

Test of Pearl 000 — Binary logic and computer architecture

Pearls of Computer Science (201700139)

Bachelor module 1.1, Technical Computer Science, EWI

September 8, 2017, 13:45–14:45

Module coordinator: Doina Bucur, Maurice van Keulen

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!
- Write your answers on this paper, in the provided boxes , and hand this in.
- Total number of points: 100.
Total number of pages: 7.

Your name:

(please underline your family name (i.e., the name on your student card), so that we know how to sort)

Your student number:

Continued on next page...

1. Binary numbers

7 pt (a) Convert the decimal number -4 to a 6-bit 2-complement binary number. Show your calculation.

7 pt (b) Convert the hexadecimal A2F to decimal, and show your calculation.

4 pt (c) Which of the following operations multiplies a binary number by 9? (one correct answer)

- A. Shift to the left by 9 positions.
- B. Shift to the right by 9 positions.
- C. Shift to the left by 3 positions and add the original (unshifted) number to it.
- D. Shift to the right by 3 positions and add the original (unshifted) number to it.
- E. Shift to the left by 9 positions and add the original (unshifted) number to it.
- F. Shift to the right by 9 positions and add the original (unshifted) number to it.

6 pt (d) Which of the following operations multiplies a 2-complement binary number by -1 ? One or more are correct; select *all* correct ones!

- A. Invert the first bit.
- B. Invert the last bit.
- C. Invert all bits.
- D. Invert all bits, and then add 1.
- E. Invert all bits, and then subtract 1.
- F. Add 1, and then invert all bits.
- G. Subtract 1, and then invert all bits.

Continued on next page...

2. Boolean logic

6 pt

(a) Give the truth table of a 3-input AND/OR-gate: if input C=1, the output is the OR of inputs A and B, otherwise, it is the AND of A and B.

A	B	C	output

6 pt

(b) Suppose you take a 2-input AND gate, and put inverters in front of both inputs. Does this as a whole work as a 2-input OR gate?

- A. No, you can never make an OR gate out of AND gate.
- B. No; but if we also put an inverter at the output, it does.
- C. Yes, and this would also work if the AND gate had more than 2 inputs.
- D. Yes, but this only works for a 2-input AND gate, not for more inputs.

Explain your answer:

8 pt

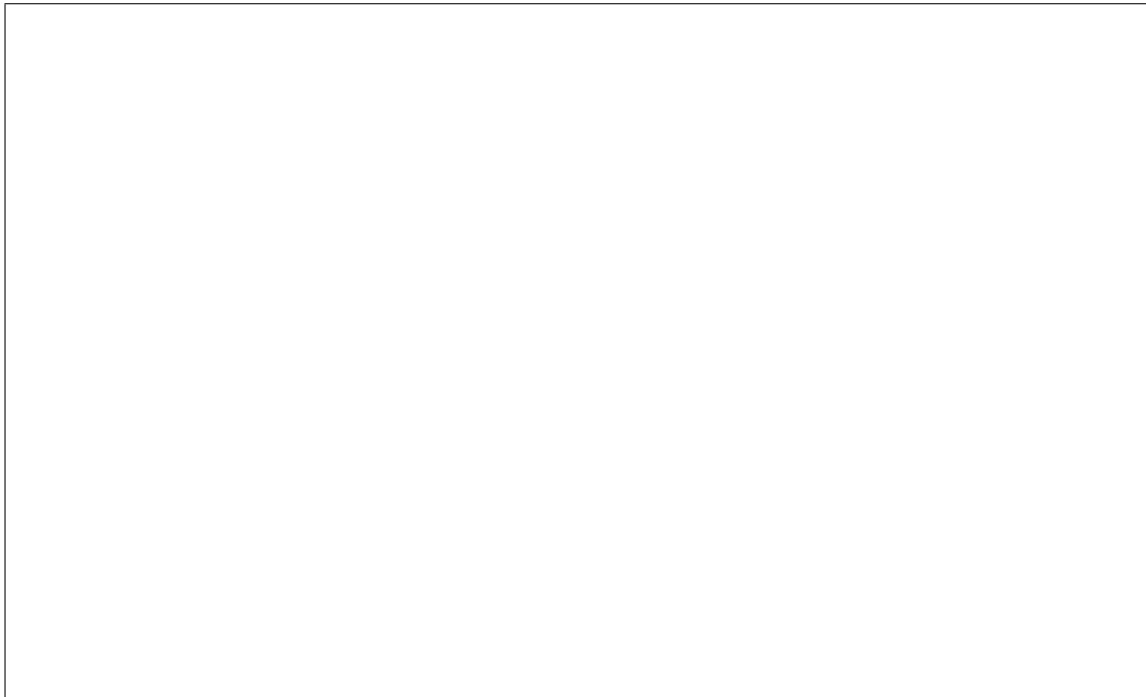
(c) Consider the following derivation in Boolean algebra. Indicate for each (numbered) equals sign which rule is applied, by putting a tickmark (✓) in the appropriate cells of the table. The “wrong” rule is to be chosen if you think that that step is not correct. (It is possible that a rule is used multiple times, or not at all, in this derivation; however, each step uses only a single rule.)

