CSE 340

Rejwan Shah
23241108
Fall 2023

Course Name : Computer Architechture.

Faculty : Sumaiya Tanjil Khan (STK)

email : ext. sumaiya.tanjil @ bracu.ac.bd.

Consultation :

- Sunday ⇒ 12:30 – 01:50 pm
  ⇒ 3:30 – 4:50 pm
- Tuesday ⇒ 9:30 – 11:00
  3:30 – 5:00
- Thursday ⇒ 11:00 – 02:00 pm.

UB 809 09

Mark Distribution :

- Attendance ⇒ 5%.
- Quizes ⇒ 15% (best 3 out of 4/5)
- Assignment ⇒ 15%
- Mid ⇒ 25%
- Final ⇒ 40%

Raw mark

- Course ft (Theory + Math) based

Midterm Syllabus :

- Chapter - 1
- Chapter - 2
- Chapter - 3

Final Syllabus :

- Chapter - 4

Chapter - 1

Computer Abstraction and Technology

The Computer Revolution:

- Progress in Computer technology.
  - Underpinned by **Moore's Law** *

→ If you have a chip of small area and we have n transistors in the chip. Then in the next 2 years, same area এর chip এ 2n transistor থাকবে (fit করতে পারব)

⇒ **The number of chips (Transistor) present in a single chip would get double in every 2 years.**

- Makes Novel (advanced) application feasible.
  - ⇒ Computer in automobiles.
  - ⇒ Cell Phones.
  - ⇒ Human Genome Project
  - ⇒ World wide web.
  - ⇒ Search Engines

- **Computers are pervasive**

⇒ ব্যাপকভাবে ব্যবহৃত হয়, বর্তমান carry করা easy.

• Different Types of Computers:

• Desktop Computer:

⇒ General purpose, variety of software.

⇒ Subject to cost / performance trade off.

⇒ One, Machine, One user at a time

} Varies on the performance you want or on your budget.

• Server Computer:

⇒ Network based

⇒ High Capacity, performance, reliablity

⇒ Range from small servers to building sized

⇒ One machine, multiple user, at a time.

• Embeded Computer:

⇒ Hidden as component of Systems.

⇒ Stringent power / performance / cost constraints.

⇒ Build for specific reason

⇒ Particular কোনো কাজের জন্য create করা।

উদাহরণ: miero-oven, washing-machine.

• Cloud Computer:

⇒ Server বিহীন ব্যবস্থা.

⇒ সেবা use করতে পারবে, example: Colab

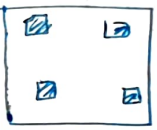
⇒ need internet.

Single Core:

- One CPU
- One Chip

Multi-Core

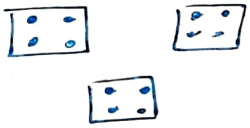
- Many CPU
- One chip



- each proc sharing main memory & peripherals.

Many-Core

- Many-Core
- Many CPU



Understanding Performance:

Performance depends on three things:

1. Algorithms.
2. Programming Language, Compiler, Architecture.
3. Processor and memory systems.
4. I/O Systems

Algorithms:

⇒ Determines numbers of operations executed

Programming Language, Compiler, Architecture:

→ Determines numbers of machine instructions executed per operation

Processor and memory Systems:

⇒ Determines how fast instructions are executed

I/O System:

⇒ Determines how fast I/O Operations are executed.

I/O মানে input / output.

The Processor Market:

- Traditional feature phone এর demand কমছে।
- Smartphone, Tablets এর demand PC থেকে বেশি.
  ⇒ Smartphone এর demand tablet থেকে বেশি।

Learning Outcomes:

- Procedure of translating programs into machine code.
  ⇒ And how hardware executes them.
- Hardware / Software interface.
- Determining factor of program performance and how to improve.
- How hardware designers improve performance.
- What's parallel processing.

"Increase CPU performance মানে decrease execution time."

~~execution / CPU time কম হলে তবে performance~~

Below your programm. ★★★

① Application Software :

⇒ Written in high level language.

② System Software.

- Compiler

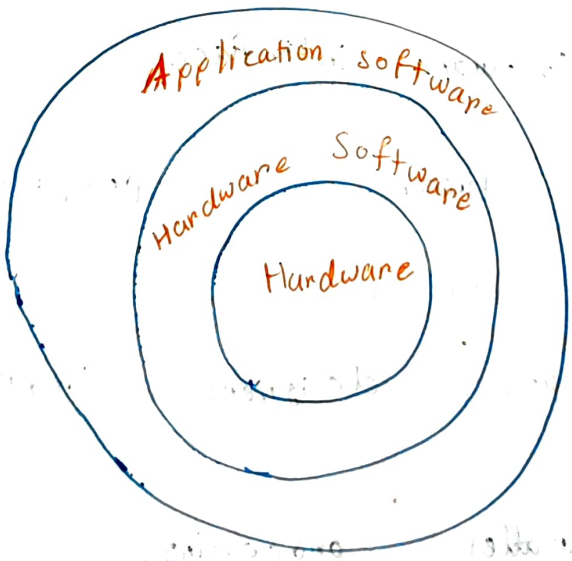
  ⇒ Translates high level language to machine code.

