Assignment 03 - Stack machines

Task 1-

1:- To add two integers together, we first should add them to the stack (push), ~~push the operands~~ pop twice to get the operands, and push the result back into the stack. This gives us a total of 5 references.

2:- The reason is that, while stack machines specify operands on the top of the stack, ~~and~~ only ~~specify~~ uses the values of this operands, register machines also need to specify the source and destination registers. We can say that instructions need more than 6bits, but the lenght will be determined by factors like architecture, even going to infinity in some models

3:- Register machines save the state of the processor and curent registers in a set of registers, while stack machine just save the stack and program pointers to the stack, which ~~allows it~~ avoids having to save the state of registers. This usually give them ~~an~~ advantage.

Task 2-

1:- $4\cdot(7+8\cdot9)-1$ → 1 4 7 8 9 MUL ADD MUL SUB

2.- $(96-(4+44\cdot(3-1)+7)\cdot25$

96 25 7 4 44 3 1 SUB MUL ADD ADD MUL SUB

3.- $(5^3/(2+3))/5$ → 5 5 5 5 2 3 ADD MUL MUL DIV DIV

Task 3-

1.-

4 → 000100

2 → 000010

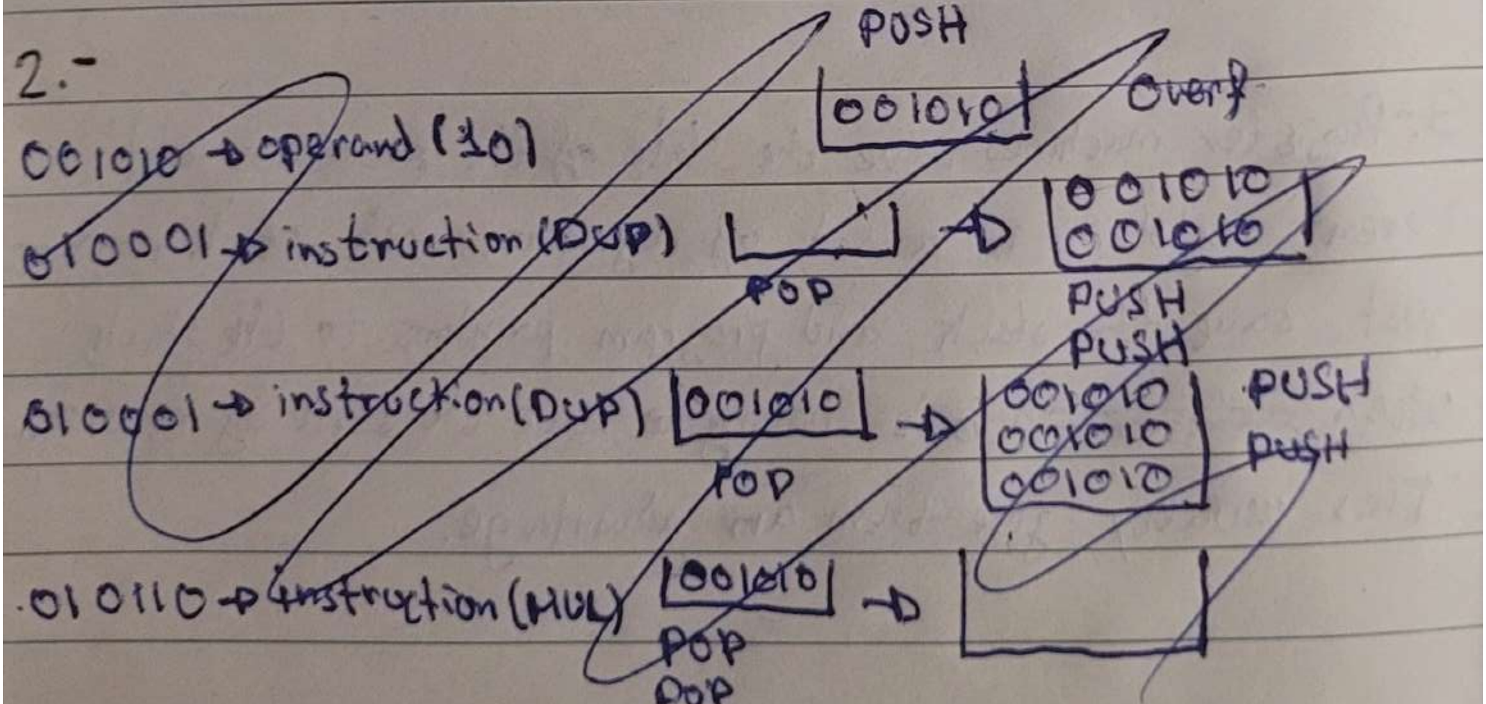
3 → 000011

• → 010110

+ → 010100

/ → 010111

2.-



001010 → operand (10)

010001 → instruction (DUP)

010001 → instruction (DUP)

010110 → instruction (MUL)

Representation	Meaning	Stack	Overflow flag
001010	10	001010 PUSH	F
010001	DUP	POP → 001010 PUSH 001010 PUSH	F
010001	DUP	001010 POP → 001010 PUSH 001010 PUSH	F
010110	MUL	001010 POP POP → 001111 PUSH 001010 PUSH	T (Result 15)
011111	XOR	POP POP → 000101 PUSH	F
000100	4	000100 000101 PUSH	F
011011	SHR	POP → 001010 PUSH POP	F
000100	4	000100 001010 PUSH	F
011001	MOD	POP → 000010 PUSH	F
000110	6	000110 000010 PUSH	F
011000	EXP	POP → 001111 PUSH POP	T
100010	SPACE	100010 001111 PUSH	F
110110	S	110110 100010 001111 PUSH	F
101000	E	101000 110110 100010 001111 PUSH	F
110101	R	110101 101000 110110 100010 001111 PUSH	F
010000	STP	unchanged	F