Experiment 3

Designing the Memory System

3.1 Background

The 8086 CPU has addressing capability of 1 Mega Bytes as well as a 16-bit data bus. Basically, EPROM and SRAM memory chips are byte organized. The 8086 CPU requires that the memory be organized as two banks called the even and odd banks. The EPROM memory is to store resident programs that must run when the microprocessor system is powered on. The SRAM memory is used to store application programs that are ready to run as well as to provide some reserved locations for the proper operations of interrupts. The designer must be careful with bank selection especially when dealing with read/write memories. In this case, byte operations with one bank must be done while enabling one of the memory banks and disabling the other. This experiment should expose the student to a number of issues such as:

1. Partitioning of the memory address space,
2. Distinguishing between addressing capability and physical memory,
3. Dealing with partial and exhaustive addressing, arid
4. Studying address decoding techniques

3.2 Objective

Designing and implementing a small memory system using SRAM and EPROM memory chips

3.3 Equipment

- The prototype-board that already includes an 8086 CPU operating in minimum mode with clock generator and a fully demultiplexed data and address buses,

- Two 8 Kbytes SRAM memories (6264), and
- Two 8 Kbytes EPROM memories (2764)

3.4 Procedure

In this project, each group will design a memory system consisting of two memory modules. The first is a 16 KByte SRAM starting at address 00000h. The second is a 16 KByte EPROM ending at address FFFFh (Why?). Refer to your text book for more details about memory system design.

1. Each group discuss and answer the following issues:
    a. What is the number of address lines required to address the two modules?
    b. How to design the 32 KBytes memory system using partial decoding?
    c. How to design the same memory system using exhaustive (full) decoding?
    d. What address lines to be connected to the even and odd banks of the two modules?
    e. How to distinguish the even and odd banks of the SRAM module?
    f. Is it necessary to distinguish the even and odd banks of the EPROM module? Why?
2. Design the above memory system by using two 8 KBytes SRAM memories (6264) and two 8 KBytes EPROM memories (2764). Show your design to the lab instructor and implement it after validation. Your design should show, for each memory chip, what should be connected to:
    a. Address and data pins
    b. Chip select ($\overline{CS}$)
    c. Output Enable ($\overline{OE}$)
    d. Write Enable ($\overline{WE}$)

3. Implement the decoder shown in Figure 3.1 which takes as input $\overline{RD}$, $\overline{WR}$ and $M/\overline{IO}$, and produces *memory read* ($\overline{MEMR}$), *memory write* ($\overline{MEMW}$), *I/O read* ($\overline{IOR}$) and *I/O write* ($\overline{IOW}$) signals.

4. After completing the wiring, each group must carry out visual and electrical testing of the connections (e.g. using multimeters) as well as doing necessary corrections.

