## Appendix A

## 4052/4054 FIRMWARE INSTRUCTIONS

The following table gives the hex code, opcode abbrevation, and address mode for the 4052/4054 system firmware instructions.

* 4052 CUSTOM CODE

44 NEW (14 OF THESE ARE FLOATING PT)
REST OF INVALID OPCODES ARE TRAP(ed)

| | Hex Code | Opcode | Address Mode |
|---|---|---|---|
| * | 00 | TRAP | NONE |
| | 01 | NOP | INHERENT |
| * | 02 | NOP2 | INHERENT |
| * | 03 | SFA | INHERENT |
| * | 04 | TRAP | NONE |
| * | 05 | TAP | INHERENT |
| | 06 | TAP | INHERENT |
| | 07 | TPA | INHERENT |
| | 08 | INX | INHERENT |
| | 09 | DEX | INHERENT |
| | 0A | CLV | INHERENT |
| | 0B | SEV | INHERENT |
| | 0C | CLC | INHERENT |
| | 0D | SEC | INHERENT |
| | 0E | CLI | INHERENT |
| | 0F | SEI | INHERENT |
| | 10 | SAB | INHERENT |
| | 11 | CBA | INHERENT |
| * | 12 | TAPX | INHERENT |
| * | 13 | TPAX | INHERENT |
| * | 14 | ADXI | IMMEDIATE |
| * | 15 | ASPI | IMMEDIATE |
| | 16 | TAB | INHERENT |
| | 17 | TBA | INHERENT |
| * | 18 | SDA | INHERENT |
| | 19 | TRAP | NONE |
| * | 1A | NLDXX | INHERENT |
| | 1B | ABA | INHERENT |
| * | 1C | NLDAX | INHERENT |
| * | 1D | NLDBX | INHERENT |
| * | 1E | NSTAX | INHERENT |
| * | 1F | JMPAX | INHERENT |
| | 20 | BRA | RELATIVE |
| * | 21 | SDB | INHERENT |
| | 22 | BHI | RELATIVE |

| Hex Code | Opcode | Address Mode |
|---|---|---|
| 23 | BLS | RELATIVE |
| 24 | BCC | RELATIVE |
| 25 | BCS | RELATIVE |
| 26 | BNE | RELATIVE |
| 27 | BEQ | RELATIVE |
| 28 | BVC | RELATIVE |
| 29 | BVS | RELATIVE |
| 2A | BPL | RELATIVE |
| 2B | BMI | RELATIVE |
| 2C | BGE | RELATIVE |
| 2D | BLT | RELATIVE |
| 2E | BGT | RELATIVE |
| 2F | BLE | RELATIVE |
| 30 | TSX | INHERENT |
| 31 | INS | INHERENT |
| 32 | PULA | INHERENT |
| 33 | PULB | INHERENT |
| 34 | DES | INHERENT |
| 35 | TXS | INHERENT |
| 36 | PSHA | INHERENT |
| 37 | PSHB | INHERENT |
| 38 | JMPIN | EXTENDED |
| 39 | RTS | INHERENT |
| 3A | FPSHD | DIRECT |
| 3B | RTI | INHERENT |
| 3C | FPSHX | INDEXED |
| 3D | FPSH | EXTENDED |
| 3E | WAI | INHERENT |
| 3F | SWI | INHERENT |
| 40 | NEGA | INHERENT |
| 41 | FPSHI | IMMEDIATE |
| 42 | FPULD | DIRECT |
| 43 | COMA | INHERENT |
| 44 | LSRA | INHERENT |
| 45 | FPULX | INDEXED |
| 46 | RORA | INHERENT |
| 47 | ASRA | INHERENT |
| 48 | ASLA | INHERENT |
| 49 | ROLA | INHERENT |
| 4A | DECA | INHERENT |
| 4B | FPUL | EXTENDED |
| 4C | INCA | INHERENT |
| 4D | TSTA | INHERENT |
| 4E | FDUP | INHERENT |
| 4F | CLRA | INHERENT |

| | Hex Code | Opcode | Address Mode |
|---|---|---|---|
| | 50 | NEGB | INHERENT |
| ☆ | 51 | FSWP | INHERENT |
| ☆ | 52 | FADD | INHERENT |
| | 53 | COMB | INHERENT |
| | 54 | LSRB | INHERENT |
| ☆ | 55 | FSUB | INHERENT |
| | 56 | RORB | INHERENT |
| | 57 | ASRB | INHERENT |
| | 58 | ASLB | INHERENT |
| | 59 | ROLB | INHERENT |
| | 5A | DECB | INHERENT |
| ☆ | 5B | FMUL | INHERENT |
| | 5C | INCB | INHERENT |
| | 5D | TSTB | INHERENT |
| ☆ | 5E | FDIV | INHERENT |
| | 5F | CLRB | INHERENT |
| | 60 | NEGX | INDEXED |
| ☆ | 61 | FNORM | INHERENT |
| ☆ | 62 | PSHRET | DIRECT |
| | 63 | COMX | INDEXED |
| | 64 | LSRX | INDEXED |
| ☆ | 65 | RTRN | DIRECT |
| | 66 | RORX | INDEXED |
| | 67 | ASRX | INDEXED |
| | 68 | ASLX | INDEXED |
| | 69 | ROLX | INDEXED |
| | 6A | DECX | INDEXED |
| ☆ | 6B | PSHX | INHERENT |
| | 6C | INCX | INDEXED |
| | 6D | TSTX | INDEXED |
| | 6E | JMPX | INDEXED |
| | 6F | CLRX | INDEXED |
| | 70 | NEG | EXTENDED |
| ☆ | 71 | STROKE | INHERENT |
| ☆ | 72 | EC | INHERENT |
| | 73 | COM | EXTENDED |
| | 74 | LSR | EXTENDED |
| ☆ | 75 | PULX | INHERENT |
| | 76 | ROR | EXTENDED |
| | 77 | ASR | EXTENDED |
| | 78 | ASL | EXTENDED |
| | 79 | ROL | EXTENDED |
| | 7A | DEC | EXTENDED |
| ☆ | 7B | TRAP | NONE |

