

Computer Architecture HW #1

Due: Friday, Sept. 4 (5 PM in ITT 305 mailbox or under my office door, ITT 313)

1. Compare zero-, one-, two-, three-address, and the load & store machines by writing programs to compute

$$X = A * B * B + C * D;$$

$$Y = (X + B - D) / (X - C);$$

for each of the five machines (3-address, 2-address, 1-address, 0-address, and load/store). The instructions available for use are as follows. Your program should not wipe out the values in A, B, C, or D.

3 Address	2 Address	1 Address (Accumulator machine)	0 Address (Stack machine)
MOVE (X ← Y)	MOVE (X ← Y)	LOAD M	PUSH M
		STORE M	POP M
ADD (X ← Y + Z)	ADD (X ← X + Y)	ADD M	ADD
SUB (X ← Y - Z)	SUB (X ← X - Y)	SUB M	SUB
MUL (X ← Y * Z)	MUL (X ← X * Y)	MUL M	MUL
DIV (X ← Y / Z)	DIV (X ← X / Y)	DIV M	DIV
		Notes: “SUB M” performs AC = AC - M “DIV M” performs AC = AC / M	Notes: “SUB” performs POP T POP T2 T3 = T2 - T PUSH T3 “DIV” performs POP T POP T2 T3 = T2 / T PUSH T3

Load/Store Architecture - operands for arithmetic operations must be from/to registers. For example, to perform the high-level statement “Z = X - Y” we need the code:

```
LOAD R2, X
LOAD R3, Y
SUB R4, R2, R3
STORE R4, Z
```

2. Assume 8-bit opcodes, 32-bit absolute addressing, 5-bit register numbers, and 32-bit operands. Compute the number of bits needed during program execution for each programs from question 2. Complete the following table.

	3 Address	2 Address	1 Address	0 Address	Load & Store
Number of program bits read during program execution					
Number of bits of data transferred to and from memory					
Total number of bits read and written during program execution					