

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Sekvenčné a kombinačné štruktúry na báze obvodov Altera CPLD

Miroslav ŠULÍK

DIPLOMOVÁ PRÁCA

2008

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Katedra elektroniky a multimediálnych telekomunikácií

**Sekvenčné a kombinačné štruktúry na báze obvodov Altera
CPLD**

DIPLOMOVÁ PRÁCA

Miroslav Šulík

Vedúci diplomovej práce:

Doc. Ing. Miloš Drutarovský, CSc.

Konzultant diplomovej práce:

Doc. Ing. Miloš Drutarovský, CSc.

Košice 2008

Analytický list

Autor:	Miroslav Šulík
Názov práce:	Sekvenčné a kombinačné štruktúry na báze obvodov Altera CPLD
Jazyk práce:	slovenský
Typ práce:	Diplomová práca
Počet strán:	83
Akademický titul:	Inžinier
Univerzita:	Technická univerzita v Košiciach
Fakulta:	Fakulta elektrotechniky a informatiky (FEI)
Katedra:	Katedra elektroniky a multimediálnych telekomunikácií (KEMT)
Študijný odbor:	Elektronika a telekomunikačná technika
Mesto:	Košice
Vedúci DP:	Doc. Ing. Miloš Drutarovský, CSc.
Konzultanti DP:	Doc. Ing. Miloš Drutarovský, CSc.
Dátum odovzdania:	5. máj 2008
Dátum obhajoby:	5. jún 2008
Kľúčové slová:	Sekvenčné obvody, Kombinačné obvody, CPLD, Altera, Quartus
Kategória Konspekt:	Technika, technológia, inžinierstvo; Elektronika
Citovanie práce:	Šulík, Miroslav: Sekvenčné a kombinačné štruktúry na báze obvodov Altera CPLD. Diplomová práca. Košice: Technická univerzita v Košiciach, Fakulta elektrotechniky a informatiky, 2008. 83 s.
Názov práce v AJ:	Sequential and combination structures based on Altera CPLD circuits
Kľúčové slová v AJ:	Sequential circuit, combination circuit, CPLD, Altera, Quartus

Abstrakt v SJ

Dokument v teoretickej časti obsahuje opis možnosti konfigurovania FPGA a CPLD obvodov a špecializovaných pamätí firmy Altera. Opis je rozdelení na tri časti, kde prvá časť opisuje konfiguračné metódy pre jednotlivé rodiny obvodov Altera. Druhá časť opisuje používané konfiguračné formáty a v tretej časti sa opisuje konfiguračný hardvér, pričom sa podrobnejšie opisuje USB Blaster Download Cable. V praktickej časti sa opisuje vyvinuté zariadenie a rozoberajú sa jednotlivé časti konštrukcie. V tejto časti sa nachádza i prúdová analýza, ktorá je pre danú úlohu dôležitá. V predposlednej kapitole sú uvedené testovacie príklady realizovaného prípravku. Jednotlivé príklady sú riešené v prostredí Altera Quartus II pre grafický i VHDL režim prostredia s kompletnou časovou a funkčnou simuláciou.

Abstrakt v AJ

The document contains the description of possibilities in configuration of FPGA and CPLD circuits and specialized Altera memories in theoretical part. The description is divided in to the three parts. First part describes configuration methods for the families of Altera circuits. The second part describes configuration formats that were used. Configurable hardware with focus on USB Blaster Download Cable is described in the third part. Development of the device and the single part of construction are described in practical part. In this part current analysis that is important for the task can be found. Tested examples of the device are presented in the penult part. The examples are solved in an Altera Quartus II environment for graphical and VHDL mod with the complete functional and time simulation.

TECHNICKÁ UNIVERZITA V KOŠICIACH

Fakulta elektrotechniky a informatiky

Katedra elektroniky a multimediálnych telekomunikácií
akademický rok: 2007/2008

**ZADANIE
DIPLOMOVEJ PRÁCE**

pre: **Miroslav Š u l í k**

Odbor: Elektronika a telekomunikačná technika

Študijný program:

Vzhľadom k tomu, že ste splnili požiadavky učebného plánu, zadáva Vám dekan fakulty na návrh vedúceho vedecko-pedagogického pracoviska v zmysle zákona o VŠ č.131/2002 Z.z a Študijného poriadku TU §15, ods. 3, túto tému záverečnej práce:

**Sekvenčné a kombinačné štruktúry na báze obvodov
Altera CPLD**

POKYNY PRE VYPRACOVANIE

Osnova práce:

Navrhnete a technicky zrealizujete prípravok na precvičovanie základných kombinačných a sekvenčných obvodov pomocou obvodu Altera EPM3064A. Navrhnete ďalšie komponenty prípravku tak, aby ho bolo možné použiť s minimom externých zariadení (napájacích adaptérov a pod.). Ako konfiguračné rozhranie použijete USB JTAG adaptér vyvinutý na KEMT FEI TU v Košiciach. Pre navrhnutý prípravok vytvorte demonštračné príklady na otestovanie vybraných kombinačných a sekvenčných obvodov v grafickom aj VHDL režime prostredia Altera Quartus II a kompletnou funkčnou a časovou simuláciou s cieľom ich využitia vo výučbe špecializovaných predmetov. V teoretickej časti práce opíšte základné možnosti využitia USB adaptéra pre konfigurovanie Altera FPGA a CPLD obvodov a špecializovaných sériových pamätí. Tieto módy otestujte na vybraných konfiguráciách vývojových dosiek dostupných na KEMT FEI.

Rozsah laboratórných a grafických prác : podľa potreby

Rozsah záverečnej práce: odporúčaný počet strán 40 a viac

Zoznam odporúčanej literatúry:

1. <http://www.altera.com>

Vedúci záverečnej práce: doc. Ing. Miloš Drutarovský, CSc.

Konzultant: doc. Ing. Miloš Drutarovský, CSc.

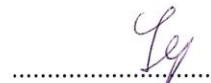
Dátum zadania záverečnej práce: 3.4.2008

Dátum odovzdania záverečnej práce: 5.5.2008

V Košiciach, dňa 3.4.2008



prof. Ing. Liberios Vokorokos, PhD.
dekan fakulty



prof. Ing. Dušan Levický, CSc.
vedúci vedecko-pedagogického pracoviska

Čestné vyhlásenie

Čestne vyhlasujem, že som celú záverečnú prácu vypracoval samostatne a literatúru, ktorú som použil, uvádzam v zozname.

Ďalej vyhlasujem, že nemám námietky proti zapožičaniu a zverejneniu mojej záverečnej práce alebo jej časti so súhlasom katedry.

Košice, 5. máj 2008

.....

vlastnoručný podpis

Pod'akovanie

Ďakujem Doc. Ing. Milošovi Drutarovskému, CSc., vedúcemu diplomovej práce za jeho cenné a podnetné rady. Taktiež ďakujem firme Elcom spol. s r.o., Prešov za poskytnutie obvodov Altera CPLD EPM3064A.

