

Test of Pearl 000 — Binary logic and computer architecture
Pearls of Computer Science (202001022)
Bachelor module 1.1, Technical Computer Science, EWI
September 11, 2020, 13:45–14:45

Module coordinator: Doina Bucur
Instructor: Pieter-Tjerk de Boer

- You may use 1 A4 document with your own notes for this exam and a *simple* calculator.
- Scientific or graphical calculators, laptops, mobile phones, books etc. are not allowed.
Put those in your bag now!
- Questions marked with MC must be answered on the separate multiple-choice form, at the number indicated in the circle.
- Other questions have a box in which you can write the answer on this paper; this paper must be handed in.
- Total number of points: 100.
Total number of pages: 6.

Your name:

Your student number:

1. Binary numbers

4 pt

(a) What is the decimal number +4 expressed as a 5-bit 2-complement binary number?

- A. 00011
- B. 00100
- C. 00101
- D. 01000
- E. 10100
- F. 10011
- G. 11000

(Feel free to use the open space besides MC questions for scratch)

MC01

4 pt

(b) Considering the 1-complement and 2-complement number representation scheme(s), which has/have the property that the left-most bit indicates the sign? (ignore the number 0, as it has no sign)

- A. Neither of them.
- B. Only the 1-complement scheme.
- C. Only the 2-complement scheme.
- D. Both of them.

MC02

4 pt

(c) Convert decimal 2592 to hexadecimal.

- A. A20
- B. A02
- C. 0A2
- D. 2A0
- E. 02A
- F. 20A
- G. none of the above

MC03

- 4 pt (d) Convert the 2-complement hexadecimal number F0 to decimal:
(i.e., first convert to 2-complement 8-bit binary and then to decimal)
- A. -240
 - B. -16
 - C. -15
 - D. 0
 - E. 15
 - F. 16
 - G. 240

MC04

- 4 pt (e) Suppose we shift a 4-bit unsigned binary number X one position to the left; we fill the gap at the right by a 0, and the left-most bit is dropped so the result again fits in 4 bits. Which statement is true?
- A. This calculates $2X$, for any X
 - B. This calculates $2X$ correctly if and only if $X \geq 0$
 - C. This calculates $2X$ correctly if and only if $X > 0$
 - D. This calculates $2X$ correctly if and only if $X > 1$
 - E. This calculates $2X$ correctly if and only if $X < 7$
 - F. This calculates $2X$ correctly if and only if $X < 8$
 - G. This calculates $2X$ correctly if and only if $X < 15$

MC05

- 4 pt (f) What does “positional” notation mean?
- A. Each digit indicates its own position.
 - B. The position of a digit indicates its weight.
 - C. Each digit has a position that does not change.
 - D. A digit changes when moved to a different position.
 - E. The sign is determined by the digit in the first position.

MC06

2. Boolean logic

- 4 pt (a) Give the truth table of a “conditional inverter”: its output is the same as input X if input Y is 0, and it is the inverse of input X if input Y is 1. Note that you have to answer 4 multiple choice questions here: for 0 choose A, for 1 choose B.

X	Y	output
0	0	MC07
0	1	MC08
1	0	MC09
1	1	MC10

12 pt (b) Consider the following derivation in Boolean algebra. For each step, indicate on the multiple-choice form which relationship is used, using the following options:

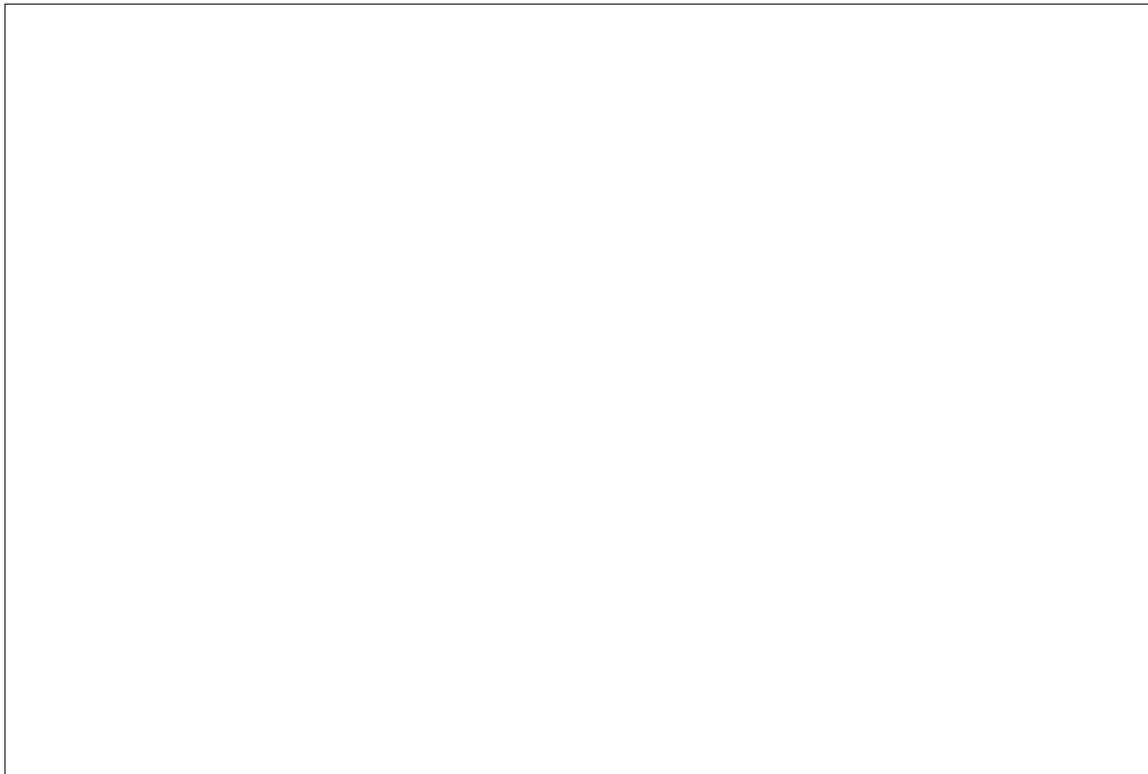
- A commutative property
- B identity property
- C complement property
- D distributive property
- E DeMorgan's theorem
- F this step is not correct

