Eötvös Lóránd Tudományegyetem
Programozási nyelvek és fordítóprogramok tanszék

Strukturált formában tárolt programszöveg

Diplomamunka

Témavezető
Králik Barnabás
(volt) Doktorandusz hallgató

Készítette
Thier Richárd István
Programtervező Informatikus MSc

2018. március 6.

Témabejelentő

Az elfogadott és aláírt témabejelentőt a dolgozat második oldalaként szükséges elhelyezni, amelyet az adminisztrációban lehet átvenni. Itt tehát már a tanszéki pecséttel ellátott és aláírt lapot kell belefűzni (ezen oldal *helyett*, ez az oldal csak útmutatás). Az elektronikusan feltöltött dolgozatba már nem szabad beleszerkeszteni ezt a feladatkiírást.

Tartalomjegyzék

Kivonat **2**

Abstract **3**

1. Bevezetés **4**

2. Markdown-eszközök **5**
2.1. A dokumentum lefordítása Linux alatt 5
2.2. Alapadatok megadása 5

3. A dolgozat formai kivitele **7**
3.1. A dolgozat kimérete 7
3.2. A dolgozat nyelve 7
3.3. A dokumentum nyomdatechnikai kivitele 8

4. A Markdown-sablon használata **9**
4.1. Ábrák és táblázatok 9
4.2. Felsorolások és listák 10
4.3. Képletek 10
4.4. Irodalmi hivatkozások 12
4.5. A dolgozat szerkezete és a forrásfájlok 14

5. Összefoglaló **16**

Köszönetnyilvánítás **17**

A. Függelék **21**
A.1. Válasz az „Élet, a világmindenség, meg minden” kérdésére 21

Kivonat

A dolgozat témája egy strukturált formában tárolt programszöveg és feldolgozási mód vizsgálata, amely egy fordítóprogram-motor alapját képezheti - a 3D futtató-motor rendszerekkel analóg módon. Az említett 3D-motor technológia már láthatóan uralja a 3D játékfejlesztés és megjelenítés területét, de akárcsak a fordítóprogramok esetében, korábban a 3D technológiát is ad-hoc, egyedi fejlesztések jellemezték. Idővel az eljárások szoftverkönyvtárakká, keretrendszerekké és legvégül az iparban is használt teljeskörű futtatómotorrá fejlődtek. A szoftverkönyvtárak lényegében a hasznos függvények gyűjteményét jelentik, a keretrendszerek már egy kiterjeszthető rendszert adnak, de a fordítóprogram és a programozási nyelvek technológiájában a következő lépés, teljeskörű futtatómotor még nem létezik.

A dolgozatban egy új strukturált reprezentációt, plugin-kezelési és scriptelési módszert definiálunk, elemzünk, vizsgálunk, ami egy C++14 implementációval megvalósítva, továbbfejleszthető alapot szolgáltat egy ilyen futtatómotor kifejlesztéséhez. A készülő kezdeti rendszer, a „turul" (Tree-Using Really Unorthodox Language) a futtatókörnyezetet fogja képezni, az erről külön is leválasztható adatreprezentácó pedig a „tbuf" (vagy turbobuf) nevű alrendszerben található. Ez utóbbit a fő forrásokról leválasztva, külön kezeljük, mert egy gyorsan olvasható, egyedi faszerkezetet leíró reprezentációt ad. Az így kapott megoldás „nyelv"-ként történő elnevezését az adja, hogy bár egy futtatókörnyezetről van szó, annak adatszerkezete - és az ezzel történő „scriptelés" - egy programnyelvet képez. Az utóbbiak alapján tehát a megoldásra nem csak egy programozási és fejlesztési környezetként, hanem egy eddig nem látott metaprogramozási képességgel felruházott önmódosító nyelvként is tekinthetünk: az elnevezésben ezt a terminológiát követtük.

