CS 125 - Lecture 3

Objectives:

- Pseudo code – Motivation and Examples
- Computers are not magic
- Sign up for Turing's Craft
- Laptop setup issues? Post screenshot on Piazza

3. **Representing algorithms?**
(http://userpages.wittenberg.edu/bshelburne/Comp150/Algorithms.htm)

- Use natural languages
- Use formal programming languages
- **Pseudo-Code** - natural language constructs modeled to look like statements available in many programming languages

Pseudo-Code is a numbered list of instructions to perform some task.

1. *ordered sequence of operations*
2. each instruction is computable
3. complete

Three Categories of **Algorithmic Operations:**

1. sequential operations – instructions executed in order
2. conditional "question asking" operations – select from alternatives
3. iterative operations (loops) – repeating a block of instructions

4. Computing a Quiz Average: Pseudo-code to calculate a quiz average

1. get number of quizzes
2. sum := 0
3. count := 0
4. while count < number of quizzes
   - get quiz grade
   - sum = sum + quiz grade
   - count = count + 1
5. average = sum / number of quizzes
6. display average

1. Computer Science Terminology – did your neighbor do the readings?
Discuss with your neighbor what a Computer Scientists means by the following terms and give an example of each:

- fetch-execute cycle
- CPU
- opcode
- operand
- register
- condition code

2. Arithmetic can be done with binary numbers:

https://www.youtube.com/tv#/watch?v=GcDshWmhF4A

$$\begin{array}{r} 11110_2 \\ +\ 00111_2 \\ \hline \end{array}$$

2

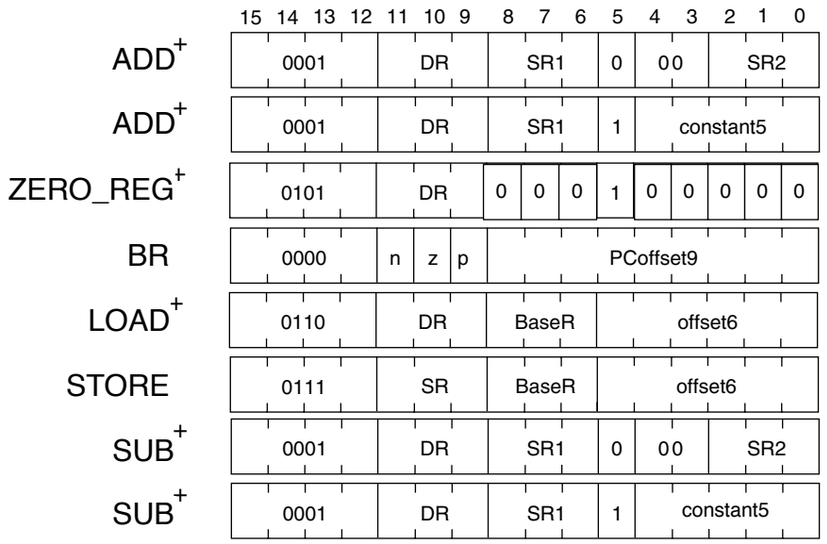
5. Write pseudo-code to print the highest quiz score:

6. A simple machine language:

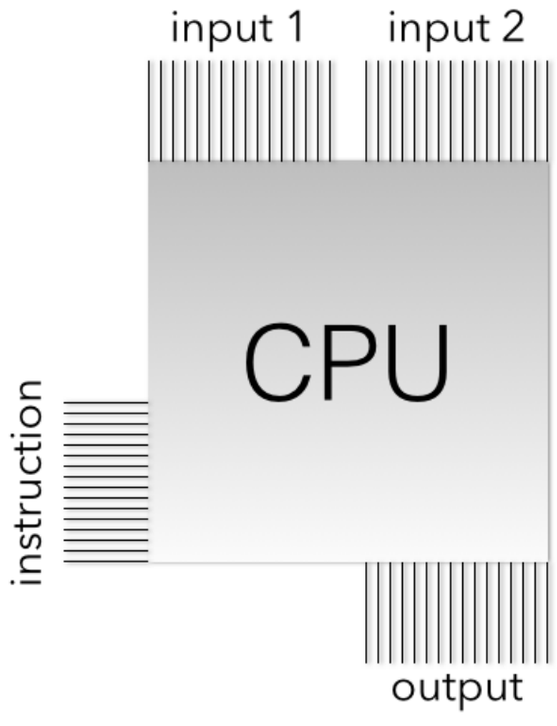
ZERO_REG DESTR – puts a zero in the specified register.
ADD SRCR1 + SRCR2 -> DESTR – add two registers together and write the result in the destination register.
ADD SRCR1 + CONSTANT -> DESTR – add a register to a constant and write the result in the destination register.
SUB SRCR1 - SRCR2 -> DESTR – subtract one register from another and write the result in the destination register.
SUB SRCR1 - CONSTANT -> DESTR – subtract a constant from a register and write the result in the destination register.
LOAD DESTR <- [BASER + CONSTANT] – add the value of a register to a constant to compute a memory address and copy 4 bytes starting at that address to the destination register.
STORE SRCR1 -> [BASER + CONSTANT] – add the value of a register to a constant to compute a memory address and copy the source register to 4 bytes of memory starting at that address.
BR.___ PCOFFSET – the branch instruction specifies a combination of condition codes (n, z, p); if any of the specified condition codes holds a 1, the PC is set to PC + 2 + 2(*PCOFFSET)*. Otherwise PC is set to PC + 2.

