

Facultatea de Automatică și Calculatoare

Specializarea: Calculatoare și Tehnologia Informației

Disciplina: Proiectarea Sistemelor Numerice

**Proiect:**

**Intersecție semaforizată în cruce cu senzori**

Îndrumător Laborator: Realizator:

Diana Irena Pop Apostu Iulian-Eduard

Cuprins

**1 Proiectare5**

1.1 Specificația Proiectului 5

1.2 Schema Bloc/Componente

**2. Justificarea soluției alese10**

**3. Manual de utilizare și întreținere** **12**

**4. Posibilități de dezvoltări ulterioare** **16**

**5. Bibliografie** **17**

1. Proiectare

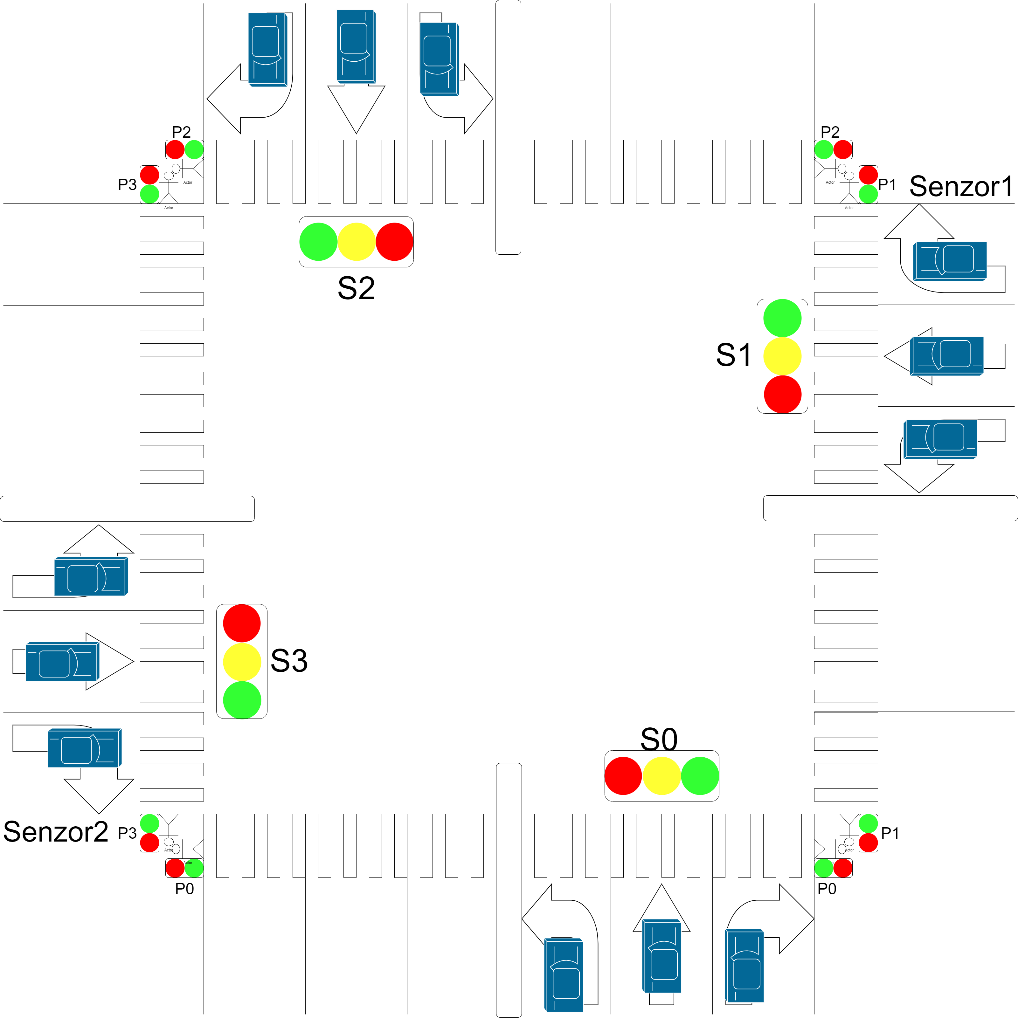
1.1. Specificația Proiectului

Se cere să se proiecteze o intersecție semaforizată în cruce cu senzori.

