Address Decoding in the H89

Paul Laba
Revision 2.5
November 10, 2015

Abstract

Address Decoding in the Heath Company's H89 computer system refers to the process of decoding the Z80 processor's address lines during a memory read, write or refresh cycle, or an I/O read or write cycle, and selecting the correct memory or I/O device to process the request. Memory address decoding is handled in the H89 by the U516 and U517 address decoder ROMs; I/O address decoding is handled by the U550 I/O port decoder ROM. These parts are located on the H89's CPU logic board.

The use of ROMs (including PROMs, EPROMs, EEPROMS, etc.) for memory address and I/O port decoding is a fast and effective way to decode addresses and port numbers on systems that contain more than a few memory or I/O devices, as is the case with the H89. ROM decoders reduce circuit complexity, minimize the number of components required, and allow the designer to easily modify the decoding scheme by reprogramming or replacing the ROM.

This document is divided into two main parts. Part I describes the H89's memory address decoding. Part II describes the H89's I/O port decoding.

Three appendices are included at the end of this document. Appendix A lists the 256 states of the U517 memory bank decoder's A0-A7 address lines and the corresponding value of the D0-D7 data lines. Appendix B lists the 32 states of the U516 page decoder's A0-A4 address lines and the corresponding value of the D0-D7 data lines. Appendix C lists the 256 states of the U550 I/O port decoder's A0-A7 address lines and the corresponding value of the D0-D7 data lines.

Hobbyists interested in developing an H89-based emulator can hopefully use this document and its appendices to emulate the U516, U517 and U550 decoder ROMs. Doing so will provide a highly accurate emulation of the H89's memory and I/O decoder design and implementation, including the less obvious mirrored writes and ORG0 memory remapping features.

If you would like to comment on this document, or you discover any technical errors within (I'm sure there are many), please contact me at prlaba@comcast.net.

Paul Laba
October 2015

Part 1: Memory Address Decoding

This part describes in detail how the H89 decodes the Z80's A0-A15 address lines during a memory read, memory write or memory refresh request, and selects the appropriate memory device to service the request. It also explains how addresses that fall within the first 8K (bank 0) are remapped to different memory devices when ORG0 mode is enabled.

The Z80's 64K Address Space

The Z80 processor contains 16 address lines, A0 to A15, allowing it to address a maximum of 65,536 (64K) individual memory locations in the range 0x0000 to 0xFFFF.[1] These locations are typically distributed across multiple memory devices.

The H89 logically divides the 64K memory address space into eight 8K segments, called *banks*. The table below lists the eight 8K memory banks and their associated address range:

Bank	Size	Address Range
0	8K	0x0000 to 0x1FFF
1	8K	0x2000 to 0x3FFF
2	8K	0x4000 to 0x5FFF
3	8K	0x6000 to 0x7FFF
4	8K	0x8000 to 0x9FFF
5	8K	0xA000 to 0xBFFF
6	8K	0xC000 to 0xDFFF
7	8K	0xE000 to 0xFFFF

The H89 further divides each 8K memory bank into eight 1K segments, called *pages*. The table below lists the eight 1K pages within memory bank 0 and their associated addresses:

Page	Size	Address Range
0	1K	0x0000 to 0x03FF
1	1K	0x0400 to 0x07FF
2	1K	0x0800 to 0x0BFF
3	1K	0x0C00 to 0x0FFF
4	1K	0x1000 to 0x03FF
5	1K	0x1400 to 0x17FF
6	1K	0x1800 to 0x1BFF
7	1K	0x1C00 to 0xFFFF

[1] Hexadecimal notation is used throughout this document unless otherwise noted. If you're an Octal enthusiast, you'll have to do your own conversions.

A closer examination of the bank table above reveals that the Z80's three high-order address lines, A13, A14 and A15, form a 3-bit *bank address* that identifies which memory bank, 0 to 7, contains the 16-bit address. Similarly, the next three high-order address lines, A10, A11 and A12, form a 3-bit *page address* that identifies which page, 0 to 7, within the A13-A15 memory bank contains the 16-bit address.

For example, the 16-bit address 0x6A57 looks like this on the Z80's address lines:

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	1	1	0	1	0	1	0	0	1	0	1	0	1	1	1

Bank Page

The three high-order bits, 011B, indicate that this address falls within bank 3. The next three high-order bits, 010B, indicate that the address falls within page 2. So this address maps to a memory location in bank 3, page 2.

The H89 Memory Devices

The H89 contains a variety of memory devices, each handling its own range of addresses. Some memory devices, such as the U518 system ROM, U520 floppy disk ROM and U522/U524 floppy disk RAM, are reserved for use by the operating system and are collectively referred to as *system memory*. System memory devices always occupy addresses within bank 0 (0x0000 to 0x1FFF).

Other memory devices, in the form of individual 16K dynamic RAM (DRAM) devices, are available for general use by both the operating system and user programs. These devices are collectively referred as *user RAM*. User RAM occupies addresses within banks 1 through 7 (addresses 0x2000 to 0xFFFF).

The table below lists the H89's eight memory banks and their use:

Bank	Address Range	Usage
0	0x0000 to 0x1FFF	System memory (8K)
1	0x2000 to 0x3FFF	User RAM (56K) [2]
2	0x4000 to 0x5FFF	
3	0x6000 to 0x7FFF	
4	0x8000 to 0x9FFF	
5	0xA000 to 0xBFFF	
6	0xC000 to 0xDFFF	
7	0xE000 to 0xFFFF	

[2] By default only 56K of the maximum 64K user RAM is accessible. The H89 supports a feature called *ORG 0 mode* that allows an operating system to access the full 64K of user memory. ORG 0 mode is explained later.

Physically, user RAM consists of one to four 16K DRAM devices.[3] The first three, *user RAM 0*, *user RAM 1* and *user RAM 2*, are physically located on the CPU logic board. The fourth is an optional 16K memory expansion card that plugs into the P503/P509 accessory card connectors on the CPU logic board. All H89 systems were shipped with at least 16K of user RAM installed; users could then purchase and install one or two additional 16K user RAM kits to increase the on-board user RAM to 32K or 48K, or purchase and install the 16K memory expansion card to increase user RAM to its full 64K capacity.

Here's the same table as above, this time listing the four user RAM devices:

Bank	Address Range	Device
0	0x0000 to 0x1FFF	System memory (8K)
1	0x2000 to 0x3FFF	User RAM 0 (16K)
2	0x4000 to 0x5FFF	
3	0x6000 to 0x7FFF	User RAM 1 (16K)
4	0x8000 to 0x9FFF	
5	0xA000 to 0xBFFF	User RAM 2 (16K)
6	0xC000 to 0xDFFF	
7	0xE000 to 0xFFFF	16K memory expansion card (8K)
		16K memory expansion card (8K)

Note that the User RAM 0, User RAM 1 and User RAM 2 devices each occupy two contiguous memory banks. More importantly, note that *only 8K of the 16K memory expansion card's memory is accessible*, because there are not enough banks in the Z80's 64K address space to address the card's full 16K of memory. In effect, half of the expansion card's 16K memory is "wasted." As we'll see later, the H89's *ORG0* remapping feature provides a way to access both 8K banks on the expansion card.

Configuring User RAM

Two jumpers on the CPU logic board, *JJ501* and *JJ502,* indicate how much user RAM is installed in the system. Each jumper can be set in its "0" or "1" position.

The table below shows the possible positions of the JJ501 and JJ502 jumpers and the installed user RAM devices indicated by each pair:

JJ502	JJ501	Size	User RAM Installed
0	0	16K	User RAM 0
0	1	32K	User RAM 0 and 1
1	0	48K	User RAM 0, 1 and 2
1	1	64K	User RAM 0, 1 and 2, plus the 16K memory expansion card

[3] The term *device* here refers to a set of eight MK4116 16K x 1-bit dynamic RAM (DRAM) chips wired together to operate as a single 16K x 8-bit device. We'll continue referring to these 16K DRAM devices without regard to their individual chips, unless otherwise noted.

Be aware that the JJ501 and JJ502 jumpers can be configured to indicate *less* user RAM than is actually installed. For example, the jumpers might be set to indicate that 16K of user RAM is installed, even though 48K of user RAM is physically present. However, the jumpers should not be configured to indicate *more* user RAM than is actually installed; doing so will likely cause operating system errors.

The U516 and U517 Memory Address Decoders

The CPU logic board contains two ROM (Programmable Read-Only Memory) chips, U516 and U517, which together decode the target address for all Z80 memory read, write or refresh requests and select the correct memory device(s) to service the request.[4]

The U517 ROM serves as a *memory bank address decoder*. It connects to the three highest Z80 address lines, A13-A15, and selects the memory device assigned to the 8K bank containing the target address. Besides memory bank decoding, the U517 decoder also performs *ORG0 mode* address remapping, as well as handling unmapped read and write requests (requests whose target address does not map to any installed memory device). The U517 decoder is selected whenever the Z80 CPU activates its /MREQ status line, indicating a memory read, memory write or memory refresh request (determined by the Z80's /RD, /WR and /RFSH status lines). Otherwise the U517 decoder is deselected, which deselects all memory devices in the system.

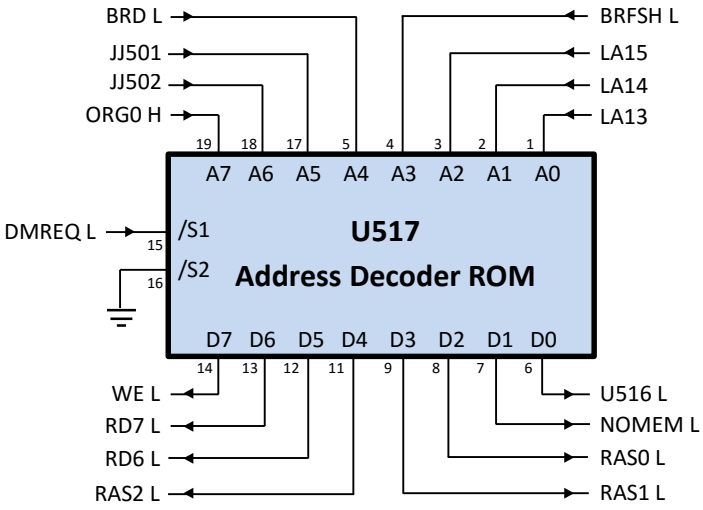
The U516 ROM serves as a *system memory page address decoder* for addresses that fall within bank 0 (0x0000 to 0x1FFF). It monitors the Z80's A10-A12 address lines and selects the system memory device (system ROM, floppy ROM, floppy RAM, optional system ROM or optional system RAM) assigned to the page in bank 0 containing the target address. The U516 decoder is selected and deselected by the U517 decoder. It is selected only when the Z80 issues a memory read or write request with a target address that falls within bank 0 and ORG0 mode is disabled. Otherwise the U516 decoder is deselected, which deselects all of its system memory devices.

The next two sections examine the U517 and U516 decoders in more detail.

[4] We'll see shortly why a single request may cause more than one memory device to be selected.

The U517 Address Decoder

The U517 address decoder is a 256 x 8-bit ROM consisting of eight address lines (A0 to A7), eight data lines (D0 to D7) and two select lines (/S1 and /S2). The diagram below shows the U517 ROM's address, data and select lines:



Note: A "/" prefix or "L" suffix on a line's label indicates that the line uses inverted logic (0 when the line is active, 1 when the line is inactive). An "H" suffix (or no "/" prefix or "H" or "L" suffix) indicates positive logic (1 when the line is active, 0 when the line is inactive).

The decoder's address lines are connected to various signals on the CPU logic circuit board, such as the Z80's latched A13, A14 and A15 address lines (labeled `LA15`, `LA14` and `LA13`) and the Z80's buffered /RD and /RFSH status lines (`BRD L` and `BRFSH L`). The decoder's data lines are used to select and deselect various memory devices, including the U516 decoder ROM, user RAM 0, RAM 1 and RAM 2, the 16K memory expansion card, and the special U521 "no memory" chip.

The U517 ROM has two active/low select lines, /S1 and /S2. Both lines must be active (0) to select the device. Because the /S2 line is tied to ground, the /S1 line alone selects and deselects the chip. The /S1 line is connected to `DMREQ L`, the delayed Z80 /MREQ status line.[5] `DMREQ L` is active (0) when the Z80 issues a memory read, memory write, or memory refresh request; otherwise DMERQ L is inactive (1).

When the U517 decoder ROM is selected (DMERQ L=0), the contents of the cell addressed by its A0-A7 address lines is transferred to its D0-D7 data lines. Memory devices connected to an active data line (0) are selected; memory devices connected to an inactive data line (1) are not selected (deselected).

[5] The internal timing of the Z80 is such that, during T1 of a Memory Read cycle, the /MREQ status line may become active *before* the /RD status line. For this reason the CPU logic board adds a 100 ns delay to the buffered /MREQ status line before routing it to the U517's /S1 select line. This ensures that all of the signals on the U517's address lines have "settled" by the time the ROM is selected.

Whenever the U517 decoder is deselected (/S1=1), all of its data lines are inactive (1), effectively deselecting all connected memory devices, including the U516 decoder ROM.

The table below lists each of the U517 decoder's eight address lines, its connection to other components on the H89 CPU logic board, and a description of its function:

Line	Pin	Connects To	Function
A0 (LA13)	1	The Z80's latched A13 address line (U511 pin 15).	These three lines indicates which bank (0 to 7) contains the target address.
A1 (LA14)	2	The Z80's latched A14 address line (U511 pin 16).	
A2 (LA15)	3	The Z80's latched A15 address line (U511 pin 19).	
A3 (BRFSH L)	4	The Z80's buffered /RFSH status line (U509 pin 18).	If this line is active (0), the Z80 has issued a memory refresh request.[6] If this line is inactive (1), the A4 address line indicates the type of memory request (read or write).
A4 (BRD L)	5	The Z80's buffered /RD status line (U509 pin 12).[7]	If this line is active (0) and A3 is inactive (1), the Z80 has issued a memory read request. If this line is inactive (1) and A3 is inactive (1), the Z80 has issued a memory write request. If the A3 address line is active (0), this line is ignored.
A5 (JJ501)	17	The JJ501 jumper's center pin.	These two lines indicates the amount of user RAM installed (16K, 32K, 48K or 64K).
A6 (JJ502)	18	The JJ502 jumper's center pin.	
A7 (ORG0 H)	19	The latched *Q5* output line from the General Purpose Port (U552 pin 15).	If this line is active (1), ORG0 mode is enabled; otherwise ORG0 mode is disabled. ORG0 mode is described in detail in a later section.

[6] Because the Z80 never activates its /RD and /RFSH status lines at the same time, the U517 decoder's A3 and A4 address lines are never both active (0) at the same time. Nonetheless, the U517 decoder ROM is programmed so that addresses of the form xxx00xxxB (A3=0, A4=0) produce the same results as xxx10xxxB (A3=0, A4=1).

[7] The U517 does not make use of the Z80's buffered /WR status line; the buffered /RD and /RFSH status lines are sufficient to determine the type of memory operation (read, write or refresh).

The table below lists each of the U517 decoder's eight data lines its connection to other components on the H89 CPU logic board, and a description of its function:

Line	Pin	Connects To	Function
D0 (U516 L)	6	The /CS (Chip Select) line of the U516 decoder ROM.	When this line is active (0), the U516 decoder ROM is selected. Otherwise the ROM is not selected.
D1 (NOMEM L)	7	The /1G and /2G select lines of the U521 "no memory" chip (see below).	When this line is active (0), the U521 chip is selected, which returns the fixed value 0x00 on the Z80's data lines. Otherwise the U521 chip is deselected.
D2 (RAS0 L)	8	The /RAS (Row Address Strobe) line of the U526-U533 user RAM 0 chips.	When this line is active (0), user RAM 0 (0K-16K) is selected. Otherwise User RAM 0 is deselected.
D3 (RAS1 L)	9	The /RAS (Row Address Strobe) line of the U534-U541 user RAM 1 chips.	When this line is active (0), user RAM 1 (16K-32K) is selected. Otherwise User RAM 1 is deselected.
D4 (RAS2 L)	11	The /RAS (Row Address Strobe) line of the U542-U549 user RAM 2 chips.[8]	When this line is active (0), user RAM 2 (32K-48K) is selected. Otherwise User RAM 2 is deselected.
D5 (RD6 L)	12	Pin 19 of the P507, P508 and P509 accessory card connectors.	When this line is active (0), the 16K memory expansion card (plugged into connector pair P503/P509) is selected. Otherwise the device is deselected.
D6 (RD7 L)	13	Pin 20 of the P507, P508 and P509 accessory card connectors.	This line was provided for future expansion. The latest U517 ROM, part 444-66, never activates its D6 data line. If such an accessory card was installed and selected via RD7 L, an updated U517 ROM part would be needed.
D7 (WE L)	14	The /WE (write enable) lines of user RAM 0, 1 and 2 (U526-U549), the 16K memory expansion card, and the /WE Lines of the U522/U524 optional system RAM chips.	When this line is active (0), it enables writing on the selected device (a write operation). When inactive (1), it disables writing on the selected device (a read operation).

Each of the data lines, with the exception of the D7 (WE L) data line, selects and deselects an individual memory device.

[8] The D4 data line is actually connected to the center pin of jumper JJ503 (JJ504 on older CPU logic boards). In its normal "B" position, JJ503 routes the D4 data line's signal to the /RAS line of the user RAM 2 chips. In its "A" position, jumper JJ503 routes the signal instead to pin 18 (labeled RD5 L) of the P507/P508/P509 accessory connectors, effectively "overriding" user RAM 2. As far as I know, no Heath accessory card ever utilized the RD5 L signal, so the JJ504 jumper is always in its "B" position.

An Example: Reading from User RAM 0

An example will help illustrate the U517 decoder's operation.

