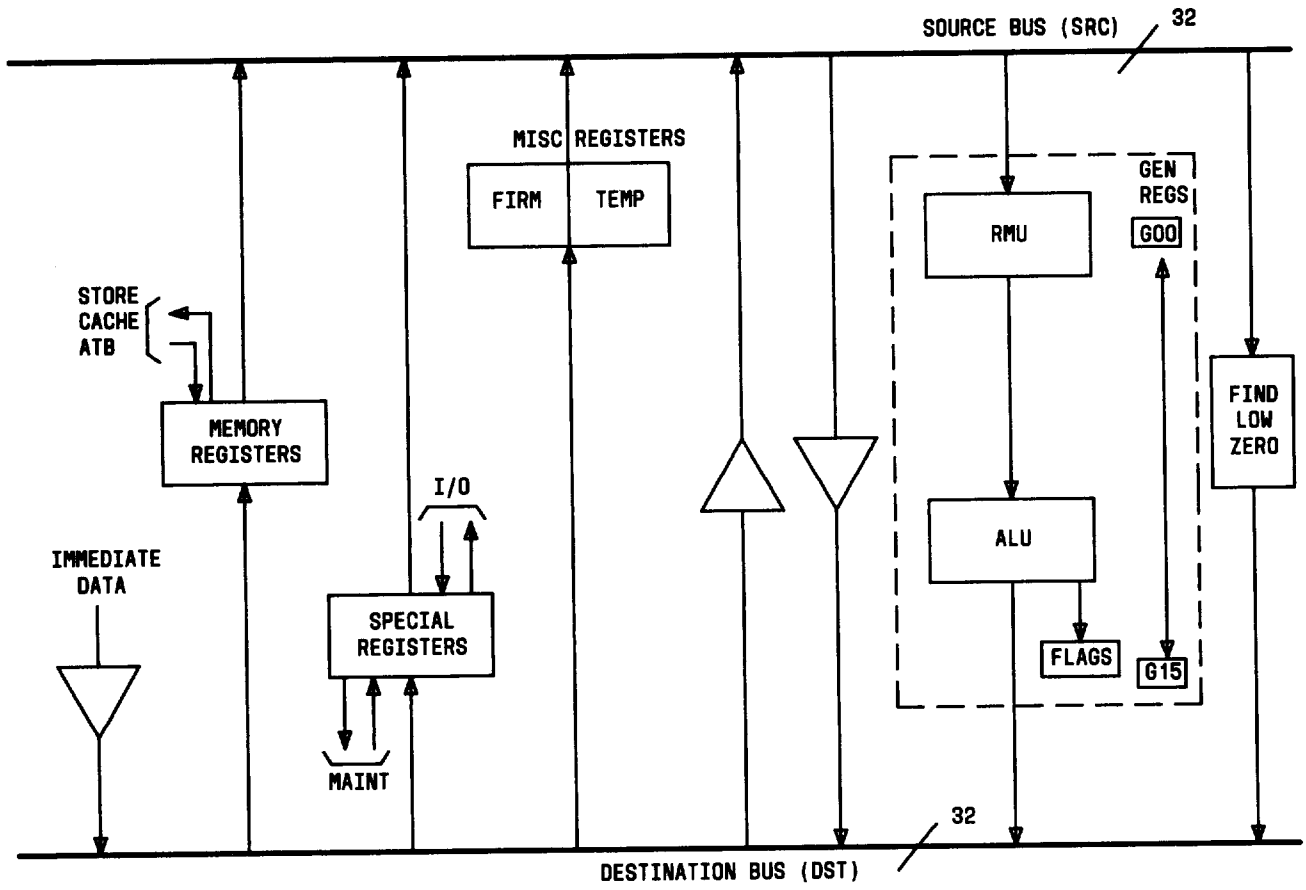



**Fig. 6—Internal CC Architecture Block Diagram**

used, data). Functions provided by the SDC are the interface and control for the store data and store instructions.

**4.11** The data manipulation unit (DMU) is comprised of two units (DMU 0, UN01 and DMU 1, UN23).

**4.12** The DMU 0 contains the following circuitry:

(a) Rotate unit

(b) Mask unit

(c) Arithmetic logic unit (ALU)

(d) Parity generator

(e) Matcher circuit

(f) Bypass ALU circuit

(g) Bidirectional gating register (BGR) (bits 35 through 32)

(h) Registers (temporary, firm, and general).

**4.13** Functions provided by DMU 0 are:

(a) Rotate right (bits 0 through 31)

(b) Bypass ALU (bits 35 through 32)

(c) Mask unwanted data

(d) Arithmetic logic functions

(e) Generation of parity bits

(f) Store for bits 35 through 32 (temporary, firm, and general)

(g) Interface between BGR and BGB.

**4.14** The DMU also contains the temporary, firm, and general registers, and the BGRs which are 36-bit registers. These registers are located in the DMU circuit packs. Bits 35 through 32 (parity) are in DMU 0 (UN01) and bits 31 through 00 are in DMU 1 (UN23). Also, the temporary and firm registers use only 16 bits (bits 15 through 00) of the 36-bit register.

**4.15** The DMU 1 contains the following circuitry:

(a) Find low zero logic circuit

(b) Mask unit

(c) ALU

(d) Bypass ALU circuit

(e) BGR (bits 31 through 00)

(f) Registers (temporary, firm, and general).

**4.16** The DMU 1 performs the following functions:

(a) Finds low zero on the source (SRC) bus

(b) Bypasses ALU (bits 31 through 00)

(c) Masks unwanted data

(d) Performs arithmetic logic operations

(e) Stores data (bits 31 through 00) in the temporary, firm, and general registers

(f) Interfaces between BGB and BGR.

**4.17** The DMU (0 and 1) is connected to the designated circuit via the DST bus and the SRC bus.

**4.18** The rotate unit functions to rotate bits, either 0, 8, 16, or 24 bit positions, as designated by MIR bits 37 through 33. The mask unit functions to blank or inhibit unwanted bits. The ALU is used to add, subtract, or logically combine data, then gate the output to the specified circuit. The find-low-zero circuit activates a signal to the destination register, indicating the least significant bit position of the first zero found in a source register (the first zero establishes priority of the operation). The BGR is used as a 2-way interface buffer between buses (BGB, SRC, and DST).

**4.19** The temporary register is comprised of eight $16 \times 4$ RAMs. Each RAM provides 4 bits of the 16 temporary registers ($8 \times 4 = 32$). The firm registers are composed of eight $4 \times 16$ RAMs in parallel that provide sixteen 32-bit registers. The matcher circuit functions as a comparator circuit; if inputs are not matched, an error condition exists. The bypass ALU circuit is used to bypass data from the ALU as required.

**4.20** Special registers (UN02 and UN03) contain 32-bit registers with bits 15 through 00 on UN02 (special register 0) and bits 31 through 16 on UN03 (special register 1). These registers are as follows:

(a) Hardware status register (HSR)—control of CC hardware

(b) System status register (SSR)—status of 3B 20D Model 1 computer

(c) Error register (ER)—error conditions

(d) Timer—timing signals

(e) Real-time counter—counts real-time

(f) Channel data register (CDR)—I/O data

(g) Interrupt mask (IM)—inhibits interrupt

(h) Interrupt set (IS)—initiates interrupt

(i) Pulse point register (PPR)—miscellaneous control points

(j) Program status word (PSW)—control of 3B 20D Model 1 computer software.

**4.21** The special registers provide storage for control, status, I/O data, interrupt control, and error data. Additionally, the special registers provide timing functions for the CC.

### C. Main Store Update Unit

**4.22** The main store update unit (MASU, UN34) is an optional unit required for direct memory access controller (DMAC) or duplex operation. The MASU functions as a logic circuit to arbitrate between the CC and DMAC as to which unit receives priority to access the main store bus (MASB). The MASU enables the data and address leads of the

MASB separately to ensure accurate data operation. Also, the MASU ensures that data changing the contents of any main store (MAS, CU 0 or CU 1) address is available to each MAS, so that each CU MAS contains the identical data.

**4.23** The MASU contains the following circuitry:

(a) MASB control

(b) Store direction and protection

(c) Update synchronization.

The MASU is connected to the MASU in the other CU via the MASU bus (duplex configuration), and to the CC, MAS, and DMAC via the MASB.

### D. Maintenance Channel

**4.24** The maintenance channel (MCH, UN22) provides the interface between the CCs in a duplex configuration via a data link between CCs. The MCH provides for the following:

(a) Monitoring the other CC

(b) Disabling/enabling the other CC

(c) Reading/writing the other CC microstore

(d) Loading and executing microinstructions in the other CC

(e) Loading and executing diagnostic microsequences in the other CC

(f) Stopping-and-switching control.

**4.25** The MCH contains the master and slave control circuitry, programmable read only memory (PROM) sequencer, stop-and-switch circuitry, and the bus interface circuitry. The MCH in each CC is connected to the other via four pairs of cables (data link). These pairs are as follows:

(a) Data high (DAH)

(b) Data low (DAL)

(c) Clock (CLK)

(d) Request (REQ).

**4.26** The MCH is interfaced to the CC via the bidirectional gating bus (BGB), microstore data bus (MSD), microstore address bus (MSA), and the maintenance bus (MTCB).

**4.27** The MCH has two ports, A and B. Port A is used for the duplex configuration. Special applications that require a DSCH to be connected to the MCH (for maintenance) use port B.

### E. Cache Store Unit

**4.28** The cache store unit (CSU) is an optional unit used to increase the real-time capabilities of the CU by reducing the access time of frequently used data. The CSU operates at 250 nanoseconds (ns) and the MAS operates at 850 ns. The CSU is comprised of a cache controller (two UN10s) and a cache memory unit (one UN11). The CSU is a temporary (changeable) store area for the contents of frequently read MAS locations (until overwritten). The CSU is connected to the CC via the cache address bus, cache data bus, and the cache control leads. Also, the CSU is connected to the MASU via the MASB (data, address, and control). If the CC is not equipped with a CSU, a UN30 interface circuit pack must be used as an interconnection circuit between the cache data bus (always equipped) and the MASB.

**4.29** The cache controller provides control of the CSU. The cache controller contains memory storage areas for addresses, which are divided into four memory modules (A through D). Each module contains 516 (16-bit) address words.

**4.30** The cache memory provides a storage area for data words. The memory storage area is divided into four memory modules (A through D). Each module contains 1024 (36-bit) words (32 data and 4 parity).

### F. Input/Output Channels

**4.31** Two positions are provided for input/output (I/O) channels in the CC unit. These two positions (0 and 1) may be any combination of SCH, DSCH, or ACHI. The I/O channels are part of the 3B computer I/O interfaces and are not considered part of the CC. However, the I/O channels are provided space and power in the CC unit.

### G. Microlevel Test Set Interface

**4.32** The microlevel test set (MLTS) interface (UN16) provides an interface between the CU and the MLTS. The MLTS is used to remote test a 3B 20D computer by a host computer or for on-site manual testing of the 3B 20D computer. The MLTS interface contains bus interfacing circuits to access the CC buses.

**4.33** The MLTS interface has access to the following buses (and circuits connected to these buses):

(a) Microstore address bus (MSA)

(b) Microstore data bus (MSD)

(c) Maintenance bus (MTCB)

(d) Bidirectional gating bus (BGB).

### H. Utility Circuit

**4.34** The utility circuit (UC, UN21) is an optional circuit pack that provides the interfacing for software testing and debugging. The UC monitors operations between the CC and the MAS. The UC contains matchers, memory, and control registers. Matchers compare address and data information. The memory provides storage area for data. Control registers provide control and timing over the UC functions.

### I. Overview

#### Registers

**4.35** Registers are located in the special register (UN02 and UN03), store data control (SDC, UN06), data manipulation unit (DMU, UN01 and UN23), and store address control (SAC, UN43). These registers can be labeled as memory registers, special registers, and miscellaneous registers. Each register can be loaded from the DST bus and enabled to the SRC bus under microprogram control. Figure 7 is a summary of the CC registers.

**4.36** The memory registers are interfaced to the main program store, memory management, and cache logic. These registers are as follows:

(a) Store data register (SDR)

(b) Store instruction register (SIR)

(c) Store address register (SAR)

(d) Program address register (PA)

(e) Instruction buffer (IB)

(f) Store control register (SCR).

All of these registers contain 36 bits (32 data bits plus 4 parity bits), except the PA register which is a 27-bit register (24 data bits plus 3 parity bits).

**4.37** The special registers are used for special data that is used throughout the CC. The special registers are as follows:

(a) Program status word (PSW)

(b) Pulse point register (PPR)

(c) System status register (SSR)

(d) Hardware status register (HSR)

(e) Error register (ER)

(f) Interrupt mask (IM)

(g) Interrupt set (IS)

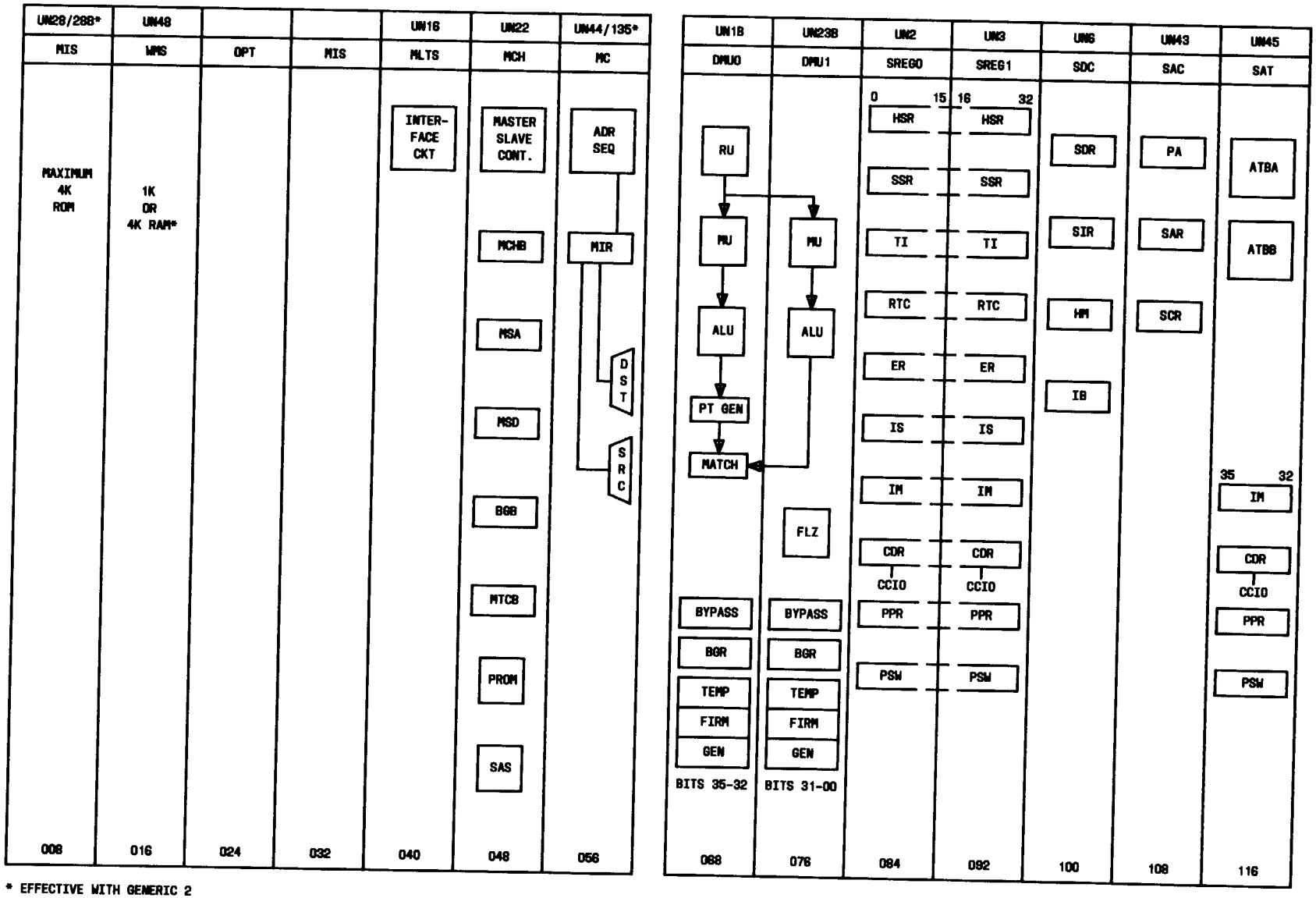
(h) Channel data register (CDR)

(i) Bidirectional gating register (BGR)

(j) Real-time clock (RTC).

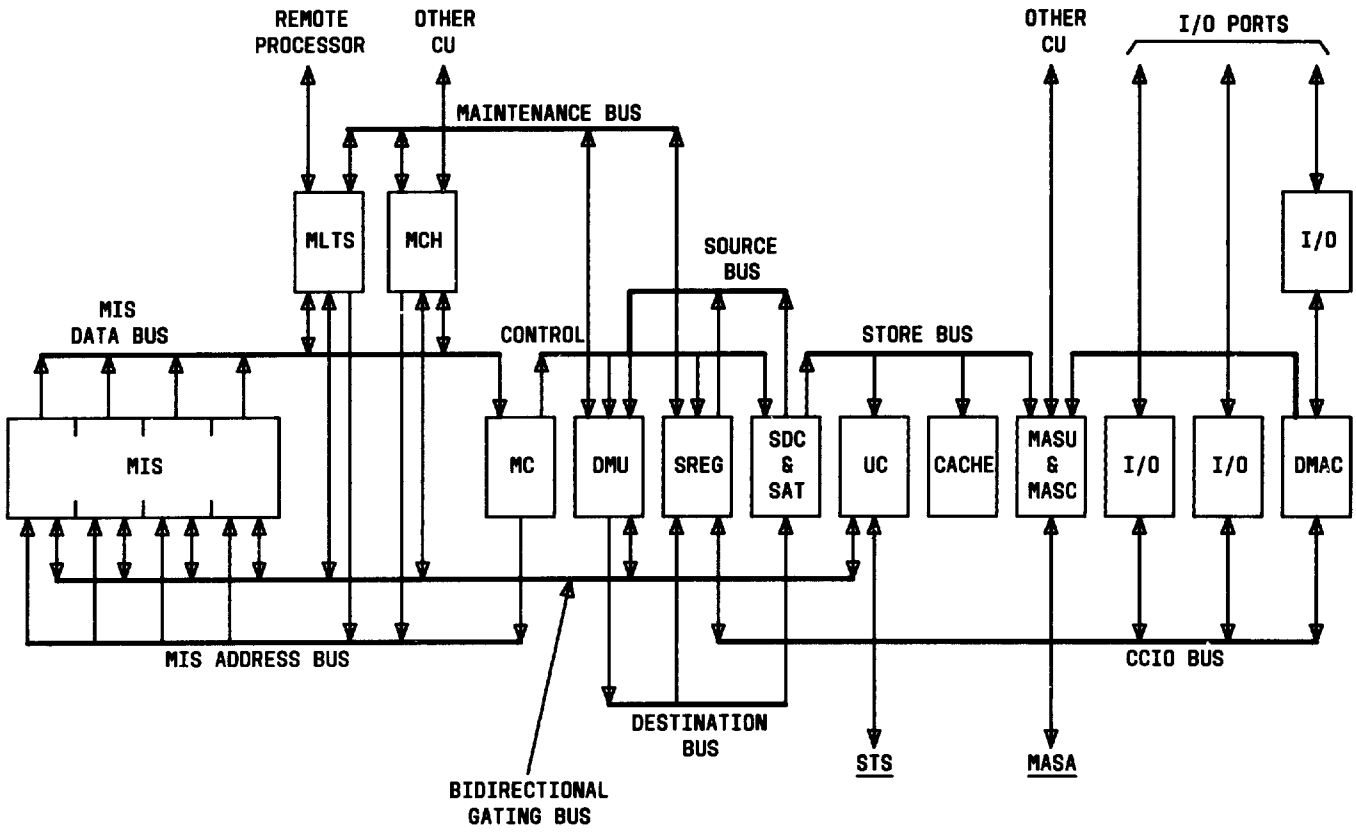
Figure 8 shows the register interconnections.

**4.38** The miscellaneous registers are two sets of 16 registers designated as temporary and firmware registers ($2 \times 16 = 32$). These registers consist of random access memory (RAM) circuits. The temporary registers are used as scratch pad registers. Temporary registers 0 through 7 are used for microcode processing and registers 8 through 15 are used for data storage during memory management microroutines. The firmware registers are used by the store address translator (SAT). Firm registers 0 through 7 are designated as either primary or secondary segment base registers. Table B lists the destinations of the temporary and firm registers. Figure




* EFFECTIVE WITH GENERIC 2

Fig. 7—CC Register Summary




LEGEND:

| | |
|---|---|
| CCIO | CENTRAL CONTROL INPUT/OUTPUT BUS |
| CU | CONTROL UNIT |
| DMAC | DIRECT MEMORY ACCESS CONTROLLER |
| DMU | DATA MANIPULATION UNIT |
| I/O | INPUT/OUTPUT |
| MASA | MAIN STORE ARRAY |
| MASC | MAIN STORE CONTROLLER |
| MASU | MAIN STORE UNIT |
| MC | MICROCONTROL |
| MCH | MAINTENANCE CHANNEL |
| MIS | MICROSTORE |
| MLTS | MICROLEVEL TEST SET |
| SAT | STORE ADDRESS TRANSLATOR |
| SDC | STORE DATA CHANNEL |
| SREG | SOURCE REGISTER |

**Fig. 8—Register Interconnections**

9 is a block diagram of the temporary, firm, and BGRs.

### Microinstructions

**4.39** The 64-bit microinstruction readout of the microstore (MIS) is decoded to provide control signals for CC operation. Execution of a main store (MAS) instruction requires a series of microinstructions (to complete the task) to be read out of the MIS, decoded and executed. Refer to Fig. 10 and Table C for the microinstruction format.

**TABLE B**

**TEMP AND FIRM REGISTER DESIGNATIONS**

| TEMP REGISTERS | |
|---|---|
| **REG. NO.** | **DEFINITION** |
| 0-7 | Microcode Processing |
| 8 | Scratch 0 |
| 9 | Scratch 1 |
| 10 | ATB—SAR |
| 11 | ATB—SDR |
| 12 | ATB—SCR |
| 13 | ATB—Q |
| 14 | ATB—PSW |
| 15 | ATB—BGR |
| **FIRM REGISTERS** | |
| 0 | SBR* |
| 1 | SBR |
| 2 | SBR |
| 3 | SBR |
| 4 | SBR |
| 5 | SBR |
| 6 | SBR |
| 7 | SBR |
| 8 | SYSBASE |
| 9 | TOP-IS |
| 10 | TOP-KS |
| 11 | DSR 0 |
| 12 | DSR 1 |
| 13 | DSR 3 |
| 14 | DSR 3 |
| 15 | Hold Get (Not Used) |

* SBR represents segment base registers.

**4.40** Six types of microinstructions are used in the CC:

- Manipulate
- Xmove
- Immediate data
- Find low zero
- Jump
- Call.

**4.41** A single microinstruction may contain the following operands and operations depending on the type of instruction. These operands and operations are listed in the order of occurrence during the execution of a microinstruction.

(1) Register source operand

(2) Rotate operation

(3) Constant source operand

(4) Mask operation

(5) Second register source operand

(6) Arithmetic or logic operation

(7) Carry operation

(8) Shift operation

(9) Register destination operand

(10) Flag operation.

**4.42** ***Manipulate*** microinstructions are used to gate data in a register to the rotate mask unit (RMU) and the arithmetic logic unit (ALU), then store the processed data in a destination register. ***Xmove*** microinstructions gate data in a source register directly to a destination register. ***Immediate data*** microinstructions contain 32 bits of data which are used in the current operation. ***Find low zero*** tests the contents of a register to find the lowest bit position which contains a zero. ***Jump*** microinstructions test some specified data bit or flag; then, as the result of the test, branch to another series of microinstructions. ***Call*** microinstructions are used to call microprograms.

**4.43** The microinstruction (64 bits) is segmented as follows: Bits 0 through 7 are reserved for the next address (NA) field in each microinstruction. Bits 44 through 47 define the microinstruction type. Bits 48 and 49 define the execution time allotted to the microinstruction in 50-ns increments. Bits 50 through 55 define store operations and some miscellaneous functions in the computer. Bits 56 through 63 are parity and check bits. These 28 bits are used in all microinstructions. The functions of the remaining 36

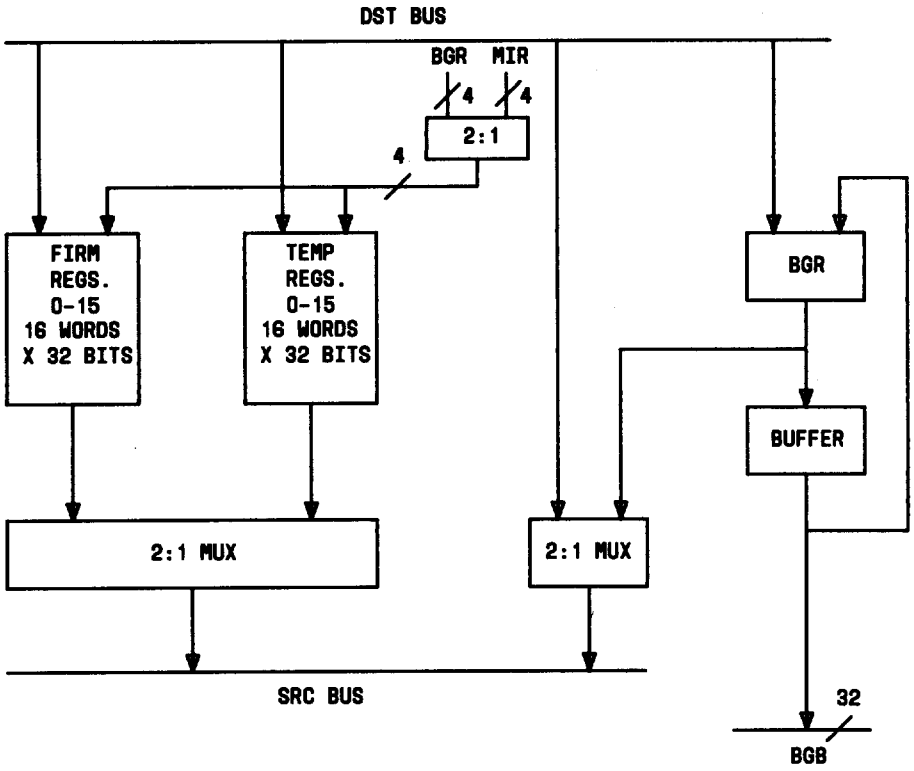



**Fig. 9—Temp, Firm, and BGR Registers**

bits depend on the type of microinstruction. Refer to Fig. 10 and Table C.

