Test of Pearl 000 — Binary logic and computer architecture

Pearls of Computer Science (201700139)
Bachelor module 1.1, Technical Computer Science, EWI

September 7, 2018, 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:

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

CORRECT ANSWERS

Your student number:

0000000

---

1. Binary numbers

4 pt (a) What is the decimal number –3 expressed as a 6-bit 2-complement binary number?

MC**01** **G**

A. 100010
B. 100011
C. 100100
D. 101100
E. 111011
F. 111100
G. 111101

4 pt (b) Considering the 1-complement and 2-complement number representation scheme(s), has/have the property that the first bit is 1 if the number is negative, and 0 if it is positive?

MC**02** **D**

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

4 pt (c) Convert hexadecimal 1F0 to decimal.

A. 240
B. 271
C. 272
D. 481
E. 495
F. 496
G. 511

MC**03** **F**

4 pt (d) Convert hexadecimal A1 to binary.

A. 00011010
B. 00010110
C. 00100110
D. 10001010
E. 10100001
F. 10100010
G. 10101000

MC**04** **E**

4 pt (e) Which of the following operations multiplies a binary number by 7?

A. Shift to the left by 3 positions.
B. Shift to the left by 4 positions.
C. Shift to the left by 5 positions.
D. Shift to the left by 2 positions and then add the original (unshifted) number to it.
E. Shift to the left by 2 positions and then subtract the original (unshifted) number from it.
F. Shift to the left by 3 positions and then add the original (unshifted) number to it.
G. Shift to the left by 3 positions and then subtract the original (unshifted) number from it.

MC**05** **G**

4 pt (f) Suppose we have 4-bit adder for unsigned numbers, which has 4 output bits and a "carry" output; the latter is essentially the 5th output bit needed if the sum exceeds 15.
Now assume we want to use this adder to perform additions of 4-bit 2-complement signed numbers. What should we do with the "carry" output to get correct results for both positive and negative numbers?
You may assume that the 2-complement input numbers are such that their sum is in the range –8 to +7, so it can be represented by the adder's 4 normal output bits.

A. Just ignore the carry bit.
B. If the carry bit is a 1, invert the 4 normal output bits.
C. If the carry bit is a 1, invert the most-significant of the 4 normal output bits.
D. If the carry bit is a 1, invert the least-significant of the 4 normal output bits.
E. If the carry bit is a 1, add 1 to the result represented by the 4 normal output bits.
F. If the carry bit is a 1, subtract 1 from the result represented by the 4 normal output bits.

MC**06** **A**

2. Boolean logic

4 pt (a) Give the truth table of a 2-input comparator: if both inputs (X and Y) are equal, the output is 1, and it is 0 otherwise. Note that you have to answer 4 multiple choice questions here; choose A for 0 or B for 1.

X	Y	output
0	0	MC**07** **B**
0	1	MC**08** **A**
1	0	MC**09** **A**
1	1	MC**10** **B**

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

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

$\overline{XY} \cdot (\overline{X} + (Y + Z))$

MC11 **E** $= (\overline{X} + \overline{Y}) \cdot (\overline{X} + (Y + Z))$

MC12 **D** $= \overline{X} + (\overline{Y} \cdot (Y + Z))$

MC13 **D** $= \overline{X} + \overline{Y}Y + \overline{Y}Z$

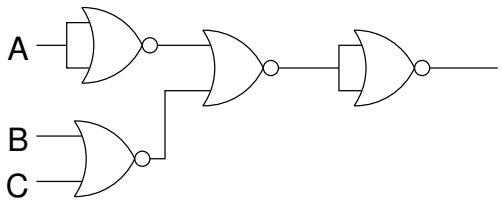
MC14 **C** $= \overline{X} + 0 + \overline{Y}Z$

MC15 **B** $= \overline{X} + \overline{Y}Z$

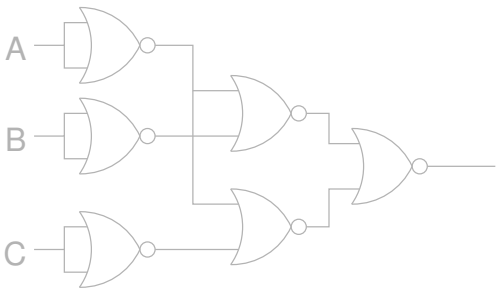
MC16 **F** $= \overline{XY}Z$

6 pt (c) Sketch a diagram implementing the following formula with only NOR gates: $\overline{A \cdot (B + C)}$

