DELFT UNIVERSITY OF TECHNOLOGY EEMCS

TU Delft

Midterm Computer Organisation CSE1400

2021-09-30

Please read the following information carefully! (the same information is on exam and on answersheet)

- Opening the exam and/or answerform before you are instructed to start is **strictly prohibited**.
- This exam consists of 13 questions. The number of points each question is worth is written next to the question (the complete exam is 4000 points).
- You have **90 minutes** to complete this exam.
- Before you hand in your answers, check that your answer form contains your name and student number on every page, also filled in using the boxes.
- Please check carefully which question should be answered where and stick to the indicated space.
- The use of books, notes, calculators, smart watches, and other aids is **strictly prohibited**.
- Any answer that you provide to an open question should be backed up by an explanation and/or the accompanying math!
- Fill in the answer form with blue/black **pen**. If you make a mistake on the answer form, there is extra space on the back. If you use this: **indicate this clearly in the original question box that your answer is or continues in the extra space.**
- You can solve the exercises in any order, and the order of the exercises is **not** determined by their difficulty.

.

1. (200 points) As we all know, the ENIAC in 1943 consumed roughly 150 kW of power and weighs 30 tons. In 1996, they made a sort of "replica" or tribute called "ENIAC-on-a-chip". This chip could do the exact same thing, but was only a few gram and measured less than 1 square centimetre.

   Explain in a few short sentences what invention(s) made this possible.

   Solution: - transistors: small, reliable, and less power

   - integrated circuits: combine many transistors in one compact microprocessor package

2. (300 points) Consider the following two boolean algebra formulas:

$$f_1(x, y, z) = \bar{a}\bar{b}c + b\bar{c} \quad (1)$$

$$f_2(x, y, z) = a\bar{c} + b\bar{c} \quad (2)$$

   For which inputs do they give different outputs? Show your work!

   Solution: $x = 0, y = 0, z = 1$ and $x = 1, y = 0, z = 0$

x	y	z	f_1	f_2
0	0	0	0	0
0	**0**	**1**	**1**	**0**
0	1	0	1	1
0	1	1	0	0
1	**0**	**0**	**0**	**1**
1	0	1	0	0
1	1	0	1	1
1	1	1	0	0

3. (300 points) Otto, Thomas and Stefan were fighting for the honorary title of "Candy King", and Thomas was slowly taking the lead with his weekly candy deliveries to the TAs. Otto realised that in order to make a comeback, he had no choice but to bring out his secret weapon: Eugenia[1] biscuits. Unfortunately, while he was dropping off a package of goodies, the bottom of it broke and covered one of Otto's Karnaugh maps with Eugenias.

Otto quickly realised that this event created an excellent exam question and he took a picture of the Karnaugh map. Now he tasks you with filling in the spots that were covered by the Eugenias.

You also know that the minimum **Sum of Products** for this Karnaugh map was:
$(\overline{A} \cdot \overline{B} \cdot \overline{D}) + (B \cdot \overline{C}) + (\overline{B} \cdot C \cdot D)$.

Draw the original Karnaugh map and show your work.

CD \ AB	00	01	11	10
00	1	1	1	0
01	0	1	1	0
11	1	0	0	1
10	1	0	0	0

Solution:

4. (300 points) For the following conversions, decide whether they are possible or not. If yes, give the resulting conversion. **You do not need to show your work.**

[1]Eugenias are sandwich biscuits filled with chocolate cream and rum flavour. These treats were famous with Romanian children because of the chocolate filling, and they were popular with the parents too, because they didn't cost too much.

(a) Convert the number 222_{10} into the following representations:

i. 8-bit 1's Complement

Solution: Not possible

ii. 12-bit BCD

Solution: `0010 0010 0010`

(b) Convert the number -222_{10} into the following representations:

i. 10-bit Sign & Magnitude

Solution: `10 1101 1110`

ii. 16-bit 2's Complement

Solution: `1111 1111 0010 0010`

5. (300 points) Max and Amy now also realise that the `public static double gramsToPounds()` they implemented yesterday, was implemented in a very inefficient way. Amy remembers that the original formula was: $\frac{weight}{1000} * 2.205$ and now proposes the following change: $\frac{weight}{1024} * 2.25$.

Give at least one advantage and one disadvantage of Amy's method.

Solution: advantage: division by a power of 2 is faster because we can do a subtraction in the exponent instead of a floating point division; the mantissa also has only 2 bits (of which one is hidden), which can make operations on it slightly faster

disadvantage: losing precision

6. (350 points) ISAac the architect wants to make an instruction set for his friend Ana the computer scientist, and he wants it to have as many instructions as possible. Unfortunately the computer he wants to use only supports instructions of at most 80 bits and ISAac wants to have instructions that can access 61 registers, have 2 memory operands which can access 420 MiB of byte-addressable memory and use hard coded integer values that are within the range of [-85,42].

