Универзитет у Београду

Електротехнички факултет

Пројекат из основа рачунарске технике 2

Управљање DC мотором и мерење броја обртаја

Студенти:

Лука Симић 2017/0353

Алекса Богдановић 2017/0578

Садржај

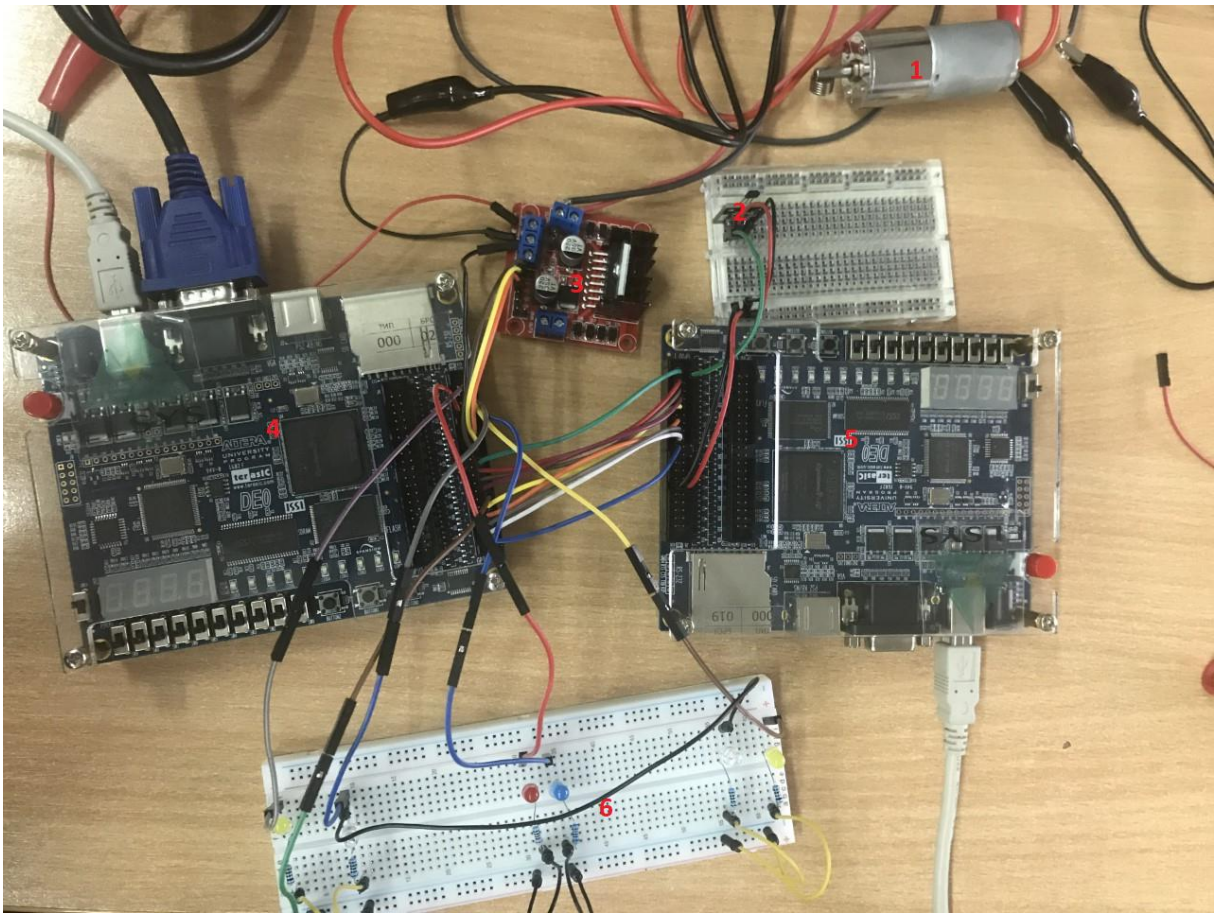
Тематика пројекта

- Управљање DC мотором, односно линеарна контрола брзине и смера окретања DC мотора техником Pulse Width Modulation помоћу коришћења L298N H-most електричног кола.
- Модуларна телеметрија која коришћењем магнетног поља показује основну информацију о мотору – број обртаја.
- Могућност да пројекат буде модуларан, односно да служи као основа за друге пројекте сличног типа и да без икакве промене може да се брзо и лако надограђује, нпр. пројекат може да служи за прављење аутомобила додавањем новог мотора, а на основу броја обртаја може се лако израчунати брзина аутомобила.

