

Robolab : Assignment 3  
by  
IHSAN AHMAD SIDDIQUEE

Task 1

1. For a simple 'ADD' operation of two integers, the data cache is referred 'five' times in a stack machine.

e.g.  $A + B$ , where A, B are two operands

- steps:
- ① Push 'A' to memory.
  - ② Push 'B' to memory.
  - ③ Pop 'A' from memory.
  - ④ Pop 'B' from memory.
  - ⑤ 'A+B' result push to memory.

So, for a total of 5 data cache references.

2. common stack machine instructions can easily be stored into 6 bit or less, whereas register machines requires multiple registers to do the same instruction per ALU to select the operands. The register machines average about 16 bits per instruction including the operands. Register machines uses wider offset of fields to push and pull operands code for instruction, whereas, stack machine uses only one operand code for the instruction.
3. Register machines saves the register on a stack and may go through interrupt handler code. Stack machines respond to interrupts quickly because most of the parameters already on a stack and there's no need to push them from there to store in a different stack. hence, having an advantage over the Register machines.

## Task 2

1.  $4 * (7 + 8 * 9) - 1$
2.  $(96 - (4 + 44 * (3 - 1) + 7) * 25)$
3.  $(5^3 / (2+3)) / 5$

$$4 * (7 + 8 * 9) - 1$$

$\rightarrow 4 \ 7 \ 8 \ 9 * + * 1 -$

$$(96 - (4 + 44 * (3 - 1) + 7) * 25)$$

$\rightarrow 96 \ 4 \ 44 \ 3 \ 1 - * + 7 + 25 * -$

$$(5^3 / (2+3)) / 5$$

$\rightarrow 5 \ 3 ^ 2 \ 3 + / 5 /$

1.  $4 * (7 + 8 * 9) - 1$

Input	stack	RPN
4	4	
*	*	4
7	7*	4
8	8*	47
+	8*	47
8	8*	478
*	8*	478
9	8*	4789
)	8*	4789*
-	8*	4789*-
1	-	4789*-1-

3.  $(5^3 / (2+3)) / 5$

$\rightarrow (5 * 5 * 5 / (2+3)) / 5$

Input	stack	RPN
(	(	
5	5	5
*	5*	5
5	5*	55
*	5*	55*
5	5*	55*5
/	55*	55*5*
(	55*	55*5*
2	55*	55*5*2
+	55*	55*5*2
3	55*	55*5*23
)	55*	55*5*23+
)		55*5*23+1
/	1	55*5*23+1
5	1	55*5*23+1/5

2.  $(96 - (4 + 44 * (3 - 1) + 7) * 25)$

Input	stack	RPN
(	(	
96	(	96
-	(-	96
(	(-	96
4	(-l	964
+	(-l+	964
44	(-l+	9644
(	(-l+*	96444
3	(-l+*	964443
-	(-l+*	964443
1	(-l+*	9644431
)	(-l+*	9644431-
+	(-l+*	9644431-*+
7	(-l+	9644431-*7
)	(-l+)	9644431-*7+
*	(-l+)	9644431-*7+
25	(-l+)	9644431-*7+25
)	(-l+)	9644431-*7+25*-

### Task 3

1. 4 2 2 3 \* + \* 2 /

4	2	2	3	*	+	*	2	/
000100	000010	000010	000011	010110	010100	010110	000010	010111

Instruction	Binary Representation.	stack
4	0 0 0 1 0 0	
2	0 0 0 0 1 0	
2	0 0 0 0 1 0	
3	0 0 0 0 1 1	
*	0 1 0 1 1 0	2 * 3 or, 6
+	0 1 0 1 0 0	2 + 6 or, 8
*	0 1 0 1 1 0	4 * 8 or, 32
2	0 0 0 0 1 0	
/	0 1 0 1 1 1	32 / 2 or, 16 → final message.
STP	0 1 0 0 0 0	End of Execution.

Current content : 16

2. Binary Representation.	Abbreviation	Stack
0 0 1 0 1 0	10	10
0 1 0 0 0 1	DUP	10 10 (10 Duplicate 10)
0 1 0 0 0 1	DUP	1010 10 (10 Duplicate 10)
0 1 0 1 1 0	MUL	10 100 (10 * 10)
0 1 1 1 1 1	XOR	110 (10 XOR 10)
0 0 0 1 0 0	4	110 4 (110 bin 110 1110)
0 1 1 0 1 1	SHR	6 (110110, 110 Dec 6)
0 0 0 1 0 0	4	64
0 1 1 0 0 1	MOD	2 (6 MOD 4 → 2)
0 0 0 1 1 0	6	2 6
0 1 1 0 0 0	Exp	64 (2^6 → 64)
1 0 0 0 1 0	Space	64 (Add space)
1 1 0 1 1 0	S	69S
1 0 1 0 0 0	E	69SE
1 1 0 1 0 1	R	69SER → final message.
0 1 0 0 0 0	STP	End of Execution.

Current content : 69 SER