

Jiří Král

ŘEŠENÉ PŘÍKLADY VE VHDL

HRADLOVÁ POLE FPGA PRO ZAČÁTEČNÍKY

Praha 2010



Upozornění!

Vážení čtenáři,

na naší webové adrese: **<http://shop.ben.cz/cz/121309>** v sekci download, se nachází soubory ke stažení:



ke stažení vývojové prostředí ISE WebPack 10.1
(pozor soubor **WebPack.zip** 14 MB)



ke stažení soubory (.brd, .sch) k Příloze 2 (**Priloha2.zip**)



barevné obrázky z Přílohy 1 (**121309obr.zip**)

Ing. Jiří Král

Řešené příklady ve VHDL – Hradlová pole FPGA pro začátečníky

Bez předchozího písemného svolení nakladatelství nesmí být kterákoli část kopírována nebo rozmnožována jakoukoli formou (tisk, fotokopie, mikrofilm nebo jiný postup), zadána do informačního systému nebo přenášena v jiné formě či jinými prostředky.

Autor a nakladatelství nepřijímají záruku za správnost tištěných materiálů. Předkládané informace jsou zveřejněny bez ohledu na případné patenty třetích osob. Nároky na odškodnění na základě změn, chyb nebo vynechání jsou zásadně vyloučeny.

Všechny registrované nebo jiné obchodní známky použité v této knize jsou majetkem jejich vlastníků. Uvedením nejsou zpochybněna z toho vyplývající vlastnická práva.

Veškerá práva vyhrazena

© Ing. Jiří Král, Praha 2010

© Nakladatelství BEN – technická literatura, Věšínova 5, Praha 10

Jiří Král: Řešené příklady ve VHDL – Hradlová pole FPGA pro začátečníky

BEN – technická literatura, Praha 2010

1. vydání

ISBN 978-80-7300-257-2

Obsah

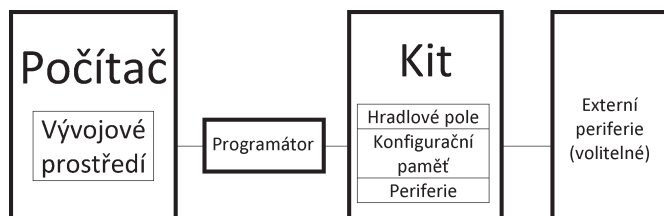
ÚVOD	4
ZAČÍNÁME PROGRAMOVAT	6
1. program – „Hello world“	7
2. program – sčítačka	17
3. program – blikající ledka	21
4. program – běžící světlo	26
5. program – morseovka	31
6. program – sedmisegmentový displej	37
7. program – sedmisegmentový displej jinak	43
8. program – sčítačka s displejem	46
9. program – stopky	51
10. program – hodiny	60
11. program – budík	67
12. program – zvuk	78
13. program – PWM	81
14. program – PWM jinak	86
15. program – efekt	89
16. program – řízení otáček motoru	94
17. program – řízení otáček motoru s reverzací	99
18. program – obousměrný port typu otevřený kolektor	104
19. program – paměti	109
20. program – Digital Clock Manager	117
21. program – asynchronní led	120
ZÁVĚR.....	122
LITERATURA A JINÉ ZDROJE	122
PŘÍLOHA	123

Hradlová pole vždy patřila do sféry profesionálů a ti, pro které je elektronika koníčkem, se jim, bohužel, vyhýbali. Důvody byly a i nadále zůstávají různé. U první generace hradlových polí to jistě bylo i proto, že nabízené funkce nebyly příliš pokročilé, a když už bylo potřeba, dal se daný problém řešit pomocí klasických integrovaných obvodů. Později již byly nabízeny i hodně složité možnosti, ale hradlová pole se tím vzdálila tomu, co musí koníček poskytovat. Hru a zábavu. Proto se řada amatérských elektroniků věnuje hrátkám s jednočipovými mikropočítači, ale hradlová pole jsou pro ně bílým místem. To je jistě škoda, protože současná hradlová pole 3. generace, hovoří se o nich jako o FPGA, již jsou schopny poskytnout i tu potřebnou zábavu. A také tvůrčí seberealizaci, která může snadno přejít do profesionální podoby. Protože spektrum aplikací hradlových polí má trvale vzestupnou tendenci, jistě není nikdy pozdě se začít jejich programování věnovat.

Tato knížka vám formou řešených příkladů ukáže, jak začít. Pokud vás tato problematika zaujme, může vás nasměrovat i z hlediska profesní orientace.

Jak začít

V úvodu je třeba si položit otázku, co všechno budete potřebovat. Tato otázka je velmi komplexní a patří k ní i potřeba vstupních znalostí. Důležitá je i volba techniky, se kterou budete pracovat. Přesto, že v principu je možné učit se programování s prakticky libovolným hradlovým polem, je volba rozhodující pro množství komplikací, se kterými se později setkáte.