Кораци у развоју пројекта

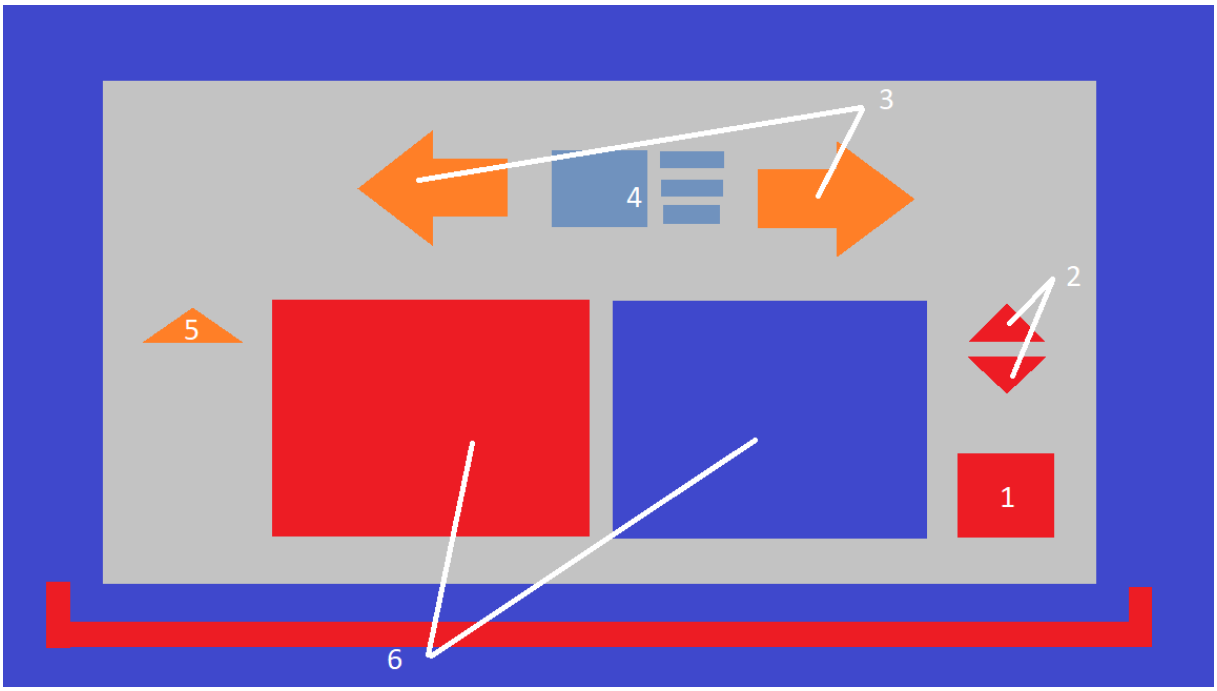
1.) Генерисање PWM сигнала који је главни за покретање и регулацију брзине мотора
2.) VGA протокол – приказивање левог, десног и сва четири показивача правца, информације у ком смеру се окреће мотор, информације о укљученим светлима и информације да ли је мотор укључен. Ово чини средишњи део пројекта, јер смо даље на основу неких сигнала из овог дела пројекта користили за даљу обраду.
3.) Повезивање са L298N колом, атестирање мотора и повезивање сигналних диода – повезали смо нашу прву (main) плочицу са L298N колом, повезали и прикључили напајање +12V DC на L298N и наше диоде које представљају леви, десни, сва четири показивача правца, света и полицијска светла смо такође повезали. Атестирали смо да ли мотор лепо реагује на све наше PWM сигнале, да ли стабилно мења смер, да ли диоде светле како треба и да ли то све лепо интерагује са нашим ВГА протоколом.
4.) Hall Effect A3144 сензор – Повезивање сензора за детекцију магнетног поља и пропратна логика која мери време и рачуна број обртаја.
5.) Комуникација између плочица – прослеђивање израчунате вредности RPM друге плочице (плочице за телеметрију) првој (main) плочици и приказивање те вредности на 7 Segment display.
6.) Комплетно повезивање свих претходних делова, финално атестирање, исправљање уочених грешака приликом атестирања.
7.) Демонстрација финалне верзије пројекта.

Изглед делова и плочице



1) DC мотор 12V
2) Hall Effect сензор A3144
3) L298N
4) Main плочица за VGA, диоде и исписује број обртаја мотора
5) Плочица која рачуна број обртаја мотора
6) Диоде за фарове, ретровизоре и сирене

VGA



Приказ пројекта на монитору

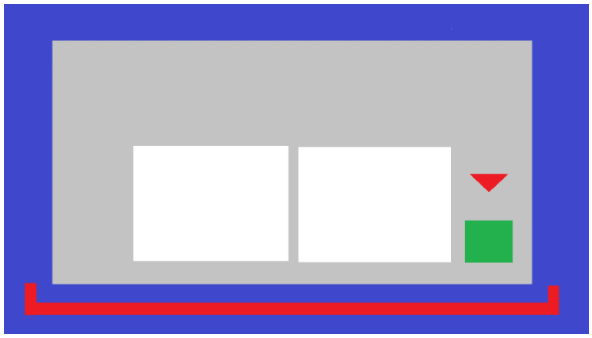
1) Ознака за рад мотора (црвено означава да је мотор угашен, зелено означава да је мотор упаљен)
2) Ознака која показује у ком се смеру окреће мотор (стрелица на горе – CW у смеру кретања казаљке на сату, стрелица на доле – CCW у обрнутом смеру од смера кретања казаљке на сату).
3) Ознака за мигавце
4) Ознака да ли су упаљени фарови
5) Ознака за сва четри показивача правца
6) Ознака за сирене

Портови на плочици

- Sw0 – пали и гаси мотор
- Sw1 – пали и гаси фарове
- Sw2 – десни мигавац
- Sw3 – леви мигавац
- Sw4 – смер кретања мотора
- Sw5 – сва четири показивача правца
- Sw6 – сирене

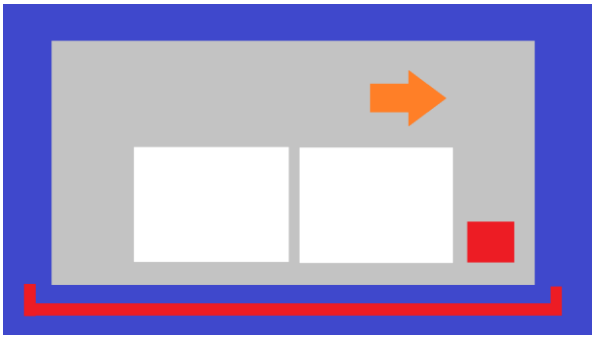
Случајеви

Када упалимо мотор на слици се ознака за мотор претвара у зелени квадрат и уколико sw4 није активан онда је приказана стрелица на горе.

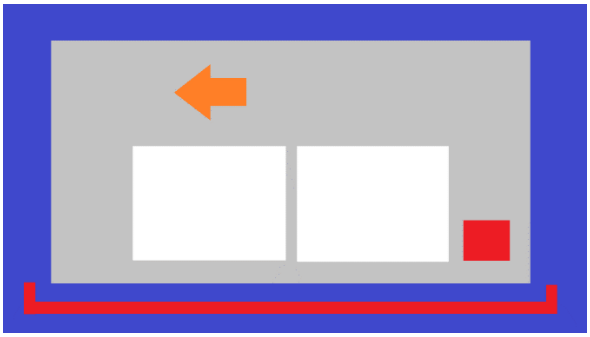


Када упалимо мотор и уколико је sw4 активан онда је приказана стрелица на доле.

Напомена : Уколико је мотор угашен не приказује се ни једна стрелица.

