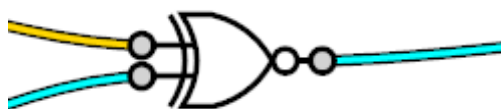


Gymnázium Jiřího Wolkerův, Prostějov, Kollárova 3



JEDNODUCHÝ SIMULÁTOR LOGICKÝCH OBVODŮ

SimLogic

Maturitní práce IVT

2010

Vedoucí práce:	Mgr. František Střídecký
Autor práce:	Petr Kalandra
Třída:	08.B

Prohlašuji, že jsem maturitní práci vypracoval samostatně s využitím uvedených pramenů a literatury a nemám závažný důvod proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb. o právu autorském.

V Prostějově dne 25.4.2010

Podpis:.....

OBSAH

1	Úvod.....	5
2	Pojmy a zkratky použité v práci.....	6
3	Analýza.....	7
3.1	Úvod analýzy.....	7
3.2	Výběr platformy.....	7
3.3	Vizuální řešení.....	8
3.3.1	Ovládací okno.....	8
3.3.2	Kreslení.....	8
3.4	Požadavky.....	9
4	Návrh.....	10
5	Popis aplikace.....	11
5.1	Standardy.....	11
5.2	Nabídky aplikace.....	11
5.3	Klávesové zkratky.....	12
5.4	Panel vlastností.....	12
5.5	Pomocná okna.....	13
5.5.1	Tabulka pravdivostních hodnot.....	13
5.5.2	Nastavení analýzy obvodu.....	14
5.5.3	O aplikaci.....	14
5.6	Hlavní pracovní plocha aplikace.....	14
5.6.1	Manipulace s objektem.....	14
5.6.2	Manipulace se spojem.....	15
5.6.3	Soupis ovládání editoru spojů.....	15
5.7	Módy simulace.....	16
5.7.1	Okamžitý mód.....	16
5.7.2	Mód simulace.....	16
6	Závěr.....	17
6.1	Souhrn.....	17
6.2	Předpokládané využití.....	17
6.3	Budoucnost projektu.....	17

7	Seznam literatury a internetových zdrojů.....	18
8	Použitý software.....	18
9	Přílohy.....	19
A	Schéma hlavní nabídky.....	19
B	Popis činnosti všech implementovaných objektů.....	20
C	Ukázková zapojení.....	25
D	Soupis obsahu přiloženého CD.....	26

1 Úvod

Během výuky matematiky a potažmo elektrotechniky zahrnuté v rámci fyziky a matematicko-fyzikálního semináře jsem si povšiml nedostatečného prostoru věnovaného Booleově logice či logickým obvodům. Toto téma bylo vždy probíráno stručně a studenti měli občas problém danou problematiku pochopit. Dle mého názoru je to zajisté škoda, neboť se jedná o základ celé moderní digitální výpočetní techniky. Například definice Turingova stroje vychází z předpokladu, že veškeré matematické operace lze zapsat pouze pomocí elementárních operací, které jsou mimo jiné jednoduše realizovatelné právě za pomoci logických obvodů. Můžeme tedy prohlásit, že za cenu složitosti obvodu je možné sestavit takřka jakýkoliv algoritmus. Možným důkazem této teorie je i existence programovatelných hradlových polí (PLD), která jsou koncipována jako jednoúčelové rychlé výpočetní jednotky, a která do jisté míry mohou nahradit i MCU. Přemýšlel jsem tedy nad tím, jak vzbudit zájem o tuto problematiku v řadách studentů a vytvořit tak vyučujícím větší prostor k výuce tohoto tématu a usnadnit ji. Hlavním předpokladem pro tuto variantu je především interaktivita a dostupnost aplikace. Dal jsem se tedy do hledání a našel spoustu simulátorů. Každý z nich měl svá pro, ale i svá proti. Mezi nejčastější proti patřila například cena aplikace, omezení pouze na základní hradla či složité ovládání. Nakonec jsem se rozhodl pro vytvoření vlastní aplikace, která by byla jednoduchá na ovládání a přesto zahrnula většinu základních logických obvodů.

Cílem této práce je seznámit čtenáře se vznikem a obsluhou aplikace, v jedné z příloh i principem některých základních logických obvodů. Problematika logických obvodů patří svým rozsahem nad rámec práce, a proto v ní nikde není hlouběji popsána. Taktéž není cílem aplikace vytvořit komplexní editor elektronických schémat.

2 Pojmy a zkratky použité v práci

.Net je platforma od Microsoftu usnadňující vývoj aplikací dostupná pro Windows, Web aj.

BCD (Binary coded decimal) je ukládání desítkových čísel do čtyřbitových bloků (nibblů).

Booleova logika definuje matematické operace sčítání, násobení a negace nad čísly 0 a 1.

CSV (Comma separated values) je textový formát určený k přenosu tabulkových dat.

C# je striktně objektový programovací jazyk z dílny Microsoftu vycházející z jazyka C++.

Distinctive shape je způsob zakreslení hradel pomocí různých tvarů.

