

## Chapter 10

### ROM (Read Only Memory)

সংজ্ঞা:

ROM হলো এমন এক ধরনের মেমরি যেখানে ডেটা বা প্রোগ্রাম স্থায়ীভাবে সংরক্ষিত থাকে, এবং বিদ্যুৎ চলে গেলেও সেই ডেটা হারায় না।

👉 এজন্য একে বলে non-volatile memory।

#### ◆ বৈশিষ্ট্য:

- ROM-এ ডেটা পূর্বেই লেখা থাকে (pre-programmed)।
- ROM শুধু read করা যায়, লেখা যায় না।
- ROM ব্যবহার করা হয় যেখানে ডেটা স্থায়ীভাবে অপরিবর্তিত থাকতে হবে, যেমন BIOS, firmware ইত্যাদি।

#### ◆ ROM-এর বিভিন্ন ধরন:

##### 1. ROM (Fixed ROM):

ফ্যাক্টরিতেই স্থায়ীভাবে প্রোগ্রাম করা হয়। পরিবর্তন করা যায় না।

##### 2. PROM (Programmable ROM):

একবার প্রোগ্রাম করা যায়, কিন্তু পরে মুছে ফেলা যায় না।

প্রোগ্রাম করার সময় “fuse” পুড়িয়ে সেট করা হয়।

##### 3. EEPROM (Erasable Programmable ROM):

- বারবার প্রোগ্রাম করা যায়।
- মুছে ফেলতে হলে ultraviolet (UV) আলোতে প্রায় 20 মিনিট রাখা লাগে।
- প্রোগ্রাম করা হয় “EPROM Programmer” নামের ডিভাইস দিয়ে।

##### 4. EEPROM (Electrically Erasable PROM):

- বিদ্যুৎ দিয়েই erase ও reprogram করা যায়।
- একে Flash Memory নামেও ডাকা হয়।

##### 5. Flash Memory:

- EEPROM-এর উন্নত রূপ।
- কম্পিউটারের BIOS, মোবাইল ফোন, ক্যামেরা, পেনড্রাইভ, SD কার্ড—সবখানেই ব্যবহৃত হয়।
- বিদ্যুৎ ছাড়াও ডেটা থাকে (non-volatile)।

### \* Difference between PROM, EPROM and EEPROM

বিষয়	PROM	EPROM	EEPROM
<b>Full Form</b>	Programmable Read Only Memory	Erasable Programmable Read Only Memory	Electrically Erasable Programmable Read Only Memory
<b>Programming Method</b>	একবার প্রোগ্রাম করা যায় (fuse burning)	প্রোগ্রাম করা যায় EPROM programmer দিয়ে	প্রোগ্রাম ও মুছে ফেলা যায় বৈদ্যুতিকভাবে (electrically)
<b>Erasing Method</b>	Erase করা যায় না	UV light (ultraviolet light) দ্বারা erase করা হয়	Electrical signal দিয়েই erase করা যায়
<b>Reusability</b>	একবার ব্যবহারের পর আর ব্যবহার করা যায় না	বারবার erase ও reprogram করা যায়	সহজেই বারবার erase ও reprogram করা যায়
<b>Erase Time</b>	— (not erasable)	প্রায় 15–20 মিনিট UV আলোতে	খুব দ্রুত (মাইক্রোসেকেন্ড লেভেল)
<b>Erase Method Location</b>	—	চিপটিকে ডিভাইস থেকে খুলে UV আলোতে দিতে হয়	ডিভাইসের ভিতরেই (in-circuit) erase করা যায়
<b>Volatility</b>	Non-volatile (power off হলেও ডেটা থাকে)	Non-volatile	Non-volatile
<b>Cost</b>	সস্তা	মাঝারি	তুলনামূলকভাবে বেশি
<b>Speed</b>	দ্রুত পড়া যায়	পড়া দ্রুত, লেখা ধীর	পড়া ও লেখা দুটোই দ্রুত
<b>Common Use</b>	স্থায়ী firmware বা configuration ডেটা	পুরনো BIOS chip বা test firmware	আধুনিক BIOS, Flash memory, pen drive, SSD ইত্যাদি



## R \$RAM (Static RAM)

সংজ্ঞা:

SRAM হলো এমন মেমরি যা বিদ্যুৎ থাকা পর্যন্ত ডেটা ধরে রাখে, কিন্তু বিদ্যুৎ বন্ধ হলে মুছে যায়।

👉 এজন্য একে বলে volatile memory।

#### ◆ বৈশিষ্ট্য:

- ডেটা রাখতে কোনো “refresh” দরকার হয় না (তাই একে বলে *static*)।
- খুব দ্রুত কাজ করে।
- কিন্তু দাম বেশি, তাই সাধারণত **cache memory** হিসেবে ব্যবহৃত হয়।
- ছোট সাইজের হয় (1MB-এর নিচে সাধারণত)।

#### ◆ উদাহরণ:

- 4016 SRAM ( $2K \times 8$ )
- 6116 SRAM, 62256 SRAM ( $32K \times 8$ )

#### ◆ পিন ও কন্ট্রোল:

- CS (Chip Select): কোন চিপ সক্রিয় হবে তা নির্ধারণ করে।
- OE (Output Enable): আউটপুট সক্রিয় করে (রিড করার সময়)।
- WE (Write Enable): রাইট করার সময় ব্যবহৃত হয়।

#### ⚡ অ্যাক্সেস টাইম:

- SRAM-এর অ্যাক্সেস টাইম খুব দ্রুত (যেমন 120ns বা 150ns)।
- ক্যাশ মেমরিতে ব্যবহৃত SRAM-এর স্পিড আরও বেশি (1ns পর্যন্ত)।

## ৩ DRAM (Dynamic RAM)

### সংজ্ঞা:

DRAM-ও **volatile memory**, কিন্তু এটি ডেটা **capacitor**-এর চার্জ আকারে ধরে রাখে, যা সময়ের সাথে কমে যায়।

👉 তাই প্রতি 2–8 মিলিসেকেন্ডে refresh করতে হয়, নাহলে ডেটা হারিয়ে যায়।

#### ◆ বৈশিষ্ট্য:

- বড় সাইজের মেমরি তৈরি করা যায় (যেমন  $256M \times 8$ )।
- কম খরচে বেশি ক্ষমতা পাওয়া যায়।
- ধীরে কাজ করে SRAM-এর তুলনায়।
- প্রতিনিয়ত refresh করা লাগে (controller-এর মাধ্যমে)।

#### ◆ কেন Dynamic?

কারণ এর স্টোরেজ সেলগুলো “capacitor”-এর উপর নির্ভর করে, যেগুলোর চার্জ সময়ের সাথে নষ্ট হয়, তাই বারবার “refresh” দিতে হয়।

#### ◆ Address Multiplexing:

- DRAM-এ address pin সংখ্যা কমানোর জন্য row আর column address একসাথে ব্যবহৃত হয়।
- যেমন  $64K \times 4$  DRAM (TMS4464)-এ 16-bit অ্যাড্রেসকে দুই ধাপে পাঠানো হয়:
  - প্রথমে RAS (Row Address Strobe) signal দিয়ে row পাঠানো হয়।
  - তারপর CAS (Column Address Strobe) signal দিয়ে column পাঠানো হয়।

#### ✿ Difference between SRAM and DRAM

বিষয়	SRAM (Static RAM)	DRAM (Dynamic RAM)
Full Form	Static Random Access Memory	Dynamic Random Access Memory
Data Storage Method	Data সংরক্ষণ হয় flip-flop সার্কিটের মাধ্যমে	Data সংরক্ষণ হয় capacitor-এর চার্জ আকারে
Refresh Requirement	Refresh দরকার হয় না	প্রতি কয়েক মিলিসেকেন্ড পরপর refresh দরকার হয়
Speed	খুব দ্রুত (fast access time)	তুলনামূলকভাবে ধীর
Cost	দাম বেশি	দাম কম
Density (Storage capacity)	কম (small size, কম bit প্রতি chip)	বেশি (বড় মেমরি সাইজ তৈরি করা যায়)
Power Consumption	বেশি (কারণ flip-flop সার্কিট সবসময় power নেয়)	কম (capacitor-based design)
Used As	Cache memory (CPU cache, registers)	Main memory (RAM modules)
Complexity of Circuit	জটিল সার্কিট (6 transistor per bit)	সহজ সার্কিট (1 transistor + 1 capacitor per bit)
Volatility	Volatile (power off হলে ডেটা হারায়)	Volatile (power off হলে ডেটা হারায়)
Access Time	কম (দ্রুত)	বেশি (ধীর)