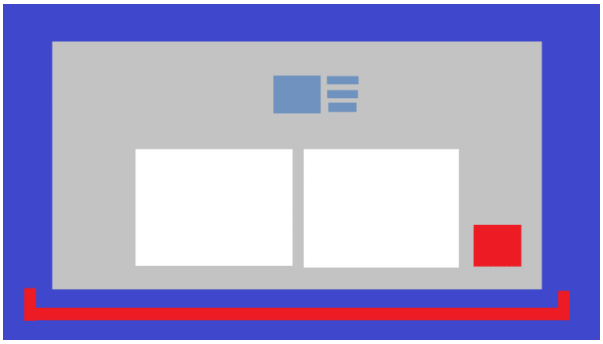


Када је sw2 активан пали се десни мигавац, може да ради без активности мотора.

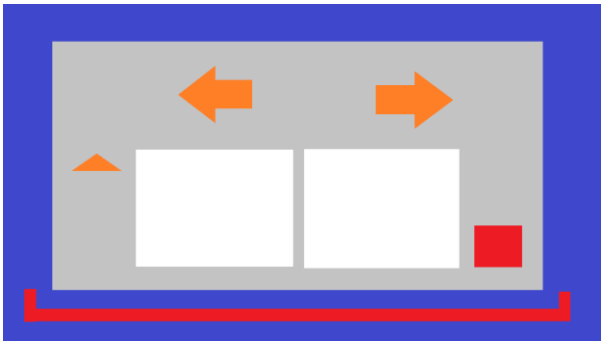


Када је sw3 активам пали се леви мигавац, Може да ради без активности мотора.

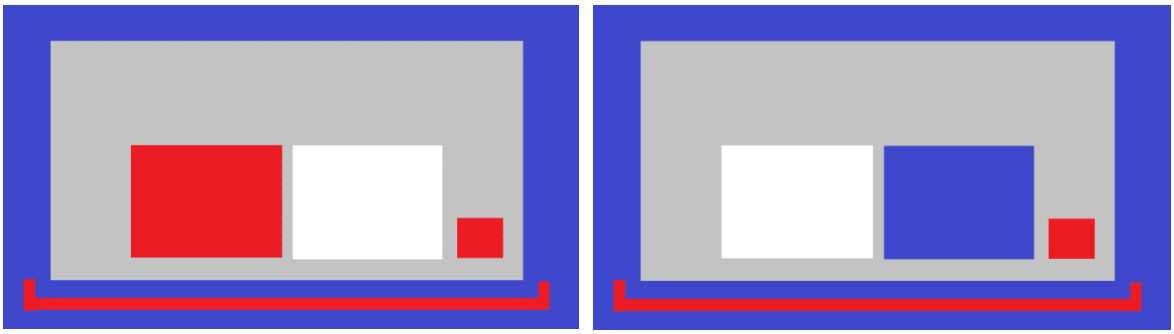
Напомена : Уколико су активни и sw3 и sw4 неће приказати ни један ни други мигавац



Када је sw1 активан приказује се ознака за фарове, може да ради без активности мотора.



Када је sw5 активан приказује се ознака за сва четири показивача правца, може да ради без активности мотора.



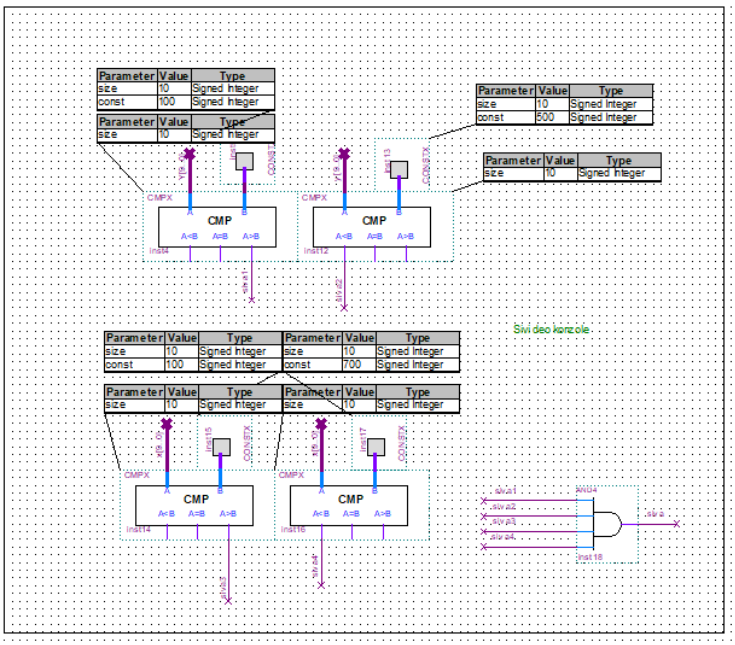
Уколико је активан sw6 пали се сирена која наизменично мења боје из црвене у плаву.

Напомена : Могу у исто време да раде и сирена и сва четири показивача правца и фарови без активности мотора.