Fiecare direcție de mers are 6 benzi de circulație. Pi reprezintă semafoare pentru treceri de pietoni, cu 2 culori (roșu și verde), iar Si reprezintă sensurile pentru automobile, cu semafoare cu 3 culori (roșu, galben și verde). Culoarea galbenă a semaforului pentru automobile este corelată cu culoarea roșie la semafoarele de pietoni.

Sistemul are două regimuri diferite de funcționare: regimul test și regimul normal. În regimul test se introduc valori predefinite si cu ajutorul lor se verifica funcționarea semafoarelor și a senzorilor.

Intersecția este prevăzută cu senzori de detectare a automobilelor pe direcția

orizontală (considerată cu prioritate), în ambele sensuri. Aceștia sunt fixați cu 30 m înainte de intrarea in intersecție, pe fiecare banda. Daca senzorii detectează un flux de mașini cu o anumita frecventa pe direcția orizontala (aceasta frecventa este programabila), atunci se va modifica structura de comanda încât sa se repete timpul de funcționare a sistemului pe acea direcție si abia apoi sa se reia ciclul normal de funcționare. 

1.2 Schema Bloc

Diagram

Description automatically generated

Componente:

-Divizor de frecvență: primește ca intrare clock-ul principal.

-Memorie ROM.

-2 porți logice AND: pentru senzori.

-2 porți logice OR: pentru a decide dacă se va repeta ciclul în funcție de senzor.

ROM:

Memoria ROM are ca intrare Clock-ul divizat care va “parcurge” memoria ce are 4 biți de adresă (lungimea adresei de biți) și lungime de date 14:

A picture containing text, clock

Description automatically generated

Table

Description automatically generated with low confidence

Conținutul memoriei ROM este alcătuit din 14 numere scrise în baza 16 ce reprezintă aprinderea și stingerea semafoarelor, numărul respectiv în baza 2 reprezentând fiecare culoare, stinsă pe 0 și aprinsă pe 1, numere parcurse de către Clock-ul divizat:

Conținutul memoriei ROM:



De exemplu, pentru prima valoare: 94C în binar este 00100101001100:

Cel mai semnificativ bit reprezintă starea culorii verde a ultimului semafor (0 adică stins).

Cel mai puțin semnificativ bit reprezintă starea culorii roșie a primului semafor (0 adică stins).

Primii 3 biți (cei din stânga) reprezintă ultimul semafor, “001b” indicând faptul că semaforul este roșu.

Următorii 3 biți reprezintă al treilea semafor, “001” indicând faptul că semaforul este roșu.

Următorii 2 biți reprezintă semaforul pentru pietoni(avem un semafor universal pentru toate trecerile de pietoni, toți pietonii trecând simultan), “01” indicând faptul ca toate semafoarele pentru pietoni sunt roșii.

Următorii 3 biți reprezintă cel de-al doilea semafor, “001” indicând faptul ca semaforul este roșu, iar ultimii 3 biți (cei din dreapta) reprezintă primul semafor, “100” indica faptul ca primul semafor este verde.

Putem observa că pentru culoarea galben la cel de-al doilea (ciclul 951h) și la ultimul semafor (ciclul 1149h) se repetă de 3 ori adică 3 cicluri. Acest lucru este datorat celor doi senzori care verifică fluxul de mașini în timp ce culoarea galben este aprinsă.

Același lucru poate fi observat și la cel pentru pietoni in ciclul 0989h , dar nu pentru a verifica senzorii, ci pentru a umple memoria ROM pentru a nu exista cicluri goale care sa oprească toate semafoarele

Senzorii:

Cei doi senzori sunt alcătuiți fiecare dintr-o poarta logică AND cu doua intrări. O intrare este Clock-ul cu ajutorul căruia se poate schimba frecvența, iar cea de a doua este starea culorii galben pentru cel de al doilea semafor, respectiv pentru ultimul semafor destinat mașinilor.

