



# **I2S-VÄYLÄLIITYNNÄN TOTEUTUS FPGA- PIIRILLE**

**Joel Junttila**

**Ohjaaja: Jukka Lahti**

# SÄHKÖTEKNIIKAN KOULUTUSOHJELMA

2016

**Junttila J. (2016) I2S-väylän toteutus FPGA-piirille.** Oulun yliopisto, sähkötekniikan koulutusohjelma. Kandidaatintyö, 16 s.

## TIIVISTELMÄ

Tutkimuksen aiheena oli tehdä toimiva I2S-väylä protokolla ja tutkia voiko piiriä luoda FPGA-piirille. Piirille tehtiin RTL-toteutus SystemVerilog kovonkuvauskielellä. Sen jälkeen piirille ajettiin FPGA-synteesi Alteran Cyclone 5 GX FPGA-piirille. Suurin kellotaajuus oli 263,5 MHz ja logiikkalohkoja käytettiin 18 kappaletta. Lopuksi tutkittiin päästiinkö tavoitteisiin.

**Avainsanat:** I2S, FPGA-piiri, SystemVerilog

**Junttila J. (2016) I2S bus protocol implementation to FPGA.** University of Oulu, Degree Programme in Electrical Engineering. Bachelor's thesis, 16 p.

## **ABSTRACT**

**The subject of the Bachelor's Thesis was to create a workable I2S bus protocol and to examine if it is possible to create the I2S circuit to FPGA. RTL-model was made to the circuit with a hardware description language SystemVerilog. After that FPGA synthesis was execute to the Altera's Cyclone 5 GX. The maximum clock frequency was 263,5 MHz and 18 logic blocks were used. Finally, the end results were examined to see if they met the objectives of this study.**

**Keywords: I2S, FPGA, SystemVerilog**

# SISÄLLYSLUETTELO

TIIVISTELMÄ.....	2
ABSTRACT.....	3
SISÄLLYSLUETTELO.....	4
LYHENTEET.....	5
1. JOHDANTO.....	6
2. I2S.....	7
2.1 I2S perusteet.....	7
2.2 Tutkimuksen suunnittelu.....	8
3. RTL-KOODAUS.....	9
3.1 SystemVerilog.....	9
3.2 RTL-koodaus.....	9
4. FPGA-SYNTEESI.....	11
4.1 FPGA-piirin perusteet.....	11
4.2 Käytettävä FPGA-piiri.....	12
4.3 Synteesi.....	13
4.4 Synteesin tulokset.....	14
5. YHTEENVETO.....	16
6. LÄHTEET.....	17

## Lyhenteet

ASM	Algorithmic State Machine, algoritmien tilakaavio
FPGA	Field-programmable gate-array, uudelleen ohjelmoitavissa oleva digitaalinen mikropiiri.
I2S	Inter-IC sound, sarjamuotoinen standardi digitaalisille audisignaaleille.
SCK	Continuous serial clock, sarjakello
WS	Word select
MSB	Most significant bit, eniten merkitsevä bitti
LSB	Less significant bit, vähiten merkitsevä bitti
RTL	Register transfer level, rekisterisiirtotaso
SoC	System of Chip, Elektroninen järjestelmä yhdellä mikropiirillä.
LUT	Look up table, Logiikkalohko, joka sisältää muistitaulukon.
ALM	Adaptive Logic Module, LUT:sta kehitetty versio.

# 1. JOHDANTO

Työn tarkoituksena on luoda I2S-väylä protokolla ja tutkia voiko piiriä toteuttaa FPGA-piirille. Piirin toiminta ja testaus tehdään Systemverilog-kovonkuvauskielellä. Piiri varmennetaan Menthor Graphics:n QuestaSim-simulaatio-ohjelmalla. Piirin synteesi tehdään Quartus 2-ohjelmalla. Synteesin jälkeen tutkitaan piirin kokoa, aika-analyysejä ja kuinka paljon I2S-väylä kuluttaa FPGA-piirin resursseja. Lopuksi tulosten pohjalta tarkastellaan kannattaako I2S-väylän toteutusta tehdä kyseisellä FPGA-piirillä.

Tutkimustani voi hyödyntää esimerkiksi valittaessa sopivaa väylä ratkaisua eri digitaalisille piireille. Tutkimukseni antaa myös yleiskuvan FPGA-piirille tehtävästä omasta piiristä aina suunnittelusta toteutukseen. Lisäksi tutkimukseni antaa empiiristä tietoa I2S-väylästä, josta löytyy tällä hetkellä melko vähän tietoa internetistä.

