



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Katedra elektrických pohonů a trakce

Možnosti využití FPGA pro řízení pohonů

Usage of FPGA for controlling electric drives

Diplomová práce

Studijní program: Elektrotechnika, Energetika a Management

Studijní obor: Elektrické pohony

Vedoucí práce: Ing. Jan Bauer, Ph.D.

Petr Zakopal
Praha 2023

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Zakopal** Jméno: **Petr** Osobní číslo: **483802**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra elektrických pohonů a trakce**
Studijní program: **Elektrotechnika, energetika a management**
Specializace: **Aplikovaná elektrotechnika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Oživení pracoviště s měničem DCM a PLC SIMATIC

Název bakalářské práce anglicky:

Workpalce with Rectifier DCM and PLC SIMATIC

Pokyny pro vypracování:

- 1) Seznamte se s měničem řady DCM firmy SIEMENS
- 2) Oživte základní regulační smyčky měniče (otáčkovou, proudovou)
- 3) Prostudujte možnosti záznamu průběhů z měniče pomocí PLC nebo dotykového panelu
- 4) Pomocí PLC SIMATIC S1200 a dotykového panelu realizujte vzdálené ovládání a monitoring měniče
- 5) Na dotykovém panelu vytvořte obrazovku pro nastavování otáček nebo momentu motoru napájeného měničem

Seznam doporučené literatury:

- [1] Weidauer J., Messer R. Electrical Drives, Publics Erlangen, 2014
- [2] SCE Training Curriculum. Siemens AG, 2016
- [3] Durry B. The Control Techniques Drives and Controls Handbook 2nd ed., IeT, 2009
- [4] Pavelka J., Koblir P. Elektrické pohony a jejich řízení. 3. přepracované vydání. Praha: České vysoké učení technické v Praze, 2016. ISBN 978-80-01-06007-0.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Jan Bauer, Ph.D., katedra elektrických pohonů a trakce FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **24.01.2021**

Termín odevzdání bakalářské práce: **21.05.2021**

Platnost zadání bakalářské práce: **30.09.2022**

Ing. Jan Bauer, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

PROHLÁŠENÍ

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne _____

Petr Zakopal

PODĚKOVÁNÍ

Nam quis commodo justo. Mauris diam metus, mattis sed rutrum in, volutpat sit amet sem. Nam bibendum commodo porttitor. Quisque eget lectus rutrum, molestie tortor id, iaculis nunc. Sed et maximus ipsum. Vivamus vel facilisis nisl. Curabitur eu nibh nec erat mollis finibus at in sapien. Mauris viverra sapien neque, nec lacinia odio laoreet eu. Quisque consectetur eros ac orci interdum scelerisque.

ABSTRAKT

Nam quis commodo justo. Mauris diam metus, mattis sed rutrum in, volutpat sit amet sem. Nam bibendum commodo porttitor. Quisque eget lectus rutrum, molestie tortor id, iaculis nunc. Sed et maximus ipsum. Vivamus vel facilisis nisl. Curabitur eu nibh nec erat mollis finibus at in sapien. Mauris viverra sapien neque, nec lacinia odio laoreet eu. Quisque consectetur eros ac orci interdum scelerisque.

Klíčová slova: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis aliquam finibus sagittis. Nunc venenatis, augue quis luctus dictum, elit justo pharetra leo, nec viverra purus dui at quam.

ABSTRACT

Nam quis commodo justo. Mauris diam metus, mattis sed rutrum in, volutpat sit amet sem. Nam bibendum commodo porttitor. Quisque eget lectus rutrum, molestie tortor id, iaculis nunc. Sed et maximus ipsum. Vivamus vel facilisis nisl. Curabitur eu nibh nec erat mollis finibus at in sapien. Mauris viverra sapien neque, nec lacinia odio laoreet eu. Quisque consectetur eros ac orci interdum scelerisque.

Keywords: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis aliquam finibus sagittis. Nunc venenatis, augue quis luctus dictum, elit justo pharetra leo, nec viverra purus dui at quam.

