8086/8088 Hardware Specifications

- **Power supply:**
  - +5V with tolerance of ±10%;
  - 360mA.

- **Input characteristics:**
  - Logic 0 – 0.8V maximum, ±10μA maximum;
  - Logic 1 – 2.0V minimum, ±10μA maximum.

- **Output characteristics:**
  - Logic 0 – 0.45V maximum, 2mA maximum;
  - Logic 1 – 2.4V minimum, -400μA maximum.
Pinout

- Pins are divided into three separate buses:
  - Address (output);
  - Data (input/output);
  - Control (input/output);
Pinout

- **Address/Data bus:**
  - AD7-AD0 (8088) or AD15-AD0 (8086).
  - These lines are multiplexed address and data.

- **Address bus:**
  - A15-A8 (8088).
  - These are address lines only.

- **Address/Control (status) bus:**
  - A19/S6 – A16/S3.
  - These lines are multiplexed address and status.
  - S6 is always logic 0, S5 indicates condition of the IF flag bits, S4 and S3 indicate which segment is being accessed during the current bus cycle.
Pinout

Control bus:
- RD’
  - When asserted it indicates a read operation is happening.
- READY
  - When READY is logic 0 the microprocessor inserts wait states into the timing of the processor.
- INTR
  - Used for peripherals to request a hardware interrupt.
- TEST’
  - This pin is tested by the WAIT instruction, if asserted WAIT behaves as a NOP, otherwise the WAIT instruction waits for TEST’ to become logic 0.
- NMI
  - Similar to INTR except it cannot be masked.
Pinout

- **Control Bus**
  - **RESET**
    - Causes the processor to reset itself.
  - **CLK**
    - Clock input to the processor.
  - **Vcc**
    - Power supply connection, 5.0V, ±10%.
  - **GND**
    - Power supply connection. Note that both ground pins must be connected for proper operation.
  - **MN/MX’**
    - Selects minimum or maximum mode of operation.
  - **BHE’/S7**
    - Bus high enable is used in the 8086 to enable the most significant data bus during a read or write operation. S7 is always logic 1.
Pinout

- **Minimum Mode:**
  - IO/M’(8088) or M/IO’(8086)
    - Indicates if the processor is accessing a memory address or an I/O port address.
  - WR’
    - When asserted it indicates a write operation is happening.
  - INTA’
    - Signal a response to an interrupt request.
  - ALE
    - Indicates that the address/data bus contains address information.
  - DT/R’
    - Data transmit/receive indicates that the data bus is transmitting or receiving information.
Pinout

- **Minimum Mode:**
  - **DEN**
    - Data bus enable activates the external data bus buffer.
  - **HOLD**
    - This input receives DMA - direct memory access requests.
  - **HLDA**
    - When asserted this pin acknowledges that the processor entered a hold state.
  - **SS0’**
    - Equivalent to S0 pin in maximum mode.
Pinout

- **Maximum Mode:**
  - S2’, S1’ and S0’
    - These status bits indicate the function of the current bus cycle.
  - RO’/GT1’ and RO’/GT0’
    - Request/grant pins used for DMA during maximum mode operation.
  - LOCK’
    - Used to lock peripheral off the system.
  - QS1’ and QS0
    - Queue status bits.
The combination of some of the pins of the microprocessor indicates several different functions:

- Bus cycle status;
- Bus control functions;
- Queue status.

### Table 9-5: Bus cycle status (8088) using SS0

<table>
<thead>
<tr>
<th>IO/M</th>
<th>DT/R</th>
<th>SS0</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Interrupt acknowledge</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Memory read</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Memory write</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Halt</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Opcode fetch</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>I/O read</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>I/O write</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Passive</td>
</tr>
</tbody>
</table>

### Table 9-6: Bus control functions generated by the bus controller (8288) using S2, S1, and S0

<table>
<thead>
<tr>
<th>S2</th>
<th>S1</th>
<th>S0</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Interrupt acknowledge</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>I/O read</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>I/O write</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Halt</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Opcode fetch</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Memory read</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Memory write</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Passive</td>
</tr>
</tbody>
</table>

### Table 9-7: Queue status bits

<table>
<thead>
<tr>
<th>QS1</th>
<th>QS0</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Queue is idle</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>First byte of opcode</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Queue is empty</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Subsequent byte of opcode</td>
</tr>
</tbody>
</table>
8284A Clock Generator

- The 8284 provides the 8086/8088 system with:
  - Clock generation;
  - RESET sync.;
  - READY sync.
  - TTL peripheral clock signal.
Buffering and Latching

- **74LS245** – Octal Bus Transceivers with Tri-state Outputs.

- **74LS373** – Octal Transparent Latch with Tri-state Outputs.
Buffering and Latching
Buffering and Latching
Bus Timing

---

Two diagrams illustrating bus timing with labels:
- **CLK**: Clock signals
- **ADDRESS**: Address signals
- **ADDRESS/DATA**: Address and data signals
- **RD** (Read)
- **WR** (Write)

The timelines are segmented into:

1. **T1**
2. **T2**
3. **T3**
4. **T4**

**Top Diagram**:
- **One Bus Cycle**
- **Valid Address**
- **Data from Memory**

**Bottom Diagram**:
- **One Bus Cycle**
- **Valid Address**
- **Data Written to Memory**
Wait State

- Wait state is an extra clocking period ($T_w$) inserted between cycles $T_2$ and $T_3$. 

