College of Computer Science
Northeastern University
COM 1722: Freshman Honors Seminar

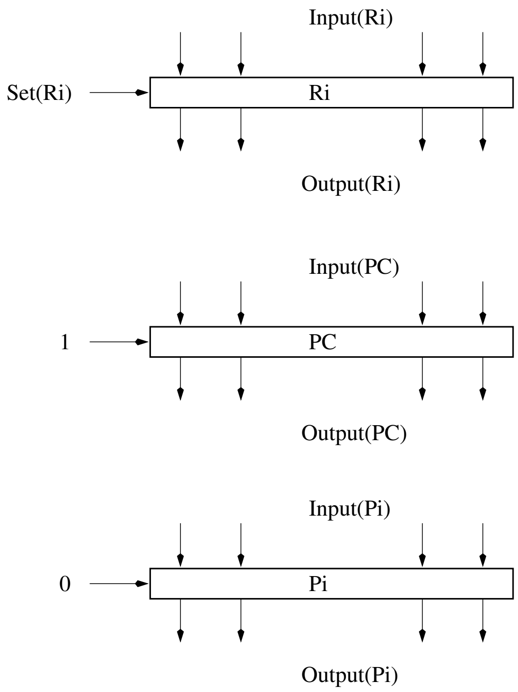
Winter 2003
Handout 7
4 March 2003

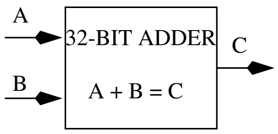
Design of a Simple Processor

Our design will proceed in steps.

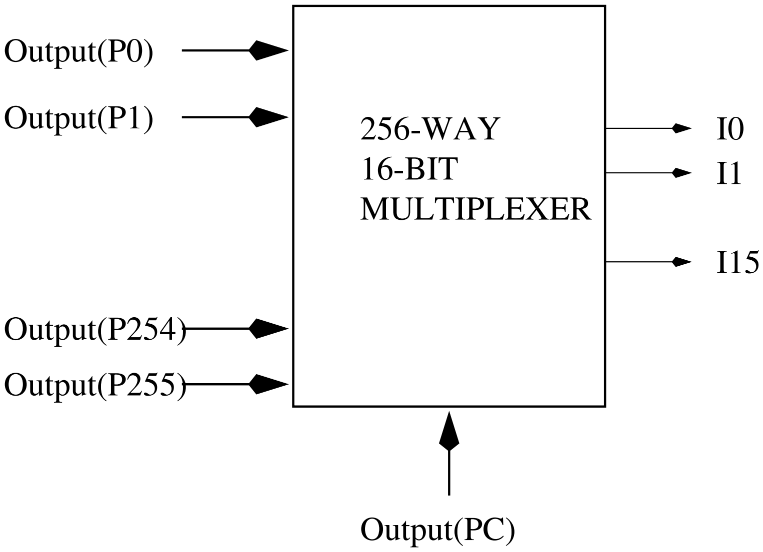
1. We lay out the registers (giving names to the input bits, output bits, and registers for convenience).
2. We design the circuits for extracting the next instruction and incrementing the PC.
3. We design the circuits for implementing the add, negate, load, and jump-if-zero operations.
4. We design the circuit that determines the input and the set bits for the data registers and the PC based on the above circuits.
5. In each of the above circuits, we clearly label the inputs and outputs. These circuits can be put together by matching the appropriate labeled inputs and outputs.

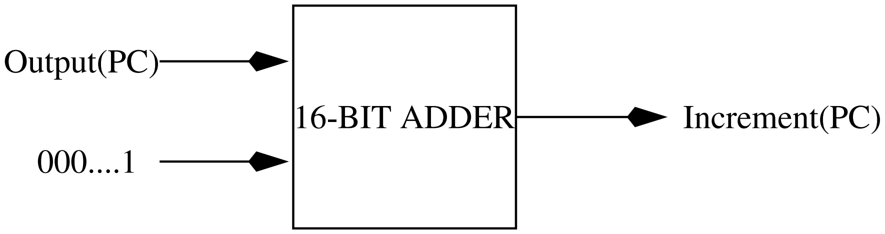
Registers. Here is a layout of the registers. Note that the set bit for PC is always 1 and the set bit of program registers is always 0. So the PC changes at every clock cycle and the program registers never change. The set bits for the data registers will be determined by the instruction.



