

# Tema 1 AC

## Sensors\_input:

Se vor masura pe rand cate o pereche de senzori 1-3, respectiv 2-4. Daca unul din senzori dintr-o pereche are valoare nevalida ( $=0$ ) atunci se anuleaza toata perechea si cu siguranta urmatoarea pereche va avea valori valide conform cerintei. Astfel folosesc 3 if-uri :

1-Vedem daca un senzor din perechea 1-3 este nevalid dupa care facem suma celorlalti 2 senzori,

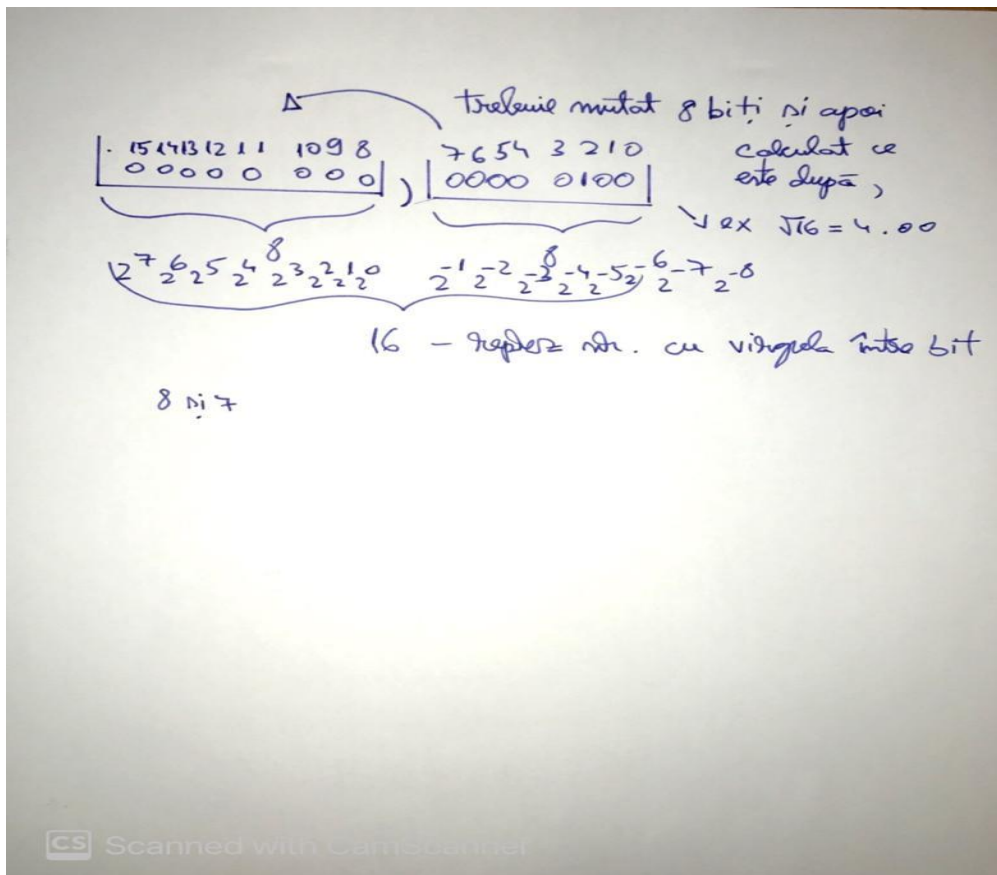
apoi pentru a face rotunjirea inaltimii ne uitam la ultimul bit (daca acesta e 0 atunci numarul este par, iar daca acesta este 1 atunci numarul este impar) in urma conditiei putem avea cazul unul cand este par si impartim direct la 2 pentru a afla media si cazul doi cand este impar adunam 1 si apoi impartim la 2.

2-Asemenea doar ca pentru perechea 1-3 cu 2-4 nevalida.

3-Toti cei 4 senzori sunt valizi fac suma lor, o impart la 2 dupa care aplic aceasi metoda de mai sus pentru a realiza rotunjirea.

La final concatenez toti biti a lui sum pentru a reprezenta height.

