ECE220 Midterm 1 Review

Author: Members of HKN

1 Memory-Mapped I/O

(a) What is the difference between a callee-saved and a caller-saved register?

(b) What is the difference between polling I/O vs. Interrupt-Driven I/O?

(c) Fill in the blanks of the following statements relating to Memory-Mapped I/O.

Certain device registers are ________to certain ________locations. However, the registers physically are ________from the memory. Memory-mapped device registers are a common way to ________computer systems with devices.

2 Traps and Subroutines

(a) Why would we need to have service routines (known as TRAPs)? Name three reasons.

(b) **Shifting :(**

Write a subroutine that performs a logical left shift of R1 by the value stored in R2, and reports the result in R1. You may assume that R2 contains a positive number. Only R6 and R1 can be modified/read from, and only R2 can be read from.

```
.ORIG x3100
SHIFTL
; code starts here









; code ends here
RET
.END
```

(c) **Permute Quarters**:

A value stored in a 16-bit LC3 register can be divided into four equal parts of four bits each:

$$X = X_1\ X_2\ X_3\ X_4$$

Write a subroutine, PERMUTE, that reorders R5 as follows:

$$\text{R5} = X_1\ X_3\ X_2\ X_4$$

Assume all registers are caller-saved. You should not use loops. You may assume the existence of a DIV subroutine, which divides as expected. There are also some labels for you to use. [Hint: the previous subroutine might be useful]

```
.ORIG x3100
; assume code for entry point is omitted
;
;
;

PERMUTE
; write the code here













; code ends here
RET
; code omitted
;
.HALT ; labels below
CLEARX2X3 .FILL #0xF00F
FOUR .FILL #4
X2 .FILL 0x0F00
X3 .FILL 0x00F0
SAVER7 .FILL #0
.END
```

(d) **Trap Concepts**

Given the figures below, determine the memory address of the TRAP vector table that will be accessed and which TRAP service routine will be executed.

User Program ASM Code
;
;
TRAP 0xAA
;
;

Address	Value
0xAA	0x05C0
0x1AA	0x05D0
0x2AA	0x05E0
0x3AA	0x05F0
...	
0x05C0	Routine α
0x05D0	Routine ω
0x35C0	Routine κ
0x35D0	Routine π

y

Trap Vector Table Entry:

a. 0xAA

b. 0x1AA

c. 0x2AA

d. 0x3AA

Trap Routine Executed:

a. α

b. ω

c. κ

d. π

Of the following steps executed during a TRAP, in what order are they executed?

a. Return to User Program

b. Execute Trap Routine

c. Access Trap Vector Table

1)

2)

3)

3 Stack Operations

(a) Given the following input sequence of numbers: “24609846117”, write a sequence of pushes and pops that produces this output: “64098116472”.

(b) Two Parts:

   i. Write the expression $(((4 * 2) + 1)/3) + 5$ in postfix notation.

   ii. Write the following postfix expressions in mathematical notation and indicate what they evaluate to (if they are not valid, write “not valid”)

      - 6721*-5/*

      - 83+632-+1*+-

      - 2233*2+5-+8/-

      - 89+6-44

(c) **MP2 Postfix Calculator:** This sequence is input to the console: 445+3/8*-= Draw the stack (and where the stack pointer points to) after:

   i. 5 has been input

   ii. + has been input

   iii. * has been input

   iv. = has been input

Assume that the stack pointer points to the address **one above the most recent-pushed entry.** Remember that a POP does NOT remove an item from memory but simply changes the stack pointer!

4 C Programming

(a) What will be the output of the following C Program?

```c
int main() {
  int i;
  for (i = 3; i < 13; i ++ )
  {
    if (i % 3 == 1)
    {
      printf("Bong\n");
            }
    if (i % 2 == 0)
    {
      printf("Ding\n");
      continue;
    }
    printf("Odd\n");

  }
  return 0;
}

```

(b) What is the return value of this program?

```c
int foo(int x, int y);

int main()
{
  int x = 3;
  int y = 4;

  x = y + foo(x,y);
  y = x - foo(x,y);

  return x + y;



}

int foo(int x, int y)
{
  int a = x + y;
  int b= x - y;
  a += x--;
  y ++;
  a += (y + 1);
  return b + a + --x;


}


```

5 Conceptual Questions

(a) What is the order of access for a stack abstract data type?

(b) Define overflow and underflow.

(c) True or False. Please explain your answer.

i. Interrupts are more efficient than polling.

ii. There are up to 8 possible TRAP service routines.

iii. TRAPs shield programmers from system specific details.

iv. PSR and PC are pushed to the User stack before executing an interrupt service routine.

v. An item is deleted after being pushed off the stack.

vi. TRAP service routines are provided as part of the system code.