Lecture 1-3: August 9 and 11, 2021
Computer Architecture and Organization-II
Biplab K Sikdar

The machine *computer* came in 1940's. The study on *computer architecture* is the accumulation of concepts, developed and refined over many years, towards the design of this powerful computing machine - today's *Computer*.

0.1 Today's Computer

Today's conventional computer stores the set of instructions in its memory and executes (process) them one by one. The five major components of a computer are shown in Figure 1.

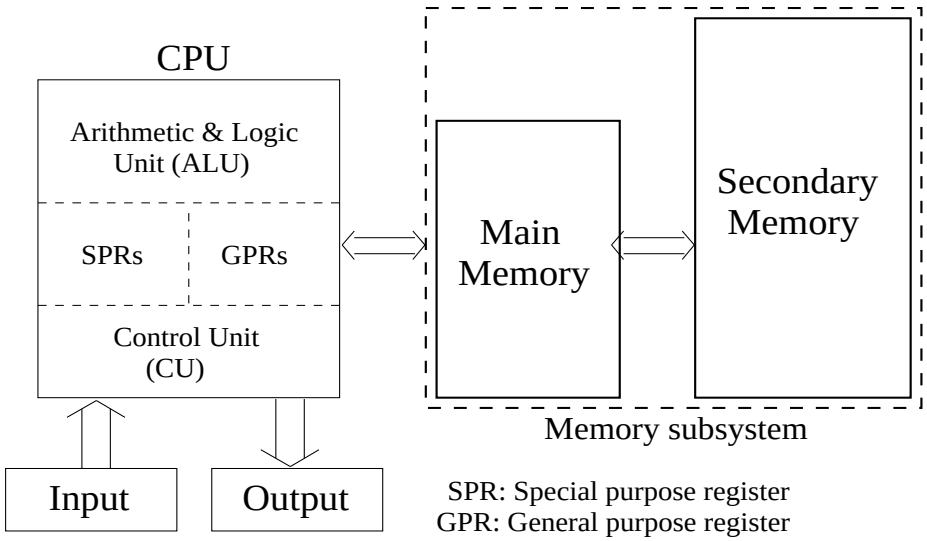


Figure 1: Basic organization of a machine computer

Today's computer is nothing but an information processor (Figure 2).

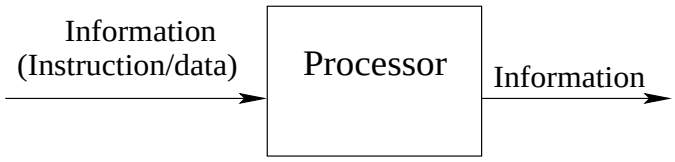


Figure 2: The computer

0.2 Technology Trends

Development of VLSI technology made it possible today that the performance of today's microprocessor is even comparable with that of a supercomputer (Figure 3).

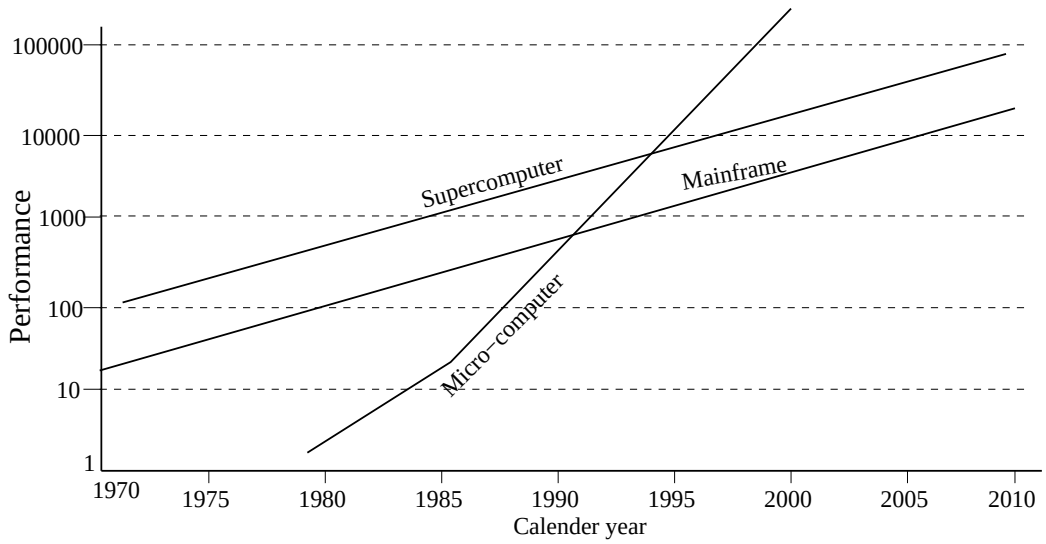


Figure 3: Performance trend

It is due to the improved compaction density within a chip (processor/memory etc.).

