Student name and email: Lasse Thue Drongesen

Student number: 20223489

Exam info

Exam is written

- There are 100 points in total, which will be used to decide the grade.
- There are 7 Parts with 3 exercises in each.
- Each exercise gives 4 or 5 %.

Censorship

There is external censorship.

Exam Guidelines:

- The exam must be completed at AAU campus; it will be 4 hours.
- If you think that the task formulation is deficient / incorrect, then write which additional assumptions / changes you use as a basis for your answer.
- Remember to clearly write the name and study number on the answer.

Permitted aids:

- Common IT tools such as text editor / word processing, PDF annotation tool, Calculator.
- All tools that are introducing in the course.
- Formats of the entire LC-3 instruction set (page 148 of ICS book).
- Slides and notes from the course

Part I: Data Types, and Operations

A. (4%) Consider the hexadecimal number B4. Perform the following conversions and interpretations:

a. Convert the hexadecimal number B4 to its **binary** equivalent.

Answer: 10110100

b. Convert the hexadecimal number B4 to its **decimal** equivalent.

Answer: 128 + 32 + 16 + 4 = 180

c. Interpret the binary equivalent of B4 as an **unsigned 8-bit integer**.

Answer: 11 * 16 + 4 = 180

d. Interpret the binary equivalent of B4 as a **signed 8-bit integer** using **one's complement.**

Answer: 01001011 = 64 + 8 + 2 + 1 (flip sign) = -75

e. Interpret the binary equivalent of B4 as a **signed 8-bit integer** using **two's complement.**

Answer: 76

B. (5%) Calculate the following expression in two's complement and then convert the result to decimal:

1100110 − 1001001 + 111011

Note: The given instruction for answering seems wrong for this expression so I've used the method taught in the course for calculating two's complement expressions

1100110 – 1001001 =

1100110 + 0110110 = 0011100

0011100 + 1111011 = 0010111

Answer: 23

C. (5%) Evaluate the following expression in two's complement, perform the bitwise AND between the first two numbers, then add the result to the third number, and finally convert the result to decimal: **(1011101 AND 1001010) + 1110001**

1011101 AND 1001010 = 1001000

1001000 + 1110001 = 0111001

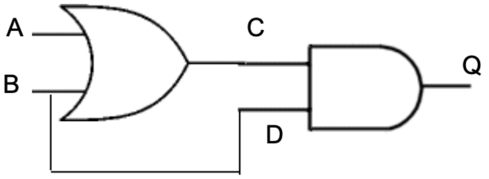
= 57

Part II: Digital Logic

A. (5%) Fill in the truth table for the following logical expression:
(q OR r) AND (NOT(p) OR NOT(q))

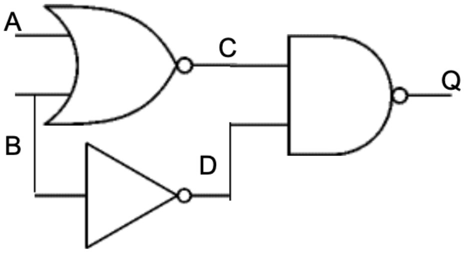
p	q	r	NOT(p)	NOT(q)	q OR r	NOT(p) OR NOT(q)	(q OR r) AND (NOT(p) OR NOT(q))
1	**0**	**0**	**0**	**1**	**0**	**1**	**0**
0	**1**	**0**	**1**	**0**	**1**	**1**	**1**
0	**0**	**1**	**1**	**1**	**1**	**1**	**1**
1	**1**	**0**	**0**	**0**	**1**	**0**	**0**
0	**1**	**1**	**1**	**0**	**1**	**1**	**1**
1	**0**	**1**	**0**	**1**	**1**	**1**	**1**
1	**1**	**1**	**0**	**0**	**1**	**0**	**0**
0	**0**	**0**	**1**	**1**	**0**	**1**	**0**

B. (4%) The figure below shows a logic circuit and its truth table. Complete the below truth table.



A	B	C	D	Q
0	0	0	0	0
1	0	1	0	0
1	1	1	1	1
0	1	1	1	1

C. (5%) The figure below shows a logic circuit and its incomplete truth table. Complete the below truth table:



A	B	C	D	Q
0	0	1	1	0
1	1	0	0	1
1	0	0	1	1
0	1	0	0	1

Part III: Von Neumann Model

A. (4%) The table below shows the memory from a von Neumann model counter. If the program counter contains 10100, state the data that will be placed in **Memory Address register** and in **Memory Data register.**

Assuming it's before the next instruction phase, and NOT between copying the contents of the PC in the MAR and Incrementing the PC, the contents will be as follow

- State the data that will be placed in Memory Address register?
  Answer: 10011
- State the data that will be placed in Memory Data register
  Answer: 10101111

Address	Contents
10001	11001101
10010	11110001
10011	10101111
10100	10000110
10101	00011001
10110	10101100

B. (5%) Below are the key stages in a typical von Neumann cycle, but they are listed out of order. Number the stages from 1 to 6 to correctly sequence them according to their occurrence in the cycle.