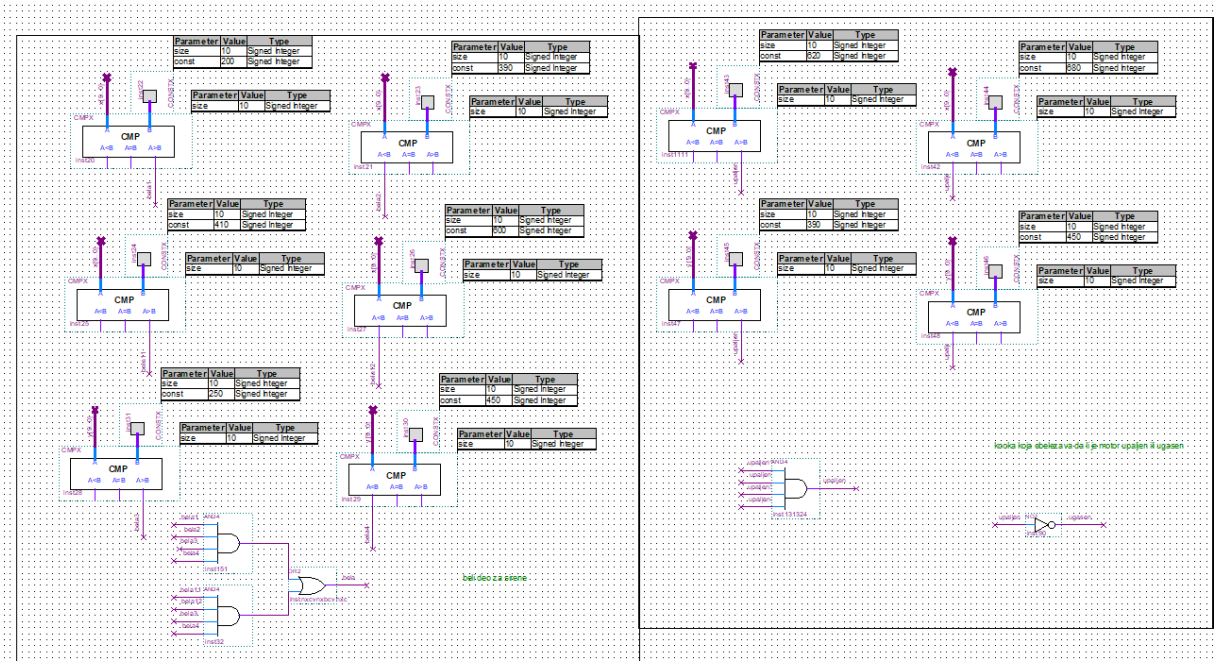
Списак реализованих датотека/шема

- Seven segment digit interface.vhd
- Binary 2BCD.vhd
- add10.bdf
- add20.vhd
- add1.bdf
- REGX.vhd
- MP2X.vhd
- CONSTX.vhd
- RisingEdge.bdf
- CMPX.vhd
- CLK_DIVIDER.vhd
- VGAController.bdf
- Block1.bdf
- mux4-1.bdf
- MUX16-1.bdf