The notable progress: SSI $\rightarrow$ MSI $\rightarrow$ LSI $\rightarrow$ VLSI $\rightarrow$ ULSI.

A Exponential growth was first predicted by Gordon E Moore (Moore's Law) in 65.

Moore's Law: Computing power of a m/c doubles every 18/24 months.

Moore's prediction is accurate. Noticed in Intel's microprocessor family (Figure 4).

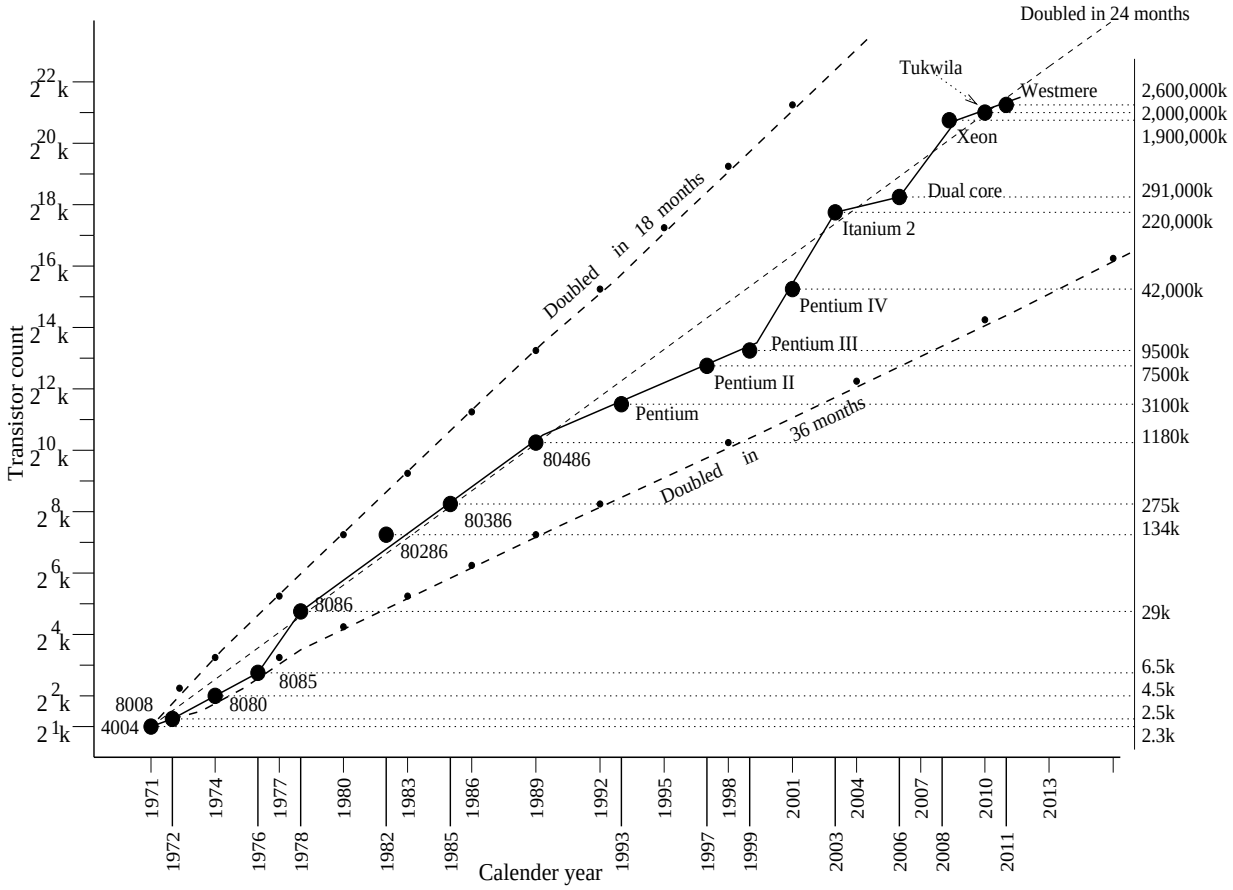


Figure 4: Intel processors' transistor count

The number of transistor integrated per chip gets doubled every 24 months.