| Hex Code | Opcode | Address Mode |
|---|---|---|
| 7C | INC | EXTENDED |
| 7D | TST | EXTENDED |
| 7E | JMP | EXTENDED |
| 7F | CLR | EXTENDED |
| 80 | SUBAI | IMMEDIATE |
| 81 | CMPAI | IMMEDIATE |
| 82 | SBCAI | IMMEDIATE |
| 83 | TRAP | NONE |
| 84 | ANDAI | IMMEDIATE |
| 85 | BITAI | IMMEDIATE |
| 86 | LDAAI | IMMEDIATE |
| 87 | TRAP | NONE |
| 88 | EORAI | IMMEDIATE |
| 89 | ADCAI | IMMEDIATE |
| 8A | ORAAI | IMMEDIATE |
| 8B | ADDAI | IMMEDIATE |
| 8C | CPXI | IMMEDIATE |
| 8D | BSR | RELATIVE |
| 8E | LDSI | IMMEDIATE |
| 8F | TRAP | NONE |
| 90 | SUBAD | DIRECT |
| 91 | CMPAD | DIRECT |
| 92 | SBCAD | DIRECT |
| 93 | TRAP | NONE |
| 94 | ANDAD | DIRECT |
| 95 | BITAD | DIRECT |
| 96 | LDAAD | DIRECT |
| 97 | STAAD | DIRECT |
| 98 | EORAD | DIRECT |
| 99 | ADCAD | DIRECT |
| 9A | ORAAD | DIRECT |
| 9B | ADDAD | DIRECT |
| 9C | CPXD | DIRECT |
| 9D | TRAP | NONE |
| 9E | LDSD | DIRECT |
| 9F | STSD | DIRECT |
| A0 | SUBAX | INDEXED |
| A1 | CMPAX | INDEXED |
| A2 | SBCAX | INDEXED |
| A3 | TRAP | NONE |
| A4 | ANDAX | INDEXED |
| A5 | BITAX | INDEXED |
| A6 | LDAAX | INDEXED |
| A7 | STAAX | INDEXED |

| | Hex Code | Opcode | Address Mode |
|---|---|---|---|
| | A8 | EORAX | INDEXED |
| | A9 | ADCAX | INDEXED |
| | AA | ORAAX | INDEXED |
| | AB | ADDAX | INDEXED |
| | AC | CPXX | INDEXED |
| | AD | JSRX | INDEXED |
| | AE | LDSX | INDEXED |
| | AF | STSX | INDEXED |
| | B0 | SUBA | EXTENDED |
| | B1 | CMPA | EXTENDED |
| | B2 | SBCA | EXTENDED |
| ☆ | B3 | TRAP | NONE |
| | B4 | ANDA | EXTENDED |
| | B5 | BITA | EXTENDED |
| | B6 | LDAA | EXTENDED |
| | B7 | STAA | EXTENDED |
| | B8 | EORA | EXTENDED |
| | B9 | ADCA | EXTENDED |
| | BA | ORAA | EXTENDED |
| | BB | ADDA | EXTENDED |
| | BC | CPX | EXTENDED |
| | BD | JSR | EXTENDED |
| | BE | LDS | EXTENDED |
| | BF | STS | EXTENDED |
| | C0 | SUBBI | IMMDIATE |
| | C1 | CMPBI | IMMEDIATE |
| | C2 | SBCBI | IMMEDIATE |
| ☆ | C3 | TRAP | NONE |
| | C4 | ANDBI | IMMEDIATE |
| | C5 | BITBI | IMMEDIATE |
| | C6 | LDABI | IMMEDIATE |
| ☆ | C7 | TRAP | NONE |
| | C8 | EORBI | IMMEDIATE |
| | C9 | ADCBI | IMMEDIATE |
| | CA | ORABI | IMMEDIATE |
| | CB | ADDBI | IMMEDIATE |
| ☆ | CC | ADAX | INHERENT |
| ☆ | CD | WADAX | INHERENT |
| | CE | LDXI | IMMEDIATE |
| ☆ | CF | TRAP | NONE |
| | D0 | SUBBD | DIRECT |
| | D1 | CMPBD | DIRECT |
| | D2 | SBCBD | DIRECT |
| ☆ | D3 | TRAP | NONE |

| Hex Code | Opcode | Address Mode |
|---|---|---|
| D4 | ANDBD | DIRECT |
| D5 | BITBD | DIRECT |
| D6 | LDABD | DIRECT |
| D7 | STABD | DIRECT |
| D8 | EORBD | DIRECT |
| D9 | ADCBD | DIRECT |
| DA | ORABD | DIRECT |
| DB | ADDBD | DIRECT |
| DC | SBUG | INHERENT |
| DD | CBUG | INHERENT |
| DE | LDXD | DIRECT |
| DF | STXD | DIRECT |
| E0 | SUBBX | INDEXED |
| E1 | CMPBX | INDEXED |
| E2 | SBCBX | INDEXED |
| E3 | MVLR | INHERENT |
| E4 | ANDBX | INDEXED |
| E5 | BITBX | INDEXED |
| E6 | LDABX | INDEXED |
| E7 | STABX | INDEXED |
| E8 | EORBX | INDEXED |
| E9 | ADCBX | INDEXED |
| EA | ORABX | INDEXED |
| EB | ADDBX | INDEXED |
| EC | MVRL | INHERENT |
| ED | WADX | EXTENDED |
| EE | LDXX | INDEXED |
| EF | STXX | INDEXED |
| F0 | SUBB | EXTENDE |
| F1 | CMPB | EXTENDED |
| F2 | SBCB | EXTENDED |
| F3 | CPCH | IMMEDIATE |
| F4 | ANDB | EXTENDED |
| F5 | BITB | EXTENDED |
| F6 | LDAB | EXTENDED |
| F7 | STAB | EXTENDED |
| F8 | EORB | EXTENDED |
| F9 | ADCB | EXTENDED |
| FA | ORAB | EXTENDED |
| FB | ADDB | EXTENDED |
| FC | TRAP | NONE |
| FD | PCH | IMMEDIATE |
| FE | LDX | EXTENDED |
| FF | STX | EXTENDED |

## 4052/4054 SYSTEM BLOCK DESCRIPTION

The description presented here is a general overview of the 4052 and 4054 systems. For a detailed description refer to Section 7 (ALU, MCP, AND MAS CIRCUIT BOARDS THEORY OF OPERATION) or to the I/O sections (keyboard, mag tape unit, display, or GPIB). Refer to Figure 6-9 and Figure 6-10 for the following description. The only difference between the 4052 and the 4054 systems is the display circuitry.

### 4052/4054 System Overview

The 4052/4054 system consists of the I/O units (display, keyboard, backpacks, internal mag tape unit, and GPIB) and the processor boards (ALU, MCP, and MAS Boards). The I/O Board interfaces the processor boards to the I/O units. The functions of the processor boards and I/O Board can be summarized as follows:

- ALU - Arithmetic Logic Unit. This is the microprogrammed processor board containing the arithmetic logic units (RALUs). The system microcode, the major processor registers (accumulators, index register, stack pointer, etc.), and master clock circuitry. This board, through the mircroprograms and related hardware, controls the fundamental system operations and execution of firmware (macro) level instructions.

- MCP - Memory Command Processor. Directing memory operations, the MCP plays a major role in the execution of firmware instructions and coordinating data transfers between the ALU and memory. Some of its' primary functions are: determining the source and destination of address/data information being passed to the Memory Access Sequencer, relaying information on the number of bytes involved in a memory transfer or which memory space is involved in the operation, and upkeep of the firmware level program counter (the only processor-related register not contained on the ALU). It also is involved in the operation of the MBUS (main 16-bit bus between the MCP and MAS).

- MAS - Memory Access Sequencer. Under the direction of the MCP, the MAS performs the actual memory accesses for read/write operations. Besides containing the timing logic for these operations, the MAS also holds address decoding logic for accessing the I/O and ROM Pack address spaces. All system RAM and ROM (firmware) are located on this board.

- I/O - Input/Output. Providing the interface to the outside world, the I/O board generates the necessary timing for the operation of the keyboard, magtape, display, GPIB, and peripheral/ROMPACK circuitry.

