VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

GRAFICKÉ UŽIVATELSKÉ ROZHRANÍ PRO PLATFORMU REPOMO

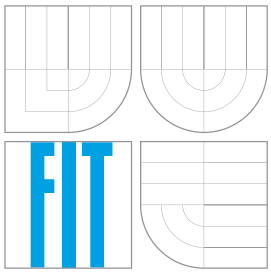
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PETR ŠIMEK

BRNO 2009

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

GRAFICKÉ UŽIVATELSKÉ ROZHRANÍ PRO PLATFORMU REPOMO

GRAPHICAL USER INTERFACE FOR REPOMO PLATFORM

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — PETR ŠIMEK
AUTHOR

VEDOUCÍ PRÁCE — doc. Ing. LUKÁŠ SEKANINA, Ph.D.
SUPERVISOR

BRNO 2009

Abstrakt

Rekonfigurace hardwaru se stala současným trendem, kdy pevnou obvodovou strukturu dokážeme softwarově konfigurovat. REPOMO je rekonfigurovatelný čip obsahující 32 polymorfních logických členů. Termín „polymorfní“ znamená, že obvod dokáže měnit svou logickou funkci v závislosti např. na úrovni napájecího napětí. Tato práce si klade za úkol seznámit čtenáře s problematikou polymorfismu v elektrických obvodech, konstrukcí a funkcí čipu REPOMO. Hlavním cílem práce bylo navrhnout a implementovat grafické uživatelské rozhraní, které usnadní vývoj aplikací pro čip REPOMO.

Abstract

Today, reconfigurable hardware is happend tendency when we can configure static circuit structure in the same way as software. REPOMO is a reconfigurable chip which contains 32 polymorphic gates. Polymorphism means that a gate can change its logic function as a response to the power supply voltage. The purpose of this thesis is to introduce the area of polymorphic circuits and construction and function of REPOMO chip. The main goal of the thesis is proposed and implemeted graphical user interface which make easy progress applications for chip REPOMO.

Klíčová slova

Polymorfismus, polymorfní logický člen, grafické uživatelské rozhraní, FITkit, REPOMO.

Keywords

Polymorphism, polymorphic gate, graphical user interface, FITkit, REPOMO.

Citace

Petr Šimek: Grafické uživatelské rozhraní pro platformu REPOMO, bakalářská práce, Brno, FIT VUT v Brně, 2009

Grafické uživatelské rozhraní pro platformu REPOMO

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením doc. Ing. Lukáše Sekaniny, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Petr Šimek
20.5.2009

Poděkování

Chtěl bych poděkovat doc. Ing. Sekaninovi, Ph.D. za vedení a možnost zpracovat tak zajímavé zadání bakalářské práce a Ing. Vašíčkovi za další odborné konzultace.

Obsah

Kapitola 1

Úvod

Termínem hardware s možností rekonfigurace máme na mysli zařízení s pevnou fyzickou strukturou, ale s možností změny vykonávané funkce, která je prováděna až v době po ukončení výrobního procesu [5].

Polymorfní elektronika může být považována za novou technologii po realizaci rekonfigurace, která je schopna začlenit do jedné kompaktní struktury logické členy s možností provádět více logických funkcí než právě jednu. Polymorfní logické členy, jejichž logická funkce závisí na úrovni napájecího napětí, jsou speciální třídou polymorfních logických členů [6].

Problematikou polymorfních logických členů se zabývá Fakulta informačních technologií VUT v Brně a výsledkem její činnosti je čip REPOMO, obsahující mimo jiné 32 logických členů, které dokáží měnit svou logickou funkci podle úrovně napájecího napětí.

Doposud byl čip REPOMO programován pomocí poněkud nepřehledných skriptů nebo ručně.

1.1 Cíl práce

Mým úkolem a cílem této práce je nastínit problematiku polymorfních logických členů a obvzláště se seznámit s čipem REPOMO, který obsahuje polymorfní logické členy. Dále navrhnout a implementovat grafické uživatelské rozhraní pro usnadnění práce s tímto obvodem a na závěr otestovat navržené schéma zapojení a demonstrovat dosažené výsledky.

1.2 Struktura práce

V následující kapitole přiblížím problematiku polymorfních hradel. V kapitole třetí představím už konkrétní řešení polymorfismu, a to čip REPOMO. Poté bude následovat kapitola o důležitém systému, bez kterého by nebylo možno s čipem pracovat, a to o FITkitu. Pátá kapitola je věnována obecným znalostem z oblasti grafických uživatelských rozhraní, v šesté kapitole je popsáno navržené GUI. V předposlední sedmé kapitole jsou prezentovány experimentální výsledky - ověření funkčnosti programu a poslední kapitola zhodnocuje dosažené výsledky.

Kapitola 2

Polymorfní elektronika

Pokrok v technologii CMOS umožnil návrh a rozvoj inteligentních adaptivních systémů. Tyto systémy mohou intergrovat inteligentní snímání, počítání a řízení v jedné křemíkové struktuře. Adaptace může být dosaženo tehdy, pokud je alespoň část obvodu konfigurovatelná. V některých aplikacích se provádí adaptace jen jednou, a to s cílem odstranění nedokonalostí zhotoveného obvodu, kterými jsou například kolísaní frekvence, vadné paměťové místo apod. Ve složitých případech se proces adaptace provádí prakticky ustavičně, protože se obvod přizpůsobuje dynamickému prostředí [6].

2.1 Polymorfní logické členy

Polymorfní logické členy, které představil tým z NASA SPL, reprezentují novou technologii po realizaci rekonfigurace. Polymorfní logické členy mohou být řízeny vnějším prostředím ve formě teploty, napájecího napětím, světla, signály apod. Například polymorfní logický člen AND/OR, závislý na teplotě, má v prvním módu při teplotě 27°C logickou funkci AND, v druhém módu při teplotě 125°C logickou funkci OR [6].

2.2 Polymorfní logický člen NOR/NAND (NASA SPL)

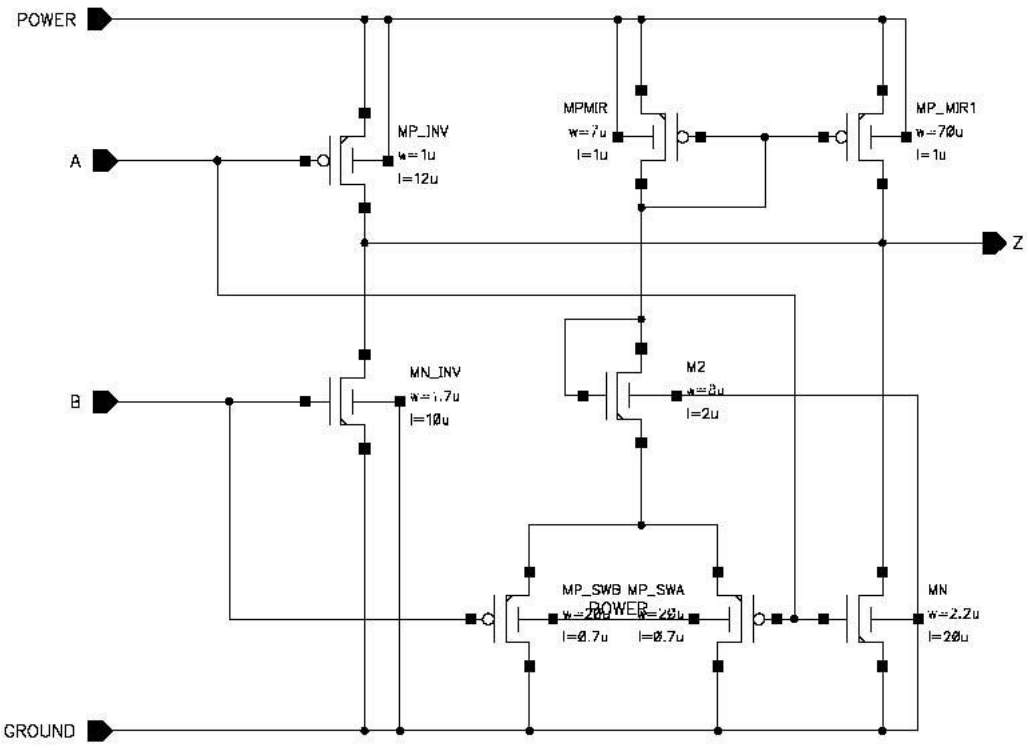
Jedním z polymorfních prvků je hradlo NOR/NAND, řízené úrovní napájecího napětí. Toto hradlo obsahuje 6 tranzistorů a je vyrobeno technologií 0,5 μm . Při napájecím napětí Vdd = 1,8 V hradlo pracuje jako NOR a při Vdd = 3,3 V jako NAND. Oproti běžnému číslicovému logickému členu, který považuje za logickou 0 napětí od 0-0,8 V a logickou 1 od 3-5 V, tak polymorfní člen pracuje v případě logické 1 v obou módech s identickými napěťovými úrovněmi, to jest logická 1 je reprezentována napětím 1,8 V a nezávisí, jestli je Vdd = 1,8 V nebo Vdd = 3,3 V (za povšimnutí stojí, že logická 0 je vždy reprezentována 0 V) [6].

2.3 Polymorfní logický člen NOR/NAND (FIT)

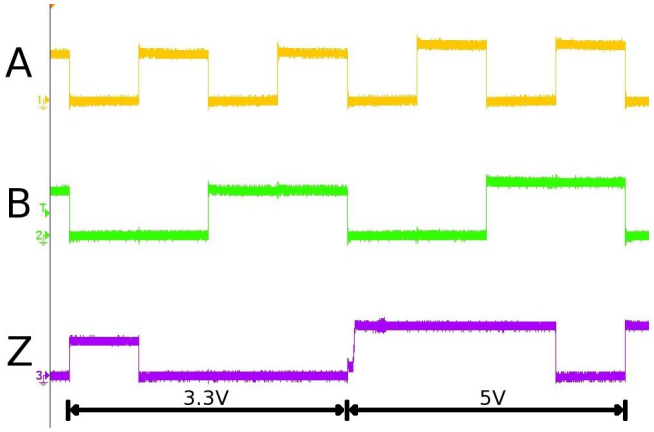
V čipu REPOMO se používá poněkud odlišný člen NOR/NAND, který byl vyroben technologií 0,7 μm. První mód pro Vdd=3,3 V definuje logickou funkci jako NOR a druhý mód pro Vdd=5 V definuje logickou funkci jako NAND. Schéma členu je zobrazeno na obrázku 2.1.

Obrázek 2.2 ukazuje jeho chování. První dva tranzistory na vstupech A a B slouží jako invertory a garantují správné logické úrovně výstupů pro vstupní kombinace 00 a 11. Tyto kombinace způsobí stejný výstup pro oba režimy členu. Rozdílné chování členu nastává při ostatních vstupních kombinacích: Pro výstupní hodnotu logická 0 u NORu a logická 1 u NANDu. V módu NOR je hodnota výstupu 0 dosažena otevřením tranzistou MN (nebo MN_inv respektive). Jak je tranzistor MP_MIR1 zavřený, napájecí napětí není dostatečné, aby otevřelo tranzistor MP_MIR a tranzistor M2. V módu NAND je na výstupu zajištěna hodnota 1 otevřeným tranzistorem MP_MIR1. Tento tranzistor je mnohem větší než zbylé MN tranzistory a tak převyšuje jejich nulový potenciál. Na obrázku 2.3 je ukázáno výstupní napětí při vstupní kombinaci „10“ a různých úrovní Vdd. Při Vdd = 0-3,8 V, člen se chová jako NOR (na výstupu je logická 0). Pro Vdd větší jak 3,9 V se člen chová jako NAND (výstup je v logické 1) [6].