Peripheral/memory technology are also important to exploit computation power of a processor. Memory capacity grows exponentially with time. As memory design adopts latest VLSI technology, Moore's law is also applicable to memory chip.

B Major objective of processor design - develop high speed, low cost, low power but high performance tiny CPUs.

Performance of a CPU increases with the increase in transistors within the chip.

Increase in number of transistors implies more functions for computation.

Intel's 4004 μp (1971) - 2.3k transistors, chip area 12 mm^2 with 10 μm technology.

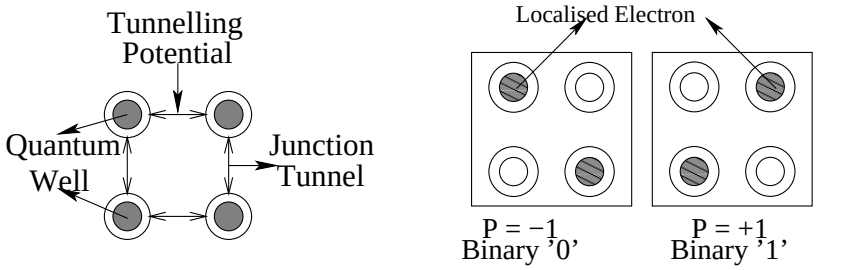
$\Downarrow$

Intel's 10-core Xeon (2010) - 512 mm^2, 260 crore transistors, 32 nm technology.

Current projection - scaling process of CMOS technology will end up to 7 nm.

C Quantum computation, molecular & nano technologies are expected to take birth.

Quantum-dot Cellular Automata (QCA) relies on coulombic interactions (Figure 5).



(a) Structure of a QCA Cell (b) QCA Cell with two polarization

Figure 5: QCA cell

0.3 Computer Performance

The growth of computer performance is superlinear with cost (Figure 6).

Target of a new architectural proposal is to realize a cost effective design.

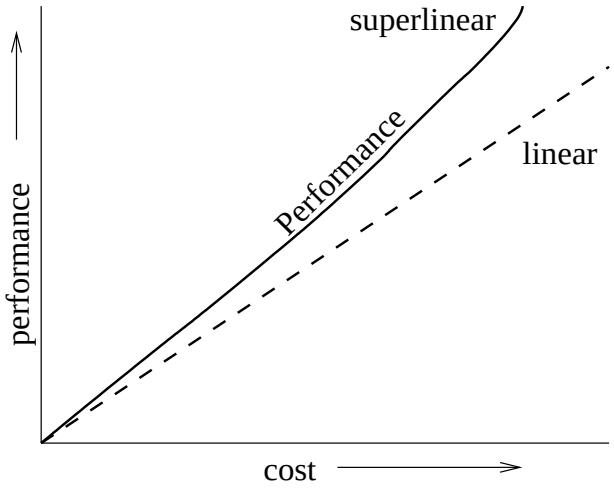


Figure 6: Performance and cost relationship

A Grosch (1940's) postulates a relation - Grosch's law.

Grosch's law: Computation power of a system is proportional to square of its cost.

Performance $p = k \times C^2$, where k is a constant and C is the cost.

B Measure of CPU performance can be defined as

$$\textit{performance} = \frac{1}{CPU\ execution\ time}$$

If CPU_1 takes longer time to finish execution of a program than CPU_2, then CPU_1 is of low performance.

If CPU_1 takes $3t$ and CPU_2 takes t time, then CPU_2 peformance 3 times better.

Though absolute measure of performance of a CPU is not realizable.

CPU $execution\ time$

Time taken by a CPU to execute a program can be measured as

CPU $execution\ time$

= $(number\ of\ instructions) \times (clocks\ per\ instruction\ (CPI)) \times (seconds\ per\ clock)$

= $\frac{number\ of\ instructions \times CPI}{clock\ rate}$.

Therefore, improved performance means

(i) Reduction in number of instructions in a program,
(ii) Reduction of CPI, and
(iii) Increase in the clock rate.

