McGill University, Department of Electrical and Computer Engineering

ECSE 324 Computer Organization - Final Exam Fall 2022

Dec 13, 2022 6:30:00 PM

Examiner: Prof. Christophe Dubach **Associate Examiner:** Prof. Brett Meyer

Exam:	**Closed Book** Printed on both sides of the page Answer on Exam Exam should be returned
Crib sheets :	**Not permitted**
Dictionaries :	Translation & Regular
Calculators :	**Not permitted**

Special instructions: You can do rough work on the blank pages. Rough work will not be used for grading. For all multiple-choices questions, only select a single answer. If more than one answer is selected, you will get zero for that question. Filling up the answer-key on the last page is worth 1 point.

No questions allowed. In case you suspect a mistake/bug in a question, write down your assumptions and answer the question as best as you can.

Part A: General knowledge, multiple choice questions [1 point] each. **[43 points]**

A.1 Computing abstractions

1. Assuming that 0x 90 12 4F EE is the in-memory representation of a big-endian number, the equivalent little-endian representation is:

a. 0x EE 90 12 4F
b. 0x 12 EE 90 4F
c. 0x EE 4F 12 90
d. 0x 90 4F EE 12
e. 0x EE F4 21 09

2. Which of the following data accesses is unaligned?

a. Read a 32-bit word at 0x02
b. Read a 16-bit half-word at 0x02
c. Read a 32-bit word at 0x04
d. Read a 32-bit word at 0x00
e. Read a 8-bit word at 0x02

3. In a Von Neumann architecture,

a. Instructions and data are stored in the same memory
b. Instructions and data are stored in separate memories
c. Signed integers are represented as 2's complement
d. Signed integers are represented as 1's complement
e. The word size is 32 bits

4. How many bytes of memory are allocated for the following code example (ignoring alignment and assuming integers are 32 bits)?

```
int arr [10] ;  char x;
```

a. 11
b. 14
c. 41
d. 44
e. None of these

5. How many unique addresses does a 33-bit system have?

a. 8GB
b. 4GB
c. 2GB
d. 1GB
e. None of these

6. Which of the following statements is true?

a. Memory is always non-volatile
b. I/O is used to exchange data with humans or other devices
c. Programs are burned into processors at design time
d. Every motherboard contains a Wi-Fi chip
e. Programs can be stored in memory but not on an HDD/SSD

7. What is the top of stack after executing the following sequence of operations?

Push 1; Push 2; Pop; Push 3; Pop

a. 0
b. 1
c. 2
d. 3
e. None of these

8. For the following C structure, what is the address of variable "e" if variable "a" is allocated at address 0x00? Assume that integers are 32 bits and must be aligned at a 4-byte boundary in the struct.

```
struct test{
  int a;
  char b;
  char c[2];
  int d;
  char e;
}
```

a. 0x04
b. 0x08
c. 0x0C
d. 0x10
e. None of these

A.2 ISA

9. How wide is an ARM instruction (assuming it is not a Thumb instruction)?

a. 32 bits
b. 16 bits
c. 64 bits
d. 256 bits
e. ARM instructions are variable-length

10. ADD R4, R5, #24 corresponds to the following operation:

a. R4 = R5 + 24
b. R5 = R4 + 24
c. R0 = R4 + R5 + 24
d. R24 = R4 + R5
e. None of the above

11. Which of the following statements is false?

a. An ISA specifies how the hardware implements instructions
b. An ISA defines an interface between hardware and software
c. Different processors may implement the same ISA
d. x86-64, ARMv7-A, Power ISA 3.0, and RISC-V are ISAs
e. The ISA consists of the instruction set, information about how memory is organized, how to access memory, and more

12. What is the effective address for the following ARM load instruction?

```
LDR R4, [R1, R0, LSL#2]
```

a. R1 + (2 * R0)
b. (R1 + R0) * 4
c. R1 + R0
d. (R1 * 4) + R0
e. None of the above

