

TARTU ÜLIKOOL  
LOODUS- JA TEHNOLOOGIATEADUSKOND  
Tehnoloogiainstituut

**Tarmo Kolsar**

**Suunitlus riistvaraplatvormi valmiseks  
sõltuvalt lähteülesandest**

**Bakalaureusetöö (12 EAP)**

Juhendaja: M. Sc. Margus Rosin

Kaitsmisele lubatud: .....

Juhendaja: .....

allkiri, kuupäev

Tartu 2014

## Sisukord

1 Sissejuhatus.....	3
2 Platvormid.....	4
2.1 7400 seeria kiibistik.....	4
2.2 FPGA .....	4
2.3 Atmel AVR .....	4
2.4 x86.....	5
3 Ülesanded.....	6
3.1 Hindamiskriteeriumid .....	6
3.2 Loogikatehe.....	6
3.3 Registrate töötlemine.....	7
3.4 Fibonacci arvude leidmine.....	7
3.5 Jõumeetodil väärtuse tuvastamine .....	8
4 Vahendid.....	9
4.1 7400 simuleerimine.....	9
4.2 Arduino Leonardo.....	9
4.3 Basys2.....	10
4.4 x86.....	11
5 Tulemused.....	12
5.1 Programmeerimiseks nõutavad oskused .....	16
6 Kokkuvõte.....	18
Tsiteeritud teosed.....	19
Summary in English.....	21
Lisad .....	22
Terminid .....	23
Litsents.....	24

## 1 Sissejuhatus

Tänapäeval on palju võimalusi luua riistvaralahendusi oma elu lihtsustamiseks ja automatiseerimiseks. Olgu selleks automaatne lilledekastja, mis regulaarselt kastab taime ja lülitab valgustust, või kaameraga lennumasin, mis suudab tuules end stabiilselt hoida. Kui inimesel tekib idee või vajadus iseseivalt planeerida ja koostada riistvaraprojekt, võib tekkida probleem kõikide võimalike mikrokontrollerite seast projektiks sobiliku valimisega. Mikrokontrolleri vahetamine keset arendusprotsessi on vägagi ajakulukas. Seega on tähtis alustada sobilikuima riistvaraplatvormiga. Õige riistvaraplatvormi valimisele proovib järgnev lõputöö anda kätte soovitusliku suuna.

Bakalaureusetöö ülesehitus on kaheosaline. Esimeses osas on loodud katsed, mis esindavad traditsioonilisi, igapäevaselt riistvarale püstitatud ülesandeid. Neid ülesandeid on katsetatud erinevatel platvormidel, mis on antud lõputöö raames vaadeldavad. Teises osas on uuritud valitud riistvaraplatvormide tugevusi ja nõrkusi. Samuti on loodud ülevaatlik hinnang milline riistvaraplatvorm sobib lähteülesannete lahendamiseks kõige paremini.

Töö koosneb kuuest peatükist: sissejuhatausest, platvormide tutvustamisest, ülesannete püstitamisest, ülesannete jaoks valitud riistvaralahenduste kirjeldusest, tulemuste võrdlemisest ja kokkuvõttest. Platvormide tutvustamise peatükis uuritakse olemasolevaid riistvaralahendusi. Riistvaralahenduste peatükis võetakse vaatluse alla iga platvormi esindav riistvara ja nende spetsifikatsioonid. Ülesannete peatükis luuakse sobivad ja mitmekülgsed harjutused. Tulemuste peatükis tuuakse välja riistvara peal jooksutatud ülesannete tulemused ja nendest tehtud järeldused. Programmid, mis koostati seoses ülesannetega, on kaasatud CD plaadiga.

## 2 Platvormid

Antud töö raames valiti välja neli uuritavat platvormi. Järgnevad valikud tehti kuna need on nii Tartu Ülikoolis õpetatavates kavades kui ka üldiselt eelistatud platvormid tootearengus. Nendeks on: 7400 seeria TTL kiibistik, FPGA, Atmel AVR tehnoloogial loodud mikroprotsessorid ja x86 käsustikul loodud arhitektuur.

### 2.1 7400 seeria kiibistik

Tegemist on kõige levinumate transistor-transistor loogika mikrokiipidega<sup>1</sup>, mis võivad endas sisaldada alates lihtsamatest loogika väratitest kuni keerulisemate aritmeetika-loogikaplokkideni.

Töö raames valiti 7400 seeria mikrokiibid nende lihtsuse, ajaloolise väärtuse ja kasutuse eluea ning sellest tuleneva laia kasutajatoe tõttu. Esimene kaheksa NAND väratiga 7400N loodi 1964. aastal ja omab tänapäevaks sadu kasutusesolevaid sugulasi. Ülesannete lahendamisel ei ole kasutusel programmeerimiskeeli, kuna lahendus koosneb kiipide õiges loogilises järjekorras omavahel ühendamises. Küll aga on lahenduse disainimiseks kasutusel vastav tarkvara. Ühenduste struktuuri loomiseks on tarvis osata loogika värateid ja nende tõeväärtustabeleid.

Erinevate kiibistikega on võimalik piiratud oskustega luua madala keerukusega rakendusi, mis on kulude poolest soodsad ja tarbivad vähe voolu. Kõrgema keerukuse puhul kasvab mikrokiipide arv kiiresti, kuna need on seotud otseselt loogikalülituste arvuga.

### 2.2 FPGA

FPGA (*Field-programmable gate array*) ehk programmeeritav ventiilmaatriks on mitmekülgne mikroskeem, mis lubab kasutajal luua oma konfiguratsiooni vastupidiselt tehases tehtud sätetele. Skeemi töö määrab kasutaja riistvara kirjeldamise keelega. FPGA koosneb programmeeritavatest loogikakomponentidest, mida kutsutakse loogikaplokkideks, ja seadistatavatest vaheühendustest, mille abil saab loogikaplokke omavahel kokku ühendada. Kõige väljapaistvamalt lahendatakse FPGAdega paralleelarvutus ülesandeid nagu videotöötlus, kõnetuvastus või krüptograafia.