**4.44** The RMU is inserted in the data path between the source bus and the ALU. The RMU can preprocess data going to the ALU by rotating the data from 0 to 31 places, and by masking a data word with a predefined mask. There are 512 mask words available to use in the mask operation.

**4.45** The RMU performs three separate functions: byte rotations, bit rotations, and masking. Refer to Fig. 11. Bytes of an input data word are rotated in the byte rotator circuit. Output of the byte rotator is applied to the bit rotator, which completes the 0 through 31 place rotation of the data word. The rotated word then may be ANDed or ORed with any one of the 512 mask words. Figure 12 illustrates the byte rotation logic. The bit rotator is shown in Fig. 13.

**4.46** The mask words are stored in a 512-word by 32-bit ROM. Implementation of the mask function is accomplished by 32 AND-OR-INVERT gates at the input of the ALU. A lead from the microinstruction register (MIR) controls the type of operation to be performed. A logic "1" on this lead causes an AND, while a logic "0" causes an OR function to be performed.

**4.47** The ALU is comprised mainly of an array of eight bipolar 4-bit computer-integrated circuits (2901A). A 2901A integrated circuit has three distinct elements: a dual-port 16-word RAM, an arithmetic-logic section, and a temporary (Q) register. Any of the 16 words stored in the RAM may be

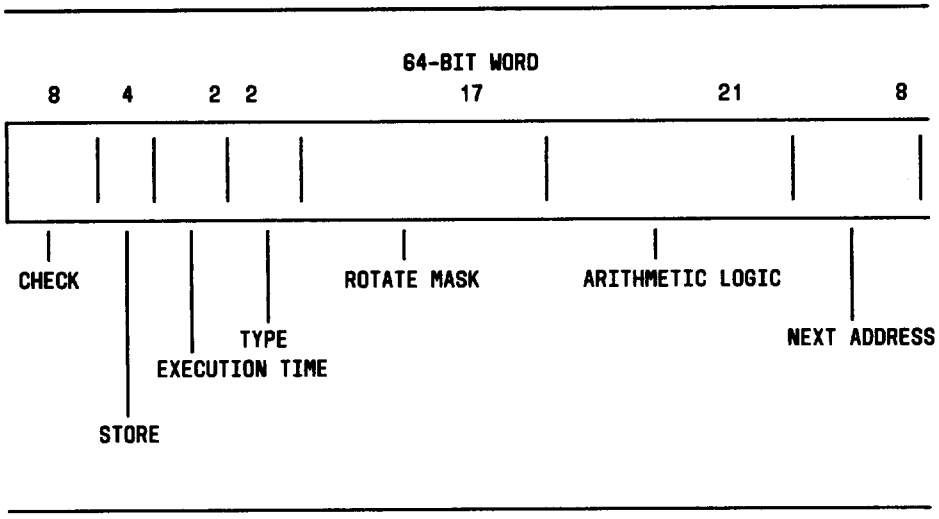



**Fig. 10—Microinstruction Word Format**

accessed by either the A-address or the B-address. These two operands may be accessed, processed by the arithmetic-logic section, then stored in the B-address location or the Q register by a single microinstruction. The eight RAMs in the 2901As form the set of general registers G00 through G15.

**4.48** The arithmetic-logic section can perform three binary arithmetic and five logic operations on its two 4-bit input operands. Additional functions can be performed by manipulating the carry-in bit. Control of the 2901A is accomplished by two 3-bit input fields and a 2-bit field. These are the ALU source, ALU function, and destination control field. Each of these fields is decoded by the 2901A to establish the gating paths to execute the desired function. Output of the 2901A is via a 2-input tri-state multiplexer. Table D depicts the available ALU functions.

**4.49** Two sets of flag bits are provided so that results of the operations can be tested. The sets are general and temporary, while the flags in each set are carry, overflow, sign, and all-zero condition. These flags are under microprogram control and any bit may be used to perform a condition test. The flags in the general set form the low four bits of the program status word (PSW), and the flags in the temporary set form the low four bits of the hardware status register (HSR).

**4.50** In general, the ALU receives data stored at a location addressed by the ALU A address as one operand, data at a second location addressed by the B address as a second operand, performing the specified function in the arithmetic section and gating the result in the B address location. The A and B addresses specify both external data locations and words in the 16-word RAM inside the ALU.

**4.51** The Q register in the ALU functions as an accumulator or temporary register. It is also used during multiplication and division operations. The eight Q registers in the 2901As form a 32-bit hardware register.

**4.52** The A and B addresses may come directly from the microinstruction in the MIR or from designated registers. Refer to Fig. 14 for a simplified logic diagram of the A and B addressing.

### Microprogram Unit

**4.53** The microprogram unit consists primarily of the following subunits:

(a) The microinstruction register (MIR) which contains the current microinstruction

(b) The instruction buffer (IB) which contains the current MAS instruction being executed

(c) The microaddress register (MAR) which contains the address of the current microinstruction

## TABLE C

## MICROINSTRUCTION DECODING

| MANIPULATE | |
|---|---|
| BIT | FUNCTION |
| 0-7 | Next Address |
| 8-11 | Destination |
| 12-15 | Register |
| 16-17 | Flag |
| 18-26 | Instruction |
| 27-28 | Carry |
| 29-32 | Mask Class |
| 33-37 | Rotate Amount |
| 38-39 | Rotate Mask Unit Operation |
| 40-43 | Source |
| 44-47 | Type |
| 48-49 | Time |
| 50-55 | Store |
| 56-63 | Check |

| JUMP | |
|---|---|
| BIT | FUNCTION |
| 0-7 | Next Address |
| 8-23 | Jump Address |
| 24-32 | — |
| 33-39 | Condition |
| 40-43 | — |
| 44-47 | Type |
| 48-49 | Time |
| 50-55 | Store |
| 56-63 | Check |

| FIND LOW ZERO | |
|---|---|
| 0-7 | Next Address |
| 8-11 | Destination |
| 12-15 | Register |
| 16-23 | Fill Data Byte 2 |
| 24-31 | Fill Data Byte 1 |
| 32-39 | Fill Data Byte 0 |
| 40-43 | Source |
| 44-47 | Type |
| 48-49 | Time |
| 50-55 | Store |
| 56-63 | Check |

| IMMEDIATE DATA | |
|---|---|
| 0-7 | Next Address |
| 8-39 | Data Word |
| 40-43 | Destination |
| 44-47 | Type |
| 48-49 | Time |
| 50-55 | Store |
| 56-63 | Check |

| CALL | |
|---|---|
| 0-7 | — |
| 8-23 | Call Address |
| 24-39 | Call Address |
| 40-43 | — |
| 44-47 | Type |
| 48-49 | Time |
| 50-55 | Store |
| 56-63 | Check |

| XMOVE | |
|---|---|
| 0-7 | Next Address |
| 8-11 | Destination |
| 12-15 | Register |
| 16-39 | — |
| 40-43 | Source |
| 44-47 | Type |
| 48-49 | Time |
| 50-55 | Store |
| 56-63 | Check |

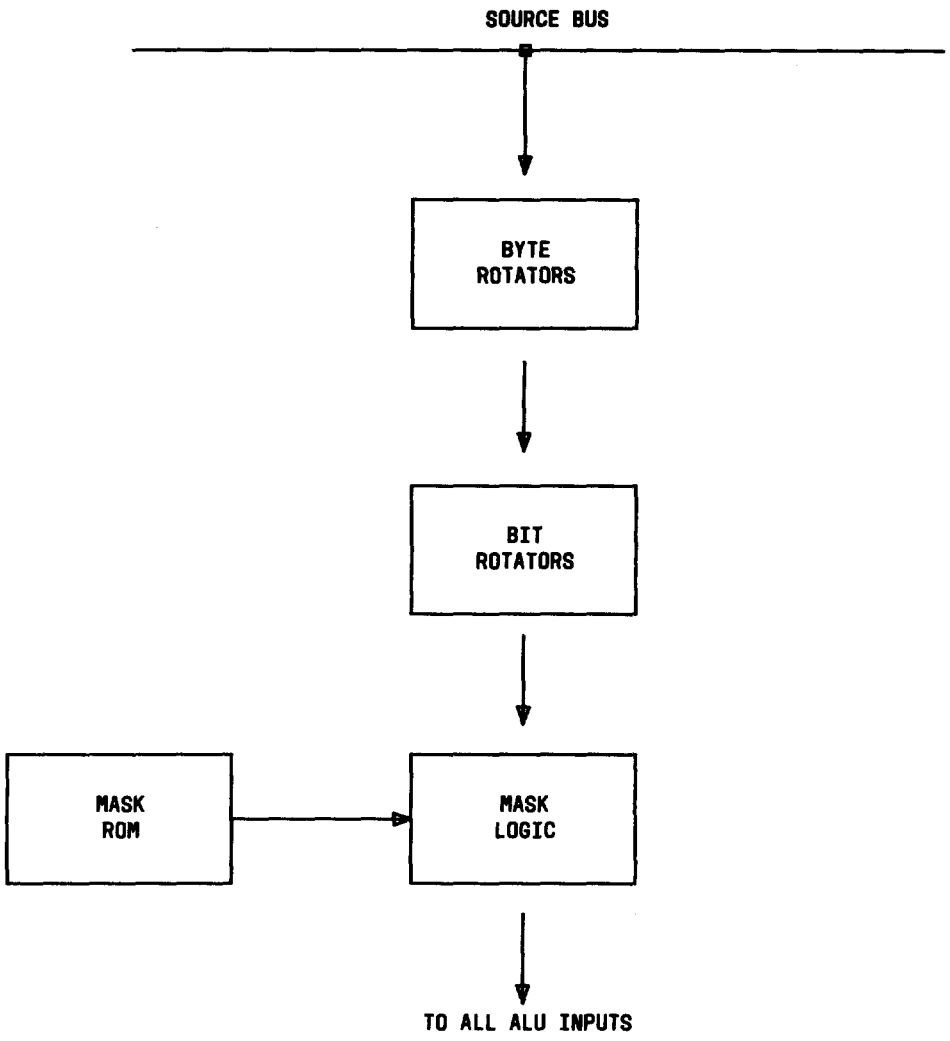



**Fig. 11—Rotate Mask Unit**

(d) The return address register (RAR) which contains the return address for a subroutine.

**4.54** The microprogram unit performs three functions:

- Microaddress sequencing
- Microinstruction decoding
- Timing generation.

**4.55** Each microinstruction contains the address of the next microinstruction (the NA field in the word format). The microinstruction address consists of two 8-bit bytes. The most significant byte is called the page number, while the least significant byte is called the line number. The NA field specifies the line number; the page number of the current microinstruction is taken to be the page number of the next microinstruction. This is practical since most sequences of microinstructions are very short compared to the number of microinstructions that can be stored on one page of microstore.

**4.56** Other address alternatives are required for different circumstances. An interrupt service request on a maintenance reset (MR) causes a jump to a predefined, fixed address. Jump addresses are needed when a decision is made to jump based on some tested value, when a jump to a new page in memory is necessary, or when a subroutine is called.

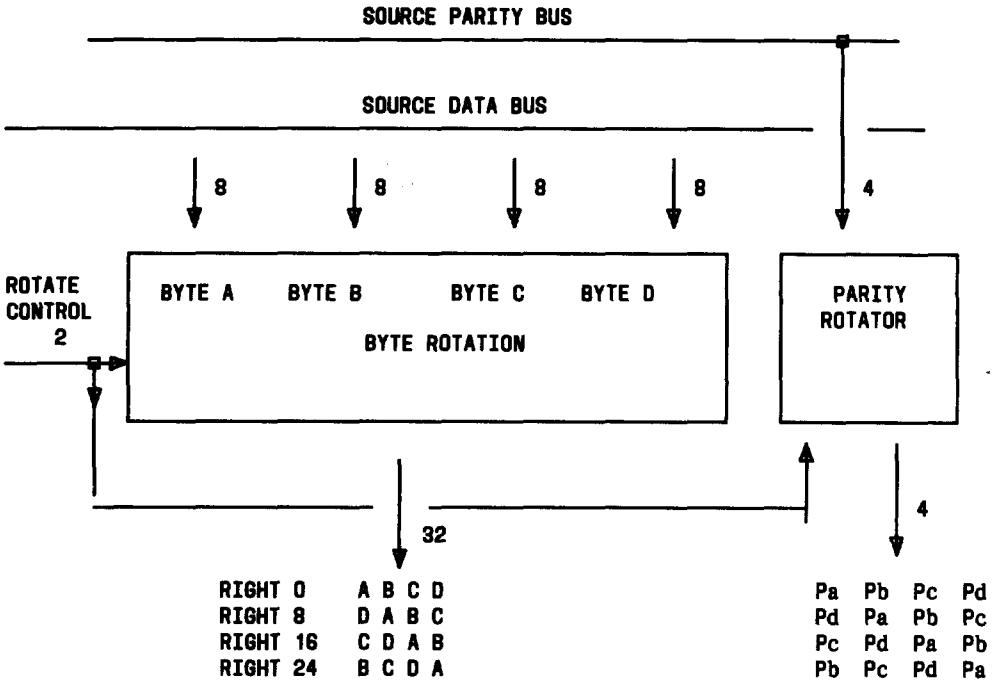



**Fig. 12—Byte Rotation**

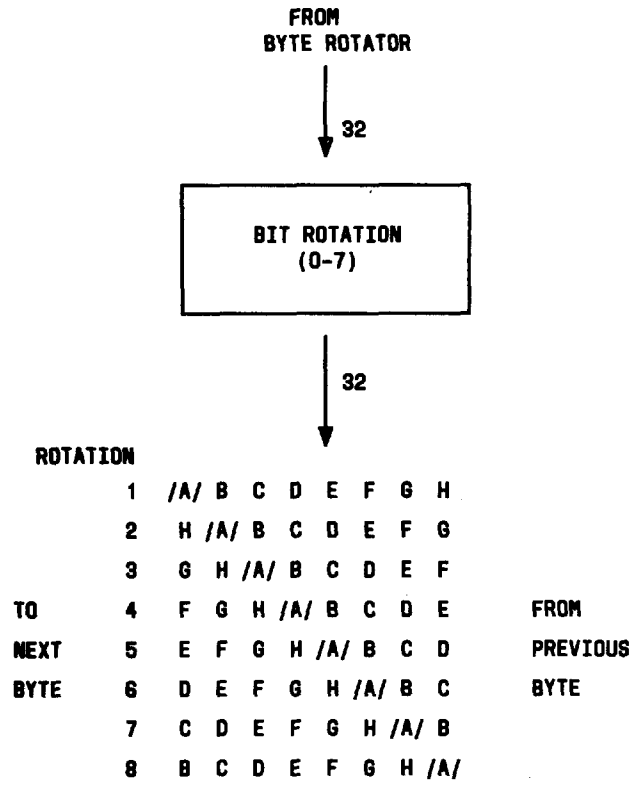



**Fig. 13—Bit Rotator**

In these cases, the new address comes from a field in the microinstruction that is normally used for other purposes.

**4.57** When execution of one instruction is completed and a new instruction is fetched from the MAS, the address of the first microinstruction is decoded from the Opcode of the instruction, together with certain bits of the program status word (PSW), and then forced into the MAR.

**4.58** Microaddress sequences consist of multiplexers which select the desired type of address and the 16 flip-flops which form the MAR. The output of the microaddress sequencer is through an array of tri-state drivers.

### Microinstruction Decoding

**4.59** Each microinstruction contains a number of mapped fields, ie, fields that are used for different purposes at different times. Control signals may come directly from the various fields of the microinstruction or they may come from other sources. Multiplexers are used to select the appropriate inputs. The decoder also generates signals to identify the type of microinstruction, which in turn specifies

**TABLE D**

**ARITHMETIC LOGIC UNIT FUNCTIONS**

| SOURCES | |
|---|---|
| **MNEMONIC (NOTE)** | **DEFINITION** |
| A | Data located at the "A" address |
| B | Data located at the "B" address |
| Q | Q register |
| Z | Zero |
| D | Data input (output of the rotate mask unit) |
| **ALU FUNCTIONS** | |
| **ARITHMETIC** | **LOGICAL** |
| ADD<br>Sub R<br>Sub S | OR, AND, NRAND (compliment R, logical AND S),<br>XOR (exclusive OR), XNOR (exclusive NOR) |
| **DESTINATIONS** | |
| **MNEMONIC** | **DEFINITION** |
| K | External register |
| B | Register defined by the "B" address field |
| Q | Q register |
| KB | Register defined by the "B" address field and an external register |

***Note:*** Source combinations allowed: AQ, AB, ZQ, ZB, ZA, DA, DQ, DZ.

the function that each of the mapped fields is to perform. Refer to Fig. 8 for a block diagram of the internal computer architecture. Table C lists the various fields decoded from a microinstruction.

**4.60** The purpose of decoding is to provide instructions and data to the ALU. The basic ALU functions are listed in Table D while the 32 ALU instructions are listed in Table E. In this Table E, A and B refer to general registers inside the ALU, while Q is the third ALU register. Figure 15 is a block diagram of the ALU.

### Timing

**4.61** The CC clock circuitry establishes the proper intervals for the following operations:

- Execution time
- Asynchronous wait
- Step pause.

**4.62** Four fixed values of execution times are provided: 150, 200, 250, and 300 nanoseconds (ns). Asynchronous wait provides waits in multiples of 50 ns when a source or destination register is unavailable and when a store access is not yet complete. Step pause provides multiples of 50-ns delays to accommodate single-step operation.

**4.63** Clock timing for a single microinstruction consists of eight timing intervals: T0, T1, T2, T3, T4, TS, TD and TP. T0 through T4 are each 50 ns long, while intervals TS, TD, and TP are multiples of 50 ns. T0 and T1 form the basic 100-ns timing interval. Intervals T2, T3, and T4 are added as necessary when intervals of 150, 200, and 250 ns are required.

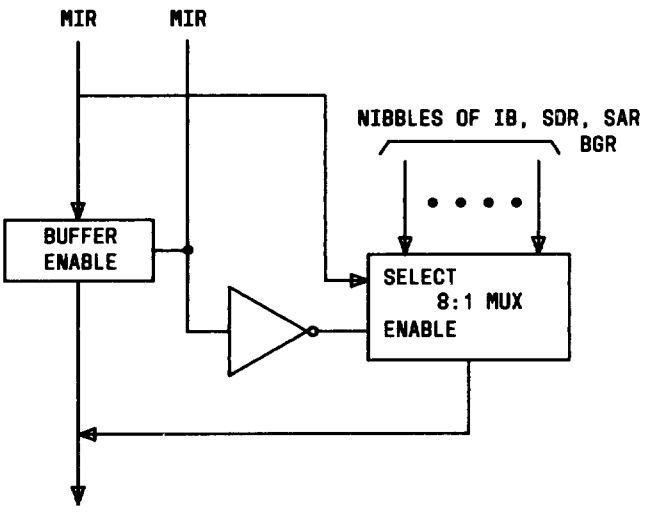



**Fig. 14—A and B Address Control**

Intervals TS and TD are added to provide an asynchronous wait, while TP adds the step pause interval. Figure 16 is a simplified logic diagram of the clock circuit.

### Cache Operation

**4.64** An optional cache store unit (CSU) provides a means of reducing access and cycle times of store operations. The CSU consists of three high-speed memory arrays, each containing 512 words, and has an access time of 200 ns. Also, the CSU provides a 2K-word buffer for use as an interrupt stack. Each word read from MAS is stored in the cache memory. Each time an MAS read is requested, part of the address is decoded to read the corresponding address in the cache memory. If the requested data word is in cache, it is returned from the CSU and the MAS read operation is inhibited. Thus, frequently used data words will usually be found in the cache memory.

### Input/Output Operation

**4.65** The channel data register (CDR) is the interface between the CCIO bus and the CC. Data to be outputted is loaded into the CDR from the destination (DST) bus, then is gated onto the CCIO bus from the CDR. Input data from the CCIO bus is gated to the CDR, then transferred to the source (SRC) bus. Refer to Fig. 17.

**4.66** The CCIO bus is used to transmit data, address, and control signals between the CC and the I/O processors. Each I/O channel is addressed by a 3-out-of-6 code on the address leads CIOAD00 through CIOAD50 (from the hardware status register [HSR]). When the addressed I/O device detects its code, it returns the same code to the CC on leads CIOAK00 through CIOAK50. The CC transmits the read data signal (CIORD0) from bit 0 of the pulse point register (PPR) onto the CCIO bus.

**4.67** Verification of the three address-acknowledge signals is performed in the source register (SREG) 1, but only to determine that the three signals have been received, not that the signals are correct. If the signals are not received, bit 17 in the ER is set. Refer to Section 254-301-100 (Input/Output Interfaces, Description and Theory of Operation, 3B 20D Computer) for further details concerning the I/O system.

### Memory Management

**4.68** Memory management in the CC is performed by the store address translator (SAT) which provides the interface between the store address register (SAR) and main store (MAS). The SAT accepts a logical address (referred to as "virtual") from the SAR and translates that to the actual physical address of a cache or MAS location. This permits blocks of data to be stored in any unused location in the MAS.

**4.69** The SAT functionally consists of two 512-word by 27-bit RAMs (called address translation buffers A and B [ATBA and ATBB]), a 10-bit counter, address, main store, and invalidation multiplexers, write control, compare and protection logic, and cache interface. Refer to Fig. 18 for a block diagram of the SAT and Fig. 19 for the address translation buffer (ATB) word format.

**TABLE E**

**BASIC ALU INSTRUCTIONS**

| INSTRUCTION | SOURCE R (NOTE) | SOURCE S (NOTE) | DESTINATION H (NOTE) |
|---|---|---|---|
| 000xxx000 | A | Q | Q |
| 001xxx000 | A | Q | (K) |
| 010xxx000 | A* | Q | B |
| 011xxx000 | A | Q | B |
| 000xxx001 | A | B | Q |
| 001xxx001 | A | B | (K) |
| 010xxx001 | A* | B | B |
| 011xxx001 | A | B | B |
| 000xxx010 | Q | Q | Q |
| 001xxx010 | D | Q | (K) |
| 010xxx010 | D* | Q | B |
| 011xxx010 | O | B | (K) |
| 010xxx011 | O* | B | B |
| 011xxx011 | O | B | B |
| 000xxx100 | O | A | Q |
| 011xxx100 | O | A | (K) |
| 010xxx100 | O* | A | B |
| 011xxx100 | O | A | B |
| 000xxx101 | D(j) | A | Q |
| 001xxx101 | D(j) | A | (K) |
| 010xxx101 | D(j)* | A | B |
| 011xxx101 | D(j) | A | B |
| 000xxx110 | D(j) | Q | Q |
| 001xxx110 | D(j) | Q | (K) |
| 010xxx110 | D(j)* | Q | B |
| 011xxx110 | D(j) | Q | B |
| 000xxx111 | D(j) | Q | Q |
| 001xxx111 | D(j) | Q | (K) |
| 010xxx111 | D(j)* | Q | B |
| 011xxx111 | D(j) | Q | B |

***Note:*** A and B are ALU internal registers and Q is the third ALU register. D(j) is an external special register.

* = ALU bypassed.

**4.70** The ATBs contain address translation information and each is divided into eight 64-word blocks or "task areas." Each entry in the ATBs is divided into four fields: tag field, protection field, relocation address field, and parity field.

(a) The tag field contains 7 address tag bits (bits 6 through 0). These address tag bits are compared to the SAR bits 16 through 14 and bits 23 through 20 to determine whether there is a "hit" (the required translation information is in the entry).