Obsah

Zoznam obrázkov	10
Zoznam tabuliek	12
Zoznam symbolov a skratiek	13
Úvod	15
1 Programovanie a konfigurovanie CPLD, FPGA obvodov	
a špecializovaných sériových pamätí firmy Altera	16
1.1 Konfigurovanie rôznych rodín obvodov Altera	18
1.2 Konfiguračné formáty	21
1.3 Konfiguračný hardver.....	22
2 Univerzálna vývojová doska CPLD start kit	24
2.1 Opis zariadenia	24
2.2 Opis konštrukcie	27
2.2.1 Napájacia časť.....	27
2.2.2 Obvod CPLD EMP3064ATC44-10.....	29
2.2.3 Konfiguračné rozhranie JTAG ISP (konektor)	31
2.2.4 Generátor hodinových impulzov.....	33
2.2.5 Tlačidlá, prepínače	34
2.2.6 Bzučiak	35
2.2.7 LED diódy, dvoj číslicový sedem-segmentový displej.....	36
2.3 Prúdová analýza zapojenia	38
2.4 Zapojenie prepojek	38
3 Testovacie projekty zariadenia	40
3.1 Prevodník z BCD na sedem-segment	40
3.1.1 Zadanie úlohy.....	40
3.1.2 Riešenie.....	40
3.1.3 Syntéza	41
3.1.4 Schematické zapojenie obvodu.....	42
3.1.5 Riešenie pomocou VHDL.....	45
3.1.6 Výsledok úlohy	48
3.2 Demultiplexor.....	49
3.2.1 Zadanie úlohy.....	49
3.2.2 Riešenie.....	49

3.2.3	Syntéza	49
3.2.4	Zapojenie obvodu.....	50
3.2.5	Riešenie pomocou VHDL.....	52
3.2.6	Výsledok úlohy	53
3.3	Multiplexor	53
3.3.1	Zadanie úlohy.....	54
3.3.2	Riešenie.....	54
3.3.3	Syntéza	54
3.3.4	Riešenie pomocou VHDL.....	56
3.3.5	Výsledok úlohy	58
4	Záver.....	59
	Zoznam použitej literatúry	60
	Prílohy.....	62

Zoznam obrázkov

Obr. 1 Typická štruktúra obvodov PAL	17
Obr. 2 Typická štruktúra FPGA obvodu.....	17
Obr. 3 Konfiguračný hardvér USB Blaster Download Cable firmy Altera.....	22
Obr. 4 Bloková schéma konfiguračného zariadenia USB Blaster	23
Obr. 5 Bloková schéma CPLD start kit	24
Obr. 6 Opis jednotlivých elementov prototypovej dosky CPLD start kit.....	25
Obr. 7 Zapojenie jednotlivých elementov profesionálnej dosky CPLD start kit.....	26
Obr. 8 Napájací kábel prípravku CPLD start kit	27
Obr. 9 Napájacia časť prípravku CPLD start kit.....	28
Obr. 10 Bloková schéma rodiny MAX výrobcu Altera.....	29
Obr. 11 Zapojenie obvodu CPLD EPM3064 na CPLD start kite	30
Obr. 12 Zapojenie konfiguračného rozhrania JTAG	32
Obr. 13 Zapojenia oscilátora a) a prepojky JP4 na pripojenie k GCLK2	33
Obr. 14 Rozdelenie jednotlivých prepínačov pre DIP prepínač	34
Obr. 15 Zobrazenie zapojenie tlačidiel a) a DIP prepínača b) na prípravku CPLD start kite.....	35
Obr. 16 Spôsob zapojenia bzučiaka na CPLD start kit.....	36
Obr. 17 Rozmiestnenie DG1 a DG2 na sedem-segmentovom displeji.....	37
Obr. 18 Zapojenie LED a sedem-segmentového displeja na CPLD start kit	37
Obr. 19 Význam zapojenia prepojky JP 1	39
Obr. 20 Zobrazenie zapojenej a nezapojenej prepojky.....	39
Obr. 21 Karnaughove mapy pre prevodník BCD na sedem-segment.....	42
Obr. 22 Priradenie príslušných pinov v projekt prevodník.....	42
Obr. 23 Schéma zapojenia prevodníka BCD na sedem-segment	43
Obr. 24 Výsledok funkčnej simulácie pre zapojenie prevodníka BCD na sedem- segment	44
Obr. 25 Výsledok časovej simulácie pre prevodník BCD na sedem-segment	45
Obr. 26 Zapojenie prevodníka z BCD na sedem-segment vytvoreného vo VHDL	46
Obr. 27 Výsledok funkčnej simulácie prevodníka z BCD na sedem-segment vytvoreného vo VHDL	47
Obr. 28 Výsledok časovej simulácie prevodníka z BCD na sedem-segment vytvoreného vo VHDL	48
Obr. 29 Schematické zapojenie demultiplexra	51
Obr. 30 Priradenie pinov v projekte demultiplexor	51

Obr. 31 Zapojenie demultiplexra vytvoreného vo VHDL	52
Obr. 32 Zapojenie vývodov demultiplexra riešeného pomocou VHDL k jednotlivým pinom CPLD obvodu.....	53
Obr. 33 Bloková schéma zapojenia multiplexora	55
Obr. 34 Zapojenie Multiplexora	55
Obr. 35 Zapojenie vstupov a výstupov multiplexra k jednotlivými vývodmi CPLD obvodu.	56
Obr. 36 Zapojenie multiplexra pre simulácie	56
Obr. 37 Zapojenie multiplexra riešeného pomocou jazyka VHDL.	57
Obr. 38 Upravené zapojenie multiplexra riešeného pomocou VHDL pre simuláciu	57
Obr. 39 Zapojenie vstupov a výstupov multiplexra k jednotlivým vývodom CPLD obvodu	57

Zoznam tabuliek

Tab. 1 Využitie jednotlivých metód konfigurovania rodín obvodov Altera.....	19
Tab. 2 Typické využitie konfiguračných metód konfiguračnými zariadeniami.....	19
Tab. 3 Odporúčaný počet zariadení pre n-Bitový konfiguračný PS mód.....	20
Tab. 4 Napät'ové úrovne vstupov a výstupov pri zmene napätia na VCCIO	30
Tab. 5 Zapojenie pinov CPLD obvodu a konektora JTAG rozhrania	33
Tab. 6 Priradenie pinov CPLD obvodu s oscilátorom a JP4	34
Tab. 7 Priradenie pinov CPLD obvodu tlačidlám a DIP prepínačom	35
Tab. 8 Priradenie pinov CPLD obvodu bzučiaku	36
Tab. 9 Priradenie pinov CPLD obvodu pre LED diódy a obe čísla sedem- segmentového displeja.....	37
Tab. 10 Spotreba jednotlivých komponentov a celková spotreba zariadenia.....	38
Tab. 11 Pravdivostná tabuľka prevodníka z BCD na sedem-segment	41
Tab. 12 Pravdivostná tabuľka demultiplexra.....	50
Tab. 13 Pravdivostná tabuľka 4-vstupového multiplexora.....	54

