EE234

Microprocessor Systems

Midterm Exam

Nov. 13, 2020. (2:10pm – 3pm)

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

Name:

WSU ID:

Problem	Points	
1	10	
2	20	
3	30	
4	30	
Total	90	

Problem #1 (Stack, 10 points)

R0 has 0xEEEE 0000, R1 has 0xCCCC 0000, and R2 has 0x0000 AAAA. Show the contents of the stack memory and the stack pointer after the PUSH or POP operations below.

	0xFF20	0xFFFF 0123
	0xFF1C	0x1234 FF12
SP →	0xFF18	
	0xFF14	
	0xFF10	
	0xFF0C	

⇒ PUSH {R0}

0xFF20	0xFFFF 0123
0xFF1C	0x1234 FF12
0xFF18	
0xFF14	
0xFF10	
0xFF0C	

⇒ PUSH {R2}

0xFF20	0xFFFF 0123
0xFF1C	0x1234 FF12
0xFF18	
0xFF14	
0xFF10	
0xFF0C	

⇒ PUSH {R0}

0xFF20	0xFFFF 0123
0xFF1C	0x1234 FF12
0xFF18	
0xFF14	
0xFF10	
0xFF0C	

⇒ POP {R2}

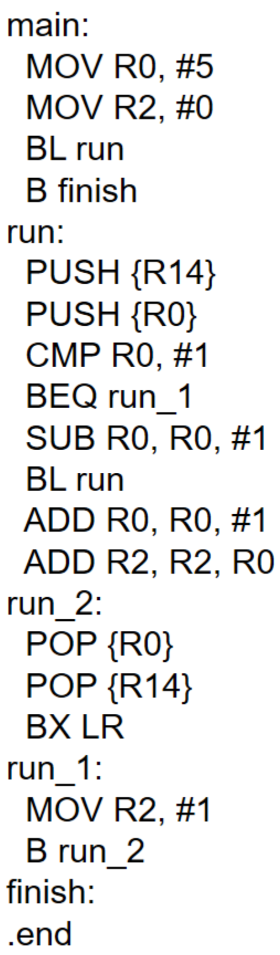
0xFF20	0xFFFF 0123
0xFF1C	0x1234 FF12
0xFF18	
0xFF14	
0xFF10	
0xFF0C	

⇒ PUSH {R1}

0xFF20	0xFFFF 0123
0xFF1C	0x1234 FF12
0xFF18	
0xFF14	
0xFF10	
0xFF0C	

Problem #2 (Stack, 20 points)

Answer the following questions for the assembly code shown below.



```
main:
  MOV R0, #5
  MOV R2, #0
  BL run
  B finish
run:
  PUSH {R14}
  PUSH {R0}
  CMP R0, #1
  BEQ run_1
  SUB R0, R0, #1
  BL run
  ADD R0, R0, #1
  ADD R2, R2, R0
run_2:
  POP {R0}
  POP {R14}
  BX LR
run_1:
  MOV R2, #1
  B run_2
finish:
.end
```

(1) (10 points) What is the value stored in R2 when the program ends?

(2) (20 points) How many times is the “PUSH {R14}” statement executed?

Problem #3 (Stack, 30 points)

Answer the following questions for the assembly code shown below.

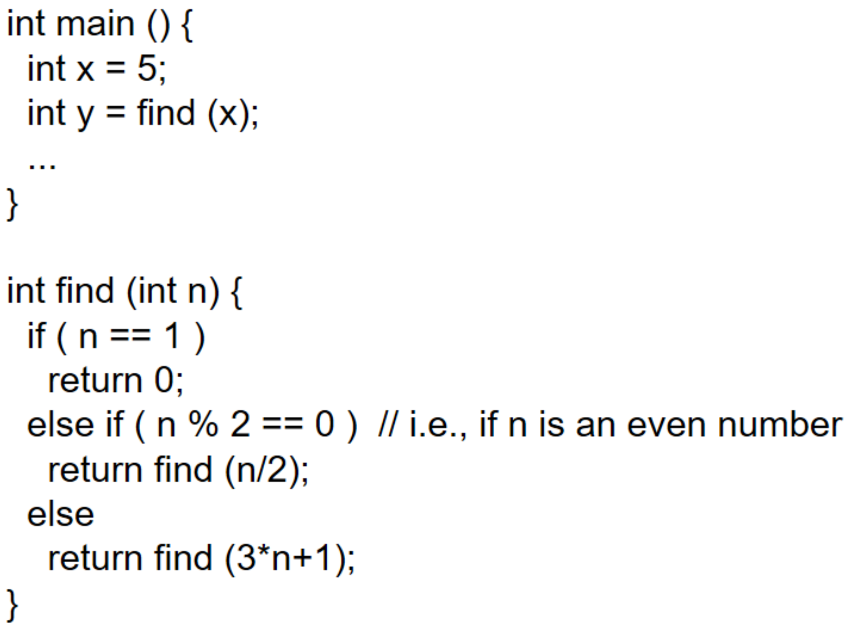
```
main:
  MOV R0, #5
  MOV R2, #0
  BL run
  B finish
run:
  PUSH {R14}
  PUSH {R1}
  PUSH {R0}
  CMP R0, #1
  BEQ run_1
  CMP R0, #2
  BEQ run_1
  SUB R0, R0, #1
  BL run
  MOV R1, #0
  ADD R1, R1, R2
  SUB R0, R0, #1
  BL run
  ADD R1, R1, R2
  MOV R2, R1
run_2:
  POP {R0}
  POP {R1}
  POP {R14}
  BX LR
run_1:
  MOV R2, #1
  B run_2
finish:
.end
```

(1) (10 points) What is the value stored in R2 when the program ends?

(2) (20 points) How many times is the “PUSH {R14}” statement executed?

Problem #4 (C to Assembly, 30 points)

Write an assembly code for the following C code.



```
int main () {
  int x = 5;
  int y = find (x);
  ...
}

int find (int n) {
  if ( n == 1 )
    return 0;
  else if ( n % 2 == 0 )  // i.e., if n is an even number
    return find (n/2);
  else
    return find (3*n+1);
}
```

- Use BL and BX for the recursive function calls.
- R0 is used for the variable x.
- R1 is used for the variable y.
- For the division, use the logical shift right (LSR) instruction. For the multiplication, use the logical shift left (LSL) and addition (ADD) instructions.
- Here is my code for the main function. You implement the “find” function in assembly.

```
main:

  MOV R0, #5  // int x = 5;

  BL find

find:

  // you implement this function
```