Objectives

You will be able to:

- State the function of the pins of the 8088
- List the functions of the 8088 data, address, and control buses
- State the differences in the 8088 microprocessor in maximum mode versus minimum mode
- State the function of the pins of the 80286
- Describe the differences between real and protected modes
- Describe the operation of the 80286 data, address, and control buses
- Describe the purpose of the expansion slots of the IBM PC AT (ISA)
- Describe the ISA bus system

8086 8088 Processor

- Introduced in 1978.
- Internally and Externally 16-bit processor.
- AD0-AD7 are address-data lines.
  - ALE high means AD0-AD7 carry address.
  - ALE low means AD0-AD7 lines carry data
- 88 in minimal mode reduces the memory interface hardware.

8088 Address and Data Bus

- **74LS373** 8-bit (octal) D latch latches the address bits A0-A7 when ALE is HIGH, and keeps them available when ALE is LOW.
74LS373
Octal D-Latch with Enable

- It contains 8 D-latches.
- A D-latch transfers D-input to Q-output while G is high,
- It latches Q-status while G is low.
- ~OC low turns the outputs high impedance (not connected).

<table>
<thead>
<tr>
<th>Function Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output Control</td>
</tr>
<tr>
<td>L</td>
</tr>
<tr>
<td>L</td>
</tr>
<tr>
<td>L</td>
</tr>
</tbody>
</table>

8086 - 8088 Address Space

- 8088 has 20 address lines: A19 … A0 to address an 8-bit data in the memory.
  - Address lines A7 – A0 are multiplexed with Data lines D7 – D0.
  - Its address space is $2^{20} \times 8$-bit = 1 MByte.
- For a stable operation, all addresses must be latched to drive memory devices properly.

Typical Memory Control Lines

- ~CS (~chip select = ~CE : ~chip enable)
- ~WE (Write Enable)
- ~OE (Output Enable)

Memory Write and Read Control

- ~CS or ~CE must be active (=low) to read or write a memory location.
- ~WE must be active (=low) to write a data to addressed location
- ~OE must be active (=low) to read a data from the addressed location.
Memory Cycles

READ CYCLE
- Processor sends
  - Address to read
    - Address decoder generates CS. Memory is activated
  - ~OE low, starts reading memory

WRITE CYCLE
- Processor sends
  - Address to write into
    - Address decoder generates CS. Memory is activated
  - Data to write into memory
  - ~WE low, starts writing data to memory

Memory
- decodes address
- selects the register,
- connects data lines to the selected register

Processor
- transfers the memory contents to a processor register.
- makes ~OE high to end the reading process.

8088 8-bit I/O instructions

- 8086 and 8088 have four isolated IO instructions.
  - in al, <p8> ; p8 is 8-bit port number
  - mov dx, <p16>; p16 is 16-bit port number
  - in al, dx
  - out <p8>, al
  - mov dx, <p16>
  - out dx, al

- In accessing to the port address.
  - in instructions make 8086 to assert an ~IOR active,
  - out instruction make 8086 to assert an ~IOW active

8088 Simple Output Port

- 74LS373 latches output data on D0-7 when port is addressed and also ~IOW is low
  - Design for out 99h, al

Execution of “mov al,12h” “out 99h,al” at CS:IP =01A7:0018

- out 99h,al is coded by two bytes: E6h 99h.

<table>
<thead>
<tr>
<th>cycle</th>
<th>A[19..8]</th>
<th>AD[0..7]</th>
<th>~WR</th>
<th>~RD</th>
<th>~IO/~M</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALE</td>
<td>01Ah</td>
<td>88h</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>mem.read</td>
<td>01Ah</td>
<td>E6h</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ALE</td>
<td>01Ah</td>
<td>89h</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>mem.read</td>
<td>01Ah</td>
<td>99h</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ALE</td>
<td>000h</td>
<td>99h</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>io.write</td>
<td>000h</td>
<td>12h</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>ALE</td>
<td>01Ah</td>
<td>8Ah</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

- Processor
  - fetches E6h from the memory.
  - fetches the port address byte, 99h
  - Writes contents of AL (=12h) to port 99h.
  - When OUT instruction is over processor starts to fetch the next instruction.