Zoznam symbolov a skratiek

AS	Active Serial, aktívny sériový
ASIC	Application Specific Integrated Circuit, zákazkový integrovaný obvod
BCD	Binary Coded Decimal, binárne kódovanie dekadických čísel
CMOS	Complementary Metal Oxide Semiconductor, dopovaní metal oxidový polovodič
CPLD	Cell Programmable Logical Device, programovateľný logický obvod využívajúci bunkovú architektúru
EEPROM	Electrically Erasable Programmable Read Only Memory, programovateľná pamäť s elektricky mazateľným obsahom
FPGA	Field Programmable Gate Array, programovateľné logické polia
IOB	Input/Output Blok, vstupno/výstupný blok
ISP	In System Programming, programovanie v cieľovom systéme
JTAG	Joint European Test Action Group, skupina, ktorá vytvorila testovací štandard IEEE 1149.1-1990
LB	Logic Block, logický blok
LED	Light-Emitting Diode, dióda vyžarujúca svetlo
MAX	Multiple Array matriX, viac rozmerná matica
nTRST	Test ReSeT, testovací reset
PAL	Programmable Array Logic, (obvod pozostávajúci z programovateľného poľa AND pevne pripojené k OR
PPA	Passive Parallel Asynchronous, pasívny paralelný asynchrónny
PPS	Passive Parallel Synchronous, pasívny paralelný synchrónny
PS	Passive Serial, pasívny sériový
PSA	Passive Parallel Synchronous, pasívny paralelný synchrónny
TAP	Test Access Port, prístupový port testov
TCK	Test ClOcK, testovacie hodiny
TDI	Test Data In, vstupné testovacie dáta

TDO	T est D ata O ut, výstup testovacích dát
TMS	T est M ode S elect I nput, výber režimu testu
USB	U niversal S ynchronous B us, univerzálna synchronná zbernica
VHDL	V ery H igh I ntegrated C ircuit D escription L anguage, jazyk na programovanie obvodov veľmi vysokej integrácie
UDNF	Úplná D isjunktívna N ormálna F orma
Ω	O hm, jednotka elektrického odporu

Úvod

Časy keď v elektronike prekvitala analógová technika sú dávno preč. V dnešnej dobe je svet založený na digitálnej technike. Analógové signály sa podľa možností čo najskôr digitalizujú. Hlavnou výhodou číslicových signálov je nadobudnutie iba konečného počtu hodnôt. Tým sa líšia od analógových signálov, ktoré nadobúdajú nekonečný počet hodnôt, sú spojité.

Tento rozdiel je podstatný pre zníženie nárokov na presnosť obvodov a tým aj na ich cenu. Navyše číslicové obvody majú výhodu v spracovaní informácii v rôznej forme (text, čísla, obrázky) a sú necitlivé na zmenu parametrov svojich súčiastok, hlavne pri kolísaní teploty.

Aj tieto výhody prispeli k tomu, že sa číslicové obvody objavujú skoro všade. Od jednoduchých malých zariadení ako je blikajúce svetielko na bicykli cez domáce spotrebiče až po veľmi zložitú počítače a laboratórne prístroje.

Typov číslicových obvodov je nespočetné množstvo. Každý typ obvodu sa špecializuje na inú oblasť elektrotechniky. Práca je zameraná na zostrojenie prípravku osadeného CPLD obvodom Altera EPM3064A. CPLD obvody sú zlatá stredná cesta medzi programovateľnými logickými obvodmi, pretože sú zložitejšie ako GAL obvody no jednoduchšie ako FPGA obvody. Princíp programovania GAL, CPLD a FPGA obvodov je veľmi podobný a preto je CPLD obvod EPM3064A veľmi dobrá voľba pre začiatočníkov.

Pre programovanie daných obvodov sa využíva vývojové prostredie Quartus II firmy Altera. V tejto práci sa vývojové prostredie Quartus II využíva na vytvorenie testovacích projektov pre navrhovaný prípravok. Testovacie projekty sú vytvorené v grafickom aj VHDL režime s funkčnými a časovými simuláciami. Uvedené projekty zobrazujú výhodu využívania opisného jazyka VHDL pre riešenie úloh.

1 Programovanie a konfigurovanie CPLD, FPGA obvodov a špecializovaných sériových pamätí firmy Altera

Preberaná problematika bude demonštrovaná na vybraných produktoch firmy ALTERA [9]. Aplikované princípy sú však využité aj inými výrobcami FPGA (Field Programmable Gate Array) obvodov.

Firma Altera patrí medzi popredných výrobcov CPLD (Complex Programmable Logic Device), FPGA a ASIC (Application Specific Integrated Circuit) obvodov. Jej ponuka by sa dala rozdeliť do štyroch základných skupín [11].

CPLD obvody

Lacné FPGA obvody – poskytujúce malý výkon

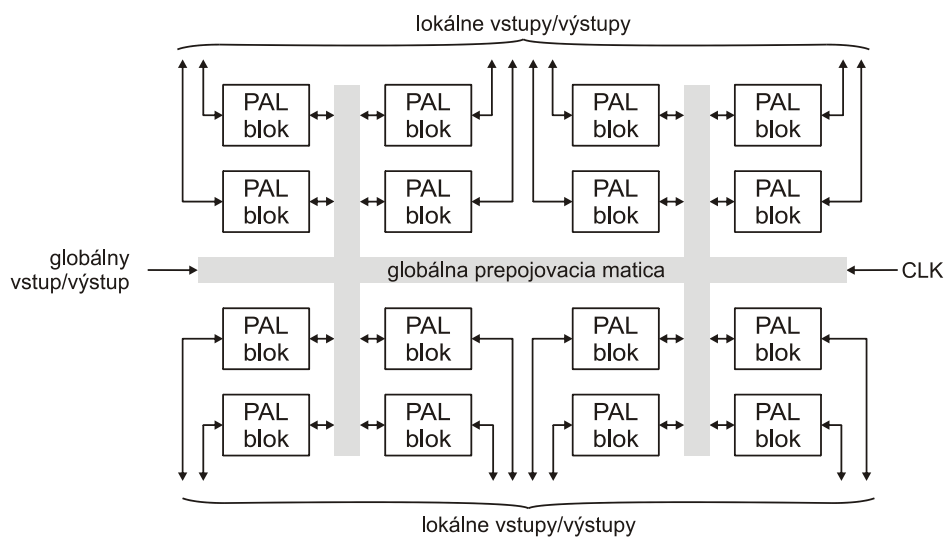
Drahé FPGA obvody – poskytujúce veľký výkon

Štruktúrované ASIC obvody

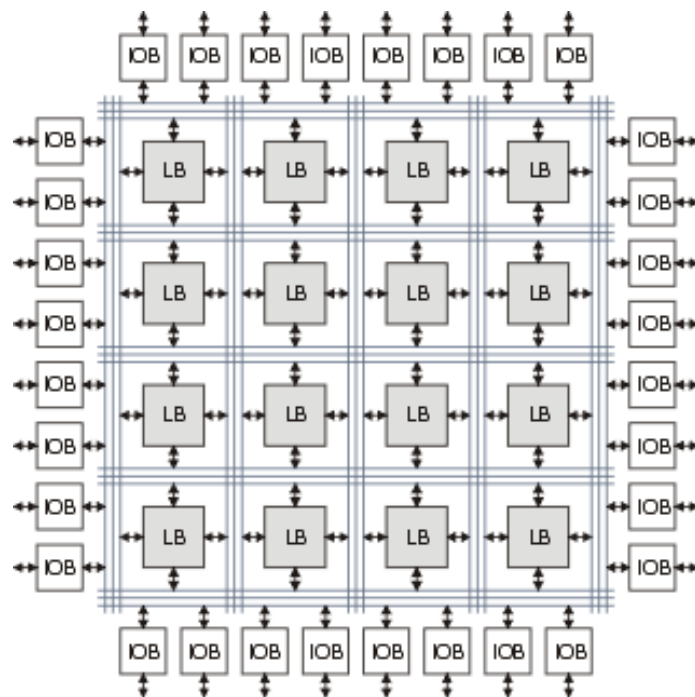
Rozdiely medzi CPLD a FPGA obvodmi vychádzajú z ich štruktúry. Keďže CPLD majú štruktúru blízku PAL (Programmable Array Logic), sú vhodné na konštrukciu kombinatorickej logiky s mnohými vstupmi a s dobre odhadnuteľnými oneskoreniami (vďaka „priamočiarej“ štruktúre). Nie sú však vhodné pre rozsiahlu sekvenčnú logiku, keďže každý klopný obvod zaberie jednu makrobunku, ktorých je obmedzený počet. Naopak, FPGA nie sú vhodné pre rozsiahlu kombinatorickú logiku, keďže signály by museli prechádzať veľkým množstvom buniek. Naopak, aj zložitá sekvenčná logika sa implementuje pomerne jednoducho. Keďže väčšina aplikácií si vyžaduje práve sekvenčné obvody a nie kombinatorické, na obvody FPGA sa kladie väčší dôraz a sú vyrábané aj v mimoriadne veľkých a rýchlych verziách.

