

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA ELEKTRONICKÁ

KATEDRA ŘÍDÍCÍ TECHNIKY



BAKALÁŘSKÁ PRÁCE

IR komunikace pomocí FPGA

Praha, 2007

Novotný Jan

Abstrakt

Tato bakalářská práce se zabývá přenosem dat pomocí infračervených paprsků.

V teoretické části jsou popsány nejpoužívanější způsoby modulace, datový přenos pomocí IrDA a protokoly používané dálkovými ovladači.

Praktická část zahrnuje připojení modulu infraportu k hradlovému poli FPGA. Součástí práce je také vytvoření systému pro ovládání inraportu v jazyce VHDL a jeho naprogramování pro příjem a vysílání informace v různých formátech. V systému je využit mikrokontrolér Picoblaze.

V textu jsou popsány základní části programu jazyka VHDL, které lze nalézt na přiloženém CD.

Abstract

This bachelor thesis is aimed to data transfer by infrared radiation. Theoretical section familiarizes with the most using modulation, data transfer and protocols for remote controls.

A practical part of the thesis describes interconnection between the infraport modul and FPGA. There is mentioned system implemented in FPGA for controlling infrared module, which allows the machine receives and transmissions information in several formats. The system is created in VHDL language and the system uses microcontroler PicoBlaze. All programs and projects created for the thesis are enclosed on the CD.

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v příloženém seznamu.

V Praze, dne

.....

podpis

Poděkování

Na tomto místě bych chtěl poděkovat vedoucímu mé práce ing. Romanu Bartosinskému za cenné připomínky a rady při řešení problému související s bakalářkou prací.

Katedra řídicí techniky

Školní rok: 2006/2007

Zadání bakalářské práce

Student: Jan Novotný
Obor: Kybernetika a měření
Název tématu: IR komunikace pomocí FPGA

Zásady pro vypracování:

1. Seznamte se s přenosem dat pomocí IR komunikace.
2. Seznamte se s návrhem a implementací hardwaru pro obvody FPGA pomocí nástroje Xilinx ISE a s mikro-kontrolérem PicoBlaze.
3. Navrhněte a implementujte připojení modulu IR portu k mikro-kontroléru PicoBlaze.
4. Demonstrujte funkčnost implementace řízením vývojové desky pomocí dálkového ovladače.

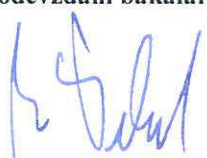
Seznam odborné literatury:

- Peter J. Ashenden, The Designer's Guide to VHDL, San Francisco, California, 1995
- Matthew Aubury, RC 10 Platform Development Manual, Celoxica, UK, 2005
- Rob Otte, Low-Power Wireless Infrared Communications, Norwell, USA, 1999


Vedoucí bakalářské práce: Ing. Roman Bartosinski

Datum zadání bakalářské práce: zimní semestr 2006/07

Termín odevzdání bakalářské práce: 15. 8. 2007


Prof. Ing. Michael Šebek, DrSc.
vedoucí katedry




Prof. Ing. Zbyněk Škvor, CSc.
děkan

V Praze, dne 6. 3. 2007

ÚVOD	3
1 IR KOMUNIKACE	4
1.1 IRDA	4
1.1.1 FYZICKÁ VRSTVA	4
1.1.2 POUŽÍVANÉ RYCHLOSTI A FORMÁT DAT.....	4
1.2 MODULACE IR.....	5
1.2.1 PULSNÍ MODULACE.....	5
1.2.2 FSK MODULACE.....	6
1.2.3 BI - PHASE MODULACE.....	6
1.2.4 PULSNĚ ŠÍRKOVÁ MODULACE	7
1.2.5 SHRnutí JEDNOTLIVÝCH MODULACÍ	7
1.3 PROTOKOLY PRO DÁLKOVÉ OVLADAČE.....	7
1.3.1 PROTOKOL RC-5	7
1.3.2 NEC STANDARD	8
2 HRADLOVÉ POLE	9
2.1 FPGA.....	9
2.2 JAZYKY PRO PROGRAMOVÁNÍ FPGA.....	10
2.2.1 PROGRAMOVACÍ JAZYK VHDL	10
2.3 PICOBLAZE	11
2.4 RC10.....	12
3 POPIS REALIZACE	13
3.1 MODUL INFRAPORTU	13
3.1.1 PŘIPOJENÍ MODULU INFRAPORTU K DESCE RC10.....	13
3.2 ARCHITEKTURA	14
3.2.1 SPOLUPRÁCE JEDNOLIVÝCH KOMPONENT	14
3.2.2 VSTUPNÍ A VÝSTUPNÍ PORTY ARCHITEKTURY RIZENI.....	15
3.2.3 PROCESY	15
3.2.4 SIGNÁLY.....	15
3.2.5 PAMĚŤ RAM.....	16
3.3 HODINOVÉ SIGNÁLY	16
3.4 REŽIMY PŘENOSU DAT	17
3.4.1 REŽIM TIME	18
3.4.1.1 Filtr.....	18

3.4.2 REŽIM UART	18
3.5 PICOBLAZE	19
3.5.1 PŘÍKAZY PRO SÉRIOVOU KOMUNIKACI.....	20
3.6 DEKÓDOVÁNÍ SIGNÁLU OVLADAČE.....	20
3.7 OBSAZENOST HRADLOVÉHO POLE.....	22
ZÁVĚR	24
POUŽITÁ LITERATURA.....	25
PŘÍLOHY	26

ÚVOD

Přenos dat prostřednictvím infračerveného paprsku patří mezi jednu z nejstarších forem moderní bezdrátové datové komunikace. Infračervené rozhraní je dnes již běžnou součástí komunikační výbavy mobilních telefonů a infraportem jsou vybaveny prakticky všechny kapesní počítače. Disponují jím i některé tiskárny, fotoaparáty a videokamery. Standardně je elektronika infračerveného rozhraní součástí i většiny nových základních desek pro osobní počítače, avšak optickou část je nutno v případě potřeby doplnit o IR přenos. Pravděpodobně největším konkurentem infračerveného rozhraní je technologie Bluetooth, která je, stejně jako IrDA, relativně levná a snadno použitelná. V současnosti obě tyto technologie existují vedle sebe a doplňují se navzájem. Prakticky každý mobilní telefon vyšší a v některých případech i střední třídy využívá obě komunikační rozhraní.

Vzhledem ke stávajícímu rozšíření a širokému spektru možných aplikací lze předpokládat, že i přes některé své záporné vlastnosti má infračervené rozhraní před sebou stále ještě budoucnost, i když je skutečností, že bude postupně vytlačováno právě stále více se rozšiřující technologií Bluetooth.

V práci se budu zabývat IR komunikací. Pomocí jazyka VHDL navrhnu implementaci přenosem dat ve formátu IrDA a jejími specifikacemi a zaznamenám infračervený kódový signál dálkového ovladače.

1 IR KOMUNIKACE

1.1 IrDA

IrDA je standard vytvořený IrDA konsorciem (Infrared Data Association), který definuje, jak bezdrátově přenášet digitální data pomocí infračerveného záření. IrDA ve svých specifikacích definuje standardy jak fyzických koncových zařízení, tak protokolu, jímž komunikují IrDA zařízení. IrDA standard vznikl z potřeby mobilně propojit různá zařízení mezi sebou (hlavní využití IrDA je pro spojení notebooku či různých osobních komunikátorů, ale IrDA rozhraním jsou vybavovány například mobilní telefony, PDA, video kamery atd.)