Atunci când culoarea unuia dintre cele două semafoare devine galben, dacă senzorul corespunzător benzii este aprins (indică un număr mare de mașini), atunci ciclul se va repeta,

Însă culoarea galben va rămâne aprinsă (un lucru ce poate fi modificat). Ciclul respectiv se poate repeta o singură dată.

Ieșirea din poarta AND (din senzori) va intra într-o poarta OR cu două intrări, cealaltă intrare fiind starea culorii verde, acest lucru făcând posibilă aprinderea culorii verde pentru a doua oară.

Fără această poartă OR, reaprinderea culorii verde a celor două semafoare cu senzori nu poate fi posibilă datorită faptului că memoria ROM nu are un ciclu special pentru cazul în care senzorul indică un flux mare de mașini, iar cele doua semnale(cel din ROM și cel din poarta AND) vor interfera, repetarea ciclului nefiind posibilă.

Exemplu pentru cazul în care senzorul indică flux ridicat de mașini:

A picture containing calendar

Description automatically generated A picture containing diagram

Description automatically generated A picture containing text, clipart

Description automatically generated A picture containing diagram

Description automatically generated

La prima vedere, o metodă pentru a nu adăuga cele două porți OR ar fi oprirea memoriei ROM pentru a reaprinde culoarea verde, conectând ieșirea din poarta AND direct la led și la memoria ROM, însă acest lucru nu este posibil: Diagram

Description automatically generated

2. Justificarea soluției alese

Pentru implementare am ales să folosesc memorie ROM în detrimentul utilizării a mai multor numărătoare pentru fiecare culoare și semafor datorită faptului că în cerință nu au fost impuse restricții și preferințe legate de timp, nefiind precizată o anumită durată de timp pentru fiecare culoare a semaforului.

Cât despre “parcurgerea” intersecției, am ales o metodă eficientă pentru derularea ciclurilor

astfel încât timpul de așteptare pentru fiecare direcție de mers să fie relativ mic. Parcurgerea este alcătuită din 6 cicluri ce vor avea loc în următoarea ordine:

-în primul ciclu, vor trece mașinile de pe banda S0

-în al doilea ciclu, vor trece mașinile de pe banda S1

-în al treilea ciclu, vor trece pietonii

-în al patrulea ciclu, vor trece mașinile de pe banda S2

-în al cincilea ciclu, vor trece mașinile de pe banda S3

-în al șaselea ciclu, vor trece pietonii

Această metodă de parcurgere poate fi regăsită in organigrama în care cele 6 cicluri au fost realizate cu ajutorul a 9 stări ce includ și verificarea senzorilor:

Application, Teams

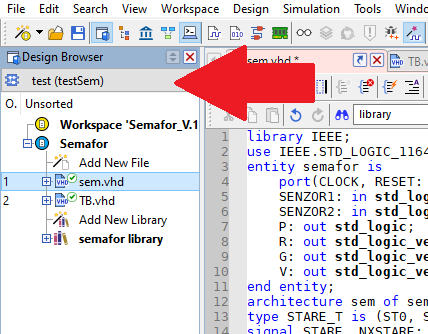
Description automatically generated

3. Manual de utilizare și întreținere

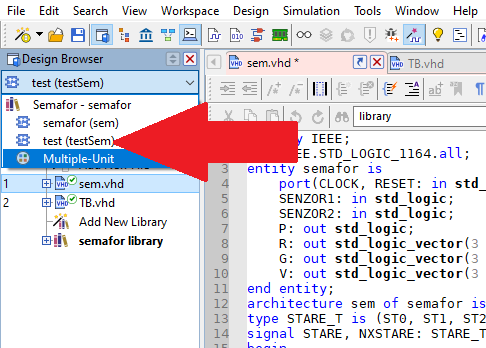
Se va rula fișierul Semafor.aws ce va duce la deschiderea întregului proiect în Active-HDL.