A dolgozat célja elsősorban a strukturált formában tárolt és feldolgozott programszöveg, a scriptelhető elképzelések által felvetett lehetőségeknek, illetve annak az elemzése, hogy ez az elképzelés milyen módon különbözik a fordító-fordítók (pl. lex, yacc) keretrendszer-jellegű képességeitől. A dolgozat másodsorban kitér az elképzelés megvalósításához használható algoritmusokra, a hatékony implementációs lehetőségekre. A dolgozattal párhuzamosan készülő szoftver-komponensek egy kiterjeszthető futtató-motort adnak, amely további kutatások alapját képezheti és nem csak oktatási célra - de egyszerűbb kísérlegi fordítóprogramok, vagy programozási nyelvek kiértékeléséhez is használható.

Abstract

In this document we investigate a structured source code representation and processing approach with a goal to create „programming language engines" with similar architectural choices of the so-called „3D engines". These 3D engines are visibly dominating the field of 3D game and content development, but for a long time not only compilers, but also the games were created in ad-hoc ways, which later evolved into code „libraries", „frameworks" and more recently/finally complete „engines". While a code library is just a collection of useful methods, and a framework is just a set of rules that define an extensible system - engines provide an all-encompassing and structured way to handle the specifics of a field. Until now I do not know about any works that approach the field of compiler construction using this structured way.

In this document we define, analyse and investigate the structured data-based approach and „plugin scripting" that serves as an extensible architecture of a modern C++14 implementation for the forementioned system. The system is called turul (Tree-Using Really Unorthodox Language) and the readable-binary representation of trees that are used through the solution is called „tbuf" (or turbo-buf) for its hoped processing speed. Despite we are talking about an „engine" for languages, the system can be thought as an extensible language with unseen metaprogramming support too and in the final naming we stick to this terminology.

In our goal to investigate possibilities of more structured ways of compiler constructrion using the engine approach, the differences of the engine approach and compiler-compilers (like lex and yacc) and the underlying algorithms of the efficent implementation is documented as a reference with proper baseline analysis of the ideas. Implementation serves as an extensible engine that can serve as the basis for further research and useful not only in teaching - but for smaller experimental compiler construction to evaluate programming language ideas too.

1. fejezet

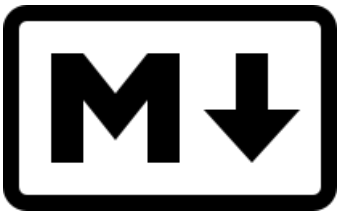
Bevezetés

A bevezető tartalmazza a diplomaterv-kiírás elemzését, történelmi előzményeit, a feladat indokoltságát (a motiváció leírását), az eddigi megoldásokat, és ennek tükrében a hallgató megoldásának összefoglalását.

A bevezető szokás szerint a diplomaterv felépítésével záródik, azaz annak rövid leírásával, hogy melyik fejezet mivel foglalkozik.

2. fejezet

Markdown-eszközök



2.1. ábra. A Markdown nyelv logója

2.1. A dokumentum lefordítása Linux alatt

A dokumentum fordításához szükséges a Pandoc és a TeXLive LaTeX-disztribúció telepítése:

```
sudo apt-get install texlive-base pandoc pandoc-citeproc
```

A diplomaterv a `make` parancssal fordítható le, a konfigurációt a `Makefile` állomány tartalmazza.

2.2. Alapadatok megadása

A diplomaterv alapadatait (cím, szerző, konzulens, konzulens titulusa) a `Makefile` fájlban lehet megadni a változók módosításával:

```
AUTHOR=...
TITLE=...
```

Az alapértelmezett kimenet a PDF, de különböző kimeneteket is megadhatunk a `make` parancs céljának változatásával:

```
make pdf
make html
make mobi
```

3. fejezet

A dolgozat formai kivitele

A dokumentum kialakítása során törekedtünk az ELTE IK MSc szakdolgozati tudnivalóknak való megfelelésre. Az itt található információk egy része is innen került átvételre. Az eredeti dokumentum az alábbi linken érhető el: https://www.inf.elte.hu/content/programtervezo-informatikus-bsc-programozo-matematikus-szakdolgozat-programtervezo-matematikus-dipt.1138?m=192.