1.1.1 Fyzická vrstva

K přenosu informace infračerveným přenosovým kanálem je zapotřebí vysílač a přijímač infračerveného záření převádějící elektrický signál na optické záření a naopak. Pro infračervený přenos se dosud používala výhradně oblast blízkého infračerveného záření s vlnovou délkou v rozmezí 840 - 960nm. V současné době se však již objevuje nový standard pokrývající pásmo 700 - 1600nm. Na tuto vlnovou délku jsou citlivé i mnohé CCD kamery. Přijímačem jsou PIN fotodiody, které pracují v generačním režimu (při dopadu světla na přijímač "vyrazí" světlo elektrony, které se odvádí do filtru, ten propustí jen ty frekvence povolené pro daný typ IrDA modulace). Existuje přímá úměra mezi energií dopadnutého záření a nábojem, který optická část přijímače vygeneruje.

1.1.2 Používané rychlosti a formát dat

IrDA zařízení dle normy IrDA 1.0 pracují do vzdálenosti 1.0 m při bitové chybovosti BER (bit error ratio, poměr chybně přenesených bitů ku správně přeneseným) 10^{-9} a maximální úrovni okolního osvětlení 10klux (denní svit slunce). Tyto hodnoty jsou definovány pro nesouosost vysílače a přijímače 15 stupňů, pro jednotlivé optické prvky se měří výkon do 30 stupňů.

Rychlosti jsou pro IrDA v. 1.0 od 2400 do 115200 kbps. Formát dat je stejný jako na sériovém portu, tedy asynchronně vysílané slovo uvozené startbitem (viz kapitola o obr.1.1).

Norma IrDA 1.1 navíc definuje rychlosti 576 kbit/s a 1,152 Mbit/s. Pro tyto rychlosti je využito pulsního kódování o délce čtvrtiny doby impulsu. Při těchto rychlostech je již nutné signál vysílat synchronně.

Dále verze infračerveného rozhraní označovaná jako verze 1.4 nebo zkratkou VFIR umožňuje maximální rychlost až 16 Mb/s. Bývá však využívána jen ve zcela výjimečných případech. Nejnovější projekt označovaný zkratkou UFIR dokáže pracovat dokonce s rychlostí nad 100 Mb/s.

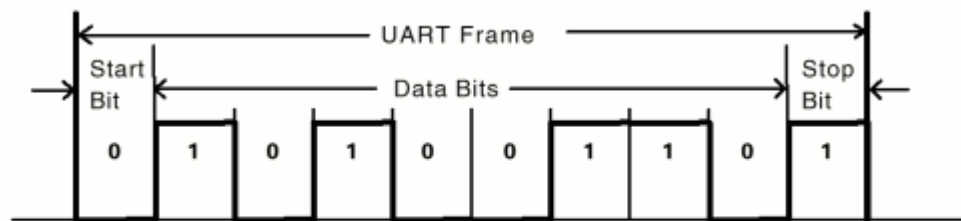


Figure 11a. UART Frame

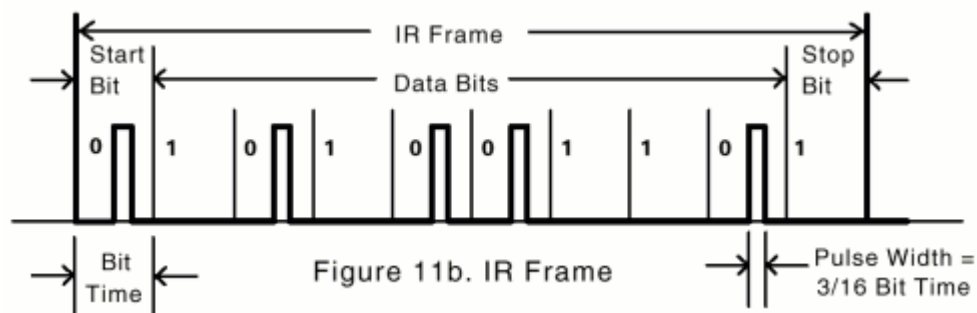


Figure 11b. IR Frame

Obr. 1.1 Formát dat IrDA

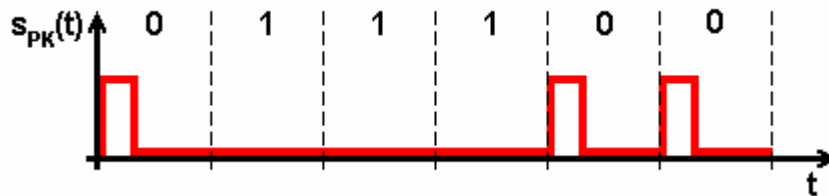
Vysílací strana může používat buď $3/16$ délky trvání bitu, nebo může užít fixní délku optického impulsu $1,6 \mu\text{s}$.

1.2 Modulace IR

1.2.1 Pulsní modulace

Pulsní modulace (znázorněná na obr. 1.1) je jedním z nejjednodušších způsobů kódování dat přenášených infračerveným zářením. Principem této modulace je rozdělení signálu na stejně velká tzv. časová okna. V tomto okně se buď vyskytne, nebo nevyskytne impuls konstantní délky, zpravidla podstatně menší, než je délka časového okna. Výskyt tohoto impulsu v okně je považován za logickou nulu, jeho absence pak za logickou jed-

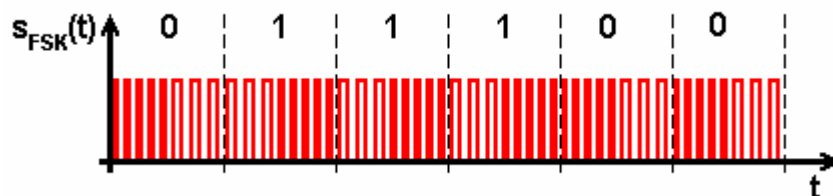
ničku. Hodinové impulsy na přijímací straně se synchronizují s hranou přijímaného signálu. Při přenosu většího bloku jedničkových bitů, kdy nejsou vysílány žádné impulsy, by mohl přijímač vypadnout ze synchronismu. Proto se zde používá tzv. bit-stuffing, kdy po určitém počtu jedničkových bitů je vyslán navíc jeden nulový. Tento bit je pak třeba na přijímací straně opět odstranit.



Obr. 1.2 Pulsní modulace

1.2.2 FSK modulace

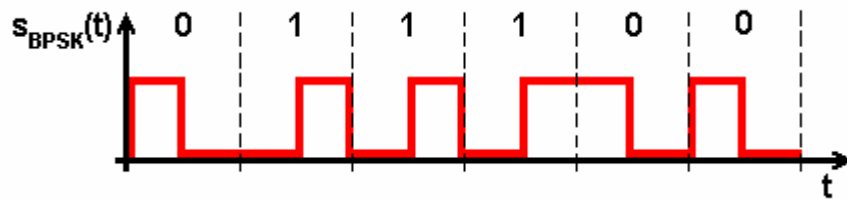
FSK (frequency shift keying) se používá jako velice bezpečná metoda přenosu. Princip spočívá v modulování bitů pomocí dvou frekvencí, jak je znázorněno na obr. 1.2. Nevýhodou je vyšší cena zařízení, vyšší energetická náročnost a pomalejší datový přenos. Z těchto důvodů se FSK modulace používá pouze výjimečně v systémech s požadavky na vysokou bezpečnost přenosu.



