Pin-Outs

- Both **8086** and **8088** come in **40-pin DIP packages**.
- **8086** has a **16-bit data bus** (AD0–AD15).
- **8088** has an **8-bit data bus** (AD0–AD7).
- Main difference: **8086 transfers 16-bit data at once → faster**.
- Some minor differences in control pins:
    - **8086:** has **M** pin.
    - **8088:** has **M/IO** pin.
    - **Pin 34:** 8086 → **/S7**, 8088 → **SS0**.

Power Supply

- Both need **+5 V** (tolerance ±10%).
- Max supply current:
    - 8086 → **360 mA**
    - 8088 → **340 mA**
- Operating temp: **32°F to 180°F** (not very wide).
- CMOS versions (80C86 / 80C88) are low-power (**10 mA**) and work in extreme temps (**-40°F to +225°F**).

DC Characteristics

- **Input pins:** Need very small current (leakage current).
    - Logic 0 → max 0.8 V
    - Logic 1 → min 2.0 V
- **Output pins:** Can drive standard logic circuits.
    - Logic 0 → max 0.45 V
    - Logic 1 → min 2.4 V
- Noise immunity slightly less than standard because 0.45 V vs 0.4 V for logic 0.
- **Fan-out:** Max **10 loads per output pin**; use buffer if more.

Recommended connections: TTL (74LS, 74ALS) or CMOS (74HC) logic.

Pin Connections (Example)

- **AD0–AD7 (8088):** Multiplexed Address/Data pins.
    - When **ALE = 1** → they carry **address**.
    - When **ALE = 0** → they carry **data**.
    - High-impedance during **HOLD**.

(8086 has AD0–AD15; function is the same, just 16 bits.)

Key Point for Exams:

- 8086 → 16-bit bus → faster.
- 8088 → 8-bit bus → cheaper, slower.
- Power: +5V, 360/340 mA.
- Multiplexed AD pins → address or data depending on ALE.
- Fan-out ≤ 10 without buffer.

Fan-Out Table

- **Fan-out** = How many standard logic devices a microprocessor output can drive.
- Recommended fan-out for 8086/8088 pins:

Logic Family	Sink Current	Source Current	Max Fan-out
TTL (74)	1.6 mA	40 µA	1
TTL (74LS)	0.4 mA	20 µA	5
TTL (74ALS)	0.1 mA	20 µA	10
TTL (74F)	0.5 mA	25 µA	10
CMOS (74HC)	10 µA	10 µA	10
NMOS	10 µA	10 µA	10

Tip: Fan-out >10 → use a **buffer**.

Status Bits S3 & S4 (8086/8088)

S4	S3	Segment Function
0	0	Extra segment
0	1	Stack segment

1	0	Code or no segment
1	1	Data segment

Address / Data Pins

- **A15–A8 (8088):** Upper 8 bits of memory address, high-impedance during HOLD.
- **AD15–AD8 (8086):** Upper 8 bits of address/data bus (multiplexed).
    - ALE = 1 → address
    - ALE = 0 → data
    - HOLD → high-impedance
- **A19–A16 / S6–S3:** Multiplexed for address and status bits.
    - S6 = always 0
    - S5 = IF flag (interrupt enable)
    - S4–S3 = segment info (see table above)
    - Can address **4 memory banks** using S4/S3 as A21/A20.

Control Pins (Easy Explanation)

Pin	Function
RD	Read: when 0 → microprocessor reads data from memory/I/O
WR	Write: when 0 → microprocessor writes data to memory/I/O
READY	Wait state input. 0 → microprocessor waits; 1 → normal operation
INTR	Interrupt request. Checked only if IF = 1
NMI	Non-maskable interrupt. Works even if IF = 0
RESET	Reset microprocessor. Starts from FFFF0h and disables interrupts
CLK	Clock input. Must be 33% duty cycle (high 1/3, low 2/3)
VCC / GND	Power supply (+5V) and ground (two GND pins)
MN/MX	Select minimum mode (+5V) or maximum mode (GND)
ALE	Address latch enable. Indicates bus has address
DT/R	Data transmit/receive. Shows direction of data
DEN	Data bus enable for external buffers
HOLD / HLDA	HOLD requests DMA; HLDA acknowledges HOLD
INTA	Interrupt acknowledge signal
TEST	Tested by WAIT instruction (used with 8087 coprocessor)

BHE / S7	Bus High Enable (8086 only). Enables D15–D8
M / IO	Memory or I/O selection

