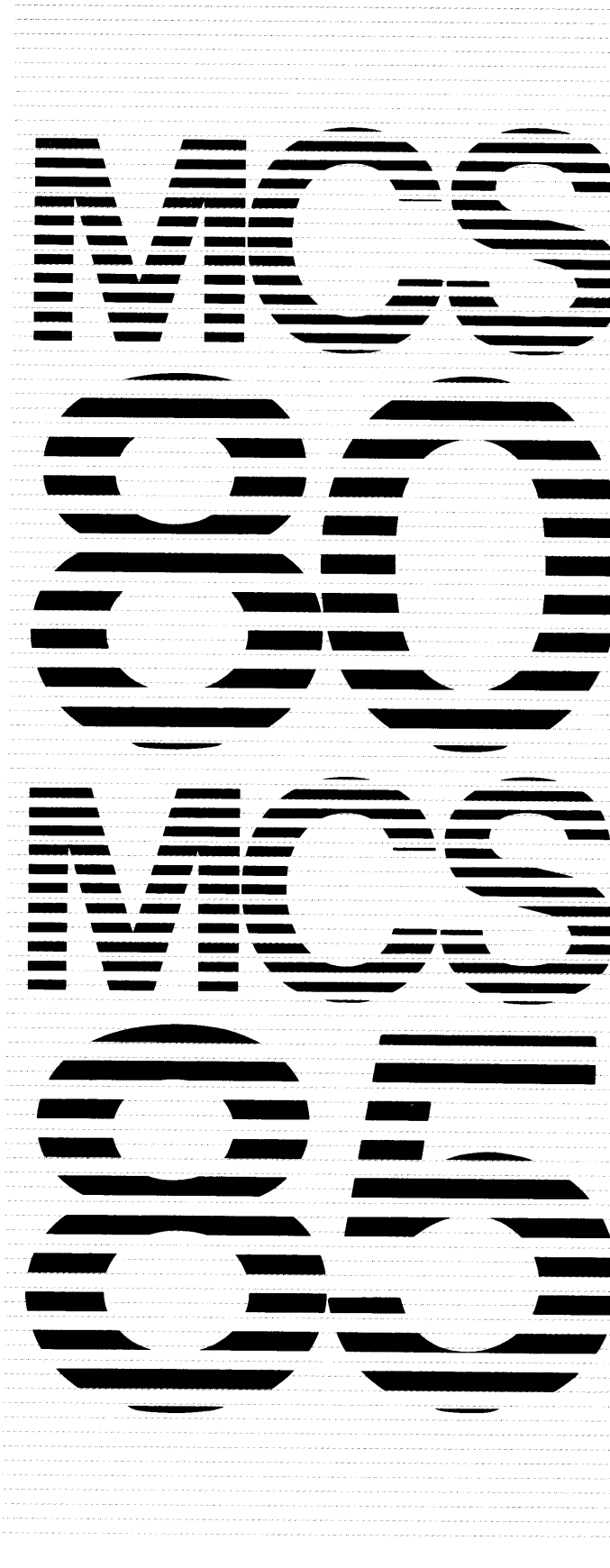
Chapter 3
8085A System Operation and Interfacing



CHAPTER 3
8085A SYSTEM OPERATION AND INTERFACING

3.1 INTERFACING TO THE 8085A

The 8085A interfaces to both memory and I/O devices by means of READ and WRITE machine cycles, the timing of which are identical. During each machine cycle the 8085A issues an address and a control signal, then either sends data out on the bus or reads data from the bus. The 8085A may be performing a READ machine cycle, but what it reads could be a ROM, RAM, I/O device, peripheral device, or nothing.

There is no distinction between data, instruction opcodes, and I/O port numbers except the way the CPU interprets what it reads from the bus. If an opcode is what would logically appear on the bus, the CPU will treat as an opcode whatever does appear there; if an I/O port number is to be expected, what appears will be interpreted as a port number. The same is true for a WRITE cycle. The 8085A issues an address, data, and a control signal. Unless it is requested to WAIT (by use of the READY line) it will complete the cycle and proceed to the next. Regardless of whether there is a device present to accept the data, the CPU executes one instruction at a time, in sequence, until told to do otherwise. The program controls the sequence and nature of all machine cycles until an interrupt occurs.

There are two ways of addressing I/O devices in the MCS-85 system. If the IO/$\overline{M}$ output from the CPU is used to distinguish between I/O and memory READ and WRITE cycles, then that system is said to employ standard, or I/O-mapped, I/O. If IO/$\overline{M}$ is not so used, the CPU does not distinguish between I/O and memory, and its system employs memory-mapped I/O. Each method of addressing I/O has advantages and disadvantages.

3.2 MEMORY-MAPPED I/O

3.2.1 Advantages of Memory-Mapped I/O

Since the processor doesn't distinguish I/O from memory using this addressing scheme, you can take advantage of the larger instruction set that references the memory address space. Instead of only being able to transfer a byte of data between the accumulator and the I/O port (using INPUT and OUTPUT instructions), you can now program arithmetic and logic operations on port data as well as move data between any internal register and the I/O port. Consider the new meaning of the following instructions:

Examples:

MOVr,M	(Input Port to any Register)
MOV M,r	(Output any Register to Port)
MVI M	(Output immediate data to Port)
LDA	(Input Port to ACC)
STA	(Output from ACC to Port)
LHLD	(16-Bit Input)
SHLD	(16-Bit Output)
ADD M	(Add Port to ACC)
ANA M	(AND Port with ACC)

3.2.2 Disadvantages of Memory-Mapped I/O

While memory instructions may increase the flexibility of the I/O system, there are some drawbacks. Since I/O devices are now addressed as memory, there are fewer addresses available for memory. A common practice is to use address bit 15 (A_{15}) to distinguish memory from I/O. (See Figure 3-2 and accompanying discussion.) If $A_{15} = 0$ then memory is being addressed; if $A_{15} = 1$, I/O is being addressed. This particular scheme limits the maximum amount of memory that can be used to 32k bytes. A further disadvantage of memory-mapped I/O is that it takes 3 bytes of instruction and 13 clock cycles using the LDA or STA instructions to specify moving a byte of data between the accumulator and an I/O device, whereas the INPUT and OUTPUT instructions require only two bytes and 10 clock cycles. This is because the I/O address space is smaller (only 256 bytes) and therefore requires fewer bits to completely specify an address. A futher advantage of using the INPUT and OUTPUT Instructions is that it allows the easy connection of the MCS-80 peripherals to the MCS-85 multiplexed bus. If you memory-map the MCS-80 peripherals to the MCS-85 bus, you must either latch the lower address bits with an 8212 or use a portion of the memory address space by connecting the chip selects and address lines of the ports to the unmultiplexed upper eight lines of the address bus.