Dynamická knihovna je část kódu oddělená od aplikace; může být načtena za chodu.

GUI (Graphical User Interface) je grafické rozhraní pro komunikaci aplikace s uživatelem.

Logický obvod je obvod složený z hradel. Jejich kombinací vznikají číslicové systémy.

MCU (Micro-controller unit) je jednočipový procesor využívaný k řešení konkrétní úlohy.

Mono je komunitní open-source projekt umožňující spustit .Net aplikace i na jiných OS.

Náběžná hrana je přechod z logického stavu 0 na 1.

Objekt je jakýkoli prvek na pracovní ploše aplikace.

Ovládací prvek je jednoúčelový prvek v GUI.

Pin umožňuje zapojení prvku do obvodu. Podle směru signálu se dělí na vstupní a výstupní.

PLD (Programovatelné hradlové pole) je soustava hradel. Slouží jako obdoba MCU.

PLD vychází z teorie, že lze sestavit libovolný logický obvod jen z NAND či NOR hradel.

Rectangular shape je způsob zakreslení hradel pomocí obdélníku a vepsaného textu.

Sestupná hrana je přechod z logického stavu 1 na 0.

Spoj je v rámci aplikace chápán jako propojení dvou pinů.

Tooltip je rámeček s textem, jež se objeví při najetí kurzoru na ovládací prvek.

Úrovně HIGH/LOW je jiné pojmenování pro logické stavy 1/0.

Uzel je místo propojení dvou či více vodičů (spojů).

3 Analýza

3.1 Úvod analýzy

Po delší úvaze, a převážně z časových důvodů, jsem se rozhodl práci rozdělit do dvou samostatných částí, a to na **Editor schémat** a **Rozšířené GUI**.

Editor schémat obsahuje třídu ovládacího prvku samotného editoru a dále pak třídy reprezentující například **Pin** obvodu nebo **Součástku**. Každý jednotlivý základní logický **obvod** dědí třídu součástky a vytváří množinu vstupních a výstupních pinů jako členské proměnné. Dále se objekt editoru stará o samotnou simulaci, výpočty či vizualizaci výsledného zapojení.

Rozšířené GUI je samotná spustitelná aplikace, která instancuje prvek editoru z příslušné dynamické knihovny a stará se o obsluhu stavového řádku, nabídek či dalších dialogů jakými jsou například nastavení analýzy obvodu či dialog vizualizace pravdivostní tabulky. Mimo jiné má na starosti také export získaných dat do formátu CSV.

Další část analýzy je vyčleněna výběru platformy, požadavkům na aplikaci, průběhu vývoje a příčinám, které vedly k výslednému řešení.

3.2 Výběr platformy

Při výběru platformy hrál hlavní roli čas. Bylo potřeba aplikaci vyvíjet rychle a efektivně ale přitom pečlivě a kvalitně. Z tohoto důvodu jsem upřednostnil jazyk C# a platformu Microsoft .NET, dnes již ve své předposlední verzi 3.5, neboť obsahovala nástroje pro jednoduchou tvorbu uživatelského rozhraní, spoustu užitečné nastavbové funkcionality, jež platforma poskytuje za cenu téměř nepatrného poklesu výkonu. Výhodou platformy je mimo jiné i snadná přenositelnost mezi operačními systémy díky komunitnímu projektu Mono. Jazyk C# jsem pak zvolil převážně pro jeho přehlednost a přirozený styl zápisu kódu.

3.3 Vizualní řešení

3.3.1 Ovládací okno

Při bližším ohledání existujících aplikací si není možné nepovšimnout vývoje nabídek a celkově designu aplikací v poslední době. Dochází k posunu aplikací blíže k běžnému uživateli a často i k omezování a následného obtěžování uživatelů pokročilých. Aplikace jsou koncipovány jako kombinace velké obrázkové lišty nástrojů a pracovní plochy. Tento přístup je zajisté funkční, ale v důsledku i omezující. Rozhodl jsem se proto ponechat ideu samostatné interaktivní pracovní plochy, ale poněkud pozměnil návrh nástrojového panelu. Nový graficky zajímavý styl panelu by byl náročný po stránce časové i designové a u takto „jednoduché“ aplikace i neefektivní. Ponechal jsem proto starší styl nabídek a doplnil jej o velké ikonky, jež by měly způsobit částečné zpřehlednění nabídky. Vizualizace pravdivostní tabulky pak využívá obrázků pro jednoznačnou vizualizaci stavu v kombinaci s tooltipem obsahujícím textový popis stavu. Původní vzhled s čísly 0 a 1 nebyl příliš přehledný.