Each instruction has the same length and should follow this format:

OP code	register	direct	direct	immediate

ISAac needs your help to figure out:

(a) **Which number representation to use** for the hard coded values, such that the number of instructions is the largest? Explain why.

(b) **How many instructions** can this ISA have?

Hint: to have the largest number of instructions possible you will need to figure out which number representation uses the least number of bits in this case

Solution:

1. Excess-x (can be specified as Excess-85), or anything that makes it a range of 128 values that use 7 bits to be represented.
2. 61 registers => 6 bits for reg operand
   128 immediate values => 7 bits for imm operand
   420 MiB => $2^9 \cdot 2^{20}$ B => 29 bits for each dir operand
   7 + 6 + 29 + 29 = 71 bits => 9 bits for opcode (80 - 71 = 9)
   9 bits => $\mathbf{2^9 = 512}$ **instructions**

7. (300 points) The combinatorial circuit shown in Figure 1 implements which of the logic formulas below?

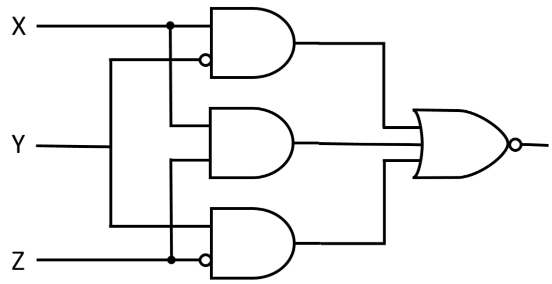


Figure 1: A combinatorial circuit.

A. $(\overline{X} + Y)(X + Z)(\overline{Y} + Z)$
B. $(\overline{X} + Y)(\overline{Y} + Z)(\overline{X} \cdot \overline{Z})$
C. $(\overline{X} + Y)(\overline{Y} + Z)(\overline{X \cdot Z})$
D. $X\overline{Y} \; + \; XZ \; + \; Y\overline{Z}$

8. (350 points) Amy the hippo (A), Max the marmot (M), and Luke (L) all live in the same house. They heard that smart appliances are all the rage nowadays, so they decide to make a smart heating system which turns on and off automatically depending on who is home. Amy and Max don't care too much about the temperature, but Luke is often cold. When all three are home, they produce enough warmth that heating is not necessary. As such, they devise the following equation for their smart heating:

$H = L \cdot \overline{(A \cdot M)}$

Can you help them design the circuitry? In your drawing, clearly indicate when lines connect and when they do not.

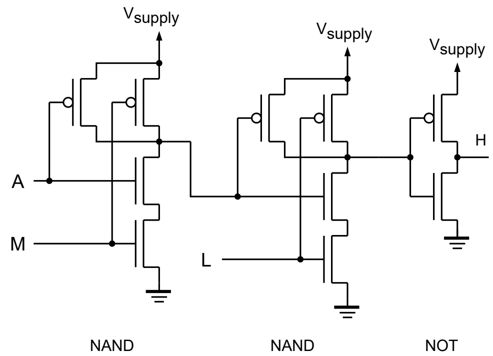
(a) Fill out the truth table for the equation.

(b) Draw a CMOS circuit which implements the equation. Use at most 12 transistors.

Solution:

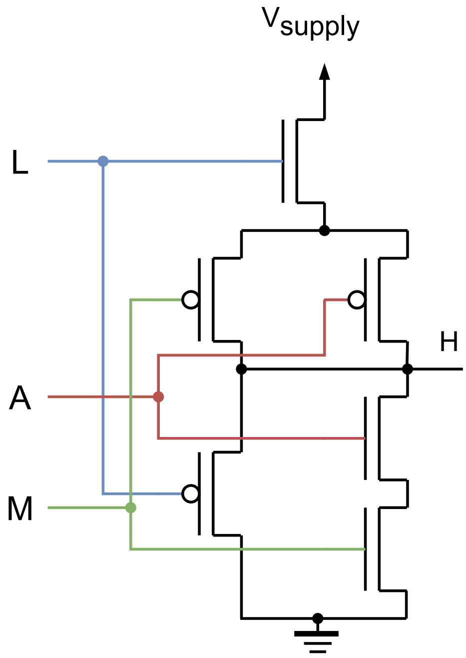
L	**A**	**M**	**H**
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

The correct solution can use a NAND, NAND and a NOT:



$\overline{\overline{L \cdot \overline{(A \cdot M)}}} = L \cdot \overline{(A \cdot M)}$

We also accept a circuit like the following, though according to the book this is technically not allowed (Pull up should use NMOS only and pull down should use PMOS only).



