Section 7

ALU, MCP, MAS, AND I/O CIRCUIT BOARDS THEORY OF OPERATION

INTRODUCTION

The ALU Board, MCP Board, and MAS Board are the processor and memory for the 4052/4052A and 4054/4054A Desktop Computers. This section covers the theory of operation and general timing of these three circuit boards. Although the 4052, 4052A, 4054, and 4054A all use different I/O Boards, their operation is quite similar. For details as to operation of the I/O Board, see Section 8, Keyboard Theory of Operation, Section 9, Tape Unit Theory of Operation, and Section 10, GPIB Theory of Operation.

This manual covers the different boards used in the 4052 and 4054 and the 4052A and 4054A Desktop Computers. Each board is commented so you will be able to identify the correct board for your computer system.

You should refer to the following in order to fully understand and use this section:

- o Section 6 and Appendix A of this manual
- o 4052/4052A Parts and Schematics Service Manual
- o 4054/4054A Parts and Schematics Service Manual

ALU BOARD

Two major versions of the ALU Board are used. They differ only in the clock circuitry. The latest version of the ALU Board uses a modular clock, while the earlier version uses a clock built of discrete components.

The ALU Board (Figure 7-1) contains the arithmetic logical unit, firmware instruction decode-execution, interrupt control, and system registers (except for the Program Counter which resides on the MCP Board). The ALU Board interfaces to the MCP Board and is connected to the interrupt lines from the rest of the computer system. The ALU Board contains the following functional units:

- Instruction Register
- Fetch Control
- Wait Control
- Memory Controller Handshake
- Clocks
- Sequence Instruction Branch
- Sequence Micro ROM Branch
- Interrupt Generator
- Micro Sequencer Branch Control
- RALU
- RALU Constant Load
- Shift Control
- Condition Code Set-up and Register
- Micro ROM
- Micro ROM Sequencer
- Pipeline Register

Instruction Register

The Instruction Register is a clocked latch used to store the firmware instruction opcode. The opcode is latched into the register at the rise of FRCLK-1 from the MCP Board. The opcode is supplied by the FRA0 thru FRA7 lines (MB8 thru MB15 are the same lines). The output of the Instruction Register is used by the Fetch Control ROM and the Sequence Instruction Branch.

Fetch Control

The Fetch Control is a ROM addressed by the output of the Instruction Register. The ROM decodes the firmware instruction opcode for use by the MCP Board during the instruction execution. The WONFI-0 and WONFO-0 are used by the Wait Control to delay the ALU Board operation during certain firmware instructions.

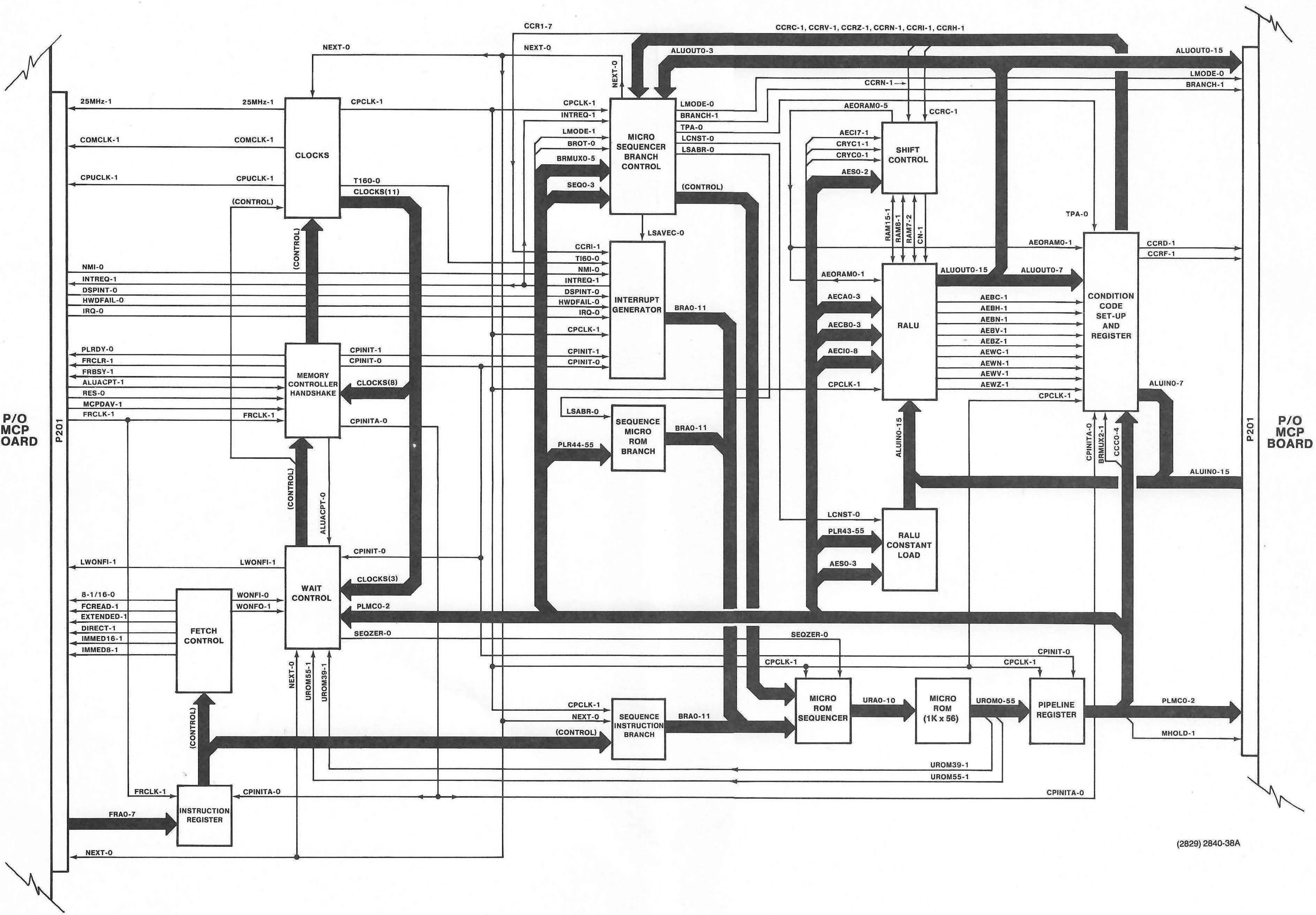


Figure 7-1. ALU Block Diagram.

Sequence Instruction Branch

The Sequence Instruction Branch is a tri-state buffer used as a source of a Micro ROM address for the Micro ROM Sequencer. The Micro ROM starting address for a firmware instruction is X '3XX', where "XX" is the hexadecimal value of the firmware opcode. The NEXT-0 control line from the Micro Sequencer Branch Control enables output of the buffer when the firmware instruction is to be executed.

Sequence Micro ROM Branch

The Sequence Micro ROM Branch is a tri-state buffer used as a source of a Micro ROM address for the Micro ROM Sequencer. The address comes from the Pipeline Register (latched Micro ROM output). This allows the Micro ROM to supply the next Micro ROM address. The LSABR-0 control line from the Micro Sequence Branch Control enables the output of the buffer when the Micro ROM branch is taken.

