Submission

(Reading Assignment - 1)

(CSN-221)

Different Computer Architectures

by

Jitesh Jain

(Enrol number: 19114039)

(B. Tech CSE)

(Batch O2)

Abstract

Technology is growing very fast, every day we see new things and changes to technology. There is a lot of history for computer systems. Nearly 60 years. Computer Architecture has changed from day one to till today. Based on Computer Architecture new computer systems are developed. Computer Architecture is different for microcomputers, minicomputers, mainframes, laptops, palmtop, tablets, mobile phones which are used as smart computers. Each device is having its own architecture. In this report, I explain a number of computer architectures with the basic concepts behind the same.

Contents

Von Neumann Architecture [1]

Introduction

Von Neumann architecture is based on the stored-program computer concept, where instruction data and program data are stored in the same memory. This design is still used in most computers produced today.

Concept

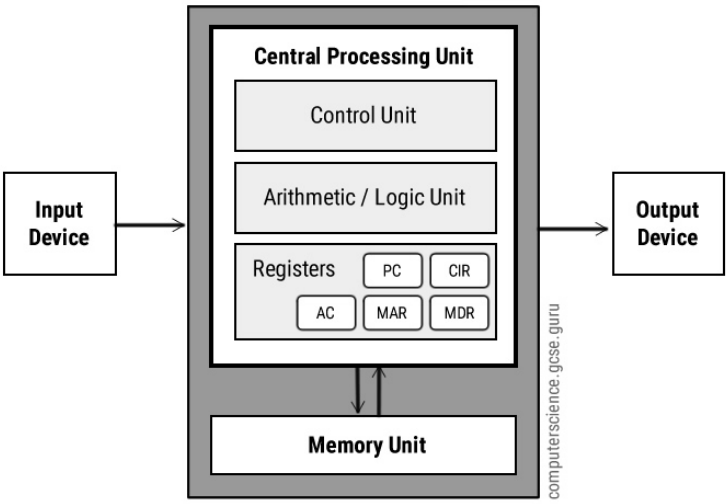
The basic concept behind the von Neumann architecture is the ability to store program instructions in memory along with the data on which those instructions operate. Until von Neumann proposed this possibility, each computing machine was designed and built for a single predetermined purpose. All programming of the machine required the manual rewiring of circuits, a tedious and error-prone process.

Explained

Von Neumann architecture is composed of three distinct components:

1. The **CPU**, which can be considered the heart of the computing system, includes three main components:
    - CU determines the **order** in which instructions should be executed and controls the retrieval of the proper operands.

    - ALUs perform all mathematical and Boolean operations.
    - The registers are temporary storage locations to quickly store and transfer the data and instructions being used.
2. The computer's memory is used to store program instructions and data.
3. The I/O interfaces allow the computer's memory to receive information and send data to output devices



Von Neumann Architecture

The preceding components are connected to each other through a collection of signal lines known as a bus.

Execution

Initially, a program has to be loaded into the memory. Once the program is in memory, the operating system then schedules the CPU to begin executing the program

instructions. Each instruction to be executed must first be retrieved from memory. This retrieval is referred to as an instruction fetch. After an instruction is fetched, it is put into a special register in the CPU, called the instruction register (IR). While in the IR, the instruction is decoded to determine what type of operation should be performed. The instruction is then performed, and the results are stored back into memory and/or registers. This process is repeated for each instruction of the program until the program's end is reached.

Data Flow Architecture

Introduction [2]

Dataflow architecture is a computer architecture that directly contrasts the traditional von Neumann architecture or control flow architecture.

Concept [3]

Dataflow architectures do not have a program counter (in concept): the executability and execution of instructions are solely determined based on the availability of input arguments to the instructions so that the order of instruction execution is unpredictable, i.e. behavior is nondeterministic.

Explained [3]

A software system is created when information goes through a series of transformations. Data Flow Architecture depicts the workflow followed to create a software system. In a data flow architecture, information is pulled into the system which then flows through several modules and goes through transformation until destination (output or a data store) is achieved. In Data flow architecture, transformations can be reused and modified.

Execution [4]

There are three types of execution sequences between modules:

I. *Pipe and Filter:*

A. Filters are the processes that run at the same time, it means that they can run as different threads, coroutines, or be located on different machines entirely.

B. Each pipe is connected to a filter and has its own role in the function of the filter.

C. The filters are robust where pipes can be added and removed at runtime.

D. A filter reads the data from its input pipes and performs its function on this data and places the result on all output pipes. If there is insufficient data in the input pipes, the filter simply waits.