#### 🧠 সারসংক্ষেপে:

- SRAM = স্ক্রুত, দামি, cache-এর জন্য
- DRAM = ধীর, সস্তা, main memory-এর জন্য



## Address Decoding (অ্যাড্রেস ডিকোডিং) কী?

Address decoding হলো এমন একটি প্রক্রিয়া যার মাধ্যমে microprocessor-এর address signal থেকে নির্দিষ্ট একটি memory chip বা I/O device নির্বাচন (select) করা হয়।

👉 সহজভাবে বললে:

মাইক্রোপ্রসেসর অনেক বড় address space (যেমন 1MB) ব্যবহার করে, কিন্তু প্রতিটি memory chip বা device ছোট জায়গা (যেমন 2KB, 4KB ইত্যাদি) নেয়।

তাই “কোন address এ কোন chip কাজ করবে” — এটা নির্ধারণ করার জন্য address decoding দরকার।

### ◆ কেন Address Decoding দরকার?

ধরা যাক,

- 8088 মাইক্রোপ্রসেসরের ২০টি address pin (A0–A19) আছে → তাই সে 1MB ( $2^{20}$  bytes) address করতে পারে।
  - কিন্তু একটা 2716 EPROM চিপের মাত্র ১১টি address pin (A0–A10) আছে → অর্থাৎ সে  $2K \times 8 = 2048$  bytes (2KB) ডেটা রাখতে পারে।
- ◆ যদি আমরা শুধু A0–A10 EPROM-এর সাথে সংযুক্ত করি, তাহলে প্রসেসর ওই 2KB জায়গার মধ্যেই ঘুরে ঘুরে কাজ করবে।
- ◆ ফলে পুরো 1MB memory space-এর বাকি অংশ “অচেনা” থেকে যাবে।

👉 এই mismatch ঠিক করার জন্যই address decoder ব্যবহার করা হয়।

### ✿ Address Decoder কী করে?

Address decoder দেখে address bus-এর কোন লাইনগুলোতে 1 বা 0 আছে, তার ভিত্তিতে নির্ধারণ করে কোন memory chip select হবে।

অর্থাৎ:

নির্দিষ্ট address range (যেমন 00000H–007FFH বা F8000H–F87FFH) এ গেলে শুধু সেই নির্দিষ্ট chip “enable” হবে।

## ◆ Simple NAND Gate Decoder (সহজ উদাহরণ)

ধরা যাক, আমরা 2KB ( $2K \times 8$ ) EPROM ব্যবহার করছি।

👉 এজন্য A0–A10 সরাসরি EPROM-এর address পিনে যাবে।

👉 বাকি address line A11–A19 (মোট ৯টা) আমরা NAND gate-এর ইনপুটে দিচ্ছি।

### \* কীভাবে কাজ করে:

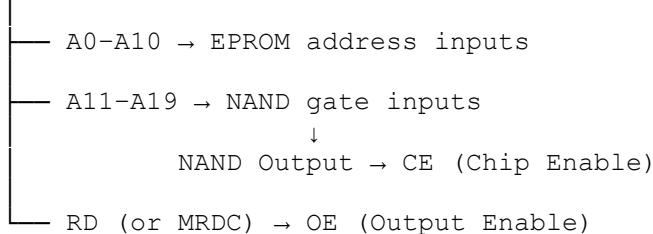
- যখন A19–A11 সবগুলো 1 হয়,  
তখন NAND gate-এর আউটপুট হয় 0 (Low)।
- এই Low সিগনালটি দেওয়া হয় EPROM-এর CE (Chip Enable) পিনে।
  - অর্থাৎ,  $CE = 0$  হলে EPROM কাজ করে।
- তারপর 8088 এর RD signal বা MRDC (Memory Read Control) সিগনাল OE (Output Enable) পিনে যায়।
  - $CE = 0$  এবং  $OE = 0$  হলে  $\rightarrow$  EPROM ডেটা আউটপুট দেয়।

### ■ সুতরাং:

যখন address bus-এর উপরের ৯টি বিট (A19–A11) সব 1 হয়, তখন শুধুমাত্র সেই EPROM চিপটি “select” হয় এবং data পাঠায়।

## ◆ উদাহরণস্মরণ চিত্র (বর্ণনা আকারে)

8088 Address Bus (A19–A0)



## ✿ সারসংক্ষেপ:

বিষয়	ব্যাখ্যা
মূল উদ্দেশ্য	কোন address-এ কোন মেমরি চিপ কাজ করবে তা নির্ধারণ করা ✓
কেন দরকার	মাইক্রোপ্রসেসরের address line সংখ্যা ও মেমরির address pin সংখ্যা মেলে না ✓
ডিকোডিং ডিভাইস	NAND gate, decoder IC (যেমন 74LS138), PLD ইত্যাদি ✓
ফলাফল	নির্দিষ্ট address range অনুযায়ী সঠিক মেমরি chip সক্রিয় হয় ✓

## Chapter 11



### I/O Interface (Input/Output Interface) কী?

I/O Interface হলো এমন একটি যোগাযোগ মাধ্যম (communication link) যার মাধ্যমে মাইক্রোপ্রসেসর বাইরের ডিভাইসগুলোর (যেমন: কীবোর্ড, মনিটর, প্রিন্টার, সেন্সর ইত্যাদি) সাথে ডেটা আদান-প্রদান করে।

👉 অর্থাৎ, এটা হলো মাইক্রোপ্রসেসর ও বাইরের বিশ্বের মধ্যে সংযোগের সেতু।



### কেন I/O Interface দরকার

মাইক্রোপ্রসেসর নিজে সরাসরি বাইরের ডিভাইসের সাথে কথা বলতে পারে না, কারণ:

- প্রসেসর কাজ করে ডিজিটাল বাইনারি সিগন্যাল (0 ও 1) দিয়ে।
- কিন্তু বাইরের ডিভাইসের কাজ করার পদ্ধতি অনেক ভিন্ন।

👉 তাই একটি I/O Interface circuit লাগে, যা ডেটা ফরম্যাট ও timing মেলাতে সাহায্য করে।



### I/O Instructions

I/O ডিভাইসের সাথে যোগাযোগ করার জন্য মাইক্রোপ্রসেসরে কিছু নির্দিষ্ট নির্দেশ (instruction) আছে:

Instruction	কাজ
IN	ইনপুট ডিভাইস থেকে ডেটা পড়ে (data read করে)
OUT	আউটপুট ডিভাইসে ডেটা পাঠায় (data write করে)
INS	ইনপুট ডিভাইস থেকে মেমরিতে ডেটা পাঠায় (string আকারে)
OUTS	মেমরি থেকে আউটপুট ডিভাইসে ডেটা পাঠায় (string আকারে)

## ◆ IN এবং OUT Instruction-এর কাজ

### 1. IN instruction

- ব্যবহার হয় I/O device → Processor ডেটা আনার জন্য।
- ডেটা accumulator-এ (AL, AX বা EAX) আসে।
  - ◆ উদাহরণ: IN AL, 60H → Port 60H থেকে ডেটা পড়বে।

### 2. OUT instruction

- ব্যবহার হয় Processor → I/O device-এ ডেটা পাঠানোর জন্য।
  - ◆ উদাহরণ: OUT 60H, AL → AL এর মান Port 60H তে পাঠাবে।

## ◆ Port Address (I/O Address)

প্রতিটি I/O ডিভাইসের একটা নির্দিষ্ট address (port number) থাকে।

যেভাবে memory-র প্রতিটি cell-এর ঠিকানা থাকে, ঠিক তেমনি I/O ডিভাইসেরও একটি ঠিকানা থাকে।

- **8-bit Port address:** 00H – FFH
- **16-bit Port address:** 0000H – FFFFH

■ 8086/8088 মাইক্রোপ্রসেসর ১৬-বিট পর্যন্ত I/O address ব্যবহার করতে পারে।  
অর্থাৎ, মোট 65536 (64K) I/O port address সম্ভব।