Suppose the Z80 issues a memory read request during T1 of a Memory Read cycle, by activating (0) its /MREQ and /RD status lines. The target address is 0x2000 (the lowest address in User RAM 0). Assume that ORG0 mode is disabled, and that the JJ501 and JJ502 jumpers are both in their "1" position, indicating that 64K of user RAM is installed. When the `DMREQ L` signal line becomes active (0), the U517 decoder ROM is selected. At this point the decoder's address lines contain the value 0x69:

A7	A6	A5	A4	A3	A2	A1	A0
0	1	1	0	1	0	0	1

- Latched Z80 A13 address line (1)
- Latched Z80 A14 address line (0)
- Latched Z80 A15 address line (0)
- Buffered Z80 /RFSH status line (inactive)
- Buffered Z80 /RD status line (active)
- JJ501 jumper position (1)
- JJ502 jumper position (1)
- ORG 0 mode (disabled)

The cell at address 0x69 is read and its contents, 0xFB, is returned on the D0-D7 data lines:

D7	D6	D5	D4	D3	D2	D1	D0
1	1	1	1	1	0	1	1

- Deselects U516 decoder ROM
- Deselects U521 "no memory" chip
- **Selects User RAM 0**
- Deselects User RAM 1
- Deselects User RAM 2
- Deselects 16K memory expansion card (RD6)
- Deselects future memory card (RD7)
- **Indicates a read operation**

As this example illustrates, using a ROM to decode a target address and map it to the correct memory device is simple and fast. The U517 decoder ROM's eight address lines provide for 256 individual input "states." Each state is the address of a cell within the ROM whose contents are returned on the decoder's eight data lines. Those lines are used to select and deselect their connected memory devices, and indicate the type of memory operation to perform (read or write).

Decoding Unmapped Memory Addresses

If the U517 decoder's A3 address line (`BRFSH L`) is 1 and the A4 address line (`BRD L`) is 0, then the Z80 has issued a memory read request. If the *A5* and *A6* address lines (JJ501, JJ502) indicate that no user RAM is installed at the memory bank specified by the decoder's *A0*, *A1* and *A2* address lines, then the decoder's D1 data line (`NOMEM L`) is activated.

This selects the U521 "no memory" chip, a 74LS240 octal inverter/driver chip with 3-state outputs. All eight input lines of the U521 are tied to +5V. Thus, when the U517 decoder selects the U521 chip by activating (0) its tied-together `1G` and `2G` lines (each select line controls four of the eight inverters) which in turn causes the chip to load its inverted output lines (0) onto the Z80's data lines.

For example, suppose the *A5* and *A6* address lines are both 0, indicating that 16K of user RAM is installed. In this configuration, all addresses that fall within banks 3 through 7 are unmapped. If the Z80 issues a memory read request with a target address that falls within any of these unmapped banks, the decoder selects the U521 chip, which returns the value 0x00 on the Z80's data lines.

This feature provides compatibility with Heath's earlier H8 computer, where the Z80's address and data lines were inverted on the CPU board. The H8 and H89 system monitor code makes use of this feature to locate the highest valid address in user RAM. They do this by writing a non-zero value to each possible user RAM address, then reading back the same address. If the returned value is zero (the result of an unmapped memory read) then the monitor knows that no RAM is installed at that address.[9]

Note that the U521 "no memory" device is selected for unmapped memory *reads* only. If the Z80 issues a memory *write* request with an unmapped target address, the U517 decoder simply deactivates *all* of its data lines, effectively deselecting all memory devices and ignoring the write request.

Note also that selection of the U521 "no memory" device is based solely on the JJ501 and JJ502 jumpers, regardless of how much user RAM is physically installed. For example, suppose the JJ501 and JJ502 jumpers indicate that 32K of user RAM is installed, but only the first 16K of user RAM is actually installed (user RAM 0). A memory read request whose target address falls within the second 16K of user RAM (user RAM 1) will *not* cause the U517 decoder to select the U521 "no memory" device, since the JJ501 and JJ502 jumpers indicate that memory is present. Instead, the decoder will activate its D3 data line, selecting the (non-existent) 16K user RAM 1 device. Since the user RAM 1 device is not physically present to respond to the Z80's read request, the request will go "unserviced."[10]

Selecting a DRAM Device

Each 16K user RAM device is comprised of eight 16K x 1-bit DRAM chips. Reading and writing DRAM is more complex than reading and writing static RAM, because DRAMs must be addressed using separate row and column addresses. As such, the number of address lines on a DRAM chip is generally *half* the number of lines needed to address the equivalent number of cells on a static RAM or ROM chip.

For example, the MK4116 16K x 1-bit DRAM chips that make up the H89's User RAM organizes its cells as a table of 128 rows and 128 columns (128 x 128 = 16384). Seven address lines, A0 through A6, are required to specify a row or column address in the range 0 to 127.

Because a DRAM requires separate row and column addresses, those addresses must be loaded onto the DRAM's address lines in two separate steps (a process called *address multiplexing)*.

[9] Note that this has nothing to do with how much memory is physically installed on the system, only the positions of the JJ501 and JJ502 jumpers.

[10] It does not appear that the CPU logic board uses pull-up resistors on its data bus. As a result, the value "read" by the Z80 at the end of an unserviced read request is indeterminate.

DRAM chips provide two control lines for loading the row and column addresses: a /RAS (*Row Address Strobe*) line to latch the row address on the chip's address lines into its internal row memory decoder, and a /CAS (*Column Address Strobe*) line to latch the column address on the address lines into its internal column memory decoder. In a typical read or write operation, the row address is first loaded onto the chip's address lines and its /RAS line activated (0). At this point the chip's internal row memory decoder contains the row address of the cell to be read or written. With the /RAS line still active, the column address is then loaded onto the chip's address lines and its /CAS line activated (0), latching the columns address. At this point the chip's internal row and column memory decoders contain the complete row and column address of the target cell, so the chip then proceeds to read or write the addressed cell. Once the read or write operation completes, the /RAS and /CAS lines can be deactivated (1).

The U517 decoder controls only the /RAS lines of the 16K user RAM chips. A separate circuit, called the *Dynamic RAM Address Multiplexer* (*Mux* for short), is responsible for loading the row and column addresses from the latched Z80 address lines onto each chips' address lines, and activating and deactivating the chips' /CAS lines.[11] Timing between the U517 decoder and the Mux circuit is obviously critical; the /RAS lines, controlled by the U517 decoder, and the /CAS lines, controlled by the Mux circuit, must be activated and deactivated at precisely the right time and in the right sequence.

In the context of this document, when the U517 decoder "selects" a User RAM device by activating its /RAS line, we will consider that step sufficient to trigger the read or write operation, even though the Mux circuit must also load the correct row and column addresses and activate the device's /CAS line as part of the same operation.

Refreshing Dynamic RAM

Refreshing a DRAM chip does not require loading a column address onto its address lines or activating its /CAS line; loading a row address onto the DRAM's address lines and activating its /RAS line is sufficient to refresh all of the cells within the addressed row. As such, during a memory refresh cycle the Mux circuit mentioned above loads a row address from the latched Z80 address lines onto the DRAM's address lines, but does *not* load a column address nor activate the DRAM's /CAS line.

Unlike memory read and write requests, where the U517 decoder selects a single target device, memory refresh requests cause the U517 decoder to select *all* installed user RAM devices.

If the Z80 issues a memory refresh request — the decoder's A3 address line is active (0) — the U517 decoder activates (0) *all* of the /RAS lines connected to *installed* User RAM chips, as determined by the decoder's A5 (JJ501) and A6 (JJ502) address lines, without regard to its A0, A1 and A2 address lines. In this case the decoder's D0 (`U516 L`) and D1 (`NOMEM L`) data lines are inactive (1), as are the data lines connected to any uninstalled User RAM chips.

For example, suppose the system contains 32K of user RAM (A5=1, A6=0). When the Z80 issues a memory refresh request, the A3 (`BRFSH L`) address line is 0. When the U517 decoder is selected, it activates its D3 (`RAS0 L`) and D4 (`RAS1 L`) data lines, selecting both user RAM 0 and user RAM 1. No other devices are selected.

[11] The Mux circuit operates on *all* DRAM chips' address lines and /CAS lines at the same time, without regard to which DRAM might be selected by the U517 decoder during a memory read or write operation. That's not a problem, since the read or write operation is not triggered until/unless *both* the /RAS and /CAS lines are activated.

Similarly, if the system contains 64K of user memory (*A5*=1, *A6*=1), a memory refresh request causes the decoder to activate its D3, D4, D5 and D6 data lines, selecting user RAM 0, user RAM 1, user RAM 2 and the 16K memory expansion card.

Writing to Bank 0 (System Memory)

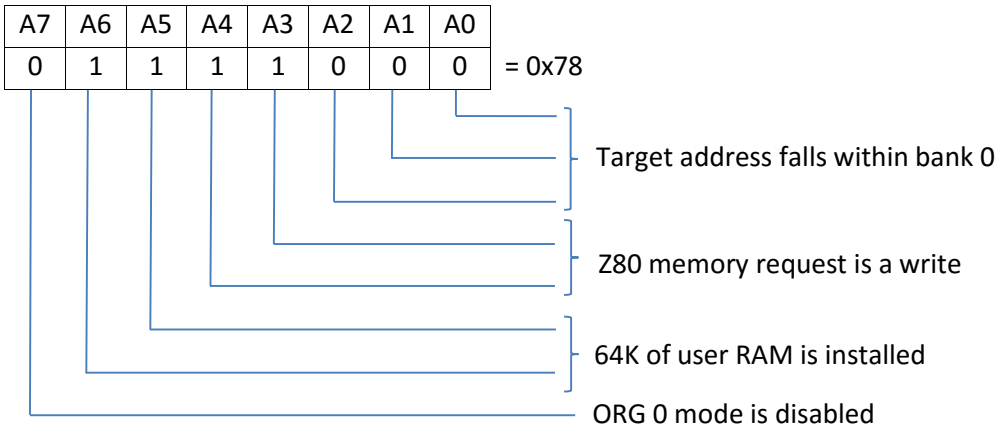
One of the most surprising behaviors of the U517 decoder is its handling of memory write requests with a target address that falls within bank 0 (0x0000 to 0x1FFF). As described earlier, such requests cause the U517 decoder to activate its D0 (`U516 L`) and D7 (`WE L`) data lines, essentially routing the write request to the U516 decoder for further decoding and processing. The surprise is that, on systems containing 64K of user RAM (jumpers JJ501 and JJ502 are both) the U517 decoder *also activates its D5 (*`RD6 L`*) data line (0), selecting the 16K memory expansion card* (!).

This means that the U517 decoder not only routes a bank 0 write request to the U516 decoder, it also writes to the 16K memory expansion card. In effect, *any* write to a bank 0 target address is "mirrored" to a cell in the 16K memory expansion card.

This "mirrored write" behavior only occurs when *all* of the following conditions are true:

- The Z80 issues a memory write request.
- The request's target address falls within bank 0 (0x0000 to 0x1FFF).
- 64K of user RAM is installed (JJ501=1, JJ502=1).
- ORG0 mode is disabled.

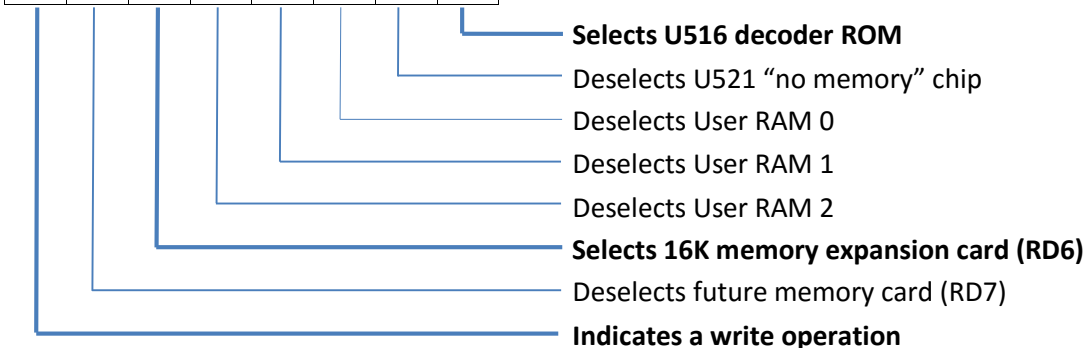
The above conditions are met when the U517 decoder's eight address lines are in this one state:



The contents of address 0x78 on the U517 decoder's data lines look like this:

D7	D6	D5	D4	D3	D2	D1	D0
0	1	**0**	1	1	1	1	**0**

= 0x5E



Before answering the "why" question for this behavior, let's first consider the "how" question; specifically, which cells on the 16K memory expansion card are written to by this "mirrored write" operation.

Recall that a 16K memory device, including the 16K memory expansion card, requires 14 address lines (A0 to A13) to access all of its 16,384 memory locations. The high-order address line, A13, effectively "divides" the device's memory locations into two 8K segments. When the A13 address line is 0, the device's *lower* 8K segment is accessed. When the A13 address line is 1, the device's *upper* 8K segment is accessed.

Also recall that the U517 decoder selects the U516 decoder when the Z80's A13-A15 address lines are 0, indicating an address that falls within bank 0 (0x0000 to 0x1FFF).

Thus, when a mirrored write to the 16K memory expansion card occurs, the Z80's A13 address line is always 0, so the write operation will target a memory cell within the expansion card's *lower* 8K segment.

When a *normal* read or write to the 16K expansion card occurs (with ORG0 mode still disabled), the target address falls within bank 7 (0xE000 to 0xFFFF). For all values in that address range the A13 address line is 1, which means the read or write operation will target a memory cell within the expansion card's *upper* 8K segment.

That means any data written to the 16K expansion card as part of a mirrored write operation *cannot be read*, at least not while ORG0 mode is disabled, because the card's lower 8K memory segment is *unmapped* and is therefore not accessible during normal read and write operations that target bank 7.

From a functional perspective, writing to a "writeable" system memory device in bank 0, like floppy RAM or the optional system RAM device, results in *two* concurrent writes: one to a cell in the selected system RAM device and the other to a cell in the 16K memory expansion card's lower 8K memory segment. However, writing to a *read-only* system device, such as the system ROM or the floppy disk ROM, ignores the write to the system ROM device (obviously) but still writes to a cell in the expansion card's lower 8K memory segment.

Now for the "why" question. Some H89 enthusiasts have suggested this mirrored write feature is a "useful mistake." I consider it an intentional design. The CP/M operating system, the only Heath operating system that officially supports ORG0 remapping,[12] uses this feature to initially copy the contents of the 4K system ROM (MTR-90) to pages 0-3 of the

[12] A later, never-officially-released version of HDOS, version 3.0, also includes support for ORG 0 remapping.

lower 8K segment of the 16K memory expansion card, prior to enabling ORG0 mode. It does this by reading each byte in system ROM and writing its value back *to the same system ROM address*. The U517 decoder mirrors these writes to the same address in the lower 8K segment of the 16K memory expansion card. This effectively *relocates* the system ROM code to read/write memory on the expansion card. Once ORG0 mode is enabled, CP/M can then access the expansion card's lower 8K memory segment using the *same* system ROM addresses as before, as if it were still reading from system ROM. Because the relocated system ROM code now resides in read/write memory, CP/M is free to modify the code, add or delete routines, etc.

Missing from this explanation is the details of how ORG0 memory remapping actually works. We're finally ready to describe that feature in full detail.

ORG0 Memory Remapping

The U517 decoder's A7 address line indicates if *ORG0 mode* is enabled (1) or disabled (0). This address line is connected to the Q*5* output line of the U551/U552 General Purpose Port (360Q). Writing to port 360Q with the D5 data line set (xx1xxxxxB) enables ORG0 mode; writing to port 360Q with the D5 data line clear (xx0xxxxxB) disables ORG0 mode.[13]

When ORG0 mode is enabled, the U517 decoder's A7 address line is active (1). This causes the decoder to remap all target addresses within bank 0 (0x0000 to 0x1FFF). Instead of selecting the U516 decoder ROM, the U517 decoder instead selects one of the 16K user RAM devices (User RAM 0, User RAM 1, User RAM 2 or the H88-16 16K memory expansion card).

Which user RAM device is selected depends on the amount of user RAM installed, as indicated by the JJ501 and JJ502 jumpers (connected to the U517's A5 and A6 address lines).

In the following example, assume that the system contains 64K of user memory (the U517 decoder's A5 and A6 address lines are both 1), meaning all 48K of on-board user RAM is installed, plus the 16K memory expansion card.

With ORG0 mode disabled (the U517 decoder's A7 address line is 0), the decoder responds to read and write requests with a target address within bank 0 by activating its D0 data line (0), which in turn selects the U516 decoder ROM. That decoder in turn decodes the page address and maps it to one of the system memory devices in bank 0. Similarly the U517 decoder maps target addresses within banks 1 through 7 to one of the four 16K user RAM devices by activating its D2 (`RAS0 L`), D3 (`RAS1 L`), D4 (`RAS2 L`) or D5 (`RD6 L`) data lines.

When ORG0 mode is enabled (the A7 address line is 1), the U517 decoder behaves differently. It no longer selects the U516 decoder ROM in response to a memory read or write request with target address within bank 0. Instead, it activates its D5 (`RD6 L`) data line, selecting the 16K memory expansion card. In effect, all bank 0 target addresses are remapped to the lower 8K section of the 16K expansion card.

[13] When the H89 system is powered on or reset, all of the General Purpose Port's output lines are cleared, which disables ORG 0 mode.

The table below shows how the U517 decoder maps addresses when ORG0 mode is enabled on a system with 64K of user RAM installed:

Bank	Address Range	Read Activates	Write Activates	Maps To
0	0x0000 to 0x1FFF	RD6 L	RD6 L	16K expansion card (lower 8K)
1	0x2000 to 0x3FFF	RAS0 L	RAS0 L	User RAM 0 (16K)
2	0x4000 to 0x5FFF	RAS0 L	RAS0 L	
3	0x6000 to 0x7FFF	RAS1 L	RAS1 L	User RAM 1 (16K)
4	0x8000 to 0x9FFF	RAS1 L	RAS1 L	
5	0xA000 to 0xBFFF	RAS2 L	RAS2 L	User RAM 2 (16K)
6	0xC000 to 0xDFFF	RAS2 L	RAS2 L	
7	0xE000 to 0xFFFF	RD6 L	RD6 L	16K expansion card (upper 8K)

Note that, with ORG0 mode enabled, target addresses within banks 0 and 7 are mapped to the 16K expansion card, while addresses in banks 1-6 are mapped to the same user RAM as when ORG0 mode is disabled.

Also note that, with ORG0 mode enabled, all 64K of user RAM is accessible, but all bank 0 system memory devices are not.