---

<sup>1</sup><http://www.computerhistory.org/semiconductor/timeline/1963-TTL.html>

## 2.3 Atmel AVR

Atmel AVR tehnoloogia töötati välja 1996 aastal Norra Tehnoloogiaülikooli tudengite poolt. Esimene töötav kiip loodi 1997 aastal nimetusega AT90S1200. AVR arhitektuur oli esimene mikrokiip, mis kasutas välmälu andmete salvestamiseks. Lühendil AVR ei ole loojate järgi tähendust.<sup>2</sup> Atmeli eelised seisnevad madalas voolutarbes ja kuluefektiivsuses. Antud lõputöös keskendutakse megaAVR mikrokontrolleritele, mis on välja töötatud suurte andmemahutudega opereerimiseks. Väljapaistvamate omaduste juurde kuuluvad lai toodete perekond, *picoPower* tehnoloogia, mis võimaldab väga madalat voolutarbimist, suur integratsioon, analoogfunktsionaalsus ja kiire arendus läbi *on-chip* silumise.<sup>3</sup>

## 2.4 x86

x86 arhitektuuri all mõeldakse Intel 8086 protsesoritele loodud käsustike perekonda.<sup>4</sup> 8086 disainimist alustati 1978. aasta mais Stephen Morse juhtimisel. Algselt loodud manussüsteemides ja personaalarvutites töötamiseks on x86 tänapäeval kasutuses ka serverites ja tööjaamades. Loodud arhitektuurile on oma leviku tõttu üks suuremaid valikuid programmeerimiskeelte ja operatsioonisüsteemide kogu. Antud lõputöös on oletatud, et teoreetiline riistvaralahenduse looja ei hakka disainima riistvara tasemel oma projekti, vaid kasutab olemasolevaid riistvaraplatvorme. Kui projekt läheb masstootmisesse, saab teha uue kulude hinnangu spetsiifilise riistvaralahenduse loomiseks.

---

<sup>2</sup><https://www.youtube.com/watch?v=HrydNwAxbcY>

<sup>3</sup><http://www.atmel.com/products/microcontrollers/avr/megaavr.aspx>

<sup>4</sup><http://www.pcworld.com/article/146957/article.html>

## **3 Ülesanded**

Et saavutada ülevaade kõikide riistvara platvormide tugevuste ja nõrkuste kohta, on tarvis luua ülesanded, mis tooksid välja eri riistvaralahenduste omapärad. Kõikidele valitud lähteülesannetele ei ole koostatud igas valitud platvormil programme. Välja on jäetud olukorrad, kus konkreetse platvormi tarbeks programmi koostamine ületaks oluliselt kasutatava loogika mahu võrreldes teiste platvormidega või poleks võimalik täielikku funktsionaalsust sünteesida.

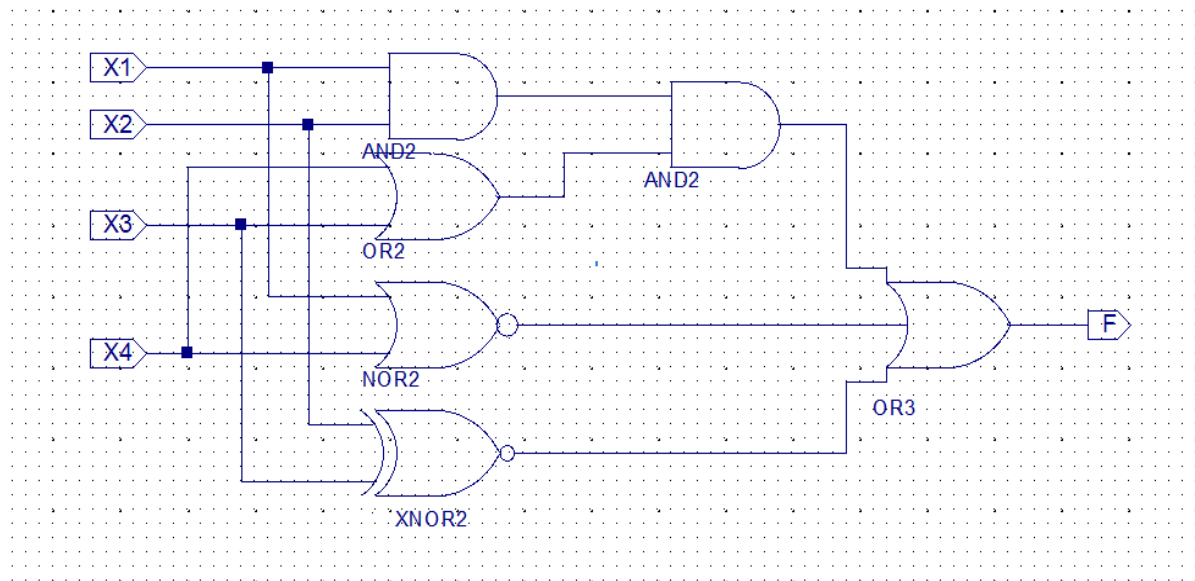
### **3.1 Hindamiskriteeriumid**

Et oleks võimalik luua järeltõlki ülesannetest, on vajalik mõõta riistvara sooritusi. Kuna platvormide kasutusvaldkonnad on väga erinevad, on tarvis mõõta erinevaid aspekte, et tuua välja eri platvormide tugevaid külgi. Peamisteks vaadeldavateks kriteeriumiteks sai antud lõputöö raames valitud:

- programmi keerukus.
- programmeerija vajavate oskuste tase.
- riistvara lahenduskiirus.
- platvormide umbkaudne maksumus.

### **3.2 Loogikatehe**

Osa igapäevaselt kasutusel olevast elektroonikast ei vaja kiiret arvutusvõimsust. Selle tõttu on üheks antud lõputöö raames vaadeldavaks ülesandeks teostada lihtne loogikatehte arvutus. Rõhk ei saa olema tulemuse arvutamise kiirusel, vaid ülesande lahendamise keerukusel ja milliseks kujuneb antud platvormi maksuvus. Tegemist on nelja sisendi ja ühe väljundiga funktsiooniga. Selline funktsionaalsus on näiteks kasutuses mitme lülitiga valgusinstallatsioonides, lihtsates liikumisandurites ja erinevates automaatustes jne. Loogikalülitustest on kasutusel AND, OR, NOR, ja XNOR väratid. Funktsionaalsuse loogikaskeem on toodud skeemil 1.