3.1. A dolgozat kimérete

Minimálisan elvárható 50, de az optimális kiméret 60–80 oldal (függelékkel együtt). A bírálók és a záróvizsga bizottság sem szereti kifejezetten a túl hosszú dolgozatokat, így a bruttó 90-100 oldalt nem érdemes jelentősen túlszárnyalni. Egyébként függetlenül a dolgozat kiméretétől, ha a dolgozat nem érdekfeszítő, akkor az olvasó már az elején a végét fogja várni. Érdemes zárt, önmagában is érthető művet alkotni.

3.2. A dolgozat nyelve

Mivel Magyarországon a hivatalos nyelv a magyar, ezért alapértelmezésben magyarul kell megírni a dolgozatot. Aki külföldi posztgraduális képzésben akar részt venni, nemzetközi szintű tudományos kutatást szeretne végezni, vagy multinacionális cégnél akar elhelyezkedni, annak célszerű lehet angolul megírnia diplomadolgozatát. Mielőtt a hallgató az angol nyelvű verzió mellett dönt, erősen ajánlott mérlegelni, hogy ez mennyi többletmunkát fog a hallgatónak jelenteni fogalmazás és nyelvhelyesség terén, valamint – nem utolsó sorban – hogy ez mennyi többletmunkát fog jelenteni a konzulens, illetve a bíráló számára. Egy nehezen olvasható, netalán érthetetlen szöveg teher mindenki számára.

3.3. A dokumentum nyomdatechnikai kivitele

A diplomamunkát kemény kötésben kell leadni , 1 példányban. Fedőlapjának színe fekete, aranyszínű feliratokkal. A lap mérete: A4-es méret, színe fehér; betűméret: 12 pont. Sorokra vonatkozó megkötés: sorkizárt igazítás, 1,5-es sortávolság.

Standard margó: - belső: 3,5 cm - külső: 2,5 cm - alsó: 2,5 cm - felső: 2,5 cm

4. fejezet

A Markdown-sablon használata

Ebben a fejezetben röviden, implicit módon bemutatjuk a sablon használatának módját, ami azt jelenti, hogy sablon használata ennek a dokumentumnak a forráskódját tanulmányozva válik teljesen világossá. Amennyiben a szoftver-keretrendszer telepítve van, a sablon alkalmazása és a dolgozat szerkesztése Markdownban a sablon segítségével tapasztalataink szerint jóval hatékonyabb, mint egy WYSWYG (*What You See is What You Get*) típusú szövegszerkesztő esetén (pl. Microsoft Word, OpenOffice).

A LaTeX tördelőrendszer használatához képest a Markdown nyelv több megszorítást is tartalmaz, így az elkészített dokumentum általában kevésbé testreszabható. Cserébe viszont a Markdownban készült dokumentumok exportálhatók HTML vagy e-könyv (EPUB) formátumba is.

4.1. Ábrák és táblázatok

A képeket a veszteségmentes PNG, valamint a veszteséges JPEG formátumban érdemes elmenteni.

Az egyes képek mérete általában nem, de sok kép esetén a dokumentum összmérete így már szignifikáns is lehet. A dokumentumban felhasznált képfájlokat a dokumentum forrása mellett érdemes tartani, archiválni, mivel ezek hiányában a dokumentum nem fordul újra. Ha lehet, a vektorgrafikus képeket vektorgrafikus formátumban is érdemes elmenteni az újrafelhasználhatóság (az átszerkeszthetőség) érdekében.

A képek beillesztésére a Markdown eszközök fejezetben mutattunk be példát. Az előző mondatban egyúttal az automatikusan feloldódó ábrahivatkozásra is láthatunk példát.

A táblázatok használatára az alábbi táblázat mutat példát.