(b) The protection field consists of 4 bits (bits 10 through 7). Bits 10 through 8 are set if the requested memory page is readable, writable, or exe-

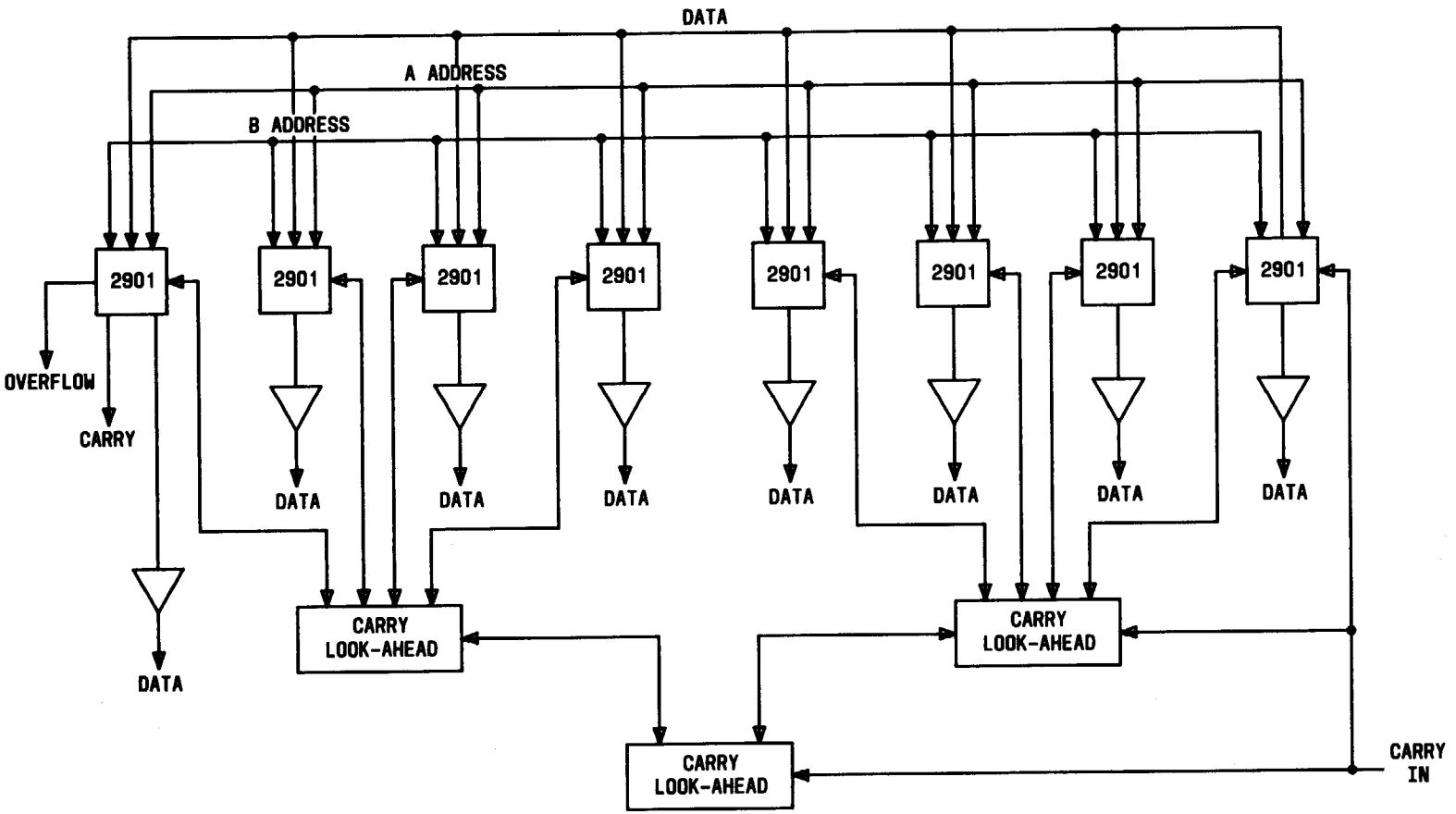



**Fig. 15—Arithmetic Logic Unit, Internal Architecture**

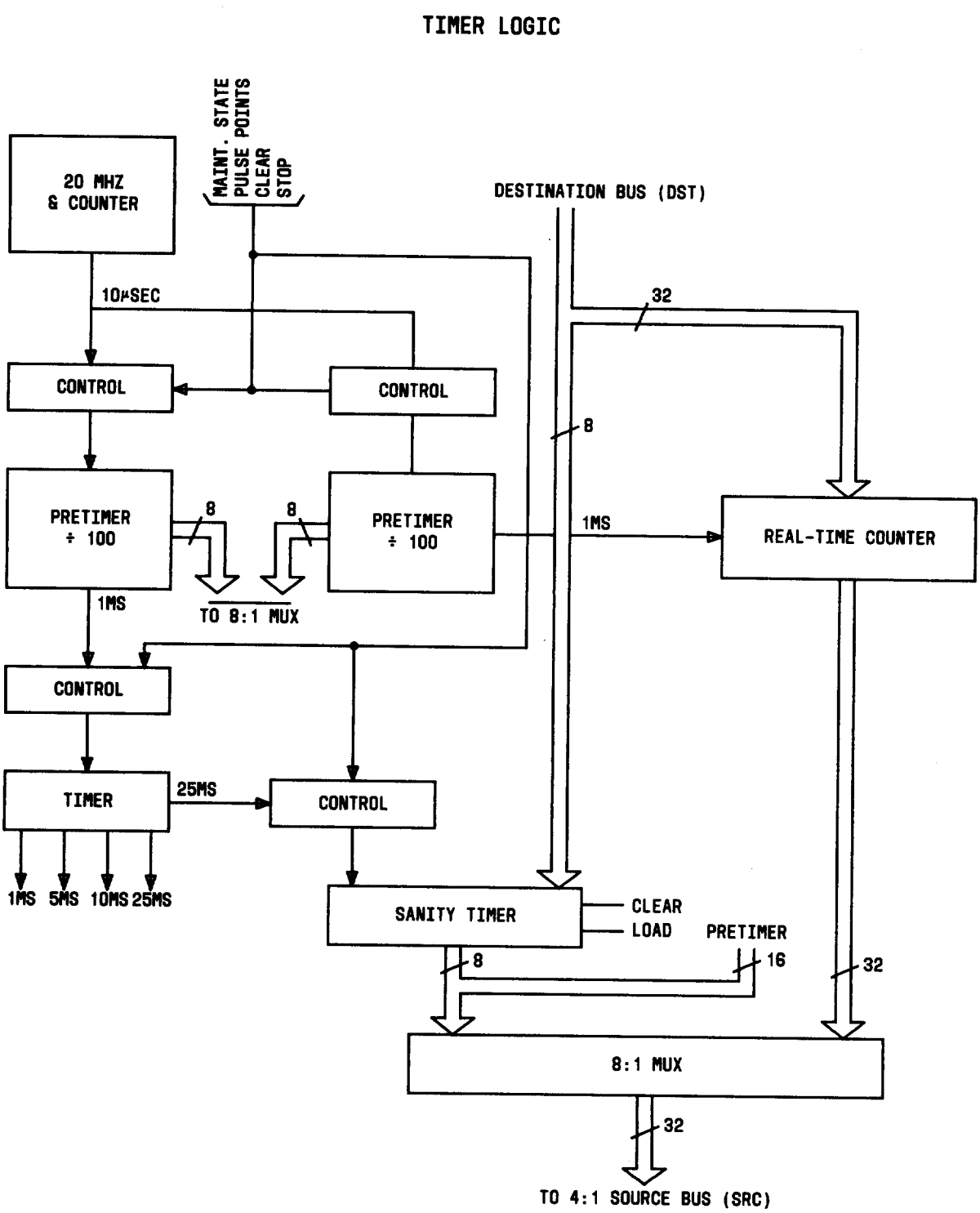



**Fig. 16—CC Clock Circuits**

cutable. Bit 7 is a valid bit that determines the validity of an entry.

(c) The relocation address field contains the 13 most significant bits (bits 23 through 11) of the

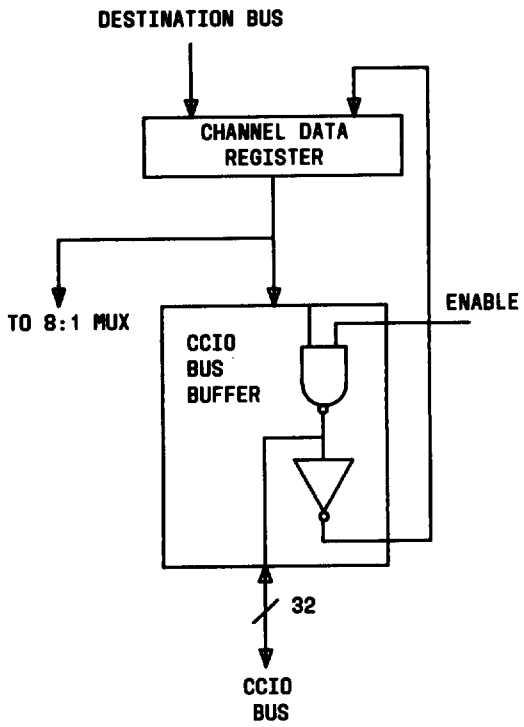



**Fig. 17—Channel Data Register Diagram**

physical address. These bits are sent to the cache and the MAS.

(d) The parity field contains 3 parity bits (bits 34 through 32). These bits correspond to the three bytes (24 bits) contained in the above three fields.

**4.71** The ATB may be bypassed if no address translation is required. In this case, SAR bits 23 through 11 are gated directly onto the relocation address output of the ATBA so that the virtual address and the physical MAS address are the same.

**4.72** The store control register (SCR) contains the SAT control flip-flops. Bits 0 through 10 and 27 through 29 are used to control the SAT. Bit 27 selects ATBA and ATBB while bit 28 determines whether the SAR or the counter furnishes the ATB address. Bit 29 isolates the ATBs from the DST bus and causes an all-zero word to be used as data. Bit 10 controls the counter formed by bits 0 through 9.

**4.73** Bits 7 through 9 are not incremented during the counter operation but are used to select the "task area" of the ATBs. Bit 9 is used as the most significant bit of the counter. Clearing bit 10 starts the counter and, when bit 21 is incremented to a 1, the counter stops. In addition, the counter may be loaded with a predefined value when bit 21 is a logic 1. Figure 20 is a block diagram of the counter.

**4.74** Store control register (SCR) bit 29 causes the invalidation multiplexer to gate either all zeros or the DST bus to the ATBs. This causes the ATBs to contain all zeros during an invalidate operation.

**4.75** The address multiplexer selects the source of the ATB address according to the state of SCR bit 28. When bit 28 is a logic 0, the address from the counter is selected and, when bit 19 is a logic 1, SAR bits 11 through 13 and bits 17 through 19 are used as an address.

**4.76** The write control logic sends a write enable signal to either ATBA or ATBB according to the states of the SCR bits 27 and 28. If the SCR bit 28 is set, bit 18 selects ATBA and ATBB to receive its address from the counter. When bit 28 is clear, the write enable signal is applied to whichever module is invalid. If both are valid, the write enable is applied to each alternately.

**4.77** The compare logic in each ATB module compares the tag field with bits 14 through 16 and bits 20 through 23 of the SAR. If there is a match, a "hit" signal is sent to the cache memory and to the SAC (UN43). If neither ATB matches the SAR, an ATB "miss" signal is generated which inhibits the Store Go signal, and causes a microinterrupt. When the SCR bit 20 is a 0, the ATB is bypassed which forces an ATB "hit."

**4.78** When an ATB "hit" occurs, protection checks are made to determine whether memory operation is correct. Any of the following errors inhibit Store Go, clear access enable (SCR bit 16), and cause a microinterrupt:

(a) Read being performed and ATB read bit not set

(b) Write being performed and ATB write bit not set

(c) Fetch being performed and ATB execute bit not set.

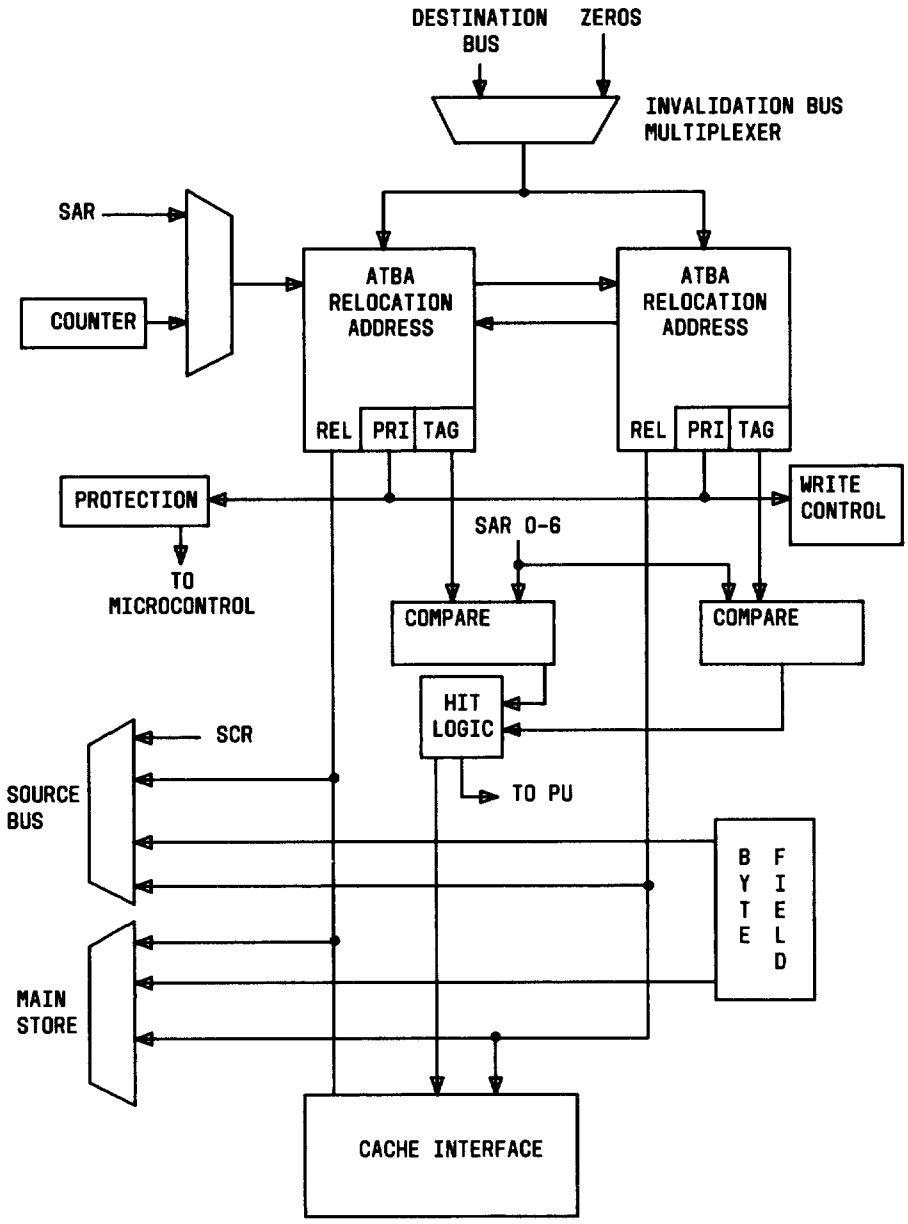



**Fig. 18—Store Address Translator Block Diagram**

**4.79** In order to form the physical address of a MAS location, bits 0 through 10 of the SAR are used as bits 0 through 10 of the MAS address. The relocation address, bits 11 through 23 of the ATB, are joined to bits 0 through 10, and all 24 bits are gated onto the MAS bus. The relocation address from the ATBs address any one of the 1024 pages, while the 11 least significant bits from the SAR address any of the 2K bytes in a memory page.

**4.80** When an ATB "miss" occurs (that is, when the SAR tag bits 14 through 16 and bits 20 through 23 do not match ATB bits 0 through 7), the ATB "miss" routine is initiated. This microcode routine uses tables of words in MAS to determine the relocation address. The microroutine first calculates which segment base register (SBR) contains the segment table address by using the type of instruction (read, write, or instruction fetch), PSW register bits

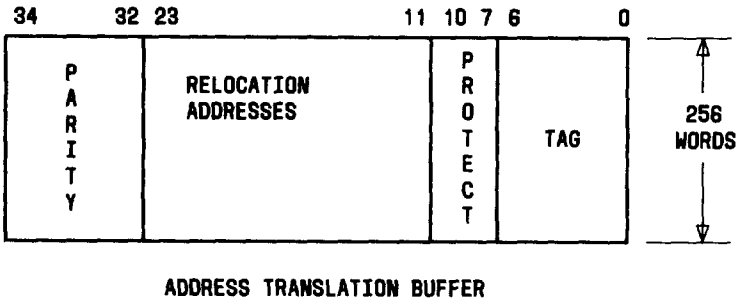



Fig. 19—ATB Word Format

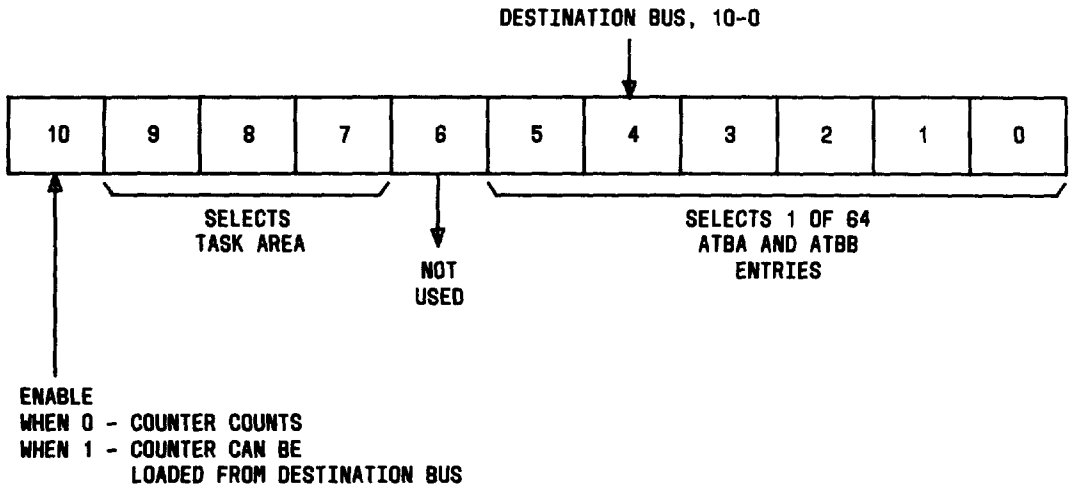



Fig. 20—Counter Layout

8 and 9, and bits 10 through 12 or 13 through 15. This address, plus SAR bits 17 through 23, yields the address of one of the entries in the segment table. Each segment table entry (STE) contains the address of a page table. The page table address, plus SAR bits 11 through 16, yields the address of one of the page table entries (PTE). This address contains the upper 13 bits of the relocation address which are combined with bits 0 through 10 of the SAR to produce the complete MAS physical address. This address is also used to update the ATB. The sequence of events is as follows:

(1) Instruction type + PSW 8, 9, + PSW 10 through 12 or 13 through 15 = SBR select

(2) SBR 0 through 21 + SAR 17 through 23 = STE address

(3) STE 0 through 21 + SAR 11 through 16 = PTE address

(4) PTE 11 through 23 + SAR 0 through 11 = relocation address

(5) Relocation address + SAR 0 through 10 = MAS physical address.

Figure 21 depicts the ATB "miss" addressing and Fig. 22 depicts the ATB sequencing.

**4.81** The cache interface circuitry transfers the contents of the relocation address field of the ATB, together with the least significant byte of the SAR, to the cache memory. The cache uses the ATB "hit" signals to determine which of the physical addresses is valid. In addition, parity is recalculated.

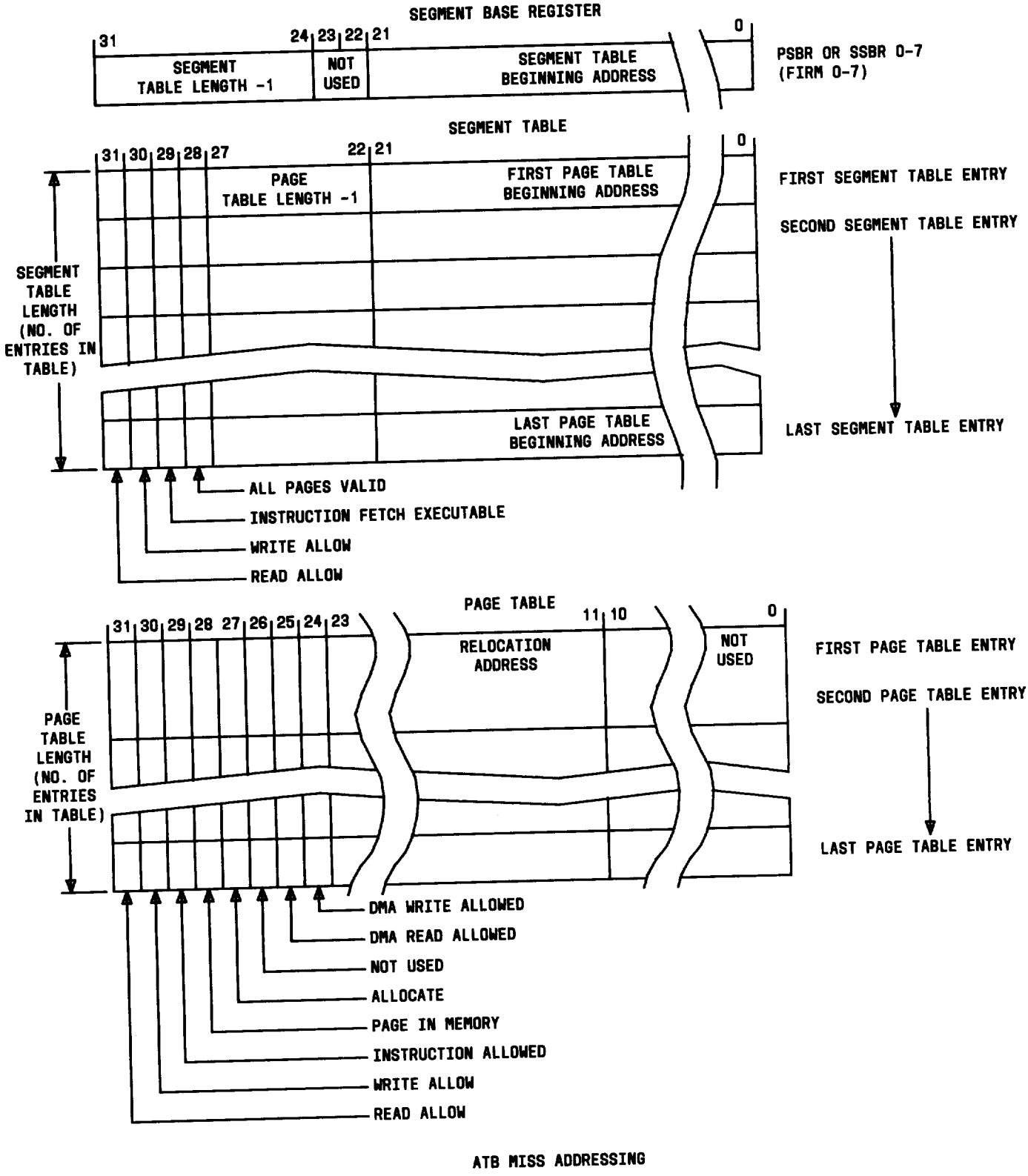



Fig. 21—ATB "Miss" Addressing Layout

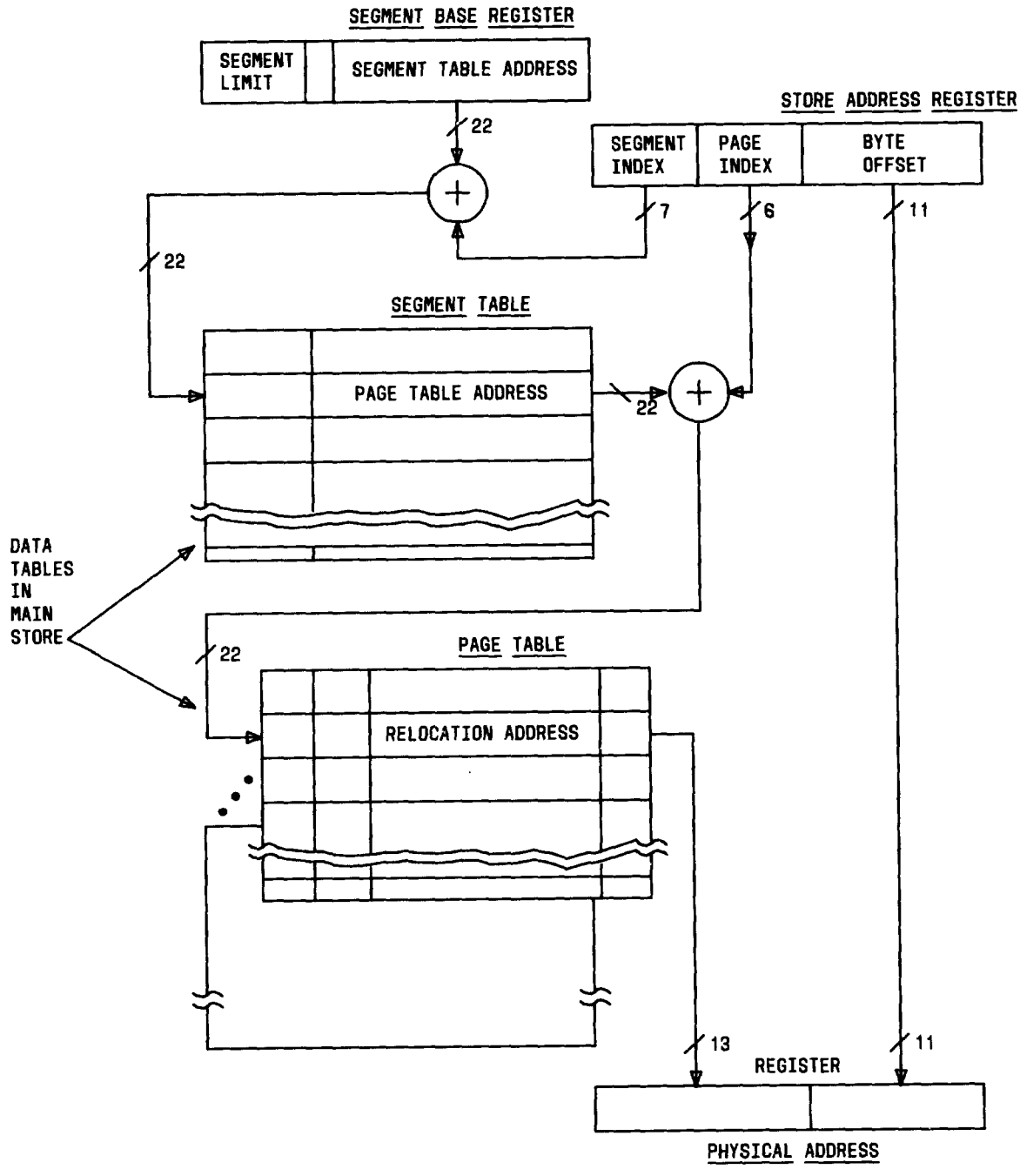



**Fig. 22—ATB Sequencing Layout**

**4.82** The MAS multiplexer determines which address from the cache interface logic will be transferred to the MAS. If ATBA has a "hit," the MAS receives the A address; if not, the B address is sent. If the ATB bypass mode is enabled, the A address is sent. The source multiplexer gates an ATB output onto the source (SRC) bus. Bit 40 in the MIR selects the SCR output or one of the ATB outputs selected by the SCR bit 18.

## 5. THEORY OF OPERATION

**5.01** The CC is comprised of the following units:

(a) Microstore (MIS)

(b) Central processing unit (CPU)

(c) Main store update unit (MASU)—DMAC/ duplex configuration

(d) Maintenance channel (MCH)—duplex configuration

(e) Cache store unit (CSU)—optional

(f) I/O channels—2 positions

(g) Microlevel test set (MLTS) interface—optional

(h) Utility circuit (UC)—optional.