## 2.I2S

### 2.1 I2S perusteet

Inter-IC sound eli I2S-väylä on PHILIPS:n esittelemä väyläratkaisu digitaaliselle audio signaalille. Standardi esiteltiin vuonna 1986 ja siitä julkaistiin päivitetty versio vuonna 1996.

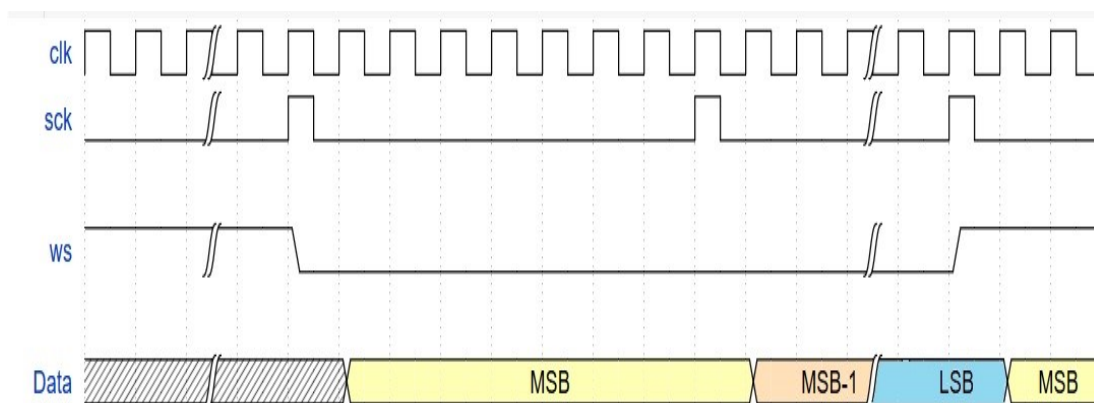
Väylän tehtävänä on toimia sarjalinkkinä digitaaliselle audiodatalle kaiutinpiirissä. Väylän sisääntulossa on datasiinaalit ja mahdolliset sallintasignaalit (enable signaalit). Ulostulossa sarjamuotoiset datasiinaalit, sisäinen kellosignaali SCK ja sananvalinta signaali WS.

Väylään saapuva data on muutettava sarjamuotoiseksi datasiinaaliksi. Data lähetetään ulos väylästä MSB bitti ensin, koska lähettimellä ja vastaanottimella voi olla eri sanapituudet.

Sisäinen kellosignaali muodostetaan ulkoisesta kellosignaalista. Sisäinen kellosignaali on muutaman kertaluvun pienempi kuin ulkoinen kello. Väylän sisäiset prosessit tapahtuvat aina sisäisen kellon nousevalla reunalla. Tämä siksi, että väylä ei tarvitse nopeaa kellosignaalia ja siten suurempia datanopeuksia voidaan käyttää myöhemmin.

Sananvalintasiinaalin WS tehtävänä on valita lähetettävä datakanava. Esimerkiksi jos WS on 1, lähetetään kanava 1 ja jos WS on 0, lähetetään kanava 2.

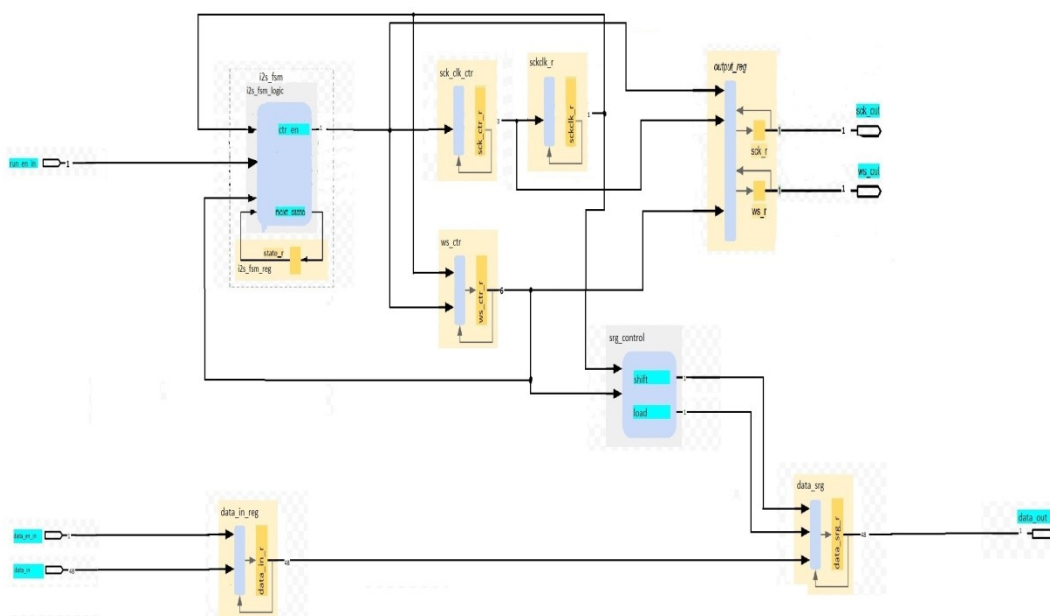
Sananvalintasiinaali vaihtaa tilaansa SCK:n nousevalla reunalla aina, kun datakanava on lähetetty. WS -signaali pitää siis tilaansa niin kauan kun yksi kanava on lähetetty. Signaalit on esitetty kuvassa 1. [1]



