

IMPLEMENTATION OF 8-BIT MICROPROCESSOR INTO FPGA CHIP

Ondřej Wallezký

Bachelor Degree Programme (3), FEEC BUT

E-mail: xwalle00@stud.feec.vutbr.cz

Supervised by: Marek Bohrn

E-mail: bohrn@phd.feec.vutbr.cz

Abstract: This paper summarizes thesis that deals with design of architecture compatible with one of 8-bit microcontrollers from Microchip company for FPGA implementation. First part of this paper describes design concept of this architecture. It mentions basic issues and concept design of data memory, describes I/O port design and principal function of system controller. Second part describes method used to design this system as well as way to change some of its parameters and program the system.

Keywords: PIC, Microchip, architecture, FPGA, ASIC, microcontroller, design

1. ÚVOD

Cílem článku je seznámit čtenáře o bakalářské práci, jejímž cílem je navrhnout 8bitový mikroprocesor, který by byl na úrovni instrukčního souboru kompatibilní s jedním z 8bitových mikrokontrolérů firmy Microchip. Tento mikroprocesor má být následně implementován do obvodu FPGA. Důvodem vzniku této práce je nedostatek kvalitních obvodů tohoto typu. Tento obvod by totiž bylo možno při správném návrhu použít jako řídicí blok obvodu ASIC, což je sekundárním cílem práce. Pro autora se navíc jedná o užitečnou zkušenost, jelikož se jedná o návrh komplexního číslicového systému.

První část článku se zabývá koncepcí návrhu obvodu, druhá část seznámí čtenáře o použité metodice realizace tohoto obvodu.

2. NÁVRH OBVODU

Pro návrh daného mikroprocesoru je nutná znalost obvodu, který byl vybrán jako vzor [1]. Důvodem jsou požadavky na kompatibilitu mikroprocesoru s tímto obvodem. Procesor musí být schopen provádět program na základě instrukcí používaných cílovým mikrokontrolérem [2] a jeho chování musí co nejvíce s tímto mikrokontrolérem korespondovat pro co nejlepší přenositelnost programu.

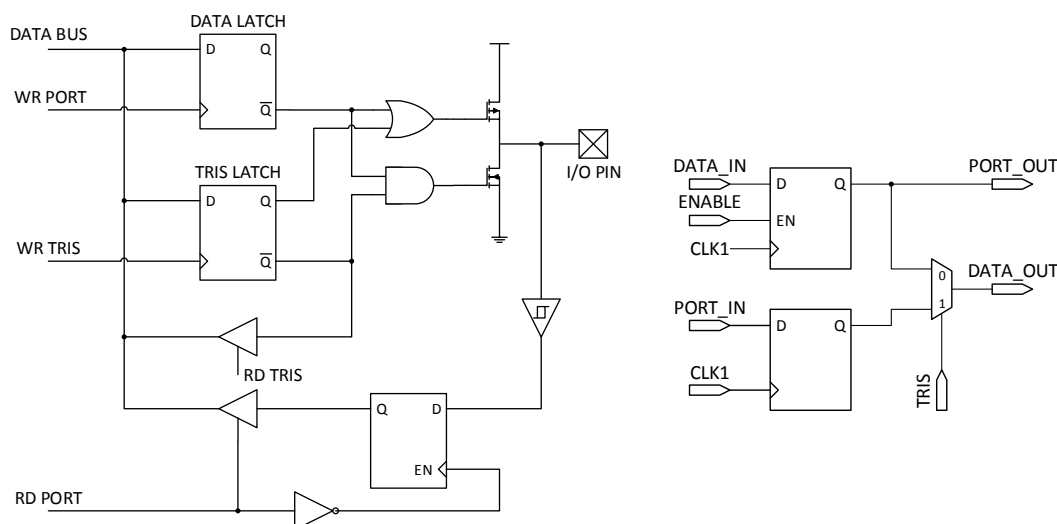
Vzhledem k implementaci jádra do obvodu FPGA nelze uvažovat realizaci analogových bloků. V navrhovaném jádře jsou z periferních obvodů použity jen čítače a časovače [1] kvůli jejich širokému využití v praxi. Některé funkce těchto obvodů (např. asynchronní režim časovače Timer 1) nejsou z důvodu synchronního návrhu podporovány, případně je původní funkce pozměněna nebo nahrazena funkcí jinou.