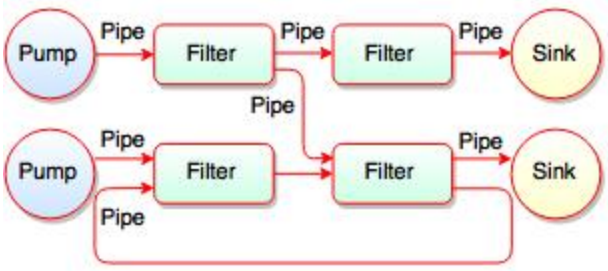


Fig. Pipes and Filters

II. *Batch Sequential:* In Batch sequential, separate programs are executed in order and the data is passed as an aggregate from one program to the next.

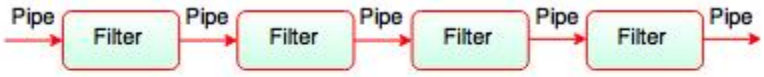


Fig. Batch Sequential

III. *Process Control:*

A. Process Control Architecture is a type of Data Flow Architecture, where data is neither batch sequential nor pipe stream.

B. In process-control architecture, the flow of data comes from a set of variables that control the execution of the process.

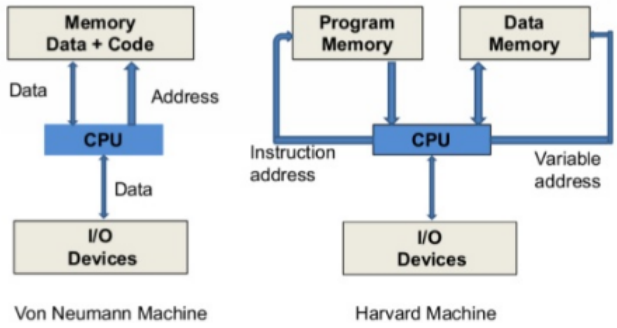
Harvard Architecture [5]

Introduction [6]

The **Harvard architecture** is a computer architecture with separate storage and signal pathways for instructions and data.

Concept [7]

Harvard architecture is used when data and code is present in different memory blocks. A separate memory block is needed for data and instruction. Data can be accessed by one memory location and instruction can be accessed by a different location. It has data storage entirely contained within the central processing unit (CPU). CPU can read and write instructions and process data access. Harvard architecture has different access codes and data address spaces that is, the instruction address zero is not the same as data address zero.



Quantum Computer Architecture [8]

Introduction

Quantum computing is essentially harnessing and exploiting the amazing laws of quantum mechanics to process information. A traditional computer uses long strings of “bits,” which encode either a zero or a one. A quantum computer, on the other hand, uses quantum bits, or qubits.

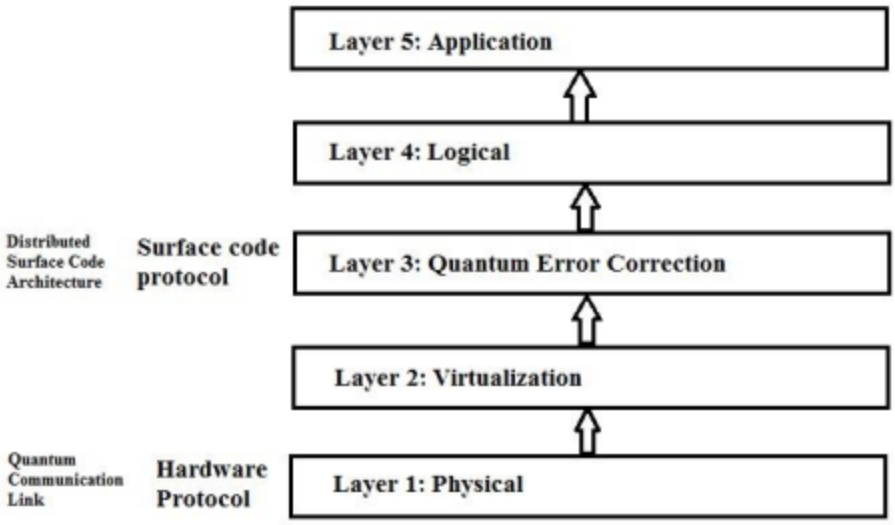
Concept

Quantum computers have the potential to perform certain calculations significantly faster than any silicon-based computer.