Stage	Sequence number
Execute the instruction fetched	5
Decode the instruction	4
Increment the program counter	1
Fetch the instruction from memory	3
Store results in the memory or registers	6
Evaluate the address of the instruction	2

C. (4%) Which component of the von Neumann architecture is primarily responsible for executing operations on data?

Choose the correct answer from the following options:

A) Memory

B) Control Unit

C) Arithmetic Logic Unit (ALU)

D) Input/Output

Answer: C) Arithmetic Logic Unit (ALU)

Part IV: LC-3 and Assembly

A. (5%) Given the following assembly code snippet, what are the memory addresses corresponding to the symbols **TAG1**, **STORAGE**, and **ARRAY**?

```
.ORG X3100
   JMP TAG1
   NOP
TAG1:   ADD R4, R5, R6
      BRnz STORAGE
STORAGE: .FILL XFFFF
ARRAY:   .BLKW 5
```

Answer:

Symbol	**Adress**
TAG1	X3102
STORAGE	X3104
ARRAY	X3105

B. (5%) Below is an assembly program that attempts to load and store values into memory locations labeled **X**, **Y**, and **Z** and then operate on these values. There are several errors in this code. For each, describe the error and indicate whether it will be detected at assembly time or run time.

```
.ORG X4000
   LD R1, X
   LD R2, Y
   ADD R3, R1, R4
   ST R3, Z
   AND R1, R1, #0
   ADD R1, R1, R2
   ST R1, RESULT
   BRnzp START
X:  .FILL X1001
Y:  .FILL X1002
```

```
Z:  .FILL X1003

RESULT: .BLKW 1

START: .STRINGZ "Restart"
```

Problems:

Assuming the Program wishes to operate on the contents of the memory addresses filled at

Label X, Y. The use of LD and instruction is wrong, as the memory addresses x1001, x1002, x1003 would be loaded into the registers rather than the contents of those memory addresses. LDI instructions would be the correct choice in this case. These are errors that will be detected during runtime, as its valid to still use LD but it will not give the expected result.

The same problem happends when using ST R3, Z. Here the contents of R3 will simply be stored at address Z, not at the memory address given at label Z. Here a STI instruction could have been used. The error will again be detected at runtime when evaluating the register contents during the execution.

C. (5%) Write an assembly program to divide a number by **3** by repeated subtraction. Store the quotient in **R4** and the remainder in **R5**.

Code:

```
     .ORIG x3000

     AND R4, R4, #0 ; Quotitent

     AND R1, R1, #0

     ADD R1, R1, #3 ; Number to divide with

     LD R2, NUMBER ; Operand

     ADD R5, R2, #0 ; Copy the number into R5

     NOT R3, R1          ; Convert 3 -3 in 2s compl

     ADD R3, R3, #1

LOOP ADD R5, R2, R3 ; Remainder

     BRn NEGATIVE

     BRz INC_QUOT
```

```
        ADD R4, R4, #1

        BRnzp LOOP


NEGATIVE NOT R5, R5  ; Get the positvie value of the remainder

         ADD R5, R5, #1

         BRnzp END


INC_QUOT ADD R4, R4, #1

END    HALT

NUMBER     .FILL 8
```

Part V: Process and Disk

A. (5%) Consider the set of **5** processes with their arrival times and burst times given below. Assuming that the CPU scheduling policy is **Round Robin** with a time quantum of **3 units**, calculate the average turnaround time.

Process	Arrival time	Burst Time
P1	**1**	**8**
P2	**2**	**6**
P3	**3**	**4**
P4	**4**	**5**
P5	**5**	**2**

P1	P2	P3	P4	P5	P1	P2	P3	P4	P1
4	7	10	13	15	18	21	22	24	26

P1 Turnaround = 26 – 1 = 25

P2 Turnaround = 21 – 2 = 19

P3 Turnaround = 22 – 3 = 19

P4 Turnaround = 24 – 4 = 20

P5 Turnaround = 15 – 5 = 10

Answer: AVG turnaround time = (25 + 19 + 19 + 20 + 10)/5 = 18,6

(5%) Consider a disk with a rotational speed of **7,200 RPM**, a track capacity of **600 sectors**, and an average sector size of **512 bytes**. Calculate the average access time for this disk, including the seek time, rotational latency, and transfer time, given the following conditions: Average seek time is estimated at **9 ms**. Data transfer rate is **100 MB/s**.

