Hardware details of 8088

- The 8088 is a 16-bit microprocessor
  - 8-bit bidirectional data bus. (requires two read cycles to capture 16 bits of data.)
  - 20-bit address bus can access over 1 million bytes of memory \((2^{20} = 1 \text{ MB})\)
  - Control signals are provided that enable external circuitry to take over the 8088's buses
  - two interrupt lines are included to provide maskable and nonmaskable interrupt capability.
Operation Modes

• The 8088 can operate in either one of two modes of operation:
  – (1) Minimum Mode
  – (2) Maximum Mode

• Some pins have dual functions, one for minmode and the other for maxmode.

• Minimum Mode:
  – The simplest and least expensive mode.
  – All the control signals for memory & I/O operations are generated by the processor.

• Maximum Mode:
  – Some of the control signals must be externally generated (requires an external bus controller 8288)
  – Allows the system to use an external coprocessor such as 8087 (floating-point coprocessor).

Pin layout of 8088
8088 Pin Functions

- $V_{CC}$, GND and CLK
  - there are two pins for ground (GND)
  - one for Vcc
    - 8088 operates on a single, positive supply of 5 V ±10%
    - supply current is 340 mA at room temperature.
  - The CLK input requires a digital waveform with a 33% duty cycle.
  - The minimum clock period is 200 ns, corresponding to a frequency of 5 MHz.

8088 Pin Functions

- AD0 – AD7, A8 – A19
  - These signals constitute the 8088's 20-bit address bus and 8-bit data bus.
  - AD0 - AD7 are the processor's multiplexed address/data bus.
8088 Pin Functions

- ALE (address latch enable)
  - is an output signal used to demultiplex the 8088’s address/data bus.
  - The processor uses ALE to indicate when AD0 through AD7 contain address information

![Diagram of ALE and address/data bus]

**FIGURE 10.5** Demultiplexing the 8088’s address/data bus
8088 Pin Functions

• DT/R, WR, RD, and IO/M
  – DT/R (data transmit/receive) is an output used in a minmode system to control the direction of data flow in the bidirectional buffer used on the data bus.
    • When DT/R is low data should flow into the 8088.
    • When DT/R is high, the 8088 is outputting data.
  – WR (write) is an active low output used to indicate when the processor is writing to a memory or I/O
  – RD (read) is an active low output used to indicate when the processor is reading a memory or I/O.
  – IO/M is an output that indicates whether the current bus cycle is a memory access or an I/O access.

• Figure shows how OR gates can be used to decode different read/write operations
8088 Pin Functions

• NMI, INTR, and INTA
  – These three signals control the activity of external hardware interrupts.
  – NMI and INTR are inputs and function identically in either processor mode.
  – INTA is an output available only in minmode.

MEMORY SYSTEM DESIGN
MEMORY SYSTEM DESIGN

• Microprocessor-based memory system, whether EPROM or RAM, have standard buses connecting them to the microprocessors. These are
  – the control bus,
  – the data bus,
  – the address bus.

Bus Buffering

• A microprocessor address or data buses may be overloaded when they are connected to external memory.
  – For this reason, we will buffer the address and data buses.
Accessing Memory

• Some output signals are used to drive a memory unit, these are
  – RD
  – WR
  – IO/M
Memory read cycle timing

![Memory read cycle diagram]

Memory write cycle timing

![Memory write cycle diagram]
DESIGNING A MEMORY ADDRESS DECODER

- The main function of a memory address decoder is to monitor the state of the address bus and determine when the memory chips should be enabled. (i.e. RAMs or ROMs)
Examples

Example 11.1: A circuit containing 32KB of RAM is to be interfaced to an 8088-based system, so that the first address of the RAM (also called the base address) is at 48000H. What is the entire range of RAM addresses? How is the address bus used to enable the RAMs? What address lines should be used?

Example 11.2: A 16KB EPROM section, with a starting address of 30000, is to be added to an existing memory system. The following circuitry will properly decode the entire address range, 30000 to 33FFF. The 8-input NAND gate in Figure 11.11 is used to detect the 3 pattern on the upper four address bits, and the 0s on A13 and A14. Six address lines are used in this decoder, because the other 14 are needed to address one of 16K possible byte locations in the EPROM. The EPROM is directly addressed by A0 through A12.
Example 11.3: Two 32KB memories, an EPROM with a starting address of 60000 and a RAM with a starting address of 70000, are needed for a new minicomputer memory system. Figure 11.12(a) shows how the EPROM is enabled, and Figure 11.12(b) shows how the RAM is enabled. In this design, it is only necessary to use a 4-input NAND gate to do the decoding of the "6" or "7" part of the address range.

Experienced digital designers can detect binary patterns, and the reward in finding a pattern is generally a reduction in the digital circuitry needed to implement a desired function. Did you notice that the address ranges for the RAM and EPROM in the previous example are very similar? In fact, they are identical, except for the $A_{16}$ address bit. Let us look at another example to see how we can use pattern detection to simplify the required hardware.

Example 11.4: The RAM and EPROM sections of Example 11.3 are enabled by the simplified decoder presented in Figure 11.13. Do you see how the NAND gate is used to detect
Partial Address Decoding

- Some applications may require small memories.
- e.g. a system consisting 8K words of EPROM and 8K words of RAM needs only 14 address lines.
  - The first 13 (A0 – A12) go directly to the EPROM and RAM,
  - and the last address line A13 is used to select either the EPROM or the RAM.

Example 11.6: A 16KB block of memory, composed of two 8KB EPROMs, is to have a starting address of 4000H. What is the address range for each EPROM? What circuitry is needed to implement a partial address decoder for a minmode system?
Example 11.7: A 32KB EPROM needs a starting address of 30000, and a 32KB RAM needs a starting address of 20000. The circuitry in Figure 11.17 shows how these addresses are partially decoded. In this example, three-input NAND gates are used to do the decoding. All three inputs must be high for the output to go low (and enable the memories). IO/M is inverted, so that it presents a 1 when low. \( A_{17} \) is connected directly, because it is high in both the RAM and EPROM address ranges. Only \( A_{18} \) changes. It is low for the RAM range and high for the EPROM range. Address lines \( A_0 \) through \( A_{14} \) are used by the memories themselves.

**FIGURE 11.17 Partial address decoder for 32KB EPROM at 30000, and 32KB RAM at 20000**

---

A complete RAM/EPROM memory

- Required hardware for
  - 8KB of EPROM (located at FE000) and
  - 8KB of RAM (located at 00000).