Obr. 1.3 FSK modulace

1.2.3 Bi - phase modulace

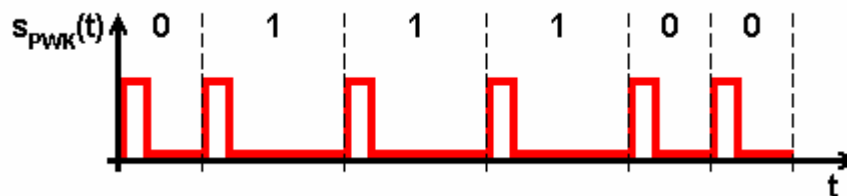
Princip bi-phase modulace je znázorněn na obr.1.3 a spočívá opět v rozdělení signálu na tzv. časová okna. Tato časová okna mají stejnou velikost a pro detekci daného bitu se bere změna úrovně signálu uvnitř tohoto okna. Pokud je změna kladná, tj. z log.0 na log.1, je bit vyhodnocen jako 1, v opačném případě jako 0.



Obr. 1.4 Bi – phase modulace

1.2.4 Pulsně šířková modulace

Principem pulsně šířkové modulace je opět rozdělení signálu do tzv. časových oken, tentokrát však ne stejně dlouhých (viz obr.1.4). Jednotlivá časová okna jsou synchronizována náběžnou hranou signálu a detekce bitu se provádí podle délky okna. Kratšímu oknu odpovídá bit 0, delšímu pak bit 1.



Obr. 1.5 Pulsně šířková modulace

1.2.5 Shrnutí jednotlivých modulací

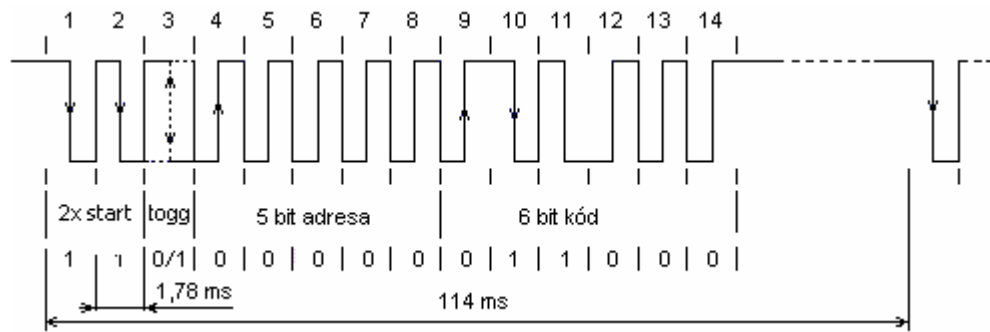
Všechny druhy modulace, s výjimkou modulace FSK, je možno provozovat ve dvou módech. První možností je modulace kódovaného signálu na nosnou frekvenci, druhou pak pulsní přenos. V praktických aplikacích se modulace na nosnou frekvenci používá u bi-phase kódování, pulsní přenos, často také nazývaný „flash“ mód, je častější u pulsně šířkové modulace.

1.3 Protokoly pro dálkové ovladače

1.3.1 Protokol RC-5

Přenosový standard RC-5 byl vyvinut pro dálkové ovladače spotřební elektroniky a tomu odpovídá i struktura rámce s netypickou délkou dat 6 bitů. Je zde použito bi-phase kódování s modulací na nosný kmitočet 36 kHz. Každý bit obsahuje burst 32 impulsů o již zmíněné frekvenci. Přenos rámce začíná dvěma start bity následovanými toggle bitem.

Toggle bit mění svou hodnotu s každým rámcem, je tedy takto možné identifikovat nedoručení předchozího rámce. Informační blok rámce sestává z pěti adresových bitů, určujících konkrétní ovládané zařízení a z šesti datových bitů, představujících vlastní příkaz. Vlastní rámec je vysílán tak dlouho, dokud je stisknuté příslušné tlačítko na ovladači. Struktura RC-5 protokolu s příkladem datového rámce je uvedena na obr.1.6.



Obr. 1.6 Protokol RC-5

1.3.2 NEC standard

Jako RC5 rovněž používá dodatečnou modulaci na nosnou frekvenci 36 kHz, ale základem je pulsně šířkové klíčování. Konstantní délky rámce je dosaženo dvojnásobným opakováním přenášených dat (v přímé a v invertované podobě), což slouží k zabezpečení přenosu proti chybám. Na začátku každého rámce je po dobu 9 ms vysílán shluk impulsů, pomocí něhož nastaví přijímací modul svou citlivost. Dále následuje osmibitová adresa a příslušný příkaz taktéž osmibitový. Jestliže je tlačítko ovladače stisknuto delší dobu, pak se za prvním kompletním rámcem vysílají pouze zaváděcí impulsy a jen jeden bit.

2 HRADLOVÉ POLE

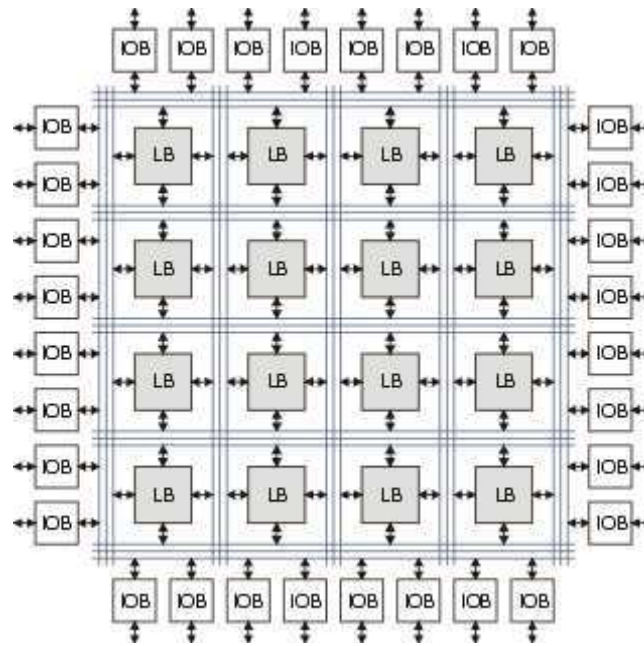
2.1 FPGA

Field Programmable Gate Array je samostatná třída aplikačně specifických IO, která umožňuje realizovat elektronický systém v jediném integrovaném obvodu naprogramováním jeho propojení a nastavení logickými bloky. Nastavení propojení je možné neomezeně krátit a měnit a je provedeno řádově ve stovkách milisekund až sekundách.

Obvody architektury FPGA jsou založeny na malých generátorech logických funkcí s pamětmi (LUT tabulky), klopných obvodech a obvodech sloužících k horizontálnímu a vertikálnímu propojení. Největší současné obvody FPGA obsahují statisíce LUT tabulek a klopných obvodů.

Obvody architektury kombinují výhody PLD obvodů s výhodami plně zákaznických VLSI obvodů a umožňují implementaci rozsáhlých číslicových systémů. Díky podobnosti s plně zákaznickými obvody se dnes používají pro návrh a simulaci jak FPGA obvodů, tak i obvodů plně zákaznických podobné návrhové systémy.

Základem FPGA jsou tři stavební prvky. První prvek jsou programovatelné logické bloky, které obsahují logické prvky (generátory logických funkcí - tabulky LUT, klopné obvody) a lokální propojovací pole. Druhým stavebním prvkem je programovatelné horizontální a vertikální propojení, posledním prvkem jsou programovatelné vstupní/výstupní bloky. V poslední době k těmto blokům přibyly další obvody jako násobičky, paměti, obvody pro úpravu hodinových signálů. Velmi zjednodušené blokové schéma FPGA je znázorněno na obrázku 2.1.