Skeem 1. Loogikatehte skeem.

### 3.3 Registrate töötlemine

Keskmise ja suurema keerukusega programmide puhul on kasulik, kui kasutatakse registreid, kuhu saab salvestada hetkeväärtusi tulevaseks töötlemiseks. Registrate kasulikkus asub hoitavate andmemahutude loomises, väärtused on kirjutatud mälli ja neid saab kasutada programmi järgnevates lõikudes ning pole vaja uuesti välja arvutada. Seetõttu on loodud ülesanne, mis esmalt salvestab ja seejärel kasutab mitmesse registrisse kirjutatud väärtusi. Ülesanne koosneb alljärgnevatest, järjestikku täidetavatest alamülesannetest:

- Kirjuta registrisse x1 väärtus 1010.
- Liiguta bitväärtust 2 kohta paremale ja salvesta see x2 registrisse.
- Kirjuta kolmandasse registrisse x3 väärtus 0101.
- Liida x2-le x3 ja lahuta saadud vastusest x1.
- Muuda x2 kõige madalam bit üheks.
- Muuda x2 väärtus vastupidiseks.
- Väljasta x2.

### 3.4 Fibonacci arvude leidmine

Tänapäeval on trend liikuda suurema ja kiirema arvutusvõimsuse suunas.<sup>5</sup> Sellise vajaduse testimiseks erinevatel riistvaraplatvormidel on tarvilik luua ülesanne, mis nõuaks eeskätt võimalikult kiiret arvutusjõudlust. Üheks ülesandeks on antud lõputöö raames valitud Fibonacci jada arvude arvutamine. Fibonacci jada on arvude jada mille iga liige, välja arvatud esimese kahe arvu puhul, on eelmiste liikmete summa. Mõõdetakse, mitmenda tsükklini jõuab programm kolme sekundi jooksul.

### 3.5 Jõumeetodil väärtuse tuvastamine

Tegemist on 32biti pikkuse järjendiga, mis sisaldab endas binaararvu. Ülesanne on jõumeetodiga tuvastada järjend. Antud ülesannet on võimalik lahendada erinevalt, kuid parim juhtum oleks, kui otsitav salasõna jagatakse osadeks ja tuvastatakse paralleelselt. Praegusel juhul ei kasuta x86 arhitektuur riistvara puhul rohkem kui ühte lõimu. Rohkemad lõimud võimaldavad paralleeltöötlust, kuid antud lõputöö raames neid ei kasuta, kuna kõikidel riistvaralahendustel pole mitme tuuma tuge.<sup>6</sup> x86 puhul lahendatakse ülesanne sarnaselt FPGAga, otsitava väärtuse ala jagatakse osadeks, kuid kasutades lineaartöötlust, lahendatakse nende osade tuvastamine üksteise järel, mitte paralleelselt. Reaalne kokkupuude antud ülesandel päriselu kasutuses oleks paroolide jõumeetodil ära arvamine.

---

<sup>5</sup>[https://www.wcu.edu/ceap/houghton/readings/technology\\_trends.html](https://www.wcu.edu/ceap/houghton/readings/technology_trends.html)

<sup>6</sup><http://preshing.com/20120208/a-look-back-at-single-threaded-cpu-performance/>

## 4 Vahendid

### 4.1 7400 simuleerimine

Xilinx ISE visuaalprogrammeerimise liideses on kasutajal võimalik disainida 7400 seeria mikrokiipide arhitektuuriga sarnane loogikaahel. Uurimustööks vajaliku järelduse tegemiseks piisas, kui disain loodi Xilinx ISE keskkonnas ja ülejäänud vajalik informatsioon saadi 7400 seeria dokumentatsioonist. Võrdlemine käis 7400N mudeliga millel on 8 NAND väratit.

### 4.2 Arduino Leonardo

Atmel ATmega arhitektuuri katsetamiseks on valitud Arduino Leonardo arendusplaat.<sup>7</sup> Leonardol on Atmega32U4 mikrokiip, mis töötab 16MHz taktsagedusel. Lisaks on veel sisseehitatud USB side tugi eemaldades vajaduse sekundaarsele protsessorile ja võimaldades luua virtuaalse COM port ühenduse. Riistvaraplatvorm võib seetõttu näida ühenduses olevale arvutile kui arvutihiire või klaviatuurina. Järgneval tabel on Leonardo spetsifikatsioon:

<b>Mikrokontroller</b>	<b>ATmega32u4</b>
<b>Töötamispinge</b>	<b>5V</b>
<b>Sisendpinge (soovituslik)</b>	<b>7-12V</b>
<b>Sisendpinge (limiit)</b>	<b>6-20V</b>
<b>Digitaalsed I/O Pinnid</b>	<b>20</b>
<b>PWM kanaleid</b>	<b>7</b>
<b>Analoog sisendkanaleid</b>	<b>12</b>
<b>DC vool I/O Pinnile</b>	<b>40mA</b>
<b>DC vool 3.3V Pinnile</b>	<b>50mA</b>
<b>Flash memory</b>	<b>32KB (ATmega32u4) millest 4KB on Bootloaderil</b>
<b>SRAM</b>	<b>2.5KB (ATmega32u4)</b>
<b>EEPROM</b>	<b>1KB (ATmega32u4)</b>
<b>Taktsagedus</b>	<b>16MHz</b>

Tabel 2. Arduino Leonardo spetsifikatsioonid

Kasutatav tarkvara on Arduino poolt loodud spetsiaalne integreeritud arenduskeskkond Arduino IDE versioon 1.0.5 millel on Leonardo mudelit toetav tugi. Programmeerimiseks kirjutatati antud uurimustöö raames *Wiring* põhjal baseeruvat C/C++ implementeeritud programmeerimiskeelt.<sup>8</sup>