3.3 ADDRESS ASSIGNMENT

3.3.1 Decoding

Besides memory-mapped I/O, another practice is to only partially decode the address bus when generating chip selects. Every device has a given number of unique addresses associated with it. The 8355, for instance, has 2k bytes of ROM and therefore has 2k addresses associated with the ROM. Any one of these 2k addresses can be uniquely specified by a pattern on the 11 ($2^{11} = 2k$) address lines. However, since the 8355 must work with other devices in a system, it isn't enough to simply specify the 11 bits; further bits of information must be used to locate the 2k bytes within the 65k address space. The 2k bytes are located by the use of chip enable (CE) inputs to the 8355 chip. If the 8355 were to occupy the first 2k bytes of the memory address space, it would, strictly speaking, be necessary to decode the fact that A_{15}-A_{11} were all zeroes, and use that condition as a chip enable. Then the 8355 would be selected only when the address bus was less than 2k.

However, if other 2k blocks of addresses aren't being used, you may combine those addresses and not decode all of the upper five address lines for chip enables. In fact, in a small system you may need to decode only one bit of address, which is to say connect that bit of the address bus to the chip enable line of the 8355. If you connect A_{11} to the $\overline{CE}$ line of the 8355 and tie CE to V_{CC}, then the 8355 would be selected whenever the memory address was less than 2k. (See Figure 3-1A.)

However, it will also be selected whenever memory locations 4k-6k, 8k-10k, 61k-63k (i.e., whenever bit $A_{11} = 0$) is addressed. If the programmer is aware of this, and if there are no other devices assigned to the other address spaces, then it may be an acceptable condition. Care must be taken, however, to ensure that at no time will two different devices be selected simultaneously. Whenever one device is selected, that memory address must deselect all other devices. If two devices are selected simultaneously for a READ operation, the electrical conflict on the bus may damage one or both parts. Note also that the address bus may reflect an undesired address during T_5, T_6 of an opcode fetch cycle and during address bus transitional periods in T_1 (this is illustrated in Chapter 2). Therefore, all memory and I/O devices must qualify their selection with $\overline{RD}$ or $\overline{WR}$, or the address on the bus at the falling edge of the ALE, so as to ignore all spurious addresses.

3.3.2 Linear Selection

Using an address bit as a chip select is referred to as linear selection. The direct consequence of linear selection is that you cut the available address space in half for each single address bit used as a chip enable. If this penalty is too high, you can always use an 8205 one-of-eight decoder. Also, some chips have multiple chip enables, which allows for some automatic decoding of the address. (See Figures 3-1B and 3-1C.)

One drawback to linear selection is that the memory addresses of the different parts are not contiguous. For example, if three 8355s are addressed using linear selection, one might be located at 0-2k, the next at 6k-8k, and the next at 10k-12k. The programmer must recognize these page boundries and jump over them.

3.4 INTERFACING TO THE 8155/8156, 8355/8755A

3.4.1 I/O Mapped I/O:

This section describes some of the techniques involved in connecting the MCS-85 combination memory and I/O chips to the 8085A as I/O devices.

Figure 3.1A shows one 8355 connected to the 8085A bus. (In the interest of simplicity, only the chip enable and IO/$\overline{M}$ lines are shown; the other lines are connected as shown in Figures 3.6, 3.7 or 3.8.) Notice that CE is tied to V_{CC} and $\overline{CE}$ is connected to A_{11}. This is because after RESET the processor always starts executing at location 0. Since the ROM normally contains the program, it must be selected when the address is all zeroes.

One consequence of the ROM being selected by an all-zero address is that the I/O ports on the chip will be selected only when $A_{11} = 0$. This is because the I/O ports and the memory have common chip enables, therefore forcing the selection conditions of one onto the other. Furthermore, since the IO/$\overline{M}$ line of the chip is connected to the IO/$\overline{M}$ line of the 8085A, the port has I/O mapped I/O. The I/O ports can be accessed only by use of the INPUT and OUTPUT instructions; since these are the only instructions that cause IO/$\overline{M}$ to go high.

The boxes to the right of the chip in Figure 3.1A indicate the memory addresses and I/O Port numbers required to access the chip. As a result of the linear selection technique used, there are many "don't care" bits (marked by "X"s) in the address. While they don't affect the addressing of this device, they may affect other