Obr. 2.1 Struktura FPGA

Od začátku devadesátých let do dnešní doby se kapacita FPGA obvodů zvětšila více než 200krát, rychlost se zvýšila 20krát, cena se snížila více než 300krát^[4]. Díky snadné a rychlé změně obsahu FPGA ho použijeme hlavně tam, kde potřebujeme vysoký výpočetní výkon. Obvody FPGA zažívají velmi rychlý vývoj.

2.2 Jazyky pro programování FPGA

Naprogramovat FPGA obvod lze více způsoby. Buď popisem v jazyku typu HDL (Hardware Description Language) nebo vytvořením obvodového schématu logického systému. V současné době je použitelnější využití některého jazyka typu HDL. Pro jazyk existují dva silné standardy nabízející návrhářům téměř podobné možnosti, a to u nás velmi rozšířený jazyk VHDL a jazyk Verilog, oblíbený zejména na americkém kontinentě.

2.2.1 Programovací jazyk VHDL

Very High Speed Integrated Circuits Hardware Description Language (VHDL) je programovací jazyk, který byl navržen a optimalizován pro popis číslicových obvodů a systémů. Začátek vzniku lze přisoudit projektu VHSIC v roce 1981 v rámci programu ministerstva obrany USA. Pak následoval dlouhý vývoj a po několika inovacích je platný standard z roku 2002 (IEEE Std 1076-2002). Díky tomu, že je znát otevřený standard, je možné pro návrh používat simulátory i celá vývojová prostředí různých výrobců.

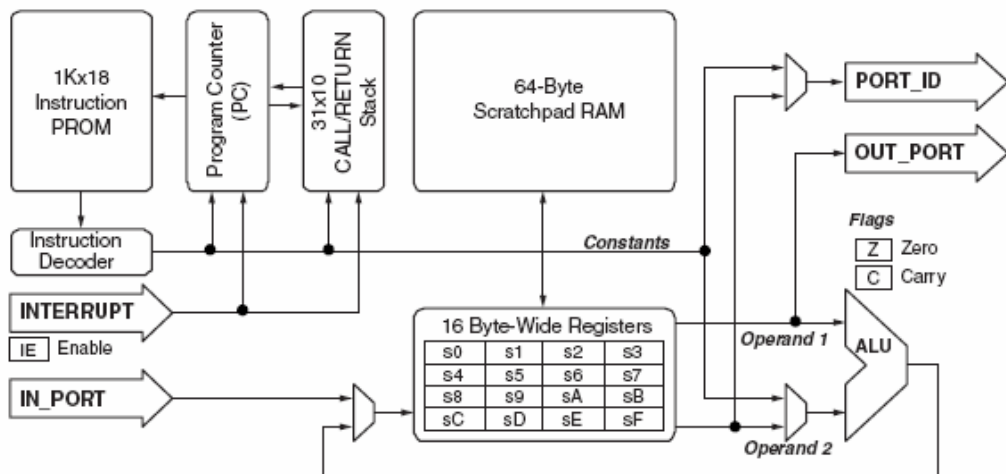
Jazyk kombinuje rysy jazyka pro *modelování a simulaci*, jazyka *návrhového*, jazyka pro *testování* a jazyka pro popis *topologie* (netlist). VHDL zahrnuje mnoho rysů vhodných pro popis chování elektronických součástí od logických hradel po mikroprocesory a zákaznické obvody.

2.3 Picoblaze

Picoblaze je volně dostupný procesor určený pro použití FPGA a CPLD. Je navržen pro snadné použití do FPGA obvodů, protože má velmi jednoduchou instrukční sadu a pro jeho použití není zapotřebí žádných dalších externích periférií. Je vytvořen v jazyce VHDL a je založený na architektuře RISC. Obsahuje

- 16-ti bitové pracovní registry,
- 1K instrukcí programovatelných na chip,
- aritmeticko logickou jednotku (ALU) s CARRY and ZERO příznaky,
- 64 –byte RAM paměť,
- 256 vstupních a 256 výstupních portů.

Blokové schéma procesoru je znázorněno na obr.2.2.



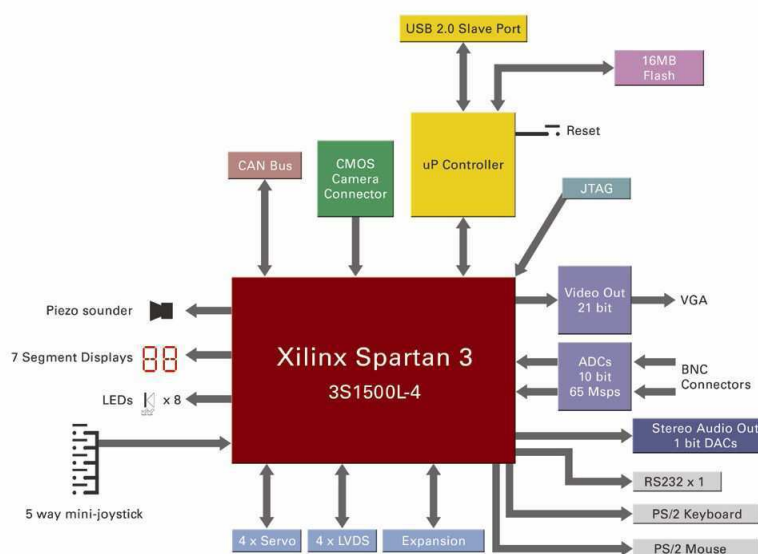
Obr. 2.2 Blokové schéma mikroprocesoru PicoBlaze

2.4 RC10

RC10 je vývojová deska navržena pro průzkum a vývoj FPGA obvodů a určena pro tvorbu aplikací v nízkoúrovňových programovacích jazycích. Deska poskytuje k použití moderní FPGA a vstupní/výstupní zařízení, díky němuž je možné připojit širokou škálu aplikací. S počítačem komunikuje pomocí USB kabelu.

Obsahuje hradlové pole Xilinx Spartan 3 XC3S1500L-4-FG320, 16MB Flash paměť a mnoho dalších periférií (viz obr. 2.3):

- USB konektor,
- videovstup,
- audiovstup,
- sériovou linku RS-232,
- PS/2 konektor,
- CAN sběrnici,
- 8 LED diodu,
- sedmissegmentový LED zobrazovač,
- pětipolohový joystick.



Obr. 2.3 Periferie vývojové desky RC10

3 POPIS REALIZACE

3.1 Modul infraportu

Pro infračervený přenos signálu je použit modul, který obsahuje pin pro příjem/vysílání (TXD,RXD) IR a dva piny pro nastavení MD0, MD1 (viz příloha A). Základ modulu tvoří obvod HSDL3612 od firmy Agilent. Obvod obsahuje IR vysílač, PIN diodu a řídicí analogový obvod. Modul je navržen tak, aby se dal přímo připojit k obvodům s pulsně šířkovou modulací/demodulací. Řízení výkonu vysílání modulu probíhá nastavením hodnot na pinech MD0, MD1 a je určeno podle tabulky 3.1.

Tabulka 3.1 – řízení výkonu infraportu

Mode 0	Mode 1	Rx funkce	Tx funkce
1	0	vypnuto	vypnuto
0	0	SIR	plný výkon
0	1	SIR	2/3 výkonu
1	1	SIR	1/3 výkonu

3.1.1 Připojení modulu infraportu k desce RC10

Deska RC10 má vyvedené vývody pro zapojení periferií. Konektor modulu infraportu k pinům FPGA na desce zapojím podle tabulky 3.2. Kromě toho jsou prostřední piny konektoru (5,6) jsou zaslepené, aby nešel modul připojit jiným než jedním způsobem.