3.3.1 Kreslení

Pro potřeby samotného simulátoru bylo nezbytné vyvinout vlastní systém ovládacích prvků odpovídající jednoduchému editoru schémat. Proto byl vytvořen ovládací prvek editoru (ve významu ovládacího prvku platformy .NET), jež je jakousi virtuální plochou pro samotné ovládací prvky schématu (Piny, Objekty, Spoje). Veškeré vykreslování těchto prvků probíhá samostatně uvnitř knihovny editoru a není využito žádných externích bitmap, což díky své vektorové podstatě zajišťuje škálovatelnost (zoomování) plochy bez ztráty kvality. Je použito vyhlazování poskytnuté platformou .NET, což může působit rozostřený obraz při malém měřítku, avšak lepší obraz při výchozím přirozeném měřítku. Editor je inspirován profesionálními editory schémat jakým je například Eagle, ale nesnaží se být tak obsáhlý. Standardům odpovídající a komplexní schematický editor by byl nad rámec této práce.

3.4 Požadavky

Cílem této práce bylo vytvořit tzv. lightweight editor a simulátor logických obvodů, který by bylo možné využít při výuce matematiky, fyziky či informatiky. Důraz byl kladen především na jednoduchost užívání a rychlost, tak aby byl snáze použitelný pro studenty i učitele a neměl vysoké nároky na výkon PC. Aplikace si také neklade za cíl být komplexním editorem. To se projevuje například překrýváním některých spojů. Tento nedostatek patří mezi další plánované body vývoje. Jedním z požadavků byla možnost exportovat výsledky simulace. Jako exportní formát jsem zvolil CSV, neboť je jednoduše generovatelný, jednoduše zpracovatelný a je podporován většinou tabulkových editorů a generátorů grafů.

4 Návrh

V této části se pokusím popsat strukturu zdrojových kódů a jeho tříd. První částí, kterou se budu zabývat je část **aplikace**. Tato část obsahuje následující třídy v abecedním pořadí:

AboutBox má na starosti zobrazení informativního dialogu *O aplikaci*.

AnalysisSettings obsluhuje dialog, který má za úkol získat uživatele požadavky exportu analýzy.

Csv je třída zajišťující průběžný a snadný export dat do souboru ve formátu CSV.

MainForm je samotné hlavní okno. Spojuje editor s GUI a předává informace v rámci aplikace.

Program vytváří hlavní okno aplikaci a obsahuje hlavní funkci a třídu programu.

TruthTable obstarává vizualizaci pravdivostní tabulky a její ukládání do souboru CSV.

Knihovna **editoru** pak obsahuje v abecedním pořadí tyto třídy:

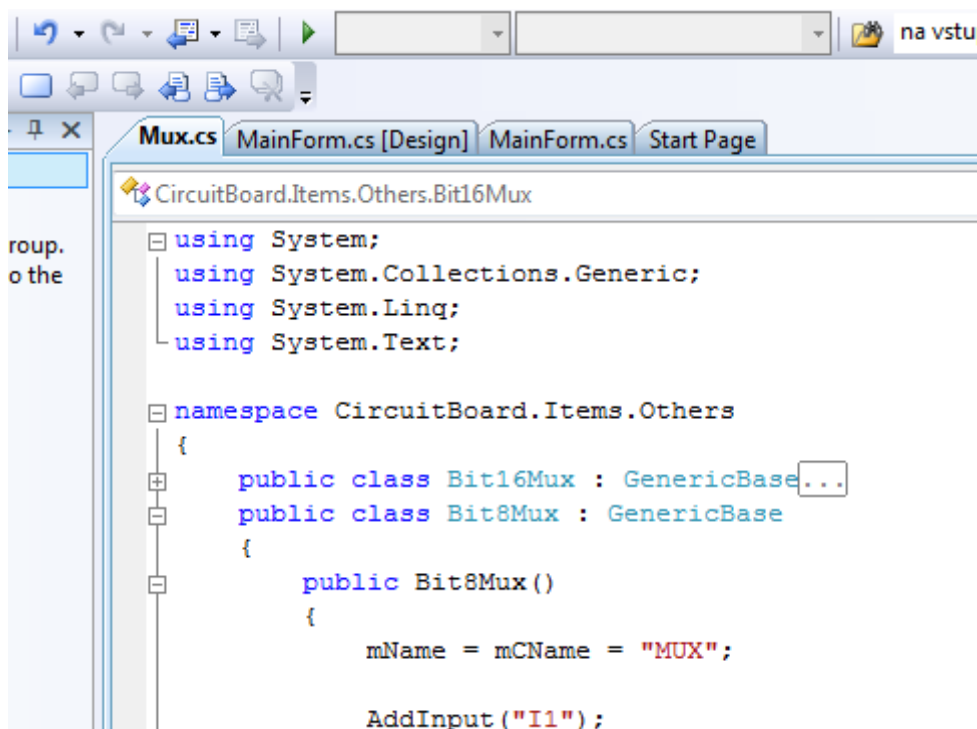
Item uchovává informace o objektu (součástce). Stará se i o její vizualizaci.

IStorage je interface pro komunikaci s exportem CSV, který se nachází v hlavní aplikaci.

Pin uchovává informace o vstupu/výstupu součástky a jejich propojení.

Scheme má za úkol vykreslení obvodu a zpracování dat ze simulace a její obsluhu.

Další třídy které dědí rozhraní **Item** jsou vlastní implementace jednotlivých prvků (**Display**).