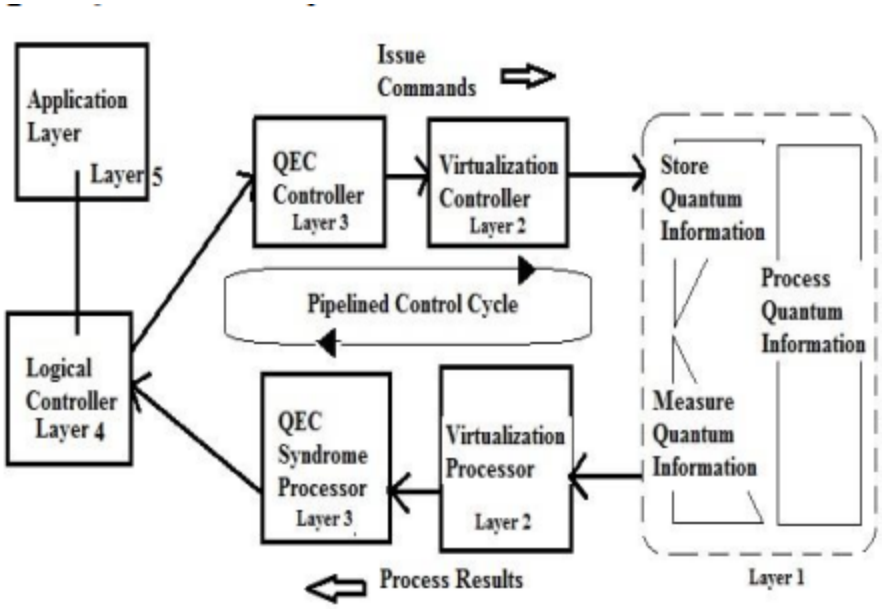
Explained

Quantum information is stored in the electron spin states of a charged quantum dot controlled by ultrafast optical pulses. Optical control makes this system very fast, scalable to large problem sizes, and extensible to quantum communication or distributed architectures. The design of this quantum computer centers on error correction in the form of a topological surface code, which requires only local and nearest-neighbor gates. We analyze several important issues

of the surface code that are relevant to architecture, such as resource accounting and the use of Pauli frames.



Quantum Computer Architecture



Central Architecture of Quantum Computer

While a classical three-bit state and a quantum three-qubit state are both eight-dimensional vectors, they are manipulated quite differently for classical or quantum computation. For computing, in either case, the system must be initialized, for example into the all-zeros string |000>, corresponding to the vector (1,0,0,0,0,0,0,0).

In classical randomized computation, the system evolves according to the application of stochastic matrices, which preserve that the probabilities add up to one (i.e., preserve the L1 norm).

In quantum computation, on the other hand, allowed operations are unitary matrices, which are effectively rotations (they preserve that the sum of the squares adds up to one, the Euclidean or L2 norm). (Exactly what unitaries can be applied depending on the physics of the quantum device.) Consequently, since rotations can be undone by rotating backward, quantum computations are reversible. (Technically, quantum operations can be probabilistic combinations of unitaries, so quantum computation really does generalize classical computation).

Finally, upon the termination of the algorithm, the result needs to be read off. In the case of a classical computer, we sample from the probability distribution on the three-bit register to obtain one definite three-bit string, say 000. Quantum mechanically, we measure the three-qubit state, which is equivalent to collapsing the quantum state down to a classical distribution (with the coefficients in the classical

state being the squared magnitudes of the coefficients for the quantum state, as described above), followed by sampling from that distribution. Note that this destroys the original quantum state. Many algorithms will only give the correct answer with a certain probability. However, by repeatedly initializing, running, and measuring the quantum computer's results, the probability of getting the correct answer can be increased. In contrast, counterfactual quantum computation allows the correct answer to be inferred when the quantum computer is not actually running in a technical sense, though earlier initialization and frequent measurements are part of the counterfactual computation protocol.

Parallel Computing Architecture [9] (Flynn's taxonomy)

Introduction

In Flynn's taxonomy, the processors are classified based on the number of concurrent instruction and data streams available in the architecture.

Explained

1. ***SISD***
    a. SISD is a computer architecture in which a single uni-core processor executes a single instruction stream, to operate on data stored in a single memory.
    b. A sequential computer that exploits no parallelism in either the instruction or data streams.
    c. Examples of SISD architecture are the traditional uniprocessor machines like PC or old mainframes.

2. ***SIMD***
    a. SIMD is a computer with multiple processing elements that perform the same operation on multiple data points simultaneously. It performs

parallel computations, but only on a single instruction at a given moment.