There are four types of data carriers in the 4052/4054 system:

- ADDRESS AND DATA. The 16-bit data or the 16-bit memory space address is carried on this bus between the MCP Board and the MAS Board.

- ADDRESS ONLY. The 16-bit address from the MCP Board is split into two 16-bit address busses on the MAS Board: an even address bus and an odd address bus. The even address bus is used for even RAM and ROM addressing, for external addressing of the backpacks, and for the I/O Board addressing. The odd address bus is used for odd RAM and ROM.

- DATA ONLY. The 16-bit data between the MCP Board and the MAS Board is split into two 8-bit busses: an even data bus and an odd data bus. The even data bus is for even RAM and ROM data, and for external data for the backpacks and the I/O Board. The odd data bus is for the odd RAM and ROM data and 16-bit external data (first external byte).

- OTHER.The rest consists of Micro ROM addressing, timing signals, control lines, I/O data, and handshaking between the circuit boards.

The Micro ROM on the ALU Board performs a partial system check during power up (refer to 4052/4054 MICROCODE). When finished with the power up operation, the microcode is controlled by firmware instructions from the MAS Board or Backpacks. The firmware instruction is latched into the Instruction Register. The Instruction Register is decoded and the microcode sequence for the firmware instruction is executed. During each microcode instruction the microcode word is latched into the Pipeline Register. This allows the ALU Board control to set up the address of the next microcode word to be executed.

Data between the ALU Board and the MCP Board is transfered over two 16-bit busses: one for data to the RALU and one for data from the RALU. The RALU and Condition Code Logic consist of 16 registers, an arithmetic logic unit, Condition Code Register, and shift logic.

Part of the microcode controls the operation and data-address transfer between the MAS Board, the MCP Board, and the ALU Board. The MCP Board has a Program Counter, Zero High/Sign Extended, ALU input and output latches, and a Memory Byte Latch. The Program Counter is used for addressing of the firmware instruction. The ALU input and output latches are used to store RALU data. The Memory Byte Latch and the Zero High/Sign Extended are used together during firmware instruction fetching and execution.

The MCP Board controls the address and read or write of data on the MAS Board. The address from the MCP Board is received by the Address Bus Logic. For non-external addressing the address is put on the odd address bus, and the address plus one is put on the even address bus. For external addressing, the address is put on the even address bus and on the external address lines. If only one byte is needed the external addressing is finished, but if two bytes are needed the even address is incremented and the external address is enabled again for the second byte. When 16 bits of data from external are read, the first byte is latched and put on the odd data bus, while the second byte is placed on the even data bus.

The data to and from the MCP Board to the MAS Board splits into an 8-bit odd data bus and an 8 bit even data bus on the MAS Board. Depending on the operation and the address (Figure 6-11), even data is active on either the low byte or the high byte of MCP Board 16-bit data but not to both bytes. The odd data is active on the byte not used by the even data. The Data Bus Logic can cross the data bytes between the MAS Board and the MCP Board.

The I/O Board contains the external interface to the internal mag tape drive, the keyboard, the display, and the GPIB Interface.

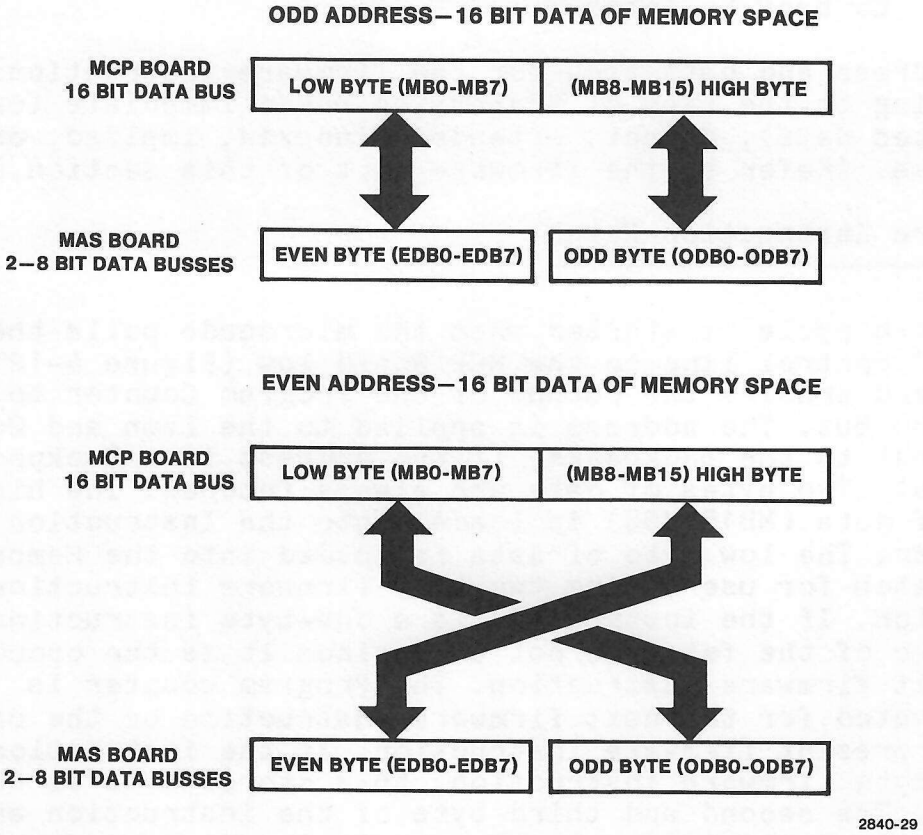


Figure 6-11. MCP Board and MAS Board Data Bytes Transfer.

## 4052/4054 FIRMWARE INSTRUCTION DATA TRANSFER

The 4052/4054 firmware instructions are processed by the processor in the following sequence:

- Fetch firmware instruction.
- Decode instruction.
- Execute the instruction.
- Honor pending interrupts.
- Go back to fetch.

The address and data flow for the firmware instructions vary depending on the type of addressing used: immediate (one or two bytes data), direct, extended, indexed, implied, or relative. (Refer to the firmware part of this section.)

### Firmware Instruction Fetch

The fetch cycle is started when the microcode pulls the MHOLD-0 control line to the MCP Board low (Figure 6-12). The MCP Board enables the output of the Program Counter to the MB15-MB0 bus. The address is applied to the Even and Odd ROM (external to the Backpacks, if the address is a Backpack address). Two bytes of data are always fetched. The high byte of data (MB15-MB8) is loaded into the Instruction Register. The low byte of data is loaded into the Memory Byte Latch for use during two-byte firmware instruction execution. If the instruction is a one-byte instruction, the low byte of the fetch is not used since it is the opcode for the next firmware instruction. The Program counter is incremented for the next firmware instruction or the operand of the present firmware instruction. If the instruction is a three-byte firmware instruction, this stored byte is discarded. The second and third byte of the instruction are fetched together (refer to three-byte immediate and extended instructions).