Obvody CPLD – sú obvody založené na CMOS (Complementary metal-oxide semiconductor) technológií. Vyvinuli sa z PLD (Programmable Logic Device) obvodov, ktoré majú obmedzené vlastnosti. Výrobcovia preto začali spájať viacej takýchto obvodov na jednom čipe spoločne s potrebnými prostriedkami ich prepojenia. Každý výrobca využíva trochu inú štruktúru obvodov. CPLD od rôznych výrobcov sa obvykle líšia v realizácii blokov vlastnej programovateľnej logiky aj keď väčšinou vychádzajú z klasickej štruktúry PAL zobrazenej na Obr. 1.

Obvody FPGA – majú z programovateľných obvodov najvšeobecnejšiu štruktúru a obsahujú najviac logiky. Súčasné najväčšie FPGA obvody obsahujú až 10-ky miliónov ekvivalentných hradiel (typické dvojevstupové hradlo NAND). Typickú štruktúru obvodu FPGA znázorňuje Obr. 2. Bloky označené IOB (Input/Output Block) predstavujú vstupno-výstupné obvody pre každý vstupno-výstupný vývod FPGA. Tieto bloky obvykle obsahujú register, budič, multiplexor a ochranné obvody. Bloky LB (Logic Block) predstavujú vlastné programovateľné logické bloky.



Obr. 1 Typická štruktúra obvodov PAL



Obr. 2 Typická štruktúra FPGA obvodu

1.1 Konfigurovanie rôznych rodín obvodov Altera

Jednotlivé rodiny obvodov Altera [9] sa môžu konfigurovať týmito základnými metódami.

- Aktívna (FPGA generuje riadiace a synchronizačné signály)
- Pasívna (riadiace signály generuje konfiguračné zariadenie)
- JTAG (Joint Test Action Group)

Tieto základné metódy sú rozdelené nasledovne [1]

- Active serial (AS), aktívnu sériovú – konfiguračné dáta sú prenášané po jednom bite počas jedného hodinového cyklu
- Passive Serial (PS), pasívnu sériovú – konfiguračné dáta sú prenášané po jednom bite počas jedného hodinového cyklu
- Passive Parallel Synchronous (PPS), pasívnu paralelnú synchronnú – konfiguračné dáta o veľkosti 8-bitov sa na prvú nábežnú hranu signálu DCLK zachytia v pamäti FPGA a následne je potrebných osem dobežných hrán k prevedeniu paralelných dát na sériové.
- Fast Passive Parallel (FPP), rýchlu pasívnu paralelnú – konfiguračné dáta sú prenášané jeden bajt počas jedného hodinového cyklu
- Passive Parallel Asynchronous (PPA), pasívnu paralelnú asynchrónnu – prenos ôsmich dátových signálov riadia asynchrónne riadiace signály
- Passive Serial Asynchronous (PSA), pasívnu sériovú asynchrónnu – prenos jedného bitu riadia asynchrónne signály
- JTAG – konfiguračné dáta sú prenášané sériovým JTAG rozhraním

Využitie jednotlivých metód konfigurovania jednotlivých rodín obvodov je opísaný v Tab. 1.

Typické využitie konfiguračných metód rôznymi zariadeniami zobrazuje Tab. 2.

Tab. 1 Využitie jednotlivých metód konfigurovania rodín obvodov Altera

Configuration Scheme	Device Family									
	Stratix II	Stratix, Stratix GX	Cyclone II	Cyclone	APEX II	APEX 20K, APEX 20KE, APEX 20KC	Mercury	ACEX 1K	FLEX 10K, FLEX 10KE, FLEX 10KA	FLEX 6000
Passive Serial (PS)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Active Serial (AS)	✓		✓	✓						
Fast Passive Parallel (FPP)	✓	✓			✓					
Passive Parallel Synchronous (PPS)						✓	✓	✓	✓	
Passive Parallel Asynchronous (PPA)	✓	✓			✓	✓	✓	✓	✓	
Passive Serial Asynchronous (PSA)										✓
Joint Test Action Group (JTAG)	✓	✓	✓	✓	✓	✓	✓	✓	✓	(1)

Tab. 2 Typické využitie konfiguračných metód konfiguračnými zariadeniami

Konfiguračná metóda	Typické využitie
Passive Serial (PS)	Konfigurácia s vylepšenými konfiguračnými zariadeniami (EPC16, EPC8 a EPC4), EPC2, EPC1, EPC1441, sériovým asynchrónnym rozhraním mikroprocesora, USB Blaster USB Port download cable, Master Blaster communication cable, ByteBlasterII or ByteBlasterMV
Active serial (AS)	Konfigurácia so sériovými konfiguračnými zariadeniami (EPCS1 a EPCS4)
Passive Parallel Synchronous(PPS)	Konfigurácia sa paralelným synchrónnym rozhraním procesora
Fast Passive Parallel (FPP)	Konfigurácia s vylepšeným konfiguračným zariadením alebo paralelným synchrónnym rozhraním procesora kde 8-bitov z konfiguračných dát je načítaných pri každom hodinovom cykle, 8-krát rýchlejší ako PPS
Passive Parallel Asynchronous (PPA)	Konfigurovanie pomocou paralelného asynchrónneho rozhrania mikroprocesora
Passive Serial Asynchronous (PSA)	Konfigurovanie pomocou sériového asynchrónneho rozhrania mikroprocesora
Join Test Action Group (JTAG)	Konfigurácia prostredníctvom IEEE Std. 1149.1

Špeciálne konfiguračné pamäte.

Najnovší konfiguračný štandard rieši potrebu veľkokapacitnej konfigurácie skombinované s priemyselným štandardom. Rozšírené konfiguračné rozhranie výrazne redukuje požadovaný čas konfigurácie.

Nové konfiguračné rozhranie poskytuje tieto konfiguračné módy [12]:

- 1-bit passive serial (PS)
- 2-bit passive serial
- 4-bit passive serial
- 8-bit passive serial
- Fast passive parallel (FPP)

Tento rozšírený konfiguračný mód umožňuje súbežne konfigurovať viacero obvodov alebo zariadení naraz. Konfigurované obvody nemusia byť tej istej rodiny ani tej istej hustoty integrácie no musia podporovať rozšírený konfiguračný mód SRAM-based. Potom pomocou ôsmich bitov sa môže naraz konfigurovať až osem zariadení. Počet pripojených konfigurovaných obvodov závisí na počte bitov tak ako to zobrazuje Tab. 3.