ORG0 Remapping on "Small" Memory Systems

Next we'll consider how the U517 decoder ROM remaps bank 0 addresses on systems with *less* than 64K of user RAM installed. For example, suppose a system contains 32K of user memory (JJ502=0, JJ501=1). The table below shows the memory map with ORG0 mode enabled:

Bank	Address Range	Read Activates	Write Activates	Maps To
0	0x0000 to 0x1FFF	RAS1 L	RAS1 L	User RAM 1 (lower 8K)
1	0x2000 to 0x3FFF	RAS0 L	RAS0 L	User RAM 0 (16K)
2	0x4000 to 0x5FFF	RAS0 L	RAS0 L	
3	0x6000 to 0x7FFF	RAS1 L	RAS1 L	User RAM 1 (upper 8K)
4	0x8000 to 0x9FFF	NOMEM L	None	No-memory Device
5	0xA000 to 0xBFFF	NOMEM L	None	
6	0xC000 to 0xDFFF	NOMEM L	None	
7	0xE000 to 0xFFFF	NOMEM L	None	

In this configuration, the U517 decoder remaps target addresses within bank 0 to the lower 8K segment of User RAM 1, while addresses within bank 4, which previously mapped to the lower 8K segment of User RAM 1, are now unmapped. Target addresses within banks 1-3 are mapped the same as when ORG0 mode is disabled.

This means all read requests with a target address within bank 4 will now select the U521 "no memory" device, while all write requests with a target address within bank 4 will be ignored. In no case will ORG0 mode cause the U517 decoder to remap a target address to more than one memory device.

To continue our example: If we increase user RAM to 48K (with ORG0 mode still enabled), the U517 decoder will remap all target addresses within bank 0 to the lower 8K segment of User RAM 2, while addresses within bank 6, which previously mapped to the lower 8K segment of User RAM 2, are now unmapped.

Finally, consider the 'pathological' case of ORG0 mode with only 16K of user RAM installed. In this case the U517 decoder remaps target addresses within bank 0 to the lower 8K segment of User RAM 0, while addresses within bank 2, which previously mapped to the lower 8K segment of User RAM 0, are now unmapped, as are all addresses within banks 3-7

The table below shows the U517 decoder's ORG0 address remapping for different amounts of user RAM.

User RAM installed	Bank 0 addresses remapped to	Unmapped addresses
16K	Lower 8K of user RAM 0	Banks 2-7, including upper 8K of user RAM 0
32K	Lower 8K of User RAM 1	Bank 4-7, including upper 8K of user RAM 1
48K	Lower 8K of User RAM 2	Bank 6-7, including upper 8K of user RAM 2
64K	Lower 8K of memory expansion card	None

The beauty of the ORG0 remapping design is its simplicity: the U517 decoder performs ORG0 remapping simply by activating one of its User RAM select lines (`RAS0 L`, `RAS1 L`, `RAS2 L` or `RD6 L`) instead of selecting the U516 decoder. No actual manipulation or translation of the Z80's address lines is required.

Appendix A lists each of the U517 decoder ROM's 256 address line states (0x00 to 0xFF) and the corresponding data line states. The suffix '(ORG0 remap)' in the *Result* column indicates a case where the U517 decoder remaps a bank 0 address to a different memory device when ORG0 mode is enabled.

Memory Device Addressing

Each memory device in the H89 connects to a subset of the Z80's 16 address lines (A0 to A15). The device's size determines how many address lines are actually connected.[14]

For static memory devices like system ROM, floppy ROM and floppy RAM, the number of address line is a function of its size, in bytes, calculated as $Log_2(size)$. For example, the 2K system ROM requires 10 address lines ($Log_2(2048) = 10$), A0-A9, to access each of its individual memory locations. The 1K system RAM device requires one less address line ($Log_2(1024) = 9$), A0-A8, to access each of its individual memory locations.

As explained earlier, dynamic RAM devices like the H89's 16K user RAM 0, user RAM 1 and user RAM 2 chips, plus the 16K memory expansion card chips, contain half the number of address lines as a static memory device. For purposes of

[14] Memory devices' address lines are not actually connected to the Z80's address lines. Instead they are connected to the outputs of two 74LS73 octal D latch chips, U510 and U511, which latches the Z80's address lines whenever the Z80 activates its /MREQ or /IORQ status line. These lines are often labeled, LA0, LA1, … LA15 on schematics.

this discussion we'll treat each of the H89's User 16K RAM devices, which contain only 7 address lines, as if they contained the full complement of 14 address lines.

Because a memory device does not connect to all 16 of the Z80's address lines, its internal address — the address it "sees" on its own address lines — is equal to or less than the value of the target address available on the Z80's address lines. For example, the 2K system ROM device's internal address range is 0x000 to 0x0800; a 16K User RAM device's internal addresses range is 0x0000 to 0x5FFF.[15]

The U517 decoder selects a 16K User RAM device by activating its D2 (`RAM 0 L`), D3 (`RAM 1 L`), D4 (`RAM 2 L`) or D5 (`RD6 L`) data lines. When a data line is inactive (1), its corresponding device is deselected. When a data line is active (0), the device's address lines specify which memory location to access for the read or write operation.[16]

Consider how the U517 decoder remaps target addresses within banks 1 or 2 to User RAM 0 (assume that ORG0 mode is disabled). The table below lists several target addresses that fall within bank 1 or 2, and their corresponding address on the RAM 0 device's own address lines:

Z80 Address Lines	Bank	User RAM 0 Address Lines
0x2000	1	0x2000
0x3000	1	0x3000
0x3FFF	1	0x3FFF
0x4000	2	0x0000
0x5000	2	0x1000
0x5FFF	2	0x1FFF

If you examine the table closely, you'll notice that all of the addresses listed in the *Z80 Address Lines* column are in increasing order, from 0x2000 (the lowest address within bank 1) to 0x5FFF (the highest address within bank 2). But the corresponding addresses in the User *RAM 0 Address Lines* column do not always increase. Here's why:

When the value of the Z80's address lines is 0x2000 (the first address in bank 1), the U517 decoder selects User RAM 0 by activating its D2 (`RAS0 L`) data line. The value of the User RAM 0 device's address lines is 0x2000, the same as the Z80's address lines. So the first address "seen" by the User RAM 0 device is 0x2000, not 0x0000. As the Z80 addresses increase from 0x2000 to 0x3FFF, the User RAM 0 device's address lines match the Z80's address lines. But when the Z80 address lines change from 0x3FFF (the highest address in bank 1) to 0x4000 (the lowest address in bank 2), the User RAM 0 device's address lines change from 0x3FFF to 0x0000. That's because the Z80's A14 address line for the address 0x4000 is 1, but the A14 address line is not connected to the User RAM 0 device. As the Z80 addresses increase from 0x4000 to 0x5FFF, the value of the User RAM 0 device's address lines also increases, but from 0x0000 to 0x1FFF.

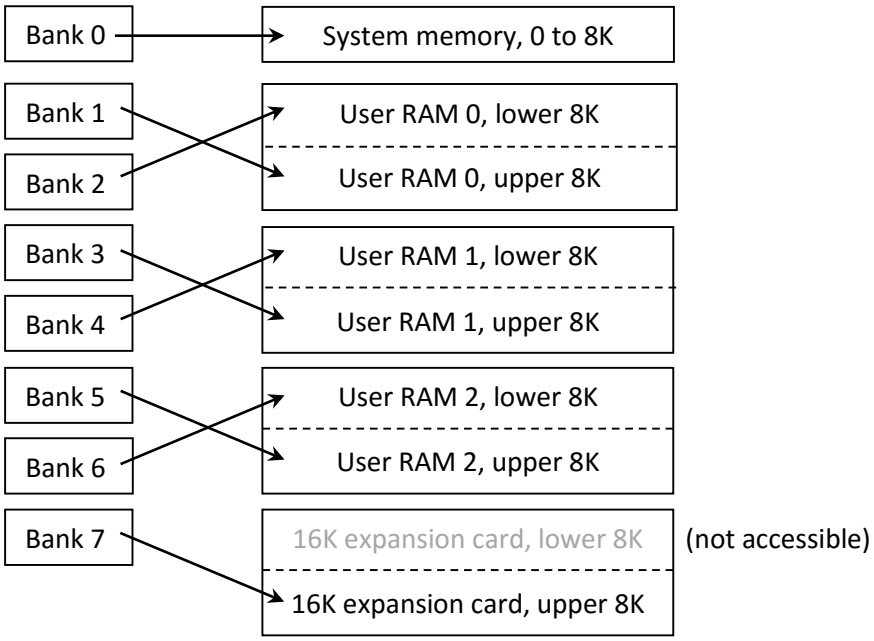
The result of this decoding/mapping is that, for each 16K User RAM device, the first 8K of Z80 addresses map to the user RAM device's *upper* 8K segment, while second 8K of Z80 addresses map to the device's *lower* 8K segment.

[15] A memory device's internal address range is 0 to n, where n is the device's size – 1.
[16] The actual process of reading or writing an individual memory location in a DRAM device was explained in an earlier section.

The table below shows this address mapping for all memory devices (ORG0 mode still disabled):

Bank	Z80 Address	Device Selected	Device Address Lines
0	0x0000 to 0x1FFF	System memory	0x0000 to 0x1FFF (8K)
1	0x2000 to 0x3FFF	User RAM 0	0x2000 to 0x3FFF (upper 8K)
2	0x4000 to 0x5FFF		0x0000 to 0x1FFF (lower 8K)
3	0x6000 to 0x7FFF	User RAM 1	0x2000 to 0x3FFF (upper 8K)
4	0x8000 to 0x9FFF		0x0000 to 0x1FFF (lower 8K)
5	0xA000 to 0xBFFF	User RAM 2	0x2000 to 0x3FFF (upper 8K)
6	0xC000 to 0xDFFF		0x0000 to 0x1FFF (lower 8K)
7	0xE000 to 0xFFFF	Memory Expansion Card	0x2000 to 0x3FFF (upper 8K)

Here's a graphical view of the above table, with 64K of user RAM installed and ORG0 mode disabled:



Note that bank 7 addresses map to the *upper* 8K segment of the 16K memory expansion card. The card's *lower* 8K segment is not accessible when ORG0 mode is disabled.[17]

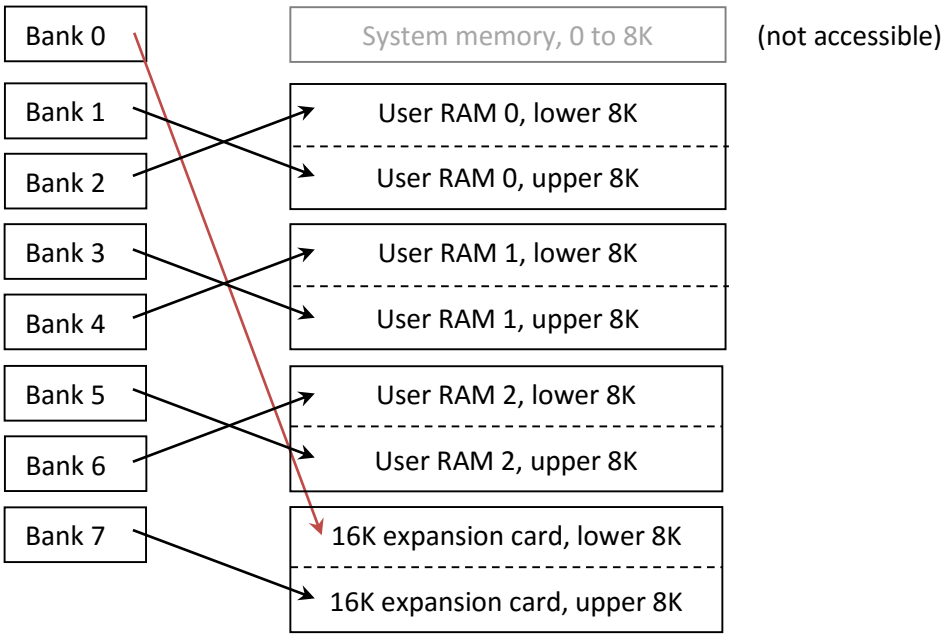
ORG0 Remapping Revisited

It is hopefully apparent now why ORG0 memory mapping does not require any manipulation or translation of the Z80's address lines. When ORG0 mode is enabled, any bank 0 address (0x0000 to 0x1FFF) causes the U517 decoder to deselect the U516 decoder and instead select the *highest installed User RAM device*. When that User RAM device is

[17] As explained earlier, the lower 8K segment of the 16K memory expansion card can still be written to with ORG 0 mode disabled, by writing to an address within bank 0 — a feature called *mirrored write*). However, the card's lower 8K segment cannot be *read* unless ORG 0 mode is enabled.

selected, its address lines match the Z80 address lines (0x0000 to 0x1FFF). In the case of the 16K memory expansion card, the Z80 address maps to the card's previously unmapped lower 8K segment.

Here's the same graphical view, this time with ORG0 mode enabled:



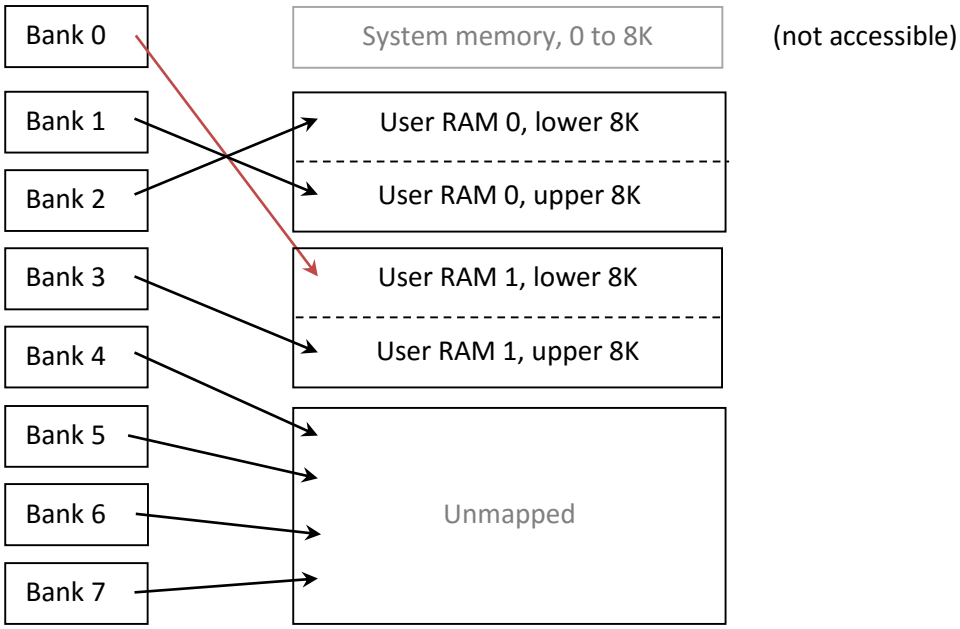
With ORG0 mode enabled, the previously inaccessible lower 8K segment of the memory expansion card is now accessible via the remapped bank 0 addresses. All system memory (system ROM, floppy ROM, floppy RAM, etc.) are inaccessible when ORG0 mode is enabled.

Earlier it was noted that, with ORG0 mode disabled, if you write to a bank 0 address, the U517 decoder would also write to the otherwise inaccessible lower 8K segment of the 16K expansion card. With ORG0 mode enabled, these same bank 0 addresses now map to the same lower 8K memory locations on the memory card.

For example, if the value 0x12 is written to address 0x0000 with ORG0 mode disabled, that value also is written to the expansion card's internal address 0x0000 (the first address in its lower 8K segment). This location is *not readable* when ORG0 mode is disabled (reading address 0x0000 will read from the system ROM, not the memory expansion card). If the program then enables ORG0 mode and reads address 0x0000, the value 0x12 will be returned, since the address 0x0000 will be remapped to the 16K expansion card's internal address 0x0000.

ORG0 mode on Systems with Less Than 64K of User RAM

To close out this discussion, consider what happens when ORG0 mode is enabled on a system with *less* than 64K of user RAM installed (one or both of the U517's A5 (JJ501) and A6 (JJ502) address lines is 0). Here's the graphical view again, this time with 32K of user RAM installed and ORG0 mode enabled:



Notice that target addresses within bank 0 are remapped to the *lower* 8K segment in User RAM 1, while target addresses within bank 4, which are normally mapped to the same lower 8K segment in User RAM 1, are now unmapped.

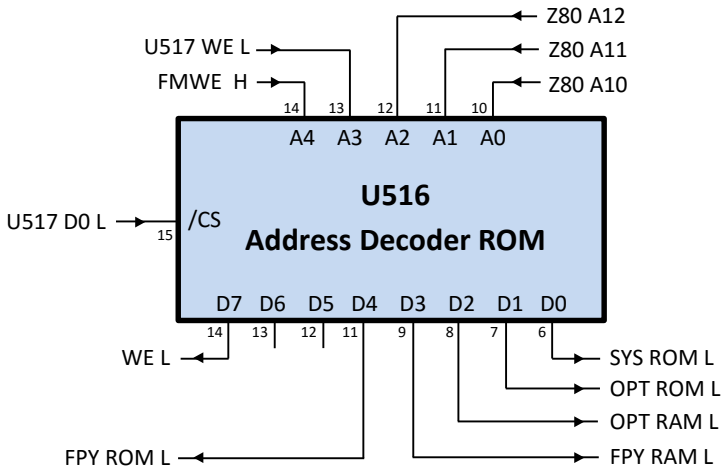
The U517 decoder "unmaps" target addresses within bank 4 simply by deactivating its D2 (/`RAS1 L`) data line. Instead, the decoder:

- Activates its D1 (`NOMEM L`) data line for read requests (data line D7=1), which causes the U521 "no memory" chip to return the value 0x00 on the Z80's data lines.
- Deactivates *all* of its data lines for write requests (data line D7=0), effectively ignoring the request.

To summarize, when ORG0 mode is enabled, the U517 decoder remaps all target addresses within bank 0 to the lower 8K segment of the *highest installed 16K user RAM device*, based on the setting of the JJ501 and JJ502 jumpers.