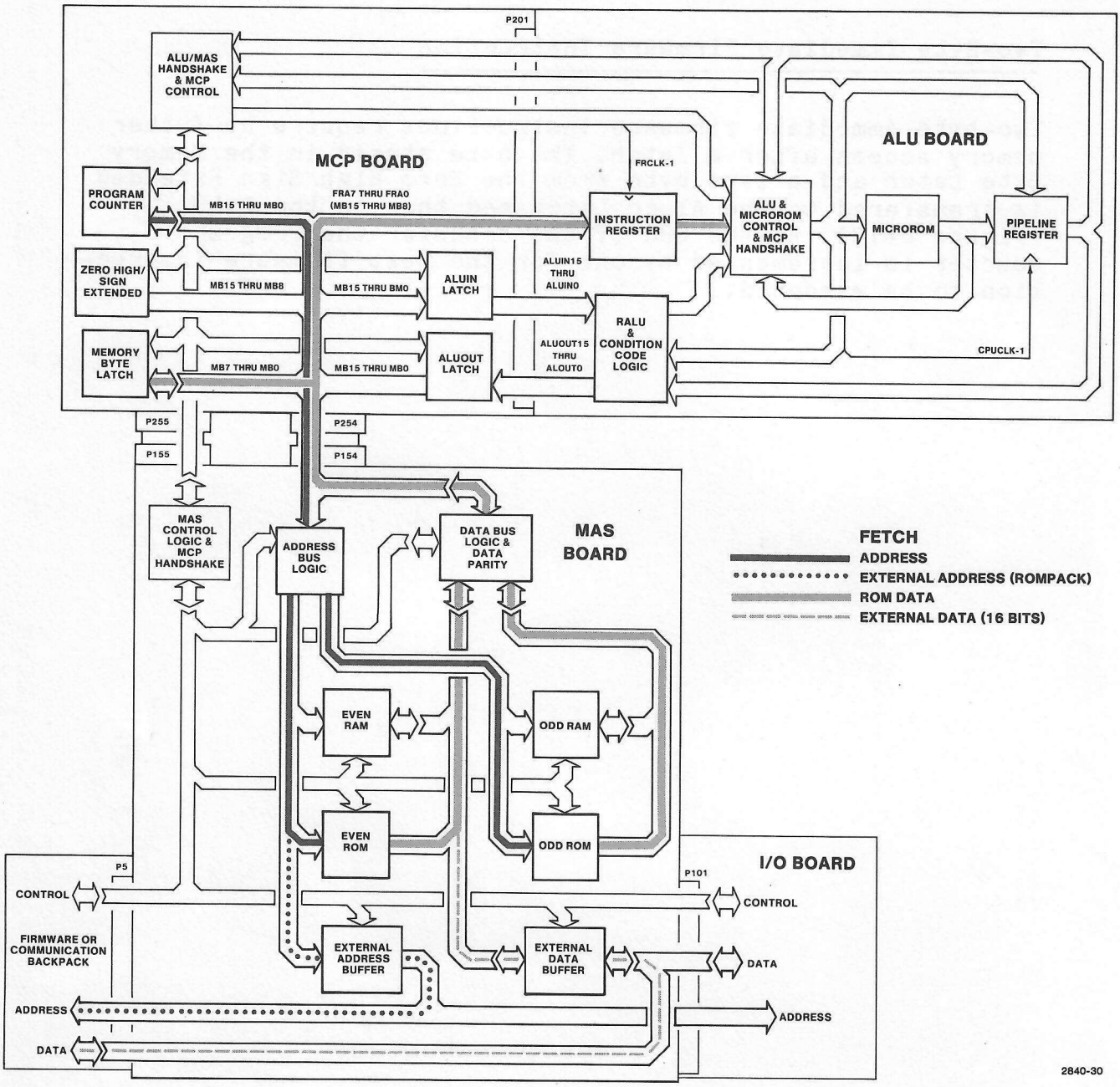


**Figure 6-12. 4052/4054 Firmware Instruction Fetch.**

## Two-Byte Immediate Firmware Instruction

Two-byte immediate firmware instructions require no futher memory access after a fetch. The byte stored in the Memory Byte Latch and a zero byte from the Zero High/Sign Extended is transfered to the Aluin Latch and then to the RALU (Figure 6-13). At the end of the transfer the Progran counter is incremented by one for the next firmware instruction to be executed.