Obr. 1 Co potřebujete?

Je tedy třeba vyjasnit si, s jakým hradlovým polem budete pracovat, jak je dostupný potřebný hardware a software a v neposlední řadě je to i dostupnost získání rady v okamžiku, kdy nebudete vědět jak dál. Je třeba si uvědomit, že úvaha, která vede k výslednému rozhodnutí, je velmi subjektivní.

Vstupní znalosti

Hradlová pole patří mezi digitální obvody. Proto je vhodné, ale nikoliv nutné, mít alespoň základní znalosti číslicové techniky. Ty lze, samozřejmě kromě školy, získat studiem odborné literatury, např. [1] nebo [2]. V době internetu však existuje nepřeberná řada zdrojů, kde je možné si doplnit potřebné vědomosti. V krajním případě stačí zadat neznámý pojem do vyhledávače a najít vhodný odkaz. Praktické zkušenosti ukazují, že je opravdu možné postupovat od programování ve VHDL k rozvoji znalostí číslicové techniky.

Hardware

Tuto kapitolu je nutné začít u vlastních hradlových polí a jejich výrobců.

Není příliš těch, kteří tyto součástky zvládají. Mezi ty hlavní patří firmy Xilinx, Altera, Lattice a Actel. Pořadí je uvedeno podle velikosti produkce. Existují však ještě i další, minoritní výrobci. Za zmínku stojí firmy Silicon Blue, Achronix a Abound Logic.

Samozřejmě výrobci mají ve své nabídce řadu typů, z nichž si konstruktéři mohou vybírat. Není však předmětem této knihy probírat jednotlivé nabízené řady a typy. Podrobnosti je možné nalézt na webových stránkách výrobců [3] až [9].

Vlastní hradlové pole tedy budeme dále považovat za danou věc. K jednotlivým typům existují několikaset stránkové dokumentace a kromě rozsahu je možnou komplikací, že jsou v angličtině. Obecné principy je možné nastudovat v dostupné literatuře, např. [10].

Samotné hradlové pole však nestačí. Až na výjimky musí být doplněno o konfigurační paměť. V každém případě potřebujeme zdroj hodinových impulzů a nějaké periferie nebo alespoň konektory, které by umožnily je připojit. Také je nutné nezapomenout na programátor, který nám zajistí zavedení programu do hradlového pole, případně do konfigurační paměti. Pokud k tomu zohledníme ještě otázku pouzder, kdy v naprosté většině případů nelze uvažovat o vlastnoručním zapájení, je rozumné pořídit si ke zkoušení něco z výroby profesionálů. Ti vyrábí tzv. kity. Jedná se o desku plošného spoje, která je osazena hradlovým polem a vším, co je k vlastní funkci nezbytné. Kromě toho je kit vybaven různými periferiemi. Ovšem problém je, co vybrat, protože nabídka je velmi široká. Pro začátečníka, který v tuto chvíli v hradlových polích objevil svého koníčka, bude zřejmě rozhodujícím kritériem podpora výrobce, dostupnost, cena a podpora komunity. Určitě bude dobré vybírat z nabídky velkého výrobce, kde je předpoklad, že vyberete něco, co je rozšířené a tudíž všeobecně známé.

Největším výrobcem hradlových polí FPGA je jednoznačně firma Xilinx. Podporu ze strany tohoto výrobce je možné ohodnotit jako velmi dobrou. Kity s hradlovými poli firmy Xilinx nabízí řada výrobců. Zajímavé jsou např. ty, které vyrábí firma Digilent [11]. Dostupné jsou buď přímo u výrobce, nebo u dodavatelů. Podle typu hradlového pole a složitosti kitu je i různá cena. Pro první seznámení s hradlovými poli zřejmě výběr padne na některý z tzv. low cost typů. Z produkce firmy Xilinx to bude některý z řady Spartan3 nebo Spartan3E. Zde je však třeba upozornit na rychlost dalšího vývoje a tak je dnes možné uvažovat i o typu Spartan6, který výrobce rovněž řadí mezi low cost. V nejnovější nabídce je kit Spartan-6 FPGA SP601 Evaluation Kit. V této chvíli však jsou tři důvody, proč není možné předpokládat okamžité použití. Nejsou praktické zkušenosti, vybavení kitu je velmi omezené a v neposlední řadě je to cena.

Podporu komunity je možné hledat na různých diskusních fórech. Jedno je přímo na stránkách firmy [12]. Další je např. [13].

Pokud má být předchozí shrnutí konkretizováno, lze z nabídky vybrat tři typy kitů, které jsou uvedeny v příloze.