Tabulka 3.2 – připojení pinů FPGA k RC10

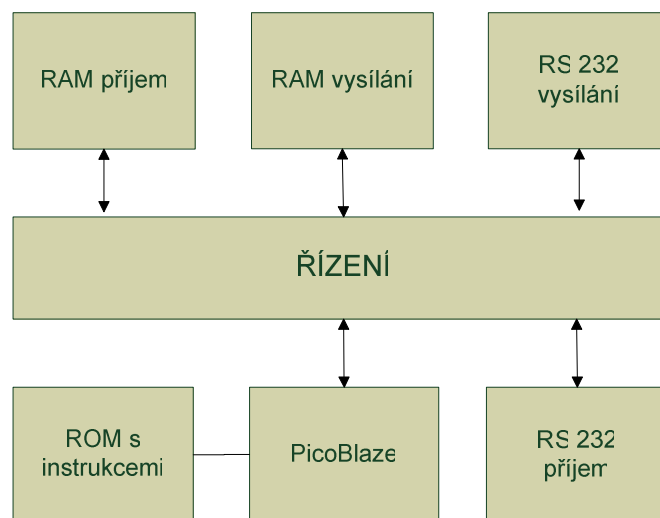
Signál	Konektor modulu	Konektor RC10	Vývod FPGA
VCC	8	44 (VCC33)	-
GND	4	40 (GND)	-
TXD	9	45 (IO33)	H17
RXD	7	43 (IO32)	H18
MD0	1	37 (nCS0)	G16
MD1	2	38 (nCS1)	G15

3.2 Architektura

VHDL umožňuje rozdělit systém, který má hierarchickou stavbu, na jednotlivé funkční bloky, které se dají snadněji naprogramovat, zřehlednit a odsimulovat.

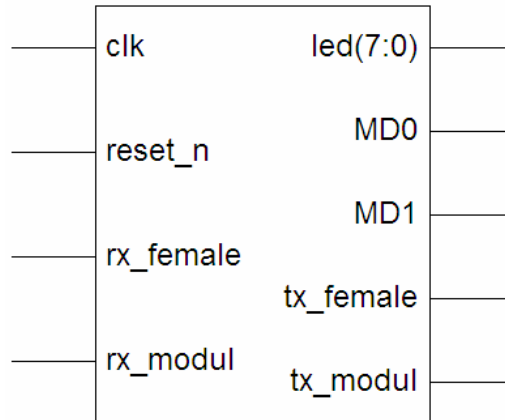
3.2.1 Spolupráce jednotlivých komponent

Základ tvoří architektura RIZENI, tj architektura, v níž jsou jednotlivé funkční bloky propojeny. Do ní je napojena komponenta RAM pro příjem a RAM pro vysílání. Dále je napojen mikrokontrolér PicoBlaze se svým programem uloženým v paměti ROM. Pro komunikaci se sériovým portem jsou připojeny komponenty. Vše je zobrazeno na obr. 3.1



Obr. 3.1 Spolupráce komponent

3.2.2 Vstupní a výstupní porty architektury RIZENI



Obr. 3.2 Entita RIZENI

Entita RIZENI je zobrazena na obr.3.2. Mezi vstupní porty patří hodiny *clk*, resetovací vstup *reset*. Výstupními porty jsou signály pro LED diody, které signalizují některé signály z programu, porty pro MD0, MD1 sloužící k nastavení režimu módu modulu jednoduchého infraportu, dvojice portů pro komunikaci se sériovou linkou počítače *tx_female*, *rx_female* a dvojice portů pro komunikaci s jednoduchým infraportem *tx_modul*, *rx_modul*.

3.2.3 Procesy

Procesy se v jazyce VHDL vykonávají sekvenčně, tedy podobně jako v jiných programovacích jazycích. Navenek se však proces chová jako paralelní příkaz. Program obsahuje procesy pro čas na synchronizaci se sériovou linkou a procesy na měření délky časových pulsů pro přijímání/vysílání.

Architektura RIZENI obsahuje základní procesy. Je v ní umístěn proces pro příjem/vysílání. Podle povolení nastaveného režimu se v procesu vykonává přijímání/vysílání dat.

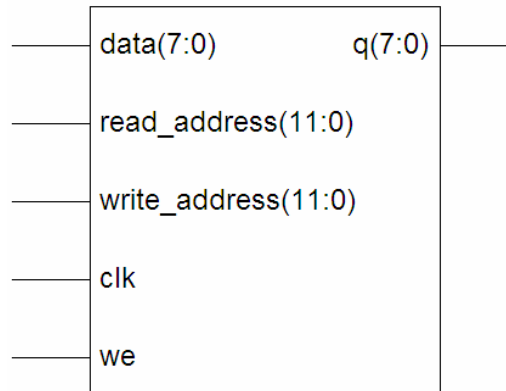
3.2.4 Signály

Mezi ovládací signály programu patří signály pro povolení, jsou to *enable_send_time* (povolení odesílání v režimu time), *enable_send_uart* (povolení odesílání v režimu uart), *enable_read_time* (povolení příjmu v režimu TIME), *enable_read_uart*

(povolení příjmu v režimu UART). Vždy musí být aktivní pouze jeden z výše uvedených signálů, protože hardware (jednoduchý infraport) nedokáže současně přijímat a vysílat.

Program obsahuje mnoho dalších pomocných signálů (pro ovládání RAM, synchronizaci a další).

3.2.5 Paměť RAM



Obr. 3.3 Entita RAM

Implementace obsahuje dvě paměti RAM (entita zobrazena na obr. 3.3). Jedna obsahuje 256 x 8 bitů pro odesílání dat a druhá má obsah 4096 x 8 bitů pro přijímání dat.

Obě mají stejnou strukturu. Zápis na adresu, která je přivedena na vstupní port *write_address* se koná, jestliže je přivedena log. 1 na vstupní port *we*. Číst data se dají vždy po nastavení adresy vstupního portu *read_address* na výstupu portu *q*.

V architektuře řízení jsou adresy nastavovány tak, aby se při příjmu po příchodu prvního pulsu data postupně zapisovala až do úplného naplnění. Při odesílání jsou odeslána jenom ta data, která byla vložena do paměti.

Všechny bloky, které spolu spolupracují musí být řízené hodinovým signálem, aby nedocházelo k hazardním stavům.

3.3 Hodinové signály

Výsledkem programování není program, ale složitý obvod složený ze jednoduchých logických obvodů, protože musíme zabránit hazardním stavům. Zabráníme tomu tak, že každý proces budeme synchronizovat společným časovým signálem. Celá implementace je

synchronně řízená hodinovým impulsem (signál *clk*) s frekvencí 48 MHz (je to rychlost, na které pracuje RC10).

Pro přijímání/odesílání dat je určen signál *clk_in/clk_out*, který je nastaven děličkou ze signálu *clk*. Přes sériovou linku nastavíme tiky signálu příkazem TIME DIVISIOR, kde DIVISIOR je hexadecimální číslo plus jedna, kterou dělička dělí signál. Například nastavíme-li hodnotu DIVISIOR na 0002. Signál *clk_in* pracuje na frekvenci 16 MHz.

V režimu UART nastavujeme děličku pro určitou rychlost odesílání v baudech (tj. jednotka udávající počet změn stavu přenosového média za jednu sekundu). Chceme-li například nastavit rychlost na 9600 baudů, nastavíme hodnotu na

$$DIVISOR = \frac{48 \cdot 10^6}{BaudRate} - 1 = \frac{48 \cdot 10^6}{9600} - 1 = 4999 = (1387)h.$$

Uvádím tabulku pro rychlosti používané při IrDA přenosu (tabulka 3.3).