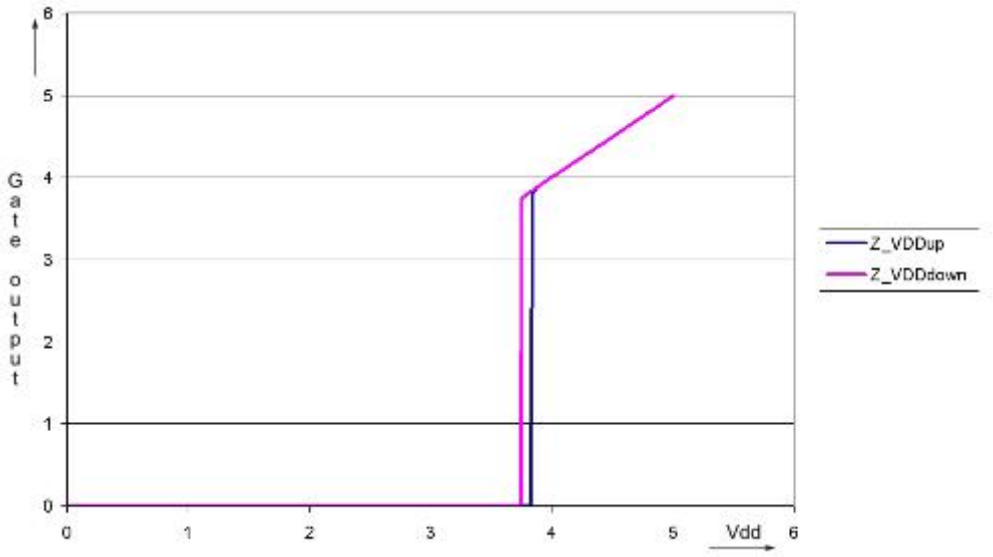
Měřené charakteristiky členu ukazují, že změna logické funkce probíhá okolo Vdd = 3,8 V. Stabilní výstupní hodnoty jsou pro Vdd menší jak 3,7 V a Vdd větší jak 3,9 V. V okolí 3,9 V může výstupní hodnota oscilovat. To se může dít díky měnícímu se příkonu při přepínání tranzistorů v logickém členu. Avšak funkční rozsah nestability je adekvátně malý a snížený hysterezí [6]. Mezní parametry jsou odvozeny od použité technologie a ukázány v tabulce 2.1. Elektrické parametry hradla můžeme vidět v tabulce 2.2.



Obrázek 2.1: Polymorfní hradlo NOR (Vdd = 3,3 V)/NAND (Vdd = 5 V) (převzato z [6])



Obrázek 2.2: Chování polymorfního logického členu NOR/NAND, měřeno při 5 kHz (převzato z [6])



Obrázek 2.3: Výstupní napětí pro vstup „10“ a rozdílné Vdd (převzato z [6])

	min.	**max.**
Vdd	-0,5 V	7 V
vstupně/výstupní napětí	-0,5 V	Vdd + 0,5 V
provozní teplota	0°C	+70°C
tranzistorová teplota	-40°C	+125°C

Tabulka 2.1: Některé mezní hodnoty hradla NOR/NAND (převzato z [6])

	Vdd = 3,3 V	**Vdd = 5 V**
V_{iL}	max. 0,79 V	max. 1,09 V
V_{iH}	min. 1,37 V	min. 1,78 V
V_{oL}	0,0 V	0,0 V
V_{oH}	3,29 V	4,97 V
f_{max}	38,6 MHz	56,2 MHz

Tabulka 2.2: Naměřené parametry hradla NOR/NAND (převzato z [6])

2.4 Použití polymorfních logických členů

Použití polymorfních členů je různé. Jsou například vhodné pro dynamické systémy, kde potřebujeme časteji rekonfigurovat obvod, pro realizaci skryté funkce nebo sloučení logiky se senzory. Byly rovněž využity v oblasti diagnostiky a testování [8].

Kapitola 3

REPOMO

Zkratka REPOMO[1] je odvozena z anglického názvu REconfigurable POlymorphic MOdule (rekonfigurovatelný polymorfní modul). Jedná se o nový rekonfigurovatelný čip s polymorfními hradly. Tento čip byl vyvinut pro výzkum elektrických vlastností polymorfních obvodů a demonstraci použití polymorfní elektroniky.

3.1 Popis REPOMO

Obsahuje pole 32 konfigurovatelných hradel (viz obrázek 3.1), kde každé z nich může vykonávat AND, OR XOR nebo polymorfní NOR/NAND funci, která je řízená úrovní napajecího napětí. Čip má 4 vstupy a 8 výstupů, což se hodí pro tvorbu malých polymorfních kombinačních obvodů [7]. Na obrázku 3.1 jsou vstupní data označena jako data_in; první výstup je značen jako data4_out a druhý jako data8_out.

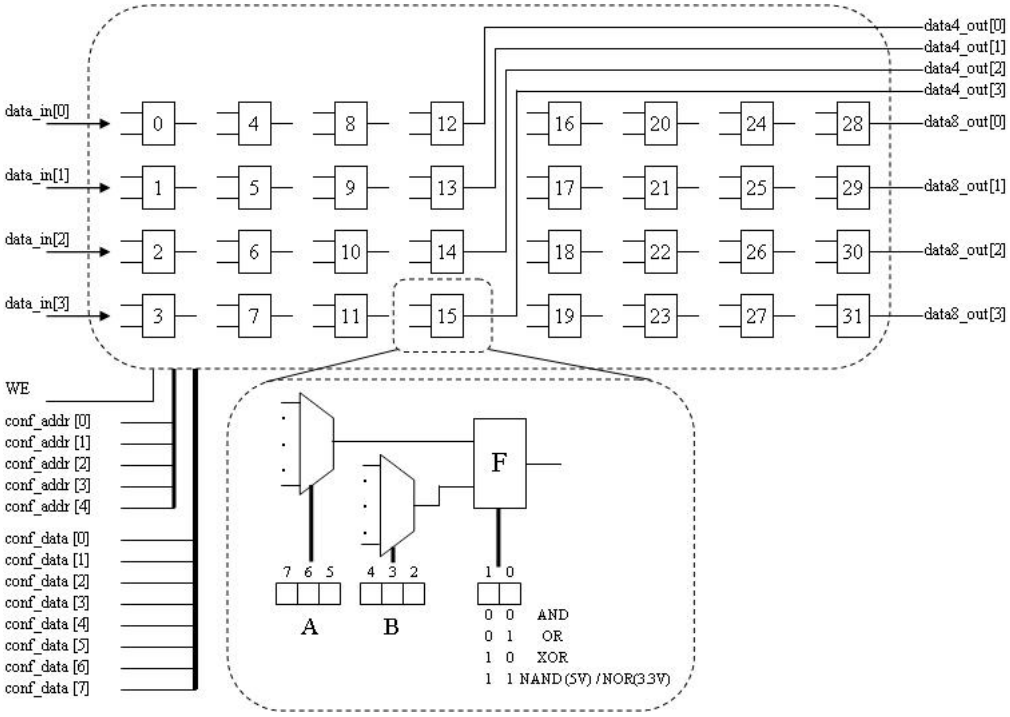
3.2 Konfigurace jednotlivých členů

Jak je vidět na obrázku 3.1, každý logický člen je konfigurován pomocí 8 bitů. Nejnižší 2 bity určují logickou funkci, zakódování je uvedeno v tabulce 3.1. Poté následují dvě trojice bitů, kde každá z nich určuje výstup logického členu, ke kterému je připojen jeden ze vstupů konfigurovaného členu. Každý ze vstupů hradla může být připojen pouze k logickým členům umístěným vlevo od něj, a to pouze o max. 2 sloupce zpět. Tuto situaci demonstruje obrázek 3.2.

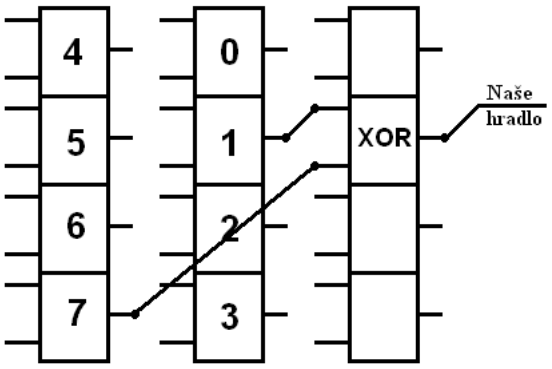
Logická funkce	Hodnota dec	Hodnota bin
AND	0	00
OR	1	01
XOR	2	10
NAND(5V)/NOR(3,3V)	3	11

Tabulka 3.1: Určení logické funkce polymorfního hradla v REPOMO

[1] Někdy je čip také označován jako REPOMO32.



Obrázek 3.1: Vnitřní struktura REPOMO (převzato z [7])



Obrázek 3.2: Určení hodnoty vstupu polymorfního hradla v REPOMO

Kapitola 4

FITkit

Platforma FITkit vznikla na Fakultě informačních technologií s cílem umožnit studentům, aby mohli navrhovat a prakticky realizovat nejen softwarové, ale také hardwarové projekty či dokonce celé aplikace [16].

4.1 Důležité komponenty

FITkit obsahuje 16-bitový nízkopříkonový mikrokontrolér rodiny MSP430 firmy Texas Instruments [13] a další periferie. Nejvýznamnějším obvodem na kitu je FPGA (Field-Programmable Gate Array) programovatelné hradlové pole XC3S50-4PQ208C řady Spartan 3 firmy Xilinx. Ačkoliv pohledem na velikost FPGA obvodu se může zdát, že se jedná o zastaralou technologii, opak je pravdou. Pouzdro PQ208 bylo zvoleno záměrně - jednak kvůli jednoduššímu osazení, jednak kvůli ceně. Rodina Spartan-3, společně s rodinou Virtex, představuje moderní rekonfigurovatelné řešení. Zatímco obvody rodiny Spartan jsou určeny pro levnější aplikace, rodina Virtex nabízí vysoce výkonné řešení pro náročné specifické úlohy [11].

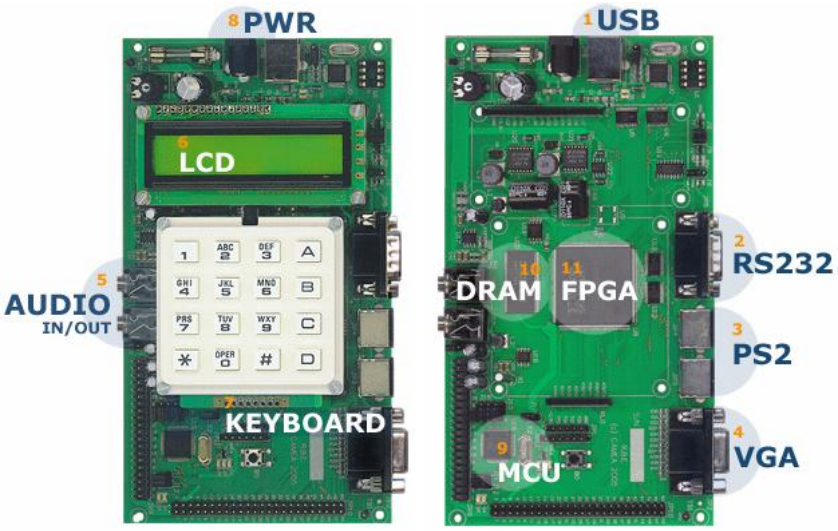
Další důležitou součástí kitu je USB převodník. Ten poskytuje dva kanály (A a B) umožňující komunikaci po USB a tím dva nezávisle konfigurovatelné kanály. Převodník je kompatibilní s USB 2.0 Fulistl Speed. Podporované režimy jednotlivých kanálů:

- MPSSE (Multi-Protocol Synchronous Serial Engine Interface) - speciální režim, který umožňuje implementaci synchronních sériových protokolů (I2C, SPI, JTAG, atd.). Rychlost toku dat může být až 5,6 Mbit/s. MPSSE je k dispozici pouze na kanálu A.
- FIFO (bitbang) - paralelní přenos dat.
- UART - seriový přenos (RS232,RS422,RS485).

Kanál A, který je připojen k FPGA obvodu, umožňuje komunikovat s PC v libovolném z podporovaných režimů. Uvnitř FPGA obvodu lze tedy např. vytvořit zařízení, které bude možné po I2C ovládat z PC. Jinou možností je napojit kanál A na řadič seriového kanálu a z PC zařízení ovládat přes virtuální COM port. Naopak, v FPGA může sloužit pouze jako „drát“ mezi zařízením připojeným k FITkitu a počítačem. Ke komunikaci s kitem je možné využít:

- knihovny libkitclient, která umožňuje přímý přístup k FTDI bez nutnosti zjišťovat přiřazený COM port a
- tzv. virtuálního COM portu, který je automaticky vytvořen při připojení FITKitu.

Kanál B je připojen k programovacím pinům mikrokontroléru (RESET,TST) a dále k pinům seriového rozhraní (RxD, TxD). Pomocí tohoto kanálu lze tedy mikrokontroler programovat a komunikovat s ním přes terminálový program. Možnostem komunikace s mikrokontrolérem se věnuje dokument Komunikace s FITkitem [14].