OBSAH

1	Úvod.....	1
2	Embedded Systems	2
2.1	Application Specific Integrated Circuit	3
2.2	Hardware Accelerated Applications	3
3	Programovatelné hradlové pole – FPGA	5
3.1	Vývoj FPGA z PLD	5
3.2	Aktuální složení FPGA	5
3.2.1	Generátory funkcí	6
3.2.2	Paměťové elementy	7
3.3	Popis funkce	7
3.4	Programování	7
3.5	Využití	7
3.6	Spotřeba	7
3.7	Výpočetní výkon a propustnost	7
3.7.1	Aplikace v elektrických pohonech	7
3.8	Vývojová deska Digilent Zybo	7
3.8.1	Parametry a složení	7
3.8.2	Možné alternativy	7
4	Matematický model stroje	7
4.1	Představení stroje	7
4.2	Odvození modelu	7
4.3	Optimalizace modelu	7
5	Program pro FPGA a CPU	7
5.1	Použité nástroje	7
5.1.1	Xilinx Vivado	7
5.1.2	Xilinx Vitis	7
5.1.3	Petalinux	8
5.1.4	Programovací prostředí – operační systém Linux	8
5.2	Tvorba HW architektury Xilinx Vivado	8
5.3	Tvorba Petalinux	8
5.4	Tvorba SW pro CPU a FPGA	8
6	Představení pracoviště	8
7	Dosažené výsledky	8
	Závěr	9
	Literatura	10

Příloha A	Seznam symbolů a zkratk	11
A.1	Seznam symbolů	11
A.2	Seznam zkratk	11

SEZNAM OBRÁZKŮ

2 - 1	Blokové schéma Embedded systému a řízeného fyzikálního systému. (převzato a upraveno z [2])	3
3 - 1	Blokové schéma složení moderních FPGA.....	5
3 - 2	Základní koncept uspořádání FPGA.	6
3 - 3	Ukázka, jakým způsobem realizuje funkční generátor požadovanou funkci pomocí SRAM a MUX.	7

SEZNAM TABULEK

3 - 1	Pravdivostní tabulka ukázkové funkce, realizované v generátoru funkcí, umístěném v logickém bloku FPGA.....	7
-------	---	---

1 Úvod

V době, kdy byla od elektrických pohonů požadována spolehlivost, vysoká účinnost a nenáročný ovšem kvalitní řízení, byly k řízení využívány samotné digitální signálové procesory. Postupem času dochází ke zjištění, že výkon DSP není dostatečný a na některé aplikace, kde je vyžadováno provedení značné množství náročných výpočtů za co nejkratší čas, nejsou vhodné. Proto nastupuje éra logických programovatelných polí (FPGA), které jsou schopny tyto výpočty provést s velmi nízkými nároky na energii a za velmi krátký čas.

V mnoha odvětvích se již začíná využívat embedded systém s Application Specified Hardware, který je určen pouze na využití v předem dané aplikaci. Tento hardware slouží v dané aplikaci k jedinému účelu, který vykonává a na který je optimalizován. Tím se liší od procesoru, který vykonává mnoho instrukcí a využít ho pouze jako samostatnou výpočetní jednotku je z hlediska energetické i finanční náročnosti nevýhodné. Implementace hradlových polí přináší nejen v řízení elektrických pohonů zvýšení výpočetního výkonu, ale také snižování energetické náročnosti řízení.

Perspektiva logických programovatelných polí a hardwareově urychlovaných aplikací je podpořena jejich využíváním i mimo obor elektrických pohonů a trakce. Z důvodu jejich veliké propustnosti, vysokých výpočetních výkonů a nízké energetické náročnosti jsou využívány v AI, machine learningu, zpracování obrazu, těžení kryptoměn a jiných nepohonářských aplikacích.

Nevýhodou problematiky FPGA je jejich složitější programovatelnost z hlediska tvoření aplikace. Aplikace je tvořena určitým postupem (workflow), který kladne vysoké nároky na vzdělání a zkušenosti vývojářů. Většina FPGA je programována pomocí jazyků Verilog či VHDL, které mohou pro softwarově orientované programátory představovat značnou překážku. Proto bylo vyvinuto tvoření aplikací pomocí vyšší úrovně syntézy (HLS), kdy je možné tvořit programy ve vyšších programovacích jazycích jako je například C, C++ či Python. HLS umožnilo rapidní rozšíření a využití Embedded FPGA Accelerated Applications v mnoha aplikacích a značně vylepšilo vývojářský požitek (developer experience, DX) při tvorbě aplikací.