4.1. táblázat. Az órajel-generátor chip órajel-kimenetei.

Órajel	Frekvencia	Cél pin
CLKA	100 MHz	FPGA CLK0

Órajel	Frekvencia	Cél pin
CLKB	48 MHz	FPGA CLK1
CLKC	20 MHz	Processzor
CLKD	25 MHz	Ethernet chip
CLKE	72 MHz	FPGA CLK2
XBUF	20 MHz	FPGA CLK3

4.2. Felsorolások és listák

Számozatlan felsorolásra mutat példát a jelenlegi bekezdés:

- *első bajusz:* ide lehetne írni az első elem kifejését,
- *második bajusz:* ide lehetne írni a második elem kifejését,
- *ez meg egy szakáll:* ide lehetne írni a harmadik elem kifejését.

Számozott felsorolást is készíthetünk az alábbi módon:

1. *első bajusz:* ide lehetne írni az első elem kifejését, és ez a kifejtés így néz ki, ha több sorosra sikeredik,
2. *második bajusz:* ide lehetne írni a második elem kifejését,
3. *ez meg egy szakáll:* ide lehetne írni a harmadik elem kifejését.

A felsorolásokban sorok végén vessző, az utolsó sor végén pedig pont a szokásos írásjel. Ez alól kivételt képezhet, ha az egyes elemek több teljes mondatot tartalmaznak.

Listákban a dolgozat szövegétől elkülönítendő kódrészleteket, programsorokat, pszeudokódokat jeleníthetünk meg.

```
* *első bajusz:* ide lehetne írni az első elem kifejését, és ez a kifejtés így néz ki,
* *második bajusz:* ide lehetne írni a második elem kifejését,
* *ez meg egy szakáll:* ide lehetne írni a harmadik elem kifejését.
```

A listákban definiálható a használt programozási nyelv (pl. `java`, `latex`, `bash` stb.).

4.3. Képletek

Ha egy formula nem túlságosan hosszú, és nem akarjuk hivatkozni a szövegből, mint például a $e^{i\pi} + 1 = 0$ képlet, *szövegközi képletként* szokás leírni. Csak, hogy másik példát is lássunk, az $U_i = -d\Phi/dt$ Faraday-törvény a $\mathrm{rot}E = -\frac{dB}{dt}$ differenciális alakban adott Maxwell-egyenlet felületre vett integráljából vezethető le. Látható, hogy a LaTeX-fordító a sorközöket betartja, így a szöveg szedése esztétikus marad szövegközi képletek használata esetén is.

Képletek esetén az általános konvenció, hogy a kisbetűk skalárt, a kis félkövér betűk ($\mathbf{v}$) oszlopvektort – és ennek megfelelően $\mathbf{v}^T$ sorvektort – a kapitális félkövér betűk ($\mathbf{V}$) mátrixot jelölnek. Ha ettől el szeretnénk térni, akkor az alkalmazni kívánt jelölésmódot célszerű külön alfejezetben definiálni. Ennek megfelelően, amennyiben $\mathbf{y}$ jelöli a mérések vektorát, ϑ a paraméterek vektorát és $\hat{\mathbf{y}} = \mathbf{X}\vartheta$ a paraméterekben lineáris modellt, akkor a *Least-Squares} értelemben optimális paraméterbecslő $\hat{\vartheta}_{LS} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$ lesz.

