

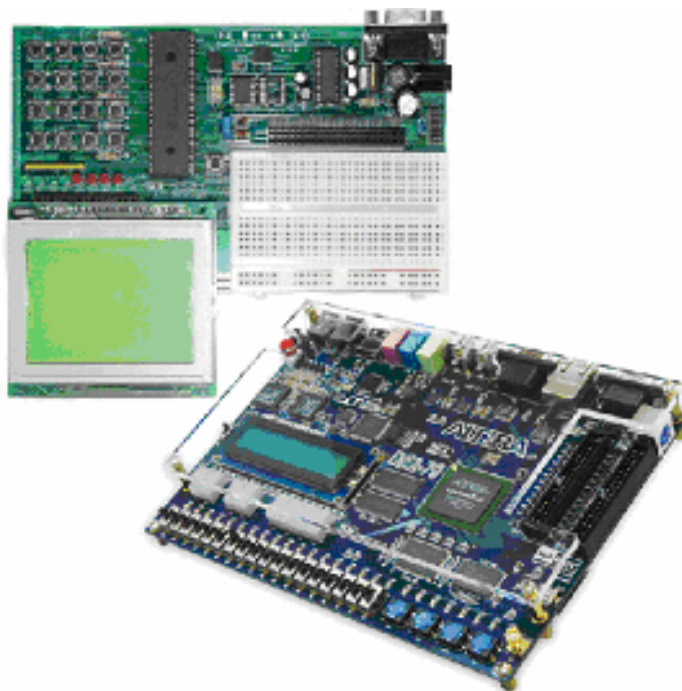


NVD
Rit

Digitalur og analogur elektronikkur

Starvsstovuvenjingar

Magnus Danielsen
Benadikt Joensen
Ólavur Simonsen



VERITGERN
Thesis

TØKNIFRÁGREIÐING
Technical Report

UNDIRVÍSINGARTILFAR
Teaching Material

UPPRIT
Notes

NVDRit 2009:07

Heiti / Title **Digitalur og analogur elektronikkur –
Starvsstovuvænjingar**

Høvundar / Authors Magnus Danielsen
Benadikt Joensen
Ólavur Simonsen

Ritslag / Report Type *Undirvísingartilfar/Teaching Material*
Forsíða/Frontpage DE2-70 development board, Altera and Terassic
Proton Development Board, Crownhill Ltd.

NVDRit 2009:07
© Náttúruvísindadeildin og høvundurin
ISSN 1601-9741
Útgevvari / Publisher Náttúruvísindadeildin, Fróðskaparsetur Føroya
Bústaður / Address Nóatún 3, FO 100 Tórshavn, Føroyar (Faroe Islands)
Postrúm / P.O. box 2109, FO 165 Argir, Føroyar (Faroe Islands)
• • • • • +298 352550 • +298 352551 • nvd@setur.fo

INNIHALDSYVIRLIT

I.	Digital rás til stýring av trafikkljósi	s. 5
II.	Switch-mode regulering til DC motor	s. 13
III.	Kombinatoriskar rásir	s. 17
IV.	CMI – Koding og deking til ljósleiðara dataskipan	s. 23
V.	PIC mikrokontrollari – menning við Proton Development Board	s. 31

Føroyskt abstrakt

Henda frágreiðing umfatar leiðbeiningar til starvsstovuvønjingar í lærugreinunum ”Elektronikkur” og ”Digitalur elektronikkur og mikroteldur”, ið innganga í útbúgvingina BSc í KT Verkfrøði á Náttúruvísindadeildini á Fróðskaparsetri Føroya.

English abstract:

The present report includes instructions for laboratory exercises for courses in ”Electronics” and ”Digital electronics and microprocessors” intended to be part of the education in ICT Engineering at University of The Faroe Islands.

I. Digital rás til stýring av trafikkljósi

Magnus Danielsen

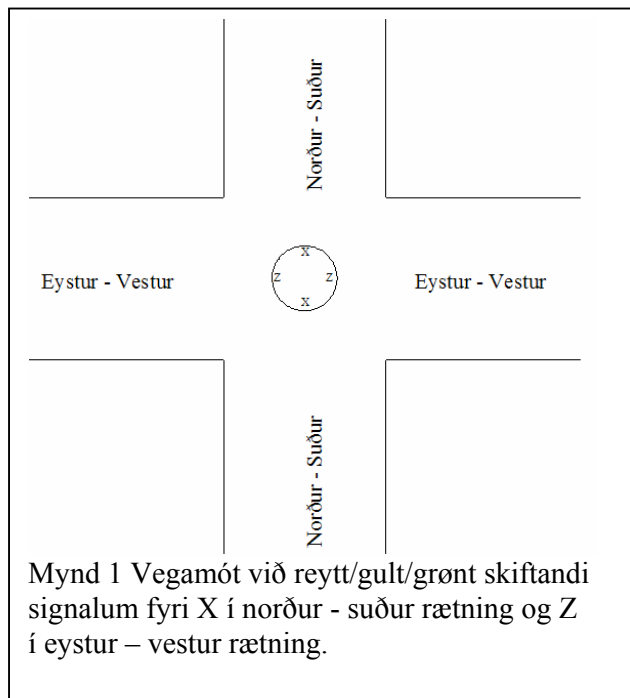
Inngangur

Vanligt er at nýta ljóssignal til stýring av trafikkinum á vegamótum. Vit kanna í hesi venjingini bert eitt einfalt dømi, har ljóssignal er sett upp at stýra biltrafikki í einum vegamóti, sum víst í mynd 1.

Raðfylgjan kann vera við trimum ymiskum støðum: 1) grønt ljós fyri “koyr”, 2) gult ljós fyri “fyrirreika teg at steðga”, 3) reytt ljós fyri “steðga”, og síðan endurtikið 1) grønt ljós “koyr” o.s.v.

Í summum skipanum er kortini raðfylgjan tillagað soleiðis, at raðfylgjan er í fyra ymiskum støðum: 1) grønt ljós fyri “koyr”, 2) gult ljós fyri “fyrirreika teg at steðga”, 3) reytt ljós fyri “steðga”, 4) reytt og gult ljós fyri “fyrirreika teg at koyra”, og so endurtikið 1) grønt ljós “koyr” o.s.v.

Vanligt er, at fyri gongufólk verður bert nýtt skift millum reytt og grønt ljós, men verður ikki viðgjørt nærmari her.



Í venjingini verður ein stýriskipan roynd. Fyri at stýra raðfylgjuni av ljósunum verður digital streymrás uppbygd. Fleiri mátar eru at náa sama máli. Í hesi venjingini er valt at uppbyggja eina tíðarraðfylgjurás (timer IC) til klokkusignalgerða. Ein teljari verður nýttur til at avgera raðfylgjuna í ljósregulering-ini. Dekodarar verður nýttur til at peika á tey einstøku tíðarbilini í signalraðfylgjuni. Umframt eru kombinatoriskar rásir og flipp-floppar nýttur í venjingini. Sum ljóssignalgerðar verða nýttar reyðar, gular og grønar ljósdiodur. Fyri at spara um talið á IC rásum, verða dekodarar og móguliga aðrar

rásir við “open collector” útgangi nýttir. Hendan venjingin tekur støði í *skjali 1*, funnið á internetinum. (Viðmerkjast skal, at í hesum skjali er feilur í diodurásini í mynd á síðu 6 og 7. Tí verður rásin broytt nakað í hesi venjingini.) Vert er at hefta seg við, at fleiri móguleikar eru fyri at uppbyggja rás, ið hevur sama úrslit. Í hesi venjingini er dentur lagdur á, at studentarnir læra um nakrar IC (Integrated Circuit) streymrásir, ið eru á marknaðinum, til at byggja funktiónina upp sum eina hardware

loysn. Verður harafturímóti ein mikróprosessari brúktur, kann sama funktiún verða skapt sum ein blanda software/hardware loysn.

Uppsetingarnar í hesi venjingini skulu roynast bæði 1) við telduforritinum PSpice og 2) í eini uppseting av elektronisku lutunum í starvsstovuni. Stutt leiðbeining til PSpice er í kapitlið 10 í skjali 2.

Tað er týðningarmikið at fyrireika seg væl og, at **allir teir teoretisku spurningarnir eru svaraðir, áðrenn møtt verður í starvsstovuni**. Frágreiðing um venjingina skal verða skrivað og skjalprógvað við PSpice rástekningum (diagrammum) og útskriftum av simuleringum, tekningum av royndaruppsetingum og mátiúrslitum. Frágreiðingin skal enda við einari stuttari fakligari meting av úrslitunum og niðurstøðu (diskussión og konklusión).

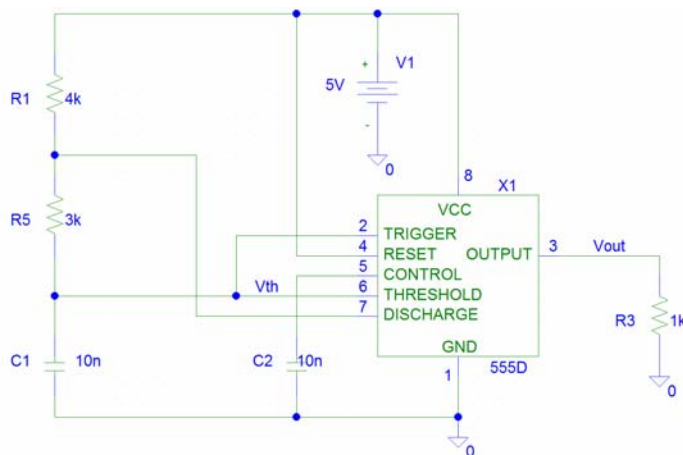
Vit royna í fyrsta umfari eina trýstøðu uppseting og velja tíðarbilini, har reytt, gult og grønt ljós eru av hóskaði tíðarlongd, t.d tíðarraðfylgjuna í talvuni, har hvør puntur svarar til tíðarbili T (sek):

Norður – Suður	R	R	R	R	R	R	R	R	R	R	R	G	G	G	G	G	G	Y	Y
Eystur – Vestur	R	G	G	G	G	G	G	Y	Y	R	R	R	R	R	R	R	R	R	R

1. Tíðarraðfylgjuras - klokkusignalgerði

Í mynd 2 er ein tíðarraðfylgjuras (“timer”) víst, har virðini av mótstøðum og kondensatorum eru vald tilvildarligt. Á útganginum á 555 verður skaptur ein periodiskur fýrkantaður spenningur við niðara og ovara virði ávikavist umleið 0 og 5 volt. C_1 , R_5 og R_1 avgera periodustøddina. Kondensatorurin C_2 kann veljast frítt innan fyri víð mørk, men er settur til 10 nanofarad í hesi uppstillingini. Sí annars datablaðið fyri “timer”-IC

rásina, *skjal 3*, og ein viðgerð av rásini finst í *skjali 4*.



Mynd 2 ”Timer 555”-rás nýtt til klokkusignalgerða.

Spurningar at svara áðrenn møtt verður í starvsstovuni:

- 1.1. Greið frá, hvussu henda rásin virkar, og kann geva fyrkantaðar pulsar út (V_{out}). Brúka uppgivnu keldur og databløðum at svara spurninginum.
- 1.2. Vís og grundgev fyri:
 - At kondensatorurin C_1 hevur uppløðingartíðina $t_1=0,693 \cdot (R_1 + R_5) \cdot C_1$
 - At kondensatorurin C_1 hevur avløðingartíðina $t_2=0,693 \cdot R_5 \cdot C_1$
 - At signalfrekvensurin hevur støddina givna við hesum formlinum
$$f = \frac{1}{T} = \frac{1,44}{(R_1 + 2R_5) C_1}$$
- 1.3. Finn virði av R_1 , R_5 og C_1 soleiðis, at periodan verður $T = 1$ sekund? (T er sett til 1 sekund av praktiskum áðum. Í veruligari skipan skal T vera nakað longri, t.d. 3 – 5 sekund.)

Spurningar at svara við brúk av teldusimuleringsforritinum PSpice:

- 1.3. Ger eina uppseting við Pspice forritinum av tíðarfylgurrásini í telduni við funnu virðunum fyri R_1 , R_5 og C . Prenta tekningina (diagrammið) til at seta í frágreiðingina.
- 1.4. Ger eina simulering av útgangsspenninginum, og spenninginum yvir C_1 sum funktión av tíðini. Prenta myndina til at skjalfesta úrslitið.

Spurningar at svara í starvsstovuni:

- 1.5. Ger eina starvsstovuuppseting av tíðarfylgurrásini. Skjalfest uppsetingina við eini tekning (diagrammi).
- 1.6. Mátu útgangsspenningin og spenningin yvir C_1 sum funktión av tíðini.
- 1.7. Hvussu neyvt er frekvensurin f tann rætti.

