Fall 2023
Exam III: Revenge of the C
2023-12-07

Instructions:

This exam contains 14 pages (including this cover page) and 24 questions.

You have **180 minutes** to complete the examination.

For this exam, you have been given a separate scantron answer sheet to fill in your responses. We scan this into Gradescope so color in the bubble without going outside of the lines.

It is **OK** to write on your test pages.

It is **NOT OK** to write anything on your scantron answer sheet except for your name, computing ID, signature, and coloring in your bubbles (small circles) to answer the questions.

All questions on this test are single-select unless otherwise specified in a specific question.

You may not use a calculator or notes.

We will use the following data type sizes:

Types	size in bits
`char`	8
`short`	16
`int` and `float`	32
`long` and `double`	64

Function arguments are in (in order) `%rdi`, `%rsi`, `%rdx`, `%rcx`, `%r8`, `%r9`; return values are in `%rax`.

The next several pages contain reference material which you are welcome to refer to during the test if you would like.

Our Example ISA

This is the same ISA used in HW03 and HW04, but presented to fit onto one printed page.

Each instruction is one or two bytes, with the meaning of those bytes being:

Not all instructions have the second byte; those that do describe it below as the byte "at `pc + 1`".

In the table below `rA` means "the value stored in register number `a`" and `rB` means "the value stored in register number `b`."

`icode`	`b`	Behavior	add to `pc`
0		`rA = rB`	1
1		`rA += rB`	1
2		`rA &= rB`	1
3		`rA` = read from memory at address `rB`	1
4		write `rA` to memory at address `rB`	1
5	0	`rA = ~rA`	1
5	1	`rA = -rA`	1
5	2	`rA = !rA`	1
5	3	`rA = pc`	1
6	0	`rA` = read from memory at `pc + 1`	2
6	1	`rA +=` read from memory at `pc + 1`	2
6	2	`rA &=` read from memory at `pc + 1`	2
6	3	`rA` = read from memory at the address stored at `pc + 1`	2
7		if `rA <= 0`, set `pc = rB`	N/A
7		if `rA > 0`, do nothing	1

If the first bit of the byte at `pc` is `1` instead of `0`, the above text does not define what the instruction means, but some other source (such as a question on this exam) might. If it has no defined meaning either here or elsewhere, leave the `pc` and all other registers and memory values unchanged.

1 Representing Numbers in Memory

1. Convert 442_5 **in Base 5** to **Base 7** (Hint: Convert Base 5 to Base 10 and Base 10 to Base 7)?

   (A) `122`

   (B) `316`

   (C) `233`

   (D) `214`

2. How would -0.75 be represented on a 16-bit machine? Let's assume that we are using IEEE 754 Half-precision float, which uses a 5-bit biased exponent. (The format for IEEE 754 is sign bit, exponent bit, and then mantissa).

   (A) `0xB99A`

   (B) `0xDCDA`

   (C) `0xDCDB`

   (D) `0xF400`

   (E) `0xBA00`

   (F) `0xF001`

   (G) `None of the above`

3. What is -53_{10} in 8-bit Two's Complement?

   (A) `11001011`

   (B) `11001010`

   (C) `00110101`

   (D) `11101011`

4. Assume 0x4A436C12C53281AD (64 bit number) is stored at memory address 0x29, what is the value at 0x2B assuming it is stored in little endian vs big endian?

   (A) `Little Endian: 0x43 Big Endian: 0x81`

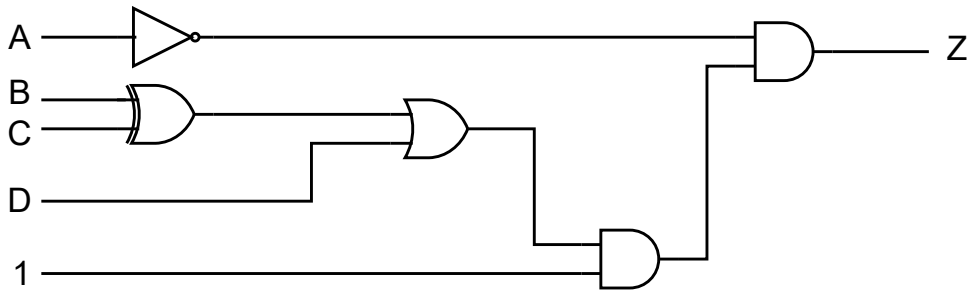
   (B) `Little Endian: 0x6C Big Endian: 0x32`