Kuva 1. Havainnekuva sck, ws –ja data signaaleista.

## 2.2 Tutkimuksen suunnittelu

Tutkimuksen suunnittelussa määritettiin aluksi sisääntulot ja ulostulot. Vaatimuksista selvisi, että sisääntulossa on ainakin datan sisääntulo ja ulostuloissa ovat ulkoinen kello clk, nollaus signaali asynkrooninen reset, WS, SCK ja Datan ulostulo. Loput signaalit saatiin lohkokaaviosta. Lohkokaavio on esitetty kuvassa 2. Lohkokaavion lähtökohtana oli digitaalitekniikka 3:ssä käytetty I2S-lohkokaavio, jota kehitettiin edelleen yksinkertaisemmaksi. Siitä poistettiin turhia lohkoja ja signaaleja sekä nimettiin signaaleja uudestaan. Lohkokaaviosta saatiin tilakoneen sallintesignaali run\_en\_in ja sisääntulo rekisterin sallintesignaali data\_en\_in. Lohkokaaviosta on jätetty ulkoinen kello clk ja asynkrooninen reset-signaali selkeyden vuoksi piirtämättä.



Kuva 2. I2S-väylän lohkokaaavio.

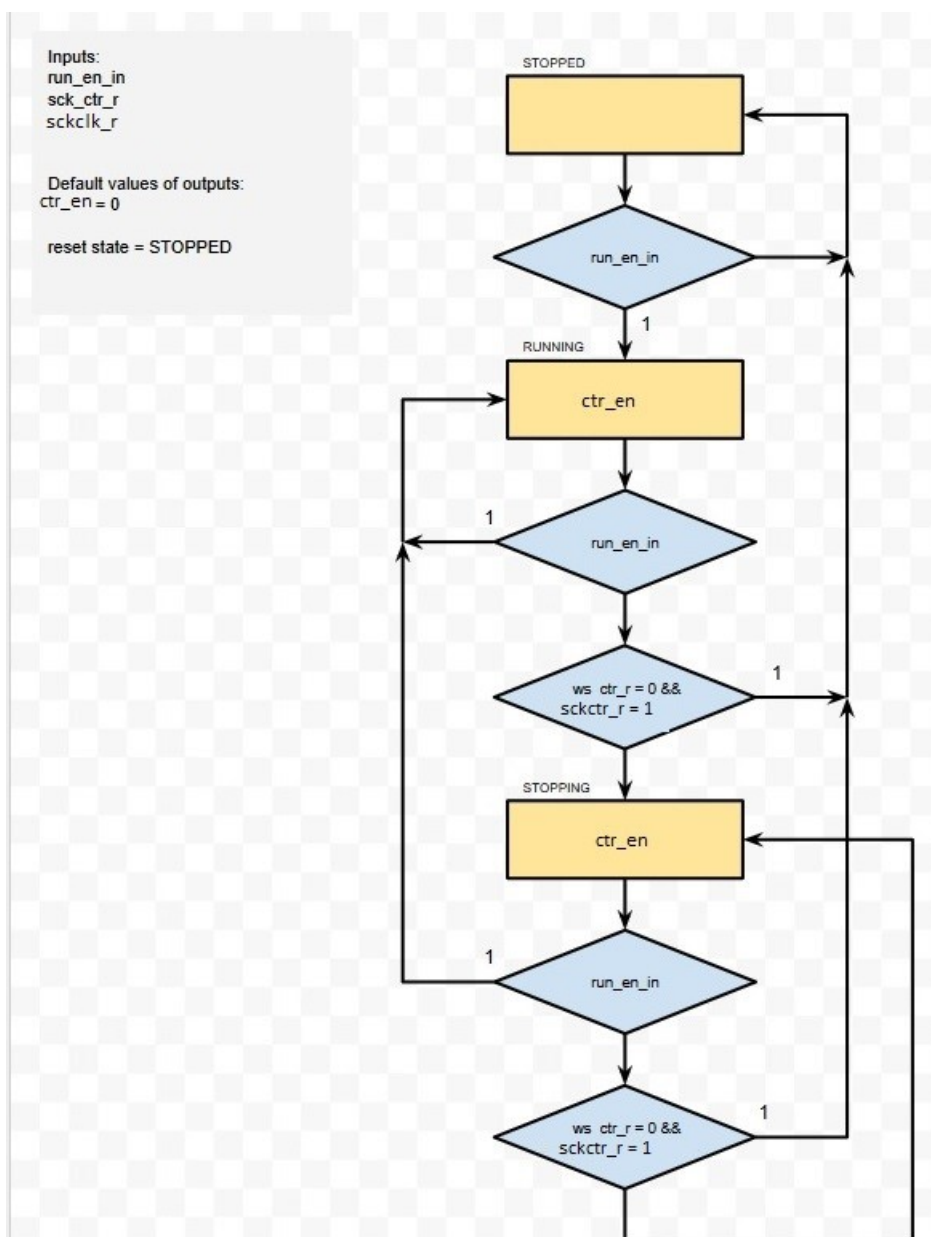
Data\_in\_reg:n tehtävänä on toimia sisääntulorekisterinä. Rekisteriin ladataan väylään tuleva data. Rekisteriä sallitaan data\_en\_in sallintesignaalilla. Data\_srg on siirtorekisteri, johon ladataan data\_in\_reg:stä tuleva signaali. Siirtorekisteriä ohjaa srg\_control logiikkalohko. Jos shift signaali on 1, dataa siirretään siirtorekistereihin. Jos load signaali on 1, dataa ladataan data\_in\_reg:stä siirtorekisteriin. Dataa siirretään siirtorekisteriin kohti MSB bittiä eli LSB bitti siirtyy ensimmäisenä siirtorekisteriin. Nämä rekisterit täytyisi tehdä kaikille kanaville, jos kanavia olisi enemmän kuin yksi.

