EE234

Microprocessor Systems

Midterm Exam 2

Nov. 12, 2021. (2:10pm – 3pm)

Instructor: Dae Hyun Kim (daehyun@eecs.wsu.edu)
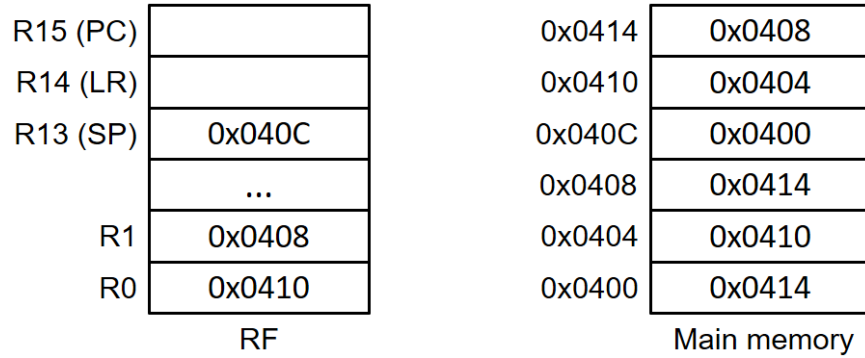
Name:

WSU ID:

Problem	Points	
1	10	
2	20	
3	40	
Total	70	

Problem #1 (Stack, 10 points)

<u>* Notice that we assume that SP points to the topmost available memory space.</u>

The following shows the register file (RF) and the main memory map.

R15 (PC)			0x0414	0x0408
R14 (LR)			0x0410	0x0404
R13 (SP)	0x040C		0x040C	0x0400
	...		0x0408	0x0414
R1	0x0408		0x0404	0x0410
R0	0x0410		0x0400	0x0414
	RF			Main memory

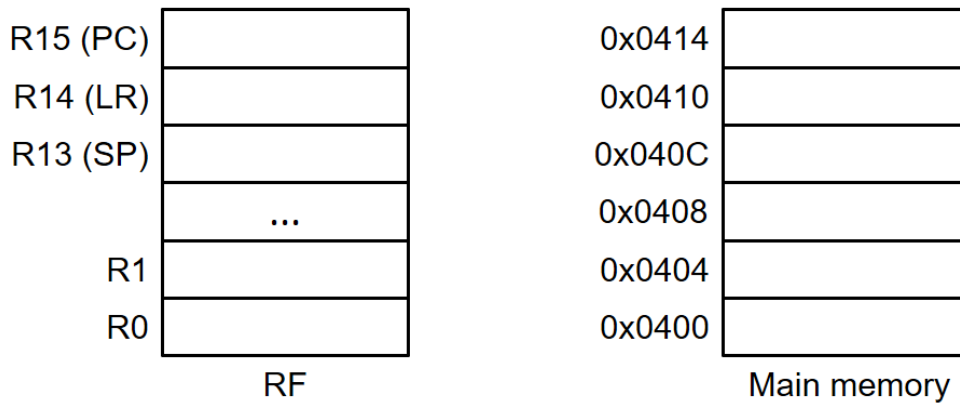Fill in R0, R1, and the main memory map below after the following instructions are executed.

PUSH {R0}

PUSH {R0}

POP {R1}

POP {R0}

POP {R0}

ADD R1, R0, R1

R15 (PC)			0x0414	
R14 (LR)			0x0410	
R13 (SP)			0x040C	
	...		0x0408	
R1			0x0404	
R0			0x0400	
	RF			Main memory

Problem #2 (Stack, 20 points)

Answer the following questions for the assembly code shown below.

```
main:
  MOV R0, #3
  BL run_1
  B end

run_1:
  PUSH {LR}
  PUSH {R0}
  BL run_2
  POP {R0}
  ADD R1, R1, #2
  POP {LR}
  BX LR

run_2:
  PUSH {LR}
  PUSH {R0}
  CMP R0, #1
  BEQ run_3
  SUB R0, R0, #1
  BL run_1
  ADD R1, R1, #1
  POP {R0}
  POP {LR}
  BX LR

run_3:
  MOV R1, #3
  POP {R0}
  POP {LR}
  BX LR
end:  // end of code
```

(1) (10 points) What is the value stored in R1 when the program ends?

(2) (10 points) How many times is the "PUSH {R14}" statement executed?

Problem #3 (Subroutines and Stack, 40 points)

```
main:
  MOV R0, #10
  BL run_1
  B end

run_1:
  PUSH {LR}
  CMP R0, #2
  BLT run_1_end
  SUB R0, R0, #1
  BL run_2
  POP {LR}
  BX LR
run_1_end
  MOV R1, #1
  POP {LR}
  BX LR

run_2:
  PUSH {LR}
  CMP R0, #2
  BLT run_2_end
  SUB R0, R0, #2
  BL run_1
  POP {LR}
  BX LR
run_2_end
  MOV R1, #2
  POP {LR}
  BX LR

end:  // end of code
```

(1) (10 points) What is the value stored in R1 when the program ends?

(2) (10 points) Will the code still work if the initial value of R0 ("MOV R0, #10") is very large (e.g., "LDR R0, =0xFFFFFFFF")? Explain if the code has any potential problem.

(3) (20 points) Rewrite the code (with or without subroutine calls) so that it can work without any problem for any initial value of R0 (assume that R0 has an unsigned value). Try to minimize the # lines (except address label lines) in your code. You can use R1-R12 for temporary registers.

If you code doesn't work, you will get max. 10 points. If you code works, you will get min. 10 points + extra points (# lines ≤15: +10 points, 16: +9 points, 17: +8 points, 18: +7 points, 19: +6 points, ≥20: +5 points)