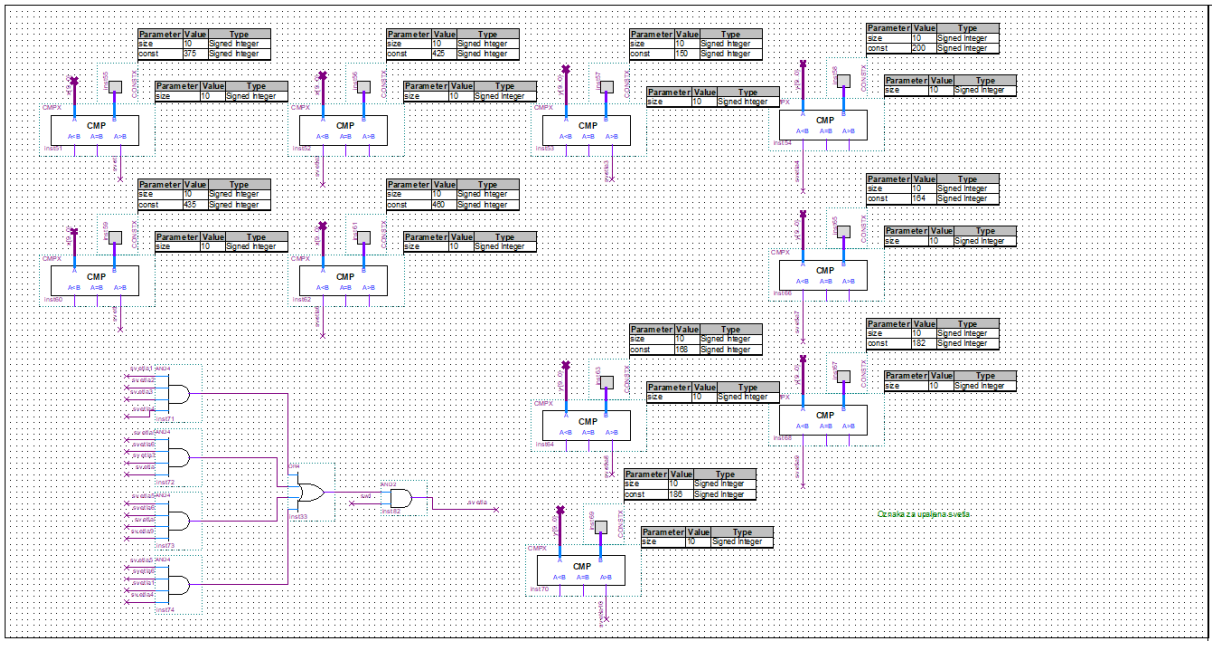
Приказ шеме у quartus-y за VGA-a



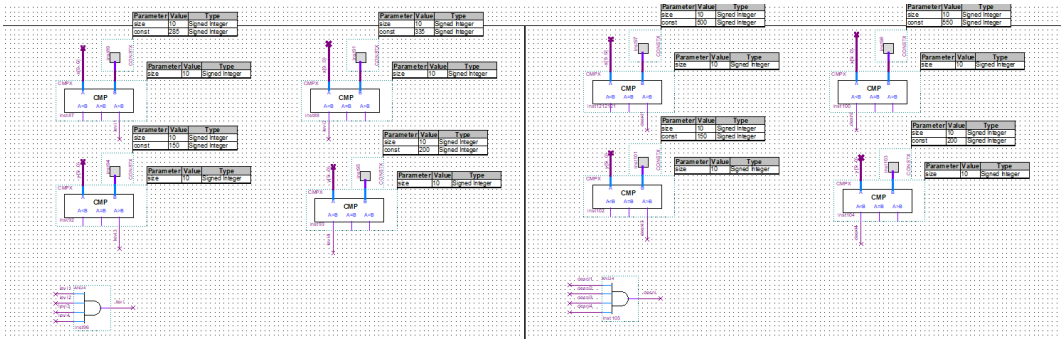
Сиви део конзоле



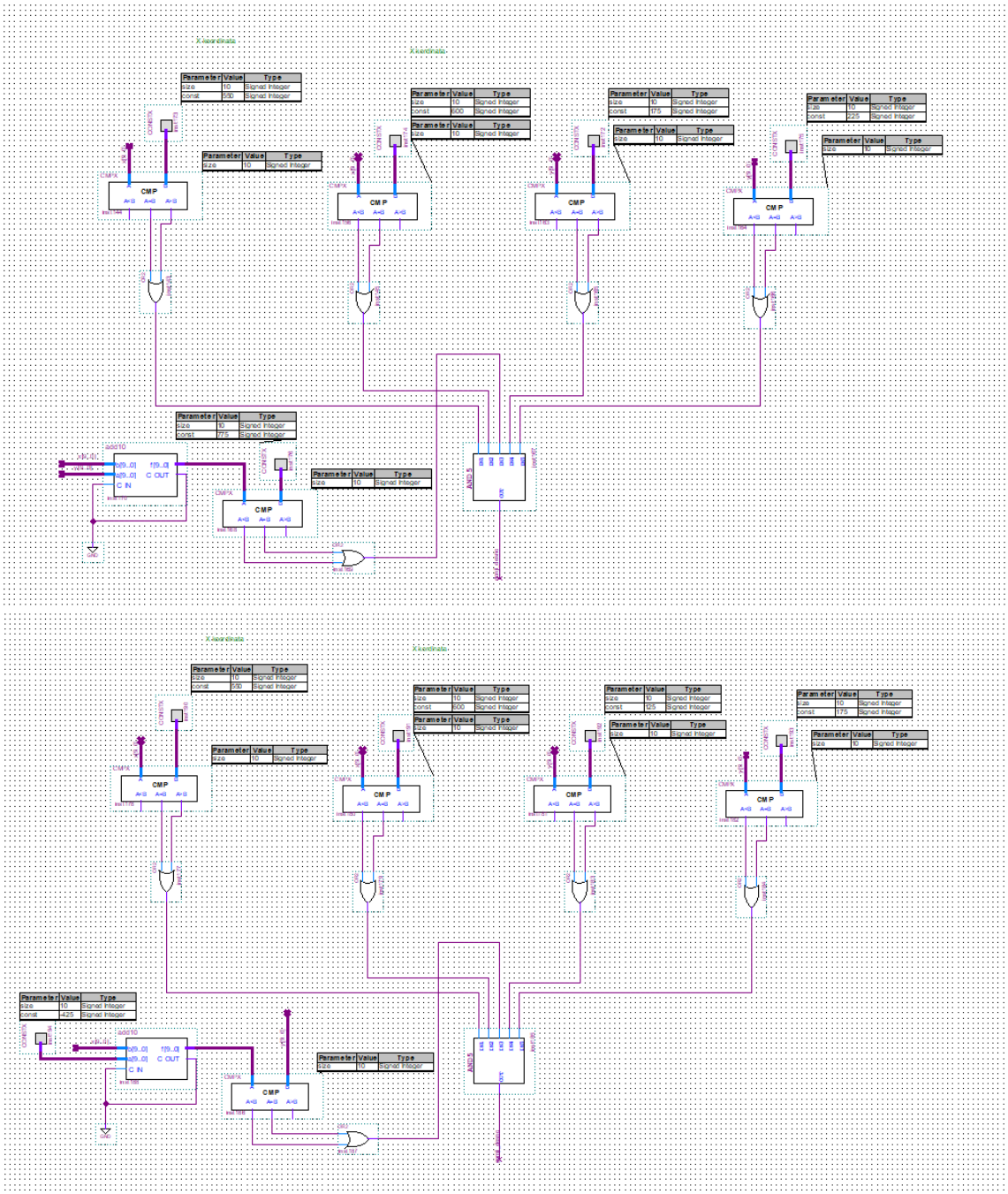
Приказ позадине за сирене и мотор



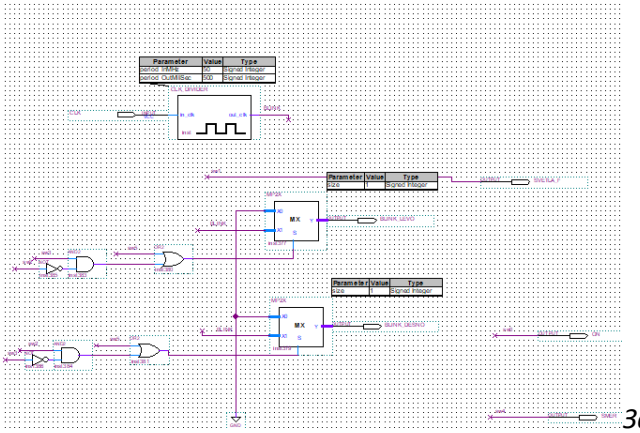
Фарови



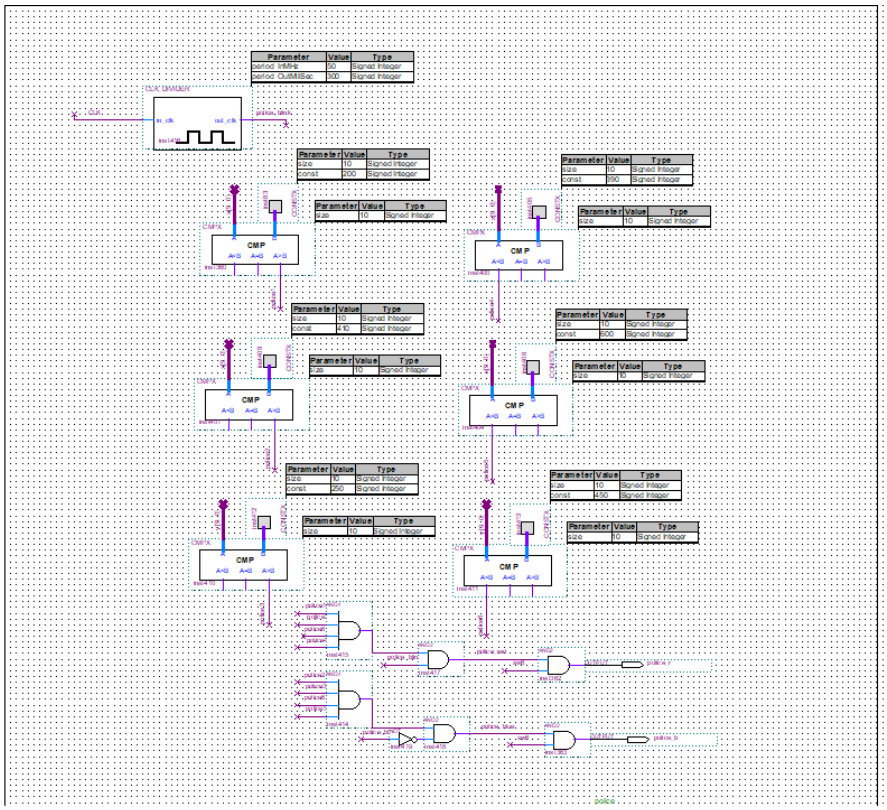
Леви и десни мигавац



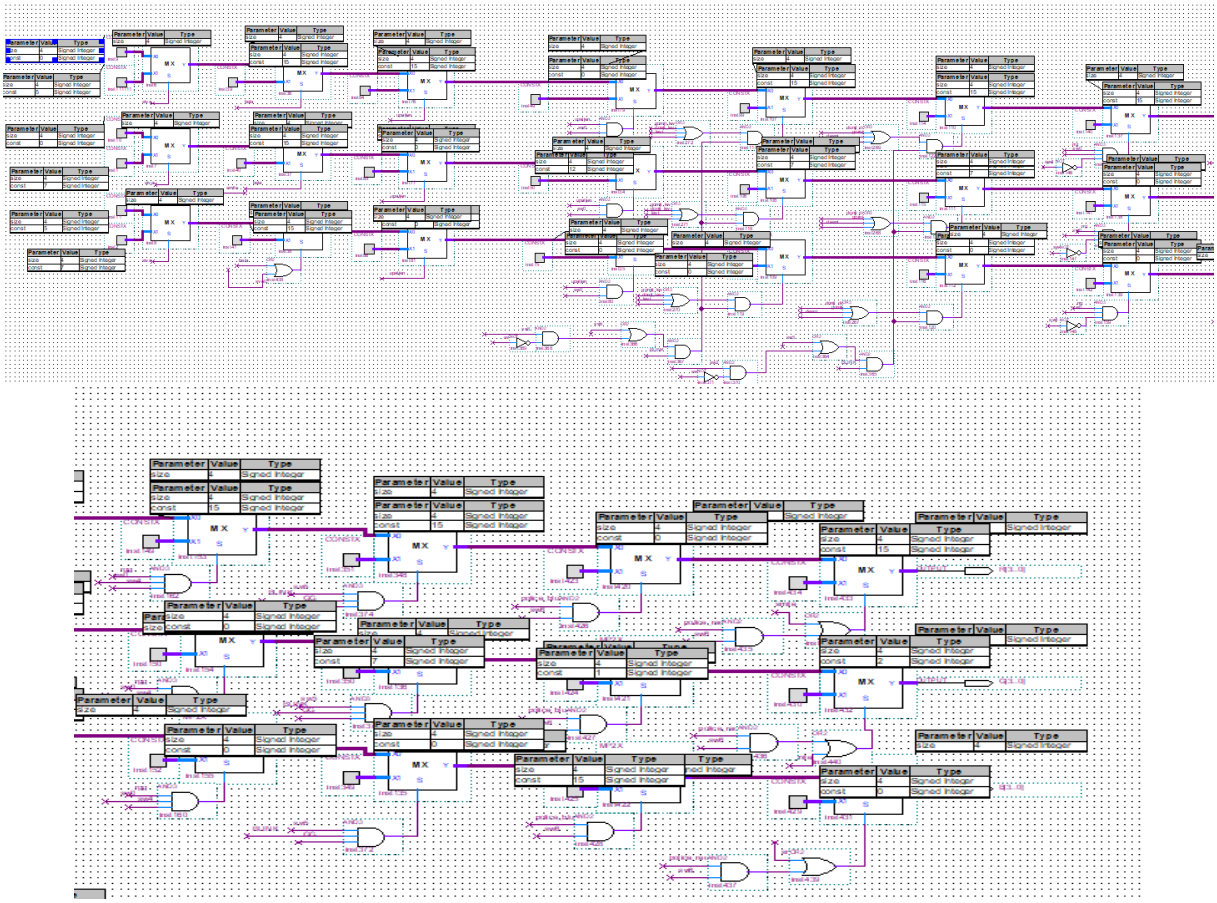
Начин на који је нацртан један троугао, на исти начин су нацртани и остали троуглови



За блинк мигавца



За блинкање сирене



Мултиплексери преко којих шаљемо боје и делове које треба да прикаже на монитору

Диоде

- Када је активан sw1 пале се беле лед диоде које представљају фарове.