## Square\_root:



## Digital - Fast CORDIC

Am luat exemplul  $250 \Rightarrow \sqrt{250} = 15.81$

Rezoluție:  $250 = 128 + 64 + 32 + 16 + 8 + 2$

$$\sqrt{x} = y \Rightarrow y * y = x$$

$$in = 11111010$$

$$out = \underbrace{00001111}_{15} \underbrace{11001111}_{81}$$

virgula între bitul 8 și 7

$$base = 128 = 1000\ 0000$$

$$128 \times 128 = 16384$$

$$y = 0000\ 0000\ 0000\ 0000$$

i = 0:

$$y = 0000\ 0000\ 1000\ 0000$$

$$\text{if } (1000\ 0000\ 0000\ 0000 > 11111010)$$

$$y =$$

$$base \gg 1 \Leftrightarrow 0100\ 0000$$

i = 1

Ne comparăm bitii pe rând

i = 8

$$\text{iese din for} \Rightarrow y = 0000\ 0000\ 0000\ 1111$$

Înțelegem 8 bitii la sfârșitul rezultatului pt a

face partea întregă ca în out

iau iar  $base = 128$  deoarece s-a modificat din urmă  
 Voi salva într-o nouă variabilă în de 16 biti  
 shifting la stânga cu 8 biti =)

in - new = 1111 1010 0000 0000

folosesc o buclă while, iar pentru a fi nimerizată  
 voi avea un număr fix de iterații după cum  
 e prezentat în laborator

aplic în principiu același cod de mai sus cu fân  
 aleg o variabilă aux  $Y = y * y$  pe 32 de biti pe care  
 o shiftz după ce  $* y$  - ul rezultat din for-ul  
 precedent

i=0  
 028 if  $\gg 8$   
 0000 1111 0000 0000 \* 0000 1111 0000 0000  $\gg 8$   
 1111 1010 0000 0000

728

la final vom avea  $y = p. \text{intregă}$ , 11001111

comentariu

# CORDIC

$$b_{\text{max}} = 128 = 1000\ 0000$$

$$2^7$$

$$\sqrt{x} = y$$

$$y * y = x$$

$$\sqrt{0} = 0$$

Ex:  $y = \overset{1\ 2\ 3\ 4}{0000} \overset{5\ 6\ 7\ 8}{0000} \quad x = 0$

prima iteratie din algoritmul

$$y = \overset{1\ 2\ 3\ 4}{1000} \overset{5\ 6\ 7\ 8}{0000} \text{ - iteratii}$$

$$\text{dacă } 1000\ 0000 \times 1000\ 0000 > 0000\ 0000$$

$$y -= b_{\text{max}}$$

$$y = 0000\ 0000$$

$$b_{\text{max}} \gg 1 \Leftrightarrow 0100\ 0000$$

a 2a iteratie

$$y = 1000\ 0000$$

$$\text{de. } 0100\ 0000 \times 0100\ 0000 > 0000\ 0000$$

$$y = 0000\ 0000$$

$$b_{\text{max}} \gg 1 \Leftrightarrow 0010\ 0000$$



$$\sqrt{16} = 4$$

$$y = 0000 \ 0000 \quad x = 0001 \ 0000$$

prima itera

$$\textcircled{1} \quad \text{if } 1000 \ 0000 \times 1000 \ 0000 > 0001 \ 0000$$

$$y = 0 \dots$$

$$\text{banc} \gg 1 \Leftrightarrow 0100 \ 0000$$

$$\textcircled{2} \quad \text{if } \begin{matrix} y = 0100 \ 0000 \\ 0100 \ 0000 \times 0100 \ 0000 > 0001 \ 0000 \\ \vdots \\ 0010 \ 0000 \end{matrix}$$

$$\textcircled{3} \quad \begin{matrix} y = 0010 \ 0000 \\ \text{if } \dots > \dots \text{ da} \\ \vdots \\ 0001 \ 0000 \end{matrix}$$

