CS31001 COMPUTER ORGANIZATION AND ARCHITECTURE

Debdeep Mukhopadhyay,
CSE, IIT Kharagpur

References/Text Books

- ***Theory:***
  - ***Computer Organization and Design, 4th Ed, D. A. Patterson and J. L. Hennessy***
  - ***Computer Architceture and Organization, J. P. Hayes***
  - *Computer Architecture, Berhooz Parhami*
  - *Microprocessor Architecture, Jean Loup Baer*
- ***Laboratory:***
  - *Douglas Smith, HDL Chip Design (for Verilog)*
  - *SPIM Tutorial :pages.cs.wisc.edu/~larus/**spim**.html*

Pre-Midsem Syllabus

- Overview: Terms and Taxonomy
- Instruction Set Architecture:
  - Instruction and Addressing
  - Procedures and Data
  - Assembly Languages Programs:
    - SPIM simulator and Debugger
- Numbers and Computers:
  - Number Representations
  - Adders
  - Multipliers
  - Floating Point Arithmetic

Pre-MidSem Syllabus

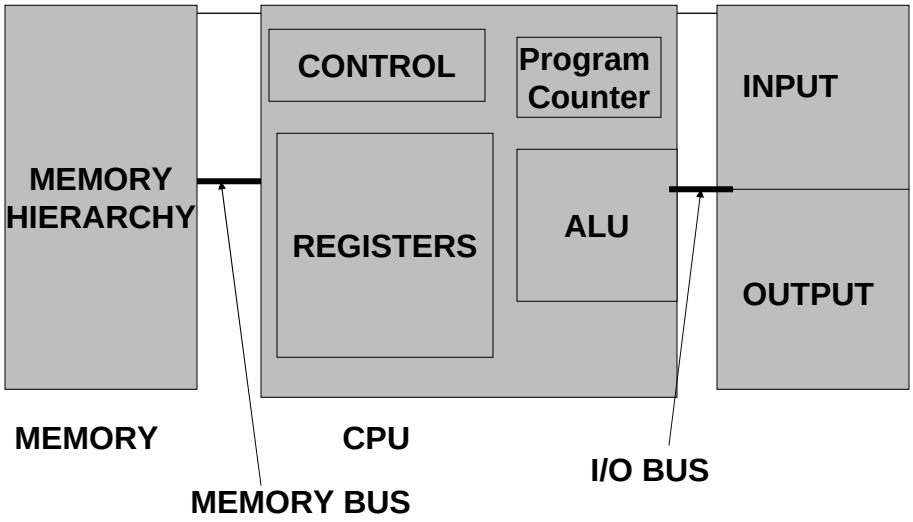
- Design of a Processor:
  - Instruction Execution Steps
  - Control Unit Design: Microprogramming
  - Pipelined Data Paths
    - Performance
    - Hazards

An Overview

Von Neumann Model

- 1903-1957
- Contributed to give a very basic model, often referred to as **Von Neumann model**

The von Neumann Machine



The Stored Program Concept

- Programs are sequence of instructions stored in the memory.
- The CPU consists of:
  - Program Counter (PC) indicates the address of the next instruction to be executed.
  - ALU (Arithmetic Logic Unit)
    - performs arithmetic and logical operations
  - Registers: High Speed storage of operands.
  - Control Unit: That interprets instructions and causes them to be executed.

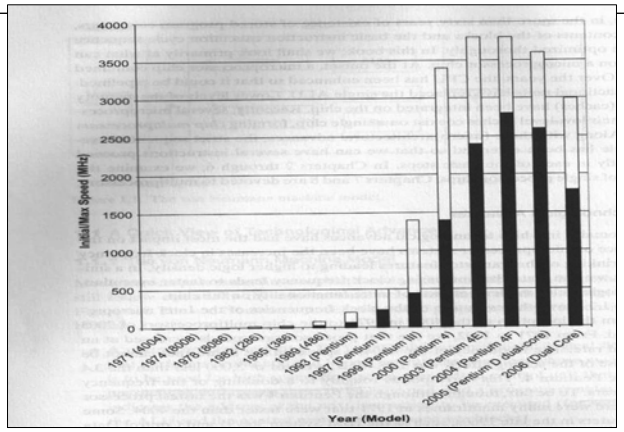
The Stored Program Concept

- The memory stores the instructions, data and intermediate results. Memory has several hierarchy:
  - registers being the highest level and the fastest form.
- Input/ Output: Transmits and receives results and messages (information) from and to the outside world respectively.