FIGURE 3-1A SINGLE CHIP

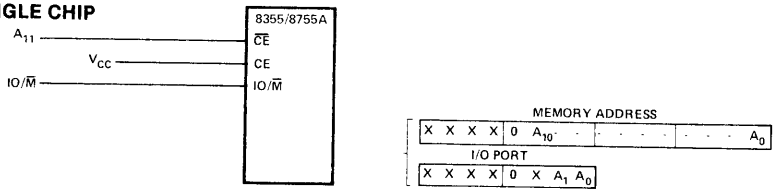


FIGURE 3-1B MULTIPLE CHIPS

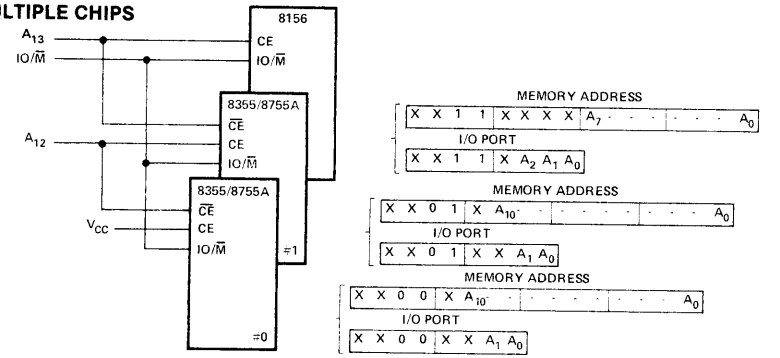


FIGURE 3-1C FULLY DECODED AND EXPANDED

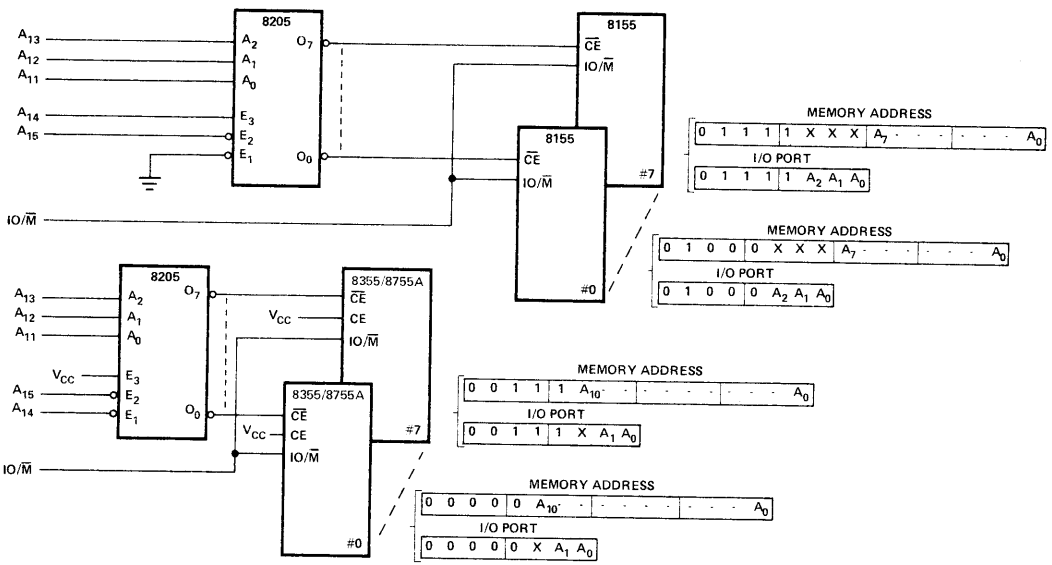


FIGURE 3-1D SEPARATE CHIP ENABLES FOR I/O AND MEMORY

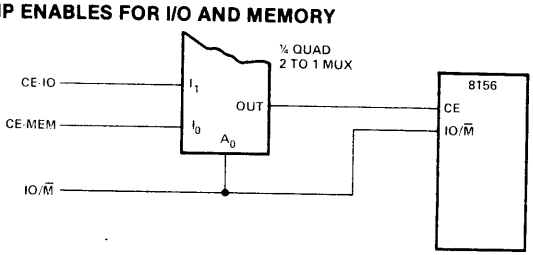


FIGURE 3-1 MCS-85™ PERIPHERALS WITH I/O MAPPED I/O

devices in the system, which would force them to be either ones or zeroes. Remember that two devices may not be selected simultaneously; thus each device must have an address that not only selects itself, but also deselects all other devices. If there are any bits which are truly "don't cares," they are customarily assigned to be zero. If all the "X" bits in Figure 3.1A were "don't cares," then the chip could be addressed as memory locations 0-2k, and I/O Ports 0-3.

Figure 3.1B shows a slightly larger system of two 8355s and one 8156. Notice that 8355 No. 1 uses its two chip enable lines to decode $A_{12} = 1$, $A_{13} = 0$. It is possible to address each of the chips without selecting any of the others. Also notice that there are some illegal addresses (e.g., $A_{12} = 0$, $A_{13} = 1$) that would cause two of the devices to turn on simultaneously. The programmer must not use these addresses.

Figure 3.1C shows a larger MCS-85 system. Two 8205s are used to completely decode the addresses. There are some interesting points to observe here. First, while some of the devices have multiple possible address (i.e., they have some "don't care" bits), there aren't any addresses which can cause simultaneous selection of two or more parts. Second, the I/O and memory portions of the 8x55 components share chip enables, so they are forced to live with each other's constraints. Third, only one 8205 is required per eight chips for the decoding; that's an overhead of only 1/8 of a chip per part.

Figure 3.1D shows a remedy to the problem illustrated in Figure 3.1C, namely that I/O and memory portions of the chip are forced to live with each other's chip enable constraints. By using a quad 2 to 1 multiplexer, the chip enables of the I/O and memory portions of four chips can be independently assigned.

3.4.2 Memory-Mapped I/O:

Figure 3.2A shows an 8355 connected to the 8085A. Since the IO/$\overline{M}$ pin of the 8355 is connected to A_{15}, whenever $A_{15} = 1$ the I/O ports will be accessed. While A_{15} could be set to 1 either by a memory or by an I/O instruction, in this situation the port is usually accessed only by the memory instructions. You may access ports either as memory locations (where $A_{15} = 1$ refers to a memory address of 32k or higher) or as I/O ports (where $A_{15} = 1$ refers to an I/O address of 128 or higher, since bits A_8-A_{15} are a

FIGURE 3-2A SINGLE CHIP

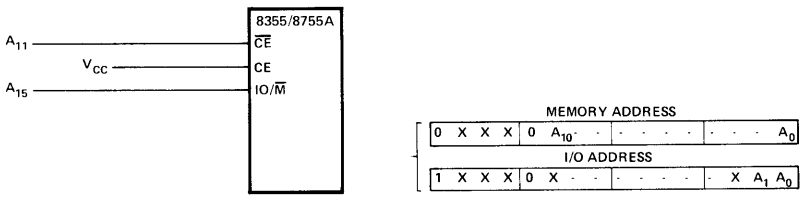


FIGURE 3-2B MULTIPLE CHIPS

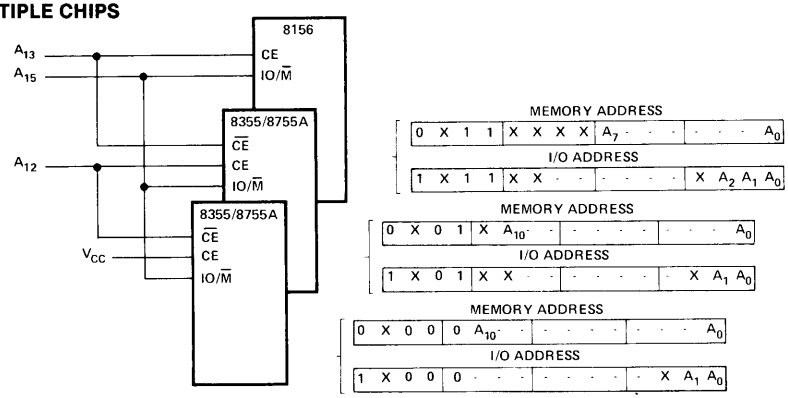


FIGURE 3-2 MCS-85™ PERIPHERALS WITH MEMORY-MAPPED I/O

replication of bits A_0-A_7). Assuming that memory-mapped I/O is used, the addresses are shown in the boxes to the right in Figure 3-2. If you want to be sure that neither the I/O nor the memory is ever selected by any INPUT or OUTPUT instruction, then the chip enable must be conditioned by IO/$\overline{M}$ = 0.