   (C) `Little Endian: 0x81 Big Endian: 0x43`

   (D) `Little Endian: 0x32 Big Endian: 0x6C`

2 Circuits

5. Given the following circuit:



   What is the value of Z for these tests:

```
Test 1: A=1, B=0, C=1, D=1
Test 2: A=0, B=1, C=1, D=0
```

   (A) `Test 1: 0 Test 2: 0`

   (B) `Test 1: 0 Test 2: 1`

   (C) `Test 1: 1 Test 2: 0`

   (D) `Test 1: 1 Test 2: 1`

6. Assume I want to write this program in Toy ISA

```
int main(){
    int i = 0;
    int x = 5;
    while i<=30:
        x=x+5;
        i=i+1;
}
```

I have gotten the following:

```
60 00 64 05 68 [A] 6C 1E [B] 6D 01 65 05 7E
```

What is the value for position [A] and [B] (Assume pc begins at 0)?

(A) `A: 09 B: 5C`

(B) `A: 04 B: 67`

(C) `A: 08 B: 5D`

(D) `A: 12 B: 51`

(E) `A: 09 B: 5D`

(F) `A: 16 B: 6A`

(G) `A: 08 B: 78`

3 Toy Single Cycle Machine

Here are all the components required to construct a single-cycle machine that implements our Toy ISA. Connect these components to execute the following instruction: `if rA <= 0, set pc = rB`. Ensure that your machine functions correctly in all stages, including fetch, decode, execute, memory, and write-back. If a stage is not required for this instruction, you can skip wiring it. The controller will set C_4 to 3 for this instruction.

To complete the diagram, indicate which wire each given wire should be connected to. E represents the Flags and this Line is connected to the Controller.

For example, if wire A should be connected to wire V, we would write this as $V \to A$. You do not have to use all the wires. Leaving the appropriate boxes blank is considered the correct answer; filling a box unnecessarily will be considered incorrect.

Function codes for ALU are

code	function
001	and
010	addition
011	subtraction
100	shift

You can also reference the output of register file directly use `RD2` or `RD1`. For example $RD1 \to F$