- Operating System

  ⇒ Manages memory and storage scheduling tasks.

③ Hardware

⇒ Processor, memory, I/O controllers.



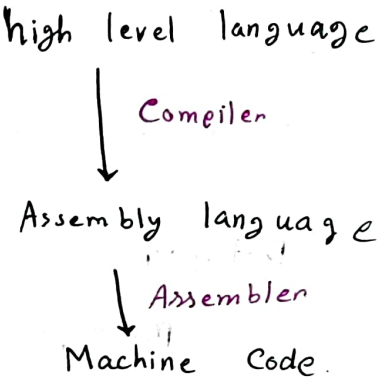
Levels of Programme Code

- **High level language:**

⇒ Almost similar to human language

⇒ Level of abstraction closer to problem domain.

⇒ Provides for productivity and portablity.



Assembly Language: (MIPS — 32 bit architecutre)

⇒ close to machine language.

⇒ Textual rep. of instructions.

```
c = a+b
add $c, $a, $b
```

Hardware representations:

⇒ Uses machine language

⇒ Binary digits (bits).

⇒ Encoded instructions and data.

Components of Computers:

- Same component for all kind of computer.

⇒ Desktop, Server embeded

- Input / Output devices

⇒ UI device.

display, keyboard, mouse.

⇒ Storage devices,

Hard disk, CD-DVD, SSD, flash

⇒ Network adapters.

Inside the Processor (CPU): ★★

- Datapath
- Control
- Cache memory

Datapath:

memory তে বিভিন্ন data store করছি; input, output থেকে, input - output থেকে data memory তে যাচ্ছে। data display করতে চাইলে তা memory থেকে আবার output device এ আসবে, কোনো instruction execute করতে চাই তাহলে তা data গুলো ALU বা অন্য কোনো unit এ যাচ্ছে, ALU computer memory থেকে instruction আনবে,

data গুলো যে এক জায়গা থেকে অন্য জায়গায় যাচ্ছে, এগুলো হচ্ছে datapath এর মাধ্যমে যাচ্ছে, প্রত্যেকটা data datapath এর মাধ্যমে connected প্রত্যেকটা unit এর সাথে, " data pass হয় data path এর মাধ্যমে।"

Performs operation on data

⇒ Combination of devices and their connecting path.
Devices communicate through this path.

Control:

⇒ Controls the sequence of operations, sequence of instruction execution

[Which instruction should go where; how it should execute, which hardware it should access]

⇒ Control Unit controls any operation.

Cache Memory:

প্রত্যেক CPU এর একটা cache memory থাকে, যখনই কোনো instruction execute হওয়া লাগে তখনই তা ram থেকে এর প্রয়োজনীয় data টা cache memory তে নিয়ে আসে। cache memory তে নিয়ে আসার পর CPU তে instruction গুলো execute হয়।

⇒ Small fast SRAM memory for immediate access of data.

⇒ Very very small, very expensive beuz they are very fast you can access data from the memory devices. That's why they are inside the processors. beuz processor operates very quickly

- The bigger the cache memory the more beneficial it is for the processor.

★ AMD Barcelona was a 4 Cores Processor

Abstractions: ★★

→ Hides the lower level detail from the user and makes the higher level detail visible to the user.

- Helps us deal with complexity

  ⇒ Hides lower-level detail.

- Instruction Set Architecture (ISA)

  ↳ The hardware / Software interface.

  ↳ instruction set for a particular architecture.

  ↳ allows high level programmes to be translated into assembly instruction and then translates into machine instruction

- Application binary Interface (ABI)

  → The ISA plus system software interface

  → Allows the system to identify where the data is going to be stored in the registers; how much data will be transferred

• Implementation.

⇒ The details underlying and interface

Definitions:

Computer architecture: ★★★★★

এর definition ৩ টা জিনিস নিয়ে চলবে.

- hardware.
- Software.
- hardware, software এমনভাবে create করতে হবে যাতে specific goal গুলো meet করে.

design করতে হবে এবং এমন hardware select করতে হবে, একসাথে interconnect করতে হবে এবং সাথে সাথে তার সাথে compatible software interface ও create (hardware interface) করতে হবে, যেন আমাদের computer computing system টা meets functional (যেমন feature রাখতে চাই ওগুলো যাতে functional থাকে), performance (যেন high হয়), energy consumption (কম হয়), cost কম হয় এবং other specific goal meet করে.