Figure 3.2B shows a somewhat larger system, also using memory-mapped I/O. As in Figure 3.1B care must be exercised to ensure that no two devices are accessed simultaneously. You can see that considerable memory address space is used up as a result of using memory-mapped I/O.

3.5 INTERFACING TO MCS-80™ PERIPHERALS

3.5.1 I/O Mapped I/O:

For want of a better name, the Intel® 825x, 827x, and 829x series peripherals are referred to here as MCS-80 peripherals because unlike the 8155/56, 8355 and 8755A, they are compatible with the nonmultiplexed MCS-80 system bus.

To interface to an MCS-80 peripheral, you must provide a constant address, a chip select, and $\overline{RD}$ or $\overline{WR}$. Since the upper address lines (A_8-A_{15}) of the 8085A are nonmultiplexed, they can be tied directly to the peripherals, as shown in Figure 3.3A. To provide I/O mapped I/O, use either linear selection (keeping the I/O and memory addresses noncoincidental), or condition the chip selects $\overline{WR}$ with IO/$\overline{M}$ = 1. Figure 3.3A shows a technique of gating the chip selects with IO/$\overline{M}$ = 1, using an 8205. This technique also allows more I/O devices to be used than linear selection would. Note that this technique relies on the fact that the I/O Port number is copied onto A_8-A_{15} as well as A_0-A_7 during an INPUT or OUTPUT instruction.

Figure 3.3B shows an alternative approach to interfacing to MCS-80 components. By latching the lower 8 bits of address with an 8212, and decoding the control signals with an 8205, you create an exact copy of the MCS-80 (8080A, 8224, 8228) bus. You may then use whatever circuits have been previously developed for the 8080. The total cost is one 8212 and one 8205. Since the same signals might have needed buffering anyway (and the 8212 and 8205 provide buffering of their outputs), the extra component overhead ranges from little to nothing.

3.5.2 Memory-Mapped I/O:

Exactly the same techniques used to memory map the MCS-85 apply to the MCS-80 I/O devices. Figure 3.4 shows an 8205 used to qualify the chip select of the I/O device with IO/$\overline{M}$ = 0. Since the MCS-80 peripherals require nonmultiplexed address lines, linear select is not too useful unless the address lines are latched. This is because connecting both the chip selects and the address lines of the MCS-80 peripherals to A_8-A_{15} would deplete all the useful addresses very quickly.

3.6 INTERFACING TO STANDARD BUS MEMORIES

Standard bus memory devices are designed to be used with nonmultiplexed address and data buses. Interfacing to standard memories is very similar to interfacing to MCS-85 memories with the exception that A_0-A_7 must be latched. Once this requirement is met, all the tricks discussed earlier can be used. Since the address lines would eventually require buffering as the system size grew, the overhead of the 8212 latch again becomes negligible.

Figure 3.5 shows the interface of the 8085A to a large block of memory, specifically 16k bytes of ROM and 8k bytes of RAM. Besides the memories, the circuit requires only 2-1/6 other parts for logical gating. If MCS-80 I/O parts were used, the 8212 latch could be shared between the two groups, further reducing the gating overhead per IC. Sixteen 2142 chips and eight 2316E chips are used in this design. The data bus, address lines 8-10, and control signals in this system all should be buffered. This applies to any system with the number of memory devices represented here.

Wherever two or more parts are paralleled on the same bus, they must be 3-state devices such as the 2142 RAM, 2316E ROM, 2716 EPROM, 2332 ROM, 2732 EPROM, and 2364 ROM, which have either an output disable (OD) input or multiple chip select (CS) inputs. To prevent bus contention, only one memory device may be output-enabled at a time in this configuration; the outputs of all others must be deselected during $\overline{RD}$.

For additional information on interfacing standard memory devices, please read Section 2 of Appendix I and the Intel applications note AP-30 "Application of Intel's 5V EPROM and ROM Family for Microprocessor Systems" available from: Intel, Literature Dept., 3065 Bowers Ave., Santa Clara, CA 95051.

3.7 DYNAMIC RAM INTERFACE:

For interfacing the dynamic RAM, Intel makes a single-component dynamic RAM refresh controller, the 8202, which interfaces the 8085A to multiplexed-address-bus dynamic RAMs like

FIGURE 3-3A DECODED CHIP SELECTS

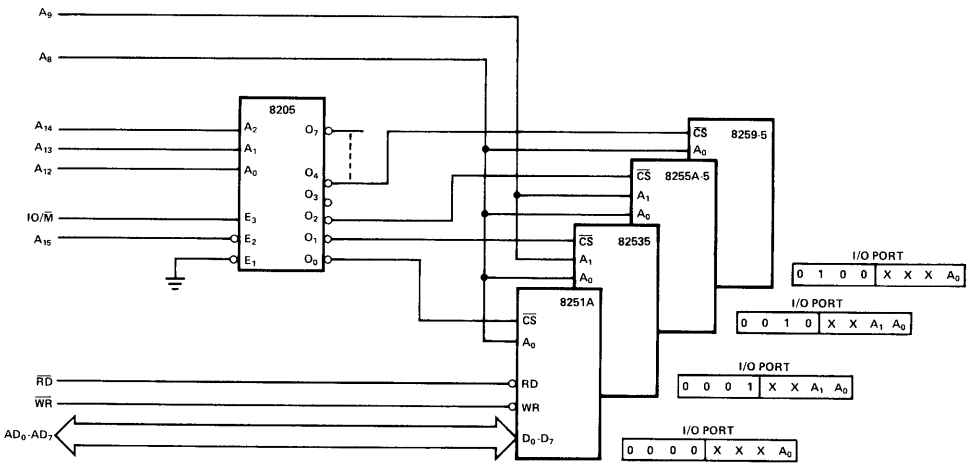


FIGURE 3-3B DECODED CONTROLS AND LATCHED ADDRESS (MCS-80™ TYPE BUS)