The U516 Page Decoder

The U516 decoder is a 32 x 8-bit PROM consisting of five address lines A0-A4, eight data lines D0-D7 and a chip select line (/CS). The diagram below shows the basic layout of the U516 decoder:[18]



The U516 decoder serves as a memory *page* decoder for target addresses within bank 0 (0x0000 to 0x1FFF), where all system memory devices reside. When the U517 decoder is selected during a Z80 memory read or write cycle, and it detects a target address within bank 0, it selects the U516 decoder by activating its /CS line (0). If the U517 decoder is not selected, or it does not detect a bank 0 address, the U516 decoder's /CS line is inactive (1).

When the U516 decoder is deselected — its /CS line is inactive (1) — all of its eight data lines are also inactive (1). When the decoder's /CS line is activated (0), it loads the contents of the cell addressed by its five address lines onto its eight data lines. The state of those data lines determine which system memory device, if any, is selected, and the type of operation to perform (read or write).

The U516 decoder monitors three of the Z80's latched address lines, `LA10`, `LA11` and `LA12`. These three lines form a 3-bit page address, 0 to 7, indicating which 1K page in bank 0 contains the target address. The table below lists the page map for all system memory devices residing in bank 0:

Page	Address Range	System Memory Device	Size
0	0x0000 to 0x03FF	System ROM (U518)	2K
1	0x0400 to 0x07FF		
2	0x0800 to 0x0BFF	Optional system ROM (U519)	2K
3	0x0C00 to 0x0FFF		
4	0x1000 to 0x13FF	Optional system RAM (U522, U524)	1K
5	0x1400 to 0x17FF	Floppy RAM (U523, U525)	1K
6	0x1800 to 0x1BFF	Floppy ROM (U520)	2K
7	0x1C00 to 0x1FFF		

[18] Not shown are the Vcc and GND pins, or the eight 1K ohm pull-up resistors that connect each data line to +5V.

The system ROM and floppy ROM devices are 2K x 8-bit ROM chips. The system ROM chip contains code for initializing the system at startup, as well as a basic "monitor" program for booting a mounted disk, setting the system date and time, examining and changing the contents of memory, etc.[19] The floppy ROM chip contains code and common routines for communicating with the H88-1 floppy disk controller.

The floppy RAM device is a 1K x 8-bit static RAM device consisting of two 1K x 4-bit static memory chips, U523 and U525. Floppy RAM is used by the operating system to maintain information about the currently mounted disk. It can be write-enabled or write-protected under software control.

Sockets are provided for an optional 2K x 8-bit ROM chip (U519) and optional 1K x 8-bit RAM chips (U522 and U524). These sockets are typically unpopulated.[20]

The table below lists the U516 decoder's five address lines and their connections:

Line	Pin	Connects to
A0 (LA10)	10	The latched Z80 A10, A11 and A12 address lines. These three lines form a 3-bit page address (0 to 7).
A1 (LA11)	11	
A2 (LA12)	12	
A3 (WE L)	13	The D7 (WE L) data line (pin 14) from the U517 decoder ROM.[21] This line, when active (0), indicates that the Z80 has issued a memory write request.
A4 (FMWE H)	14	The FMWE H line[22] from pin 12 of the P512 accessory card connector, where the H88-1 Floppy Disk interface card is installed. This line, when active (1), indicates that floppy RAM is write-enabled.; when inactive (0), indicates that floppy RAM is write-protected.

The decoder's A0, A1 and A2 address lines indicate which page (0-7) in bank 0 contains the target address, as described earlier.

The A3 address line is connected to the U517 bank decoder's D7 (WE L) data line. The U517 decoder activates this line (0) to indicate a Z80 memory write request, and deactivates this line (1) to indicate a Z80 memory read or refresh

[19] The original system ROM was MTR-88. Later ROM versions, MTR-89 and MTR-90, were released to support newer disk drives and provide additional monitoring commands. The MTR-88 and MTR-89 are 2K ROM devices, while the MTR-90 is a 4K ROM device.

[20] To my knowledge Heath never populated the U519 optional system ROM or U522/U524 optional system RAM sockets.

[21] The CPU Logic Circuit Board schematic incorrectly shows this U516 pin as not connected to anything other than its pull-up resistor.

[22] The FMWE H line is mislabeled on some CPU Logic Circuit Board schematics as FMWEH L. This is an active/high line, so the L suffix is incorrect. The corrected line name, FMWE H, is used throughout this document.

request. In response to a memory write request, the U516 decoder deactivates all of its data lines, *except* in these cases:

- The target address is within page 4 (the optional system RAM), or
- The target address is within page 5 (the floppy RAM device) *and* the A4 address line indicates that floppy RAM is write-enabled (1).

The A4 address line is connected to pin 12 of the P512 accessory card connector, where the optional H88-1 Floppy Disk Controller card is installed. This line, when active (1), indicates that the controller's associated floppy RAM is write-enabled. When inactive (0), this line indicates that the floppy RAM is write-protected (read-only).

The table below lists each of the U516 decoder's data lines and its function:

Line	Pin	Connects To
D0 (SYS ROM L)	1	The /CS (Chip Select) line of the U518 system ROM chip. Activating this line (0) selects the system ROM.
D1 (OPT ROM L)	2	The A pin of the JJ508 jumper.[23] Activating this line (0) when the JJ508 jumper is in its "A" position selects the U519 optional system ROM. Activating this line when the JJ508 jumper is in its "B" position selects no device.
D2 (OPT RAM L)	3	The /CS (Chip Select) lines of the U522 and U524 optional static RAM chips. Activating this line (0) selects the optional system RAM.
D3 (FPY RAM L)	4	The /CS (Chip Select) lines of the U523 and U525 floppy RAM chips. Activating this line (0) selects the floppy RAM.
D4 (FPY ROM L)	5	The /CS (Chip Select) line of the U520 floppy ROM chip *and* the B pin of the JJ508 jumper. Activating this line (0) selects the floppy ROM device and, if the JJ508 jumper is in the "B" position, also selects the optional U519 system ROM device.
D5	6	Unused. This line's state is always inactive (1).
D6	7	Unused. This line's state is always inactive (1).
D7 (WE L)	9[24]	The /WE (Write Enable) line of the U523 and U525 floppy RAM chips. If this line is active (0) when the Floppy RAM device is selected, and the floppy RAM device is write-enabled (see below), the device performs a write operation. If this line is inactive (1) when the floppy RAM device is selected, the device performs a read operation.

The MTR-90 System ROM

Heath released three different system ROM parts during the life of the H89 (several third party manufacturers also released their own system ROM part to support their own add-on hardware). The MTR-88 and MTR-89 system ROMs are 2K x 8-bit parts, while the MTR-90 system ROM is a 4K x 8-bit part.

Heath could have released the MTR-90 system ROM as two separate 2K x 8-bit parts, one installed in the U518 system ROM socket and the other installed in the U519 optional ROM socket. Instead, Heath took advantage of the denser 4K x

[23] This jumper is JJ507 on newer H89 systems (seven jumpers instead of eight).
[24] This pin is incorrectly labeled as pin 8 on some H89 schematics.

8-bit ROM parts packaged in the same 24-pin DIP as the previously released 2K x 8-bit ROM parts. That allowed Heath to deliver the MTR-90 ROM as a single 4K x 8-bit part, installed in the same U518 system ROM socket as the previous MTR-88 and MTR-89 system ROM parts. The only problem was that the U518 socket was never designed to support a 4K x 8-bit part. It provided 11 address lines (A0-A10) to handle addresses up to 2K, but lacked the extra address line (A11) required to handle addresses up to 4K.

So Heath came up with a rather clever solution. The JJ505 jumper (JJ506 on older CPU boards), located near the U518 and U519 ROM sockets, was originally designed to tie the U518 and U519 sockets' V_{BB} lines (pin 21) to either +5V or ground. Pin 21 happened to be the additional A11 address line on the newer 4K x 8-bit ROM parts. So Heath "repurposed" the jumper, instructing its MTR-90 ROM customers to remove the jumper's shunt and install a special jumper wire connecting the jumper's center pin to pin 14 of the P507, P508 or P509 accessory connector. This wire routed the latched A11 address line (LA11) from pin 14 of the accessory connectors to pin 21 of the U518 and U519 ROM sockets, thereby extending the U518 system ROM's address range from 2K to 4K.[25]

One other change is required to support the 4K MTR-90 ROM: a different U516 Memory Decoder ROM part. The U516 decoder ROM, part 444-41, maps addresses within pages 0 or 1 (0x0000 to 0x07FF) to the 2K MTR-88 or MTR-89 system ROM, and addresses within pages 2 or 3 (0x0800 to 0x0FFF) to the optional 2K system ROM (U519). A different U516 decoder ROM, part 444-83, maps all page 0 and 1 addresses to the same U518 system ROM device, but also maps all page 2 and 3 addresses to the U518 system ROM device. It does not map any addresses to the optional U519 system ROM (the decoder's D1 data line is always inactive (1)).

With the 4K MTR-90 system ROM and updated U516 decoder ROM installed, the page map for the bank 0 system devices now look like this:

Page	Address Range	System Memory Device	Size
0	0x0000 to 0x03FF	System ROM (U518)	4K
1	0x0400 to 0x07FF		
2	0x0800 to 0x0BFF		
3	0x0C00 to 0x0FFF		
4	0x1000 to 0x13FF	Optional system RAM (U522, U524)	1K
5	0x1400 to 0x17FF	Floppy RAM (U523, U525)	1K
6	0x1800 to 0x1BFF	Floppy ROM (U520)	2K
7	0x1C00 to 0x1FFF		

Writing to Floppy RAM

The A4 address line FMWE H (Floppy Memory Write Enable) indicates whether or not writing to floppy RAM is allowed. When the A4 address line is active (1), writing to floppy RAM is permitted. When the A4 address line is inactive (0), writing to floppy RAM is prohibited.

The A4 address line originates at control port 177Q on the H88-1 Floppy Disk interface card, connected to the P506 and P512 connectors. The FMWE H line can be set or cleared under software control by writing to the floppy controller's control port. Writing to port 177Q with bit 7 set (1xxxxxxxB) activates the FMWE H line (1) and allows writing to floppy

[25] Note that this modification precluded the use of a 2K ROM part in either the U518 system ROM socket or the U519 optional ROM socket. Users wanting to revert back to the MTR-88 or MTR-89 system ROM part would need to remove the special jumper wire.

RAM; writing to port 177Q with bit 7 clear (0xxxxxxxB) deactivates the FMWE H line (0) and prohibits writing to floppy RAM.

When the U516 decoder detects a Z80 read request (the decoder's A3 (WE L) address line is inactive) with a target address within page 5, the decoder activates its D7 (WE L) data line, which selects the floppy RAM device. When the decoder detects a write request (its D7 address line is active) with a target address within page 5, the decoder selects or deselects the floppy RAM, depending on the state of its A4 (FMWE H) address line. If the A4 address line is active (writing to floppy RAM is allowed), the decoder activates its D3 and D7 data lines, triggering a write to the floppy RAM device. If the A4 address line is inactive (writing to floppy RAM is prohibited), the decoder deactivates *all* of its data lines, ignoring the write request.

U516 Address Decoding in Action

Appendix B lists the 32 possible states of the U516 decoder's address lines and corresponding data lines. Listed below is a subset of that table, with active data lines highlighted.[26] Address line states for page 5 addresses (floppy RAM) have been expanded to show the effect of the A4 (FMWE H) address line.

Address Lines						Data Lines						
FMWE H	U517 WE L	Z80 A12	Z80 A11	Z80 A10		WE L	FPY ROM H	FPY RAM H	OPT RAM H	OPT ROM H	SYS ROM H	
A4	A3	A2	A1	A0	Page	D7	D4	D3	D2	D1	D0	Result
x	0	0	0	0	0	1	1	1	1	1	1	No device selected
x	1	0	0	0		1	1	1	1	1	0	Reads System ROM
x	0	0	0	1	1	1	1	1	1	1	1	No device selected
x	1	0	0	1		1	1	1	1	1	0	Reads system ROM
x	0	0	1	0	2	1	1	1	1	1	1	No device selected
x	1	0	1	0		1	1	1	1	0	1	Reads optional ROM
x	0	0	1	1	3	1	1	1	1	1	1	No device selected
x	1	0	1	1		1	1	1	1	0	1	Reads optional ROM
x	0	1	0	0	4	1	1	1	0	1	1	Reads optional RAM
x	1	1	0	0		1	1	1	0	1	1	Reads optional RAM
0	1	1	0	1	5	1	1	0	1	1	1	Reads Floppy RAM
1	1	1	0	1		1	1	0	1	1	1	Reads Floppy RAM
0	0	1	0	1		1	1	1	1	1	1	No device selected
1	0	1	0	1		0	1	0	1	1	1	Writes Floppy RAM
x	1	1	1	0	6	1	1	1	1	1	1	No device selected
x	1	1	1	0		1	0	1	1	1	1	Reads Floppy ROM
x	0	1	1	1	7	1	1	1	1	1	1	No device selected
x	1	1	1	1		1	0	1	1	1	1	Reads Floppy ROM

x = don't care

Some observations:

- No more than one system memory device is selected for any given address lines state.[27]
- If a memory read is requested (A3=1), the device corresponding to the target address's memory page is selected and the D7 (WE L) data line is inactive (1).

[26] No columns are listed for the unused D5 and D6 data lines.
[27] Not so with the U517 decoder; see the section *Writing to Bank 0 (System Memory)*.

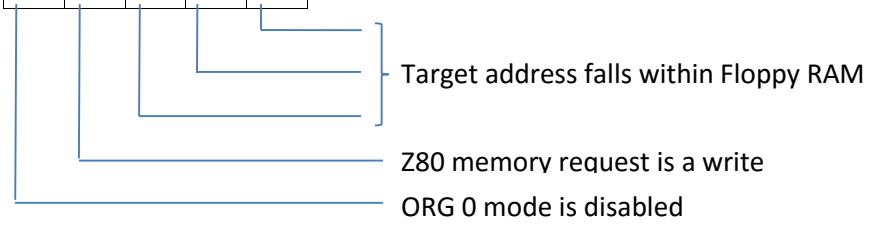
- If a memory write is requested (A3=0) with a target address *outside* of page 4 (optional RAM) or page 5 (floppy RAM), no device is selected (all data lines are 1).

- If the Z80 issues a memory write request (A3=0) with a target address within page 4 (optional system RAM), that device is selected (D2=0). But the D7 (`WE L`) data line is 1, indicating a read request (!). That's not a problem because the U522 and U524 optional RAM chips' /WE (write enable) lines are not the U516 decoder's D7 (`WE L`) data line; they're connected instead to the U517 decoder's D7 (`WE L`) data line.

- If the Z80 issues a memory write request (A3=1) with a target address within page 5 (floppy RAM), that device is selected (D3=0) *if and only if* the floppy RAM is write-enabled (A4=1). If the floppy RAM is write-protected (A4=0), then no device is selected (all data lines are 1).

- The D7 (`WE L`) data line is activated (0) for exactly one of the 32 possible address states:

A4	A3	A2	A1	A0
0	0	1	0	1



ORG0 mode and Floppy RAM

As explained earlier, when ORG0 mode is enabled, the U517 bank decoder remaps all target addresses within bank 0 (0x0000 to 0x1FFF) to the lower 8K of the highest installed 16K User RAM device. What does this mean when reading or writing floppy RAM?

When ORG0 mode is enabled, the U516 decoder is no longer selected; all of its data lines are inactive (1). As such, *none* of the system devices controlled by the U516 decoder are selected, including the floppy RAM device. When the Z80 issues a memory read or write request with a target address within bank 0, page 5 (0x1400 to 0x17FF) the U523 and U525 floppy RAM chips are *not* selected; instead the U517 decoder directs the write to page 5 of the lower 8K bank of the highest installed 16K User RAM device.

When ORG0 mode is enabled, the state of the U516 decoder's A4 (`FMWE H`) address line *does not matter*. Programs can freely write to addresses in the remapped floppy RAM address range, even though the U516 decoder's A4 line from the controller might indicate that the floppy RAM is write-protected.

Here's an example: Suppose an H89 system contains 64K of user RAM (the 16K expansion card is installed). Suppose also that the U516 decoder's A4 (`FMWE H`) address line is 0, indicating that the floppy RAM is write-protected.

When ORG0 mode is enabled, addresses in bank 0, page 5 (0x1400 to 0x17FF), normally mapped to the floppy RAM, are instead remapped to bank 7, page 5 (0xF400 to 0xF7FF). If a program writes a value to address 0x1400 (the lowest address in floppy RAM), that value will instead be written to address 0xF400 of the 16K expansion card, even though the floppy RAM device is write-protected.

With no way for the hardware itself to write-protect floppy RAM when ORG0 mode is enabled, it becomes the responsibility of the operating system software to write-protect the remapped floppy RAM addresses (0xF400 to 0xF7FF in the example above). Unfortunately, because the Z80 architecture provides no special executive or supervisor modes to separate and protect operating system memory from user memory, a user program is free to write to *any* writable memory location in the H89, including the relocated 1K floppy RAM addresses remapped to user RAM. This in turn could corrupt data on a mounted floppy disk.

Even if the Z80 supported such modes, the H89 would also need a way to prevent a user program from writing to port 177Q, which ultimately write-enables and write-protects floppy RAM.

In Summary

The H89's U517 and U516 ROM address decoders provide powerful and efficient memory address decoding with a minimum number of parts. They also provide a valuable memory remapping feature (ORG0) that allows access to all 64K of user RAM.

Part II: I/O Address Decoding

This section describes in detail how the H89 decodes the Z80's A0-A7 address lines during an I/O read or write operation, and selects the appropriate I/O device to service the request.

Port Numbers

During an I/O read or write operation the Z80's A0-A7 address lines contain the *port number* of the target I/O device. This allows the Z80 to access up to 256 distinct I/O devices.[28]

The H89 supports several I/O devices. Some I/O devices are assigned a single port number while others are assigned multiple port numbers. For example, the *General Purpose Port* (GPP) I/O device, embedded on the CPU logic board, is assigned the single port number 362Q.[29] This device consists of an 8-section DIP switch (SW501) that can be read by issuing an I/O read to port 362Q, and an 8-bit latch (U552) that can be written by performing an I/O write to the same port.