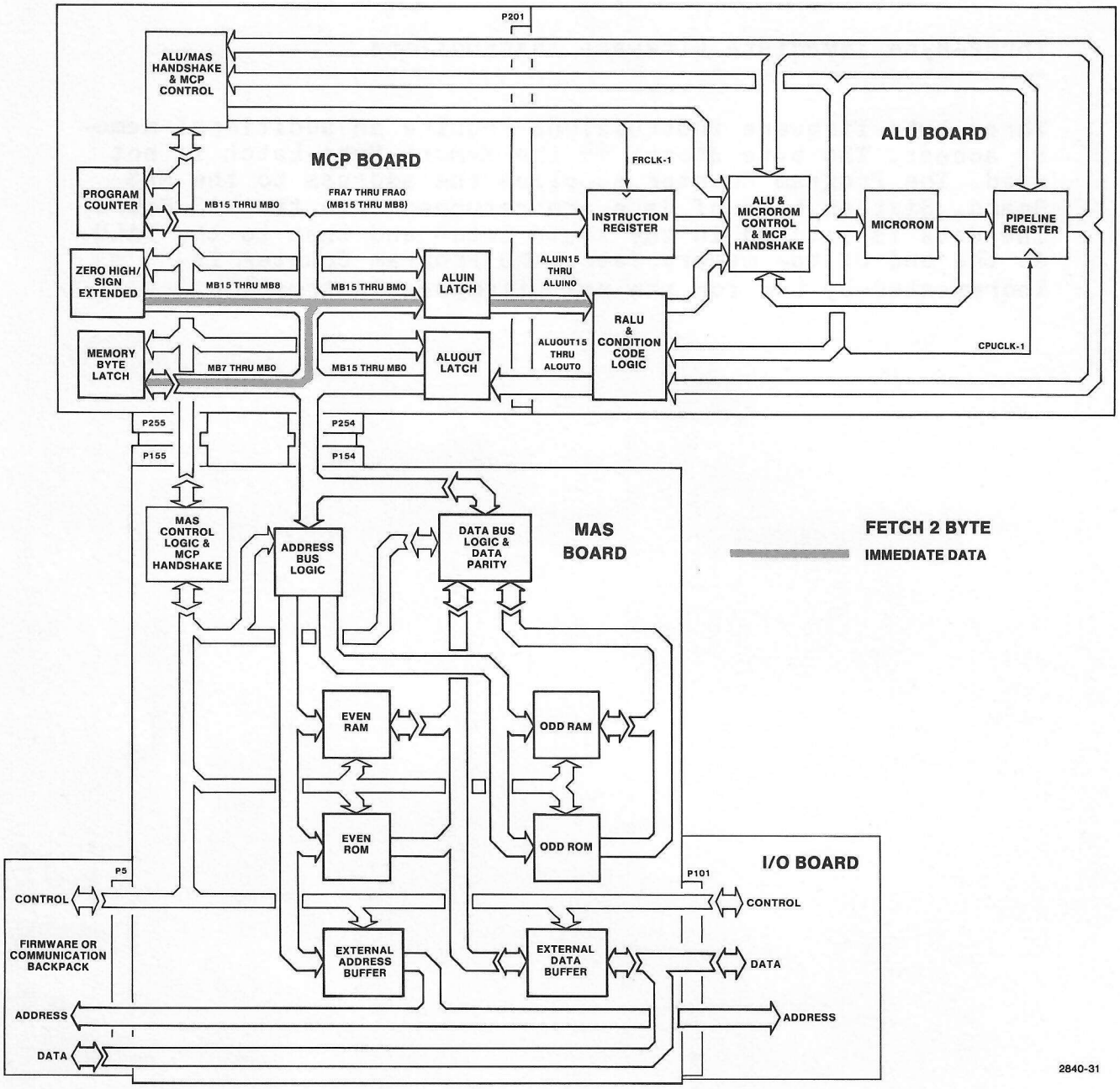


Figure 6-13. 4052/4054 Two-Byte Immediate Firmware Instruction.

## Three-Byte Immediate Firmware Instructions

Three-byte firmware instructions require an additional memory access. The byte stored in the Memory Byte Latch is not used. The Program Counter supplies the address to the MAS Board. Sixteen bits of data are returned from the MAS Board. The data is latched in the Aluin Latch and then to the RALU. At the end of the memory fetch the Program Counter is incremented by two for the next firmware instruction.

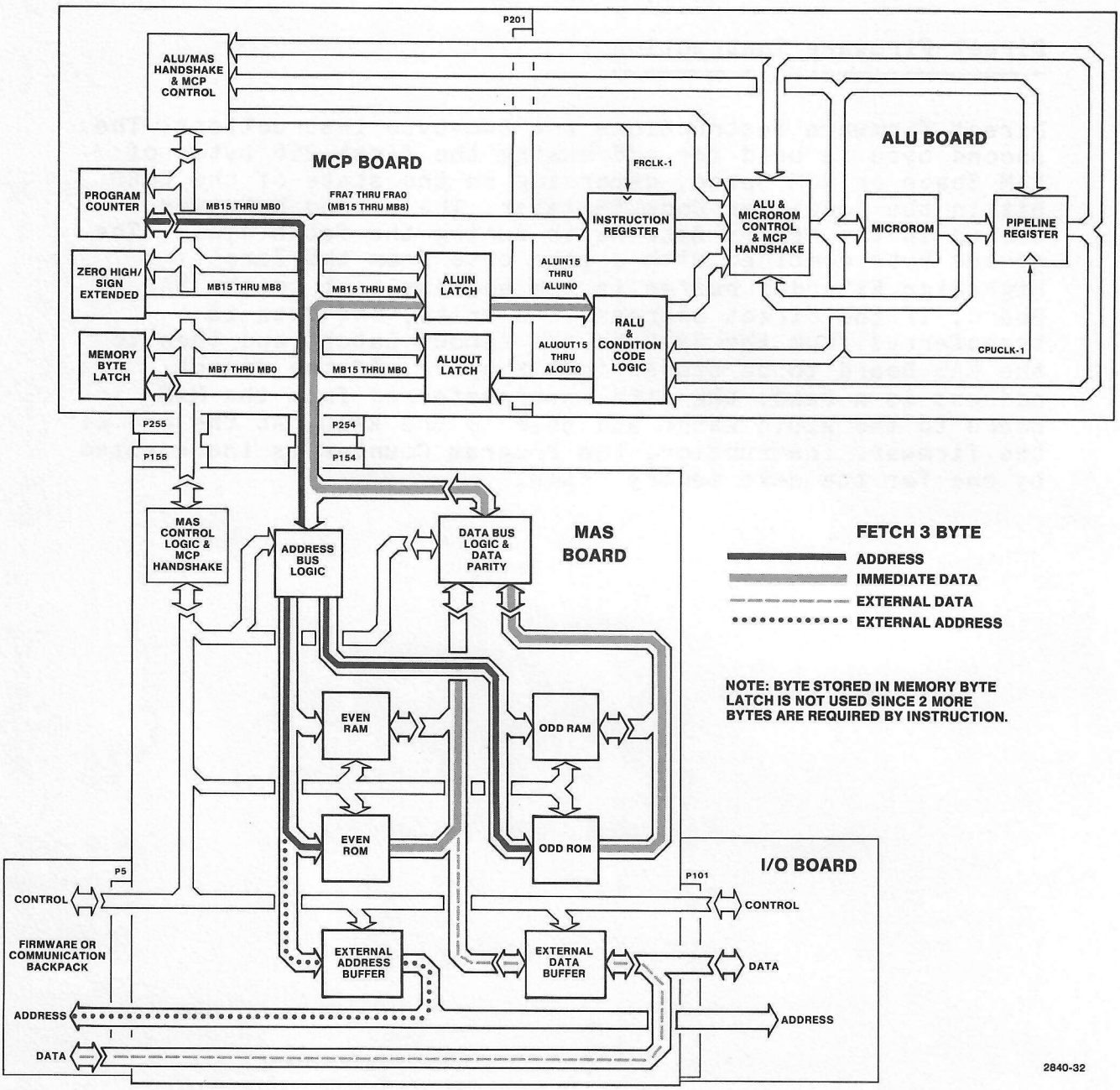


**Figure 6-14. 4052/4054 Three-Byte Immediate Firmware Instruction.**

## Direct Firmware Instruction

Direct firmware instructions are two-byte instructions. The second byte is used for addressing the first 256 bytes of RAM Space or ROM Space, depending on the state of the CCRD bit in the Condition Code Register. The second byte was stored in the Memory Byte Latch during the fetch cycle. The second byte combined with a zero byte from the Zero High/Sign Extended buffer is the address sent to the MAS Board. If the direct address is a write, the data is transferred from the RALU to the Aluout Latch, and then to the MAS Board to be stored in RAM space. If the direct address is a read, the data is transferred from the MAS Board to the Aluin Latch and then to the RALU. At the end of the firmware instruction, the Program Counter is incremented by one for the next memory fetch.