5 Popis aplikace

Na tomto místě bych rád seznámil uživatele se základním principem činnosti a ovládáním aplikace.

5.1 Standardy

Aplikace se snaží dodržovat platné mezinárodní elektrotechnické standardy. Odchylkou je snad jen vykreslování spojů jako beziérových kubických křivek.

Pro zobrazení hradel je použita norma *distinktivní* (*Distinctive shape*) zatímco pro vykreslení složitějších obvodů norma *obdélníková* (*Rectangular shape*) s textovým popisem pinů pro větší univerzálnost.

5.2 Nabídky aplikace

V aplikaci jsou použity pouze dvě nabídky, a to nabídka hlavní a nabídka tvorby objektů.

Hlavní nabídka obsahuje především manipulaci s projekty vytvořenými v programu (nabídka *Soubor*), dále pak ovládání simulace a animace (nabídka *Simulace*) a analýzy (nabídka *Analýza*).

V nabídce *soubor* naleznete načtení/uložení projektu a také vytvoření projektu nového.

V nabídce *simulace* naleznete krokování i restart simulace a možnosti pro obsluhu animace, jako například ovládání rychlosti a její zastavení.

V nabídce *analýza* pak můžete vygenerovat tabulku pravdivostních hodnot nebo soubor CSV obsahující stavy na vstupech a výstupech v jednotlivých krocích.

Nabídka tvorby objektů obstarává seznam objektů, které je možné umístit na pracovní plochu. Jedná se zpravidla o zdroje signálu, měřicí sondy, samotné součástky či pomocné objekty jakým je například popisek.

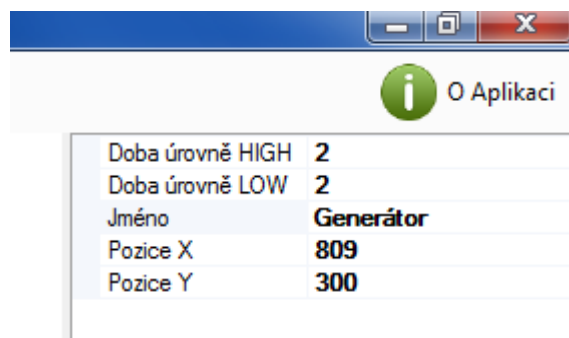
5.3 Klávesové zkratky

Zde naleznete globální klávesové zkratky, které lze použít na pracovní ploše i při aktivní nabídce. Ovládací klávesy pracovní plochy se nachází v kapitole Soupis ovládání editoru spojů.

Ctrl + N	Vytvoří nový projekt
Ctrl + S	Uloží projekt
Ctrl + O	Načte projekt
Ctrl + T	Vygeneruje tabulku pravdivostních hodnot
Ctrl + A	Vygeneruje záznam stavů na vstupech a výstupech do souboru CSV
Alt + F4	Ukončí aplikaci
F1	Zobrazí dialog <i>O aplikaci</i>
F3	Restartuje simulaci
F4	Přejde na další krok simulace
F5 – F10	Změna rychlosti animace (F5 odpovídá nejnižší rychlosti a F10 nejvyšší)
F11	Zastaví animaci

5.4 Panel vlastností

Jedná se o panel, který se objeví u pravého okraje hlavního okna po označení součástky v editoru. Umožňuje nastavit jednotlivé vlastnosti součástky, mezi které patří například poloha nebo pojmenování.



5.5 Pomocná okna

5.5.1 Tabulka pravdivostních hodnot

Tento dialog zobrazuje vygenerovanou tabulku pravdivostních hodnot. Je rozčleněn na dvě části, a to na nabídku a samotnou tabulku. **Nabídka** obsahuje tlačítko *Uložit* jež vygeneruje soubor CSV s obsahem tabulky a tlačítko *Zavřít* sloužící k ukončení tohoto dialogu. **Tabulka** pak slouží k vizualizaci pravdivostních hodnot. A obsahuje sloupec *Stav*, následovaný sloupci vstupních hodnot, oddělovačem a sloupci vstupních hodnot. Sloupce s hodnotami se jmenují podle Názvu nastaveného u příslušné součástky. Sloupec *Stav* obsahuje informaci o potenciální správnosti výsledku. Obsahuje jednu z následujících ikon:



Tato ikona značí, že řádek je z největší pravděpodobností odpovídá realitě.



Tato ikona naproti tomu značí, že některá z hodnot nebyla určena.

Možné ikony ve sloupci, který vyjadřuje stav jsou následující:



Na vstupu/výstupu se nachází úroveň LOW (logická 0).

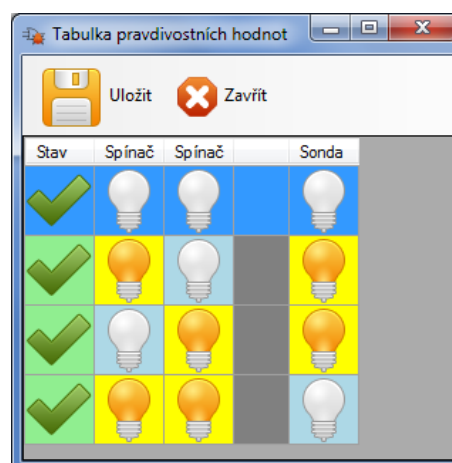


Na vstupu/výstupu se nachází úroveň HIGH (logická 1).