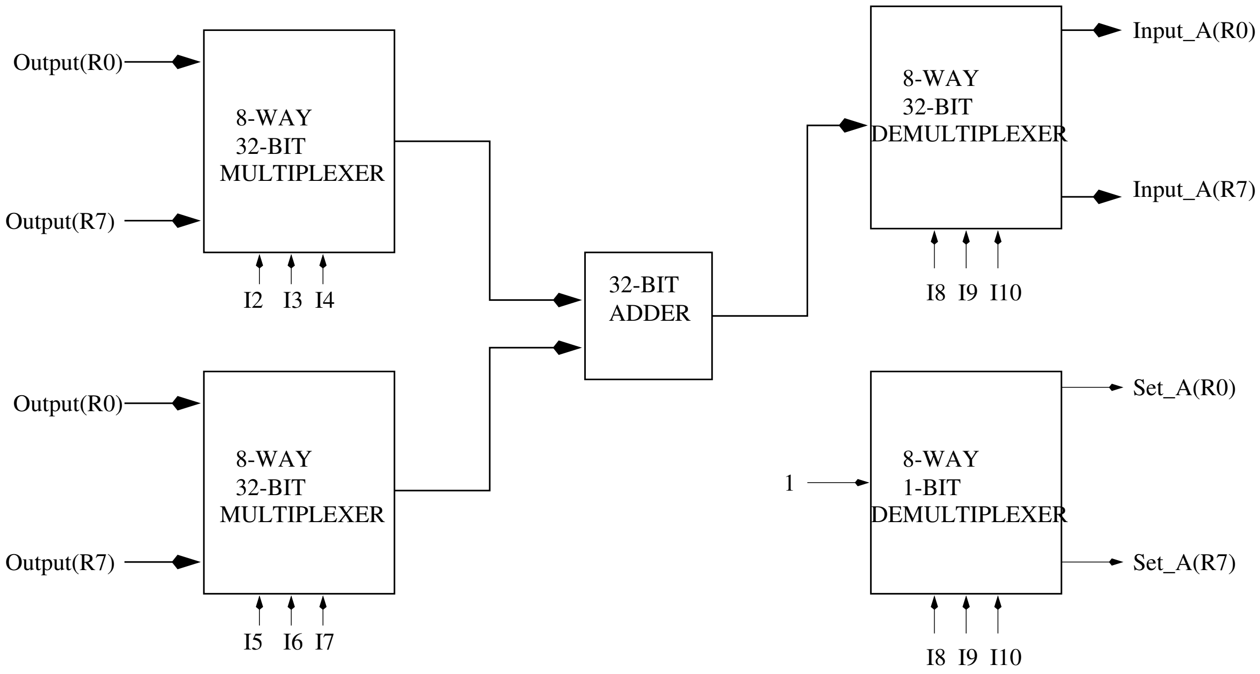


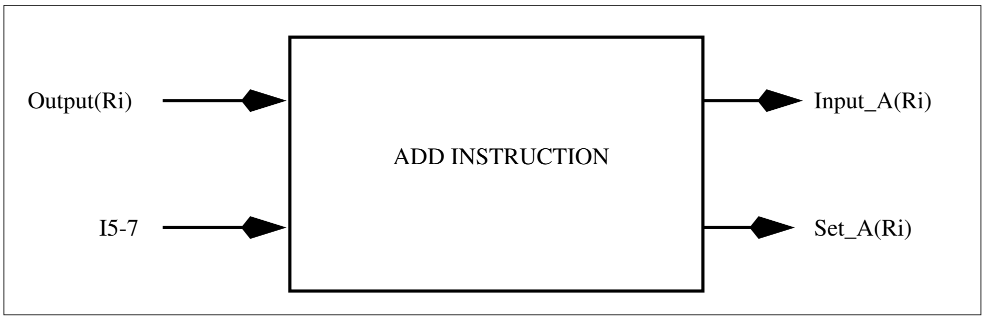
Extracting the instruction and incrementing the PC. The following are the circuits for extracting the next instruction using the PC, and calculating the increment of PC. Note that whether the PC will be actually incremented (or set to something else) will be determined by the instruction, as we will see shortly.



