**Computer Architecture For System design -**

আমরা যখন System Design শেখার চেষ্টা করি, তখন দেখি বিভিন্ন জায়গায় latency, caching, throughput, bottleneck এর মতো শব্দ আসে। কিন্তু এই শব্দগুলোর মূল ধারণা আসলে আসে **Computer Architecture** থেকে।

**তাই আজকের এই সেকশন এ আমরা শিখবো :-**

· CPU কীভাবে কাজ করে

· RAM, SSD, Disk কোথায় লাগে

· কেন Cache দরকার

· Latency কী ? Bandwidth কী ?

**আসলে কেনই বা আমাদের Computer Architecture এ দরকার : -**

System Design বলতে আমরা বুঝি – একটা ওয়েব অ্যাপ কিভাবে কাজ করে, কীভাবে সেটা লাখো ইউজারকে সার্ভ করে।  
এইটা বুঝতে গেলে আমাদের জানতে হবে:

1.কত দ্রুত আমরা ডেটা আনতে পারি

2.কোথায় latency বেশি

3.কিভাবে ডেটা রাখতে হবে দ্রুত access এর জন্য

উদাহরণ: Facebook এ ফিড এক সেকেন্ডের মধ্যে আসে কিভাবে?  
কারণ তাদের → CPU + RAM + Cache + SSD – এই hardware গুলো কাজ করে efficientভাবে।

**CPU কী?**

**CPU** এর পুরো নাম হলো **Central Processing Unit**। এটি একটি কম্পিউটারের **মস্তিষ্কের মতো** — অর্থাৎ, যেকোনো ইনপুট (data/instruction) নিয়ে সেটিকে প্রসেস করে আউটপুট তৈরি করে।

**CPU কীভাবে কাজ করে?**

1. CPU প্রথমে মেমোরি থেকে **Instruction** (নির্দেশনা) আনে। এই নির্দেশনাগুলো প্রোগ্রামে লেখা থাকে এবং RAM-এ থাকে।
2. আনা ইনস্ট্রাকশনকে CPU বোঝার মতো ভাষায় **ডিকোড** করে। মানে, CPU বুঝে কী করতে হবে — যোগ করতে হবে নাকি গুণ?
3. ইনস্ট্রাকশন অনুযায়ী কাজ করে। যেমন: ২টি সংখ্যা যোগ করা, কোনো ডেটা মেমোরিতে রাখা বা অন্য কোথাও পাঠানো।

## CPU-এর মূল উপাদান

**ALU (Arithmetic Logic Unit)**  
এখানে হিসাব (যোগ, বিয়োগ) ও যুক্তি (AND, OR) কাজগুলো হয়।

**CU (Control Unit)**  
কে কখন কী করবে, সেই নির্দেশনা দেয় CU।

**Registers**  
অস্থায়ীভাবে ডেটা ধরে রাখে খুব দ্রুত কাজ করার জন্য। যেমন: instruction register, program counter।

**Core and Thread**

· **Single-core CPU** একসাথে একটি কাজ করতে পারে।

· **Multi-core CPU** একসাথে একাধিক কাজ (parallel processing) করতে পারে।

· **Thread** মানে CPU-র কাজের লাইন। একেকটি core একাধিক thread চালাতে পারে।

**RAM, SSD, Disk — কী এবং System Design-এ কিভাবে লাগে?**

**RAM (Random Access Memory) ,**

এটি কম্পিউটারের **অস্থায়ী মেমোরি**। যে ডেটা ও প্রোগ্রামগুলো CPU বর্তমানে ব্যবহার করছে, সেগুলো RAM-এ থাকে।বিদ্যুৎ বন্ধ হলে সব ডেটা মুছে যায় (volatile memory)।

#### System Design-এ কোথায় লাগে?

**Caching** — frequently accessed ডেটা RAM-এ রেখে performance বাড়ানো হয় (যেমন Redis, Memcached)

**In-Memory Processing** — ডেটাবেস বা API থেকে বারবার না এনে RAM-এ রেখে দ্রুত প্রসেস করা

**Latency-sensitive Systems** — যেমন real-time game server, recommendation engine

**SSD (Solid State Drive)**

এটি একটি **permanent storage**, কিন্তু আগের হার্ডডিস্কের চেয়ে অনেক দ্রুত।

**System Design-এ কোথায় লাগে?**

· **Fast Storage for Databases** — যেমন high-performance DB server-এ SSD ব্যবহার করা হয়

· **Startup Time Optimization** — application/server দ্রুত শুরু করা

· **Log Storage & Access** — দ্রুত লগ access করতে হলে SSD খুব কার্যকর

**Disk / HDD (Hard Disk Drive):**

#### System Design-এ কোথায় লাগে?

**Cold Storage** — পুরনো, কম ব্যবহার হয় এমন ডেটা রাখতে

**Backup Systems**

**Data Lake / Archival Systems** — বিশাল পরিমাণ ডেটা জমিয়ে রাখার জন্য

**Caching কী?**

**Caching** মানে হলো — frequently used data-কে **temporary memory (যেমন RAM)**-এ রেখে future-এ দ্রুত access করা।

Example: যেমন, তুমি যদি বারবার একি query দাও — "product details" — তাহলে বারবার database hit না করে আগে থেকে RAM-এ store করে রাখা ডেটা serve করা হয়।

**কেন Caching লাগে? --**

· ✅ **Performance Boost**

Cache থেকে ডেটা পড়া database থেকে অনেক গুণ দ্রুত

· ✅ **Latency কমায়**

User wait করে না, milliseconds-এ response পায়

· ✅ **Database Load কমায়**

Repeated query-গুলো cache থেকে provide হওয়ায় DB presssure কমে

· ✅ **Cost-efficient**

কম latency + কম infra cost = ভালো system efficiency

যদিও caching নিয়ে আমি সম্পূর্ণ আরেকটি ডকুমেন্ট বানাবো সেখানে আরো অনেক বিশদ ভাবে আলোচনা করবো যেমন , System Design-এ Caching কোথায় কোথায় হয়?কেমন কেমন Cache আছে? Real-life Example ইত্যাদি

****Latency** আর **Bandwidth** — এই দুইটা term System Design, Networking এবং Performance Optimization এ **core concept**।**

**Latency (লেটেন্সি) কী?**

**Latency** মানে হলো: **একটি ডেটা রিকোয়েস্ট পাঠানো থেকে রেসপন্স আসা পর্যন্ত সময়।**

এটি সময়ের মাপ — সাধারণত milliseconds (ms) এ মাপা হয়।

### উদাহরণ:

তুমি ব্রাউজারে গুগল খুললে, server থেকে গুগলের homepage আসতে **যত সময় লাগে**, সেটাই latency।

**Bandwidth (ব্যান্ডউইথ) কী?**

**Bandwidth** মানে হলো:

**একক সময়ে কত ডেটা আদান-প্রদান করা যায় তার পরিমাণ।**

এটি মাপা হয় **Mbps, Gbps** (Megabits/second) এ।

**System Design-এ কোথায় লাগে?**

| **Concept** | **কেন গুরুত্বপূর্ণ?** | **উদাহরণ** |
| --- | --- | --- |
| **Latency** | Real-time system, User experience | Chat app, stock trading platform |
| **Bandwidth** | High-volume data transfer | Video streaming, CDN delivery, backups |