Introduction to the TC-201 computer

The simulated machine TC-201 has 16-bit memory registers and 12-bit addresses. Your homework will involve writing a Scheme simulator for this computer, which will be capable of running machine language programs for the TC-201. Data is moved into and out of the random access memory and manipulated in the machine via instructions stored in the memory itself. Thus, the TC-201 is a stored program computer, also called a Von Neumann computer. (Compare with "Harvard Architecture" in which program and data occupy separate memories.)

Each instruction for the TC-201 is represented by a pattern of 16 bits. The leftmost 4 bits of the instruction are the opcode, which specifies which operation is to be done. With 4 bits, there are 16 possible different opcodes, with integer values of 0 through 15.

The remaining 12 bits of an instruction are the address, which is used to specify the memory location to be used in the operation. This is a single-address format for instructions. If two values are required (for example, for ADD), the other value is in another 16-bit register called the accumulator, abbreviated ACC. We now describe the effects of each type of instruction.

If the opcode is 0, the instruction is a HALT; execution of a HALT instruction causes the computer to stop. The address portion of the HALT instruction is ignored.

If the opcode is 1, the instruction is a LOAD. For example, the instruction:

```
0001 0000 0000 0011
----                 opcode bits are 0001, which is 1
     ---- ---- ----  address bits are 0000 0000 0011, which is 3
```

may be represented symbolically as

```
LOAD 3
```

because the opcode is LOAD, and the address is 3. The effect of this instruction is to copy the contents of memory register 3 into ACC (the accumulator).

The opcode 2 denotes a STORE instruction, for example:

```
0010 0000 0000 1010      STORE 10
```

The effect of this instruction is to copy the contents of ACC into memory register 10, overwriting the previous contents of memory register 10.

The opcode 3 denotes an ADD instruction, for example:

```
0011 0000 0001 0100      ADD 20
```

The effect of this instruction is to add the contents of memory register 20 to the contents of ACC. notation.) The sum is in ACC and the contents of register 20 are unchanged. The Arithmetic Error Bit (AEB) is set to 1 if the sum produced a result that cannot be correctly represented in 16 bits; otherwise, the AEB is set to 0.

Opcode 4 is the SUBTRACT instruction, which is analogous to the ADD instruction. The contents of the memory register addressed by the address of the instruction are subtracted from ACC. The difference

is in ACC and the contents of the memory register are unchanged. The AEB is set to 1 if the difference produces a result that cannot be correctly represented in 16 bits; otherwise, the AEB is set to 0.

Opcode 5 is the INPUT instruction, used to read a number typed in by the user and put its value in the ACC.

```
0101 0000 0000 0000      INPUT
```

This instruction reads in a number from the user and puts its value in the ACC. The address field of the instruction is not used and the contents of memory are unchanged.

Opcode 6 is the OUTPUT instruction, used to print out a number from the ACC for the user.

```
0110 0000 0000 0000      OUTPUT
```

This instruction prints out the value of the number in the ACC. The address field of the instruction is not used and the contents of the ACC and memory are unchanged.

Opcode 7 is the JUMP instruction, an unconditional jump. After it is executed, the next instruction that the machine executes is the one in the register addressed by the address of the jump instruction. In addition to the registers already described, the TC-201 has another register, the program counter, or PC. The PC holds the address in memory of the next instruction to be executed. The effect of the JUMP instruction is to copy the address field of the instruction into the PC.

```
0111 0000 0001 1110      JUMP 30
```

When this instruction is executed, it causes the machine to begin executing instructions starting with the instruction in register 30.

Opcode 8 is the SKIPZERO instruction, a conditional skip. This instruction tests whether the ACC holds the number 0, and the next instruction is skipped if so. If the ACC does not hold 0, then the next instruction is executed as usual. The address field of the instruction is not used.

Opcode 9 is the SKIPPOS instruction, another conditional skip. If the ACC holds a positive number, then the next instruction is skipped. If the ACC does not hold a positive number, then the next instruction is executed as usual. The address field of the instruction is not used.

Opcode 10 is the SKIPERR instruction, another conditional skip. If the AEB holds a 1, then the next instruction is skipped. If the AEB holds a 0, then the next instruction is executed as usual. In either case, the AEB is set to 0. The address field of the instruction is not used.

For now, the rest of the opcodes (11 through 15) are unused.