### A. Microstore

**5.02** The microstore (MIS) unit is comprised of the following:

(a) Writing microstore (WMS, ▶UN48◀)

(b) Read-only microstore (MIS, UN28 ▶or UN28B effective with Generic 2◀).

#### Writable Microstore

**5.03** The WMS is a 1K ▶or 4K effective with Generic 2◀ writable microinstruction store circuit. The microinstruction is a 64-bit word used by the CU and stored in the RAM in the WMS. Microinstructions (microsequences) may be read over the BGB or the MSD bus. However, data can only be written via the BGB. The hardware status register (HSR) and pulse point register (PPR) provide the control signals for reading and writing via the BGB. Microinstructions are provided to the CU via the MSD bus. The instructions address (in the WMS) is provided from the CU via the MSA bus. The WMS is writable (loaded) via the BGB, and the instructions are accessed (read) using the MSD bus.

**5.04** Microinstructions used by the CU are accessed by the address input data (location to be read) received via the MSA bus. A valid address on the MSA bus will result in the output of the 64-bit microinstruction onto the MSD bus or onto the BGB if the BGB is not being used. This is indicated by the BGB enabled lead (BGBEN0IO) being in a high state (inactive is the low state which means that the BGB is active).

**5.05** The microstore address is an 18-bit address received from the MSA bus. The least significant 10 bits (0 through 9) contain the physical address of the RAM. Bits 10 and 11 signify an error condition when they are in a high state. ▶Effective with Generic 2, the least significant 12 bits (0 through 11) contain the physical address of the RAM.◀ Bits 12 and 13 are used to select the WMS. Bits 14 and 15 are not used except to signify an error condition if in the high state (normally, these bits are always in the low state). Bits 16 and 17 are parity bits for the address data but are not used by the WMS.

**5.06** With a valid MSA bus input (address), the input enable signals of the MSA bus tri-state buffers are active and gate the output of the addressed data in the RAM. This address does not have to be a microstore address, since the BGB is still at the addressed RAM if its enable lead (BGBEN0IO) is low. Bits 0 through 9 ▶or bits 0 through 11 effective with Generic 2◀ of the input microstore address are gated to the RAM only when the latch enable flip-flop signal (LEN1) is set to the low state. The latch enable flip-flop signal is set to the low state when the BGB enable input signal (BGBEN0IO) is not active (set to high state). A microstore address cannot be read when the BGB is being used (BGBEN0IO).

**5.07** The signal MSDV00 is the output indicating if the microstore data on the MSD bus is valid (high indicates invalid data and low indicates valid data). The valid microstore data is gated onto the MSD bus when the WMS is not being read or written via the BGB, and a valid microstore address is present.

**5.08** The WMS BGB inputs are mainly used for writing data into the WMS memory (RAM). The BGB is a 36-bit used for addressing, writing, and reading the WMS. Using the hardware status register (HSR), the pulse point register (PPR), and the bidirectional gating register (BGR) data, the address is gated to the WMS, and 32 bits of data can then be written into or read from the WMS.

**5.09** Enable signal BGBEN0IO controls the enabling of the write and read functions via the BGB. The BGB signal BGBRWIO sets the read/write mode of the BGB. A low input enables the write mode, and a high input enables the read mode. However, when providing an address via the BGB, the read/write input is set to the high state (read mode). The PPR signal (BGBPP0IO) on the BGB latches the address in the WMS to the BGB address latch, and pulse points BGBPP1IO and BGBPP2IO control writing the low and high data bits. When BGBPP1IO is gated low, it controls the low 32 bits of the microinstruction. The pulse point either reads or writes data from or to the WMS or RAM. When BGBPP2IO is gated low, it controls the high 32 bits of the microinstruction. The pulse point either reads or writes data from or to the WMS RAM.

**5.10** The BGB address is a 16-bit address that is gated from the BGR. The low 10 bits (0 through 9 for bits 0 through 11 effective with Generic 2) define the physical address to be read. These bits are fanned out to the inputs of the RAM. When the address is correct, the acknowledgment signal (BGBAM0) is gated to the low state.

**5.11** The BGB enable signal (BGBEN0IO) is set low to enable the read/write control logic. When BGBEN0IO is set low, this enables the latch enable flip-flop (LEN). The latch enable flip-flop will be set (low) when the correct address is present and the pulse point is set low. The latch enable flip-flop is set when signal LEN1 is in a high state. This enables the read enable (REN1) and write enable (WEN1) circuit gates. The latch enable (LEN1) drives the Store Go (BBGBATSGO) signal to the low state, which controls the output of the address latch (BGBAL). The address is available to the WMS RAM when the pulse point zero (BGBPP0IO) is gated low, resulting in an acknowledgment signal (BGBALCKP) being clocked and then latching the address into the BGB address latch.

**5.12** To enable the write logic, the BGB signal BGBRWIO is set high and ANDed together with the latch enable signal (LEN1). This ANDed signal (WEN1) enables the write logic. Write enable low (WEL1) and write enable high (WEH1) for the low and high data bits write pulses are needed to write the microinstruction into the WMS RAM because the BGB is a 36-bit bus and the microinstruction is 64 bits. After the address is sent to the WMS RAM, the low 32 bits of the microinstruction are gated onto the BGB, and the pulse point one signal (BGBPP1IO) is gated low. The resulting signal (WELPO) from the ANDed inputs BGBPP1IO, WEN1, and WEL1 provides for writing the low 32 bits of the data into the WMS RAM. This procedure is also used to write the high 32 bits of the microinstruction into the WMS RAM. The high 32 bits are gated onto the BGB and the pulse point two signal (BGBPP2IO) is gated low. The resulting signal (WEHPO) from the ANDed inputs BGBPP2IO, WEN1, and WEH1 provides for writing the high 32 bits of data into the WMS RAM.

**5.13** Reading the WMS from the BGB is essentially the same as writing the WMS from the BGB. The BGB enable signal (BGBWIO) is set to the high state (read mode). The WMS RAM address is gated from the bus and latched into the WMS. The BGB enable input signal (BGBEN0IO) is set low to enable the latch enable flip-flop (LEN) to the high state, and latches the address data into the BGB address latch (BGBAL). To read the WMS RAM data, the BGB enable signal (BGBWIO) is set to the low state. This input along with the latch enable signal (LEN1) results in the read enable gate (REN1) being set to the active state. Pulse point signals one (BGBPP1IO) and two (BGBPP2IO) are gated to the low state to read the low and high 32 bits of the microinstructions, respectively. Both pulse point signals one (BGBPP1IO) and two (BGBPP2IO) are ORed together at the LH1 gate circuit. The output of LH1 gate circuit is ANDed with the read enable (REN1) signal to generate the Store Go signal (BGBOTSGO). When the Store Go signal is low, the WMS RAM gates the data to be read onto the BGB. The WSM RAM outputs are connected to the BGB via a bit select register. Microstore data bus select signal (MSDSEL1) controls the selection of the 32-bit low and high data bits to be gated onto the bus (low state gates low bits and high state gates high bits), when pulse point two signal (BGBPP2IO) is set to the low state.

**5.14** Since the BGB is a 36-bit bus, data comprises 32 bits and parity comprises 4 bits (most significant bits 32 through 35). Parity is also included in the microinstruction by bits 56 through 63, which maintains odd parity over the microinstruction. Parity for the BGB is generated by an odd parity bit generator and a data selector. The odd parity generator inputs are the WMS RAM bits 56 through 63, which are microinstruction parity bits, and outputs an odd parity bit when the parity is designated as requiring an additional bit. This generates parity for the parity bits stored in the WMS RAM. This parity bit with the

microinstruction parity bits (56 through 62) are input signals to the data selector.

**5.15** The Store Go signal (BGBOTSGO) and microstore data bus select signal (MSDSEL1) control the data selector. The Store Go signal (BGBOTSGO) is in the low state when in the read mode and either pulse point signal one (BGBPP1IO) or two (BGBPP2IO) is gated low. When the Store Go signal (BGBOTSGO) is low, the parity bits are gated out onto the bus. The microstore data bus select signal (MSDSEL1) selects the data to be gated out, a low state gates the low data parity bits, and a high state gates the high data parity bits when pulse point two signal (BGBPP2IO) is gated low.

**Read-Only Microstore**

**5.16** The CC may have a maximum of three MIS (UN28 ◗or UN28B effective with Generic 2◖) circuit packs. The 64-bit microinstruction is stored in the programmable read only memory (PROM) on the MIS. The MIS (one ◗UN28/UN28B◖) has a maximum capacity of 4096 (1K by 4 PROMs) microinstructions. The MIS can be partially equipped in 1K increments (1K, 2K, 3K, or 4K). A selector switch is provided to manually set the memory range for a partially equipped MIS. The CC addresses the microinstruction in the MIS via the MSA bus and reads data from the MIS via the MSD bus.

**5.17** Reading the microinstruction from the MIS consists of providing a valid 18-bit address. Bits 0 through 9 define the physical address of the memory in the MIS. These bits are used to address 1024 locations in each of the four memory ranges. Bits 10 and 11 are decoded and used as inputs to the selector switch. The selector switch will enable the memory for a valid address and generate an error signal for an invalid address. Bits 12 and 13 are used to select one of the MIS circuit packs, each with a unique address. Bits 14 and 15 are not used; however, if either is at a high state, an error signal will be generated indicating an invalid address. This error signal (MSDV00), when high, indicates an invalid address and, when low, indicates a valid address. Bits 16 and 17 are the address parity bits and are not used on the MIS.

**5.18** The microstore data valid output signal (MSDV00) is an acknowledgment that a valid address was received and the output data was valid. Bits 10 and 11 are address signals used to select the memory range (determined by the selector switch setting). These bits are decoded by a 2-to-4 line decoder, and selects one of the four memory ranges (1K, 2K, 3K, and 4K). The output signal (CSA0) is wired directly to the enable input of the PROMs so that a minimum of 1024 locations can always be accessed. The remaining three decoder signals (CSK 1, CSK 2, and CSK 3) are input signals to the selector switch (eight selector switches are provided, but only six are connected). Each of these signals is wired to two selector switches. Depending on how the selector switches are set, the outputs are connected to the input of the PROMs or used as part of the microstore data valid output function (MSDV00). See the data in Table F for the selector switch settings for the memory ranges.

**5.19** The MIS address is the physical address (physical position) on the circuit pack. The MIS address is gated to the WMS and verified. If the address is incorrect, an error bit is set and gated to the error register. The address match signal, the maintenance channel data release signal, and the MIS address are ORed and the resulting two signals are used as the MIS data bus signals. The MIS data bus signals drive the MIS data output buffers. If the MIS address is in an out-of-range signal (high state), the MSA outputs will be gated off. The maintenance channel data release input signal inhibits the MSD bus output signals whenever the MCH requires the MSD bus.

**B. Central Processing Unit**

**5.20** The central processing unit (CPU) is comprised of the following units:

(a) Microcontrol (MC)—UN44 ◗or UN135 effective with Generic 2◖

(b) Store address translator (SAT)—UN45

(c) Store address control (SAC)—UN43

(d) Store data control (SDC)—UN06

(e) Data manipulation units (DMUs)—UN01 and UN23

(f) Special registers—UN02 and UN03.

TABLE F

SELECTOR SWITCH SETTING

| MEMORY ADDRESS | MEMORY RANGE | SWITCHES (NOTE) | | | | | |
|---|---|---|---|---|---|---|---|
| | | SW1 | SW2 | SW3 | SW4 | SW5 | SW6 |
| 0—3FF | 1K | 1 | 0 | 1 | 0 | 1 | 0 |
| 0—7FF | 2K | 1 | 0 | 1 | 0 | 0 | 1 |
| 0—BFF | 3K | 1 | 0 | 0 | 1 | 0 | 1 |
| 0—FFF | 4K | 0 | 1 | 0 | 1 | 0 | 1 |

*Note:* 1 = open switch and 0 = closed switch.

**Microcontrol**

**5.21** The microcontrol (MC) UN44 or UN135 effective with Generic 2 in the CC processes instructions and addresses microsequences. When the MC is ready to process a new instruction located in the store instruction register (SIR), the MC gates the SIR and the SIR transmits the instruction to the halfword multiplexer (HM). The HM then gates the instruction to the instruction buffer (IB).

**5.22** The MC contains the following circuitry:

(a) Microstore (MIS) address sequence

(b) Microinstruction register (MIR)

(c) Source decoders

(d) Destination decoders.

**5.23** The microsequence uses the information stored in the IB to define what registers or data are to be worked on.

**5.24** The address sequence logic in the MC develops the MIS address by using the Opcode bits in the HM (bits 31 through 24) and the program status word (PSW) (bits 17 and 16). The address sequencer than develops an 18-bit MIS address (2 parity bits and 16 information bits). These bits are gated to the MIS via the microstore address (MSA) bus. The MIS decodes the MSA bus and reads out the first microinstruction of the microsequence. The 64-bit microinstruction is gated out of the MIS via the microstore data (MSD) bus. The MSD bus then gates the microinstruction to the microinstruction register (MIR).

**5.25** The MIR decodes the microinstruction it receives from the MIS. The following is a list of microinstructions decoded by the MIR.

- Manipulate
- Xmove
- Immediate data
- Find low zero
- Jump
- Call.

**5.26** If the decoded microinstruction is to manipulate data, it is gated to the DMU for further processing. The remaining decoded microinstructions are gated to the source decoder or the destination decoder.

**5.27** The source decoder selects which register is to be gated to the source (SRC) bus with a decoded microinstruction. These registers are the instruction buffer (IB), halfword multiplexer (HM), store data register (SDR), and the store instruction register (SIR).

**5.28** Decoded microinstructions destined to the destination decoder are gated to the destination (DST) bus via registers.

**Store Address Translator**

**5.29** The store address translator (SAT, UN45) provides the following functions:

(a) Translation of the 24-bit virtual address generated by the store address control (SAC) to a 24-bit physical address for the main store/cache

(b) Protection from incorrect and/or unauthorized access

(c) Storage of parity for the special registers.

**5.30** The 24-bit virtual address is divided into a 7-bit segment field, a 6-bit page field, and an 11-bit offset byte. Each operation can have a maximum of 128 segments, with each segment having a maximum of 64 pages. A page is defined to be 2K consecutive bytes starting on a 2K byte boundary. Therefore, the SAT protection is on a page basis (the 11-bit offset is not used for translation or page protection). The translation and protection data for every page is maintained in the segment and page tables by the CU. The tables for all active operations are located in the main store (MAS). The translation and protection data for the most recently used pages (for eight operations) are stored in the address translation buffer (ATB). Whenever the CU attempts to access the MAS, the virtual address is loaded into the store address register (SAR). (The SAR is part of the SAC.)

**5.31** If the memory management is enabled, the 13 most significant bits of the SAR (7-bit segment field and 6-bit page field) are used to access the ATB and derives the physical address. If the translation and protection data is present in the ATB, then the 13-bit physical page number is gated to the cache/main store. At the same time, the 11-bit offset data is also gated to the cache (if equipped) by the SAC. However, if the translation and protection data is not present in the ATB, a microinterrupt is generated which initiates a microroutine that fetches the required data from the MAS and loads the data into the ATB. Also, if the page is protected, the access is inhibited and an error bit is activated. This results in the generation of a protection violation microinterrupt. If bit 7 of the program status word (PSW) or bit 20 of the store control register (SCR) is 0, the memory management is disabled, and the virtual address is gated to the cache as a physical address. Also, the protection check operations are inhibited.

**5.32** The SAT consists of the following circuits:

(a) ATB accessing logic

(b) Address translation buffer (ATB)

(c) Cache address driver logic

(d) MAS address multiplexer logic

(e) Hit detection logic

(f) Protection check logic

(g) ATB data input multiplexer logic

(h) ATB write logic

(i) Interrupt stack logic

(j) Source (SRC) bus multiplexer logic

(k) Self-checking match logic

(l) Parity checker logic

(m) Hardware error detection logic

(n) Special register parity logic.

**5.33** When the CC initiates a store access, the virtual address is loaded in the SAR with the control data (read, write, etc) being loaded into the SCR. The outputs of the SAR and SCR are used by the ATB logic to generate an address to access the ATB. The relocation addresses generated by the ATB are gated to the cache (if equipped) by the cache address driver logic. Simultaneously, the "hit" logic and protection logic circuits determine whether the relocation addresses generated by the ATB are valid. If the ATB contains the required relocation address and protection data for the access, a "hit" is activated by the "hit" logic. If the ATB entry is invalid for any reason, a "miss" is activated by the "hit" logic. The "miss" data is transmitted to the SAC and the cache. The SAC then initiates the ATB "miss" microroutine. When a "hit" is activated, the protection logic checks, using the protection data from the ATB, and deter-

mines if the page access is allowed. If not allowed access to the page, an error bit is set. When a "hit" is activated and protection logic check is correct, the physical address is gated to the cache. However, if the cache access generates a cache "miss," the physical address is gated via the store address multiplexer to the MAS.

### *ATB Accessing Logic*

**5.34** The basic function of the ATB accessing logic is to generate the address to access the ATB by using bits from the SCR, SAR, and the PSW. The ATB accessing logic unit has the capability of accessing up to a maximum of 1024 entries of the ATB. The ATB has 512 entries. Bit 6 of the ATB address is unused.

**5.35** The ATB addressing logic is duplicated to drive the 28 memory devices that make up the ATB and for better error detection.

**5.36** The ATB address consists of two selections:

(a) ***Block Select:*** The 512-word ATB is divided into 8 blocks of 64 entries each.

(b) ***Entry Select:*** Used to select one of the 64 entries in the selected block.

### *Address Translation Buffer (ATB)*

**5.37** The ATB consists of two memories, ATBA and ATBB, each of which has 512 entries. The width of an ATB entry is 24 bits (3 bytes). Each byte has one parity bit. Figure 19 shows the format of an entry. The function of each field in the entry is shown in paragraph 4.70.

### *Cache Address Driver Logic*

**5.38** The function of the cache address driver logic is to send the physical address to the cache memory unit. The two most significant bytes of the physical address is generated in the SAT while the least significant byte is sent directly to the cache by the SAC. To improve performance (by overlapping "hit" calculations in the ATB and cache), the 13-bit reallocation address from both ATBA and ATBB are buffered by inverters and sent to the cache.

**5.39** When the ATB is bypassed, ATBA is disabled and SAR bits 23 through 11 are routed to the cache and MAS.

### *MAS Address Multiplexer Logic*

**5.40** The function of the MAS address multiplexer and data logic is to route the reallocation address and a subset of access control information to the MAS over the store address (SA) bus.

**5.41** When there is a "hit" in the ATBA matchers, the reallocation address field of the ATBA is selected and sent over the SA bus. If there is a "hit" in the ATBB matchers, the reallocation address field of the ATBB is routed over the SA bus. The SA bus is controlled by the store update.

### *Hit Detection Logic*

**5.42** The primary function of the "hit" detection logic is to determine whether the reallocation address and protection information for the page that is being accessed is available within the ATB. The circuit is duplicated for self-checking.

**5.43** When the ATB is in an unbypassed mode, bits 7 of the ATBA and ATBB are checked for validity.

**5.44** When the ATB is unbypassed and when anyone of the ATB entries is valid, then SAR bits 23 through 20 and 16 through 14 are compared against bits 6 through 0 of the ATB.

**5.45** When ATBA bit 7 is a 1 and if bits 23 through 20 and 16 through 14 of the SAR (bits 23 through 20 and 16 through 14 of the SAR are called SARTAG) matches with ATBA bits 6 through 0, a "hit" is indicated in ATBA and the relocation address is available in bits 23 through 11 of the ATBA. Similarly, when ATBB bit 7 is a 1 and if the SARTAG matches with bits 6 through 0 of the ATBB, a "hit" is indicated in ATBB and the required relocation address is available in bits 23 through 11 of the ATBB.

**5.46** When there is a "hit" in the ATBA or when an ATB is bypassed, the cache uses the relocation address derived from the ATBA. Similarly, if there is a "hit" in the ATBB, the cache uses the relocation address derived from the ATBB.

### *Protection Check Logic*

**5.47** The function of the protection logic is to check for any access violation like writing into a write-protected page. This circuit is duplicated for self-checking. The protection checks are inhibited when the ATB is bypassed or when there is a "miss" in the ATB.

### *ATB Data Input Multiplexer Logic*

**5.48** This circuit provides the data that is required during an ATB write operation. The data for the write operation can be supplied from the DST or loaded with all 0s. Loading from the DST is required for diagnostics and inputting translation and protection data. The ATB is loaded with the three least significant bytes (bits 23 through 0) from the DST, and the CU implements the ATB as a destination register. When the ATB is to be invalidated during a switch or start-up operation, the CU loads all 0s in the ATB to establish a known starting point.

**5.49** During a normal store operation, the ATB inhibit signals are low, and the ATB is loaded from the DST during an ATB "miss" operation. When the ATB is being invalidated, the inhibit signals are high, and 0s are loaded into the ATB.

### *ATB Write Logic*

**5.50** This circuit functions to control the writing of data generated by the ATB data input multiplexer logic circuit into the ATB. When the ATB is being invalidated (loaded with 0s), the write-enable signals of the ATB memories are active low while bits 5 through 0 of the counter in the store control register (SCR) are allowed to go through all 64 possible states.

**5.51** When the inhibit signal is high and the control store signal is low, the counter in the SCR is used to access the ATB. Then the ATBB store signal is implemented if the data is to be loaded in the ATBA (high) or ATBB (low).

**5.52** During a normal store operation, the store address register (SAR) and program status word (PSW) are used to access the ATB. This enables the SAT to have complete control of where the data is to be written. When the entries in ATBA and ATBB are both invalid, the new data is written into ATBA. However, if only one entry is invalid, the new data replaces the invalid data. Also, if both ATBA and ATBB entries are valid, the new data is written into either ATBA or ATBB depending upon the state of control (high is ATBB and low is ATBA). A comparator-type circuit is used to prevent a race condition between the ATBA and ATBB.

### *Interrupt Stack Logic*

**5.53** This circuit enables the 8K-byte interrupt stack in the cache memory unit. It also provides a self-checking function to ensure correct operations. The following conditions must be satisfied before the interrupt stack logic is enabled:

(a) Cache is present

(b) Cache is not bypassed

(c) Interrupt stack enable bit in the PSW is active

(d) Kernel address is accessed

(e) Virtual address is segment 4 and pages 0 through 3.

**5.54** When the interrupt stack logic is enabled, the ATB is bypassed. In this state the ATB is set in the off state, and protection checks and parity checks of the ATB are inhibited.

### *Source (SRC) Bus Multiplexer Logic*

**5.55** This circuit functions to route the output of the ATB and other miscellaneous data via the SRC bus to the CU. The buffered outputs of ATBA and ATBB are gated into the A and B inputs of the source multiplexer and outputted onto the SRC bus to the designated circuit.

### *Self-Checking Match Logic*

**5.56** The function of the self-checking match logic is to check the various duplicated circuits in the store address translator (SAT).

**5.57** Each of the duplicated circuits in the SAT contains an "x" and "y" portion. During a normal store access, the "x" and "y" portions of each circuit are compared against each other. A mismatch in any one of the circuits indicates a hardware fault.

**5.58** The self-checking match logic is checked periodically by system diagnostics.

### *Parity Checker Logic*

**5.59** The parity checker logic checks the parity of bits in the SAR, PSW, and the ATB used in the SAT.

**5.60** The two most significant bytes of the SAR (virtual address) are checked for parity. Also, the two least significant bytes of the ATBA and ATBB are checked for parity. Parity over the least significant byte of the SAR and most significant bytes of the ATBA and ATBB are checked at the cache memory unit and the MAS. The output of the parity checkers is an active low. Parity is also checked in the PSW bits that control the address translation process.

### *Hardware Error Detection Logic*

**5.61** This logic transmits error signals to the SAC in the event of a detected hardware fault in the SAT circuit pack.

**5.62** The following is a list of faults that could be detected:

(a) Parity error in the SAR, PSW, and ATB

(b) A mismatch in the circuits that are duplicated

(c) Multiple "hits" from the ATB (a "hit" generated by both ATBA and ATBB during an access).

### *Special Register Parity Logic*

**5.63** The purpose of this logic is to provide the parity bits for registers that are resident in the special register circuit pack.

#### Store Address Control

**5.64** The store address control (SAC, UN43) located in the CC contains the following circuitry:

(a) Present address register (PA)

(b) Store address register (SAR)

(c) Increment logic

(d) Store control register (SCR).

### *Present Address Register (PA)*

**5.65** The PA register is 27 bits in width including three parity bits, one for each byte. The PA is the program counter. A new address is clocked into the PA each time an instruction is fetched into the store instruction register (SIR).

### *Store Address Register (SAR)*

**5.66** The SAR is also 27 bits wide including three parity bits, one for each byte. The SAR has the same inputs as the PA. A new address is clocked into the SAR each time the MAS is accessed. This includes fetches into the SIR, reads into the store data register (SDR), and writes from the SDR.

### *Increment Logic*

**5.67** If the 3B 20D Model 1 computer is executing a program sequentially, the PA is gated to the increment logic where the address is incremented by half words (+2) or by full words (+4). The output is then gated to the SAR.

### *Store Control Register (SCR)*

**5.68** The SCR is 32 bits wide. The SAR provides most of the control for the MAS accesses. The low bits are used to invalidate the address translation buffer (ATB). The 12th bit is the pipeline pointer. The next 4 bits are the PA shadow, which aides in determining what address was being executed at the time of a failure. The 2 high bits provide control during the accesses. Bits 23 through 16 form the command portion of the SCR.

**5.69** The SAR is a zero active register which performs the function defined for a specific bit when the bit equals zero.

#### Store Data Control

**5.70** The store data control (SDC, UN06) circuit is used to buffer data coming from the main store (MAS) (read), data going to the MAS (write), and instructions coming from the MAS (fetch). This buffering of data is necessary to increase real-time effectiveness and is done by providing an overlap of store access functions during the same time the CC is decoding and executing instructions. The SDC

also aligns Opcodes during instruction fetches to their required bit positions to begin a microsequence. The SDC contains the following circuitry:

(a) Store data register (SDR)

(b) Store instruction register (SIR)

(c) Halfword multiplexer (HM)

(d) Instruction buffer (IB).