Protože může být náročné vytvořit vlastní architekturu, složenou z CPU a spolupracujícího FPGA, je vhodné při prvotním vývoji aplikace využít dostupné vývojové desky obsahující již předpřipravené propojení jednotlivých komponent. Součástí těchto vývojových desek bývá také mnoho vstupů a výstupů (I/O) pro snadnější využití při lazení a tvoření aplikace. V této práci je využívána vývojová deska Zybo od firmy Digilent. Ovšem autor v textu představuje další možnosti, které mohou být pro konkrétní aplikace a využití vhodnější.

Tato práce se zajímá o aplikace a možné využití FPGA při řízení elektrických pohonů. Autor v ní představuje základní principy Hardware Accelerated Applications, z jakého důvodu je tento přístup perspektivní a proč je vhodné se orientovat tímto směrem.

2 Embedded Systems

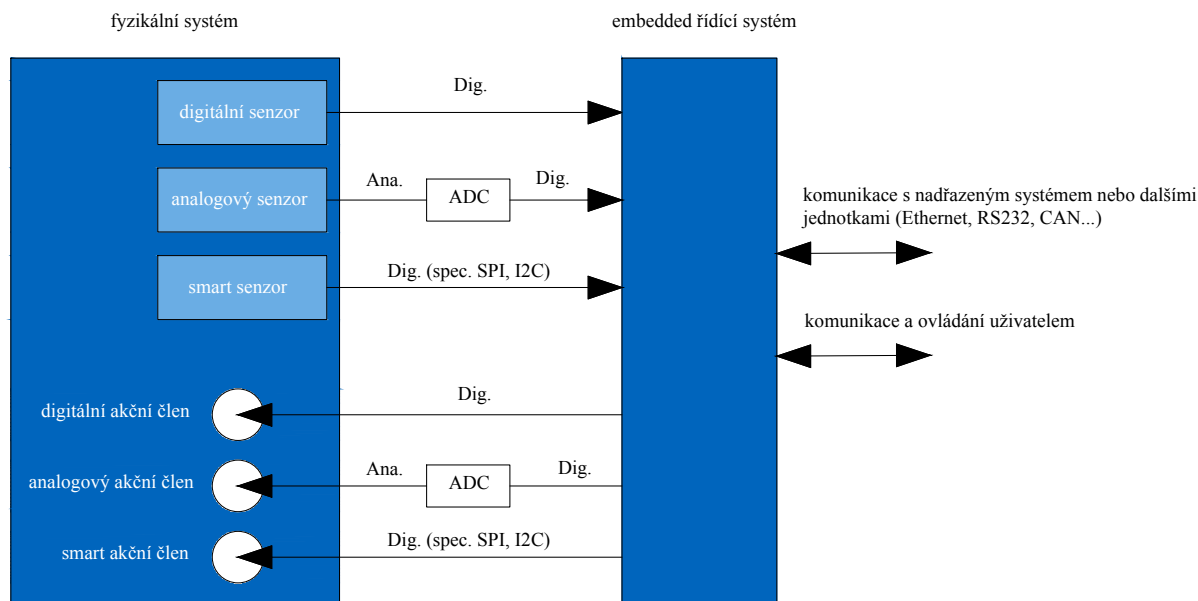
Embedded systémy je název pro skupinu zařízení, obecně systémů, které je možné charakterizovat jako specifické výpočetní zařízení, resp. počítače, které jsou určeny pro podporu funkce nebo řízení nějakého většího celku, produktu nebo fyzikálního systému. Oproti tomu osobní počítač je sice výpočetní zařízení, ale nelze mluvit o embedded systému, protože je určen pro mnoho univerzálních aplikací. [1]

Dalším důležitým rozdílem mezi *Embedded System* a obecným výpočetním zařízením je ten, že v případě embedded systému je interakce mezi systémem a uživatelem uměle omezena na základní ovládání či kontrolu funkce. Není předpokládáno, že by uživatel, jež aplikaci embedded systému využívá, výrazným způsobem zasahoval do jeho funkce. Naopak obecný výpočetní systém je uzpůsoben na podstatné zásahy uživatele. [1] [2]