<sup>7</sup><http://arduino.cc/en/Main/arduinoBoardLeonardo>

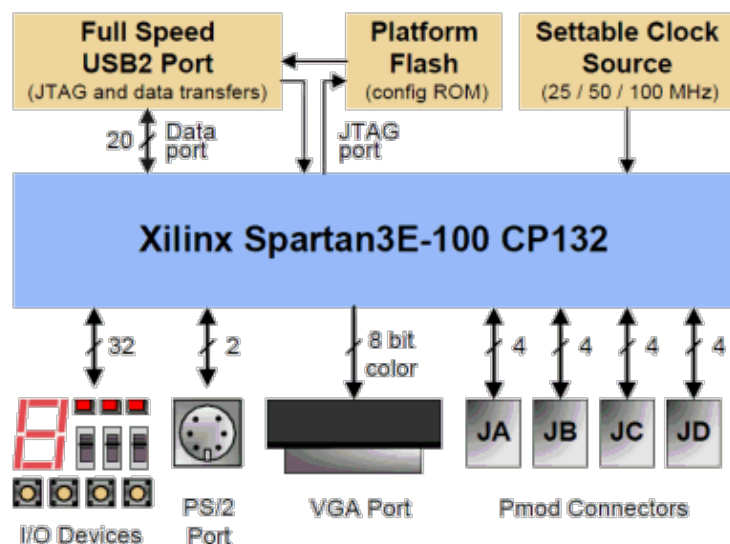
<sup>8</sup><http://arduino.cc/en/Reference/HomePage>

### 4.3 Basys2

FPGA ülesannete läbiviimiseks on kasutusel Digilent'i poolt valmistatud Basys2 arendusplaat Spartan 3E FPGA-ga.<sup>9</sup> Programmeerimine toimub Xilinx ISE Design Suite versioon 14.7 tarkvaraga ja programmi platvormile laadimine läbi Adept tarkvara versiooniga 2.4.2. Basys2 on valmis kasutamiseks arendusplatvorm, mille eesmärk on pakkuda kasutajale kergesti programmeeritav riistvaraalus. Programmeerimiskeeleks valiti VHDL, mis on loodud laiaulatusliku paralleelarvutus programmeerimiskeelena. Arendusriistvara spetsifikatsioonid on:<sup>10</sup>

<b>Integraalskeem</b>	<b>Xilinx Spartan-3e FPGA</b>
<b>Väratite arv</b>	<b>100000</b>
<b>Taktkiirus</b>	<b>25/50/100MHz</b>
<b>Visuaalne side</b>	<b>8LEDi, 4-ühikuline 7-segmendiga ekraan</b>
<b>Sisend</b>	<b>4 nuppu, 8 lülitit</b>
<b>Port</b>	<b>Atmel AT90USB2 USB2</b>
<b>Lisad</b>	<b>Neli 6-viiguga porti väliste seadmete (Pmod) ühendamiseks</b>
<b>Mälu</b>	<b>Xilinx Flash ROM</b>

Tabel 1. Basys2 spetsifikatsioonid



Joonis 1. Basys2 ehitus<sup>11</sup>

<sup>9</sup>[http://www.digilentinc.com/Products/Detail.cfm?](http://www.digilentinc.com/Products/Detail.cfm?Prod=BASYS2&NavTop=2&NavSub=649&CFID=4653109&CFTOKEN=3690b1b8335e3ea5-FA93AC6D-5056-0201-02B202C0485603F7)

<http://www.digilentinc.com/Products/Detail.cfm?Prod=BASYS2&NavTop=2&NavSub=649&CFID=4653109&CFTOKEN=3690b1b8335e3ea5-FA93AC6D-5056-0201-02B202C0485603F7>

<sup>10</sup>[http://www.digilentinc.com/Data/Products/BASYS2/Basys2\\_rm.pdf](http://www.digilentinc.com/Data/Products/BASYS2/Basys2_rm.pdf)

<sup>11</sup><http://www.digilentinc.com/Data/Products/BASYS2/BASYS2-block-380.gif>

#### 4.4 x86

x86 arhitektuuri testimiseks on kasutusel 2013. aastal komplekteeritud lauaarvuti. Arvutusjõu annab AMD FX-8350 kaheksatuumaline 4GHz taktsagedusega protsessor. Operatsioonisüsteem on Microsoft Windows 8 Pro 64-bit. Programmeerimise keskkond on Python 3.3.2.

<b>CPU</b>	<b>AMD FX-8350 4GHz Eight-Core Processor</b>
<b>Jahuti</b>	<b>Xigmatek Aegir</b>
<b>Emaplaat</b>	<b>Gigabyte GA-990FXA-UD3</b>
<b>RAM</b>	<b>Kingston HyperX Black 1600MHz DDR3 2x4GB</b>
<b>Videokaart</b>	<b>ATI Radeon HD 4800 Series</b>
<b>HDD</b>	<b>Western Digital 2TB Green</b>
<b>SSD</b>	<b>Samsung 840 120GB</b>
<b>PSU</b>	<b>Fortron 700W</b>
<b>Korpus</b>	<b>Zalman Z11</b>

Tabel 3. x86 arhitektuuriks kasutatava arvutikomplekti spetsifikatsioon

## 5 Tulemused

Kuna jõumeetodil tuvastamis harjutus eelistas paralleeltöötlust, mida Arduino Leonardo ei toeta ning mikrokiibi taktsagedus on 16 MHz, siis on teada, et tööaeg jääb Basys2, mille harjutuse lahendusaeg on lühem, ja x86, mille lahenduse leidmise aeg võrreldav FPGA'ga on pikem, mõõtmiste vahele. Leonardo kood on sarnane x86 arhitektuuri koodiga ning protsesside töö järgneb sarnast tegevust. Kuna x86 kasutatava mikroprotsessori taktsagedus on 4.0Ghz, siis võib suure kindlusega väita, et Atmeli arhitektuuril jooksev programm lõpetab oma töö mitmetes kordades hiljem. Täpsema ajakulu arvutamiseks loodav kood ja selle implementeerimine ületab lahenduse kasuteguri. Küll on kasutatud Arduino't jõumeetodil arvu tuvastamist aga need on olnud tüüpiliselt madala kuni keskmise keerukusastmega ja mitte suure kombinatsioonide hulgaga.<sup>12</sup>