### *Store Data Register (SDR)*

**5.71** The SDR buffers the data for reads out of the MAS or the cache memory and also buffers the writes into the MAS and the cache.

**5.72** The data that is inputted to the SDR is gated and clocked by two sets of leads designated A or B. The A lead gates the SDR bits 19 through 00, and the B lead gates the SDR bits 35 through 20.

**5.73** Data is clocked into the SDR by either the destination decoder located in the microcontrol circuit (MC, UN44 ▶or UN135 effective with Generic 2◀) or by the store clock logic of the store address control (SAC, UN43) via input leads.

**5.74** Selection of data that is to be loaded into the SDR from either of the stores is via the A set of leads from the read logic of the SAC. The activation of these leads originated from the store control bits of the microinstruction register.

**5.75** The SDR data can be gated to the cache and MAS via SDR buffers that are activated by a lead from the main store update unit (MASU).

### *Store Instruction Register (SIR)*

**5.76** The SIR buffers the instructions fetched from the cache or MAS. The inputs for the SIR come from two sources. They are the data read leads from the cache and the MAS and from bits of the destination (DST) bus buffers. The clocking of the DST bus to the SIR is provided by leads which originate from the destination decoder located in the microcontrol or from the leads originating from the store address control fetch logic. The selection of the destination or store instruction inputs is under the control of a special set of leads from the SAC. If these leads are logic 0s, the cache data leads will load the instruction received from the cache or MAS into the SIR. Should these leads from the SAC become a logic 1, the DST bus will be loaded into the SIR during an active clock pulse from the destination decoder located in the microcontrol (MC) circuit.

**5.77** The outputs of the SIR are gated to the halfword multiplexer (HM). The SIR data and parity bits are divided into upper and lower halves. The upper half is bits 31 through 16 with parity bits 35 and 34. The lower half is bits 15 through 00 with parity bits 33 and 32. This set of outputs is the input to the HM.

### *Halfword Multiplexer (HM)*

**5.78** The HM buffers the instruction for the loading of the microaddress register. This is done so that the Opcode portion of the instruction may be used to formulate the starting address of the microsequence necessary to perform the instruction.

**5.79** The gating of the SIR into the HM is controlled by two separate circuits, inverters that control the gating of data bits 31 through 00 and inverters that control the gating of the parity bits.

**5.80** The outputs of the HM are connected to a 4:1 multiplexer and the instruction buffer (IB). The outputs going to the IB are a bit-for-bit transfer.

### *Instruction Buffer Register (IB)*

**5.81** The IB retains the instruction that is being acted upon by the microcontrol sequence necessary to perform that instruction. The IB register has two sets of inputs. One set is gated from the DST bus multiplexers and the other set originates at the outputs of the HM.

**5.82** The MC circuit clocks the DST bus so that the IB can be loaded. Other leads inputted to the IB from the DST bus originate from the store address control.

**5.83** Outputs of the IB register indirectly select the general registers in the DMU. Other outputs of the IB register indirectly select the amount of rotation to be performed on the destination buffer in the DMU. In addition to these outputs, the IB also has outputs to an IB parity-checking circuit. If a parity error is detected, a CC stop-and-switch is activated.

The outputs of the IB can also be gated onto the source (SRC) bus.

#### Data Manipulation Unit

**5.84** The data manipulation unit (DMU) receives data from the SRC bus and gates manipulated information onto the DST bus. The CC is equipped with two circuit packs that form the DMU. The DMU circuit packs, UN01 and UN23, each contain several circuits used in the manipulation of data:

(a) Rotate mask unit (RMU)

(b) Arithmetic logic unit (ALU)

(c) Matcher circuit

(d) Parity generator

(e) Bypass logic

(f) Find low zero (FLZ in DMU 1 only).

### *Rotate Mask Unit (RMU)*

**5.85** The RMU is used when it is necessary for the operating system to move information or to work on part of the information. The RMU contains two types of circuitry, the rotate unit (RU) and the mask unit (MU).

**5.86** The RU is located in DMU 0 (UN01) only. Bits 43 through 40 and 15 through 12 of the microinstruction register (MIR) field determine which register is to be gated onto the SRC bus and inputted to the RU. The RU then performs two operations, byte rotations and bit rotations. The RU rotates data or information right from 0 through 31 bit positions. The data passes through the RU unchanged. The outputs of the RU bit rotator supply the inputs of the MU. Whenever data is rotated, it loses its relationship to the parity bits; therefore, parity is always recalculated.

**5.87** The MU is located in both DMU 0 and DMU 1. The masking function is duplicated in the DMU. The MU receives its inputs from the RU via leads bits 31 through 00. The MU has two modes of operation, and both MUs perform the operations simultaneously. The modes of operation are as follows:

(a) ***Not Active:*** Data passes through the MU unchanged.

(b) ***Active:*** A mask is set up to screen out unwanted data, and only that data to be processed is allowed through to the ALU.

**5.88** Mask information or data comes from one of two sources: directly from the instruction, or from a programmable read only memory (PROM) that is built into the MU. If the mask information comes from the PROM, a microinstruction will supply the address of the mask it desires.

**5.89** Masking with a 32-bit mask value stored in a 512-word by 32-bit PROM is performed by the mask logic on the data output of the bit rotator.

**5.90** The PROM requires nine address lines to address one of the 512 words. Five of these address lines are shared between the byte and bit rotator, as well as the five low-address bits for the PROM. The other four address bits define the mask class.

**5.91** Each bit of the data word has an AND-OR-INVERT gate at the input of the ALU (IC 2901) to perform the correct function. One lead is brought directly from the microinstruction register (MIR) and is used to indicate the type of operation to be performed. A logic 1 will cause an AND function to be performed while a logic 0 will cause the OR function to be performed. Refer to Fig. 23.

**5.92** Whenever there is a logic 1 in the mask, the value of the bit lead is gated to the ALU and the unmasked bits are made into logic 0s.

### *Arithmetic Logic Unit (ALU)*

**5.93** The ALU is duplicated in the DMU unit to assure that the DMUs are working properly. The ALU receives input data from the mask unit (MU). The ALU can perform the arithmetic and logic operations on the inputs or let the data pass through unchanged.

**5.94** The ALU performs the following operations:

- Addition

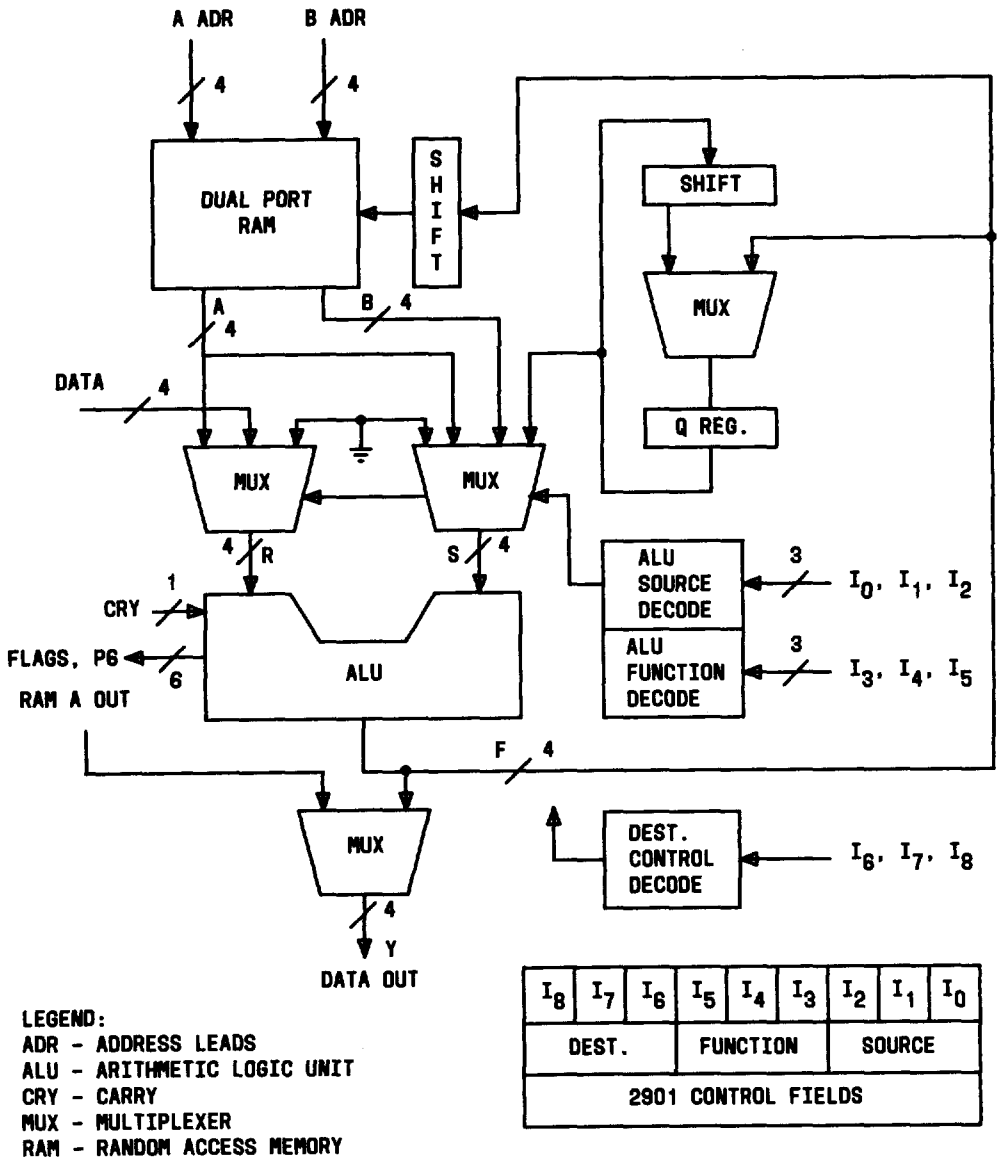



**Fig. 23—Block Diagram 2901A**

- Subtraction
- Logical OR
- Logical AND
- Logical XOR (exclusive OR).

To detect errors, both of the ALUs perform the same operation on the same data simultaneously.

### *Matcher Circuit*

**5.95** The ALU in DMU 1 gates the results of its operation to the DST bus on bits DST 31 through 00 and its image appears at a matcher in the DMU 0 via DST 31 through 00. Then the outputs of DMU 0 and DM1 are matched. If there is a mismatch, the DMU error lead becomes active and sets a bit in the error register (ER).

### *Parity Generator*

**5.96** The ALU in DMU 0 provides the input to the parity generator. The parity generator computes the parity and gates it out onto the DST bus where it joins the information bits DST 31 through 00 from DMU 1.

### *Bypass Logic*

**5.97** Bypass logic is located in both DMU 0 and DMU 1. The operating system program in the 3B computer does not require the manipulation of every word it processes. Therefore, the bypass logic circuitry is used when it is necessary to gate information from one register to another without manipulation. To do this, the bypass logic gates the SRC bus or parts of the MIR to the DST bus with the word unchanged. The microprogram determines whether the SRC bus or MIR gets gated to the DST bus.

### *Find Low Zero (FLZ)*

**5.98** The operating system determines what needs to be done by reading the information stored in registers or data words. The information may indicate that a unit needs to be tested or that a task needs to be performed.

**5.99** Bit positions indicating that some task needs to be performed are commonly referred to as flags. The FLZ logic is used to identify which bit position is equal to a logic 0, indicating that work needs to be done. Once the work is started, the operating system program sets the bit to a logic 1, preventing the system from looping on the same task over and over. The FLZ microinstruction gates the selected register to the SRC bus at the same time that it enables the FLZ circuitry in DMU 1. The FLZ searches for the rightmost (lowest) logic 0 in the information presented on the SRC bus. Then the FLZ circuit responds by indicating if a logic 0 was found and, if so, the bit position of the lowest logic 0. Next, the FLZ responds by using the DST bus leads DST 05 through 00. A logic 0 on DST 05 indicates a logic 0 was found, and a logic 1 indicates a logic 0 was not found. Leads DST 04 through 00 indicate the bit position where a logic 0 was found by using the decimal weights of the bit positions.

### Special Registers

**5.100** Special registers are contained on the UN02 and UN03 circuit packs. Bits 15 through 00 are on UN02 (special register 0) and bits 31 through 16 are on UN03 (special register 1). The registers contained in these circuit packs are as follows:

(a) Hardware status register (HSR)

(b) System status register (SSR)

(c) Error register (ER)

(d) Timer

(e) Real-time counter (RTC)

(f) Channel data register (CDR)

(g) Interrupt set (IS) and interrupt mask (IM)

(h) Pulse point register (PPR)

(i) Program status word (PSW).

### *Hardware Status Register (HSR)*

**5.101** The HSR is a 32-bit register which contains hardware and control status information. It may be loaded from the DST bus under microprogram control except for bits 4 through 7 which are read only.

### *System Status Register (SSR)*

**5.102** The SSR contains status information such as system configuration, maintenance and recovery information, and inputs from certain manual switches. Some SSR bits are loaded from the DST bus while others are read only.

### *Error Register (ER)*

**5.103** The ER is an error-detecting device and is comprised of 32 bits. Bits 0 through 10 are used for stop-and-switch type errors and main store (MAS) parity errors.

**5.104** When bit 0 is logic 0, there is a source bus/bit rotate parity error. A logic 0 on bit 1 will indicate a microcontrol parity error. When bit 2 is a logic 0, a mismatch of the auxiliary and the primary

results clock has been detected. A logic 0 on ER bit 3 indicates that an IB register parity error has been detected. When bit 4 is a logic 0, an ATB error is present. An error in the cache is present when ER bit 5 is a logic 0. The main store controller "A" (MYSERA) has an error when bit 6 is a logic 0, indicating an error has been detected in the MAS. When bit 7 is a logic 0, a My Main Store Controller Time Out error has been detected. A logic 0 on bit 8 indicates a data parity error has been detected on read operations in My Store (MYSERC). The DMU has an error when bit 9 is a logic 0. When ER bit 10 is a logic 0, an error has been detected in the My Store Address Controller.

**5.105** Error bits 11 through 26 are used for various classes of interrupt errors. In this field, errors are classified into four categories:

(a) Less serious hardware errors; ie, I/O errors, main store refresh parity error

(b) Errors related to the other CC in a duplex configuration

(c) Software-related errors; ie, privileged instruction error, ATB protection violation, accessing unequipped memory

(d) Memory management related errors; ie, ATB page fault.

**5.106** Error bit 27 is not used. Error bits 28 through 31 are used to trap the DST bus parity when the bidirectional gating register (BGR) is specified as the destination. Thus, these bits are not error bits but provide a means for examining the parity bits in the computer.

### *Timer*

**5.107** Timers in the CC count down the 20-MHz clock to produce intervals of 1, 5, 10, and 25 ms with a pulse width of 10 $\mu$s. The timer signals are available at the backplane for user options.

### *Real-Time Counter (RTC)*

**5.108** The RTC is a 32-bit synchronous counter normally incremented at a 1 ms rate. It can be loaded from the DST bus and accessed under microprogram control. A maintenance state bit can be used to inhibit counting by the RTC; it can then be stepped by a pulse point from the PPR.

### *Channel Data Register (CDR)*

**5.109** The CDR is a 32-bit register used to move data between the DST bus and the I/O bus (CCIO). It can, therefore, be loaded from either the DST bus or the CCIO bus. Refer to Fig. 17.

### *Interrupt Set (IS) and Interrupt Mask (IM) Registers*

**5.110** The IS register is a 32-bit register whose bits may be set by external signals (interrupts) or by microprogram control. The bits are only cleared by the microprogram. When a bit is set in the IS and recognized by the computer, the section specified for that particular interrupt bit is taken. The IM register is a 32-bit register whose bits are set or cleared by the microprogram. Each bit in the IM corresponds to the same bit in the IS. Setting of any bit in the IM prevents the recognition of the corresponding interrupt signal in the IS. Figure 24 shows the I/O interrupt structure and Fig. 25 shows the IS and IM registers.

### *Pulse Point Register (PPR)*

**5.111** The PPR is used to generate control pulses. It normally contains all zeros. An immediate data instruction is used to set any particular pulse bit, and a second immediate data instruction clears the bit. The pulse width is the execution time of the two microinstructions and can vary from 300 to 600 ns in 50-ns increments. Refer to Fig. 26.

### *Program Status Word Register (PSW)*

**5.112** The PSW is a 32-bit word used to control program function and record program status. Refer to Fig. 26 for the PSW block diagram and Table G for the PSW word format. The PSW is loaded from the DST bus under microprogram control.

## C. Main Store Update Unit

**5.113** The main store update unit (MASU, UN34) is required in a duplex CU or when the CU is equipped with a direct memory access (DMA). When the MASU is present in a duplex configuration, the MASU arbitrates data flow between the CC, DMA, main store(s) of the on-line computer, and the main store(s) of the off-line CU.

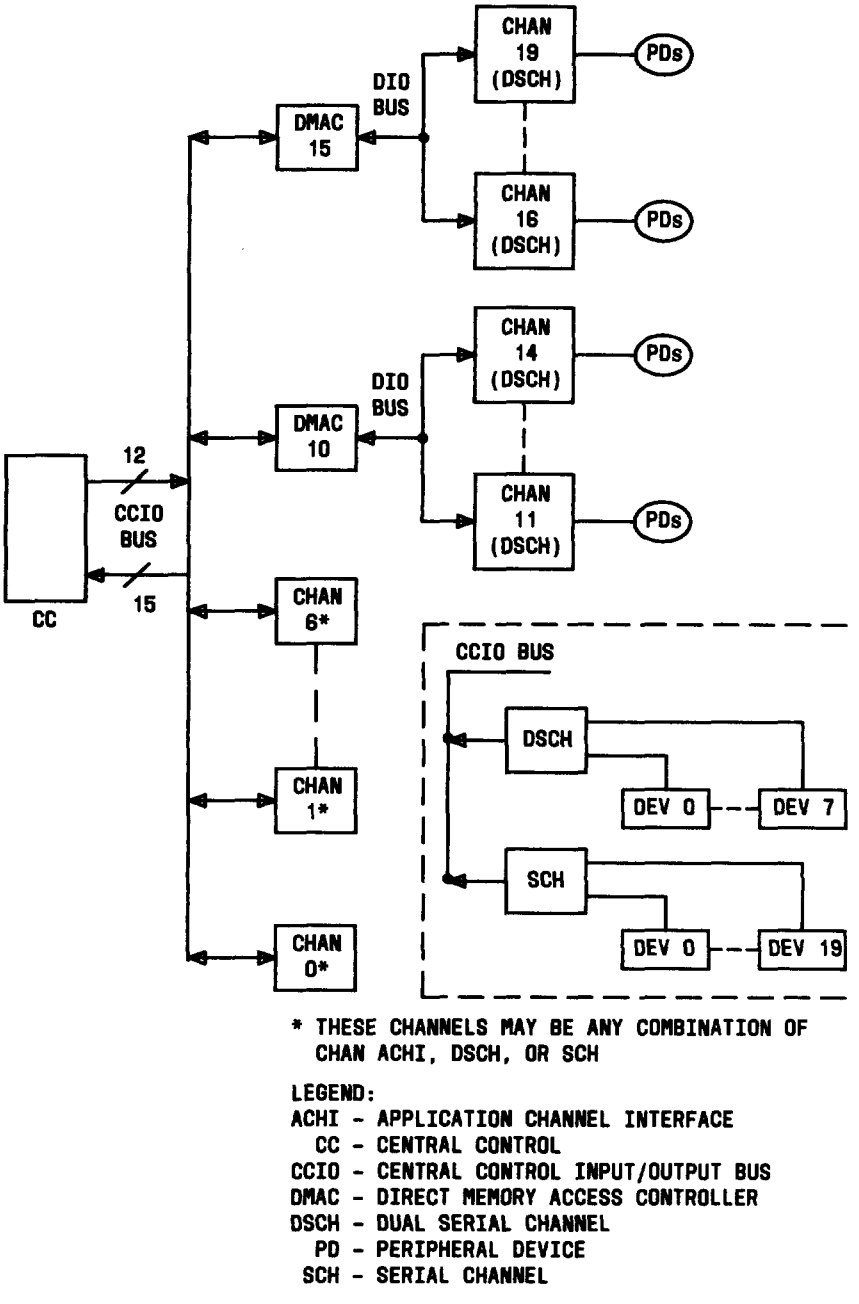



**Fig. 24—I/O Interrupt Structure**

**5.114** In addition to controlling data flow in the on-line CC by using the main store bus (MASB), the MASU also constantly keeps the data in the MAS of the off-line CU updated by using the main store update bus (MASUB). This allows the on-line CC to keep both the on-line MAS and the off-line MAS updated with the latest information.

**5.115** The MASU contains the following circuitry:

(a) MASU protection and direction

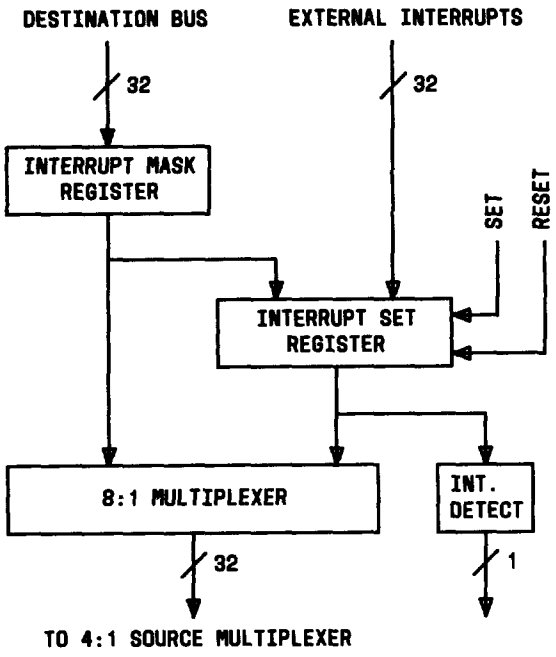



**Fig. 25—Interrupt Set and Interrupt Mask Registers**

(b) MASB enabling

(c) Priority

(d) Address bus enabling

(e) Data bus enabling.

### MASU Protection and Direction

**5.116** A bidirectional bus is used for the MASU to establish an interface from the on-line to the off-line 3B 20D Model 1 computer. This bus transmits main store data address and commands from the on-line MASU circuit to the off-line MASU circuit. These bus leads are connected to the bus transceivers of the UN34 circuit pack. Thirty-two bits of store data and 4 parity bits interface with the off-line MASU circuit. The MASU protection and direction determines the leads that should be enabled so that data being transmitted to or received from the off-line MASU will not be interfered with by other leads containing data being transmitted to other destinations located in the on-line CC.

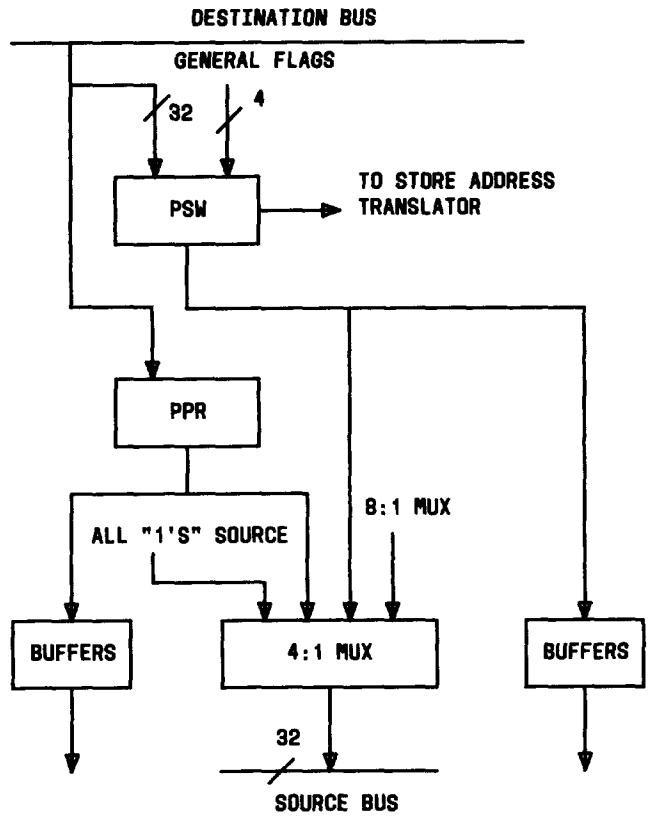



**Fig. 26—Pulse Point and Program Status Word Regis-**

### MASB Enabling

**5.117** The MASU has separate enabling leads for the address and data portions of the main store bus (MASB). The address and data portions of the on-line CC do not connect through the MASU, although the placing of data and read/write data on the MASB is under control of the MASU through its enabling leads. A request for bus usage is sent to the MASU from the CC or direct memory address (DMA) in the form of a Store Go signal. The MASU prioritizes simultaneous requests. The MASU enables the unit receiving the priority in order that the selected unit may place information on the MASB.

TABLE G

PROGRAM STATUS WORD

| BIT | ACCESS READ-WRITE | MNEMONIC | FUNCTION |
|---|---|---|---|
| 0 | RW | C | Carry Flag |
| 1 | RW | N | Negative Flag |
| 2 | RW | V | Overflow Flag |
| 3 | RW | Z | Zero Flag |
| 4 | RW | — | Emulation Control |
| 5 | RW | — | |
| 6 | RW | — | |
| 7 | RW | — | |
| 8 | RW | — | Interrupt Execution Level |
| 9 | RW | — | |
| 10 | RW | — | |
| 11 | RW | — | |
| 12 | RW | — | — |
| 13 | RW | — | — |
| 14 | RW | — | Set Execution Level Privilege |
| 15 | RW | — | I/O, Mtce Channel Privilege |
| 16 | RW | — | System I/O Privilege |
| 17 | RW | — | Write PSW, etc, Privilege |
| 18 | RW | — | Kernel Stack |
| 19 | RW | — | Interrupt Stack |
| 20 | RW | — | Memory Management Mapping On |
| 21 | RW | — | Source SSBR |
| 22 | RW | — | Destination SSBR |
| 23 | RW | — | PSBR |
| 24 | RW | — | PSBR |
| 25 | RW | — | SSBR |
| 26 | RW | — | SSBR |
| 27 | RW | — | Spare |
| 28 | RW | — | Spare |
| 29 | RW | — | Spare |
| 30 | RW | — | Spare |
| 31 | RW | — | Spare |

**Priority Circuit**

**5.118** The priority circuit arbitrates between simultaneous request for store action between the circuits associated with the MASU. The priority circuit considers which unit is controlling the MASB and gives priority to a different unit for the next store request. Each unit associated with the MASU circuit has its own circuit for resolving priority for the MASB. The priority circuit determines which of the units is presently using the MASB by an input from the address enabling buffers.