Obrázek 4.1: Pohledy na FITkit (převzato z [15])

4.2 Knihovna libkitclient

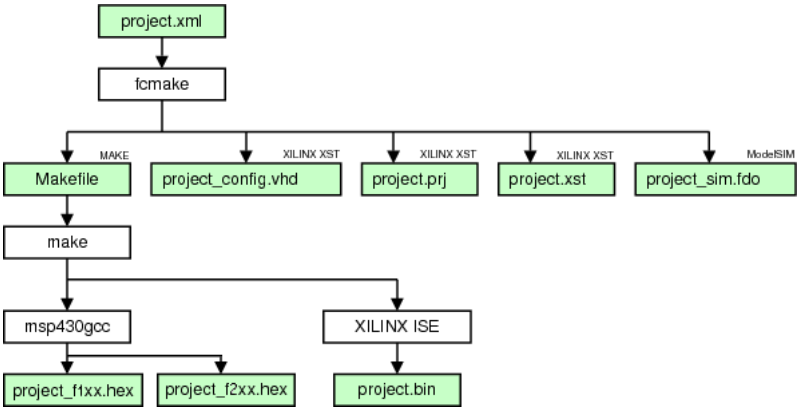
Knihovna libkitclient je multiplatformní knihovna pro správu a interakci s FITKity. Je napsána v C++ a existuje pro ni bindings pro jazyk Python, napsaný pomocí systému SWIG, který dovoluje snadnou rozšiřitelnost do dalších skriptovacích jazyků. Knihovna poskytuje jednotné API v prostředí Windows i Linux. Ve Windows je implementována nad knihovnou FTD2XX.dll, v prostředí Linux nad modifikovanou verzí libftdi.

Mezi přednosti patří umožnění přímého přístupu k FITkitu přes USB, čímž odpadá nutnost složitě konfigurovat virtuální COM porty, jejichž přiřazení je poněkud nedeterministické. Další výhodou je možnost pracovat s FITkitem z vlastní aplikace [12].

4.3 Překladový systém

Z důvodu sjednocení překladu pod systémem Windows a Linux byl vytvořen jednoduchý překladový systém založený na XML souborech. XML soubor (typicky projekt.xml) popisuje projekt, závislosti na dalších zdrojových souborech a umožňuje definovat specifická nastavení. Kromě překladu slouží XML pro generování seznamu aplikací a zjišťování informací o konkrétní aplikaci.

Schéma překladového systému je zobrazeno na obrázku 5.1. XML soubor slouží jako vstup programu fcmake, který na základě svého obsahu vygeneruje skripty pro XILINX ISE a Modelsim a Makefile soubor, se kterým je možné již pracovat pomocí standardních nástrojů. XML soubor obsahuje seznam všech zdrojových souborů a knihoven, které projekt pro překlad potřebuje, parametrů apod. Vygenerovaný soubor Makefile obsahuje pouze informace týkající se překladu konkrétního projektu a dále se odkazuje na sadu překladových



Obrázek 4.2: Schema překladového systému (převzato z [10])

pravidel umístěnou v adresáři base. XML soubor tvoří tři části:

- popis projektu,
- část obsahující informace pro generování kódu pro mikrokontroler (MCU),
- část s informacemi potřebnými pro vygenerování konfigurace pro FPGA.

Popis projektu obsahuje stručný název projektu, popis projektu, jméno autora, email a aktuální verzi (revizi), která obsahuje datum poslední modifikace.

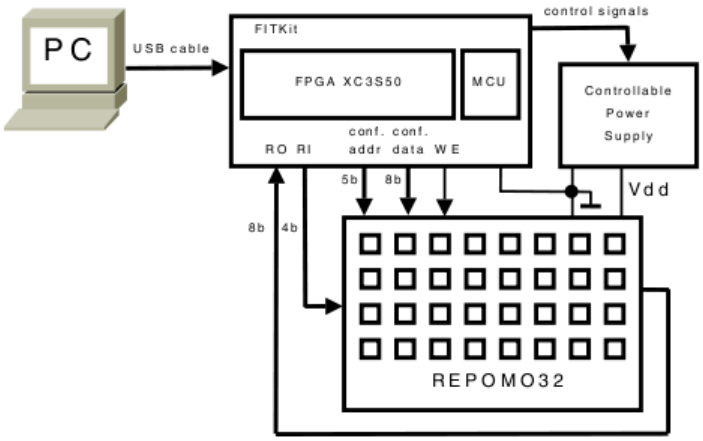
MCU sekce slouží ke specifikaci souborů, které jsou zapotřebí pro překlad kódu pro mikrokontroler a případných dalších parametrů.

FPGA sekce používá zcela stejný princip definice zdrojových souborů a závislostí jako MCU sekce, tzn. elementy <file>, <files> a <include>. Na rozdíl od MCU je zapotřebí brát ohled na pořadí vkládání souborů, neboť ModelSIM není schopen řešit automaticky závislosti [10].

4.4 Ovládání REPOMO pomocí FITkitu

K tomu, aby bylo poskytnuto uživatelsky příjemné rozhraní, je REPOMO připojeno k desce zvané FITkit, viz obrázek 4.3. FITkit je vybaven malým FPGA (Xilinx FPGA Spartan 3 XC3S50), mikrokontrolérem a různými periferními zařízeními. Obvod FPGA pracuje na frekvenci 6 MHz. V aktualní implementaci je REPOMO přímo připojeno na FPGA, které pracuje s napájením 3,3 V. Je ověřeno, že toto napětí postačuje pro řízení REPOMO pracujícího na napěťových úrovních 3,3-5 V. Software pro PC byl vyvinut tak, že může komunikovat s FPGA pomocí USB standardu. Software je zodpovědný za konfiguraci REPOMO, vytváření vstupních signálů a čtení výstupních hodnot. FITkit byl vybrán proto, že poskytuje požadované rozhraní a uspokojuje požadavky komunikační propustnosti. FITkit užívá FIT (fakulta informačních technologií) k výzkumu a výuce (používá se již více než 1000 kusů) a navíc je dobře provedena technická podpora. FITkit může ovládat zdroj napájecího napětí, které generuje Vdd pro REPOMO [7]. Proto, aby bylo možné podporovat

napájecí napětí pro dvoufunkční polymorfní hradla (3,3 V a 5 V), ale také toto napájecí napětí různě měnit, je potřeba programovatelného napájecího zdroje.



Obrázek 4.3: Blokové schéma rozhraní FITkit - REPOMO (převzato z [7])

4.5 Zapojení REPOMO za pomoci FITkitu

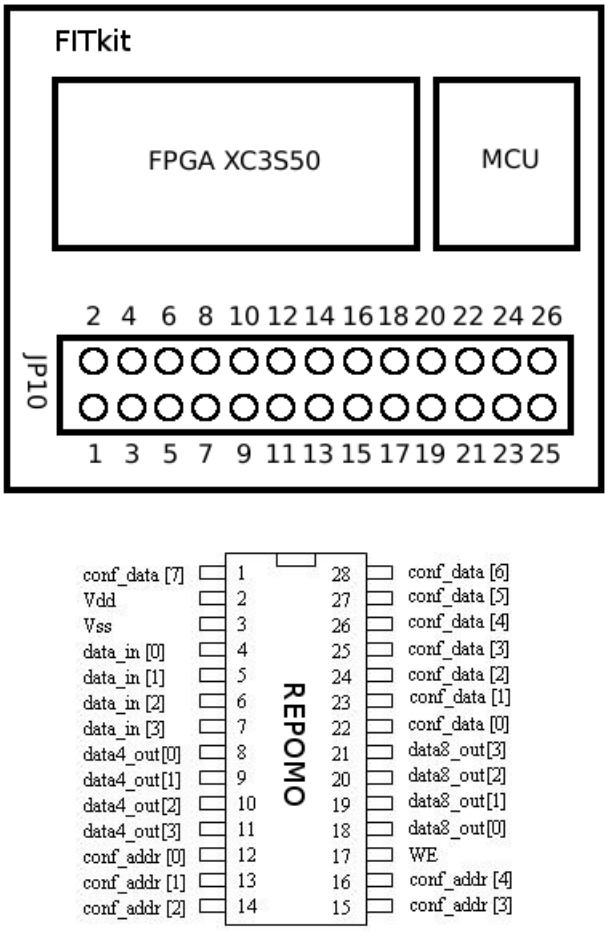
Pro snadné zapojení je potřeba:

- FITkit,
- REPOMO,
- nepájivé kontaktní pole,
- vodiče,
- počítačové konektory,
- 1 x rezistor 10kΩ

Pro propojení FITkitu s čipem REPOMO je potřeba vyvést řídící signály z konektoru JP10 na FITkitu a přivést je na požadované vývody čipu REPOMO. Zapojení viz obrázek 4.4. Jednotlivé signály lze snadno vyvést pomocí vodičů a počítačových konektorů. Pro jednoduché připojení těchto signálů k REPOMO je dobré vložit čip REPOMO do napájivého kontaktního pole. Příklad hotového zapojení viz obrázek 4.5.

4.6 Programování REPOMO

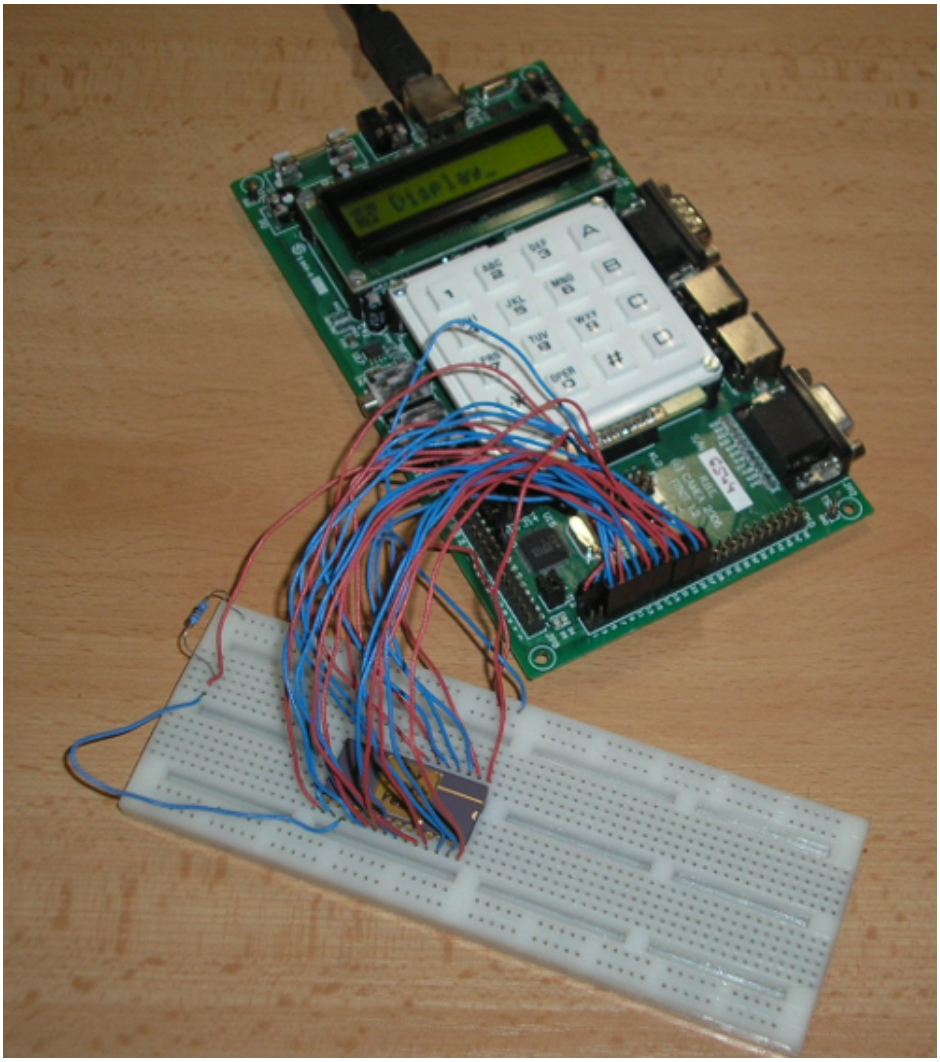
Jednou ze základních komponent pro programování REPOMO je knihovna libkitclient, která přistupuje přímo k FITkitu. Proto, aby bylo možno přeposílat data do čipu, bylo nutno vytvořit speciální obod (v jazyce VHDL). Naštěstí tento obvod již existoval dříve. Jedná se o konečný automat, který obsahuje stavy podle potřeby vykonávané funkce. Chceme-li



Popiska JP10:

1 - Vdd=5 V
2 - Vdd=3,3 V
3 - Vss pro 5 V
4 - Vss pro 3,3 V
5 - conf_addr[0]
6 - conf_addr[1]
7 - conf_addr[2]
8 - conf_addr[3]
9 - conf_addr[4]
10 - WE (přes rezistor 10kΩ připojen na VSS)
11 - conf_data[0]
12 - conf_data[1]
13 - conf_data[2]
14 - conf_data[3]
15 - conf_data[4]
16 - conf_data[5]
17 - conf_data[6]
18 - conf_data[7]
19 - data8_out[0]
20 - data8_out[1]
21 - data8_out[2]
22 - data8_out[3]
23 - data_in[0]
24 - data_in[1]
25 - data_in[2]
26 - data_in[3]

Obrázek 4.4: Zapojení FITkitu a REPOMO



Obrázek 4.5: Příklad hotového zapojení FITkit - REPOMO

zapsat konfiguraci čipu, využijeme stav SWriteConf, který projde celé pole programovatelných hradel a zapíše jednotlivé konfigurace pro každé hradlo. Chceme-li vyčíst hodnoty z REPOMO, využijeme stav SEvaluate, který vyčte hodnoty výstupu data8_out. Tyto stavy jsou volány pomocí předem napsaných funkcí v jazyce C, které využívají už zmíněnou knihovnu libfitkit. Autorem funkcí a kódu v jazyce VHDL je Ing. Vašíček.