READY — Wait state input

- **What it does:** CPU checks this pin to see if external devices are ready.
- **Logic:**
  - 0 → CPU **waits** (stops) until READY becomes 1
  - 1 → CPU operates **normally**
- **Bangla meaning:** যদি কোনো external device data দিতে সময় নেয়, তাহলে CPU একটু অপেক্ষা করবে। READY = 1 হলে CPU চলে যায়।

INTR — Interrupt request

- **What it does:** External devices can ask CPU to stop current task and handle something urgent.
- **Checked only if IF = 1** → CPU only responds if **Interrupt Enable flag** is set.
- **Bangla meaning:** অন্য কোনো device CPU কে বলছে, "আমার সাথে কাজ করো।" CPU তখন IF = 1 হলে এই request মানে।

NMI — Non-maskable interrupt

- **What it does:** Urgent interrupt that **cannot be ignored**, even if IF = 0.
- **Bangla meaning:** সবচেয়ে জরুরি interrupt। CPU বাধ্যতামূলকভাবে এটা handle করবে।

RESET — Reset microprocessor

- **What it does:** CPU resets and starts execution from **FFFF0h**.
- **Also disables interrupts** temporarily.
- **Bangla meaning:** Reset বোঝায়, CPU আবার fresh start নেবে। Interrupts বন্ধ থাকবে যতক্ষণ না normal শুরু হয়।

CLK — Clock input

- **What it does:** Gives the timing signal for CPU operations.
- **Duty cycle:** High 1/3, Low 2/3 → CPU expects **high for 1/3 of cycle, low for 2/3**
- **Bangla meaning:** CPU এই clock signal অনুযায়ী step by step instruction execute করে।

READY	অপেক্ষা করতে হবে কি না দেখবে
INTR	normal interrupt request
NMI	বাধ্যতামূলক urgent interrupt
RESET	CPU fresh start
CLK	timing signal

What is Fan-out?

Fan-out হল **একটি logic gate কতগুলো অন্য gate-কে drive করতে পারে** তা বোঝায়।

- সহজভাবে বললে:এক gate এর output কতগুলো gate এর input-কে ঠিকভাবে signal দিতে পারে।
- Example: যদি একটি NOT gate-এর output **10টা অন্য gate-এর input চালাতে পারে**, তাহলে তার **fan-out = 10**।

Why "≤10 without buffer"?

- Some gates have **limited driving strength** — মানে একদম বেশি input এ signal পাঠালে voltage বা current drop করতে পারে।
- **≤10 without buffer** মানে:

কোনো extra buffer বা amplifier ছাড়া, ওই gate 10 টা input পর্যন্ত safely drive করতে পারবে।

- যদি input এর সংখ্যা 10 এর বেশি হয় → **signal degrade হতে পারে** → ভুল logic behavior।
- **Gate** = water tap
- **Input of other gates** = glasses
- **Fan-out = number of glasses tap safely fill করতে পারে**

যদি tap-এ 10 glasses পর্যন্ত পানি ঠিকমতো যায় → fan-out = 10
11th glass → পানি ঠিকমতো যাবে না → signal degrade

Fan-out = একটি gate কতগুলো অন্য gate চালাতে পারে। "≤10 without buffer" মানে 10 এর বেশি gate চালাতে গেলে extra buffer লাগবে।

Bus Cycle Status (8088) — Table 9–5

- The bus cycle tells what the CPU is doing on the system bus.

SS0	IO/	M	DT/R	Function

0	0	0	0	Interrupt acknowledge
0	0	1	1	Memory read
0	1	0	0	Memory write
0	1	1	1	Halt
1	0	0	0	Opcode fetch
1	0	1	1	I/O read
1	1	0	0	I/O write
1	1	1	1	Passive (idle)

This table shows the **8085 microprocessor's status signals** (`SS0`, `IO/`, `M`, `DT/R`) and the corresponding **operation the CPU is performing**.

- **SS0** → Status signal 0 (together with other signals, defines current operation)
- **IO/** → `0` for Memory, `1` for I/O
- **M** → Memory/Opcode distinction
- **DT/R** → Data Transmit/Receive (1 = CPU reads, 0 = CPU writes)
- **Function** → What the microprocessor is actually doing

How to read it (step by step)

1️⃣ Look at the **status signals**: `SS0`, `IO/`, `M`, `DT/R`
2️⃣ Combine their values — the microprocessor **decodes these internally** to know what it's doing.
3️⃣ Check the **function column** → tells the **actual operation**.