Sck\_clk\_ctr on sisäisen kellon laskuri. Päätettiin, että SCK-signaali voisi olla seitsemäs osa ulkoisesta kellosta, joten laskuri laskee seitsemän ulkoista kellonreunaa ja palaa nolnaan. Laskuria sallitaan tilakoneelta tulevalla ctr\_en sallintesignaalilla. Sck\_clk\_ctr:n signaali menee Sckclk\_r rekisteriin, joka menee



tilaan 1 aina kun Sck\_clk\_ctr on laskenut seitsemään. Ws\_ctr on sananvalinta signaalin laskuri. Datasignaalin ajateltiin olevan 2x24 bittiä pitkä, joten WS-laskuri laskee 47 SCK-kellon sykliä ja palaa nolnaan. Tätäkin laskuria sallitaan ctr\_en signaalilla. I2S\_fsm\_logic on tilakone, joka luo laskureille sallintaginaalin ctr\_en. Ctr\_en toimii myös sallintasignaalin srg\_control-logiikkalohkolla. I2S\_fsm\_reg on tilakoneen tilamuisti. Srg\_control on kombinaatiologiikkalohko, joka ohjaa siirtorekisteriä luomalla shift ja load signaalit. Output\_reg on ulostulo rekisteri, joka luo ws\_out- ja sck\_out-signaalit. Output\_reg myös poistaa kombinaatiologiikan aiheuttamia hasardeja.

Tilakoneen ASM-kaaviossaatiin myös valmiina. Sitäkin muokattiin hieman omaan tilakoneeseen sopivaksi. ASM-kaavion tarkoitus on kuvata yksikäsitteisesti tilakoneen toimintaa. ASM-kaavio on esitetty kuvassa 3.[2]



Kuva 3. ASM-kaavio.

## 3.RTL-KOODAUS

### 3.1 Systemverilog

I2S-väylän luominen tehtiin Systemverilog kovonkuvauskielellä. Systemverilog on vuonna 2009 julkaistussa standardissa nimetty uudelleen Verilogista Systemverilogiin. Verilogin juuret ovat taas 1980-luvulla. Systemverilog on siis suhteellisen tuore kovonkuvauskieli. Kuten tutkimuksen suunnittelussa huomataan, Systemverilogilla kuvataan piiri RTL-tasolla. Siinä piirin toiminta kuvataan rekistereinä ja kombinaatiologiikkana. Systemverilogissa rekisterit kuvataan `always_ff`-prosessilla ja kombinaatologiikka `always_comb`-prosessilla.

Piirin toiminta pyritään kuvaamaan mahdollisimman hyvin rekisterisiirtotasolla ja piirille tehdään testausta varten testipenkki. Testipenkissä piirin sisääntuloihin syötetään haluttuja signaaleja. Sen jälkeen piiri simuloidaan ja katsotaan, mitä tuloksia ulostuloista tulee. Tämän pohjalta päätellään toimiiko piiri oikein.

