

Computer Architecture Theory + Lab (CS 305/341)

Assignment 2: Basics of ISA Due Date: 08/09/20 (Theory Assignment 1)

Name: Niraj Mahajan

Roll Number: 180050069

1. In the layered diagram of a computer system why does

- (a) The OS appear above the Architecture layer

Answer: The OS is responsible for defining tasks such as process management, handling IO, Memory management, etc. All these tasks utilize the basic functionalities (eg ALU) defined by the computer architecture. For example, whenever a new process starts, the OS has to copy the code of the process onto the user stack. But this 'copying' is done under the hood by the ISA, without the OS knowing how it is implemented, and hence the OS is a layer above the Architecture.

- (b) The Microarchitecture appear below the Architecture (ISA) ?

Keep your answer brief (few lines) and to the point.

Answer: The Architecture (ISA) is responsible to define the instructions necessary for the processor, like ALU/ Jumps, whereas, the microarchitecture is responsible for defining lower level implementations necessary for supporting the basic operations defined in ISA. Continuing the example in the previous part, we know that the ISA is responsible for implementing the assembly code to load and move code onto the userspace stack, but the implementation of these load instructions is done by the Microarchitecture. Hence, the Microarchitecture appears below the ISA in the layered diagram.

(PTO for Question 2)

2. In the table below, you need to fill in the column labelled “Code Density” (expressed in bytes). Assume that each machine (except for the stack machine) has 16 registers, that the width of the opcode field is 6 bits and that a memory address is 32 bits. Assume, for simplicity, that the only available addressing mode available on all machines is the direct mode. You are only expected to enter values in the “Code Density” column - no explanation is required.

No.	Architectural Style	No. Operands in ALU instr	Max. No. Memory operands in ALU instr	# instructions to compute $x = ab + cde$	Code density	Example
1	Stack	0	0	10	31.5 ceil = 32	-
2	Accumulator-based	1	1	8	31.5 ceil = 32	Intel 8085
3	Register-Memory	2	1	7	33.25 ceil = 34	X86, IBM 360
4	Register-Memory	3	1	7	35.25 ceil = 36	-
5	Memory-Memory	3	3	4	51 ceil = 51	VAX
6	Memory-Memory	2	2	6	52.5 ceil = 53	VAX
7	Register-register	3	0	10	40.5 ceil = 41	MIPS, ARM

3. In the program for the register-register architecture, we used 8 general purpose registers (GPRs). Rewrite the program assuming we had only 4 available GPRs. How many instructions would the program have?

Answer:

The program has 10 instructions.

- a. LD R1 A
- b. LD R2 B
- c. MUL R1 R1 R2
- d. LD R2 C
- e. LD R3 D
- f. MUL R2 R2 R3
- g. LD R3 E
- h. MUL R2 R2 R3
- i. ADD R1 R1 R2
- j. ST R1 X