Do embedded systému obecně vstupují vstupní signály, které jsou následně zpracovány a poté vybrané výsledky výpočtů jsou v podobě výstupní signálů výstupním produktem systému. Tyto produkty mohou pomocí akčních členů zasahovat do řízeného systému či aplikace. Vstupní signály většinou přicházejí ze speciálních snímačů, kompatibilních s embedded systémem (senzor teploty, senzor tlaku, senzor zrychlení, gyroskop apod.). Naopak jeho výstupem signály, vedoucí na konektory, na kterých se dle požadavků objevuje např. specifická hodnota napětí. Nebo mohou na výstupních pinech být připojené LED signalizace, komunikační sběrnice některých komunikačních systémů nebo výstupní LDC displaye. Způsob, kterým jsou kódovány vstupní a výstupní signály je většinou specificky určený aplikaci daného systému. [1]

K obecnému výpočetnímu systému je možné připojit vstupní periferie klasických osobních počítačů – myš, klávesnice, mikrofon. Jako hlavní výstupní periferie obecného systému je monitor. Jeho komunikace s periferiemi je většinou standardizována tak, aby bylo možné periferie libovolně zaměňovat bez změny funkčnosti. [1].

Na obrázku 2 - 1 je zobrazeno názorné blokové schéma řízení fyzikálního systému pomocí embedded systému. Tyto bloky mezi sebou komunikují pomocí digitálních signálů. Pokud tyto signály nejsou digitální, musí se před zpracováním v embedded systému zdiskretizovat.



Obr. 2 - 1 Blokové schéma Embedded systému a řízeného fyzikálního systému. (převzato a upraveno z [2])

2.1 Application Specific Integrated Circuit

S tématem embedded systémů se pojí pojem hardware, který je určen pro jedinou aplikaci. Tato skupina zařízení se nazývá *Application Specific Integrated Circuit*, popř. *Hardware* (ASIC, ASHW). V této oblasti je opět využíváno přesvědčení, že pokud je architektura HW přímo specializovaná na jednu aplikaci, je vysoká pravděpodobnost, že ji bude vykonávat bezchybně, kvalitně a rychle.

Tyto aplikace jsou využívány v širokém spektru oborů jako je např. zpracování zvuku, videa, výpočtů apod. Tyto ASIC mohou také vykonávat potřebné rychlé výpočty pro matematické modely elektrických strojů, které jsou využívány např. pro HIL.

Než je tento specifický obvod vytvořen, je nutné jej navrhnut, vyzkoušet a odladit. K tomu slouží logická programovatelná pole, ve kterých je možné požadovaný HW navrhnut a odladit před velkou produkcí ASIC. Pokud velká produkce není z ekonomických důvodů možná, jsou FPGA využívány i v produkční oblasti, kde je HW struktura, která by byla přítomna na ASIC, vytvořena přímo na FPGA.

2.2 Hardware Accelerated Applications

V mnoha aplikacích, nejen při řízení elektrických pohonů, je vyžadováno, aby výpočty nebo zpracování dat probíhalo vysokou rychlostí. Tento problém nemůže být většinou vyřešen použitím běžného procesoru (CPU), který je optimalizován na provádění obecných komplexních funkcí řízení běhu programu, komunikace či přesunu dat. V moderním světě dochází k exponenciálnímu nárůstu množství dat, které je potřeba zpracovat. Aby tyto data bylo možné v požadovaném čase zpracovat, je třeba využít specifický HW, který bude schopen požadavky rychlosti a výkonu uspokojit. Tento proces se nazývá *Hardware Acceleration*. [3]

Princip hardwaerové akcelerace spočívá v přesunu výpočetně náročných aktivit na zvláštní oddělený hardware. Celkové řízení běhu aplikace a komunikace je ovšem stále přítomno na řídicím CPU. Oddělený hardware, na kterém dochází k akceleraci výpočtů je optimalizován na vykonávanou úlohu a jeho využití přináší zefektivnění běhu celkové aplikace. [3]

Struktura, ve které je využíváno více oddělených hardwarových procesorových a akceleračních jednotek, se často nazývá heterogenní. [3]

Hardwaerová akcelerace poskytuje rychlejší výpočty než CPU, protože využívá značné úrovně paralelismu výpočtů. Oproti tomu klasické CPU vykonává jednotlivé instrukce sériově. I v případě, že CPU má více jader a využívá více vláken, nemůže se úrovní paralelismu při dané energetické náročnosti HW vyrovnat.

Pro HW akceleraci je v mnoha oblastech využíváno několik druhů jednotek, které jsou optimální pro dané aplikace.