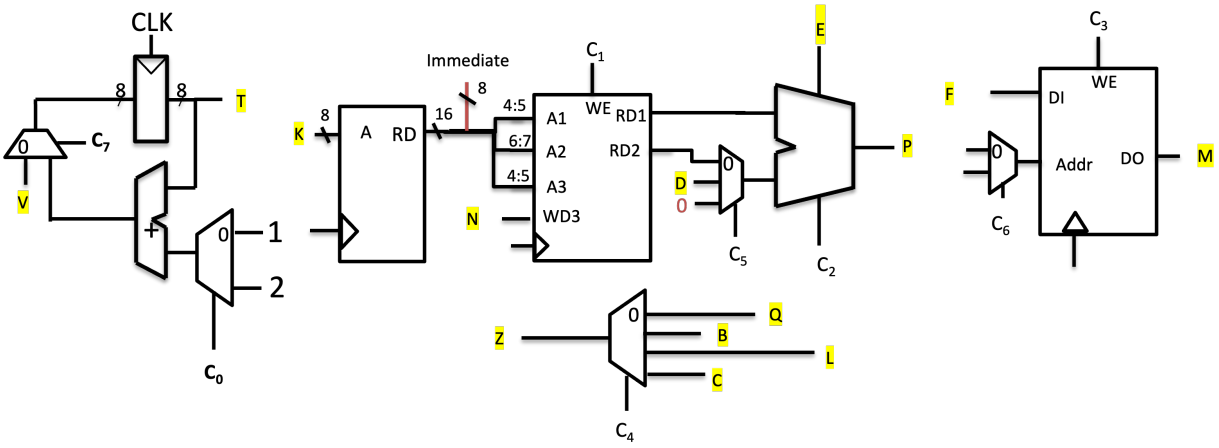


Figure 1: Single Cycle Machine Single Design

7. Select correct wiring from the list below.

   (A) $RD1 \to V, T \to K$

   (B) $RD2 \to V, T \to K$

   (C) $V \to RD1, T \to F$

   (D) $V \to RD2, T \to N$

   (E) $T \to K, P \to L, Z \to N, V \to M, F \to RD2$

   (F) $T \to K, P \to L, Z \to N, V \to M, F \to RD1$

   (G) $T \to K, P \to Q, Z \to N, V \to M, F \to RD1$

8. What are the control signals set to?

   (A) $C0 \to X$, $C1 \to 0$, $C2 \to 3$, $C3 \to 0$, $C4 \to X$, $C5 \to 1$, , $C6 \to X$, $C7 \to 0$
   (B) $C0 \to X$, $C1 \to 0$, $C2 \to 1$, $C3 \to 0$, $C4 \to X$, $C5 \to X$, $C6 \to X$, $C7 \to 1$
   (C) $C0 \to 1$, $C1 \to 0$, $C2 \to 1$, $C3 \to 0$, $C4 \to X$, $C5 \to X$, $C6 \to X$, $C7 \to 1$
   (D) $C0 \to 1$, $C1 \to 0$, $C2 \to 4$, $C3 \to 0$, $C4 \to 1$, $C5 \to 1$, $C6 \to X$, $C7 \to 0$
   (E) $C0 \to 0$, $C1 \to 0$, $C2 \to 4$, $C3 \to 0$, $C4 \to 1$, $C5 \to 1$, , $C6 \to X$, $C7 \to 1$
   (F) $C0 \to 0$, $C1 \to 0$, $C2 \to 3$, $C3 \to 0$, $C4 \to 1$, $C5 \to 1$, $C6 \to 1$, $C7 \to 1$
   (G) $C0 \to 1$, $C1 \to X$, $C2 \to 3$, $C3 \to 1$, $C4 \to X$, $C5 \to 0$, $C6 \to 0$, $C7 \to 0$

4 Pointers and Assembly

9. Assume the following code was given and the registers are **signed 8 bit registers**

```
movb $0x75, %dil
movb $0x59, %al
addb %al, %dil
```

   Note: These are the 8-bit registers (%dil is %rdi and %al is %rax)
   What are the flag values after the add?

   (A) `SF:0 OF:0 CF:1 ZF:0`
   (B) `SF:0 OF:1 CF:0 ZF:0`
   (C) `SF:0 OF:1 CF:1 ZF:0`
   (D) `SF:1 OF:1 CF:0 ZF:0`
   (E) `SF:1 OF:1 CF:1 ZF:0`
   (F) `SF:1 OF:1 CF:0 ZF:1`
   (G) `SF:0 OF:1 CF:1 ZF:1`

10. What function is below: Assume the value of x is stored in `%rdi` and the value of y is stored in `%rsi`

```
leal    (%rsi,%rsi,2), %eax
addl    %edi, %eax
retq
```

   (A) `y*3 + x`
   (B) `x+y+2`
   (C) `x+2*y`
   (D) `x+2*y+2`
   (E) `xy`
   (F) `x+y-2`
   (G) `(y-2)*y+y/2`

   You've finished 10 questions! Great job! Keep going!

11. Given the following code

```
uint8_t c = 0x20;           \\ stored at address 0x04
uint8_t d = 0x30;           \\ stored at address 0x08
uint8_t e = 0x14;           \\ stored at address 0x0C
int *a = &c;                \\ stored at address 0x1C5
int *b = &d;                \\ stored at address 0x1D0
*b  = 2*c + d;
int **l = &a;               \\ stored at address 0x2C0
**l = *a + 3*(*b) ;
int *r = &d;                \\ stored at address 0x2D0
int **n = &r;               \\ stored at address 0x3D8
```

What are the values of **n, &(*l), and r?