Emellett kiemelt, sorszámozott képleteket is megadhatunk, ennél az `equation` és a `eqnarray` környezetek helyett a korszerűbb `align` környezet alkalmazását javasoljuk (több okból, különféle problémák elkerülése végett, amelyekre most nem térünk ki). Tehát

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \tag{4.1}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x}, \tag{4.2}$$

ahol $\mathbf{x}$ az állapotvektor, $\mathbf{y}$ a mérések vektora és $\mathbf{A}$, $\mathbf{B}$ és $\mathbf{C}$ a rendszert leíró paramétermátrixok. Figyeljük meg, hogy a két egyenletben az egyenlőségjelek egymáshoz igazítva jelennek meg, mivel a mindkettőt az & karakter előzi meg a kódban. Lehetőség van számozatlan kiemelt képlet használatára is, például

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u},\\ \mathbf{y} &= \mathbf{C}\mathbf{x}.\end{aligned}$$

Mátrixok felírására az $\mathbf{A}\mathbf{x} = \mathbf{b}$ inhomogén lineáris egyenlet részletes kifejtésével mutatunk példát:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}. \tag{4.3}$$

A `\frac` utasítás hatékonyságát egy általános másodfokú tag átviteli függvényén keresztül mutatjuk be, azaz

$$W(s) = \frac{A}{1 + 2T\xi s + s^2T^2}. \tag{4.4}$$

A matematikai mód minden szimbólumának és képességének a bemutatására természetesen itt nincs lehetőség, de gyors referenciaként hatékonyan használhatók a

következő linkek:

- http://www.artofproblemsolving.com/LaTeX/AoPS_L_GuideSym.php,
- http://www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf,
- ftp://ftp.ams.org/pub/tex/doc/amsmath/short-math-guide.pdf.

Ez pedig itt egy magyarázat, hogy miért érdemes `align` környezetet használni: http://texblog.net/latex-archive/maths/eqnarray-align-environment/.

4.4. Irodalmi hivatkozások

Az irodalmi hivatkozások alkalmazására javasolt a BiBTEX használata, ezért ez a sablon is ezt támogatja. Ebben az esetben egy külön szöveges adatbázisban definiáljuk a forrásmunkákat, és egy külön stílusfájl határozza meg az irodalomjegyzék kinézetét. Ez, összhangban azzal, hogy külön formátumkonvenció határozza meg a folyóirat-, a könyv-, a konferenciacikk stb. hivatkozások kinézetét az irodalomjegyzékben (a sablon használata esetén ezzel nem is kell foglalkoznia a hallgatónak, de az eredményt célszerű ellenőrizni). A felhasznált hivatkozások adatbázisa egy `.bib` kiterjesztésű szöveges fájl, amelynek szerkezetét az alábbi kódrészlet demonstrálja. A forrásmunkák bevitelekor a sor végi vesszők külön figyelmet igényelnek, mert hiányuk a BiBTEX-fordító hibaüzenetét eredményezi. A forrásmunkákat típus szerinti kulcsszó vezeti be (`@book` könyv, `@inproceedings` konferenciakiadványban megjelent cikk, `@article` folyóiratban megjelent cikk, `@techreport` valamelyik egyetem gondozásában megjelent műszaki tanulmány, `@manual` műszaki dokumentáció esetén stb.). Nemcsak a megjelenés stílusa, de a kötelezően megadandó mezők is típusról-típusra változnak. Egy jól használható referencia a http://en.wikipedia.org/wiki/BibTeX oldalon található.

```
@BOOK{Wettl04,
  author = "Ferenc Wettl and Gyula Mayer and Péter Szabó",
  title = "\LaTeX~kézikönyv",
  publisher = "Panem Könyvkiadó",
  year = 2004
}
@ARTICLE{Candy86,
  author = "James C. Candy",
  title = "Decimation for Sigma Delta Modulation",
  journal =  "{IEEE} Trans.\ on Communications",
  volume = 34,
  number = 1,
  pages = "72--76",
```

```
  month = jan,
  year = 1986,
}
@INPROCEEDINGS{Lee87,
  author = "Wai L. Lee and Charles G. Sodini",
  title = "A Topology for Higher Order Interpolative Coders",
  booktitle = "Proc.\ of the IEEE International Symposium on Circuits and Systems",
  year = 1987,
  vol = 2,
  month = may # "~4--7",
  address = "Philadelphia, PA, USA",
  pages = "459--462"
}
@PHDTHESIS{KissPhD,
  author = "Peter Kiss",
  title = "Adaptive Digital Compensation of Analog Circuit Imperfections for Cascaded D
  school = "Technical University of Timi\c{s}oara, Romania",
  month = apr,
  year = 2000
}
@MANUAL{Schreier00,
  author = "Richard Schreier",
  title = "The Delta-Sigma Toolbox v5.2",
  organization = "Oregon State University",
  year = 2000,
  month = jan,
  note = "\newline URL: http://www.mathworks.com/matlabcentral/fileexchange/"
}
@MISC{DipPortal,
    author = "Budapesti {M}űszaki és {G}azdaságtudományi {E}gyetem, {V}illamosmérnöki é
  title = "{D}iplomaterv portál (2011 február 26.)",
  howpublished = "\url{http://diplomaterv.vik.bme.hu/}",
}
```