Graphics Processing Units (GPUs) jsou jednotky, které převážně slouží k akceleraci zpracování a renderování grafických úloh. V době rapidního rozvoje elektroniky a SW je možné využití GPUs v mnoha odvětvích umělé inteligence (AI) či kreativních odvětvích. GPUs jsou využívány v aplikacích, kde není kladen veliký důraz na nízkou odezvu (latenci). [3]

Tensor Processing Units (TPUs) jsou jednotky, které slouží k provádění algoritmů strojového učení (machine-learning, ML). Jejich přímé datové propojení umožňuje velmi rychlý a přímý přenos dat. Díky přímému připojení nevyžadují využití pamětí, které by přenos dat zpomalovali. [3]

Field Programmable Gate Arrays (FPGAs) jsou jednotky, ve kterých není při výrobě pevně daná HW struktura. To umožňuje vytvoření, resp. naprogramování HW dle požadavků akcelerované aplikace. FPGAs jsou využívány i v on-line výpočtech matematických modelů elektrických strojů. Při realizaci této práce je pro akceleraci využíváno právě těchto programovatelných polí.

3 Programovatelné hradlové pole – FPGA

3.1 Vývoj FPGA z PLD

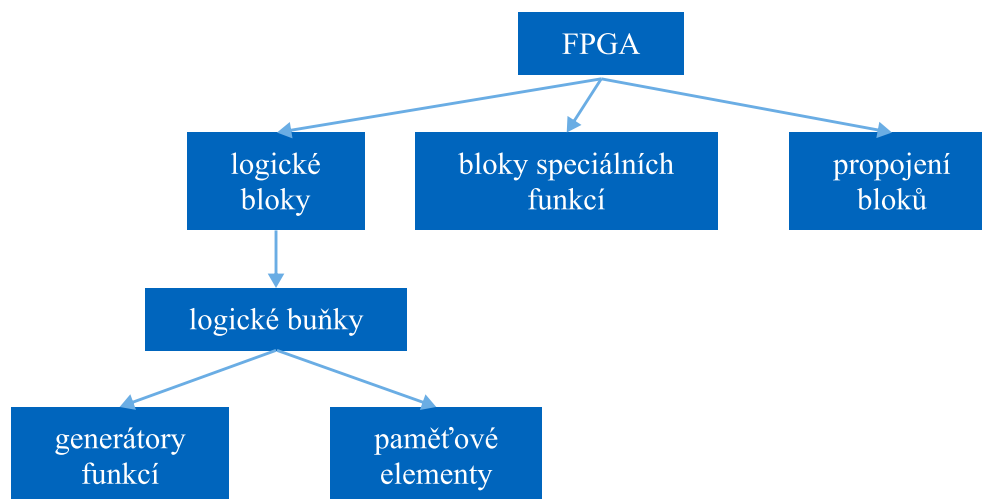
Programovatelné hradlové pole jsou zařízení, jejichž historický vývoj stojí na programovatelných logických zařízeních (programmable logic devices, PLD). První PLD fungovala na principu Booleových funkcí součtu násobení (sum of products). Tyto zařízení obsahovala matici více vstupových bloků AND a OR. Programování požadované probíhalo pomocí přerušování vstupů do těchto logických bloků. Později byly do PLA přidány D klopné obvody s multiplexory. Díky těmto součástím bylo možné vytvářet logické kombinační a sekvenční obvody, resp. automaty. Posledním vylepšením PLA, které stálo před zrodem FPGA, spočívalo v umístění více PLA bloků (skládajících se z AND, OR, multiplexeru a D klopného obvodu) na jeden integrovaný čip. Programovatelné spojení různých PLA bloků a výstupů umožnilo vytvořit požadovanou funkci. [1]