### 3.2 RTL-koodaus

Aluksi määriteltiin sisääntulot, ulostulot, rekisterit, sallintaginaalit ja tilakoneen tilat. Koodaus päätettiin aloittaa lohkokaaavion pohjalta. Piiri on herkkä nousevalle kellonreunalle, joka näkyy `always_ff`-rakenteessa. Jokaisen rekisterin alkuun laitettiin asynkronisen `reset:n` kuvaus. Piiri koostuu `always_comb`- ja `always_ff`-lohkoista. Lopuksi piirille tehtiin testipenkki, ja piiri simuloitiin QuestaSim-simulaatio-ohjelmalla. Simuloinnin perusteella pääteltiin, että piiri toimii.

`Always_ff`-rakenteesta esitetään esimerkkinä siirtorekisterin kuvaus. Siirtorekisterin SystemVerilog-koodi on esitetty kuvassa 4. Aluksi kuvataan piirin toiminta positiivisella kellonreunalla ja asynkronisen `reset:n` toiminta negatiivisella kellonreunalla. Seuraavaksi kuvataan asynkronisen `reset:n` toiminta. Reset-signaali asettaa siirtorekisterin tilan nolaksi, jos `reset:n` tila menee nolaksi. Sitten tarkistetaan `shift-` ja `load-`signaalien tilaa `if-` ja `else if` -rakenteella. Siirtorekisterin kuvauksessa dataa otetaan ensimmäiseen eli LSB-rekisteriin ja lähdetään siirtämään kohti viimeistä eli MSB-rekisteriä. Viimeisellä `else-lauseella` ei ole merkitystä piirin toiminnan kannalta. Se lisättiin varmistamaan, että jos siirtorekisteri lohkoon tulee odottamattomia virheitä, niin siirtorekisteriin ei jää dataa sisälle.

```

// data_srg
always_ff @(posedge clk or negedge rst_n)
begin: data_srg
  if(rst_n == '0)
    data_srg_r <= '0;
  else
    begin
      if(srg_load == '1)
        data_srg_r <= data_r;
      else if (srg_shift == '1)
        begin
          data_srg_r[47:1] <= data_srg_r[46:0];
          data_srg_r[0] <= data_r[0];
        end
      else
        data_srg_r <= '0;
    end
end: data_srg

```

Kuva 4. Siirtorekisterin RTL-koodi.

Always\_comb-rakenteesta esitetään esimerkkinä siirtorekisterin kontrollilogiikka. Logiikan SystemVerilog-koodi on esitetty kuvassa 5. Rakenteessa aluksi tarkastellaan sallintasignaali ctr\_en- ja sck-kellon tiloja. Jos if-lause toteutuu, mennään tarkastelemaan ws-laskurin tilaa. Jos if-lause ei toteudu asetetaan srg\_shift- ja srg\_load-signaalit tilaan nolla. Jos ws-laskuri on tilassa nolla, asetetaan srg\_load tilaan yksi ja srg\_shift tilaan nolla. Muuten srg\_load on tilassa nolla ja srg\_shift on tilassa yksi.

```

// srg_control
always_comb
begin: srg_control
  if(ctr_en == '1 && sck_ctr_r == 6)
    begin
      if (ws_ctr_r == 0)
        begin
          srg_shift = '0;
          srg_load = '1;
        end
      else
        begin
          srg_shift = '1;
          srg_load = '0;
        end
    end
  else
    begin
      srg_shift = '0;
      srg_load = '0;
    end
end: srg_control

```

Kuva 5. Siirtorekisterin kontrollilogiikka.