- Када је активан sw2 пали се десна наранџаста лед диода која се пали и гаси наизменично (на 0,5с).
- Када је активан sw3 пали се лева наранџаста лед диода која се пали и гаси наизменично(на 0,5с).
- Када је активан sw5 пали се и лева и десна наранџаста лед диода које се пале и гасе наизменично (на 0,5с).
- Када је активан sw6 пале се наизменично плава и црвена лед диода (на 0,3с).

Портови:

- Десна наранџаста лед диода – AB20
- Лева наранџаста лед диода – AA20
- Беле лед диоде – AB19
- Плава лед диода – AA18
- Црвена лед диода – AB18

Управљање мотором

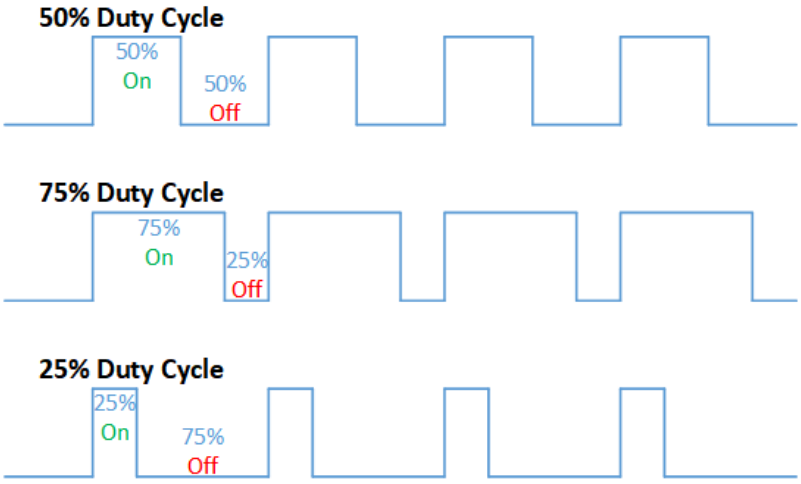
Управљање мотором се врши уз помоћ L298N дуал H-бридге кола.