**Address Bus Enabling**

**5.119** The address enabling circuit allows the address from the on-line store address translator (SAT) to be gated through to the main store controller (MASC) and also via the MASUB to the off-line MASC when the on-line MASU is in the update mode. If the request was from the DMA, the DMA address enable also gates the physical address from the DMA store address register onto the MAS address bus. After a unit has received priority from the MASU and transmitted data, priority is shifted to a different unit requiring use of the MASB.

**Data Bus Enabling**

**5.120** The data bus enabling circuit allows the data that is to be written into the MAS to have access to the MASB. When the MASU is in the update mode, the data must be received by the off-line MASC in order to keep the off-line main store(s) loaded with current data.

**D. Maintenance Channel**

**5.121** The maintenance channel (MCH, UN22) provides a diagnostic and maintenance access to a 3B 20D Model 1 computer and between the CUs in a duplex configuration.

**5.122** Two serial data output ports (A and B) are provided on the MCH circuit pack. When used in a duplex configuration, port A of MCH CU 0 is connected via a data link to port A of MCH CU 1. This data link provides a diagnostic, status, and maintenance data path between the two CUs. Also, the MCH is interfaced to the CU via the BGB, MSD bus, MSA bus, and the maintenance buses.

**5.123** The MCH contains the following circuits:

(a) Programmable sequencer

(b) Bus control logic

(c) Transmit and receive logic

(d) Master logic

(e) Slave logic

(f) BGB interface control logic

(g) Stop-and-switch logic.

Figure 27 shows a block diagram of the MCH.

**Programmable Sequencer**

**5.124** A programmable read only memory (PROM) sequencer controls the actions of the MCH. The MCH sequencer has three basic modes of operation: idle, master, and slave.

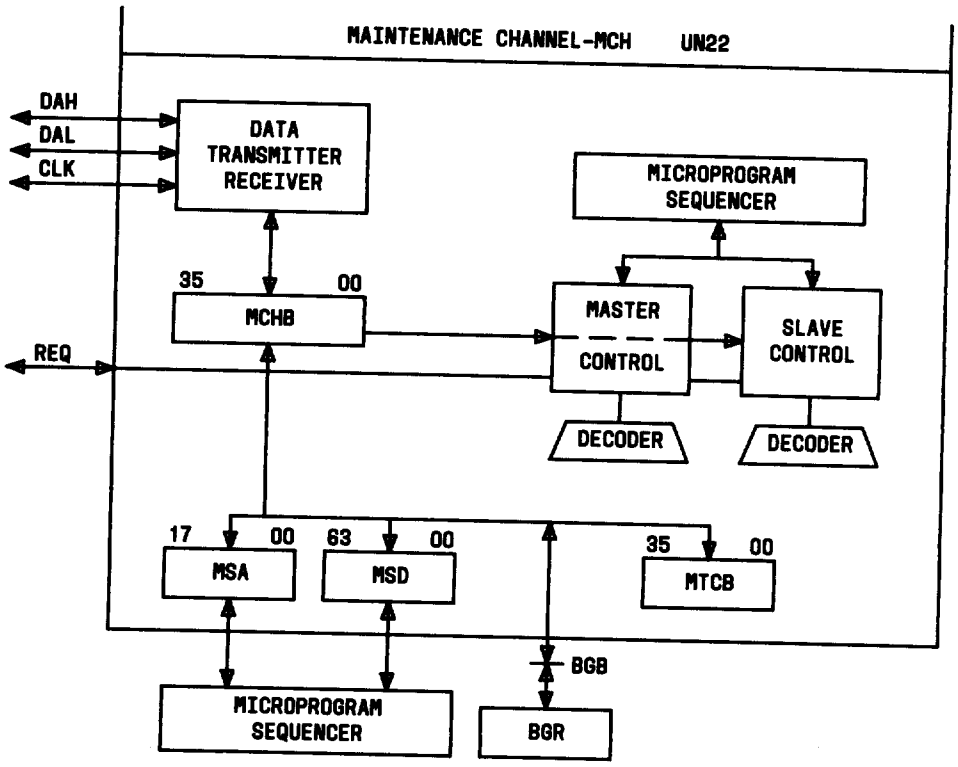
(a) ***Idle:*** When the MCH sequencer is in the idle mode, it is in a waiting state. While in the waiting state, the MCH searches for signals that will put the sequencer into either the master or slave mode. After either of these modes has entered and the desired operation has been performed, the sequencer returns to the idle state.

(b) ***Master:*** The master mode is used in a duplex configuration when the on-line MCH communicates with the off-line MCH. When the sequencer is in the master mode, a command is written into the on-line MCH, and then the MCH executes that command. The executable commands involve the transmitting of commands or data to (or receiving status or data from) the off-line MCH.

(c) ***Slave:*** All slave operations begin with the MCH receiving a serial message. This message controls the MCH regardless of its source, hence the term "slave." This serial message can come from another MCH or from a dual serial channel (DSCH). The serial message is shifted into the MCH, decoded, executed, and a response is sent back to the master computer in the slave mode.

**5.125** Two major data buses are used for data transfer within the MCH. The internal data bus (IDB) is used for most of the MCH operations. A second bus, the external data bus (ODB), is provided for the data paths that are controllable by the 3B 20D Model 1 computer. The two buses are isolated by buffers, which give the MCH the ability to execute microinstructions even if the computer has data enabled onto the ODB.

**5.126** All data movement and the control of the MCH is controlled by the PROM sequencer. The sequencer enables data onto the buses, clocks the proper registers at the proper times, enables dedicated control signals, controls transmitting and re-




LEGEND:

BGB - BIDIRECTIONAL GATING BUS
BGR - BIDIRECTIONAL GATING REGISTER
CLK - CLOCK
DAH - DATA HIGH
DAL - DATA LOW
MCH - MAINTENANCE CHANNEL
MSA - MICROSTORE ADDRESS BUS
MSD - MICROSTORE DATA BUS
MCHB - MAINTENANCE CHANNEL BUS
MTCB - MAINTENANCE BUS
PROM - PROGRAMMABLE READ ONLY MEMORY
REQ - REQUEST

**Fig. 27—Maintenance Channel Block Diagram**

ceiving serial messages, decodes commands, and tests for various conditions. Three portions of the MCH operate independently of the sequencer. The first is the shifting in of a serial message which is controlled by the transmit/receive (TR) control. A serial message may arrive at any time, and the TR control shifts the message in and notifies the sequencer that a message is present. The second is the stop-and-switch logic. It transmits and receives switch messages between duplex computers when certain error conditions exist in the system. The last is the master interface. It allows the master computer to read and write the maintenance channel buffer (MCHB), read the master status register, and write the master command independently of the sequencer.

**Bus Control Logic**

**5.127** For the MCH to maintain and diagnose the 3B 20D Model 1 computer, it must control and monitor several of the buses, the most important of which is the microstore data bus (MSD). Using the MSD register located in the MCH and a control signal to the computer, the received serial data may be loaded into the MSD register, the microstore removed from the MSD, and the MSD register enabled onto the MSD bus. When the CC is single-stepped by the MCH, the data placed on the MSD will be executed as a microinstruction. This replaces the normal microinstruction source, the microstore.

**5.128** The MCH may also monitor the MSD bus as well as drive it. Data patterns may be placed on the MSD and then returned and compared with

the original pattern to check the MSD register and drivers in the MCH. Therefore, proper operation of the MSD and associated logic can be verified before executing the microinstructions.

**5.129** The MCH may also remove CC data from the microstore address bus (MSA) and load the MSA with data stored in the MSA register. Data on the MSA may also be read into the MCH. As with the MSD, the data read in may be the data enabled out for loop-around checking.

**5.130** The MCH can also read the bidirectional gating bus (BGB) and the maintenance bus (MTCB). The BGB is enabled when the proper slave command is received. The MTCB is different, however, because a select field is associated with it. The select field tells the computer which of the six sources to supply to the MTCB. Because reading the MTCB does not interfere with the operation of the computer, the MCH can monitor status, error conditions, etc, in an operational computer.

#### Transmit and Receive Logic

**5.131** The transmit and receive logic (TR) provides the serial interface for the MCH. In the slave mode, it receives the DSCH messages on either of the two ports and informs the PROM sequencer that a message is present. When the operation is complete, it accepts the return response and transmits it out on the port which initially received the request. In the master mode, the TR logic functions in the reverse, accepting the start code and data from the MCH and transmitting it to the off-line computer, and then receiving the return message sent by the off-line computer.

**5.132** The TR logic consists of the differential line drivers and receivers, which interface to the DSCH-type bus, and the shift register necessary to receive and transmit the messages. There are two separate shift registers, one for the entire message and one for the start code portion only. By shifting both registers with the same incoming clock, the start code shift register gives an indication of any message present after four clock intervals. (The beginning of every start code message is 4 bits long.) This allows the PROM sequencer to decode the start code while the TR logic shifts in the rest of the message. The TR logic also determines which port to transmit response to and blocks the other port while an operation is in progress.

#### Master Logic

**5.133** Master commands usually involve transmitting and receiving a serial message to and from the off-line MCH. There are a few master commands that are provided primarily for testing the master portion of the MCH and do not involve any transmissions. The master commands are 16 bits long. The low 8 bits contain the actual command to the on-line MCH. The next 6 bits contain the start code the on-line 3B 20D computer wants the MCH to transmit to the off-line MCH. This field, used only when the command involves transmitting or receiving information, gives the on-line computer the ability to send any desired start code. The upper 2 bits of the master command register are not used.

**5.134** All master commands begin with the MCH in the idle state. The computer determines if the MCH is in this state by testing bit 0 (idle) of the master status register. If the bit is set (a logic 1), the MCH is idle. If the idle bit is not set (a logic 0), the MCH is either executing a previous command or hung in an unknown state. If the computer desires, it may issue an initialize pulse point to the MCH; this initializes the MCH and puts it into the idle mode. Once in the idle mode, the master command register and the maintenance channel buffer (MCHB) can be written by the computer.

**5.135** The master command register is written over the bidirectional gating bus (BGB) with the command to be executed by the on-line MCH. To write the master command register, the 3B 20D computer must place data on the BGB and then activate the proper pulse point. The MCHB is written in a similar manner using a different pulse point. The MCHB needs to be written only if the command loaded requires data, such as send data or send command. Once these registers have been loaded, the CU can activate the execute pulse point. This pulse point clears the idle bit in the master status register and informs the MCH sequencer that an execute pulse has been received.

**5.136** Once the execute indication is recognized by the MCH sequencer, the idle state is left and the master mode entered. The MCH sequencer now controls all activity of the MCH until the idle mode is entered again.

**5.137** All master commands begin after the execute pulse point is detected. The first step of

the master sequencer is to decode the master command. This is done by enabling the master command register onto the internal data bus (IDB). Next, the MCH sequencer does a 16-way branch on the low 4 bits of the IDB. This decodes the command into 1-out-of-16 and branches to the transmit long sequence. The MCHB contains the data or command that is to be sent to the off-line CU, and is enabled from the MCHB onto the external data bus (ODB) and then to the IDB. At the same time, the start code which was loaded into the master command register by the computer is enabled through the start code multiplexer (SC MUX) and onto a dedicated path to the transmit/receiver (TR) register start code bit positions. After the TR register is clocked, it contains the full serial message in the proper format to be sent to the off-line MCH. Clocking the TR register also turns on the serial line drivers and starts them outpulsing the clock signal.

**5.138** The transmit signal is now activated by the sequencer. This signal tells the TR logic to begin shifting the data out of the on-line MCH to the off-line MCH. The MCH stays in this shifting mode until the entire message has been transmitted, at which time the transmit signal is removed. The TR register is then cleared and the sequencer enters a loop waiting for the response from the off-line CU. The return response is shifted into the TR register and the start code shift register by the received clock sent by the off-line MCH. When the start code shift register is full, the sequencer is notified that the response is back. The sequencer then stops outpulsing the clock and goes into another loop, waiting for the incoming clock from the off-line MCH to stop. When this occurs, the serial portion of the operation is complete and the sequencer is now ready to check the return code and complete the master sequence.

**5.139** The sequencer enables the returned start code onto the IDB and does a 16-way branch on it. This checks the return code sent back by the off-line MCH so the sequencer can notify the on-line CU whether or not the transmission was completed properly. Based on the results of the start code test, the master status register is loaded with the idle bit (bit 0), a one. If there were any errors in the master sequence or a bad return code was received, the proper additional bits are also set. Finally, the TR register is cleared and the sequence returns to the idle state to wait for further commands. If the on-line CU had been reading the master status register and testing for the idle bit, it would now find the idle bit set, indicating that the MCH is ready for another operation. By looking at other bits in the register, the on-line CU could determine whether or not the operation was successful.

#### Slave Logic

**5.140** All slave operations begin with the MCH receiving a serial message. The message may come from any source that uses the DSCH format. The received message causes the MCH to take one of four possible actions: return data, return status, receive data, or receive a command. When the MCH has completed the requested action, a serial message is sent to the on-line CU to indicate completion and to return any data requested. Since the duplex configuration is of more importance, slave operations will not be discussed in depth.

#### BGB Interface Control Logic

**5.141** The bidirectional gating bus (BGB) interface control logic provides the interface between the computer and the MCH. The computer uses three of the bits in the BGB control field of the hardware status register (HSR) and five of the bits in the pulse point register (PPR) to handle the reading, writing, and controlling of the on-line MCH. The bits in the HSR provide a dedicated enable to each of the circuits using the BGB, a direction bit to determine whether a read or write is requested, and three read-select bits for selecting a particular source within the circuit. The MCH uses only the least significant bit of the read-select field as the CU can read only two registers, the MCHB register and the master status register.

#### Stop-and-Switch Logic

**5.142** When the on-line CU in a duplex configuration detects a fatal hardware error, system control must be transferred to the off-line CU. This transfer of control, referred to as stop-and-switch, is implemented by the MCH. The MCH sends an interrupt to the off-line CU if the on-line CU develops a fatal fault.

**5.143** When a DSCH is used as an interface to the MCH, a control pulse of 550 nanoseconds (ns) or greater transmitted from the MCH to the DSCH is detected as an interrupt. This interrupt request (stop-and-switch) is then gated to the CU.

**5.144** The MCH gates a 550-ns pulse out of both ports whenever a stop-and-switch error occurs. In addition, the MCH receiver will initialize the CU upon detection of the 550-ns pulse. In a duplex system the MCHs are connected together and, if MCH sends out a 550-ns pulse, the other MCH, upon reception, will start the CU by activating the initialize signal, maintenance reset function (MRF). In nonduplex arrangements, the 550-ns pulse is sent out on both ports and is interpreted by the receiving DSCHs as an interrupt.

**5.145** The MCH uses a PROM sequencer to generate and detect these 550-ns pulses. This sequencer is similar to the MCH control sequencer. It has an 8-bit pipeline register (a pipeline register speeds up the execution time) with five of the outputs used as address inputs to the 32-word, 8-bit PROM. The pipeline register is clocked every 100 ns by the MCH clock. On every clock pulse the high four address bits are loaded from the PROM and the least significant address bit is loaded from a buffer flip-flop. The sequence stored in the PROM is such that if the buffer flip-flop is cleared, the sequencer will loop at address zero.

**5.146** The received stop-and-switch signals are fed to the data input of the buffer flip-flop. This flip-flop is clocked every 100 ns by the MCH clock. When the received pulse is active, the buffer flip-flop will set, changing the least significant address bit from a 0 to a 1. The address bit remains a 1 as long as the incoming pulse is present. The sequencer then begins executing the data stored in the PROM, making the sequencer transfer from one odd address to the next every 100 ns. When the buffer flip-flop is cleared, the sequencer will go to the next even address rather than to the next odd one. If this occurs after 5 clock intervals (500 ns) but before 6 intervals, the incoming pulse has a width of 500 to 600 ns. Since 550 ns falls in the middle of this range, the sequencer assumes it has received such a pulse and activates the B output of the pipeline register. This output causes the MRF to be sent to the computer if it is not on-line, or sets the MCH error bit in the computer error register (ER) if it is on-line. If the buffer flip-flop is cleared either before 5 clock intervals or after 6, the sequencer returns to the address zero loop and takes no action.

**5.147** The sequencer also generates the 550-ns pulse when a stop-and-switch is requested by the computer. The stop-and-switch request is stored in a flip-flop. The flip-flop output starts the sequence by setting the buffer flip-flop as the received stop-and-switch pulses do. It also enables the bus drivers and the data input to the driver. This data input comes from a latch connected to the A output of the sequencer. The latch is cleared every 100 ns by the MCH clock signal. At the first odd address the sequencer enables the A output, setting the latch, and causing the stop-and-switch pulse to begin. The A output is kept active by the sequencer for 500 ns and then removed. The polarity of the clock signal used to clear the latch is such that it will remain set for an additional 50 ns. When the latch is cleared, the pulse being transmitted is removed, resulting in an accurate pulse width of 550 ns.

**5.148** The sequencer continues through the sequential odd address until it can guarantee that clearing the buffer flip-flop will not cause the MRF output to fire. When this has been insured, the C output is activated, clearing the stop-and-switch flip-flop.

### E. Cache Store Unit

**5.149** The optional cache store unit (CSU) is a 2K word high-speed memory which functions as a local storage area for the most frequently used data words. Since the access time to the CSU is approximately 200 ns compared to 800 ns for a main store (MAS) access, real time can be reduced when a data word is stored in the CSU.

**5.150** Each word read from MAS is also stored in the cache memory, and the cache address is linked to the MAS address of that word. If a succeeding fetch request is to the same MAS address, the word in cache is returned to the CC and the remainder of the MAS fetch is terminated. If the word is not found in cache, it is fetched from MAS to the CC and is also stored in the CSU. In this manner, the 2K most often fetched words will usually be found in the CSU. The actual contents of the CSU will vary according to what program segments are executing, but will change at a relatively slow rate. Figure 28 is a block diagram of the CSU.

**5.151** The 24-bit address is gated on leads SA000 through SA230. Since the CSU operates only on full 32-bit (4-byte) data words, leads SA000 and SA010 are ignored because they define bytes within a data word. Bits SA020 through SA100 address the 512-word address tag and cache store memory group.

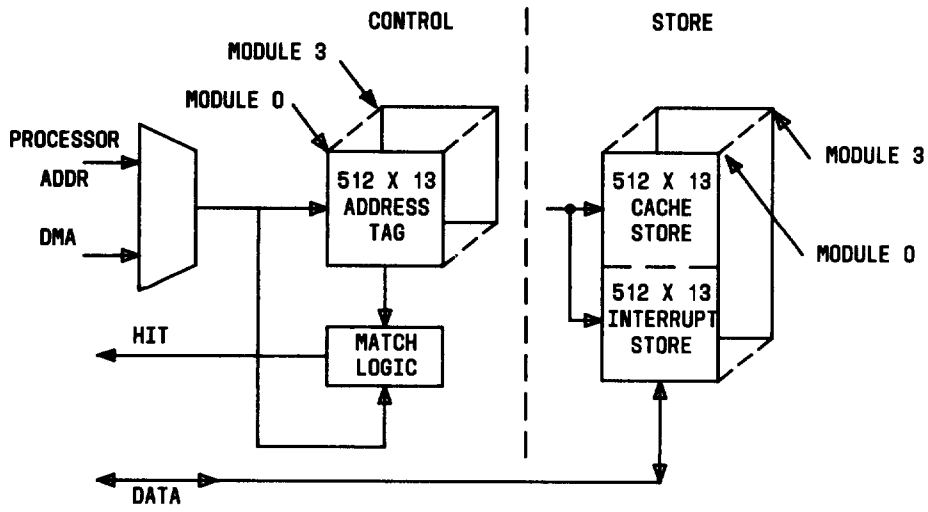



**Fig. 28—Cache Store Unit Block Diagram**

The 13 most significant bits of the address, bits SA110 through SA230, control matching logic that determines whether the addressed word is in the CSU.

**5.152** The CSU control section contains four 512-word by 13-bit groups which contain the 13-bit address tags (bits SA110 through SA230). The data word corresponding to each of these tags is stored in the store section of the CSU. A "hit" occurs when there is a match between the 13 most significant bits of a MAS address and one of the address tags, when the validation bit is set, and the CC is performing a memory read.

**5.153** When the system is initialized, all CSU locations are cleared. As MAS accesses are made, the same words are placed into the CSU until all four 512-word groups are filled. Subsequent MAS reads replace the CSU words in a manner determined by a random replacement algorithm.

**5.154** Memory accesses are initiated with a Cache Go signal. This begins a search in the CSU tag stores for an address match. If a valid match is found, the CSU generates the "hit" signal. The associated data word is then gated onto the cache data bus. The CC accepts the "hit" signal as being a CSU complete and gates the data on the cache bus into SIR or SDR.

**5.155** If an address match is not found, the CC times out in 250 ns and generates the MAS read signal. When the addressed word is returned from MAS, the cache control stores the 13 most significant address bits in one of the address tag groups and sets the associated validation bit. The data word is stored in the cache store at an empty location if the cache is full. The next access to the same address will find the word in cache, if the word has not been replaced in the meantime.

**5.156** There is some contention between the CC and DMA for MAS access. The majority of CC accesses will be to the CSU while the DMA is accessing the MAS. The DMA read operations do not affect the CSU reads or writes, but the DMA writes to the MAS must also determine whether the same data word is also in the CSU. If this is so, the CSU data word must also be changed.

**5.157** The CSU arbitrates simultaneously the DMA writes and CC requests to the CSU. After the CC issues the Cache Go signal, it tests the status of the DMA busy/idle lead to ensure that the CSU has recognized the Cache Go signal. If the CSU is busy with a DMA write-hit check, the CC waits until the CSU becomes idle.

**5.158** The CSU also functions as a subroutine stack so that write operations on both the CSU and MAS can be performed instead of the normal mode

of updating the cache data, only if there is a "miss" during a cache read operation.

**5.159** The bypass mode of operation allows direct access to the MAS by the CC. The bypass mode is initiated by the cache bypass signal, which inhibits "hit" signals from being generated and inhibits normal CSU operations. Data in the CSU is kept updated by the normal hit-checks during write operations.

**5.160** A maintenance signal allows special access by the CC to the CSU for maintenance operations and the CSU initialization. The maintenance mode allows access to the address tag memories, cache data memories, and control registers in the CSU. Initialization causes all locations to be invalidated so that the cache memory is logically empty until the MAS accesses begin.

**5.161** The CSU contains a functionally 2K-word buffer which is used as an interrupt stack. As noted earlier, the cache memory consists of four groups of 512 words (one memory page). Since the cache memories are 1K-word semiconductor memory elements, the remaining 512 words in each group are linked to form a separate 2K-word interrupt stack. Addressing and control of the interrupt stack is done by the CC.

### F. Input/Output Channels

**5.162** The CC unit provides two positions for I/O channels. These positions (0 and 1) may be any combination of the serial channel (SCH), dual serial channel (DSCH), or application channel interface (ACHI).

#### Serial Channel

**5.163** The serial channel (SCH, UN26) can communicate with a maximum of 20 peripheral devices (PDs) via individual transformer-coupled cables, one cable for input and one cable for output for each device. A serial message is transmitted to the selected device, and a response is returned. Each message contains a 3-bit start code and a 36-bit data message. The SCHs may also connect to 16-bit devices, in which case software commands cause a 21-bit serial message to be transmitted (16 data bits, 2 parity bits, and the 3-bit start code).

**5.164** All signals on the SCH are transformer-coupled to provide dc fault isolation and are bipolar to prevent a dc voltage level from appearing on the cable.

#### Dual Serial Channel

**5.165** A dual serial channel (DSCH, UN09) communicates with a maximum of 16 PDs over separate pairs of I/O cables to each device. For write operations, two serial messages are simultaneously transmitted to the selected device and two responses are received. For read operations, two messages are sent to the device and two messages are received. The message format is identical to the SCH. The DSCH is used as an I/O channel or a DMAC channel.

#### Application Channel Interface

**5.166** The application channel interface (ACHI, UN19) provides an isolated parallel interface between a PD and the CU. The ACHI consists of a 36-bit input register, a 36-bit output register, flag flip-flops, and interface logic to the CCIO bus. The interface to the PD is via differential dc-coupled line drivers and receivers.

### G. Microlevel Test Set Interface

**5.167** The microlevel test set interface (MLTS, UN16), an optional circuit pack located within the MLTS, interfaces the MLTS to the CC. The MLTS interface permits external access to the microinstruction structure, buses, and registers of the 3B 20D Model 1 computer. The MLTS must be installed in the CC to provide access by the MLTS. The MLTS can test, debug, and troubleshoot the computer core hardware and firmware (microprogram).

### H. Utility Circuit

**5.168** The utility circuit (UC, UN21) is an optional unit that functions to monitor operations between the CU and computer memory. The UC consists mainly of matchers (comparators) and transfer-trace circuitry. Refer to Table H for the types of matchers provided by the matcher logic.

**5.169** Programmable operations to the matchers, memories, and control registers are via the bidirectional gating bus (BGB). Also, matcher setups and interrogations are via the BGB.

TABLE H

UTILITY CIRCUIT MATCHER LOGIC

| TYPE | NUMBER | FUNCTION |
|---|---|---|
| Address | 4 | Memory Address |
| Data | 4 | Memory Data |
| Access | 4 | Memory Access (read, write, fetch, etc) |
| Block | 1 | Memory Address Range |
| Utility | 4 | Software Process |
| Condition | 1 | Stop-and-Switch, Maintenance Reset Function and External Input |

**5.170** The transfer-trace circuitry records program transfers and process changes (utility identification changes). Program transfers include direct program transfers, calls to and returns from subroutines, and transfers to/from interrupt routines. For each of these, the UC stores a FROM and a TO address in a 256-word transfer-trace memory. Process changes are recorded each time program execution changes from one process to another. Each change results in writing a utility identification (UID) value for the new process into the transfer-trace memory. Since process changes and program transfers share the same memory, flag bits are assigned to each memory word. These flags (bits) record the type of entry of each word.