## 4.FPGA-SYNTEESI

### 4.1 FPGA-piirin perusteet

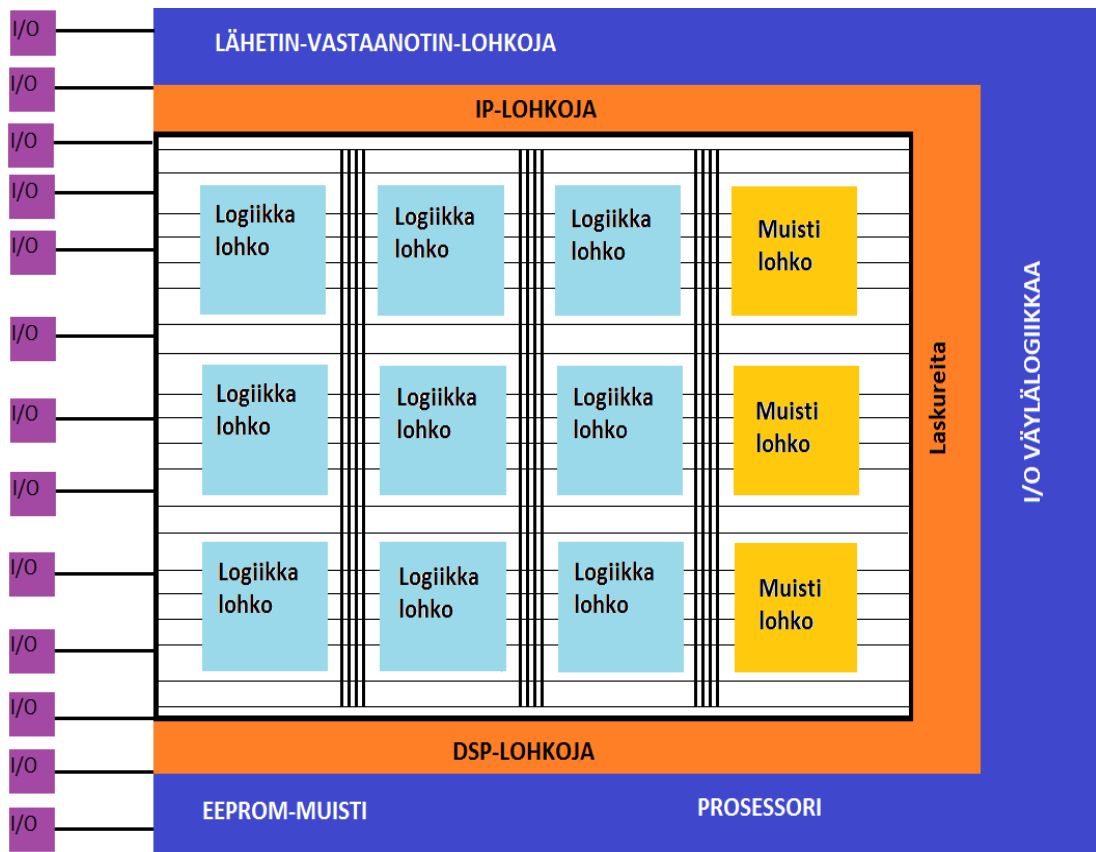
FPGA-piiri (field-programmable gate-array) on uudelleenkonfiguroitavissa oleva mikropiiri. Siten piirillä voidaan teoriassa toteuttaa mitä tahansa digitaalisia piirejä. FPGA-piiri on kehitetty 1980-luvulla. FPGA-piiri valmistajia on nykyisin kaksi isoa yritystä: Xilinx ja Altera.

FPGA-piirien toiminnallisuus koostuu ohjelmoitavissa olevista logiikkalohkoista, D-kiikuista ja niiden välisistä johdotuksista. Logiikkalohkot sisältävät muistielementin eli RAM-muistia. Logiikkalohkon sisääntulot talletetaan eräänlaiseen muistitaulukkoon ja taulukon tuloista voidaan toteuttaa esimerkiksi AND-funktio. Logiikkalohkoja kutsutaan LUT:ksi (look up table). Logiikkalohkoilla voidaan toteuttaa mikä tahansa logiikkafunktio riippuen logiikkalohkon sisääntuloista. Esimerkiksi jos logiikkalohkolla on neljä sisääntuloa, sillä voidaan toteuttaa mikä tahansa neljä-muuttujainen logiikkafunktio. Lohkojen lähdöt voidaan tallettaa edelleen muistiin D-kiikkuihin.

Monimutkaisten logiikkafunktioiden toteuttamiseen tarvitaan useita logiikkalohkoja, jotka kytketään toisiinsa johtimilla. Myös johtimet ovat uudelleenkonfiguroitavissa, jotta piiristä saadaan täysin uudelleenkonfiguroitava. Nämä logiikkalohkot on asetettu matriisi muotoon. Niiden väliin on muodostettu pysty -ja vaakasuorassa kulkevia johdotuskanavia.

Lohkojen tulot ja lähdöt on kytketty pystysuoriin johdotuskanaviin kytkimien kautta. Lisäksi pystysuorat ja vaakasuorat johtimet on kytketty toisiinsa ohjelmoitavien kytkimien kautta johtimien risteyskohdassa. Ohjelmoitavien kytkimien ohjaussignaalien arvot on talletettu muisteihin. Näin muistien sisältöä muuttamalla voidaan muodostaa haluttuja kytkentöjä. FPGA-piiri sisältää myös kello- ja reset-signaalit, jotka on valmiiksi johdotettu kaikille kiikuille.