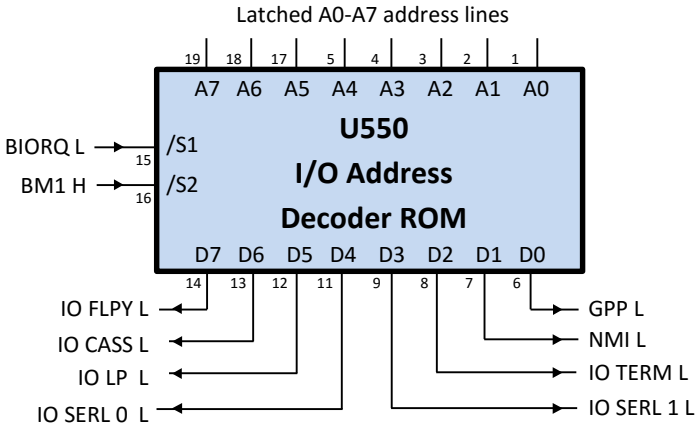
The H89 also supports several optional I/O devices in the form of accessory cards that plug into the accessory connector pairs P504/P510, P505/P511 or P506/P512. For example, the optional H88-1 Floppy Disk interface card, which plugs into accessory connector pair P506/P512, is assigned a range of port numbers, 374Q through 377Q. These ports numbers all map to the same physical device (the card's H17 floppy disk controller); circuitry on the card decodes the port number to determine which controller function to perform.

The U550 I/O Port Decoder

The U550 I/O port decoder, a 256 x 8-bit ROM, is responsible for mapping each 8-bit port number to a target I/O device. It functions very similar to the U517 memory decoder ROM, in that its eight active/low data lines each control (select and deselect) a specific I/O device in response to a specific address (port number) on its eight address lines.

[28] The Z80's A8-A15 address lines also contain meaningful values during an I/O operation, depending on the I/O instruction executed. With some effort, these address lines could also be used to extend the range of I/O devices. The H89 makes use of only the A0-A7 address lines during all I/O requests.
[29] H89 I/O port numbers are typically expressed in octal notation. That practice is followed in this document, adding the Q suffix to avoid any confusion.

The diagram below shows the U550 ROM's address, data and select lines:



The U550 decoder, like the U517 memory address decoder, contains two active/low select lines, /S1 and /S2, both of which must be active (0) to select the ROM. The /S1 line is connected to the buffered output of the Z80's /IORQ status line (`BIORQ L`), while the /S2 line is connected to the buffered and inverted output of the Z80's /M1 status line (`BM1 H`). That means the U550 decoder is selected when the Z80's /IORQ line is active (0) and its /M1 line is inactive (1).

The reason for including the Z80's /M1 status line in the selection of the U550 I/O decoder is because the Z80 activates its /IORQ line (0) for two different types of machine cycles: an I/O Read/Write cycle, and an Interrupt Acknowledge cycle (when the Z80 acknowledges a maskable interrupt request). The Z80 activates its /M1 status line (0) during an Interrupt Acknowledge cycle to differentiate that cycle from an I/O Read/Write cycle. As a result, the U550 I/O decoder is only selected during an I/O Read/Write cycle, not an Interrupt Acknowledge cycle.

Each of the U550 decoder's A0-A7 address lines is connected to its corresponding latched Z80 address line (`LA0-LA7`). The 8-bit address (a.k.a. port number) determines which data line is activated (0), which in turn determines which I/O device is selected.

The table below lists each of the U516 decoder's data lines and its function:

Line	Pin	Connects To
D0 (`GPP L`)	1	Pins 2 and 5 of the U553 quad OR chip. When pin 2 is active (0) and pin 1, connected to the buffered Z80 /RD status line (`BRD L`), is active (0), the OR gate's output (pin 3) is active, which selects the General Purpose Port's U551 hex tri-state buffer, which routes the outputs of the SW501 8-section switch bank to the Z80's D0-D7 data lines during an I/O Read cycle. When pin 5 is active (0) and pin 4, connected to the buffered Z80 /WR status line (`BWR L`) is active (0), the OR gate's output (pin 6) is active, which selects the U552 8-line latch, which stores the contents of the Z80's D0-D7 data lines during an I/O Write cycle.

Line	Pin	Connects To
D1 (NMI L)	2	The Z80's /NMI control line. Activating this line (0) initiates a non-maskable interrupt. In response, the Z80, at the end of its current instruction cycle, pushes the current value of the PC on the external stack and jumps to address 0x0066.
D2 (I/O TERM L)	3	The /CS2 line (pin 14) of the U561 Terminal Port (8250 Asynchronous Communications Element, or *ACE*). This chip serves as the primary interface between the CPU logic board and the Terminal logic board, which controls the H89's video display and keyboard. The U561 chip has three select lines, the active/high CS0 and CS1 lines, and the active/low /CS2 line. All three lines must be active to select the chip. Because the CS0 and CS1 lines are permanently tied to +5V, only the /CS2 line is used to select the chip.
D3 (I/O SERL 1 L)	4	Pin 10 of the P510, P511 and P512 accessory connectors. When the optional H88-3 Serial interface card is installed in the P504/P510 or P505/P511 connector pairs, the D3 data line is used to select/deselect the U604 Data Terminal Equipment (DTE) 8250 ACE on the card.
D4 (I/O SERL 0 L)	5	Pin 9 of the P510, P511 and P512 accessory connectors. When the optional H88-3 Serial interface card is installed in the P504/P510 or P505/P511 connectors, the D4 data line is used to select/deselect the U603 Data Communication Equipment (DCE) 8250 ACE on the card.
D5 (I/O LP L)	6	Pin 11 of the P510 and P511 accessory connectors (not the P512 connector). When the optional H88-3 Serial interface card is installed in the P504/P510 or P505/P511 connectors, the D5 data line is used to select/deselect the U603 Line Printer 8250 ACE on the card.
D6 (I/O CASS L)	7	Pin 12 of the P510 and P511 accessory connectors (not the P512 connector). When the optional H88-5 Cassette interface card is installed in the P504/P510 or P505/P511 connectors, the D6 data line is used to select/deselect the cassette read/write circuitry on the card.
D7 (I/O FLPY L)	9[30]	Pin 11 of the P512 accessory connector. When the optional H88-1 Floppy Disk interface card is installed in the P506/P512 connector, the D7 data line is used to select the floppy controller circuitry on the card.

The D0-D3 data lines control I/O devices embedded on the CPU logic board; as such their meaning and purpose are fixed and cannot be changed. However, the D4-D7 lines control *optional* I/O devices plugged into the accessory connector pairs on the CPU board. The names of those signals, taken from the H89's CPU logic board's schematic, suggest a specific I/O device, but in fact can be any device plugged into the correct accessory connector and tied to the connector's documented pin for selection.

[30] This pin is incorrectly labeled as pin 8 on some H89 schematics.

For example, the D6 data line, named `I/O CASS L`, is used to select the optional H88-5 Cassette interface card. But the same D6 data line was later used instead to select newer disk interface cards like the H/Z-47 and H/Z67, replacing the Cassette interface card.

Heath released two versions of the U550 I/O decoder ROM part: the older 444-43, which supported the H88-5 Cassette interface card, and the later 444-61, which dropped the H88-3 Cassette interface card and replaced it with a second disk interface card. Both parts support the H88-1 (H17) Floppy Disk interface card and the H88-3 Serial I/O Interface card.

As mentioned earlier, an I/O device can "own" multiple port numbers. Although there is nothing in the design of the U550 I/O decoder ROM to preclude a device owning a set of non-contiguous port numbers, in all known cases, the port numbers of an I/O devices are always contiguous.

The table below lists the I/O devices supported by the 444-61 I/O decoder ROM, their port numbers and the U550 data line used to select the device:

I/O Device	Port Number(s)	U550 Data Line
H/Z-37 5 ¼" soft-sectored Floppy Disk interface card	170Q to 173Q or 174Q to 177Q	D6 or D7
H/Z-47 8" Floppy Disk interface card	170Q to 173Q or 174Q to 177Q	D6 or D7
H/Z-67 Winchester + Floppy Disk interface card	170Q to 173Q or 174Q to 177Q	D6 or D7
H88-1 5 ¼" hard-sectored Floppy Disk interface card	174Q to 177Q	D7
H88-3 Serial I/O Interface card (DCE)	320Q to 327Q	D4
H88-3 Serial I/O Interface card (DTE)	330Q to 337Q	D3
H88-3 Serial I/O Interface card (LP)	340Q to 347Q	D5
Console Serial Port	350Q to 357Q	D2
H8 Front Panel (compatibility mode)	360Q, 361Q, 372Q, 373Q	D1
General Purpose Port	362Q	D3

Some comments are in order about the U550's D6 and D7 data lines and how they are used to select the various disk interface cards.

When the U550 I/O decoder ROM part 444-61's address lines contain one of the port numbers 174Q to 177Q, it activates *both* its D6 and D7 data lines. This is the only instance where the U550 activates more than one data line for a given port number. The D6 data line selects, via pin 11 of the P510 and P511 accessory connectors, whichever disk interface card is plugged into connector pair P504/P510 or P505/P511. The D7 data line selects, via pin 11 of the P512 accessory connector, whichever disk interface card is plugged into connector pair P506/P512. Initially this was the H88-1 Floppy Disk interface card, but that card could be replaced by one of the newer disk interface cards.

At first glance it might appear that having the U550 ROM activate both its D6 and D7 data lines in response to port numbers 174Q to 177Q could result in two different disk interface cards responding to the same port number, leading to a possible bus conflict when they each load their input value onto the data bus. But that's not the case.

Recall that the U550's D7 data line is tied to pin 12 of the P512 connector, while the D6 data line is tied to pin 11 of the P510 and P511 connectors. In no case are *both* of those data lines connected on to pins on the *same* connector. Whichever disk interface card is installed in connector pair P506/P512 can only be selected by the U550's D7 data line, since the D6 data line's signal is not present on the P512 connector. Similarly, whichever disk interface card is installed in connector pair P504/P510 or P505/P511 can only be selected by the U550's D6 data line, since the D7 data line's signal is not present on the P510 or P511 connectors.[31]

This design allows any two supported disk interface cards to be used together:

- If one of the disk interface cards is the H88-1 (which must be installed in the P506/P512 connector pair and can only be selected by the U550's D7 data line), then the second disk interface card is installed in the P504/P510 or P505/P511 connector pair and is jumpered so that the card is selected by the U550's D6 data line.
- If the H88-1 card is not used, then one of the two disk interface cards is installed in the P506/P512 connector pair and is jumpered so that the card is selected by the U550's D7 data line, while the second disk interface card is installed in the P504/P510 or P505/P511 connector pair and is jumpered so that the card is selected by the U550's D6 data line.

Whichever disk interface card is installed in the P506/P512 connector pair *must* be configured to respond to port numbers 174Q to 177Q.[32] The other disk interface card, installed in either the P504/P510 pair or P505/P511 pair, must be configured to respond to port numbers 170Q to 173Q.

See *Appendix C* for a table listing the U550 I/O Decoder ROM (part 444-61) 256 address line states and corresponding data line values.

In Summary

The H89 decodes I/O port numbers via the U550 I/O port decoder ROM, similar to the way it decodes memory addresses via its U516 and U517 memory address decoder ROMs.

The use of a ROM for I/O decoding significantly reduced parts and design complexity. It also made it easy for Heath and other 3rd party manufacturers to support new I/O devices by providing an updated U550 ROM part.

[31] The only (pathological) case where a bus conflict could occur would be if two disk interface cards, neither of which was the H88-1 Floppy Disk interface card, were installed in the P504/P510 and P505/P511 connector pairs, and *both* were configured to use the U55's D6 data line for selection.

[32] The H88-1 Floppy Disk interface card, which can only be installed in the P506/P512 connector pair, always responds to I/O port numbers 174Q to 177Q; it has no on-board jumpers for assigning an alternate range of port numbers.

Appendix A: U517 Decoder ROM 444-66

The table below was created by removing the U517 decoder ROM, part 444-66, from my H89's CPU logic board, connecting it to a test harness, and cycling through the ROM's 256 address line states to capture the corresponding data line values.

The *Result* column indicates the action taken in response to each input state.

Address Lines								Data Lines								
ORG0 H	JJ502	JJ501	BRD L	BRFSH L	Z80 A15	Z80 A14	Z80 A13	WE L	RD5 L	RD6 L	RAS2 L	RAS1 L	RAS0 L	NOMEM L	U516 L	
A7	**A6**	**A5**	**A4**	**A3**	**A2**	**A1**	**A0**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**	**Result**
0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	1	Refreshes RAM 0
0	0	0	0	0	0	0	1	1	1	1	1	1	0	1	1	Refreshes RAM 0
0	0	0	0	0	0	1	0	1	1	1	1	1	0	1	1	Refreshes RAM 0
0	0	0	0	0	0	1	1	1	1	1	1	1	0	1	1	Refreshes RAM 0
0	0	0	0	0	1	0	0	1	1	1	1	1	0	1	1	Refreshes RAM 0
0	0	0	0	0	1	0	1	1	1	1	1	1	0	1	1	Refreshes RAM 0
0	0	0	0	0	1	1	0	1	1	1	1	1	0	1	1	Refreshes RAM 0
0	0	0	0	0	1	1	1	1	1	1	1	1	0	1	1	Refreshes RAM 0
0	0	0	0	1	0	0	0	1	1	1	1	1	1	1	0	Reads U516
0	0	0	0	1	0	0	1	1	1	1	1	1	0	1	1	Reads RAM 0
0	0	0	0	1	0	1	0	1	1	1	1	1	0	1	1	Reads RAM 0
0	0	0	0	1	0	1	1	1	1	1	1	1	1	0	1	Reads NOMEM
0	0	0	0	1	1	0	0	1	1	1	1	1	1	0	1	Reads NOMEM
0	0	0	0	1	1	0	1	1	1	1	1	1	1	0	1	Reads NOMEM
0	0	0	0	1	1	1	0	1	1	1	1	1	1	0	1	Reads NOMEM
0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	1	Reads NOMEM
0	0	0	1	0	0	0	0	1	1	1	1	1	0	1	1	Refreshes RAM 0
0	0	0	1	0	0	0	1	1	1	1	1	1	0	1	1	Refreshes RAM 0
0	0	0	1	0	0	1	0	1	1	1	1	1	0	1	1	Refreshes RAM 0
0	0	0	1	0	0	1	1	1	1	1	1	1	0	1	1	Refreshes RAM 0
0	0	0	1	0	1	0	0	1	1	1	1	1	0	1	1	Refreshes RAM 0
0	0	0	1	0	1	0	1	1	1	1	1	1	0	1	1	Refreshes RAM 0
0	0	0	1	0	1	1	0	1	1	1	1	1	0	1	1	Refreshes RAM 0
0	0	0	1	0	1	1	1	1	1	1	1	1	0	1	1	Refreshes RAM 0
0	0	0	1	1	0	0	0	0	1	1	1	1	1	1	0	Writes U516
0	0	0	1	1	0	0	1	0	1	1	1	1	0	1	1	Writes RAM 0
0	0	0	1	1	0	1	0	0	1	1	1	1	0	1	1	Writes RAM 0
0	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	No Device Selected
0	0	0	1	1	1	0	0	1	1	1	1	1	1	1	1	No Device Selected
0	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	No Device Selected
0	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	No Device Selected

Address Lines								Data Lines								
ORG0 H	JJ502	JJ501	BRD L	BRFSH L	Z80 A15	Z80 A14	Z80 A13	WE L	RD5 L	RD6 L	RAS2 L	RAS1 L	RAS0 L	NOMEM L	U516 L	
A7	**A6**	**A5**	**A4**	**A3**	**A2**	**A1**	**A0**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**	**Result**
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	No Device Selected
0	0	1	0	0	0	0	0	1	1	1	1	0	0	1	1	Refreshes RAM 0, RAM 1
0	0	1	0	0	0	0	1	1	1	1	1	0	0	1	1	Refreshes RAM 0, RAM 1
0	0	1	0	0	0	1	0	1	1	1	1	0	0	1	1	Refreshes RAM 0, RAM 1
0	0	1	0	0	0	1	1	1	1	1	1	0	0	1	1	Refreshes RAM 0, RAM 1
0	0	1	0	0	1	0	0	1	1	1	1	0	0	1	1	Refreshes RAM 0, RAM 1
0	0	1	0	0	1	0	1	1	1	1	1	0	0	1	1	Refreshes RAM 0, RAM 1
0	0	1	0	0	1	1	0	1	1	1	1	0	0	1	1	Refreshes RAM 0, RAM 1
0	0	1	0	0	1	1	1	1	1	1	1	0	0	1	1	Refreshes RAM 0, RAM 1
0	0	1	0	1	0	0	0	1	1	1	1	1	1	1	0	Reads U516
0	0	1	0	1	0	0	1	1	1	1	1	1	0	1	1	Reads RAM 0
0	0	1	0	1	0	1	0	1	1	1	1	1	0	1	1	Reads RAM 0
0	0	1	0	1	0	1	1	1	1	1	1	0	1	1	1	Reads RAM 1
0	0	1	0	1	1	0	0	1	1	1	1	0	1	1	1	Reads RAM 1
0	0	1	0	1	1	0	1	1	1	1	1	1	1	0	1	Reads NOMEM
0	0	1	0	1	1	1	0	1	1	1	1	1	1	0	1	Reads NOMEM
0	0	1	0	1	1	1	1	1	1	1	1	1	1	0	1	Reads NOMEM
0	0	1	1	0	0	0	0	1	1	1	1	0	0	1	1	Refreshes RAM 0, RAM 1
0	0	1	1	0	0	0	1	1	1	1	1	0	0	1	1	Refreshes RAM 0, RAM 1
0	0	1	1	0	0	1	0	1	1	1	1	0	0	1	1	Refreshes RAM 0, RAM 1
0	0	1	1	0	0	1	1	1	1	1	1	0	0	1	1	Refreshes RAM 0, RAM 1
0	0	1	1	0	1	0	0	1	1	1	1	0	0	1	1	Refreshes RAM 0, RAM 1
0	0	1	1	0	1	0	1	1	1	1	1	0	0	1	1	Refreshes RAM 0, RAM 1
0	0	1	1	0	1	1	0	1	1	1	1	0	0	1	1	Refreshes RAM 0, RAM 1
0	0	1	1	0	1	1	1	1	1	1	1	0	0	1	1	Refreshes RAM 0, RAM 1
0	0	1	1	1	0	0	0	0	1	1	1	1	1	1	0	Writes U516
0	0	1	1	1	0	0	1	0	1	1	1	1	0	1	1	Writes RAM 0
0	0	1	1	1	0	1	0	0	1	1	1	1	0	1	1	Writes RAM 0
0	0	1	1	1	0	1	1	0	1	1	1	0	1	1	1	Writes RAM 1
0	0	1	1	1	1	0	0	0	1	1	1	0	1	1	1	Writes RAM 1
0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	No Device Selected
0	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	No Device Selected
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	No Device Selected
0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	Refreshes RAM 0, RAM 1, RAM 2
0	1	0	0	0	0	0	1	1	1	1	0	0	0	1	1	Refreshes RAM 0, RAM 1, RAM 2
0	1	0	0	0	0	1	0	1	1	1	0	0	0	1	1	Refreshes RAM 0, RAM 1, RAM 2
0	1	0	0	0	0	1	1	1	1	1	0	0	0	1	1	Refreshes RAM 0, RAM 1, RAM 2