Kapitola 5

Grafické uživatelské rozhraní - GUI

Grafické uživatelské rozhraní, ve zkratce GUI (Graphical User Interface), slouží ke komunikaci člověka s počítačem a v současné době patří mezi nejdůležitější úlohy výzkumu v oblasti počítačů, a to z následujících důvodů:

- Komunikace člověka se strojem značně ovlivňuje efektivitu používání počítačů v nejrůznějších odvětvích lidské činnosti.
- Komunikace člověka s počítačem je jednou z nejméně propracovaných oblastí procesu „práce s počítačem“.
- Komunikace člověka s počítačem je současně nejužším místem při provádění většiny „běžných činností“ při práci s počítačem.
- V komunikaci se strojem je velký potenciál pro zlepšení [17].

GUI je uživatelské rozhraní, které umožňuje ovládat počítač pomocí interaktivních grafických ovládacích prvků. Na monitoru počítače jsou zobrazena okna, ve kterých programy zobrazují svůj výstup. Uživatel používá klávesnici, myš a grafické vstupní prvky jako jsou menu, ikony, tlačítka, posuvníky, formuláře a podobně.

První grafické uživatelské rozhraní (WIMP) bylo vyvinuto v roce 1973 ve vývojových laboratořích společnosti Xerox. Oblibu mezi uživateli získalo spolu s počítači Mac kolem roku 1984 a následně i v Microsoft Windows. Kromě grafických existují i jiná uživatelská rozhraní:

- textové uživatelské rozhraní (s menu, tlačítky a myší),
- příkazový řádek (příkazy se zadávají jejich zapsáním pomocí klávesnice),
- braillský řádek,
- hlasová rozhraní a další [3].

5.1 Ovládací prvek

Ovládací prvek (anglicky widget nebo také anglicky control) je na počítači základní element pro interakci programu s uživatelem. Ovládací prvek je vizuálně ztvárněn a obvykle slouží pro manipulaci s daty v daném programu.

Rodina běžných ovládacích prvků se vyvinula ze systému uživatelského rozhraní Xerox Alto firmy PARC. Různé implementace těchto základních prvků jsou obvykle v balíku widget toolkit, který programátoři používají k tvorbě uživatelského rozhraní (GUI).

Ovládací prvky jsou také někdy označovány jako virtuální, pro vhodné odlišení virtuálního tlačítka, na které lze kliknout kurzorem myši, od tlačítka např. na klávesnici.

Podobný koncept (ale odlišným způsobem) zprostředkovává desktop widget, malá specializovaná GUI aplikace, která poskytuje některé vizuální informace a/nebo nabízí jednoduchý přístup k často používaným funkcím a aplikacím, jako například zobrazování hodin, kalendář, novinky, kalkulačka apod.

Dále se potkáváme s modálním, nebo nemodálním chováním. Modální dialog neumožňuje pokračovat v programu, dokud není ukončen (např. tlačítkem OK). Nemodální dialog průběžně vrací řízení programu, a umožňuje z hlediska uživatele interaktivně působit na prvky (měnit jejich vlastnosti a chování) [3].

Termín byl poprvé použit v uživatelském rozhraní v projektu Athena v 80. letech 20. století. Toto slovo bylo zvoleno, protože „všechny podobné termíny byly již zabrané a neoznačovaly přesně daný význam“ [4].

5.1.1 Seznam běžných ovládacích prvků

- Výběrové a zobrazovací prvky
  - Tlačítko (button)
    - Přepínací tlačítko (toggle button)
    - Zaškrtávací pole (check box)
    - Přepínač (radio button)
  - Šoupátko (slider)
  - Seznam (listbox)
  - Číselník (spinner)
  - Rozbalovací seznam (drop-down list)
  - Nabídka (menu)
    - Místní nabídka (context menu)
    - Pie menu
  - Lišta nabídek (menu bar)
  - Panel nástrojů (toolbar)
  - Pás karet (ribbon)
  - Kombinované pole (combobox)
  - Ikona (icon)
  - Strom (tree view)
  - Mřížka (grid view)
- Navigace
  - Panel (tab)
  - Posuvník (scrollbar)

- Textový výstup
  - Textové pole (textbox)
  - Kombinované pole (combobox)
    - Našeptávač
- Výstup
  - Popisek (label)
  - Kontextová nápověda (tooltip)
  - Balónová nápověda (balloon help)
  - Stavový řádek (status bar)
  - Indikátor průběhu (progress bar)
  - Informační řádek - (infobar)
- Okna
  - Okno (window)
    - Modální okno (modal window)
    - Dialogové okno (dialog box)
    - Palette window
      - Inspector window

5.2 Implementační jazyk

5.2.1 Výběr implementačního programovacího jazyka a knihovny s grafickými elementy

V dnešní době máme na výběr z celé řady jak programovacíh jazyků, tak i knihoven s grafickou nádstavbou. Například:

Programovací jazyky

- C++,
- C#,
- Python,
- Java,
- Smalltalk,
- Perl,
- Ruby.

Knihovny s grafickými elementy

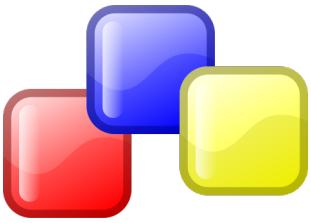
- wxWidgets,
- Qt,
- Motif,
- TCL/TK,
- GTK.

Zvoleným implemetačním jazykem pro GUI pro platformu REPOMO, byl jazyk C++ a knihovna s grafickými elementy wxWidgets. Důvod výběru byl jednoduchý: S tímto prostředím jsem obeznámen díky zapsanému semináři jazyka C++ a vytvořením několika projektů ve wxWidgets.

5.2.2 wxWidgets

wxWidgets („Windows and X widgets“, dříve známý jako wxWindows) je free software/open source multiplatformní widget toolkit. Je to knihovna základních elementů pro tvorbu grafického uživatelského rozhraní (GUI).

Vývoj začal v roce 1992 díky Julianu Smartovi, který je dodnes vývojářem jádra.



Obrázek 5.1: Logo wxWidgets (převzato z [2])

wxWidgets umožňuje zkompilovat a spustit program na několika počítačových platformách s minimálními nebo žádnými změnami kódu. Zahrnuje systémy jako Windows, Macintosh, Linux/Unix (X11, Motif, a GTK+), OpenVMS a OS/2. Verze pro embedded systémy je ve vývoji.

Knihovna je implementována v C++, ale její používání je možné v mnoha běžně používaných programovacích jazycích, mezi nimi jsou například: Python (wxPython), C#, Erlang (wxErlang), Haskell (wxHaskell), Lua (wxLua), Perl (wxPerl), Ruby (wxRuby), Smalltalk (wxSqueak), Java (wx4j) a také JavaScript (wxJS).

wxWidgets je nejlépe popsán jako nativní toolkit. Místo napodobování grafiky prvků používá nativní grafické prvky na podporovaných platformách.

20. února 2004 vývojáři wxWindows oznámili, že projekt změnil jméno na wxWidgets jako výsledek tlaku Microsoftu. Julian Smart chtěl respektovat obchodní známku Microsoftu ve Velké Británii - Windows [2].

Kapitola 6

Návrh a specifikace GUI pro platformu REPOMO

Tato kapitola obsahuje návrh a specifikaci návrhu GUI. Návrh popisuje požadavky na GUI a tyto požadavky jsou řešeny podrobněji ve specifikaci návrhu.

6.1 Návrh

Cílem této práce bylo navrhnout GUI pro platformu REPOMO, které by mělo umět vytvářet nové projekty, možnost jejich uložení a pozdějšího načtení. Každý z projektů by měl obsahovat: Kreslící plochu pro návrh schématu obvodu, ovládací prvky pro možnou simulaci obvodu či jeho přímé nahrání a vypsání hodnot z REPOMO.

6.2 Specifikace návrhu

Před spuštěním GUI (aplikace) bude potřeba naprogramovat FITkit konečným automatem, který má na starosti funkčnost přístupu k REPOMO.

Pro snadnou obsluhu bude GUI obsahovat menu a toolbar (zrychlená volba).

Menu a zrychlená volba zajistí práci s projekty, možnost nastavení a v případě nouze zavolání nápovědy.

Vytvořené projekty budou reprezentovány záložkami pro intuitivní obsluhu. Rozpracované projekty bude možno ukládat ve formě textových souborů a do každého z těchto souborů bude zapána vnitřní struktura obvodu. Načtení rozpracovaného projektu opět vytvoří uloženou strukturu.

Každá záložka bude obsahovat kreslící plochu definovanou bitmapou (obrázkem), ve které bude vykresleno pole hradel odpovídající struktuře čipu REPOMO. Toto pole bude repezentováno polem datových struktur, kde každá ze struktur bude uchovávat zapojení jednotlivých vstupů hradla a jeho logickou funkci.

Ovládání návrhu schématu bude pomocí myši, kdy vzniklé údalosti od myši budou zakreslovat návrh do definované bitmapy. Pro spojování hradel budou kresleny spojnice (úsečky), které bude možno editovat pomocí dialogového okna. Nastavení logické funkce hradla bude taktéž pomocí dialogového okna a zvolená hodnota logické funkce bude zakrelena přímo do hradla.

Navržené schéma by mělo být před nahráním do čipu REPOMO odzkoušeno a to za pomoci simulace. Pro odsimulování pro jeden vstupní vektor bude použita tzv. rychlá simu-

lace, která pro zadaný vstupní vektor a zadané vstupní napětí zobrazí výstupy jednotlivých zapojených hradel přímo v kreslící ploše. Pro simulaci více vstupních vektorů bude použita tzv. běžná simulace. Ta po zadání možnosti buď vlastních nebo všech vektorů zobrazí výsledky simulace jak pro 3,3 V tak i 5 V v oddělené části.

Pro získání hodnot z REPOMO bude potřeba otevřít spojení s FITkitem za pomocí ovládacích tlačítek, zapsat strukturu obvodu na čip a vyčíst hodnoty. Zda-li budou data z REPOMO validní zjistíme tak, že získáme výstupy jak ze simulace, tak i z REPOMO a výsledky porovnáme. Výsledek porovnání by měl být zřetelně zobrazen.

Kapitola 7

Implementace GUI a ovládání REPOMO

Jako implementační jazyk byl zvolen jazyk C++ a knihovna s grafickými objekty wxWidgets, jak je uvedeno výše. Tato kapitola je věnována popisu implemtace a řešení problémů v průběhu práce.

7.1 Struktura projektu

Jelikož řešení projektu obsahuje velké množství tříd, pojďme si je rozebrat postupně a podrobněji. Hirearchie tříd je uvedena na obrázku 7.1. Více o projektové dokumentaci spuštěním souboru index.html ve složce Repomo/doc/html na přiloženém DVD.

7.1.1 APPLICATION

Jedná se o třídu, kde její hlavní činností je vytvoření a řízení aplikace, ale mimo jiné také programuje FITkit před spuštením aplikace a kontroluje, zda-li je FITkit připojen. O obsluhu této třídy se stará třída wxApp.

