Pearls of Computer Science (202001022)

Pearl 000: Binary logic and computer architecture

Test of September 11, 2020

Answers

1. Binary Numbers

4 pt (a) B.

4 pt (b) D.

4 pt (c) A.

4 pt (d) B.

4 pt (e) F.
As long as the left-most bit is 0, it doesn't contribute to the number, so dropping it after the shifting operation has no influence. Then indeed $2X$ is calculated (since all bits from X move to a position where they have double their original weight).
If the left-most bit is 1, then dropping it of course does make the final result wrong, namely too low by 16. This is the case for $X \geq 8$.

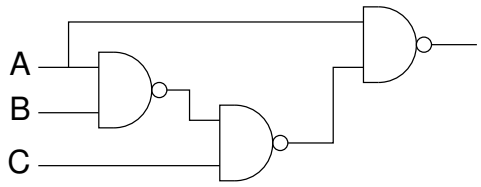
4 pt (f) B.

2. Boolean logic

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

12 pt (b) E, F, A, D, C, B
Note that the second step (MC12) may at first glance look like DeMorgan's theorem, but it isn't.

6 pt (c) First use DeMorgan to convert the + (OR) into a · (AND): $\overline{A} + \overline{B} = \overline{A \cdot B}$. Thus the whole expression becomes $\overline{\overline{\overline{A \cdot B} \cdot C} \cdot A}$ which contains only NAND operations, so drawing it is straightforward:



4 pt (d) A.
The output of an AND gate is 1 only if both inputs are 1. Feeding two such signals into another AND gate makes the latter's output 1 only if all four inputs are 1, so that's a four-input AND gate. Inverting the third gate's output makes the whole thing a four-input NAND.

Continued on next page...

3. Problem 3

	read address 1 / write address	read address 2	instruction	explanation
Timeslot 0	2	2	1	calculate R2×R2 and store this in R2
Timeslot 1	2	1	1	calculate R2 (which now contains the original value of R2 times itself) × R1 and store this in R2
Timeslot 2	1	2	0	calculate R2+R1 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.

Quite many students used a third register (e.g., R3) and apparently silently assumed that this register contained 0 initially. This is not a correct solution, as nothing was given about the initial content of other registers.

---

4. Problem 4

18 pt (a)

R17	R18	R19	R20	
2				
	7			
		1		
			4	
	5			
				BRNE to SUB R18,R17
	3			
			0	
				BRNE not taken
	2			
				BRNE to SUB R18,R19
	1			
				BRNE to SUB R18,R19
	0			
				BRNE not taken

5 pt (b) 19 since 16 instructions are executed, 3 of which are a branch that is taken and thus take 1 extra cycle each.

---

5. Problem 5

Various possible ways of writing the answer, all good:

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

Each pass through the loop first decrements R17 and then adds R17 to R19, hence the adding of first X-1, then X-2, then X-3, and so on. Also, R18 (starting at Y) is decremented at every iteration, and we stop when it reaches zero, so we do the above Y times.