Examples:

1. **Row 1:** `SS0=0, IO/=0, M=0, DT/R=0`

- IO/ = 0 → Memory operation
- DT/R = 0 → Write
- M=0 → Not opcode fetch
- Function = **Interrupt acknowledge**

2. **Row 2:** `SS0=0, IO/=0, M=1, DT/R=1`

- IO/ = 0 → Memory
- M=1 → Opcode fetch (reading instruction)

- DT/R = 1 → Read
- Function = **Memory read**

3. **Row 7:** `SS0=1, IO/=1, M=0, DT/R=0`

- IO/ = 1 → I/O
- DT/R = 0 → CPU writes
- Function = **I/O write**

- **TRICK TO REMEMBER**
- **IO/** = 0 → Memory, 1 → I/O
- **M** = 1 → Opcode fetch / Read, 0 → Data or control
- **DT/R** = 1 → CPU reads from memory/I/O, 0 → CPU writes to memory/I/O

Bus Control by 8288 Bus Controller — Table 9–6

- Status bits **S2, S1, S0** tell the bus controller what operation to perform.

S2	S1	S0	Function
0	0	0	Interrupt acknowledge
0	0	1	I/O read
0	1	0	I/O write
0	1	1	Halt
1	0	0	Opcode fetch
1	0	1	Memory read
1	1	0	Memory write
1	1	1	Passive (idle)

This table is for **8085 microprocessor status signals** that tell **what the CPU is currently doing**.

Signal	Meaning
S2, S1, S0	Status bits output by CPU to indicate current operation

How to read it:

1. **S2 = 0** → CPU is performing **control or I/O operation**
2. **S2 = 1** → CPU is performing **memory operation**

Then check **S1 and S0**:

- **S1 = 0, S0 = 0** → Interrupt acknowledge
- **S1 = 0, S0 = 1** → I/O read
- **S1 = 1, S0 = 0** → I/O write
- **S1 = 1, S0 = 1** → Halt

Similarly, for S2 = 1 (memory/opcode operations):

- **S1 = 0, S0 = 0** → Opcode fetch
- **S1 = 0, S0 = 1** → Memory read
- **S1 = 1, S0 = 0** → Memory write
- **S1 = 1, S0 = 1** → Passive / idle

CPU generates these **3-bit status signals** during every machine cycle. External devices (like memory or I/O) can check these bits to know what operation is happening.

Queue Status Bits (QS1, QS0) — Table 9–7

- Show the **internal instruction queue** status:

QS 1	QS 0	Function
0	0	Queue is idle
0	1	First byte of opcode
1	0	Queue is empty
1	1	Subsequent byte of opcode

- **QS pins** are mainly used by **8087 coprocessor**.

This table tells **what is happening in the instruction queue** of 8085.

- 8085 has a **prefetch queue** (instruction queue) — stores instructions before CPU executes them.
- QS1 and QS0 **describe the status of this queue**.
- **0,0** → CPU hasn't fetched any instruction yet
- **0,1** → First byte fetched (instruction started prefetch)
- **1,0** → Queue empty (CPU executed all prefetched instructions)
- **1,1** → Multi-byte instruction — CPU reading 2nd/3rd byte

8284A Clock Generator — Operation Made

Clock Section (Timing for CPU & Peripherals)

- **X1 & X2:** Connect a crystal → oscillator generates square wave at crystal frequency.
- **OSC output:** Taken from inverting buffer → can be used as EFI input to other 8284A chips.
- **F/ pin:**
  - F/ = 0 → use **internal crystal oscillator**
  - F/ = 1 → use **external frequency input (EFI)**
- **Divide-by-3 counter:**
  - Takes oscillator output → creates **CLK signal** for 8086/8088.
  - Also helps with **ready synchronization** and **feeds second divide-by-2 counter**.
- **Divide-by-2 counter:**
  - Output = **PCLK (Peripheral Clock)** → 1/6 of crystal frequency, 50% duty cycle.

✅ **Example:**

- 15 MHz crystal → CLK = 5 MHz (CPU), PCLK = 2.5 MHz (peripherals).