Increase in clock rate depends on hardware technology.

Number of instructions and CPI are directly related to computer architecture.

C Parallel processing can improve system performance.

Gene Amdahl, however identifies limitations of parallel processing.

Amdahl's speed up formula: *If 'f' is the fraction of run time T of a program PR required for unparallelizable computations within PR, assuming that the remaining parallelizable part of PR enjoys perfect speed up 'p' when run on 'p' processors, the overall speed up for the program execution is*:

$S_p = \frac{runtime\ of\ the\ original\ program}{execution\ time\ with\ p\ processors} = \frac{T}{fT+(1-f)\frac{T}{p}} = \frac{p}{1+(p-1)f}$ or $\frac{1}{f+(1-f)/p}$.

Here, communication overhead is ignored.

If a program is run in a system with p processors, then speed up S_p can not be p.

If $p \to \infty$, then

$S_p = \frac{1}{f}$.

That is, for a large number of processors, speedup depends on fraction of a program that can not be parallelized. If $f = 0.10$, speed up can be not more than 10.

f is the sequential bottleneck in a program.

0.4 Performance Estimation

Performance of a computer is conventionally expressed either as IPS (instruction per second), or MIPS (millions of instructions per second).

Currently GIPS (giga instructions per second), TIPS (tera instructions per second), and PIPS (peta instructions per second) [1] are also used.

These represent highest level of performance that ideally can be extracted.

Engineering and scientific community normally follows FLOPS (floating point operations per sec), MFLOPS (mega FLOPS), GFLOPS (giga FLOPS), and PFLOPS (peta FLOPS) as measure of computer performance.

CPI (clock per instruction) is considered as more realistic measure of performance.

However, performance evaluation normally relies on the average CPI

$$CPI_{average} = \frac{\sum_i Instruction\ type_i \times CPI_i}{IC},$$

CPI$_i$: clock cycles per instruction for instruction type$_i$,

IC = $\sum_i Instruction\ type_i$ -that is, instruction count.

[1] One kilo (1K) = 2^{10}, one mega (1M) = 2^{20}, one giga (1G) =2^{30}, one tera (1T) = 2^{40}, one peta (1P) = 2^{50}, one exa (1E) = 2^{60}, one zetta (1Z) = 2^{70}, one yotta (1Y) = 2^{80}.

0.5 Performance Requirements

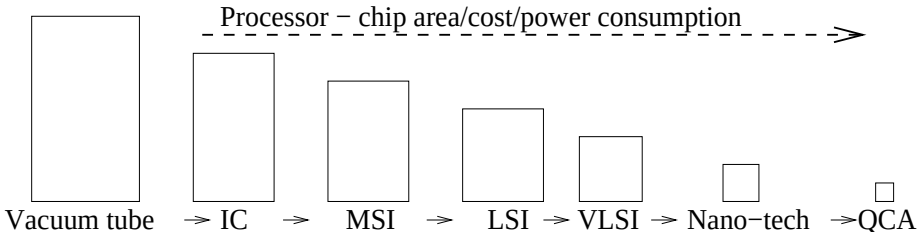


Figure 7: Chip generations

CPU speed is to be more than 10^{18} computations per sec (exa scale).

CPU/computer are tiny in size, low cost, and of low power consumption.

A Figure 7: chip area, cost, and power consumption are getting reduced with time.

It is expected that by 2030 a small CPU will be able to replace human brain.

Processor (chip) of 2040 will be representing a society (Figure 8).

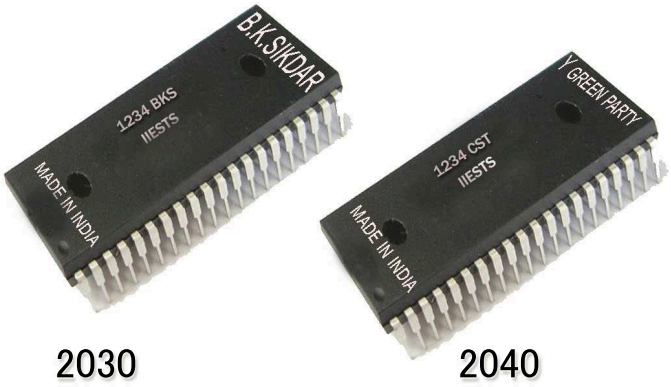


Figure 8: Our future

B Modeling of human brain may require billions of transistors.