Arduino ülesannete ajakulu leiti käsu *millis()* abil, mis väljaksudes tagastab millisekundites aja programmi käima minekust. Tulemus saadeti üle COM pordi ühenduses olevasse arvutisse. Mõõtmise aja sisse ei kuulu COM pordi kaudu andmevahetus.

x86 puhul toimus ajamõõtmine nii Python käsu *time.time()* abil kui ka *timeit.timeit()* käsuga. Esimesel puhul tagastatakse programmi algusest kulunud aeg. Kasutades sama väljakutset ülesande algul ja lõpus saab nende aegade vahest välja arvutada kulunud aja. Teisel käsul sooritatakse ülesannet niipalju kordi, kui antakse ette muutuja. Tagastatakse kõikide ülesannete kordade lahendamise koondaeg, mille jagamisel läbimise kordadega saab keskmise ajakulu. Korduv jooksutamine on vajalik tõelise keskmise võimalikult täpse leidmiseks. Ülesandeid jooksutati vähemalt 6 korda. Tabelis 3 jooksutati loogikatehte programmiosa Python's 6 korda järjest ja mõõtmisi tehti 6 korda. Tulemustest võeti keskmine ja seejärel arvutati ühe loogikatehte harjutuse tegemiseks kulunud aeg.

Loogikatehe (x86)	6 korra jooksutamine
katse 1	34.99ms
katse 2	46.00ms
katse 3	58.05ms
katse 4	57.99ms
katse 5	58.01ms
katse 6	22.00ms
keskmine	46.17ms
1 korra aeg	7.70ms

Tabel 4. Loogikatehte ülesande lahendamise aeg x86 arhitektuuris

---

<sup>12</sup><http://www.arduino passion.com/brute-force-attack-a-bios-with-arduino/>

Jõumeetodil tuvastamine (x86)	6 korra jooksutamine
katse 1	37.01ms
katse 2	17.00ms
katse 3	35.01ms
katse 4	38.01ms
katse 5	45.01ms
katse 6	39.01ms
keskmine	35.17ms
1 korra aeg	5.86ms

Tabel 5. Jõumeetodil tuvastamise ülesande lahendamise aeg x86 arhitektuuris

Registrite töötlemine (x86)	6 korra jooksutamine
katse 1	13.99ms
katse 2	40.01ms
katse 3	33.00ms
katse 4	27.01ms
katse 5	36.00ms
katse 6	51.05ms
keskmine	33.51ms
1 korra aeg	5.59ms

Tabel 6. Registrite töötlemise ülesande lahendamise aeg x86 arhitektuuris

7400 seeria transistor-transistor loogika kasutamisel Fibonacci arvu leidmiseks kolme sekundi jooksul on ainsaks aeglustavaks teguriks lülituste seisundi võtmine, kuna puudub taktgeneraator. Ühe 7400 seeria kiibi NAND lülituse aeg on testtingimustes halvimal juhul  $22\text{ns}^{13}$  dokumentatsiooni järgi. Lülituste arv kasvab eksponentsiaalselt, mistõttu pole praktiliselt kasulik antud ülesannet lahendada 7400 seeria mikrokiipidega. 7400 mikrokiibi ajakulukuseks tuleb leida kui pikk on kõige pikem väratite jada, mis skeemis tekib. Seejärel tuleb loogikatehted viia kujule mis on lahendatavad 7400 seeria mikrokiipide poolt. Antud juhul viidi kõik tehted NAND väratite kujule üle. Kuna näiteks XOR väratit saab luua neljast NAND väratist ja kolm neist on üksteise järel, tuleb ajakulul arvestada lülituste ajaliselt õige väärtuse saavutamist. Näiteks on loogikatehte harjutuse puhul kõige pikemalt tarvis läbida 10 NAND elementi. Arvestades lülituseks kuluvat maksimaalset aega 22ns saab leida lõppväärtuse õige tulemuse saamiseks  $10 \cdot 22\text{ns} = 220\text{ns}$ . Sarnast loogikat kasutati kõigi 7400 seeria mikrokiipide ajakulu arvutamiseks.

Voolutarbe puhul ei mõõdetud reaalsel harjutuse ajal mõõdetud energiakulu, vaid kasutati dokumentatsioonis märgitud komplektide voolutarbet töötamise ajal.

<sup>13</sup> <http://www.ti.com/lit/ds/symlink/sn74ls00.pdf>

Voolutarve	Komplektil	Mikrokiibil
7400	NA	105mW
Arduino	165mW	100mW
FPGA	120mW	32mW
x86	>250W	213W

Tabel 7. Voolutarbe<sup>14</sup>

Ülesannete aegade arvutamine käib kas programmi tagasiside kaudu või tarkvaral kaasas oleva statistika uurimisel. Joonisel 2 ja 3 on näha kuidas Xilinx tarkvara lahendab Fibonacci arvu tuvastamise harjutust. Järgnevad tabelid on koostatud jooksutatud programmide statistikast lähtuvalt:

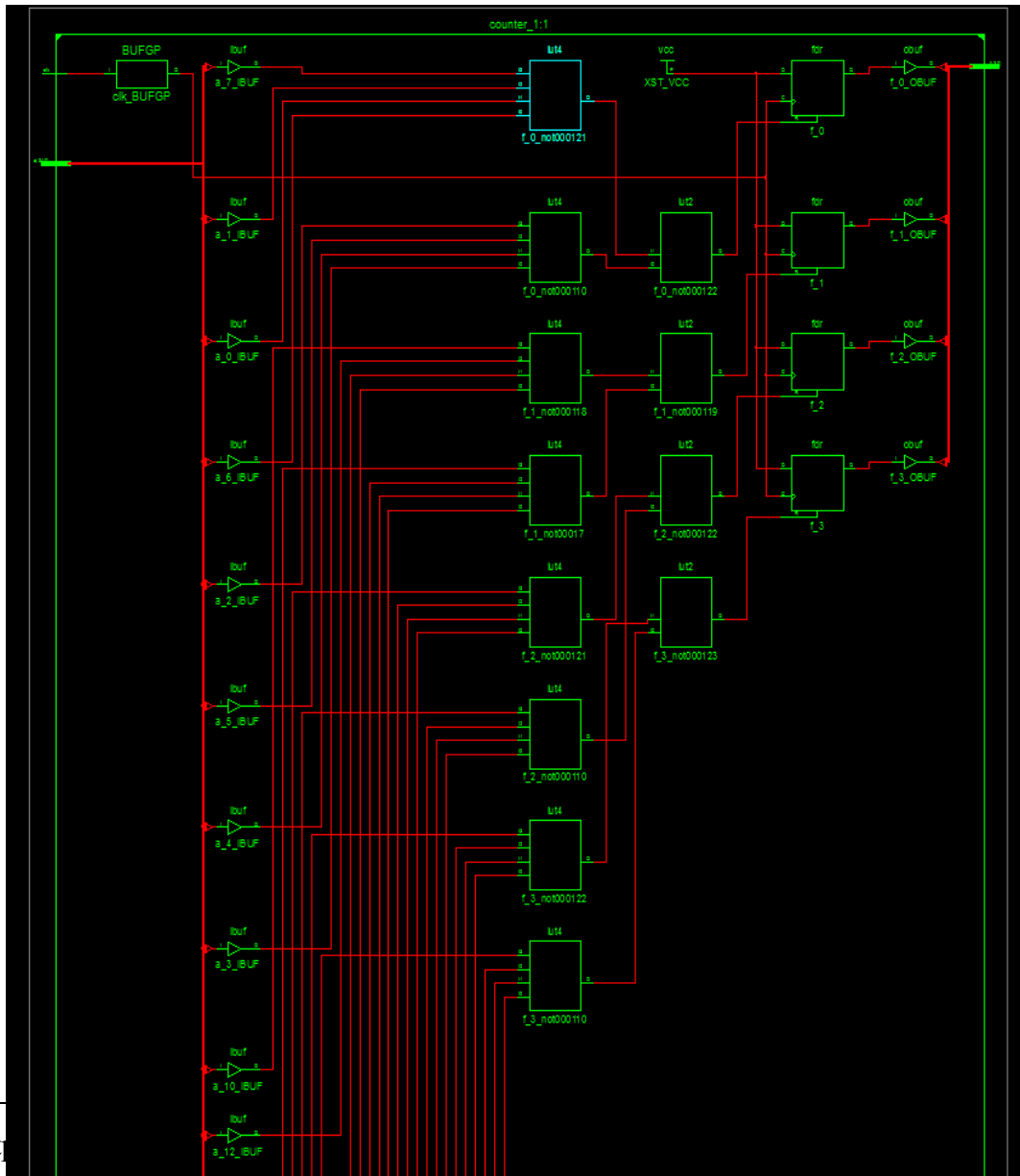
	Loogikatehe	Registrite töötlemine	Jõumeetodil tuvastamine	Fibonacci arv
<b>7400</b>	<b>220ns</b>	<b>24ns</b>	<b>13ns</b>	<b>136*10<sup>6</sup>tsüklit</b>
<b>Atmel</b>	<b>590ms</b>	<b>10ms</b>	<b>NA</b>	<b>16*10<sup>6</sup>tsüklit</b>
<b>FPGA</b>	<b>20ns</b>	<b>80ns</b>	<b>1.7ms</b>	<b>150*10<sup>6</sup>tsüklit</b>
<b>x86</b>	<b>7.7ms</b>	<b>5.5ms</b>	<b>5.9ms</b>	<b>4000*10<sup>6</sup>tsüklit</b>

Tabel 8. Harjutuste kulunud aeg ja tsüklite arv

	Mikrokiibi/integraalskeemi hind (€)	Komplekti hind (€)
7400	1.51€	3.70€
Arduino	8.29€	18€
FPGA	30.99€	65€
x86	178.80€	<900€

Tabel 9. Platvormide võrdlemine<sup>15, 16, 17, 18, 19, 20</sup>

<sup>14</sup> <http://www.bit-tech.net/hardware/2012/11/06/amd-fx-8350-review/7>



<sup>15</sup>http

ttl&s=7400

<sup>16</sup><http://www.ebay.com/itm/EVAL-FPGA-BASYS2-SPARTAN-3E-100K-Part-410-155-/390753966819?>

pt=UK\_BOI\_Electrical\_Components\_Supplies\_ET&hash=item5afac05ae3

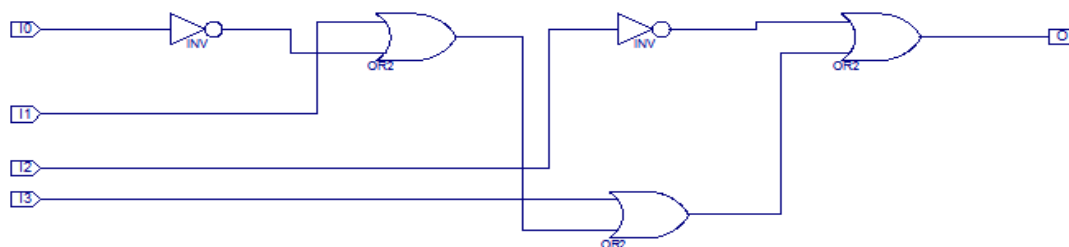
<sup>17</sup><http://arvutikeskus.ee/est/TOOTEKATALOOG/ARVUTIKOMPONENDID-Protessorid-CPU-Socket-AM3-protessorid149/AMD-CPU-FX-X8-8350-SAM3-BOX-125W-4000-FD8350FRHKBOX-AMD-186275>

<sup>18</sup>[http://store.arduino.cc/index.php?main\\_page=product\\_info&cPath=11&products\\_id=226](http://store.arduino.cc/index.php?main_page=product_info&cPath=11&products_id=226)

<sup>19</sup>[http://www.oomipood.ee/product/farnell\\_1876204/fpga-spartan-3a-200k-100vqfp-xc3s200a-4vqg100c&s=spartan-3](http://www.oomipood.ee/product/farnell_1876204/fpga-spartan-3a-200k-100vqfp-xc3s200a-4vqg100c&s=spartan-3)

<sup>20</sup> <http://www.bit-tech.net/hardware/2012/11/06/amd-fx-8350-review/7>

Joonis 2. FPGA VHDL Jõumeetodil tuvastamise harjutuse skemaatika lõige



Joonis 3. FPGA VHDL Jõumeetodil tuvastamise skemaatika LUT4\_FFDF sisu.