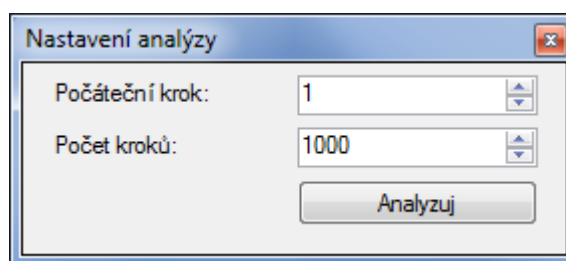
Hodnota výstupu nemohla být určena (výstup je nestabilní).

Aby bylo možné tabulku vygenerovat, musí obvod obsahovat pouze spínače (nikoli tlačítka) v maximálním počtu 16. Není doporučeno používat tuto funkci na složitější obvody, neboť by mohl být výstup, který se ustálí až po delším čase považován za nestabilní. Tato funkce je také relativně časově náročná a doba potřebná k výpočtu roste exponenciálně s počtem vstupů.



5.5.2 Nastavení analýzy obvodu

V tomto dialogu je možné nastavit počet kroků a počáteční krok analýzy. Po potvrzení těchto hodnot bude vygenerován soubor CSV se záznamem stavů na vstupech a výstupech (sondách). Hodnota *počáteční krok* umožňuje nastavit offset, od kterého se budou data ukládat. Hodnota *počet kroků* pak určuje kolik kroků má být uloženo. Maximální počet kroků je jeden milion, avšak pro zpracování tabulkovými kalkulátory se doporučuje držet pod hranicí 60 000.



5.5.3 O aplikaci

Tento dialog je čistě informativní. Obsahuje pouze verzi, copyright a informace o autorovi.

5.6 Hlavní pracovní plocha aplikace

Jedná se o nejpodstatnější část aplikace. Tato plocha se chová jako interaktivní editor schémat a zároveň zobrazuje i logické stavy na jednotlivých spojkách. S hlavní plochou je možné posouvat a také měnit její přiblížení.

5.6.1 Manipulace s objektem

Objekt je základní stavební prvek schématu. Za objekt jsou považovány součástky, zdroje signálu, sondy a také popisky. S objektem se dá libovolně posouvat po pracovní ploše aniž by došlo k přerušení spojků. Objekt lze vytvořit, duplikovat a odstranit. Také lze po jeho označení měnit jeho vlastnosti v *panelu vlastností*, který se objeví poblíž pravého okraje pracovního prostoru. Každý objekt má minimálně tři vlastnosti, a to horizontální souřadnici, vertikální souřadnici a pojmenování. Další vlastnosti jsou závislé na typu objektu a jsou zpravidla zmíněny v příloze B. Označený objekt lze rozpoznat podle zelených lomených čar v rozích.

5.6.2 Manipulace se spojem

Se spojem se nedá manipulovat přímo, ale pouze za využití pinů. Piny jsou vývody signálu ze součástek a zobrazují se jako černé horizontální čáry na okraji objektu zakončené kolečkem. Spoj je možné vytvořit postupným **kliknutím levým tlačítkem** na jeden vstupní (zpravidla nalevo) a jeden výstupní (zpravidla napravo) pin v libovolném pořadí. Nelze propojit dva vstupy nebo dva výstupy (tato spojení by neměla smysl, nebo by mohla být konfliktní). **Kliknutí pravým tlačítkem** na pin označí všechny spoje z něj vycházející. Takto označený spoj lze klávesou **Del** odstranit. Označení lze zrušit kliknutím do prázdného prostoru. Označený spoj se vykresluje zelenou barvou. Neoznačený spoj má barvu podle logické úrovně. Úroveň HIGH má oranžovou barvu, úroveň LOW barvu modrou. Fialová barva značí, že daný spoj právě vytváříte.

5.6.3 Soupis ovládání editoru spojů

Myš

Klik levým tlačítkem na objekt – označení objektu

Klik levým tlačítkem na pin – zahájení kreslení spoje

Klik levým tlačítkem na kolečko u spínače/tlačítka – přepnutí stavu spínače

Klik levým tlačítkem do volného prostoru – odznačení objektu, konec kreslení spoje

Klik pravým tlačítkem na objekt – označení objektu

Klik pravým tlačítkem na pin – označení spoje, který k pinu náleží

Klik pravým tlačítkem do volného prostoru – vyvolá nabídku tvorby objektu

Tah levým tlačítkem – přesunutí objektu pod kurzorem (je-li nějaký)

Tah pravým tlačítkem – posun pracovní plochy

Tah prostředním tlačítkem – přiblížení/oddálení pohledu

Točení kolečkem myši – přiblížení/oddálení pohledu

Dvojklik levým tlačítkem na objekt – vytvoří nový objekt stejného typu a posune jej

Dvojklik levým tlačítkem do volného prostoru – vyvolá nabídku tvorby objektu

Klávesnice

Escape během výpočtu – přeruší probíhající výpočet

Escape během kreslení spoje – přeruší kreslení spoje

Insert – vyvolá nabídku tvorby objektu

Delete – odstraní zvolený spoj či objekt

Šipky vlevo / vpravo – restart / další krok simulace

5.7 Módy simulace

Aplikace obsahuje dva módy simulace obvodu, takzvaný *Okamžitý mód* a *Mód simulace*. Aktuálně zvolený mód se zobrazuje v levé části stavového řádku hlavního okna.