More compaction of transistors on a chip strikingly increases power consumption.

In Electronic Discrete Variable Automatic Computer (EDVAC of 50's), it is 50KW.

For super computer Jaguar (2012), peak power $\simeq$ 10 MW.

Growth of power requirement in INTEL processors are shown in Figure 9.

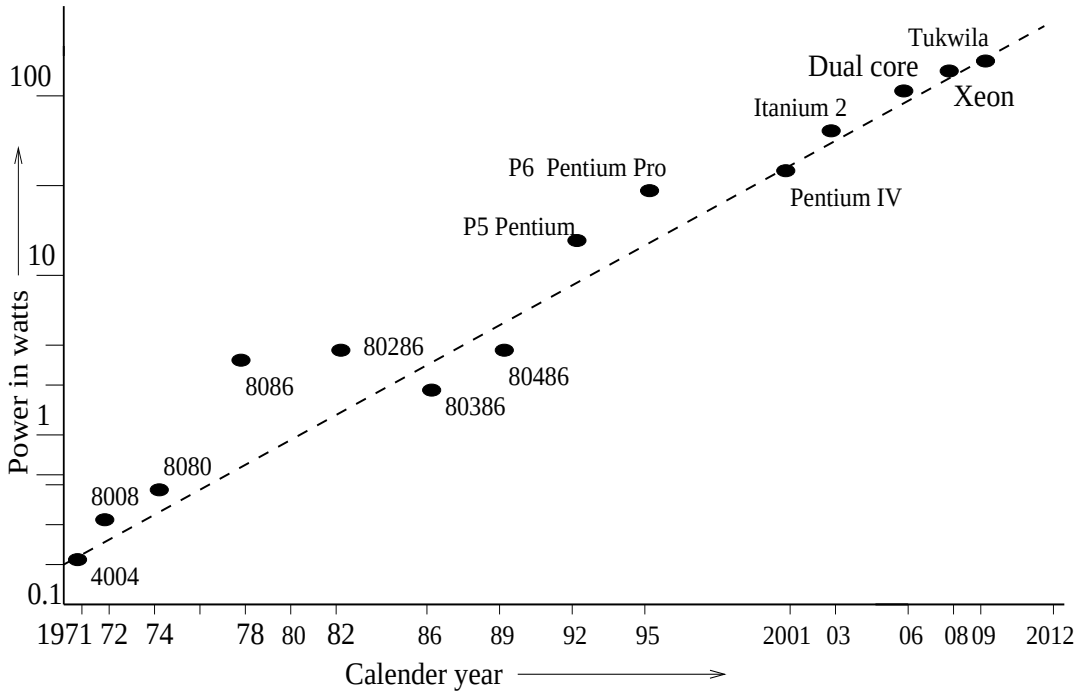


Figure 9: Power dissipation of INTEL processors

0.6 Von Neumann Architecture

1946-48 von Neumann architecture/IAS (immediate access store) architecture

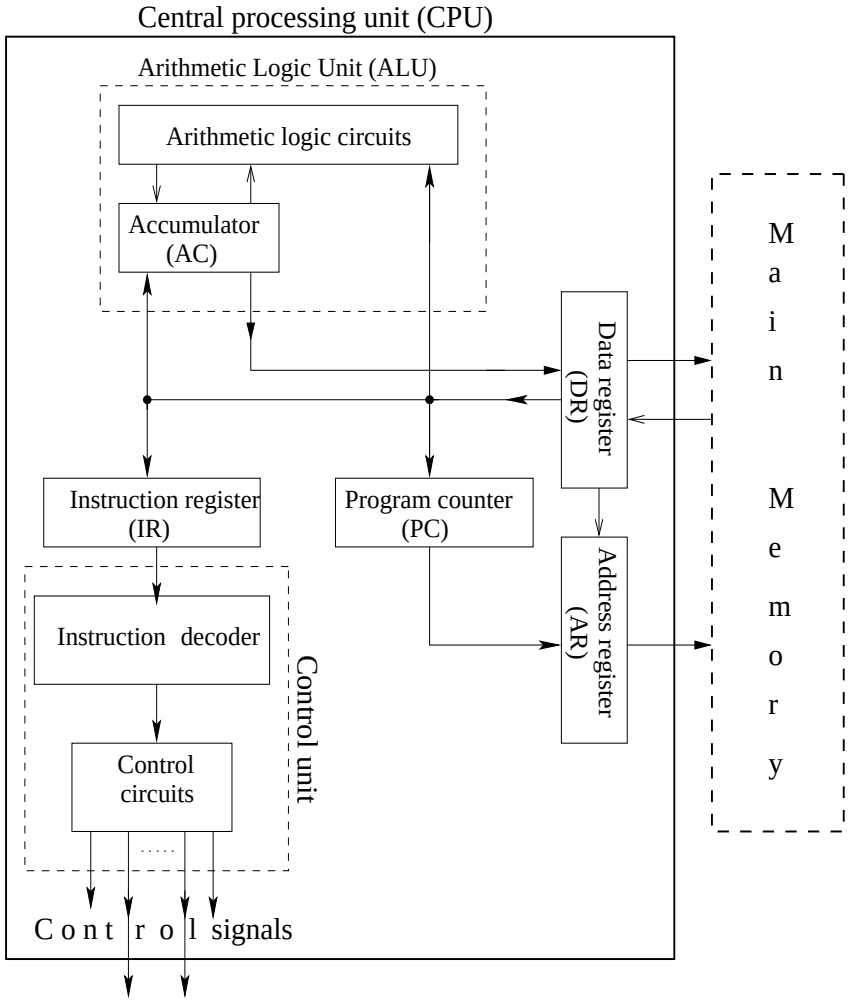


Figure 12: The basic von Neumann architecture

0.7 Extension of Basic Organization