Instruction Execution Cycle

- Next instruction is fetched from memory.
- Control Unit decodes the instruction.
- Instruction is executed:
  - ALU based
  - **load** from a memory location to a register
  - **store** from a register to the memory
  - testing condition of a potential branch
- PC is updated (incremented except for a branch)
- Repeat

Brief History

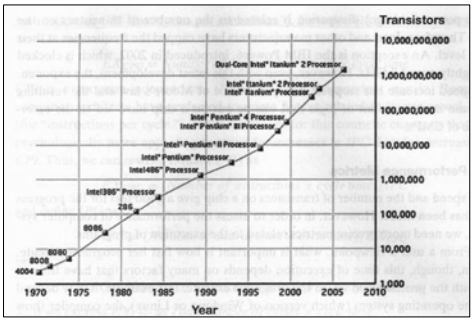


Black bars: frequency at introduction

White bars: peak frequency

- Evolution of Intel Microprocessors (from Jean Baer, Microprocessor Architecture, Cambridge University Press)

Moore's law for Intel Microprocessors

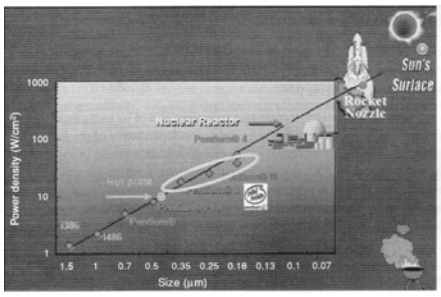


- Evolution of Intel Microprocessors (from Jean Baer, Microprocessor Architecture, Cambridge University Press)

Some figures

- 1971: Intel 4004, 1.08 MHz, 2,300 transistors
- 2003: Intel Pentium 4, 3.4 GHz, 1.7 billion transistors
  - Frequency increases roughly double per 2.5 years
  - Number of transistors roughly double every two years (Moore's Law).
- How will the trend continue in the future?

Power Dissipation in Intel Processors



- Evolution of Intel Microprocessors (from Jean Baer, Microprocessor Architecture, Cambridge University Press)

Performance Metrics

- Raw Speed and number of transistors give a good indication of the developments made by the processors.
- However we need more precise metrics.
  - to evaluate how fast a program executes
- But it depends on several factors:
  - Operating System
  - Compiler
  - Network
  - Nature of Program

Program Independent Metrics and Benchmarks

- Program Independent metrics: Metrics which are not affected by the types of programs.
- Benchmarks: A suite of programs, that indicate the load of the processor.

Instructions Per Cycle (IPC)

- Metrics to asses the micro-architecture and the memory hierarchy, should be independent of the IO subsystem.
- Define, EX_{CPU} as the time to execute a program (a collection of instructions), when the code and data both reside in the memory.
  - EX_{CPU}=Number of Instructions x Time to execute one instruction

CPI vs IPC

- Now, Time to execute one instruction=Cycles Per Instruction (CPI) x Cycle time
  - CPI is a program-independent, clock frequency independent metric.
- IPC (Instructions per cycle)=1/CPI
  - More intuitive as one tries to increase IPC (rather than decreasing CPI).

IPC = (Number of Instructions x cycle time)/EX_{CPU}

Performance

- Can be defined as the reciprocal of EX_{CPU}.
- Three important factors:
  - Compiler driven: For a given set of Instructions (called as ISA-Instruction Set Architecture), and a given program, it decides the number of instructions.
  - Micro-architecture design and implementation: Smaller the CPI or more the IPC, better the performance.
  - Technology: Decides the cycle time.

Reduction of Instructions does not necessarily make the programs faster

- Consider the example, where a multiplication program can be realized by shift and add.
- The number of instructions increase.
- But the overall time required reduces.
  - This is because, not all instructions take the same number of cycles.

Conclusions

- Von Neumann Model: The Load Store Architecture
- The Instruction Cycle.
- The trend in processor designs, Moore's law.
- Performance Metrics.