Interrupt Generator

The Interrupt Generator uses four interrupt input lines. The order of interrupt line priority from highest to lowest is HWDFAIL-0, NMI-0, IRQ-0, AND DSPINT-0. The HWDFAIL-0 and NMI-0 interrupt lines are latched if there is a transition of a high to a low state. The interrupt is not lost during execution of a firmware instruction. IRQ-0 and DSPINT-0 are lost if they are not low when CPCLK-1 goes high. IRQ-0 and DSPINT-0 are maskable interrupts. When the Condition Code Register bit CCRI-1 is high, IRQ-0 and DSPINT-0 are disabled. They can not produce an interrupt until CCRI-1 goes low.

CPCLK-1 goes high at the start of each micro instruction cycle latching all active interrupts into the storage latch (U62). If an interrupt was latched, INTREQ-1 goes high. INTREQ-1 is used by the Micro Sequence Branch Control for interrupt branching at the end of a firmware instruction. INTREQ-1 is also used to prevent the MCP Board from fetching the next firmware instruction.

Interrupts are honored at the end of each firmware instruction. If there is no interrupt, the MCP Board fetches the next firmware instruction to be executed. If there is an interrupt, the Interrupt Generator supplies a Micro ROM address to the Micro ROM Sequencer. The Micro ROM address is enabled by LSAVEC-0 from the Micro Sequence Branch Control.

Micro Sequence Branch Control

The Micro Sequence Branch Control supplies control lines for the Micro ROM Sequencer, selects the active inputs for conditional branching, controls which micro code address is active to the Micro ROM Sequencer, and controls the Micro ROM constant to the RALU.

The Micro Sequence Branch Control consists of a ROM (U235), a counter (U175), two data selectors (U100 and U108), and a decoder (U185). The ROM is addressed by Field 7 Micro ROM Sequence control (SEQ0 thru SEQ3) and by the branch true line (BRANCH-1). DO1 thru DO4 of the ROM controls the Micro ROM Sequencer. DO5 of the ROM controls the loading of data from the RALU (ALUOUT0 thru ALUOUT3) into the counter. The counter is used for Micro ROM instruction looping. DO6 of the ROM controls when the counter increments. DO7 and DO8 of the ROM addresses one-half of the decoder. The output of the decoder determines where the next Micro ROM branch address is from: Sequence Instruction Branch (NEXT-0); Sequence Micro ROM Branch (LSABR-0); or Interrupt Generator (LSAVEC-0). The other half of the decoder is addressed by BRMUX0-1 and BRMUX1-1 from the Micro ROM instruction and enabled by the inverse of LMODE-1. The output of the decoder is used for the loading (LCNST-0) of a constant from the Micro ROM or transferring (TPA-0) the Condition Code Register to the RALU. The two data selectors are addressed by BRMUX0-1 thru BRMUX5-0 from the Micro ROM instruction. The data lines supply the conditions to be selected. The outputs of the two data selectors are OR-ed together (U110A) and EX-OR-ed (U165B) with BROT-0 from the Micro ROM instruction. When BROT-0 is low, the selected condition is not inverted. When BROT-0 is high, the selected condition is inverted.

Micro ROM Sequencer

The Micro ROM Sequencer addresses to the Micro ROM. On power up, the SEQZER-0 line goes low. The Micro ROM Sequencer outputs an address of all zeros. The Micro ROM starts a system check from this address. During the power up check, the Micro ROM address is supplied by either the sequencers internal counter or by the BRA0 thru BRA11 lines from the Sequence Micro ROM Branch. When the power up sequence is finished, the Micro ROM Sequencer also gets addresses from the Sequence Instruction Branch (firmware instruction execution) and the Interrupt Generator (system interrupt acknowledge).

Micro ROM

The Micro ROM contains all the micro instructions for power up and firmware instruction execution. The Micro ROM contains 1K words of 56 bits per word. For explanation of the Micro ROM output refer to the Micro Code part of Section 6 of this manual.

The Micro ROM address is supplied by the Micro ROM Sequencer. The output of the Micro ROM is latched into the Pipeline Register. Two output lines UROM39-1 and UROM55-1 are used by the Wait Control.

Pipeline Register

The Pipeline Register stores the Micro ROM data while it is being used by the processor. Storage of the micro instruction allows the Micro ROM to be addressed for the next Micro ROM output before the present micro instruction is finished.

At reset the SEQ0 thru SEQ3 section of the Pipeline Register is cleared by CPINIT-0 to ensure no false control of the Micro ROM Sequencer. The Micro ROM data is latched on the rise of CPCLK-1 during each micro instruction cycle.

RALU Constant Load

The RALU Constant Load is a buffer that allows the transfer of 16 bits of data from the Micro ROM instruction to the ALUIN0 thru ALUIN15 lines. The buffers are enabled when LCNST-0 is low. The data comes from the Pipeline Register lines (AES0 thru AES2 and PLR43 thru PLR55).

RALU

The RALU consists of four 2901-type, 4-bit slices and a carry look-ahead generator. The RALU Block Diagram (Figure 7-2) shows the flow of data within the 2901s and the relationship of the RALU to the Shift Control and the Condition Code Set-up and Register. The control of the RALU is from the Micro ROM instruction (AECI0 thru AECI8, AECA0 thru AECA3, and AECB0 thru AECB3). Refer to the micro code description in Section 6 for the internal operation control.

Data to the RALU is on the ALUIN0 thru ALUIN15 lines and data out is on the ALUOUT0 thru ALUOUT15 lines. Both 8-bit and 16-bit conditions are supplied to the Condition Code Set-up and Register.

Shift Control

The Shift Control is controlled by AECI7, CRYC0 thru CRYC1 and AES0 thru AES2 from the Micro ROM instruction. The Shift Control selects the lines that are active during shifts. The Shift Control selects the carry input to the RALU ("0", "1", CCRC-1, and CCRC-1 inverted).