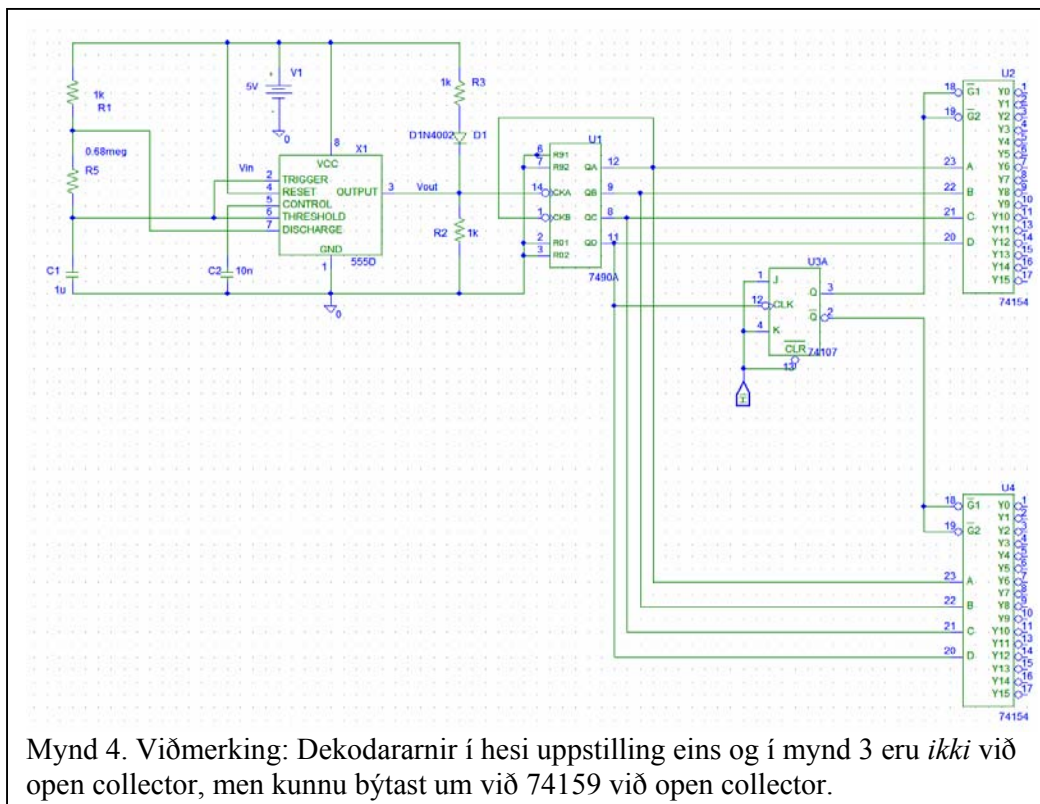
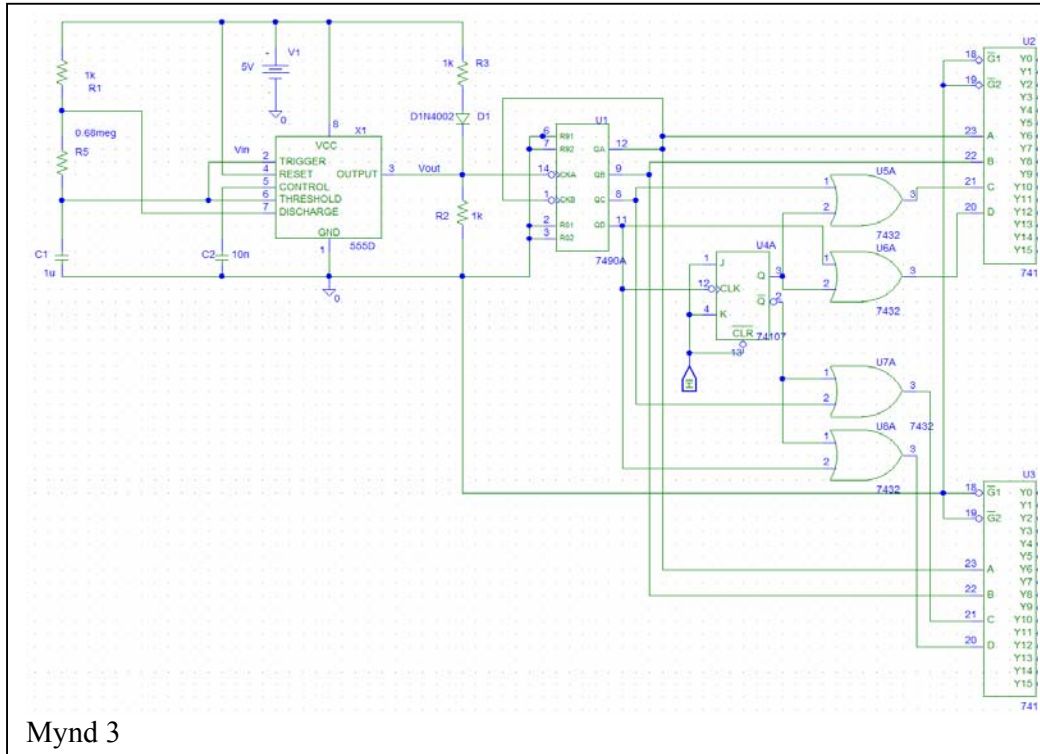
2. Teljara og dekodararás stýrd av klokkusignalgerði

Endamálið við rásini er at gera 20 útgangir, sum eftir túri vera aktivir (í hesi royndini eru rásútgangirnir aktivt-lágir).

Klokkusignalgerðin umrøddur í 1. verður nú brúktur sum innsignal til ein dekadiskan teljara. Útgangsbittini frá teljaranum verða dekodaði í tveimum dekodarum, harav hvør ger 10 útgangssignal, t.e. 20 í alt. Bert eitt av teimum skal verða aktivt (lágt). Teljarin skal sostatt fáa fyrst annan dekodaran at skifta sekventielt gjøgnum 10 ymsar støður og síðani hin.

Dekodararnir, ið verða brúktir, hava ”open collector”, tað merkir, at teir skulu hava eina ytri last, tá teir verða settir inn í eina veruliga rás. Teir hava í veruleikanum 16 útgangir hvør, men vit brúka bert 10 teir fyrstu beinleiðis sum útsignal í hesi royndini.

Ein JK flipp-floppur (JK vippa) (skjal verður brúktur til at stýra hvør av dekodarunum skal svara uppá útgangin frá teljaranum. Hetta kann verða gjørt á tvinnanda mátar, sum víst er í mynd 3 og 4. Úrslitið frá dekodarunum er tað sama.



Spurningar at svara við brúk av teldusimuleringsforritinum PSpice:

- 2.1. Ger eina uppseting við Pspice forritinum av tíðarfylgurrásini í telduni sum framhald av uppsetingini frá 1.1 við antin at brúka uppseting mynd 3 ella 4. Prenta tekningina (diagrammið) til at seta í frágreiðingina. (Uppsetingin í mynd 4 er einfaldari, og hevur færri komponentar og er tí lættari at byggja víðari uppá uttan at renna seg ímóti avmarkingunum í (evalueringsversiónini av Pspice forritinum).
- 2.2. Ger eina simulering av útgangssignalunum á báðum dekodarunum (Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8, Y9, Y10, Y11, Y12, Y13, Y14 og Y15) og útgangssignalunum frá teljaranum, umframt útgangsspenninginum Vout og spenninginum yvir C_1 sum funktiión av tíðini. Tulka úrslitið við. Prenta myndina til at skjalfesta úrslitið.

Spurningar at svara í starvsstovuni:

- 2.3. Ger eina starvsstovuuppseting av samlaðu rásini í mynd 3 ella 4. Skjalfest uppsetingina við eini tekning (diagrammi).
- 2.4. Mátu útsignalini frá dekodarunum og teljaranum saman við útgangsspenninginum og spenninginum yvir C_1 sum funktiión av tíðini. Skjalfest úrslitini.
- 2.5. Grundgev fyri, at bert útsignalini Y0...Y9 verða brúkt til at stýra ljóssignalinum.

3. Trafikkjós stýrd av dekodarauppseting

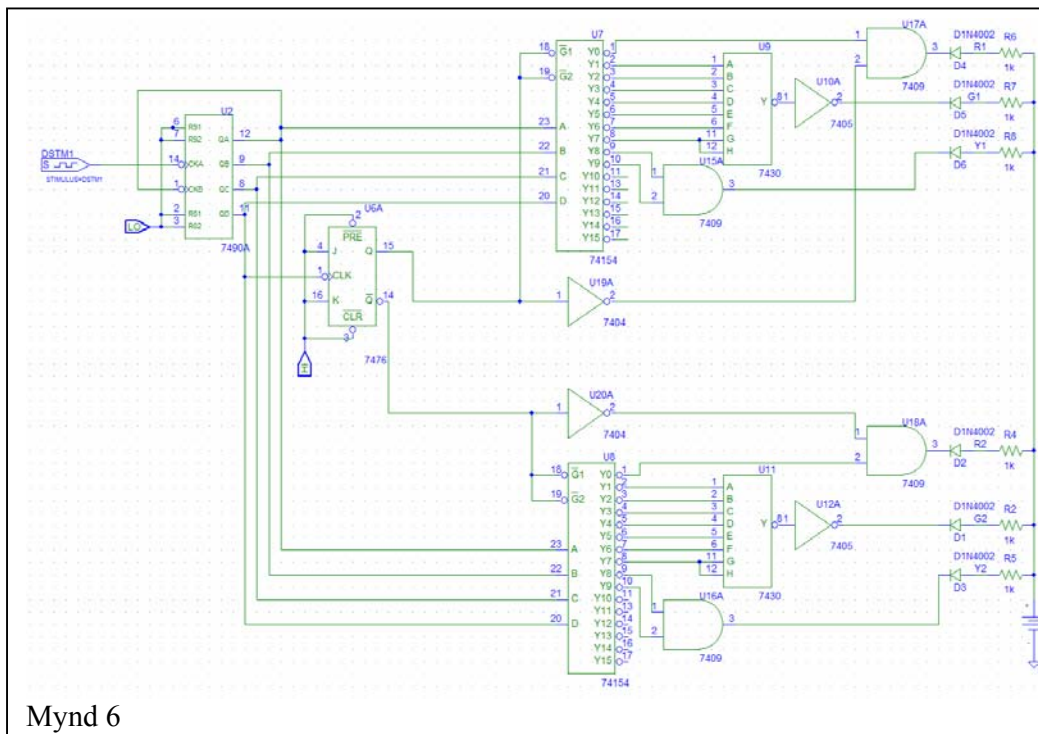
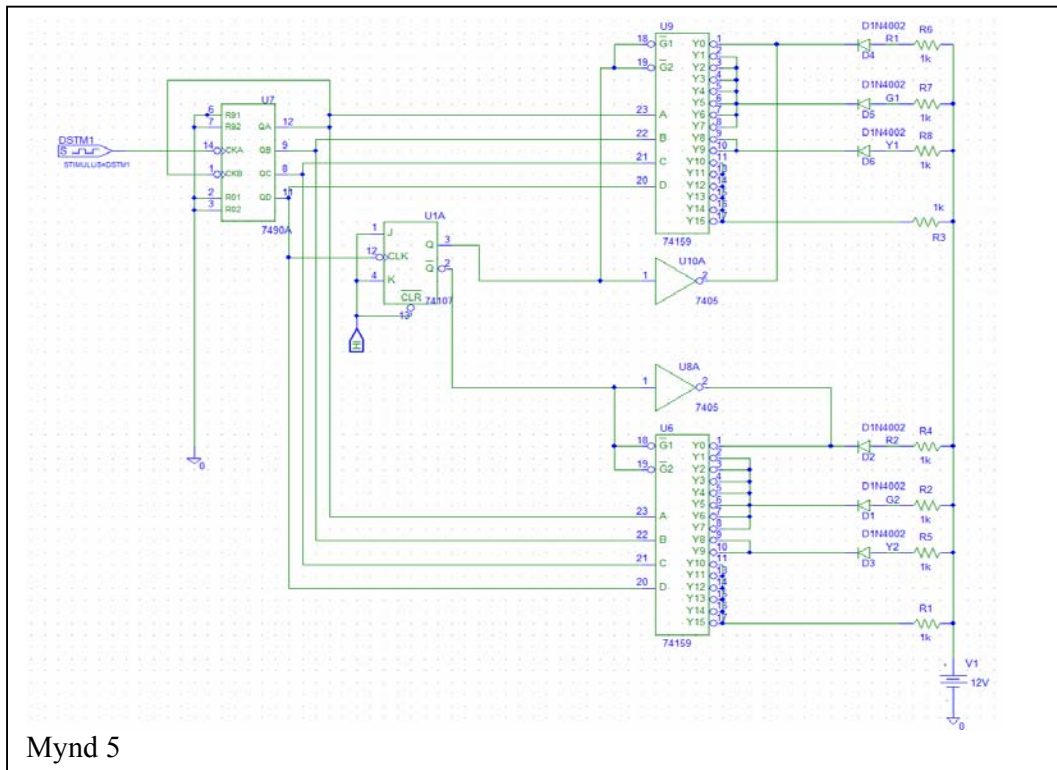
Uppsetingin frá parti 2 verður nú útbygd við reyðum, gulum og grønum ljósdiodum til at mynda eitt modell av trafikkjósum, ið verða stýrd av dekodarauppsetingini. Ein møguleiki av hesi uppsetingini er vístur í mynd 5. Her er Pspicemodell av klokkusignalgerða sett í staðin fyri tann uppbygdu klokkusignalgerðarrásina frá parti 1 fyri at sleppa undan at verða avmarkaður av evaluerings-treytunum í Pspice.