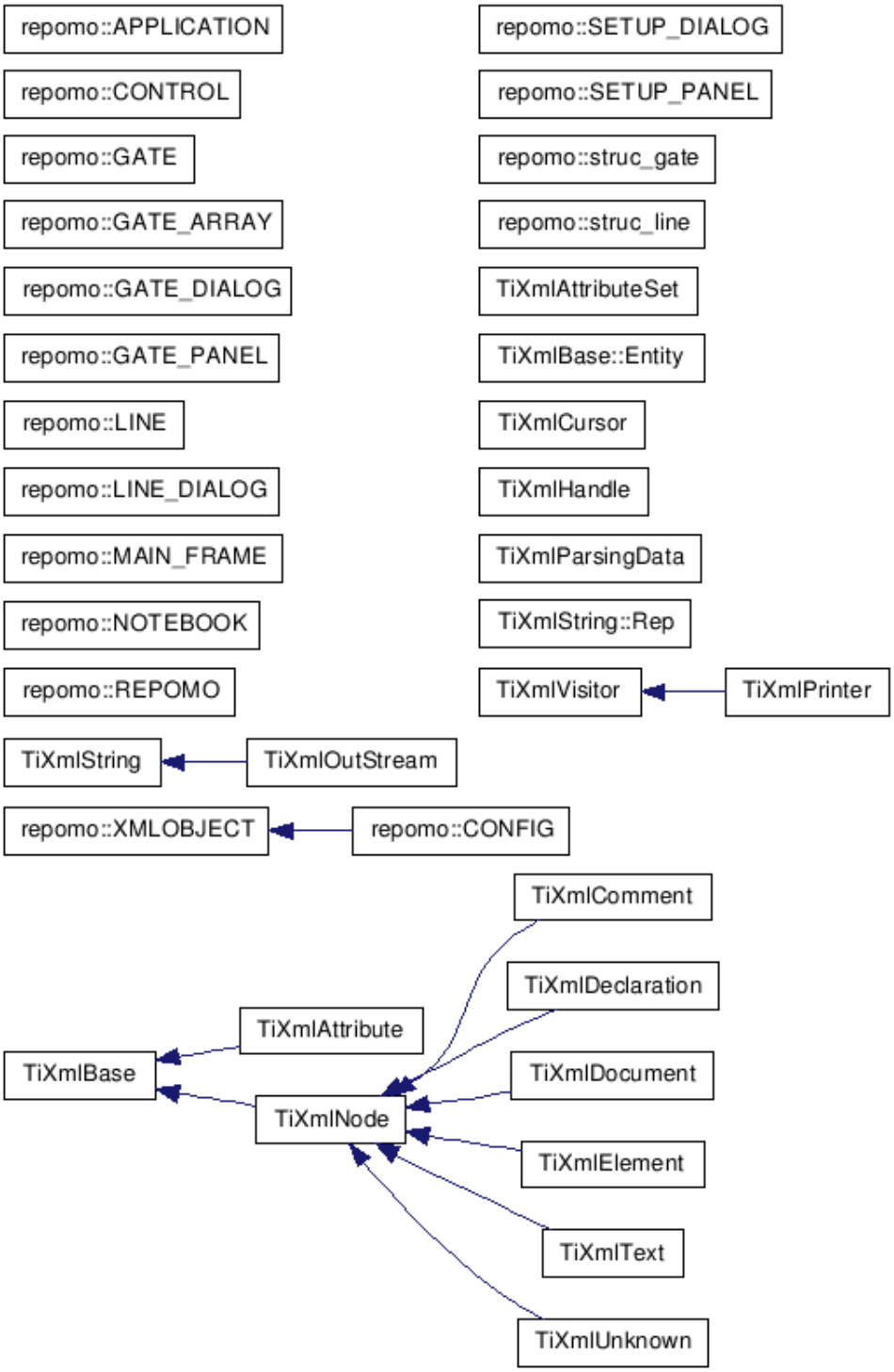
7.1.2 MAIN_FRAIM

Jedná se o jednu z nejdůležitějších částí projektu, a to objekt hlavního okna, do kterého se bude vše vykreslovat. Obsahuje hlavní menu se základní prací s aplikací a projektem, toolbar nebo-li zrychlenou volbu, stavový řádek zobrazující zprávy při nějaké události, a to nejdůležitější panel záložek do kterého se přidávají jednotlivé záložky (nové projekty). Ukazatele na záložky se ukládají do zásobníku a při případném jejím zavření či změně jazyka se prochází zásobník a provede se daná událost.

Hlavní okno je třídy wxFrame, v ní jsou umístěny všechny komponenty.

Hlavní menu je třídy wxMenu a obsahuje položky:

- Projekt - nabízí základní práci s projektem (viz obrázek 7.2):
  - Nový - vytvoří novou záložku (nový projekt).
  - Načíst - vytvoří novou záložku (rozpracovaný projekt).

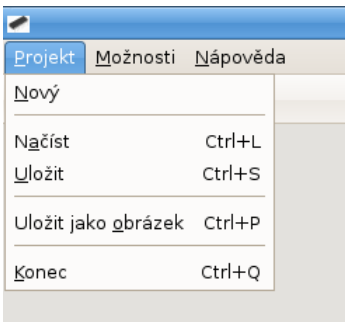


Obrázek 7.1: Hirearchie tříd

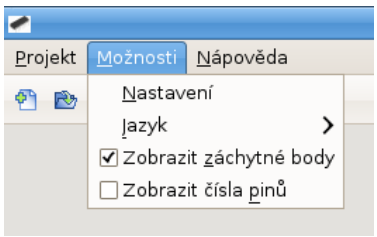
  - Uložit - uloží rozpracovaný projekt jako textový soubor.
  - Uložit jako obrázek - uloží kreslící plátno jako obrázek.
  - Konec - ukončí aplikaci

- Možnosti - možnosti aplikace (viz obrázek 7.3):
  - Nastavení - zobrazí dialogové okno pro barevné nastavení kreslícího plátna.
  - Jazyk - výběr ze 2 možností a to angličtina nebo čeština, čeština jako defaultní.
  - Zobrazit záchytné body - zobrazí u vstupů a výstupů záchytné body, kde lze kreslit spojnici.
  - Zobrazit čísla pinů - zobrazí u vstupů a výstupů čísla vývodů na čipu.

- Nápověda - zobrazí nápovědu nebo informace o aplikaci:
  - Manuál - zobrazí předem vytvořenou nápovědu v chm formátu.
  - O programu - zobrazí základní údaje o aplikaci.



Obrázek 7.2: Ukázka menu a položky Projekt



Obrázek 7.3: Ukázka menu a položky Možnosti

Nový vytvoří novou záložku, tato záložka se naplní komponentami a bude přidána do panelu záložek.

Načíst je poněkud složitější. Tato volba vytvoří novou záložku, dojde k jejímu naplnění komponentami a přidání do panelu záložek. Potom se získají data ze zvoleného textového souboru a naplní se pole datových struktur. Zvolený textový soubor obsahuje údaje o každém z 32 hradel (viz obrázek 7.4). Každé hradlo je reprezentováno datovou strukturou obsahující údaje o připojení vstupů a logické funkci. Logická funkce je dána tabulkou 3.1, vstupy jsou dány dle obrázku 3.2. Údaje ze souboru uloží do pole struktur struc_gate a

```
#polymorfni median/parita 3 vstupy (i2,i1,i0,-) 1 vystup

223 --- 012 ---
450 --- --- 261
#--- --- --- ---
#--- --- --- ---
--- 462 --- ---
--- --- 411 743
--- --- --- ---
--- --- 672 ---
--- --- --- ---
660 --- --- ---
```

Obrázek 7.4: Struktura souboru popisující uložený projekt

údaje zobrazí. Prázdné řádky nebo řádky začínající znakem # se přeskakují. Znak # se bere jako komentář. Pomlčka vyjadřuje nezadanou hodnotu.

Uložit: Uloží rozdělaný projekt, jako výše uvedený textový soubor. Postupně prochází pole a zapisuje údaje do zvoleného souboru.

Uložit jako obrázek: Uloží kreslící plochu jako obrázek (bitmapu).

Nastavení: Zobrazí dialogové okno pro výběr barev komponentů v kreslící ploše. Více u objektu SETUP_DIALOG 7.1.6.

Jazyk: Podle zvoleného jazyka buď nechá implicitní názvy v anglickém jazyce nebo přidá do katalogu český překlad. Využívá se možnosti wxLocale přímo wxWidgest a pro překlad byl použit program Poedit. Pro překlad se prochází zásobník a překlad se provádí pro každou záložku zvlášť.

Zobrazit záchytné body: Reinicializuje kreslící bitmapu a zobrazí nebo schová záchytné body, které jsou součástí vykreslených objektů hradel.

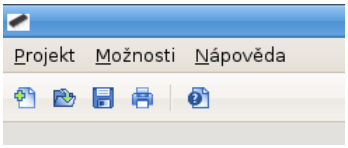
Zobrazit čísla pinů Reinicializuje kreslící bitmapu a zobrazí nebo schová čísla vývodů čipu, které jsou součástí vykreslených objektů vstupů či výstupů.

Manuál: Zobrazí nápovědu ve formátu chm, kde byl pomocí programu HTML Help Workshop vytvořen zdrojový soubor hhp pro možnost této nápovědy.

O programu: Pomocí wxAboutDialogInfo zobrazí základní informace o aplikaci a autorovi.

Toolbar je třídy wxToolBar, umístěn pod menu viz obrázek 7.5. Jedná se o rychlou volbu, kde kliknutím přímo na danou ikonu můžeme přímo provádět základní práci s projektem. Obsahuje:

- Nový
- Načíst
- Uložit
- Uložit jako obrázek
- Nápověda



Obrázek 7.5: Zrychlená volba (toolbar)

Panel záložek je třídy wxAuiNotebook, což pracuje stejně jako normální záložky, ale navíc obsahuje zavírací tlačítko a události tohoto tlačítka.

Stavový řádek je třídy wxStatusBar, kde se zobrazují různá hlášení při provádění událostí.

Všechny přednastavené údaje jsou načteny z XML souboru pro nastavení, kde tento soubor upravuje třída CONFIG. Pro práci z XML soubory byla zvolena open source knihovna TinyXML.

7.1.3 TinyXML

Jedná se o jednoduchý, malý XML parser, který může být snadno integrován do různých programů. TinyXml se vyvinul zpětnou vazbou z komunity a stál práci mnoha přispěvatelů. Je to jednoduchý, stabilní, základní XML parser používaný spousty open source a komerčních produktů [1].

7.1.4 CONFIG

Tato třída má na starosti uložit nebo vyčíst data z XML souboru, v našem případě settings.xml. Tento soubor obsahuje, uživatelem nebo programátorem při prvním spuštěním, předdefinované údaje jako:

- Jazyk
- Záchytné body
- Zobrazit čísla pinů
- Barvy - více v SETUP_DIALOGU
  - Popiska
  - Hradlo
  - Spojnice
  - Označení hradla
  - Označení spojnice
  - Popiska rychlé simulace

7.1.5 NOTE_BOOK

Další velmi důležitou část aplikace představuje nová záložku. O obsluhu se stará třída wxScrolledWindow a to proto, že umožňuje scrollování při změně rozlišení. Záložku můžete vidět na obrázku 7.6 a obsahuje:

- kreslící plochu,
- roletky pro vstupní hodnoty rychlé simulace,
- výstupní pole rychlé simulace,
- ovládání rychlé simulace,
- ovládání běžné simulace
- zápis konfigurace a výpisu hodnot z REPOMO,
- vstupní vektory,
- výstupní data.

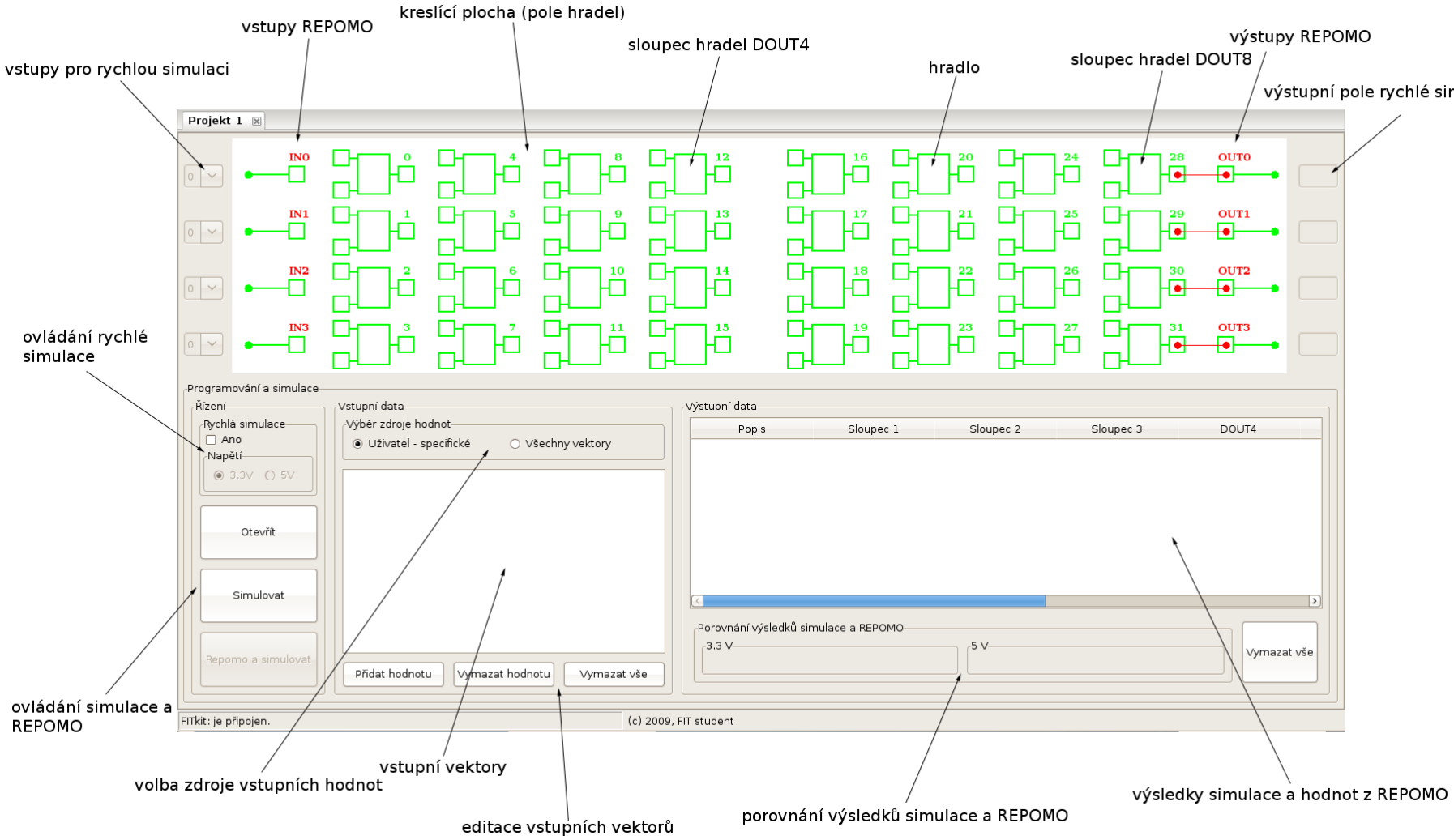
Kreslící plocha je řízena více třída, proto si je pojďme projít postupně:

Třída GATE_PANEL vytvoří wxPanel, do kterého umístí bitmapu pomocí třídy GATE_ARRAY (popsána v následujícím odstavci). Úkolem GATE_PANELU je reakce na události od myši, a to kliknutí pravého či levého tlačítka, a také se stará o vykreslení logických hodnot u rychlé simulace, více v odstavci 7.1.5.