Address Lines								Data Lines								
ORG0 H	JJ502	JJ501	BRD L	BRFSH L	Z80 A15	Z80 A14	Z80 A13	WE L	RD5 L	RD6 L	RAS2 L	RAS1 L	RAS0 L	NOMEM L	U516 L	
A7	**A6**	**A5**	**A4**	**A3**	**A2**	**A1**	**A0**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**	**Result**
0	1	0	0	0	1	0	0	1	1	1	0	0	0	1	1	Refreshes RAM 0, RAM 1, RAM 2
0	1	0	0	0	1	0	1	1	1	1	0	0	0	1	1	Refreshes RAM 0, RAM 1, RAM 2
0	1	0	0	0	1	1	0	1	1	1	0	0	0	1	1	Refreshes RAM 0, 1 and 2
0	1	0	0	0	1	1	1	1	1	1	0	0	0	1	1	Refreshes RAM 0, 1 and 2
0	1	0	0	1	0	0	0	1	1	1	1	1	1	1	0	Reads U516
0	1	0	0	1	0	0	1	1	1	1	1	1	0	1	1	Reads RAM 0
0	1	0	0	1	0	1	0	1	1	1	1	1	0	1	1	Reads RAM 0
0	1	0	0	1	0	1	1	1	1	1	1	0	1	1	1	Reads RAM 1
0	1	0	0	1	1	0	0	1	1	1	1	0	1	1	1	Reads RAM 1
0	1	0	0	1	1	0	1	1	1	1	0	1	1	1	1	Reads RAM 2
0	1	0	0	1	1	1	0	1	1	1	0	1	1	1	1	Reads RAM 2
0	1	0	0	1	1	1	1	1	1	1	1	1	1	0	1	Reads NOMEM
0	1	0	1	0	0	0	0	1	1	1	0	0	0	1	1	Refreshes RAM 0, 1 and 2
0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	1	Refreshes RAM 0, 1 and 2
0	1	0	1	0	0	1	0	1	1	1	0	0	0	1	1	Refreshes RAM 0, 1 and 2
0	1	0	1	0	0	1	1	1	1	1	0	0	0	1	1	Refreshes RAM 0, 1 and 2
0	1	0	1	0	1	0	0	1	1	1	0	0	0	1	1	Refreshes RAM 0, 1 and 2
0	1	0	1	0	1	0	1	1	1	1	0	0	0	1	1	Refreshes RAM 0, 1 and 2
0	1	0	1	0	1	1	0	1	1	1	0	0	0	1	1	Refreshes RAM 0, 1 and 2
0	1	0	1	0	1	1	1	1	1	1	0	0	0	1	1	Refreshes RAM 0, 1 and 2
0	1	0	1	1	0	0	0	0	1	1	1	1	1	1	0	Writes U516
0	1	0	1	1	0	0	1	0	1	1	1	1	0	1	1	Writes RAM 0
0	1	0	1	1	0	1	0	0	1	1	1	1	0	1	1	Writes RAM 0
0	1	0	1	1	0	1	1	0	1	1	1	0	1	1	1	Writes RAM 1
0	1	0	1	1	1	0	0	0	1	1	1	0	1	1	1	Writes RAM 1
0	1	0	1	1	1	0	1	0	1	1	0	1	1	1	1	Writes RAM 2
0	1	0	1	1	1	1	0	0	1	1	0	1	1	1	1	Writes RAM 2
0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	No Device Selected
0	1	1	0	0	0	0	0	1	1	0	0	0	0	1	1	Refreshes RAM 0, 1, 2 and RD6
0	1	1	0	0	0	0	1	1	1	0	0	0	0	1	1	Refreshes RAM 0, 1, 2 and RD6
0	1	1	0	0	0	1	0	1	1	0	0	0	0	1	1	Refreshes RAM 0, 1, 2 and RD6
0	1	1	0	0	0	1	1	1	1	0	0	0	0	1	1	Refreshes RAM 0, 1, 2 and RD6
0	1	1	0	0	1	0	0	1	1	0	0	0	0	1	1	Refreshes RAM 0, 1, 2 and RD6
0	1	1	0	0	1	0	1	1	1	0	0	0	0	1	1	Refreshes RAM 0, 1, 2 and RD6
0	1	1	0	0	1	1	0	1	1	0	0	0	0	1	1	Refreshes RAM 0, 1, 2 and RD6
0	1	1	0	0	1	1	1	1	1	0	0	0	0	1	1	Refreshes RAM 0, 1, 2 and RD6
0	1	1	0	1	0	0	0	1	1	1	1	1	1	1	0	Reads U516

Address Lines								Data Lines								
ORG0 H	JJ502	JJ501	BRD L	BRFSH L	Z80 A15	Z80 A14	Z80 A13	WE L	RD5 L	RD6 L	RAS2 L	RAS1 L	RAS0 L	NOMEM L	U516 L	
A7	**A6**	**A5**	**A4**	**A3**	**A2**	**A1**	**A0**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**	**Result**
0	1	1	0	1	0	0	1	1	1	1	1	1	0	1	1	Reads RAM 0
0	1	1	0	1	0	1	0	1	1	1	1	1	0	1	1	Reads RAM 0
0	1	1	0	1	0	1	1	1	1	1	1	0	1	1	1	Reads RAM 1
0	1	1	0	1	1	0	0	1	1	1	1	0	1	1	1	Reads RAM 1
0	1	1	0	1	1	0	1	1	1	1	0	1	1	1	1	Reads RAM 2
0	1	1	0	1	1	1	0	1	1	1	0	1	1	1	1	Reads RAM 2
0	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	Reads RD6
0	1	1	1	0	0	0	0	1	1	0	0	0	0	1	1	Refreshes RAM 0, 1, 2 and RD6
0	1	1	1	0	0	0	1	1	1	0	0	0	0	1	1	Refreshes RAM 0, 1, 2 and RD6
0	1	1	1	0	0	1	0	1	1	0	0	0	0	1	1	Refreshes RAM 0, 1, 2 and RD6
0	1	1	1	0	0	1	1	1	1	0	0	0	0	1	1	Refreshes RAM 0, 1, 2 and RD6
0	1	1	1	0	1	0	0	1	1	0	0	0	0	1	1	Refreshes RAM 0, 1, 2 and RD6
0	1	1	1	0	1	0	1	1	1	0	0	0	0	1	1	Refreshes RAM 0, 1, 2 and RD6
0	1	1	1	0	1	1	0	1	1	0	0	0	0	1	1	Refreshes RAM 0, 1, 2 and RD6
0	1	1	1	0	1	1	1	1	1	0	0	0	0	1	1	Refreshes RAM 0, 1, 2 and RD6
0	1	1	1	1	0	0	0	0	1	0	1	1	1	1	0	Writes U516 and RD6 (mirrored)
0	1	1	1	1	0	0	1	0	1	1	1	1	0	1	1	Writes RAM 0
0	1	1	1	1	0	1	0	0	1	1	1	1	0	1	1	Writes RAM 0
0	1	1	1	1	0	1	1	0	1	1	1	0	1	1	1	Writes RAM 1
0	1	1	1	1	1	0	0	0	1	1	1	0	1	1	1	Writes RAM 1
0	1	1	1	1	1	0	1	0	1	1	0	1	1	1	1	Writes RAM 2
0	1	1	1	1	1	1	0	0	1	1	0	1	1	1	1	Writes RAM 2
0	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	Writes RD6
1	0	0	0	0	0	0	0	1	1	1	1	1	0	1	1	Refreshes RAM 0
1	0	0	0	0	0	0	1	1	1	1	1	1	0	1	1	Refreshes RAM 0
1	0	0	0	0	0	1	0	1	1	1	1	1	0	1	1	Refreshes RAM 0
1	0	0	0	0	0	1	1	1	1	1	1	1	0	1	1	Refreshes RAM 0
1	0	0	0	0	1	0	0	1	1	1	1	1	0	1	1	Refreshes RAM 0
1	0	0	0	0	1	0	1	1	1	1	1	1	0	1	1	Refreshes RAM 0
1	0	0	0	0	1	1	0	1	1	1	1	1	0	1	1	Refreshes RAM 0
1	0	0	0	0	1	1	1	1	1	1	1	1	0	1	1	Refreshes RAM 0
1	0	0	0	1	0	0	0	1	1	1	1	1	0	1	1	Reads RAM 0 (ORG0 remap)
1	0	0	0	1	0	0	1	1	1	1	1	1	0	1	1	Reads RAM 0
1	0	0	0	1	0	1	0	1	1	1	1	1	1	0	1	Reads NOMEM
1	0	0	0	1	0	1	1	1	1	1	1	1	1	0	1	Reads NOMEM
1	0	0	0	1	1	0	0	1	1	1	1	1	1	0	1	Reads NOMEM
1	0	0	0	1	1	0	1	1	1	1	1	1	1	0	1	Reads NOMEM

Address Lines								Data Lines								
ORG0 H	JJ502	JJ501	BRD L	BRFSH L	Z80 A15	Z80 A14	Z80 A13	WE L	RD5 L	RD6 L	RAS2 L	RAS1 L	RAS0 L	NOMEM L	U516 L	
A7	**A6**	**A5**	**A4**	**A3**	**A2**	**A1**	**A0**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**	**Result**
1	0	0	0	1	1	1	0	1	1	1	1	1	1	0	1	Reads NOMEM
1	0	0	0	1	1	1	1	1	1	1	1	1	1	0	1	Reads NOMEM
1	0	0	1	0	0	0	0	1	1	1	1	1	0	1	1	Refreshes RAM 0
1	0	0	1	0	0	0	1	1	1	1	1	1	0	1	1	Refreshes RAM 0
1	0	0	1	0	0	1	0	1	1	1	1	1	0	1	1	Refreshes RAM 0
1	0	0	1	0	0	1	1	1	1	1	1	1	0	1	1	Refreshes RAM 0
1	0	0	1	0	1	0	0	1	1	1	1	1	0	1	1	Refreshes RAM 0
1	0	0	1	0	1	0	1	1	1	1	1	1	0	1	1	Refreshes RAM 0
1	0	0	1	0	1	1	0	1	1	1	1	1	0	1	1	Refreshes RAM 0
1	0	0	1	0	1	1	1	1	1	1	1	1	0	1	1	Refreshes RAM 0
1	0	0	1	1	0	0	0	0	1	1	1	1	0	1	1	Writes RAM 0 (ORG0 remap)
1	0	0	1	1	0	0	1	0	1	1	1	1	0	1	1	Writes RAM 0
1	0	0	1	1	0	1	0	1	1	1	1	1	1	1	1	No Device Selected
1	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	No Device Selected
1	0	0	1	1	1	0	0	1	1	1	1	1	1	1	1	No Device Selected
1	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	No Device Selected
1	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	No Device Selected
1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	No Device Selected
1	0	1	0	0	0	0	0	1	1	1	1	0	0	1	1	Refreshes RAM 0 and 1
1	0	1	0	0	0	0	1	1	1	1	1	0	0	1	1	Refreshes RAM 0 and 1
1	0	1	0	0	0	1	0	1	1	1	1	0	0	1	1	Refreshes RAM 0 and 1
1	0	1	0	0	0	1	1	1	1	1	1	0	0	1	1	Refreshes RAM 0 and 1
1	0	1	0	0	1	0	0	1	1	1	1	0	0	1	1	Refreshes RAM 0 and 1
1	0	1	0	0	1	0	1	1	1	1	1	0	0	1	1	Refreshes RAM 0 and 1
1	0	1	0	0	1	1	0	1	1	1	1	0	0	1	1	Refreshes RAM 0 and 1
1	0	1	0	0	1	1	1	1	1	1	1	0	0	1	1	Refreshes RAM 0 and 1
1	0	1	0	1	0	0	0	1	1	1	1	0	1	1	1	Reads RAM 1 (ORG0 remap)
1	0	1	0	1	0	0	1	1	1	1	1	1	0	1	1	Reads RAM 0
1	0	1	0	1	0	1	0	1	1	1	1	1	0	1	1	Reads RAM 0
1	0	1	0	1	0	1	1	1	1	1	1	0	1	1	1	Reads RAM 1
1	0	1	0	1	1	0	0	1	1	1	1	1	1	0	1	Reads NOMEM
1	0	1	0	1	1	0	1	1	1	1	1	1	1	0	1	Reads NOMEM
1	0	1	0	1	1	1	0	1	1	1	1	1	1	0	1	Reads NOMEM
1	0	1	0	1	1	1	1	1	1	1	1	1	1	0	1	Reads NOMEM
1	0	1	1	0	0	0	0	1	1	1	1	0	0	1	1	Refreshes RAM 0 and 1
1	0	1	1	0	0	0	1	1	1	1	1	0	0	1	1	Refreshes RAM 0 and 1
1	0	1	1	0	0	1	0	1	1	1	1	0	0	1	1	Refreshes RAM 0 and 1

Address Lines								Data Lines								
ORG0 H	JJ502	JJ501	BRD L	BRFSH L	Z80 A15	Z80 A14	Z80 A13	WE L	RD5 L	RD6 L	RAS2 L	RAS1 L	RAS0 L	NOMEM L	U516 L	
A7	**A6**	**A5**	**A4**	**A3**	**A2**	**A1**	**A0**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**	**Result**
1	0	1	1	0	0	1	1	1	1	1	1	0	0	1	1	Refreshes RAM 0 and 1
1	0	1	1	0	1	0	0	1	1	1	1	0	0	1	1	Refreshes RAM 0 and 1
1	0	1	1	0	1	0	1	1	1	1	1	0	0	1	1	Refreshes RAM 0 and 1
1	0	1	1	0	1	1	0	1	1	1	1	0	0	1	1	Refreshes RAM 0 and 1
1	0	1	1	0	1	1	1	1	1	1	1	0	0	1	1	Refreshes RAM 0 and 1
1	0	1	1	1	0	0	0	0	1	1	1	0	1	1	1	Writes RAM 1 (ORG0 remap)
1	0	1	1	1	0	0	1	0	1	1	1	1	0	1	1	Writes RAM 0
1	0	1	1	1	0	1	0	0	1	1	1	1	0	1	1	Writes RAM 0
1	0	1	1	1	0	1	1	0	1	1	1	0	1	1	1	Writes RAM 1
1	0	1	1	1	1	0	0	1	1	1	1	1	1	1	1	No Device Selected
1	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	No Device Selected
1	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	No Device Selected
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	No Device Selected
1	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	Refreshes RAM 0, 1 and 2
1	1	0	0	0	0	0	1	1	1	1	0	0	0	1	1	Refreshes RAM 0, 1 and 2
1	1	0	0	0	0	1	0	1	1	1	0	0	0	1	1	Refreshes RAM 0, 1 and 2
1	1	0	0	0	0	1	1	1	1	1	0	0	0	1	1	Refreshes RAM 0, 1 and 2
1	1	0	0	0	1	0	0	1	1	1	0	0	0	1	1	Refreshes RAM 0, 1 and 2
1	1	0	0	0	1	0	1	1	1	1	0	0	0	1	1	Refreshes RAM 0, 1 and 2
1	1	0	0	0	1	1	0	1	1	1	0	0	0	1	1	Refreshes RAM 0, 1 and 2
1	1	0	0	0	1	1	1	1	1	1	0	0	0	1	1	Refreshes RAM 0, 1 and 2
1	1	0	0	1	0	0	0	1	1	1	0	1	1	1	1	Reads RAM 2 (ORG0 remap)
1	1	0	0	1	0	0	1	1	1	1	1	1	0	1	1	Reads RAM 0
1	1	0	0	1	0	1	0	1	1	1	1	1	0	1	1	Reads RAM 0
1	1	0	0	1	0	1	1	1	1	1	1	0	1	1	1	Reads RAM 1
1	1	0	0	1	1	0	0	1	1	1	1	0	1	1	1	Reads RAM 1
1	1	0	0	1	1	0	1	1	1	1	0	1	1	1	1	Reads RAM 2
1	1	0	0	1	1	1	0	1	1	1	1	1	1	0	1	Reads NOMEM
1	1	0	0	1	1	1	1	1	1	1	1	1	1	0	1	Reads NOMEM
1	1	0	1	0	0	0	0	1	1	1	0	0	0	1	1	Refreshes RAM 0, 1 and 2
1	1	0	1	0	0	0	1	1	1	1	0	0	0	1	1	Refreshes RAM 0, 1 and 2
1	1	0	1	0	0	1	0	1	1	1	0	0	0	1	1	Refreshes RAM 0, 1 and 2
1	1	0	1	0	0	1	1	1	1	1	0	0	0	1	1	Refreshes RAM 0, 1 and 2
1	1	0	1	0	1	0	0	1	1	1	0	0	0	1	1	Refreshes RAM 0, 1 and 2
1	1	0	1	0	1	0	1	1	1	1	0	0	0	1	1	Refreshes RAM 0, 1 and 2
1	1	0	1	0	1	1	0	1	1	1	0	0	0	1	1	Refreshes RAM 0, 1 and 2
1	1	0	1	0	1	1	1	1	1	1	0	0	0	1	1	Refreshes RAM 0, 1 and 2