![Diagram showing clock and ready signal with time intervals]
Wait State Generation

![Diagram of wait state generation]
Minimum Vs. Maximum Mode

- **Minimum mode** is the least expensive way to operate a 8086/8088 system.
  - Control signals are generated by processor.
  - Good backward compatibility with earlier 8085A 8 bit processor.

- **Maximum mode** provides greater versatility at a higher cost.
  - Control signals are generated by external controllers.
  - Can be used with the 8087 math coprocessor.
  - Can be used with multiprocessor systems.
Minimum Mode
Memory Devices

- May be classified as:
  - ROM;
  - Flash;
  - SRAM;
  - DRAM.

- Connections:
  - Address;
  - Data;
  - Selection;
  - Control.
Address Decoding

- Addresses must be decoded to properly select a memory chip or port.
- This decoded signal will select specific devices that will communicate with the processor through the data and control buses.
- Several different methods may be used in address decoding:
  - Gates;
  - Decoders:
  - ROMs;
  - PLDs.
Address Decoding

FIGURE 10–13  A simple NAND gate decoder used to select a 2716 EPROM memory component for memory locations FF800H–FFFFFFH.

FIGURE 10–15  A circuit that uses eight 2764 EPROMs for a 64K × 8 section of memory in an 8088 microprocessor-based system. The addresses selected in this circuit are F0000H–FFFFFH.
Address Decoding

TABLE 10–1  The 82S147 PROM programming pattern for the circuit of Figure 10–17.

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>(OE) (A8) (A7) (A6) (A5) (A4) (A3) (A2) (A1) (A0)</td>
<td>(O0) (O1) (O2) (O3) (O4) (O5) (O6) (O7)</td>
</tr>
<tr>
<td>0 0 0 1 1 1 1 0 0 0</td>
<td>0 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>0 0 0 1 1 1 1 0 0 1</td>
<td>1 0 1 1 1 1 1 1</td>
</tr>
<tr>
<td>0 0 0 1 1 1 1 0 1 0</td>
<td>1 1 0 1 1 1 1 1</td>
</tr>
<tr>
<td>0 0 0 1 1 1 1 0 1 1</td>
<td>1 1 0 1 1 1 1 1</td>
</tr>
<tr>
<td>0 0 0 1 1 1 1 1 0 0</td>
<td>1 1 1 1 0 1 1 1</td>
</tr>
<tr>
<td>0 0 0 1 1 1 1 1 0 1</td>
<td>1 1 1 1 1 0 1 1</td>
</tr>
<tr>
<td>0 0 0 1 1 1 1 1 1 0</td>
<td>1 1 1 1 1 1 0 1</td>
</tr>
<tr>
<td>0 0 0 1 1 1 1 1 1 1</td>
<td>1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>all other combinations</td>
<td>1 1 1 1 1 1 1 1</td>
</tr>
</tbody>
</table>
Address Decoding
8088 Example

- Interfacing to an 8088 processor of 512K bytes of SRAM using sixteen 62255.
- The 62255 is a 32k X 8 SRAM.
- Memory is located from 00000H to 7FFFFFFH.
8086 Example

Note: A0 is labeled BLE (Bus low enable) on the 80386SX.

<table>
<thead>
<tr>
<th>BHE</th>
<th>BLE (A0)</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Both banks enabled for a 16-bit transfer</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>High bank enabled for an 8-bit transfer</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Low bank enabled for an 8-bit transfer</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>No banks enabled</td>
</tr>
</tbody>
</table>
8086 Example

- Interfacing:
  - 32k X 16 EPROM;
  - 4 X 27128 EPROM;
  - 0F0000H-0FFFFFH;
  - 64K X 16 SRAM;
  - 4 X 62256 SRAM;
  - 00000H-1FFFFFH
I/O Interfacing

- May be classified as:
  - Isolated I/O;
  - Memory mapped I/O.

- Instructions:
  - IN accumulator, source;
  - OUT destination, accumulator.
Basic I/O Interfaces
Debouncing

- Mechanical switches bounce when they are actuated. A circuit is needed to ensure that the output of the switch provides a single transition upon the switch actuation, instead of a sequence of transitions. This circuit is called a debouncer.
Port Example

- 16 bits output port decoded at addresses 40H and 41H.
- It used two latches (74ALS374) and a 16L8 used to decode the addresses.
The Intel 82C55 programmable peripheral interface is a low cost device that allow the user 24 I/O connections which may be grouped in different ways with up to three different modes of operation.
The 82C55 is programmed by sending a command byte to the control register. This command byte defines how the 82C55 will work. It defines the mode of operation and which ports are input or output.

<table>
<thead>
<tr>
<th>$A_1$</th>
<th>$A_0$</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Port A</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Port B</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Port C</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Command Register</td>
</tr>
</tbody>
</table>
8 Digit LED Display Example
20481 LCD Display Example

- 4 lines by 20 characters display that accepts ASCII code as data.

- Few connections necessary for operation:
  - 8 data;
  - 3 control;
    - R/W’(1=read,0=write)
    - RS(1=data,0=command)
    - E(1=enabled)

- Commands are defined in table 11-3 of text.
Keyboard Interface Example
Stepper Motor Interface Example

- Rotation is accomplished in full or half steps.
  - Full steps sequence:
    - 33H, 66H, 0CCH, 99H
  - Half step sequence:
    - 11H, 33H, 22H, 66H, 44H, 0CCH, 88H, 99H.