Tabulka 3.3 – převod Baud Rate na DIVISOR

Baud rate	(DIVISIOR) dec	(DIVISIOR) hex
2400	19999	4E1F
4800	9999	270F
9600	4999	1387
19200	2499	09C3
38400	1249	04E1
57600	833	0341
115200	416	01A0

3.4 Režimy přenosu dat

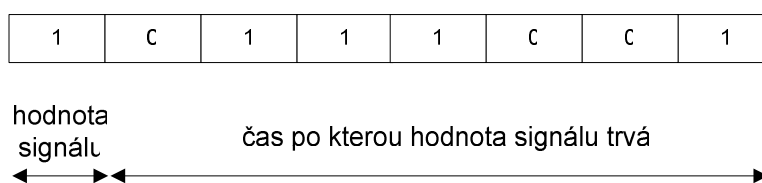
Program pro příjem IR signálu pracuje ve dvou režimech TIME a UART. V režimu TIME se jedná pouze o přenos IR a zaznamenání signálu, v režimu UART se jedná o datovou komunikaci podle normy IrDA.

3.4.1 Režim TIME

V režimu TIME program přijímá do RAM časovou změnu signálu v takovém formátu, že bit č.7 značí hodnotu signálu a úroveň signálu. Bity 0 až 6 zaznamenává počet jednotek času daný předděličkou.

Příklad:

Binární kód 10110001 znamená (viz obr. 3.3), že byla zachycena změna na log. 1 a trvala (0110001)_b = 49 časových jednotek.



Obr. 3.3 Kódování dat v režimu UART

Při odesílání nevydrží hodnota signálu infraportu na log.1 stále (pouze dobu označovanou v manuálech jako $t_{pw.}$), proto jsem zvolil jiné kódování dat. Součet hodnoty log.1 a log.0 dá příslušnou vysílanou hodnotu. Informace je ukrytá v periodě signálu.

3.4.1.1 Filtr

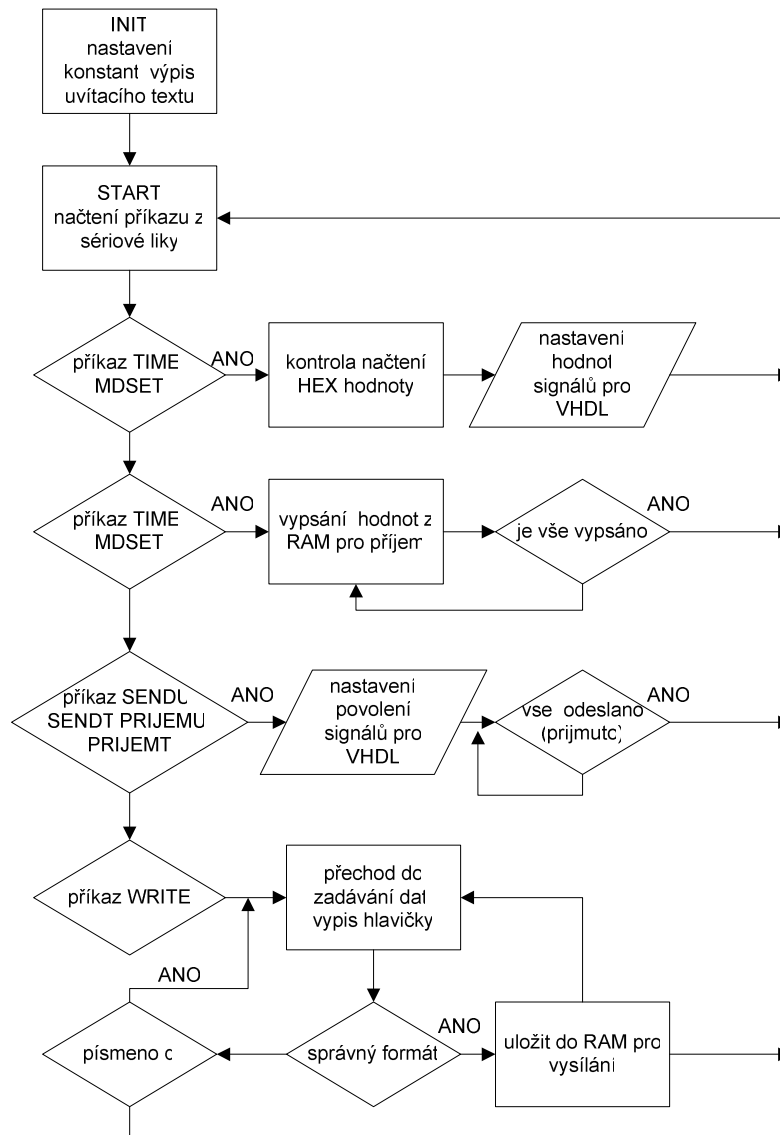
Dálkový ovladač vysílá kód na vysoké nosné frekvenci – mnoho krátkých pulsů, které nenesou informaci o stisknutém tlačítku. Pro zápis kódu by bylo potřeba velké množství paměti, které je těžko realizovatelné. Proto se mnoho krátkých pulsů považuje za log. 1. Doba mezi pulsy je připočtena k hodnotě log.1 Sériovou linkou nastavím hodnotu proměnné, o kterou se má filtrovat (příkazem FILTR).

3.4.2 Režim UART

V režimu UART se první vyšle startbit, poté data ukončená stopbitem (viz obr. 1.1 Formát dat IrDA) V tomto režimu program přijímá data do RAM pro příjem, vysílá data z RAM pro vysílání.

3.5 Picoblaze

V Picoblazu zpracovávám informace sériové linky a nastavuji jednotlivé signály (např. povolení příjmu/vysílání) a kontroluji činnost jednotlivých signálů z registru (jestli je RAM plná a data byla uložena). Zjednodušený vývojový diagram je na obrázku 3.4. Program začíná výpisem uvítacího textu a nastavením základních konstant. Pak čeká od sériové linky příkaz, který následně provede.



Obr. 3.4 Vývojový diagram programu v PicoBlaze

Příkazem tedy řídím činnost celého systému, zadávám data, která chci poslat, čtu data a nastavuji hodnoty konstant.

3.5.1 Příkazy pro sériovou komunikaci

Jednotlivé příkazy pro nastavení programu a čtení dat přes sériovou linku:

TIME xxxx – nastavení časové základny, kde x je hexadecimální číslo

MDSET xx – nastavení hodnot *MD0*, *MD1*

00 – oba jsou nulové

01 – *MD0* je ‘1’, *MD1* je ‘0’

02 – *MD0* je ‘0’, *MD1* je ‘1’

03 – *MD0* je ‘1’, *MD1* je ‘1’

WRITE – uloží data pro odesílání (ukončení režimu zadávání je písmenem q)