Abstract interface দিয়ে hardware and software connect থাকে

Same কাজ দুই compiler কম execution এ করতে পারে তত ভালো.

ISA: ***

⇒ An abstract interface between hardware and software (lowest level) of a machine.

⇒ That encompasses all the necessary informations.

⇒ To write a machine language program.

⇒ That will run correctly, including instructions, registers, memory access, I/O and so on.

⇒ assembly language হচ্ছে instruction set নির্ভর high level language যেমন এর hardware level এ নিয়ন্ত্রণ করার জন্য ISA.

⇒ বিভিন্ন architecture এ যে কয়টা instruction থাকে.

Datapath: **

⇒ datapath is a set of ~~computer architecture~~ functional unit (ALU, Multiplier) that carries out data processing.

Central part of many CPU.

⇒ datapath along with control unit, make up of the CPU, that largely regulates interaction between the datapath and data itself.

⇒ Datapaths are usually stored in registers or memory.

We can create large datapath by joining more than one datapath.

⇒ CPU এর মধ্যের বিভিন্ন component এর মধ্যে data আদান প্রদান করার জন্য যে path টা থাকে তাই datapath.

datapath CPU এর প্রত্যেকটা component এর মাঝে connected

Memory Hierarchy:

⇒ Used in computer architecture while discussing performance issues in computer architectural design; algorithm prediction and lower level programming construct.

⇒ memory hierarchy করার সময় response time কে parameter হিসেবে ধরা হয়।

⇒ "response time যত কম memory তত efficient".

⇒ সবচেয়ে efficient memory কে hierarchy তে উপরে রাখি.

⇒ register and cache সবচেয়ে উপরে;

আমাদের CPU যতবার memory কে access করতে যতটুকু time লাগে তা response time.

- reg
- cache
- ram
- USB, Flash
- Harddrive.
- Tape backup

efficieny বাড়ে
cpu time কমে
cost, size ও বাড়ে,

যেসব memory তে access করতে সবচেয়ে কমসময় লাগে তাদের size এ তুলনামূলক ছোট হয়, Cache তুলনামূলক expensive. Cache তুলনামূলক better.

main target হলো cpu efficiency বাড়ানো,

Multiprocessor এর সুবিধা:

- Simultaneously কাজ করতে পারি,
- more efficient then single core processor.
- improves computing speed, performance and cost-effectiveness
- Multi processor:
  - tightly coupled computer system having two or more processors each sharing main memory and peripherals

instruction তথ্যগুলো এমনভাবে processor কে যায়

কারণ হচ্ছে, হয় এটা এমনভাবে করে যাতে যাতে একটা chunk এর বিভিন্ন information অন্য chunk এর উপর dependent না হয়

Role of Computer Architect:

- look backward
  - Analyze and evaluate the past.
- look forward.
  - Be the dreamer, be creative (কিভাবে design করবো)
- look up:
  - Understand important problem and their nature.
- look down:
  - Understanding the capabilities of underlying technology.
  - future technology; এর future এর কথা চিন্তা করে design করছে হবে. সাধারণত,

execution time কত কম cpu performance তত ভালো।

যদি faster processor দিয়ে old processor replace করি বা multiple processor add করি তাহলে, response time কমবে এবং throughput বাড়বে, which is a good thing.

Response time:

⇒ একটা task complete হতে যতটুকু time লাগে।

Throughput:

Total work done per unit time.

CPU time এবং execution time same.

- Performance, $P = \frac{1}{ET}$ ← Execution time.

- $\frac{P_A}{P_B} = \frac{E_B}{E_A} = \frac{\text{CPU time}_B}{\text{CPU time}_A}$

- Time period, $T = \frac{1}{f}$ ← frequency

- Frequency, $f = \frac{1}{T}$

Time taken to run a program 10 s on A, 15 s on B. Which processor is fast-

যার value ছোট তার nominator এ হারর.

$$\Rightarrow \frac{E_B}{E_A} = \frac{15}{5} = 1.5.$$

∴ A is 1.5 times faster than B

Elapsed time is the total response time.

∴ Elapsed time = Response time.

cpu time.

↳ Total time spent on a given job.

একটা Cycle (Clock) Complete হতে যে time লাগে তা Time (Clock) period; T.

↳ duration of a clock cycle.

Clock frequency → Cycles per second

↳ $f = \frac{1}{T}$

Formula:

$$\text{CPU time} = \text{CPU Clock Cycles} \times \text{Clock Cycle time}$$

$$= \frac{\text{CPU Clock Cycles}}{\text{Clock rate}}$$

Clock Cycle time $= \frac{1}{\text{Clock rate}}$

- Performance can be improved by,
  - Reducing number of clock cycles.
  - Increasing Clock rate (decreasing clock cycle time)
  - Must often trade off clock rate against cycle count.