2.1. DATOVÁ PAMĚŤ

Datová paměť se skládá ze dvou částí. První z nich je paměť RAM o velikosti 192 B. Je realizována jako paměťový blok s asynchronním výstupem [3]. Druhou částí jsou registry, které mají funkci vyhrazenou pro různé systémové účely. Vzhledem k potřebě vést interní vodiče mimo datovou sběrnici jsou tyto registry realizovány prostřednictvím klopných obvodů.

2.2. VSTUPY A VÝSTUPY

Navržený obvod má 3 vstupně-výstupní porty o volitelné šířce 1–8 bitů, z toho jeden port (PORTB) má rozšířenou funkcionalitu o podporu přerušení při změně hodnoty na libovolném vstupním pinu. Z důvodu absence třístavových hradel v obvodech FPGA má každý port oddělené vstupní a výstupní vodiče. Nastavení připojení vstupu či výstupu k datové sběrnici je realizováno pomocí příslušného registru TRIS, který v původním obvodu slouží k nastavení jednotlivých pinů portu jako vstupní nebo výstupní. Tímto je zajištěna kompatibilita s původním obvodem. Obrázek 1 srovnává navržený port s obecným portem mikrokontroléru PIC.



Obrázek 1: Obecný 8bitový port obvodu PIC (vlevo) [2] a navržený port (vpravo)

2.3. ZPRACOVÁNÍ INSTRUKCÍ

Obvod je řízen programem uloženým v paměti ROM. Instrukce jsou adresovány prostřednictvím programového čítače, jehož šířka je volitelná v rozsahu 11–14 bitů. Tato šířka automaticky určuje kapacitu paměti ROM, ze které jsou adresované instrukce načítány do instrukčního registru řadiče každý čtvrtý takt hlavního hodinového signálu. Kromě šířky programového čítače lze nastavit počet úrovní zásobníku instrukcí.

Struktura řadiče je rozdělena na dvě části. První částí je instrukční dekodér, který dekoduje instrukci právě uloženou v instrukčním registru a převede ji na údaje potřebné k vykonání instrukce. Druhou částí je řídicí část, která je realizována jako soustava souběžně běžících stavových automatů. Řízení činnosti procesoru je pak jednotlivým automatům přidělováno instrukčním dekodérem prostřednictvím multiplexoru. Každý z těchto automatů reprezentuje skupinu instrukcí, které sdílí způsob toku dat uvnitř obvodu.

Provádění instrukce je rozděleno do dvou fází – načtení instrukce a její provedení. Během provádění instrukce probíhá načtení instrukce následující a dochází tak k jejich zřetězení. Obě fáze mají délku jednoho instrukčního cyklu, jenž je pevně dán a trvá 4 takty hlavního hodinového signálu. V případě instrukce skoku dojde k přepsání programového čítače až po načtení následující instrukce, a pro další cyklus je proto načtená instrukce nahrazena instrukcí NOP, která neprovádí žádnou činnost. [2]