(Note: check each step for equality only to the previous line. So if one step is incorrect (option F), the next step by itself can still be correct, even though of course it's no longer equal to what we started with.)

$$\overline{\overline{X + Y}} + \overline{Y}X + \overline{\overline{Y + Z}}$$

- MC11 = $XY + \overline{Y}X + \overline{\overline{Y + Z}}$
- MC12 = $XY + \overline{Y}X + \overline{Y}Z$
- MC13 = $XY + X\overline{Y} + \overline{Y}Z$
- MC14 = $X(Y + \overline{Y}) + \overline{Y}Z$
- MC15 = $X \cdot 1 + \overline{Y}Z$
- MC16 = $X + \overline{Y}Z$

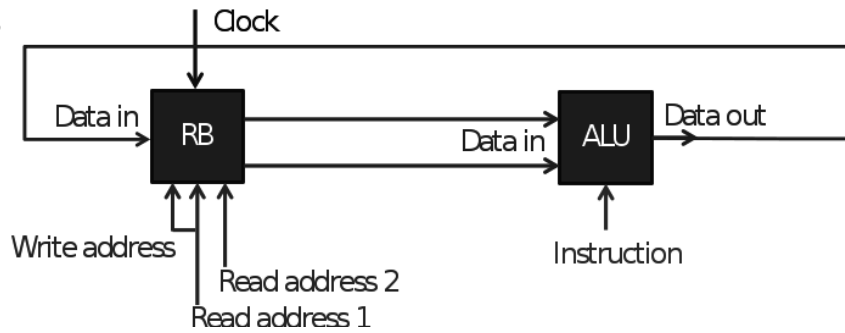
6 pt (c) Sketch a diagram implementing the following formula with only NAND gates: $\overline{\overline{\overline{A + B}}} \cdot C \cdot A$



- 4 pt (d) Suppose you take two 2-input AND gates, and feed their outputs into a third 2-input NAND gate. Does this as a whole work as a 4-input NAND gate?
- A. Yes, it does.
 - B. No, a 4-input NAND gate is not well-defined.
 - C. No, for that we should replace the third NAND gate by an AND gate.
 - D. No, for that we should replace the first two AND gates by NAND gates.
 - E. No, for that we should do both of the above-mentioned changes.
 - F. No, for that we should replace the third NAND gate by a NOR gate.
 - G. No, for that we should replace the first two AND gates by NOR gates.

MC17

12 pt 3. Problem 3



The ALU of the processor above has two instructions: 0 = 'ADD' and 1 = 'MUL'. The register bank (RB) contains four 8-bit registers. Give for this processor the program for computing $R1 \times R2 \times R2 + R1$ and storing the result into R1. (You may not need all timeslots.)

	read address 1 / write address	read address 2	instruction
Timeslot 0			
Timeslot 1			
Timeslot 2			
Timeslot 3			
Timeslot 4			

Continued on next page...

4. Problem 4

Consider the AVR program at the right.

("BRNE" means "BRanch if Not Equal", "SUB" means "Subtract".)

Assume that each instruction takes 1 clock cycle, except jumping to a different address, which takes 2 clock cycles.

```
LDI R17, $02
LDI R18, $07
LDI R19, $01
LDI R20, $04
SUB R18, R17
SUB R20, R17
BRNE -3
SUB R18, R19
BRNE -2
```

18 pt

- (a) Write in the below table the status of the registers after each instruction, one instruction per line. If a register doesn't change from one line to the next, you may leave it blank. If the instruction is a BRNE, use the "branch" column to write down whether a jump is performed, and if so, to where (e.g., "branch to LDI R19, \$01").

R17	R18	R19	R20	branch

5 pt (b) How many clock cycles does the program (of the previous page) take? Explain.

15 pt **5. Problem 5**

What is the mathematical function that is computed by the code below?

Write as a function of X and Y, e.g. $f(X, Y) = X + Y$, and explain.

Assume that X and Y are larger than 0; the result is available in R19.

```
LDI R17, $X
LDI R18, $Y
LDI R19, $00
LDI R21, $01
label1:
SUB R17, $01
ADD R19, R17
SUB R18, R21
BRNE label1
```

End of this exam. — Scratch space below this line