- The processor puts the contents of AL on Data-bus. Port shall be enabled by decoder since address is 99h, and shall latch 12h since bus controller generates ~IOW
After “out 99h,al” Instruction

- Processor starts fetching and executing the next instruction.
- But the latched value 12h stays at port-99h.
- It makes LED2 to light, but LED1 to stay dark.

The outputs \([\sim \text{WR}=0, \sim \text{RD}=1 \text{ and IO}/\sim \text{M}=1]\) from the processor makes \(G=1\). LS7373 latches \(Q[0..7]=12h\) and keeps it latched at the port outputs until the next out 99h,al instruction.

8088 Simple Input Port

- **74LS244** buffers input data to D0-7 when port is addressed and also \(\sim \text{IOR}\) is low.
- Design for \(\text{in al,9Fh}\)

The processor reads the port states. The port buffer is normally disabled. Address decoder enables it since the address is 9Fh, and \(\sim \text{IOR}\) is active.

74LS244 Three-State Buffer

- Dual 4-bit three state buffer with active low enable.
- Output takes one of 3 states: “high”, “low” or “high-impedance”.
  - High impedance means output is not connected.

Execution of “in al,9Fh” at CS:IP =01A7:0018

- **in al,9Fh** is coded by two bytes: E4h 9Fh.
- **Processor**
  - fetches E4h from the memory.
  - fetches the port address byte, 9Fh
  - Reads the port inputs (=FE) into AL register.
  - When IN instruction is over processor starts to fetch the next instruction.
80286 16-bit I/O instructions

- 80286 supports 16-bit I/O in a single I/O bus cycle.
  - `in ax, <p8>` ; port(p8) → al ; port(p8+1) → ah
  - `in ax, dx` ; port(dx) → al ; port(dx+1) → ah
  - `out <p8>, ax` ; al → port(p8) ; ah → port(p8+1)
  - `out dx, ax` ; port(dx) → al ; port(dx+1) → ah

8086 - 8088 Control Bus

- 8088 has three control signals to specify bus operation:
  - `IO/~M` specifies that bus cycle is either for IO or for Memory access.
  - `~RD` specifies that bus cycle is not read cycle.
  - `~WR` specifies that bus cycle is not write cycle.
- For independent memory and IO systems we need independent signals for Memory-Write, Memory-Read, IO-Write, and IO-Read.

PC Memory and IO control Signals

- IBM-PC uses
  - `MEMW, MEMR` for Memory-Write, Memory-Read,
  - `IOW, IOR` for IO-Write, and IO-Read

8088 Memory Read Timing

- A memory-read bus-cycle takes four clock cycles, T1, T2, T3, and T4.
- ALE high during T1 transfers AD0-AD7 to A0-A7.
- 8088 reads AD0-AD7 at the end of T3 cycle.
- Memory circuit may request additional wait cycles, using the ~READY input of 8088.
Memory Wait Cycles

Consider processor clock $T=50\text{ns}$, memory read cycle time $T_{RC} = 280\text{ns}$.

a) Draw the 8088 memory read timing diagram while it reads $5Ah$ from memory location $0A4401h$.

- Read-bus-cycle takes 7 clock cycles ($=350\text{ns}$).
- 3 of these cycles are wait cycles.

8088 Bus Control Outputs

- **ALE** (address latch enable) indicates a valid address is available on AD0…AD7.
- $\sim$**DEN** (active low data enable) enables data buffer.
- $\sim$**RD** (active low read) defines bus cycle is read cycle.
- $\sim$**WR** (active low write) that bus cycle is write cycle.
- $\sim$**READY** (active low ready) for slow devices to insert wait states until $\sim$READY=low.
- **NMI** (positive edge triggered non-maskable-interrupt)
- **INTR** (active-high interrupt) External interrupts.
- **RESET** (active-high reset) Resets the IP and Segment Registers to CS=FFFFh, IP =SS =DS =ES = 0000h.
- **CLOCK** (clock) to synchronize the bus states.
- **~READY** (active low ready) for slow devices to insert wait states until ~READY=low.