9. (350 points) After yesterdays'[2] issues with weight, Max the Marmot is wondering how much hippopotamuses normally weigh. Luckily, he comes across a small group of 4 and manages to get them all on a scale one by one. To preserve space in his notebook, he stores the weights in excess- −1200 (so excess minus 1200) with 12 bits. He records the following weights:

   1. 0101 1011 0011
   2. 0100 0000 0000
   3. 0000 0000 0000
   4. 0000 0100 1001

   (a) Help Max calculate the average weight of these 4 hippos **while staying in excess- −1200 notation**. Show your work!

   (b) Convert the final average weight from excess- −1200 to a normal radix-10 weight.

Solution: a. add them all up:

```
0101 1011 0011
0100 0000 0000
0000 0000 0000
0000 0100 1001  +
------------------
1001 1111 1100  then to divide by 4: shift by 2
0010 0111 1111
```

b. 1839 kg

10. (300 points)

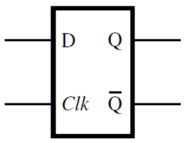
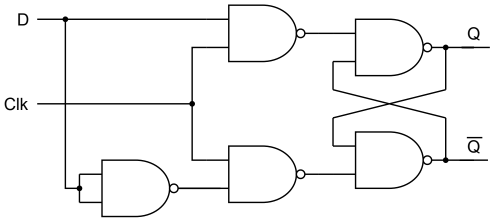


Figure 2: Gated D-latch (graphical symbol).

Sequential circuits, like the gated D-latch shown in Figure 2, are driven by a clock signal to control the timing of their operation (storing a value in case of the gated latch). As with combinatorial circuits they can be built using just universal gates.

Draw a picture implementing a gated D-latch using just *one type* of universal gate (i.e., NAND xor NOR).

[2] for people from the future: yesterday refers to the OOP Midterm of September 2021



Solution:

11. (250 points) Hannah the Hippo went to the store to buy groceries for her birthday cake. To her horror, the store is celebrating radix-day. All the prices are in different radices! Unfortunately, Hannah can only count in hexadecimal (radix-16). Can you help her figure out the total price of her shopping cart **in hexadecimal**? Show your work!

Ingredient	**Price**
flour	101111_2
eggs	306_8
butter	213_{10}
sugar	73_{12}
salt	35_{16}
milk	1132_4

Solution: You can convert everything to decimal:

Ingredient	**Price**
flour	47_{10}
eggs	198_{10}
butter	213_{10}
sugar	87_{10}
salt	53_{10}
milk	94_{10}
sum	692_{10}

And then the sum to hexa: $2B4_{16}$

12. (350 points) For 2 years Mando has been keeping track of how much time Grogu spends on eating frogs. Unfortunately he has been using 2 different time tracking devices (1 for each year), that both use the same special floating point notation, similar to IEEE-754. The format is as follows:

- 1 sign bit
- 7 bit exponent, in excess-64
- 8 bit mantissa (with 1 hidden bit - not included in these 8)

The total amount of time Grogu spent in the first year is **0 1000010 11110000** s and in the second year he spent **0 1000011 10101100** s. Now Mando wants to sum up these two numbers and display them on a clock-like device, which uses BCD as the number representation.

Hint: the easiest way might be to transform the numbers into radix-10 before transforming them into the BCD format

Solution:

1.11110000 with exp $= 2^2$ (1000010) $=>$ 111.110

1.10101100 with exp $= 2^3$ (1000011) $=>$ 1101.011

```
  111.110   (= 7.75)
+ 1101.011  (= 13.375)
------------------------
  10101.001 (= 21.125)
```

0010	0001	.	0001	0010	0101

13. (350 points) Consider the following piece of Assembly code and the current state of the stack:

```
1  popq    %rdi
2  call    foo
3
4  pushq   $74
5  pushq   %rax
6
7  movq    16(%rsp), %rax
8
9  foo:
10         pushq   %rbp
11         movq    %rsp, %rbp
12
13         subq    $16, %rsp
14         movq    %rdi, (%rsp)
15         movq    $44, -8(%rbp)
16
17         popq    %rax
18
19         pushq   $400
20
21         movq    %rbp, %rsp
22         popq    %rbp
23         ret
```

Address	Content	Pointers
2400	"HAHAHAHA"	RBP
2392	13	
2384	42	RSP
2376	"HelloWor"	
2368	"ld! Hehe"	
2360	928	
2352	134	
2344	64	

Table 1: State of the Stack

(a) What is the state of the stack after line 7 is executed? *Don't forget to include the pointers!*

Solution:

Address	Content	Pointers
2400	"HAHAHAHA"	RBP
2392	13	
2384	74	
2376	42	RSP
2368	44	
2360	400	
2352	134	
2344	64	

(b) What value does `%rax` have after line 7 is executed?

Solution: 13

End of exam