3.2 Aktuální složení FPGA

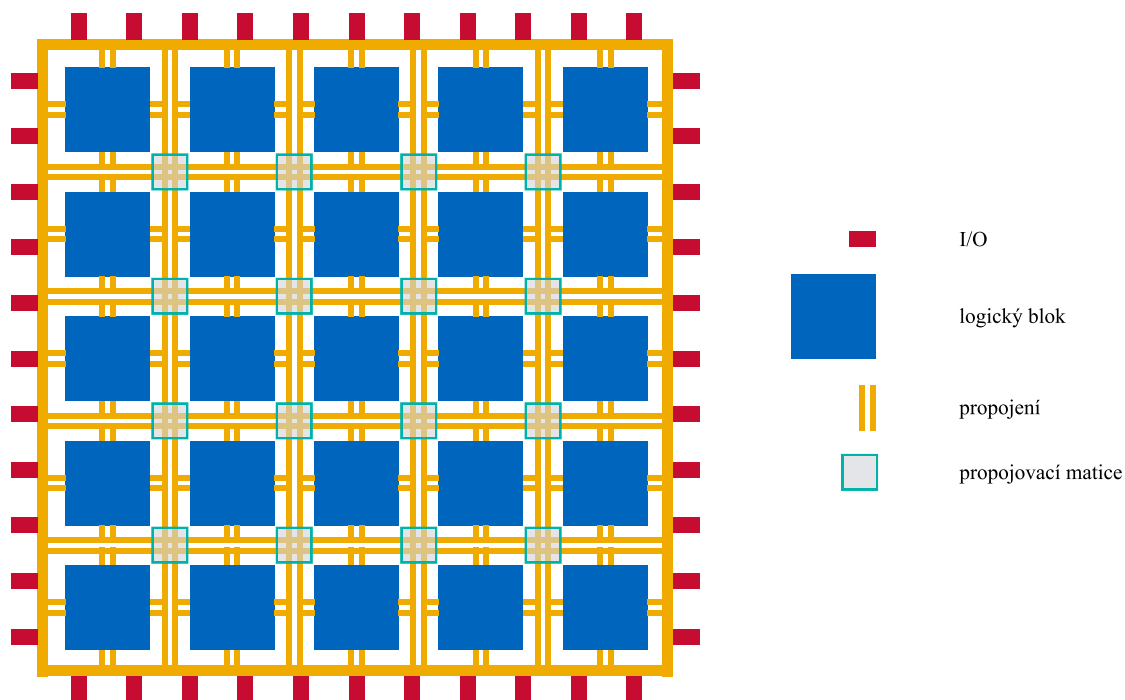
Moderní FPGA se skládají z 2D matice propojených programovatelných logických bloků, bloků speciálních funkcí a propojů vytvořených pomocí CMOS technologie. Po obvodě FPGA jsou rozmístěny vstupní a výstupní piny (I/O), připojené na zvláštní logické bloky. Použité logické bloky se skládají z mnoha logických buněk, které se skládají z generátorů funkcí a paměťových elementů. [1]

Na obr. 3 - 2 je možné pozorovat názorné schéma základního konceptu uspořádání FPGA. Na schématu jsou vyznačeny logické bloky, jejich propojení, propojovací matice pro aktivování jednotlivých propojů a vstupů a výstupů (I/O) FPGA.

I přesto, že se tato práce věnuje využití FPGA pro řízení elektrických pohonů je vhodné představit základní části FPGA a nastínit jejich funkci.



Obr. 3 - 1 Blokové schéma složení moderních FPGA.



Obr. 3 - 2 Základní koncept uspořádání FPGA.

3.2.1 Generátory funkcí

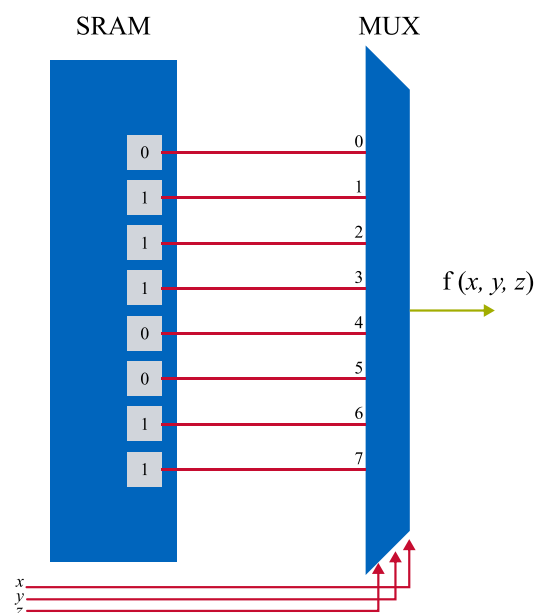
Oproti předchůdcům PLD, které pro generování funkcí používaly logická hradla tvořená CMOS tranzistory, využívají FPGA generátory funkcí.

