Computer Architecture 2020 - 2021

Harkeerat Singh Sawhney

1. Structured Computer Organization

(a) Explain the difference between the translation and interpretation of instructions?

Both translator and interpretation are different ways of converting the instructions to machine readable instructions. Translator converts the whole code in machine language program. In the other hand interpretation convert instruction by instruction in machine language program.

∴ Translation = Replacing each instruction of L1 by an equivalent sequence of instruction in L0. Hence the resulting program consists entirely of L0 instruction

Interpretation = L0 takes one instruction from L1, examine it and then executes it.

(b) Explain the key characteristics of digital logic level, the microarchitecture level, the ISA level, and the operating system machine level.

DIGITAL LOGIC LEVEL (LEVEL 0):

At the lowest level there are Gates.

Gate: Each gate has one or more digital inputs (signals from 0 or 1). It can compute simple functions as output such as AND or OR. Each gate is built up of at most handful of transistors. Small number of gates can be combined to form a 1-bit memory, which can store 0 or 1. 1-bit memories can be combined in groups of 16, 32, or 64 to form registers. Each register holds a single binary number up to some maximum.

MICROARCHITECTURE LEVEL (LEVEL 1):

- Collection of 8 to 32 registers are seen to form a local memory or a circuit called an ALU.
- ALU is capable of performing simple arithmetic operations.
- Registers are connected to the ALU to form a Data Path, over which the data flows.

INSTRUCTION SET ARCHITECTURE LEVEL (LEVEL 2)

(c) Explain the boundary between software and hardware, given the above levels.

As the technology improves the boundary between software and hardware reduces by a lot. This is because it depends on the factors of costs, speed, reliability, and frequency of expected changes to put the features either in hardware or software.

(d) Explain how the terms "level", "abstraction", and "virtual machine" relate.

Virtual Machines are the different levels of a machine in which each machine has a different level of machine language. For example

Level 2 — Virtual Machine M2 with machine language L2

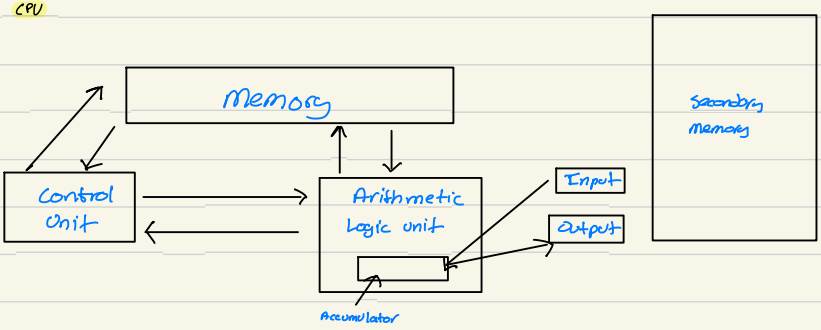
Programs in L2 are either interpreted by interpreters running on M1 or M0, or are translated to L1 or L0

L0 is machine language which is written in Binary which computer can understand.

(e) Explain the structure of the "Von Neumann Machine".

Basis of all digital computer even now. It had 5 basic parts:
- Arithmetic Logic unit
- Control Unit
- Input and output equipment
- Memory

The memory consisted of 4096 words (word holding 40 bits). Nowdays this is combined into a CPU



2. The Computer Spectrum

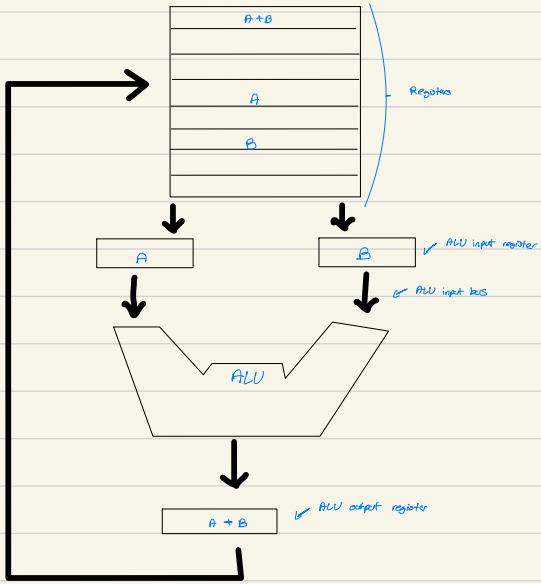
(a) Explain Moore's Law

The observation in which each new generation had four times as much memory as its predecessor, it was realized that the number of transistors on a chip was increasing at a constant rate and this predicted growth would continue for decades to come is known as Moore's Law. Nowdays Moore's law is often expressed as the number of transistors doubling every 18 months.

3. Processor Organization

(a) Explain the structure of a data path.

The internal organization of part of a typical von Neumann CPU is called as Data Path. Data Path is typically consisted of the registers (typically 1 to 32), the ALU (Arithmetic Logic Unit) and several buses connecting the pieces.



The data path of a typical von Neumann machine

(b) Explain the purpose of the register, the control unit, and ALU.

ALU (Arithmetic Logic Unit) performs operations such as addition, subtraction and other simple operations on its inputs, and then yielding a result in the output register, which can be stored back into register

Register is a small very fast storage area inside CPU. It is used to stor intermediate values from calculations or instructions that is needed again immediately.

