

TEMA 1

DESCRIEREA SOLUȚIEI GENERALE

Pentru rezolvarea temei, am ales soluțiile pe care le-am considerat a fi cele mai intuitive și simple de urmat. Am folosit modelul din anexă pentru a înțelege mai bine cum trebuie construite modulele și care trebuie să fie gândirea și scopul din spatele rezolvării temei. Metoda CORDIC implementată pentru modulul square_root a fost relativ ușor de înțeles, și am utilizat site-ul oferit în anexă pentru a înțelege cum trebuie implementat algoritmul.

În continuare, voi descrie implementarea fiecărui modul în parte.

MODULUL SENSORS_INPUT

Am implementat modulul sensors_input prin folosirea a trei "if"-uri în interiorul unui always, în următorul fel:

- Primul "if" este folosit pentru a verifica situația în care unul dintre senzorii 1 sau 3 sunt 0, deoarece, în acest caz, întreaga pereche de senzori ar fi anulată complet. În acest caz, înălțimea ar fi egală cu media aritmetică a celorlaltor doi senzori (2 și 4).

```
testing: S1 = 150, S2 = 0, S3 = 20, S4 = 3;  
testing: S1 = 0, S2 = 34, S3 = 0, S4 = 174;
```

În acest exemplu (cazul 1), cel puțin una dintre S1 și S3 este 0 și, deci, înălțimea = $(34+174)/2 = 208/2 = 104$ (număr întreg).

```
testing: S1 = 0, S2 = 160, S3 = 0, S4 = 43,
```

În acest exemplu (cazul 2), cel puțin una dintre S1 și S3 este 0. Înălțimea = $(160+43)/2 = 203/2 = 101,5$ (număr rațional). Cum am observat din anexa de pe site-ul temei, 101,5 trebuie să se rotunjească la următoarea zecimală, la 102. Așadar, am adăugat un +1 la suma dintre cei doi senzori înainte de a-i împărți, deoarece am observat că în cazul al 2-lea, adăugarea unui +1 ajută la obținerea răspunsului corect, rotunjit, iar în primul caz, răspunsul corect nu este afectat.

Pentru cazul 1: $(34+174+1)/2 = 209/2 = 104,5$ (care ar fi aproximat de Xilinx tot la 104, care este răspunsul dorit de noi).

Pentru cazul 2: $(160+43+1)/2 = 204/2 = 102$ (răspunsul corect, dorit de noi).

- Al doilea "if" este similar cu primul, dar pentru perechea de senzori S2 și S4
- Al treilea "if" cuprinde cazul în care niciun senzor nu este 0 și am procedat similar ca în cazurile anterioare, dar am adăugat +2, care nu afectează în mod negativ rezultatul final, dar ajută la rotunjirea corectă a numerelor finale.

MODULUL DISPLAY_AND_DROP

Am implementat modulul display_and_drop tot prin folosirea a 3 "if"-uri în interiorul unui always. Am inițializat toate output-urile cu 0 și am analizat cele 3 cazuri, folosindu-mă de anexa din laboratorul 3 pentru a putea afișa mesajele COLD, HOT și DROP pe display-ul nostru.

- Cazul 1: $t_act < t_lim$, și $drop_en = 0 \Rightarrow$ mesajul "COLD" va fi cel afișat
- Cazul 2: $t_act < t_lim$, și $drop_en = 1 \Rightarrow$ mesajul "DROP" va fi cel afișat
- Cazul 3: $t_act > t_lim$, și $drop_en = 1 \Rightarrow$ mesajul "HOT" va fi cel afișat

MODULUL SQUARE_ROOT

Acest modul l-am implementat folosind metoda CORDIC, cu ajutorul site-ului din anexa temei. În interiorul always-ului, am inițializat out-ul cu 0. Input-ul este reprezentat pe 8 biți și, astfel, m-am folosit de o bază cu valoarea 128 ($= 2^7$), deoarece sunt 8 biți de modificat, de la 0 la 7. În interiorul unei instrucțiuni "for" (care este sintetizabilă, pentru că are un număr finit de pași), output-ului i se va aduna baza. Se verifică condiția dacă pătratul lui out este mai mare decât input-ul și, în caz afirmativ, din output se scade baza, care nu ar fi trebuit să fie adăugată. În caz negativ, baza rămâne adăugată la output-ul nostru. După verificarea acestei condiții, înainte de închiderea instrucțiunii "for", este nevoie ca baza să fie shiftată la dreapta, adică să fie împărțită la 2 la fiecare iterație. Până în acest punct, se trec testele pentru toate pătratele perfecte, deoarece partea întreagă este corectă, dar lipsește partea fracționară.

Pentru a rezolva și problema cu partea fracționară, am creat două variabile auxiliare: un produs pe 32 biți, în care vom stoca pătratul output-ului (de 16 biți), și o variabilă pe 16 biți, unde vom stoca input-ul dat (care era pe 8 biți), shiftat la stânga. Se realizează același procedeu ca mai devreme și, suplimentar, variabila auxiliară care stochează pătratul este shiftată la dreapta cu 8, pentru ca verificarea condiției să se realizeze corect.

Sima Maria-Cristina

Grupa 331AB

MODULUL BAGGAGE_DROP

În acest modul am realizat apelul celorlaltor trei module (`sensors_input`, `square_root` și `display_and_drop`) în ordinea corectă. De asemenea, în acest modul am adăugat wire-urile necesare pentru realizarea conexiunilor dintre module și am inițializat `t_act` cu valoarea $out/2$, ca în cerința temei (`out`-ul fiind aici rădăcina pătrată a înălțimii).