FPGA-piirit sisältävät myös laskureita, DSP-lohkoja eli digitaalisia signaalin käsittelyä, sulautetun prosessorin, nopeaa I/O väylälogiikkaa, IP-lohkoja, logiikkaohjaimia ja sulautettua muistia. Lisäksi piirit sisältävät nopeita monen gigabitin lähetin-vastaanotinpareja, mikä mahdollistaa piirin toiminnan laajalla taajuuskaistalla esimerkiksi ethernetin käytössä. Näin ollen FPGA-piirien käyttö on erittäin laajaa. Kuvassa 6 on esitetty havainnollistava kuva FPGA-piiristä.[3][4][6]



Kuva 6. Havainnekuva FPGA-piiristä.

## 4.2 Käytettävä FPGA-piiri

Tutkimuksen FPGA-piiriksi valikoitui Alteran Cyclone 5 tuoteperheen Cyclone 5 GX. Itse piiriä ei kuitenkaan ostettu vaan kyseiselle piirille ajettiin synteesi Quartus 2 – ohjelmalla. Cyclone 5 on Alteran valmistama 28nm kokoluokan SoC-piiri. Altera julkaisi Cyclone 5 -tuoteperheen joulukuussa 2012. Cyclone on Alteran halvin ja pienitehoisin tuoteperhe. Cyclone 5 tuoteperhe on suunniteltu kuluttajalaitteiden FPGA-piiriksi. Cyclone 5 GX on puolestaan Cyclone 5 sarjan halvin versio, joka tukee 614Mbps - 3,125 Gbps:n lähetin-vastaanotin sovellusten toiminnan. [5][6][7]

Taulukko 1. Cyclone 5 GX tärkeimmät resurssit.

Sisääntulot ja ulostulot	Logiikkasolut	Muisti	Suurin kellotaajuus
268	150 000	6860 kb	550MHz

### 4.3 Synteesi

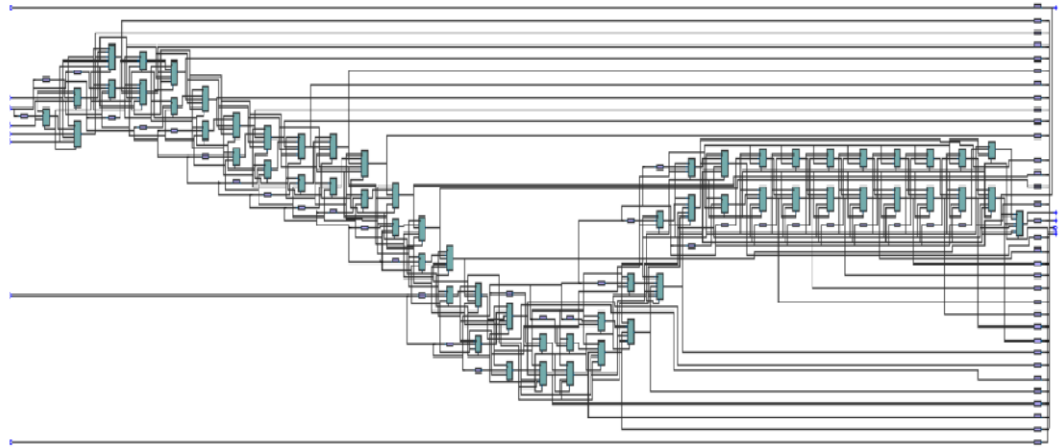
Synteessin tarkoituksena on tuoda esiin mahdolliset piirin virheet, joita RTL-simuloinnissa ei tule esille. Synteesiohjelma luo piiristä piirivalmistajan kirjastojen perusteella logiikkaporttitason kuvauksen, joka koostuu logiikkaporteista ja kiihuista. Sen pohjalta synteesiohjelma käy läpi koko piirin johdotuksen ja logiikkalohkot. Se laskee piirille ajoituksia, esimerkiksi laskee johdotuksen pisimmän viiveen. Lisäksi se kertoo kuinka paljon resursseja kyseinen piiri käyttää FPGA-piiriltä. Jos virheitä ei tullut ja piirin toiminnallisuuteen ollaan tyytyväisiä, voidaan piirille ajaa porttitason simulointi.

### 4.4 Synteessin tulokset

Piirin luominen Cyclone 5 GX FPGA-piirille onnistui. Synteesi ei havainnut yhtään virhetä piiristä, joten jatkotoimenpiteitä sen suhteen ei aiheutunut. Piirille ei asetettu resurssi- ja aikavaatimuksia, joten sekään ei aiheuttanut mitään toimenpiteitä. Pisin viive oli Sckclk\_r rekisteriltä ws\_ctr\_r:n viidennen bitin rekisteriin(0,509 ns). Synteessin luoma kellotaajuus jäi huomattavasti maksimi kellotaajuudesta, mikä johtuu muunmuassa viiveistä. Piiri ei käyttänyt kyseisen FPGA-piirille integroituja laitteita ja sovelluksia, kuten IP-lohkoja tai DSP-lohkoja. Synteesi loi myös piirille logiikkaporttitason piirikaavion, joka on esitetty kuvassa 7.

Taulukko 2. Synteessin tulokset.

Rekistereitä	109 kpl
Sisääntulo ja ulostulo pinnejä	102kpl
Käytetty kellotaajuus	263,5 MHz
Logiikkalohkot eli LUT:t	18kpl



Kuva 7. Logiikkaporttitason piiri.

## 5.YHTEENVETO

Työssä oli tarkoitus tehdä I2S-väyläprotokolla ja tutkia voiko piiriä luoda FPGA-piirille. Työ jaettiin kahteen osaan: RTL-koodaukseen ja FPGA-piiri toteutukseen. RTL-koodauksessa oli enemmän työtä, koska piiriä jouduttiin muokkaamaan vielä, kun ensimmäinen synteesi oli ajettu FPGA-piirille. Työ eteni muuten jouhevasti onnistuneen suunnitelman ansiosta.

Työn tavoitteisiin kuitenkin päästiin, koska I2S-väylälle saatiin tehtyä FPGA-piiri toteutus. Käytetty FPGA-piiri Cyclone 5 GX oli kuitenkin siinä mielessä väärä valinta, koska sen suurimpaa osaa resursseista ei käytetty. Esimerkiksi Cyclone 5 GX:n ALM lohkoja ei käytetty kuin 45kpl kun piirillä ALM-lohkoja on 56 480kpl. ALM-lohkoja on uusissa FPGA-piireissä ja ne ovat kuten LUT-lohkot, mutta niillä pystyy muodostamaan monimutkaisempia logiikkafunktioita kuin LUT:lla.

Cyclone 5 GX maksaa 179\$. Hintaan sisältyy piirikortti, jossa on eri liittimiä ja portteja kuten USB-portti. Piirikortti on valmis kaikkeen testaukseen. Pelkkien Cyclone 5 GX mikropiirien hinnat lähtevät 100\$ ylöspäin aina 400\$ asti riippuen piirien ominaisuuksista. Vertailuna piiri olisi voitu tehdä esimerkiksi Cyclone 3 FPGA-piirille, jonka eri mallien hinnat ovat 10\$ luokkaa. Tietenkin esimerkiksi Cyclone 3 FPGA-piirien ajoitusominaisuudet ovat eri luokkaa kuin Cyclone 5 FPGA-piirien.

Tutkimukselle jäi myös hieman kehitettävää. Esimerkiksi I2S-väylän testaus jäi kokonaan simulointien varaan. Kehitysehdotuksena olisi piiriä voitu testata esimerkiksi DE2-piirikortilla. Se olisi kyllä laajentanut tutkielmaa moninkertaisesti.



## 6.LÄHTEET

- [1] I2S Specification  
URL:  
[web.archive.org/web/20060702004954/http://www.semiconductors.philips.com/acrobat\\_download/various/I2SBUS.pdf](http://web.archive.org/web/20060702004954/http://www.semiconductors.philips.com/acrobat_download/various/I2SBUS.pdf)
  
- [2] Christopher R. Clare: Designing Logic Systems Using State Machines, McGraw-Hill 1973, ISBN 0-07-011120-0.
  
- [3] Digitaalitekniikan perusteet  
URL: [https://optima oulu.fi/learning/id76/bin/doc\\_show?id=216722](https://optima oulu.fi/learning/id76/bin/doc_show?id=216722)
  
- [4] Digitaalitekniikka 3  
[https://optima oulu.fi/learning/id76/bin/doc\\_show?id=235126](https://optima oulu.fi/learning/id76/bin/doc_show?id=235126)
  
- [5] Cyclone V Device Overview  
URL:  
[https://www.altera.com/content/dam/altera-www/global/en\\_US/pdfs/literature/hb/cyclone-v/cv\\_51001.pdf](https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/hb/cyclone-v/cv_51001.pdf)
  
- [6] Cyclone V Device Datasheet  
URL:  
[https://www.altera.com/en\\_US/pdfs/literature/hb/cyclone-v/cv\\_51002.pdf](https://www.altera.com/en_US/pdfs/literature/hb/cyclone-v/cv_51002.pdf)
  
- [7] In the beginning  
URL: [https://www.altera.com/solutions/technology/system-design/articles/\\_2013/in-the-beginning.html](https://www.altera.com/solutions/technology/system-design/articles/_2013/in-the-beginning.html)