Spurningar at svara við brúk av teldusimuleringsforritinum PSpice:

- 3.1. Ger eina uppseting í telduni við Pspice forritinum av tíðarraðfylgurrásini í mynd 4 sum framhald av uppsetingini frá 2.1. Prenta tekningina (diagrammið) til at seta í frágreiðingina.
- 3.2. Ger eina simulering av ljósdiodusignalunum. Prenta myndina til at skjalfesta úrslitið.
- 3.3. Ger eina nágreiniliga tuling av funktiiónini av uppsettu ljósdiodurásini við valdu dekodarunum, ið hava "open collector" útgang. Tulka eisini úrslitini.

Spurningar at svara í starvsstovuni:

- 3.4. Ger eina starvsstovuuppseting av samlaðu rásini í mynd 4. Skjalfest uppsetingina við eini tekning (diagrammi).
- 3.5. Vís, at ljósdiodurnar tendra og slókna í rættari raðfylgju.



Tað visir seg at vera torført at finna tann brúkta dekodaran 74159 við open collector. Tí er eitt alternatívt uppdkot sett upp í mynd 6, har dekodarin 74154, ið *ikki* hefur open collector er nýttur. Tað krevur so nakrar eyka gatir, har summar hava open

collector. Um ljósini eru ljósdiodur, sum víst, er kanska móguligt at fáa uppsetingina at virka við vanligum gatum eisini.

Eyka spurningar

3.6. Hvussu kann uppsetingin verða broytt so, at ljósini skifta í hesi raðfylgju:

Norður – Suður	R	R	R	R	R	R	R	R	R	R	R/Y	G	G	G	G	G	G	Y	Y
Eystur – Vestur	R/Y	G	G	G	G	G	G	G	Y	Y	R	R	R	R	R	R	R	R	R

Har R/Y stendur fyri at bæði reytt og gult lýsa samstundis

- 3.7. Simulera hesa uppseting við Pspice
- 3.8. Vís, at hon eisini virkar í starvsstovuni.

4. Tilvísingar

1. ”Traffic Light Control Circuit”, <http://home.cogeco.ca/~rpaisley4/20step.html>
2. Magnus Danielsen, “INNLEIÐANDI ELEKTRONIKKUR, Støði undir digitalum elektronikki”, NVDrit 2005:12, 2005, sí Kapittul 10. PSpice simuleringforrit, stutt leiðbeining
http://setur.fo/uploads/tx_userpubrep/NVDrit2005_12.pdf
3. National Semiconductor, ”LM555 – Timer”,
<http://www.national.com/ds/LM/LM555.pdf>
4. Tony van Roon: “555 Timer/Oscillator Tutorial”,
<http://www.uoguelph.ca/~antoon/gadgets/555/555.html>
5. Dekodari: 4 til 16
<http://www.aeromech.usyd.edu.au/MTRXLAB/ref/components/TTL/DM74LS154.pdf>. ella
<http://web.mit.edu/6.115/www/datasheets/74hc154.pdf>.
6. Dekodari: 4 til 10 BCD
http://www.nxp.com/acrobat_download/datasheets/74HC_HCT42_CNV_2.pdf

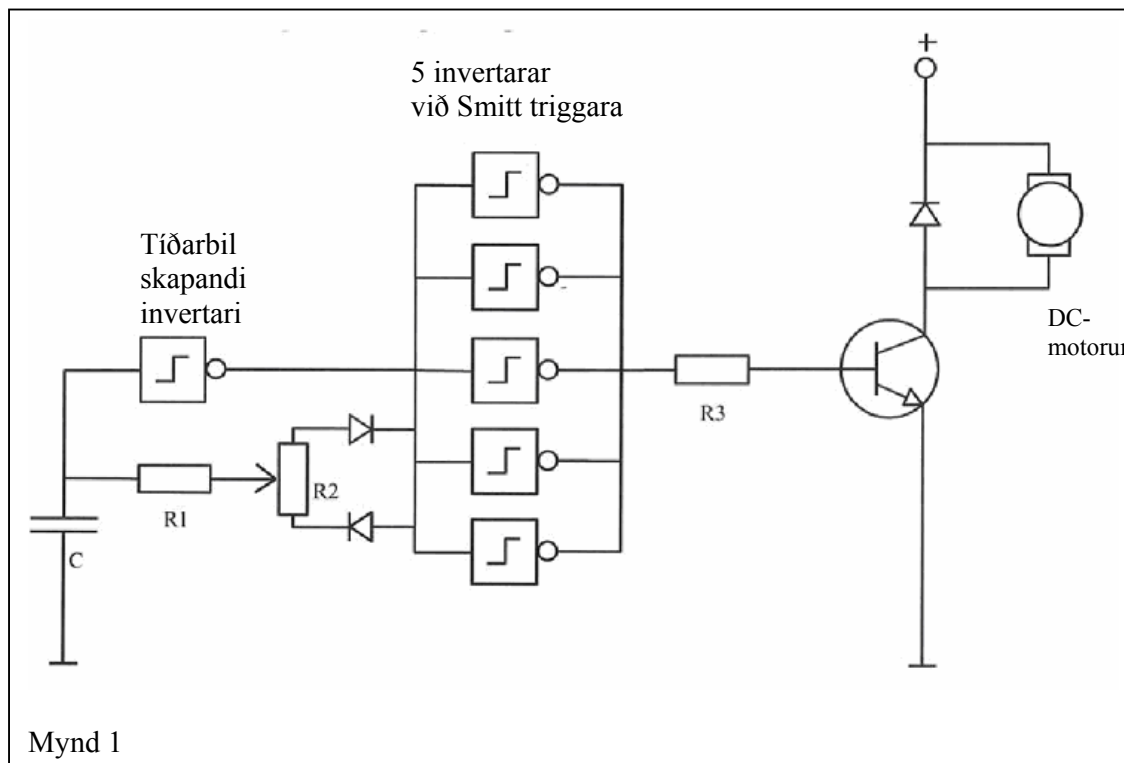
II. Switch-mode regulering til DC motor

Benadikt Joensen

Í venjingini verður ein rás sett upp til regulering av DC motori við støði í hugtakinum “swiltch-mode regulering”, har motorurin fær streym sum streympulsar, konstantir í stødd, men við skiftandi tíðarlongd soleiðis, at miðalvirðið á streyminum skiftir alt eftir longdini av pulsunum og stýrir, hvussu skjótt motorurin skal ganga. Trý streymrásupp-skot eru til, hvussu hetta kann verða gjørt. Tíðarbilgerandi eindin av uppsetingunum er ein astabilur multivibrator, uppbygður við ávikavist invertara við innbygðum schmitt triggara, operatiónsstyrkjara og “555-timara”. Streymberandi stýrieindin er eisini ymisk í uppsetingunum, har antin ein bipolerur transistor ella ein felteffekt transistor (FET) verður brúktur.

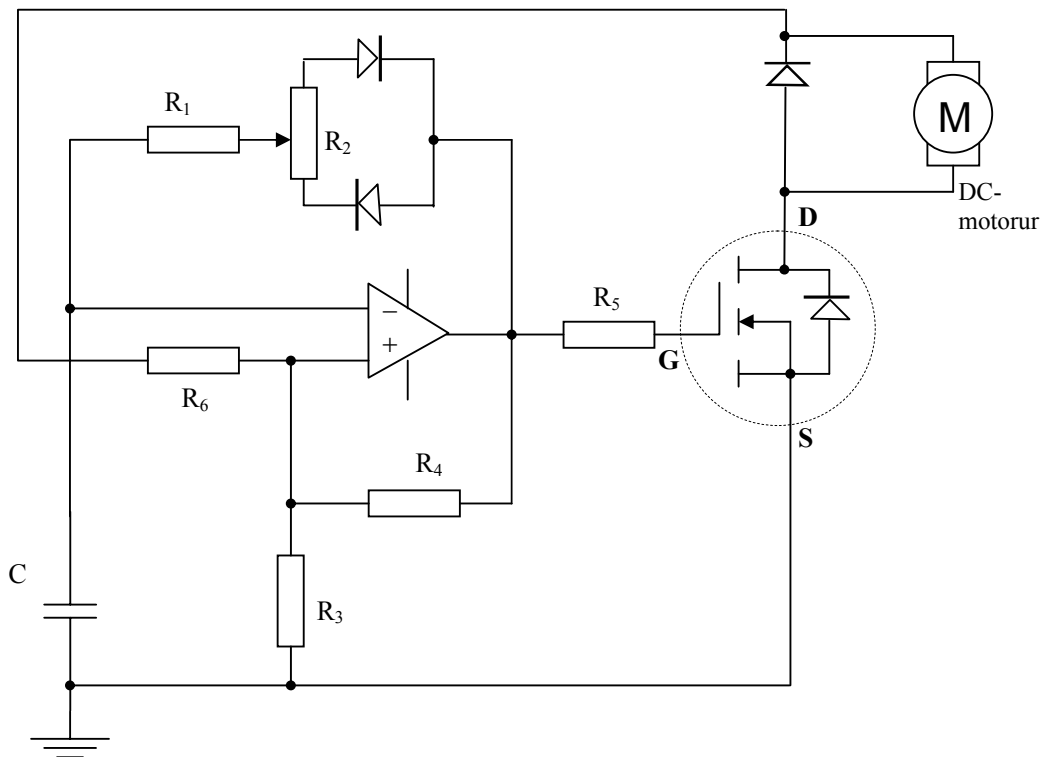
1. Uppseting við tíðarbil gerandi eind við støði í invertara

Sum víst er í mynd 1 er tíðarbil skapandi eindin ein invertari við innbygðum schmitt triggara, og eini positivari “feed-back” rás uppbygd av mótstøðuni R_1 , potentiometrinum R_2 , og tveimum diodum, umframt einum kondensatori C , ið saman við mótstøðunum avger tíðarbilið, tá streymurin til motorin skal vera høgur. 5 inverrarar við innbygðum schmitt triggaram í parallell eru settir í seriu við mótstøðu R_3 og binda tíðarbileindina við ein DC motor gjøgnum streymst stýrieindina, ið her er ein bipolerur transisitor.