For all instructions other than the branch, PC is set to PC + 2. Any instruction that writes a general-purpose register also set the condition code bits: if new value is negative then n=1, else n=0; if new value is zero then z=1, else z=0; if new value is positive then p=1, else p=0.

7. Decoding an instruction:



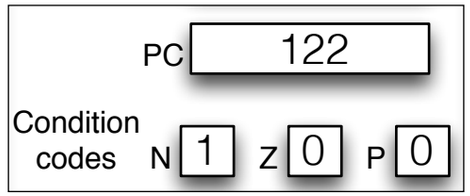
8. Schematic of a CPU:

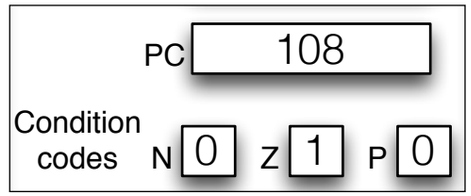


9. Decoding 16 bit-string instructions:

```
0111011001000100

0001100011000010

0101101000100000

0000110000001100
```

10. All about that branch, 'bout that branch …





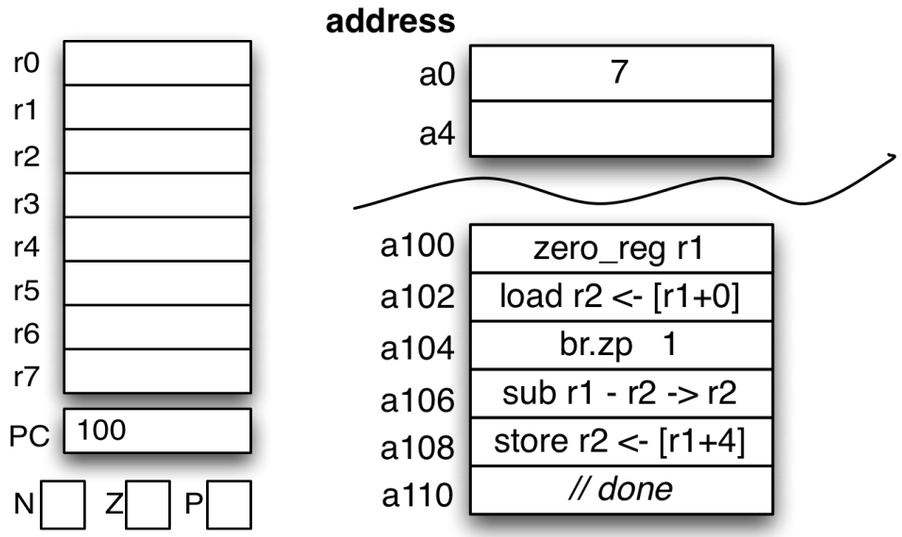
BR.N 6 PC =

BR.NZ -6 PC =

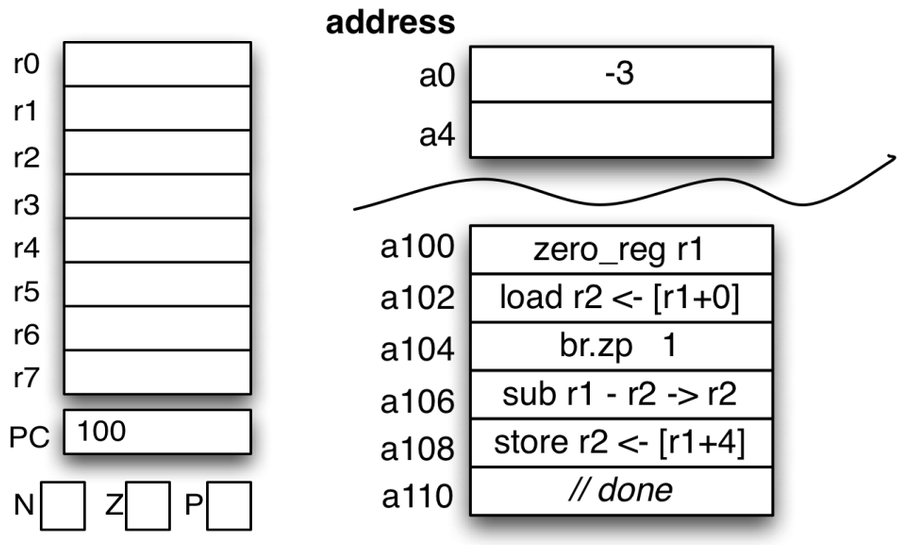
BR.NZP 22 PC =

BR.ZP -10 PC =

11. Execute a machine code program - me:



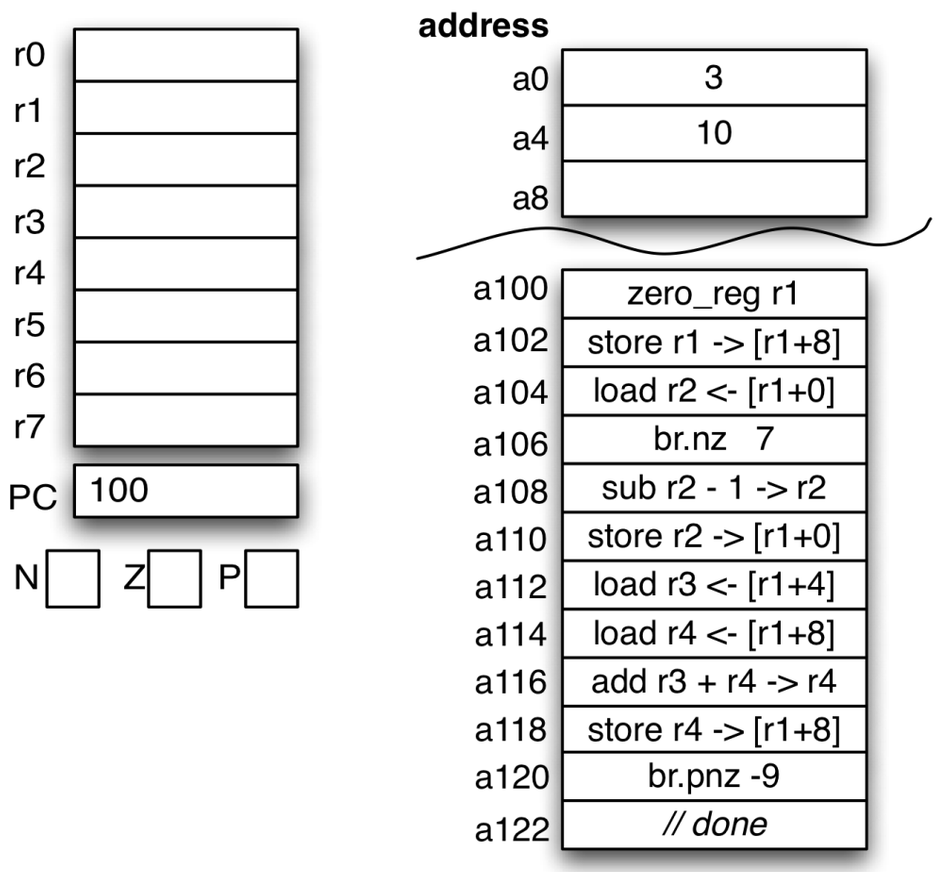
12. Execute a machine code program - you:



13. What does this code do?

Workspace:

14. Execute a machine code program - you:



15. What does this code do?

Workspace: