

KEHITYSALUSTAN SUUNNITTELUPROSESSI

Altera MAX 7000 ohjelmoitavaan piiriin perustuva kehitysalusta

LAHDEN AMMATTIKORKEAKOULU

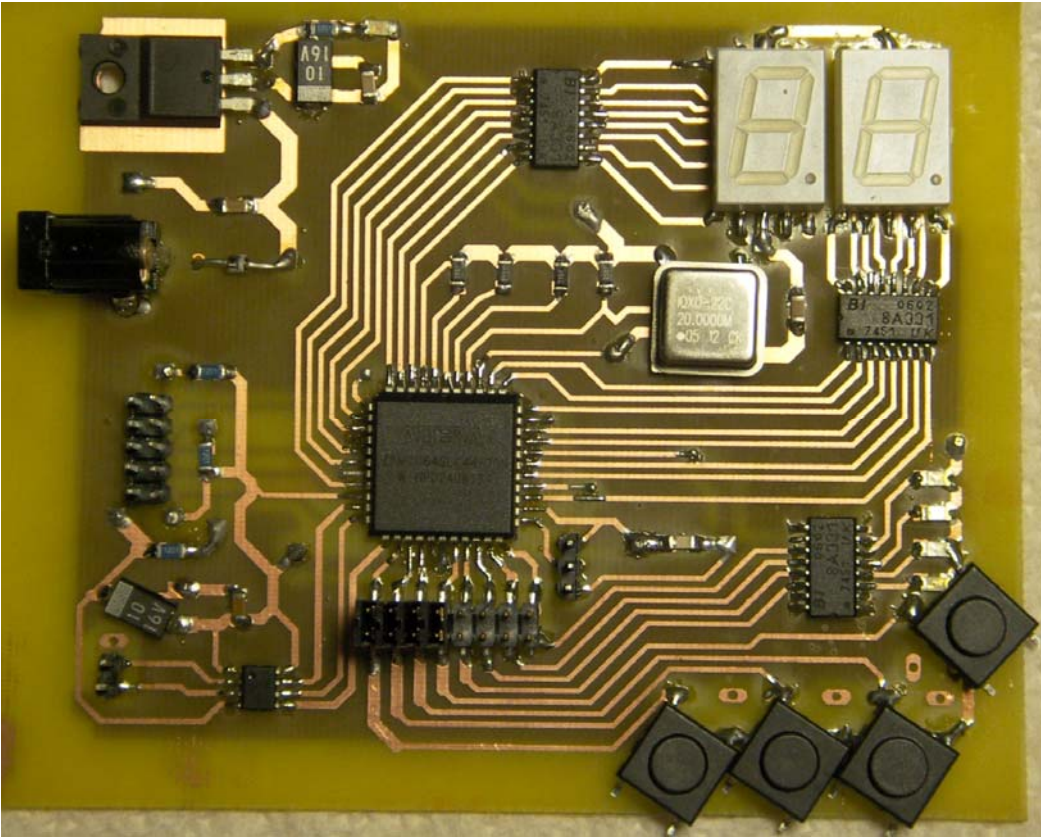
Tietotekniikan koulutusohjelma

Tietokone-elektroniikka

Opinnäytetyö

Syksy 2006

Janne Norolampi



Lahden ammattikorkeakoulu
Tietotekniikan koulutusohjelma

NOROLAMPI, JANNE: Kehitysalustan suunnitteluprosessi
Altera MAX 7000 ohjelmoitavaan piiriin perustuva
kehitysalusta

Tietokone-elektroniikan opinnäytetyö, 46 sivua, 22 liitesivua

Syksy 2006

TIIVISTELMÄ

Tämän työn tarkoituksena oli suunnitella, valmistaa sekä käyttöönottaa EPM7064S44 piiriin perustuva kehitysalusta. Piiri kuuluu Alteran MAX 7000 CPLD-piirisarjaan. Työ tehtiin Koulutuskeskus Salpauksen käyttöön, ja sitä on tarkoitus käyttää aikuisopiskelijoiden syventävissä elektroniikan opinnoissa.

Kehitysalustan lähtökohtana oli tehdä helppokäyttöinen alusta, jota oppilaiden on helppo käyttää ja jolla voidaan opettaa VHDL-kielen käyttöä Alteran omassa Quartus II kehitysympäristössä. EPM7064S44-piirille ohjelma syötetään suoraan JTAG-liitännän kautta ByteBlaster-ohjelmointikaapelilla. Kehitysalustan käyttöönottoa varten tehtiin myös opas Quartus II ohjelman käyttöönotosta sekä itse kortin käyttöönotosta (LIITE 8).

Kehitysalustan suunnittelun yhteydessä, tutustuttiin piirilevysuunnittelun teoriaan ja piirilevysuunnittelun sääntöjä noudatettiin kehitysalustan suunnittelussa. Piirilevyn suunnitteluun käytettiin CadSoftin EAGLE- piirilevynsuunnitteluohjelmaa, joka on helppokäyttöinen ohjelma myös aloittelijoille ja josta saa ilmaisen version valmistajan kotisivuilta. Kehitysalustasta tehtiin kaksi versiota, jossa jälkimmäisessä versiossa korjattiin ensimmäisen version puutteet ja virheet.

Ohjelmoitavia piirejä tutkittiin teoriaosuudessa. Siinä tutkittiin ohjelmoitavien piirien historiaa, CPLD- ja FPGA-piirien eroja sekä ohjelmoitavien piirien kytkinten toteutustapoja. Lisäksi tehtiin katsaus suurimpien valmistajien tärkeimmistä piiriperheistä ja niiden ominaisuuksista.

Valmiin kehitysalustan toimintaa testattiin VHDL-ohjelman avulla. Ohjelma toimi kehitysalustalla halutunlaisesti, joten voidaan todeta kehitysalustan suunnittelun, valmistuksen sekä itse testiohjelman onnistuneen vaatimusten mukaisesti.

Avainsanat: Ohjelmoitavat piirit, kehitysalusta, VHDL-kieli, Quartus II

Lahti University of Applied Sciences
Faculty of Technology

NOROLAMPI, JANNE: Development kit design process
Development kit based on Altera MAX 7000 pro-
grammable logic device

Bachelor's thesis in Computer Electronics, 46 pages, 22 appendices

Autumn 2006

ABSTRACT

The purpose of this project was to design, produce and take in use a development kit based on Altera EPM7064. The circuit is a part of the Altera MAX 7000 CPLD family. The development kit was made for Salpaus Further Education and it is supposed to be used in electronic studies of adult education groups.

Programmable logic devices were studied in the theoretical part. The study covered the history of programmable logic devices, differences between CPLD and FPGA circuits, and connectors of the circuit. A survey was also made on the most important circuit families of the largest manufacturers.

The starting point in creating the development kit was to make a board which would be easy to use and which could be used in teaching the VHDL language in Altera's own development environment called Quartus II. Programs are transferred to the EPM7064S44 circuit straight through a JTAG port using a ByteBlaster programming cable. A guide was also written about the use of Quartus II and the use of the card itself.

When designing the development kit, the theory of circuit board design was examined and its rules were followed in creating the development kit. CadSoft's EAGLE Layout editor was used in the design. It is easy to use and can be loaded from the creator's website without charge. A second version of the development kit was created to fix problems in the first version.

The completed development kit was tested with the VHDL program. The program worked on the development kit as desired. Based on that, we can state that designing and producing the development kit and the test program itself worked successfully.

Keywords: programmable logic device, development kit, VHDL language, Quartus II

LYHENNELUETTELO

AD	Analogy-to-Digital
AHDL	Altera Hardware Description Language
ASIC	Application Specific Integrated Circuit
CMOS	Complementary Metal Oxide Semiconductor
CPLD	Complex Programmable Logic Device
DDR	Double Data Rate
DSP	Digital Signal Processor
EAGLE	Easily Applicable Graphical Layout Editor
EEPROM	Electrically Erasable Programmable ROM
EPROM	Erasable Programmable ROM
FCRAM	Fast Cycle RAM
FPGA	Field Programmable Gate Array
IC	Integrated Circuit
IEEE	Institute of Electrical and Electronics Engineers
I/O	Input/Output
IP	Intellectual Property
ISP	In-System-Programming
JTAG	Joint Test Action Group
LUT	Look-up Table
MACO	Masked Array for Cost Optimization
MOS	Metal Oxide Semiconductor
OE	Output Enable
PHDL	Philips Hardware Description Language
RAM	Random Access Memory
ROM	Read Only Memory
SDRAM	Synchronous Dynamic RAM
SRAM	Synchronous RAM
TTL	Transistor Transistor Logic
VHDL	Very High Speed Integrated Circuit Definition Language

SISÄLLYS

1 JOHDANTO.....	1
2 OHJELMOITAVAT PIIRIT.....	2
2.1 HISTORIA	2
2.2 CPLD JA FPGA-PIIRIEN PERUSARKKITEHTUURIT	3
2.2.1 Ohjelmoitavien kytkimien toteutus tavat.....	5
2.2.2 IP-lohkot.....	7
2.3 ASIC VAI OHJELMOITAVA PIIRI	8
2.4 VALMISTAJAT.....	10
2.4.1 Xilinx.....	10
2.4.2 Altera	12
2.4.3 Lattice Semiconductor	14
2.4.4 Actel	16
2.4.5 Achronix.....	18
3. KEHITYSALUSTAN SUUNNITTELU	19
3.1 TYÖN VAATIMUKSET	19
3.2 KÄYTETYT KOMPONENTIT.....	19
3.3 PIIRILEVYSUUNNITTELU	22
3.3.1 EAGLE.....	22
3.3.2 Piirikaavio	23
3.3.3 Layout	25
3.3.4 piirilevyn valmistus.....	29
3.4 OHJELMOINTI.....	31
3.4.1 VHDL	32
3.4.2 Quartus II	33
3.4.3 ByteBlaster	34
3.5 ONGELMAT	36
4. TESTAUS JA MITTAUS	38
5. YHTEENVETO.....	41
LÄHTEET.....	42
LIITTEET	46

1 JOHDANTO

Tässä opinnäytetyössä tutkitaan ohjelmoitavia piirejä teoriassa ja käytännössä. Käytännön osuus suoritetaan rakentamalla kehitysalusta, jonka ytimenä toimii Alteran ohjelmoitava piiri. Työ tehdään koulutuskeskus Salpauksen käyttöön, ja sitä on tarkoitus käyttää aikuisopiskelijoiden syventävissä elektroniikan erikoisammattitutkintoon johtavissa opinnoissa.

Työn lähtökohtana oli tehdä kehitysalusta, jota opiskelijoiden on helppo käyttää. Vaatimuksina oli löytää Alteran valikoimasta piiri, jota pystyy ohjelmoimaan Alteran Quartus II ohjelmalla ja joka on yhteensopiva ByteBlaster-ohjelmointisovittimen kanssa, tämä tarkoitti, että piirissä piti olla JTAG-liitäntä mahdollisuus ohjelmointia varten ja sen tuli olla yhteensopiva IEEE1149.1-1990-Standardin kanssa. Kehitysalustalla on tarkoitus opettaa Quartuksen käyttöä ja VHDL-ohjelmointia.

Työssä sovelletaan insinööritutkintoon johtavien luentojen ja harjoitteluiden mukana tuomaa teoria tietoa ja käytännön osaamista. Teoriassa tutkitaan ohjelmoitavienpiirien historiaa ja niiden erilaisia kytkinten toteutustapoja. Myös piirilevyn suunnittelun teoria käydään läpi. Työtä varten opetellaan käyttämään kahta ennestään tuntematonta ohjelmaa: piirilevynsuunnitteluohjelmaa EAGLE:ä ja ohjelmointiin tarkoitettua Alteran Quartus II kehitysympäristöä.

Kehitysalustasta tehdään kaksi versiota, joista ensimmäisellä testataan kehitysalustan toimivuutta Quartuksen ja ByteBlasterin kanssa. Jälkimmäisessä versiossa korjataan ensimmäisen version puutteet ja viat sekä keskitytään enemmän piirilevynsuunnittelun sääntöjen noudattamiseen. Se tullaan kasamaan kokonaan pintaliitoskomponenteilla.

2 OHJELMOITAVAT PIIRIT

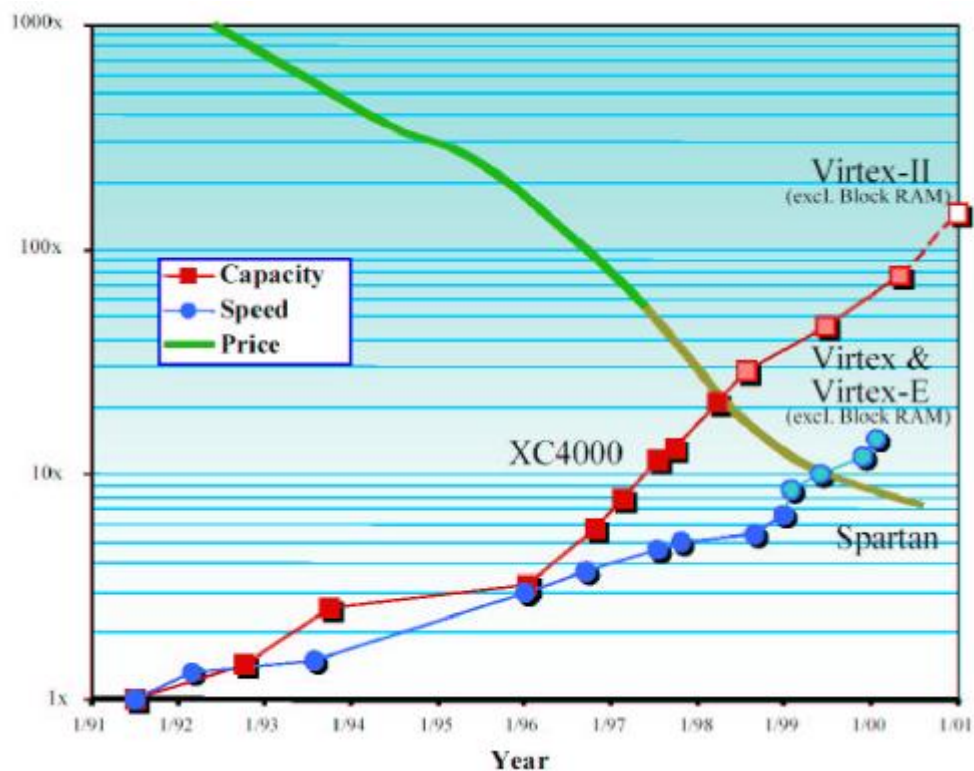
Ohjelmoitavat piirit on kehitetty irtologiikan korvaajiksi (TTL, CMOS ym.). Niiden etuihin kuuluvat mm. uudelleen ohjelmointi, (jolloin voi korjata tai muuttaa laitetta), ne vievät myös vähemmän tilaa ja ovat halvempia kuin irtologiikka sekä ovat sisäisesti häiriöttömiä. (Veijalainen 2004.)

Nykyään on jo mahdollista monissa tapauksissa korvata sovelluksessa mikro-ohjain tai jopa DSP ohjelmoitavalla piirillä. Ohjelmoitavat piirit syövät jatkuvasti markkinoita muilta piireiltä. Monitoimintaiset tuotteet yleistyvät sekä kodinelektronikassa että kännyköissä, jolloin laitteisiin tarvitaan jatkuvasti joiltain osin muutoksia ja päivityksiä. Ohjelmoitava piiri on ainoa ratkaisu, jolla tämä voidaan toteuttaa kustannustehokkaasti.

2.1 Historia

Ohjelmoitavat piirit ovat hyvin nopeasti kehittyvä mikropiiriteknologianala, 1990-luvun lopulla monet suuryritykset vaihtoivat omistajaa tai luopuivat kokonaan leikistä todettuaan teknologian liian vaikeaksi tai muuten itselleen sopimattomaksi. AMD myi Vantis-yhtiönsä Latticele, Philips myi CoolRunner-CPLD-piiriperheensä Xilinxille, Intel myi CPLD-piirinsä Alteralle ja Motorola luopui kokonaan ohjelmoitavista piireistään uusien omistajiansa myötä. Jäljelle jäi yrityksiä, jotka alun perinkin olivat keskittyneen vain ohjelmoitaviin FPGA ja CPLD- piireihin. Joitain pienyrityksiä nousi kokeilemaan onneaan alalla, lannistumatta suurten yritysten omistaja vaihdoksista. Jotkin näistä uusista yrityksistä menestyivät, toiset taas eivät, ja parhaiten menestyneet päätyivät jonkun suuren valmistajan haltuun. Suuri osa suuristakin yrityksistä valmistavat piirinsä alihankintana Siemensin tai IBM:n tehtailla, jotkut harvat valmistajat ovat perustaneet omat puolijohdetehtaat. (Wikström 1999, 91.)

Viimevuosina ohjelmoitavat piirit ovat kehittyneet jopa nopeammin kuin yleisprosessorit. Nykyään ne sisältävät jo kymmeniä miljoonia portteja ja nopeudet ovat satoja megahertsejä, kun muutama vuosi sitten niissä oli vain muutamia tuhansia portteja, ja nopeudet olivat muutamia megahertsejä. Samalla piirien hinnat porttia kohti ovat laskeneet nopeasti, mikä on mahdollistanut ohjelmoitavien piirien käytön yhä useammassa laitteissa. (Laakkonen 2003, 9.)



Kuvio 1. Ohjelmoitavien piirien kehitys (Laakkonen 2003, 9.)

2.2 CPLD ja FPGA-piirien perusarkkitehtuurit

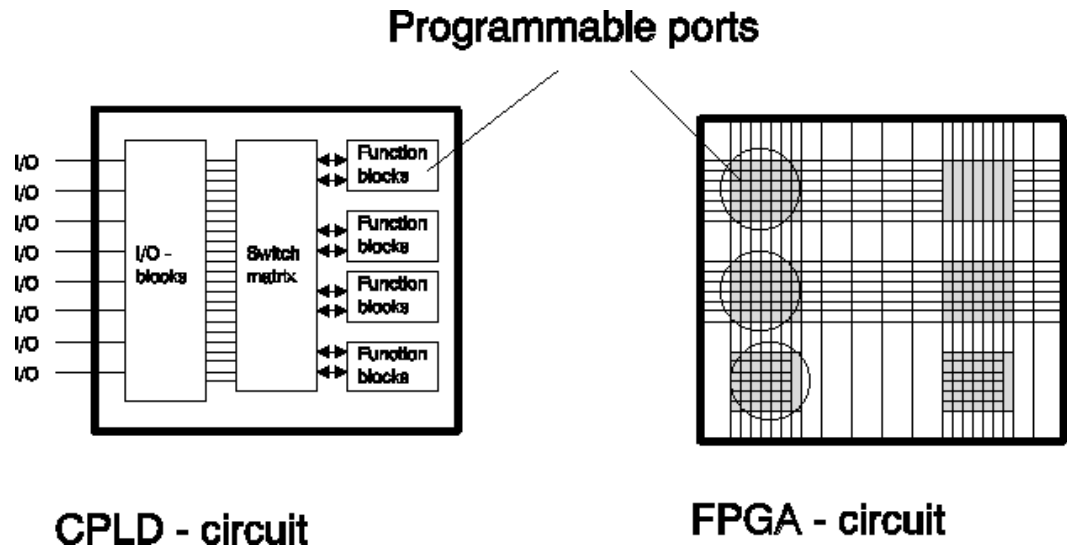
Yleisesti ohjelmoitavat piirit jaetaan kahteen pääryhmään: CPLD (Complex programmable Logic Device) ja FPGA (Field- Programmable Gate Array). Molemmat piirit muodostuvat erilaisista loogisista elementeistä ja niiden välisistä yhdysjohdoista. Loogiset elementit ovat porttipiirejä ja kiikkuja, kuten digitaalipiireissä yleensäkin. Yhdysjohdot ja loogiset elementit ovat piirissä kiinteitä, eikä niiden koostumusta tai lukumäärää voi jälkikäteen

muuttaa, joten ohjelmoitavuus onkin elementtien välisissä liitännöissä. piirissä on miljoonia kytkimiä joiden asentoa voidaan muuttaa, joten piirin peruselementeistä voidaan muodostaa lukematon määrä erilaisia kokonaisuuksia. (Wikström 1999, 91–95.)

Oikeastaan ainoa ero näiden perusarkkitehtuurien välillä on se, kuinka loogiset elementit on ryhmitelty piirin sisällä. CPLD-piirit sisältävät suuria makrosoluja, jotka on yhdistetty globaalilla reititysverkolla, kun taas FPGA-piiri sisältää paljon pieniä logiikkasoluja, jotka on yhdistetty kytkinverkolla. CPLD-piireissä on siten sisäinen rakenne enemmän etukäteen määritelty, kuin FPGA-piireissä. (Wikström 1999, 91–92; Tuominen.)

Yleensä CPLD-piirit ovat FPGA-piirejä yksinkertaisempia, josta seuraa hyvä nopeus ja helposti ennustettava käyttäytyminen, kun taas FPGA-piiri sisältää suuremman rekisterimäärän. Koska FPGA-piiri muodostuu suuresta joukosta pieniä soluja ja solut voidaan yhdistää toisiinsa lukemattomilla eri tavoilla, voidaan niillä siten toteuttaa CPLD-piirejä joustavammin eri toimintoja. FPGA-piirien joustavuudesta johtuen uusien piirien tarjonta on painottunut näihin piireihin. Tosin joustavuudella on huonotkin puolet, FPGA-piirit ovat vaikeampia suunnitella ja vaativat raskaampia työkaluja. (Wikström 1999, 91–93; Tuominen.)

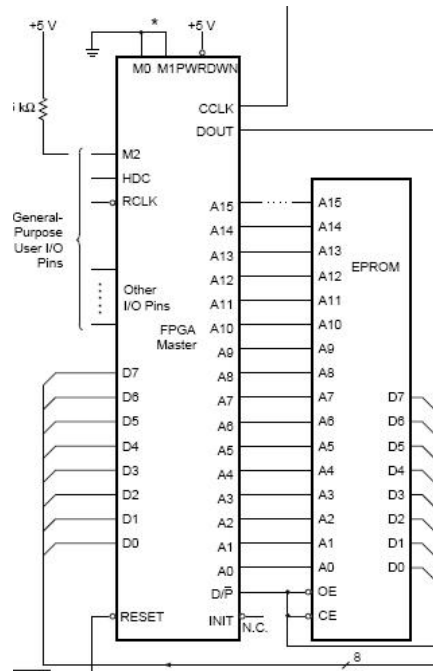
CPLD- (FPGA)-piirejä saa myös niin kutsuttuina ISP-versioisina (ISP = In System Programmable). Tämä tarkoittaa, että piiristä tulee tuoda muutama pinni liittimelle, jonka kautta voidaan komponenttiin syöttää uusi ohjelma irrottamatta itse komponenttia piirilevyiltä. (Tuominen.)



Kuvio 2. FPGA-piirit ohjelmoidaan portti kerrallaan koko alalta, kun taas CPLD-piireissä ohjelmoitavat alueet on ryhmitelty makrosoluja sisältäviksi function-blockeiksi. (Tuominen)

2.2.1 Ohjelmoitavien kytkimien toteutus tavat

Ohjelmoitavien piirien kytkimet voidaan toteuttaa usealla erilaisella menetelmällä, joilla kaikilla on hyvät ja huonot puolet. FPGA-piirit toteutetaan nykyään enimmäkseen SRAM-Pohjasina, jolloin MOS-transistoreita ohjataan SRAM-muistimoduulilla. SRAM-muisti tyhjentyy aina, kun virta katkaistaan, ja FPGA-piiri täytyy siksi ohjelmoita joka kerta uudestaan, kun se käynnistetään. Yleensä ohjelma tallennetaan EPROM-muistiin, josta se latautuu automaattisesti, kun piiri käynnistetään. Latauksen voi hoitaa myös isäntäprosessori. (Wikström 1999, 91–94.)



Kuvio 3. EPROM-muistin kytkentä FPGA-piiriin. (Xilinx3000.pdf)

Tästä on se hyöty, että piirit voidaan helposti ohjelmoida piirilevyllä, eikä tarvita piirin irrottamista ja erillisiä ohjelmointilaitteita. Toisaalta tarvitaan erillinen muistipiiri, joka vie tilaa ja maksaa rahaa. FPGA-piiri ei ole heti valmis, kun virrat on kytketty vaan vasta ohjelman latauksen jälkeen. Suurimmissa piireissä lataus voi kestää sekunninkin, joka on aivan liikaa joihinkin sovelluksiin. (Tuominen)

Toinen tyypillinen FPGA-piirin kytkinrakenne perustuu antisulakkeeseen, jossa aluksi kaikki kytkimet ovat auki jolloin antisulakkeen eriste estää virran kulun. Ohjelmoitaessa jännitepulssi läpäisee eristeen, jolloin kytkin sulkeutuu ja antisulake muuttuu johtavaksi. Kytkintä ei voida enää aukais- ta, joten tämäntyyppiset piirit ovat kertaohjelmoitavia. (Wikström 1999, 91–95.)

Hyvänä puolena on ylimääräisten muistipiirien tarpeettomuus sekä piirin pysyminen aina toimintavalmiudessa. Antisulake kytkin on myös huomattavasti pienempi kuin SRAM-pohjainen kytkin. Toisaalta taas antisulake rakenne on valmistusteknisesti kaikkein vaikeinta toteuttaa, ja niitä ei voi uudelleen ohjelmoida, jolloin tuotekehityksen prototyyppivaiheessa piirien

kulutus on melko suurta. Ohjelman täytyy myös olla toimivuudeltaan varmaa, mikä on melko vaikeaa toteuttaa. Antisulakepiirit sopivat hyvin satelliitteihin poikkeuksellisen hyvästä säteilyn kestosta johtuen ja se on niiden eräs erikoislaatuinen etu. (Wikström 1999, 91–95.)

Kolmas kytkinrakenne on EEPROM- tai Flash-transistorit, joita käytetään enimmäkseen CPLD-piireissä. Tässä menetelmässä yhdistyvät SRAM- ja antisulaketeknologian edut. Ei tarvita ulkoista muistia, kytkimet ovat pieniä, ja piiri on uudelleen ohjelmoitavissa piirilevyllä. Valitettavasti vain tämä vaihtoehto on kaikista hankalin toteuttaa ja siksi karsii valmistajia. (Wikström 1999, 91–95.)

2.2.2 IP-lohkot

Ohjelmoitavien piirien koko kasvaa kokoajan, josta seuraa että, piireille on mahdollista ohjelmoida aina vaan monimutkaisempia sovelluksia. Samalla piirien ohjelmointiin kuluva aika on kasvanut huomattavasti. Ohjelmointi on kallista ja aikaa vievää, joten avuksi on kehitetty IP-lohkot (Intellectual Property). Helpompaa olisi käyttää normaaleja mikropiirejä, mutta laitteiden koko on pienentynyt huomattavasti eikä piirilevyillä ole tilaa enää useille normaali piireille, vaan on siirrytty käyttämään suuria ohjelmoitavia piirejä (piirit eivät kuitenkaan fyysisesti ole isoja). (Salmela 1998, 20.)

Suunnitteluohjelmat ovat kehittyneet, mutta siitä huolimatta ei ole taloudellisesti järkevää suunnitella kaikkea alusta alkaen. IP-lohkoja käyttämällä voidaan tehdä uudelleenkäytettäviä lohkoja, jotka toteuttavat tietyn funktion ja sisältävät määrätyt rajapinnat. Näitä lohkoja on mahdollista yhdistellä, jolloin valmiita lohkoja käyttämällä on mahdollista toteuttaa monimutkaisia logiikkapiirejä. IP-lohkoja löytyy kaupallisina, mutta niitä on myös mahdollista tehdä itse ja koota näin oma IP-lohkokirjasto. (Salmela 1998, 20–21.)

IP-lohkoja voi hankkia useaan eri suunnittelu prosessiin, eivätkä ne yleensä ole riippuvaisia piirin valmistajasta. Lohkoja valmistavat yritykset voivatkin myydä kehittämiään lohkoja kaikille suunnittelijoille välittämättä siitä, mitä piiriä kukin käyttää. Myytävä lohko voi olla vaikka mikroprosessoriydin tai signaalinkäsittelyyn tehty yksikkö. (Salmela 1998, 20–22.)

IP-lohkojen käyttämisestä joutuu yleensä maksamaan valmistajalle ja korvaus vaihtelee tapauskohtaisesti. Valmistaja ei yleensä halua julkaista lohkojen sisäistä rakennetta, joten lohkojen kopiointi on vaikeaa. On myös olemassa vapaasti käytettäviä lohkoja, mutta näiden laatu ja vastuukysymykset ovat kyseenalaisia. Näitä lohkoja voi toki käyttää pohjana omatekemälle lohkolle. (Salmela 1998, 20–22.)

Lohkojen ostaminen ei ole ongelmaton, lohkoissa voi esiintyä pieniä virheitä tai niiden nopeus ei ole halutunlainen. Lohkojen korjaukseen voi mennä aikaa enemmän, kuin jos olisi rakentanut lohkon alusta asti itse, toisaalta toimiva lohko nopeuttaa huomattavasti suunnitteluprosessia. Lohkojen toiminnan varmistamiseksi on esim. Alteran Quartus II ohjelmassa mahdollisuus rajata, jonkun myytävän lohkon käyttö tietyksi ajaksi. Aika voidaan rajata vaikka viikkoon, ja sitten yritys tekee päätöksen ostaako lohkon vai ei. Kannattaa myös selvittää, onko ostettava lohko todellakin niin vaikea tehdä, että se kannattaa ostaa ulkopuoliselta yritykseltä, eikä tehdä sitä itse. (Salmela 1998, 20–22.)

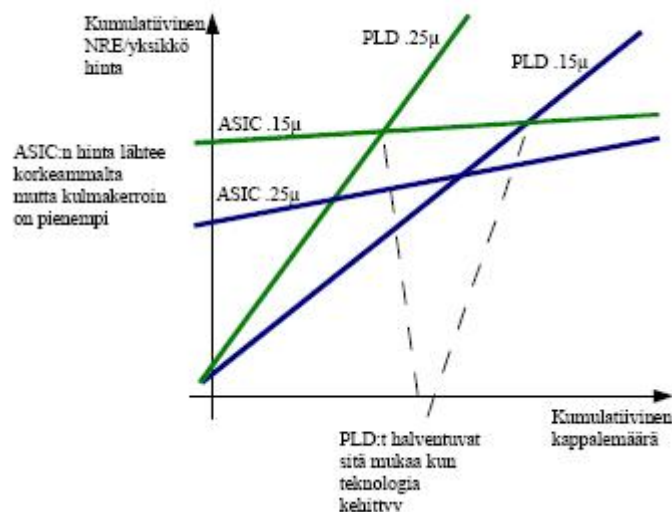
2.3 ASIC vai Ohjelmoitava piiri

Elektroniikan yleinen kehityssuunta on kohti yhden sirun järjestelmiä, mikä tarkoittaa yhä laajempaa integrointia. Tästä on seurauksena sovelluskohtaisten ja asiakaskohtaisten mikropiirien voimakkaasti lisääntyvä käyttö. Toinen kehityssuunta on kohti lyhyempiä tuotekehitysjaksoja. Tämä seikka puolestaan on johtanut ohjelmoitavien logiikkapiirien lisääntyvään käyttöön.

ASIC-piirit ovat halvin tapa toteuttaa jokin toiminto, mutta niiden haittana ovat suuret kertaluonteiset aloituskustannukset ja ensimmäisten piirien pitkä toimitusaika, ohjelmitavuuden puute, kalliit kehityskustannukset sekä hidas tuotekehitys. Hyvänä puolena ASIC-piireillä on niiden deterministinen toiminta sekä piirien halvat hinnat suurissa erissä. Ohjelmitavien piirien etu taas on se, että ne saadaan käyttöön välittömästi eikä aloituskustannuksia ole. Niissä kehityskustannukset, nopea kehitystyö sekä uudelleenohjelmitavuus. Niiden huonona puolena taas on kallis hinta. (Wikström 1998, 50.)

Ohjelmitavien-piirien hinnat ovat jatkuvasti halventuneet valmistusprosessin kehityksen myötä, jolloin piirien kokonaiskustannukset suurillakin tuotantomäärillä pärjäävät vertailussa ASIC-piireihin. (Wikström 1998, 50–51.)

Ohjelmitavia piirejä käytetään yhä suuremmissa määrin prototyyppivaiheessa ja ensimmäisissä tuotantosarjoissa. Ellei sarjan koko ole kyllin suuri kuolettamaan ASIC-piirien kiinteät kustannukset, täytyy koko tuotannon perustua ohjelmitaviin piireihin. (Laakkonen 2003, 11.)



Kuvio 4. Kustannukset ohjelmitavat piirit vs. ASIC-piirit (Laakkonen 2003, 11.)

2.4 Valmistajat

Alla käsitellään suurimpia ohjelmoitavien piirien valmistajia sekä niiden tärkeimpiä piirisarjoja. Kaikista suurin ohjelmoitavien piirien valmistaja on Xilinx, toisena tulee Altera ja kolmanneksi suurin on Lattice Semiconductor. Neljäntenä on tällä hetkellä Actel, myös muita merkittäviä valmistajia käsitellään.

2.4.1 Xilinx

Xilinx tekee pääasiassa FPGA-piirejä, mutta sen valikoimiin kuuluu myös kaksi CPLD-piirisarjaa. Xilinxin tehokkain piirisarja on Platform FPGA, josta uusin on tällä hetkellä Virtex-5 Multi-Platform FPGA.

Virtex sarjaan kuuluu Virtex E/EM, Virtex-II, Virtex-II Pro, Virtex-4 Multi ja Virtex-5 Multi piiriperheet, joista tehokkain Virtex-5 on valmistettu 65 nanometrin tekniikalla, siinä on enimmillään 330 000 ohjelmoitavaa logiikkasolua ja 1200 I/O-liitäntää. I/O liitännät tukevat DDR-siirtoa 1,2 gigabitin sekunti nopeudella. ytimen jännite on 1,0V ja I/O-porttien 1,2-3,3Voltia. Toiminnallinen kellotaajuus on 550MHz. Kehityksestä kertoo se, että suorituskyky on 30% parempi ja tehonkulutus 35% vähemmän kuin Virtex-4-piireillä, jotka julkaistiin parivuotta takaperin. (Xilinx.Virtex 2006.)



Kuvio 5. Virtex-5 Multi-Platform FPGA (Xilinx 2006 a.)

Spartan FPGA-piirisarjaan kuuluu Spartan -XL, -II, -IIE, -3L, -3 ja -3E mallit, Spartan on Virtexin halvempi versio, joka sisältää samoja ominaisuuksia, mutta ne ovat halvempia ja pienempiä. Valmistustekniikka on 90nm. Siinä on enimmillään 74 880 ohjelmoitavaa logiikkasolua ja 712 I/O-liitäntää. Portteja on max. 5 000 000. (Xilinx.Spartan 2006.)

EasyPath piirit ovat EasyPath versiot Spartan-3- ja Virtex-4- siruista. Piiri-perheen idea on yksinkertainen: FPGA-piirejä testataan asiakkaan omia vaatimuksia vastaan ja mikäli ne täyttyvät, piiri voidaan myydä jopa 80% varsinaista FPGA-piiriä halvemmalla. Piiri on kehitetty ASIC-piirien kilpailijaksi. Xilinxin mukaan ASIC-muunnoksen suunnittelu ja piille saaminen vie aikaa yli 30 viikkoa. EasyPath-versiot asiakas saa FPGA-prototyypeistään n.10viikossa, lisäksi sirulle voidaan taata sama toiminnallisuus kuin alkuperäiselle FPGA-piirille. Suunnittelu kulut jäävät perinteisistä ASIC-toteutuksista selvästi. Xilinx puhuu muutamien kymmenien tuhansien kustannuksista, kun ASIC-siruihin liikutaan usein jo miljoonatasolla. (Xilinx.Easypath 2006.)

CoolRunner CPLD-piirisarjaan kuuluu Coolrunner II ja CoolRunner XPLA3 piiri-perheet. CoolRunner on vähän virtaa kuluttava piiri, jota löytyy max. 512 makrosolun kokoisena ja kellotajuus on max. 323 MHz. piirillä on

käytetty CoolCLOCK-tekniikkaa eli piirin kiikut muuttavat tilaansa kellon molemmilla reunoilla. Jolloin piirillä käytettyjen kellojen taajuus voidaan puolittaa ja säästetään tehoa. Piirit ovat EEPROM-pohjaisia ja uudelleenohjelmoitavissa. (Xilinx.CoolRunner 2006.)

XC9500 CPLD-piirisarja on valmistettu FastFLASH-tekniikalla, jolloin piirit voidaan uudelleen ohjelmoida. Piirit ovat tarkoitettu tietoliikennesovelluksiin ja ne kuluttavat vähän tehoa. Piirejä löytyy enimmillään 288 makrosolun kokoisina ja 300MHz Nopeudella. Pin-to-pin (tiedonsiirto viive nastojen välillä) viive on ainoastaan 3.5 nanosekunttia. (Xilinx.XC9500 2006.)

Erikoispiirisarjat, Xilinx valmistaa myös useita piirisarjoja ilmailu ja avaruusteollisuuden, sekä autoteollisuuden käyttöön. Nämä sarjat ovat suunniteltu kestäämään vaikeat olosuhteet, kuten suuret lämpötilan vaihtelut ja avaruuden säteily. Piirejä on saatavilla max. 6 000 000 porttisina. (Xilinx.Aerospace and defence.)

2.4.2 Altera

Myös Alteralla on sekä FPGA-että CPLD-piirejä tuotannossa. Lähes kaikki piirisarjat ovat SRAM-pohjaisia ja tarvitsevat näin ulkoisen muistin ohjelman lataukseen. Poikkeuksen tekee MAX 7000-sarjan piirit, jotka perustuvat EEPROM-tekniikkaan. Tehokkain piirisarja tällähetkellä on Stratix II. CPLD-piirien valmistuksessa Altera on markkinajohtaja.

Stratix FPGA-piirisarjaan kuuluu Stratix, Stratix GS, Stratix II ja Stratix II GS piiriperheet, jotka on valmistettu 90 nanometrin tekniikalla, niissä on max. 179 400 ohjelmoitavaa logiikkasolua ja 1170 I/O-liitäntää. I/O liitännät tukevat DDR-siirtoa 1,5 gigabitin sekuntinopeudella. ytimen jännite on 1,0V ja I/O:n 1,2-3,3V. Toiminnallinen kellotaajuus on parhaimmillaan 653MHz. piireillä on myös max. 96 DSP-lohkoa, jotka toimivat 450 MHz

kellotaajuudella. Stratix-piiri tukee useita I/O-standardeja ja rajapintoja kuten esimerkiksi RapidIO, SPI, HyperTransport, Gigabit Ethernet. Myös paikalliset muistiväylät SDRAM, SRAM ja FCRAM ovat tuettuja. (Altera.Stratix 2006.)



Kuvio 6. Stratix II GX (Altera 2006 a.)

Cyclone FPGA-piirisarjaan kuuluu Cyclone ja Cyclone II piiriperheet. Cyclone II piirit ovat valmistettu 90nm tekniikalla ja ne on tarkoitettu kulutuselektronikkaan korvaamaan esimerkiksi DSP-prosessorit. Ne ovat hinnaltaan huomattavasti Stratix sarjan piirejä halvempia ja ne kilpailevat jo nykyään ASIC-piirien kanssa normaalissa kulutuselektronikassakin. Niissä on max. 68 410 ohjelmoitavaa logiikkasolua ja 622 I/O-liitäntää. (Altera.Cyclone 2006.)

MAX II CPLD-piireissä käytettävä uusi arkkitehtuuri perustuu 0,18 mikronin flash-prosessiin, jolloin tehonkulutus on pudonnut kymmenyksen aikaisemmasta sukupolvesta ja suorituskyky on samalla kaksinkertaistunut. MAX II piireissä on myös ensimmäisenä reaaliaikainen ISP, jolloin piiri voidaan uudelleen ohjelmoida toimintoja keskeyttämättä. Parhaimmillaan logiikkaelementtjä on 2210 ja I/O-liitäntöjä 272. Piirit toimivat 2.5, 3.3 ja 5 voltin jännitteillä. (Altera.MAX II 2006.)

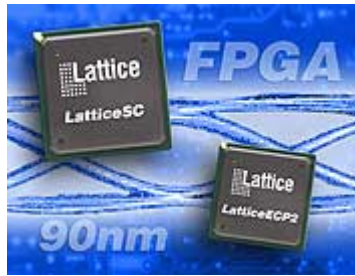
MAX 7000 CPLD-piirisarja on yksi markkinoiden nopeimmista pienistä piirisarjoista. Sarjaan kuuluvat 7000B piirit, jotka toimivat 2,5 voltin käyttö-

jännitteellä, 7000AE piirit, jotka toimivat 3,3 voltin käyttöjännitteellä ja 7000S piirit, jotka toimivat 5 voltin käyttöjännitteellä ja niitä voi muista sarjan piireistä poiketen ohjelmoida JTAG-liitännän kautta. I/O-portteja on enimmillään 212, makrosoluja 512 ja portteja 10 000kpl. Piirit perustuvat EEPROM-pohjaiseen tekniikkaan ja muistavat siten virtakatkojen jälkeen ohjelman. Piirit tukevat myös yleisempiä tiedonsiirtostandardeja. Tästä sarjasta tuli valittua piiri rakennettavaan kehitysalustaan piiri, jota käytetään oli malliltaan EPM7064SLC44-10. (Altera.MAX 7000 2006.)

2.4.3 Lattice Semiconductor

Lattice on siirtynyt suurempien kilpailijoidensa perässä valmistamaan piirejään 90nm:n tekniikalla. Lattice on väkisinkin jäänyt jälkeen pahimmista kilpailijoistaan, ja sen tehokkaimmat SC-sarjan piirit kilpailevat lähinnä Xilinxin Virtex-4:n ja Alteran Stratix II-piirien kanssa.

SC-Extreme Performance FPGA-piirisarja vastaa toiminnallisuudeltaan kilpailijoita, mutta siitä puuttuu sulautettu suoritin. Ohjelmallisensuorittimen siihen kuitenkin saa, joten sisäänrakennetun suorittimen puuttuminen ei loppupelissä ole niin hirveä asia. Tosin SC-piireille ollaan vasta kehittämässä ohjelmallista suoritinta. Uutuutena piireillä on MACO-lohko (Masked Array for Cost Optimization). MACO-lohko on ennalta määritelty ASIC-tyyppinen lohko, jossa jokin tietty toiminnallisuus voidaan puristaa kymmesosan piitilaan ja tuottaa kaksinkertaisella suorituskyvyllä, verrattuna FPGA-matriisiin. Lohkoja on maksimissaan 12 kpl. Nopeudeltaan piirit toimivat 500Mzn kellotaajuudella ja niissä on maksimissaan 115 200 ohjelmoitavaa logiikkasolua ja 972 I/O-liitäntää. I/O liitännät tukevat DDR-siirtoa 1,82 gigabitin sekuntinopeudella. (Lattice.SC-Extreme 2006.)



Kuvio 7. Latticen uudet SC ja ECP2-piirisarjan piirit (Lattice 2006 a)

ECP2 FPGA-piirisarja on Latticen uusi halpa piiriperhe. Ykkössukupolveen verrattuna piirien tiheys on kaksinkertaistunut, I/O-nopeus kaksinkertaistunut, DSP-teho kolminkertaistunut ja hinta puolittunut. LUT-yksiköitä saa 6-68 000 ja sulautettua muistia maksimissaan megabitin verran. I/O-liitäntöjä on maksimissaan 628 kpl. (Lattice.ECP2 2006.)

XP-piirisarjan piirit ovat flash-pohjaisia ja erottuvat siksi kilpailijoistaan, koska ne perustuvat SRAM-tekniikkaan. Piirit on toteutettu 0,13 mikronin tekniikalla. Nämäkin piirit ovat tarkoitettu ASIC-piirien tilalle kulutuselektronikan laitteisiin. XP-piireille on istutettu pieniä flash-muistiviipaleita, joiden ansiosta piirit voidaan käynnistää hyvinkin nopeasti. LUT-yksiköitä saa 3100-19700. I/O-liitäntöjä on enimmillään 340 kpl:ta. Piirit toimivat 1.2, 1.8, 2.5 ja 3.3 voltin käyttöjännitteillä. (Lattice.XP 2006.)

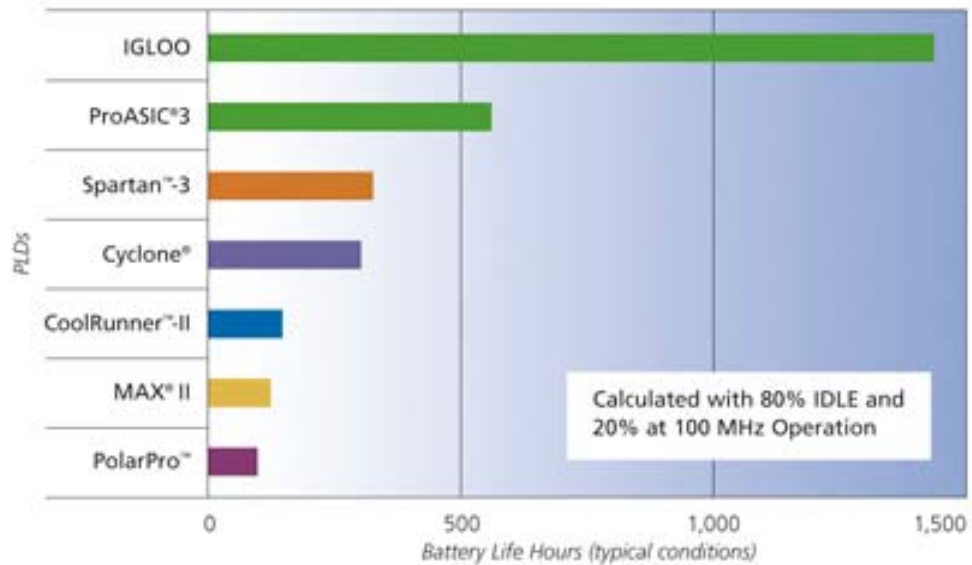
ispMACH 4000 CPLD-piirisarjaan kuuluvat B,C,V ja Z piiriperheet. B,C ja V tyyppin piirit toimivat nopealla 400MHz kellotaajuudella, kun taas vähän virtaa kuluttavat Z-sarjalaiset toimivat 267 MHz taajuudella. Piirit on toteutettu 0,18 mikronin tekniikalla. Pin to pin viive on nopeahko 3,0ns. Perheestä riippuen piirit toimivat joko 1,8v, 2,5v tai 3,3voltin jännitteillä. Makrosoluja on 32-512 kpl. I/O-liitäntöjä enimmillään 212. Piirit toimivat laajalla lämpötila alueella: -40 to +130°C. (Lattice ispMACH 4000 2006.)

ispLSI CPLD-piirisarja on vähän vanhempi Latticen piirisarja, jota käytetään vielä ainakin opetus ja ylläpitotarkoituksiin. Latticen kotisivuilta ei enää kyseistä sarjaa löytynyt, eikä myöskään ELFA:n luettelosta, joten saatavuus voi olla vaikeaa. Piiriperheitä, joita sarjaan kuuluu, ovat 1000, 2000, 3000, 5000 ja 8000 mallit. Kellotaajuus on 110–125 MHz ja portteja 2000–20000. I/O-portteja oli enimmillään 288 kpl.

2.4.4 Actel

Actel tekee ohjelmitavia logiikkapiirejä useisiin eri käyttötarkoituksiin. Suurimmat piirisarjat ovat Flash-pohjaisia, ja ne muistavat ohjelman virtakatkosten jälkeen. Pienemmät piirit taas ovat antislake-pohjaisia ja muistavat myös ohjelman, mutta niitä ei voi uudelleen ohjelmoida.

IGLOO FPGA-piirisarja on Actelin uusin, nopein ja vähän virtaa kuluttava piirisarja. Flash-pohjainen piiriperhe koostuu useasta erimallista, joiden porttien lukumäärät ovat 30 000 – 3 miljoonan välillä. Kellotaajuus on 350MHz ja I/O-portteja on enimmillään 616kpl. Uutuutena on virran kulutuksen vähentämiseksi kehitetty FlashFreeze-tekniikka, joka sammuttaa I/O-liikenteen, SRAM-muistin ja kellopiirin säilyttäen kuitenkin rekisterien tiedot. Piiri saadaan sammutettua ja käynnistettyä yhden mikrosekunnin kuluessa. Piirit on tehty kilpailemaan markkinaosuuksista ASIC- ja CPLD-piirien kanssa kannettavissa laitteissa. Sitä on kaavailtu myös älypuheliimiin, mutta piirit ovat niin uusia, että vielä niitä tai muitakaan FPGA-piirejä ei ole kännyköissä/älypuhelimissa käytetty. Piirit toimivat 1,2 / 1,5 voltin käyttöjännitteellä. RAM-muistia on enimmillään 504 kilobittiä. (Actel.IGLOO 2006.)



Kuvio 8. IGLOO-piirien virran kulutuksen vertailutaulukko (Actel 2006 a)

ProASIC FPGA-piirisarjaan kuuluvat piirit ovat myös FLASH-pohjaisia ja niitä on todella monta eri piiriperhettä. Uusin on ProASIC- 3, joka valmistetaan 13 mikronin tekniikalla, portteja on enimmillään 3 miljoonaa ja I/O-liitäntöjä 616kpl. RAM-muistia on enimmillään 504 kilobittiä. Piirit voidaan lukita niin, ettei niiltä pystytä lukemaan tietoja jälkikäteen, tätä ominaisuutta kutsutaan Flashlock-nimellä. Kaikki ProASIC sarjan-piirit toimivat 2,5 ja 3,3 voltin käyttöjännitteillä. (Actel.ProASIC 2006.)

ProASIC plus-piirit ovat valmistettu 22 mikronin tekniikalla ja niissä on enimmillään miljoona porttia, 712 I/O-liitäntää, 198 kilobittiä RAM-muistia ja ne toimivat 350 MHz:n kellotaajuudella. ProASIC piirit ovat noin puolet pienempiä ja tehottomampia, kuin uudemmat plus sarjan piirit. (Actel.ProASIC 2006.)

Axcelerator FPGA-piirisarja on antisulake-pohjainen sarja, joka on valmistettu 0,15 mikronin tekniikalla. Piirisarjan sisäinen kellotaajuus on enimmillään 500 MHz, järjestelmänopeus 350MHz, portteja on 125 000-kaksi miljoonaa ja I/O-liitäntöjä on enimmillään 684kpl. Piirillä on ohjalmapihainen 8051-suoritin, mutta "kovakoodattua" versiota ei ole tarjolla. (Actel.Axcelerator 2006.)

SX-A /SX FPGA-piirisarja on antisulake-pohjainen nopea mutta silti halpa piirisarja. Sisäinen kellotaajuus on 350MHz, silti niiden virrankulutus on suhteellisen pieni, ja pin to pin viive 3,7 ns. Portteja on 12 000-108 000 ja I/O-liitäntöjä enimmillään 360kpl. piirit toimivat 2,5, 3,3 ja 5, voltin käyttöjännitteillä. (Actel SX-A /SX 2006.)

MX FPGA-piirisarja on antisulake-pohjainen ja Actelin halvin piirisarja. Suurin nopeus on 250 MHz, portteja on 2000-36 000 kpl, käyttöjännite on 3,3 tai 5 voltia, pin to pin viive on 5,6 ns ja I/O-portteja 57-202kpl. (Actel MX 2006.)

2.4.5 Achronix

Achronix on tuntemattomampi pieni piilaaksolainen FPGA-piirien valmistaja, joka on kehittänyt 90nm tekniikalla valmistetun prototyypin, joka toimii hulpealla 1,93 gigahertsillä. Kaupallisten versioiden on tarkoitus toimia 1,6–2,2 GHz nopeudella. Tämä ULTRA-piirisarjaksi nimetty piiri on toiminut testeissä 0,2–3,9 voltin jännitteillä ja -196 - +130 asteen lämpötiloissa. (Achronix 2006)

Toinen kehitteillä oleva piirisarja on XTREME-sarja, joka toimii gigahertsin nopeudella ja kestää uskomattoman 390 asteen lämpötilan muutoksen. piirin pitäisi toimia -260 - +130 asteen lämpötiloissa. piirisarjaa kehitellään, siis todella vaativiin olosuhteisiin. (Achronix 2006)

Achronix aikoo tuoda ensimmäiset gigahertsin nopeudella toimivat piirit markkinoille ensi vuoden aikana (vuonna 2007). (Achronix 2006)

3. KEHITYSALUSTAN SUUNNITTELU

Lähtökohtana oli suunnitella ja toteuttaa helppokäyttöinen kehitysalusta, koulutuskeskus Salpauksen käyttöön. Kehitysalustaa on tarkoitus käyttää aikuisopiskelijoiden syventävissä elektroniikan erikoisammattitutkintoon johtavissa opinnoissa.

Koulutuskeskuksella ei ennestään ollut tämäntyyppistä helppokäyttöistä kehitysalustaa, joten työ tuli heille tarpeeseen. Koulutuskeskuksella ennestään olevat alustat olivat monimutkaisia ja vaikeita käyttää. Niitä ei voinut käyttää kursseilla opetukseen, koska lyhytkestoisilla kursseilla ei ollut tarpeeksi aikaa käytettävänä alustojen käytön opiskeluun.

3.1 Työn vaatimukset

Vaatimuksina oli löytää Alteran valikoimasta ohjelmoitava piiri, jota pystytään ohjelmoimaan Alteran Quartus 2-ohjelmalla joka on yhteensopiva ByteBlaster-ohjelmointikaapelin kanssa. Jotta piirin ohjelmointi onnistuisi ByteBlasteria käyttämällä, piirissä täytyi olla JTAG-liitäntä, jonka tuli olla yhteensopiva IEEE1149.1-1990-Standardin kanssa. Käytettävät komponentit sai valita vapaasti, mutta niiden täytyi löytyä ELFA:n valikoimasta. Piirilevyn suunnittelussa tuli käyttää Saksalaisen CadSoftin EAGLE-piirilevynsuunnitteluohjelmaa, josta on saatavilla ilmainen versio valmistajan kotisivuilta. Piirilevy valmistettiin Koulutuskeskus Salpauksen tiloissa syövyttämällä, ja tarkoitus oli käyttää yksipuoleista levyä.

3.2 Käytetyt komponentit

Kehitysalustasta piti tehdä helppokäyttöinen, joten FPGA-piirit hylättiin heti alussa ja keskityttiin etsimään piiriä Alteran CPLD-piirien valikoimasta. Piirissä ei tarvinnut olla älyttömiä määriä I/O-liitäntöjä, joten piirin etsintä keskittyi malliston pienimpiin malleihin. Sopiva piiri löytyi MAX 7000 piiri-

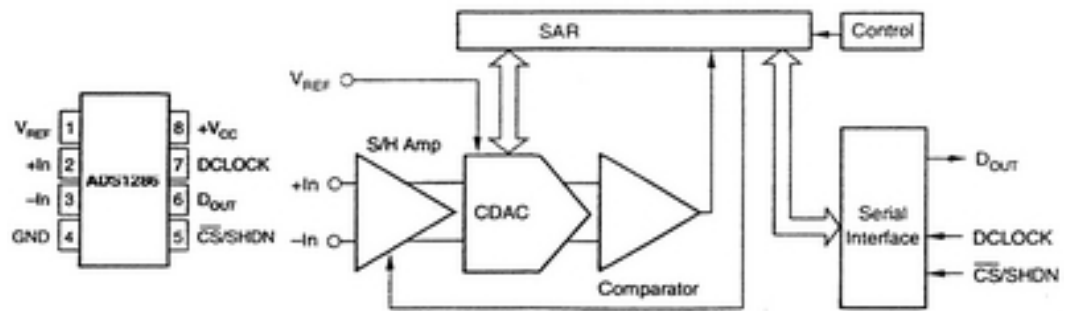
sarjasta, ja kun käyttöjännitteen haluttiin olevan 5 voltia ja piiri täytyi pysyä ohjelmoimaan JTAG-liitännän kautta, käytettävä piiri valittiin 7000S-piiriperheestä. Lopulliseksi piiriksi valittiin EPM7064SCL44-10-piiri.

EPM7064S44-piirin ominaisuudet:

- kotelo PLCC44
- 1250 porttia
- 64 macrosolua
- 36 I/O-liitäntää
- 5 ns pin-to-pin viive
- JTAG-liitäntä
- Quartus 2 tuki

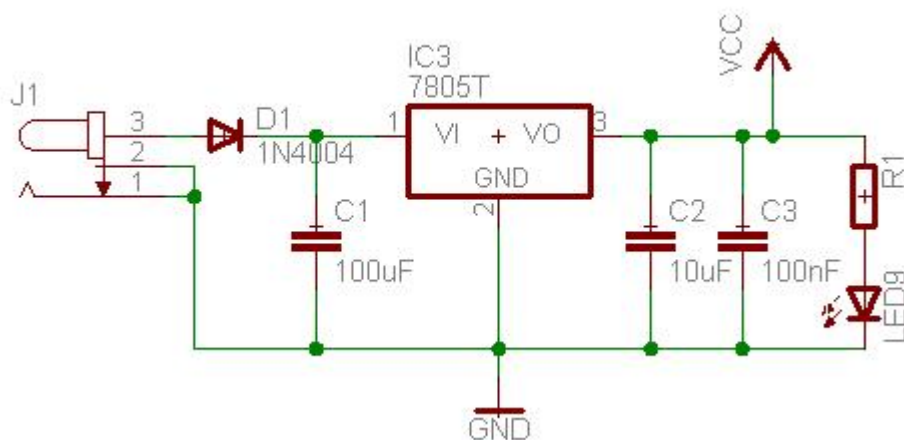
Kehitysalustasta on jätetty kaikki ylimääräinen pois, jotta se pysyy helppokäyttöisenä, ja siinä on vain perusohjelmoinnin harjoitteluun tarvittavat komponentit, jotka kuitenkin antavat tarpeeksi vaihtoehtoja ohjelmoijalle:

- 2kpl 7-segmenttinäytöjä
- 8 kpl 3mm punaisia ledejä
- rinnankytketty 8-bittinen AD-muunnin 1286P



Kuvio 9. Vähän virtaa kuluttavan 1286P AD-muuntimen sisuskalut

Piirilevyllä tehtiin oma virtalähde 5 voltin 7805T regulaattorilla, näin virran syöttö kehitysalustalle hoituu normaalilla 2,5mm pistokkeella varustetulla 6-9 voltin virtalähteellä. kytkentään laitettiin myös estodiodi 1N4005, estämään oikosulun tapahtumista, jos virtapistoke kytketään väärinpäin. 5mm punainen led-valo kertoo kun piirilevyllä tulee virtaa. Suoto- ja häiriöpoistokondensaattorit nähdään kuvioista (10).



Kuvio 10. 5 voltin virtalähteen kytkentä

Oskillaattoriksi valittiin C-MAC:n valikoimasta 20MHz kellotaajuudella toimiva IQXO-22-oskillaattori, joka toimii 5 voltin käyttöjännitteellä ja on 8-pinnisessä metallisessa DIL-kotelossa.

Muita peruskomponentteja piirilevylle valittiin seuraavat:

- 2kpl 8-vastuksen (330 ohm) vastusverkoja 898-3
- kanta EPM-7064S44-piirille
- virtapistoke PCB 2,5mm
- piikkirimaa, vastuksia ja kondensaattoreita

Kehitysalustan versio kakkosessa käytettiin samoja komponentteja kuin ensimmäisessäkin, tosin ne olivat kaikki pintaliitosmalleja, lukuun ottamatta itse EPM7064-piiriä, jota ei enää ollut saatavilla pintaliitoskomponenttina. EPM7064-piiri on PLCC-kotelossa, ja se kiinnitettiin piirilevylle ilman kantaa, jotta vedot pysyisivät levyn yläpuolella. Ensimmäisestä versiosta kiinteät kytkimet jäivät pois, ja niitä haluttiin kakkosversioon. Piirissä ei ollut enää vapaita I/O-liitäntöjä, joten neljä Lediä päätettiin korvata kytkimillä.

3.3 Piirilevysuunnittelu

Komponenttien valitsemisen jälkeen, ruvettiin suunnittelemaan Kehitysalustalle piirilevyä. Piirilevyn suunnittelussa käytettiin CadSoftin EAGLE-piirilevynsuunnitteluohjelmaa.

3.3.1 EAGLE

EAGLE (*Easily Applicable Graphical Layout Editor*) on nimensä mukainen helppokäyttöinen graafinen suunnitteluohjelma. Ohjelman käyttöönotto on suhteellisen helppoa, jos on aiemmin käyttänyt jotain piirilevynsuunnitteluohjelmaa. Tosin uuden ohjelman tutustumiseen menee aina aikaa ja joitain toimintoja ei meinannut millään löytyä ohjelman käyttöliittymästä.

Ohjelmasta on saatavilla ilmainen versio opiskelu- ja harrastekäyttöön. Ilmaisen version rajoituksia:

- Piirilevyn koko on rajattu 80 x 100mm
- Piirilevyn kerrosten lukumäärä on rajattu kahteen kerrokseen
- Vain yksisivuiset piirikaaviot ovat mahdollisia

EAGLE koostuu seuraavista osista:

1. Control Panel ,ohjauspaneeli, ohjelmien käynnistys
2. Schematic Editor, piirikaavion piirto
3. Board Editor, Layout, piirilevyn suunnittelu
4. Autorouter, piirtää komponenttien väliset kytkennät
5. CAM Processor, gerber-tiedostojen luonti

Eaglen käyttäjät eivät ole sidottuja käyttämään Windows-pohjaisia työasemia, sillä Eagle:stä löytyy versiot myös Linuxille ja Macintoshille. (Cadsoft 2006.)

3.3.2 Piirikaavio

Piirikaaviossa määrätään komponenttien väliset kytkennät, siinä näkyy myös piirejä sekä muita komponentteja esittävät piirrosmerkit. Siitä käyvät esille myös signaalien, komponenttien ja liitinten tunnuksat.

Aluksi on hyvä etsiä kaikki tarvittavat komponentit suunnitteluohjelman komponenttikirjastosta sekä tehdä ne komponentit itse, joita ei löydy val-

miina kirjastosta. Omat komponentit tehdään suunnitteluohjelmasta löytyvällä komponenttieditorilla. Jos komponentteja ei ole vielä tilattu, eikä jostain tiettyä komponenttia löydy kirjastosta, se kannattaa vaihtaa vastaavaan tuotteeseen, joka löytyy kirjastosta. Varsinkin suurimpien komponenttien luominen itse, on aloittelijalle todella aikaa vievää puuhaa. Tässä työssä A/D-muunnin vaihdettiin toisen merkkiseen, kun alkuperäisen suunnitelman mukaista muunninta, ei ollut saatavilla EAGLE:n kirjastossa. EAGLE:n komponenttikirjasto vaikuttaa aluksi todella sekavalta, ja tarvittavat komponentit saattavat löytyä mitä erikoisempien lyhenteiden takaa.

Kun tarvittavat komponentit on löydetty ja laitettu työpöydälle, niin ne järjestellään työpöydälle siten, että signaalit kulkevat vasemmalta oikealle sekä ylhäältä alaspäin. Komponenttien väliset kytkennät voidaan aloittaa heti, kun komponentit on aseteltu oikeille paikoilleen. Toki komponentteja on mahdollista liikutella myös kytkentöjen piirtämisen jälkeen, mutta se on työläämpää.

Työ kannattaa yleensä jakaa toiminnallisiin kokonaisuuksiin eri sivuille, joka selkeyttää kokonaisuutta. Eaglen ilmaisversiossa ei valitettavasti ollut mahdollista käyttää kuin yksi sivuista piirikaaviota, joten virtalähde on samalla sivulla piirretty vasempaan yläkulmaan, kun se olisi ollut parempi olla omalla sivullaan (LITE 1). Toisaalta virtalähde on niin yksinkertainen, että se voi hyvinkin sijaita pääpiirustuksen ohessa yhdellä sivulla.

Viimeistään tässä vaiheessa kannattaa tutkia komponenttivalmistajien datasivuja, jotta kaikki suodatuskondensaattorit, sekä ylös- ja alasvetovastukset tulee kerralla oikeille paikoille. Niiden lisääminen on myöhemminkin mahdollista, mutta huomattavasti vaikeampaa. Yleensä käyttöjännitteeseen piireille voidaan käyttää suodatuskondensaattoreita arvoltaan 10-100nF. Tässä työssä IC-piireille on laitettu 100nF suodatus kondensaattorit mahdollisimman lähelle piirin maa-pinniä. Virtalähteessä on jo aikaisemmin mainitut kuvion (10) mukaiset häiriönpoistokondensaattorit.

3.3.3 Layout

Varsinainen piirilevy suunnitellaan tässä osiossa. Aluksi piirretään piirilevyn rajat, EAGLE:n ilmaisversiossa ne ovat automaattisesti 80x100mm, jota voi tarvittaessa myös pienentää. Tässä työssä rajoja olisi kyllä tarvinnut suurentaa, jotta olisi voinut tehdä halutunlaisen yksipuoleisen piirilevyn. On myös tiedettävä kuinka monikerroksista piirilevyä ollaan valmistamassa sekä piirilevyn valmistustekniikka. Tässä työssä käytettiin siis lopulta kaksikerroslevyä, kun johtimien veto ei onnistunut yksikerroslevylle, tilan puutteen takia.

Komponentti sijoittelussa on otettava monta tekijää huomioon, siksi parhaimpaan mahdolliseen tulokseen on lähes mahdoton päästä. Komponenttisijoittelulla pyritään siihen, että sijoitettavat komponentit sijoitetaan tiettyihin lohkoihin ja että vedot näiden välillä olisivat mahdollisimman lyhyitä. Komponentit näkyvät suunnitteluohjelmassa tässä vaiheessa olevan toisissaan kiinni "kuminauha" periaatteella.

Komponentit sijoitetaan siten, että digitaali- ja analogiakomponentit ovat erikseen, kuumenevat osat sijoitetaan ylös, ja niille varataan tilaa jäähdytykselle. Trimmerit ym. säädettävät komponentit sijoitetaan niin, että niitä voidaan säätää ja säädettävät piirit sijoitetaan niiden lähelle. Merkkivalot, kytkimet ja liittimet sijoitetaan niin, että ne näkee ja niitä voidaan käyttää.

Johdotus vaiheessa, "kuminauha" -johtimet, yhdistetään oikeiksi johtimiksi seuraavassa järjestyksessä:

1. Maa ja käyttöjännite
2. Kriittiset signaalivedot
3. Lyhyet, helposti toteutettavat vedot
4. Muut vedot

Maa ja käyttöjännite vedot pyritään vetämään käsin, ja näissä on syytä käyttää tarpeeksi leveää johdinleveyttä. Maa ja käyttöjännite on myös pyrittävä täydentämään silmukoiksi piirillä, jotta häiriöt pysyisivät kurissa eikä pääsisi tapahtumaan ylikuulumista. Tämän työn ensimmäisessä versiossa kaikki johdinleveydet olivat 8mils eli n. 0.2mm, joka on riittävä, mutta piirilevyn valmistus on vaikeaa syövytysmenetelmällä. Valmiista piirilevystä olikin useampi johdin poikki. Tämäkin tultiin korjaamaan jälkimmäisessä versiossa.

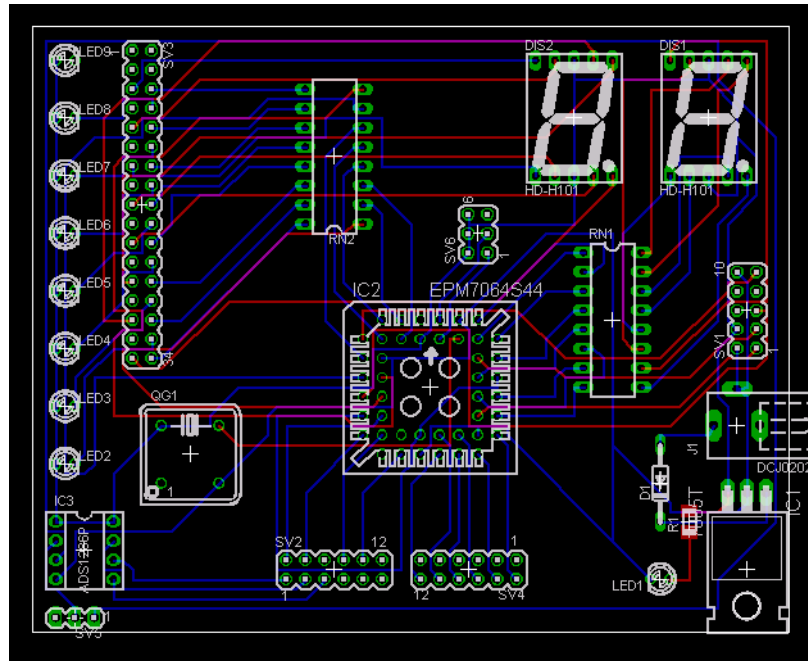
Kriittiset signaalit täytyy myös vetää käsin ja pyrkiä mahdollisimman lyhyisiin vetoihin. Kaveriksi kriittiselle signaalijohtimelle kannattaa vetää maa-johdin, joka seuraa kriittistä signaalijohdinta. On myös pyrittävä välttämään yleistä vetosuuntaa kuten väylää. Häiriöitä voidaan suodattaa pois terminoinnilla, eli ylös- ja alavetovastuksilla ja suodatuskondensaattoreilla.

Loput vedot voidaan vetää automaattisella johdotustyökalulla (Autorouter), joka laskee nopeasti tarvittavat reititysvaihtoehdot ja säästää siten todella paljon suunnittelijan aikaa. Toisaalta kone ei välttämättä osaa ottaa huomioon kaikkia tarvittavia asioita, ja reitityksen tulosta joutuu siistimään lopuksi mieleisekseen. Tietokone ei korvaa osaamista, joten on mielekästä käyttää autoreititintä vasta sitten, kun on ensin oppinut tekemään käsin hyvää jälkeä.

Kehitysalustan ensimmäisen version vedot vedettiin kokonaisuudessaan autoreitittimellä, ja kuten kuviosta(11) nähdään, kone ei onnistunut siinä kovinkaan mallikkaasti. Vaikka komponenttien optimaalista paikkaa etsittiin monta kertaa, ovat vedot täysin sekaisin ja johdinten reitityksestä johtuen niiden leveyttä ei voinut muuttaa. Johdinten täytyi mahtua komponenttien jalkojen välistä, ja siksi niiden leveys oli vain 8 milssiä. Komponentit on sijoitettu jokseenkin niin kuin pitääkin, tosin johdotusten onnistuminen oli etusijalla tässä versiossa.

Virtalähteen komponentit ovat sijoitettu oikeaan alakulmaan yhteen ryp-pääseen. Niiden yläpuolella lähellä levyn reunaa on latausliitin. Analogiset

komponentit on pyritty sijoittamaan mahdollisimman kauaksi digitaalisista piireistä (EPM7000, Oskillaattori ja A/D-muunnin). Liitteestä (2) nähdään molempien kerrosten johtimet sekä niiden epäjärjestys.



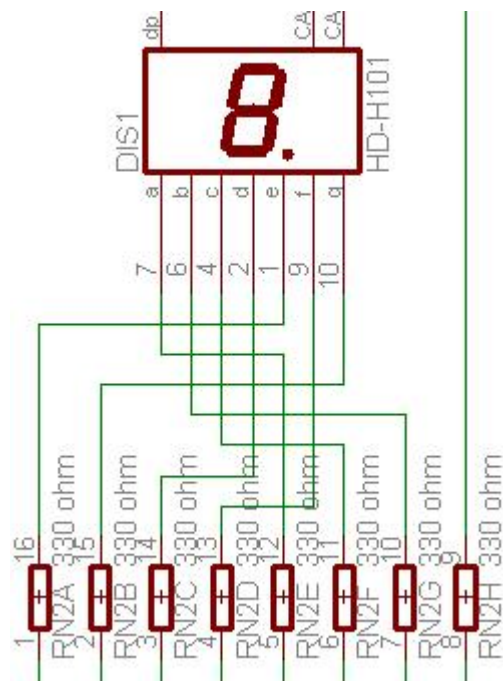
Kuvio 11. Autoreitittimen vetämät signaalit

Versio kakkosessa, joka on tarkoitettu lopulliseksi tuotteeksi, on otettu huomioon paremmin lähes kaikki asiat kuin ensimmäisessä versiossa.

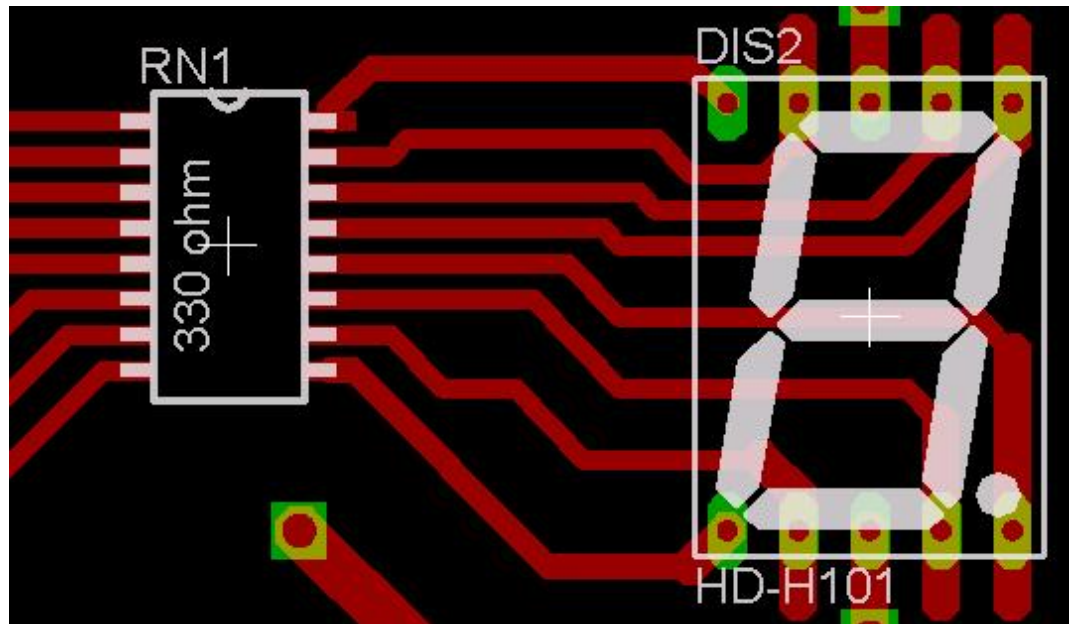
Toimintavarmuutta on parannettu jo mainituilla suodatus- ja häiriönpoistokondensaattoreilla. EPM7064 piirin OE1, OE ja GCLRn signaalit on pitkän harkinnan jälkeen vedetty kiinteästi ylös kukin oman 2.2 kilo-ohmin vastuksen kautta.

Jokainen I/O-liitin on kytketty pitkällisen harkinnan tuloksena johonkin tietyn komponentin tiettyyn pinniin. Tässä on otettu huomioon, että johtimet olisivat selkeässä järjestyksessä piirilevyllä. Selkeys puuttui ensimmäisestä versiosta. Piirikaaviosta (LIITE 4) tuli epäselvempi kuin ensimmäisen version kohdalla, mutta sille löytyy looginen selitys. Kuviossa (12) nähdään piirikaaviosta vastusverkon ja 7-segmentti näytön välinen kytkentä. Johti-

met on rumasti vedetty ristiin, mutta tämä auttoi huomattavasti johdinten vetämiseen piirilevyllä. Kuvio (13) näyttää saman asian Layoutissa. Layoutissa johtimet ovat järjestyksessä, mikä on paljon tärkeämpää kuin järjestys piirikaaviossa. Piirikaaviota piirrettäessä ensimmäiseen versioon laitettiin johtimet järjestykseseen, jolloin ne menivät ristiin Layoutissa, mikä vaikeutti huomattavasti reititystä. Jokaisen johtimen kytkentä piirikaaviossa oli tarkastettava heti Layoutista, jotta välttyttiin ristiin meneviltä johtimilta. Tämä vei aikaa huomattavan paljon, mutta piirilevystä tuli sitten huomattavasti selkeämpi (LIITE 5).



Kuvio 12. 7-Segmentti näytön kytkentä etuvastuksineen piirikaaviossa



Kuvio 13. 7-Segmentti näytön kytkentä etuvastuksineen Layoutissa

Versio kahdesta tehtiin myös kaksikerroksinen ja alapuolella menevät käyttöjännite ja maavedot. Myös JTAG-latausliittimen signaalit ovat vedetty alapuolelle. Alapuolella on lisäksi kuparitäyttö (LIITE 5).

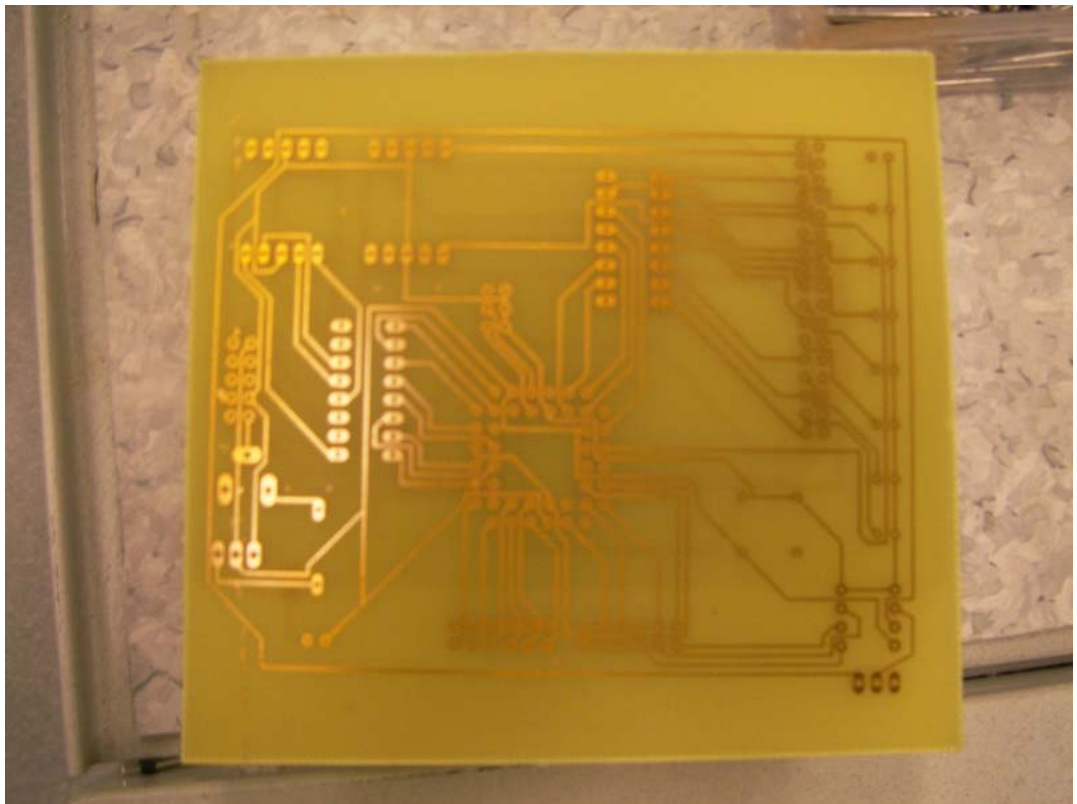
Kaikki johtimet on vedetty käsin ja autoreititin unohdettu kokonaan. Pääsääntöinen johdinten leveys on 24-32mils ja käyttöjännite- ja maavedot ovat 50-100mils leveitä, jolloin ylikuuluminen on minimoitu. Eristeväli on 16-50mils eli vähintäänkin riittävä joka paikassa.

3.3.4 piirilevyn valmistus

Piirilevy voidaan valmistaa monella eri tavalla, yleisin tapa harraste/opiskelu käytössä on syövyttäminen. piirilevy voidaan teettää jollain ulkopuolisella yrityksellä tai jyrsiä jyrsimellä, jos sellainen on käytössä.

Tässä tapauksessa kaksikerrospiirilevy valmistettiin Koulutus keskus Salkauksen tiloissa syövyttämällä. Syövytys menee yksinkertaistettuna näin:

1. Kalvoille on tulostettu piirilevyn ylä- ja alapuolen kuva. Kalvot kohdistetaan ja teipataan toisiinsa kiinni, jonka jälkeen syövytettävä piirilevy laitetaan niiden väliin.
2. Valotus. Kalvot ja piirilevy laitetaan valotuslaitteeseen, joka valottaa levyä muutaman minuutin, valotuksesta jää levyyn kuva ”muistiin”.
3. Kehitys. Valotettu piirilevy laitetaan lipeäliuokseen muutamaksi minuutiksi ja heilutellaan sitä välillä. Jos valotus on onnistunut, liukee lakka levystä valottuneista kohdista ja kuva levyllä erottuu selvästi. Kehittynyt levy huuhdellaan huolellisesti kehityksen jälkeen.
4. Syövyttäminen. Pesty levy laitetaan Ferrikloridi liuokseen, jossa se on n.20 minuuttia. Kun peittämättömät kuparifolio kohdat ovat syöpyneet, levy pestään ja siihen porataan reiät oikeisiin kohtiin.



Kuvio 14. Valmis piirilevy menossa porattavaksi

3.4 Ohjelmointi

Kaikilla piirivalmistajilla on omat ohjelmansa, joilla piirien suunnittelu prosessi toteutetaan, eivätkä ne ole välttämättä yhteensopivia toistensa kanssa. Suunnitteluprosessi menee niin, että aluksi kuvataan halutun laitteiston toiminta jollain ohjelmointikielellä.

Standardoituja ohjelmointikieliä (ohjelmankuvauskieliä) ovat VHDL ja Verilog, joista VHDL on yleisempi. Laitteistonkuvaus voidaan tehdä myös piirikaaviomenetelmällä tai C-kielellä, jotka sitten käännetään erillisellä ohjelmalla VHDL:ksi. Piirin valmistajasta riippuu, mitä kuvauskieliä on mahdollista käyttää piirin suunnittelussa, eli mitä kielien välisiä muunto-ohjelmia valmistajalla on tarjolla.

Joillakin valmistajilla on omia kieliä, jotka ovat yleensä yksinkertaistettuja ja helpommin omaksuttavissa olevia kieliä kuin VHDL tai Verilog-kielet. Latticen CPLD-piirejä voidaan ohjelmoida ABEL-kielellä, joka on nykyään joltain osin lähellä VHDL-kieltä. Alteralla taas on oma AHDL-kieli ja Philipsillä PHDL-kieli. Näiden kielien ongelmana on yhteensopivuuden puute muiden valmistajien piirien kanssa.

Seuraavaksi valmis ohjelma täytyy simuloida, jotta nähdään mahdolliset virheet ja pystytään korjaamaan ne. Erityisen tärkeää simulointi on antisulaketyyppisten piirien kanssa, joita ei voi ohjelmoida uudelleen. Monimutkaisissa piireissä täytyy eri lohkojen toiminta simuloida erikseen, muuten voi olla vaikeaa havaita, missä lohkoissa järjestelmätason virheet sijaitsevat. Simulointi säästää aikaa ja vaivaa, vaikka käytettävä piiri olisikin uudelleen ohjelmoitavissa. Suuremmille piireille ja suurimmilla ohjelmilla ohjelman lataus kestää useita minuutteja.

Simuloinnin jälkeen, logiikkasynteesissa luodaan piirin ohjelmointia varten tarvittava tiedosto. Tapahtuma on automaattinen, kunhan käyttäjä vain käynnistää sen. Eri valmistajien ohjelmat muodostavat hyvinkin erilaisia toteutuksia, ja toinen ohjelma voi tarvita huomattavasti enemmän piirin

portteja käyttöönsä kuin toinen. Tarvittava porttien määrä riippuu myös mitä ohjelmointikieltä on käytetty. Piirikaavioesitys tarvitsee yleensä vähiten portteja käyttöönsä.

3.4.1 VHDL

VHDL-kieli on syntynyt Yhdysvaltain puolustusministeriön rahoittaman suurtiheyspiirejä tutkivan projektin sivutuotteena. Ensimmäinen versio kielestä ilmestyi vuonna 1985. Vuonna 1987 se hyväksyttiin IEEE:n standardiksi sen jälkeen, kun siihen ensin oli tehty lukuisia parannuksia ja muutoksia. IEEE 1076-standardi määrittelee VHDL-kielen täydellisesti, mutta täysin siirrettävien kuvausten tekeminen osoittautui vaikeaksi. Eri valmistajien simulaattoreilla saadut tulokset olivat poikkeavia keskenään, joten Standardikomitea julkaisi lisästandardin IEEE 1164, joka määrittelee joukon simuloinnissa tarvittavia datatyyppejä. Yhdessä standardit määrittelevät VHDL-kielen yksikäsitteisesti. (Tolonen, 1996.)

Luonteeltaan VHDL-kieli on samanlainen kuin muutkin ohjelmointikieliet. Se sisältää kaikki korkeantason ohjelmointikielten ominaisuudet, mutta siinä on lisäksi tuki todelliselle rinnakkaiskäsitteilylle. VHDL-kieltä ei ajeta prosessissa, vaan sillä tehdään halutunlainen kytkentä piirille. VHDL:ää voidaan käyttää kokonaisten järjestelmien määrittelyyn, simulointiin hallintaan ja testaukseen. Se on ennen kaikkea simulointityökalu, jolla kuvattujen järjestelmien toimivuus voidaan varmistaa mahdollisimman aukottomasti.

VHDL-kieli on suhteellisen vaikea oppia, jos sen kaikkia ominaisuuksia käytetään. Siirtyminen toisesta kielestä VHDL:ään ei kuitenkaan ole hirveän vaikeaa.

VHDL-koodi, jolla kehitysalustaa testattiin, löytyy kommentoituna tämän työn luvusta (4) mittaukset ja kokonaisuudessaan liitteistä (LIITE 7).

3.4.2 Quartus II

Kehitysalustalla sijaitsevan ohjelmitavan piirin ohjelmointiin ja suunnitteluun käytetään Alteran Quartus II ohjelmaa. Ohjelmointi suoritetaan VHDL- ja C-kielellä. Testataan myös ohjelman luomista piirikaaviolla, jolloin elektroniikan peruskomponentit yhdistetään näytölle piirrettävässä schemassa, ja ohjelma tekee niiden perusteella itsenäisesti piirille kääntyvän ohjelmatiedoston.

Quartus II Web Edition 6.1 on opiskelijoille ja harrastelijoille tarkoitettu versio, johon saa 4kk lisenssin ilmaiseksi. Quartus on Alteran oma suunnitteluohjelma, joka on tarkoitettu ohjelmitavien piirien:

- suunnittelun syöttöön
 - lohkoaviona
 - piirikaaviona
 - korkean tason ohjelmointikielellä (VHDL,C,)
 - valmislohkona
- synteessin tekemiseen
- toiminnan verifiointiin ja simulointiin
 - testipenkit
 - ajoitusanalyysi
 - tehonkulutuksen estimointi
 - signaalianalyysi
- piirin ohjelmointiin.

Uusin versio Quartus II:sta on versio 6.1. Ohjelmiston suurin uudistus on alan ensimmäinen TimeQuest- ajoitusanalysointilaite, joka tuottaa kattavan tuen teollisuus-standardille Synopsysin Design Constraints (SDC)- ajoitusformaatile.

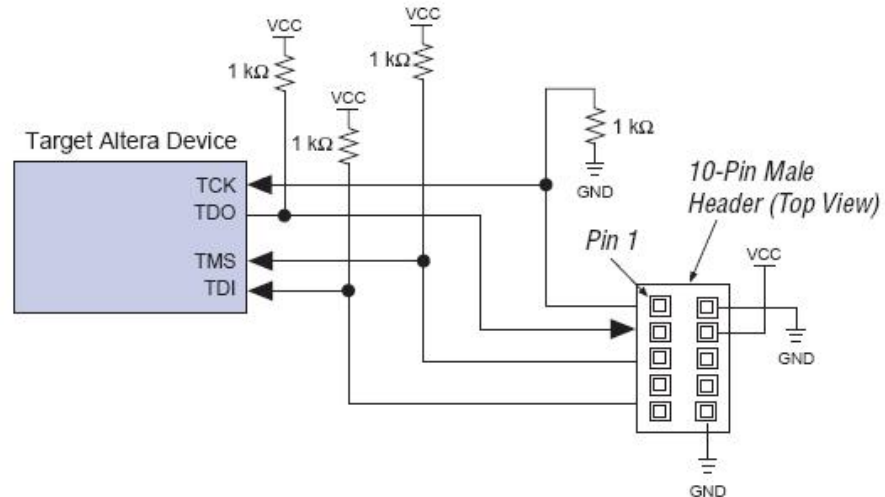
Ajoitusanalysointilaitteen avulla voidaan luoda ja analysoida aiempaa vaativimpia kellotusrakenteita sisältäviä suunnitelmia. Myös työkalun ryhmätyöskentelyn ominaisuuksia on kehitetty suuntaan, jossa projektipäällikkö voi hallita lohkojen välisiä ajoitusrakenteita suorituskyvyn maksimoimiseksi.

3.4.3 ByteBlaster

Valmis ohjelma siirrettiin piirille JTAG-liitännän kautta ByteBlaster-ohjelmointikaapelilla. Kaapeli liitetään tietokoneen LPT-porttiin ja ajurien asennuksen jälkeen, se on heti käytettävissä. Ohjelmointi tapahtuu Quartus II ohjelmistoa käyttäen ja siirrettävä tiedosto on sof-tiedosto, joka on valmistettu synteessissä. Piirilevylle tehtiin 10-pinninen JTAG-portti, johon ByteBlaster-kaapelin toinenpää liitetään. JTAG-liittimeen tulevat signaalit ovat:

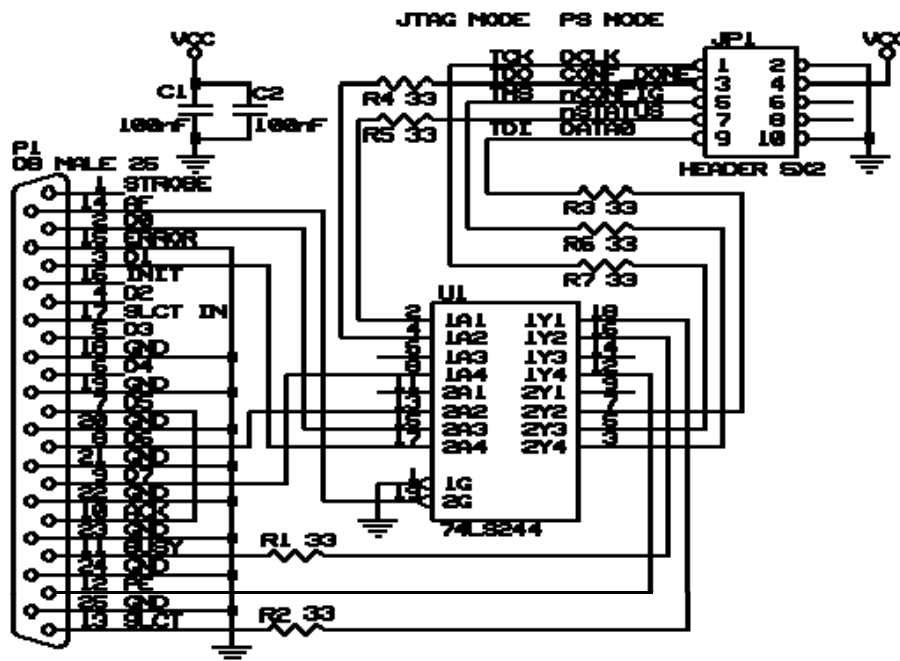
- TCK, Test clock, tahdistussignaali
- TDO, Test data output, saadaan ohjelma ulos testausta varten
- TMS, Test mode select, valitaan tilaksi ohjelmointi tai tavallinen
- TDI, Test data input, ohjelma syötetään sisään.

Lisäksi ByteBlaster-kaapeli tarvitsee käyttöjännitteen ja maan piirilevyltä toimiakseen. Ylös- ja alasveto vastukset täytyy muistaa laittaa paikoilleen, muuten ohjelman lataus ei toimi halutunlaisesti. (Kuva 15).



Kuva 15. JTAG-ohjelmointiportin kytkentä

ByteBlaster ohjelmointi kaapeli maksaa Elfan luettelossa 215 euroa, mutta sen voi helposti rakentaa itsekin (Kuvio 16). Koko kaapelin toiminta perustuu yhden 74HC244 väyläohjain piirin ympärille. Altera ei enää itse julkaise kaapelin rakennetta esittävää piirikaaviota, vaikka se oli esillä ByteBlasterin Data Sheetissä vielä vuonna 1999.



Kuvio 16. ByteBlasterin piirikaavio (Woody Johnson)

3.5 ongelmat

Ensimmäiset ongelmat tulivat, kun piti selvittää EPM7000-sarjan piirien sopivuutta projektiin. Lähinnä ohjelmitavuuteen Quartus II-ohjelmalla ja yhteen sopivuuteen ByteBlaster-ohjelmointikaapelin kanssa, ei meinannut löytyä varmuutta mistään. Varmuus saatiin lopulta Alteran tuesta.

Eagle oli minulle uusi ohjelma, mistä aiheutui joitain ongelmia. Komponentit olivat erinimisiä Elfan luettelossa kuin Eaglen kirjastossa, joten sopivia komponentteja sai kaivaa jonkun aikaa. A/D-muunnin vaihdettiin malliin, joka löytyi suoraan Eaglestä, kun alkuperäistä ei löytynyt.

Optimaalinen piirilevyn suunnittelu ei onnistunut, kun piirilevyn koko oli rajattu 80x100mm. Piirilevyllä meinasi tulla väkisinkin ahdasta, ja yksipuoleisen piirilevyn valmistus ei tullut kysymykseenkään, vaikka se oli alkuperäinen suunnitelma. Ensimmäisessä versiossa, tuhraantui todella paljon aikaa etsiessä optimaalista osien ja vetojen sijoittelua. Silti parhaassakin tapauksessa siinä olisi ollut 16 kpl hyppylankoja, joten sitä ei sitten koskaan valmistettu, vaan piirilevystä tehtiin suosiolla kaksikerroksinen. Tosin siinäkin oli ongelmia niin valmistuksen kuin komponenttien juottamisen kanssa. Kaksikerros piirilevyn valottaminen ei meinannut onnistua, koska käytetylle 2-puoleiselle piirilevyllä piti etsiä sopiva valotusaika, mikä onnistui vasta kolmannella yrittämällä. Siltikin osa vedoista oli poikki piirilevyllä, ja niitä joutui sitten korjaamaan.

Kun piirilevy oli saatu valmistettua, tuli eteen toinen ongelma. Komponenttien juottaminen ei onnistunut joka paikkaan, mikä johtui itse komponenttien rakenteesta. Kolvia oli vain mahdollista saada tungettua joidenkin komponenttien jalkoihin, kun jalat jäivät komponentit alle piiloon. Hyppylankoja oli lopulta 8 kpl tässä sauraavassakin versiossa. Kokonaan uusi versio tullaan tekemään, kunhan kortin ominaisuudet on saatu ensin testattua tällä kyseisellä kortilla. Valmis versio tehdään kokonaan pintaliitoskomponenteilla.

Esille tuli myös EAGLEN muutama huonopuoli, ohjelma ei tarkista ollenkaan kytkentöjä, ja yksi kytkentä oli jäänyt kytkeytymättä, vaikka se näyttää piirikaaviossa kytketyltä. Ohjelma ei myöskään näytä komponenttien VCC ja GND nastoja, ellei niitä erikseen oteta näkyviin, niinpä se ei myöskään osannut yhdistää kaikkia tarvittavia nastoja käyttäjännitteeseen, vaan taas joutui vetämään hyppylankoja.

Kun kortti oli juotettu kasaan niin, yritettiin siihen saada yhteyttä Quartuksen ja ByteBlasterin kanssa. Yhteys kortille ei kuitenkaan onnistunut. Käytössä oli myös Alteran oma kehitysalusta, jota olin käyttänyt Quartuksen opettelussa. Yllättäen tähänkään kehitysalustaan ei saatu yhteyttä ja Alteran tukeen soittamalla saimme tietää, että Tietokoneen LPT-portti täytyi BIOS:sta muuttaa kaksi suuntaiseksi, ja ByteBlaster-kaapeli tarvitsi vielä ajuritkin toimiakseen. Ajurien asennus oli melko työlästä, koska Windows tunnistaa ByteBlasterin, ääni, video tai peliohjaimena, mutta automaattisesti se ei tunnista koko ByteBlasteria. Quartuksen kotisivuilta löytyi ajurien asennusohje, joten asennus onnistui lopulta ohjeita seuraamalla. ByteBlaster tarvitsi vielä TDI ja TMS signaaleihinsa ylösveto vastukset, sekä TCK-signaaliin alasvetovastuksen. Nämä olivat jääneet pois, koska niitä ei ollut ensin luetussa datasivussa ollenkaan. Kaiken tämän jälkeen testiohjelma saatiin lopulta ladattua piirille.

ByteBlaster tarvitsee toimiakseen tietokoneelta LPT-portin, jota ei löytynyt käytettävästä kannettavasta tietokoneesta, joten ohjelman lataukseen tarvittiin toinen tietokone.

Uuden ohjelman harjoittelu on aina haastavaa, näin oli myös Quartuksen kanssa. Alteran kotisivuilta löytyi havainnollinen video, kuinka Quartus otetaan käyttöön, mistä oli korvaamatonta apua uuden ohjelman opettelussa. Suomenkielellä ei löytynyt yhtään opasta käyttöönottoon, joten sanakirja oli kovassa käytössä.

4. TESTAUS JA MITTAUS

Kehitysalustan toiminnan testausta varten tehtiin VHDL-ohjelma, jonka toiminta ja rakenne kerrotaan alla. Koko ohjelma on liitteenä (LIITE7).

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
```

Esimerkissä käytettiin ieee kirjastoa, josta oli otettu käyttöön kaikki ominaisuudet, ".all" komennolla/päätteellä.

```
ENTITY numerot IS
PORT ( dip1,dip2,dip3,dip4,dip5,dip6,dip7,dip8 : bit;
      seg : OUT STD_LOGIC_VECTOR (6 DOWNTO 0));
END numerot;
```

Ohjelman "kotelo" luodaan tässä vaiheessa, kotelon nimi on "numerot".

Sisääntulot (dip1,dip2,dip3,dip4,dip5,dip6,dip7 ja dip8) ovat dippi kytkimiltä tulevia bitti muotoisia signaaleja (1 tai 0). Ulostulo "seg" on bittikartta tyyppinen, 7-segmentti näytölle menevä 7-bittiiä leveä signaali. Yhteensä nämä signaalit vievät piiriltä 14 kpl:ta I/O-liitäntöjä.

```
ARCHITECTURE näyttö OF numerot IS
BEGIN
```

```
PROCESS (dip1,dip2,dip3,dip4,dip5,dip6,dip7,dip8)
BEGIN
  if dip1 = '1' then
    seg <= "1111001";
  else

    if dip2 = '1' then
      seg <= "0100100";
    else

      if dip3 = '1' then
        seg <= "0110000";
      else

        if dip4 = '1' then
          seg <= "0011001";
        else

          if dip5 = '1' then
```

```

    seg <= "0010010";
else

    if dip6 = '1' then
        seg <= "0000010";
    else

        if dip7 = '1' then
            seg <= "1111000";
        else

            if dip8 = '1' then
                seg <= "0000000";
            else
                seg <= "1111111";
            end if;
        end if;
    end if;
end if;
end if;
end if;
end if;
end if;
end if;
end if;
end process;
END näyttö;

```

Arkkitehtuurivaiheessa luodaan sisältö ”numerot” kotelolle, nimeltään ”näyttö”. Ohjelma toimii siten, että painettaessa dip1-kytkintä syötetään näytölle bitit 1111001, jolloin kaikki paitsi kaksi näytön ledeistä ovat sammuksissa (näytöt ovat yhteisanodi-tyyppisiä). Näistä kahdesta ledistä muodostuu numero 1 ja vastaavasti, kun dip2-kytkintä painetaan, tulee näyttöön numero 2. Näin toimitaan numeroon 8 asti (kytkimiä on 8 kpl:ta). Viimeinen IF-lause sammuttaa kaikki ledit näytöstä, kun mitään kytkintä ei paineta; ilman tätä IF-lausetta näytön kaikki ledit palaisivat, kun yksikään kytkin ei ole painettuna.

Ohjelma testattiin kehitysalustan ensimmäisessä versiossa, kun toinen versio ei ollut vielä valmis. Ylimääräisiin I/O-liitäntöihin liitettiin ohjelmassa käytetty 8-osainen dip-kytkin. Ohjelma on laitettu myös PROCESS- yksikköön, koska alunperin oli tarkoitus tehdä koodille jatkoa. PROCESS on

itsenäinen yksikkö VHDL-kielessä, joka kytkeytyy käyttöön, kun saa määrätyn herätteen. Tässä tapauksessa kytkimet toimii herätteenä.

A/D-muuntimelle oli tarkoitus tehdä sen toimintaa testaava ohjelma, mutta se jäi ajanpuutteen takia tekemättä. Yleismittarilla mittaamalla, sen fyysiset liitännät ovat kuitenkin kunnossa.

5. YHTEENVETO

Kehitysalusta täytti sille alussa annetut vaatimukset, se oli helppokäyttöinen ja ohjelmoitavissa Quartus II-ohjelmalla. Ohjelman syöttäminen piirille onnistui myös vaatimusten mukaisesti ByteBlaster-ohjelmointikaapelilla. Työlle ei annettu mitään sähköisiä arvoja, jotka sen tuli täyttää, näin ollen niihin ei kiinnitetty huomiota. Testiohjelma toimi kuten pitikin, joten kehitysalusta voidaan katsoa toimivaksi.

Kehitysalustan suunnittelussa noudatettiin piirilevysuunnittelun sääntöjä. Piirilevyn selkeyteen vaikutti eniten se, että jokainen johdinten kytkentä harkittiin tarkasti. Myös autoreitittimen todettiin olevan huono ratkaisu tämän kokoisessa suunnitelmassa, ja kaikki vedot vedettiin käsin.

Kehitysalusta tehtiin aikuisopiskelijoiden käyttöön, ja sen käyttöönoton piti olla nopeaa ja helppoa. Eniten aikaa työssä meni Quartuksen käyttöönottoon ja piirinojelmointiin tarvittavan ByteBlaster-kaapelin toimintakuntoon saattamiseen, joten oppilaille kirjoitettiin ohjeet, joilla käyttöönotto onnistuu helposti ja nopeasti.

Ohjelmoitavien piirien valmistajien kotisivuilta tiedon kalastaminen oli myös melko työlästä, ja eri valmistajat ilmoittavat tietonsa piireistään hieman eritavalla, joten piirien vertailu toisiinsa oli vaikeahkoa. Kaikki valmistajat kyllä ilmoittivat nopeimpien ja isoimpien piiriensä tiedot selvästi, mutta koulu tai harrastekäytössä käytettyjen piirien tiedot sai kaivamalla kaivaa.

Kehityskortin piirilevy tehtiin syövyttämällä Koulutuskeskus Salpauksen tiloissa ja kasattiin tavallisella juotosasemalla. Pintaliitoskomponenttien juottaminen onnistui, koska käytettävissä juotosasemassa oli 0,4mm kärki. Muita käytettyjä laitteita oli PC-tietokone, jossa oli LPT-portti, jota ByteBlaster käytti. Kannettavalla tietokoneella ei ohjelman lataus onnistunut piirille, kun siitä puuttui kyseinen LPT-portti.

LÄHTEET

Veijalainen, T. 2004. Ohjelmoitavat piirit. Luento Lahden ammattikorkeakoulu, Tekniikan laitos, Lahti.

Wikström, K. 1999. Muuttuvat FPGA- ja CPLD-markkinat. Prosessori 11/99, 91-95.

Laakkonen, O. Esiselvitys ohjelmoitavien logiikkapiirien tämän hetken tilanteesta ja niille löytyvistä IP-lohkoista ja työkaluista [verkkodokumentti]. Lappeenranta: Lappeenrannan teknillinen korkeakoulu, Sähkötekniikan osasto, 2003 [viitattu 21.9.2006]. Saatavissa: <http://www.ee.lut.fi/staff/Ossi.Laakkonen/fpga.pdf>

Tuominen, P. Signaaliprosessorit ja FPGA-piirit [online]. [viitattu 30.9.2006]. Saatavissa: <http://www.cc.puv.fi/~pt/fpga2.htm>

Salmela, J. 1998. IP-lohkot ja co-design. Prosessori 3/98, 20-22.

Wikström, K. 1998. Ohjelmoitavilla nopeasti markkinoille. Prosessori 3/98, 49-52.

Xilinx. xilinx3000.pdf [online] [viitattu 18.10.2006] Saatavissa: direct.xilinx.com/bvdocs/publications/3000.pdf

Xilinx. Virtex [online] [viitattu 20.10.2006] Saatavissa: http://www.xilinx.com/products/silicon_solutions/fpgas/virtex/virtex5/index.htm

Xilinx. Spartan [online] [viitattu 20.10.2006] Saatavissa: http://www.xilinx.com/products/silicon_solutions/fpgas/spartan_series/index.htm

Xilinx. EasyPath [online] [viitattu 20.10.2006] Saatavissa:

http://www.xilinx.com/products/silicon_solutions/fpgas/easypath/index.htm

Xilinx. CoolRunner [online] [viitattu 20.10.2006] Saatavissa:

http://www.xilinx.com/products/silicon_solutions/cplds/coolrunner_series/index.htm

Xilinx. XC9500 [online] [viitattu 20.10.2006] Saatavissa:

http://www.xilinx.com/products/silicon_solutions/cplds/xc9500_series/index.htm

Xilinx. Aerospace and defence [online] [viitattu 20.10.2006] Saatavissa:

http://www.xilinx.com/products/silicon_solutions/market_specific_devices/aero_def/index.htm

Altera. Stratix [online] [viitattu 21.10.2006] Saatavissa:

<http://www.altera.com/products/devices/stratix3/st3-index.jsp>

Altera. Cyclone [online] [viitattu 21.10.2006] Saatavissa:

<http://www.altera.com/products/devices/cyclone2/cy2-index.jsp>

Altera. MAX II [online] [viitattu 21.10.2006] Saatavissa:

<http://www.altera.com/products/devices/cpld/max2/mx2-index.jsp>

Altera. MAX 7000 [online] [viitattu 21.10.2006] Saatavissa:

<http://www.altera.com/products/devices/max7k/m7k-index.html>

Lattice Semiconductor. SC Extreme [online] [viitattu 22.10.2006] Saatavissa:

<http://www.latticesemi.com/products/fpga/sc/index.cfm>

Lattice Semiconductor. ECP2 [online] [viitattu 22.10.2006] Saatavissa:

<http://www.latticesemi.com/products/fpga/ecp2/index.cfm>

Lattice Semiconductor. XP [online] [viitattu 22.10.2006] Saatavissa:
<http://www.latticesemi.com/products/fpga/xp/index.cfm>

Lattice Semiconductor. XP [online] [viitattu 22.10.2006] Saatavissa:
<http://www.latticesemi.com/products/cpldspld/ispmach4000bcv.cfm>

Actel. Igloo [online] [viitattu 23.10.2006] Saatavissa:
<http://www.actel.com/products/igloo/>

Actel. ProASIC [online] [viitattu 23.10.2006] Saatavissa:
<http://www.actel.com/products/pa3/>

Actel. Axcelerator [online] [viitattu 23.10.2006] Saatavissa:
<http://www.actel.com/products/axcelerator/>

Actel. SX-A / SX [online] [viitattu 23.10.2006] Saatavissa:
<http://www.actel.com/products/sxa/index.html>

Actel. MX [online] [viitattu 23.10.2006] Saatavissa:
<http://www.actel.com/products/mx/index.html>

Achronix semiconductor corporation. [online] [viitattu 25.10.2006] Saatavissa: <http://www.achronix.com/>

Eagle CadSoft. [online] [viitattu 30.10.2006] Saatavissa:
<http://www.cadsoft.de/>

Tolonen, J. VHDL, logiikkasuunnittelijan ohjelmointikieli [verkkodokumentti] 1996 [viitattu 4.11.2006] Saatavissa:
<http://koti.mbnet.fi/siliconf/JukkaTolonen/index.html>

Altera. Quartus II [online] [viitattu 6.11.2006] Saatavissa:
<http://www.altera.com/products/software/products/quartus2/qts-index.html>

Johnson, W. Build your own ByteBlaster [online]. 2001 [viitattu 6.11.2006]

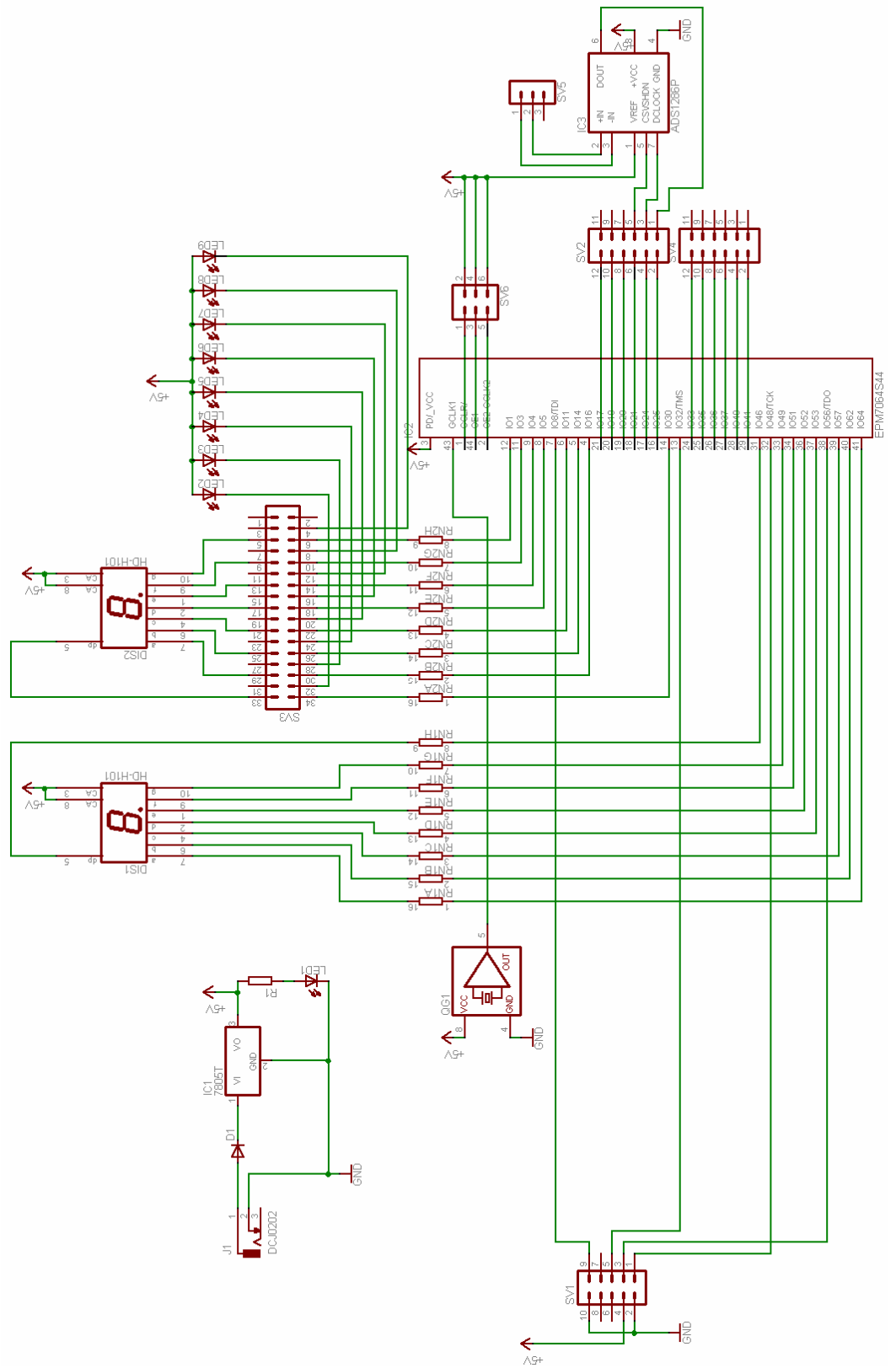
Saatavissa:

<http://opencollector.org/history/freecore/Build%20your%20own%20ByteBlaster!.htm>

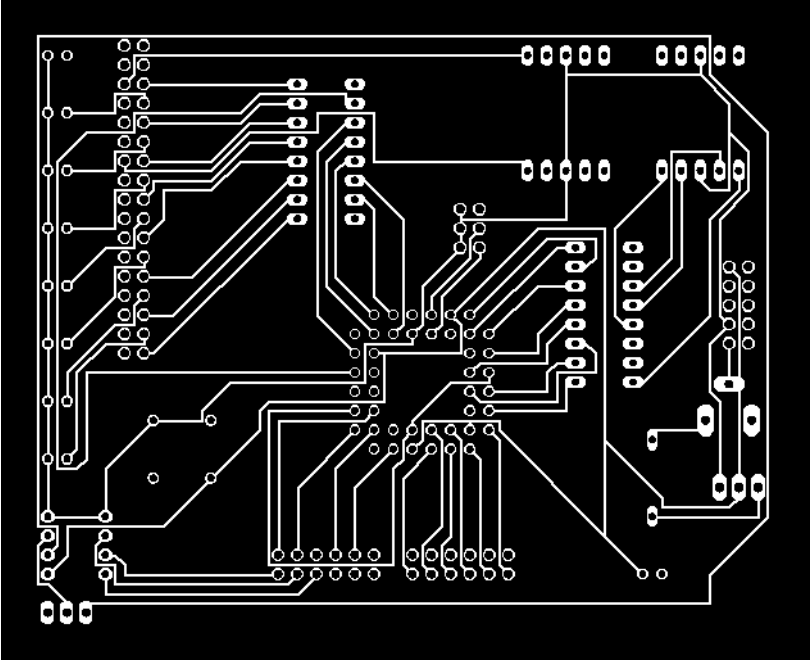
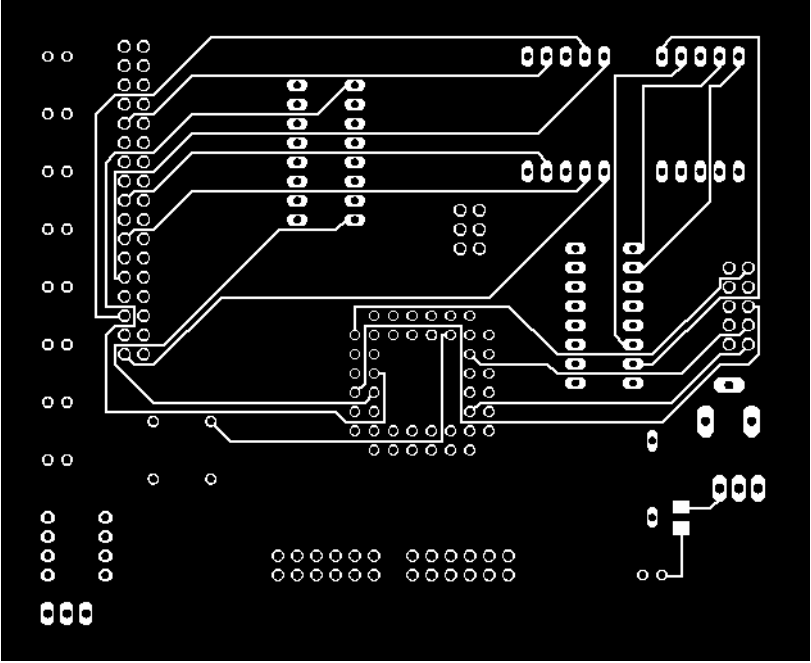
LIITTEET

- LIITE 1 Piirikaavio V1.1
- LIITE 2 Foliot ylä- ja alapuoli V1.1
- LIITE 3 Komponenttien paikat V 1.1
- LIITE 4 Piirikaavio V2.0
- LIITE 5 Foliot ylä- ja alapuoli V2.0
- LIITE 6 Komponenttien paikat V 2.0
- LIITE 7 VHDL-koodi, testiohjelmasta
- LIITE 8 Kehitysalustan ja Quartuksen käyttöönotto-opas

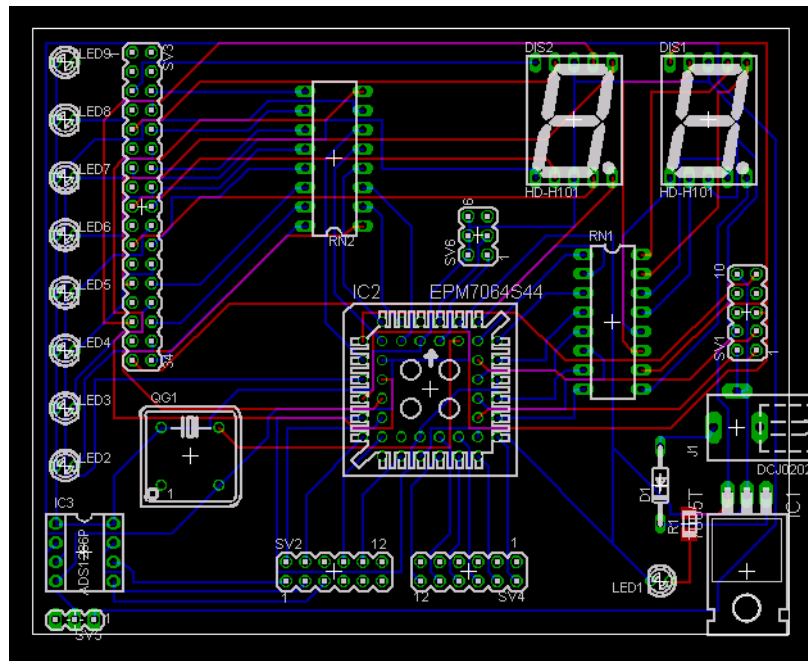
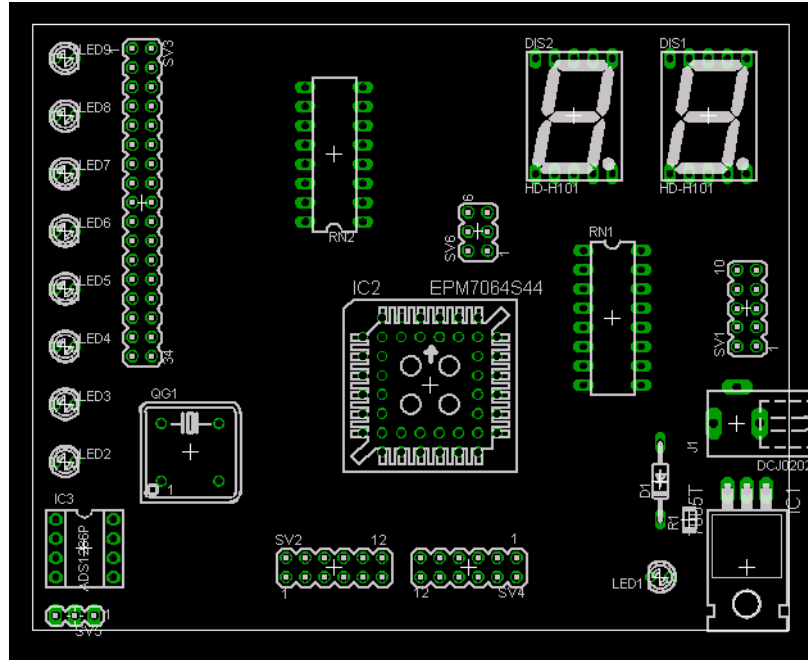
LIITE 1



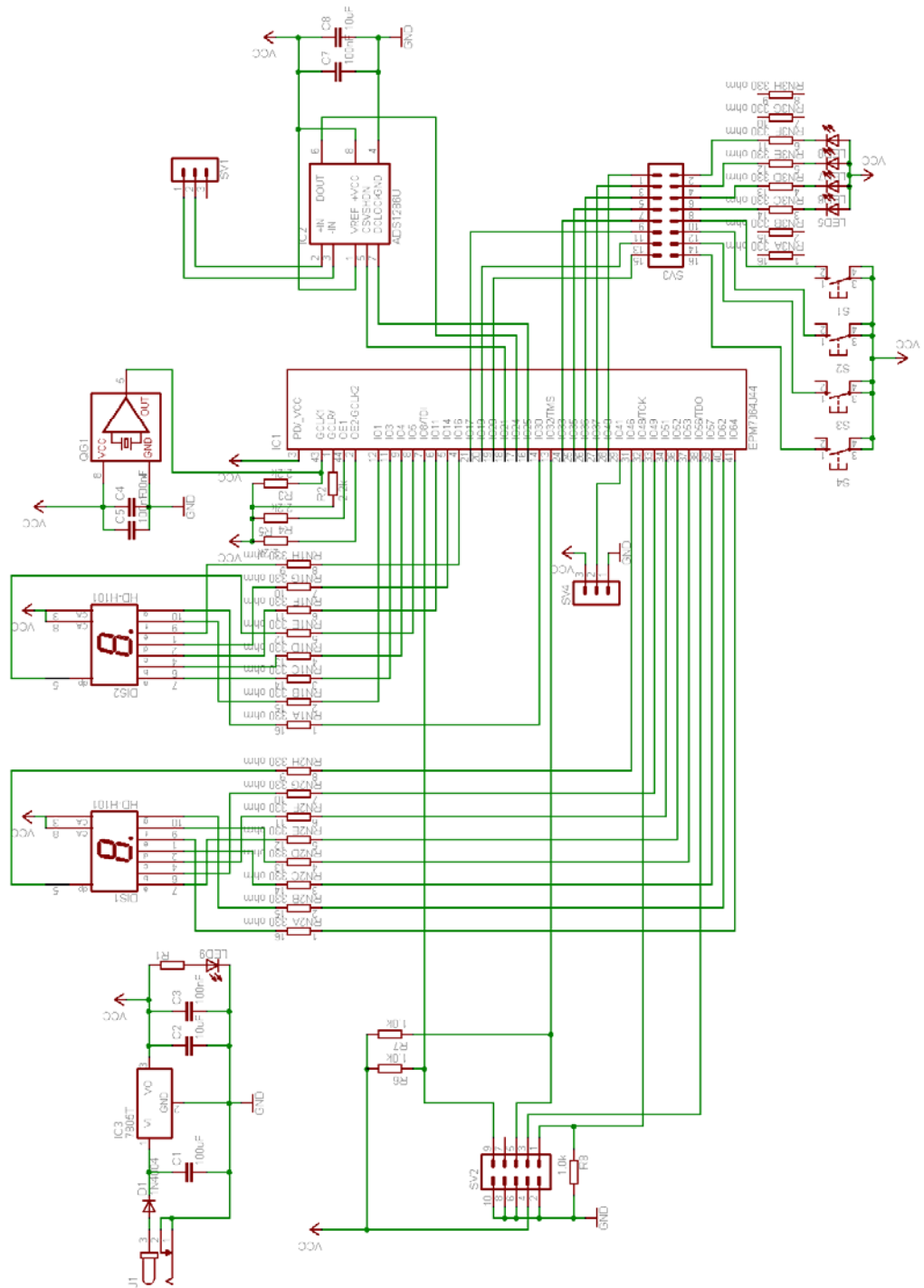
LIITE 2

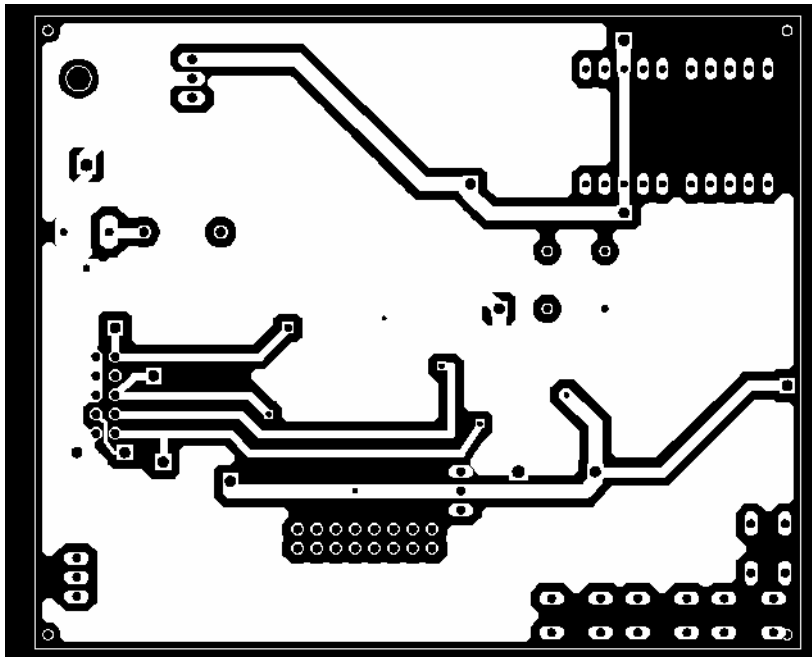
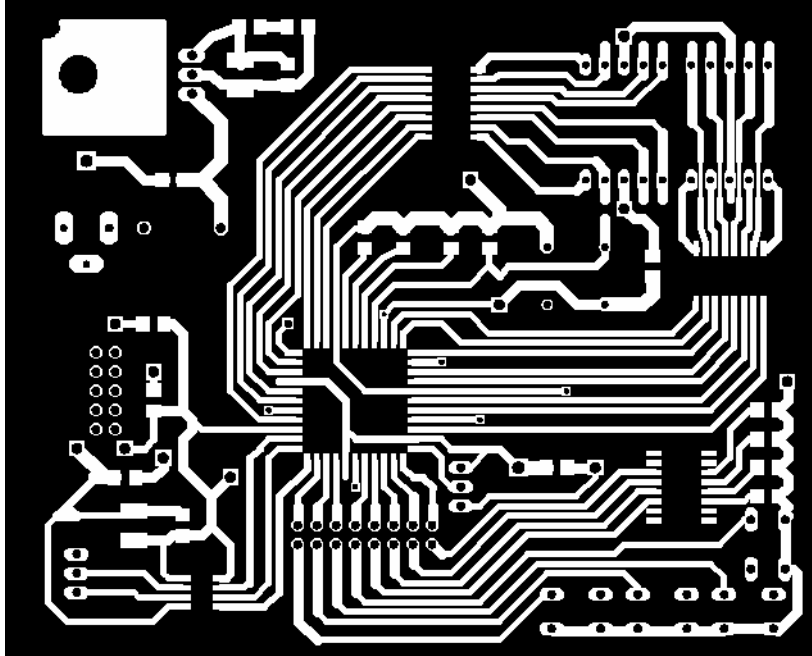


LIITE 3

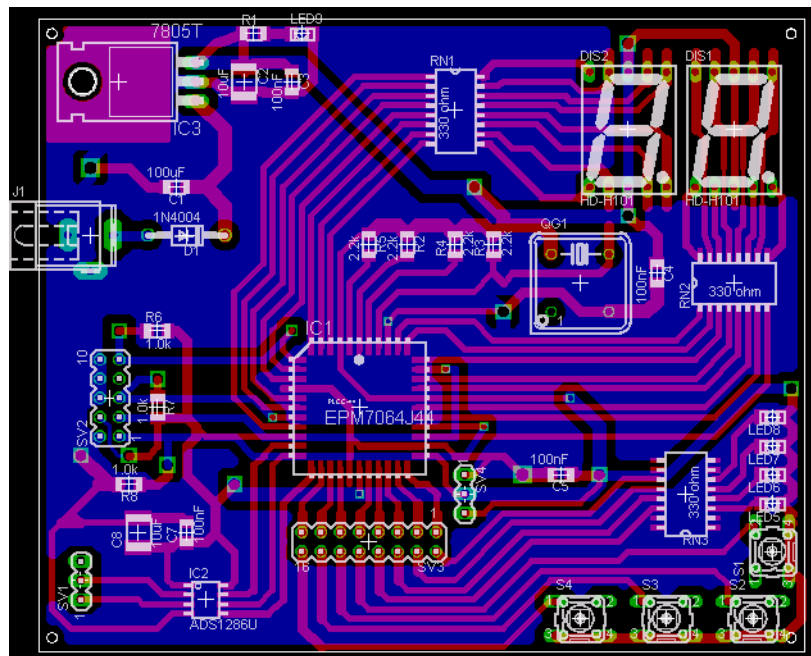
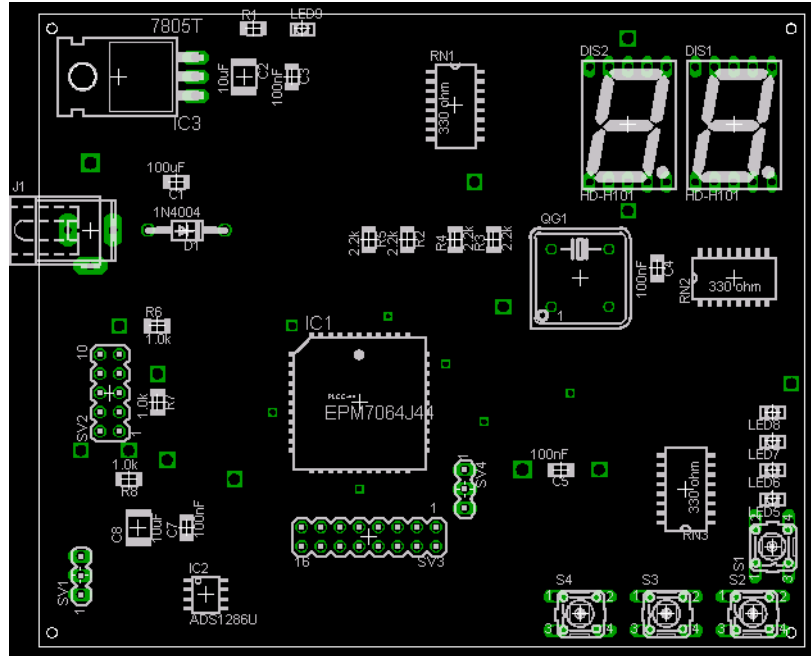


LIITE 4





LIITE 6



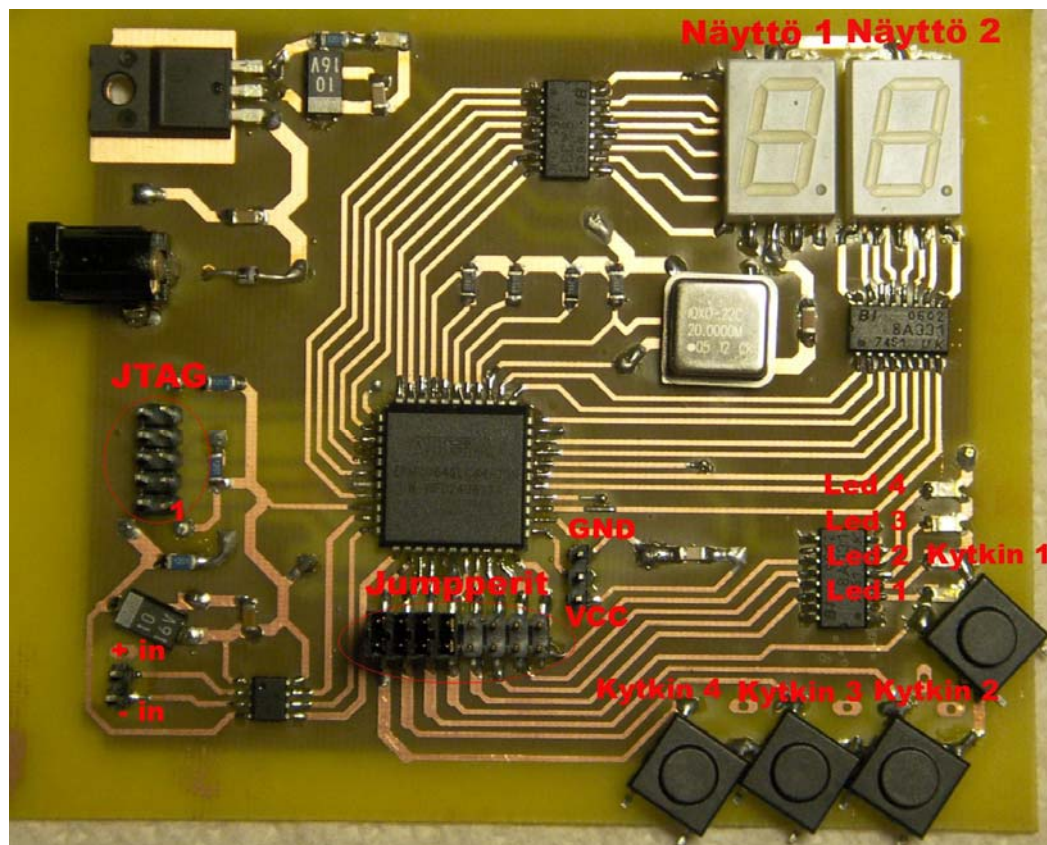
Käyttöopas

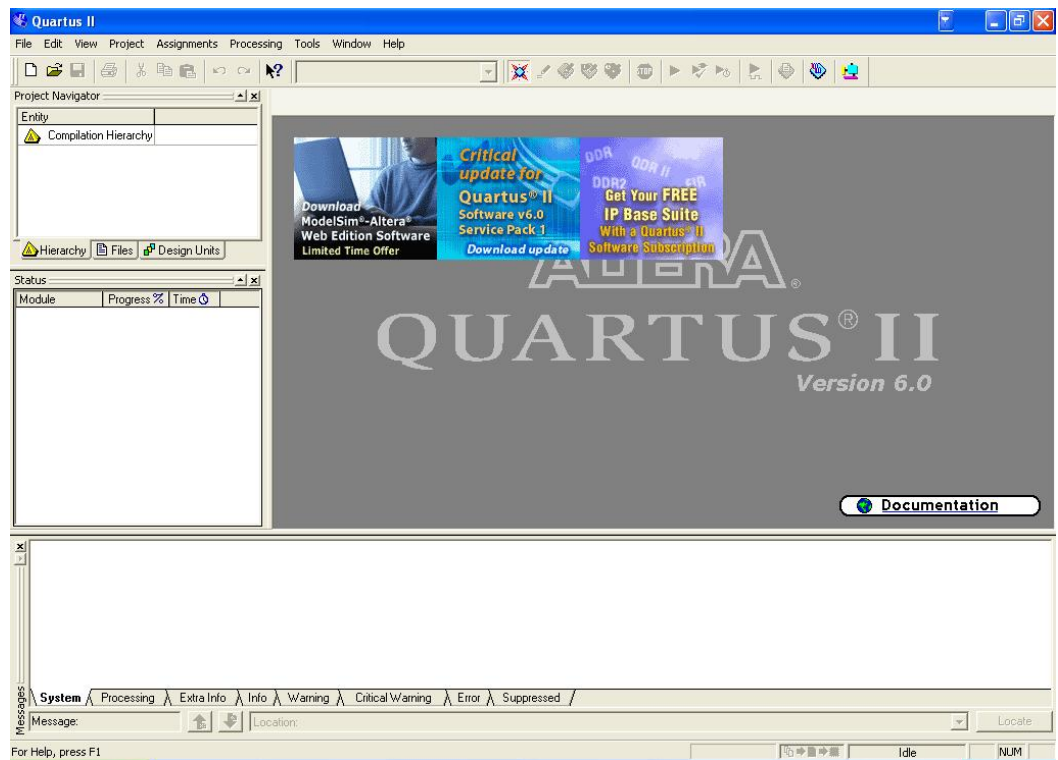
Altera MAX 7000-pohjaisen kehitysalustan ja Quartus II ohjelman käyttöönottamiselle

LAHDEN AMMATTIKORKEAKOULU
Tietokone-elektroniikka
Syksy 2006
Janne Norolampi

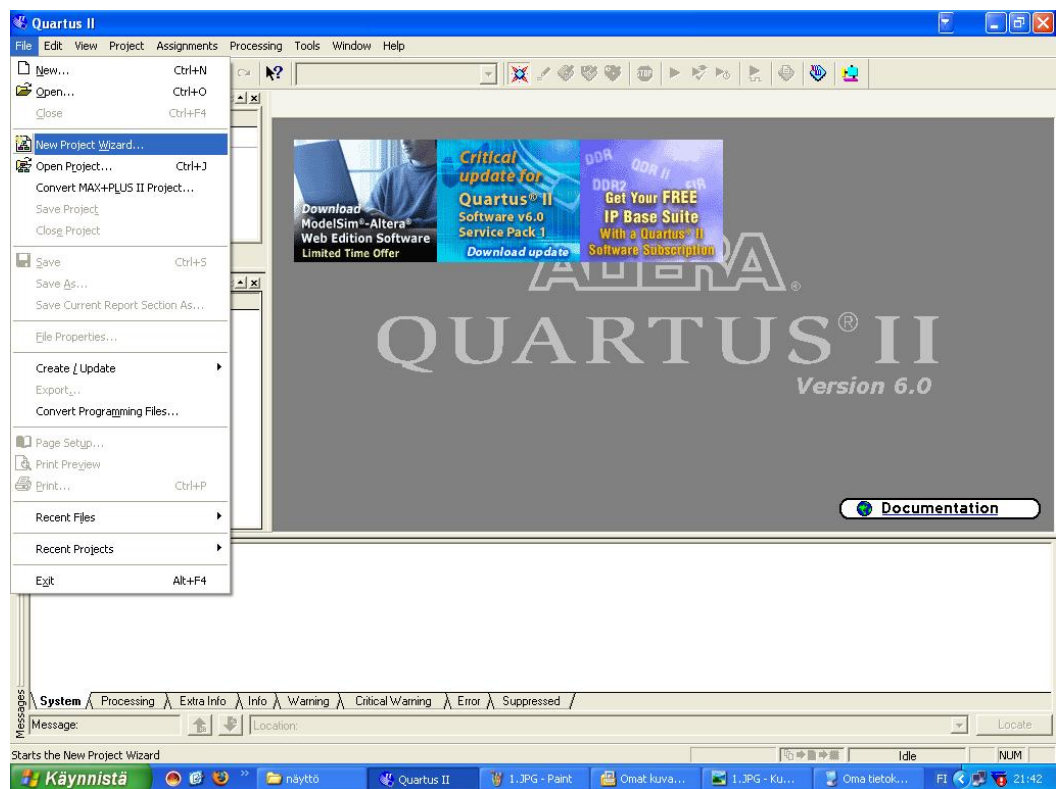
Nämä ohjeet ovat tarkoitettu käytettäväksi Alteran kehitysalustan kanssa, joka on rakennettu EPM7064S44 piirin ympärille. Esimerkki kuvasarja Quartuksen käyttöönotosta on tehty niin, että VHDL-koodi on jo valmiiksi kirjoitettu ja se halutaan simuloida, kääntää ja siirtää piirille. Koodin tulee olla tallennettu muodossa.vhd. Koodin voi kirjoittaa vaikka notepadilla kunhan tallentaa sen haluttuun muotoon. Koodin muokkaaminen onnistuu kyllä Quartuksella. Ohjeita soveltamalla voidaan käyttää mitä tahansa kehitysalustaa tai piiriä.

Kehitysalustalla on kaksi 7-segmentti näyttöä, 4kpl ledejä ja 12-Bittinen A/D-muunnin, ohjelmoinnin harjoittelua varten. Lisäksi on 4kpl kytkimiä ja mahdollisuus lisätä kytkimiä tai muita laitteita, kun jumppereilla kytketään ledit ja jo olemassa olevat kytkimet pois käytöstä. Virransyöttöön käytetään 6-9 voltin virtalähdettä jossa on 2,5mm pistoke, maa on pistokkeen ulkokehällä.

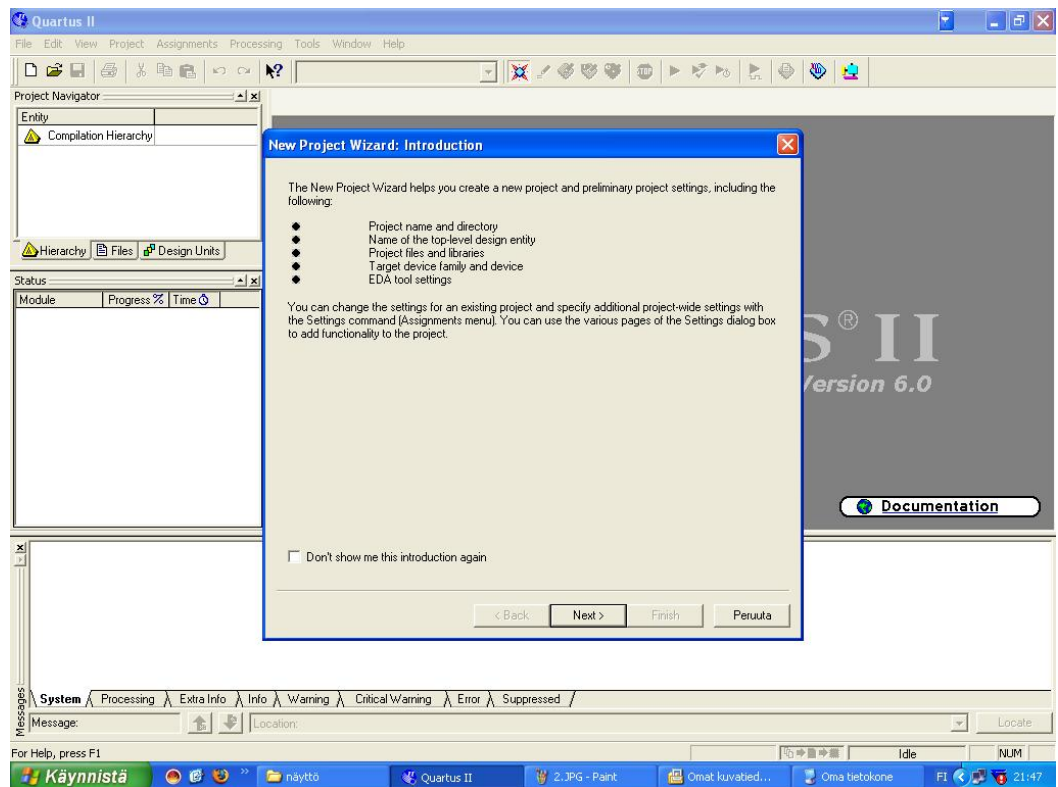




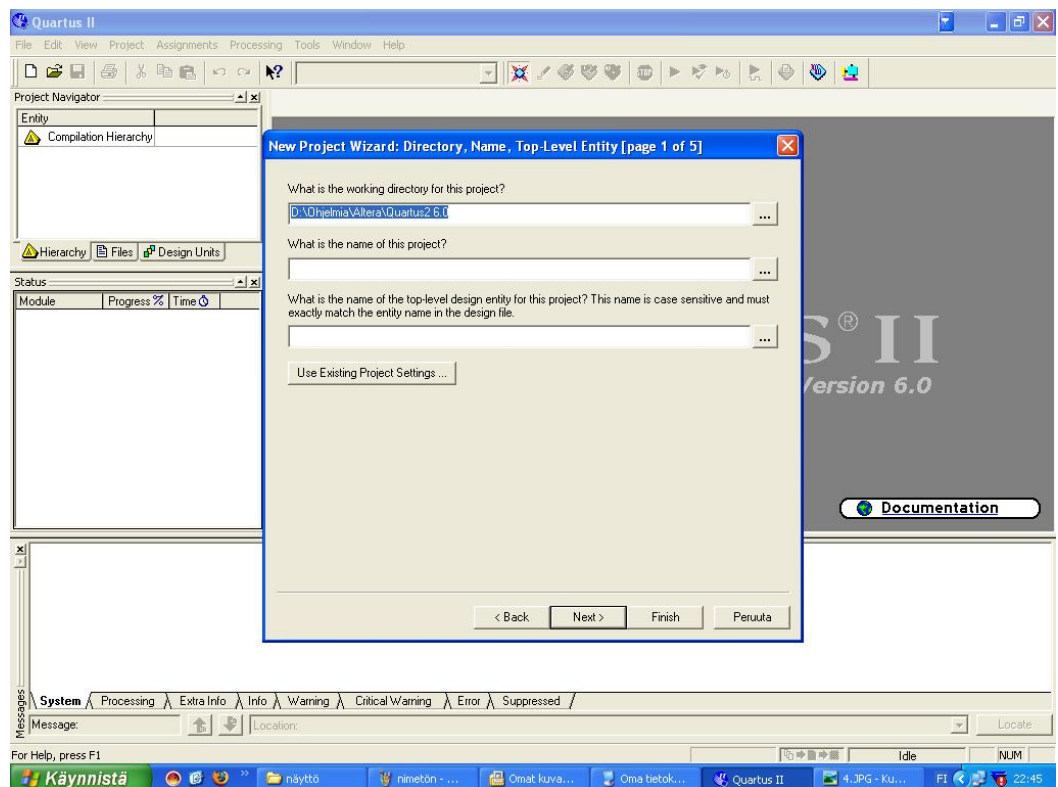
Kun Quartus II ohjelma avataan se näyttää tältä. Quartus tarkistaa avautuessaan saatavilla olevat päivitykset ja sen käynnistyminen saattaa hieman kestää.



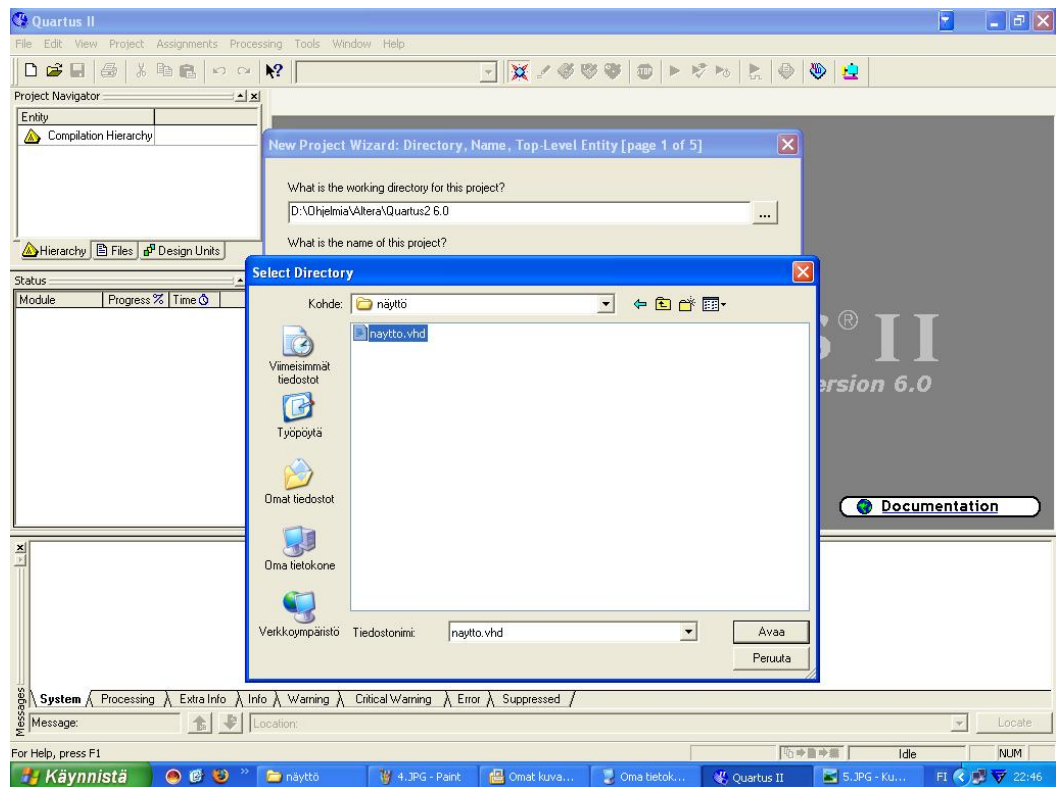
Ensimmäiseksi tehdään uusi projekti. Vasemmasta yläkulmasta valitaan File -> New Project Wizard...



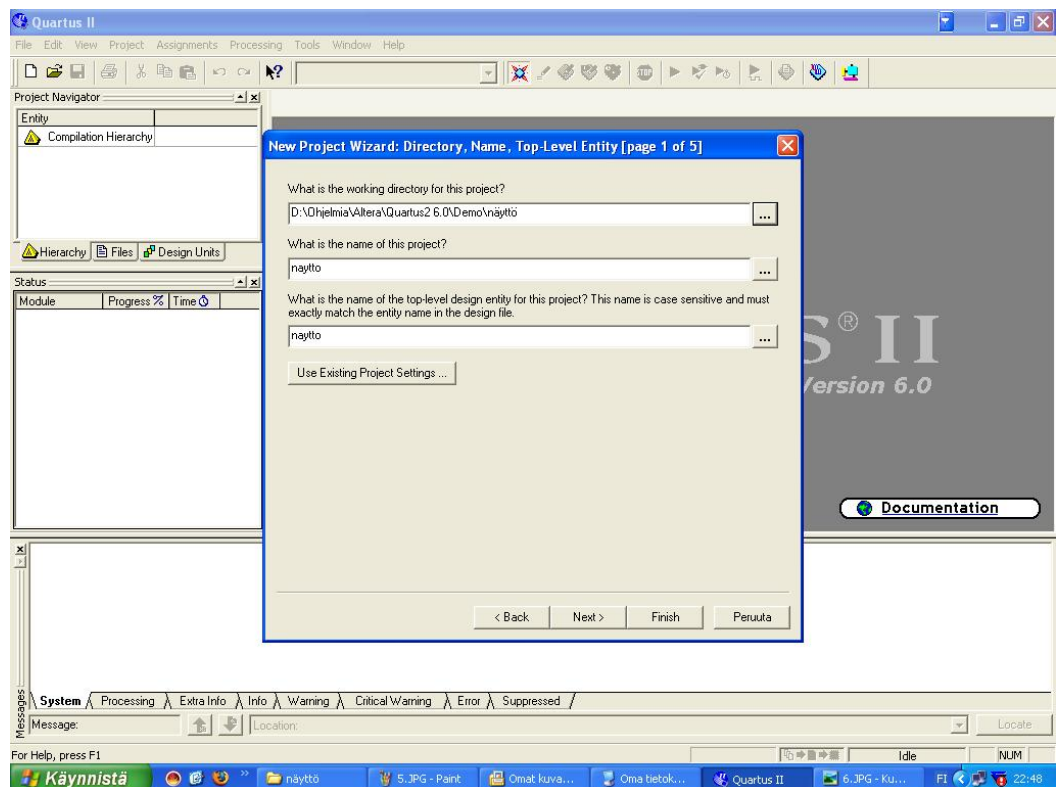
Näyttöön avautuu tämän näköinen ikkuna, josta painetaan ”Next >”



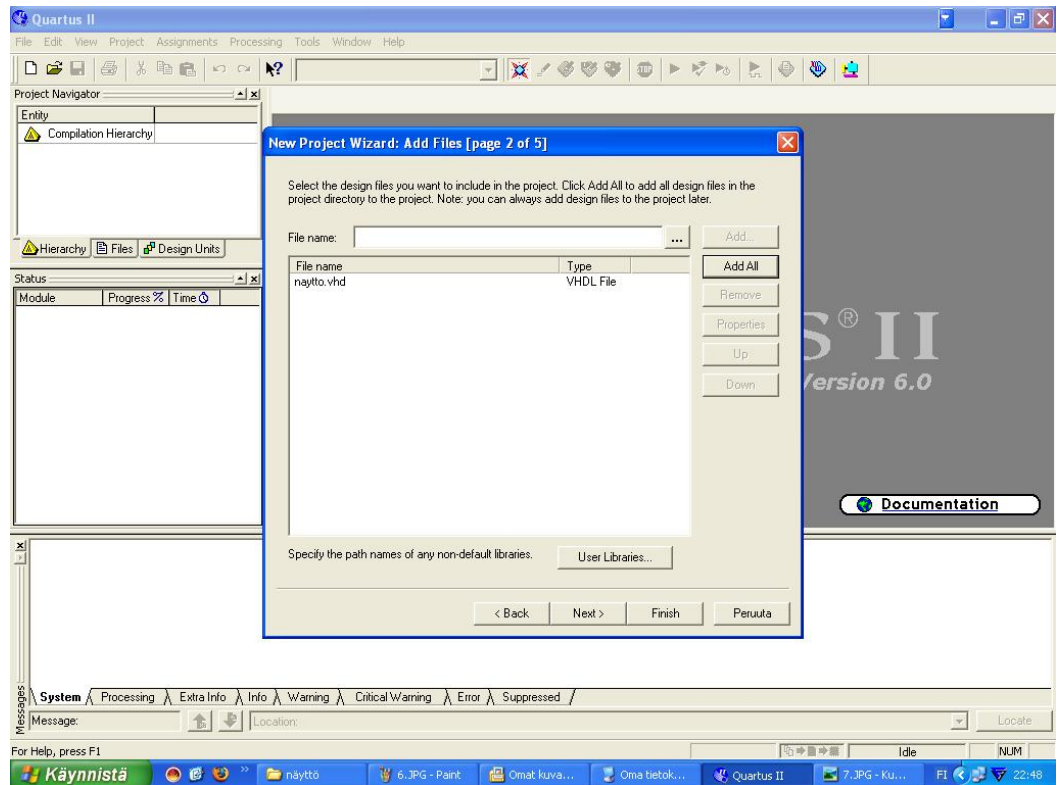
Seuraavaksi avautuu [page 1 of 5]. Oikeasta ylälaidasta painetaan ylintä, ”kolmen pisteen” kuvaketta.



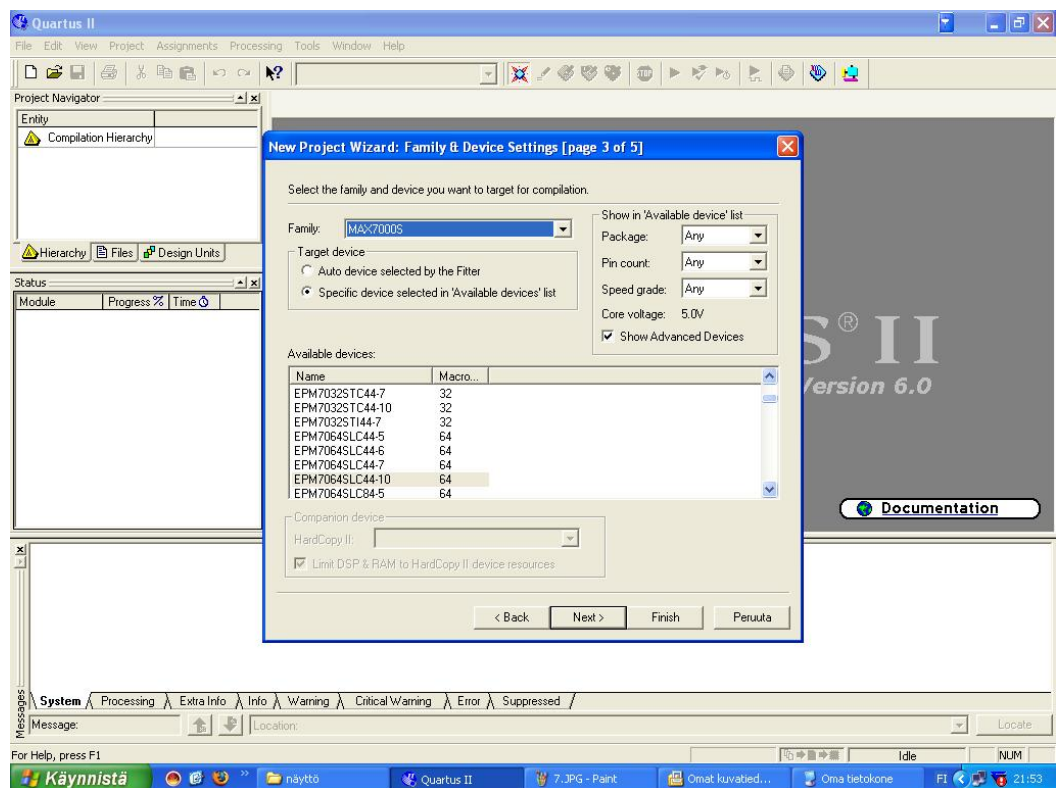
Valitaan avautuvasta ikkunasta kohde, jossa on VHDL-koodi jota on tarkoitus käyttää, tiedostomuoto on .vhd. Seuraavaksi Painetaan "Avaa".



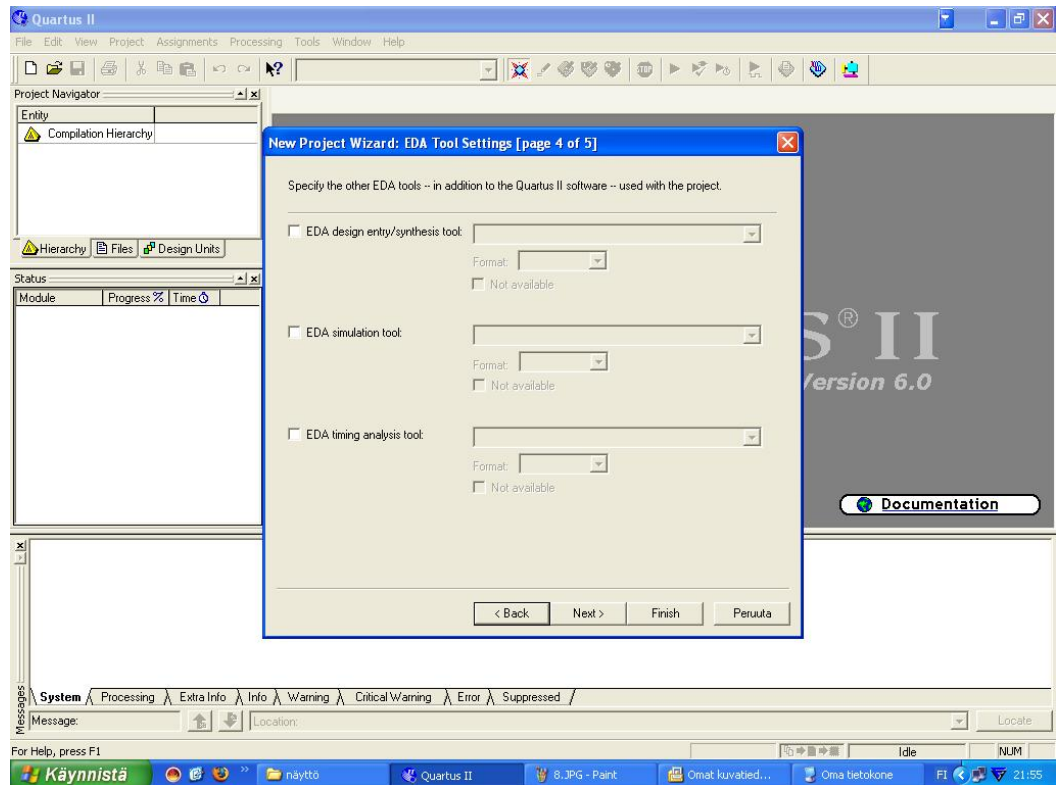
Projectin nimeksi tulee valitun koodin nimi. Painetaan "Next >"



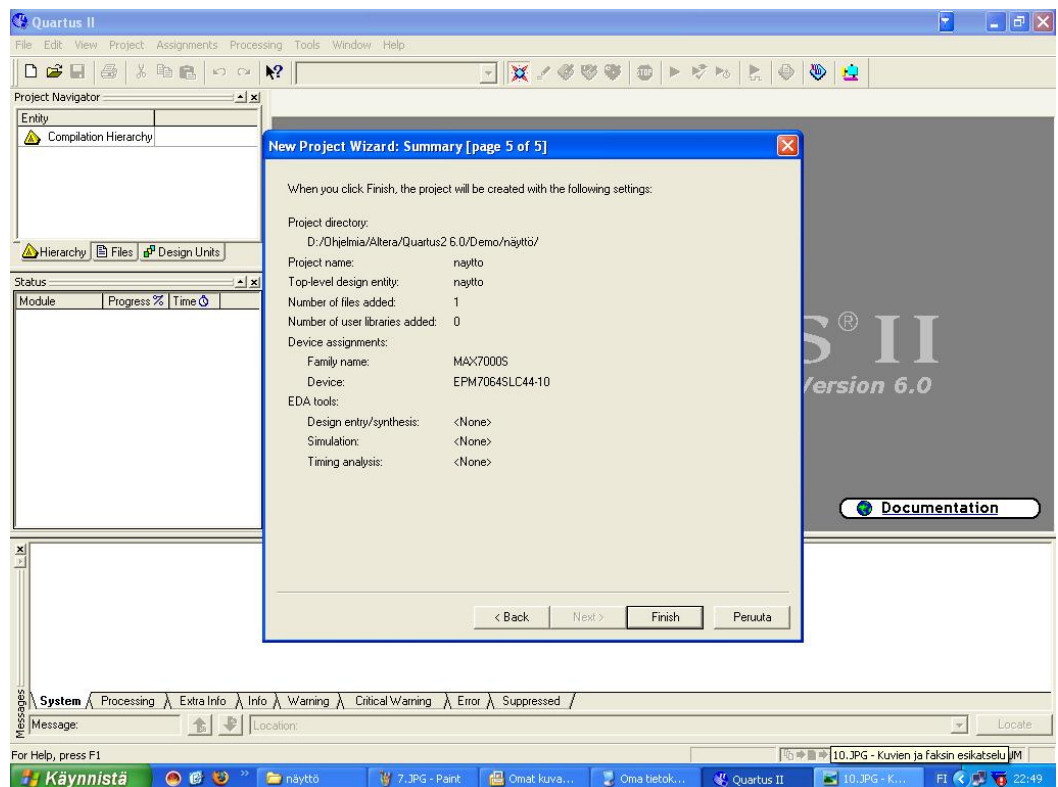
Avautuvasta ikkunasta [page 2 of 5], painetaan "Add All", jolloin .vhd tiedosto sisällytetään projectiin. Seuraavaksi painetaan "Next >"



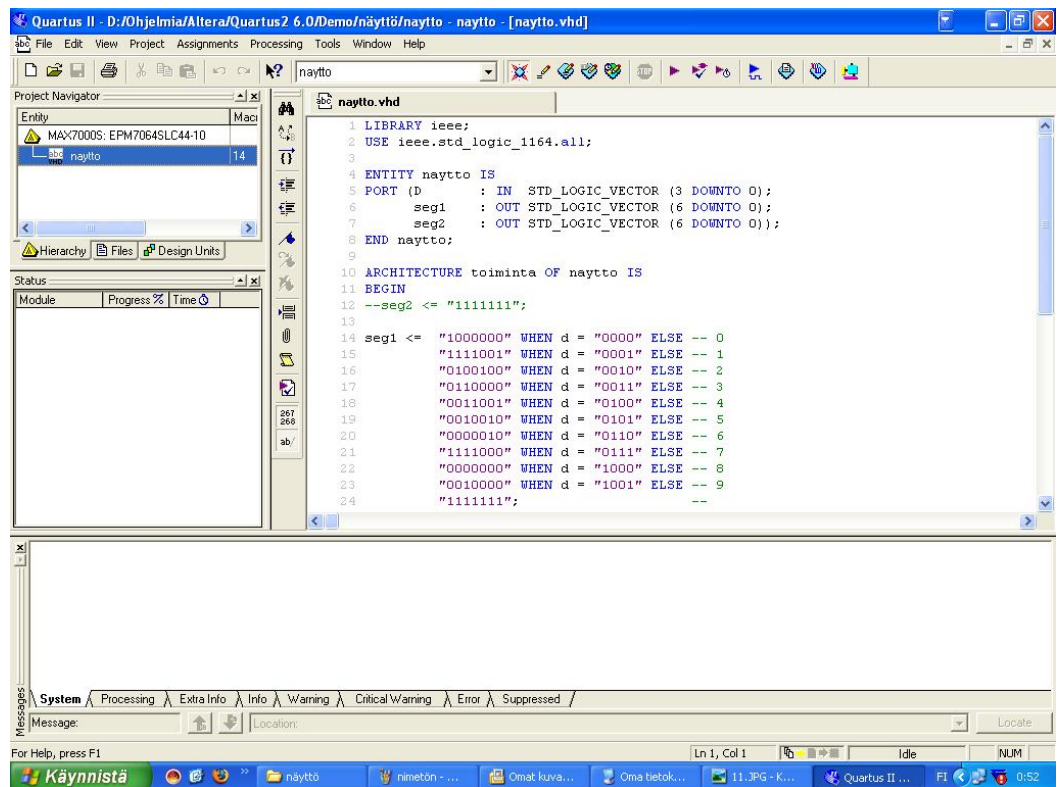
[Page 3 of 5] kohdassa valitaan käytettävä piiri. Family on MAX7000S ja tarkempi malli on EPM7064SLC44-10 ja kyseessä 64 Macrocellinen versio. Seuraavaksi painetaan "Next >"



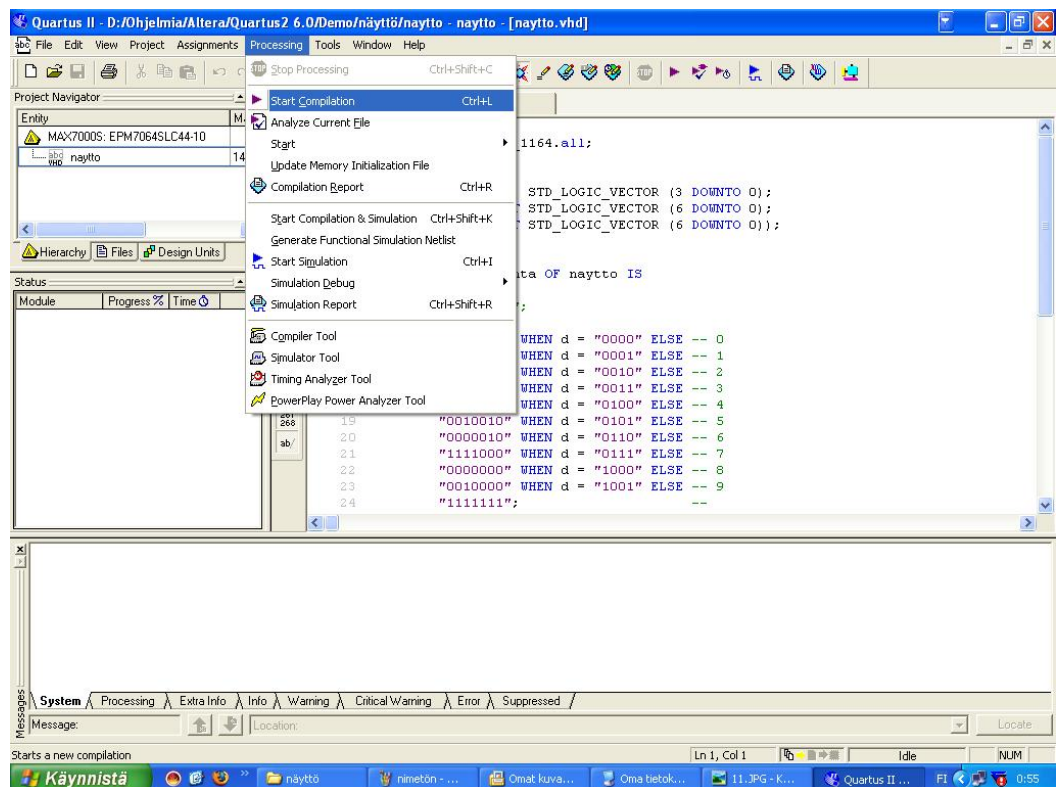
[page 4 of 5] painetaan vain "Next >"



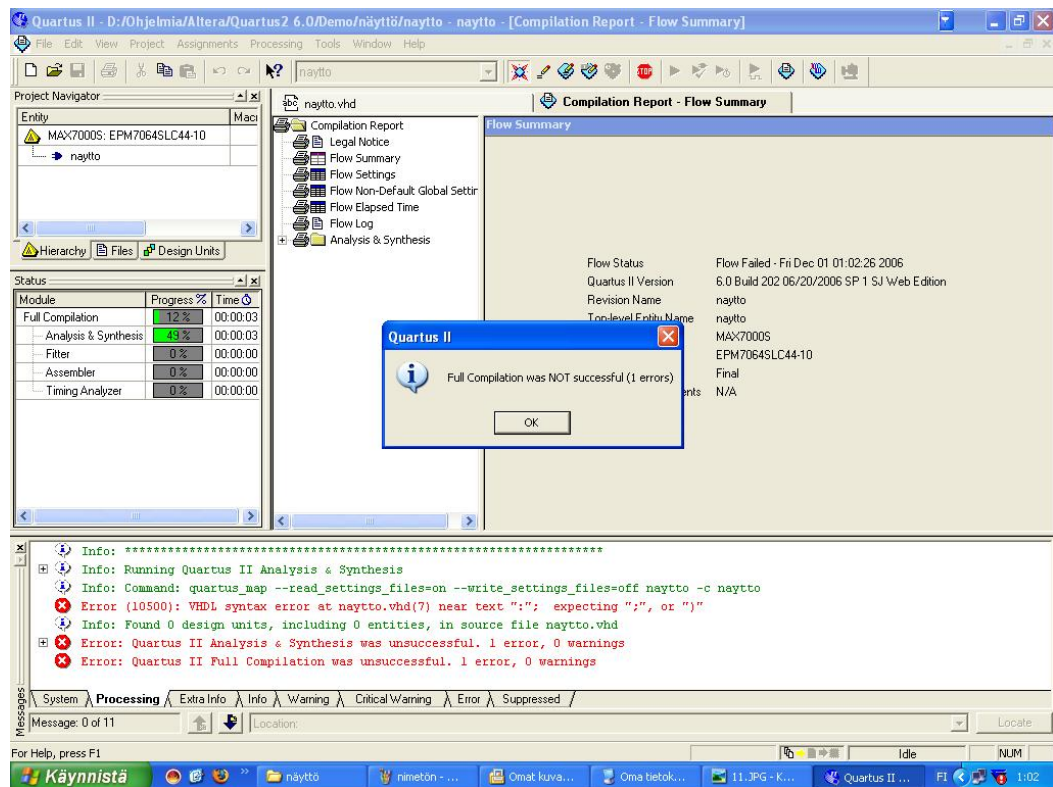
[page 5 of 5] nähdään yhteenveto käytettävästä projectista. Painetaan "Finish".



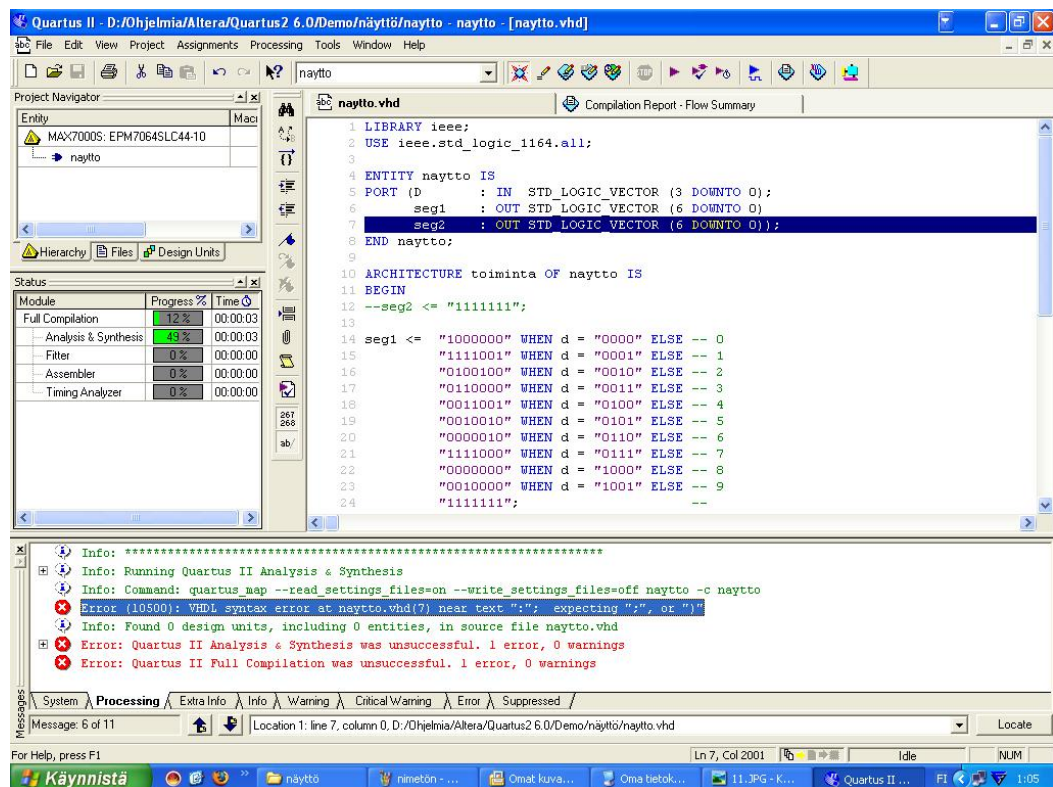
Vasempaan ylälaitaan ilmestyy Project Navigatorin Entity kohtaan luotu projecti, jota painamalla nähdään kirjoitettu VHDL-koodi. Tässä kohdassa sitä voi myös muokata halutunlaiseksi.



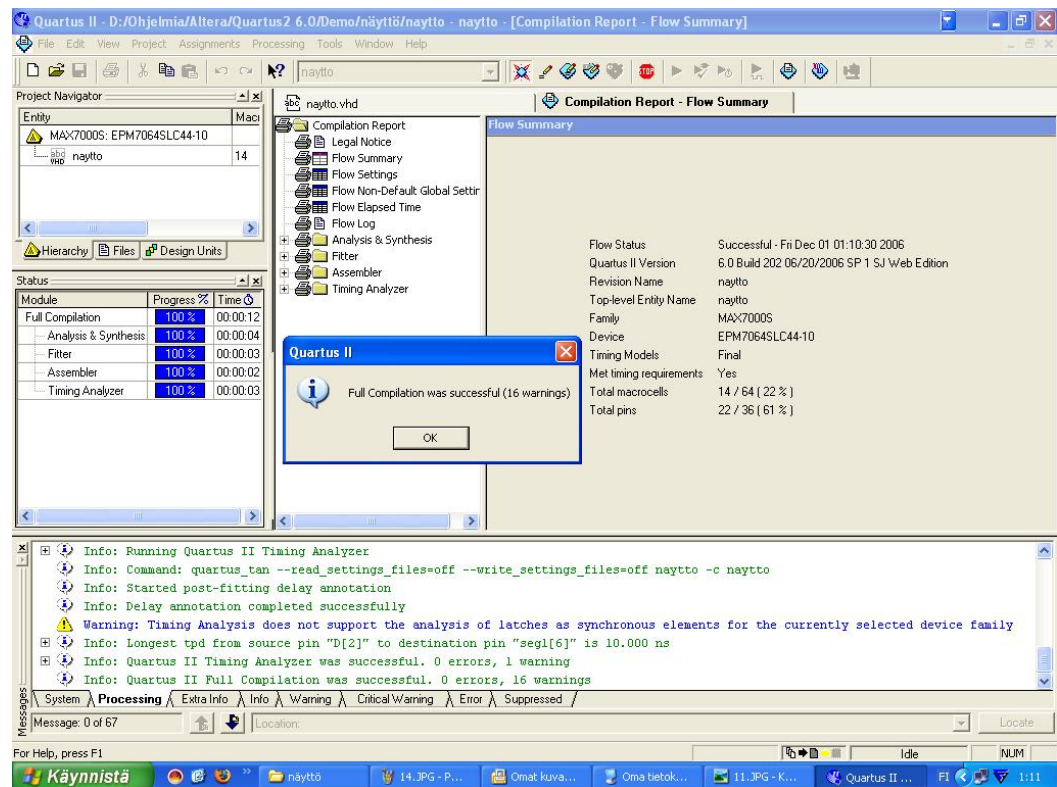
Seuraavaksi piiri voidaan koettaa kääntää. Valitaan "Processing -> Start Compilation".



Jos koodissa on virhe ohjelma ilmoittaa siitä kyllä. Virheet ja varoitukset näkyvät alhaalla olevasta ikkunasta.



Klikkaamalla virhettä ohjelma kertoo sen paikan koodissa, tässä tapauksessa riviltä 6 puuttuu puolipiste rivin perästä.



Kun virheet on korjattu ja koodi käännetty uudestaan. Quartus kertoo siitä paljon tietoja ja yhteen vedossa nähdään tämän ohjelman vievän 14 macrocellia käytettävistä 64:stä. I/O-liitäntöjä kulutetaan 22 kpl, kun niitä on kaikkiaan käytössä 36 kpl.

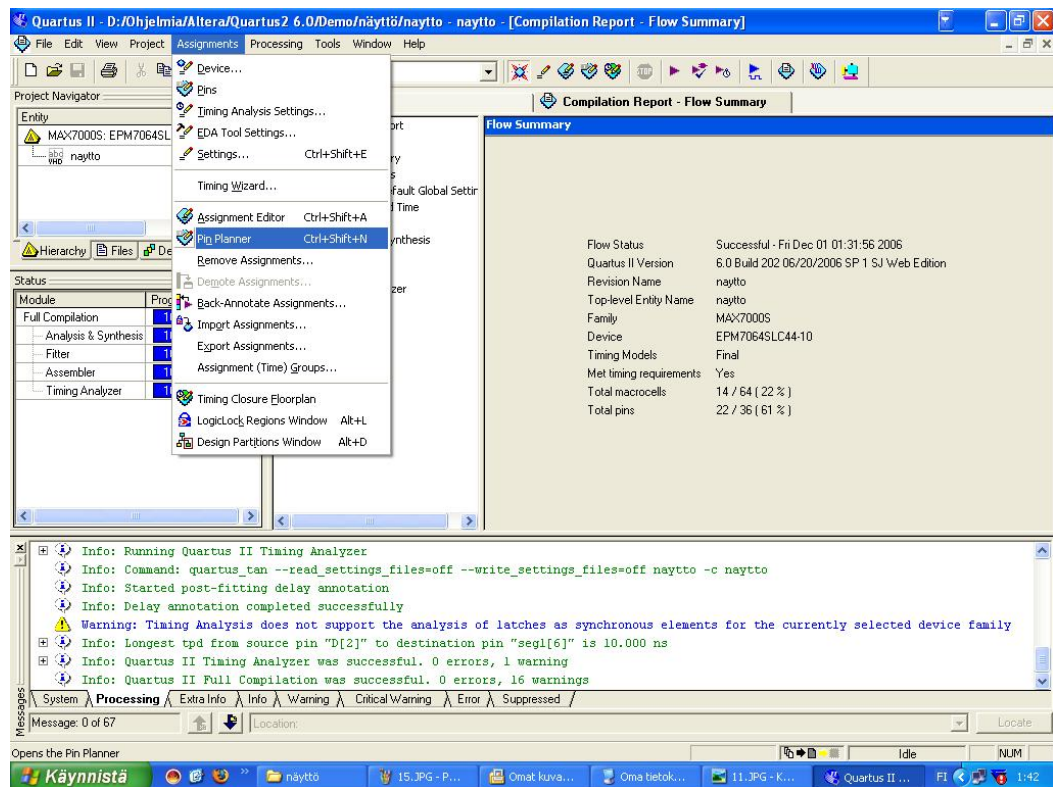
Tässä esimerkissä tehtiin kaikki käännökseen vaiheet kerralla, ne voidaan kuitenkin tehdä tarvittaessa erikseenkin. Vasemmassa laidassa "Status" ikkunasta nähdään neljä erivaihetta:

Analysis & Synthesis vaiheessa käydään koodi läpi ja raportoidaan mahdollisista virheistä. Synthesis muodostaa yhteisen tietokannan tiedostoille.

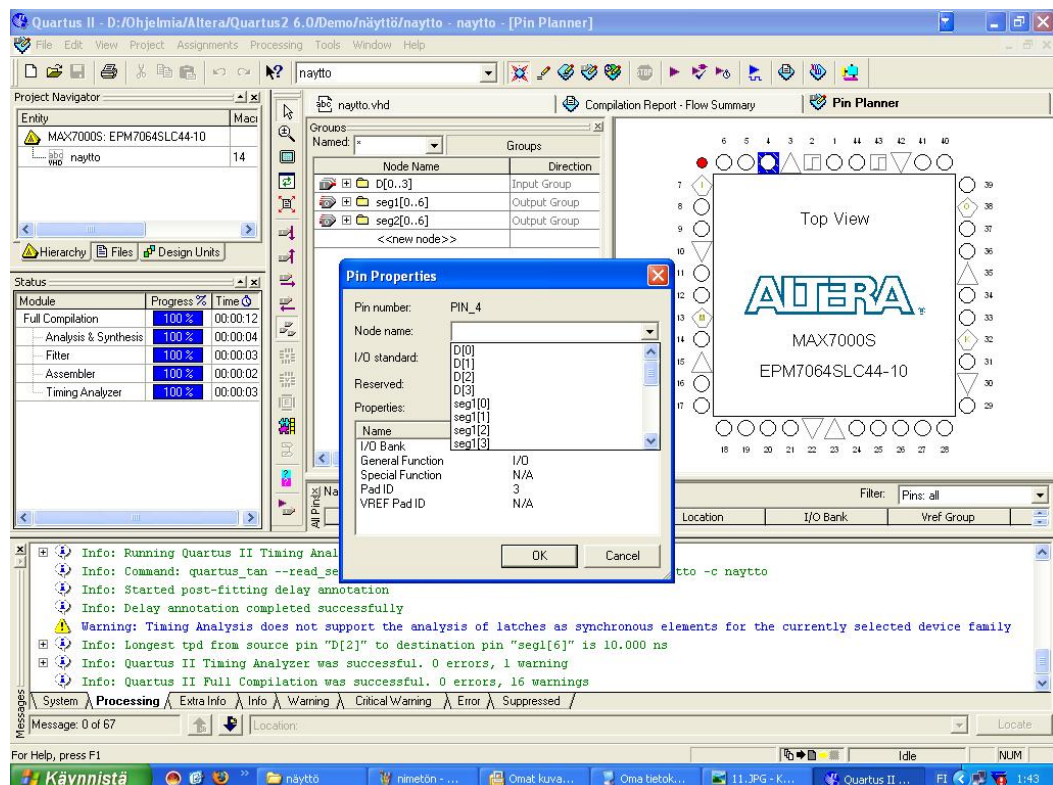
Fitter sovittaa projectin piirille sopivaksi.

Assembler tekee tarvittavan tiedoston, joka ladataan piirille.

Timing Analyzer selvittää signaalien viiveet ja raportoi niihin mahdollisesti liittyvät virheet.



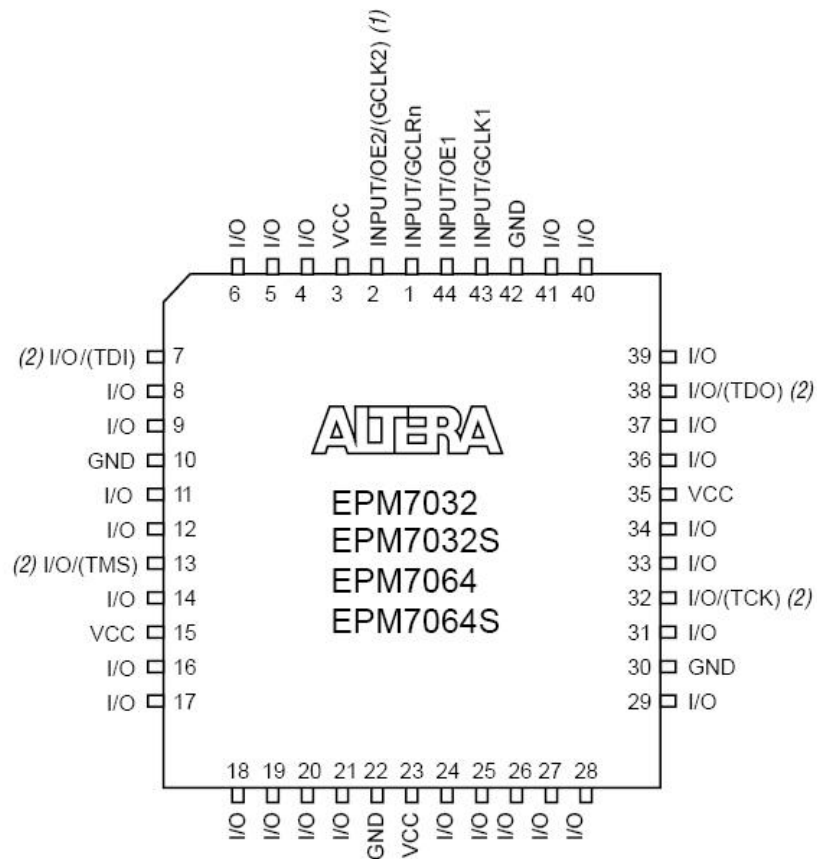
Seuraavaksi täytyy kertoa ohjelmalle, mikä piirin I/O-liitännöistä on mikäkin. Toiminto löytyy kohdasta Assignments -> Pin Planner.



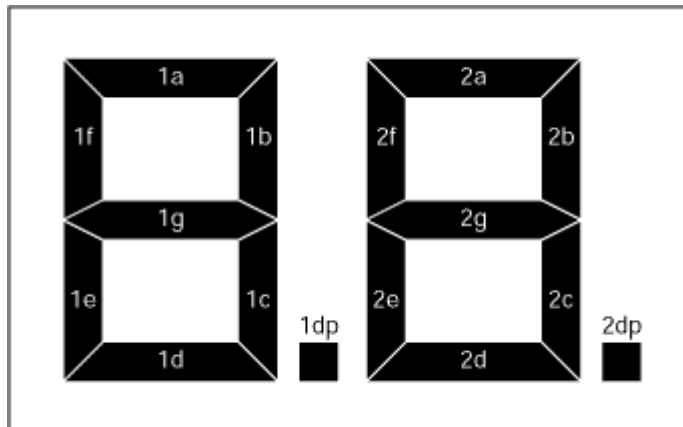
Tupla klikkaamalla jotain pinniä, aukeaa "Pin Properties" ikkuna, jossa "Node name" kohtaan laitetaan haluttu I/O-signaalin nimi. Tässä kehitysalustassa olevan PIN_4-signaalin oikea kytkentä on seg1[4]. I/O-signaaliit kytkeytyvät seuraavasti tässä kehitysalustassa:

7064 Pin	Tyyppi	Näyttö 1	Näyttö 2	Ledit	Kytkimet	A/D	Muuta
1	VCC						
2	VCC						
3	VCC						
4	Output	e					
5	Output	d					
6	Output	c					
7	TDI						Lataus
8	Output	piste					
9	Output	b					
10	GND						
11	Output	a					
12	Output	f					
13	TMS						Lataus
14	Output	g					
15	VCC						
16	Input					DCLOCK	
17	Input					Dout	
18	Input					CS/SHD N	
19	Input				4		Jumpperi
20	Input				3		Jumpperi
21	Input				2		Jumpperi
22	GND						
23	VCC						
24	Input				1		Jumpperi
25	Output			1			Jumpperi
26	Output			2			Jumpperi
27	Output			3			Jumpperi
28	Output			4			Jumpperi
29	Input						VCC/GN D
30	GND						
31	Output		piste				
32	TCK						Lataus
33	Output		b				
34	Output		c				
35	VCC						
36	Output		a				
37	Output		f				
38	TDO						Lataus
39	Output		d				
40	Output		g				
41	Output		e				
42	GND						
43	VCC						
44	VCC						

Pinnit 19-28 (pois lukien 22-23) voidaan jumppereiden avulla kytkeä, myös ulkoiseen laitteeseen. Ne voivat olla joko inputtaja tai outputteja. Pinnin 29 avulla voidaan piirille syöttää tarvittaessa VCC tai GND jumpperin avulla.

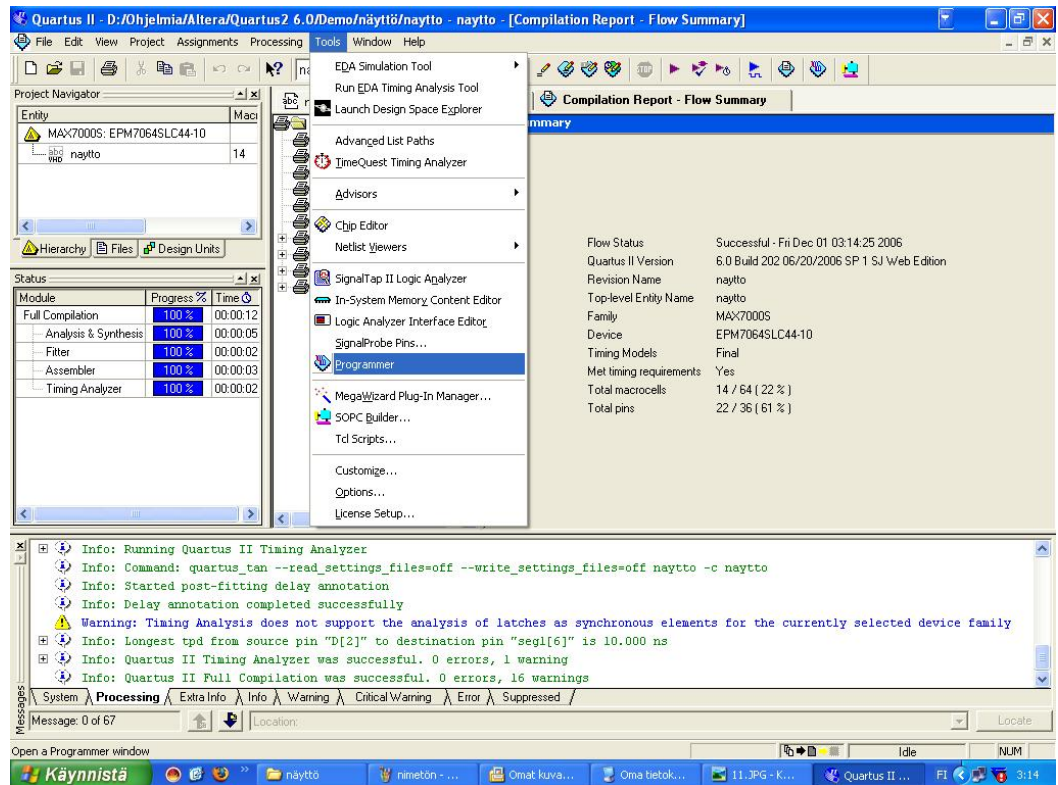


Piirin nastoja voi vertailla yllä olevaan taulukkoon ja etsiä kuhunkin koodiin sopivat pinnit.

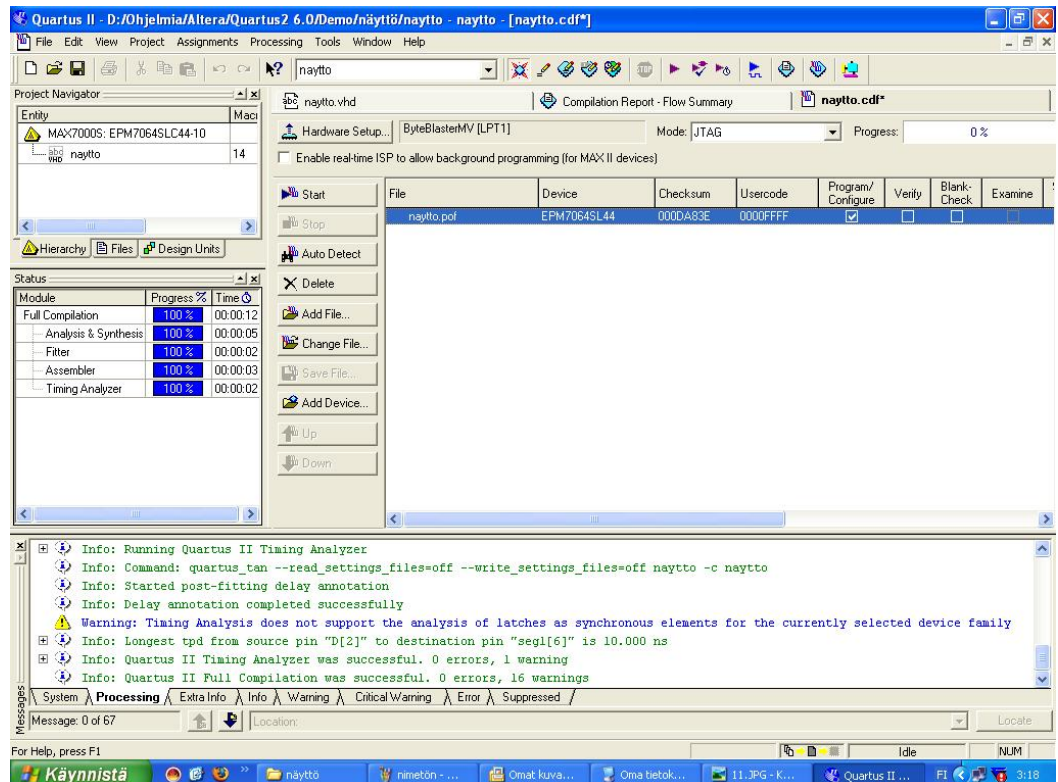


Näyttöjen pinnit ovat siis seuraavat (poimittu em. taulukosta):

Näyttö	7064 Pin näyttö 1	7064 Pin näyttö 2
a	11	36
b	9	33
c	6	34
d	5	39
e	4	41
f	12	37
g	14	40
piste	8	41



Pinnien paikalleen laittamisen jälkeen siirrytään lataamaan ohjelmaa kehitysalustalla sijaitsevalle piirille. Käynnistetään Tools -> Programmer.



Auennesta ikkunasta ylhäältä Hardware Setupista valitaan tarvittaessa ohjelmointikaapeliksi ByteBlasterMV ja Modeksi laitetaan JTAG. Piirille ladattava tiedosto on tyypiltään .pof. Program/Configure kohtaan laitetaan ruksi ja painetaan "Start", jolloin Progress kohdassa rupeaa prosentit juoksemaan ja tiedosto siirtyy piirille. Ladattu ohjelma toimii kehitysalustalla seuraavasti:

Kytkimet 4,3,2,1	Näytön näyttämä luku
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9

Eli kun painetaan kytkintä 1, näyttöön tulee lukema "1". Vastaavasti esimerkiksi numero 7 saadaan näkyviin näytölle, kun painetaan samanaikaisesti kytkimet 1, 2 ja 3 pohjaan. Kytkimillä täytyy ikään kuin saada aikaan binaariluku, jonka arvoa vastaava numero tulee palamaan näyttöön.

Simulointitiedosto luodaan seuraavasti: File -> New -> Other Files -> Vector Waveform File. Lisää simuloinnista löytyy Quartuksen "Help" osiosta.

HUOM! Muista kytkeä ByteBlaster-kaapeli kehitysalustalle niin päin, että punainen raita tulee etusivulla näkyvän kuvan JTAG-liittimen, numero 1 kohdalle.