2. Uppseting við tíðarbil gerandi eind við stöði í operatíonsstyrkjara

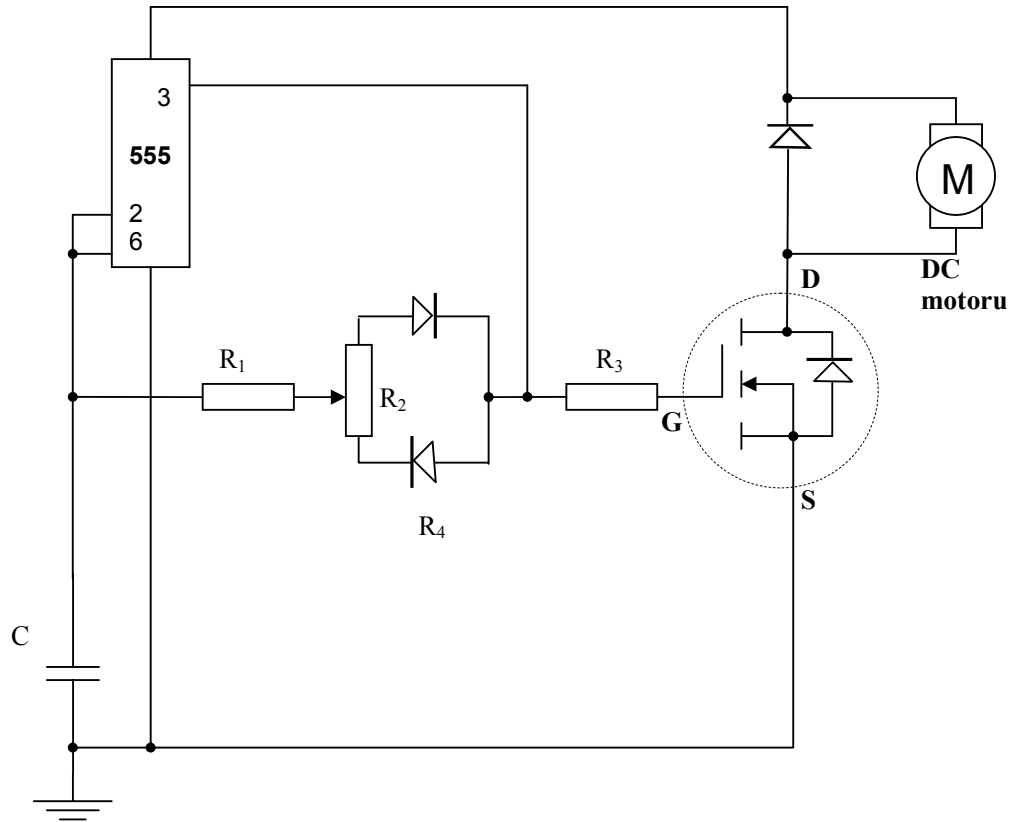
Sum víst er í mynd 2 er tíðarbil skapandi eindin ein operatíonsstyrkjari í astabilari multivibratorrás, við eini positívri “feed-back” rás uppbygd av mótstöðuni R_1 , potentiometrinum R_2 , og tveimum diodum, umframt einum kondensatori C , ið saman við mótstöðunum R_3 , R_4 og R_6 verður ein hóskandi spenningsstödd lögð fyri tann lága spenningin í pulsðini. Útgangurin frá astabila multivibratorinum verður bundin gjøgnum mótstöðuna R_5 saman við ein DC motor gjøgnum streymberandi stýrieindina, ið her er ein felteffekt transistor (FET). Felteffekt transistorin er tryggjaður móti ov høgum spenningi við diodu.



Mynd 2

3. Uppseting við tíðarbil gerandi eind við stöði í operatíonsstyrkjara

Sum vís er í mynd 3 er tíðarbil skapandi eindin ein “555-timara” bundin til mótstöðuni R_1 , potentiometrinum R_2 , og tvær diodur, umframt ein kondensator C , ið saman við mótstöðunum avger tíðarbilið, tá streymurinn til motorinn skal vera høgur. Útgangurinn frá “555-timaranum” verður bundin gjøgnum mótstöðuna R_3 saman við ein DC motor gjøgnum streymberandi stýrieindina, ið her er ein felteffekt transistor (FET). Felteffekt transistorin er tryggjaður móti ov høgum spenningi við diodu.



Mynd 3

4. Venjingarskrá

Venjingarskráin er tann sama uttan mun til omanfyri givnu streymrásir.

- **Útrokningar áðrenn uppstillingin verður sett saman:**

- 1 Vel eina av nevndu rásunum til at uppbyggja og kanna.
- 2 Avger ella rokna passandi virði fyri allar mótstöðurnar R_i og kondensatorin C , sum eru við til at fastleggja frekvensin. (Um t.d. R er givin, skal C roknast).
- 3 Set streymrásina saman sambært funnu elektronisku lutunum.

- **Mátingar (Minst til at avlesa øll relevant virði á ossilloskopinum!!)**

Máta við ossilloskopi, hvussu hesi signal síggja út við t/T umleið 0,5, har t er pulslongdin og T er periodulongdin:

- 4 Spenningurin yvir mótstöðuna, ið er bundin til basis ella gate (t.e. R_3 ella R_5).
- 5 Inngangsspenningurin á transistorinum (t.e. basisspenningur ella gatespenningur í mun til jørð).
- 6 Útspenningur á transistorinum (t.e. kollektor-emitter spenningur ella drain-source spenningur)

Máta við ossilloskopi, hvussu hesi signal síggja út við t/T umleið 0,1, umleið 0,5 og umleið 0,9, har t er pulslongdin og T er periodulongdin.

- 7 Inn- og útspenningur á ossillatorrásini
- 8 Spenningur yvir motorin og streymur gjøgnum motorin
- 9 Hygg at streyminum í einum resistansi, sum verður settur í staðin fyri motorin

Máta við ossilloskopi og voltmetri:

- 10 Hvussu hongur motorspenningurin saman við t/T fyri t/T umleið 0,1 - 0,3 - 0,5 - 0,7 og 0,9.

- **Útrokningar aftaná mátingina:**

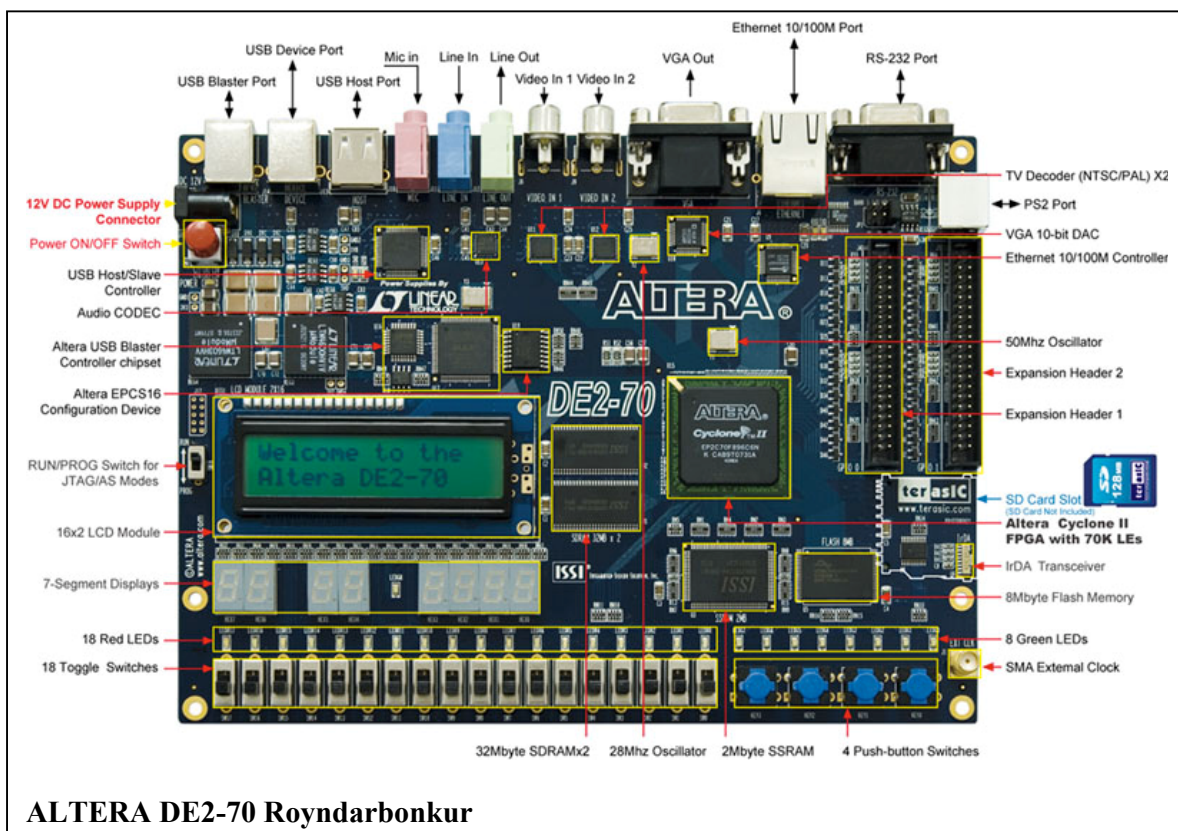
- 11 Samanber allar mátingar við útrokningar.
- 12 Skriva frágreiðing um venjingina, við frágreiðandi teksti um, hvussu reguleringin virkar, úrslitum av útrokningunum, úrslitum av mátingunum, samanbering av útrokningum og mátingum, og einari diskussión av, hvussu samsvarið er millum hesi.

III. Kombinatoriskar rásir

Magnus Danielsen

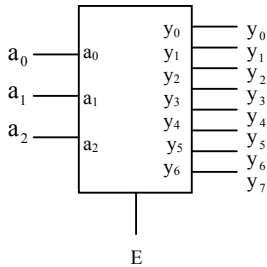
Í hesi venjingini verður Quartus II brúkt til forritan av FPGA (ella CPLD) rásum við brúk av VHDL og Schematic. Sum dømi um VHDL og Schematic forritan av kombinatoriskum rásum verða her tikin upp til viðgerðar hesi dømi: 1.Dekodrar, 2.Innkodrar, 3.Komparator, og 4.Ein lítill stýriskipan til at tendra og slökkja útljýsið til eini sethús til tess at tendra, tá hósandi er, og fyri at spara orku.

Til forritanina av FPGA verður ein ALTERA DE2-70 royndarbockur brúktur sum vístur er í myndini. Altera DE2-70 royndarbockurin hevur umframt ein FPGA, nógva útgerð, millum annað tveir innbygdar klókkusignalgerðum 28 MHz og 50 MHz, 18 digitalar 0/1-signal umskiftikontaktum, 4 0/1-signal trústkontaktir, 18 reyðar og 9 grønar ljósdiodur til lesing av digitalum signalum, LCD display og fleiri markamót til tilknýtta útgerð og mátitól til FPGA'in. Til betri útgreining av móguleikum og handfaring av hesum bocki umframt brúk av Quartus II til VHDL forritan og Schematic forritan verður víst til tilvísingar [1-4].



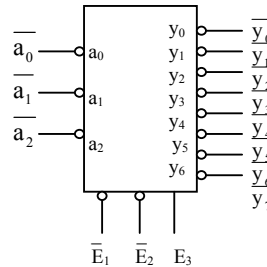
1. Dekodarar

Viðgjördir verða oktalir dekodarar. Ein oktalur dekodari er ein dekodari, ið hefur $2^3 = 8$ útgangssignal og 3 inngangssignal soleiðis, at tað binera talið, ið er myndað av inngangs-signalunum peikar á útgangin (t.e. ger útgangin aktivan), ið hefur hetta talið sum nummar. Útgangirnir kunnu antin vera aktivt høgir ella lágir. Í venjingini verða gjördir tveir oktalir dekodarar, sum vístir eru í mynd 1.1 og mynd 1.2.



Mynd 1.1

Oktalur dekodari við aktivt høgum inn- og útgangum



Mynd 1.2

Oktalur dekodari 74138

Oktalur dekodari – aktivt høgir inn- og útgangir. Ein enable inngangur

Oktali dekodarin, sum vístur er í mynd 1.1, hefur hesar eginleikar:

- 3 aktivt høgur inngangir a_0, a_1, a_2
- 8 aktivt høgur útgangir $y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7$
- 1 enable inngang E, ið er aktivt høgur.

1.1 Skriva eitt VHDL forrit fyri dekodaran

1.2 Simulera dekodaran við at uppseta eitt signal royndarforrit ("waveform program"), ið vísir úrslitini av útgangssignalunum

$y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7$ fyri allar móguleikar av inngangssignalunum $a_0, a_1, a_2 = 000, 001, \dots, 111$, og enable = $E = 0, 1$.

1.3 Sniðgev forritið til ein Altera DE2-70 royndarbank við einari integreraðari rás av FPGA slagnum soleiðis, at ljósdiodur vísa ymsu signalini, og inngangssignalini og enable signalið verða sett við hartil hoyrandi kontaktum.

1.4 "Downloada" forritið á FPGA'in.

1.5 Royn forritaða FPGA'in, skjalfest úrslitið og samanber við simuleringina.

Oktalur dekodari 74138 – aktivt lágir inn- og útgangir. Fleiri enable inngangir.

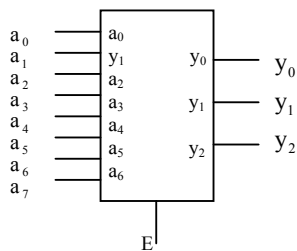
Oktali dekodarin, vístur í mynd 1.2, hefur eginleikar sum oktalur dekodari 74138, t.e.

- 3 aktivt lágur inngangir $\overline{a_0}, \overline{a_1}, \overline{a_2}$
- 8 aktivt lágur útgangir $\overline{y_0}, \overline{y_1}, \overline{y_2}, \overline{y_3}, \overline{y_4}, \overline{y_5}, \overline{y_6}, \overline{y_7}$
- 3 "enable" inngangir $\overline{E}_1, \overline{E}_2, E_3$, har ein er aktivt høgur, og tveir eru aktivt lágir.