Computer A: 2GHz clock, 10s CPU time.

Designing Computer B.

- Aim for 6s CPU time.
- Causing computer B to require 1.2 times as many clock cycles as computer A for this program.
- Can do faster clock, but causes 1.2× Clock Cycles

How fast Computer B's Clock would be?

(Clock rate 572 62)

$$\Rightarrow \text{Clock rate}_B = \frac{\text{number of Clock Cycles}_B}{\text{CPU time}_B}$$

$$= \frac{1.2 * \text{number of Clock cycles}_A}{6}$$

$$\text{Clock cycle}_A = \text{CPU time}_A \times \text{clock rate}_A$$

$$= 10\,s \times 2\,GHz$$

$$= 20 \times 10^9$$

$$\therefore \text{Clock rate}_B = \frac{1.2 * 20 \times 10^9}{6} = 4\,GHz$$

"CPU time যত কম তত better"

CPI ⇒ Clock cycle per instruction

↳ একটা instruction execute করতে যতটা Clock cycle লাগে.

"ISA same মানে instruction count same."

Clock cycles = instruction count × CPI

∴ CPU = Clock Cycles * Clock cycle time

= instruction count ∝ CPI ∝ clock cycle time

$$= \frac{\text{instruction count} \times \text{CPI}}{\text{clock rate}}$$

Computer A: Cycle time = 250 ps , CPI = 2.0

Computer B: Cycle time = 500 ps x CPI = 1.2

Same ISA.

Which is faster and by how much?

$\Rightarrow$ CPU time$_A$ = instruction count x CPI$_A$ x Clock cycle time$_A$

$= 1 \times 2.0 \times 250 \times 10^{-12}$ s

$= 500 \times 10^{-12}$ s

CPU time$_B$ = instruction count x CPI$_B$ x Clock cycle time$_B$

$= 1 \times 1.2 \times 500 \times 10^{-12}$

$= 600 \times 10^{-12}$ s

$\therefore \frac{\text{CPU time}_B}{\text{CPU time}_A} = \frac{600 \times 10^{-12}}{500 \times 10^{-12}} = 1.2$

$\therefore$ CPU A is 1.2 times faster than B.

যদি different instruction class এর জন্য দেয়া থাকে,

$$\text{Clock Cycles} = \sum_{i=1}^{n} (CPI_i \times \text{instruction count}_i)$$

Weighted average CPI,

$$CPI = \frac{\text{Clock Cycles}}{\text{instruction count}} = \sum_{i=1}^{n} \left(CPI_i \times \underbrace{\frac{\text{instruction count}_i}{\text{instruction count}}}_{\text{Relative frequency}}\right)$$

☑ Alternative compiled code sequences using instructions in classes A, B, C. Find out Clock cycles and Avg. CPI.

Class	A	B	C
CPI for class	1	2	3
IC in sequence 1	2	1	2
IC in sequence 2	4	1	1

⇒ Sequence – 1:

Clock cycles = (1×2) + (2×1) + (3×2)

= 10

∴ Avg CPI = $\frac{10}{5}$ = 2.0

Sequence – 2

Clock Cycles = (4×1) + (1×2) + (3×1)

= 9.

∴ Avg CPI = 9/6 = 1.5

- Performance depends on:
  - ⇒ Algorithm: affects IC, Possibly CPI.
  - ⇒ Programming language: affects CPI, IC.
  - ⇒ Compiler: Affects CPI, Ie
  - ⇒ ISA: affects, IC, CPI, T_c

$$\text{Power} = \underbrace{\text{Capacitive load}}_{\text{Transistor এর উপর Capacitive load নির্ভর}} \times \text{Voltage}^2 \times \text{Frequency}$$

~~CPU time~~ Power বাড়লে, CPU time কমবে; which is good.

Capacitive load কমানো better.

CPU তে multiprocessor থাকলে better; as, একই time এ processor switch করে করে simultaneously কাজ করতে পারে;

multiprocessor এ CPU time কমে যায়, Processing time বেড়ে যায়.

Suppose a new CPU has, 85% of ~~capabilities~~ capacitive load of old CPU and 15% voltage and 15% frequency reduction?