A stílusfájl egy `.sty` kiterjesztésű fájl, de ezzel lényegében nem kell foglalkozni, mert vannak beépített stílusok, amelyek jól használhatók. Ez a sablon a BiBTEX-et használja, a hozzá tartozó adatbázisfájl a `mybib.bib` fájl. Megfigyelhető, hogy az irodalomjegyzéket a dokumentum végére (a `\end{document}` utasítás elé) beillesztett `\bibliography{mybib}` utasítással hozhatjuk létre, a stílusát pedig ugyanitt a `\bibliographystyle{plain}`

utasítással adhatjuk meg. Ebben az esetben a `plain` előre definiált stílust használjuk (a sablonban is ezt állítottuk be). A `plain` stíluson kívül természetesen számtalan más előre definiált stílus is létezik. Mivel a `.bib` adatbázisban ezeket megadtuk, a BiBTEX-fordító is meg tudja különböztetni a szerzőt a címtől és a kiadótól, és ez alapján automatikusan generálódik az irodalomjegyzék a stílusfájl által meghatározott stílusban.

Az egyes forrásmunkákra a szövegből továbbra is a `@[...]` paranccsal tudunk hivatkozni, így a fenti kódrészlet esetén a hivatkozások rendre `[@Wettl04]`, `[@Candy86]`, `[@Lee87]`, `[@KissPhD]`, `[@Schreirer00]` és `[@DipPortal]`. Az irodalomjegyzékben alapértelmezésben csak azok a forrásmunkák jelennek meg, amelyekre található hivatkozás a szövegben, és ez így alapvetően helyes is, hiszen olyan forrásmunkákat nem illik az irodalomjegyzékbe írni, amelyekre nincs hivatkozás.

Mivel a fordítási folyamat során több lépésben oldódnak fel a szimbólumok, ezért gyakran többször (TeXLive és TeXnicCenter esetén 2-3-szor) is le kell fordítani a dokumentumot. Ilyenkor ez első 1-2 fordítás esetleg szimbólum-feloldásra vonatkozó figyelmeztető üzenettel zárul. Ha hibaüzenettel zárul bármelyik fordítás, akkor nincs értelme megismételni, hanem a hibát kell megkeresni. A `.bib` fájl megváltoztatáskor sokszor nincs hatása a változtatásnak azonnal, mivel nem mindig fut újra a BibTeX fordító. Ezért célszerű a változtatás után azt manuálisan is lefuttatni (TeXnicCenter esetén `Build/BibTeX`).

