Pearls of Computer Science (202001022)

Pearl 000: Binary logic and computer architecture

Test of September 17, 2021

Answers

---

1. Binary numbers

4 pt (a) D.

4 pt (b) E.

4 pt (c) C.

4 pt (d) D.

4 pt (e) A.

4 pt (f) D.
Note that B is wrong because the allowed range includes –128 (but not +128); like the 4-bit example in the lectures covered –8 to +7.

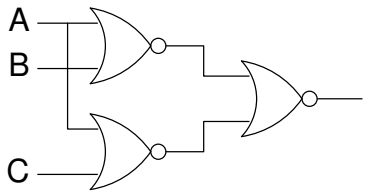
---

2. Boolean logic

4 pt (a) A, A, B, A

12 pt (b) B, F, E, A, D, B or C
In the last step two rules are used: first the complement property to show that $\overline{PQ} + PQ = 1$, and then the identity property saying $1R = R$. This was a mistake in the exam, so we accepted either answer B or C as correct.

6 pt (c) First use DeMorgan to convert the $\cdot$ (AND) into a + (OR): $\overline{A} \cdot \overline{B} = \overline{A + B}$. Thus the whole expression becomes $\overline{\overline{A + C} + \overline{A + B}}$ which contains only NOR operations, so drawing it is straightforward:



4 pt (d) D.

An easy way of seeing that the proposed circuit with three 2-input NOR gates can't work as a 4-input NOR gate, is by noting that going from one of the inputs to the output one goes through two inversions, which cancel each other. So if all inputs are 1, the output will also be 1, and vice versa. That's the opposite of what a single NOR gate does.

Answer D is right. If we replace the first two NOR gates by OR gates, then if at least one of the four inputs is a 1, one of the inputs to the remaining NOR gate will be 1, forcing its output to be a 0, as intended.

Continued on next page...

3. Problem 3

	read address 1 / write address	read address 2	instruction	explanation
Timeslot 0	2	1	1	calculate R1$\times$R2 and store this in R2
Timeslot 1	1	1	0	calculate R1+R1 and store the result in R1
Timeslot 2	1	2	1	multiply R1 (which now contains double the original content of R1) and R2 (which now contains the product of the original R1 and R2) and store this in R1
Timeslot 3				

Be careful not to overwrite the contents of a register prematurely, e.g. by initially storing a result into R1 while you still need R1's original contents later on.

There are other solutions, by observing that the expression can be rewritten as $2\times R1^2\times R2$.

---

4. Problem 4

18 pt (a)

R17	R18	R19	R20	
1				
	1			
		0		
			0	
		1		
		0		
2				
	3			
			0	
				BREQ to ADD R19,R18
		3		
		1		
3				
	6			
			1	
				jump not performed

5 pt (b) 17 since 16 instructions are executed, 1 of which is a branch that is taken and thus takes 1 extra cycle.

Continued on next page...

5. Problem 5

Various possible ways of writing the answer, all good:

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

Each pass through the loop adds both R17 (equal to X) and R18 to R19, and decrements R18 by 1. Thus, first X and Y are added to R19, then X and Y-1, and so on, until R18 reaches 0.