Address Lines								Data Lines								
ORG0 H	JJ502	JJ501	BRD L	BRFSH L	Z80 A15	Z80 A14	Z80 A13	WE L	RD5 L	RD6 L	RAS2 L	RAS1 L	RAS0 L	NOMEM L	U516 L	
A7	A6	A5	A4	A3	A2	A1	A0	D7	D6	D5	D4	D3	D2	D1	D0	Result
1	1	0	1	1	0	0	0	0	1	1	0	1	1	1	1	Writes RAM 2 (ORG0 remap)
1	1	0	1	1	0	0	1	0	1	1	1	1	0	1	1	Writes RAM 0
1	1	0	1	1	0	1	0	0	1	1	1	1	0	1	1	Writes RAM 0
1	1	0	1	1	0	1	1	0	1	1	1	0	1	1	1	Writes RAM 1
1	1	0	1	1	1	0	0	0	1	1	1	0	1	1	1	Writes RAM 1
1	1	0	1	1	1	0	1	0	1	1	0	1	1	1	1	Writes RAM 2
1	1	0	1	1	1	1	0	1	1	1	1	1	1	1	1	No Device Selected
1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	No Device Selected
1	1	1	0	0	0	0	0	1	1	0	0	0	0	1	1	Refreshes RAM 0, 1, 2 and RD6
1	1	1	0	0	0	0	1	1	1	0	0	0	0	1	1	Refreshes RAM 0, 1, 2 and RD6
1	1	1	0	0	0	1	0	1	1	0	0	0	0	1	1	Refreshes RAM 0, 1, 2 and RD6
1	1	1	0	0	0	1	1	1	1	0	0	0	0	1	1	Refreshes RAM 0, 1, 2 and RD6
1	1	1	0	0	1	0	0	1	1	0	0	0	0	1	1	Refreshes RAM 0, 1, 2 and RD6
1	1	1	0	0	1	0	1	1	1	0	0	0	0	1	1	Refreshes RAM 0, 1, 2 and RD6
1	1	1	0	0	1	1	0	1	1	0	0	0	0	1	1	Refreshes RAM 0, 1, 2 and RD6
1	1	1	0	0	1	1	1	1	1	0	0	0	0	1	1	Refreshes RAM 0, 1, 2 and RD6
1	1	1	0	1	0	0	0	1	1	0	1	1	1	1	1	Reads RD6
1	1	1	0	1	0	0	1	1	1	1	1	1	0	1	1	Reads RAM 0
1	1	1	0	1	0	1	0	1	1	1	1	1	0	1	1	Reads RAM 0
1	1	1	0	1	0	1	1	1	1	1	1	0	1	1	1	Reads RAM 1
1	1	1	0	1	1	0	0	1	1	1	1	0	1	1	1	Reads RAM 1
1	1	1	0	1	1	0	1	1	1	1	0	1	1	1	1	Reads RAM 2
1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	1	Reads RAM 2
1	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	Reads RD6
1	1	1	1	0	0	0	0	1	1	0	0	0	0	1	1	Refreshes RAM 0, 1, 2 and RD6
1	1	1	1	0	0	0	1	1	1	0	0	0	0	1	1	Refreshes RAM 0, 1, 2 and RD6
1	1	1	1	0	0	1	0	1	1	0	0	0	0	1	1	Refreshes RAM 0, 1, 2 and RD6
1	1	1	1	0	0	1	1	1	1	0	0	0	0	1	1	Refreshes RAM 0, 1, 2 and RD6
1	1	1	1	0	1	0	0	1	1	0	0	0	0	1	1	Refreshes RAM 0, 1, 2 and RD6
1	1	1	1	0	1	0	1	1	1	0	0	0	0	1	1	Refreshes RAM 0, 1, 2 and RD6
1	1	1	1	0	1	1	0	1	1	0	0	0	0	1	1	Refreshes RAM 0, 1, 2 and RD6
1	1	1	1	0	1	1	1	1	1	0	0	0	0	1	1	Refreshes RAM 0, 1, 2 and RD6
1	1	1	1	1	0	0	0	0	1	0	1	1	1	1	1	Writes RD6 (ORG0 remap)
1	1	1	1	1	0	0	1	0	1	1	1	1	0	1	1	Writes RAM 0
1	1	1	1	1	0	1	0	0	1	1	1	1	0	1	1	Writes RAM 0
1	1	1	1	1	0	1	1	0	1	1	1	0	1	1	1	Writes RAM 1
1	1	1	1	1	1	0	0	0	1	1	1	0	1	1	1	Writes RAM 1

Address Lines								Data Lines								
ORG0 H	JJ502	JJ501	BRD L	BRFSH L	Z80 A15	Z80 A14	Z80 A13	WE L	RD5 L	RD6 L	RAS2 L	RAS1 L	RAS0 L	NOMEM L	U516 L	
A7	**A6**	**A5**	**A4**	**A3**	**A2**	**A1**	**A0**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**	**Result**
1	1	1	1	1	1	0	1	0	1	1	0	1	1	1	1	Writes to RAM 2
1	1	1	1	1	1	1	0	0	1	1	0	1	1	1	1	Writes to RAM 2
1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	Writes to RD6

Appendix B: U516 Secondary Decoder ROM 444-41

The table below was created by removing the U516 decoder ROM, part 444-41, from my H89's CPU logic board, connecting it to a test harness, and cycling through the ROM's 32 address line states to capture the corresponding data line values.

The *Result* column indicates the action taken in response to each input state.

Address Lines					Data Lines								
FMWE H	U517 WE L	Z80 A12	Z80 A11	Z80 A10	WE L	Unused	Unused	FPY ROM L	FPY RAM L	OPT RAM L	OPT ROM L	SYS ROM L	
A4	**A3**	**A2**	**A1**	**A0**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**	**Result**
0	0	0	0	0	1	1	1	1	1	1	1	1	No device selected
0	0	0	0	1	1	1	1	1	1	1	1	1	No device selected
0	0	0	1	0	1	1	1	1	1	1	1	1	No device selected
0	0	0	1	1	1	1	1	1	1	1	1	1	No device selected
0	0	1	0	0	1	1	1	1	1	0	1	1	Writes to optional RAM
0	0	1	0	1	1	1	1	1	1	1	1	1	No device selected
0	0	1	1	0	1	1	1	1	1	1	1	1	No device selected
0	0	1	1	1	1	1	1	1	1	1	1	1	No device selected
0	1	0	0	0	1	1	1	1	1	1	1	0	Reads from system ROM
0	1	0	0	1	1	1	1	1	1	1	1	0	Reads from system ROM
0	1	0	1	0	1	1	1	1	1	1	0	1	Reads from optional ROM
0	1	0	1	1	1	1	1	1	1	1	0	1	Reads from optional ROM
0	1	1	0	0	1	1	1	1	1	0	1	1	Reads from optional RAM
0	1	1	0	1	1	1	1	1	0	1	1	1	Reads from floppy RAM
0	1	1	1	0	1	1	1	0	1	1	1	1	Reads from floppy ROM
0	1	1	1	1	1	1	1	0	1	1	1	1	Reads from floppy ROM
1	0	0	0	0	1	1	1	1	1	1	1	1	No device selected
1	0	0	0	1	1	1	1	1	1	1	1	1	No device selected
1	0	0	1	0	1	1	1	1	1	1	1	1	No device selected
1	0	0	1	1	1	1	1	1	1	1	1	1	No device selected
1	0	1	0	0	1	1	1	1	1	0	1	1	Reads from optional RAM
1	0	1	0	1	0	1	1	1	0	1	1	1	Writes to floppy RAM
1	0	1	1	0	1	1	1	1	1	1	1	1	No device selected
1	0	1	1	1	1	1	1	1	1	1	1	1	No device selected
1	1	0	0	0	1	1	1	1	1	1	1	0	Reads from system ROM
1	1	0	0	1	1	1	1	1	1	1	1	0	Reads from system ROM
1	1	0	1	0	1	1	1	1	1	1	0	1	Reads from optional ROM
1	1	0	1	1	1	1	1	1	1	1	0	1	Reads from optional ROM

Address Lines					Data Lines								
FMWE H	U517 WE L	Z80 A12	Z80 A11	Z80 A10	WE L	Unused	Unused	FPY ROM L	FPY RAM L	OPT RAM L	OPT ROM L	SYS ROM L	
A4	**A3**	**A2**	**A1**	**A0**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**	**Result**
1	1	1	0	0	1	1	1	1	1	0	1	1	Reads from optional RAM
1	1	1	0	1	1	1	1	1	0	1	1	1	Reads from floppy RAM
1	1	1	1	0	1	1	1	0	1	1	1	1	Reads from floppy ROM
1	1	1	1	1	1	1	1	0	1	1	1	1	Reads from floppy ROM

Appendix C: U550 I/O Decoder ROM 444-61

The table below was created by removing the U550 I/O decoder ROM, part 444-61, from my H89's CPU logic board, connecting it to a test harness, and cycling through the ROM's 256 address line states to capture the corresponding data line values.

Port numbers are shown in Octal. The *Activates* column indicates which I/O device(s) is activated in response to each input state. Note that for most address states the D0-D7 data lines are all inactive/high, indicating no device selected.

Address Lines									Data Lines								
LA7	LA6	LA5	LA4	LA3	LA2	LA1	LA0		I/O FLPY L	I/O CASS L	I/O LP L	I/O SERL 0 L	I/O SERT 1 L	I/O TERM L	NMI L	GPP L	
A7	A6	A5	A4	A3	A2	A1	A0	Port	D7	D6	D5	D4	D3	D2	D1	D0	Activates
0	0	0	0	0	0	0	0	000	1	1	1	1	1	1	1	1	None
0	0	0	0	0	0	0	1	001	1	1	1	1	1	1	1	1	None
0	0	0	0	0	0	1	0	002	1	1	1	1	1	1	1	1	None
0	0	0	0	0	0	1	1	003	1	1	1	1	1	1	1	1	None
0	0	0	0	0	1	0	0	004	1	1	1	1	1	1	1	1	None
0	0	0	0	0	1	0	1	005	1	1	1	1	1	1	1	1	None
0	0	0	0	0	1	1	0	006	1	1	1	1	1	1	1	1	None
0	0	0	0	0	1	1	1	007	1	1	1	1	1	1	1	1	None
0	0	0	0	1	0	0	0	010	1	1	1	1	1	1	1	1	None
0	0	0	0	1	0	0	1	011	1	1	1	1	1	1	1	1	None
0	0	0	0	1	0	1	0	012	1	1	1	1	1	1	1	1	None
0	0	0	0	1	0	1	1	013	1	1	1	1	1	1	1	1	None
0	0	0	0	1	1	0	0	014	1	1	1	1	1	1	1	1	None
0	0	0	0	1	1	0	1	015	1	1	1	1	1	1	1	1	None
0	0	0	0	1	1	1	0	016	1	1	1	1	1	1	1	1	None
0	0	0	0	1	1	1	1	017	1	1	1	1	1	1	1	1	None
0	0	0	1	0	0	0	0	020	1	1	1	1	1	1	1	1	None
0	0	0	1	0	0	0	1	021	1	1	1	1	1	1	1	1	None
0	0	0	1	0	0	1	0	022	1	1	1	1	1	1	1	1	None
0	0	0	1	0	0	1	1	023	1	1	1	1	1	1	1	1	None
0	0	0	1	0	1	0	0	024	1	1	1	1	1	1	1	1	None
0	0	0	1	0	1	0	1	025	1	1	1	1	1	1	1	1	None
0	0	0	1	0	1	1	0	026	1	1	1	1	1	1	1	1	None
0	0	0	1	0	1	1	1	027	1	1	1	1	1	1	1	1	None
0	0	0	1	1	0	0	0	030	1	1	1	1	1	1	1	1	None
0	0	0	1	1	0	0	1	031	1	1	1	1	1	1	1	1	None
0	0	0	1	1	0	1	0	032	1	1	1	1	1	1	1	1	None

Address Lines									Data Lines								
LA7	LA6	LA5	LA4	LA3	LA2	LA1	LA0		I/O FLPY L	I/O CASS L	I/O LP L	I/O SERL 0 L	I/O SERT 1 L	I/O TERM L	NMI L	GPP L	
A7	A6	A5	A4	A3	A2	A1	A0	Port	D7	D6	D5	D4	D3	D2	D1	D0	Activates
0	0	0	1	1	0	1	1	033	1	1	1	1	1	1	1	1	None
0	0	0	1	1	1	0	0	034	1	1	1	1	1	1	1	1	None
0	0	0	1	1	1	0	1	035	1	1	1	1	1	1	1	1	None
0	0	0	1	1	1	1	0	036	1	1	1	1	1	1	1	1	None
0	0	0	1	1	1	1	1	037	1	1	1	1	1	1	1	1	None
0	0	1	0	0	0	0	0	040	1	1	1	1	1	1	1	1	None
0	0	1	0	0	0	0	1	041	1	1	1	1	1	1	1	1	None
0	0	1	0	0	0	1	0	042	1	1	1	1	1	1	1	1	None
0	0	1	0	0	0	1	1	043	1	1	1	1	1	1	1	1	None
0	0	1	0	0	1	0	0	044	1	1	1	1	1	1	1	1	None
0	0	1	0	0	1	0	1	045	1	1	1	1	1	1	1	1	None
0	0	1	0	0	1	1	0	046	1	1	1	1	1	1	1	1	None
0	0	1	0	0	1	1	1	047	1	1	1	1	1	1	1	1	None
0	0	1	0	1	0	0	0	050	1	1	1	1	1	1	1	1	None
0	0	1	0	1	0	0	1	051	1	1	1	1	1	1	1	1	None
0	0	1	0	1	0	1	0	052	1	1	1	1	1	1	1	1	None
0	0	1	0	1	0	1	1	053	1	1	1	1	1	1	1	1	None
0	0	1	0	1	1	0	0	054	1	1	1	1	1	1	1	1	None
0	0	1	0	1	1	0	1	055	1	1	1	1	1	1	1	1	None
0	0	1	0	1	1	1	0	056	1	1	1	1	1	1	1	1	None
0	0	1	0	1	1	1	1	057	1	1	1	1	1	1	1	1	None
0	0	1	1	0	0	0	0	060	1	1	1	1	1	1	1	1	None
0	0	1	1	0	0	0	1	061	1	1	1	1	1	1	1	1	None
0	0	1	1	0	0	1	0	062	1	1	1	1	1	1	1	1	None
0	0	1	1	0	0	1	1	063	1	1	1	1	1	1	1	1	None
0	0	1	1	0	1	0	0	064	1	1	1	1	1	1	1	1	None
0	0	1	1	0	1	0	1	065	1	1	1	1	1	1	1	1	None
0	0	1	1	0	1	1	0	066	1	1	1	1	1	1	1	1	None
0	0	1	1	0	1	1	1	067	1	1	1	1	1	1	1	1	None
0	0	1	1	1	0	0	0	070	1	1	1	1	1	1	1	1	None
0	0	1	1	1	0	0	1	071	1	1	1	1	1	1	1	1	None
0	0	1	1	1	0	1	0	072	1	1	1	1	1	1	1	1	None
0	0	1	1	1	0	1	1	073	1	1	1	1	1	1	1	1	None
0	0	1	1	1	1	0	0	074	1	1	1	1	1	1	1	1	None

Address Lines									Data Lines								
LA7	LA6	LA5	LA4	LA3	LA2	LA1	LA0		I/O FLPY L	I/O CASS L	I/O LP L	I/O SERL 0 L	I/O SERT 1 L	I/O TERM L	NMI L	GPP L	
A7	A6	A5	A4	A3	A2	A1	A0	Port	D7	D6	D5	D4	D3	D2	D1	D0	Activates
0	0	1	1	1	1	0	1	075	1	1	1	1	1	1	1	1	None
0	0	1	1	1	1	1	0	076	1	1	1	1	1	1	1	1	None
0	0	1	1	1	1	1	1	077	1	1	1	1	1	1	1	1	None
0	1	0	0	0	0	0	0	100	1	1	1	1	1	1	1	1	None
0	1	0	0	0	0	0	1	101	1	1	1	1	1	1	1	1	None
0	1	0	0	0	0	1	0	102	1	1	1	1	1	1	1	1	None
0	1	0	0	0	0	1	1	103	1	1	1	1	1	1	1	1	None
0	1	0	0	0	1	0	0	104	1	1	1	1	1	1	1	1	None
0	1	0	0	0	1	0	1	105	1	1	1	1	1	1	1	1	None
0	1	0	0	0	1	1	0	106	1	1	1	1	1	1	1	1	None
0	1	0	0	0	1	1	1	107	1	1	1	1	1	1	1	1	None
0	1	0	0	1	0	0	0	110	1	1	1	1	1	1	1	1	None
0	1	0	0	1	0	0	1	111	1	1	1	1	1	1	1	1	None
0	1	0	0	1	0	1	0	112	1	1	1	1	1	1	1	1	None
0	1	0	0	1	0	1	1	113	1	1	1	1	1	1	1	1	None
0	1	0	0	1	1	0	0	114	1	1	1	1	1	1	1	1	None
0	1	0	0	1	1	0	1	115	1	1	1	1	1	1	1	1	None
0	1	0	0	1	1	1	0	116	1	1	1	1	1	1	1	1	None
0	1	0	0	1	1	1	1	117	1	1	1	1	1	1	1	1	None
0	1	0	1	0	0	0	0	120	1	1	1	1	1	1	1	1	None
0	1	0	1	0	0	0	1	121	1	1	1	1	1	1	1	1	None
0	1	0	1	0	0	1	0	122	1	1	1	1	1	1	1	1	None
0	1	0	1	0	0	1	1	123	1	1	1	1	1	1	1	1	None
0	1	0	1	0	1	0	0	124	1	1	1	1	1	1	1	1	None
0	1	0	1	0	1	0	1	125	1	1	1	1	1	1	1	1	None
0	1	0	1	0	1	1	0	126	1	1	1	1	1	1	1	1	None
0	1	0	1	0	1	1	1	127	1	1	1	1	1	1	1	1	None
0	1	0	1	1	0	0	0	130	1	1	1	1	1	1	1	1	None
0	1	0	1	1	0	0	1	131	1	1	1	1	1	1	1	1	None
0	1	0	1	1	0	1	0	132	1	1	1	1	1	1	1	1	None
0	1	0	1	1	0	1	1	133	1	1	1	1	1	1	1	1	None
0	1	0	1	1	1	0	0	134	1	1	1	1	1	1	1	1	None
0	1	0	1	1	1	0	1	135	1	1	1	1	1	1	1	1	None
0	1	0	1	1	1	1	0	136	1	1	1	1	1	1	1	1	None

