Instruction Register	
IRin	Inputs into the Instruction Register to Decode
IRDFout	Outputs the Data from the Decoded Instruction
IRAFout	Outputs the Address from the Decoded Instruction
SELECT Other	Selects the middle 8 bits instead of the last 8 bits

Arithmetic Logic Unit	
SELECT Y	Selects the Y Register for the ALU
SELECT 4	Selects 4 for the ALU
Yin	Inputs into the Y Register for ALU
Cout	Outputs the Carry Flag into the Adders
SET C	Sets the Carry Flag to 1 into the Adders
ALU OP0	Sets the LSb of the ALU Operation to Decode which Operation to do
ALU OP1	Sets the middle bit of the ALU Operation to Decode which Operation to do
ALU OP2	Sets the MSb of the ALU Operation to Decode which Operation to do
Zin	Inputs into the Z Register for the ALU Result
Zout	Outputs the Z Register for the ALU Result

ALU Operations	
0 0 0	ADD
0 0 1	SUB
0 1 0	Left blank for the Students to Implement
0 1 1	Left blank for the Students to Implement
1 0 0	Left blank for the Students to Implement
1 0 1	Left blank for the Students to Implement
1 1 0	Left blank for the Students to Implement
1 1 1	Left blank for the Students to Implement

Main Memory	
MARin	Inputs to the Memory Address Register
MARout	Outputs the Memory Address Register
MDRin	Inputs to the Memory Data Register
MDRout	Outputs the Memory Data Register
READ	Reads the Address in MAR and stores the data in MDR
WRITE	Writes the data in MDR into address in MAR

Program Counter	
PCin	Inputs to the Program Counter to indicate the Address of the Instruction
PCout	Outputs the Program Counter or the current Instruction Address
RESET PC	Resets the Program Counter to 0

Register	
REGin	Inputs to the Register (Decoder is implemented to select which register)
REGout	Outputs to the Register (Decoder is implemented to select which register)
SWAP OUT	Allows to output the 1st Operand instead of input

-

Opcode	Instruction	Example Instruction	Example Opcode
0	HALT	HALT	0000 0000 0000 0000
1	MOV REG, REG	MOV R1, R2	0001 0001 0000 0010
2	MOV REG, POINTER	MOV R1, [R2]	0010 0001 0000 0010
3	XOR REG, REG	XOR R4, R5	0011 0100 0000 0101
4	—	—	—
5	NOT REG	NOT R1	0101 0001 0000 0000
6	OR REG, REG	OR R4, R5	0110 0100 0000 0101
7	MOV REG, IMMEDIATE	MOV R0, 0x05	0111 0000 0000 0101
8	ADDC REG, REG	ADDC R4, R5	1000 0100 0000 0101
9	INC REG	INC R1	1001 0001 0000 0000
10	—	—	—
11	ADD REG, REG	ADD R3, R1	1011 0011 0000 0001
12	MOV REG, ADDRESS	MOV R3, [0x02]	1100 0011 0000 0010
13	SUB REG, REG	SUB R3, R1	1101 0011 0000 0001
14	MOV POINTER, REG	MOV [R1], R2	1110 0001 0000 0010
15	MOV ADDRESS, REG	MOV [0x02], R3	1111 0000 0010 0011

- **ALU OPERATIONS**
  - **ADD 0 (000)**
  - **SUB 1 (001)**
  - **XOR 2 (010)**
  - **NOT 3 (011)**
  - **OR 4 (100)**

Opcode 0 HALT

- Provided already

Opcode 1 MOV REG, REG

- REGout, REGin, END

Opcode 2 MOV REG, POINTER

- Traditional Microcode
  - REGout, MARin, READ, WMFC
  - MDRout, REGin, END
- Minecraft Microcode
  - REGout, MARin, READ
  - MDRout, REGin, END

Opcode 3 XOR REG, REG

- Traditional Microcode
    - REGout, Yin
    - REGout, SELECT Y, XOR, Zin
    - Zout, REGin, END
- Minecraft Microcode
    - REGout, Yin
    - SWAP OUT, REGout, SELECT Y, ALU OP1 (010), Zin
    - Zout, REGin, END
    - This will activate ALU Operation 010 = 2 (XOR)

Opcode 5 NOT REG

- Traditional Microcode
    - REGout, NOT, Zin
    - Zout, REGin, END
- Minecraft Microcode
    - SWAPout, REGout, ALU OP0 (001), ALU OP1 (010), Zin
    - Zout, REGin, END
    - This will activate ALU Operation 011 = 3 (NOT)

Opcode 6 OR REG, REG

- Traditional Microcode
    - REGout, Yin
    - REGout, SELECT Y, OR, Zin
    - Zout, REGin, END
- Minecraft Microcode
    - REGout, Yin
    - SWAP OUT, REGout, SELECT Y, ALU OP2 (100), Zin
    - Zout, REGin, END
    - This will activate ALU Operation 100 = 4 (OR)

Opcode 7 MOV REG, IMMEDIATE

- IRDFout, REGin, END

Opcode 8 ADDC REG, REG

- Traditional Microcode
  - REGout, Yin
  - REGout, SELECT Y, ADD, Zin
  - Zout, REGin, END
- Minecraft Microcode
  - REGout, Yin
  - SWAP OUT, REGout, SELECT Y, C out, Zin
  - Zout, REGin, END
  - Not setting any ALU Operation to activate 000 = 0 (ADD)

Opcode 9 INC REG

- Traditional Microcode
  - REGout, SET Carry in, ADD, Zin,
  - Zout, REGin, END
- Minecraft Microcode
  - SWAPout, REGout, SET C, Zin
  - Zout, REGin, END
  - Not setting any ALU Operation to activate 000 = 0 (ADD)

Opcode 11 ADD REG, REG

- Traditional Microcode
  - REGout, Yin
  - REGout, SELECT Y, ADD, Zin
  - Zout, REGin, END
- Minecraft Microcode
  - REGout, Yin
  - SWAP OUT, REGout, SELECT Y, Zin
  - Zout, REGin, END
  - Not setting any ALU Operation to activate 000 = 0 (ADD)

Opcode 12 MOV REG, ADDRESS

- Traditional Microcode
  - IRAFout, MARin, READ, WMFC
  - MDRout, REGin, END
- Minecraft Microcode
  - IRAFout, MARin READ
  - MDRout, REGin, END

Opcode 13 SUB REG, REG

- Traditional Microcode
    - REGout, Yin
    - REGout, SELECT Y, SUB, Zin
    - Zout, REGin, END
- Minecraft Microcode
    - REGout, Yin
    - SWAP OUT, REGout, SELECT Y, ALU OP0 (001), , Zin
    - Zout, REGin, END
    - Not setting any ALU Operation to activate 001 = 1 (SUB)

Opcode 14 MOV POINTER, REG

- Traditional Microcode
    - REGout, MDRin
    - REGout, MARin, WRITE, WMFC, END
- Minecraft Microcode
    - REGout, MDRin
    - SWAP OUT, REGout, MARin, WRITE, END

Opcode 15 MOV ADDRESS, REG

- Traditional Microcode
    - REGout, MDRin
    - IRAFout, MARin, WRITE, WMFC, END
- Minecraft Microcode
    - REGout, MDRin
    - IRAFout, SELECT OTHER, MARin, WRITE, END
    - SELECT OTHER lets you select the middle 8 bits of the IR
    - F00F = F < instruction 00 < address < F register