Tab. 3 Odporúčaný počet zariadení pre n-Bitový konfiguračný PS mód

Počet zariadení	Odporúčaný konfiguračný mód
1	1-bit PS
2	2-bit PS
3	4-bit PS
4	4-bit PS
5	8-bit PS
6	8-bit PS
7	8-bit PS
8	8-bit PS

1.2 Konfiguračné formáty

Program Alera Quartus II môže generovať jeden alebo viac konfiguračných a programovacích súborov [2]. Softvér pri kompilácii automaticky generuje súbory SRAM Object File (.sof) a Programmer Objekt File (.pof) pre konfiguráciu obvodu. Po malej zmene nastavení kompilátora nám program vygeneruje viacero súborov.

V krátkosti sa popíšu jednotlivé podporované súbory:

- SRAM Objekt FILE (.sof) – tento súbor sa využíva pre Passive Serial konfiguráciu programom Quartus II [22] pomocou zariadení, USB Blaster [4], MasterBlaster [7], Byte Blaster II [3], EthernetBlaster [6] a ByteBlaster MV [5]. Ako bolo spomenuté tento súbor sa generuje automaticky a všetky ostatné konfiguračné súbory sú vytvárané z tohto súboru.
- Programmer Object File (.pof) – Používaný na programovanie konfiguračných súčiastok pomocou programovacieho hardvéru Altera. Taktiež je generovaný automaticky Quartusom.
- Raw Binary File (.rbf) – Binárny súbor obsahujúci konfiguračné dáta. Dáta môžu byť priamo čítané napr. konfiguračným mikroprocesorom.
- Raw Programming Data File (.rpd) – Binárny súbor obsahujúci konfiguračné dáta pre Cyclone FPGA, uložené v konfiguračnej pamäti a používané v AS mode.
- Hexadecimal (Intel-Format) file (.hes) – ASCII súbor v Intel-HEX formáte. Používaný napr. externými programátormi, mikroprocesormi.
- Tabular Text File (.tff) – ASCII súbor obsahujúci jednotlivé položky oddelené čiarkami. Môže byť priamo začlenený do takmer ľubovoľných zdrojových kódov (napr. ASM, C, ...).
- Jam File (.jam) – ASCII súbor v špecializovanom JAM programovacom jazyku, ktorý umožňuje zapísať programovacie informácie, verifikovať a kontrolovať vymazanie súčiastky pomocou JTAG rozhrania.
- Jam Byte Code File (.jbc) – Binárna verzia Jam File umožňujúca zapísať programovacie informácie, verifikovať a kontrolovať vymazanie súčiastky.

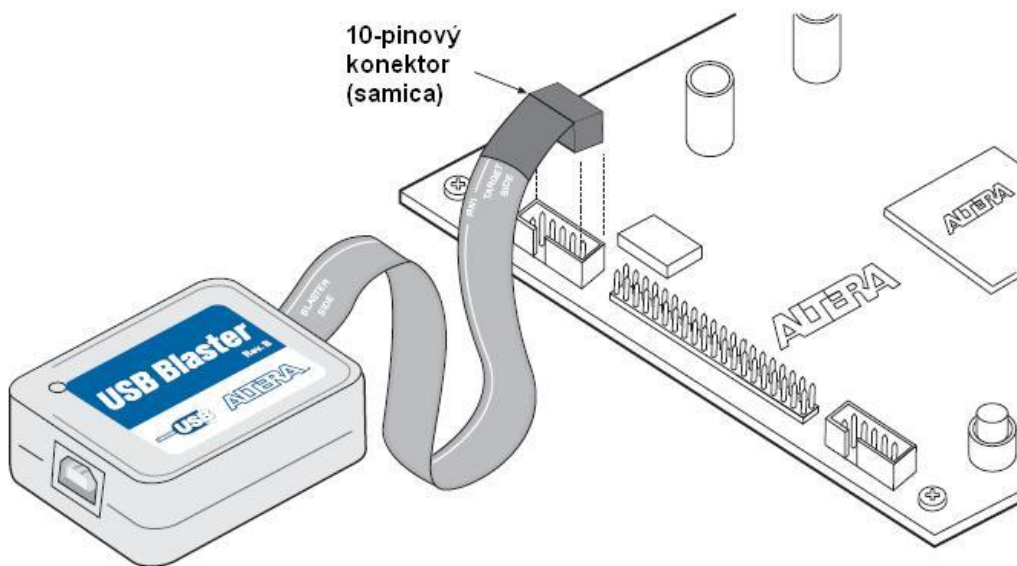
1.3 Konfiguračný hardver

Firma Altera poskytuje viacero konfiguračných nástrojov. Bohužiaľ popísať každý nástroj by bolo veľmi obširne a presiahlo by to potreby tejto práce uvedie sa opis iba jedného konfiguračného nástroja USB Blaster Download Cable, ktorý sa využíva najčastejšie.

Konfiguračný hardvér poskytnutý firmou Altera:

- ByteBlaster II Download Cable [3]
- ByteBlasterMV Download Cable [5]
- EthernetBlaster Communications Cable [6]
- MasterBlaster Serial/USB Communications [7]
- USB Blaster Download Cable [4] – je zobrazený na Obr. 3

K počítači sa pripája pomocou USB (Universal Serial Bus) konektora a k doske pomocou 10-pinového konektora. Programovacie napätie sa pohybuje v rozmedzí 1,5V až 5V v závislosti od programovaného obvodu.



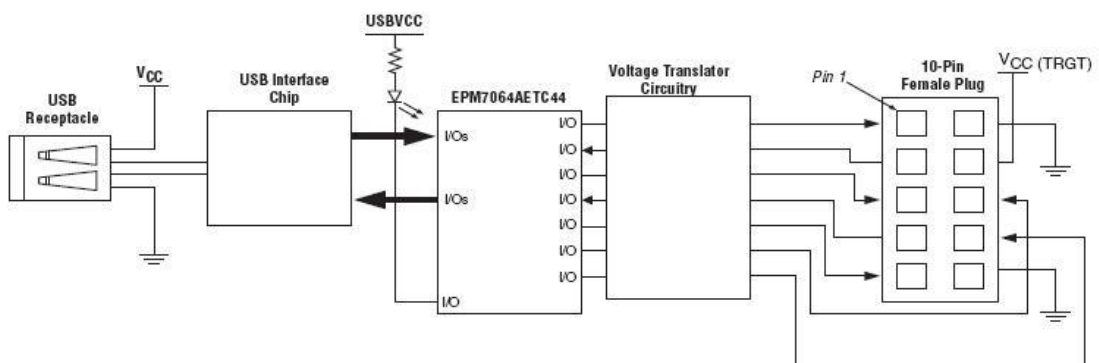
Obr. 3 Konfiguračný hardvér USB Blaster Download Cable firmy Altera

USB Blaster [4] podporuje tieto rodiny obvodov:

Stratix® series FPGAs, Cyclone® series FPGAs, MAX® series CPLDs, Arria™ GX series FPGAs, APEX™ series FPGAs, ACEX® 1K FPGAs, Mercury™ FPGAs, FLEX 10K® series FPGAs, Excalibur™ FPGAs vrátane pokročilej konfigurácie obvodov

EPC2, EPC4, EPC8, EPC16, EPC1441 a sériovej konfigurácii obvodov EPCS1, EPCS4, EPCS16, EPCS64 a EPCS128.