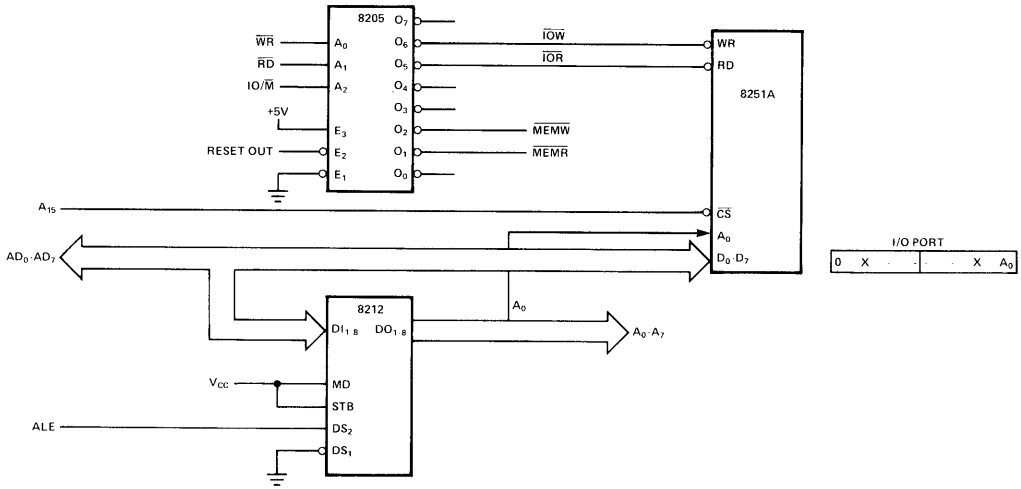


FIGURE 3-3 MCS-80™ PERIPHERALS WITH I/O MAPPED I/O

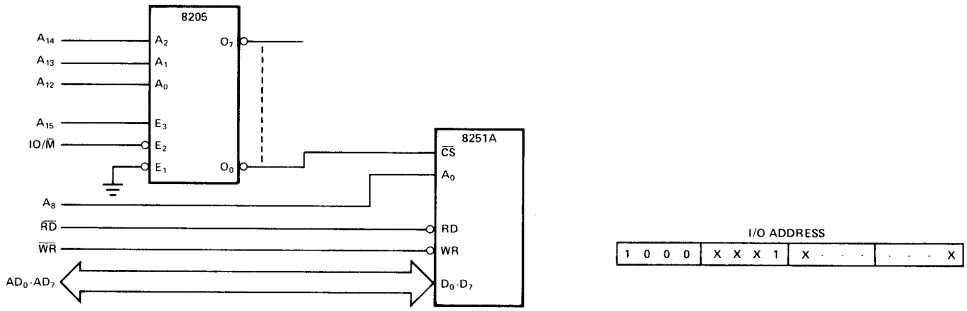


FIGURE 3-4 MCS-80™ PERIPHERALS WITH MEMORY-MAPPED I/O AND DECODED CHIP SELECTS

the Intel 2104A and 2117. The 8202 provides the necessary refreshing for such dynamic RAMs, and also provides the control signals required for accessing, selecting, and address clocking. It allows for the use of the 8085A's full capability of 64k bytes of address space with no additional buffering devices. As with other standard memory interfaces, it is necessary to demultiplex the lower 8 bits of address from the multiplexed 8085A bus, AD_{0-7}.

3.8 MINIMUM MCS-85™ SYSTEM

The Schematics of Figure 3.6 depict a minimum system core. In actual use, some of the processor control signals (TRAP, INTR, and HOLD) would have to be terminated. Also, interface logic to external devices as well as more memory and I/O devices may be desirable. The first thing one notices about the system in Figure 3.6 is the scarcity of parts required to build this system. With a minimum of parts, we have constructed a microcomputer system that has the following functions:

PARTS	FUNCTIONS
1 8085A	1 CPU (Clock cycle ≤320 ns)
1 8355/8755A	2048 Bytes of either EPROM or ROM
1 8156	256 Bytes of RAM
1 Crystal	38 I/O Lines
4 Resistors	5 Interrupts
1 Capacitor	1 Programmable Timer/ Counter
1 Diode	1 Crystal and Oscillator
1 +5 Power Supply	1 Clock
	1 Power-on Reset

By looking at the printed circuit layout of Figure 3.7, we can see that not only are there just 3 ICs, but that the interconnection of these parts is extremely easy and provides a very dense layout. Expecially notice the easy flow of the system bus on the solder side of the board.