8088 Boot ROM

- After a power-on reset, 8088 starts with CS=FFFFh, and IP=0000h.
- Physical address FFFF0h is called the reset vector. This location shall contain an instruction (usually a far jmp instruction)
- A Boot ROM should occupy the higher address block of 1MByte address space.
8088 in maximal mode

- Maximal mode is designed for 8288 bus controller IC.
- 8288 converts S0-S1-S2 to IO and Memory control signals.

<table>
<thead>
<tr>
<th>S2</th>
<th>S1</th>
<th>S0</th>
<th>Processor State</th>
<th>8288 Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Interrupt acknowledge</td>
<td>INTA</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Read input/output port</td>
<td>IORC</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Write input/output port</td>
<td>IOWC, A1OWC</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Halt</td>
<td>None</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Code access</td>
<td>MRDC</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Read memory</td>
<td>MRDC</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Write memory</td>
<td>MWTC, AMWC</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Passive</td>
<td>None</td>
</tr>
</tbody>
</table>

8288 Bus Signaling.

- ~MRDC (~memory read command) = ~MEMR.
- ~MWTC (~memory write command) = ~MEMW.
- ~IORC (~I/O read command) = ~IOR.
- ~IOWC (~I/O write command) = ~IOW.
- ~INTA (~interrupt acknowledge) = ~INTA.
- DT/~R (data transmit/~receive)
- DEN (data enable)
- ALE (address latch enable) for demux of AD0-7

8-bit Section of ISA BUS

- IBM PC-AT bus later became ISA bus.
  - ISA = Industry Standard Architecture
- IBM PC first design used 8088 (8-bit data bus) device.
  - 8-bit ISA emerged from IBM-PC 8088 design.
- IBM AT used 80286 with 16-bit data bus.
  - 16-bit ISA started with this design.

8-bit Section of ISA Bus

- IBM PC system bus later became 8-bit ISA bus.
8-bit Section of ISA Bus

- **SA[0–19]** are system address lines
- **SD[0-7]** are system data lines
- **~SMEMW, ~SMEMR, ~IOW, ~IOR, AEN** are memory and IO control lines.

80286 and 16-bit ISA Bus

- **80286 has 16-bit data bus**
  - Address lines A0-A23 can define only a single byte.
  - ~BHE completes A0 to enable D0-7 or D8-15.
    - ~BE0=0 A0=0 enables both banks (bits D0 – 15)
    - ~BE0=0 A0=1 enables bank-1 (bits D8 – 15) only
    - ~BE0=1 A0=0 enables bank-0 (bits D0 – D7) only
    - ~BE0=1 A0=1 reserved for data bus is idle.
- **80286 has two modes:**
  - Real mode uses 20 address bits
  - Protected mode uses 24 address bits.
- This course will will show only real mode operation.

<table>
<thead>
<tr>
<th>~BHE</th>
<th>A0</th>
<th>Data Bus Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Transferring on D0-D15 (16-bit)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Transferring on D8-D15 (8-bit, odd address)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Transferring D0-D7 (8-bit, even address)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Reserved (Data bus is idle)</td>
</tr>
</tbody>
</table>

80286 Read Cycle

Memory read cycle timing on 80286 bus showing timing of DEN, DT/~R, ~READY lines.

PC-AT ISA – PCI Bus

- **Additional lines contain**
  - **SD8-15** system data lines,
  - **LA17-23** address lines.
  - **SBHE** High-bank enable line.
  - **Additional interrupt lines.**
- **You have to know**
  - how many address and data lines it has.
  - main memory and i/o control lines of ISA-bus.
What Is the Next?

- We have to understand the device select mechanism in the memory and io subsystems.
  - You shall be able to answer the memory timing and wait state questions.
    - Example: A DRAM has read cycle time $T_{RC}=190\text{ns}$, but write cycle time $T_{WC}=210\text{ ns}$.
      - Find the necessary wait states for this DRAM if interfaced to an 8088 bus that works at 20MHz processor frequency.
      - Find the wait states for an 80286 bus that works at the same processor frequency.
  - You shall be able to draw a memory or IO timing chart with address and data values assigned on it.