Bloková schéma zobrazená na Obr. 4 zobrazuje hlavné bloky programátora. Jednotlivé bloky sú opísane smerom od USB konektora, ktorý zabezpečuje napájanie a komunikáciu programátora z počítačom pomocou USB kábla. Za konektorom nasleduje prevodník z USB rozhrania na paralelnú zbernicu, ktorá je privedená do ďalšieho bloku vytvoreného obvodom EPM7064A. Tento obvod sa stará o generovanie programovacích signálov na výstupnom konektore. Tieto signály však treba napäťovo upraviť aby nedošlo k poškodeniu programátora alebo zariadenia.



Obr. 4 Bloková schéma konfiguračného zariadenia USB Blaster

2 Univerzálna vývojová doska CPLD start kit

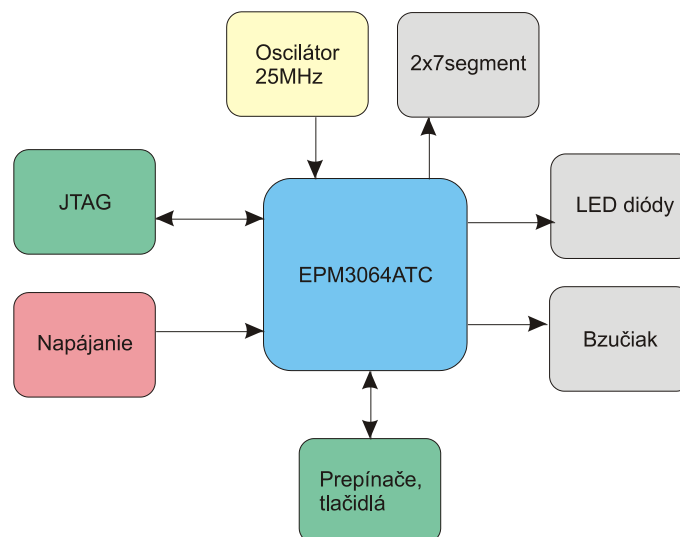
2.1 Opis zariadenia

Zariadenie bolo skonštruované vďaka myšlienke realizácie zariadenia, ktoré by slúžilo ako výučbová pomôcka pre špecializované cvičenia na katedre Elektroniky a multimediálnych telekomunikácií. Konštrukcia zariadenia je riešená tak, aby umožňovala vytvoriť čo najviac príkladov pre rôzne stupne zložitosti precvičovaných zapojení, pri čo najmenšej cene, spotrebe a s minimom externých zariadení.

Rozhodnutie využiť CPLD EPM3064A [8] poskytnutý firmou Elcom spol. s r.o., Prešov [13] bolo kľúčové, pretože obvod poskytuje integráciu 64 makro buniek, ktoré umožňujú naprogramovanie jednoduchých no pre účely výučby postačujúcich kombinačných a sekvenčných štruktúr, pri zachovaní malých rozmerov, nízkej spotrebe a nízkej cene.

Rozhodujúce pre využitie zariadenia je taktiež využitie vstupných a výstupných komponentov.

Zvyšné obvody zariadenia sa vyberali s ohľadom na nízku spotrebu a nízku cenu.

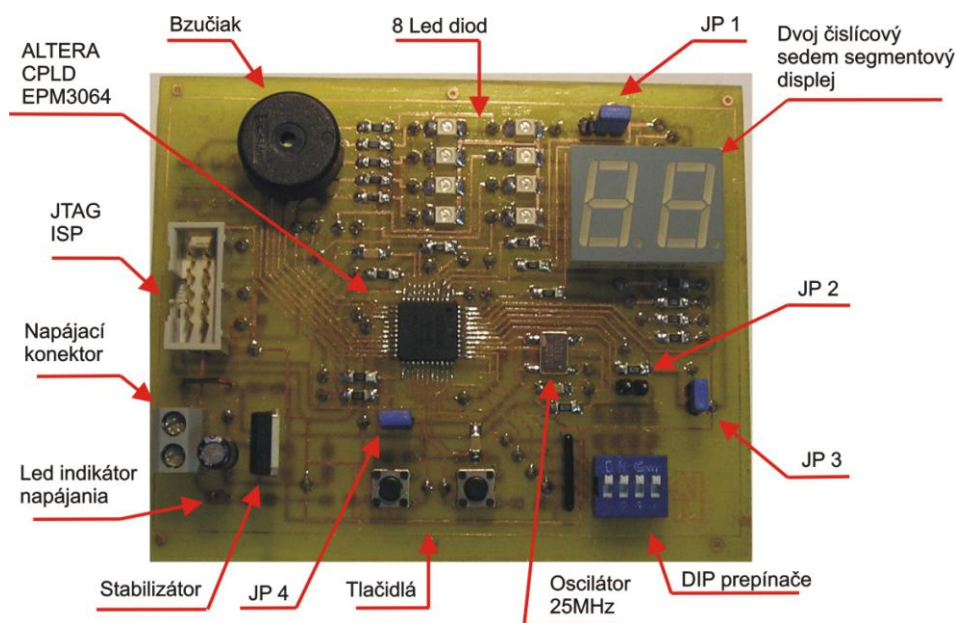


Obr. 5 Bloková schéma CPLD start kit

Obr. 5 zobrazuje blokovú schému prípravku CPLD start kit., ktorá sa skladá z týchto základných blokov:

- Vstupné komponenty z pohľadu CPLD obvodu:
 - napájací konektor – zabezpečuje napájanie prípravku
 - JTAG konektor – zabezpečujúci konfiguráciu obvodu
 - oscilátor – zabezpečujúci hodinové impulzy o frekvencii 25MHz
 - štyri prepojky – upresňujúce bližšiu konfiguráciu zariadenia. Jednotlivý význam prepojek bude opísaný neskôr
 - dve tlačidlá – zadávajúce binárne vstupy (log.1 a log.0)
 - pole 4 prepínačov – zadávajúce binárne vstupy (log.1 a log.0)
- Výstupné komponenty z pohľadu CPLD obvodu:
 - bzučiak – slúži na akustickú signalizáciu
 - osem LED diód – umožňujú zobrazenie binárneho výstupu, pričom binárny výstup je reprezentovaný rozsvietením alebo nerozsvietením LED diódy
 - dvoj číslcový sedem-segmentový display

Jednou z výhod daného zariadenia je využitie možnosti programovania v systéme (ISP – In System Programing) obvodu CPLD. Všetky operácie spojené z mazaním, zápisom a verifikáciou matice EEPROM sa riešia pomocou JTAG rozhrania. Ten umožňuje programovanie obvodu CPLD EPM3064 [8] bez nutnosti odpájania a pripájania špeciálneho programátora vyvinutého na KEMT FEI TU v Košiciach [14].