Joonisel 2 on näha kuidas Xilinx tarkvara kasutab *Lookup table*'leid, mis on FPGA arhitektuuri üks tähtsamaid komponente, et lahendada ülesannet. Mida LUT4 implementeerib on täpsemalt välja toodud joonisel 3. Joonise 2 vasakust äärest on näha signaali sisse tulemist ja paremas ääres tulemuse väljumist.

### 5.1 Programmeerimiseks nõutavad oskused

Järgnev hinnang on loodud lähtudes Tartu Ülikoolis Arvutitehnika eriala läbinud kogemustega isiku oskustest. 7400 mikrokiipide töö jaoks antud töö sisul ei ole tarvis osata programmeerimiskeelt. Selle eest tuleb osata luua loogikalülituste skeem, mis kirjeldaks ära vāratite paigutuse koos sisendite, väljunditega, pinge allika ja maandusega. Ajakulu antud üleannete lahendamiseks puudus, kuna lahendused simuleeriti tarkvara peal, mis koostati seoses teise platvormi testimisega.

Atmel ATmega arhitektuuri programmeerimiseks on tarvis C keele valdamine algsel tasemel. Ülesannete teostamine ei nõua kõrget programmeerimisoskust, kuid tarvis on teada erinevate andmetüüpide piiranguid. Programmeerimise aeg jäi pool tundi ülesande kohta.

Arduino riistvara Leonardo arendusplaadi programmeerimiseks on kasutataval keelel loodud palju lihtsustusi, et algajale programmeerijale pakkuda võimalikult laia tuge. Sellest tuleneb kahjuks probleem kirjutatava programmi ja interpretaatori vahel, sest programm ei pruugi käituda ootuspäraselt. Juhtus olukordi, kus väljund ei olnud õige, sest tarkvarapoolsed funktsioonid teisaldasid vāartuste tüüpe ootamatul viisil ilma hoiatamata. Soovitava väljundi mittetekkimise probleemi põhjust ei olnud koheselt võimalik teada saada, kuna puudusid seotud veateated, sest otsest viga koodis ei leidunud. Lahendus sellele oleks keele nüansside

parem tundmaõppimine. Ajakulu programmide loomiseks oli pool tundi ülesande kohta, kuna tutvuda tuli spetsiifilise programmeerimiskeele nüanssidega. Küll puudus käsk bittide rotatsiooni kohta, mis oleks registrite töötlemise ülesannet kergendanud. Seetõttu tuli kasutada järjendeid, mis tõstsid riistvara poolt tehtavate tsüklite arvu ja koodi pikkust.

x86 riistvarale loodi programmid Python programmeerimiskeeles. Python on tuntud kui kõrgetaseme programmeerimiskeel, mille peamine eesmärk on loetavusel. Üks silmapaistvamaid kohti on muutujate tüüpide määramine tarkvara enda poolt. Seetõttu pole tarvis rangelt määrata algselt erinevate muutujate andmetüüpe ja nende pikkusi.

VHDL programmeerimiskeele oskustase on kõige kõrgem, kuna iga harjutuse jaoks oli tarvis osata eelnevalt kindlal tasemel programmeerimiskeelt. Kõige suurema väljakutse esitas VHDL'is paralleeltöötlust nõudev harjutus, kuna tarvis oli jooksutada samaaegselt mitut protsessi.

Kogemuste järgi saab tuua järgneva graafiku:

Platvorm	Oskuste tase	Harjutuste keerukus
<b>7400</b>	<b>madal</b>	<b>madal</b>
<b>Arduino</b>	<b>keskmine</b>	<b>madal</b>
<b>VHDL</b>	<b>kõrge</b>	<b>keskmine-kõrge</b>
<b>Python</b>	<b>madal</b>	<b>madal-keskmine</b>

Tabel 10. Programmeerimiskeele nõutud tase sõltuvalt platvormist

## 6 Kokkuvõte

Tänapäeval turul olevad riistvaraplatvormid on loodud hõivama kindlaid nišivaldkondi ja iga platvorm toetub oma tugevustele jäädes tagaplaanile teistes aspektides. Ei ole ühtegi riistvaralahendust, mis oleks parim igas olukorras ja sellise leidmine ei ole olnud antud töö eesmärgiks. Kui lähteülesande lahendus peab olema madala hinnaga, on soovituslik kasutada 7400 seeria kiibistikku või Atmega AVR arhitektuuri mikroprotsessoreid. Kui eesmärgiks on jõudlus, siis pöörduda FPGA või x86 arhitektuuri poole. Kasutaja piiratud programmeerimisoskuste korral on soovituslik valida endale x86 arhitektuur, sest sellel on suur valik rakendatavaid programmeerimiskeeli, milledega kaasnevad mitmekesised kasutajatoed. Igale ülesandele on olemas talle sobiv riistvaralahendus. Lõputööle tuginedes võib edasi uurida rohkemaid platvorme ja neil sooritada laiemaid teste. Platvormidest võib valida Harvard arhitektuuril loodud *Peripheral Interface Controller* ja Texas Instruments mikrokontrollerid. Ülesannete poolest võib sooritada vastupidavus teste, kontrollimaks töötamist erinevates olukordades nagu pinge kõikumised, temperatuuri ekstreemsused ja füüsiline tugevus.