13. Assuming that R0 is initially equal to addr, what are the register contents of R0, R1, R2 and R3 after executing the program below?

```
LDR R1,[R0],#4
LDR R2,[R0,#4]!
LDR R3,[R0,#4]
```

a. R0 = addr + 8; R1 = *addr; R2 = *(addr + 8); R3 = *(addr + 12)
b. R0 = addr + 12; R1 = *addr; R2 = *(addr + 8); R3 = *(addr + 12)
c. R0 = addr; R1 = *(addr + 4); R2 = *(addr + 8); R3 = *(addr + 12)
d. R0 = addr + 4; R1 = *(addr + 4); R2 = *(addr + 4); R3 = *(addr + 8)
e. R0 = addr + 8; R1 = *(addr + 4); R2 = *(addr + 4); R3 = *(addr + 4)

14. Which instruction is equivalent to the following code?

```
        BNE LABEL1
        ADD R1, R2, R3
LABEL1:
```

a. CMP R2, R3
b. ADDEQ R1, R2, R3
c. ADD R1, R2, R3
d. ADDNE R1, R2, R3
e. None of these

A.3 Software

15. A C program is typically processed by system tools in the following order:

a. Linker -> Loader -> Assembler -> Compiler
b. Assembler -> Linker -> Compiler -> Loader
c. Compiler -> Assembler -> Linker -> Loader
d. Loader -> Assembler -> Compiler -> Linker
e. Compiler -> Loader -> Assembler -> Linker

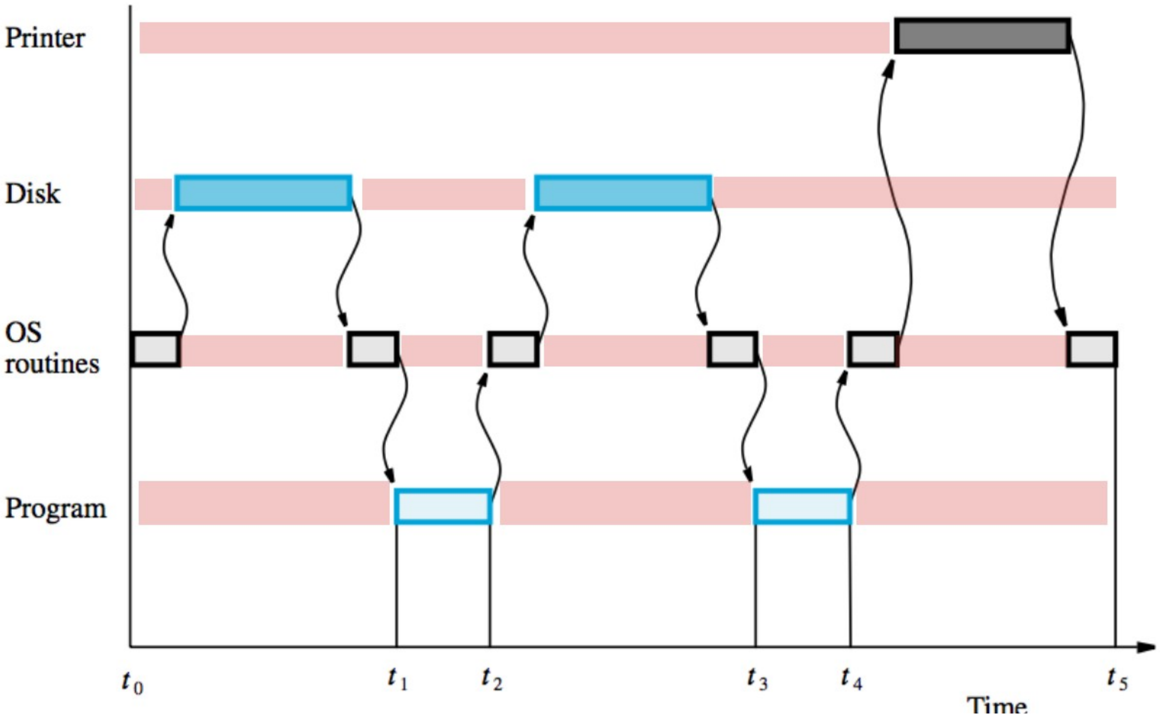
16. Which tool is responsible for constructing and deconstructing the runtime environment in which programs run?

