

Tema 1 - Baggage drop

realizată de Mihăilă Iuliana-Raluca
grupa 333AA

sensors_input

Conform cerinței, dacă un senzor este 0, acesta și perechea lui se exclud din calcul. Perechile de senzori sunt 1-3 și 2-4 și se garantează faptul că nu există senzori din ambele perechi cu valoarea 0 în același timp. Media acestora este aproximată la cel mai apropiat întreg.

Am declarat 3 variabile de **tip reg** pentru a le putea face **atribuiri blocante** în interiorul unui **bloc always**. Astfel, pentru realizarea mediei senzorilor, există trei variante (implementarea am realizat-o prin construcția **if**, folosind **teorema împărțirii cu rest**) :

1. Când unul sau ambii senzori din perechea 1-3 este/sunt 0:
 - Se calculează câtul și restul mediei celeilalte perechi de senzori
 - Apoi, folosind construcția **case**, i se va atribui mediei o valoare în funcție de rest:
 - dacă restul este 0 ('b0), mediei îi este atribuită valoarea câtul calculat anterior
 - dacă restul este 1 ('b1), mediei îi este atribuită valoarea întreagă următoare, deoarece, prin împărțirea la 2, restul 1 înseamnă 0,5 și trebuie aproximat
2. Când unul sau ambii senzori din perechea 2-4 este/sunt 0:
 - Se va proceda similar cu cazul anterior.
3. Când toți cei patru senzori au valoare nenulă:
 - Se calculează câtul și restul mediei tuturor senzorilor
 - Pentru acest **case** vor fi 4 cazuri:
 - dacă restul este 0 ('b0), mediei îi este atribuită valoarea câtul calculat anterior
 - dacă restul este 1 ('b1), mediei îi este atribuită valoarea câtul calculat anterior, deoarece, prin împărțirea la 4, restul 1 înseamnă 0,25
 - dacă restul este 2 ('b10), mediei îi este atribuită valoarea întreagă următoare, deoarece, prin împărțirea la 4, restul 2 înseamnă 0,5 și trebuie aproximat
 - dacă restul este 3('b11), mediei îi este atribuită valoarea întreagă următoare, deoarece, prin împărțirea la 4, restul 3 înseamnă 0,75 și trebuie aproximat

În acest modul am făcut o atribuire continuă (**assign**) pentru output.

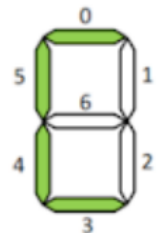
display_and_drop

Pentru afișarea mesajului corespunzător și determinarea condiției de aruncare a pachetului, am declarat 5 variabile de **tip reg** pentru a le putea face **atribuiri blocante** în interiorul **blocului always**.

Folosesc 3 construcții **if** pentru cele 3 condiții – fiecare condiție în parte va afișa un mesaj diferit și îi va atribui o valoare semnalului de alarmă (0 sau 1).

Mesajul va fi afișat pe un afișaj cu 7 segmente, așa că am scris cu ajutorul biților și a figurei alăturate, fiecare literă din cadrul mesajului (1 logic – aprins și 0 logic – stins).

În acest modul am făcut 5 atribuiri continue pentru cele 5 output-uri.



square_root

Pentru implementarea acestui modul m-am folosit de metoda CORDIC.

Înmulțirea cu 2 este echivalentă cu shiftarea cu 1 bit la stânga, iar împărțirea cu 2 este echivalentă cu shiftarea cu 1 bit la dreapta, însă în algoritmul CORDIC înmulțirea și împărțirea pot fi reduse la adunări și scăderi.

Am declarat 3 variabile de **tip reg** pentru a le putea face **atribuiri blocante** în interiorul **blocului always**. În cadrul acestui bloc am implementat metoda CORDIC astfel: i se atribuie variabilei de tip reg in_aux valoarea primită de la input shiftată cu 16 la stânga. Dacă un număr nu este un număr pătrat perfect, adăugarea de biți crește precizia rezultatului rădăcinii pătrate. Variabilei base i se atribuie valoarea 1 shiftată la stânga cu 15. Aceasta va fi shiftată la dreapta cu câte un bit în cadrul construcției **repeat** care va fi executată de 16 ori. Tot în cadrul acestei construcții, variabilei result i se adună valoarea base, și se verifică dacă pătratul acesteia este mai mare decât valoarea inputului shiftată la stânga cu 16. Dacă condiția este adevărată înseamnă că baza nu ar fi trebuit adăugată, așa că se va scădea. Rezultatul final va fi reprezentat astfel: 8 biți pentru partea întreagă și 8 biți pentru partea fracționară.

În acest modul am făcut o atribuire continuă (**assign**) pentru output.

baggage_drop

În cadrul acestui modul am apelat cele 3 module prezentate anterior și am făcut o atribuire continuă pentru a ține cont de formula $t = \sqrt{\text{height}} / 2$ din cadrul cerinței.