## Tsiteeritud teosed

- [1] Standard Logic IC Families introduced (2007) Computer history museum.  
[Online]. <http://www.computerhistory.org/semiconductor/timeline/1963-TTL.html>
- [2] The story of AVR (2014, mai) avrtvtube. [Online].  
<https://www.youtube.com/watch?v=HrydNwAxbcY>
- [3] MegaAVR Microcontrollers (2014, mai) Atmel [Online]  
<http://www.atmel.com/products/microcontrollers/avr/megaavr.aspx>
- [4] Birth of a standard (juuni, 2008) PCWorld [Online]  
<http://www.pcworld.com/article/146957/article.html>
- [5] Major trends of the fourth age of computing:  
New pressures for growing educational capacity (aprill, 2014) Houghton  
[https://www.wcu.edu/ceap/houghton/readings/technology\\_trends.html](https://www.wcu.edu/ceap/houghton/readings/technology_trends.html)
- [6] A Look Back at Single-Threaded CPU Performance (veebruar, 2012) Preshing  
on programming <http://preshing.com/20120208/a-look-back-at-single-threaded-cpu-performance/>
- [7] Basys2 board (mai, 2014) Digilent [Online].  
<http://www.digilentinc.com/Products/Detail.cfm?Prod=BASYS2&NavTop=2&NavSub=649&CFID=4653109&CFTOKEN=3690b1b8335e3ea5-FA93AC6D-5056-0201-02B202C0485603F7>
- [8] Digilent Basys2 Board Reference Manual (mai, 2014) Digilent [Online].  
[http://www.digilentinc.com/Data/Products/BASYS2/Basys2\\_rm.pdf](http://www.digilentinc.com/Data/Products/BASYS2/Basys2_rm.pdf)
- [9] Digilent Basys2 Board block (mai, 2014) Digilent [Online]  
<http://www.digilentinc.com/Data/Products/BASYS2/BASYS2-block-380.gif>
- [10] Arduino Board Leonardo (mai, 2014) Arduino [Online]  
<http://arduino.cc/en/Main/arduinoBoardLeonardo>
- [11] Language Reference (mai, 2014) Arduino [Online]  
<http://arduino.cc/en/Reference/HomePage>
- [12] Brute force attack a BIOS with Arduino (mai, 2014) Arduino Passion [Online]  
<http://www.arduinopassion.com/brute-force-attack-a-bios-with-arduino/>
- [13] Datasheet (mai, 2014) Texas Instruments  
<http://www.ti.com/lit/ds/symlink/sn74ls00.pdf> (mai, 2014)
- [14] AMD FX-8350 Power Consumption (mai, 2014) Bit-tech <http://www.bit-tech.net/hardware/2012/11/06/amd-fx-8350-review/7>
- [15] 7400 (mai, 2014) Oomipood [Online]

- [http://www.oomipood.ee/product/tme\\_7400/ic-digital-nand-channels-4-inputs-2-dip14-ttl&s=7400](http://www.oomipood.ee/product/tme_7400/ic-digital-nand-channels-4-inputs-2-dip14-ttl&s=7400)
- [16] Spartan-3e 100k Basy2 (mai,2014) Ebay [Online] [http://www.ebay.com/itm/EVAL-FPGA-BASYS2-SPARTAN-3E-100K-Part-410-155-/390753966819?pt=UK\\_BOI\\_Electrical\\_Components\\_Supplies\\_ET&hash=item5afac05ae3](http://www.ebay.com/itm/EVAL-FPGA-BASYS2-SPARTAN-3E-100K-Part-410-155-/390753966819?pt=UK_BOI_Electrical_Components_Supplies_ET&hash=item5afac05ae3)
- [17] AMD FX-X8-8350 (september, 2013) Arvutikeskus [Online] <http://arvutikeskus.ee/est/TOOTEKATALOG/ARVUTIKOMPONENDID-Protessorid-CPU-Socket-AM3-protessorid149/AMD-CPU-FX-X8-8350-SAM3-BOX-125W-4000-FD8350FRHKBOX-AMD-186275>
- [18] Arduino Leonardo with headers (mai,2014) Arduino [Online] [http://store.arduino.cc/index.php?main\\_page=product\\_info&cPath=11&products\\_id=226](http://store.arduino.cc/index.php?main_page=product_info&cPath=11&products_id=226)
- [19] FPGA Spartan-3a (mai,2014) Oomipood [Online] [http://www.oomipood.ee/product/farnell\\_1876204/fpga-spartan-3a-200k-100vqfp-xc3s200a-4vqg100c&s=spartan-3](http://www.oomipood.ee/product/farnell_1876204/fpga-spartan-3a-200k-100vqfp-xc3s200a-4vqg100c&s=spartan-3)
- [20] Power consumption and efficiency (mai,2014) Techreport <http://www.bit-tech.net/hardware/2012/11/06/amd-fx-8350-review/7>

## **Summary in English**

### **Hardware selection depending on mission objective**

Tarmo Kolsar

The purpose of this thesis is to compare different hardware platforms between each other. The platforms in question are 7400 series transistor-transistor logic microchips, Atmega AVR technology chips, FPGA and x86 instruction set based microprocessors.

In addition tests are performed on each hardware. The selected hardwares for platforms are Arduino Leonardo, Basys2 development board, tabletop PC with windows operating system and Python software. There are 4 different tests performed which include calculating Fibonacci numbers and performing mathematical logic.

In the end weaknesses and strengths are viewed at and a conclusion made for which primary objective, which platform is best suited.

## **Lisad**

Kõik harjutustes kasutatavad koodid on lisadena CD kettal. Kettal asub iga platvormi jaoks eraldi kaust, kus on sisuna ülesannete jaoks loodud programmi failid. Failide kogumaht on 488KB.

## Terminid

### FPGA

*(Field-Programmable Gate Array)*  
väliprogrammeeritav väravamassiiv või  
programmeeritav ventiilmaatriks

### IDE

*(Integrated Development Environment)*  
integreeritud arenduskeskkond

### VHDL

*(VHSIC Hardware Description Language)*  
Väga kiirete integraallülituste riistvara  
kirjeldamise keel

### LUT

(Lookup table)  
Otsingutabel

## Litsents

### Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina Tarmo Kolsar (sünnikuupäev: 31.12.1989)  
(*autori nimi*)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose  
**Suunitlus riistvaraplatvormi valimiseks sõltuvalt lähteülesandest,**  
(*lõputöö pealkiri*)

mille juhendaja on Margus Rosin  
(*juhendaja nimi*)

1. reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
2. üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
3. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
4. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, 01/06/14