GND и VCC – на њих се качи напајање између 7-35V за мотор и преко овог кола у ствари протиче струја јер FPGA плочица не може да издржи велике струје и напоне.

IN1 и IN2 – ови улази служе за регулисање смера мотора и разликујемо неколико случајева у зависности њихове комбинације:

1.) IN1=0 I IN2=1 – мотор се окреће CW
2.) IN1=1 I IN2=0 – мотор се окреће CCW
3.) IN1=1 I IN2=1 – ова комбинација се не користи, односно мотор се неће окретати
4.) IN1=0 I IN2=0 – мотор се неће окретати

ENA – улаз на кога доводимо PWM сигнал за регулисање брзине мотора.



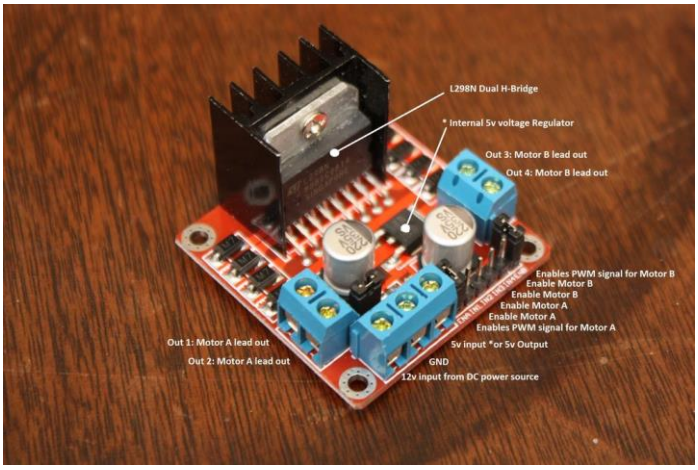
Изглед PWM сигнала

PWM сигнал генерише FPGA плочица. Наш сигнал ће бити фреквенције 200Hz, амплитуде VCC и производиће 10 различитих сигнала чији је Duty Cycle 10%, 20% , 30% , 40%, 50%, 60% , 70%, 80% , 90% и 100%.

Duty Cycle – се рачуна по формули: $D = \frac{PW}{T} \times 100\%$ је:

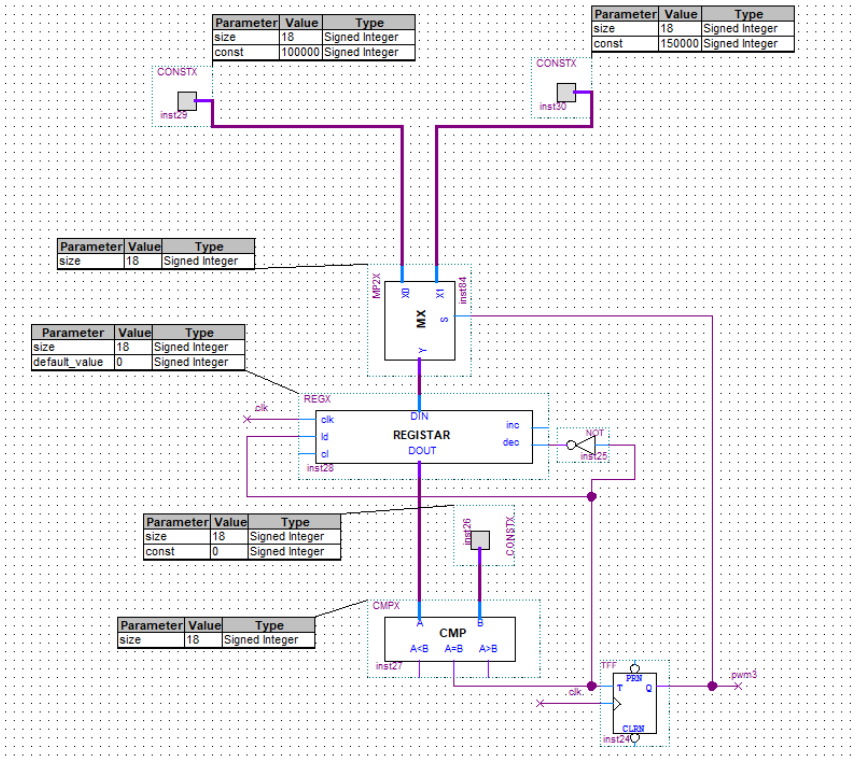
- D – Duty Cycle
- PW – Време у којем наш сигнал има вредност + VCC
- T – периода нашег сигнала

Што је овај параметар већи брзина мотора је већа и обрнуто.



Изглед L298N кола

Коло које генрише PWM сигнал – Blok1.bdf



Када бисмо направили 18bit-ни бројач који сваки пут када изброји 250,000 и тада тоглује флип флоп добили бисмо периодичан правоугаони PWM сигнал који има duty cycle 50% и фреквенцију 200Hz. Ако бисмо хтели да добијемо duty cycle нпр. 40%, урадићемо следеће. У регистар учитамо 100,000 и то декрементирамо, када то дође на нулу Т флип-флоп тоглује вредност на 0 у регистар упишемо 150,000 и сада то декрементирамо на нулу, када дође вредност регистра на нулу, Т флип-флоп тоглује вредност на 1, у регистар се упише 100,000 и процес се понавља наизменично. Тако смо добили целу периоду где нам је сигнал прво имао вредност 1, а затим вредност 0, а како је 150,000+100,000 = 250,000 а 50MHz/250,000 = 200Hz добили смо очекивани PWM сигнал.

Даље ћемо те сигнале прослеђивати на мултиплексер и како кликћемо bnt0 или бtn1 прослеђујемо све спорији или све бржи PWM а лед[9..0] ће нам показивати који duty cycle је тренутно изабран и прослеђен мотору.

Помоћу прекидача SW4 бирамо смер, а информацију о смеру ћемо имати на екрану уз помоћ VGA протокола.