Reset Section (Power-On or Manual Reset)

- Simple circuit: **Schmitt trigger + D-type flip-flop**.
- Ensures **RESET timing meets 8086/8088 requirements**:
  - RESET sampled at **positive edge** of CPU clock.
  - Flip-flop guarantees RESET goes high **within 4 clock cycles** after power-on.
- **RC circuit:**
  - Initially logic 0 → capacitor charges → logic 1.
  - Ensures RESET stays **high for ≥ 50 µs**.
- **Push-button:** Allows manual reset.
- CLK feeds CPU (divide-by-3)
- PCLK feeds peripherals (divide-by-6)
- RESET = stable logic 1 within 4 clocks, held ≥50 µs

Simple Connection Notes (Figure 9–4)

- **F/ & CSYNC → Ground** → select internal crystal oscillator
- **Crystal = 15 MHz** → provides:
  - CPU CLK = 5 MHz
  - Peripheral CLK = 2.5 MHz

"8284A generates CLK (1/3 crystal) for CPU, PCLK (1/6 crystal) for peripherals, and stable RESET using RC + flip-flop circuit. F/ selects internal/external clock, X1-X2 connect crystal."

LOCK Pin

- **LOCK** output is used to **lock peripherals**, preventing other devices from accessing the system temporarily.
- Activated by **LOCK prefix** in instructions.

8284A Clock Generator

- An **external chip** used with 8086/8088 to generate proper **CLK and RESET signals**.
- **Why needed:** Without 8284A, extra circuits needed for clock, reset, and READY signals.

8284A Pin Functions

Pin	Function
X1, X2	Connect external crystal for timing
F/ C & EFI	Select internal crystal or external clock
CLK	Clock output to 8086/8088 (1/3 crystal freq, 33% duty)
PCLK	Peripheral clock (1/6 crystal freq, 50% duty)
OSC	Oscillator output (same freq as crystal)
READY	Connects to 8086/8088 READY input for wait states
RDY1, RDY2 + AEN1, AEN2	Cause wait states (synchronize READY signal)
RES	Reset input (active-low)
RESET	Reset output to 8086/8088 RESET pin
CSYNC	Clock sync for multiple processors (ground if single CPU)
VCC / GND	Power +5V / Ground

Minimum vs Maximum Mode

- **Minimum mode:** MN/MX = +5V → for **single processor** systems.

- **Maximum mode:** MN/MX = GND → for **systems with coprocessors**.
- In maximum mode, **S0–S2** indicate the function of the current bus cycle (decoded by 8288 controller).
- **GT1 / RQ/GT0:** DMA request/grant pins (bidirectional in max mode).

8284A Operation

1. **Clock Section:**
    - Crystal → square wave → AND gate → divide-by-3 → CLK
    - Second divide-by-2 → PCLK for peripherals
    - Example: 15 MHz crystal → CLK = 5 MHz, PCLK = 2.5 MHz
    - **Reset Section:**
    - RC circuit → logic 0 → charges to logic 1 → applies RESET
    - D-type flip-flop → ensures RESET high within 4 clocks, stays ≥50 μs
    - Push-button → manual reset
    - **X1 & X2**

- এখানে একটি **crystal** (যেমন 15 MHz) সংযুক্ত করা হয়।
- Crystal + oscillator → produce করে **square wave** (timing signal)।
- এই signal দিয়ে CPU এবং peripherals চলবে।

OSC Output

- Oscillator output নেওয়া হয় **inverting buffer** এর মাধ্যমে।
- এই output অন্য 8284A chips এ EFI (External Frequency Input) হিসাবে ব্যবহার করা যায়।

F/ pin

- **F/ = 0** → 8284A **নিজের internal crystal oscillator** ব্যবহার করবে।
- **F/ = 1** → 8284A **external frequency input (EFI)** ব্যবহার করবে।

অর্থাৎ, CPU timing নিতে পারবে **internal crystal** অথবা **external clock signal** থেকে।

Divide-by-3 counter

- Oscillator output দেয়া হয় **divide-by-3 counter** এ।

- এটি **CLK signal** তৈরি করে যা **8086/8088 CPU** চালায়।
- এছাড়াও, ready synchronization এবং **দ্বিতীয় divide-by-2 counter** কে timing দেয়।

Divide-by-2 counter

- Output = **PCLK (Peripheral Clock)**
- Peripheral devices এই clock ব্যবহার করে।
- Frequency = **1/6 of crystal frequency**, Duty cycle = 50%
- Crystal = 15 MHz
- Divide-by-3 → CPU clock = 15 ÷ 3 = **5 MHz**
- Divide-by-2 → Peripheral clock = 5 ÷ 2 = **2.5 MHz**