???

$$\Rightarrow \frac{P_{new}}{P_{old}} = \frac{C_{old} \times 85\% \times (V_{old} \times 85\%)^2 \times (F \times 85\%)}{C_{old} \times V_{old}^2 \times F_{old}}$$

$$= (0.85)^4 = 0.52.$$

∴ আমরা যে মোট 0.52 এর power reduce করতে পারছি।

(P.T.O)

RISC

- Reduced Instruction Set Computer
- Instruction execute in only one clock cycle
- Variable instruction size
  - ಇಲ್ಲಿ RISC ನಲ್ಲಿ
  - ಇಲ್ಲಿ CISC ನಲ್ಲಿ
- Requires Pipelining
- Hardwire Controlled.
- Complexity pushed to the compiler.
- Instruction set is orthogonal (little overlapping of instruction functionality)
- Instruction interface with memory via fixed mechanism

CISC

- Complex Instruction Set Computer.
- Instruction take more than one clock cycle to execute.
- Fixed instruction format and uniformed length instruction.
- No pipelining.
- Micro code Controlled.
- Work well with simpler compiler
- Upward compatability within a family.
- Instruction interface with memory in multiple ways mechanisms with complex addressing modes.

MIPS architecture RISC ನಲ್ಲಿ under ನಲ್ಲಿ.

For a specific machine, total instruction count = $2 * 10^9$, Avg CPI = 3 cycles / instruction, Clock rate = 100 MHz. We want to improve the compiler to have instruction count = 10^9, new CPI = 4 cycles / instructions. clock rate = 250 MHz. By what percentage the is the new compiler improved?

⇒ Initially,

$$\text{CPU time} = \frac{ic * CPI}{\text{clock rate}} = \frac{2 * 10^{-9} \times 3}{100 \times 10^6}$$

$$= \cancel{6000 \times 10^{-20}}\ 60$$

After,

$$\text{CPU time} = \frac{10^9 \times 4}{250 \times 10^6} = 16$$

$$\therefore \%\ \text{of improvement} = \left(\frac{\text{CPU time for comp}_A - \text{CPU time for comp B}}{\text{CPU time for comp}_A} \right) \times 100$$

$$= 73.33\%$$

Assume, a new CPU has 6% less capacitive load compared to an existing CPU and has an 2% reduction in voltage ~~and frequency~~. Now calculate the reduction in overall power use.

$$\Rightarrow \frac{P_{new}}{P_{old}} = \frac{(0.94 \times \text{Capacitive load}) \times (0.98 \times \text{Voltage})^2 \times \text{Frequency}}{\text{Capacitive load} \times \text{Voltage}^2 \times \text{frequency}}$$

$$= 0.902.$$

P_{new}, P_{old} [illegible] এর মধ্যে 0.902 times improve হচ্ছে, বা power reduce হচ্ছে,

$\therefore$ Reduction in power = $(1 - 0.902) \times 100\%$

$= 9.8\%$

Spec Ratio:

Reference time এবং current execution time এর ratio কে বলা হয় spec ratio.

$$\text{Spec Ratio} = \frac{\text{Reference Time}}{\text{Execution Time}}$$

Geometric mean $= \sqrt[n]{\prod_{i=1}^{n} \text{Spec ratio}_i}$

$\therefore \sqrt[2]{SR_1 \times SR_2}$

$\sqrt[3]{SR_1 \times SR_2 \times SR_3}$

Geometric Mean

Multiple processor থাকলে প্রত্যেকটা processor এর আলাদা আলাদা spec ratio থাকবে,

execution time ratio আর Spec ratio same.

Given 2 programs,

Program A, instruction count = 10, CPI = 3, clock Cycle = 3s,

Reference time = 500 s.

Program B, instruction count = 50, CPI = 7, Clock Cycle = 3s

Reference time = 2000 s.

Find spec ratio for each program individually ? also find the geometric mean.

⇒ Program A

CPU time = IC * CPI x Clock cycle

= 10 x 3 x 3

= 90s

∴ Spec ratio = $\frac{500}{90}$

= 5.556

Program B

CPU time = Ic x CPI x Clock cycle

= 50 x 7 x 3

= 1050 s

spec ratio = $\frac{2000}{1050}$

= 1.9047.

$\therefore$ Geometric mean $= \sqrt[2]{\text{spec ratio}_A \times \text{spec ratio}_B}$

$= \sqrt[2]{5.556 \times 1.9047}$

$= 3.2537.$

Amdahl's Law:

⇒ improving an aspect of a computer and expecting a proportional improvement in overall performance.

"মূলত্ব মাত্র কোনো একটা specific feature কে যদি improve করি তাহলে overall program টার কতটুকু improvement হবে তা Amdhal's law দ্বারা calculate করি।"

$$T_{improved} = \frac{T_{affected}}{\text{improving factor}} + T_{unaffected}$$

(active / affected time; Total time)

- multiply accounts for [80s / 100s].