Obr. 6 Opis jednotlivých elementov prototypovej dosky CPLD start kit

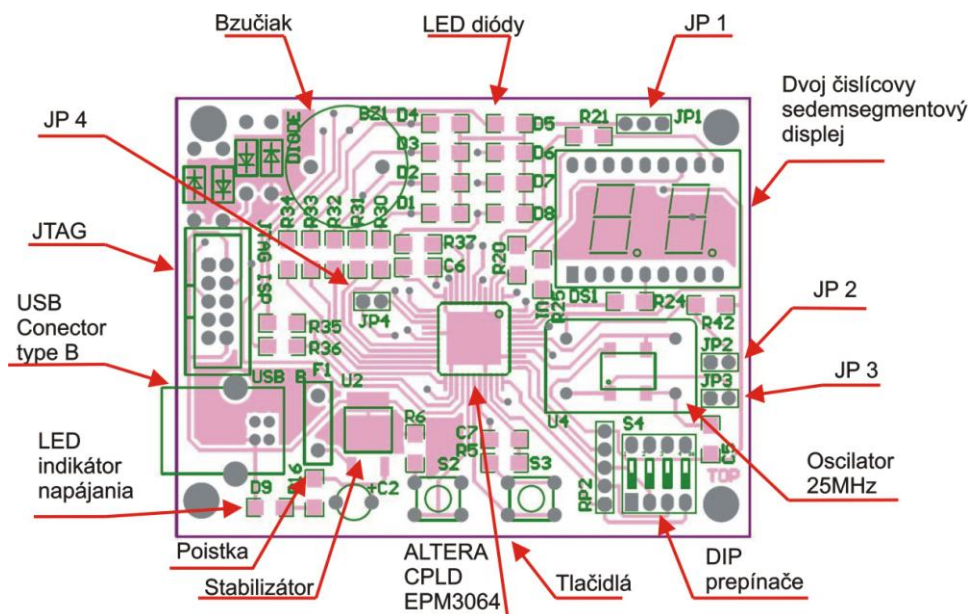
Na Obr. 6 je zobrazené zapojenie jednotlivých elementov na prototypovej doske plošných spojov. Prototyp prípravku je realizovaný za amatérskych podmienok.

Pri domácej výrobe obojstranných dosiek je nutné riešiť niekoľko problémov a i pri najväčšej snahe vyriešiť ich, môže byť výsledok neuspokojivý. Preto sa rozhodlo dať vyrobiť dosky plošných spojov profesionálne.

Na Obr. 7 je zobrazené umiestnenie jednotlivých komponentov na doske plošných spojov vyrobených profesionálne.

Počas písania tejto diplomovej práce nebola ešte profesionálne vyrobená doska plošných spojov k dispozícii. CPLD start kit prípravok s profesionálne vyrobenou doskou plošných spojov bude demonštrovaný pri obhajobe diplomovej práce.

V Prílohe. B sú uvedené výrobné dáta vo formáte Gerber a taktiež celý databázový súbor zo schémou a kompletným návrhom dosky plošných spojov pre program Protel 99SE. V Prílohe A1 je zobrazená schéma zapojenia CPLD start kitu, v Prílohe A2 je zobrazená doska plošných spojov zo strany súčiastok (Top layer) a v Prílohe A3 je zobrazená doska plošných spojov zo strany spojov (Bottom layer).



Obr. 7 Zapojenie jednotlivých elementov profesionálnej dosky CPLD start kit

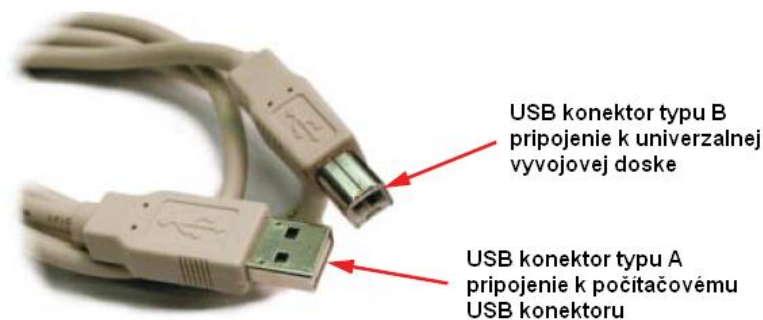
2.2 Opis konštrukcie

Elektrická schéma zariadenia je zobrazená v Prílohe A1. V nasledujúcich podkapitolách sa podrobnejšie rozoberú jednotlivé časti schémy.

Pri riešení tejto konštrukcii bola požiadavka využiť minimum externých zariadení. To sa splnilo tým, že k zariadeniu sa pripájajú iba dva externé zariadenia a to je USB JTAG programátor a zdroj napájacieho napätia, ktorý je zabezpečený cez USB rozhranie.

2.2.1 Napájacia časť

Napájanie vývojovej dosky je zabezpečené cez USB rozhranie na pracovnom počítači. Doska sa pripája k počítaču pomocou USB kábla zobrazeného na Obr. 8.



Obr. 8 Napájací kábel prípravku CPLD start kit

Základné parametre USB rozhrania

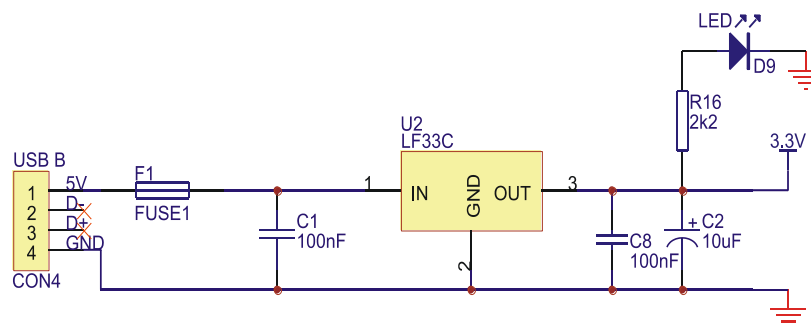
- Komunikačná rýchlosť 1,5Mbit/s do 480Mbit/s
- Komunikačná vzdialenosť do 5m
- Možnosť pripojenia viacerých zariadení
- Rozhranie obsahuje 5V napájanie a maximálne 500 mA odber

Práve posledný bod vlastností rozhrania poskytujúci stále napätie 5V a odber prúdu do 500mA sa využije ako zdroj energie. Napájanie z USB rozhrania sa prideliuje v jednotkách po 100mA. Tesne po pripojení zariadenia k rozhraniu začína identifikácia zariadenia a počas identifikácie sa nesmie odoberať prúd viac ako jednu jednotku

výkonu (100mA na 5V). Až po prebehnutí identifikácie môže zariadenie odoberať spomínaných 500mA.

Kvôli tejto vlastnosti bolo potrebné rozhodnúť, či sa využije možnosť napájania iba 100mA, alebo sa do zariadenia pridá obvod na identifikáciu (napr. FT 232 RL [23]) a získa sa možnosť odberu zariadenia do 500mA.

Z rozmerových a ekonomických požiadaviek na zariadenie sa upustilo od pridania obvodu na identifikáciu. Týmto krokom sa stanovil maximálny možný odber zariadenia na 100mA, čomu sú prispôsobené zvyšné časti zariadenia. Ako vyplýva z Obr. 9 na zariadení sa nachádza USB konektor typu B.