Třída GATE_ARRAY vytvoří bitmapu a pomocí objektů GATE a LINE zobrazí pole hradel a k nim příslušné vstupy a výstupy. Jednotlivé objekty GATE a LINE se ukládají do zásobníků, které se prochází po události kliknutí na kreslící plochu. Klikneme-li jednou levým tlačítkem myši na vstupy či výstupy hradel nebo na vstupy či výstupy REPOMO, kreslíme spojnici a přidáváme ji do zásobníku spojnic. Pokud se v blízkosti dvojkliku levého tlačítka myši nachází úsečka (zjištění pomocí analytické geometrie vzdálenosti bodu od přímky, a to za pomoci zásobníku spojnic), smaže se a vymaže údaje o sobě ze zásobníku. Klikneme-li pravým tlačítkem myši do oblasti hradla, zobrazí se nám dialogové okno pro volbu logické funkce, kterou má na starost objekt GATE_DIALOG (více v podkapitole 7.1.7). Klikneme-li pravým tlačítkem myši na oblast přímky (opět zjištění pomocí analytické geometrie), zobrazí se dialogové okno pro editaci přímky, kterou má na starost objekt LINE_DIALOG (více v podkapitole 7.1.8).

Ovládání rychlé simulace se nachází v části záložky s ovládáním (viz obrázek 7.7). Chceme-li spustit rychlou simulaci, zaškrneme Ano, díky tomu se povolí roletky a výstupy rychlé simulace po stranách kreslícího plátna a tím se ocitneme v režimu rychlé simulace. Na schématu čipu můžeme vizuálně zkontrolovat vypočtené logické hodnoty, které závisí na zadaných vstupních kombinací (roletky vlevo) a na výběru velikosti napětí u polymorfních hradel. Metoda pro tuto simulaci projde pole struktur struc_gate a postupně zjišťuje výsledné hodnoty propojených hradel a až výslednou hodnotu zapíše do polí po pravé straně kreslící plochy.

Ovládání běžné simulace se nachází v části záložky s ovládáním (viz obrázek 7.7). Máme zde 3 tlačítka. Tlačítko simulace již dobře známe z rychlé simulace, ale pokud chceme odsimulovat více vektorů najednou, musíme zvolit jiný druh simulace. K tomu slouží oblast vstupních dat, kde po jejich zadání stačí zmáčknout tlačítko „simulovat“ a zobrazí se nám výstupní hodnoty v oblasti výstupu. Výstupní hodnoty hradel se postupně počítají



Obrázek 7.6: Ukázka záložky (nového projektu)

průchodem polem struktur a získáním těchto hodnot dostáváme postupně všechny potřebné výstupy, které jen přidáme do výstupní oblasti.

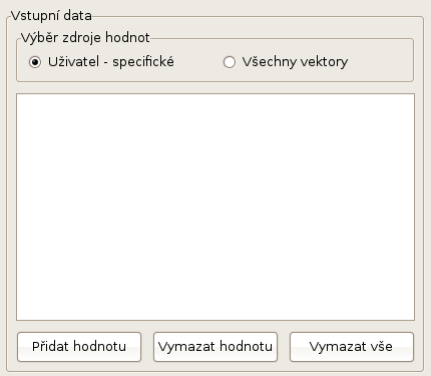
Obrázek 7.7: Řízení simulace a programování čipu REPOMO

Zápis konfigurace a výpisu hodnot z REPOMO se nachází v části záložky s ovládáním (viz obrázek 7.7). Aby mohlo dojít ke komunikaci s FITkitem a tím následně s čipem REPOMO, je potřeba navázat spojení pomocí tlačitka Otevřít. Poté můžeme tlačítkem „Repomo a simulovat“ zapsat konfiguraci obvodu, vyčíst hodnoty z čipu REPOMO a získané hodnoty porovnat se simulací. Příklad výsledku porovnání viz obrázek 7.8.

Obrázek 7.8: Oblast porovnání simulace a hodnot z REPOMO

Vstupní vektory slouží pro vlastní zadání testovacích vektorů schématu viz obrázek 7.9. Zvolíme-li uživatelem specifické vektory, tak tlačítky pod tímto boxem editujeme vstupní kombinace. Můžeme jich zadat maximálně 16, a to v binárním kódu. Zvolíme-li možnost všech vektorů, tak pomocí předem vytvořeného pole řetězců určíme vstupní vektory jako kombinace od 0 do 15 v binární podobě.

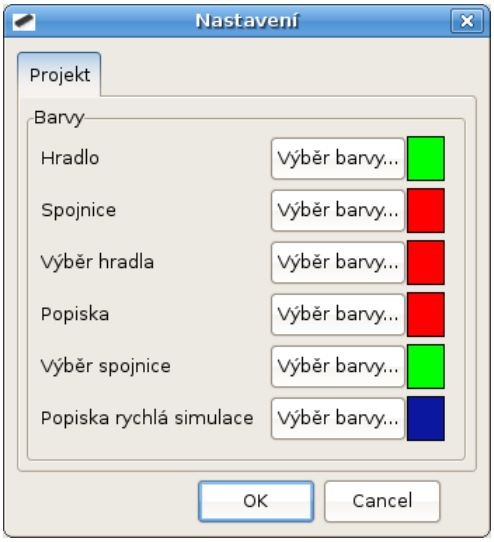
Výstupní data se pouze zobrazují ve výstupním boxu. Jsou uspořádána do 8 sloupců a 4 řádků, tak jak se nachází na kreslící ploše. Tím můžeme snadno najít námi hledané výstupy daného hradla. Jednotlivé výstupní hodnoty jsou odsimulovány pro všechny hodnoty, které jsou zadány ve vstupním boxu tzn. pokud bude 8 vstupních vektorů, u každého zapojeného hradla bude ve výsledku 8 hodnot. Těchto 8 hodnot bude zapsaných v pořadí, ve kterém byly vstupní hodnoty zadány a to od prava. Výstupy jsou označeny DOUT4 a DOUT8.



Obrázek 7.9: Ukázka oblasti pro zadání vstupních vektorů

7.1.6 SETUP_DIALOG

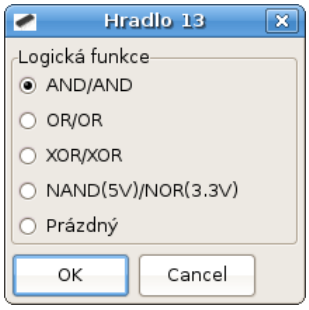
Jedná se o dialogové okno pro možnost nastavení barevného schématu projektu. Můžeme zde nastavit barvy viz obrázek 7.10. Jednotlivé barvy jsou načteny z XML souboru a zobrazeny v panelu vedle názvu barvy. Tento panel je objekt třídy SETUP_PANEL, kde o překreslení se stará třída sama, stačí ji pouze výběr barvy pomocí klasického dialogového okna wxColourDialog. Vybrané schéma se uloží do XML souboru a dojde k překreslení všech existujících záložek.



Obrázek 7.10: Ukázka dialogového okna pro nastavení

7.1.7 GATE_DIALOG

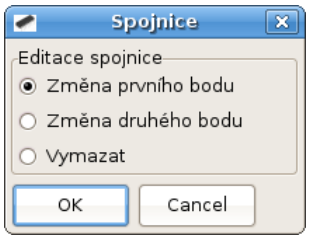
Jedná se o dialogové okno pro možnost výběru logické funkce hradla viz obrázek . Výbraná logická funkce se uloží do pole struktur a překreslí se celá kreslící plocha. Pokud bude vybrána volba „prázdný“, na daném místě se vymaže hodnota v poli struktur a dojde k překreslení kreslící plochy.



Obrázek 7.11: Ukázka dialogového okna pro volbu logické funkce

7.1.8 LINE_DIALOG

Jedná se o dialogové okno pro možnost editace spojnice (viz obrázek 7.12) v kreslící ploše. Výběrem dané funkce můžeme spojnici buď vymazat nebo změnit její první či druhý bod kliknutím na jiné místo dané pravidly.



Obrázek 7.12: Ukázka dialogového okna pro editaci spojnice

7.1.9 CONTROL

Jedná se o základní řídící prvke celé aplikace. Jde o třídu, kde se nachází všechny metody pro řízení projektu, a to překreslení aplikace při znevalidnění okna, dále metody pro zjištění pozice kliknutí tlačítka myši, kreslení spojnice, důležitou roli samozřejmě hrají načtení a uložení projektu či reinicializace kreslící plochy a v neposlední řadě ovládní REPOMO. Tato třída pro ovládání REPOMO pouze využívá již předem napsaných metod umístěných ve třídě **REPOMO**.

7.2 Řízení REPOMO

O řízení REPOMO se stará třída REPOMO, která obsahuje již předem vytvořené metody pro otevření spojení s FITkitem, zápis konfigurace, výpis hodnot z REPOMO a uzavření spojení.

7.2.1 Metody řízení

Metody **open()** a **close()** se starají o spojení FITkitu s PC přes USB rozhraní.

Metoda **writeconf(unsigned char *conf)** po předání řetězce obsahující konfiguraci o každém hradle z pole hradel se spojí s FITkitem, pomocí zvoleného stavu konečného auto-

matu zvolí zápis konfigurace, pošle dané data na potřebné piny čipu. Pro zápis konfigurace musí být každé hradlo připojeno a proto nevyužitá hradla se připojují na kombinaci 0 0 0.

Metoda **evaluate(unsigned char *reqtab)** po předání ukazatele na řetězec vyčte hodnoty z REPOMO pro předchozí konfiguraci. Vždy jsou testovány všechny kombinace vektorů a to pro oboje napětí. Pokud není přepínaný zdroj pro FITkit, tak vždy výsledek jedné z hodnot napětí neodpovídá. Tato možnost je volána údálostí tlačítka „REPOMO a simulovat“ v záložce. Podle zadaných vstupních vektorů buď zobrazí všechny získané vektory nebo u zadaných uživatelem vybere ty, které zvolil. To se děje tak, že výsledné hodnoty jsou převedeny na binární podobu a vždy nejvyšší čtveřice bitů odpovídá testovanému vektoru. Poté zobrazí jak hodnoty simulace tak i vyčtené hodnoty z REPOMO do oblasti výstupních dat.

Metoda **writeeval(unsigned char *conf, unsigned char *reqtab)**, která je použita v GUI pro programování REPOMO, vznikla sloučením metod writeconf a evaluate.

7.2.2 Možnost jiné platformy než FITkit pro komunikaci s REPOMEM

Na obrázeku 4.3 vidíme určitou abstrakci mezi PC a REPOMO. Budeme-li chtít zvolit jinou platformu pro jeho řízení, stačí pouze přepsat metody ve třídě REPOMO a mělo by vše fungovat správně i na jiné platformě. Jen je nutné dodržet názvy všech metod a jim předávané parametry. Názvy metod:

- open()
- close()
- writeconf()
- evaluate()
- writeeval()

Pokud nejsou dodrženy názvy, nemusí vše fungovat správně. Další možností je nové názvy metod přepsat přímo ve třídě **CONTROL**.