Tacces is defined as = Tavg seek + Tavg rotation + Tavg transfer

Tavg seek given at 9ms

Tavg rotation = 1/2 * 1/7200 x 60 x 1000 = 4.167ms

Tavg transfer 1/7200 x 1/600 x 60 x 1000 = 0.014ms

Answer: Tacces = 9ms + 4.167ms + 0.014ms = 13.181ms

B. (4%) Consider a system with three processes **P1, P2** and **P3** each requiring disk access to complete their operations. The seek time to the disk sectors they need access to vary due to their locations on the disk.

The seek times are as follows:

P1: 8 ms

P2: 10 ms

P3: 5 ms

Additionally, the system uses a scheduling algorithm that prioritizes processes based on the shortest seek time first (SSTF) for disk access.

Order of access: P3 -> P1 -> P2

Answer: Total disc access time = 5ms + 8ms + 10ms = 23ms

Part VI: Virtual memory

A. (5%) Assume we have a virtual memory system defined as follows:

The virtual memory size is 64 GB.

The physical memory size is 8 GB.

The page size is 4 KB.

Tasks:

Calculate the number of bits needed for the virtual page number (VPN).

64 GB = 2^36 bytes

4kb = 2^12 bytes/page

Pages = 2^36 – 2^12 = 2^24

Answer: 24 bits are required for the VPN.

Calculate the number of bits needed for the physical page number (PPN).

8 GB = 2^32 bytes

4 KB = 2^12 bytes/page

Pages = 2^32 – 2^12 = 2^20

Answer: 20 bits are required for the PPN

Calculate the page offset.

We know that 2^12 bytes/page so

2^15bits bits/page

Answer: The page offset is 15 bits.

Determine how many entries there are in the page table assuming a fully associative page table.

Answer: There are 2^24 entries as each VPN needs an entry in the page table

B. (5%) Given the following page reference string, and a memory capacity of 3 frames, calculate the page hits and page misses using the LRU replacement algorithm.

Page Reference String: 4, 2, 5, 1, 3, 2, 4, 1, 5, 3, 2, 5, 1

Str	4	2	5	1	3	2	4	1	5	3	2	5	1
Fr1	4	4	4	2	5	1	3	2	4	1	5	3	2
Fr2		2	2	5	1	3	2	4	1	5	3	2	5
Fr3			5	1	3	2	4	1	5	3	2	5	1
Hits	0	0	0	0	0	0	0	0	0	0	0	1	1
Misses	1	2	3	4	5	6	7	8	9	10	11	11	12

Answer: 1 hit and 12 misses

C. (5%) Which of the following statements accurately describes the function of **virtual memory** in a computer system?

Options:

A) Virtual memory allows the system to use hard disk space to simulate extra RAM, effectively increasing the amount of memory available for applications to use.

B) Virtual memory limits the amount of RAM accessible to the operating system, thereby enhancing the system's processing speed.

C) Virtual memory is a specialized hardware feature that directly maps video memory to the CPU for faster graphics processing.

D) Virtual memory decreases the overall system performance by reducing the frequency of data exchange between the CPU and the RAM.

Answer: A

Part VII: Multithreading

A. (5%) Explain what the following Java code does and identify any potential issues related to thread execution.

```
public class MyThread extends Thread {
    public void run() {
        for (int i = 0; i < 5; i++) {
            System.out.println("Thread running: " + i);
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                System.out.println("Thread interrupted");
            }
        }
    }

    public static void main(String[] args) {
        MyThread t = new MyThread();
        t.start();
        t.start();
    }
}
```

The run method loops 5 times through a try-catch block where the thead try to sleep (wait) for a second, whereafter it will go to the next iteration of the for-loop

If an interruptedException is thrown while sleeping in the try block, it will be catched in the catch block and print out that it's been interrupted to the system output.

In the main method it is attempted to start t twice sequentially, which will cause an error as the thread is still running while the 2nd method call on start is called.

B. (5%) Write a Java program that uses a semaphore to synchronize access to a common resource between multiple threads. Assume you have **3 threads** that need to access a shared resource called **SharedConfig** which can only be accessed by one thread at a time.

I’ve programmed in C during the course.

C. (5%) Which statement is true about the **ReentrantLock** in **Java**?

Options:

A) A ReentrantLock cannot be locked again by the thread that already holds the lock.

B) A ReentrantLock allows the thread that holds the lock to re-enter the same block of code that is locked by the same lock object.

C) ReentrantLock does not allow the lock holder to conditionally wait on a specific condition.

D) ReentrantLock automatically releases the lock when the thread that holds the lock terminates.

Answer: B)