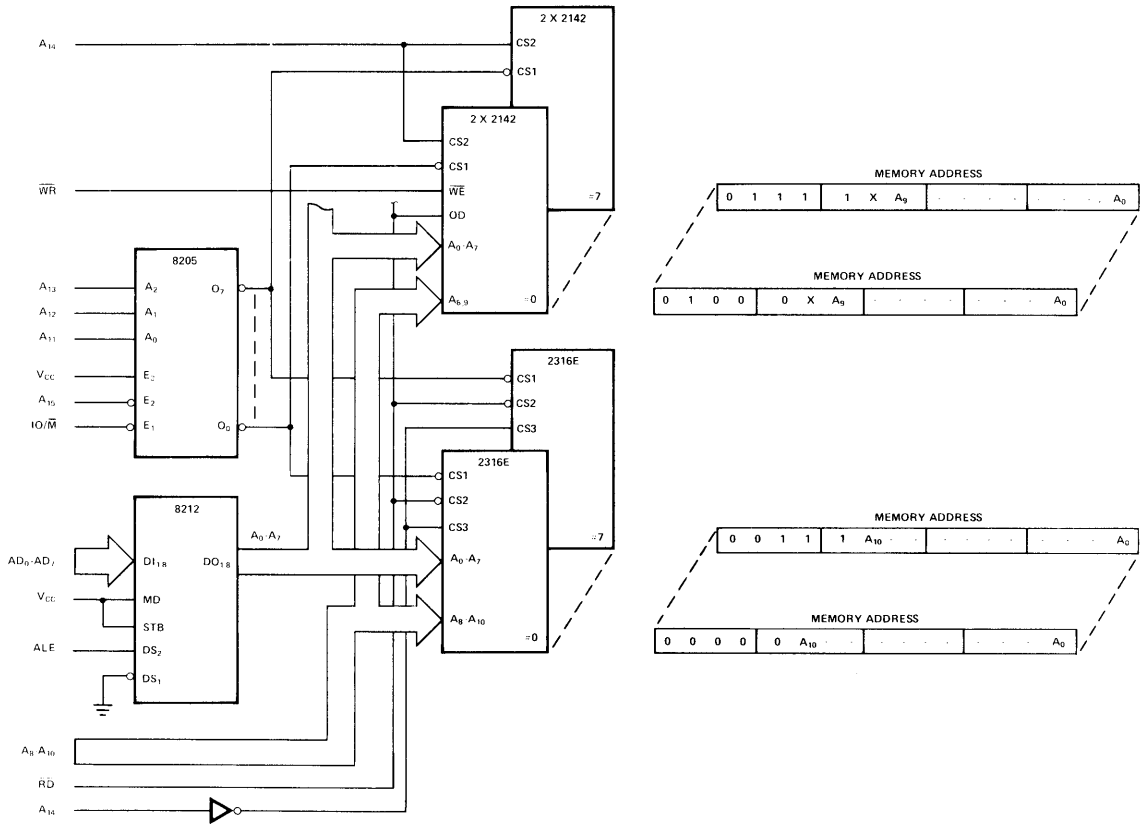
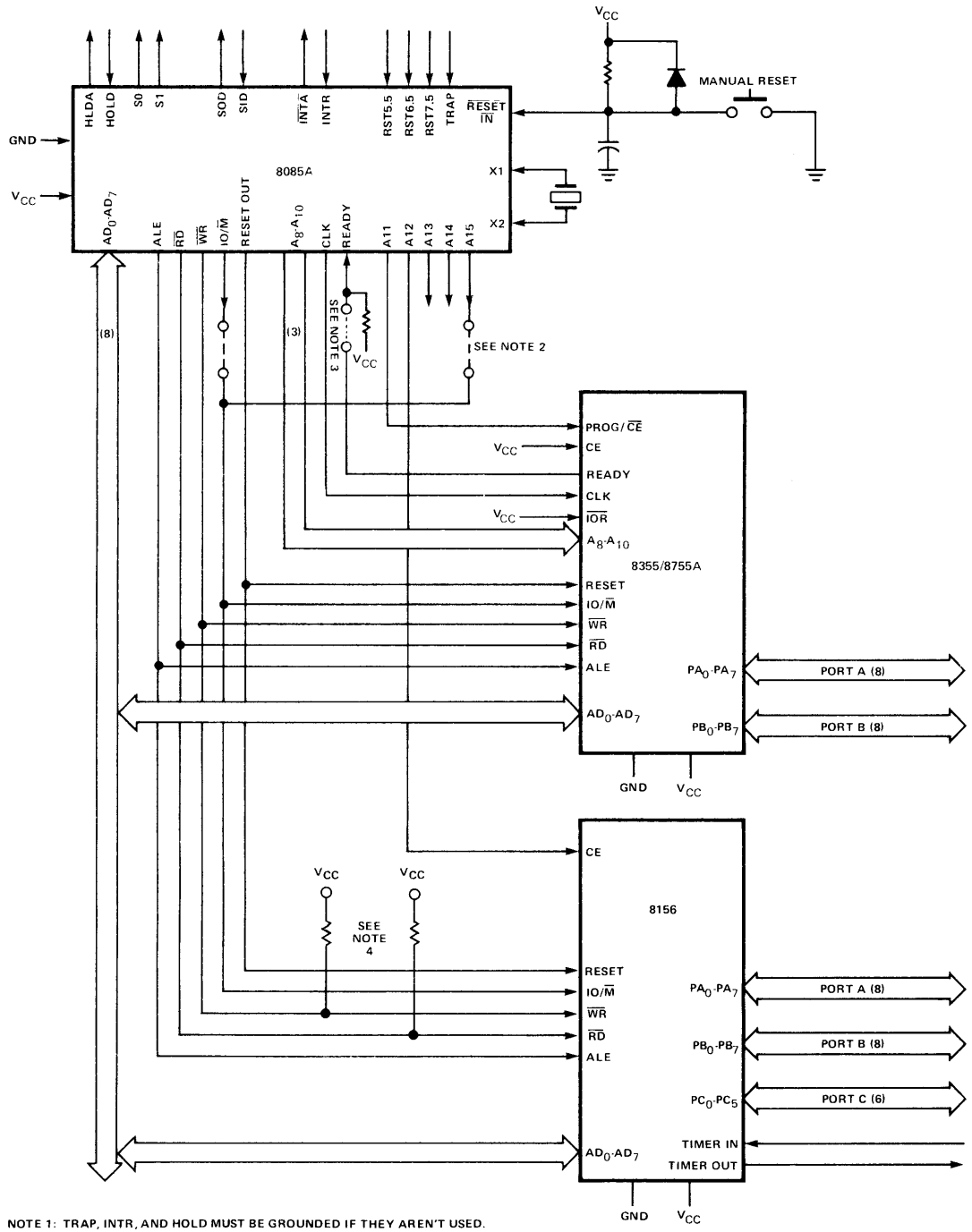
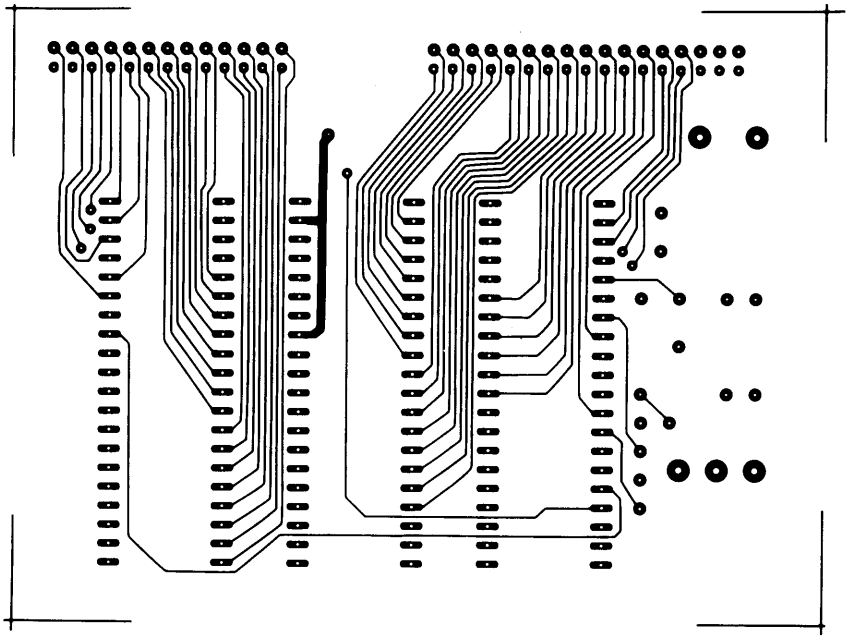