a. Assembler
b. Linker
c. Compiler
d. Loader
e. Parser

17. Which computer program is responsible for handling external symbols before generating a binary?

a. Assembler
b. Linker
c. Loader
d. Debugger
e. None of these

18. The figure shows the input/output steps of a program. Let each of the six OS routine execution intervals be 1 unit of time, with each disk operation requiring 4 units, printing requiring 6 units, and each program execution interval requiring 3 units. How long does the execution of the program take?



a. 40

b. 23

c. 20

d. 18

[✓ e. 26]

19. Assuming the operating system can schedule the concurrent execution of many programs perfectly so as to maximize the system throughout, what is the total time required to execute two identical copies of the program from the previous question, assuming there is one printer, one disk and two CPUs?

a. 26

b. 52

[✓ c. 32]

d. 44

e. 40

20. Given the following ARM assembly program, which labels are defined in the Symbol Table and which ones are known?

```
.global _start

_start:

   MOV R1, #5

LOOP1:

   ADD R1, #1
   CMP R1, #10
   BNE LOOP1

LOOP2:

   SUB R1, #4
   B LOOP3

.end
```

a. _start (location known),
LOOP1 (location known),
LOOP2 (location known),
LOOP3 (location unknown)

b. start (location unknown),
LOOP1 (location unknown),
LOOP2 (location unknown),
LOOP3 (location unknown)

c. start (location known),
LOOP1 (location known),
LOOP2 (location known),
LOOP3 (location known)

d. start (location unknown),
LOOP1 (location known),
LOOP2 (location known),
LOOP3 (location unknown)

e. start (location known),
LOOP1 (location known),
LOOP2 (location unknown),
LOOP3 (location known)

A.4 I/O

21. An interrupt request (IRQ) signal,

a. Allows the CPU to interact with the device only when it is useful
b. Allows the CPU to work on other tasks while waiting
c. Allows the CPU to sleep while waiting
d. All of the above
e. None of the above

22. Which event is the possible cause of interrupt?

a. Null pointer exceptions
b. Segmentation faults
c. Keyboard inputs
d. All of these
e. None of these

23. Which of the following statements about buses is true?

a. Bus masters initiate communication on the bus; bus slaves respond accordingly
b. Control signals indicate what and when actions are to be taken
c. Address signals indicate which bus-connected resources are requested to participate in an exchange
d. Data signals are used for the exchange itself
e. All of the above

24. Which of the following statements about buses is false?

a. Synchronous buses require careful design to ensure timing constraints are met
b. Asynchronous buses are especially suitable for short buses
c. Asynchronous bus transfers require four end-to-end delays
d. Synchronous bus transfers require two end-to-end delays
e. Asynchronous buses automatically adjust to the timing of each device

25. The following code is related to an I/O device. What is the actual purpose of the code?

```
iodev:   .word 0x00006000

         ...
         LDR      V1,  iodev

L1:

         LDRB     V2, [V1, #4]
         TST      V2, #4
         BEQ      L1
         STRB     V3, [V1]
```

a. Polling for read
b. Polling for Write
c. Echo
d. Idle
e. None of these

A.5 MEMORY

26. Which one is the fastest for a processor to access?

    a. L1 cache
    b. L2 cache
    c. Main memory
    d. Disk
    e. Register file

27. A 16x8 RAM consists of

    a. 16 bytes
    b. 128 bytes
    c. 32 bytes
    d. 64 bytes
    e. 256 bytes

28. Which statement is correct for SRAM?

    a. The size typically reaches 100GB
    b. It can be used to implement an L1 cache
    c. It can store data without power
    d. It must be off-chip
    e. None of these

29. Which statement is true?

    a. SRAM is volatile and loses state unless periodically refreshed once in a while
    b. DRAM is faster and cheaper than SRAM
    c. NVMs are used to implement the register file
    d. All of the above
    e. None of the above

30. To construct a 4Mx64 composite memory out of 1Mx8 memory chips, we need:

    a. 8 1Mx8 memory chips
    b. 16 1Mx8 memory chips
    c. 32 1Mx8 memory chips
    d. 64 1Mx8 memory chips
    e. It cannot be done