Hogy a szövegbe ágyazott hivatkozások kinézetét demonstráljuk, itt most sorban meghivatkozzuk a [6], [2], [4], [3] és az [5] forrásmunkát, valamint az [1] weboldalt.

teszt'

4.5. A dolgozat szerkezete és a forrásfájlok

A diplomatervsablon (a kari irányelvek szerint) az alábbi fő fejezetekből áll:

- *tájékoztató* a szakdolgozat/diplomaterv szerkezetéről, ami a végső dolgozatból törlendő, valamint a *feladatkiírás* (`guideline.md`), a dolgozat nyomtatott verzójában ennek a helyére kerül a tanszék által kiadott, a tanszékvezető által aláírt feladatkiírás, a dolgozat elektronikus verziójába pedig a feladatkiírás egyáltalán ne kerüljön bele, azt külön tölti fel a tanszék a diplomaterv-honlapra,
- *címoldal* (generált),
- *tartalomjegyzék* (generált),
- a diplomatervező *nyilatkozat*a az önálló munkáról (generált),
- 1-2 oldalas tartalmi *összefoglaló* magyarul és angolul, illetve elkészíthető még további nyelveken is (`abstract.md`),
- *fejezetek*: *bevezetés* (feladat értelmezése, a tervezés célja, a feladat indokoltsága, a diplomaterv felépítésének rövid összefoglalása, `chapter1.md`), a feladatkiírás pontosítása és részletes elemzése, előzmények (irodalomkutatás, hasonló alkotások), az

ezekből levonható következtetések, a tervezés részletes leírása, a döntési lehetőségek értékelése és a választott megoldások indoklása, a megtervezett műszaki alkotás értékelése, kritikai elemzése, továbbfejlesztési lehetőségek (`chapter{2..n}.md`),
- összefoglalás (`summary.md`),
- esetleges *köszönetnyilvánítás*ok (`acknowledgement.md`),
- részletes és pontos *irodalomjegyzék* (generált),
- *függelékek* (`appendices.md`).

5. fejezet

Összefoglaló

A diplomaterv összefoglaló fejezete.

Köszönetnyilvánítás

A köszönetnyilvánítás nem kötelező, akár törölhető is. Ha a szerző szükségét érzi, itt lehet köszönetet nyilvánítani azoknak, akik hozzájárultak munkájukkal ahhoz, hogy a hallgató a szakdolgozatban vagy diplomamunkában leírt feladatokat sikeresen elvégezze. A konzulensnek való köszönetnyilvánítás sem kötelező, a konzulensnek hivatalosan is dolga, hogy a hallgatót konzultálja.

Táblázatok jegyzéke

4.1. Az órajel-generátor chip órajel-kimenetei. 9

Ábrák jegyzéke

2.1. A Markdown nyelv logója 5

Irodalomjegyzék

[1] Villamosmérnöki és Informatikai Kar Budapesti Műszaki és Gazdaságtudományi Egyetem. Diplomaterv portál (2011 február 26.). http://diplomaterv.vik.bme.hu/.

[2] James C. Candy. Decimation for sigma delta modulation. *IEEE Trans. on Communications*, 34(1):72–76, January 1986.

[3] Peter Kiss. *Adaptive Digital Compensation of Analog Circuit Imperfections for Cascaded Delta-Sigma Analog-to-Digital Converters*. PhD thesis, Technical University of Timişoara, Romania, April 2000.

[4] Wai L. Lee and Charles G. Sodini. A topology for higher order interpolative coders. In *Proc. of the IEEE International Symposium on Circuits and Systems*, pages 459–462, Philadelphia, PA, USA, May 4–7 1987.

[5] Richard Schreier. *The Delta-Sigma Toolbox v5.2*. Oregon State University, January 2000.
URL: http://www.mathworks.com/matlabcentral/fileexchange/.

[6] Ferenc Wettl, Gyula Mayer, and Péter Szabó. *LaTeX kézikönyv*. Panem Könyvkiadó, 2004.

A. függelék

Függelék

A.1. Válasz az „Élet, a világmindenség, meg minden” kérdésére

A Pitagorasz-tételből levezetve

$$c^2 = a^2 + b^2 = 42.$$

A Faraday-indukciós törvényből levezetve

$$\mathrm{rot}E = -\frac{dB}{dt} \quad \longrightarrow \quad U_i = \oint\limits_{\mathbf{L}} \mathbf{Edl} = -\frac{d}{dt} \int\limits_{A} \mathbf{Bda} = 42.$$