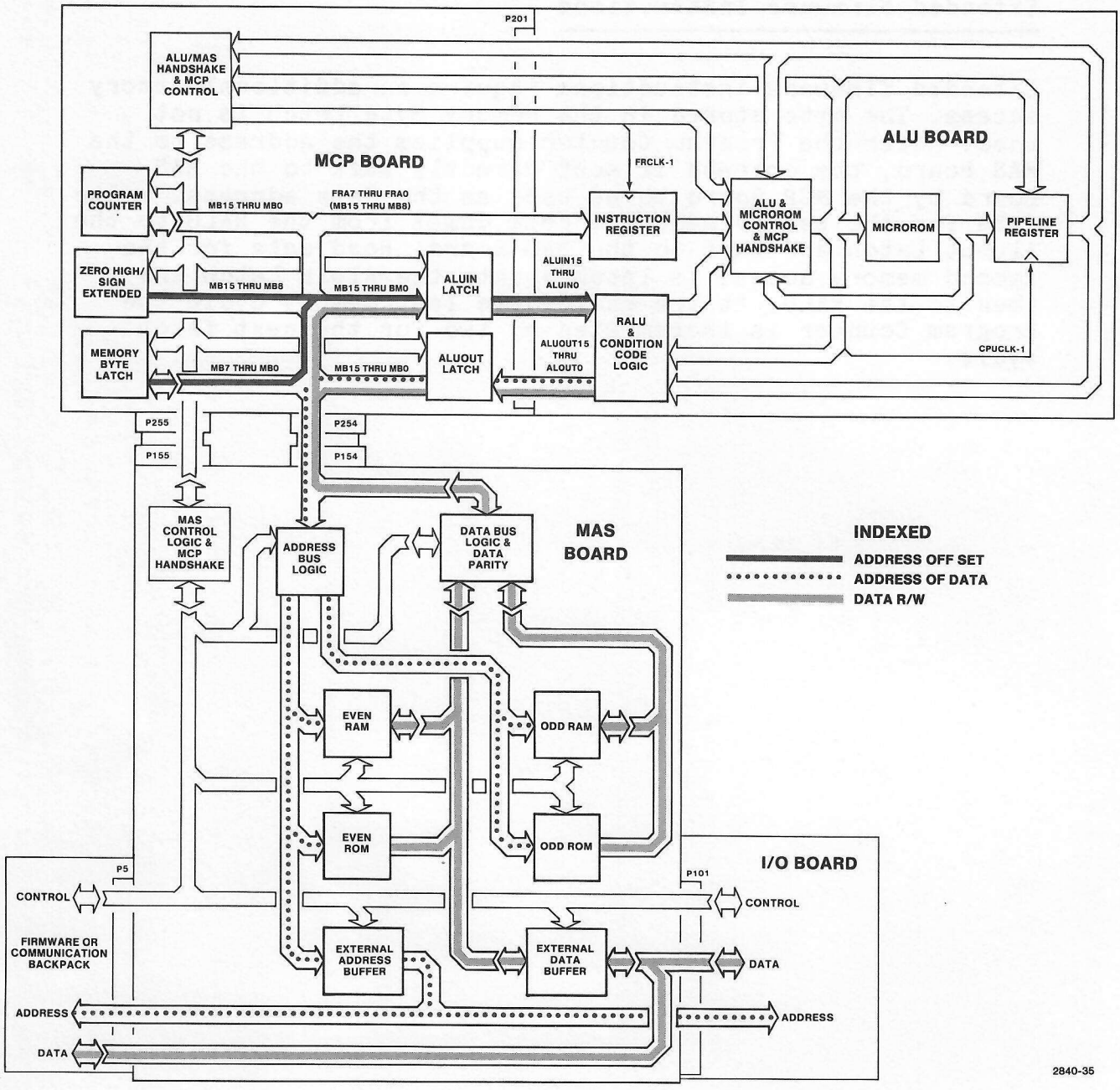


Figure 6-15. 4052/4054 Direct Firmware Instruction.

## Extended Firmware Instructions

Extended firmware instructions require an additional memory access. The byte stored in the Memory Byte Latch is not used. After the Program Counter supplies the address to the MAS Board, the operand is sent directly back to the MAS Board by the MCP Board to be used as the data address. Write data for the second memory access comes from the RALU to the Aluout Latch and then to the MAS Board. Read data for the second memory access is latched into the Aluin Latch and then to the RALU. At the end of the last memory cycle the Program Counter is incremented by two for the next fetch cycle.

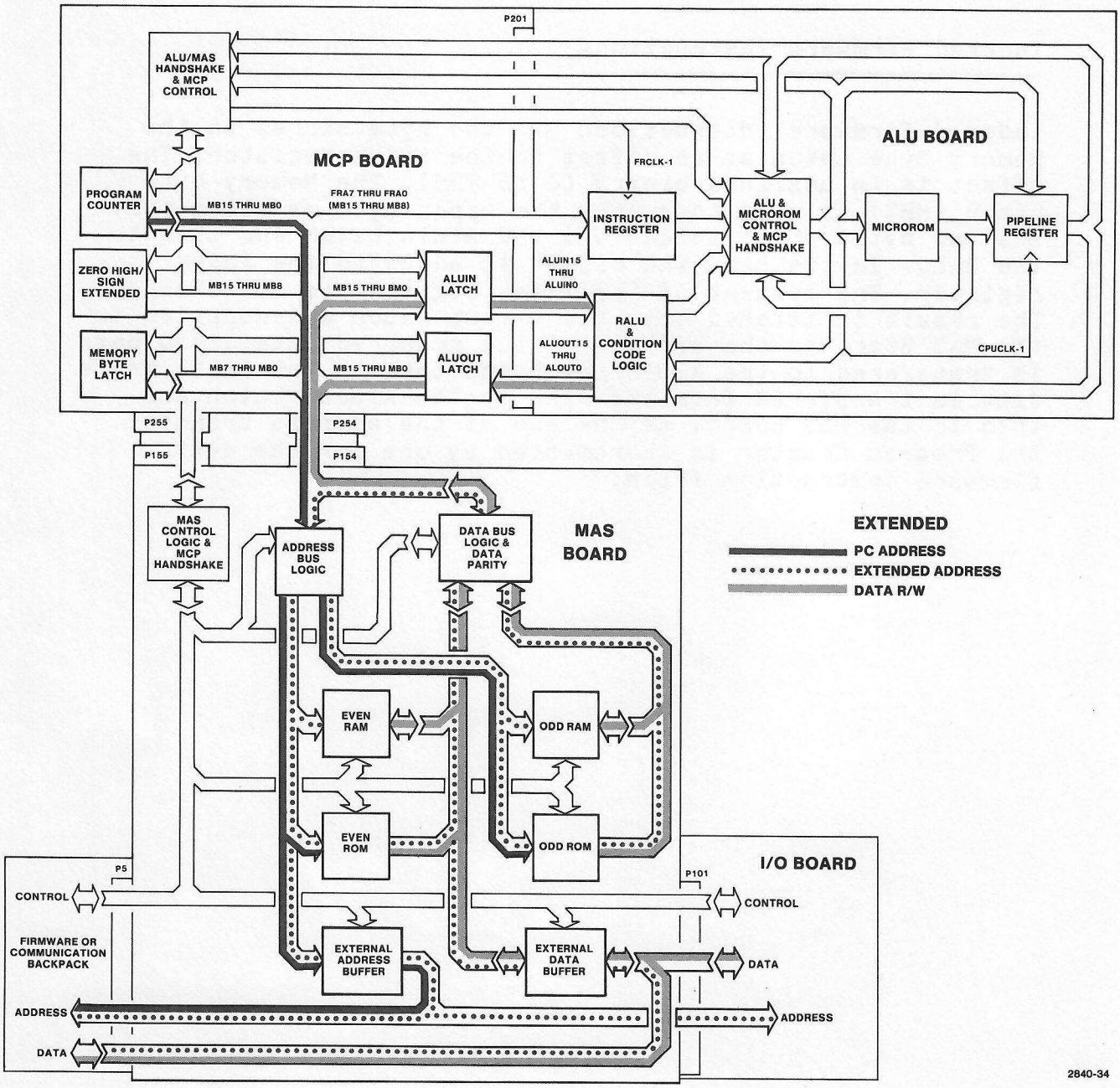


**Figure 6-16. 4052/4054 Extended Firmware Instruction.**

## Indexed Firmware Instructions

Indexed firmware instructions use the byte stored in the Memory Byte Latch as an offset to the index register. The offset is in unsigned binary (0 to 255). The Memory byte Latch (MB7) is used to supply the upper byte os all zeros. The two bytes are latched into the Aluin Latch and then to the RALU. In the RALU the offset is added to the index register. The content of the index register is not changed. The result is latched into the Aluout Latch and supplied to the MAS Board as the address of the required data. Read data is transfered to the Aluin Latch and then to the RALU. Write data is transfered from the RALU to the Aluout Latch and then to the MAS Board. At the end of the address transfer the Program Counter is incremented by one for the next firmware instruction fetch.