How much improvement in multiply performance to get 5x overall?

$$\Rightarrow \frac{100}{5} = \frac{80}{n} + 20$$

$$\Rightarrow 20 = \frac{80}{n} + 20$$

⇒ can't be done, → আর এই program টা 5x improve করা impossible

> by the way, improvement factor n এর value কখনো negative, ∞, বা 0 হওয়া যাবে না।

→ এই 100s execution time এর ভিতর 80s হল কাজ করছে বাকি 20 second idle বসে আছে।

instead of 5x improvement যদি 2x improvement হতো,

$$\frac{100}{2} = \frac{80}{n} + 20$$

$$\Rightarrow 50 = \frac{80}{n} + 20$$

$$\Rightarrow \frac{80}{n} = 30$$

$$\Rightarrow n = 2.67.$$

∴ improvement factor = 2.67.

☑ Now assume that, a particular operation takes 2.5x % of the total execution time. What improvement is required if we want 2.5 times speedup in that operation, where x is equal to 10.

$$\Rightarrow \frac{100}{2.5} = \frac{(100 \times 2.5 \times 10\%)}{n} + 75$$

$$\Rightarrow 40 = \frac{25}{n} + 75$$

$$\Rightarrow n = -0.7142$$

Value negative আসলে তার improve সম্ভব না,

∴ Since the improvement factor is negative, we can't bring this improvement.

☑ Suppose you are training a face recognition model, which is heavily dependent on a process (80%). So, you installed a graphics card with to speed up that process. Now, you observe that it is taking only 3 days to execute., as opposed to 6 days before installing the card. What is the improvement?

⇒ improved time = 3 days,

Affected time = $6 \times 0.8 = 4.8$

Unaffected time = $6 - 4.8 = 1.2$,

⇒ $\therefore 3 = \frac{4.8}{n} + 1.2$

⇒ $1.8n = 4.8$

⇒ $n = 2.67$.

percentage দিয়ে করলে,

$100 = 80 + 20$

⇒ $\frac{100}{2} = \frac{80}{n} + 20$

⇒ $n = 2.67$.

MIPS as a performance Metric:

- MIPS ⇒ Millions of instructions per second
- 1 MIPS মানে 1 second এ 1 million instruction execute হচ্ছে।
- যত MIPS মানে ভালো Unit.
- doesn't account for
  - different ISA
  - difference in complexity between instructions.

$$MIPS = \frac{\text{instruction count}}{\text{Execution time} \times 10^6}$$

$$= \frac{\text{instruction count}}{\frac{\text{instruction count} \times CPI \times 10^6}{\text{clock rate}}}$$

$$= \frac{\text{clock rate}}{CPI \times 10^6}$$

The Von Neumann Model ★★★★

- Hardware based architecture.
- "Data memory এবং instruction memory same memory তে রাখা। As a result, একসাথে একই time এ simultaneously data এবং instruction access করতে পারি না।"
- Also called "stored program computer" (instruction in memory).
  - Sequencially programme run করে।

- Stored program:
  - ⇒ instruction stored in a linear memory array
  - ⇒ Memory is unified between instruction and data

  The interpolation of a stored value depends on the controlling signals

- Sequential instruction processing.
  - ⇒ One instruction ~~fetched~~ processed at a time.
  - ⇒ Program counter (PC) identifies the current instruction that is being executed
  - ⇒ PC is advanced sequentially except for control transfer instruction.

if else condition ও সবসময় check হয়.

```
1. if :
2.      add
3.      z

4. else:
5.      multiplication
```

if condition satisfy হলে 1 থেকে 2 then 3 হবে .....