1.6 Skriva eitt VHDL forrit fyri dekodaran. Hetta kann gerast við at broyta forritið frá dekodaranum í spurningi 1.1 hóskaði.

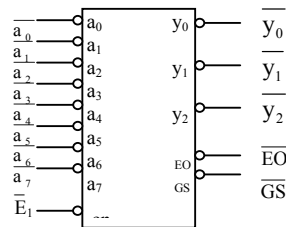
- 1.7 Simulera dekodaran við at uppseta eitt signal royndarforrit, ið vísir úrslitini av útgangssignalunum $\overline{y_0}, \overline{y_1}, \overline{y_2}, \overline{y_3}, \overline{y_4}, \overline{y_5}, \overline{y_6}, \overline{y_7}$ fyri allar móguleikar av inngangssignalunum $\overline{a_0}, \overline{a_1}, \overline{a_2} = 000, 001, \dots, 111$, og enable inngangssignalunum $\overline{E_1}, \overline{E_2}, E_3 = 000, 001, \dots, 111$. Hetta kann gerast við at broyta signal royndarforritið frá spurningi 1.2 hóskaði.
- 1.8 Sniðgev forritið til ein Altera DE2-70 royndarfonk við einari integreraðari rás av FPGA slagnum soleiðis, at ljósdiodur vísa ymsu signalini, svarandi til, at inngangssignalini og enable signalini verða sett við hartil hoyrandi kontaktum.
- 1.9 ”Downloada” forritið á FPGA’in.
- 1.10 Royn forritaða FPGA’in, skjalfest úrslitið og samanber við simuleringina.

2. Prioritets innkodarar



Mynd 2.1

Oktalur prioritets innkodari við aktivt høgum inn- og útgangum



Mynd 2.2

Oktalur prioritets innkodari 74138 við aktivt lágum inn- og útgangum

Ein oktalur innkodari hevur $2^3 = 8$ inngangssignal og 3 útgangssignal soleiðis, at tað binera talið, ið útgangssignalini mynda, er nummarið á tí innganginum, sum er aktivur. Útgangir og inngangir kunnu antin vera aktivt høgir ella lágir. Innkodarin hevur tann veikleika, at hann gevur rætt úrslit um bert ein inngangur er aktivur. Eru fleiri aktivir samstundis kemur fløkja í, og innkodarin kann ikki geva rætt úrslit.

SN54/74LS148
SN54/74LS748
FUNCTION TABLE

INPUTS		OUTPUTS											
EI	0	1	2	3	4	5	6	7	A2	A1	A0	GS	EO
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	L	H	H	L	H	L	L	H
L	X	X	X	X	L	H	H	H	L	H	L	L	H
L	X	X	X	L	H	H	H	H	H	L	L	L	H
L	X	L	H	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H

Mynd 2.3 Sannleikatalva fyri oktalan prioritets innkodara

Ein prioritets innkodari líkist ”vanliga” innkodaranum við tað, at hann virkar sum hesin, um bert ein av inngangunum er aktivur. Eru harafturímóti fleiri inngangir aktivir, gevur hann útgangsúrslitið, ið svarar til nummarið á tí innganginum, sum hevur hægst nummar. Í venjingini verða gjørdir tveir oktalir prioritets innkodarar, vísir í mynd 2.1 og 2.2. Prioritetsinnkodarin í mynd 2.2 hevur sannleika-talvuna í mynd 2.3

Oktalur prioritets innkodari – aktivt høgir inn- og útgangir. Ein enable inngangur
Oktali prioritets innkodarin, sum vísur er í mynd 2.1, hevur hesar eginleikar:

- 8 aktivt høgir inngangir
- 3 aktivt høgir útgangir
- 1 enable inngang, ið er aktivt høgur.

- 2.1 Skriva eitt VHDL forrit fyrir prioritets innkodaran
- 2.2 Simulera innkodaran við at uppseta eitt signal royndarforrit, ið vísir úrslitini av útgangssignalunum y_0, y_1, y_2 fyrir eitt hóskandi tal av móguleikum av inngangssignalunum $a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7 = 00000000, 00000001, 00000010, 00000011, \dots, 11111111$, og $\text{enable} = E = 0$.
1. Av tí at móguleikarnir her eru so nógvir, er neyðugt at avmarka royndina til hóskandi tal av viðkomandi móguleikum.
- 2.3 Sniðgev forritið til ein Altera DE2-70 royndarbank við einari integreraðari rás av FPGA slagnum soleiðis, at ljósdiodur vísa ymsu signalini, og inngangssignalini og enable signalið verða sett við hartil hoyrandi kontaktum.
- 2.4 ”Downloada” forritið á FPGA’in.
- 2.5 Royn forritaða FPGA’in, skjalfest úrslitið og samanber við simuleringina.

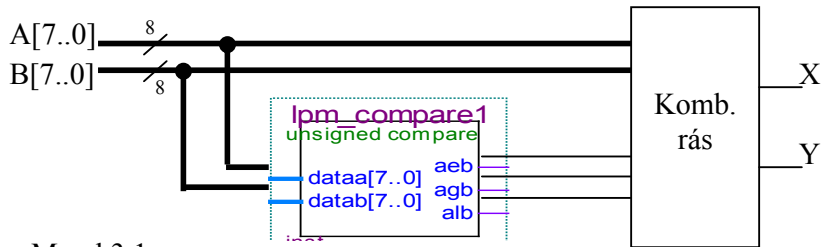
Oktalur prioritets innkodari 74148 – aktivt høgir inn- og útgangir. Ein enable inngangur, ein enable útgangur og ein ”group signal” útgangur.

Oktali prioritets innkodarin, sum er vístur í mynd 2.2 hevur somu eginleikar sum oktali prioritets innkodari 74148, t.e. hevur sambært sannleikatalvuni í mynd 2.3:

- 8 aktivt lágar inngangir
 - 1 aktivt lágan enable inngang EI (Enable In).
 - 3 aktivt lágar útgangir
 - 1 aktivt lágan GS (Group Signal ella Group Select) útgang, ið er aktivur, tá bert onkur inngangur er aktivur og EI er aktivur.
 - 1 aktivt lágan enable EO (Enable Output) útgang, ið er aktivur, tá eingin inngangur er aktivur, og EI er aktivur.
- 2.6 Skriva eitt VHDL forrit fyrir dekodaran.
Hetta kann gerast við at broyta forritið frá dekodaranum í spurningi 1.1 hóskandi.
 - 2.7 Simulera dekodarin við at uppseta eitt signal royndarforrit, ið vísir úrslitini av útgangssignalunum $\overline{y_0}, \overline{y_1}, \overline{y_2}, \overline{EO}$ og \overline{GS} allar móguleikar av inngangssignalunum $\overline{a_0}, \overline{a_1}, \overline{a_2}, \overline{a_3}, \overline{a_4}, \overline{a_5}, \overline{a_6}, \overline{a_7} = 00000000, 00000001, 00000010, 00000011, \dots, 11111111$, og $\overline{EI} = 0, 1$.
Hetta kann gerast við at broyta signalroyndarforritið frá spurningi 1.2 hóskandi.
 - 2.8 Sniðgev forritið til ein Altera DE2-70 royndarbank við einari integreraðari rás av FPGA slagnum soleiðis, at ljósdiodur vísa ymsu signalini, og inngangssignalini og enable signalini verða sett við hartil hoyrandi kontaktum.
 - 2.9 ”Downloada” forritið á FPGA’in.
 - 2.10 Royn forritaða FPGA’in, skjalfest úrslitið og samanber við simuleringina.

3. Komparator

Í Quartus II er eitt savn av forritaðum modellum av kombinatoriskum rásum til Schematic, ið hava móguleikan til at verða tillagaðir til ávís endamál. Hesar hava prinsipiellar funktiónir so sum teljarar, komparatorar ,addarar o.s.v. Tað, sum sermerkir hesi model er, at tað er móguligt at tillaga inngangir, útgangir og funktiónir av hvørjum einstøkum luti eftir ynski til at kunna nøkta tað funktión tørvur er á.



Mynd 3.1

Í hesi venjingini verður ein komparator valdur til at gera eina skipan, ið samanber tvey 8 bit biner tøl A og B og avger, um tey eru eins, samstundis sum avgjørt verður, um tey eru ólíka ella líka tøl. Royndaruppsetingin er víst í mynd 3.1.

Komparatormodellið `lpm_compare` verður brúkt. Nakrar av støddunum A og B, umframt útgangssignalini `aeb`, `agb`, og `alb` verða brúkt sum inngangssignal til eina kombinatoriska rás, ið skal gera útgangssignalini X og Y soleiðis, at

- $X = 1$, tá $A = B$, og bæði A og B eru líka tøl
- $Y = 1$, tá $A = B$, og bæði A og B eru ólíka tøl

Ger forrit við Schematic til henda komparatorin og forrita ein FPGA við honum.

- 3.1 Tillaga komparatorin soleiðis, at hann hevur tveir 8 bit datainngangir og trýggjar útgangir `aeb` (a equal to b), `agb` (a greater than b) og `alb` (a less than b).
- 3.2 Ger eina kombinatoriska rás, sum við A, B, `aeb`, `agb` og `alb` gevur tey ynsktu signalini X og Y á útganginum. Forrita hana við Schematic.
- 3.3 Simulera komparatoruppsetingina við at uppseta eitt signal royndarforrit, ið vísir úrslitini av útgangssignalunum X og Y við hóskandi úrvali av inngangssignalum A og B.
- 3.4 Sniðgev forritið til ein Altera DE2-70 royndarbank við einari integreraðari rás av FPGA slagnum soleiðis, at ljósdiódur vísa ymsu signalini, og inngangssignalini verða sett við hartil hoyrandi kontaktum.
- 3.5 ”Downloada” forritið á FPGA’in.
- 3.6 Royn forritaða FPGA’in, skjalfest úrslitið og samanber við simuleringina.

4. Stýriskipan til at tendra og slökkja útljósið við

Uttan fyri eini sethús er lampa til útljós, ið er ætlað at lýsa, tá brúk er fyri ljósi, og at vera slókt annars til tess at geva eitt nøktandi umhvørvi, men samstundis spara mest møguliga orku. Hetta kann verða gjørt við kombinatoriskari rás. Skipað skal soleiðis vera fyri, at

- Um dagin, tá ljóst er, skal lampan vera slókt
- Tá dagur tekur at halla, og tað verður myrkt og fram til miðnátt, skal lampan vera tendrað so, at tað verður lagaligt at koma til húsið.
- Eftir miðnátt verður lampan slókt fyri at spara streym
- Men kemur onkur, sjálvst aftan á miðnátt, t.d. blaðútberingin ella onkur seinur gestur inn um garðsportrið, skal lampan tendrað, men slóknar aftur, tá garðsportrið verður latið aftur (møguliga við tíðarseinking).

Ljósið verður mátað við fotodiodu, ið leiðir streym, meðan ljóst er og ongan streym í myrkri. Streymurin gevur logiska signali $L = 1$, tá fotodiodan leiðir streym, og $L = 0$, tá fotodiodan ikki ber streym.

Eitt logiskt signal K kemur frá klokku soleiðis, at $K = 1$ um klokkan er 00-12, tá lampan er slókt, um portrið er aftur, og $K = 0$ um klokkan er 12-24, tá lampan er tendrað, um myrkt er.

Eitt relé mátar um garðsportrið er aftur ella opið. Frá tí kemur eitt logiskt signal G soleiðis, at $G = 0$, um tað er aftur, og $G = 1$ um tað er opið.

Til at stýra um lampan skal vera tendrað ella slókt, verður logisk rás uppbygd við inngangssignalunum L , K og G og einum útgangssignali F , ið stýrir, um lampan er tendrað ella slókt.

Til tess at tryggja, at lampan tendar, um signalið á útganginum á logisku rásini fellur burtur, skal lampan tendra, tá $F = 0$, og lampan slóknar tá $F = 1$.

Til at gera hetta í veruleikanum verður ein FPGA programmeraður til at gera hesa logisku funktiónia F . Til at fremja hetta verða hesir spurningarnir svaraðir:

- 4.1 Ger eina sannleikatalvu av variablunum L K G og F
- 4.2 Set eitt logiskt sum av produktum upp fyri F
- 4.3 Ger eitt VHDL program fyri at finna F sum funktión av L K G
- 4.4 Simulera F
- 4.5 Endurtak 3. og 4. við Schematics
- 4.6 Programmera ein FPGA við programminum, og royn um funktiónin er røtt
- 4.7 Skjalfest hvørt punktið við útskriftum og notatum

Tilvísingar:

- [1] ALTERA DE2-70 User Manual
- [2] Getting Started with Altera DE2-70 board
- [3] Quartus II Introduction Using VHDL Design
- [4] Quartus II Introduction Using Schematic Design

IV. CMI – Koding og deking til ljósleiðara datasendingu

Magnus Danielsen

1. Inngangur

Venjingin snýr seg um greiningu av uppbygging og virkíhætti av CMI kodingsskipanini, ið m.a. verður brúkt í optiskum transmissiónsskipanum. Skipanin er uppbygd av undireind-um, sum hvør sær fyrst verður uppbygd og kannað. Síðan verða tær bundnar saman til eina samhangandi transmissiónsskipan, hvørs transmissiónseiginleikar verða kannaðir.