## ◆ Fixed Address vs Variable Address

ধরন	বর্ণনা
Fixed Address (p8)	8-bit I/O address → instruction এর মধ্যেই থাকে (যেমন IN AL, 20H)
Variable Address (DX)	16-bit address → DX register-এ থাকে (যেমন OUT DX, AL)

---

## ◆ INS & OUTS Instructions (String I/O)

এই দুইটি instruction ব্যবহার হয় একসাথে অনেক byte বা word ডেটা transfer করার জন্য।

Instruction	কাজ	ব্যবহার করা রেজিস্টার
INS	Input device → Memory	ES:DI
OUTS	Memory → Output device	DS:SI

- এগুলো REP prefix এর সাথে ব্যবহার করলে একাধিক ডেটা (string) একসাথে পাঠানো যায়।
- 

## ✿ I/O Address Decoding

যখন IN/OUT instruction চালানো হয়, তখন I/O address (port number) **address bus**-এ পাঠানো হয়।

এরপর I/O interface সার্কিট এই address decode করে নির্দিষ্ট ডিভাইস সিলেক্ট করে।

- 👉 এক কথায়:

মাইক্রোপ্রসেসর address পাঠায় → decoder দেখে কোন I/O device সাড়া দেবে।

## ✿ Types of I/O Mapping

I/O interface দু'ভাবে কাজ করতে পারে:

ধরন	ব্যাখ্যা
Isolated I/O (Direct I/O)	আলাদা I/O address space ব্যবহার করে (IN/OUT instruction প্রয়োজন)
Memory-Mapped I/O	I/O device-গুলোকে memory-এর মতো treat করা হয় (MOV instruction দিয়েই access করা যায়)

---

## সংক্ষিপ্ত সারমর্ম

বিষয়	ব্যাখ্যা
উদ্দেশ্য	Processor ও I/O ডিভাইসের মধ্যে যোগাযোগ করা
মূল নির্দেশনা	IN, OUT, INS, OUTS
Port Address	8-bit বা 16-bit হতে পারে
Address decoding	নির্দিষ্ট I/O ডিভাইস select করার প্রক্রিয়া
Mapping Types	Isolated I/O ও Memory-mapped I/O

### ◆ Basic Input and Output Interface কী?

 মাইক্রোপ্রসেসর নিজে বাইরের জিনিসগুলো (যেমন সুইচ, LED, কীবোর্ড ইত্যাদি) সরাসরি বুঝতে পারে না।  
তাই এর জন্য I/O interface দরকার — যেটা একদিকে মাইক্রোপ্রসেসরের সাথে, অন্যদিকে বাইরের হার্ডওয়্যার ডিভাইসের সাথে সংযুক্ত থাকে।

এখানে দুই ধরনের ইন্টারফেস থাকে:

1. **Input Interface** → বাইরের ডিভাইস থেকে ডেটা মাইক্রোপ্রসেসরে আনা হয় (IN instruction)
2. **Output Interface** → মাইক্রোপ্রসেসর থেকে ডেটা বাইরে পাঠানো হয় (OUT instruction)

### ● 1. Basic Input Interface

#### ◆ গঠন:

Input interface সাধারণত Three-state buffer দিয়ে তৈরি হয় (যেমন IC 74ALS244)।  
এখানে বাইরের ডিভাইস (যেমন সুইচ) buffer-এর input এ যুক্ত থাকে, আর buffer-এর output যুক্ত থাকে data bus-এর সাথে।

Switch → Buffer → Data Bus → Microprocessor

#### ◆ কাজের পদ্ধতি:

- যখন **মাইক্রোপ্রসেসর** IN instruction চালায়, তখন নির্দিষ্ট I/O port address ডিকোড হয়।
- এই ডিকোড থেকে একটি Select signal (logic 0) পাওয়া যায় যা buffer-কে “enable” করে।
- Enable হলে buffer ডেটা বাসে বাইরের ইনপুট (যেমন সুইচের অবস্থা) পাঠিয়ে দেয়।
- মাইক্রোপ্রসেসর সেই ডেটা AL (accumulator) রেজিস্টারে নিয়ে নেয়।

👉 অর্থাৎ, IN instruction চালালে সুইচগুলোর অবস্থান (ON/OFF = 1/0) মাইক্রোপ্রসেসর পড়ে নিতে পারে।

◆ **কেন Three-State Buffer লাগে?**

Buffer-এর তিনটি অবস্থা থাকে:

1. Logic 1
2. Logic 0
3. High Impedance (disconnected)

→ যখন buffer enable না থাকে, তখন সেটা high-impedance mode এ থাকে — ফলে data bus-এ কোন সমস্যা হয় না।

## ● **2. Basic Output Interface**

◆ **গঠন:**

Output interface **সাধারণত Latch বা Flip-Flop** (যেমন IC 74ALS374) দিয়ে তৈরি হয়।  
এখনে মাইক্রোপ্রসেসরের data bus ল্যাচের ইনপুটে যুক্ত থাকে, আর ল্যাচের আউটপুটে যুক্ত থাকে LED বা অন্য আউটপুট ডিভাইস।

Microprocessor → Data Bus → Latch → LEDs

◆ **কাজের পদ্ধতি:**

- যখন মাইক্রোপ্রসেসর OUT instruction চালায়, তখন AL (বা AX, EAX) রেজিস্টারের ডেটা data bus-এ আসে।
- Select signal (logic 0) পেলে latch সেই ডেটা capture করে রাখে।
- তারপর latch-এর আউটপুট (Q pins) সেই ডেটা LED বা আউটপুট ডিভাইসে পাঠায়।

👉 অর্থাৎ, OUT চালালে accumulator-এর ডেটা LED-এ দেখা যায়।

◆ **কেন Latch লাগে?**

মাইক্রোপ্রসেসর থেকে ডেটা খুব অল্প সময়ের জন্য ( $< 1 \mu\text{s}$ ) data bus-এ থাকে।  
 তাই latch সেই মানটাকে ধরে রাখে যতক্ষণ না নতুন OUT instruction আসে।  
 এভাবে LED বা অন্য ডিভাইস ডেটা ধরে রাখতে পারে।

---

## সহজভাবে মনে রাখো:

বিষয়	Input Interface	Output Interface
মূল অংশ	3-state buffer (যেমন 74ALS244)	Latch (যেমন 74ALS374)
নির্দেশনা (Instruction)	IN	OUT
ডেটা প্রবাহের দিক	Device → Microprocessor	Microprocessor → Device
ডেটা রাখা হয় কিনা	না (ডেটা সরাসরি পড়ে)	হ্যাঁ (Latch ডেটা ধরে রাখে)
উদাহরণ	Switch থেকে ইনপুট নেওয়া	LED তে আউটপুট দেখানো

---

## ছোট করে মনে রাখো:

- IN → Buffer → Read from device
  - OUT → Latch → Write to device
- 
- 

## Handshaking (হ্যান্ডশেকিং) কী?

### সংজ্ঞা:

Handshaking হল এমন একটি পদ্ধতি, যার মাধ্যমে microprocessor এবং I/O device এর মধ্যে ডেটা আদান-প্রদান সমন্বিতভাবে (synchronized) করা হয়।  
 অর্থাৎ, দুটি ডিভাইস যাতে একই গতিতে (speed matching) ডেটা পাঠানো বা গ্রহণ করতে পারে, সেটাই Handshaking-এর উদ্দেশ্য।

---

## কেন Handshaking দরকার?

সব I/O ডিভাইসের কাজের গতি মাইক্রোপ্রসেসরের মতো দ্রুত নয়।

👉 যেমন –

একটা parallel printer হয়তো প্রতি সেকেন্ডে কয়েকশো character ছাপতে পারে, কিন্তু microprocessor প্রতি সেকেন্ডে লক্ষ লক্ষ instruction সম্পাদন করতে পারে।

➡ যদি প্রসেসর একসাথে অনেক ডেটা পাঠিয়ে দেয়, তাহলে প্রিন্টার সেই ডেটা নিতে পারবে না ফলে ডেটা হারিয়ে যাবে (data loss)।