5.7.1 Okamžitý mód

Tento mód slouží převážně k editaci schématu a vyznačuje se tím, že při změně stavu spínače se okamžitě aktualizují všechny logické stavy obvodu. Aktualizace probíhá maximálně do druhé úrovně rekurze. Poté je přerušena. Tento mód se vyvolá restartem simulace.

5.7.2 Mód simulace

V tomto módu je taktéž možno editovat zapojení. Aktualizace logických úrovní není okamžitá, ale kroková. Na další krok simulace je možné přejít některou z příslušných klávesových zkratek nebo kliknutím na odpovídající tlačítko z hlavní nabídky (*režim krokování*). V *režimu animace* dochází k periodickému krokování simulace. Animaci je taktéž možno aktivovat z hlavní nabídky nebo klávesovou zkratkou. Aktuální krok se taktéž objevuje nalevo ve stavovém řádku.

Simulace (krok 6) - režim krokování

Stavový řádek aplikace

6 Závěr

6.1 Souhrn

Hlavním záměrem práce bylo vytvořit jednoduchý simulátor logických obvodů. Tento záměr byl dle mého názoru ve stanovených mezích dodržen. Navíc došlo i k rozšíření o nastavbové logické obvody a imaginární prvky sloužící k usnadnění práce. Technické zpracování je odpovídající zadání a lze jej snadno rozšířit o další funkce a objekty.

6.2 Předpokládané využití

Návrh aplikace vycházel z předpokladu, že dojde k je využívání mezi studenty a učiteli nejen technických škol, ale i gymnázií nebo jiných středních škol. Celý projekt je koncipován pro jeho snadné začlenění do výuky, ale je možné jej využít i jako pomůcku při řešení úloh této problematiky mezi širokou veřejností se zájmem o dané téma. Práce v projektu je relativně rychlá a díky škálovatelnosti vektorové grafiky jí lze dobře využít na zobrazovačích s vysokým rozlišením či ve spojení s interaktivní tabulí nebo projektorem, což také napomáhá jednoduché integraci do výuky.

6.3 Budoucnost projektu

Budoucnost tohoto projektu závisí především na jeho uživateli, neboť jejich odezva a podněty budou hlavním faktorem ovlivňujícím směr vývoje. Doufám, že dojde k většímu rozšíření na školách a tím pádem i k získání lepší odezvy. Již v tento okamžik se nabízí několik možných úprav, a to především vylepšení manipulace se spoji a jejich vykreslování. Jako další z nich mohu zmínit například možnost tisku nebo podporu tvorby vlastních objektů bez zásahu do zdrojového kódu za pomoci skriptování.

Kontakt na mou osobu v případě odezvy je možné nalézt v dialogu *O aplikaci*. Také je možné využít webových stránek projektu na adrese <http://simlogic.cz.cc>.

7 Seznam literatury a internetových zdrojů

Na tomto místě uvádím seznam tištěných zdrojů, které jsem použil převážně jako zdroj informací o vstupech a výstupech jednotlivých obvodů.

Katalog společnosti GM Electronic: *Součástky pro elektroniku*, 2001

Negsseog S.: *IC-Vergleichs-Handbuch*, Verlag für technische Literatur Conrad, 1990

Dále uvádím použité internetové zdroje.

Microsoft Developer Network (MSDN): <<http://msdn.microsoft.com>>

CodeProject: <<http://www.codeproject.com>>

C# Corner: <<http://www.c-sharpcorner.com/>>

Wikipedia: <<http://en.wikipedia.org>>

Play-hookey: <<http://www.play-hookey.com/digital>>

Icojoy: <<http://www.icojoy.com/>>

8 Použitý software

Zde je uveden seznam použitého software. Veškerý software je obsažen na přiloženém CD.

Microsoft Visual C# 2008 Express editon

Knihovny .NET Framework 3.5 SP1

OpenOffice 3.0

Adobe Acrobat Reader 8

IcoFx

XnView

9 Přílohy

Příloha A Schéma hlavní nabídky

Soubor

Nový

Načíst

Uložit

Konec

Simulace

Restartovat Simulaci

Další krok

Animace

Nastavení rychlosti animace

Zastavit

Analýza

Generovat pravdivostní tabulku

Generovat tabulku průběhu

Konec

O aplikaci

Příloha B Popis činnosti všech implementovaných objektů

Všechny níže zmíněné obvody využívající hodinový signál reagují na sestupné hrany!