Телеметрија

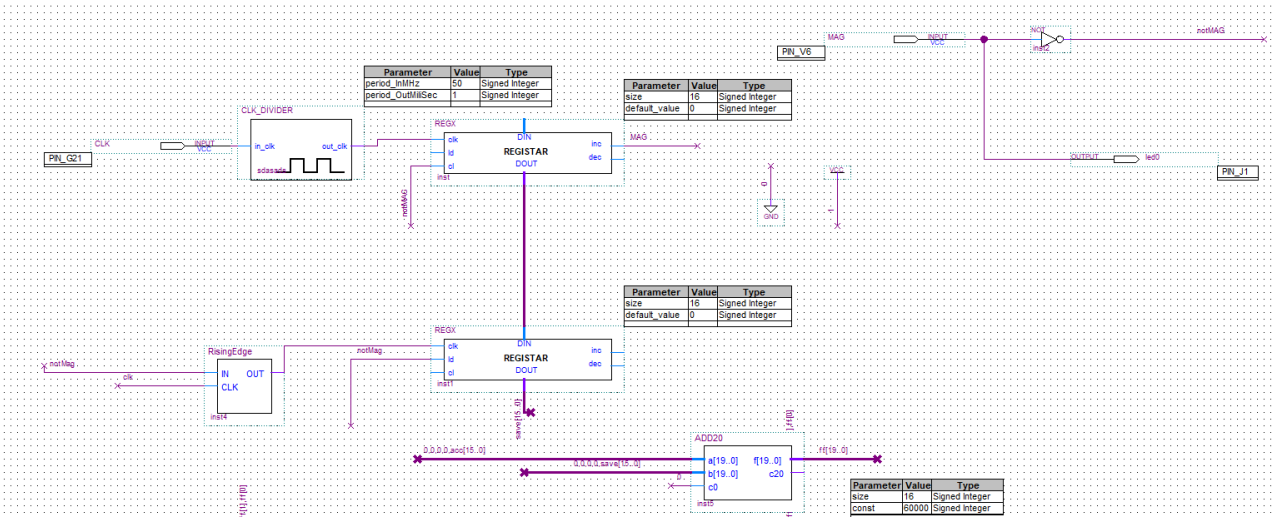
Друга FPGA плочица ће да садржи сензор A3144 (прекидачки hall effect транзистор) који це нам послужити за мерење броја обртаја по минути RPM. Сензор ради на следеци начин: када нема магнетног поља у близини он ће да нам даје логичку јединицу (+2,5V), уколико има магнетног поља у близини он ће нам давати Логичку нулу (+0V).

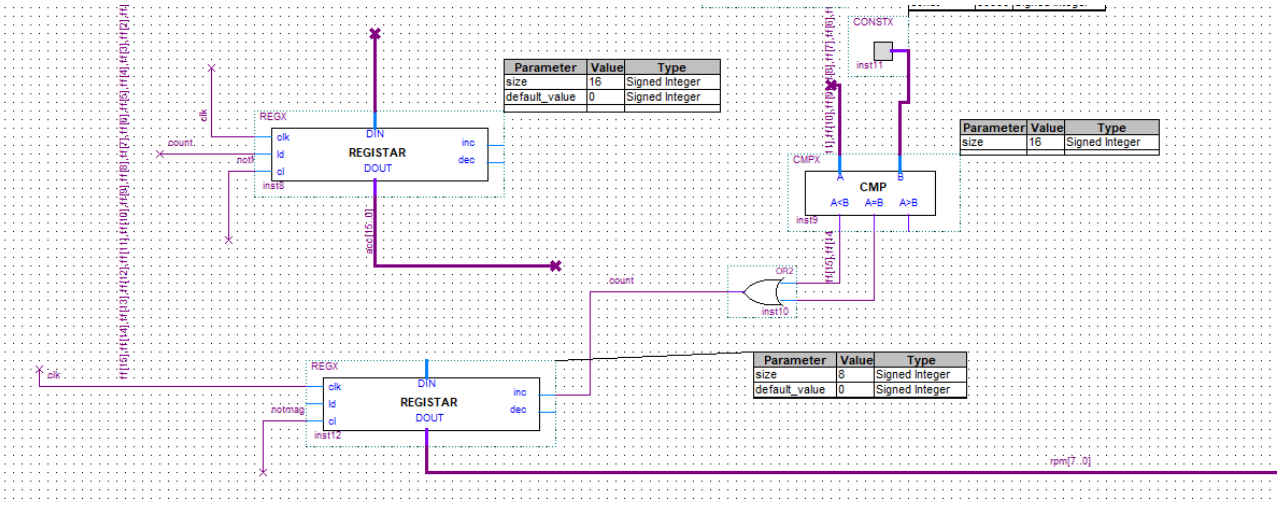
За осовину мотора ће бити прилепљен неодијумски магнет, док се осовина врти и када сензор не детектује магнет, имаћемо један бројач који се инкрементира на сваку милисекунду. Када сензор детектује магнет, сачуваћемо број милисекунди из нашег бројача који нам говори колико нам је требало милисекунди да се изврши отприлике један обртај.

Затим рачунамо по следећем програму колико ћемо имати обртаја у минути:

```
Count=0;

Rpm=0;

While(sum<=60000)

{

      sum+=brojmilisekundi;

      rpm++;

} display(rpm)
```

- Број милисекунди – је време које нам је потребно за један обртај
- RPM – број обртаја по минути
- сум – привремена сума, односно овим ћемо бројати колико се измерених милисекунди налази у 60000 милисекунди тј. једном минуту и упоредо сваки пут када испунимо услов у while петљи инкрементирамо RPM. Када изађемо из while петље као резултат добијамо број обртаја, који касније прослеђујемо првој (main) плочици и приказујемо на 7-segment display.





Када сензор не детектује магнет, тада ће се активирати Send управљачки сигнал који прима прва(main) FPGA плочица, она тада у свој регистар учита 8bit-ну вредност која представља RPM и ту вредност прикаже на 7-segment display. Тих 8 bit-ова смо проследили преко output пинова друге плочице на input пинове прве плочице (паралелни пренос).

Мане пројекта

Мана пројекта је начин на који је измерен број обртаја. Пошто смо мерили време у милисекундама, за већи број обртаја нпр. 1000rpm тренутно решење не би могло да детектује толико велике обртаје, већ би морао бројачки регистар да броји нпр. у наносекундама, а самим тим би нам и требао регистар већег капацитета. Још једна мана је то што смо рачунали да се магнет налази испред сензора у идеалном кратком времену, што у реалности није случај па долази до мале грешке приликом рачунања.