# Computer Architecture Theory + Lab (CS 305/341)

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

**Total: 100**
**-5% of total score for incorrect submission format**

1. [20 marks] In the layered diagram of a computer system why does
   (a) The OS appear above the Architecture layer
   Solution:
   i. The ISA is the interface between the software and the hardware and so all the layers of software sit above it. Of all software running on a processor, the OS, being system-level software, is closest to the ISA and the hardware. It deals with such abstractions as process, thread, locks and virtual memory.
   ii. The ISA is primarily concerned with instruction set design and the specification of the semantics of each instruction. It supports various OS abstractions - for example, it provides instructions (such as atomic operations used to implement locks/semaphores) and lower-level constructs (the PC and registers which comprise the partial state of a process/thread). In addition, the ISA (together with the microarchitecture) actively cooperate with the OS in the implementation of functions such as memory management, high-speed address translation (through TLBs) and in the protection of user address spaces.

   Grading Scheme:
   10 marks for fully correct solution.
   Breakup: 5 marks for each of the above points.
   Partial marks have been alloted depending on the concepts mentioned.

   (b) The Microarchitecture appear below the Architecture (ISA) ?
   Solution:
   i. In layered designs, it is often the case that a particular layer implements the layer above it. Since the microarchitecture implements the architecture, it is placed just below the ISA layer.
   ii. It is concerned with the different steps (stages) in the processing of instructions. It aims to streamline instruction execution and achieve high instruction throughput by pipelining and scale the memory wall through intricate design of cache memories.

Grading Scheme:
10 marks for fully correct solution.
Breakup: 5 marks for each of the above points.
Partial marks have been alloted depending on the concepts mentioned.

2. [60 marks] 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 | - |
| 2 | Accumulator-based | 1 | 1 | 8 | 31.5 | Intel 8085 |
| 3 | Register-Memory | 2 | 1 | 7 | 33.25 | X86, IBM 360 |
| 4 | Register-Memory | 3 | 1 | 7 | 35.25 | - |
| 5 | Memory-Memory | 3 | 3 | 4 | 51.0 | VAX |
| 6 | Memory-Memory | 2 | 2 | 6/ 7 | 52.5/ 61.25 | VAX |
| 7 | Register-register | 3 | 0 | 10 | 40.5 | MIPS, ARM |

Grading Scheme:

| Case No | Correct number of bytes (full for upper rounded, -2 for lower rounded) | Correct number of bits (instead of bytes) | Other cases |
|---|---|---|---|
| 1 | 8 | 4 | 0 |
| 2 | 8 | 4 | 0 |
| 3 | 10 | 5 | 0 |
| 4 | 8 | 4 | 0 |
| 5 | 8 | 4 | 0 |
| 6 | 10 (for 52.5)/ 8 (for 61.25) | 5/ 4 | 0 |
| 7 | 8 | 4 | 0 |

3. [20 marks] 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?

   Solution:

   ```
   LD R1, A
   LD R2, B
   MUL R1, R1, R2
   LD R2, C
   LD R3, D
   MUL R2, R2, R3
   LD R3, E
   MUL R2, R2, R3
   ADD R1, R1, R2
   ST R1, X
   ```

   10 instructions.

   Grading Scheme:
   20 marks for fully correct solution.
   Breakup:
   5 marks for mentioning correct number of instructions.
   15 marks for fully correct program.

   For each incorrect instruction in the program, 1.5 mark has been deducted (partial marks are allotted considering error propagation).