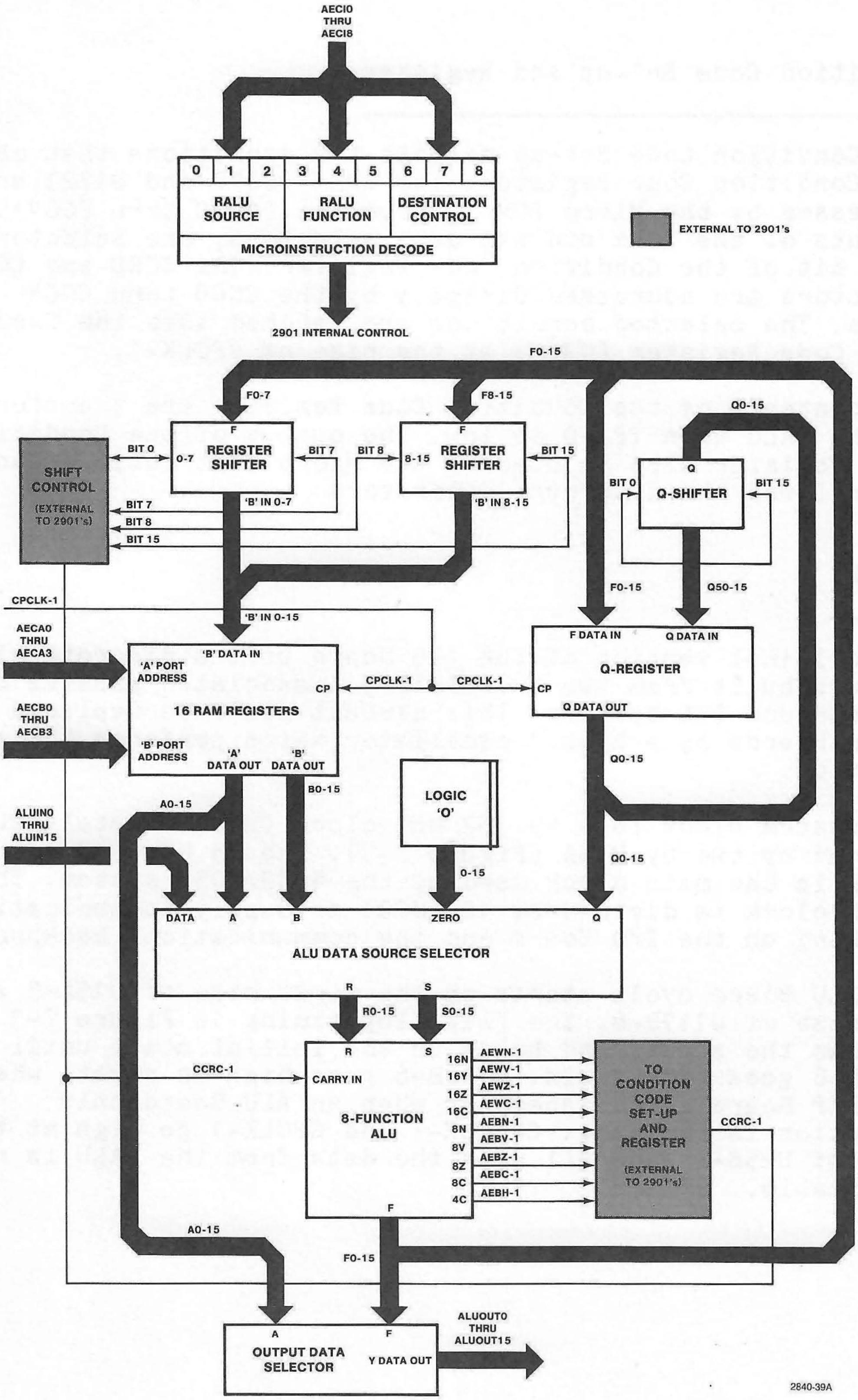


Figure 7-2. RALU Block Diagram.

Condition Code Set-up and Register

The Condition Code Set-up selects the conditions that change the Condition Code Register. Two ROMs (U170 and U172) are addressed by the Micro ROM instruction (CCCO thru CCC4). The outputs of the ROMs address data selectors, one selector for each bit of the Condition Code Register. The CCRD and CCRF selectors are addressed directly by the CCCO thru CCC4 lines. The selected conditions are latched into the Condition Code Register (U385) at the rise of CPCLK-1.

The contents of the Condition Code Register are transferred to the RALU when TPA-0 is low. The output of the Condition Code Register also is used by the Micro Instruction Branch Control and the Interrupt Generator.

Clocks

The original version of the ALU Board uses a discrete clock circuit built from two transistors, associated passive components and TTL buffers. This circuit has been replaced on later boards by a hybrid oscillator which performs the same function.

The system clock is a 49.152 MHz clock (approximately 20 ns) divided by two by U15A (Figure 7-3). The 25 MHz (40 ns) clock is the main clock used by the 4052/4054 system. The 25MHz clock is divided by 10 (U20) to supply communications clocking on the I/O Board and the communications backpack.

The ALU Board cycle starts on the first rise of U15A-5 after the rise of U117B-6. The flip-flop timing in Figure 7-3 follows the states and holds in the initial state until U117B-6 goes high again. U117B-6 goes high on reset, when the MCP Board is finished, or when an ALU Board only operation is finished. CPUCLK-1 and CPCLK-1 go high at the fall of U25B-9 and fall when the data from the RALU is ready and stable.

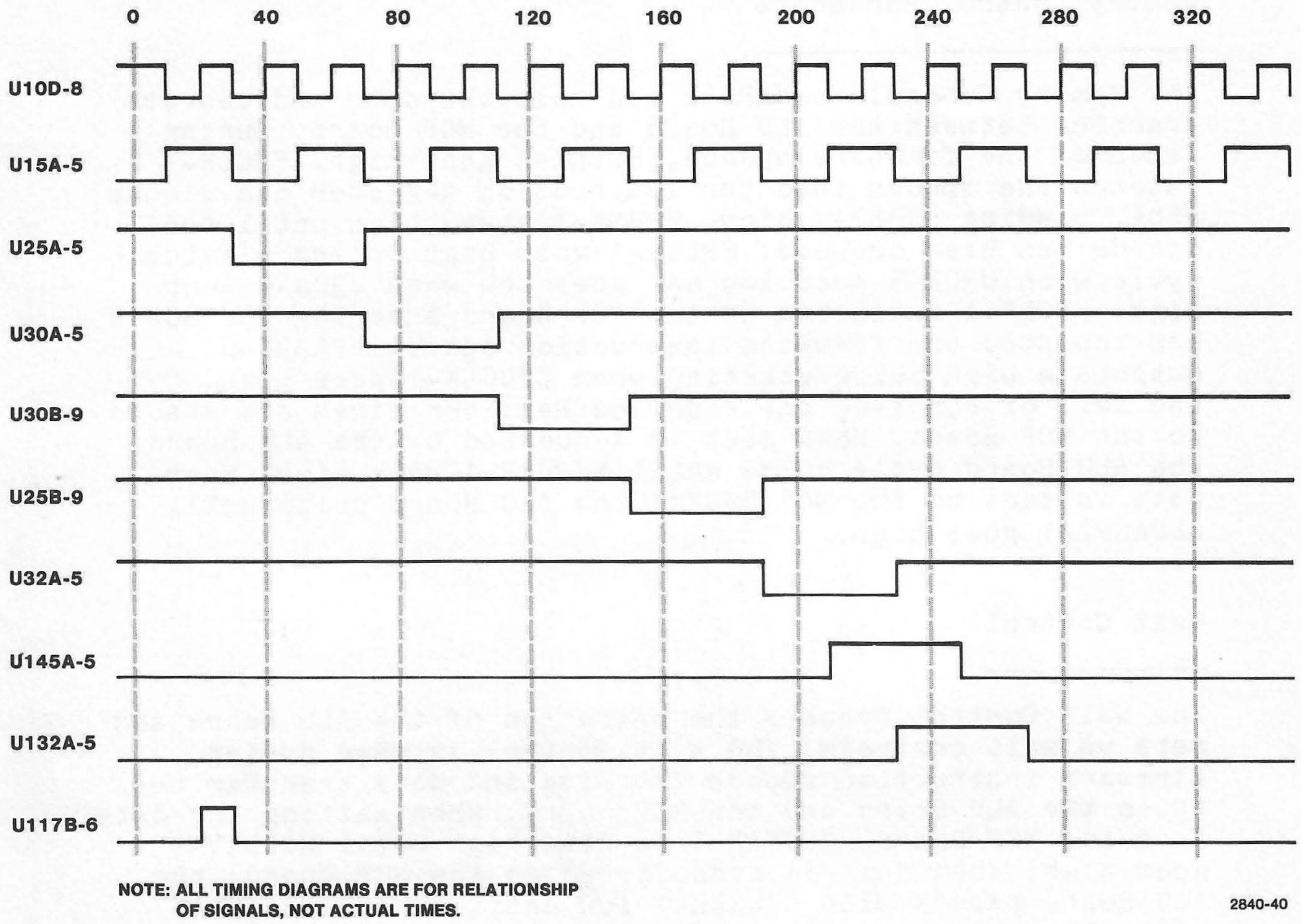


Figure 7-3. ALU Timing-Clocks.

