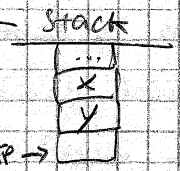
PROJECT (7) REFERENCE

[VM CODE]

contains BOTH INT or BOOL



first/top-stack cell

POP

PUSH

RAM	
0	SP
1	LCL
2	ARG
3	THIS
4	THAT
5..12	TEMP
13..15	General Purpose Registers
16..255	STATIC SEGMENT
256..2047	STACK
2048..16383	HEAP
16384..24576	I/O (16384 SCREEN, 24576 KBD)
24577..32767	UNUSED

R13 USED IN MY IMPLEMENTATION OF "POP", TO MOMENTARILY MEMORIZE THE ADDRESS OF THE CELL TO POP TO

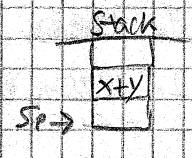
R14 USED IN MY IMPLEMENTATION OF "RETURN" TO MOMENTARILY MEMORIZE RETURN-ADDRESS

R15 USED IN MY IMPLEMENTATION OF "RETURN" TO MOMENTARILY MEMORIZE LCL.

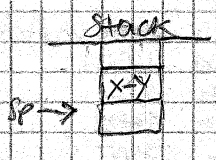
(A) COMMANDS

ARITHMETIC LOGIC COMMANDS

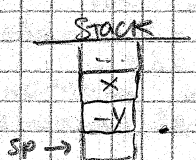
ADD



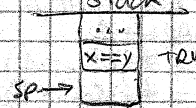
SUB



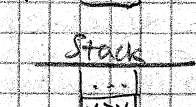
NEG



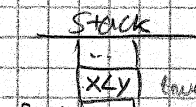
EQ



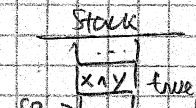
GT



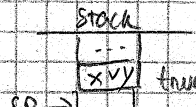
LT



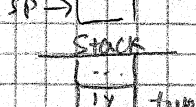
AND



OR



NOT



(B) COMMANDS

MEMORY SEGMENTS COMMANDS

PUSH/POP <index>

- LOCAL
- ARGUMENT
- THIS
- THAT
- CONSTANT (ONLY PUSH)
- STATIC → static variables
- POINTER → changes THIS/THAT
- TEMP → RAM[5..12]

push i ⇒ addr = *SP + i, *SP = *addr, SP++

pop i ⇒ addr = *SP + i, SP--, *addr = *SP

push constant i ⇒ *SP = i, SP++

push static i ⇒ ∀ foo.vm, foo.1, foo.2, foo.3 etc. variables are created and used

pop static i

push pointer 0/1 ⇒ *SP = THIS/THAT, SP++

pop pointer 0/1 ⇒ SP--, THIS/THAT = *SP

(C) COMMANDS →

(D) COMMANDS

(C) COMMANDS — BRANCHING COMMANDS

LABEL <label> → just creates a label

GOTO <label> → Unconditional jump to label

IF-GOTO <label> → CONDITIONAL JUMP (IF LAST STACK ELEMENT IS ≠0 JUMPS)

(D) COMMANDS — FUNCTION COMMANDS

FUNCTION <functionName> <nVars> → 1. LABEL <functionName>
2. PUSH 0 (AS MANY TIMES AS nVars)

CALL <functionName> <nArgs> → requires the nArgs to be pushed just before

1. push returnAddress
2. push LCL
3. push ARG
4. push THIS
5. push THAT
6. ARG = SP-5-nArgs
7. LCL = SP
8. goto functionName
9. label (functionName <callNumber>$returnAddress)

(1.–5. = SAVE OLD FRAME)

RETURN → requires last element on the stack to be return value.

1. returnAddress = *(LCL-5)
2. POP ARGUMENT 0 (pops return value to ARG 0)
3. SP = ARG+1
4. THAT = *(LCL-1)
5. THIS = *(LCL-2)
6. ARG = *(LCL-3)
7. LCL = *(LCL-4)
8. goto (functionName <callNumber>$returnAddress)

(4.–7. = REINSTATE OLD FRAME)