কিন্তু, if condition satisfy না হলে তাহলে 4 এ else এ চলে আসবে।

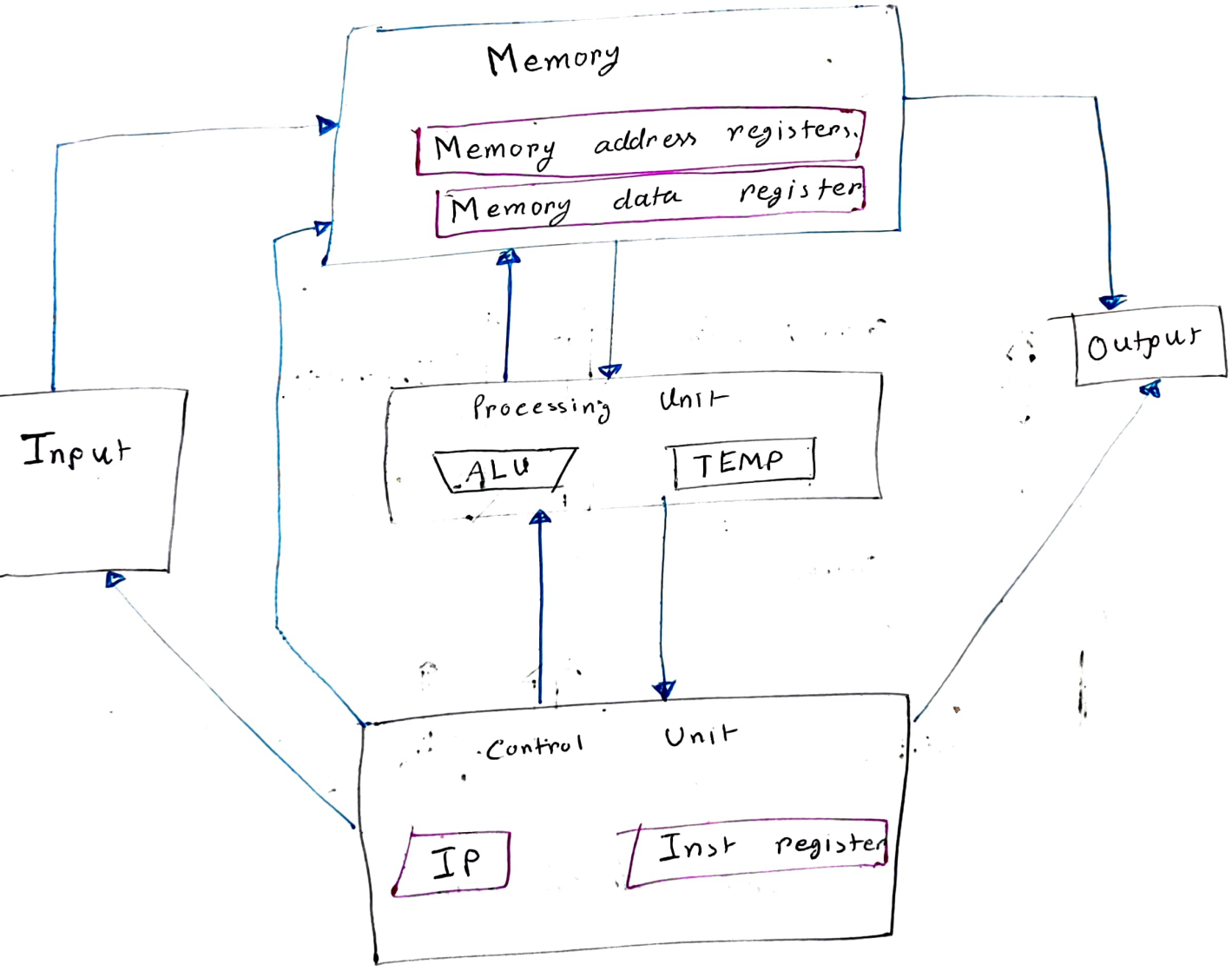


Figure: Von Neumann Architecture.

The Harvard Architecture;

- Data memory এবং instruction memory আলাদা (shared না).
  - তাই একইসাথে data এবং instruction কে access করতে পারবে, বিধায় কোনো Hazard সৃষ্টি হয় না.

- This architecture have seperate physical storage and signal pathways for instruction and data.