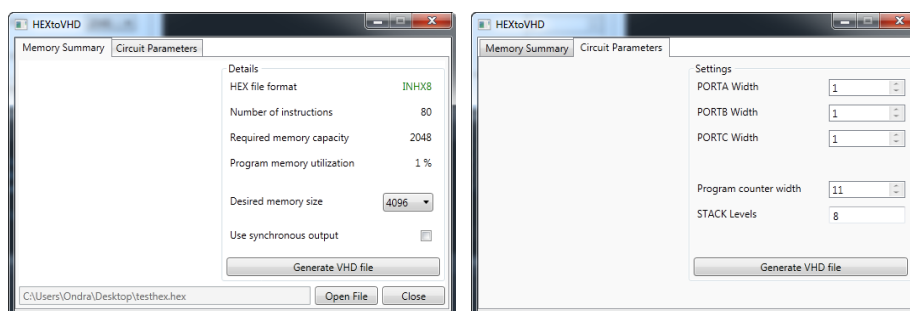
Struktura řadiče není v dokumentaci vzorového obvodu podrobně popsána. Jeho návrh je tak založen pouze na popisu chování obvodu v různých situacích, jako je např. provádění instrukce skoku nebo obsluha požadavků přerušení.

3. REALIZACE OBVODU

Realizace obvodu je provedena prostřednictvím jazyka VHDL. Je použit strukturální zápis s implementovanou genericitou pro podporu změn parametrů obvodu, konkrétně velikostí vstupně-výstupních portů a parametrů programového čítače.

Realizovaný obvod používá dvou hodinových signálů. Základem je systémový hodinový signál CLK1, který je připojen ke všem sekvenčním obvodům, doplňkovým signálem je povolovací signál CLK4, jehož perioda trvá 4 takty signálu CLK1, má střihu 25 % a reprezentuje délku jednoho instrukčního cyklu. Pro realizaci jádra v obvodu FPGA je použit synchronní reset, pro implementaci do obvodu ASIC bude na základě konzultace s odborným vedoucím použit reset asynchronní, převážně z důvodu menší plochy a nezávislosti resetu na hodinovém signálu. To vede ke dvěma samostatným popisům obvodu.

Popis obvodu je doplněn podpůrným programem běžícím na platformě .NET, který slouží k nastavení parametrů obvodu a generaci popisu programové paměti ROM ze zdrojového HEX souboru, viz Obrázek . Tento soubor je generován příslušným vývojovým prostředím. Program pro procesor tak lze psát v prostředí MPLAB poskytovaném výrobcem vzorového obvodu, tj. firmou Microchip.



Obrázek 2: Podpůrný program pro úpravu obvodu

4. ZÁVĚR

Tento článek shrnuje závěrečnou práci, již se autor zabývá. V závěrečné práci byla provedena analýza funkce vybraného mikrokontroléru PIC firmy Microchip a na jejím základě byl navržen kompatibilní 8bitový mikroprocesor. Tento mikroprocesor je doplněn o vstupně-výstupní porty, čítače a časovače, stejně tak o možnost úpravy některých parametrů, jako je např. šířka programového čítače. V mikroprocesoru nejsou z důvodu snížení komplexnosti implementovány některé funkce jako úsporný režim nebo tzv. časovač Watchdog. Instrukce s těmito perifériemi související jsou nahrazeny instrukcí NOP přímo v řadiči. V současné době probíhá testování funkce obvodu a dokončení programu.

REFERENCE

- [1] MICROCHIP TECHNOLOGY INC. *PIC16(L)F722A/723A Data Sheet: 28-Pin Flash Microcontrollers with nanoWatt XLP Technology* [online]. 2010–2012, 286 s. [cit. 2.3.2014]. Dostupné z: <http://ww1.microchip.com/downloads/en/DeviceDoc/41417B.pdf>
- [2] MICROCHIP TECHNOLOGY INC. *PICmicro™ Mid-Range MCU Family Reference Manual* [online]. Prosinec 1997, 688 s. [cit. 2.3.2014]. Dostupné z: <http://ww1.microchip.com/downloads/en/devicedoc/33023a.pdf>
- [3] XILINX INC. *Using Look-Up Tables as Distributed RAM in Spartan-3 Generation FPGAs* [online]. Ver. 2.0, 1.3.2005, 12 s. [cit. 2.3.2014]. 2.0. Dostupné z: http://www.xilinx.com/support/documentation/application_notes/xapp464.pdf