Logickou funkci je možné popsat pravdivostní tabulkou, která má určitý počet vstupů a odpovídající počet výstupů. Dle [1] je možné si představit, že se generátor dané funkce skládá ze samostatné statické paměti (SRAM), jejíž výstupy jsou přímo přivedeny na vstup multiplexeru (MUX). Signály výběru výstupů by odpovídaly vstupním proměnným a jednotlivé vstupy do MUX výstupům funkce.

Pro bližší pochopení funkce generátoru funkcí z předchozího odstavce je možné představit realizaci smyšlené logické funkce $f(x, y, z) = \bar{x}z + y$. Pravdivostní tabulka této smyšlené logické funkce je zobrazena v tab. 3 - 1. Odpovídající realizace pomocí MUX a SRAM je zobrazena na obr. 3 - 3. Tato reprezentace se nazývá look-up table (LUT). Grafické znázornění inspirováno [1].

Tab. 3 - 1 Pravdivostní tabulka ukázkové funkce, realizované v generátoru funkcí, umístěném v logickém bloku FPGA.

i	x	y	z	$f(x, y, z)$
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1



Obr. 3 - 3 Ukázka, jakým způsobem realizuje funkční generátor požadovanou funkci pomocí SRAM a MUX.

Výhoda této reprezentace funkcí oproti logickým hradlům je, že doba zpoždění signálu (propagation delay) pro funkci je konstantní. Respektivě je konstantní, pokud funkci je možné realizovat jednou LUT. Pro realizaci obecné funkce je zapotřebí multiplexer $2^n \rightarrow 1$ a SRAM s počtem buněk 2^n , kde n je počet vstupních proměnných dané funkce. [1]

3.2.2 Paměťové elementy

3.3 Popis funkce

3.4 Programování

3.5 Využití

3.6 Spotřeba

3.7 Výpočetní výkon a propustnost

3.7.1 Aplikace v elektrických pohonech

3.8 Vývojová deska Digilent Zybo

3.8.1 Parametry a složení

3.8.2 Možné alternativy

4 Matematický model stroje

4.1 Představení stroje

4.2 Odvození modelu

4.3 Optimalizace modelu

5 Program pro FPGA a CPU

5.1 Použité nástroje

5.1.1 Xilinx Vivado

5.1.2 Xilinx Vitis

- 5.1.3 Petalinux**
- 5.1.4 Programovací prostředí – operační systém Linux**
- 5.2 Tvorba HW architektury Xilinx Vivado**
- 5.3 Tvorba Petalinux**
- 5.4 Tvorba SW pro CPU a FPGA**
- 6 Představení pracoviště**
- 7 Dosažené výsledky**

Závěr

Aliquam dapibus leo velit, ultrices eleifend mi feugiat eget. Aliquam euismod facilisis turpis, nec lobortis libero aliquet sit amet. Aenean suscipit ante eget ipsum viverra hendrerit. Ut sed massa sed nisi tempus dapibus in eu enim. Nullam vitae odio laoreet, malesuada purus non, faucibus orci. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam eget odio quis enim laoreet imperdiet nec eu nunc. Maecenas ut consequat purus. Duis faucibus risus nec metus cursus placerat. Phasellus sapien justo, laoreet in pulvinar ut, maximus nec velit.

Literatura

- [1] SASS, Ronald; SCHMIDT, Andrew G. *Embedded systems design with platform FPGAs: principles and practices*. Boston: Morgan Kaufmann, 2010. ISBN 0123743338.
- [2] ANDINA, Juan J. R.; TORRE ARNANZ, Eduardo de la; VALDÉS PEÑA, María D. *FPGAs: Fundamentals, Advanced Features, and Applications in Industrial Electronics*. 1. vyd. Bosa Roca: CRC Press, 2017;2015; ISBN 9781439896990;1439896992;
- [3] What Is Accelerated Computing, and Why Is It Important? In: *Xilinx* [online]. [B.r.] [cit. 2022-11-03]. Dostupné z: <https://www.xilinx.com/applications/adaptive-computing/what-is-accelerated-computing-and-why-is-it-important.html>.

Příloha A: Seznam symbolů a zkratk

A.1 Seznam symbolů

\vec{F} (N) vektor síly

A.2 Seznam zkratk

DCM DC Master