Memory Control Handshake

The Memory Control Handshake controls the data and address transfer between the ALU Board and the MCP Board. During a fetch of the firmware opcode, FRCLK-1 goes high. FRCLK-1 latches the opcode into the Instruction Register and clocks U60A, sending FRBSY-1 high. FRBSY-1 stays high until the opcode has been decoded. FRCLR-1 goes high during a fetch cycle when U30A-5 goes low and goes low when U30A-5 goes high. FRCLR-1 indicates to the MCP Board that the ALU Board has accepted the firmware instruction opcode. PLRDY-0 outputs a high pulse starting when CPUCLK-1 goes high. On the fall of PLRDY-0, the Pipeline Register lines are stable to the MCP Board. When data is requested by the ALU Board, the ALU Board cycle holds until MCPDAV-1 goes high. When data is sent to the MCP Board, the ALU Board holds until ALUACPT-1 goes high.

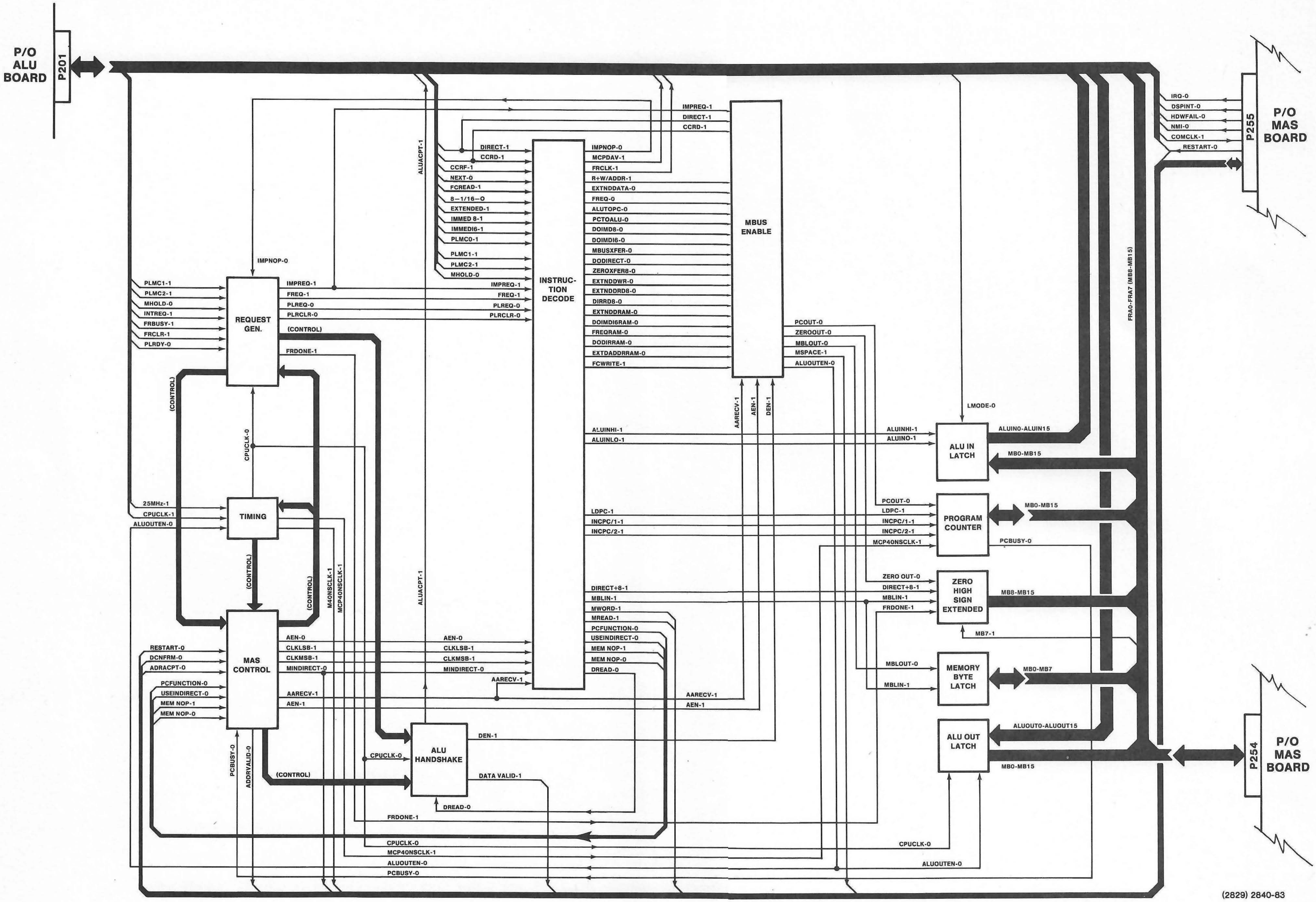
Wait Control

The Wait Control decodes the operation of the ALU Board and sets up wait controls. The Wait Control is used during firmware instruction opcode fetching and data transfer between the ALU Board and the MCP Board. When waiting for data from the MCP Board, CPUCLK-1 is held high until MCPDAV-1 goes high. When data is transferred to the MCP Board, the ALU Board pauses with CPUCLK-1 low until ALUACPT-1 goes high. The Wait Control supplies the SEQZER-0 signal to start the power up sequence of the micro code.

MCP BOARD

The MCP Board (Figure 7-4) controls the data and address between the ALU Board and the MAS Board. The MCP Board contains the Program Counter used during execution of firmware instructions. The MCP Board contains the following blocks:

- Aluin Latch
- Aluout Latch



(2829) 2840-83

Figure 7-4. MCP Block Diagram.

- Memory Byte Latch
- High Zero/Sign Extended
- Program Counter
- Request Generator
- MAS Control
- Timing
- ALU Handshake
- Instruction Decode
- Mbus Enable

Aluin Latch

The Aluin Latch consists of 16 D-type latches (U451 and U551) used to hold the data from the MB0 thru MB15 lines. The data from the Aluin Latch to the ALU Board is on the ALUIN0 thru ALUIN15 lines. The data is latched when ALUINHI-1 and ALUINLO-1 goes high. ALUINHI-1 clocks the upper byte (MB8 thru MB15) into the latch (U451). ALUINLO-1 clocks the low byte (MB0 thru MB7) into the latch (U551). The data output is controlled by LMODE-0 from the ALU Board micro instruction. When LMODE-0 is high the Aluin Latch is active on the ALUIN0 thru ALUIN15 lines.