④  $y = 0001\ 0000$   
 if ... > ... da  
 ...  
 0000 1000

⑤ ~~if~~

⑥  $y = \underline{00000100}$   
 if .16 > 16 nu

base >> 1  $\Leftrightarrow$  00000010

⑦  $y = 0000\ 0100 + 0000\ 0010$

if 0000 0110  $\widehat{x}$  > 16 ~~A~~

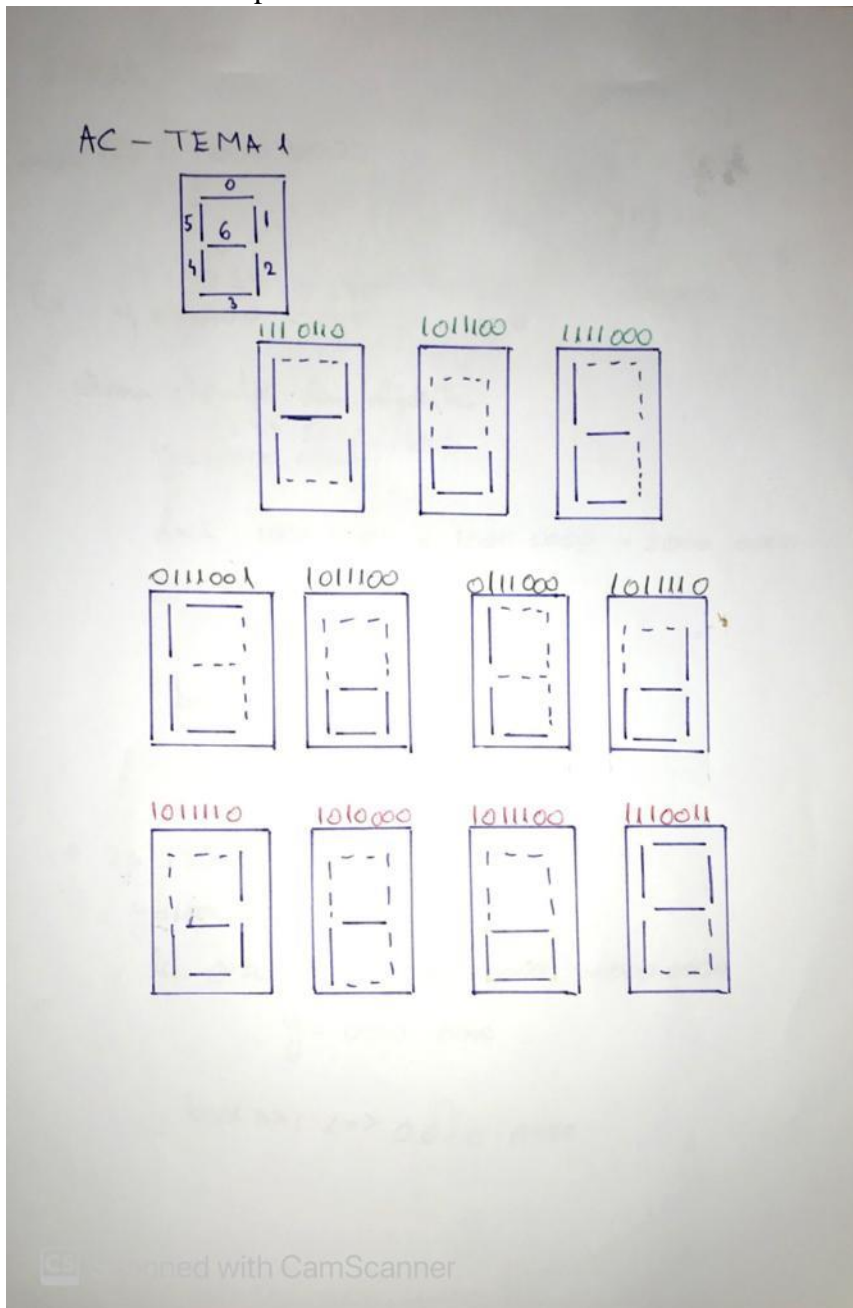
$y = \text{base}$

...

⑧ return  $y = \underline{0000\ 0100}$

**Display\_and\_drop:**

Am declarat 4 variabile de tip reg pentru cei 4 segmenti si la final i-am atribuit continuu pentru cei din output ca in laborator. Prima oara am luat o variabila de tip reg initializata cu 0 deoarece o sa am nevoie cand voi face comparatia dintre partea intreaga si fractionara a lui  $t\_act$  cu  $t\_lim$ . Pentru comparatie am concatenat primii 8 biti de la  $t\_act$  si primii 8 biti de la  $t\_lim$ , acestia reprezentand partea intreaga, respectiv urmatoorii 8 biti de la  $t\_act$  si ceilalti 8 biti de la  $t\_lim$  ramasi reprezentand partea fractionara. Am exprimat si afisat pe 7seg fiecare situatie prezentate in tabelul din tema prin 3 if-uri.

**Baggage\_drop:**

Am atribuit continuu unei variabile auxiliare timpul final impartit la 2. Specific semnalele din diagrame prin wires.