**5.171** Figure 29 is a detailed block diagram of the UC. All commands sent to the UC are decoded from the BGB data bus and control lines. The BGB data bus is duplicated in the UC and routed to all memories and registers. All UC matchers are comprised of memory arrangements with data supplied to/from these via the duplicated BGB.

**5.172** The 3B 20D Model 1 computer memory address and data and control signals are latched in the UC counters/latches. These devices latch the signals when comparing, and act as counters when reading or writing over the BGB. Outputs from these circuits supply the address to the matcher memories. The memory address is multiplexed before latching to enable selection between virtual or physical address. Instruction addresses are held in a separate latch for input to the transfer-trace memory. The instruction latch and the UID map are switched to the transfer-trace memory for program transfers and process changes, respectively. The transfer-trace counter controls the memory address at which each entry is written.

**5.173** The UID matcher compares against the UID map for each memory cycle. The map is updated by microcode each time a new process is assigned an ATB slot.

**5.174** The trigger functions receive the matcher outputs over the repeated BGB. For maintenance reasons, the matcher outputs can be substituted for user-supplied data through the BGB. Trigger Function 0 is hardwired to an 8-bit programmable event counter. The counter increments each time trigger Function 0 becomes active. The event counter output is provided as a selectible input to trigger Functions 1 through 3.

**5.175** The UC control register directs global operations within the UC. These include virtual or physical address selection, transfer-trace modes, freeze or run mode, and the condition comparator match value. The condition comparator functions as an input to all trigger functions. The UC status register provides status information of the trigger functions and interrupt logic.

**5.176** The UC uses random access memory (RAM) circuits for matching. The building block for these circuits is a 256-word by 4-bit static RAM. The address, UID, and block comparators consist of a 256 word by 12 bit circuit. The data comparator circuits are 256 words by 16 bits, and the access comparator is 16 words by 4 bits.

**5.177** Figure 30 shows the matching scheme applied to the address comparators. The RAM 0 through RAM 2, respectively, compare against byte 0 through byte 2 of the computer memory address. The column 0 output of each RAM is connected to form address comparator 0 (AC0) as columns 1 through 3 are connected to form AC1 through AC3. Match values are written into the RAMs as 1s and nonmatch values as 0s. In the example shown in Fig. 30, AC0 is set up to match on address 030100. When-

ever this value addresses the RAM arrangement, a 1 is represented on the AC0 line. Any other value would force the AC0 line to a 0 for a nonmatch condition.

**5.178** All that is required of the mask values is to write multiple 1s in the corresponding column positions. For example, writing 1s into every address position in RAM 0 for column 0 would indicate a match condition for every value of byte 0 in AC0.

**5.179** Address data is gated to each RAM through an 8-bit counter/latch. This device acts as an 8-bit counter during Test Utility System (TUS) controlled reads and writes of the comparator. The TUS has available to it two control functions associated with the counters. It can either zero all counters or increment them by one. The normal procedure for loading the comparator is to first zero the counter and then write location 0 of the RAMs. These operations are followed by incrementing the counter and writing the RAM until all 256 locations are written. The data written into the RAMs is supplied over the BGB as part of the write command.

**5.180** The counter/latch serves as latches during normal address comparing. The memory address is latched on each memory cycle and fed to the RAM address ports. The match values from the RAMs are directed to their respective trigger functions.

**5.181** The TUS-controlled reads of the comparators also use the counters for supplying the read address. Since the outputs of all the RAMs are wired together, a single read order would result in a logical "OR" of the RAM data. Therefore, three separate read orders are issued to return RAM information. Each of these orders returns data only from its respective RAM.

**5.182** The utility identification (UID) comparators use the same RAM arrangement shown in Fig. 30. The data comparator includes an additional RAM in its memory arrangement for comparing on 32 bits of data. The access comparator uses a single RAM with only 16-word locations used.

## 6. MAINTENANCE

**6.01** The CC uses a large set of self-checking logic so that most errors will be detected within a very short period of time. The error detection strategy is to use parity checking, results matching, and duplication of logic.

**6.02** Parity checking is the most commonly used error-detecting method in the CC. Most registers have four parity bits (one bit for each byte) associated with them. Parity is generated whenever data is moved from or through the ALU to a destination. Parity is checked whenever the data is used. A parity error causes bit 0 of the error register (ER) to be set, which results in a CC stop-and-switch action (except under special circumstances). The registers that maintain parity are BGR, TEMP, FIRM, PSW, PPR, CDR, IM, SAR, PA, SDR, IB, and HM. The registers that do not have parity are IS, HSR, SSR, ER, and RTC.

**6.03** Duplication of logic circuitry is used whenever parity checking does not provide confidence in error detection or when the amount of circuitry required becomes too great. The mask logic and ALUs in the DMU are duplicated, as well as the carry-look-ahead, carry in, A and B address leads, ALU instruction leads, some MAS control logic, and find-low-zero (FLZ) logic. Refer to Fig. 31 and 32.

**6.04** One of the duplicated ALUs is used for data operations while the other generates parity over the operation; therefore, if either ALU malfunctions, an error should be detected.

**6.05** The output of the parity ALU is matched with the DST bus by a set of eight comparators. Bit 9 of the ER will be set if the matcher detects an error. Refer to Fig. 32.

**6.06** The maintenance channel (MCH) provides a serial link into the microcontrol structure of the CC. This provides maintenance and diagnostic access between the CCs of a duplex system.

**6.07** The primary fault recovery technique is to switch CUs when a fault is detected in the on-line computer. Software and hardware diagnostics can then be performed on the faulty unit via the MCH.

**6.08** In a duplex configuration the MCHs of the two CUs are connected together. All communications between the two CUs take place over this path. The two serial ports on the MCH, A and B, allow it to be connected to the MCH of the other CC and to the

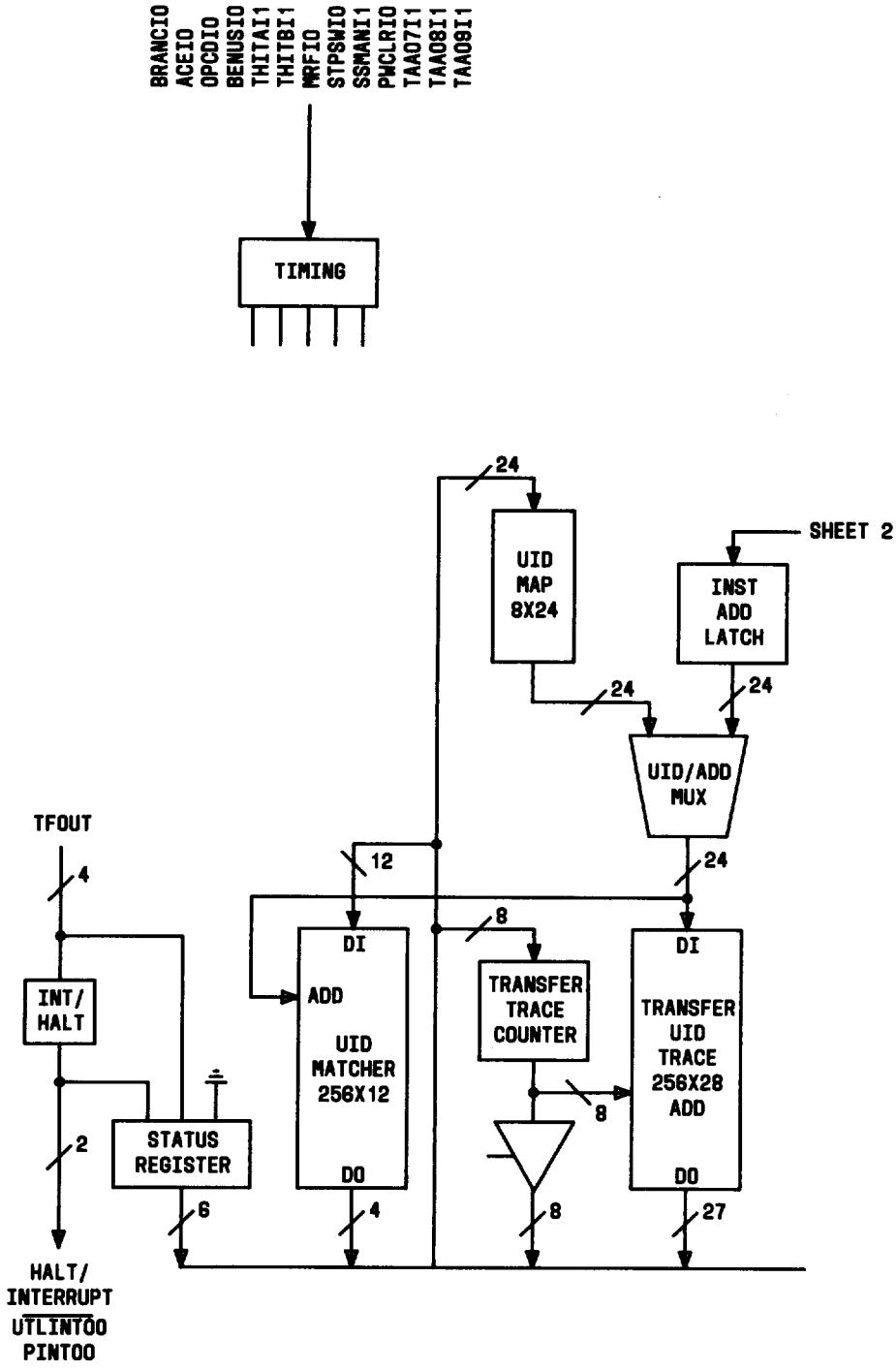



Fig. 29—Utility Circuit Layout (Sheet 1 of 2)

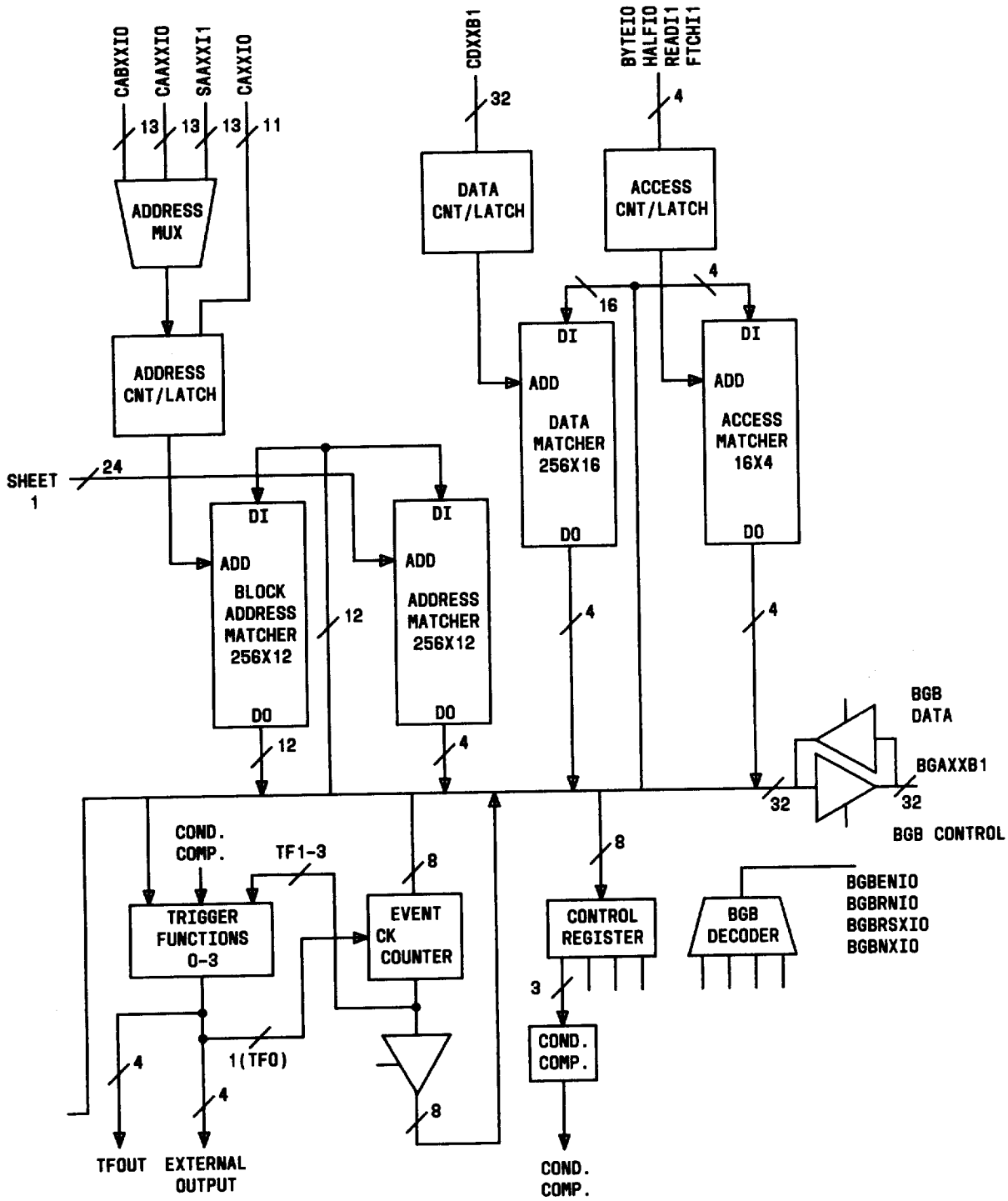



Fig. 29—Utility Circuit Layout (Sheet 2 of 2)

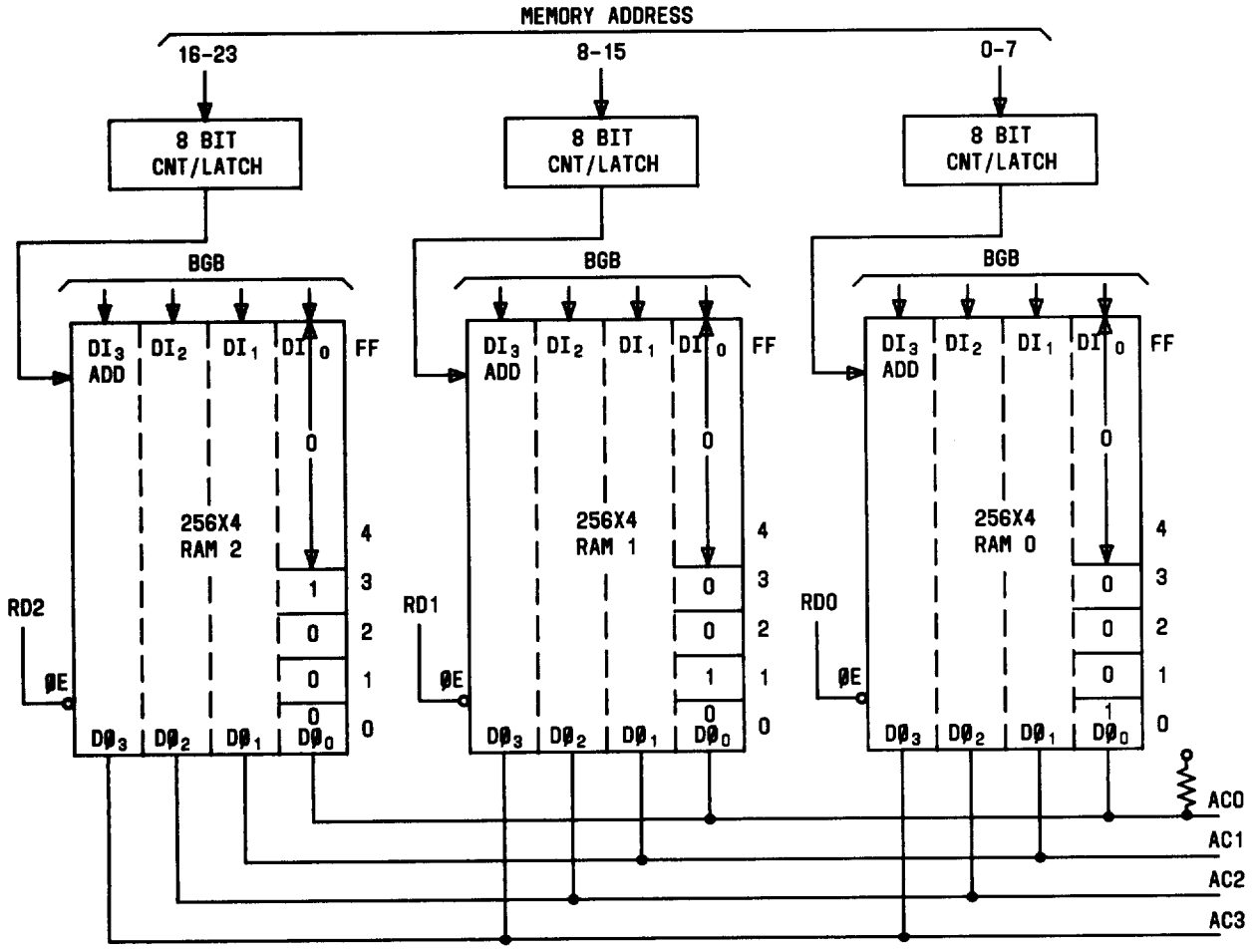



**Fig. 30—Address Comparator Block Diagram**

DSCH. The A ports of the two CCs are connected together.

**6.09** There are two major data buses in the MCH: the internal data bus, which is used for most MCH operations, and the external data bus for data paths controllable by the 3B 20D computer.

**6.10** The MCH sequencer handles all data movement and control within the MCH. The sequencer, a PROM, enables data onto the buses, clocks registers, enables control signals, controls transmitting and receiving serial messages, decodes commands, and tests for various conditions.

**6.11** The MCH is able to control and monitor several of the CC internal buses, the most important of which is the microstore data bus (MSD). The MSD register in the MCH can be loaded from the serial channel, the CC microstore isolated from the MSD bus, and the contents of the MSD register gated onto the MSD. In this manner microinstructions from the MCH can be executed in the CC in a single-step mode. The MCH can also monitor the MSD bus. Data patterns may be placed on the MSD, then compared with the original data in the MSD register. This loop-around check verifies both the transmitting and receiving paths.

**6.12** The microstore address bus (MSA) is accessed in the same manner as the MSD bus. Data can be stored in the MSA register, then gated onto the MSA bus. Loop-around checking is done or computer data used in the same manner as the MSD. The bidi-

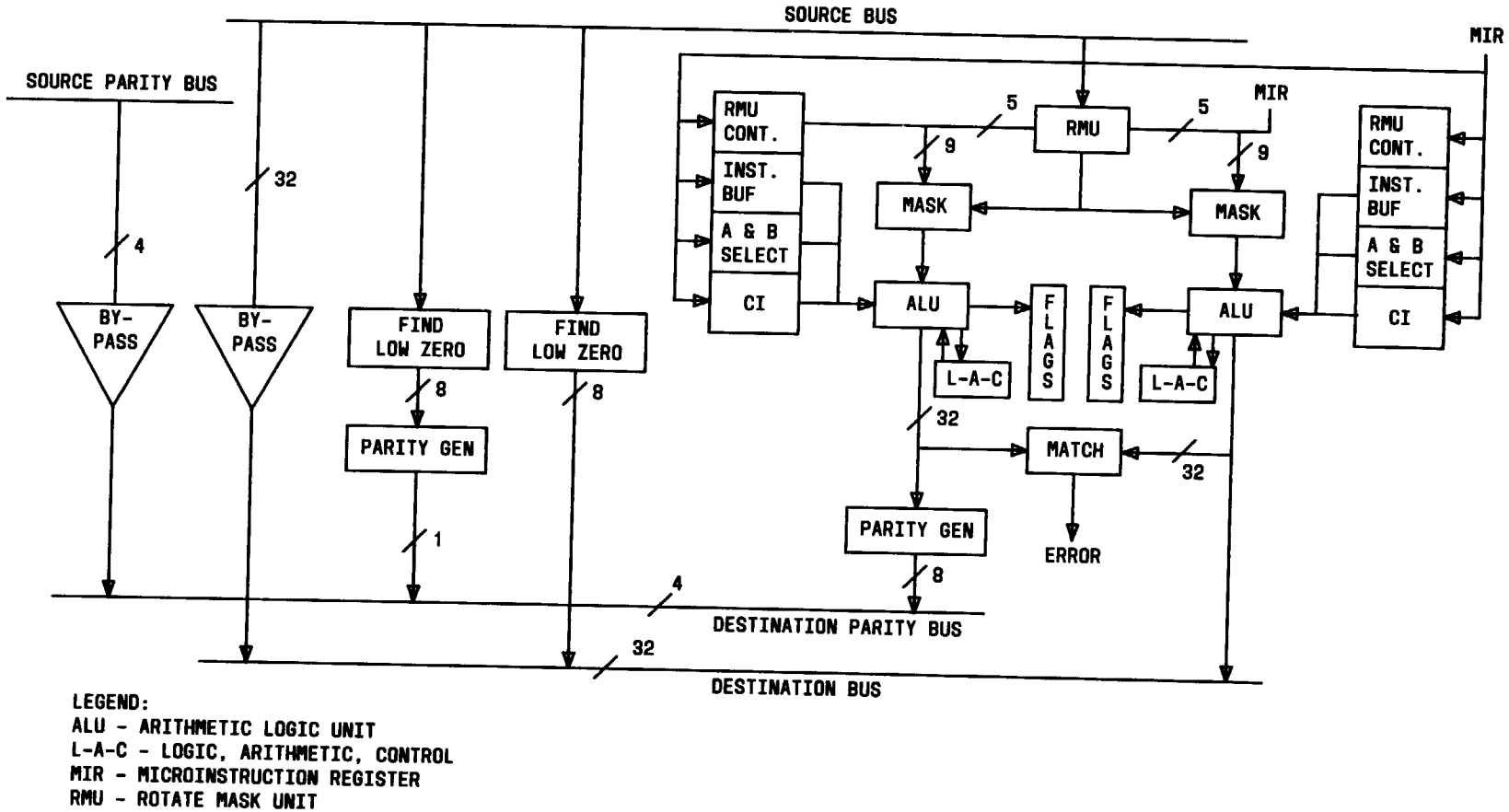



**Fig. 31—Duplication and Mask Logic Diagram**

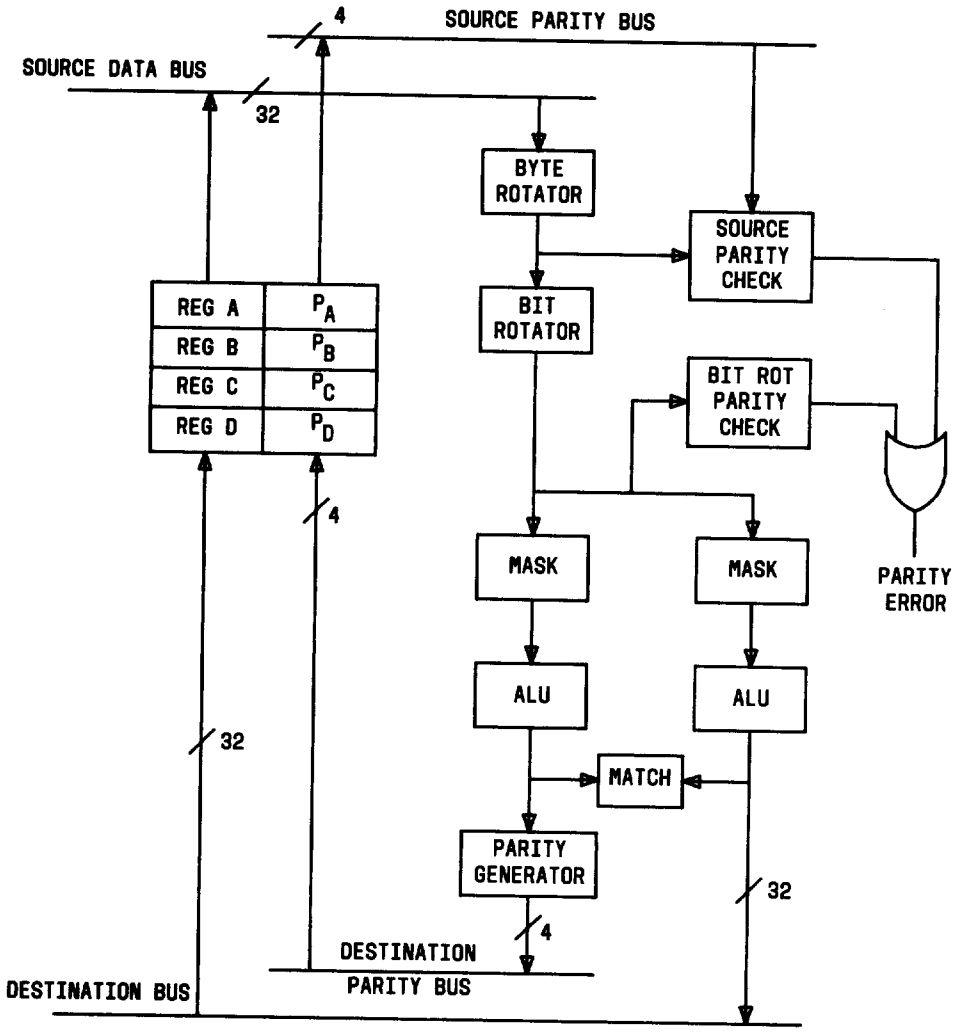



**Fig. 32—Self-Check Example**

rectional gating bus (BGB) and the maintenance bus (MTCB) can also be accessed by the MCH.

**6.13** The MCH uses the same serial data format as does the DSCH to ensure compatibility between the MCH and external inputs via the DSCH. The MCH uses a differential dc bus with five pairs of lines, two bidirectional signal data lines, two 10-MHz clock lines and a request or interrupt line. The two data lines operate in parallel, the high data bytes on one line and the low data bytes on the other. This gives an effective data transfer rate of 20 MHz.