FIGURE 3-5 STANDARD MEMORIES WITH LATCHED ADDRESS AND DECODED CHIP SELECTS

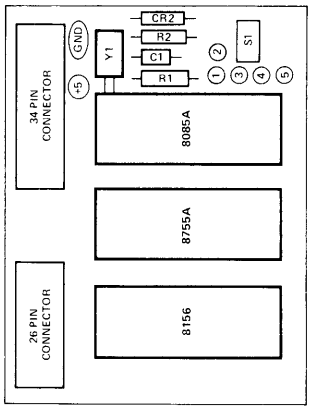


NOTE 1: TRAP, INTR, AND HOLD MUST BE GROUNDED IF THEY AREN'T USED.
NOTE 2: USE IO/M FOR STANDARD I/O MAPPING. USE A15 FOR MEMORY MAPPED I/O.
NOTE 3. CONNECTION IS NECESSARY ONLY IF ONE T_{WAIT} STATE IS DESIRED.
NOTE 4: PULL-UP RESISTORS RECOMMENDED TO AVOID SPURIOUS SELECTION WHEN $\overline{RD}$ AND $\overline{WR}$ ARE 3-STATED. THESE RESISTORS ARE NOT INCLUDED ON THE PC BOARD LAYOUT OF FIGURE 3-7.