(A) `**n: 0x30 &(*l):0x20 r:0x2D0`

(B) `**n:0x60 &(*l):0x1C5 r:0x2D0`

(C) `**n:0x70 &(*l):0x04 r:0x08`

(D) `**n:0x30 &(*l):0x04 r:0x70`

(E) `**n:0x70 &(*l):0x2C0 r:0x30`

(F) `**n:0x70 &(*l): 0x1C5 r:0x08`

(G) `**n: 0x3D8 &(*l):0x100 r:0x30`

12. Given the code:

```
typedef struct from{
  short my;
  unsigned long point;
  int ofview[10];
  char thejedi[4];
}areevil;
```

Assume a pointer to the struct is stored at memory address 0x40 with a value of 0x90, at what memory address would ofview[3] be stored? Assume no padding.

(A) `0x9E`

(B) `0xA1`

(C) `0x112`

(D) `0xA6`

(E) `0xA8`

(F) `0x106`

(G) `0x108`

13. What will the following code print?

```
#include <stdio.h>

int main() {
    char *str = "Hello World";
    printf("%c", *(str+1));
    return 0;
}
```

(A) 0

(B) ello World

(C) e

(D) l

(E) "

(F) d

(G) <space>

14. Consider the following struct. How would I print out the name of a student.

```
typedef struct {
    int id;
    char *name;
} Student;

Student std;
std.id = 123;
std.name = "Jane Doe";
Student *ptr = &std;
```

(A) `printf("%s", ptr->name);`

(B) `printf("%s", *ptr->name);`

(C) `printf("%s", ptr.name);`

(D) `printf("%s", *(ptr.name));`

(E) `printf("%s", *name->ptr);`

(F) `printf("%s", (ptr)name*);`

(G) `printf(ptr*->name*);`

15. Consider the following C code.

```
int updateValue(int *value) {
    *value = 10;
    return *value;
}
int main() {
    int num = 5;
    updateValue(&num);
    return 0;
}
```

Which of the following snippets of assembly does it correspond to.

(A)
```
updateValue:
    movl $10, %rdi
    ret
```

(B)
```
updateValue:
    movl (%rdi), %eax
    movl $10, (%eax)
    ret
```

(C)
```
updateValue:
    movl %rdi, %eax
    movl $10, (%eax)
    ret
```

(D)
```
 updateValue:
     movl $10, (%rdi)
     movl (%rdi), %eax
     ret
```

(E)
```
updateValue:
    movl $10, (%eax)
    ret
```

(F)
```
updateValue:
    movl $10, %eax
    ret
```

5 Function Pointers

The following program is a simple implementation of polymorphic program. This polymorphic program modifies its code during execution.

```
#include <stdio.h>
#include <stdint.h>

//Type def the function pointer (Part A)

---------------------------------------

void computeXOR(uint8_t *array, size_t size) {
    uint8_t xorResult = 0xFF;
    for (size_t i = 0; i < size; ++i) {
       array[i] ^= xorResult ;
    }
    return;
}

int main() {
   uint8_t myArray[] = {___________________________________}; //Part B
    size_t arraySize = sizeof(myArray) / sizeof(myArray[0]);
    computeXOR(myArray, arraySize);
    funct_ptr to_be_executed = (funct_ptr) myArray;
    int result = to_be_executed();
    printf("The result %d", result);
    return 0;
}
```

16. (Part A) Let's assume that the function in the Array takes no arguments and returns a short. Which of the following implements the `typedef` from the function pointer?

    (A) `typedef int (funct_ptr*)(void);`

    (B) `typedef int (funct_ptr)(void);;`

    (C) `typedef int (funct_ptr)();`

    (D) `typedef int funct_ptr;`

17. (Part B) We want to execute the following instruction: `movl $0x77, %eax`. The encoding of this is `B8 77 00 00 00` and the encoding for `ret` instruction is `C3`. What content should we put in the array when we write the program so that the instruction is executed?

(A) `0xB8, 0x77, 0x00, 0x00, 0x00, 0xC3`

(B) `0xB8, 0x00, 0x00, 0x00, 0x77, 0xC3`

(C) `0xC3, 0x00, 0x00, 0x00, 0x77, 0xB8`

(D) `0x47, 0x88, 0xFF, 0xFF, 0xFF, 0x3C`

(E) `0x47, 0xFF, 0xFF, 0xFF, 0x88, 0x3C`

6 Memory Errors

Consider the following snippet of code.

```
 1  #include <stdio.h>
 2
 3  int* createAndReturnArray() {
 4      int stackArray[5] = {1, 2, 3, 4, 5};
 5      int* ptr = stackArray;
 6      return ptr;
 7  }
 8
 9  int main() {
10      int* result = createAndReturnArray();
11      for (int i = 0; i < 5; i++) {
12          printf("%d ", result[i]);
13      }
14      free(result);
15      return 0;
16  }
```

18. The program above contains memory errors. Select which option (only one) contains the lines associated with the memory errors?