31. What is the overall hit ratio of the following address stream (byte-addressable memory) on a fully associative cache of five 32-bit blocks, assuming an LRU replacement policy?

read 0x00
read 0x04
read 0x08
read 0x0c
read 0x10
read 0x04

a. 0/6
b. 1/6
c. 2/6
d. 3/6
e. 4/6

32. Which statement is correct for Virtual Memory?

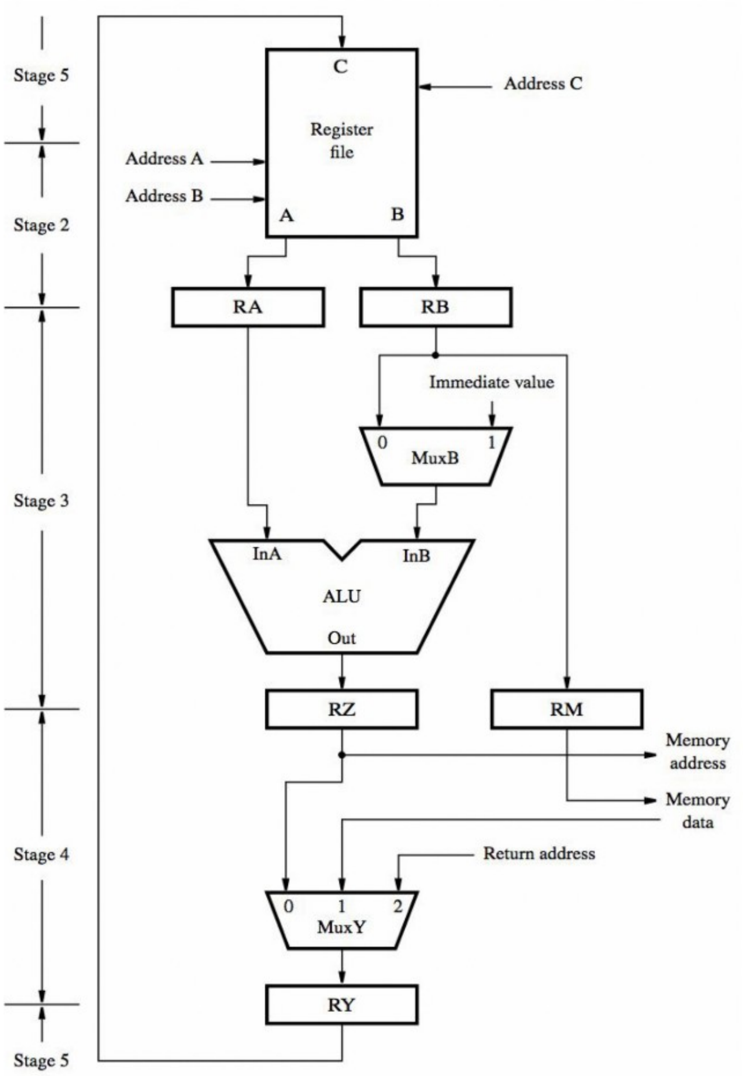
a. Page fault can be fixed by cache replacement
b. A cache miss occurs when the targeted physical address is not in a virtual page
c. The Page Table is stored in main memory
d. Virtual address can be directly used for accessing data from memory without translation
e. None of these

33. Which statement is correct for Memory Management Unit (MMU)?

a. It contains a table to help translating virtual addresses into physical ones quickly
b. Page size can be changed on the fly
c. Every processor must have a MMU
d. Translation lookaside buffer is 1-way set-associative
e. None of these

A.6 Processor

34. Suppose a processor uses the data path depicted below. If the processor executes instruction `STR R1, [R2]` , what is the content of the registers at the beginning of stage 3?



a. RA = R1, RB = R2
b. RA = R2, RB = R1
c. RA = unknown/unused, RB = R2
d. RA = unknown/unused, RB = R1
e. RA = R1, RB = unknown/unused

35. Suppose that an (in-order) ARM processor with a 5-stage pipeline executes the following instructions:

```
LDR   R1, [R2]
ADD   R3, R4, R5
```

Assumes that a cache miss introduces a 5-cycle bubble in the pipeline during the LDR instruction's fourth stage. How long do the instructions take to complete?