Using DeMorgan: $\overline{A \cdot (B + C)} = \overline{A} + \overline{B + C}$, leading to the following circuit:



alternative:



The gray picture at the right shows an alternative solution based on using the distributive property first: $\overline{A \cdot (B + C)} = \overline{AB + AC} = \overline{(\overline{A} + \overline{B})} + \overline{(\overline{A} + \overline{C})}$; it's more complicated, but worth the full points too.

4 pt (d) Suppose you take two 2-input AND gates, and feed their outputs into a third 2-input AND gate. Does this as a whole work as a 4-input AND gate?

MC17 **A**

A. Yes, it does.
B. No, a 4-input AND gate is not well-defined.
C. No, for that we should replace the third AND gate by an OR gate.
D. No, for that we should replace the first two AND gates by OR gates.
E. No, for that we should put inverters at the inputs of the third AND gate.

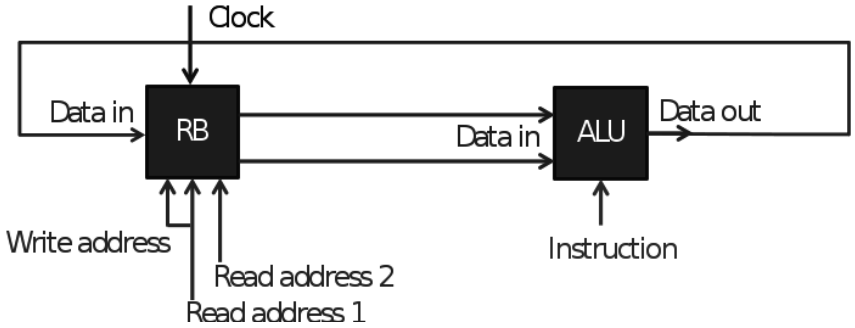
4 pt (e) Suppose you have a lot of 2-input AND gates. Can you make a NOR gate out of this?

MC18 **B**

A. Yes, thanks to DeMorgan's theorem.
B. No, you'd also need inverters.
C. No, you'd also need at least one OR gate.
D. No, you'd also need inverters and at least one OR gate.

---

15 pt **3. Problem 3**



The ALU of the processor above has two instructions: 0 = 'ADD' and 1 = 'MUL'. Furthermore it has four 8-bit registers. Give for this processor the program for computing R1 $\times$ (R2 + R3) + R2 and storing the result into R1. (You may not need all timeslots.)

	read address 1 / write address	read address 2	instruction
Timeslot 0	**3**	**2**	**0**
Timeslot 1	**1**	**3**	**1**
Timeslot 2	**1**	**2**	**0**
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”.

```
LDI   R17, $01
LDI   R18, $04
LDI   R19, $02
LDI   R20, $08
INC   R18
ADD   R18, R19
SUB   R19, R17
MOV   R21, R18
SUB   R21, R20
BRNE  -5
```

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

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	
1					
	4				
		2			
			8		
	5				
	7				
		1			
				7	
				-1	
					BRNE taken
	8				
		0			
				8	
				0	
					BRNE not taken
					(the lines showing the jump (BRNE) instructions may be omitted)

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

16 since 15 instuctions are executed, one of which is a branch that is taken and thus takes 1 extra cycle

---

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
   LDI   R20, $01
label1:
   ADD   R17, R19
   ADD   R19, R20
   SUB   R18, R20
   BRNE label1
   MOV   R20, R17
```

$$f(X,Y) \;=\; X + 1 + 2 + \cdots + Y \;=\; X + \sum_{k=1}^{Y} k \;=\; X + \frac{Y \cdot (Y+1)}{2}$$

Each pass through the loop adds R19 to R17, and increments R19, hence the adding of first 1, then 2, then 3, and so on. Simultaneously, R18 (Y) is decremented; when it reaches zero, we stop.
Full score already for the $X + 1 + 2 + \ldots + Y$ answer.

End of this exam.