7.3 CMAKE

Proto, aby bylo možno přeložit navrženou aplikaci, bylo zapotřebí použití programu CMAKE.

CMAKE je program, který po zapsání pravidel do souboru CMakeLists.txt a následným spuštěním programu cmake s parametrem -G a uvedením za ním do uvozovek cílovou platformu, vytvoří makefile pro překlad. Pro windows je potřeba přeložit pro MinGW. Poté stačí pod danou platformou program přeložit. Pod linuxem dochází ke kompilaci jiným způsobem. Více o spuštění aplikace a instalaci potřebných modulů je uvedeno v souboru INSTALL.txt ve složce Repomo na přiloženém DVD.

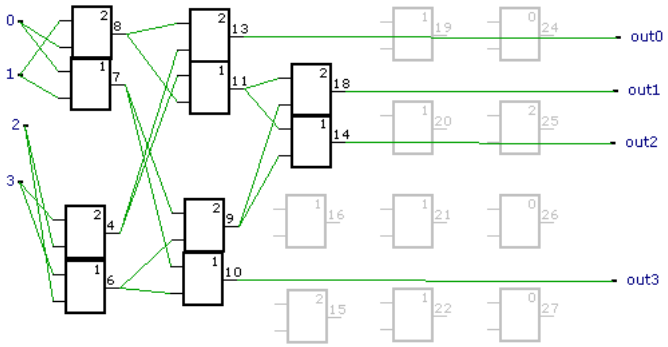
Kapitola 8

Experimentální ověření

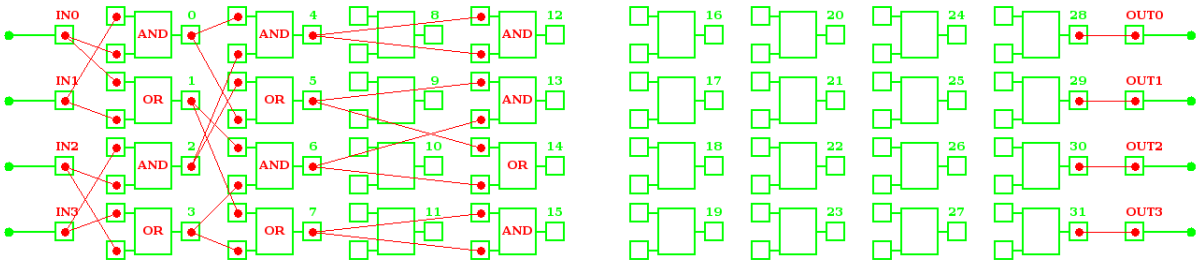
Posledním úkolem bylo navrhnout už ve funkčním GUI schéma zapojení, toto zapojení v něm přímo otestovat a ověřit správnost výsledků.

8.1 Příklad

Jako příklad uvedu řadící 4b síť převzatou z [9]. Jedná se o jednoduchý obvod, který má seřadit nuly a jedničky. Na obrázku 8.1 můžeme vidět schéma zapojení řadící sítě zobrazené v programu Chromozome viewer a na obrázku 8.2 toto schéma překreslené do hotového GUI pro platformu REPOMO.



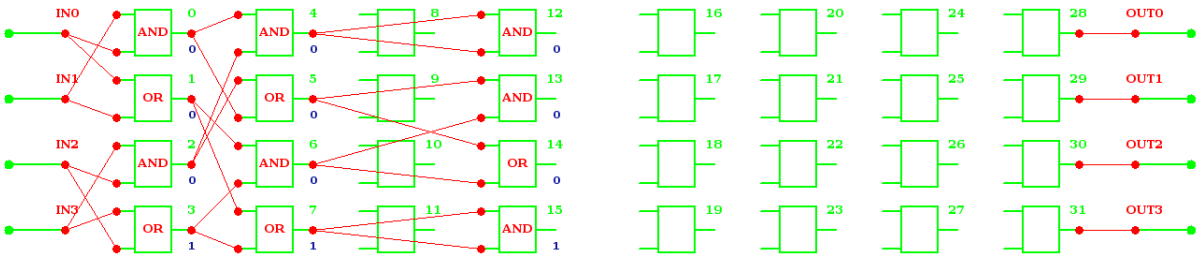
Obrázek 8.1: Příklad schématu řadicí 4b sítě v Chromozome viewer



Obrázek 8.2: Příklad schématu řadicí 4b sítě v GUI pro platformu REPOMO

8.1.1 Test A

Obvod seřazuje vstupní vektor a to tak, že na nejnižších pozicích výstupu jsou nuly a poté následují jedničky. Pro jeden vstupní vektor můžeme obvod vyzkoušet v módu rychlé simulace. Byla zvolena kombinace 0100 a nyní se podíváme, jaký bude výsledek na výstupu DOUT4 (viz obrázek 8.3). Výstupem by měla být hodnota 0001. Zvolené napětí můžeme pominout, jelikož je obvod sestaven pouze z logických součtů a součinů, které nejsou polymorfními hradly, a proto jejich logická funkce je neměnná.

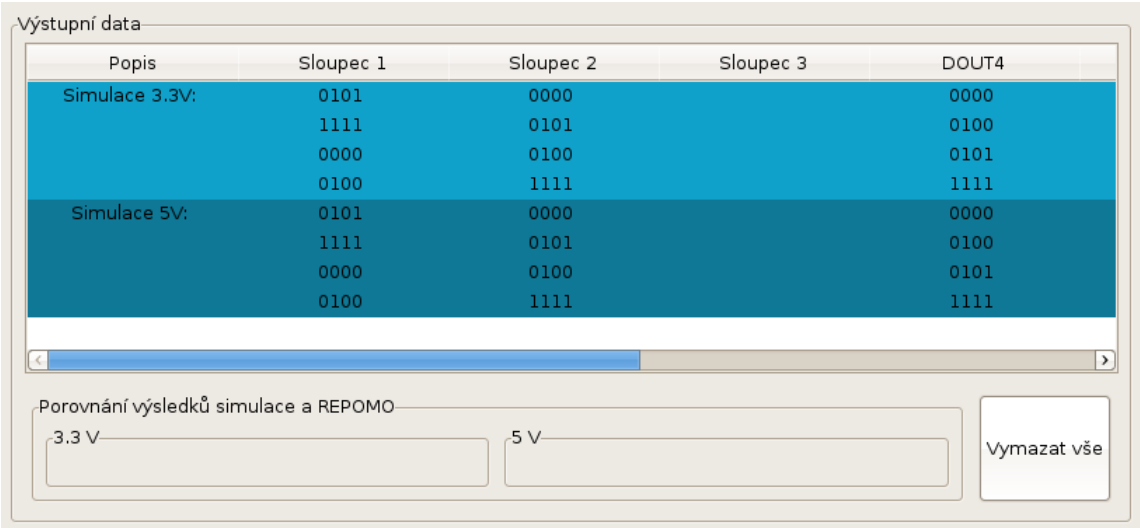


Obrázek 8.3: Výsledek rychlé simulace pro vstupní vektor 0100

Jak můžeme vidět na obrázku 8.3, tak výsledek odpovídá předpokladu. Pro kontrolu můžeme výsledek zkontrolovat vizuálně, a to pomocí zobrazených výstupních hodnot u jednotlivých hradel.

8.1.2 Test B

Nyní zkusme odsimulovat více vektorů najednou. Pro tuto volbu použijeme běžný mód simulace a zvolíme zadání vektorů uživatelem. Pomocí tlačítka „Přidat hodnotu“ přidáme kombinace 1100, 0100, 1110 a 1000 do oblasti vstupních hodnot. Poté stačí pouze stisknout tlačítko „Simulovat“ a výčtem hodnot z výstupní oblasti zjistíme výdledek (viz obrázek 8.4). Očekávané výsledky jsou: 0011, 0001, 0111, 0001.

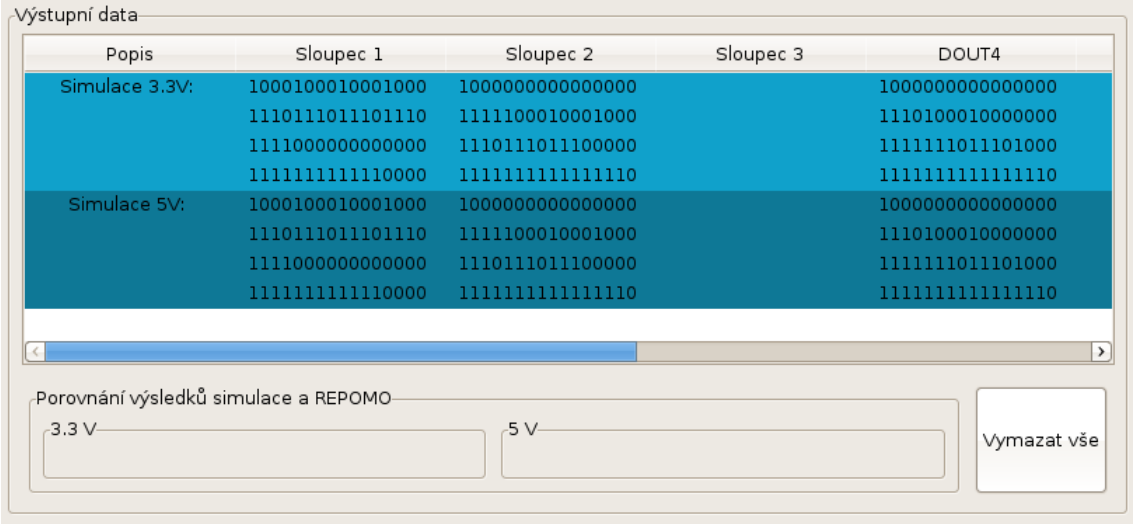


Obrázek 8.4: Výsledek běžné simulace pro vstupní vektory 1100, 0100, 1110, 1000

Výsledek simulace je vždy zobrazen jak pro napětí 3,3 V, tak i 5 V. Jelikož obvod neobsahuje polymorfní logické členy, výsledky jsou totožné pro oboje napětí. Projdeme-li postupně sloupce výsledků výstupu simulace ve sloupci DOUT4 (viz obrázek 8.4) z pravé strany, tak zjistíme, že první výslednou hodnotou je 0011. Tím se nám seřadil první zadaný vektor, a to 1100. Výsledek odpovídá předpokladu i u zbylých vektorů a díky tomu je potvrzeno správné chování simulace.

8.1.3 Test C

Další test provedeme na všech možných kombinacích vstupních vektorů. Pro tuto volbu použijeme opět běžný mód simulace, ale jako vstupní data zvolíme možnost „Všechny vektory“. Poté stačí pouze stisknout tlačítko „Simulovat“ a výčtem hodnot z výstupní oblasti zjistíme výdledek (viz obrázek 8.5).



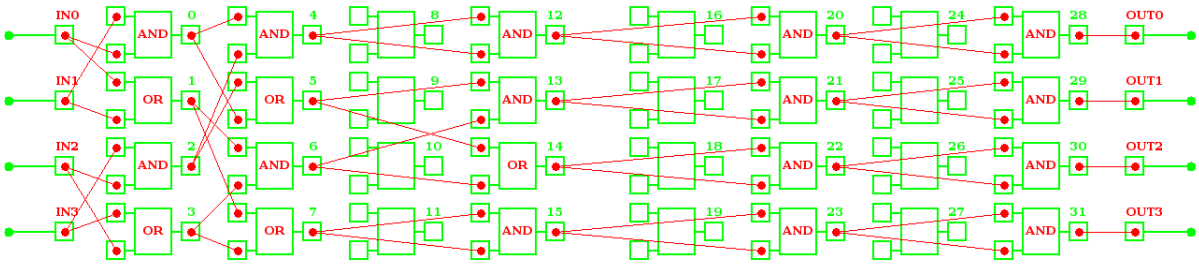
Obrázek 8.5: Výsledek běžné simulace pro všechny kombinace vstupních vektorů

Projdeme-li opět postupně výsledek simulace, zjistíme, že simulace pracuje opět správně.