👉 তাই দরকার এমন একটা পদ্ধতি, যেখানে প্রিন্টার জানাবে সে প্রস্তুত কিনা, আর প্রসেসর অপেক্ষা করবে যতক্ষণ না প্রিন্টার প্রস্তুত হয়।  
এই প্রক্রিয়াটাকেই বলা হয় handshaking।

## ⚙️ Handshaking-এর কাজের পদ্ধতি (Step by Step)

### ১) Microprocessor ডেটা পাঠাতে চায়

→ কিন্তু আগে দেখে নেয় ডিভাইস ready কিনা (যেমন প্রিন্টার ready signal দিয়েছে কিনা)।

### ২) I/O device ready হলে

→ প্রসেসর data bus দিয়ে ডেটা পাঠায়।

### ৩) ডেটা গ্রহণের পর

→ I/O device একটি acknowledge (ACK) সিগন্যাল পাঠায়, জানায় যে ডেটা গ্রহণ সম্পন্ন।

### ৪) Microprocessor ACK পাওয়ার পর

→ পরবর্তী ডেটা পাঠায়।

এইভাবে প্রতিটি ডেটা ট্রান্সফারে request-acknowledge cycle হয়।

এটাই handshaking প্রক্রিয়া।

## 🔄 Handshaking Signals (সাধারণত ব্যবহৃত সিগন্যাল)

Signal	কাজ
STB (Strobe)	Microprocessor জানায় যে ডেটা পাঠানো হয়েছে
ACK (Acknowledge)	I/O device জানায় যে ডেটা গ্রহণ করা হয়েছে

Signal	কাজ
READY / BUSY	ডিভাইস জানায় সে নতুন ডেটা নেওয়ার জন্য প্রস্তুত কিনা

## 🧠 সহজভাবে মনে রাখো:

💻 Handshaking মানে —

“Microprocessor ও I/O device একে অপরকে signal পাঠিয়ে  
ডেটা ট্রান্সফারের গতি মিলিয়ে নেয়।”

## 💡 পরীক্ষায় লেখার মতো সংক্ষিপ্ত উত্তর:

### Definition:

Handshaking is a method of data transfer control in which the microprocessor and the I/O device exchange signals to synchronize data transfer speed.

### Explanation:

Since many I/O devices (like printers) operate much slower than the microprocessor, handshaking ensures that the processor sends or receives data only when the device is ready. This prevents data loss and maintains synchronization between them.

### Example:

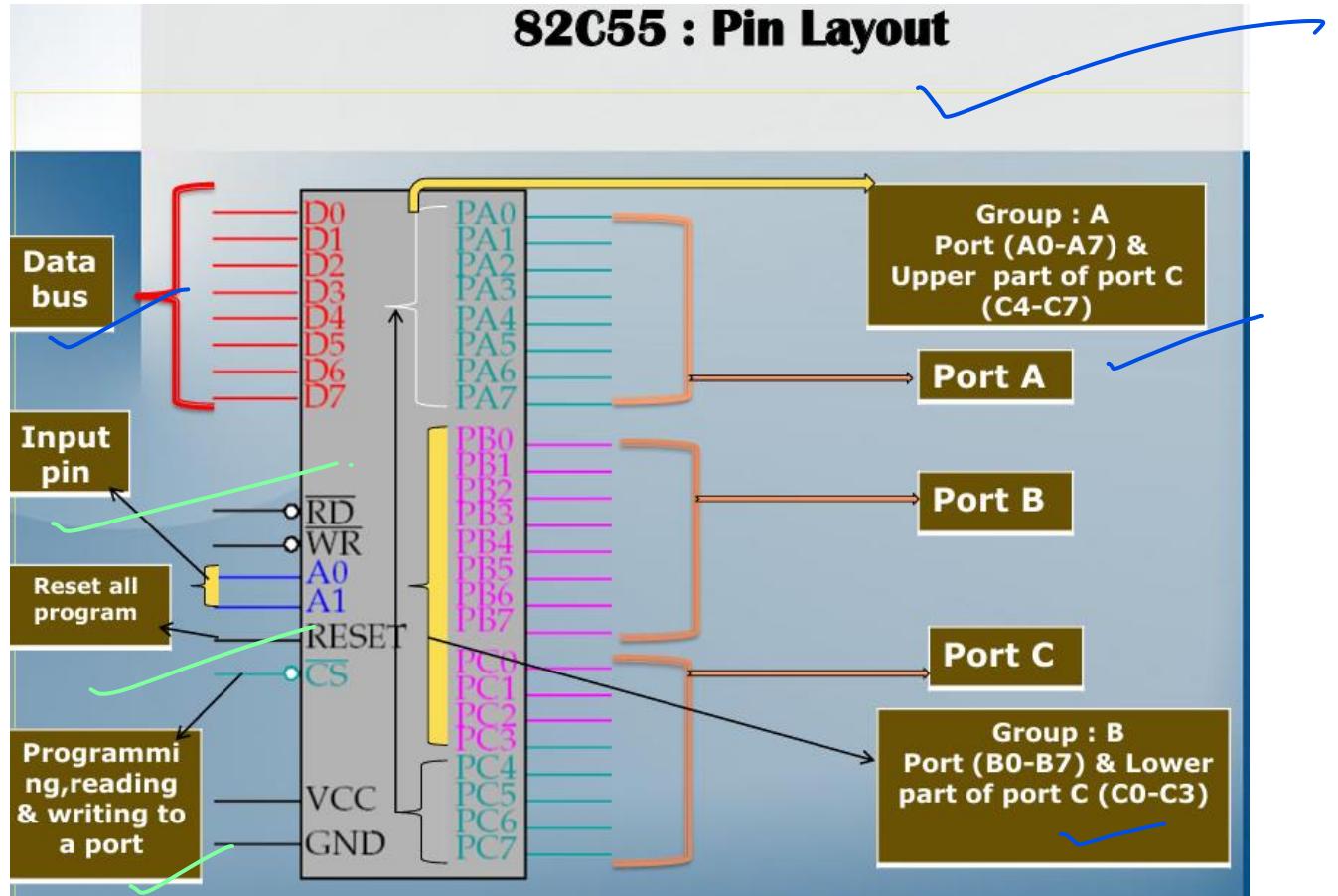
In a printer interface, the microprocessor sends data only after receiving a “ready” signal from the printer, and the printer sends an “acknowledge” signal back after accepting the data.

## ✓ 82C55 PPI – Short Description (2 Marks)

The **82C55 (Programmable Peripheral Interface)** is an **I/O device** used to interface peripherals (like keyboards, displays, printers) with a microprocessor.

It provides **24 programmable I/O lines**, divided into **three 8-bit ports**:

- **Port A** – 8-bit
- **Port B** – 8-bit
- **Port C** – 8-bit (can be split into two 4-bit ports: C upper & C lower)



#### Main Features:

- Each port can be programmed as **input or output**.
- Can operate in **three modes**:
  - **Mode 0:** Simple I/O (no handshake)
  - **Mode 1:** Handshake I/O (synchronized data transfer)
  - **Mode 2:** Bidirectional data transfer (Port A only)

#### Programming of 82C55 (2 Marks)

The 82C55 is **programmed by writing a control word to its control register**.

#### Steps to program:

1. **Select control register** using address lines (A1, A0 = 11).
2. **Send control word** from CPU to the control register.
3. Control word defines:
  - Which ports are input/output.
  - Which mode (0, 1, or 2) each group will operate in.

### Example:

```
MOV AL, 82H ; Control word (set Port A input, Port B output, etc.)
OUT 83H, AL ; Send to control register of 82C55
```

#### In short:

82C55 provides **24 I/O lines** programmable in different modes, and it is **programmed using a control word** written to its control register.