Hradla

Invertor má na výstupu opačnou hodnotu než na vstupu.

Buffer má stejné hodnoty na vstupu i výstupu. Slouží jako zpožďovací člen.

AND má na výstupu logickou jedničku, jsou-li jedničky na obou vstupech.

NAND má na výstupu přesně opačnou hodnotu jako AND.

OR má na výstupu logickou jedničku, je-li jednička alespoň na jednom vstupu.

NOR má na výstupu přesně opačnou hodnotu jako OR.

XOR má na výstupu logickou jedničku, je-li jednička právě na jednom vstupu.

XNOR má na výstupu přesně opačnou hodnotu jako XOR.

Vstupy

Spínač umožňuje uživateli měnit trvale stav na svém výstupu.

Jediný povolený vstup při generování pravdivostních tabulek.

Stav lze změnit kliknutím na kolečko uvnitř spínače.

Tlačítko umožňuje vygenerovat jednotkový puls na svém výstupu.

Stav lze změnit kliknutím na kolečko uvnitř tlačítka.

Generátor umožňuje generovat periodicky se opakující stavy na svém výstupu.

Perioda se dá nastavit ve vlastnostech objektu (*Doba úrovně HIGH/LOW*).

Trvale HIGH (1) je zdroj konstantního signálu úrovně jedna.

Trvale LOW (0) je zdroj konstantního signálu úrovně nula.

Čísla

Tyto obvody slouží jen k usnadnění simulace. Ve skutečnosti neexistují!

Jsou dostupné varianty lišící se datovým typem hodnoty.

Zdroje generují na svých výstupech binární reprezentaci daného čísla.

Číslo lze nastavit ve vlastnostech objektu (*Hodnota*).

Výpis převádí binární reprezentaci daného čísla ze vstupů a zobrazuje jej.

Čítače

Čítače slouží jako počítadla impulsů na vstupu.

Jsou dostupné i reverzní varianty, které umí počítat jen pozpátku.

Pro usnadnění práce je přítomen i uživatelský čítač, u kterého lze nastavit horní mez.

Čítače lze řetězit použitím pinu **CARRY** popřípadě spojením pinu **CLK** s horním bitem.

U čítačů se vyskytují následující piny:

CLK je vstup impulzů, které se mají počítat.

RST/RO vynuluje počítadlo, je-li na něm úroveň jedna.

UD způsobí zpětné počítání, je-li na něm úroveň jedna.

CARRY je výstupní pin, který generuje náběžnou hranu při přetečení čítače.

Sx jsou výstupní piny lišící se typem čítače.

V praxi existuje několik variant čítačů:

Binární (v programu označeny jako 4/8/16 bit)

Počítají ve dvojkové soustavě do čísla určeného počtem bitů.

Decimální (také BCD)

Počítají ve dvojkové soustavě do desítky a poté pokračují od začátku.

Johnsonův

Počítá do desíti, tak že je jednička vždy na jednom z jeho deseti výstupů.

Johnsonův binární

Počítá do desíti, ale hodnoty výstupů odpovídají následující tabulce:

Číslo	S1	S2	S3	S4	S5
0	0	0	0	0	0
1	1	0	0	0	0
2	1	1	0	0	0
3	1	1	1	0	0
4	1	1	1	1	0
5	1	1	1	1	1
6	0	1	1	1	1
7	0	0	1	1	1
8	0	0	0	1	1
9	0	0	0	0	1

Sčítačky (též Aritmetické jednotky)

Sčítají čísla ve dvojkové soustavě, která se nacházejí na pinech **A** a **B**.

Piny **Cin/Cout/CARRY** složí k přenosu informace o přetečení rozsahu.

Jednoduchá sčítačka nemá na rozdíl od plné vstupní pin **Cin**.

Pin **Cin** lze využít také pro odečtení dvou čísel.

Multiplexery (MUX)

Multiplexer slouží jako přepínač signálů.

Na výstup **X** posílá data přicházející ze vstupu **I_a**, kde **a** je adresa na pinech **A1 – An**.

Počet datových i adresních vstupů se liší podle rozsahu multiplexeru.

V aplikace je přítomen v 16/8/4/2 bitových variantách (počet bitů adresy).

Demultiplexery (DEMUX)

Demultiplexer slouží jako rozdělovač signálů.

Na výstup **O_a** posílá data přicházející ze vstupu **X**, kde **a** je adresa na pinech **A1 – An**.

Počet adresních vstupů a datových výstupů se liší podle rozsahu demultiplexeru.

V aplikace je přítomen v 16/8/4/2 bitových variantách (počet bitů adresy).

Klopné obvody

Klopné obvody jsou paměťové členy, lišící se vstupy a chováním.

Zpravidla mají výstup **Q** (normální) a **Q** (negovaný, inverzní).

Lze je rozdělit podle reakcí na vstupy následovně:

Latch (stav na vstupech je okamžitě reflektován na výstupech).

Clocked Latch (k reflexi je potřeba hodinový signál odpovídající logické jedna).

Flip-Flop (k reflexi je potřeba hodinový pulz se sestupnou hranou).