Acesta conține 2 fișiere de tip .vhd. Primul este fișier: sem.vhd reprezintă regimul normal de funcționare al semaforului, iar cel de-al doilea, TB.vhd reprezintă regimul de test (Test Bench) al acestuia. Pentru a verifica funcționalitatea, vom compila pe rând cele două fișiere cu ajutorul tastei F11, urmând ca mai apoi sa pornim simularea folosind tasta F6. Înainte de a porni simularea, trebuie să ne asigurăm ca simulăm fișierul TB.vhd. Acest lucru se realizează astfel:

Se va da click aici:



Se va alege entitatea test din lista:



După pornirea simulării, se va deschide un fișier de tip waveform astfel:

Graphical user interface, text, application

Description automatically generated

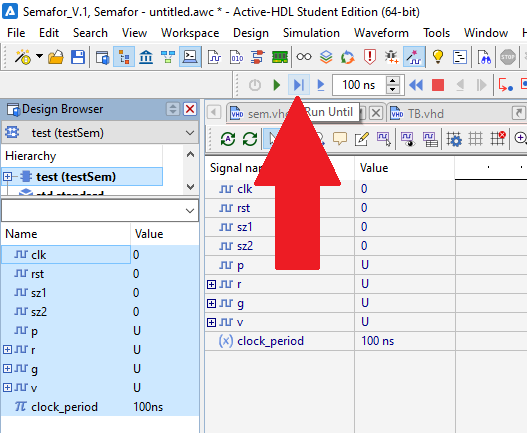
Unde se vor selecta cu ctrl-a toate semnalele din partea stângă, se vor copia cu ctrl-c și vor fi puse aici cu ctrl-v:

Graphical user interface, application, Word

Description automatically generated Graphical user interface, application, table, Excel

Description automatically generated

Se va apăsa butonul “run until” și se va selecta timpul dorit de simulare:



Waveform-ul va arăta așa:

Graphical user interface, Excel

Description automatically generated

r, g, v sunt cele trei culori pentru fiecare din cele 4 semafoare destinate mașinilor.

p este semaforul pentru toți pietonii.

Sz1, sz2 sunt cei doi senzori.

Schimbarea frecvențelor celor doi senzori se poate realiza in fișierul TB.vhd:

Text

Description automatically generated

Printr-un simplu test se poate constata faptul că pentru valorile mai mari sau egale cu 9, senzorul nu mai afectează parcurgerea, astfel ciclul respectiv nu se va mai repeta.

4. Posibilități de dezvoltări ulterioare

O primă îmbunătățire poate fi găsirea unei parcurgeri mai eficientă decât cea aleasă prezentată mai sus în organigramă.

Cea de a doua îmbunătățire (pentru implementarea cu memorie ROM în LogiSim) ar fi adăugarea altor 2 porți logice NAND pentru culoarea galben la cel de-al doilea și cel de-al patrulea semafor, astfel încât acestea să se stingă în cazul în care senzorul indică un flux mare de mașini ce va duce la repetarea ciclului, nu să rămână aprins cum este acum. Așadar, prima intrare în poarta NAND va fi chiar semnalul ce conține starea culorii galben din memoria ROM, iar cea de a doua intrare va fi ieșirea din poarta OR (care duce la culoarea verde), ieșirea porți NAND fiind led-ul pentru culoarea galben.

Cum arată acum:

A picture containing text, clipart

Description automatically generated

Led-ul galben ar trebui sa fie stins. (Acest lucru se realizează în simularea din Active-HDL)

O altă îmbunătățire care ar putea oferi veridicitate proiectului ar putea fi utilizarea altor resurse precum numărătoare pentru fiecare culoare și semafor

5. Bibliografie

1.

<https://didatec.sharepoint.com/:b:/r/sites/PSNSeriaAan2021-2022/Class%20Materials/Indrumator%20de%20Laborator/Lab10VHDL.pdf?csf=1&web=1&e=ItYhhi>

2.

<https://didatec.sharepoint.com/:f:/r/sites/PSNSeriaAan2021-2022/Class%20Materials/Cursuri?csf=1&web=1&e=R70OLT>

3. <https://www.youtube.com/watch?v=eZ2SmsUi8kI&ab_channel=GanDuumar>

4. <https://www.youtube.com/watch?v=ejpdxbkHgWo&ab_channel=NeutronNick11>