**6.14** An emergency action interface (EAI) unit is part of the CU and provides some manual control to be used primarily for troubleshooting. The EAI unit is located in the CU frame adjacent to the MAS module. It accepts inputs from the serial channel and produces control signals to initiate certain operations. Indicator lamps on the front panel provide status information (see Table I).

**6.15** The microlevel test set (MLTS) is used to access the MSA, MSD, MCHB, and BGB through a UN16 circuit pack installed at location 040 in the CC. Commands can be inputted from a user TTY terminal or from a file of commands in a host computer. These commands result in execution of special functions and sequence control.

**6.16** The MLTS receives inputs from a local TTY terminal or from a remote TTY over a tele-

**TABLE I**

**EMERGENCY ACTION INTERFACE (EAI) UNIT INDICATORS AND FUNCTIONS**

| INDICATOR | FUNCTION |
|---|---|
| Run | Indicates CU is executing macroinstructions |
| Active | Indicates that CU is on-line. |
| Emergency Action Enable | Indicates that this CU is in the emergency action mode, enabling control of other EAI functions |
| Forced On-Line | Indicates that this CU has been forced into the on-line mode |
| Forced Off-Line | Indicates that this CU has been forced into the off-line mode. |
| Status | A 1-digit hexadecimal display which indicates the last successfully executed step of the sanity diagnostic sequence, providing a software-driven verification of system sanity in the event of a TTY or duplex IOP failure. (Low bits of the System Status Register.) |

phone line, or from a computer which can be either local or remote. The CC can be operated in a single-step mode, break points can be established, results of operations can be matched with expected results, and returned data can be printed out.

**6.17** The MLTS is not expected to reside on site but will be available on short notice when maintenance problems arise that are not solved by normal diagnostic routines.

**6.18** Since the CC has no control panel of its own, communications with it take place over the serial channel from a video-type terminal and a printer. Figure 33 depicts the proposed local maintenance position. The maintenance technician can initiate software diagnostics from the video terminal and receive data on the terminal and on the printer whenever a hard copy is required.

**6.19** Manual control is provided on the power switch ABB1 located by the EAI in the main store (MAS) module 0. Operation of the ABB1 is described in Part 7.

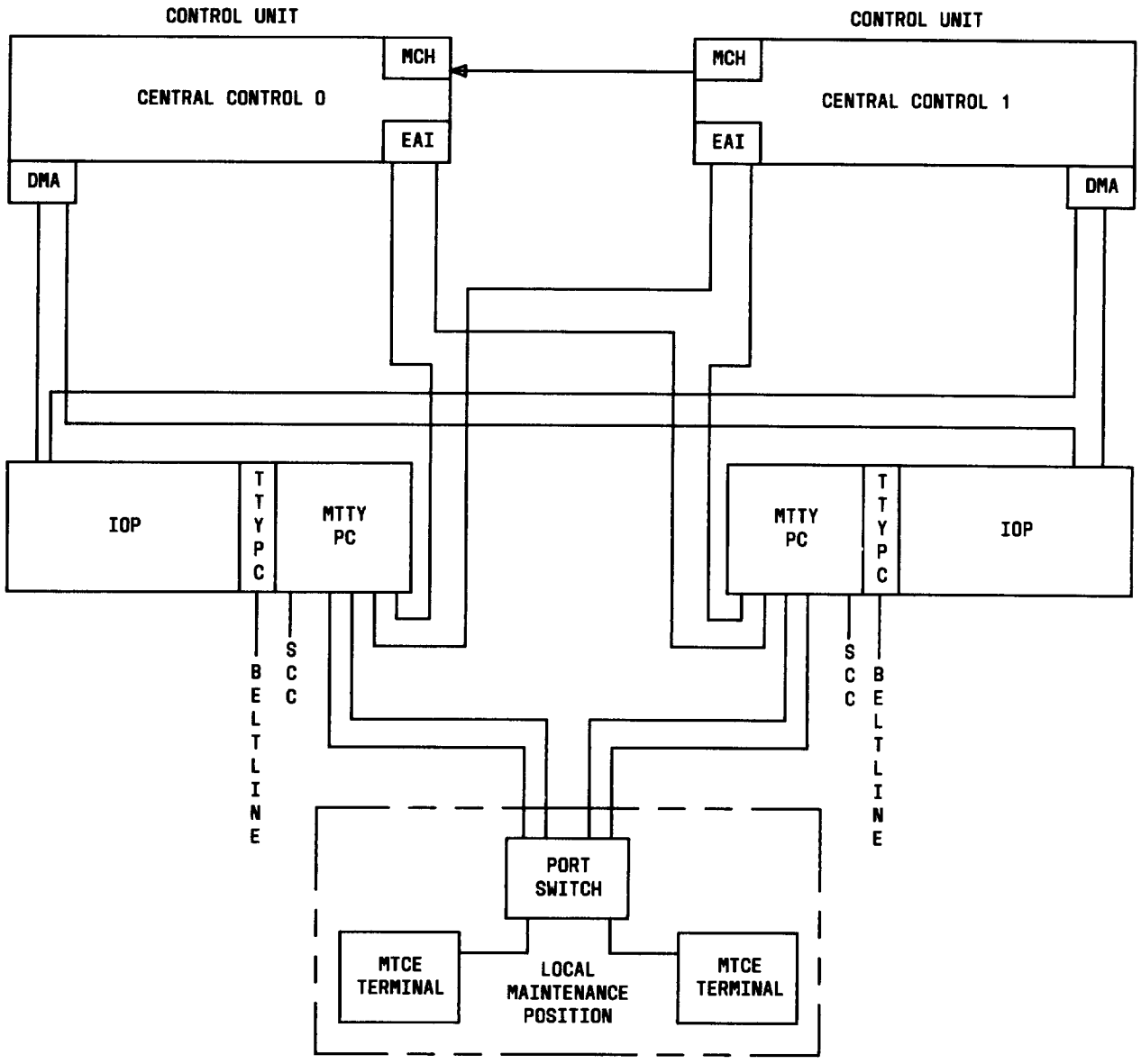
## 7. POWER

**7.01** The CC requires +5 volts for operation. The +5 volts is supplied by dc-to-dc converters located in the power unit at the lower portion of the CU frame. Two to four (maximum) 244D converters (−48 volts to +5 volts) may be used dependent upon the application. The +5 volts dc output from the 244D converters are paralleled together (to increase current capability) and distributed to the ED-4C181-30 fuse unit for distribution to the CC circuit packs. Refer to Fig. 34 for a block diagram of the power unit and distribution.

### A. ABB1 Power Switch

**7.02** The ABB1 power switch (Fig. 35) provides the control (application and removal) for the frame power (Fig. 1). The ABB1 is located in the main store module 0 unit in the CU frame (Fig. 1).

**7.03** Five indicator lights on the front of the switch indicate the state of the unit being controlled. These lights are OFF, ALM (alarm), OOS (out of ser-




LEGEND:

| | |
|---|---|
| DMA - DIRECT MEMORY ACCESS | MTCE - MAINTENANCE |
| EAI - EMERGENCY ACTION INTERFACE | MTTY - MAINTENANCE TELETYPEWRITER |
| IOP - INPUT/OUTPUT PROCESSOR | PC - PERIPHERAL CONTROLLER |
| MCH - MAINTENANCE CHANNEL | SCC - SWITCHING CONTROL CENTER |

**Fig. 33—Local Maintenance Position Interfaces**

vice), RQIP (request in progress), and ROS (request out of service).

**7.04** Five pushbutton switches provide power control. These are located on the front panel and are ON, OFF, ROS/RST (request out of service/request restore), ACO/T (lamp test/retire major alarm), and MOR (manual override).

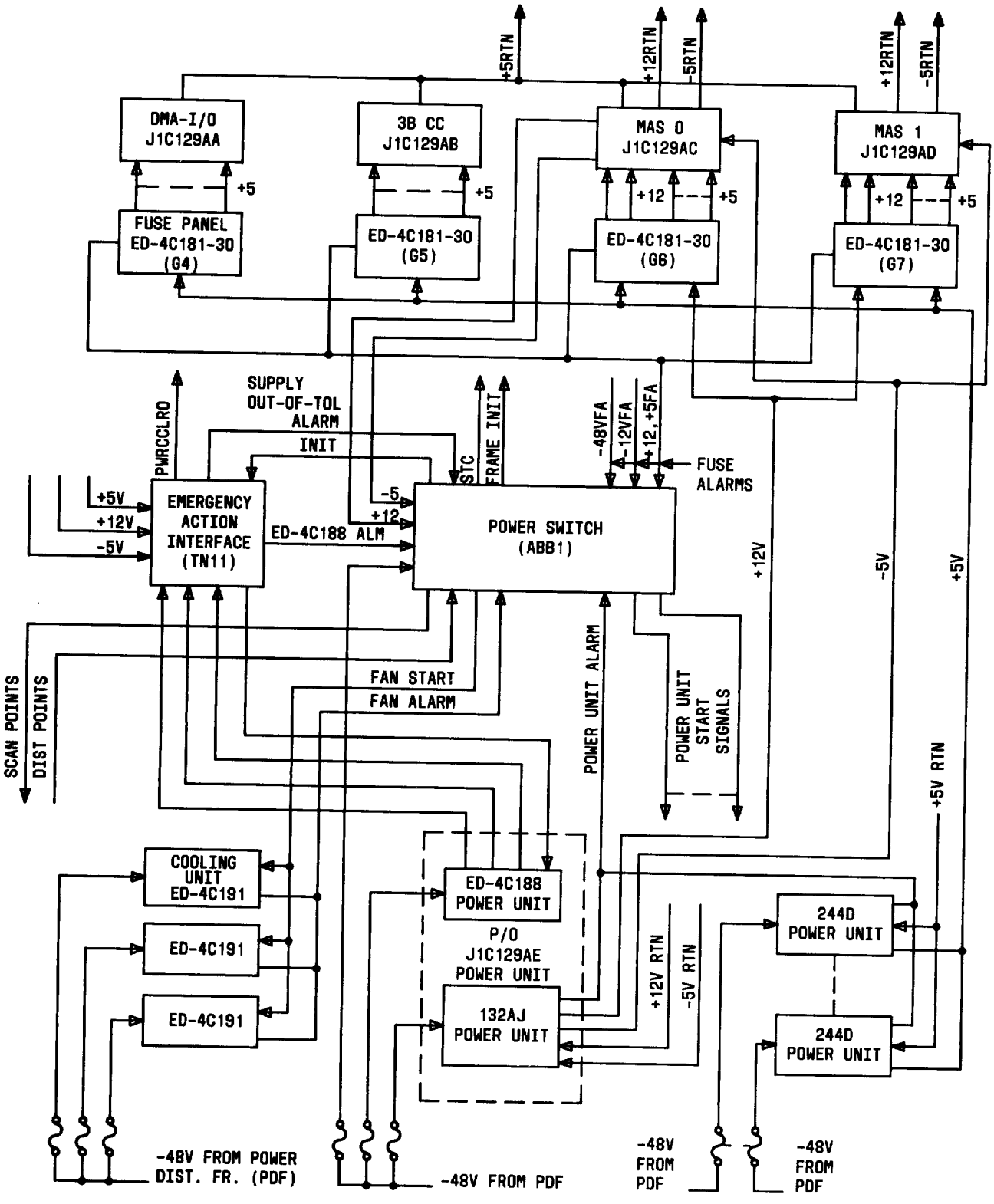



Fig. 34—Control Unit Frame Power Distribution, Control, and Alarms

**7.05** Input voltage requirements are −48 volts from the office supply and +5 volts from the 244D dc-to-dc converters controlled by the switch.

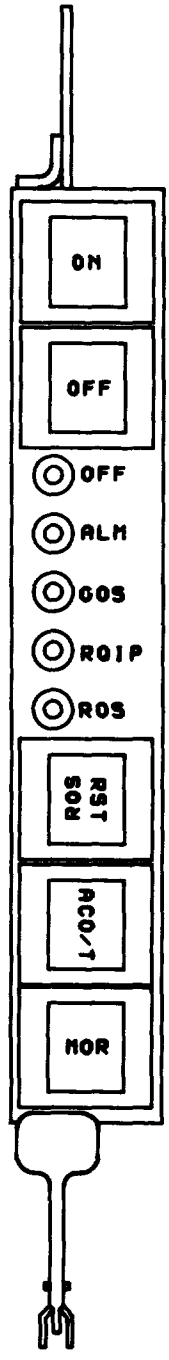



**Fig. 35—ABB1 Power Switch (Front Panel)**

**7.06** Three-phased start signals are provided by the ABB1 to control the application sequence of logic power, memory power, and I/O bus power. The following functions may be performed by the craft:

(a) Sequentially supply or remove power

(b) Initiate system request to remove from service or restore to service the associated unit

(c) Test indicator lights on the power switch

(d) Retire a major office alarm generated at the associated unit.

### B. ABB1 Controls

**7.07** The ON, OFF, and MOR are momentary push-button switches and ROS/RST and ACO/T are 2-position latching switches. The five indicators are 549-type LEDs (light emitting diodes). A description of these controls is given below:

(a) ***ON Switch:*** Momentarily depressing the ON switch when the ACO/T switch is not in its retire alarm state initiates the power-up sequence. Depressing the ON switch when the ACO/T switch is in its retire-alarm state or when frame power is up causes no change in the state of the power switch.

(b) ***OFF Switch:*** Momentarily depressing the OFF switch when the unit is in its out-of-service state initiates the power-down sequence. Depressing the OFF switch when the unit is in service or when power is off causes no change in the state of the power switch.

(c) ***ROS/RST Switch:*** Depressing the ROS/RST switch to the ROS position requests the unit be taken out of service via the scan point SCX and lights the ROS LED. Depressing the ROS/RST switch to the RST position requests that the unit be restored to service.

(d) ***ACO/T Switch:*** Depressing the ACO/T switch to the retire-alarm state tests all lamps on the power switch, provides an external lamp test signal, silences the office major alarm originated at the power switch, and extinguishes the ALM LED, if lit.

(e) ***MOR Switch:*** Simultaneously depressing the OFF and MOR (manual override) switches defeats the interlock between the OFF switch and unit out-of-service state and initiates the power-down sequence.

(f) ***OFF LED:*** The red OFF lamp is lit when the unit is in its power-off state and is extinguished when the unit is in its power-up state.

(g) ***ALM LED:*** The red ALM lamp lights to indicate the presence of power-related faults.

(h) ***OOS LED:*** The yellow OOS (out-of-service) lamp is system-activated via the OOS signal distribute point when the unit is marked out of service.

(i) ***RQIP LED:*** The green RQIP (request in progress) lamp lights to indicate the system has received a request to take the unit out of service or restore it to service. This lamp, which is system-activated via the RQIP signal distribute point, flashes to indicate the request has been denied.

(j) ***ROS LED:*** The green ROS (request out of service) service lamp is lit when the ROS/RST switch is in its ROS state.

### C. Scan, Alarm, and Signal Distribute Points

**7.08** Three scan points (SCX, SCY, and BPP), two alarm points (MJ and PA), and two signal distribute points (OOS and RQIP) are provided. Each scan and alarm point consists of an isolated metallic contact. The active "1" state appears as a resistance of less than 200 megaohms. The inactive "0" state appears as an open circuit. Each distribute point consists of an optoisolator input diode.

**7.09** ***Scan and Alarm Points:*** Table J summarizes the scan and alarm point states. On automatic power off, the MJ scan point closes and remains closed until the ACO/T switch is depressed. When power is left up in the presence of a fault, the MJ scan point remains closed until either the fault is removed (PA scan point also goes inactive) or the ACO/T switch is depressed (PA scan point remains active).

**7.10** ***Signal Distribute Points:*** The active "1" state of the RQIP signal distribute point indicates a system software acknowledgment that a request for removal from service or restoral to service of the associated unit has been made. If the request is granted, the RQIP SD point will become inactive ("0" state). If it is denied, the RQIP SD point will intermittently flash under system control. The OOS signal distribute point becomes active when the unit has been taken out of service. The RQIP and OOS indicator lamps provide a visual indication of the state

**TABLE J**

**SCAN AND ALARM POINTS**

| CONDITION | BPP | SCX | SCY | MJ | PA |
|---|---|---|---|---|---|
| Normal in service | 1 | 0 | 0 | 0 | 0 |
| Request out of servie | 1 | 1 | 0 | 0 | 0 |
| Manual power off | 0 | 1 | 1 | 0 | 0 |
| Automatic power off | 0 | 1 | 1 | 1 | 0 |
| Power up with major fault present | 1/0 | 0 | 1 | 1 | 1 |
| Power up with minor fault present (such as supply out of tolerance) | 1 | 0 | 1 | 0 | 1 |

of the RQIP and OOS signal distribute points, respectively. Table K summarizes the signal distribute point states.

### D. Power-Up Sequence

**7.11** Power-up is initiated by momentarily depressing the power ON switch. Control circuitry ensures that frame power is supplied in the proper sequence via three start signals and an initialization phase.

**7.12** In the power-off state, the initialization circuit which starts the power-up sequence is powered from a fused −48V source through a normally open ON switch (momentary contact). Initialization signals enable +5V frame converters which power transistor-transistor logic (TTL) sequence circuitry and initialize power control and alarm circuitry. This allows for the power switch to be inserted with −48V present without the danger enabling frame converters until the power ON switch is depressed.

**7.13** The start signals STA and STB are optoisolator outputs used to control converters associated with the power switch. The start signal STC is an open-collector relay driver which enables I/O bus power. On power-up, logic power precedes memory power by approximately 1300 ms, and memory power precedes the start signal by approximately 500 ms. The power-off LED is lit in the power-off state and does not extinguish until the power-up sequence is complete.

### E. Power-Down Sequence

**7.14** ***Normal Power Down:*** To prevent the inadvertent removal of frame power, the OFF switch (momentary contact) is interlocked with a system-granted out-of-service (OOS) signal. Depressing the power OFF switch causes no change in state of the circuit pack unless the OOS signal distribute point is active, in which case frame power is sequentially removed. On power down, the initialization signals and I/O bus power are removed approximately 500 ms before memory power is removed. The start signal is removed approximately 500 ms before logic power is removed. The power-off LED remains extinguished until the power-down sequence is complete.

**7.15** ***Emergency Power Down:*** Simultaneously depressing the power OFF and MOR (manual override) switches sequentially removes frame power, providing an emergency manual power control if needed. Additionally, a separate backup timing circuit (approximately 1500 ms) will release the +5V power to the circuit pack, all start signals, and hence, all frame power controlled by the power switch will release. Power should never be removed without consulting the maintenance document (TOP 254-301-811). Also, power should be removed before replacing any circuit pack or unit in the CU frame.

**TABLE K**

**SIGNAL DISTRIBUTE POINTS**

| CONDITION | RQIP | OOS |
|---|---|---|
| Normal in service | 0 | 0 |
| Removal from service or restoral to service requested with disposition pending | 1 | 0 |
| Request denied | FLASH | 0 |
| Diagnostic failure after a restore-to-service request | FLASH | 0 |
| System grants out-of-service-request | 0 | 1 |

## 8. GLOSSARY

**8.01** The following terms are defined to aid in using and understanding this section:

***Cache Store Unit (CSU)***—An optional system unit which provides a high-speed, random access, memory buffer used to reduce the apparent main store access and cycle times.

***CCIO Bus***—A group of leads used to interconnect the CC with its I/O channels and DMA unit. This bus is used to carry data, address, and control information.

***Central Control (CC)***—Control section of a 3B 20 computer. It consists of control circuitry, arithmetic and logic unit (ALU), rotate mask unit (RMU), general and special registers, maintenance channel, store address translator circuit, and optional cache memory.

***Control Unit (CU)***—That part of a 3B 20 computer which is switched on- or off-line as a unit. It consists of a CC, main store, DMA unit (when equipped), I/O channels, and the power unit.

***Data Manipulation Unit (DMU)***—This circuit contains the rotate mask unit, logic and arithmetic functions of the CC.

***Direct Memory Access Unit (DMA)***—This optional system component provides main store access control for I/O devices and other computers without requiring direct control from the CC.

***DMA Controller***—Control circuitry for the DMA.

***DMA I/O Bus***—A group of leads which interconnect the DMA controller with four DMA channel controllers.

***Dual Serial Channel (DSCH)***—Provides a 16- or 32-bit ac interface to I/O devices, and uses two serial signal paths in each direction.

***Duplex Computer***—A duplex computer consists of duplicated control units interconnected via the maintenance channel, and all peripheral hardware required for system operation. The peripheral units may include a system status panel, TTY controller, tape date controller, and moving head disk units.

***Duplex Dual Serial Bus Selector (DDSBS)***—A system component which provides the capability to connect two DSCHs from duplex CUs to a single peripheral device.

***Emergency Action Interface (EAI)***—System component which provides manual emergency control over the CU.

***Gating Bus***—A multilead bus (usually 36 leads) which carries data between the various sections of the CC.

***Halfword***—A halfword consists of two bytes.

***Initialization***—An action taken to provide the system with a known good and operating configuration.

***Insulated Gate Field Effect Transistor (IGFET)***—The storage element for one bit of data.

***I/O Processor (IOP)***—The IOP is used as a front-end processor permitting the connection of a variety of peripheral devices (terminals, magnetic tape unit, moving head disk drive, etc) to the 3B 20 computer. The IOP contains a peripheral interface controller (PIC), an IOP interface, and two microprocessor communities. Each microprocessor community connects to a number of data terminals.

***Main Store (MAS)***—The MAS unit of the computer provides the storage for program instructions and data. It utilizes insulated gate field effect transistor (IGFET) memory devices and initially will provide up to 4 megabytes of storage. It consists of an optional main store update circuit and one or two main store modules.

***Main Store Array (MASA)***—The MASA circuit provides the storage array and will initially provide 32K (K1024) words by 36 bits of storage including parity.

***Main Store Bus (MASB)***—The MASB provides address, data, and control paths needed for main store access and connects to the main store, the central control, main store update, and direct memory access.

***Main Store Controller (MASC)***—The MASC circuit provides the control interface between the main store bus and memory devices.

***Main Store Module (MASM)***—The MASM unit consists of one main store controller and 1 to 16 microstore address buses (MSAs) initially providing a maximum of 2 megabytes of storage.

***Main Store Update Bus (MASUB)***—The MASUB is an address/data control bus that interconnects the main store update circuits of each control unit in a duplex computer configuration.

***Main Store Update Unit (MASU)***—The MASU circuit interfaces the main store buses of duplex control units to permit reading, writing, and updating of each main store by each control unit. It arbitrates the use of the main store bus between the central control, direct memory access (DMA), and duplex control unit. It is required in duplex systems and any simplex systems with DMA.

***Maintenance Bus (MTCB)***—The MTCB is a 36-lead bus that permits the maintenance channel and microlevel test set/virtual panel to examine important status registers independent of the state of the CC.

***Maintenance Channel (MCH)***—The MCH is a CC component that provides serial access at a microinstruction level to and/or from a duplex or master/slave control.

***Maintenance Reset Function (MRF)***—An MRF is a hardware signal within a CC which leads to a system initialization. A CC within which an MRF is occurring is attempting to become the active CC; ie, to go on-line.

***Mask***—A bit pattern used to enable or disable specific bits.

***Microcontrol (MC)***—The MC portion of the CC controls the sequencing of the microstore (MIS) and decoding of the microinstructions, providing control signals to the CC circuitry.

***Microinstruction***—A fixed, read-only instruction which is used to form microinstruction sequences that are permanently stored in a read-only memory. The microinstruction sequences are used to implement the CC instruction set and basic control functions.

***Microlevel Test Set (MLTS)***—The MLTS is an optional test component that interfaces to the CC at the microlevel and is used to manually test a CC. The MLTS uses the same circuit pack position as the virtual panel.

***Microprocessor***—A small, low-priced, special purpose processor used to perform a specific function, eg, a peripheral controller.

***Microstore (MIS)***—The MIS is a system component that provides the memory and related interface circuitry containing the microprogram used for instruction, decoding, and overall control of the CC.

***Microstore Address Bus (MSA)***—The MSA is an 18-lead address bus that transmits the address of the desired microinstruction to the MIS.

***Microstore Data Bus (MSD)***—The MSD is a 64-lead data bus that transmits the microinstruction to the microcontrol.

***Multiplex***—The concurrent transmission of more than one information (data) stream over a single channel.

***Nibble***—Four data bits.

***Off-Line*** —A CU is off-line when it is not in the active state and is not controlling the system. The off-line CU is the unit which is not in active control of the system configuration and execution, but which may be active (executing diagnostics, for example) when performing off-line functions.

***On-Line***—A CU is on-line if it is in the active state and capable of executing code. More specifically, for a duplex computer, the on-line CU is in active control of the system configuration and execution; the mate CU, the off-line CU, may be active (executing diagnostics) but it is not in control.

***Physical Address***—Hardware memory address.

***Refresh***—The act of restoring the stored data bit in an insulated gate field effect transistor memory cell.

***Serial Channel (SCH)***—The SCH component provides a 16- or 32-bit serial data interface utilizing two serial signal paths (one in each direction). Input/output messages are effectively round-trip in nature.

***Simplex Computer***—A computer installation which is equipped with only one CU.

***Special Registers***—A group of registers in the CC which perform unique functions within the CC. These registers contain specialized data concerning system operation and are not usually available for external use. Special registers include the channel data register, bidirectional gating register, pulse point register, interrupt set register, interrupt mask register, timer, program status register, hardware status register, system status register, error register, and related circuitry.

***Stop-and-Switch***—A hardware signal generated by error detection circuits within the on-line CC which causes the on-line CU to be halted and the off-line CU to be switched to on-line.

***Store Address Translator (SAT)***—The SAT component provides the virtual address to physical address translation circuitry as well as the main store address protection circuitry. The store address register (SAR), program address (PAR), and related circuitry are included in this circuit.

***Store Data Control (SDC)***—The SDC circuit provides the store data and instruction interface function including the store data register (SDR), store instruction register (SIR), instruction buffer (IB), and related circuitry.

***Utility Circuit (UC)***—The UC is an optional circuit that provides address and data matchers and transfer-trace capability to the CC and is used for software debugging.

***Word***—A CC word consisting of 32 bits (plus 4 parity bits) which corresponds to the width of the data paths within the CC, and to the width of most registers within the CC. A word is divided into four 8-bit bytes with a parity bit associated with each byte.