(A) `line 14, line 6`

(B) `line 10, line 12`

(C) `line 5, line 10`

(D) `line 3, line 15`

(E) `line 11, line 12`

(F) `line 5, line 12`

19. Given following student struct

```
typedef struct  {
    char studentID[6];
    int grade;
}Student;
```

How would you allocate an array that stores 7 student struct elements.

(A) `struct Student* studentArray = (struct Student*)malloc(7);`

(B) `Student* studentArray = (struct Student*)malloc(7 * sizeof(Student));`

(C) `Student* studentArray = (Student*)malloc(7 * sizeof(Student));`

(D) `Student* studentArray = (void*) malloc(7);`

(E) `Student* studentArray = (Student*)malloc(sizeof(Student));`

(F) `struct Student* studentArray[7] = malloc(struct Student*);`

(G) `struct Student* studentArray[7] = malloc(6*sizeof(char)+sizeof(int));`

20. Assume you have a character array `char str[31] = "Hello,_World!_Programming!_Fun";`. Your task is to move the substring `"Programming"` to the beginning of the array,. The result should be `"Programming!!_Programming!_Fun"`

(A) `memcpy(str, str + 13, 11);`

(B) `memmove(str, str + 13, 12);`

(C) `strcpy(str, str + 13);`

(D) `strncpy(str, str + 14, 12);`

(E) `strncpy(str + 14, str, 12);`

(F) `memmove(str, str + 14, 12);`

(G) `memmove(str + 14, str, 12);`

7 Sockets

The following program implements server that returns the time when a client connects to it. Rearrange the lines to correctly implement the server.

```
----SNIP IMPORTS ---
int main(int argc, char *argv[]) {
    int sockfd, newsockfd, portno;
    struct sockaddr_in serv_addr, cli_addr;
    int size_of_client_conn_socket;
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    bzero((char *) &serv_addr, sizeof(serv_addr));

    serv_addr.sin_family = AF_INET;
    serv_addr.sin_addr.s_addr = INADDR_ANY;
```

```
    serv_addr.sin_port = htons(8080);
    size_of_client_conn_socket = sizeof(cli_addr);

1.  while (1) {
2.  close(newsockfd);
3.  write(newsockfd, ":)", 2);
4.  listen(sockfd, 5);
5.  newsockfd = accept(sockfd, (struct sockaddr *) &cli_addr,
              &size_of_client_conn_socket);
6.  bind(sockfd, (struct sockaddr *) &serv_addr, sizeof(serv_addr));
7.  }
    return 0;
}
```

21. Rearrange the lines above so that they correctly implement the server.

    (A) 5, 1, 4, 6, 2, 3, 7
    (B) 1, 5, 3, 2, 6, 4, 7
    (C) 4, 2, 6, 1, 3, 5, 7
    (D) 6, 4, 1, 5, 3, 2, 7
    (E) 2, 3, 5, 1, 4, 6, 7
    (F) 3, 6, 1, 4, 5, 2, 7
    (G) 5, 4, 1, 3, 6, 2, 7

22. How difficult was this final?

    (A) too easy
    (B) easy
    (C) fair
    (D) difficult
    (E) too difficult

23. Assuming that you wanted to take CSO2 next semester, were you able to enroll?

    (A) yes
    (B) no
    (C) Not interested in taking CSO2 next semester

24. Would you like 5% extra credit on this Exam. (you still get credit if you don't answer it)?

    (A) yes
    (B) no

    Congratulations on finishing the final and the course. It's been an absolute pleasure to TA for you guys this semester! Hope the holidays are amazing and your remaining time at UVA is great! - Adriel Barretto, Khyati Kiyawat and the TA Team.