## ✳️ 82C55 কী করে?

👉 82C55 হলো একটি **programmable I/O device**,  
যা মাইক্রোপ্রসেসরের সাথে **input/output ডিভাইস** (যেমন switch, LED, printer ইত্যাদি)  
সংযোগ করতে ব্যবহৃত হয়।  
এতে **৩টা port** থাকে —

- Port A
- Port B
- Port C

এই তিনটি port কে প্রোগ্রাম করে **input বা output** হিসেবে ব্যবহার করা যায়।

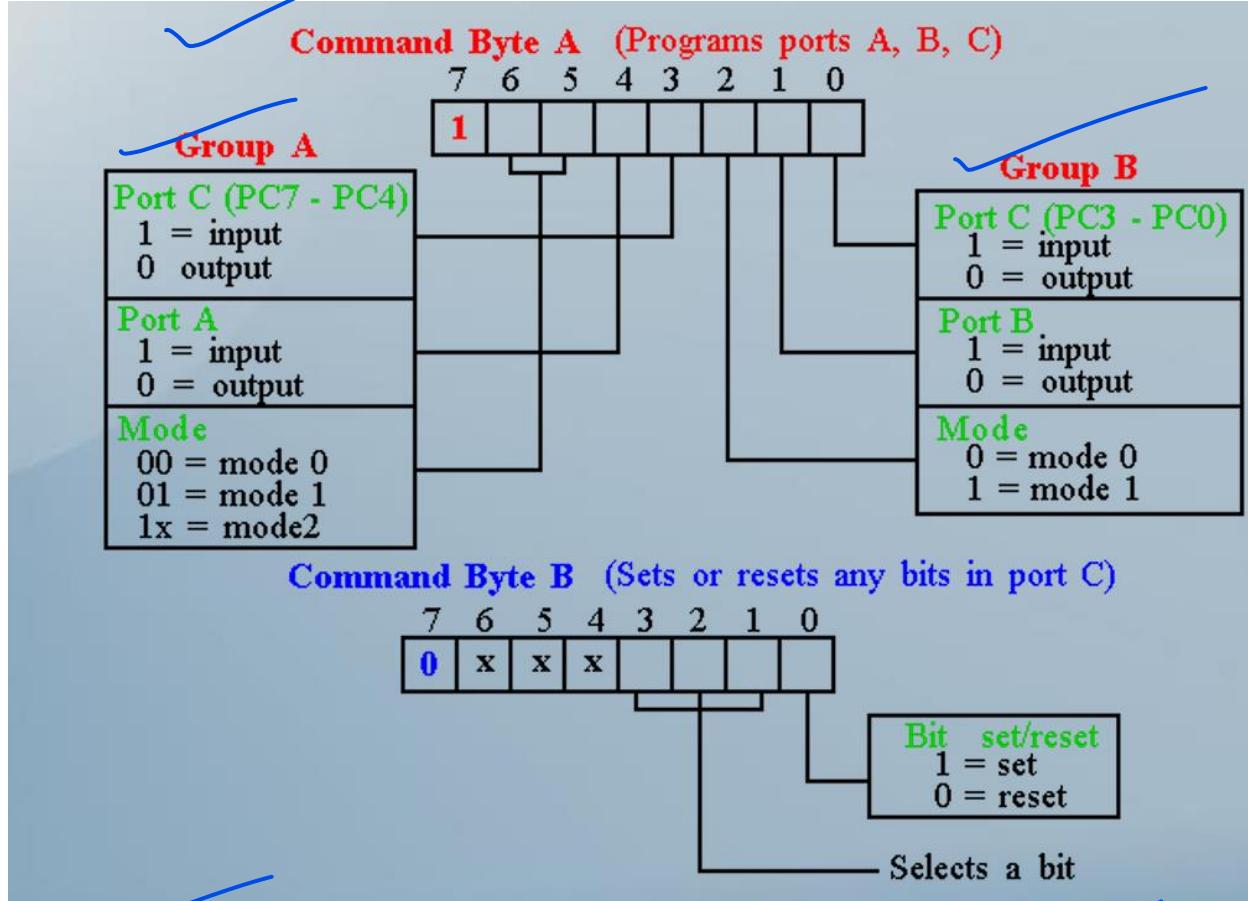
## ⚙️ Command Register (কমান্ড রেজিস্টার)

82C55-এর ভিতরে **দুটি Command Register** থাকে —  
যেগুলো দিয়ে আমরা port গুলো কীভাবে কাজ করবে (mode, input/output ইত্যাদি) তা  
নির্ধারণ করি।

### ◆ 1 Command Register এর bit 7 (MSB bit)

- Bit 7 = 1 → Command Byte A
- Bit 7 = 0 → Command Byte B

অর্থাৎ bit 7 দিয়ে নির্ধারণ হয় কোন Command Register ব্যবহার হচ্ছে।



## 2 Command Byte A

👉 এটি ব্যবহৃত হয় **Group A** এবং **Group B** এর কাজের মোড প্রোগ্রাম করার জন্য।

🧠 মনে রাখো —

82C55-এ মোট তিনটি port আছে → A, B, C

এগুলো দুইটি গ্রুপে ভাগ করা হয়:

Group	Contains	Operates in Mode
<b>Group A</b>	Port A + Upper 4 bits of Port C (PC7-PC4)	Mode 0, 1, 2
<b>Group B</b>	Port B + Lower 4 bits of Port C (PC3-PC0)	Mode 0, 1

সুতরাং, **Command Byte A** দিয়ে Group A ও Group B—দুটোই কনফিগার করা হয়।

## 3 Command Byte B

👉 এটা শুধুমাত্র Port C-এর individual bits set (1) বা reset (0) করার জন্য ব্যবহৃত হয়।  
এইটা তখনই কাজ করে, যখন 82C55 Mode 1 বা Mode 2-তে প্রোগ্রাম করা থাকে।

অর্থাৎ, Command Byte B দিয়ে তুমি Port C-এর নির্দিষ্ট pin-কে ON বা OFF করতে পারো।

## Group A এবং Group B-এর পার্থক্য:

বিষয়	Group A	Group B
পোর্টসমূহ	Port A + PC7–PC4	Port B + PC3–PC0
মোড (Mode)	Mode 0, 1, 2	Mode 0, 1
ব্যবহার	বেশি জটিল কাজ (যেমন handshaking বা bi-directional data transfer)	সহজ ইনপুট/আউটপুট কাজের জন্য
কন্ট্রোল	Group A Control bits	Group B Control bits

## সংক্ষিপ্তভাবে মনে রাখো:

- ◆ 82C55-এ ২টা Command Register থাকে
- ◆ Bit 7 দিয়ে ঠিক হয় Command A না B
- ◆ Command Byte A → Group A ও B এর মোড সেট করে
- ◆ Command Byte B → Port C-এর bits আলাদা আলাদা করে সেট/রিসেট করে (Mode 1/2 তে)
- ◆ Group A: Mode 0,1,2
- ◆ Group B: Mode 0,1

## 82C55 Operating Modes Overview

82C55 Programmable Peripheral Interface (PPI) তিনটি আলাদা operating mode এ কাজ করতে পারে:

Mode	নাম	ব্যবহারের ধরন
Mode 0	Basic Input/Output	Simple I/O operation
Mode 1	Strobed Input/Output	I/O with handshaking signals

Mode	নাম	ব্যবহারের ধরন
Mode 2	Bi-directional Bus I/O	Two-way data transfer using handshaking

## Mode 0 – Basic Input/Output Mode

◆ Main Concept:

এটি হলো সবচেয়ে সাধারণ (simple) মোড, যেখানে প্রতিটি port (A, B, C) কে input বা output হিসেবে ব্যবহার করা যায়।

◆ Features:

- প্রতিটি port কে আলাদাভাবে input বা output হিসেবে সেট করা যায়।
- No handshaking signals ব্যবহৃত হয় না।
- Data সরাসরি লিখা বা পড়া হয়।
- Output data latch হয় (মানে একবার দিলে ধরে রাখে)।
- Input data buffered থাকে।
- Port C কে আলাদা control হিসেবে ব্যবহার করা যায়।

◆ Example Use:

Switch থেকে ডেটা পড়া (input) বা LED-তে ডেটা লেখা (output)।

🧠 সংক্ষেপে:

👉 “Simple I/O mode without any control signal.”