8.1.4 Test D

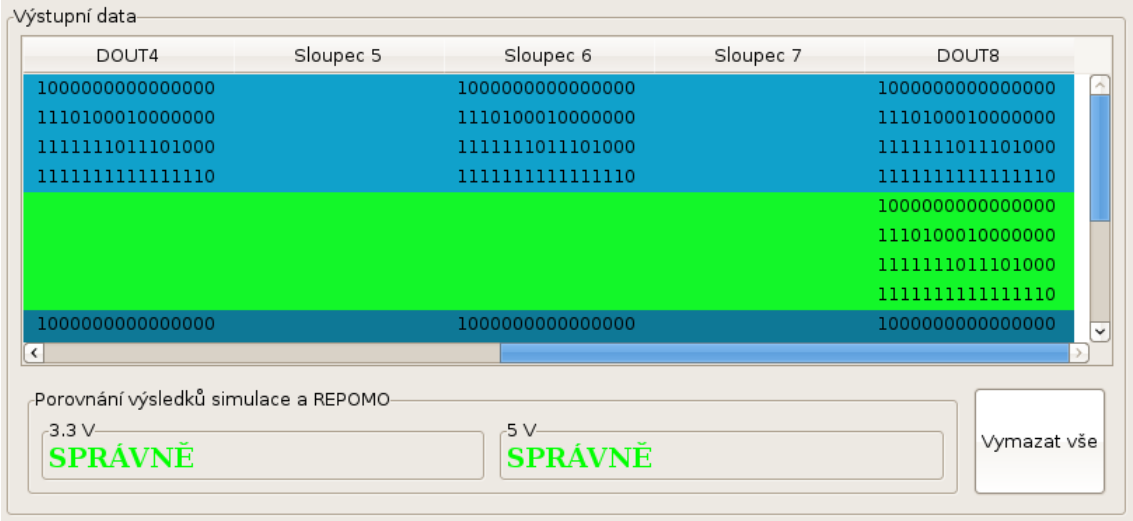
Předešlé testy byly pouze softwarového rázu. Nyní je obvod odzkoušen přímo v čipu REPOMO. Pro tento test je potřeba pozměnit schéma zapojení, a to jen vyvedením DOUT4 výstupů (4. sloupec hradel) k poslednímu sloupci hradel (viz obrázek 8.6). Tato potřeba vzniká z důvodu získání hodnot z REPOMO, protože metody pro vyčtení hodnot pracují pouze s vývody data8_out (viz obrázek 4.4).

Pomocí tlačítka „Otevřít“ navážeme spojení s FITKITEM, tím se povolí tlačítko REPOMO a simulace, kde po jeho stisknutí vyčteme hodnoty z REPOMO pro zadané vstupní vektory. Jelikož obvod zkoušíme pro všechny možné kombinace vstupních vektorů, získáme vždy šestnáctice. Program provede nejdříve simulaci pro oboje napětí a poté vyčte hodnoty z REPOMO. Výsledky porovná a do oblasti Porovnaní výsledků simulace a REPOMO ve výstupních datech zobrazí kontrolní hlášení. Výstupní data jsou odlišena barvou pozadí, a to modrá barva zvýrazní výsledky simulace a zelená zvýrazní výsledky z čipu REPOMO. Světlé barvy určují výsledek pro 3,3 V a tmavší pro 5 V. Jelikož box pro zobrazení výstupních dat má omezenou velikost, je potřeba se posunout na sloupec DOUT8 pomocí



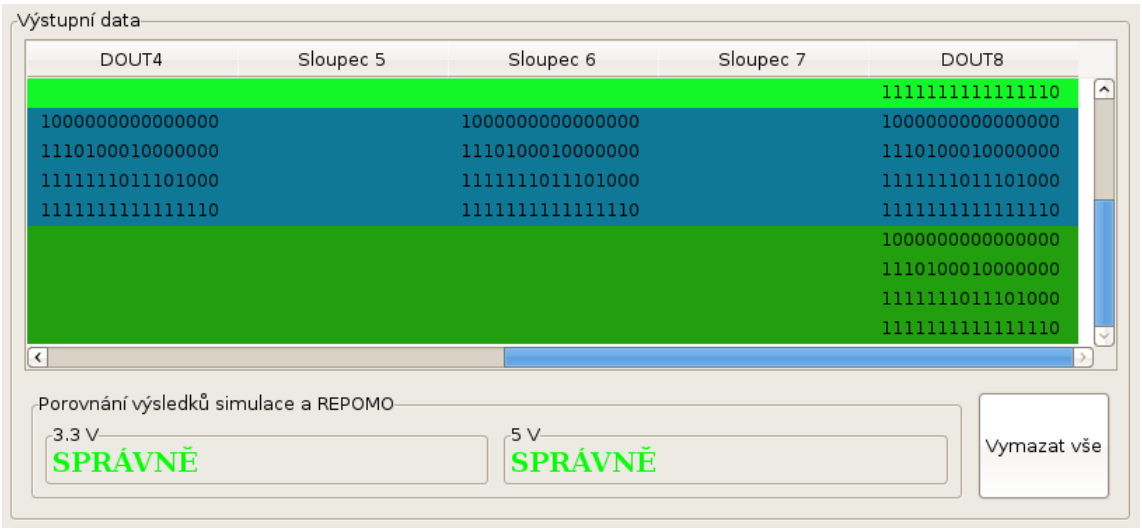
Obrázek 8.6: Upravené schéma zapojení pro simulaci a výčet hodnot z REPOMO

posuvníku. Zde výsledky nejsou popsány, ale rozlišeny barevně, viz obrázek 8.7 a obrázek 8.8.



Obrázek 8.7: Výsledek výpisu hodnot z REPOMO a porovnání s výsledkem simulace pro 3,3 V

Z obrázků 8.7 a 8.8 jasně vyplývá, že porovnané hodnoty jsou si rovny a tím simulace i výčet hodnot z čipu REPOMO jsou správné.



Obrázek 8.8: Výsledek výpisu hodnot z REPOMO a porovnání s výsledkem simulace pro 5 V

Kapitola 9

Závěr

Polymorfní elektronika je jistě jednou ze zajímavých a velmi užitečných oblastí jak v evolučním návrhu, tak i v běžné elektrotechnice. Použitím polymorfních logických členů můžeme konstruovat obvody s nižšími nároky na velikost, spotřebu apod. Jistě je to zajímavé odvětví a uvidíme, co budoucnost ukáže.

Jako hlavní přínos této práce spatřuji v možnosti snadného a lehkého programování REPOMO, možnosti návrhu a simulace polymorfních obvodů.

Při práci na tomto projektu jsem si zdokonalil poznatky z programovacího jazyka C++, práci s knihovnou wxWidgets a prohloubil jsem si znalosti elektroniky v oblasti polymorfních rekonfigurovatelných logických členů.

Literatura

[1] Grinninglizard: TinyXml. [online], 2009.
URL http://www.grinninglizard.com/tinyxml/

[2] Wikipedia: wxWidgets. [online], Poslední modifikace 21. 3. 2009.
URL http://cs.wikipedia.org/wiki/WxWidgets

[3] Wikipedia: Grafické uživatelské rozhraní. [online], Poslední modifikace 24. 4. 2009.
URL http://cs.wikipedia.org/wiki/Grafick%C3%A9_u%C5%BEivatelsk%C3%A9_rozhran%C3%AD

[4] Wikipedia: Ovládací prvek (počítač). [online], Poslední modifikace 27. 3. 2009.
URL http://cs.wikipedia.org/wiki/Ovl%C3%A1dac%C3%AD_prvek_(po%C4%8D%C3%ADta%C4%8D)

[5] Janíček, F.: *Vlastnosti a použití architektury Cell Matrix*. Diplomová práce, Fakulta informačních technologií VUT v Brně, 2004.

[6] Růžička, R.; Sekanina, L.; Prokop, R.: Physical Demonstration of Polymorphic Self-checking Circuits. In *Proc. of the 14th IEEE Int. On-Line Testing Symposium*, IEEE Computer Society, 2008, ISBN 978-0-7695-3264-6, s. 31–36.
URL http://www.fit.vutbr.cz/research/view_pub.php?id=8652

[7] Sekanina, L.; Růžička, R.; Vašíček, Z.; aj.: REPOMO32 - New Reconfigurable Polymorphic Integrated Circuit for Adaptive Hardware. In *Proc. of the 2009 IEEE Symposium Series on Computational Intelligence - Workshop on Evolvable and Adaptive Hardware*, IEEE Computational Intelligence Society, 2009, ISBN 978-1-4244-2755-0, s. 39–46.
URL http://www.fit.vutbr.cz/research/view_pub.php?id=8898

[8] Sekanina, L.; Stareček, L.; Kotásek, Z.; aj.: Polymorphic Gates in Design and Test of Digital Circuits. *International Journal of Unconventional Computing*, ročník 4, č. 2, 2008: s. 125–142, ISSN 1548-7199.
URL http://www.fit.vutbr.cz/research/view_pub.php?id=8587

[9] Vašíček, Z.: Evoluční návrh kombinačních obvodů zadaných tabulkou. Technická zpráva, Fakulta informačních technologií VUT v Brně, 2004.
URL www.fit.vutbr.cz/~sekanina/eva/cvic/xvasic11.pdf

[10] Vašíček, Z.: FITkit: Návody - překladový systém. Stránky projektu FITkit. [online], 2009.
URL http://merlin.fit.vutbr.cz/FITkit/docs/navody/kompilacev2.html

[11] Vašíček, Z.: FITkit: Popis hardwaru kitu - FPGA. Stránky projektu FITkit. [online], 2009.
URL http://merlin.fit.vutbr.cz/FITkit/docs/hardware/hw_fpga.html

[12] Vašíček, Z.: FITkit: Popis hardwaru kitu - libfitkit. Stránky projektu FITkit. [online], 2009.
URL http://merlin.fit.vutbr.cz/FITkit/docs/navody/libkitclient.html

[13] Vašíček, Z.: FITkit: Popis hardwaru kitu - MCU. Stránky projektu FITkit. [online], 2009.
URL http://merlin.fit.vutbr.cz/FITkit/docs/hardware/hw_mcu.html

[14] Vašíček, Z.: FITkit: Popis hardwaru kitu - USB. Stránky projektu FITkit. [online], 2009.
URL http://merlin.fit.vutbr.cz/FITkit/docs/hardware/hw_ftdi.html

[15] Vašíček, Z.: FITkit: Popis hardwaru kitu. Stránky projektu FITkit. [online], 2009.
URL http://merlin.fit.vutbr.cz/FITkit/hardware.html

[16] Vašíček, Z.: Stránky projektu FITkit. [online], 2009.
URL http://merlin.fit.vutbr.cz/FITkit/

[17] Zemčík, P.: Úvod, tvorba rozhraní, základní principy, tvorba uživatelských rozhraní - podpora přednášek, FIT VUT v Brně, 2007.

Dodatek A

Obsah DVD

```
DVD
|——/Repomo
|                |——/doc
|                |            |——/html
|                |            |                    |—— projektová dokumentace
|                |——/src
|                |            |——/fitkit
|                |            |                    |—— řízení připojení FITkitu
|                |            |——/img
|                |            |                    |—— obrázky aplikace
|                |            |——/langs
|                |            |                    |—— český překlad
|                |            |——/manual
|                |            |                    |—— /html
|                |            |                    |                    |—— nápověda
|                |            |——/savedprojects
|                |            |                    |—— uložené projekty
|                |            |—— config.cc
|                |            |—— config.h
|                |            |—— control.cc
|                |            |—— control.h
|                |            |—— gate.cc
|                |            |—— gate.h
|                |            |—— gate_array.cc
|                |            |—— gate_array.h
|                |            |—— gate_dialog.cc
|                |            |—— gate_dialog.h
|                |            |—— gate_panel.cc
|                |            |—— gate_panel.h
|                |            |—— line.cc
|                |            |—— line.h
|                |            |—— line_dialog.cc
|                |            |—— line_dialog.h
|                |            |—— main.cc
|                |            |—— main.h
```

```
|                    |                    |—— main_frame.cc
|                    |                    |—— main_frame.h
|                    |                    |—— notebook.cc
|                    |                    |—— notebook.h
|                    |                    |—— setup_dialog.cc
|                    |                    |—— setup_dialog.h
|                    |                    |—— setup_panel.cc
|                    |                    |—— setup_panel.h
|                    |                    |—— repomo.cc
|                    |                    |—— repomo.h
|                    |                    |—— repomo_unix.cc
|                    |                    |—— repomo_unix.h
|                    |—— INSTALL.txt
|                    |—— Readme.txt
|                    |—— Doxyfile
|——/TechReport
|                    |——/tex
|                    |                    |—— zdrojové soubory technické zprávy
|                    |—— ibp.pdf
```

Dodatek B

Dokumentace

B.1 Manuál

Spuštěním souboru index.html ve složce Repomo/src/manual/html se zobrazí nápověda pro obsluhu programu. Tatéž nápověda lze spustit přímmo z aplikace, a to v menu položkou Nápověda → Manuál nebo klávesou F1.

B.2 Projektová dokumentace

Spuštěním souboru index.html ve složce Repomo/doc/html se zobrazí projektová dokumentace aplikace ve formátu HTML vygenerovaná systémem doxygen.

B.3 Instalace

Postup instalace potřebných modulů a překlad aplikace je popsán v textovém souboru INSTALL.txt ve složce Repomo.