Starvsstovuvenjingin er áður gjørd við støði í skjalinum NVDRit 2005:13, har CMI kodari og CMI-dekodari verða uppbygdir við brúk av SSI (Small-Scale Integration) og MSI (Medium-Scale Integration) IC-chippum. Vit skulu í hesi venjingini viðgera CMI kodara og CMI dekodara við somu funktiónum, men verða teir nú gjørdir við at forrita ein VLSI (Very Large Scale Integration) FPGA chip (CPLD chippur kann eisini verða brúktur) við Schematic ella VHDL forritan. Til forritanina og simulering av skipanini verður Quartus II brúkt. Spurningarnir viðvíkjandi forritanini verða fyrst orðaðir, svarandi til brúk av Schematic. Aftan á viðgerðina av ávikavist CMI kodaranum og CMI dekodaranum við Schematic, eru somu spurningar um uppbygging og roynd av teimum, men nú forritaðir við VHDL. Tað verður rátt til, at gera tvær óheftar verkætlanir í Quartus II ein til Schematic og ein til VHDL forritið.

Tað kann vera skilagott, annahvørt tað er í Schematic ella VHDL partinum av venjingini, at lata allar undireindirnar vera uppbygdar ella forritaðar í sama skjali, men í fyrstuni óheftar av hvørji aðrari. Tá er tað lættari í øðrum umfari at binda tær sama til eina samlaða skipan.

2. Kodiskipan – endamál og uppbyggjan

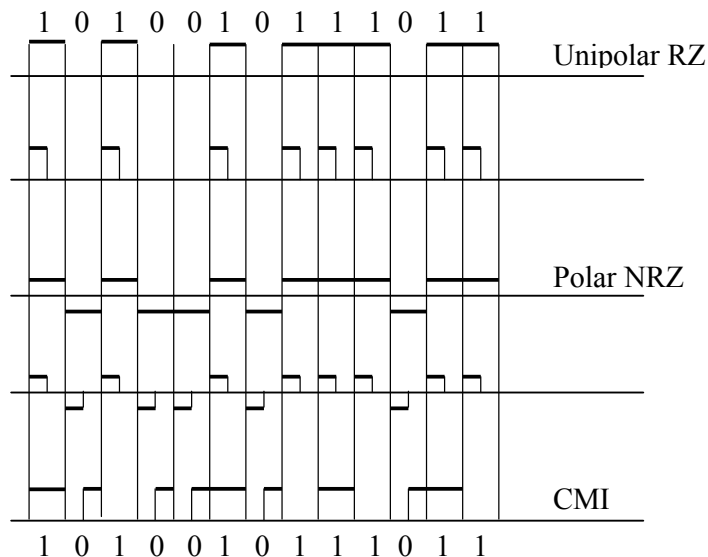
Í øllum digitalum tele- og datatransmissiónsskipanum, verða kodingsskipanir nýttar til tess at gera transmissiónina av information tryggast og best møguliga. Vanliga, tá information skal verða borin frá einum staði til annað, verður hetta gjørt við, at informationin verður gjørd um til digital signal, ið kunnu vera biner, t.e. ein røð av “0”-um og “1”-um, sum verða send hvørt aftan á annað, t.e. sekventielt (serielt), á eini transmissiónslinju.

Verður soleiðis tann bineri framburðurin (representatióin) av signalinum nýttur, býta vit tíðarásin upp í interval av ávísari longd $T=1/f$, har f verður nevndur klokkufrekvensurin. Í hvørjum av hesum intervallum senda vit eitt signalmynstur (spenningsmynstur), ið er ymiskt fyri “0” og “1”, og sigur frá, hvørt av hesum verður sent. Hesa signalmynstur-skipan nevna vit eina kodu fyri transmissiónina.

Ein tann mest einfalda, og vit kunnu kanska siga grundleggjandi, kodingin er myndað av, at spenningurin í intervallinum er 0 volt svarandi til logiskt “0” og V volt svarandi til logiskt “1”. Hesa koduna nevna vit ta bineru “unipolaru NRZ” koduna (“Non Return to Zero”). NRZ er víst í mynd 2.1 saman við øðrum kodum. Er spenningurin V volt bert í parti, vanliga helvtini av einum 1-intervalli, verður kodan nevnd “unipolar RZ” (“Return to Zero”). Er logiskt “0” myndað av spenninginum –

$V/2$, og “1” myndað av $+V/2$, fáa vit tær tilsvarandi bipolaru ella bert polaru kodurnar.

Tað er ikki altíð, at transmissiónskodurnar eru so einfaldar við tað, at tað oftast er ynskt, at tær skulu uppfylla ymsar treytir fyri, at transmissiónin skal gerast so trygg og feilfri sum gjørligt. Eginleikar í transmissiónini, ið hædd skal takast fyri, kunnu vera av ymiskum slagi. Í koparkaðalum er soleiðis ynskt, at miðalvirðið av signalspenninginum skal vera so nær null sum gjørligt av tí, at transmissión av einum javnstreymi onga nyttu ger, men bert elvir til eitt orkutap. Tí er ynskt í hesum kaðalum, at tað kodaða signalið verður borið av einum spenningi, ið í meðal er null, t.e. eins ofta er positivur sum negativur. Tí verða polarar kodur valdar til hesa transmissión. Í glasfipurkaðalum er hetta ikki nakar trupulleiki við tað, at signalið her er ljós. Harafturímóti er í øllum digitalum transmissiónsskipanum avgerandi, at tað í móttøkuendanum er møguligt at fáa fatur í einum klökkusignali, ið ger tað møguligt at markera, nær tey digitalu signalmynstrini skulu verða avlisin. Hetta klökkusignalið kann sjálvsagt verða sent í serstakari sendikanal, men er tað ein dýrur máti, sum tí ikki verður nýttur.



Mynd 2.1

Í veruligum kommunikationsskipanum verður klökkusignalið endurskapt úr sjálvum informationssignalinum, sum inniheldur ein frekvenskomposant (Fourier komposant) við sama frekvensi, sum klökkusignalið hevur, ella eitt heiltals multiplum av honum. Fyri at tryggja, at ein long røð av nullum ella eitt tølum í tí digitala signalinum ikki skal geva eitt langt tíðarbil, har kaðalsignalið, tað veri seg í koparkaðali ella ljósleiðarakaðali, er konstant, og tað tí verður torført at endurskapa klökku-signalið, verður ein ella annar formur fyri koding nýttur, sum ger, at signalið, sum sent verður, javnan skiftir við tíðini.

Í ljósleiðarasendiskipanum verður ofta tann sokallaða CMI kodan (coded mark inversion) brúkt. Hon fæst fram við at brúka hesar reglar (sí mynd 2.1):

- Logiskt “0” verður myndað í einum klökkuintervalli sum lágt signal í fyrru helvt av intervallinum og sum høgt signal í seinnu helvt av intervallinum, t.e. tað hendir eitt skift frá lágum til høgt signal í miðjuni av intervallinum. Vit

kunnu sostatt siga at “0” verður umsett til at eita “01” við einum nýggjum klökkuintervalli, ið bert hevur hálva stødd av grundklökkuintervallinum ella dataklökkuintervallinum.

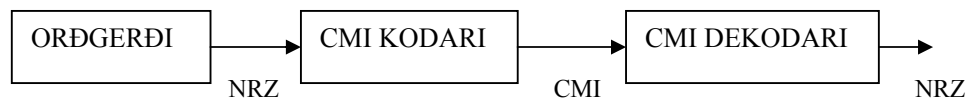
- Logiskt “1” verður myndað sum eitt støðugt signal í øllum dataklökkuintervalli-num. Hetta skiftir millum at vera lágt og høgt aðruhvørja ferð logiskt ”1” kemur fyri. Vit kunnu sostatt siga at “1” er umsett til “00” ella “11” aðruhvørja ferð tað fyríkemur.

Tí er brúk fyri at gera eitt klökkusignal við dupultum klökkufrekvensi til tess at stýra CMI koduni. Hesa klökku nevna vit CMI-klökkuna.

Í venjingini verður ein CMI kodari og ein CMI dekodari, hóskaði til brúk í einari ljósleiðaraskipan, uppbygður við forritan av VLSI rás av slagnum FPGA (ella CPLD) og royndur. Av tí at tað her snýr seg um at modulera signalið til eina ljósstrálu frá einum lasara, t.e. slökkja og tendra hann, brúka vit lágan og høgan spenning til at mynda lágt og høgt niveau í CMI koduni, sum víst í mynd 2.1

Til tess at royna skipanina er brúk fyri einum digitalum inngangssignali, ið er eitt tilvildarligt býti av “0” og “1” á tíðarásinum. Best hevði verið at havt ein tilvildarligan orðgerða ella bitgerða til hetta endamál; men til hesa venjingina byggja vit ein orðgerða við einum føstum 8 ella 16 bit orði, ið verður endurtikið periodiskt, og sum vit kunnu stilla eftir ynski.

Yvirlitsuppsetingin av skipanini sær út, sum víst í mynd 2.2. Úr orðgerðanum kemur ein røð av bittum í NRZ kodu, sum verður umgjørdur til CMI kodu í CMI kodaranum. Hetta CMI signalið er tað signalið, ið vanliga verður sent gjøgnum ljósleiðaran, sum kortini ikki er við í hesi venjingini. Í móttøkuendanum verður CMI signalið móttikið aftur, og síðan umger CMI dekodarin hesa koduna aftur til NRZ kodu.



Mynd 2.2

3. Klökkur

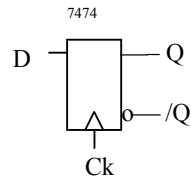
Dataklökkur er grundklökkusignalið, ið stýrir datasignalinum. Síðan hava vit eisini brúk fyri einum CMI-klökkusignali við dupult so stórum frekvensi, sum dataklökkur hevur. Eitt slíkt signal kann verða gjørt við ólineerum elektronikki út frá dataklökkusignalinum, men liggur uttan fyri evni í hesi venjingini. Vit skulu heldur ganga hinvegin og byrja við eini CMI-klökku. Við eini flip-flop skapa vit eitt dataklökkusignal við hálvum CMI-klökku frekvensi. Vit velja ein D flip-flop 7474(mynd 3.1).

At svara áðrenn møtt verður í starvsstovuni:

Spurningur 1: Ger eitt diagramm av dataklökkusignalgerðanum og vís, hvar dataklökka og CMI-klökka eru.

At svara í starvsstovuni:

Spurningur 2: Bygg dataklokkusignalgerðan upp við Quartus II Schematic og simulera hann.



D	Q	/Q
0	0	1
1	1	0

Mynd 3.1

4. Orðgerði

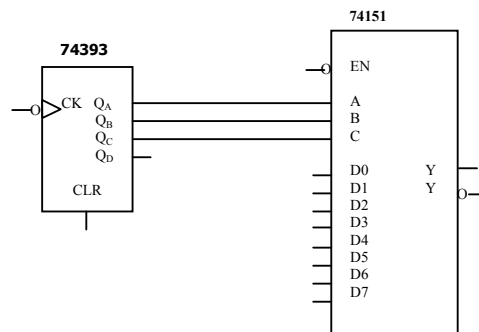
Í hesi venjing hevði verið best at brúkt ein orðgerða, ið ger tilvildarlig orð, til at royna CMI kodaran og dekodaran við. Hann er ikki tøkur. Í staðin fyri vilja vit byggja ein orðgerða, ið kann gera eina fyriskrivaða røð av bittum, ið síðan verður endurtikin periodiskt. Í byrjanini velja vit 8 bit, sum so eitt í senn við hjálp av einum multipleksara 74151 verða send út á útgangin Y. Adressuinngangirnir ABC fáa signal frá binera teljara-num 74393 sum víst á mynd 4.1. Orðið, ið verður sent út á Y-útganginum er D₀, D₁, D₂, D₃, D₄, D₅, D₆, D₇, ið er sett til logiskt "0" ella logiskt "1", svarandi til orðið, ið vit ynskja.