Several ways by which basic configuration of Figure 12 is made more powerful.

1. Additional registers are added. These aree:

- General purpose (multi purpose) registers
- Index registers
- Special purpose registers

2. Capabilities of the ALU are enhanced

3. Instruction prefetching is used

4. Special control circuitry

5. Speed up through pipelining within the CPU

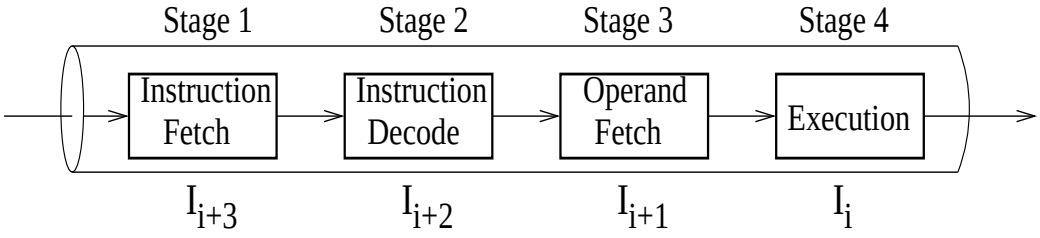


Figure 13: Pipelining

6. Parallel processing

- ALU is divided into K-parts (Figure 14) - K partitioned ALU

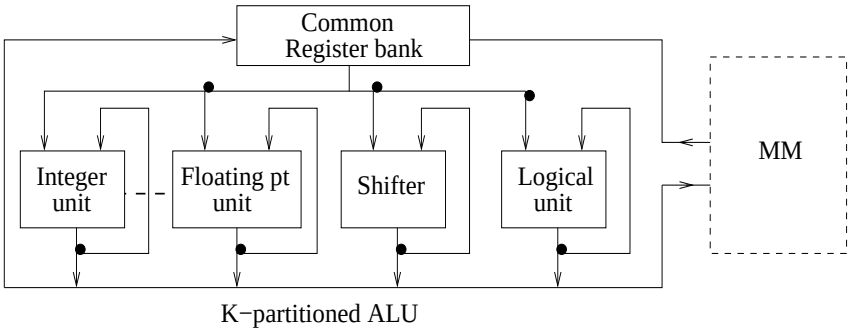


Figure 14: CPU with K-partitioned ALU

- ALU is replicated K-times (Figure 15)