Aluout Latch

The Aluout Latch consists of 16 D-type latches (U441 and U541) used to hold the data from the ALUOUT0 thru ALUOUT15 lines. The Aluout Latch output is to the MB0 thru MB15 lines. The data is latched when CPUCLK-0 goes high. The data is enabled to the MB0 thru MB15 lines when ALUOUT-0 is low.

Memory Byte Latch

The Memory Byte Latch is used during firmware instruction opcode fetch operations. The lower byte (MB0 thru MB7) is stored in the latch (U515) and the upper byte (MB8 thru MB15) is stored in the Instruction Register on the ALU Board. MB8 thru MB15 are the same lines as FRA0 thru FRA7. The data is stored in the latch when MBLIN-1 goes high. The data is enabled to the MB0 thru MB7 lines when MBLOUT-0 is low.

High Zero/Sign Extended

The High Zero/Sign Extended has two functions: it supplies a high byte of all zeros or all ones during +/- offset addressing (relative), and it supplies a high byte of all zeros during one-byte data transfer.

During a fetch operation the upper bit (MB7-1) stored in the Memory Byte Latch is clocked into a D-type latch (U681A) by MBLIN-1 going high. The D-type latch is cleared if the data in the Memory Byte Latch is to be used for 8-bit immediate, a direct address, or 8-bit data from the MAS Board to the ALU Board. The output of the D-type latch (U681A-5) is supplied to all eight inputs of a buffer (U415). The buffer output is enabled by ZEROOUT-0.

Program Counter

The Program Counter consists of a 16-bit counter (U425, U435, U525, and U535), output buffer (U421 and U521), and load-increment timing logic. The Program Counter supplies the firmware instruction address.

To load the Program Counter, LDPC-1 goes high, clocking a D-type filp-flop (U635A). The Q-not output of the flip-flop (U635A-6) enables the load line to the counter. The delay produced by the next D-type flip-flop (U635B) allows the data and load lines to settle. The last D-type flip-flop (U625A) is used to clock the counter and clear U635A.

The Program Counter is incremented by one or by two. The same configuration of flip-flops used for loading the counter is used to increment the counter. The increment-by-two flip-flop set has the last two flip-flops repeated to get the second increment pulse. The increment-by- one flip-flops are started by the INCPC/1-1 line. The increment-by-two flip-flops are started by the INCPC/2-1 line. At the time the counter is being clocked the PCBUSY-0 line is low.

The load data to the counter is from the MB0 thru MB15 lines. The counter output to the MB0 thru MB15 lines is enabled by the PCOUT-0 line to the buffers (U421 and U521).

Request Generator

The Request Generator generates three types of requests: a fetch request (FREQ-1 is high); an implied request (IMPREQ-1 is high); and pipeline request (PLREQ-0 is low). A fetch request is when a firmware instruction opcode is being fetched from the MAS Board or the backpacks. An implied request is used for two and three-byte firmware instructions. An implied request follows the firmware instruction fetch and enables the MCP Board to process the additional firmware instruction bytes (immediate, direct, extended, or relative addressing). A pipeline request provides a means by which the ALU Board has control of the MAS Board from the micro code (PLMC0 thru PLMC2) for power-up and firmware instructions that require more than an implied request.

MAS Control

The MAS Control supplies the handshaking between the MCP Board and the MAS Board. When an address is available on the MB0 thru MB15 lines, ADDRVALID-0 is low. When write data is available on the MB0 thru MB15 lines, DATAVALID-0 is low. When the address has been accepted by the MAS Board, ADRACPT-0 goes low and ADDRVALID-0 goes high. Data between the MAS Board and the MCP Board is confirmed by DCNFRM-0 going low. During a memory read operation, DCNFRM-0 going low indicates that the data is available from the MAS Board on the MB0 thru MB15 lines. During a memory write operation, DCNFRM-0 going low indicates that the data has been received by the MAS Board.

Timing

The Timing block is used for the MCP Board cycles and buffers the system clock (25MHZ-1, MCP40NSCLK-1, and M40NSCLK-1) on the MCP Board and to the MAS Board.

ALU Handshake

The ALU Handshake tells the ALU Board that the data from the ALU Board has been accepted by the MCP Board (ALUACPT-1 goes high).

Instruction Decode

The Instruction Decode takes the three request types (fetch request - FREQ-1 high; implied request - IMPLREQ-1 high; and pipeline request - PLREQ-0 low) and decodes the type of operation the MCP Board is to execute for the selected request.

The active output lines depend on which request is active. Only one type of request can be active at any one time. Most of the outputs are used by the Mbus Enable block to control data and address transfer on the MB0 thru MB15 lines.

Mbus Enable

The Mbus Enable decodes the outputs of the Instruction Decode to control the buffers and registers that are connected to the MB0 thru MB15 lines. PCOUT-0 enables the output of the Program Counter to the MB0 thru MB15 lines. MBLOUT-0 enables the Memory Byte Latch to the MB0 thru MB7 lines. ZEROOUT-0 enables the High Zero/Sign Extended to the MB8 thru MB15 lines. ALUOUTEN-0 enables the Aluout Latch to the MB0 thru MB15 lines. The MSPACE-1 output is the control line to the MAS Board. MSPACE-1 indicates the memory space (RAM or ROM) the MAS Board uses for the memory operation. If MSPACE-1 is high, RAM space is used. If MSPACE-1 is low, ROM space is used (see Figure 7-5).