Address Lines									Data Lines								
LA7	LA6	LA5	LA4	LA3	LA2	LA1	LA0		I/O FLPY L	I/O CASS L	I/O LP L	I/O SERL 0 L	I/O SERT 1 L	I/O TERM L	NMI L	GPP L	
A7	A6	A5	A4	A3	A2	A1	A0	Port	D7	D6	D5	D4	D3	D2	D1	D0	Activates
0	1	0	1	1	1	1	1	137	1	1	1	1	1	1	1	1	None
0	1	1	0	0	0	0	0	140	1	1	1	1	1	1	1	1	None
0	1	1	0	0	0	0	1	141	1	1	1	1	1	1	1	1	None
0	1	1	0	0	0	1	0	142	1	1	1	1	1	1	1	1	None
0	1	1	0	0	0	1	1	143	1	1	1	1	1	1	1	1	None
0	1	1	0	0	1	0	0	144	1	1	1	1	1	1	1	1	None
0	1	1	0	0	1	0	1	145	1	1	1	1	1	1	1	1	None
0	1	1	0	0	1	1	0	146	1	1	1	1	1	1	1	1	None
0	1	1	0	0	1	1	1	147	1	1	1	1	1	1	1	1	None
0	1	1	0	1	0	0	0	150	1	1	1	1	1	1	1	1	None
0	1	1	0	1	0	0	1	151	1	1	1	1	1	1	1	1	None
0	1	1	0	1	0	1	0	152	1	1	1	1	1	1	1	1	None
0	1	1	0	1	0	1	1	153	1	1	1	1	1	1	1	1	None
0	1	1	0	1	1	0	0	154	1	1	1	1	1	1	1	1	None
0	1	1	0	1	1	0	1	155	1	1	1	1	1	1	1	1	None
0	1	1	0	1	1	1	0	156	1	1	1	1	1	1	1	1	None
0	1	1	0	1	1	1	1	157	1	1	1	1	1	1	1	1	None
0	1	1	1	0	0	0	0	160	1	1	1	1	1	1	1	1	None
0	1	1	1	0	0	0	1	161	1	1	1	1	1	1	1	1	None
0	1	1	1	0	0	1	0	162	1	1	1	1	1	1	1	1	None
0	1	1	1	0	0	1	1	163	1	1	1	1	1	1	1	1	None
0	1	1	1	0	1	0	0	164	1	1	1	1	1	1	1	1	None
0	1	1	1	0	1	0	1	165	1	1	1	1	1	1	1	1	None
0	1	1	1	0	1	1	0	166	1	1	1	1	1	1	1	1	None
0	1	1	1	0	1	1	1	167	1	1	1	1	1	1	1	1	None
0	1	1	1	1	0	0	0	170	1	0	1	1	1	1	1	1	Disk card in P510 or P511
0	1	1	1	1	0	0	1	171	1	0	1	1	1	1	1	1	Disk card in P510 or P511
0	1	1	1	1	0	1	0	172	1	0	1	1	1	1	1	1	Disk card in P510 or P511
0	1	1	1	1	0	1	1	173	1	0	1	1	1	1	1	1	Disk card in P510 or P511
0	1	1	1	1	1	0	0	174	0	0	1	1	1	1	1	1	Disk card in P512 (D7) and/or Disk card in P510 or P511 (D6)
0	1	1	1	1	1	0	1	175	0	0	1	1	1	1	1	1	Disk card in P512 (D7) and/or Disk card in P510 or P511 (D6)
0	1	1	1	1	1	1	0	176	0	0	1	1	1	1	1	1	Disk card in P512 (D7) and/or Disk card in P510 or P511 (D6)

Address Lines									Data Lines								
LA7	LA6	LA5	LA4	LA3	LA2	LA1	LA0		I/O FLPY L	I/O CASS L	I/O LP L	I/O SERL 0 L	I/O SERT 1 L	I/O TERM L	NMI L	GPP L	
A7	A6	A5	A4	A3	A2	A1	A0	Port	D7	D6	D5	D4	D3	D2	D1	D0	Activates
0	1	1	1	1	1	1	1	177	0	0	1	1	1	1	1	1	Disk card in P512 (D7) and/or Disk card in P510 or P511 (D6)
1	0	0	0	0	0	0	0	200	1	1	1	1	1	1	1	1	None
1	0	0	0	0	0	0	1	201	1	1	1	1	1	1	1	1	None
1	0	0	0	0	0	1	0	202	1	1	1	1	1	1	1	1	None
1	0	0	0	0	0	1	1	203	1	1	1	1	1	1	1	1	None
1	0	0	0	0	1	0	0	204	1	1	1	1	1	1	1	1	None
1	0	0	0	0	1	0	1	205	1	1	1	1	1	1	1	1	None
1	0	0	0	0	1	1	0	206	1	1	1	1	1	1	1	1	None
1	0	0	0	0	1	1	1	207	1	1	1	1	1	1	1	1	None
1	0	0	0	1	0	0	0	210	1	1	1	1	1	1	1	1	None
1	0	0	0	1	0	0	1	211	1	1	1	1	1	1	1	1	None
1	0	0	0	1	0	1	0	212	1	1	1	1	1	1	1	1	None
1	0	0	0	1	0	1	1	213	1	1	1	1	1	1	1	1	None
1	0	0	0	1	1	0	0	214	1	1	1	1	1	1	1	1	None
1	0	0	0	1	1	0	1	215	1	1	1	1	1	1	1	1	None
1	0	0	0	1	1	1	0	216	1	1	1	1	1	1	1	1	None
1	0	0	0	1	1	1	1	217	1	1	1	1	1	1	1	1	None
1	0	0	1	0	0	0	0	220	1	1	1	1	1	1	1	1	None
1	0	0	1	0	0	0	1	221	1	1	1	1	1	1	1	1	None
1	0	0	1	0	0	1	0	222	1	1	1	1	1	1	1	1	None
1	0	0	1	0	0	1	1	223	1	1	1	1	1	1	1	1	None
1	0	0	1	0	1	0	0	224	1	1	1	1	1	1	1	1	None
1	0	0	1	0	1	0	1	225	1	1	1	1	1	1	1	1	None
1	0	0	1	0	1	1	0	226	1	1	1	1	1	1	1	1	None
1	0	0	1	0	1	1	1	227	1	1	1	1	1	1	1	1	None
1	0	0	1	1	0	0	0	230	1	1	1	1	1	1	1	1	None
1	0	0	1	1	0	0	1	231	1	1	1	1	1	1	1	1	None
1	0	0	1	1	0	1	0	232	1	1	1	1	1	1	1	1	None
1	0	0	1	1	0	1	1	233	1	1	1	1	1	1	1	1	None
1	0	0	1	1	1	0	0	234	1	1	1	1	1	1	1	1	None
1	0	0	1	1	1	0	1	235	1	1	1	1	1	1	1	1	None
1	0	0	1	1	1	1	0	236	1	1	1	1	1	1	1	1	None
1	0	0	1	1	1	1	1	237	1	1	1	1	1	1	1	1	None
1	0	1	0	0	0	0	0	240	1	1	1	1	1	1	1	1	None

Address Lines									Data Lines								
LA7	LA6	LA5	LA4	LA3	LA2	LA1	LA0		I/O FLPY L	I/O CASS L	I/O LP L	I/O SERL 0 L	I/O SERT 1 L	I/O TERM L	NMI L	GPP L	
A7	A6	A5	A4	A3	A2	A1	A0	Port	D7	D6	D5	D4	D3	D2	D1	D0	Activates
1	0	1	0	0	0	0	1	241	1	1	1	1	1	1	1	1	None
1	0	1	0	0	0	1	0	242	1	1	1	1	1	1	1	1	None
1	0	1	0	0	0	1	1	243	1	1	1	1	1	1	1	1	None
1	0	1	0	0	1	0	0	244	1	1	1	1	1	1	1	1	None
1	0	1	0	0	1	0	1	245	1	1	1	1	1	1	1	1	None
1	0	1	0	0	1	1	0	246	1	1	1	1	1	1	1	1	None
1	0	1	0	0	1	1	1	247	1	1	1	1	1	1	1	1	None
1	0	1	0	1	0	0	0	250	1	1	1	1	1	1	1	1	None
1	0	1	0	1	0	0	1	251	1	1	1	1	1	1	1	1	None
1	0	1	0	1	0	1	0	252	1	1	1	1	1	1	1	1	None
1	0	1	0	1	0	1	1	253	1	1	1	1	1	1	1	1	None
1	0	1	0	1	1	0	0	254	1	1	1	1	1	1	1	1	None
1	0	1	0	1	1	0	1	255	1	1	1	1	1	1	1	1	None
1	0	1	0	1	1	1	0	256	1	1	1	1	1	1	1	1	None
1	0	1	0	1	1	1	1	257	1	1	1	1	1	1	1	1	None
1	0	1	1	0	0	0	0	260	1	1	1	1	1	1	1	1	None
1	0	1	1	0	0	0	1	261	1	1	1	1	1	1	1	1	None
1	0	1	1	0	0	1	0	262	1	1	1	1	1	1	1	1	None
1	0	1	1	0	0	1	1	263	1	1	1	1	1	1	1	1	None
1	0	1	1	0	1	0	0	264	1	1	1	1	1	1	1	1	None
1	0	1	1	0	1	0	1	265	1	1	1	1	1	1	1	1	None
1	0	1	1	0	1	1	0	266	1	1	1	1	1	1	1	1	None
1	0	1	1	0	1	1	1	267	1	1	1	1	1	1	1	1	None
1	0	1	1	1	0	0	0	270	1	1	1	1	1	1	1	1	None
1	0	1	1	1	0	0	1	271	1	1	1	1	1	1	1	1	None
1	0	1	1	1	0	1	0	272	1	1	1	1	1	1	1	1	None
1	0	1	1	1	0	1	1	273	1	1	1	1	1	1	1	1	None
1	0	1	1	1	1	0	0	274	1	1	1	1	1	1	1	1	None
1	0	1	1	1	1	0	1	275	1	1	1	1	1	1	1	1	None
1	0	1	1	1	1	1	0	276	1	1	1	1	1	1	1	1	None
1	0	1	1	1	1	1	1	277	1	1	1	1	1	1	1	1	None
1	1	0	0	0	0	0	0	300	1	1	1	1	1	1	1	1	None
1	1	0	0	0	0	0	1	301	1	1	1	1	1	1	1	1	None
1	1	0	0	0	0	1	0	302	1	1	1	1	1	1	1	1	None

Address Lines									Data Lines								
LA7	LA6	LA5	LA4	LA3	LA2	LA1	LA0		I/O FLPY L	I/O CASS L	I/O LP L	I/O SERL 0 L	I/O SERT 1 L	I/O TERM L	NMI L	GPP L	
A7	A6	A5	A4	A3	A2	A1	A0	Port	D7	D6	D5	D4	D3	D2	D1	D0	Activates
1	1	0	0	0	0	1	1	303	1	1	1	1	1	1	1	1	None
1	1	0	0	0	1	0	0	304	1	1	1	1	1	1	1	1	None
1	1	0	0	0	1	0	1	305	1	1	1	1	1	1	1	1	None
1	1	0	0	0	1	1	0	306	1	1	1	1	1	1	1	1	None
1	1	0	0	0	1	1	1	307	1	1	1	1	1	1	1	1	None
1	1	0	0	1	0	0	0	310	1	1	1	1	1	1	1	1	None
1	1	0	0	1	0	0	1	311	1	1	1	1	1	1	1	1	None
1	1	0	0	1	0	1	0	312	1	1	1	1	1	1	1	1	None
1	1	0	0	1	0	1	1	313	1	1	1	1	1	1	1	1	None
1	1	0	0	1	1	0	0	314	1	1	1	1	1	1	1	1	None
1	1	0	0	1	1	0	1	315	1	1	1	1	1	1	1	1	None
1	1	0	0	1	1	1	0	316	1	1	1	1	1	1	1	1	None
1	1	0	0	1	1	1	1	317	1	1	1	1	1	1	1	1	None
1	1	0	1	0	0	0	0	320	1	1	1	0	1	1	1	1	H88-3 Serial I/O card U603 ACE (DCE)
1	1	0	1	0	0	0	1	321	1	1	1	0	1	1	1	1	H88-3 Serial I/O card U603 ACE (DCE)
1	1	0	1	0	0	1	0	322	1	1	1	0	1	1	1	1	H88-3 Serial I/O card U603 ACE (DCE)
1	1	0	1	0	0	1	1	323	1	1	1	0	1	1	1	1	H88-3 Serial I/O card U603 ACE (DCE)
1	1	0	1	0	1	0	0	324	1	1	1	0	1	1	1	1	H88-3 Serial I/O card U603 ACE (DCE)
1	1	0	1	0	1	0	1	325	1	1	1	0	1	1	1	1	H88-3 Serial I/O card U603 ACE (DCE)
1	1	0	1	0	1	1	0	326	1	1	1	0	1	1	1	1	H88-3 Serial I/O card U603 ACE (DCE)
1	1	0	1	0	1	1	1	327	1	1	1	0	1	1	1	1	H88-3 Serial I/O card U603 ACE (DCE)
1	1	0	1	1	0	0	0	330	1	1	1	1	0	1	1	1	H88-3 Serial I/O card U604 ACE (DTE)
1	1	0	1	1	0	0	1	331	1	1	1	1	0	1	1	1	H88-3 Serial I/O card U604 ACE (DTE)
1	1	0	1	1	0	1	0	332	1	1	1	1	0	1	1	1	H88-3 Serial I/O card U604 ACE (DTE)
1	1	0	1	1	0	1	1	333	1	1	1	1	0	1	1	1	H88-3 Serial I/O card U604 ACE (DTE)
1	1	0	1	1	1	0	0	334	1	1	1	1	0	1	1	1	H88-3 Serial I/O card U604 ACE (DTE)
1	1	0	1	1	1	0	1	335	1	1	1	1	0	1	1	1	H88-3 Serial I/O card U604 ACE (DTE)
1	1	0	1	1	1	1	0	336	1	1	1	1	0	1	1	1	H88-3 Serial I/O card U604 ACE (DTE)
1	1	0	1	1	1	1	1	337	1	1	1	1	0	1	1	1	H88-3 Serial I/O card U604 ACE (DTE)
1	1	1	0	0	0	0	0	340	1	1	0	1	1	1	1	1	H88-3 Serial I/O card U602 ACE (LP)
1	1	1	0	0	0	0	1	341	1	1	0	1	1	1	1	1	H88-3 Serial I/O card U602 ACE (LP)
1	1	1	0	0	0	1	0	342	1	1	0	1	1	1	1	1	H88-3 Serial I/O card U602 ACE (LP)
1	1	1	0	0	0	1	1	343	1	1	0	1	1	1	1	1	H88-3 Serial I/O card U602 ACE (LP)
1	1	1	0	0	1	0	0	344	1	1	0	1	1	1	1	1	H88-3 Serial I/O card U602 ACE (LP)

Address Lines									Data Lines								
LA7	LA6	LA5	LA4	LA3	LA2	LA1	LA0		I/O FLPY L	I/O CASS L	I/O LP L	I/O SERL 0 L	I/O SERT 1 L	I/O TERM L	NMI L	GPP L	
A7	A6	A5	A4	A3	A2	A1	A0	Port	D7	D6	D5	D4	D3	D2	D1	D0	Activates
1	1	1	0	0	1	0	1	345	1	1	0	1	1	1	1	1	H88-3 Serial I/O card U602 ACE (LP)
1	1	1	0	0	1	1	0	346	1	1	0	1	1	1	1	1	H88-3 Serial I/O card U602 ACE (LP)
1	1	1	0	0	1	1	1	347	1	1	0	1	1	1	1	1	H88-3 Serial I/O card U602 ACE (LP)
1	1	1	0	1	0	0	0	350	1	1	1	1	1	0	1	1	U561 Console Terminal Port ACE
1	1	1	0	1	0	0	1	351	1	1	1	1	1	0	1	1	U561 Console Terminal Port ACE
1	1	1	0	1	0	1	0	352	1	1	1	1	1	0	1	1	U561 Console Terminal Port ACE
1	1	1	0	1	0	1	1	353	1	1	1	1	1	0	1	1	U561 Console Terminal Port ACE
1	1	1	0	1	1	0	0	354	1	1	1	1	1	0	1	1	U561 Console Terminal Port ACE
1	1	1	0	1	1	0	1	355	1	1	1	1	1	0	1	1	U561 Console Terminal Port ACE
1	1	1	0	1	1	1	0	356	1	1	1	1	1	0	1	1	U561 Console Terminal Port ACE
1	1	1	0	1	1	1	1	357	1	1	1	1	1	0	1	1	U561 Console Terminal Port ACE
1	1	1	1	0	0	0	0	360	1	1	1	1	1	1	0	1	Z80 /NMI line
1	1	1	1	0	0	0	1	361	1	1	1	1	1	1	0	1	Z80 /NMI line
1	1	1	1	0	0	1	0	362	1	1	1	1	1	1	1	0	General Purpose Port
1	1	1	1	0	0	1	1	363	1	1	1	1	1	1	1	1	None
1	1	1	1	0	1	0	0	364	1	1	1	1	1	1	1	1	None
1	1	1	1	0	1	0	1	365	1	1	1	1	1	1	1	1	None
1	1	1	1	0	1	1	0	366	1	1	1	1	1	1	1	1	None
1	1	1	1	0	1	1	1	367	1	1	1	1	1	1	1	1	None
1	1	1	1	1	0	0	0	370	1	1	1	1	1	1	1	1	None
1	1	1	1	1	0	0	1	371	1	1	1	1	1	1	1	1	None
1	1	1	1	1	0	1	0	372	1	1	1	1	1	1	0	1	Z80 /NMI line
1	1	1	1	1	0	1	1	373	1	1	1	1	1	1	0	1	Z80 /NMI line
1	1	1	1	1	1	0	0	374	1	1	1	1	1	1	1	1	None
1	1	1	1	1	1	0	1	375	1	1	1	1	1	1	1	1	None
1	1	1	1	1	1	1	0	376	1	1	1	1	1	1	1	1	None
1	1	1	1	1	1	1	1	377	1	1	1	1	1	1	1	1	None