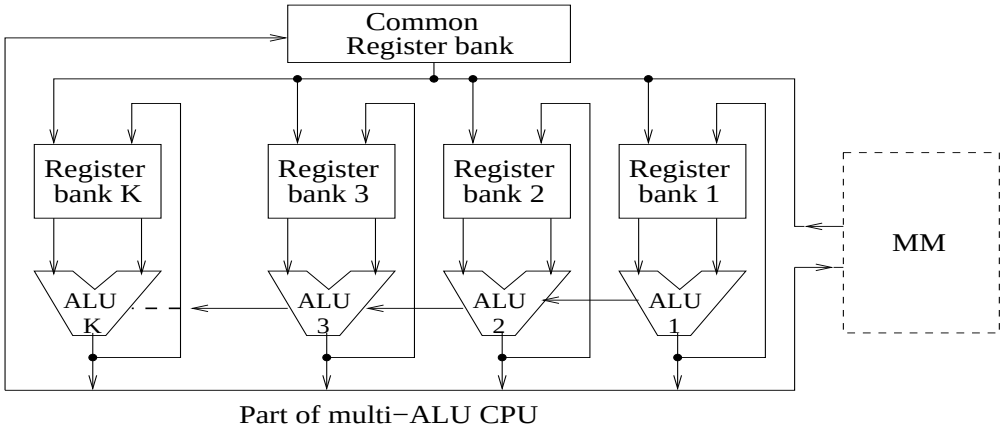


Figure 15: CPU with replicated ALUs

- Latest introductions: multiple processor core within a CPU, very large instruction word (VLIW) architecture, superscalar architecture, etc.

7. Speculative execution

Inst Level Parallelism (ILP): To ensure many inst to be executed simultaneously.

Speculative execution increases processing performance through lookahead.

Target is to keep functional units of a system as busy as possible by executing the instructions which may be needed later on.

- Control speculation:

  In Figure 16(a), when CPU decodes I_b, fate of condition C may not be known.

  To know correct exec path (I_{b+1}, I_{b+2}, $\cdots$ or I_t, I_{t+1}, $\cdots$), have to wait for C.

  CPU predicts/speculates (say, condition C is satisfied) direction of branch, and continues to execute along that path (say I_t, I_{t+1}, $\cdots$) (Figure 16(b)).

  If prediction is found wrong, results of this execution path are discarded.

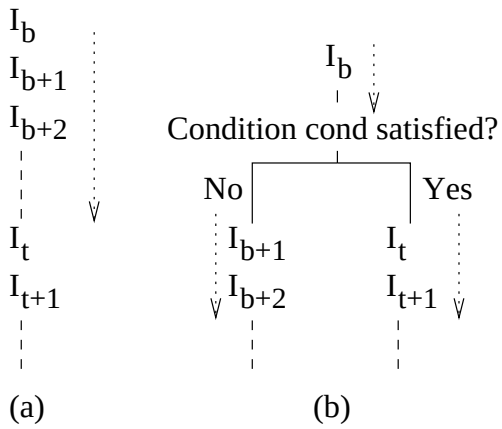


Figure 16: Control speculation

- Data speculation:

  Example 0.1 Let take the program segment.

  ```
  Load   r1, addr1
  mult   r1, r2      ⇒   r1 = r1 * r2
  sub    r3, r4      ⇒   r3 = r3 - r4
  and    r5, r1      ⇒   r5 = r5 . r1
  ```

  Execution of it can follow

at processor 1	at processor 2
load	sub
↓	
mult	
↓	
and	

  The 'load', 'mult' & 'and' are executed serially (there are dependencies).

  If data speculation for r1 is followed, then

at processor 1	at processor 2	at processor 3
	predicted (r1)	
	↓	
load	mult	sub
	↓	
	and	

  If misspeculation, reexecute.

- In eager execution, whenever program counter comes to a branch (I_b), execution continues speculatively down both the paths I_{b+1}, I_{b+2}, $\cdots$ and I_t, I_{t+1}, $\cdots$ (does not wait for decision on condition C).

  When C is resolved, results of the incorrect path are discarded.

  Prediction/speculation improves performance of a computer system, but hardware supported speculative computing increases power consumption.

8. Green computing:

Execution speed increases with the increase in processor clock frequency.

High clock frequency adds to huge power consumption[2].

Now, designers adopt transition to multicore architectures.

Architecture of pocessor core is chosen simpler to reduce average energy per operation.

[2]The dynamic power is P= $CV^2 f$. C is the capacitance, V is voltage, and f is the clock frequency. V can not be decreased significantly due to switching noise.