## Mode 1 – Strobed Input/Output Mode (With Handshaking)

◆ Main Concept:

এই মোডে data transfer হয় handshaking signal (control signal) এর মাধ্যমে — যাতে microprocessor ও I/O ডিভাইসের মধ্যে proper synchronization থাকে।

◆ Features:

- Handshaking signal ব্যবহৃত হয় (Port C এর কিছু পিন control signal হিসেবে কাজ করে)।
- Port A & Port B — দুটোই Mode 1-এ কাজ করতে পারে।
- Group A = Port A + Upper 4 bits of Port C (PC7–PC4)
- Group B = Port B + Lower 4 bits of Port C (PC3–PC0)

- প্রতিটি group-এর 8-bit data port input বা output হতে পারে।
- সব input ও output data **latched** হয়।
- Port C-এর পিনগুলো status/control signal দেয় যেমন:
  - STB (Strobe)
  - IBF (Input Buffer Full)
  - OBF (Output Buffer Full)
  - ACK (Acknowledge)
  - INTR (Interrupt Request)

◆ Example Use:

Printer, keyboard, বা sensor system — যেখানে ডেটা ধীরে ধীরে আসে বা যায় এবং synchronization দরকার হয়।

🧠 সংক্ষেপে:

👉 “I/O with synchronization using control (handshaking) signals.”



## Mode 2 – Bi-directional Bus I/O Mode

◆ Main Concept:

এই মোডটি কেবল Group A-এর জন্য (Port A) ব্যবহৃত হয়।

এখানে Port A কাজ করে একটি bi-directional (দুই দিকে ডেটা যাওয়া-আসা) data bus হিসেবে।

◆ Features:

- কেবল Group A-তে ব্যবহারযোগ্য।
- Port A → 8-bit bi-directional data bus।
- Port C → 5-bit control/status port।
- Input ও Output data দুই দিকেই latched হয়।
- Data transfer control হয় Port C-এর handshaking signal দিয়ে।
- এই মোডটি মূলত high-speed communication এর জন্য ব্যবহার হয়।

◆ Example Use:

দুই মাইক্রোপ্রসেসরের মধ্যে data exchange বা full-duplex communication।

🧠 সংক্ষেপে:

👉 “Used for two-way data transfer in Group A only.”



## সারসংক্ষেপ (Exam Table Format)

বিষয়	Mode 0	Mode 1	Mode 2
নাম	Basic I/O	Strobed I/O	Bi-directional Bus I/O
ব্যবহারের ধরন	Simple input/output	Input/output with handshaking	Two-way communication
Handshaking Signal	না	হ্যাঁ	হ্যাঁ
কাজ করে যে পোর্টে	A, B, C	A ও B	শুধুমাত্র A
Control পোর্ট	Port C optional	Port C (used for handshake)	Port C (used for control/status)
Data ল্যাচিং	Output latched	Input ও output দুটোই latched	Input ও output দুটোই latched
ব্যবহারের উদাহরণ	LED, Switch	Printer, Keyboard	CPU-to-CPU data link



## Stepper Motor – Main Concept

### 1 Stepper Motor কী?

- **Digital Motor:** এটা একটি ডিজিটাল ধরনের মোটর, কারণ এটি একদম নির্দিষ্ট ধাপে ধাপে (discrete steps) ঘোরে।
- **360° rotation** পূর্ণ করার জন্য অনেকগুলো ধাপে চলে।
- এক ধাপের প্রতি নির্ভর করে motor design বা gearing-এর উপর।

### 2 Step Size

- **Low-cost motor** → 15° per step
- **High-precision motor** → 1° per step

অর্থাৎ motor যত বেশি precision, তত ছোট step করবে।

### 3 কিভাবে step তৈরি হয়?

- Step গুলো তৈরি হয় magnetic poles এবং gearing এর সাহায্যে।
- Motor-এর ভিতরে coils (ইলেকট্রোম্যাগনেট) থাকে।
- Motor armature-এ single pole থাকে।
- Step করার সময় coils-কে চালু (energize) করা হয়।

উদাহরণ:

- Two coils energized at a time → Motor step করে full step ( $45^\circ, 135^\circ, 225^\circ, 315^\circ$ )
  - One coil energized at a time → Motor step করে light/full step এর মতো কম power
- খরচে

#### 4 Full Stepping

- Motor armature একটিমাত্র step-এ 8টি discrete positions-এ দাঁড়ায়।
- এই 8টি অবস্থাকে বলা হয় Full Stepping।
- Stepper motor step-by-step ঘুরে, প্রত্যেক ধাপ এ নির্দিষ্ট angular movement করে।

#### 5 Stepper Motor + 82C55

- 82C55 দিয়ে motor control করা যায়।
- Motor-এর coils(ইলেকট্রোম্যাগনেটিক তার বা winding) কে Port A/B এর output pins-এর মাধ্যমে control করা হয়।
- Microprocessor 82C55-এর OUT instruction পাঠিয়ে stepper motor-এর প্রতিটি coil-কে switch করে next step তৈরি করে।
- এভাবে stepper motor programmed number of steps ঘুরে নির্দিষ্ট angle-এ পৌঁছে।

#### >Main Concept সংক্ষেপে:

Stepper motor হল একটি digitally controlled motor যা discrete steps ( $45^\circ, 15^\circ, 1^\circ$  ইত্যাদি) এ ঘোরে।

82C55-এর output pins দিয়ে আমরা motor-এর coils চালাই, এবং প্রতিটি step এর মাধ্যমে motor armature নির্দিষ্ট অবস্থায় পৌঁছায়।

Full stepping মানে motor armature 8টি discrete অবস্থায় ঘোরে, প্রতিটি ধাপের জন্য নির্দিষ্ট angle থাকে।

## ✓ Stepper Motor Driving Circuit (82C55 + Darlington Pair)

### 1 Motor Driving

- Stepper motor-এর coils সরাসরি microprocessor বা 82C55 থেকে চালানো যায় না, কারণ coils বেশি current চাই।
- তাই ব্যবহার করা হয় NPN Darlington amplifier pairs।
  - Darlington pair → ছোট input current দিয়ে বড় output current দিতে পারে।
  - প্রতিটি coil-কে একটি Darlington pair দিয়ে চালানো হয়।

### 2 Circuit Overview

- 82C55 PPI → microprocessor থেকে signal নেয় → OUT instruction-এর মাধ্যমে Port A/B এর output pins-এ সিগন্যল দেয়।
- এই output signal → Darlington pair → Coil চালানো হয়।
- Motor armature step-by-step ঘোরে।
- Direction (Right/Left) এবং step সংখ্যা microprocessor-এর CX register দ্বারা নির্ধারিত হয়।

### 3 Direction & Steps

- CX register → 16-bit
  - MSB (leftmost bit) → direction
    - CX > 8000h → Right-hand direction
    - CX < 8000h → Left-hand direction
  - 15-bit remaining → number of steps

অর্থাৎ, microprocessor প্রথম bit দিয়ে direction ঠিক করে, বাকি 15-bit দিয়ে কত ধাপ (steps) motor ঘুরবে তা জানায়।

### 4 Stepper Motor Control Summary

Component	কাজ
82C55	Microprocessor থেকে signals পায়, coils চালানোর জন্য drive signal দেয়।
Darlington pair	ছেট সিগনাল কে বড় current-এ amplify করে coil চালায়।
Coils	Electromagnet → armature ঘোরায়।
CX register	Motor spin direction & step count নির্ধারণ করে।

### 💡 Main Concept সংক্ষেপে:

Stepper motor 82C55 দিয়ে control করা হয়।

82C55-এর output → Darlington amp → coil → armature।

Motor-এর spin direction নির্ধারণ হয় CX-এর MSB দিয়ে, এবং step সংখ্যা 15-bit দিয়ে।

এইভাবে microprocessor program অনুযায়ী motor ঠিকভাবে step-by-step ঘোরে।

### Coprocessor কি?

Coprocessor হলো মাইক্রোপ্রসেসরের একটি বিশেষ ধরনের সহকারী চিপ বা সার্কিট।

এর কাজ হলো মাইক্রোপ্রসেসরের কাজের বাইরে বা অতিরিক্ত কিছু বিশেষ ধরনের গণনা বা কাজ দ্রুত করা।

### মূল দিকগুলো:

1. মাইক্রোপ্রসেসরের ক্ষমতা বৃদ্ধি করে।
2. সিস্টেমের প্রসেসিং স্পিড বাড়ায়।
3. সাধারণ 32-bit প্রসেসরের সাথে অনেক ব্যবহৃত হয়, যেমন: Intel 80386, Motorola 68020।