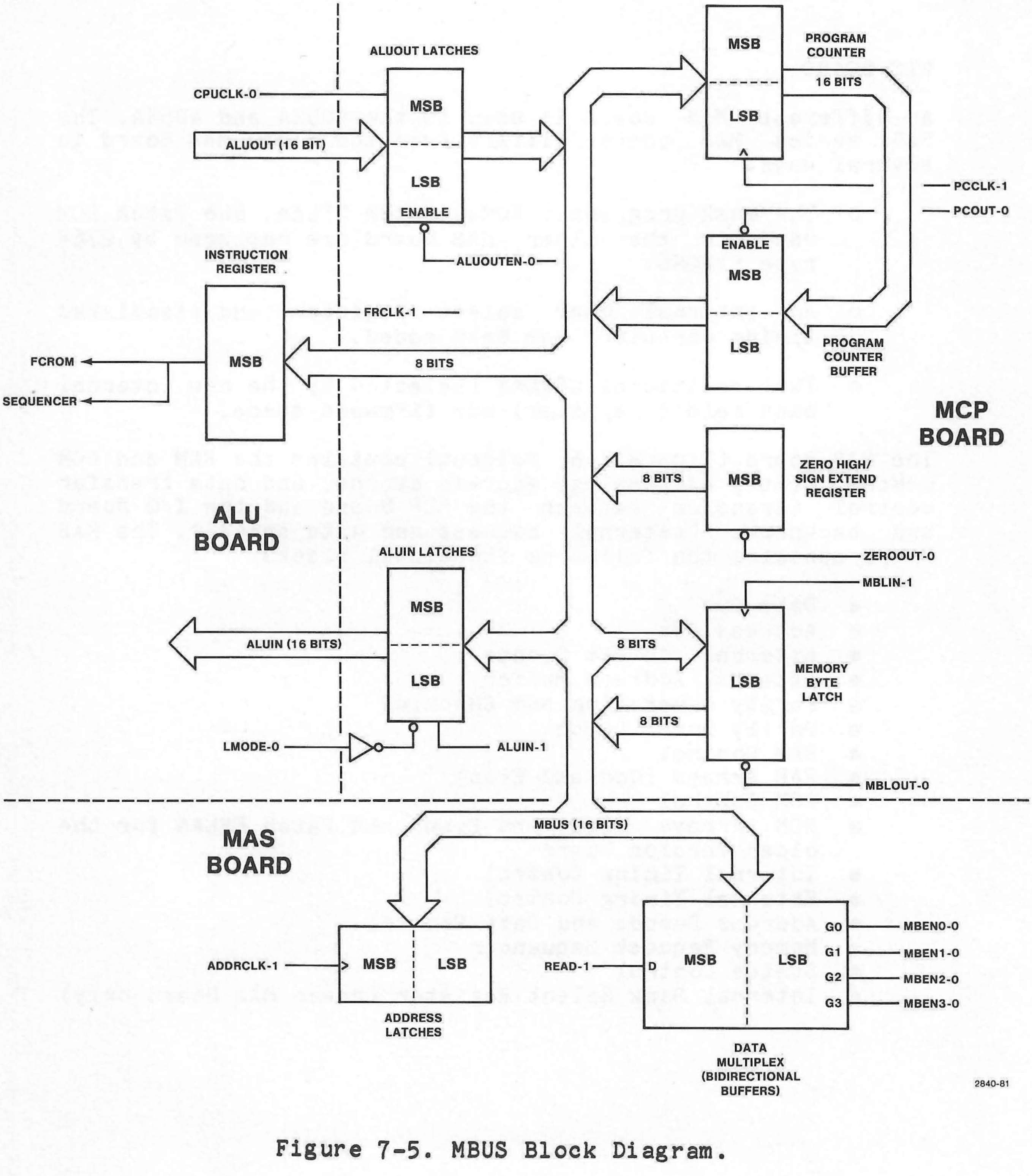


Figure 7-5. MBUS Block Diagram.

MAS BOARD

A different MAS Board is used in the 4052A and 4054A. The "A" series MAS board differs from the older MAS Board in several ways:

- o The mask programmed ROMs, Patch FPLAs, and Patch ROM used on the older MAS Board are replaced by 2764 type EPROMS.
- o An internal bank select register and associated timing circuitry has been added.
- o Two additional EPROMs (selected by the new internal bank select register) add firmware space.

The MAS Board (Figure 7-6, foldout) contains the RAM and ROM memory, memory addressing, address decode, and data transfer control (transfer between the MCP Board and the I/O Board and backpacks [external address and data space]). The MAS Board contains the following functional blocks:

- ● Data Bus
- ● Address Bus
- ● External Address Decode
- ● External Address Buffer
- ● Parity Generating and Checking
- ● Parity Error Latch
- ● RAM Control
- ● RAM Arrays (Odd and Even)
- ● ROM Control
- ● ROM Arrays (Odd and Even) and Patch FPLAs for the older version board
- ● Internal Timing Control
- ● External Timing Control
- ● Address Decode and Data Control
- ● Memory Request Sequencer
- ● Status Control
- ● Internal Bank Select Register (newer MAS Board only)

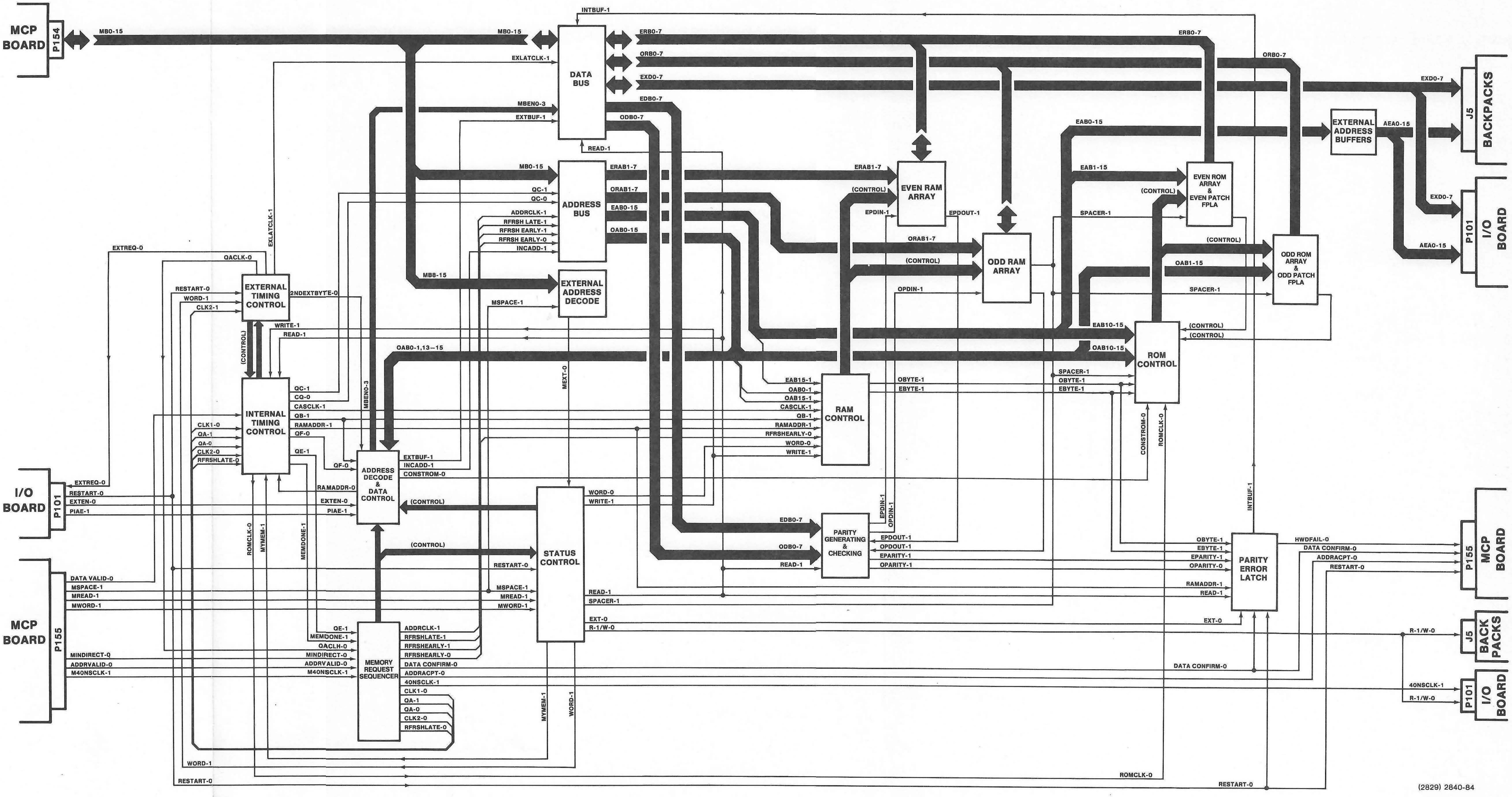


Figure 7-6. MAS Block Diagram.