a. 8 cycles
b. 10 cycles
c. 11 cycles
d. 14 cycles
e. 15 cycles

36. Consider a 5-stage pipeline without forwarding, how many cycles does it take to run these instructions till completion?

```
ADD   R2, R3, R7
ADD   R9, R2, R8
```

a. 6
b. 7
c. 8
d. 9
e. None of these

37. Consider a 5-stage pipeline with forwarding available between any two stages, how many cycles does it take to run these instructions till completion?

```
ADD   R2, R3, R7
ADD   R9, R2, R8
```

a. 6
b. 7
c. 8
d. 9
e. None of these

38. Suppose that the slowest component in a processor's pipeline is the ALU's multiplier, which limits the clock frequency to 200MHz. The next-slowest component appears in a different stage and limits the clock frequency to 600MHz. In how many stages should one split the multiplier to increase the clock frequency as much as possible?

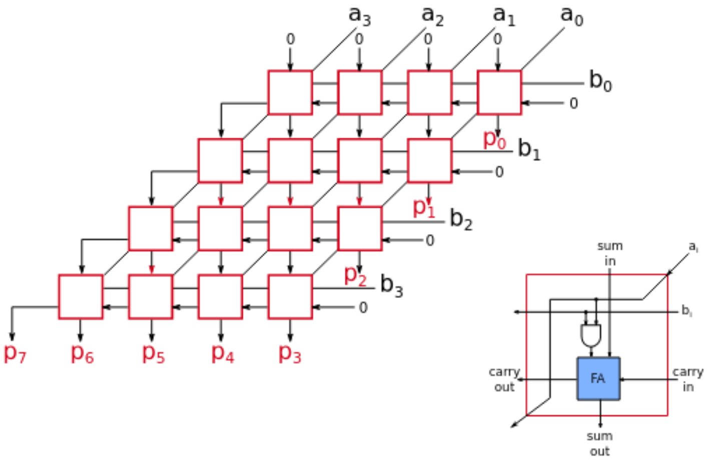
a. 1 stage
b. 2 stages
c. 3 stages
d. 4 stages
e. 5 stages

39. Pipeline stalls may be caused by

a. Structural hazards
b. Data hazards
c. Control hazards
d. All of the above
e. None of the above

A.7 Arithmetic

40. We wish to multiply two n-bits integer. An array multiplier as shown below has a maximum delay of approximately:



a. n
b. 2*n
c. 3*n
d. 4*n
e. 5*n

41 What is the maximum delay of the previous array multiplier if Carry-Save addition is used?

a. n
b. 2*n
c. 3*n
d. 4*n
e. 5*n

42. How many clock cycles does a shift and add multiplier require to multiply two 32-bit numbers, assuming shifting and adding can be done in the same clock cycle?

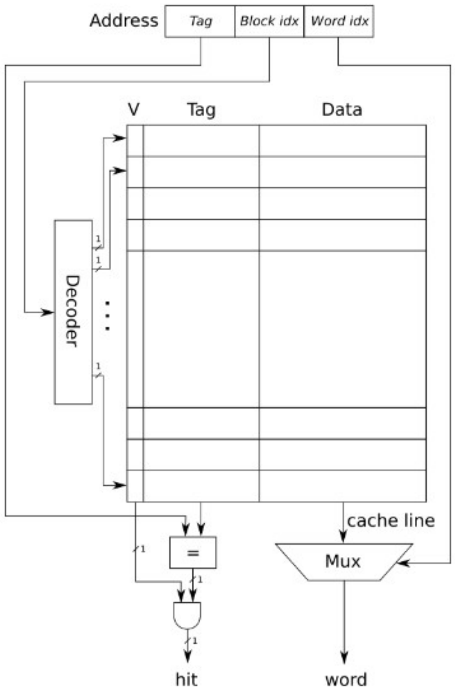
a. 8 cycles
b. 16 cycles
c. 32 cycles
d. 64 cycles
e. 128 cycles