### কেন Coprocessor ব্যবহার করা হয়?

বেসিক মাইক্রোপ্রসেসর (যেমন 8085, 8086) সব ধরনের জটিল গণনা বা বিশেষ ফাংশন করতে পারে না।

### উদাহরণ:

- Floating-point arithmetic (দশমিক সংখ্যা নিয়ে হিসাব)
- Graphics প্রক্রিয়াজাতকরণ
- Signal processing (সিগন্যাল প্রক্রিয়াজাতকরণ)
- String processing (স্ট্রিং নিয়ে কাজ)
- Square root, sine, cosine, logarithm-এর মতো ফাংশন

**সমস্যা:** সাধারণ প্রসেসর এসব কাজ সরাসরি করতে পারে না।

**সমাধান:** Coprocessor ব্যবহার করা হয়, যা এই কাজগুলো দ্রুত এবং সঠিকভাবে করতে পারে।

### Coprocessor Peripheral কেন?

- Coprocessor মেইন প্রসেসরের বাইরে থাকে এবং মেইন প্রসেসরের instruction অনুসারে কাজ করে।
- এটি মেইন প্রসেসরের সাথে BUS বা ইন্টারফেসের মাধ্যমে যোগাযোগ করে।
- অর্থাৎ, Coprocessor একটি external device যা মেইন প্রসেসরের ক্ষমতা বৃদ্ধি করে।
- যেহেতু peripheral হল “মেইন প্রসেসরের সাথে যুক্ত কোনো সহায়ক ডিভাইস”, তাই Coprocessor একটি বিশেষ ধরনের peripheral।

### Coprocessor কিভাবে মেইন প্রসেসরের সাথে সংযুক্ত হয়?

Coprocessor মেইন প্রসেসরের সাথে মূলত দুইভাবে সংযুক্ত হয়:

#### ১. Intelligent Monitor Interface

- মেইন প্রসেসর BUS-এর মাধ্যমে যে ইনস্ট্রাকশন পাঠায়, Coprocessor তা মনিটর করে।
- যদি কোনো কাজ মেইন প্রসেসর করতে না পারে, তখন Coprocessor সেটি করে।

#### ২. Special Signal Interface

- মেইন প্রসেসর যখন কোনো ইনস্ট্রাকশন এক্সিকিউট করে, তখন সে Coprocessor-এর কম্যান্ড রেজিস্টারে সেই কাজের কম্যান্ড লেখে।
- Coprocessor সেই কম্যান্ড অনুযায়ী কাজ সম্পন্ন করে এবং ফলাফল মেইন প্রসেসরকে দেয়।

## সারসংক্ষেপে

- **Coprocessor** = মাইক্রোপ্রসেসরের সহকারী চিপ
  - **কারণ** = জটিল বা বিশেষ কাজ দ্রুত করতে
  - **কাজের ধরন** = Floating-point হিসাব, গাণিতিক ফাংশন, গ্রাফিক্স, সিগন্যাল প্রক্রিয়াজাতকরণ
  - **মেইন প্রসেসরের সাথে সংযোগ** = Intelligent monitor বা Special signal interface
- 

## Coprocessor

1. **মূল কাজ:**
    - এটি মূল প্রসেসরের সাথে যোগাযোগ করতে পারে এবং কম্পিউটেশনাল কাজ করতে পারে।
    - উদাহরণ: Floating-point হিসাব, সাইন/কোসাইন, লগারিদম, গ্রাফিক্স প্রক্রিয়াজাতকরণ ইত্যাদি।
  2. **ফিচার:**
    - অতিরিক্ত ইনস্ট্রুকশন, রেজিস্টার, ডেটা টাইপ যোগ করে, যা মেইন প্রসেসর সরাসরি সমর্থন করে না।
    - ব্যবহারকারীর জন্য ক্ষমতা বৃদ্ধি করে (performance বাড়ায়)।
  3. **প্রক্রিয়া:**
    - মূল প্রসেসরের কমান্ড অনুযায়ী কাজ করে।
    - ফলাফল মেইন প্রসেসরে পাঠায়।
- 

## Peripherals (পেরিফেরাল ডিভাইস)

1. **মূল কাজ:**
    - এগুলো মূলত I/O (Input/Output) অপারেশন এর জন্য ব্যবহৃত হয়।
    - উদাহরণ: কীবোর্ড, মাউস, প্রিন্টার, ডিস্ক ড্রাইভ ইত্যাদি।
  2. **ফিচার:**
    - এটি কম্পিউটেশনাল কাজ করতে পারে না।
    - শুধু ইন্টারফেসের মাধ্যমে মেইন প্রসেসরের সাথে তথ্য আদান-প্রদান করে।
- 

## সারসংক্ষেপে তুলনা

বিষয়	Coprocessor	Peripherals
কাজের ধরন	জটিল গণনা, বিশেষ ফাংশন	শুধুমাত্র I/O অপারেশন
কম্পিউটেশন ক্ষমতা	আছে	নেই
মেইন প্রসেসরের সাথে সংযোগ	BUS বা স্পেশাল ইন্টারফেস	ইন্টারফেস রেজিস্ট্যান্স / পোর্ট
উদাহরণ	Math Coprocessor, Graphics Coprocessor	Keyboard, Printer, Disk

সংক্ষেপে বললে:

- **Coprocessor** = মেইন প্রসেসরের “সহকারী গণনাকারী”
- **Peripherals** = মেইন প্রসেসরের “I/O সহকারী”

### Statement:

"All coprocessors are peripherals, but all peripherals are not coprocessors."

ব্যাখ্যা:

#### 1. Coprocessor হলো একটি Peripheral:

- Coprocessor হলো একটি বিশেষ ধরনের peripheral device, কারণ এটি মেইন প্রসেসরের সাথে যোগাযোগ করে।
- অর্থাৎ, এটি একটি external বা auxiliary device যা প্রসেসরের কাজকে সাহায্য করে।
- উদাহরণ: Math Coprocessor, Graphics Coprocessor।
- তাই বলা যায়, “All coprocessors are peripherals”, কারণ সব coprocessor মেইন প্রসেসরের সাথে সংযুক্ত এবং তার কাজকে সমর্থন করে।

#### 2. সব Peripherals Coprocessor নয়:

- **কিন্তু** সব peripherals computation করতে পারে না, তারা শুধুমাত্র I/O কাজের জন্য ব্যবহৃত হয়।
- উদাহরণ: Keyboard, Printer, Disk drive।
- তাই বলা যায়, “but all peripherals are not coprocessors”, কারণ প্রতিটি peripheral computation বা special instruction দিতে সক্ষম নয়।

## ১. Coprocessor-এর ধরন (Types of Coprocessor)

8086 পরিবারে প্রতিটি মাইক্রোপ্রসেসরের সাথে **একটি compatible coprocessor** থাকে।

প্রধান ধরনের Coprocessor:

1. **Math Coprocessor** – সাধারণ গাণিতিক কাজের জন্য
2. **Numeric Coprocessor** – সংখ্যা সংক্রান্ত বিশেষ কাজের জন্য
3. **Numeric Data Coprocessor** – extended numeric data support
4. **Floating Point Unit (FPU) Coprocessor** – floating-point arithmetic এবং advanced mathematical operations-এর জন্য

## ২. Compatible Processor এবং Coprocessor

Processor	Compatible Coprocessor
8086 & 8088	8087
80286	80287, 80287XL
80386DX	80387DX
80386SX	80387SX
80486DX	Inbuilt FPU
80486SX	80487SX

## ৩. Main Processor এবং Coprocessor Interfacing

1. **Address-Data Bus:**
  - Multiplexed address-data bus সরাসরি main processor থেকে coprocessor-এ যায়।
2. **Status Lines:**
  - Status এবং queue status lines সরাসরি সংযুক্ত থাকে।
3. **BUSY Signal:**
  - Coprocessor-এর BUSY signal মেইন প্রসেসরের TEST pin-এ সংযুক্ত থাকে।