At svara áðrenn møtt verður í starvsstovuni:

Spurningur 3: Hvussu kunnu vit broyta orðgerðan við at brúka enable inngangin á multipleksaranum, soleiðis at 8 bit røðin, sum vit hava stillað við givnum virðum verður eftirfylgd av 8 nullum, og síðan endurtikið periodiskt.

At svara í starvsstovuni:

Spurningur 4: Set 8 bit orðgerðan upp í Quartus II Schematic. Ger broytingina sambært sp.3. Simulera orðgerðan, áðrenn og aftan á broytingina.



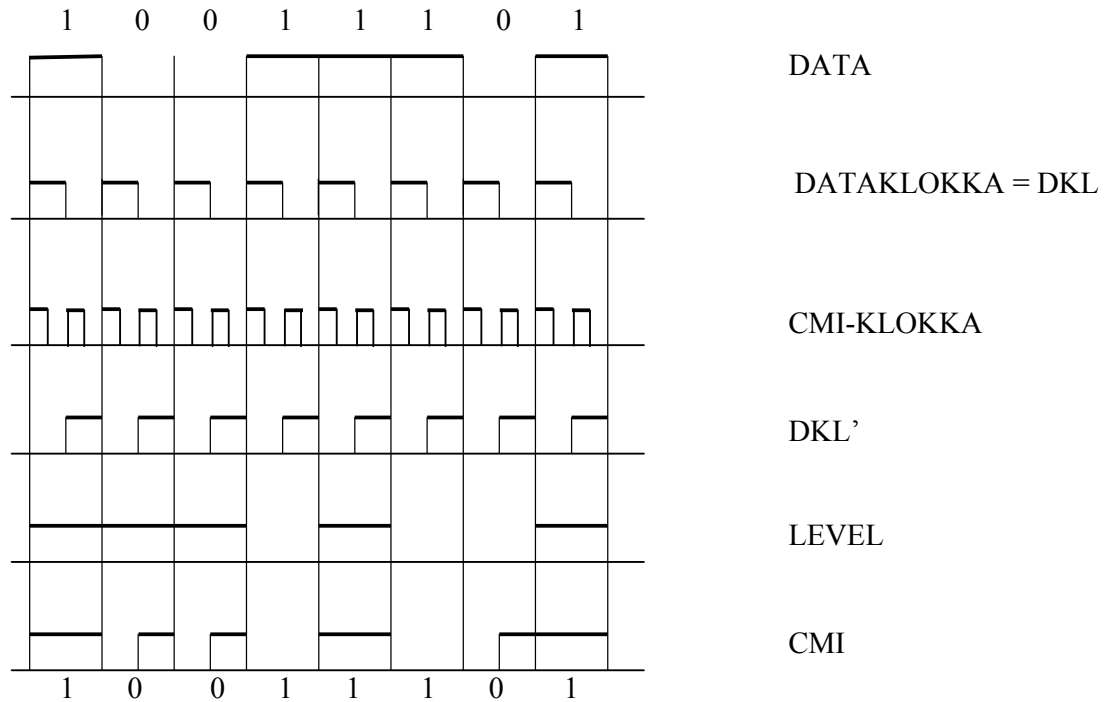
Mynd 4.1

5. CMI-kodari

Sum tað gongur fram av frágreiðingini í parti 2 og mynd 5.1, ger CMI kodarin brúk av einum klokkufrekvensi, sum er dupult so stórusum dataklokkufrekvensurin.

Nú er so spurningurin, hvørjir inngangsvariablar stýra, hvørjum virði 0 ella 1, ið CMI útgangurin skal hava í hvørjum CMI-klokkuintervalli.

Givið er, at datasignalið, ið vit nevna DATA, má vera inngangsvariabul. Vit síggja eisini, at CMI signalið, svarandi til DATA = 0, skiftir í miðjum intervalli. Hetta ger dataklokkann eisini. Tí má dataklokkusignalið “DKL” vera ein hóskandi inngangsvariabul.



Mynd 5.1

CMI signalið svarandi til DATA=1 er konstant í tveimum CMI-klokkuintervallum, men skiftir virði til 0 og 1 aðruhvørja ferð 2 CMI klokkusignal hava verið. Tí mugu vit hava ein variabul, lat okkum nevna hann LEVEL, sum stýrir CMI signalvirðinum í “1” intervallum, og skiftir virði hvørja ferð eitt DATA=1 kemur inn, men heldur virði tá DATA=0. Level er hinvegin ikki eitt útifrú komandi signal, men má verða gjørt í skipanini. Av tí at LEVEL bert broytist, um DATA hevur verið tvær ferðir 1, tá ein CMI klokkupulsur kom, má tann digitala maskinan, sum skapar LEVEL hava 4 støður svarandi til 2 støðuvariablar (2 flip-flop).

At svara áðrenn møtt verður í starvsstovuni:

Spurningur 5: Vís, at útgangssignalið $CMI = DATA' \cdot DATAKL' + DATA \cdot LEVEL$. Allar støður eru vístar á mynd 5.1.

Spurningur 6: Vís, at tann við CMI- klokkuna synkrona digitala maskinan í mynd 5.2 gevur LEVEL signalið. Hetta verður gjørt við at

- finna excitatióslíkningarnar
- skriva yvirgangs/útgangstalvu upp
- skriva støðu/útgangstalvu upp
- definera støðurnar
- tekna støðudiagramm
- vís á hvar í mynd 5.2 LEVEL signalið er.

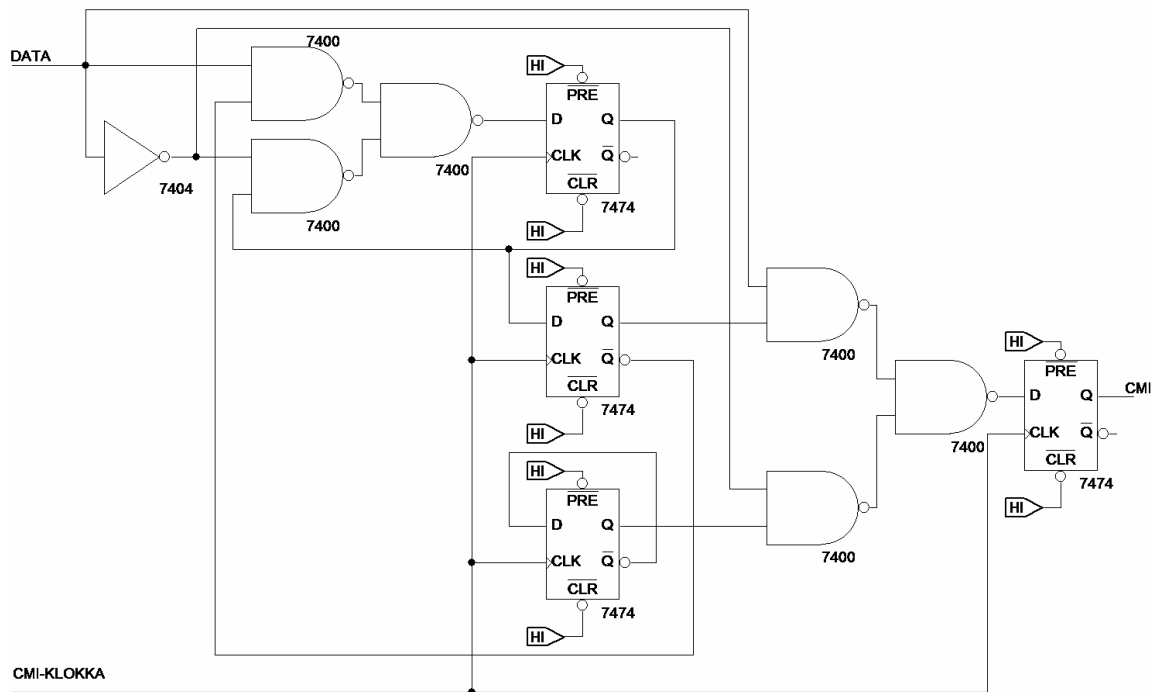
At svara í starvsstovuni um CMI-kodari við Schematic:

Spurningur 7: Bygg CMI kodaran upp í Quartus II Schematic, og simulera hann. Brúka inndata frá gjörda orðgerðanum.

Spurningur 8: Forrita FPGA chippin við gjörda Schematic forritinum

Spurningur 9: Royn og dokumentera tann gjörda Schematics forritaða CMI-kodaran

Goym gjörda Schematic forritið sum serstaka verkætlan (navngev hana t.d. CMISchematic) í Quartus II skipanini.



Mynd 5.2 CMI – kodari

At svara í starvsstovuni um CMI kodara forritaður við VHDL

Ger nýggja verkætlan í Quartus II skipanini, navngivin við t.d. CMIVHDL, og brúka nú VHDL forritan sum stöði til næstu spurningarnar.

Spurningur 10 Uppbygg klokkugerða og orðgerða í VHDL

Spurningur 11 Uppbygg CMI kodaran við Quartus II – VHDL forritan

Spurningur 12 Programmera FPGA chippin við gjörda VHDL forritinum

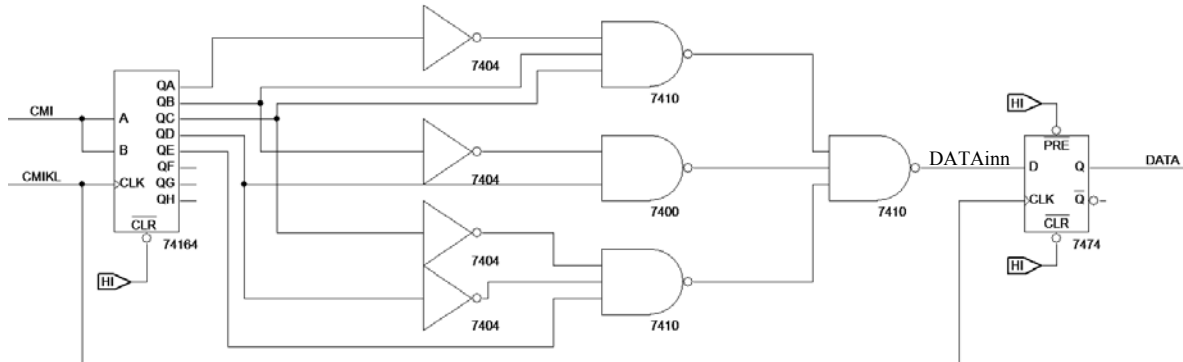
Spurningur 13 Royn og dokumentera tann gjörda VHDL forritaða CMI kodaran

6. CMI - dekodari

Ein dekodari er ein digital maskina, sum broytir CMI koduna til eitt datasignal (tað upprunaliga) við NRZ kodu sambært

CMI koda	DATA (NRZ-koda)
01	00
00	11
11	11

Í venjingini gera vit CMI dekodara við brúk av skiftiregistrum, t.d. 74164 og kombinatoriskum rásum, sum víst í mynd 6 vísir.



Mynd 6.1 CMI – dekodari bygður við skiftiregistrum

At svara áðrenn møtt verður í starvsstovuni svarandi til mynd 6.1:

Spurningur 16: Skriva sannleikatalvuna upp fyri kombinatorisku rásina DATAinn sum funktiión av QA, QB, QC, QD og QE útgangirnar í skiftiregistrinum svarandi til CMI dekodring. Vís at hendan funktiiónin kann realiserast við vístu logisku rásini í mynd 6.1. Vís at útgangsvirði DATA á CMI dekodaranum eru somu data sum inngangsvirði DATA í CMI kodaran.

At svara í starvsstovuni um CMI dekodara forritaður við Schematic:

Spurningur 17: Set rásina av CMI dekodaranum saman sum í mynd 6.1, og royn hana við Quartus II Schematic simulering

Spurningur 18 Programmera FPGA chippin við gjørda Schematics forritinum

Spurningur 19 Royn og dokumentera tann gjørda Schematics forritaða CMI dekodaran

Spurningur 20 Lat dekodaran fáa signalið frá útganginum á kodaranum, og samanber inngangssignalið til CMI kodaran við útgangssignalið á CMI dekodaranum

At svara í starvsstovuni um CMI dekodara forritaður við VHDL:

Spurningur 21 Forrita CMI dekodaran við Quartus II VHDL og royn hann við simulering.

Spurningur 22 Programmera FPGA chippin við gjørda VHDL forritinum.

Spurningur 23 Royn og dokumentera tann gjørda VHDL forritaða CMI kodaran.

Lat dekodaran fáa signalið frá útganginum á kodaranum og samanber inngangssignalið til CMI kodaran við útgangssignalið á CMI dekodaranum

7. CMI kodara – dekodara skipan

At svara í starvsstovuni:

Spurningur 24: Samanbind allar lutskipaninar og royn samlaðu skipanina við at samanbera bitsamansetingina frá orðgerðanum, og tað CMI dekodaranum gjørda signalið.