SENDU – odešle data z RAM pro vysílání režimem UART

SENDDT – odešle data z RAM pro vysílání režimem TIME

PRIJEMU – přijme data do RAM pro přijímání režimem UART

PRIJEMT – přijme data do RAM pro přijímání režimem TIME

READ – vypíše data z RAM pro přijímání

3.6 Dekódování signálu ovladače

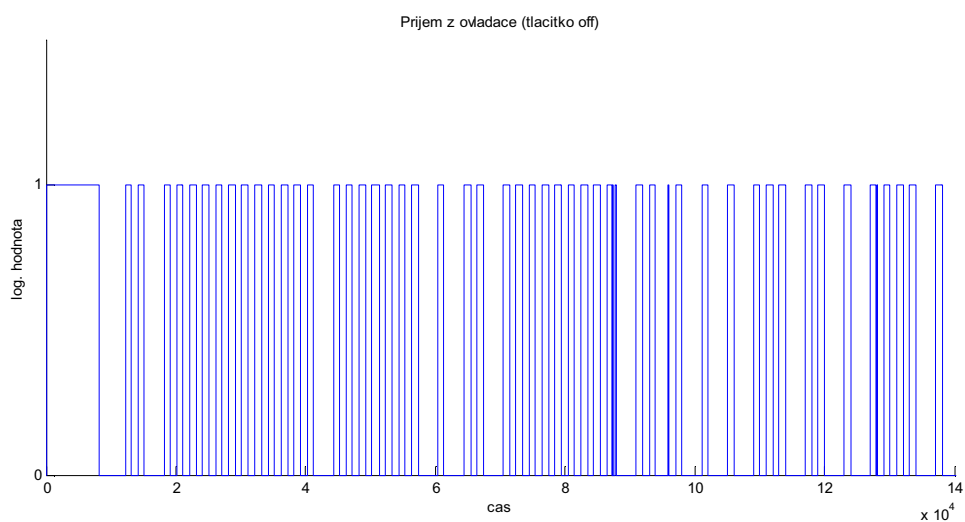
Většina firem, které vyrábí dálkové ovladače, používá svůj komunikační protokol (většinou lehce pozměněný RC-5), proto kdybychom chtěli rozpoznat jednotlivá tlačítka, je lepší zapamatovat si celý signál a ten pak vyhodnotit než dekódovat pomocí protokolu.

Pro příjem jsem použil dálkový ovladač PANASONIC k DVD rekordéru. Při stisku tlačítka **1**, **2**, **OFF** byla sejmuta data, která jsou také uložena na přiloženém CD (pro výběr dat z RAM přes sériovou linku slouží příkaz READ).

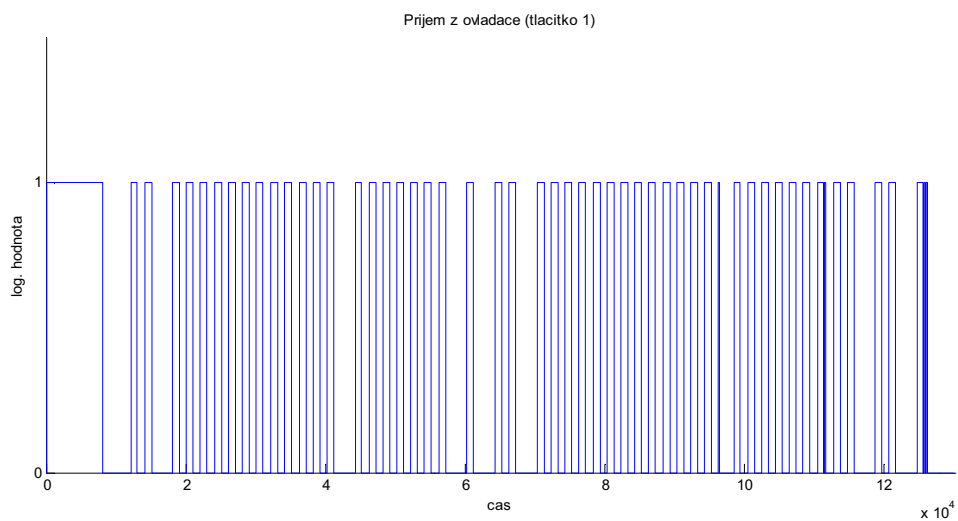
Pro vykreslení grafů byl vytvořen dekódovací m-file pro matlab (uložený na CD pod názvem *decode.m*). Jako vstup tohoto programu jsou data v zakódované formě uložena v textovém souboru *data.txt* stejném adresáři jako m-file.

Dělicí poměr DIVISOR byl nastaven na hodnotu (0014)hex = (20)dec. Jednotka grafu má rozměr

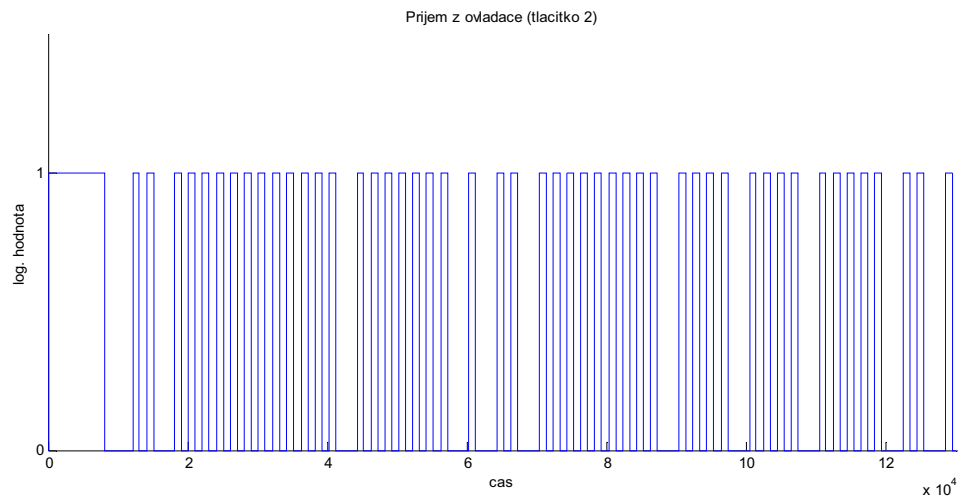
$$t_1 = \frac{DIVISOR + 1}{48 \cdot 10^6} = \frac{20 + 1}{48 \cdot 10^6} = 0,44 \mu s .$$



Obr. 3.5 Přijem ovladače při stisku tlačítka OFF



Obr. 3.6 Přijem ovladače při stisku tlačítka 1



Obr.3.7 Příjem ovladače při stisku tlačítka 2

Po vykreslení dat dostáváme časové průběhy zobrazené na obr. 3.5, obr. 3.6, obr. 3.7. Z naměřených grafů vidíme, že jednotlivá tlačítka se liší tvarem signálu. Signál je uvozen jedním startbitem, nejedná se proto o nejpoužívanější kódování RC-5.

Na internetu jsem nenašel žádné informace o kódování, které využívá tento dálkový ovladač, našel jsem pouze amatérské aplikace, které naměřily stejný průběh. Vykreslené průběhy stisknutých tlačítek byly ověřeny na osciloskopu.

3.7 Obsazenost hradlového pole

Na obrázku 3.8 můžeme vidět implementovaný návrh tak, jak je zobrazen v panelu Floorpanel. Jednotlivé bloky od sebe nelze rozlišit, protože splývají. Umístování jednotlivých elementárních funkcí po čipu je bez ohledu na hierarchii bloků.