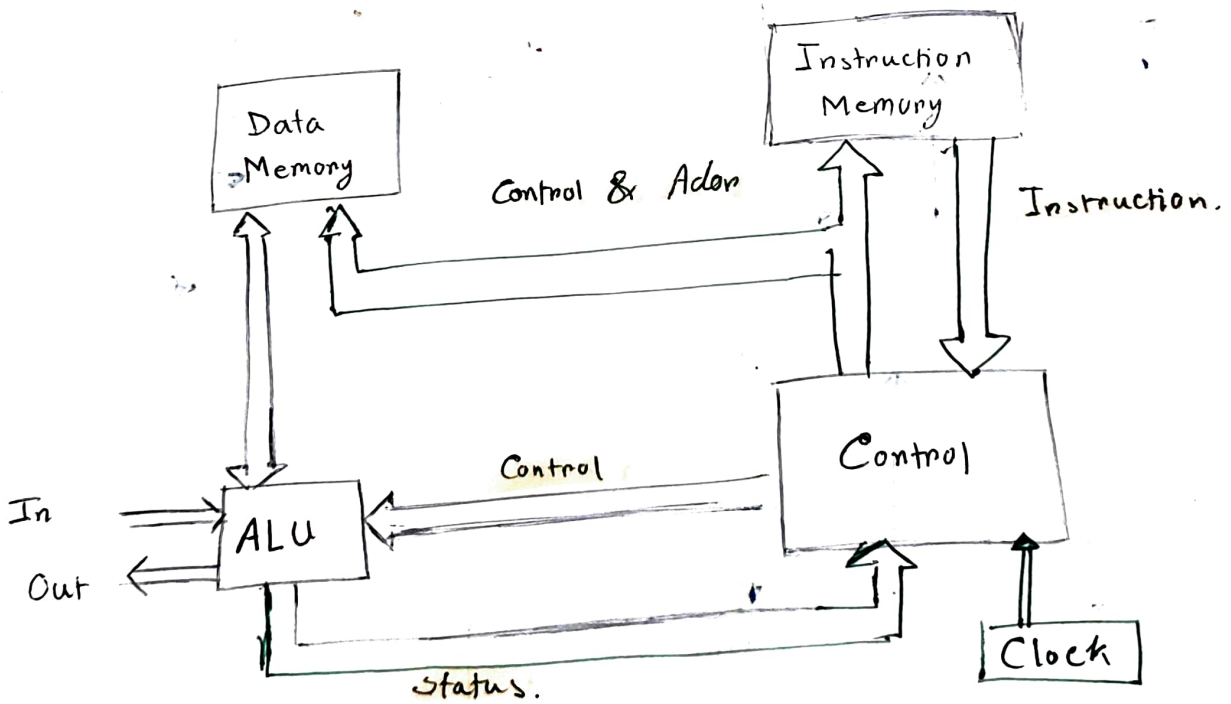


Figure: Harvard Architecture.

- Harvard architecture computer can thus be faster for a given circuit complexity because instruction fetches and data access do not contend for a single memory pathway.
- [illegible] memory to data share, characteristics share not mandatory.
  1. (Word width, timing, implementation, technology and memory access structure can differ).

- Instruction memory is often wider than data memory.
- In some system instruction can be stored in read-only memory while data memory generally requires read-write memory.

Calculate the execution time of the programs and SPEC ratio from table below as well as the summary benchmark point using geometric mean?

Benchmark Program	Instruction count $\times 10^9$	CPI	Clock rate (GHz)	Reference time (seconds)
Go	1970	1.10	3.2	10490
Hmmer	2910	0.6	3.2	9330

⇒

Go

$$\text{execution time} = \frac{1970 \times 10^9 \times 1.10}{3.2 \times 10^9}$$

$$= 677.1875$$

$$\therefore \text{Spec ratio}_{Go} = \frac{10490}{677.1875}$$

$$= 15.49053$$

Hmmer

$$\text{execution time} = \frac{2910 \times 10^9 \times 0.6}{3.2 \times 10^9}$$

$$= 545.625$$

$$\text{Spec ratio}_{Hmm} = \frac{9330}{545.625}$$

$$= 17.09965$$

$$\therefore \text{Geometric mean} = \sqrt[2]{15.49053 \times 17.09965}$$

$$= 16.2752$$

Chapter - 1 Formula's

- $P \propto \frac{1}{\text{Execution time}}$

- $\frac{P_A}{P_B} = \frac{E_B}{E_A} = \frac{\text{CPU time}_B}{\text{CPU time}_A}$

- CPU time या execution time = CPU Clock Cycles * Clock Cycle time

$$= \frac{\text{CPU Clock Cycles}}{\text{Clock rate}}$$

$$= \frac{\text{instruction count} \times \text{CPI}}{\text{Clock rate}}$$

$$= \text{instruction count} \times \text{CPI} \times \text{Clock Cycle time.}$$

- Clock Cycle time = $\frac{1}{\text{Clock rate}}$

- Same ISA ~~के लिए~~ instruction count same.

- Clock Cycles = $\sum_{i=1}^{n} (\text{CPI}_i \times \text{instruction count}_i)$

- Weighted average CPI.

$$\text{CPI} = \frac{\text{Clock Cycles}}{\text{instruction count}}$$

$$= \sum_{i=1}^{n} \left(\text{CPI}_i \times \frac{\overbrace{\text{instruction count}_i}^{\text{Relative frequency}}}{\text{instruction count}}\right)$$

- Power = Capacitive load $\times$ Voltage^2 $\times$ Frequency

- Spec ratio $= \dfrac{\text{reference time}}{\text{execution time}}$

- Geometric mean $= \sqrt[n]{\prod_{i=1}^{n} \text{spec ratio}_i}$

- Amdhal's law,

$$T_{improved} = \frac{T_{affected}}{\text{improvement factor, } n} + T_{unaffected}$$

- MIPS $= \dfrac{\text{instruction count}}{\text{execution time} \times 10^6}$

$$= \frac{\text{Clock rate}}{\text{CPI} \times 10^6}$$

- Cost per die $= \dfrac{\text{cost per wafer}}{\text{Dies per wafer} \times \text{Yield.}}$

- Dies per wafer $\approx$ Wafer area / Die area

- Yield $= \dfrac{1}{(1 + (\text{Defets per area} \times \text{Die area}/2))^2}$

Integrated Circuit cost