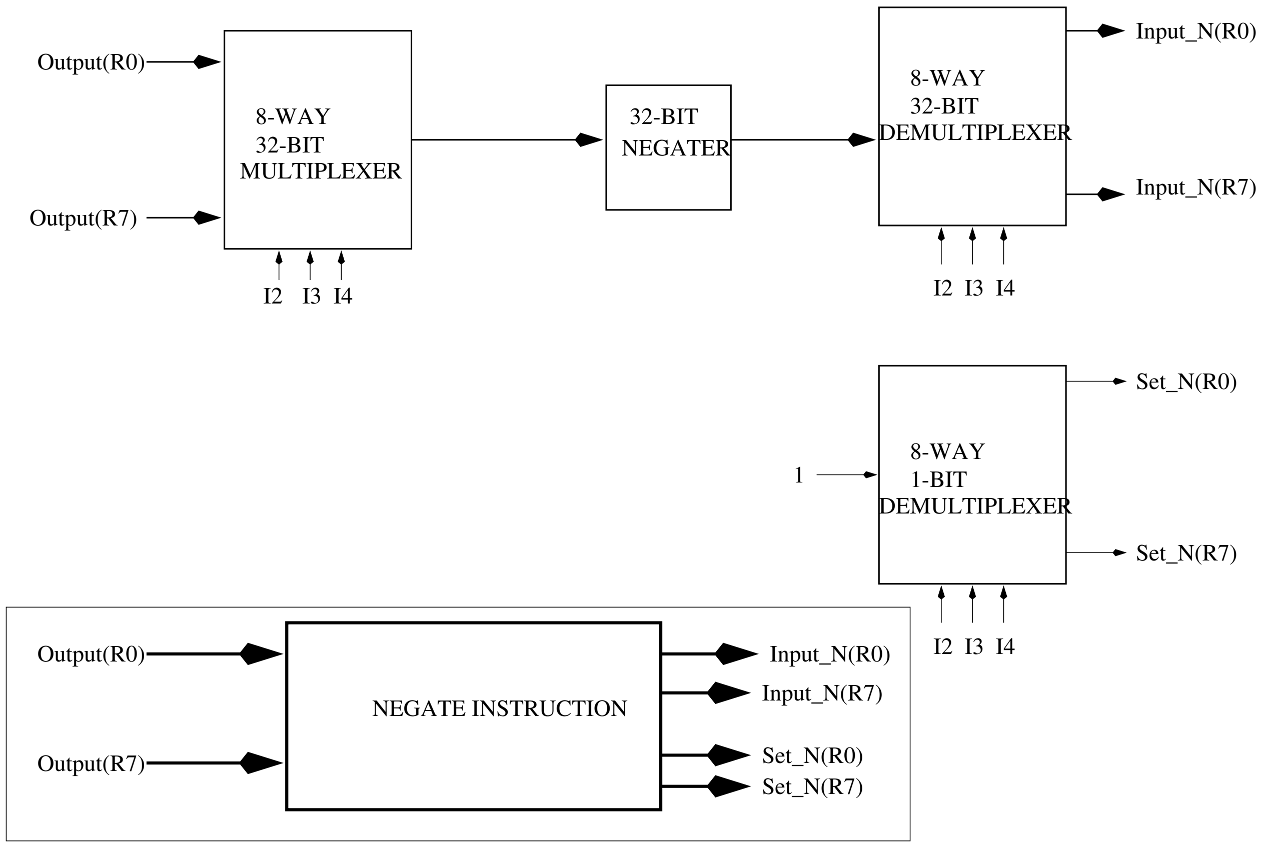


The add instruction. Here is the circuit for implementing the add instruction.

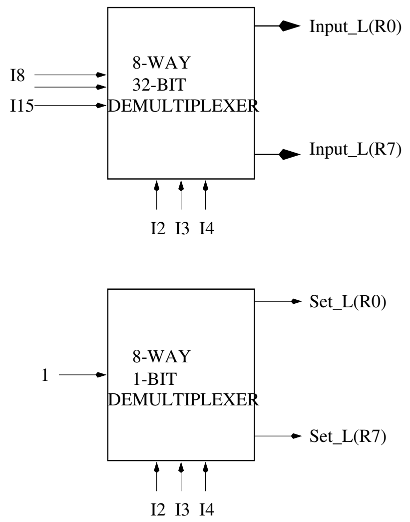


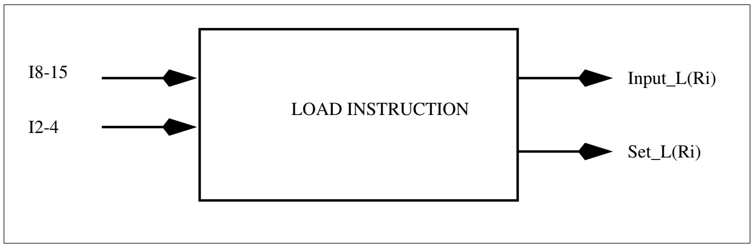


The negate instruction. Here is the circuit for implementing the negate instruction.