Spurningur 25: Vís við digitaloscilloskopinum, at transmissiónin gjøgnum kodara og dekodara er virkin eftir ætlan. Eru glitch at síggja í transmitteraða signalinum?

Spurningur 26: Hvørjum endamáli tækir tað at seta D-flip-floppar í útgangin av CMI kodaranum og í útgangin av CMI dekodaranum í mynd 5.2 og 6.1?

8. Starvsstovuútgørð

Til venjingina er henda starvsstovuútgørð tøk:

- Altera DE2-70 royndarfonkur við FPGA chippi, tveimum innbygdu klokkusignal-gerðum 28 MHz og 50 MHz, digitalum 0/1-signal kontaktum, ljósdiodum til lesing av digitalum signalum, LCD display og fleiri markamót til tilknýtta útgørð og mátitól.
- Digitalt oscilloskop við tveimum analogum signalrásum og 16 digitalum signalrásum.

Viðmerkjast skal, at, tá forritaði FPGA chippurin verður royndur, er hent at brúka trýstkontaktstýrt klokkusignal, tá ynskt er, at gongdin í skipanini skal vera so sein, at møguligt er at fylgja signalunum bit fyri bit við ljósdiodunum á royndarfonkinum.

Verður harafturímóti digitalt oscilloscop brúkt, kann annar av innbygdu klokkusignalgerðunum verða brúktur og signalgongdin sæst á skýggjanum.

Ein møguleiki er eisini at gera eina divisoreind við teljara, ið býtir innbygdað høga frekvensin við einum stórum tali, so at lágur frekvensur t.d. 1 Hz fæst burturúr til at brúka til klokkufrekvens.

V. PIC mikrokontrollari – menning við Proton Development Board

Ólavur Simonsen

1. Endamál

Endamálið við venjingini er, at fáa innlit í PIC microcontrollaran og síggja, hvussu microcontrollarar verða brúktir í praksis.

2. Útgerð

Til venjingina verður eitt Proton Development Board og forritið Proton Suite útflyggjað til hvønn lesandi, umframt USB kaðal, Serial til USB umskiftari og Streymforsýning 12VDC. Ein telda við Windows skal eisini brúkast til hvønn lesandi.

Proton Development Board

Hetta er eitt borð, ið hevur

ymskar elektroniskar

komponentar festar á borðið.

Kjarnan á borðinum er ein PIC16F877 microcontrollari.

Proton Suite

Proton Suite er eitt kompilara forrit. Tað vil siga, at forritið, ið skal brúkast til PIC microcontrollaran, verður forritað í Proton Suite á einum máli, ið menniskju skilja.

Síðan verður forritið umgjørt til ein hexadecimalan fil, ið PIC controllarin skilir.

Málið, ið brúkt verður í Proton Suite kallast PicBasic, og líkist sera nógv BASIC

málinum, ið er kent frá eldri teldum. PicBasic er helst tað teldumálið, ið mest líkist

enskum talumáli samanborðið við onnur teldumál. Hetta ger, at PicBasic er sera nemt og skjótt at læra.

3. Uppgávan

Uppgávan er býtt í fýra partar, har triggir teir fyrstu snúgva seg um at royna 3 ymsk forrit, ið verða handað (sí Appendices A, B og C), og ymsar broytingar verða gjørdar í. Síðan verður eitt forrit gjørt av teimum trimum soleiðis, at tey koma at virka saman.

3.1 Roynd við grafiskum LCD (GLCD)

Eitt forrit skal gerast í Proton Suite, og síðan leggjast niður í PIC16F877 microcontrollaran á Proton Development borðinum.

Displayið er grafiskt, so møguligt er at vísa grafik ella røðslumyndir. Hvør lesandi ger sítt egna grafiska eyðkenni til displayið. Sí Appendix A fyri at síggja eitt dømi.

3.2 Vís knappaborð á GLCD

Forritið skal aktivera knappaborðið á Proton borðinum soleiðis, at tøluni 1-16 verða representerað við knappaborðinum. Somuleiðis skal forritið kunna vísa á displayinum, ið situr á Proton borðinum, hvat tal verður trýst á. Sí Appendix B fyri at síggja eitt dømi.

3.3 Analog til Digital Convertering (ADC)

Á Proton borðinum er eitt potentiometur og í PIC controllaranum er ein analog til digital umskiftari.

Eitt forrit skal gerast, sum vísir virðið á spenninginum, tá skrúa verður uppá potentiometrið. Sí Appendix C fyri at síggja eitt dømi.

3.4 Samanseting av øllum

Ger eitt forrit, sum fyrst vísir grafiska eyðkennið. Eftir tað skal displayið vísa á, hvat trýst verður á knappaborðinum, men tá trýst verður á "1", verður virðið á spenninginum víst.

4. Mannagongd

Vegleiðing til uppsetan av Proton borði saman við teldu

- 4.1 Byrja við at innstillera Proton+ Suite á telduna
- 4.2 Innstillera appelsingula dongle (USB stikkið), ið er hjálagt. Minst til at innstillera frá fløgu, áðrenn dongle verður sett í. Eitt lítið skjal er hjálagt, sum greiður frá, hvussu innstillerað verður.
- 4.3 Innstillera USB to Serial Link.
- 4.4 Byrja Proton IDE, sum tú finnur í Start. Um tú brúkar Vista, skal tú brúka: Run as Administrator (Proton IDE)
- 4.5 Set GLCD í Proton borðið (Frá uttasta holinum til vinstru). Set USB to Serial kaðal millum teldu og Proton borð.
- 4.6 Set streymforsýning (12VDC) til Proton borðið.
- 4.7 Royndarkoyring av forriti: GLCD_Test.bas (C:\Program Files\ProtonIDE\Samples)
 - Lat GLCD_Test.bas upp í Proton IDE
 - Trýst á Compile and Program knappin.
 - Síðan verður biðið um at endurstarta Proton borðið. (Knappurin í miðuni á borðinum)
 - Um einki hendur. Er neyðugt at kanna, um COM portrið er rætt uppsett.
 - Trýst á Loader Read
 - Her sæst ein lítil menu við "COM 3" ella okkurt líknandi. Minst til, hvat COM stendur í menuini.
 - Fær síðan inn á Control Panel > System > Device Manager > Ports (COM & LPT)
 - Her skal I756X hava sama virði í COM portri, sum var uppgivið í Loader Read. Um ikki so høgra klikka á I756X, og vel Properties. Vel Port settings > Advanced. Set síðan COM Port number til sama nummar, sum fyrr sæð í Loader Read.

- Fær síðan aftur í Proton IDE og trýst á Compile and Program knappin. Síðan verður biðið um at endurstarta Proton borðið. (Knappurinn í miðuni á borðinum)
- Forritið skuldi nú verið downloadað og virkið í microcontrollaranum.
- Hetta sæst, um okkurt skilagott verður víst á GLCD'inum.

4.8 So skuldi verið klárt at fara í gongd við uppgávarnar.

Appendix A

PicBasic forrit til GLCD Display

GLCD_Roynd.bas

```
' Vís tekst og grafikk á einum grafiskum LCD (64x64)

    Include "PROTON_G4.INT"      ' Inkluderar uppsetan til PIC

' Dimensionera nakrar variablar

    Dim Xpos          As Byte    ' Setur Xpos sum eitt byte
    Dim Ypos          As Byte    ' Setur Ypos sum eitt byte

    Cls                    ' Ruddar LCD skermin
    Print At 0 , 2 , "Setursroynd 2009" ' Skrivar á linju 0,2
    DelayMS 3000           ' Bíðar í 3 sekund

    For Xpos = 0 To 10 ' Ein for lykkja ið telur frá 0 til 10
        Print At 0 , 2 + Xpos, " " ' Skrivar á pláss 2 til 12
        DelayMS 100              ' Bíðar í 100 ms
    Next                          ' Tekur næsta tal

    Print At 7 , 2 , "Eitt fellandi punkt" ' Skrivar á linju 7,2

Umaftur:                          ' Setur eitt merkið (Label)
    For Xpos = 0 To 63              ' Telur frá 0 til 63
        Ypos = Sin Xpos            ' Roknar sinus av Xpos
        Plot Xpos , Ypos           ' Teknar á (Xpos,Ypos) á GLCD
        DelayMS 10                ' Bíðar í 10 ms
    Next                            ' Tekur næsta tal
    For Xpos = 0 To 63              ' Telur frá 0 til 63
        Ypos = Sin Xpos            ' Roknar sinus av Xpos
        UnPlot Xpos , Ypos         ' Ruddar á (Xpos,Ypos) á GLCD
        DelayMS 10                ' Bíðar i 10 ms
    Next                            ' Tekur næsta tal
    GoTo Umaftur                   ' Far til merkið: Umaftur

    Include "FONT.INC"             ' Inkluderar font fíl til GLCD
```

Appendix B

PicBasic forrit til knappaborð

Knappabord.bas

```
' Forrit til at visa knappaborð á GLCD

    Include "PROTON_G4.INT" ' Inkluderar uppsetan til PIC

' Dimensionera nakrar variablar
    Dim Sula      As      Byte      ' Knappaborð súla
    Dim Rekkja    As      Byte      ' Knappaborð rekkja
    Dim Knappur   As      Byte      ' Knapp virði

    OPTION_REG.7 = 0              ' Loyvir PORTB "pullups"
    Cls                            ' Ruddar GLCD
    DelayMS 100                    ' Bíðar í 100 ms
    Print At 0 , 2 , "Setursroynd 2009" ' Skrivar á linju 0,2
    Print At 2 , 2 , "Tal:"        ' Skrivar á linju 2,2

Lykkja: GoSub HeintaKnapp          ' Heinta ein knapp frá knappaborði
    Print At 2, 6, #Knappur , " "   ' Vis ASCII knappanummar
    GoTo Lykkja                     ' Far til merkið: Lykkja

' Subrutina til at heinta ein knapp frá knappaborði
HeintaKnapp:
    DelayMS 50                      ' Slepp knappi, bíða

HeintaKnappUppi:                   ' Bíða til allir knappar eru uppi
    PORTB = 0                       ' Setur allar útgangir til lágt
    TRISB = %11110000 ' Setur niðastu 4 bein til útgangir, ovastu
4 bein til inngangir
    If (PORTB >> 4) <> %00001111 Then HeintaKnappUppi ' Um nakar
knappur er niðri, far aftur

    DelayMS 50                      ' Slepp knappi, bíða

HeintaTrystanKnapp:                ' Bíða til knappur verður trýstur
    For Sula = 0 To 3                ' 4 súlur á knappaborði
        PORTB = 0                    ' Setur allar útgangir til lágt
        TRISB = (DCD Sula) ^ %11111111 ' Setur ein súlu bein
to útgang
        Rekkja = PORTB >> 4          ' Les rekkju
        If Rekkja <> %00001111 Then KnappurHeintadur ' Um
knappur trýstur, far út
    Next                              ' Far víðari
    GoTo HeintaTrystanKnapp ' Eingin knappur trýstur, kanna aftur

KnappurHeintadur: ' Broyt rekkju og súlu til knappanummar 1 - 16
    Knappur = (Sula * 4) + (NCD (Rekkja ^ %00001111))
    Return                                     ' Subrutina liðug

Label:                                ' Setur eitt merkið (Label)
    End                                       ' Liðugt

    Include "FONT.INC" ' Inkluderar font fíl til GLCD
```

Appendix C

PicBasic forrit til Analog til Digital Convertering

ADC.bas

```
' Forrit til at visa Analog til Digital á GLCD

  Include "PROTON_G4.INT" ' Inkluderar uppsetan til PIC

  ADIN_RES = 10           ' 10-bit upplýsingur verður settur
  ADIN_TAD = FRC          ' RC OSC valdur
  ADIN_STIME = 50         ' Loyv 50us samplingstíð

  Dim Advirdi As Word     ' Dimensionera variabul
  TRISA = %11111111      ' Set all bein á PORTA til inngangir
  ADCON1 = %10000010     ' Set PORTA til analog
  DelayMS 500             ' Bíða í 500 ms
 Cls                      ' Rudda GLCD

Umaftur:
  Advirdi = ADIn 0        ' Les kanal 0 (Potentiometur) (0-1023)
  Advirdi = (Advirdi */ 500) >> 2
  Print At 0, 1, "Spenningur:"
  Print At 1, 1, Dec (Advirdi/100), ".", DEC2 Advirdi, " VDC"
  DelayMS 100            ' Bíða í 500 ms
  GoTo Umaftur           ' Far til merkið: Umaftur

  Include "FONT.INC"     ' Inkluderar font fíl til GLCD
```