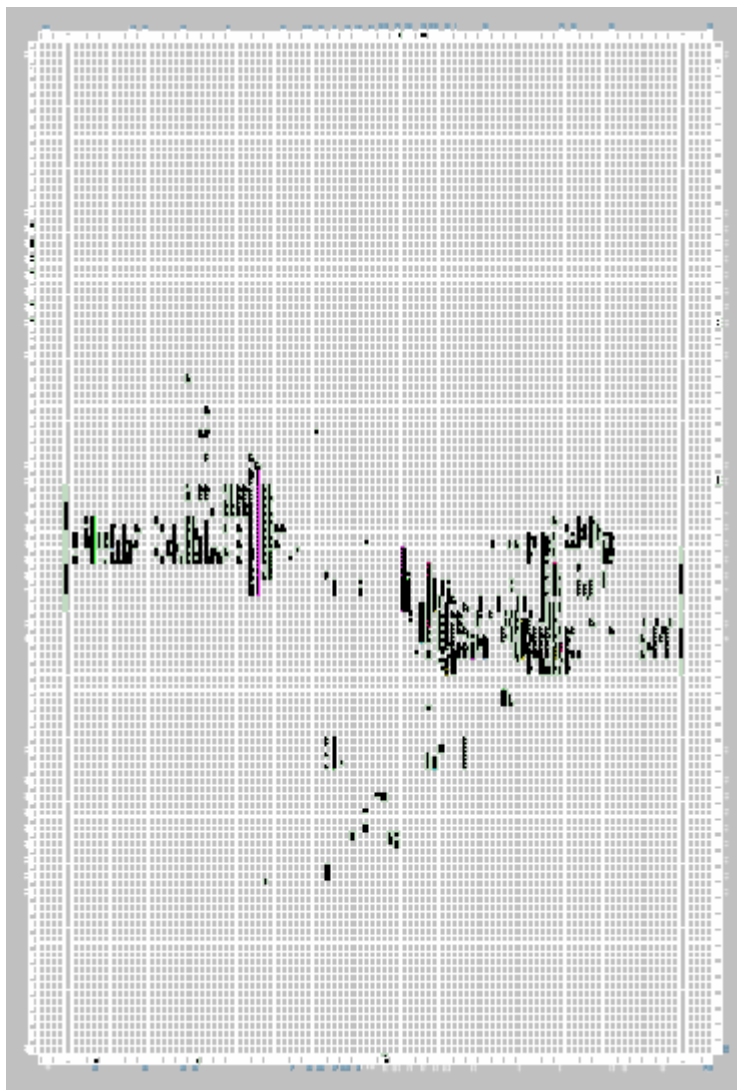
V tabulce 3.4 je shrnuto využití jednotlivých „prvků“ v FPGA.

Tabulka 3.4 Obsazenost hradlového pole

Element	Využito	K dispozici	Obsazenost
Slice Flip Flops	380	26624	1%
4 input LUTs	601	26624	2%
Obsazený Slice	466	13312	3%

Dále implementace obsahuje celkový počet čtyřvstupých LUTů je 790 z 26624 (tj. 2%). Počet logických členů 601, propojek 85, Dual Port RAM 85, 32x1 RAM 52, posuv-

ných registrů 36, IOBs 18 z 221 (tj. 8%), RAMs Block 4 z 32 (tj. 12%), GClks 1 z 8 (tj. 12%). Celkový počet elementů je **280156**. Jelikož to byla moje první velká aplikace psaná ve VHDL, vůbec jsem se nezabýval optimalizací, kterou lze obsazenost hradlového pole výrazně snížit.



Obr.3.8 Obsazenost FPGA

ZÁVĚR

Seznámil jsem se s přenosem dat pomocí IR komunikace, jazykem VHDL a instrukcemi mikrokontroléru. Vytvořil jsem systém pro připojení modulu infraportu a ovládní tak, že dokáže přijímat a vysílat data. Řešení pro přesun dat pomocí IR komunikace jsem rozdělil na dva režimy kódování. První režim TIME je spíše časovým záznamem dat než kódování. Druhý režim UART slouží pro datovou komunikaci pomocí IrDA.

Funkčnost jsem ověřil na osciloskopu a přenosem dat mezi dvěma stejnými implementacemi.

Zjistil jsem, že použitý ovladač PANASONIC k DVD rekordéru neodpovídá žádnému často používanému kódování pro dálkové ovladače.

Práce může sloužit kromě přenosu dat i jako začátek tvorby pro vytvoření učícího se dálkového ovladače. Po zaznamenání kódu dálkového ovladače se musí vytvořit vysílání zaznamenaného kódu.

Celý systém je popsán jazykem VHDL a vytvořen v návrhovém systému XILINX ISE 9.1.02i

Pro data z režimu TIME jsem vytvořil m-file na zobrazení přijatých informací do grafu.

Na přiloženém CD je kompletní návrh společně s katalogovými listy součástek, m-file a všechny dokumenty, které jsem při návrhu použil.

POUŽITÁ LITERATURA

[1] *Normy pro IR datový přenos* <cit 4. Březen 1999>

URL:<<http://hw.cz/Teorie-a-praxe/Dokumentace/ART753-Normy-pro-IR-datovy-prenos.html>>

[2] Vladimír Myslík, *IrDa - Kompletní popis*, <cit. 13. Říjen 1998>

URL:<<http://hw.cz/Teorie-a-praxe/Dokumentace/ART784-IrDa---Kompletni-popis.html>>

[3] Marek Zouzalík, *Letem světem bez drátů* <cit 21. Leden 2005>

URL:<<http://www.21stoleti.cz/view.php?cisloclanku=2005012111>>

[4] Pinker Jiří, Poupa Martin, *Číslicové systémy a jazyk VHDL*, nakladatelství BEN

[5] Jan Babčaník, *Jednoduchý infraport* <cit 7. Leden 2007 >

URL:<<http://hw.cz/Teorie-a-praxe/Konstrukce/ART1793-Jednoduchy-infraport.html>>

[6] *Přednášky, ukázkové programy* <cit 29. Listopad 2006>

URL:<<http://dce.felk.cvut.cz/msy/files/fpga/>>

PŘÍLOHY

Příloha A - jednoduchý infraport

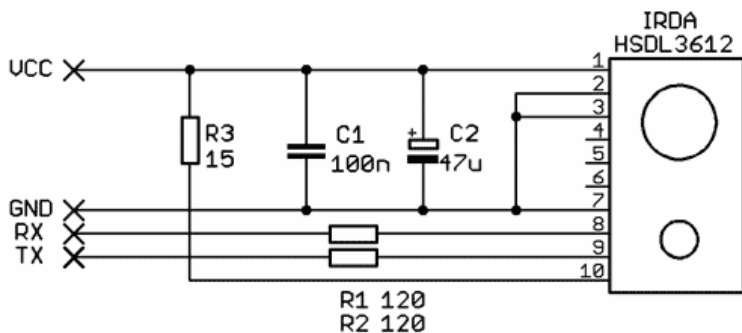
Charakteristika

- nízká cena IR datového spojení
- konstrukce ve standardu IrDA
- pracovní vzdálenost 0.01 – 1m
- přenosové rychlost 2,4 – 115,2kbit
- potlačení rušení denním světlem

Parametry

- vlnová délka 875/880nm
- napájecí napětí 2,7 – 5,2V
- spotřeba 2,5 – 5mA
- vyzařovací úhel 30°/60°
- pouzdro A3612
- pracovní teplota -20°C až +70°C

Schéma zapojení



9	7	5	3	1
10	8	6	4	2

Číslování pinů konektor:

Signály na modulu

Signál	Konektor modulu	Funkce
VCC	8	napájení
GND	4	uzemnění
TXD	9	vysílání IR
RXD	7	příjem IR
MD0	1	řízení výkonu odesílání
MD1	2	řízení výkonu odesílání

Piny č. 5,6 jsou zaslepené.

Manuál pro integrovaný obvod je obsazen na příloženém CD.

Příloha B – obsah přiloženého CD**Složky**

manual/ – manuály pro RC10, HSDL3612, Picoblaze

project/ – celý projekt vytvořený v XILINX ISE 9.1.02i se všemi zdrojovými kódy

bitstream/ – soubor pro nahrání zařízení do fpga, jednoduchý popis programu

text/ – bakalářská práce ve formátu doc a pdf

data/ – naměřená data z dálkového ovladače, grafy

Soubory

index.html podrobný popis umístění dat na CD