Existuje několik typů:

RS (Reset/Set)

Je-li na vstupu **R** hodnota jedna, na výstupu je hodnota nula.

Je-li na vstupu **S** hodnota jedna, na výstupu je hodnota jedna.

Je-li na obou vstupech nula, na výstupu se hodnota nemění.

Je-li na obou vstupech jedna, nelze stav výstupů určit (zakázaný stav).

JK (jméno nenese žádný význam, mnemotechnická pomůcka JUMP - KILL)

Ekvivalent obvodu **RS** (**J** odpovídá **S**, **K** odpovídá **R**)

Je-li na obou vstupech jedna, výstupy se negují (chybí zakázaný stav).

Pojmenován po vědci Jacku Killbym, který jej vyvinul.

JK/SC (Jump-Kill/Set-Clear)

Ekvivalent obvodu **JK**.

Lze vnutit stav výstupu pomocí nuly na vstupech **S** a **C** (Set/Clear).

Je-li nula na obou vstupech **S** i **C**, pak na obou výstupech bude jedna.

Je-li jedna na obou vstupech **S** i **C**, pak chování odpovídá obvodu **JK**.

D (Data)

Jedná se o paměťový člen.

Na výstup запиše hodnotu vstupu **D**, právě když přijde hodinový pulz.

Vstupy **R** a **S** (Reset a Set) odpovídají vstupům **S** a **C** u obvodu **JK**.

T (Toggle)

Při příchodu hodinového pulzu neguje své výstupy.

Používá se nejčastěji jako frekvenční dělič.

Posuvné registry**Serial -> Parallel (SP SHIFT)**

Paměťový člen sloužící k převodu sériového signálu na paralelní.

Při příchodu hodinového pulzu dojde k uložení dalšího bitu dat.

Při ukládání se data posunou o bit a na pin **S1** se přiřadí stav vstupu **DATA**

Parallel -> Serial (PS SHIFT)

Paměťový člen sloužící k převodu paralelního signálu na sériový.

Pro vykonání jakékoliv operace je potřebný hodinový pulz.

Je-li **MODE** ve stavu jedna, dojde k uložení dat ze vstupů **D1** až **D8**.

V opačném případě se chová stejně jako SP SHIFT.

Na výstupním pinu **DATA** se vždy nachází aktuální bit sériových dat.

Piny **S1** až **S8** obsahují aktuální data z paralelního vstupu.

Během hodinového pulzu dochází i k posunu dat na pinech **S1** až **S8**.

BCD Dekodéry

Vycházejí z vyjádření desítkového čísla ve dvojkové soustavě.

BCD definuje stavy od 0000 do 1001 (binárně).

BCD na 1 z 10

Nastaví výstup určený BCD adresou na vstupech do stavu jedna.

1 z 10 na BCD

Na výstupech se nachází binární adresa vstupu s hodnotou jedna.

Na vstupech musí být vždy právě jedna jednička, jinak je generována chyba.

Chyba se projevuje logickou jedničkou na výstupu **ERR**.

BCD na 7seg Displej

Obvod slouží k převodu čtyřbitového čísla na vstupy displeje.

Jsou definována i čísla 10 až 15 (zobrazí se jako A až F).

Je-li jednička na vstupu **ERR**, bude na displeji zobrazena pomlčka.

Sonda

Slouží k monitorování logického stavu na spoji.

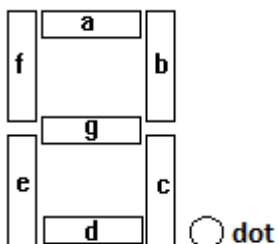
Jediný druh výstupu při generování pravdivostních tabulek či analýze.

Displej

Reprezentuje sedmi segmentový LED displej.

Hodnoty na vstupech určují zda daný segment svítí.

Segmenty (vstupy) odpovídají následujícímu obrázku:

**Popisek**

Slouží k zobrazení textu ve schématu a nijak neovlivňuje výsledný obvod.

Zobrazený text odpovídá vlastnosti *Jméno*.

Příloha C Ukázková zapojení

K práci jsou přiloženy mimo jiné i tyto tři ukázkové projekty:

Ukázka multiplexu

Tento projekt slouží jako ukázka pro generování analytického CSV souboru.

Ukázka XOR

Tento projekt slouží jako ukázka generování pravdivostní tabulky XOR hradla.

Ukázka Simulace

Jedná se o ukázku čítače s displejem. Slouží k prezentaci animace a krokování.

Příloha D Soupis obsahu přiloženého CD

Disk je logicky členěn do složek podle významu obsahu.

Složka *Binaries* obsahuje spustitelné soubory aplikace SimLogic.

Složka *Documents* obsahuje tuto práci v elektronické podobě v několika formátech.

Složka *Samples* obsahuje ukázkové projekty (viz příloha C).

Složka *Software* obsahuje programy použité při tvorbě práce či potřebné k jejímu spuštění.

Složka *Sources* obsahuje zdrojové kódy aplikace SimLogic v jazyce C#.