Data Bus

The Data Bus provides the data path between the internal data registers and the internal (system ROM and RAM) or external (I/O Board or backpack) memory space. Data direction is controlled by the READ-1 line. When READ-1 is low (memory write) the data is transferred from the MB0 through MB15 to the EDB0 through EDB15 and ODB0 through ODB15 lines. WHen READ-1 is high (read memory) the data goes in the other direction.

The memory spaces are divided into even and odd addressing of data bytes. Data transfer between the MAS Board and the MCP Board is two bytes or one byte (Figure 7-7). When one byte is transferred, MB0 thru MB7 is active with EDB0 thru EDB7 (U195) for even addresses and with ODB0 thru ODB7 (U165) for odd addresses. External one-byte data transfers are between EXD0 thru EXD7 and EDB0 thru EDB7 (U495) for both even and odd external addresses (EDB0 thru EDB7 is active with MB0 thru MB7). During an odd address or an external (even and odd addresses read only, write to external is one-byte only) two-byte data transfer, MB0 thru MB7 is active with EDB0 thru EDB7 (U185); MB8 thru MB15 is active with ODB0 thru ODB7 (U170). The first external byte (EXD0 thru EXD7) is latched in U160 and output to ODB0 thru ODB7. The second external byte is transfered from EXD0 thru EXD7 to EDB0 thru EDB7 (U495). When two bytes of data at an even address are transferred, MB0 thru MB7 is active with ODB0 thru ODB7 (U165) and MB8 thru MB15 is active with EDB0 thru EDB7 (U185). The MBEN0 thru MBEN3 control lines control these data transfers.

The ODB0 thru ODB7 lines are buffered to the odd RAM and ROM (ORB0 thru ORB7) by U270. The EDB0 thru EDB7 lines are buffered to the even RAM and ROM (ERB0 thru ERB7) by U395.

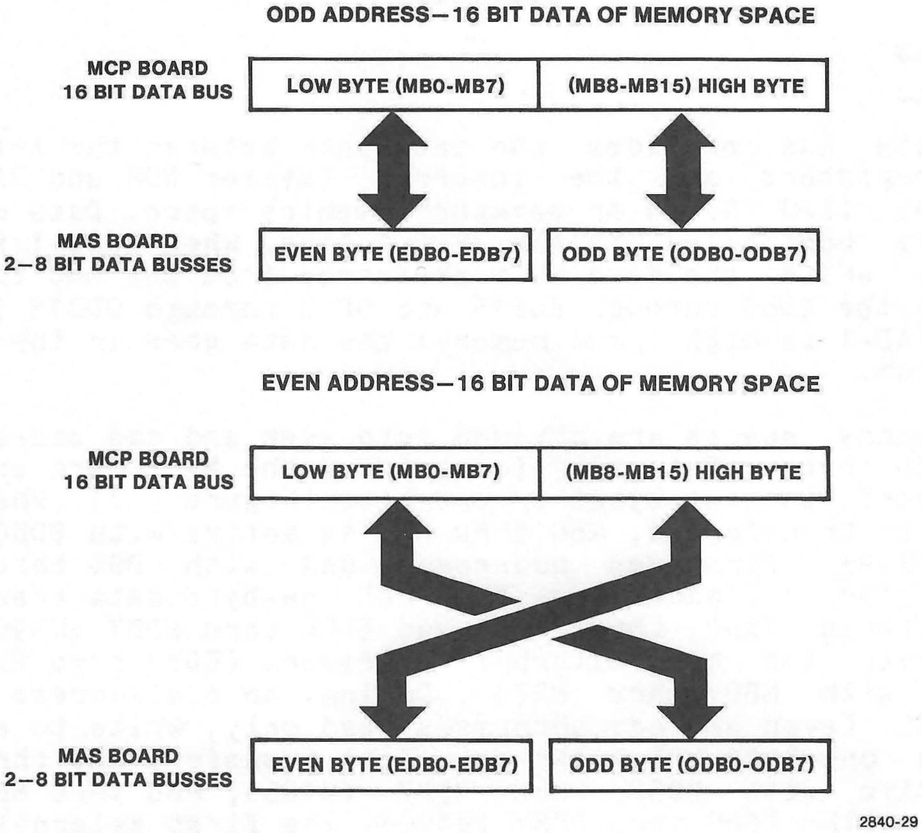

ODD ADDRESS—16 BIT DATA OF MEMORY SPACE
MCP BOARD 16 BIT DATA BUS
LOW BYTE (MB0-MB7)
(MB8-MB15) HIGH BYTE
MAS BOARD 2—8 BIT DATA BUSSES
EVEN BYTE (EDB0-EDB7)
ODD BYTE (ODB0-ODB7)
EVEN ADDRESS—16 BIT DATA OF MEMORY SPACE
MCP BOARD 16 BIT DATA BUS
LOW BYTE (MB0-MB7)
(MB8-MB15) HIGH BYTE
MAS BOARD 2—8 BIT DATA BUSSES
EVEN BYTE (EDB0-EDB7)
ODD BYTE (ODB0-ODB7)
2840-29

Figure 7-7. Data Byte Transfer.

Address Bus

The Address Bus latches the address from the MCP Board, buffers the destination of the address, and supplies the address counting for dynamic RAM refresh. The address from the MBO through MB15 lines is latched (via U180 and U190) and supplied to the OABO through OAB15 lines.

Internal memory (both ROM and RAM is arranged into two banks, an Odd Bank and and Even Bank. These two banks are addressed by the lines EAB1 through EAB15 (Even Address Bank 1-15) and OAB1 through OAB15 (Odd Address Bank 1-15). An adder consisting of U385, U390, and U480 increments the EABO through EAB15 lines for odd word addresses and two-byte external addresses. The input to the adder is from OABO through OAB15 and the output of the adder is sent to lines EABO through EAB15.

Single Byte Operations

During single byte operations, the addresses present on the EAB1-15 lines and the addresses present on the OAB1-15 lines are identical. Selection of the odd or even memory bank is done by line OAB-0.

Word (Two-byte) Operations

During word or two-byte operations, both banks are addressed regardless of the state of OAB-0. When the first byte of a two-byte operation is located at an even address (on MBO-15), the addresses on the two banks will be the same. However, when the first byte of the two-byte operation is located at an odd address, the EAB1-15 address is the OAB1-15 address incremented by the adder (the value of OAB1-15 + 1).

External Addressing

A two-byte external address is sent to the External Address Buffer on the EABO-15 lines in two memory cycles. During the first cycle, the address from the latches (U180 and U190) is placed onto the bus and also into the adder. During the second memory cycle, the adder increments the address by one (INCADD-1 goes high).

RAM Refresh Addressing

The dynamic RAM memory is refreshed by periodically cycling through the OAB0-7 lines. A counter (U295) is incremented whenever the line RFSHLATE-1 goes high. The output of the counter is buffered by U290, and whenever RFSHEARLY-0 goes low, the buffered address is placed on the OAB0-7 lines.

Dynamic RAM Addressing

The dynamic RAMs are addressed through buffers U460 and U465 from lines OAB1-14 to lines ORAB1-7 and buffers U430 and U435 from lines EAB1-14 to lines ERAB1-7.

External Address Decode

The External Address Decode enables the external space for access. The MEXT-0 line is low for RAM space address (MSPACE-1 is high) X'FF00' to X'FFFF' and for ROM space address (MSPACE-1 is low) X'0000' to X'3FFF'. The RAM space of external is used for the I/O Board addressing. The ROM space of external is used for the switchable backpack address (16K).