$$(A + \bar{B} + C)(\overline{A + \bar{B}}) \stackrel{(1)}{=} (A + \bar{B} + C)(A + B) \stackrel{(2)}{=} A + (\bar{B} + C) \cdot B \stackrel{(3)}{=} A + \bar{B}B + CB \stackrel{(4)}{=} A + \bar{B}B + BC \stackrel{(5)}{=} A + 0 + BC \stackrel{(6)}{=} A + BC$$

step	commutative	identity	complement	distributive	DeMorgan	wrong
(1)						
(2)						
(3)						
(4)						
(5)						
(6)						

6 pt

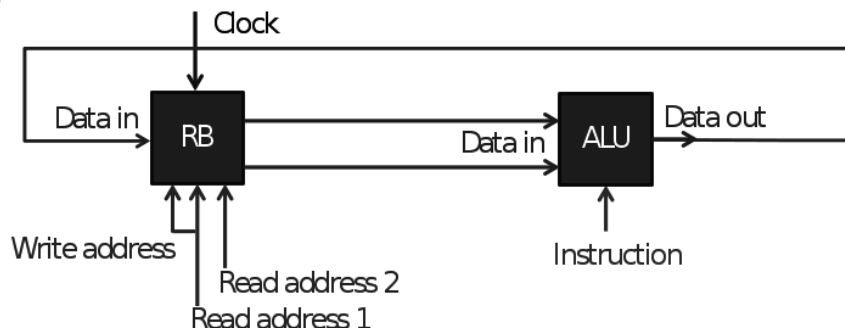
(d) Sketch a diagram implementing the following formula with only NAND gates: $A \cdot (\bar{B} + \bar{C})$



Continued on next page...

15 pt

3. Problem 3



The ALU of the processor above has two instructions: 0 = 'ADD' and 1 = 'MUL'. Furthermore it has 4 8-bit registers. The starting value for register R4 equals 0. Give for this processor the program for computing $R1 + (R2 * R3) + R1$ 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			
Timeslot 5			

Continued on next page...

4. Problem 4

Given this AVR program; "BRNE" means "BRanch if Not Equal", "INC" means "Increment (add 1)", "SUB" means "Subtract".

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

```
LDI R17, $03
LDI R18, $02
LDI R19, $01
LDI R20, $00
ADD R18, R17
SUB R18, R19
INC R19
MOV R21, R19
SUB R21, R17
BRNE -6
```

15 pt

- (a) Fill in the below table with the status of the registers after each instruction; if a register doesn't change from one line to the next, you may leave it blank.

R17	R18	R19	R20	R21

- 5 pt (b) How many clockcycles 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, and the result is available in R20.

```
LDI R17, $X
LDI R18, $Y
LDI R19, $01
label1:
SUB R18, R19
BREQ label2
ADD R17, R17
JMP label1
label2:
MOV R20, R17
```

End of this exam.