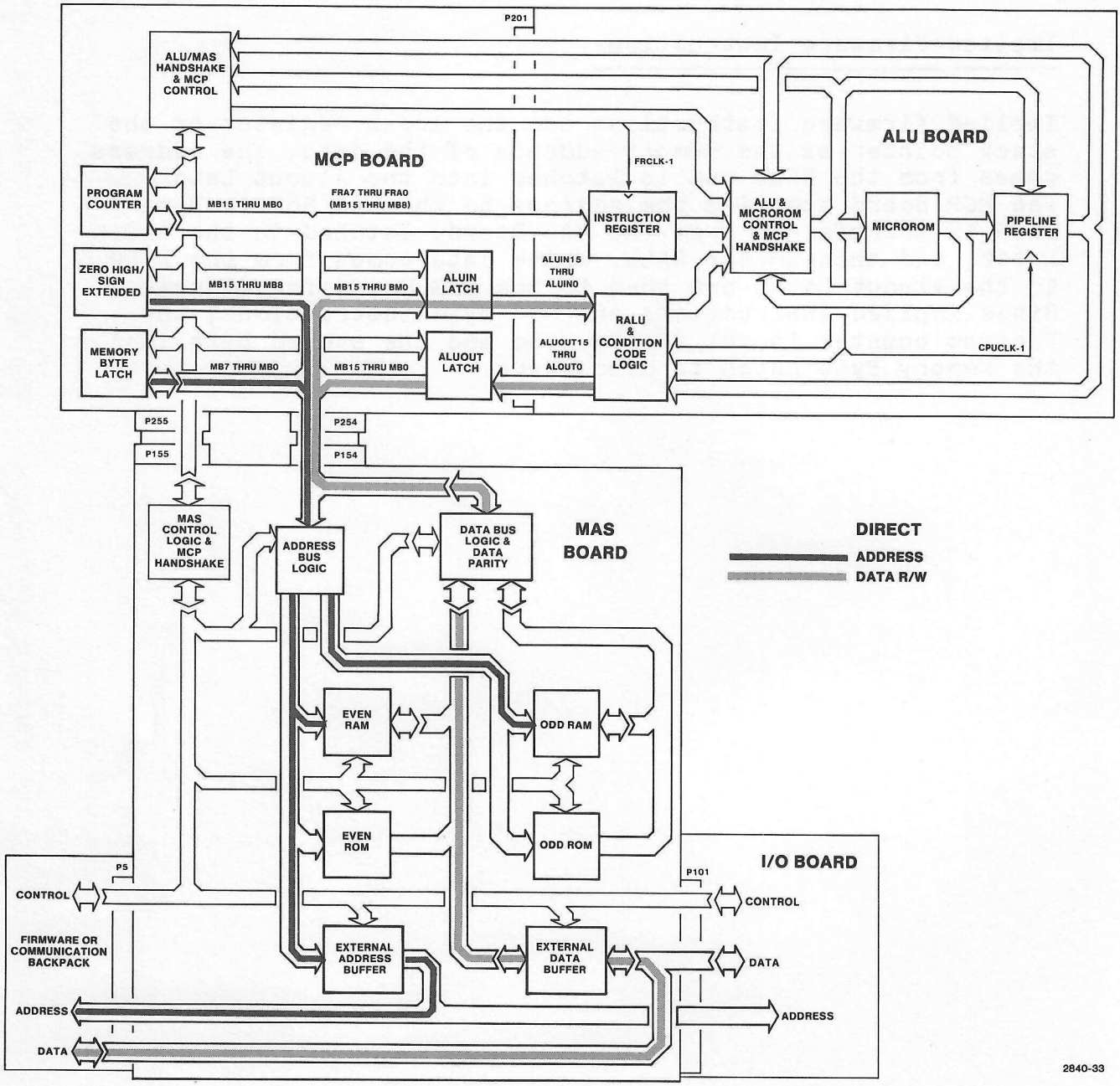


Figure 6-17. 4052/4054 Indexed Firmware Instruction.

## Implied Firmware Instructions

Implied firmware instructions use the index register or the stack pointer as the memory address of the data. The address comes from the RALU and is latched into the Aluout Latch. The MCP Board supplies the address to the MAS Board. Read data is transferred from the MAS Board, latched in the Aluin Latch, and then to the RALU. Write data comes from the RALU to the Aluout Latch and then to the MAS Board for storage. Since implied instructions are one byte instructions, the Program counter is not incremented and the stored byte in the Memory Byte Latch is discarded.