External Address Buffer

The External Address Buffer is enabled at all times. The input to the buffers (U485 and U490) is from the EAB0 thru EAB15 lines. The output is to the AEA0 thru AEA15 lines. The AEA0 thru AEA15 lines address the I/O Board and the backpacks.

Parity Generating and Checking

The Parity Generating and Checking consists of two parity generating intergrated circuits, one for even RAM space (U285) and one for odd RAM space (U260). The data is from the ODB0 thru ODB7 (odd RAM space) and EDB0 thru EDB7 (even RAM space). During a RAM space write, the generators create an even parity output to OPDIN-1 and EPDIN-1 for storage in RAM space. During a read of data, the generators input the data with the parity stored during write and create an output. If there is an error OPARITY-1 or EPARITY-1 goes high.

Parity Error Latch

The Parity Error Latch is used to generate a HWDFAIL-0 interrupt. The conditions required for the interrupt are a read operation of the RAM Array (U345A-3 is high) and either an odd parity error (OPARITY-1 is high) or an even parity error (EPARITY-1 is high). During one-byte memory read, only the RAM space addressed parity is active (odd or even). The LED, DS15, is turned on by a parity error.

RAM Control

The RAM Control does the decoding for one or two-byte access, read/write control, row addressing (RAS) control, and column addressing (CAS) control. WORD-0 controls the one and two-byte memory access. When WORD-0 is low, both even and odd RAM arrays are active (two-byte read/write). When WORD-0 is high, OAB0-1 selects the active RAM array (one-byte read/write). WRITE-1 controls the reading and writing of both RAM arrays. WRITE-1 high is a write operation. Each RAM bank (even and odd) is divided into two 16K-byte sections. EAB15-1 selects the active 16K section of the Even RAM Array. OAB15-1 selects the active 16K section of the Odd RAM Array. Row address select (RAS) is activated for refresh by RFRSHEARLY-0 going low and for memory access by EAB15-1 and OAB15-1. RFRSHEARLY-0 activates all four 16K sections of RAM. Column address select (CASCLK-1) goes high to select the RAMs for read/write addressing.

RAM Arrays (Odd and Even)

The Even RAM Array and the Odd RAM Array functions exactly the same; therefore, only one is described here. Each RAM array is divided into two 16K by 9 bit sections. Eight bits are used for data, and the ninth bit is used for storing the parity. The control and selection is from the RAM Control. The address is supplied in two 7-bit parts from the ERAB1 thru ERAB7 lines. The row address (RAS) input is used for refresh of memory and data access addressing. The row address is latched into the RAMs, and then the column address (CAS) is supplied to the RAMs. Column address is not used for the RAM refresh.

ROM Control

The ROM Control selects the ROMs addressed by the computer system. These may be the Odd ROM array, the Even ROM Array, or in the case of the older boards, the Patch ROMs.

The address decoders, U420 and U470, select the active ROM. The address decoders are enabled by SPACER-1, OBYTE-1, and EBYTE-1 when ROMCLK-0 goes low. CONSTROM-0 selects the constant ROMs.

On the older boards using FPLAs and Patch ROMs, the FPLAs are used to detect patch addresses. The flag outputs (U863-19 and U845-19) are used to disable the address decoders when the FPLAs detect a patch address. The FPLAs supply the byte or bytes at the trap memory address. The FPLAs' flag output also disables the constant ROMs (U810 and U893) to allow patching of constants.

On the newer boards, the ROM Control can be disabled by placing a strap at J419.

The Even ROM Array and Even Patch FPLA, and the Odd ROM Array and Odd Patch FPLA are functionally the same; therefore, only one is described here. The Patch FPLA looks at all addresses. When a memory patch address has been received, the flag output (pin 19) goes low. The FPLA outputs a data byte to the data bus instead of the ROMs. In each ROM array there are four 8K by 8-bit ROMs. Three ROMs are used for firmware instructions, and one is used for constant data. The constant ROM has an address in the lower 8K of RAM space. The patch ROMs are either 512 by 8-bit ROMs or 256 by 8-bit ROMs. The patch ROMs are used for patching large sections of memory and are accessed by instructions executed from the FPLAs.

Internal Timing Control

The Internal Timing Control develops the RAM and ROM timing signals needed for memory access, and provides timing for system handshake signals needed four internal memory operations. The timing is developed by a series of flip-flops. At the beginning of a memory operation, each flip-flop in the series is clocked, in succession, to a logical 1. Then the series is sequentially clocked to a logical 0 until all flip-flops have been zeroed. Combination of the flip-flop's outputs supplies the MAS Board with the signals used to control the Data and Address Busses, the RAM and ROM clocks, and the system handshake signals.

External Timing Control

The External Timing Control supplies the timing signals when an external access is required. When high, the WORD-1 signal sets up the timing for two-byte external access by setting U102A. When U102A is set, it forces the I/O board to complete two cycles by placing the external timing control into a mode which requires two DAX-0 pulses before EXTREQ-0 is removed.

Address Decode and Data Control

The Address Decode and Data Control supplies the selection of memory space (RAM, ROM, constant ROM, or external), the read/write control lines (for the MAS Board and the I/O Board), and the Data Bus buffer selection. The memory select control lines are SPACER-1, RAMADDR-0, CONSTROM-0, and EXT-0. The read/write control lines are R-1/W-0, WRITE-1, and READ-1. The Data Bus buffer lines (MBEN0 thru MBEN3) are supplied from a ROM (U250).

Memory Request Sequencer

The Memory Request Sequencer starts the memory access and controls the dynamic RAM refresh cycles. The MAS Board starts a memory access when ADDRVALID-0 goes low, MINDRECT-0 goes low, or there is a refresh request from U120B-9. ADDRVALID-0 is the control line used for address from the MCP Board. MINDIRECT-0 is used during an extended addressing mode (the data out of the MAS Board is the address of the next memory access). The control lines used for refresh are RFRSHEARLY (-1 and -0) and RFRSHLATE (-1 and -0).

Status Control

The Status Control latches information to perform the memory accesses. The following information is latched in U225: read/write (MREAD-1 is high for read operations); one or two-byte data (MWORD-1 is high for two bytes); RAM or ROM space (MSPACE-1 is high for RAM space); and external access (MEXT-0 is low for an external access).

External Bank Select (New Board Only)

The External Bank Select is a single bit addressed register which is written to in parallel with the ROM Pack Bank Select Register. When bit six is high, U380B is set. This disables external memory accesses in ROM space and enables EPROMs U863 and U845. When U380B is cleared, normal external memory accesses are again enabled.

I/O BOARD

The I/O Board contains the following functional areas:

- o I/O Control, Address Decode, and Clocking
- o Magnetic Tape Interface Logic
- o Keyboard Interface Logic
- o Display Interface Logic
- o GPIB Interface

Two versions of the I/O Board exist for the 4052/4052A and two for the the 4054/4054A. They differ in that the newer board has the GPIB control performed by a single integrated circuit.

General interfacing is done through PIAs and discrete logic located on the I/O Board. You should be familiar with the Manufacturer's Data sheets before attempting to diagnose or repair problems involving PIAs or the GPIB Interface circuit.