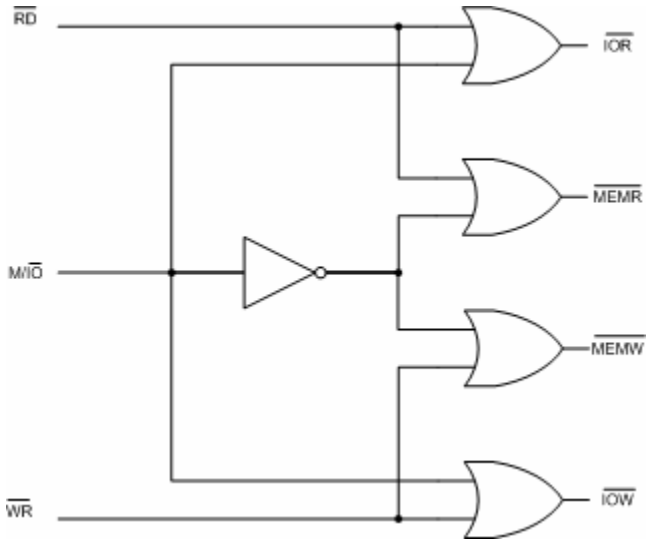
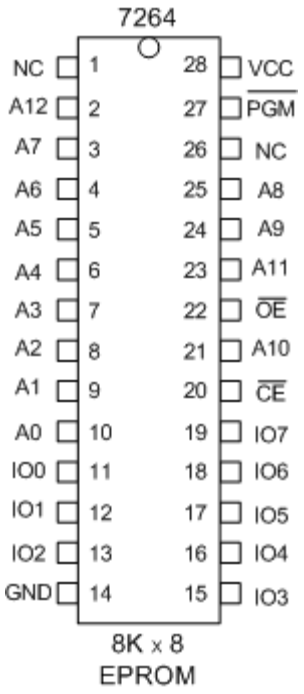
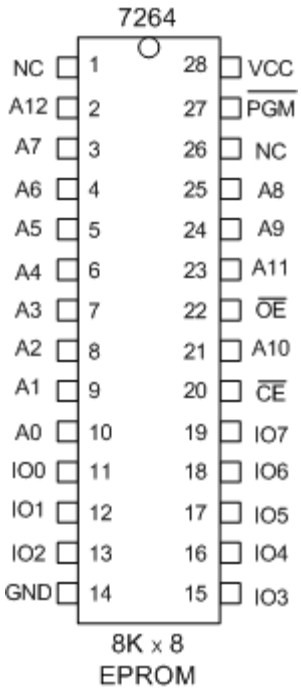
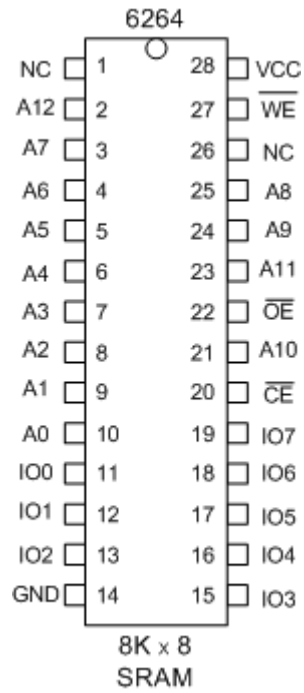
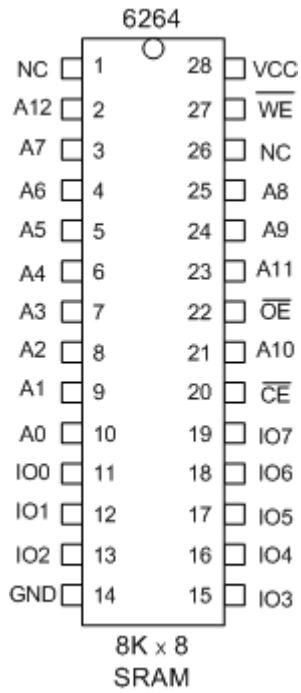


Figure 3.1: Generating the four memory and I/O control bus signals from the 8086's RD, WR and I/O signals.


7264
NC 1
A12 2
A7 3
A6 4
A5 5
A4 6
A3 7
A2 8
A1 9
A0 10
IO0 11
IO1 12
IO2 13
GND 14
28 VCC
27 PGM
26 NC
25 A8
24 A9
23 A11
22 OE
21 A10
20 CE
19 IO7
18 IO6
17 IO5
16 IO4
15 IO3
8K x 8
EPROM


7264
NC 1
A12 2
A7 3
A6 4
A5 5
A4 6
A3 7
A2 8
A1 9
A0 10
IO0 11
IO1 12
IO2 13
GND 14
28 VCC
27 PGM
26 NC
25 A8
24 A9
23 A11
22 OE
21 A10
20 CE
19 IO7
18 IO6
17 IO5
16 IO4
15 IO3
8K x 8
EPROM


6264
NC 1
A12 2
A7 3
A6 4
A5 5
A4 6
A3 7
A2 8
A1 9
A0 10
IO0 11
IO1 12
IO2 13
GND 14
28 VCC
27 WE
26 NC
25 A8
24 A9
23 A11
22 OE
21 A10
20 CE
19 IO7
18 IO6
17 IO5
16 IO4
15 IO3
8K x 8
SRAM


6264
NC 1
A12 2
A7 3
A6 4
A5 5
A4 6
A3 7
A2 8
A1 9
A0 10
IO0 11
IO1 12
IO2 13
GND 14
28 VCC
27 WE
26 NC
25 A8
24 A9
23 A11
22 OE
21 A10
20 CE
19 IO7
18 IO6
17 IO5
16 IO4
15 IO3
8K x 8
SRAM