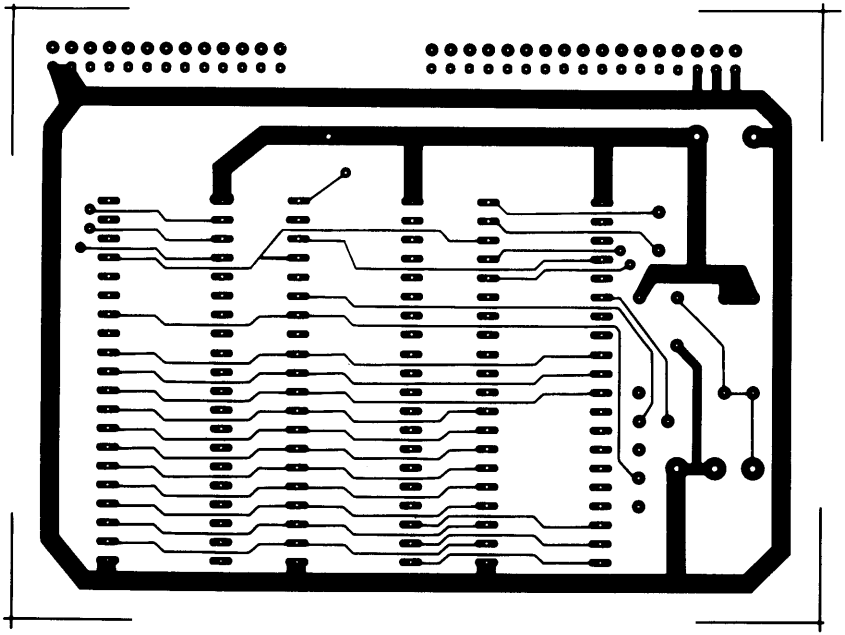
FIGURE 3-6 MINIMUM 8085 SYSTEM



COMPONENT SIDE
SCALE: ≈1:1

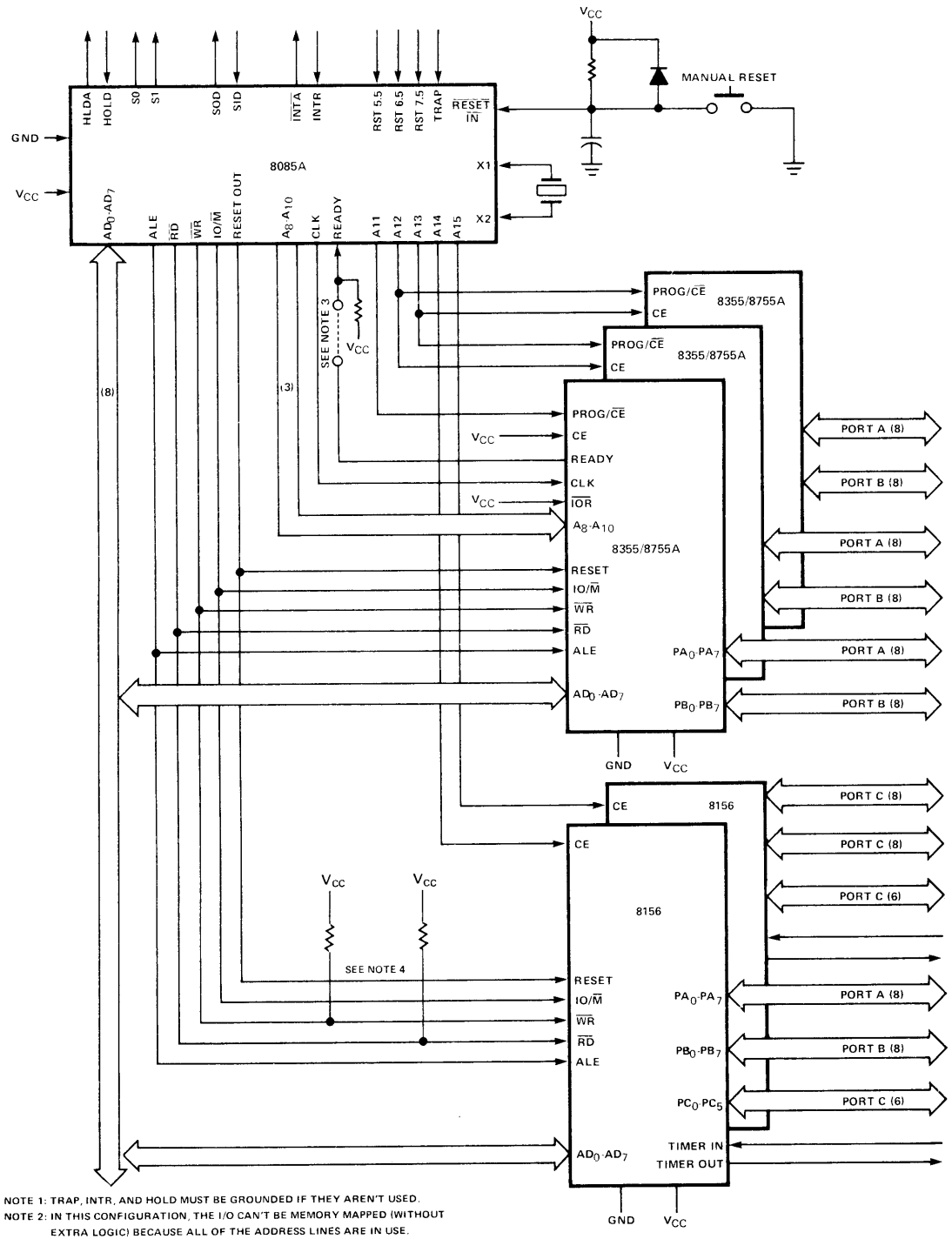


COMPONENT LAYOUT



SOLDER SIDE
SCALE: ≈1:1

FIGURE 3-7 PRINTED CIRCUIT LAYOUT



NOTE 1: TRAP, INTR, AND HOLD MUST BE GROUNDED IF THEY AREN'T USED.
NOTE 2: IN THIS CONFIGURATION, THE I/O CAN'T BE MEMORY MAPPED (WITHOUT EXTRA LOGIC) BECAUSE ALL OF THE ADDRESS LINES ARE IN USE.
NOTE 3: CONNECTION IS NECESSARY ONLY IF ONE T_{WAIT} STATE IS DESIRED.
NOTE 4: PULL-UP RESISTORS RECOMMENDED TO AVOID SPURIOUS SELECTION WHEN $\overline{RD}$ AND $\overline{WR}$ ARE 3-STATED.

FIGURE 3-8 EXPANDED SYSTEM

3.9 EXPANDED MCS-85™ SYSTEM

Figure 3.8 shows the circuit Figure 3.6 expanded to its maximum size without the use of any extra logic. In an extremely small board area we can fit:

PARTS	FUNCTION
1 8085A	1 CPU (Clock cycle
3 8355/8755A	≤ 320ns)
2 8156	ROM/EPROM
1 Crystal	6144 Bytes
4 Resistors	512 Bytes RAM
1 Capacitor	76 I/O Lines
1 Diode	5 Interrupts
	2 Programmable Timer/Counters
	2 Serial I/O Lines
	1 Crystal and Oscillator
	1 Clock
	1 Power-on Reset

3.10 MCS-85™ SYSTEM WITH 8185

The 8185 1K-byte static RAM chip is another multiplexed-bus component that insures that the most highly integrated systems can be built with MCS-85 components. Figure 3.9 shows a 4-chip MCS-85 system schematic with the following characteristics:

PARTS	FUNCTION
1 8085A	1 CPU
1 8185	ROM/EPROM
1 8156	2048 Bytes
1 8355/8755A	1280 Bytes RAM
	38 I/O Lines
	5 Interrupts
	1 Timer/Counter
	2 Serial I/O Lines

The 8185 also has power-down capability. By connecting $\overline{CE}_1$ to IO/$\overline{M}$ from the 8085A the 8185 will be powered down during I/O operations and Interrupt Acknowledge cycles.

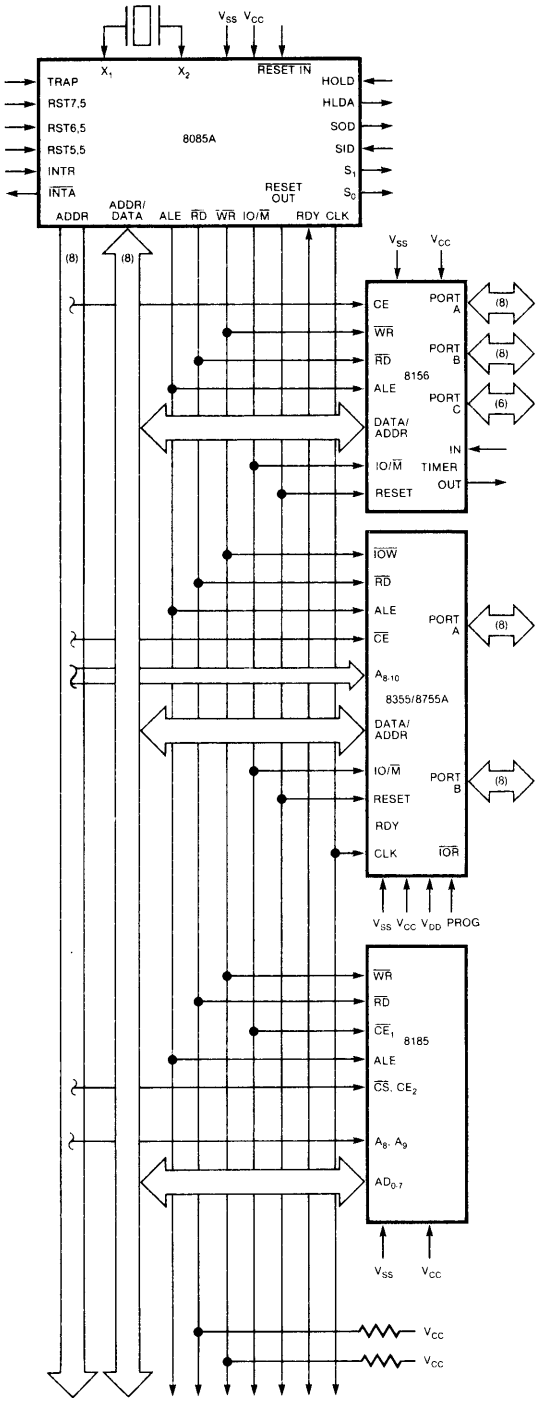


FIGURE 3-9 MCS-85 SYSTEM WITH 8185