43. What is the hexadecimal representation for 63.375 in fixed point with an 8-bit integer and an 8-bit fraction?

[✓ a. 3F60]
b. 3E60
c. 1E10
d. 3F10
e. None of these

B. Caches **[6 points]**

In all the following questions, we will assume a 32-bits address size, a word size of 32 bits, a 16-bytes cache line size and a 4MB cache (data part) with the following design:



44. What type of caches this circuit corresponds to? **[1 point]**

a. Fully-associative cache
b. Direct-mapped cache
c. Two-ways set associative cache
d. Four-ways set associative cache
e. None of these

45. How many cache lines are in this cache? **[1 point]**

a. 2^16
b. 2^18
c. 2^19
d. 2^20
e. None of these

46. How many bits are in the “Word idx” part of the address ? **[1 point]**

a. 32
b. 18
c. 2
d. 16
e. None of these

47. How many bits in the “Tag” part of the address? **[1 point]**

a. 32
b. 18
c. 2
d. 16
e. None of these

End of multiple choice questions: make sure to fill in the answer-key on the last page of the exam for questions 1 – 47, this is worth 1 point!

----

48. What is the ‘v’ bit used for? **[1 point]**

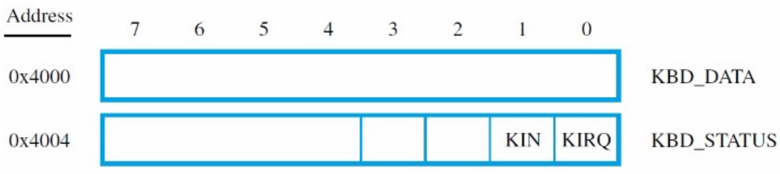
Check if the data is valid

49. Why is there a “Mux” at the bottom of the cacheline? **[1 point]**

To select a single word out of the cacheline.

C. Assembly & friends **[8 points]**

Assume that the data and status register of a keyboard device are memory mapped as follows, and that the KIN bit indicate if new data is ready:



50. Complete the following ARM program to read the keyboards character and store them one by one in the buffer until the carriage return character is detected. **[4 points]**

```
.global _start

.equ    CR, 13                  // ascii code for carriage return

kbd:    .word 0x00004000        // keyboard base address
buffer: .space 44               // allocate 44 bytes of buffer space

_start:
        LDR     V1, kbd
        LDR     V2, =buffer     // assume converted into MOV by assembler
READ:
        LDRB    V3, [V1, #4]
        TST     V3, #2
        BEQ     READ
        LDRB    V4, [V1]
        STRB    V4, [V2], #1

        TEQ     V4, #CR
        BNE     READ

        // rest of the program
```

51. What would happen if more than 80 characters are provided as an input before a carriage return is encountered? **[1 point]**

The instructions would be overridden by the input characters.

52. Assuming the first declared label starts at address 0, what is the address of the label READ?
(hint: think carefully about where the first label is) **[1 point]**

56 in decimal or 0x38 in hexadecimal

53. Assume that you have crafted a very small piece of assembly program which send the entire content of the memory over a network machine you control to steal some sensitive information. You have assembled your code and encoded its binary representation which, we will assume, is equivalent to the following 44 ascii characters: EVILprogramTHATsendsMEMORYcontentOVERtheWEB_

Furthermore, let's assume that the binary representation in hexadecimal of an unconditional branch instruction which jumps back 52 instructions is: eafffff1 and (part of) the ascii table is given below.

Hex value	Char	Hex value	Char
E8	è	F4	ô
E9	é	F5	õ
EA	ê	F6	ö
EB	ë	F7	÷
EC	ì	F8	ø
ED	í	F9	ù
EE	î	FA	ú
EF	ï	FB	û
F0	ð	FC	ü
F1	ñ	FD	ý
F2	ò	FE	þ
F3	ó	FF	ÿ

What input would you provide to the original program to execute your evil plan and send the content of the memory over the web? Write the sequence of ascii characters. **[2 points]**

EVILprogramTHATsendsMEMORYcontentOVERtheWEB_XXXXXXXXêÿÿñ where X is any character

Partial points:
0.5 pt if êÿÿñ appears in the sequence
0.5pt if EVILprogramTHATsendsMEMORYcontentOVERtheWEB_ appears in the sequence