## ৪. 8087 Coprocessor Architecture

### 8087 Coprocessor-এর মূল অংশ:

#### 1. Control Unit (CU):

- Microprocessor system bus-এর সাথে interface তৈরি করে।
- Coprocessor এবং Microprocessor-এর operations sync করে।

#### 2. Execution Unit (EU):

- সব Coprocessor instructions execute করে।
- 8-register stack আছে, প্রতিটি register 80-bit।
- Stack operands এবং results বাধে।

### 8087-এর ফিচার:

- 68 numeric processing instructions যোগ করে।
- 80-bit register stack প্রদান করে।
- 7 data types support করে: 16, 32, 64-bit integer; 32, 64, 80-bit floating point; 18-digit BCD operands।
- Arithmetic, trigonometric, exponential এবং algorithmic instructions support করে।

---

## ৫. 80287 Coprocessor

- 8087-এর enhanced version, 80286 প্রসেসরের জন্য।
- HMOS technology তে design করা।
- Floating point, extended integer এবং BCD support।
- Reset হওয়ার পর real address mode-এ কাজ করে।
- Protected address mode-এ switch করা যায়, কিন্তু reset ছাড়া real mode-এ ফিরে আসা যায় না।

---

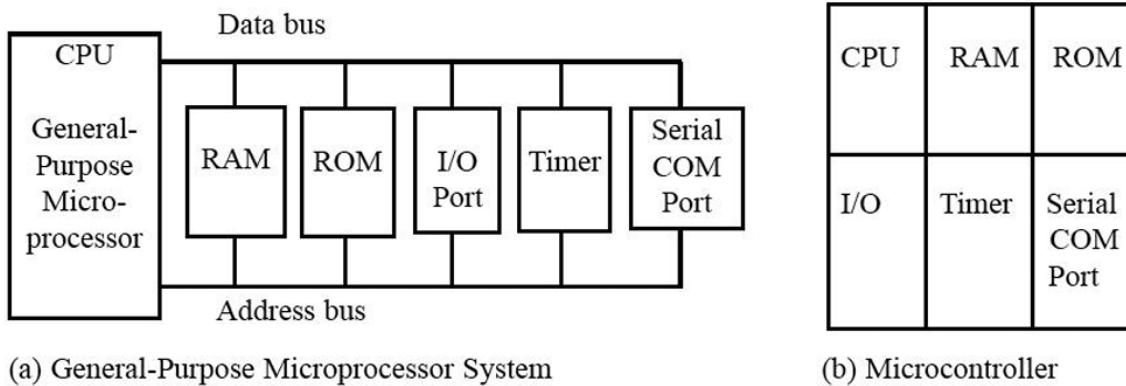
## ৬. 80387 Coprocessor

- 80386 প্রসেসরের জন্য numeric coprocessor।
- Floating-point, extended integer, এবং BCD data types support।
- 32, 64, 80-bit floating point; 64-bit integer; 18-digit BCD operands।
- তিনটি functional unit আছে, যা parallel কাজ করতে পারে।
- 80386 এবং 80387-এর মধ্যে সব communication software-এর জন্য transparent।

## সারসংক্ষেপ:

- Coprocessor হলো মেইন প্রসেসরের সহকারী, যা advanced computation ও floating point calculations করে।
  - প্রতিটি প্রসেসরের জন্য compatible coprocessor থাকে।
  - 8087, 80287, 80387 প্রভৃতি coprocessor প্রাথমিকভাবে numeric এবং floating-point operations-এর জন্য ব্যবহৃত।

- Microcontroller versus general-purpose microprocessor



## **Figure 1–1** Microprocessor System Contrasted With Microcontroller System

## • Microcontrollers for embedded systems

**Table 1–1**  
Some Embedded Products  
Using Microcontrollers

<b>Home</b>	<b>Office</b>
Appliances	Telephones
Intercom	Computers
Telephones	Security systems
Security systems	Fax machine
Garage door openers	Microwave
Answering machines	Copier
Fax machines	Laser printer
Home computers	Color printer
TVs	Paging
Cable TV tuner	<b>Auto</b>
VCR	Trip computer
Camcorder	Engine control
Remote controls	Air bag
Video games	ABS
Cellular phones	Instrumentation
Musical instruments	Security system
Sewing machines	Transmission control
Lighting control	Entertainment
Paging	Climate control
Camera	Cellular phone
Pinball machines	Keyless entry
Toys	
Exercise equipment	

Topic	Microprocessor ( $\mu$ P)	Microcontroller ( $\mu$ C)
Definition	A CPU on a single chip — needs external memory and I/O devices to work.	A complete computer system on a single chip (CPU + Memory + I/O + Timers).
Examples	Intel 8086, Pentium, Core i7	Intel 8051, PIC, Arduino (ATmega328), ARM Cortex-M
Memory	Requires external RAM/ROM.	Has built-in (internal) RAM and ROM.
I/O Ports	Must be connected through external chips.	Built-in I/O ports are available.
Working Type	Used for complex calculations, heavy tasks, and general-purpose computer design.	Performs specific control-oriented tasks; used in embedded systems.
Cost & Power	Expensive and consumes more power.	Cheaper and consumes less power.
Speed	Generally much higher (in GHz).	Comparatively lower (in MHz).
Applications	Used in PCs, laptops, and servers.	Used in washing machines, microwave ovens, cars, and IoT devices.

## ● Choosing a microcontroller

Company	Web Site
Intel	<a href="http://www.intel.com/design/mcs51">www.intel.com/design/mcs51</a>
Atmel	<a href="http://www.atmel.com">www.atmel.com</a>
Philips/Signetics	<a href="http://www.semiconductors.philips.com">www.semiconductors.philips.com</a>
Infineon	<a href="http://www.infineon.com">www.infineon.com</a>
Dallas Semi/Maxim	<a href="http://www.maxim-ic.com">www.maxim-ic.com</a>
Microchip	<a href="http://www.microchip.com/design-centers/microc">http://www.microchip.com/design-centers/microc</a>

Feature	8051	8052	8031
ROM (on-chip program space in bytes)	4K	8K	0K
RAM (bytes)	128	256	128
Timers	2	3	2
I/O pins	32	32	32
Serial port	1	1	1
Interrupt sources	6	8	6

**Table 1–4** Comparison of 8051 Family Members

### Features of 8051 Microcontroller (1 Mark)

- 8-bit CPU (The 8051 is an 8-bit processor, meaning that the CPU can work on only 8 bits of data at a time. Data larger than 8 bits has to be broken into 8-bit pieces to be processed by the CPU)
- 4 KB on-chip ROM
- 128 bytes on-chip RAM
- 32 I/O pins (4 ports × 8 bits)
- Two 16-bit timers/counters
- Full duplex serial communication
- 64 KB external memory support
- 6 interrupt sources

## OVERVIEW OF THE 8051 FAMILY

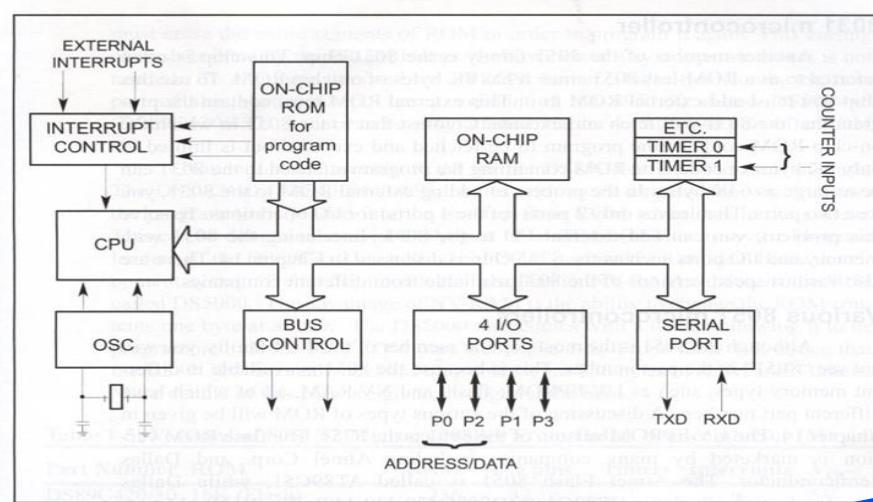


Figure 1-2. Inside the 8051 Microcontroller Block Diagram