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.

111100
Using the appropriate weights, this is $-2^5+2^4+2^3+2^2 = -32+16+8+4 = -4$.

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

$10 \cdot 16^2 + 2 \cdot 16^1 + 15 \cdot 16^0 = 2560 + 32 + 15 = 2607$

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.

C

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.

D,G

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
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

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.

B

Explain your answer:

You should show this using either truth tables, or a derivation using De Morgan's law.

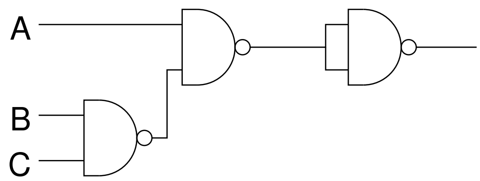
8 pt (c) Consider the following derivation in Boolean algebra. Indicate for each (numbered) equals sign which rule is applied, by putting a tickmark (✓) in the appropriate cells of the table. The "wrong" rule is to be chosen if you think that that step is not correct. (It is possible that a rule is used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

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

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: $A \cdot (\overline{B}+\overline{C})$

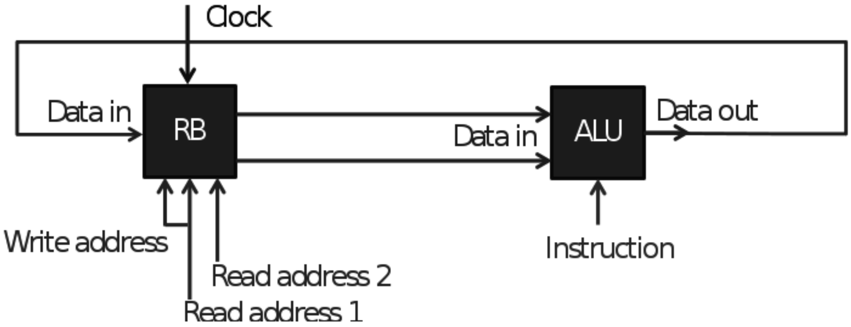
Using DeMorgan: $A \cdot (\overline{B}+\overline{C}) = A \cdot \overline{BC}$, leading to the following circuit:



There's an alternative solution, based on using the distributive property first: $A \cdot (\overline{B}+\overline{C}) = A\overline{B} + A\overline{C} = \overline{\overline{A\overline{B}} \cdot \overline{A\overline{C}}}$.

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	2	3	1
Timeslot 1	2	1	0
Timeslot 2	1	2	0
Timeslot 3			
Timeslot 4			
Timeslot 5			

Many variations are possible which are also correct, e.g., using R3 rather than R2 to store the intermediate result.

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	
3					
	2				
		1			
			0		
	5				
	4				
		2			
				2	
				-1	
					BRNE
	7				
	5				
		3			
				3	
				0	
					BRNE

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

17, since 16 instructions 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
label1:
   SUB   R18, R19
   BREQ label2
   ADD   R17, R17
   JMP label1
label2:
   MOV   R20, R17
```

$f(X, Y) = X \cdot 2^{Y-1}$

Each pass through the loop adds R17 to itself, thus doubling the contents. This is done R18–1 times; the –1 because the decrement and check of R18 is before the doubling of R17.

End of this exam.