Control Unit acts like a manager in a computer as it recieves orders from RAM in form of instruction and decode that instruction down into specific commands for other components inside computer system. It also directs the data flow and the operation of the ALU.

(c) Explain the difference between RISC and CISC

Both RISC and CISC are terms for different architectures.

A RISC machine is a computer which uses only simple commands that can be divided into several instructions which then can achieve low-level operation within a single CLK cycle. In simple words it is a CPU design plan based on simple orders and acts fast.

A CISC machine is a computer in which a single instruction can perform numerous low-level operations which is accomplished by multi-step processes. In simple words it is a CPU design plan based on single commands, which are skilled in executing multi-step operations.

Difference between RISC and CISC:

RISC	CISC
• Average clock cycle per instruction (CCPI) is 1.5	• Average clock cycle per instruction (CCPI) is range of 2 and 15.
• Performance is optimized with more focus on software	• Performance is optimized with more focus on hardware.

4. Parallelism

(a) Explain the differences and similarities of the various kinds of instruction-level parallism (pipelining, superscalar architectures)

Instruction level parelism means to get more instruction / sec out of the machine. There are two methods to do this.

Pipelinning

Pipelining is an concept in which there is an ability to fetch instructions from memory in advance, so that they are there when needed. These instructions are stored in set of registers called prefetch buffer. Prefetching divides instruction execution up into two parts: fetching and actual execution. Pipelinize carries this concept further in which instruction execution is divided into dozen of parts.

Pipeline have different steps. Below is a pipeline with 5 stages:

S1:	1	2	3	4	5	6	7	8	9
S2:		1	2	3	4	5	6	7	8
S3:			1	2	3	4	5	6	7
S4:				1	2	3	4	5	6
S5:					1	2	3	4	5
	1	2	3	4	5	6	7	8	9

TIME →

Superscaler Architectures

Superscaler Architecture is an architecture which involves two pipelines. In order for this to happen both the pipelines must be connected to instruction fetch unit. Each pipeline would have its own ALU for parallel operations. The two instructions must not conflict over resource usage and neither must depend on the result of other. However this approach was too expensive, so only one pipeline was used and was hence called superscaler Architectures.

(b) Explain the principle of pipelining

Pipelining basically tends to reduce the time for a lot of instructions to be executed. It brings the instructions from memory in advance in register called prefatched buffer. Hence, depending on instruction execution the time for one instruction to be execute decreases.

5. Machine-Level Representation of Data

(a) Explain why today's computers use a binary data representation

The reason why today's computers use binary data representation is because its more efficient. This means that digital information can be stored by distinguishing between different values of some continuous physical quantity, such voltage or current. If there are more values which have to be distinguished, there would less separation between adjacent values and less reliable memory.

(b) Explain reasons for groupings bits into memory cells (common cell size: 8 bits = 1 byte)

Memory cell is consisted of 8 bits of information. The purpose of memory cells is represent 1 bit of information. 1 bit = 8 byte. A bit is too smallest data type to represent an information.

(c) Explain the purpose of a memory address

A memory adress represents a number which is assigned to each byte in computer's memory. CPU could then use it to track where data and instructions are stored in RAM.

(d) Explain aligned and unaligned memory acceses.

A memory access is said to be alligned when the data being accessed is n bytes long and the datum adress is n-byte alligned. Otherwise it is said to be unaligned.

(e) Explain big endian and little endian architectures.

A byte in a word can be numbered from left-to-right or right-to-left.

System in which numbering begins at the high oder = big endian

System in which numbering begins at the low oder = little endian

Big Endian

0	J	I	m	
4	S	m	I	T
8	H	0	0	0
12	0	0	0	21
16	0	0	1	4

Little Endian

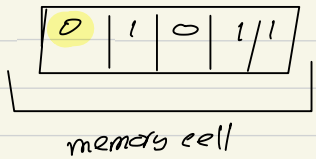
0		m	I	J
4	T	I	m	S
8	0	0	0	H
12	0	0	0	21
16	0	0	1	4

6. Error Detetion and Correction

(a) Explain the terms "memory word", "code word", "data bits", and "redundant (check bits".

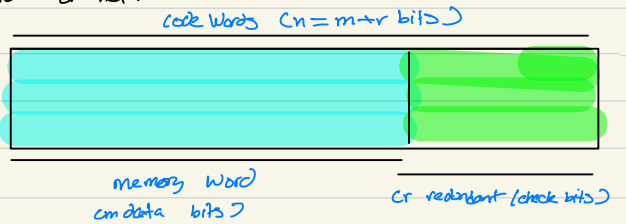
Memory word is consisted with a specific amount of byte. For example in a 32 bit machine a word would have 32 bits

Bit is the smallest measure for information. A bit consists of 8 bytes of information.



0 = 1 bit

Redundant (check bits is the extra bits in the code word to check if code word is valid or not.



(b) Given the numbers of data bits, m, and the number data bits, m, and the number of redundant bits, r, determine the number of possible code words, the number of possible memory words, and the number of valid code words

Floating Point

$n = f \times 10^e$ → way of separating the range from the precise is to express numbers in the familiar scientific notation.

- f → mantissa
- e → positive or negative integer called the exponent

Range is determined by the number of digits in the exponent.

Precision is determined by the number of digits in mantissa.