b. Example, an array processor

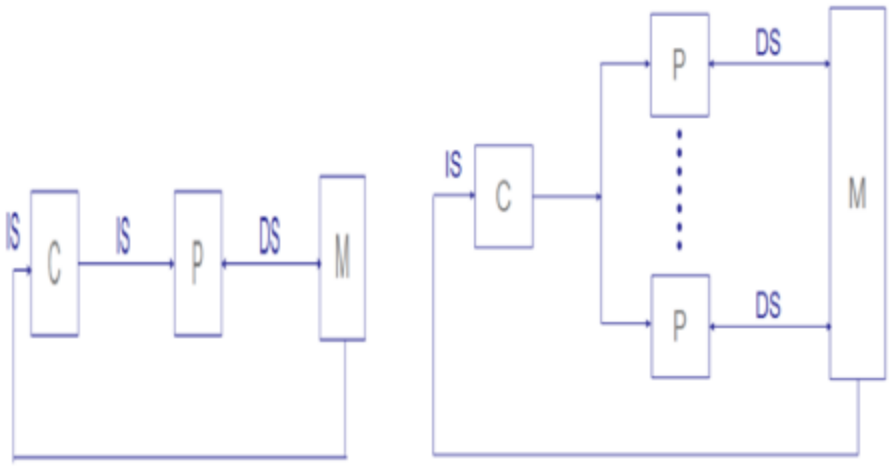


Fig.3 SISD and SIMD

3. ***MISD***

a. MISD is a parallel computing architecture where many functional units perform different operations on the same data.

b. Architecture generally used for fault tolerance.

c. Heterogeneous systems operate on the same data stream and must agree on the result.

d. Examples include the Space Shuttle flight control computer.

4. MIMD

a. MIMD machines include a number of processors that function asynchronously and independently.

b. At any time, different processors execute different instructions on different pieces of data.

c. Multiple autonomous processors simultaneously executing different instructions on different data.

d. Distributed systems are generally recognized to be MIMD architectures; either exploiting a single shared memory space or a distributed memory space.

MISD & MIMD

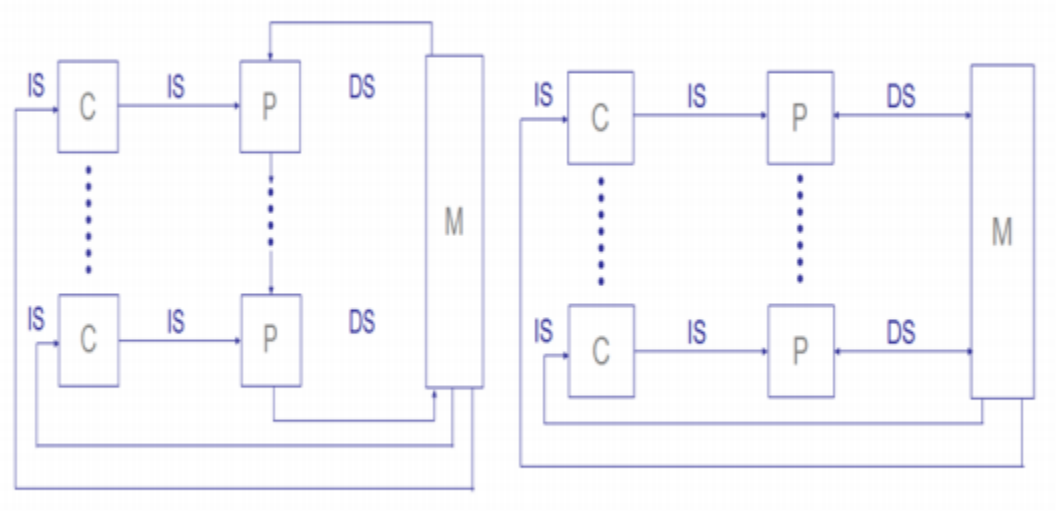


Fig.4 MISD and MIMD

References

[1]	http://www2.cs.siu.edu/~cs401/Textbook/ch2.pdf
[2]	https://en.wikipedia.org/wiki/Dataflow_architecture
[3]	https://www.educba.com/data-flow-architecture/
[4]	https://www.tutorialride.com/software-architecture-and-design/data-flow-architecture.htm#:~:text=Data%20flow%20architecture%20is%20a,architecture%20has%20been%20successfully%20implemented.
[5]	https://tdck.weebly.com/uploads/7/7/0/5/77052163/03_-_harvard_architecture_comparison.pdf
[6]	https://en.wikipedia.org/wiki/Harvard_architecture#:~:text=The%20Harvard%20architecture%20is%20a,pathways%20for%20instructions%20and%20data.&text=The%20term%20originated%20from%20the,data%20in%20electro%2Dmechanical%20counters.
[7]	https://www.educba.com/types-of-computer-architecture/
[8]	https://ieeexplore.ieee.org/document/8250619#:~:text=Based%20on%20Computer%20Architecture%20new,are%20used%20as%20smart%20computers.&text=Quantum%20Computer%20is%20based%20on%20using%20device%20at%20required%20amount.
[9]	https://www.rsisinternational.org/journals/ijrsi/digital-library/volume-5-issue-4/71-77.pdf