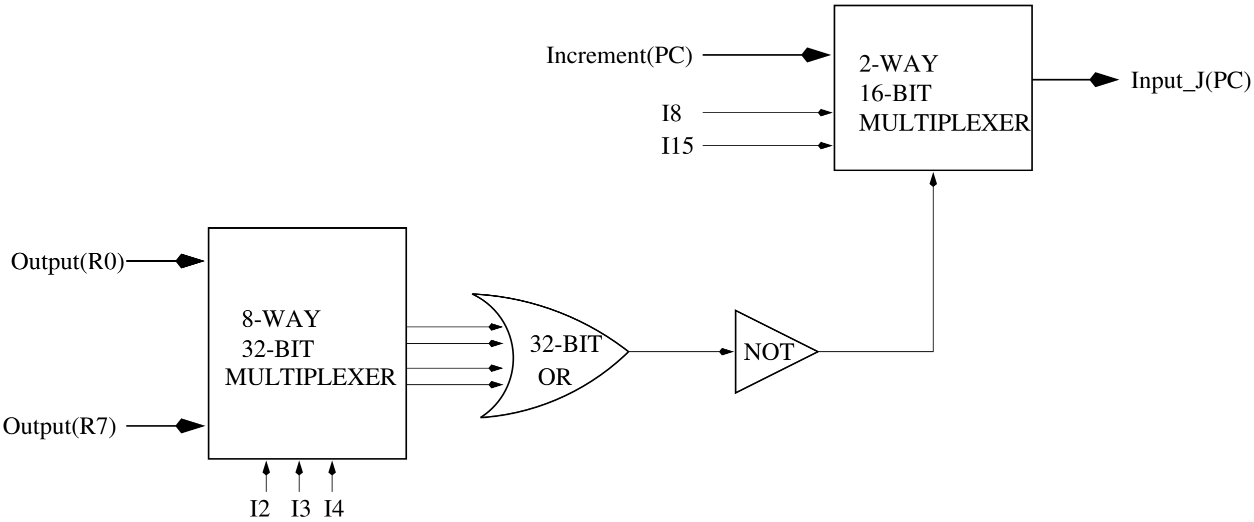


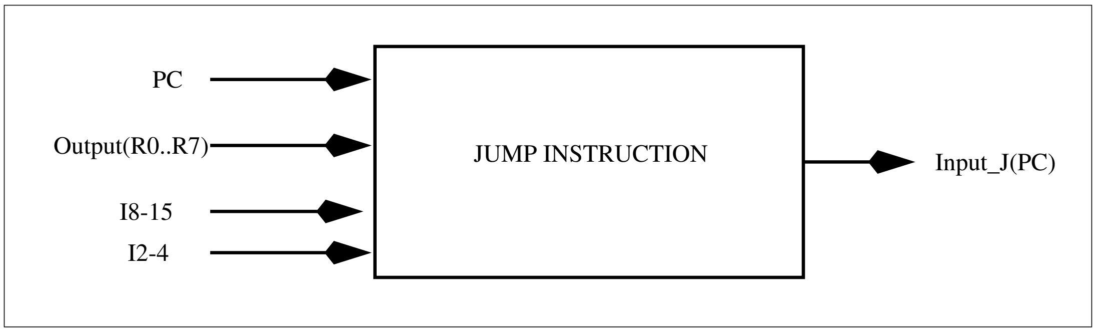
The load instruction. Here is the circuit for implementing the load instruction.





The jump-if-zero instruction. Here is the circuit for implementing the jump-if-zero instruction.





Storing the new values into the data registers and PC. The values computed by the add, negate, load, and jump-if-zero circuits give us potential values that need to be stored in the data registers. Whether the registers need to be updated and what their new values will be is determined by the particular instruction type. The following circuit pushes in the correct value into the registers. It also determines the correct value of the PC. The inputs to the program registers are the same as their outputs (since they are not allowed to change).