Vývojové prostředí

Programování hradlových polí je kvalitativně něco úplně jiného, než známe z postupů ve vyšších programovacích jazycích. Je třeba vykonat řadu postupných kroků od zadání až po naprogramování obvodu. Proto každý výrobce poskytuje software, který nám toto všechno umožňuje. Souhrnně pak tento sw nazýváme vývojové prostředí. Vývojové prostředí je tedy nástroj programátora. Je zvykem poskytovat základní funkce zdarma, byť pouze po předchozí registraci a s různě omezenou dobou platnosti licence. Je třeba zdůraznit nepřenositelnost vývojových prostředí mezi jednotlivými výrobci. Vzhledem k naznačené orientaci na hradlová pole firmy Xilinx je tedy logické využití vývojového prostředí od téže firmy. To je nabízeno pod názvem ISE Web Pack. Samozřejmě i tento software podléhá velké dynamice vývoje. I když jsou jednotlivé verze velmi podobné, přesto jsou některé změny velmi zásadní. Navíc rozsah funkcí vývojového prostředí neumožňuje jednoduchý návod na jeho použití. Proto bývá zvykem nabídnout vhodný tutoriál. Pro začátečníky se pak nutně jedná o velmi rozsáhlý materiál, kterým by velmi narostl rozsah této knížky. Aby k tomu nedošlo a zároveň byla zachována aktuálnost této knihy, je zpracovaný tutoriál v elektronické podobě k dispozici na stránkách nakladatelství.

Programovací jazyky

Zvládnutí dnešních hradlových polí předpokládá zvládnutí vhodného programovacího jazyka. Ty hlavní v současné době existují dva, VHDL a Verilog. Oba jsou rovnocenné, co se poskytovaných funkcí týká, i když se samozřejmě navzájem liší. Pro rozhodnutí, kterému jazyku se věnovat, je možné v podstatě jen kritérium rozšíření toho kterého jazyka. Jazyk Verilog dominuje v Asii a v USA. V Evropě (a tedy i u nás) je převaha jazyka VHDL. Je třeba zmínit, že existují snahy přenést programování na úroveň programovacích jazyků C nebo C++.

Jazyk VHDL

Jazyk VHDL byl původně vyvinut především pro modelování a simulaci rozsáhlých číslicových systémů na základě zadání ministerstva obrany USA. Tento vývoj začal v roce 1981 a první vydání se objevilo v roce 1987 ve formě standardu IEEE Std 1076–1987. Dnes nej-používanější verze je z roku 1993 a to přesto, že je následovaly v letech 1999, 2000 a 2002 ještě další tři verze. V této době je očekáván nový standard. Ten však jistě nebude mít vliv na aktuálnost toho, co si v této knize ukážeme. Bližší podrobnosti o historii a vývoji jazyka VHDL je možné si přečíst v různé literatuře, např. [10].

Pro nás je podstatné, že nám jazyk VHDL umožní popsat řešení, které je tzv. syntetizovatelné. Tedy takové, které budeme schopni převést do podoby konfiguračního souboru. Tento soubor nám pak po zavedení do hradlového pole zajistí fyzickou funkci respektující zadání.

Základní vlastnosti jazyka VHDL jsou následující:

- Otevřený standard.
- Návrh může být zahájen bez znalosti cílového obvodu.
- Přenositelnost kódu.
- Umožňuje přímou simulaci.
- Umožňuje implementaci do vybraného obvodu.

K zadání, které má program realizovat, přistupuje jazyk VHDL jako k „černé skřínce“. Tu pak popisuje zvnějšku a zevnitř. Zvnějšku je to propojení pomocí tzv. vstupních a výstupních portů. Uvnitř pak je to popis chování, které zajišťuje požadovanou funkcionalitu.

Pokud se podíváme na možnosti jazyka VHDL, musíme konstatovat, že jsou velmi rozsáhlé. Protože však tato knížka je určena pro začátečníky, je třeba současně zdůraznit, že není nutné znát všechno. Už z mála lze vytvořit zajímavé programy.

Začínáme programovat

Je známou zkušeností, že programování se nedá naučit teoreticky. Proto dále budeme postupovat řešením příkladů. Tento postup bude samozřejmě od nejjednoduššího ke složitějšímu. Náš postup bude využívat jen to, co je bezpodmínečně nutné. Jedná se o následující kroky.

- Založení projektu.
- Vytvoření zdrojových souborů.
- Kontrola syntaxe, tedy jestli vývojové prostředí rozumí tomu, co jsme napsali.
- Simulace. Ta nám slouží k odhalení, zda vývojové prostředí nerozumí našemu programu jinak, než jsme to mysleli.
- Vytvoření ucf souboru, který nám popisuje fyzické propojení našeho obvodu na vývody z pouzdra.
- Vygenerování konfiguračního souboru bit (pro hradlové pole) nebo mcs (pro konfigurační paměť).

Tutoriál ukazuje mnohem více možností vývojového prostředí, ale to přesahuje záměr této knihy.