```
push constant 7
push constant 8
add
```

Virtual machine code

VM Translator

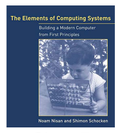
```
@7
D=A
@SP
AM=M+1
A=A-1
M=D

@8
D=A
@SP
AM=M+1
A=A-1
M=D

@SP
AM=M-1
D=M
A=A-1
M=D+M

(END)
@END
0;JMP
```

Assembly instructions



Chap 7

VM to Hack Translator

9 arithmetic / logical commands

1. add
2. sub
3. neg
4. eq
5. gt
6. lt
7. and
8. or
9. not

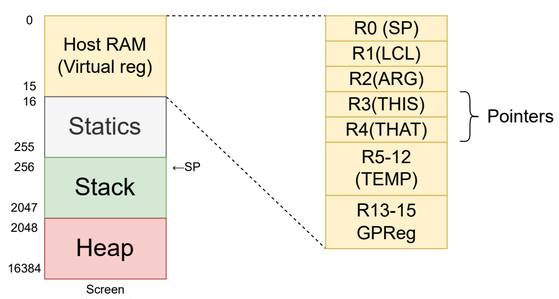
2 memory access commands, 8 memory segments

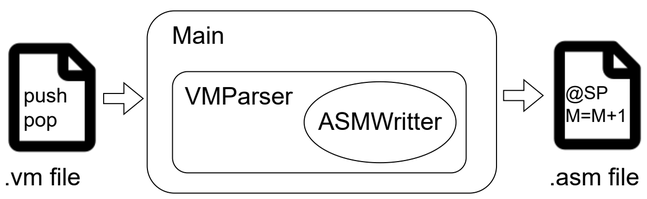
1. push
2. pop

1. constant
2. static
3. local
4. argument
5. pointer
6. temp
7. this
8. that

stack

Hack virtual machine segments

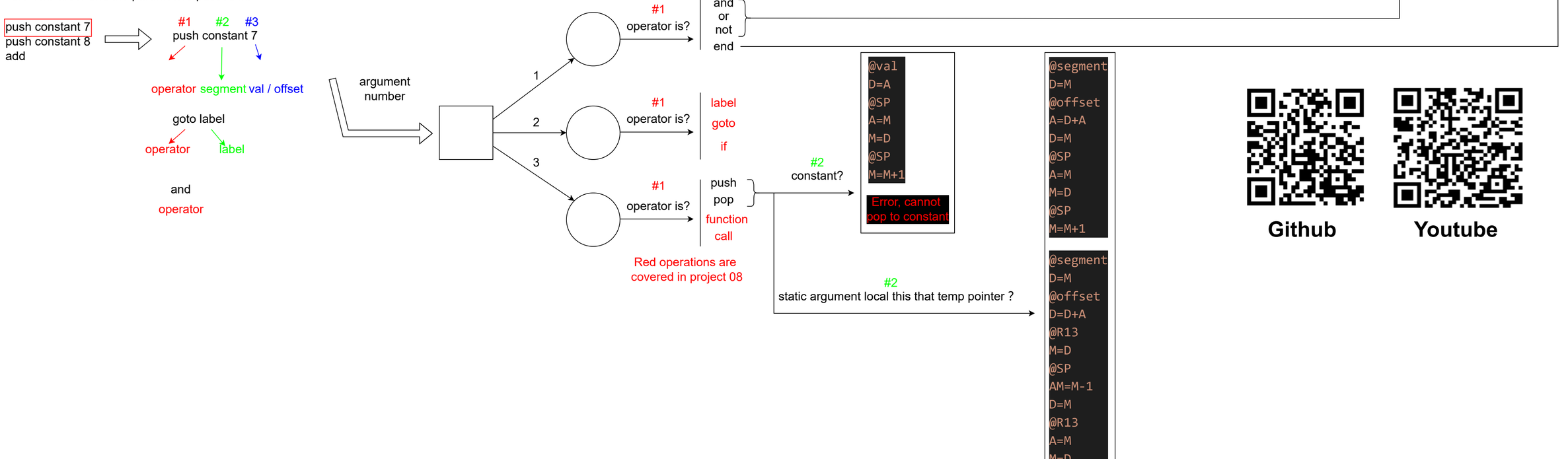




Main: read .vm file line by line, write translated assembly code into .asm file.

VMParser: Break the command in each line into operation, memory segment and offsets (or constant value), then call the asm code writer functions in ASMWritter.

ASMWritter: generates the assembly code based on the called function by the VMParser and the parameters passed in.



Red operations are covered in project 08

Github

Youtube