Opcode	Instruction	Registers	Bits			
0 (0000)	Register to Register	$R_A \rightarrow R_Y$	Op (4)	R_A (4)	X (4)	R_Y (4)
1 (0001)	Immediate to Register	$I \rightarrow R_Y$	Op (4)	X (8)		R_Y (4)
			Immediate (16)			
2 (0010)	Register to Memory	$R_A \rightarrow M_{RB}$	Op (4)	R_A (4)	R_B (4)	X (4)
3 (0011)	Memory to Register	$M_{RA} \rightarrow R_Y$	Op (4)	R_A (4)	X (4)	R_Y (4)
4 (0100)	Bitwise OR	$R_A \mid\mid R_B \rightarrow R_Y$	Op (4)	R_A (4)	R_B (4)	R_Y (4)
5 (0101)	Bitwise NOT	$\sim R_A \rightarrow R_Y$	Op (4)	R_A (4)	X (4)	R_Y (4)
6 (0110)	Bitwise AND	$R_A \&\& R_B \rightarrow R_Y$	Op (4)	R_A (4)	R_B (4)	R_Y (4)
7 (0111)	Bitwise XOR	$R_A \wedge R_B \rightarrow R_Y$	Op (4)	R_A (4)	R_B (4)	R_Y (4)
8 (1000)	Add	$R_A + R_B \rightarrow R_Y$	Op (4)	R_A (4)	R_B (4)	R_Y (4)
9 (1001)	Subtract	$R_A - R_B \rightarrow R_Y$	Op (4)	R_A (4)	R_B (4)	R_Y (4)
A (1010)	Logical NOT	$!R_A \rightarrow R_Y$	Op (4)	R_A (4)	X (4)	R_Y (4)
B (1011)	Logical Shift Right	$R_A >> 1 \rightarrow R_Y$	Op (4)	R_A (4)	X (4)	R_Y (4)
C (1100)	Push* (Details below)	(R_A or PC+2) → Top stack	Op (4)	R_A (4)	Op_2 (1)	X (7)
D (1101)	Pop** (Details below)	Top stack → (R_Y or PC)	Op (4)	X (4)	Op_2 (1)	X (3)	R_Y (4)
E (1110)	Unconditional Jump	$R_A \rightarrow$ PC	Op (4)	R_A (4)	X (8)		
F (1111)	Conditional Jump*** (Details below)	If (R_B ?? 0): $R_A \rightarrow$ PC	Op (4)	R_A (4)	R_B (4)	Flags (4)	

*Push:

Op_2	Behavior
0	$R_A \rightarrow$ Top stack
1	PC+2 $\rightarrow$ Top stack

A push instruction with Op_2 = 1 is essentially the first part of a function call: We store PC+2 so that PC+1 can be the jump instruction and PC+2 will be the instruction memory return address.

**Pop:

Op_2	Behavior
0	Top stack $\rightarrow R_Y$
1	Top stack $\rightarrow$ PC

***Conditional Jump:

Flags	Comparison
0000	N/A (Instruction functions as a NOP)
0001	$R_B == 0$
001X	R_B != 0
01XX	$R_B < 0$
1XXX	$R_B > 0$