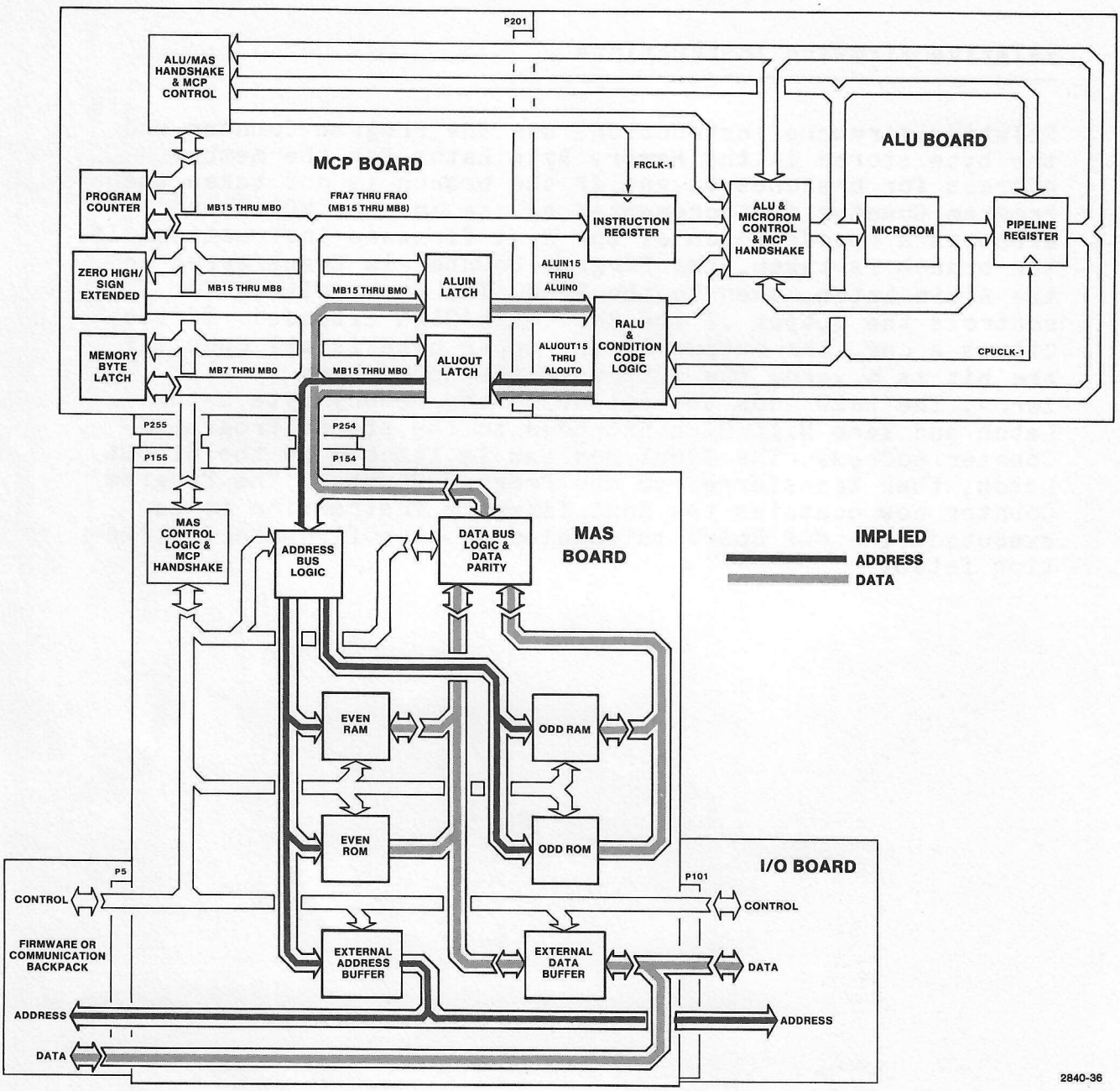


**Figure 6-18. 4052/4054 Implied Firmware Instruction.**

## Relative Firmware Instructions

Relative firmware instructions use the Program Counter and the byte stored in the Memory Byte Latch for the memory address for branches taken. If the branch is not taken the Program Counter is incremented by one and the MCP Board executes a opcode fetch of the next firmware instruction. If the branch is taken, the Program Counter is transferred to the Aluin Latch, then to the RALU. The upper bit (MB7) controls the output of the Zero High/Sign Extended. If the bit is a one, the output of the upper byte is all ones. If the bit is a zero, the output of the upper byte is all zeros. The RALU adds the output of the Memory Byte Latch Latch and Zero High/Sign Extended to the stored Program Counter address. The final address is latched in the Aluout Latch, then transferred to the Program Counter. The Program Counter now contains the next firmware instruction to be executed. The MCP Board is enabled to do a firmware instruction fetch.

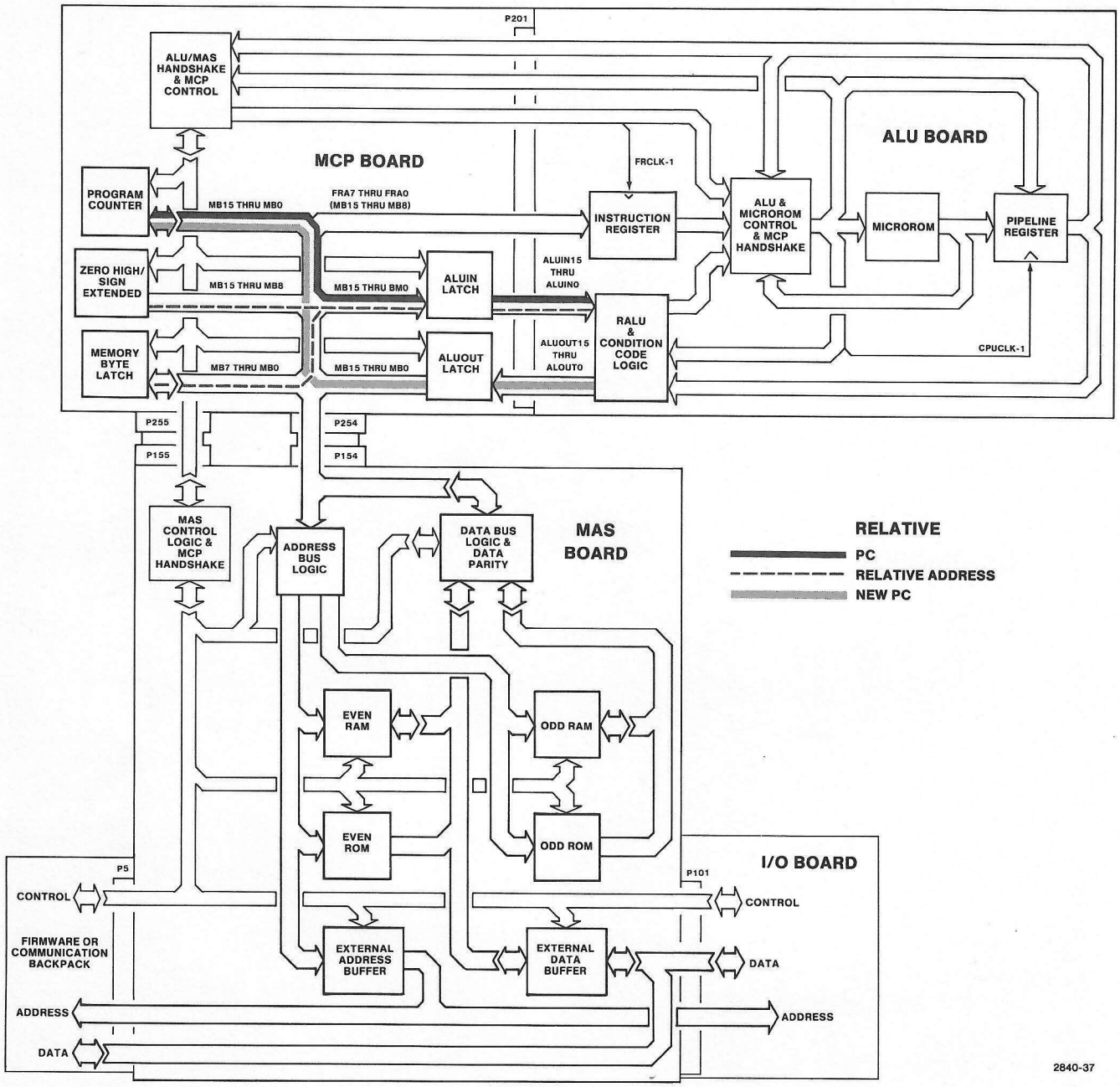


**Figure 6-19. 4052/4054 Relative Firmware Instructions.**