

Nume și Prenume: _____
Grupa: _____

Simulare Examen ASC

~ Tutori: Dumitrache G. Larisa, Furculeșteanu Bianca-Maria ~

- Fie $x=10.25$ și $y=27.5$.
 - Converțiți x și y în baza 2.
 - Converțiți mai departe în baza 16, fără să treceți în baza 10.
 - Calculați $y-x$ lucrând direct în baza 16.
 - Converțiți rezultatul scăderii în baza 10.
 - Determinați reprezentarea internă ca single a lui y , binară și hexa.
 - Calculați $x+y$ folosind algoritmul de adunare în virgulă mobilă pentru formatul single (se va lucra cu reprezentările binare, în notație științifică, și, la final, se va converti rezultatul în baza 10).
 - Interpretați ca nr în baza 10 reprezentarea internă hexa, în virgulă mobilă, 0x8C, considerând formatul dat de $n=8$ (dimensiunea locației) și $k=2$ (dimensiunea câmpului caracteristică).
- Fie $f: B_2^3 \rightarrow B_2^2$, $f(x, y, z)=(f_1(x, y, z), f_2(x, y, z))$; unde $f_1, f_2: B_2^3 \rightarrow B_2$, $f_1(x, y, z)=x*y$, $f_2(x, y, z)=1$ dacă și numai dacă cel puțin 2 dintre variabilele x, y, z au valoarea 0.
 - Construiți tabelul de valori al lui f și scrieți f_1, f_2 în FND și FNC.
 - Implementați f folosind un PROM.
 - Implementați f folosind multiplexori elementari, apoi reduceți la maxim nr multiplexorilor elementari și desenați modelul simplificat.
 - Implementați f folosind un codificator.
- Considerăm implementarea procesorului MIPS cu un singur ciclu. Fie programul:

n: .word 5 lw \$t0, n et: li \$v0, 10
m: .word 6 la \$t2, n sw \$t3, 8(\$t2) syscall
q: .word 1 li \$t3, 5 add \$t3, \$t3, \$t0
.text: lw \$t4, q sub \$t3, \$t3, \$t4
main: li \$t5, 21 bne \$t3, \$t5, et

- Pentru instrucțiunile add, sw și bne scrieți câmpurile din reprezentarea lor internă (ex: rs, rt, opcode etc.) și scrieți-le atât în binar, cât și în hexa.
- Completați tabelul.

	1	3	5	7	8	ALU zero	d	Branch	MR	MW	ALU Src	ALU op (2b)	ALU Ctrl (3b)	PC	Mem[q]
Inițial	-	-	-	-	-	-	-	-	-	-	-	-	-	a	1
sw	a														
bne															

Dacă s-ar mai executa a doua oară instrucțiunea de sw, cât va fi Mem[q]=?

- Adăugați procesorului implementare instrucțiunii **addp rt, rs** ($rt=rt+rs$).

RegDst	ALU Src	Mem to Reg	Reg Write	MR	ALU op (2b)	ALU Ctrl (3b)