Obr. 9 Napájacia časť prípravku CPLD start kit

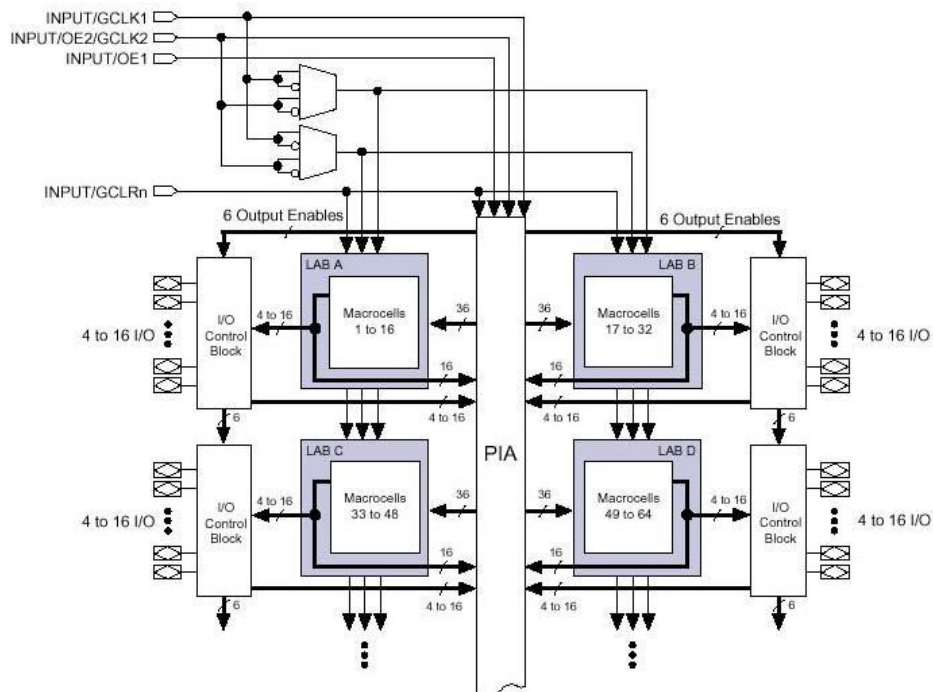
Za konektorom nasleduje vratná poistka F1 [15] (pozri Obr. 9), ktorá pri odbere prúdu viac ako 100mA odpojí obvod od napájania a pri opätovnom poklese prúdu pod 100mA uzavrie obvod a zariadenie môže pokračovať v prevádzke. Výhoda tejto poistky pozostáva vtom, že nie je potrebné priamo zasahovať do zariadenia, a taktiež nie je nutné vymieňať súčiastku, čo je ekonomicky výhodné.

USB rozhranie poskytuje napájacie napätie 5V, no hlavným obvodom zariadenia je obvod EPM3064ATC44-10 [8] firmy Altera [9], ktorý vyžaduje napájacie napätie 3,3V. Preto je potrebné pomocou stabilizátora napätia upraviť napätie na hodnotu 3,3V. Na túto úlohu je zvolený stabilizátor LF33CDT [10] s nízkym poklesom napätia na výstupe (typicky 0,45V pri 500mA) a nízkou spotrebou (typicky 500uA) s maximálnym výstupným prúdom až 1A.

Kondenzátory C1 a C8 sú blokovacie kondenzátory. LED dióda D9 [16] (červenej farby) signalizuje, že je zariadenie pod napätím.

2.2.2 Obvod CPLD EMP3064ATC44-10

Jedná sa o CPLD programovateľný obvod na báze CMOS EEPROM s architektúrou MAX (Multiple Array MatriX) z rodiny MAX3000 [8]. Bloková schéma rodiny MAX je zobrazená na Obr. 10.



Obr. 10 Bloková schéma rodiny MAX výrobcu Altera

Základné vlastnosti obvodu:

3,3V programovanie v systéme (ISP)

Vylepšený algoritmus ISP pre rýchlejšie programovanie, pull-up rezistory na I/O pinoch počas ISP

Oneskorenie pin to pin 4,5ns, hodinová frekvencia až do 222 MHz

I/O rozhranie kompatibilné s 5V, 3,3V a 2,5V logickými úrovňami, pričom jadro je napájané 3,3V

Prevedenie v puzdrách TQFP, PQFP, PLCC

Obsahuje 1 250 hradiel, 64 makro buniek, 4 logické polia blokov a maximálne 66 používateľských I/O pinov (v závislosti od použitého puzdra súčiastky).

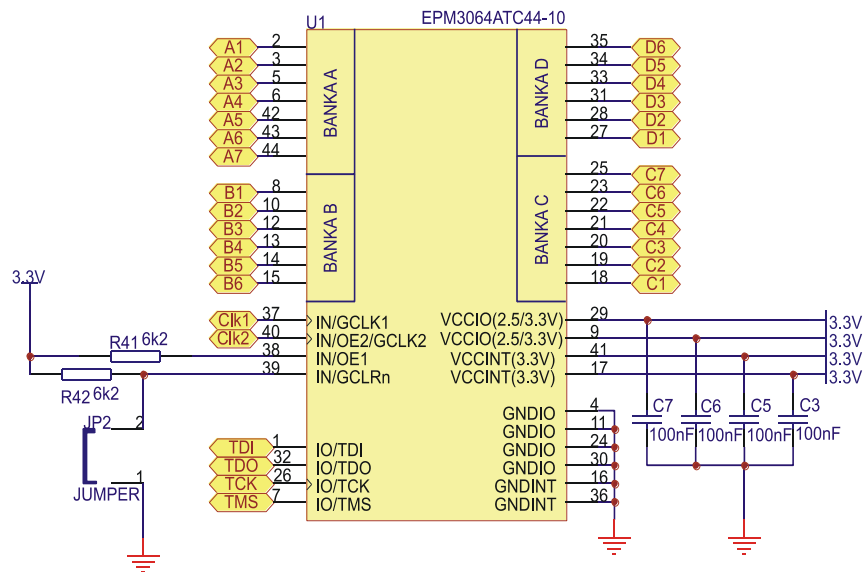
Zapojenie obvodu je zobrazené na Obr. 11.

Pri návrhu tohto zariadenia bol k dispozícii obvod v 44 pinovom PQFP puzdre. Týchto 44 pinov sa môže rozdeliť do 4 základných skupín.

- 4 vyhradené vstupy, ktoré môžu byť použité ako obyčajné vstupy, alebo ako vysokorýchlostné, globálne riadiace signály (clock, clear, a dva signály output enable) pre každú makro bunku a I/O pin
- 10 napájacích vstupov (6 je pripojených na zem, 2 sú pripojené na napájacie napätie, a dve slúžia na konfiguráciu výstupných napätí pozri Tab. 4)
- 4 I/O vývody, ktoré sa používajú pre JTAG rozhranie
- 26 I/O vývodov, ktoré môžu byť nakonfigurované ako vstupy alebo výstupy

Postupne bude opísané pripojenie všetkých vývodov a Prílohe A (P3) je v tabuľke uvedený prehľad zapojenia vývodov obvodu EMP3064 [8] k jednotlivým komponentom zariadenia.

Medzi najväčšie špecifiká zapojenia patrí možnosť pripojiť na vývody VCCIO napätie 3.3V alebo napätie 2.5V podľa závislosti požadovaného napätia na výstupe. Táto hodnota ovplyvňuje napäťové úrovne na výstupe obvodu tak ako to zobrazuje tabuľka Tab. 4.



Obr. 11 Zapojenie obvodu CPLD EPM3064 na CPLD start kite

Tab. 4 Napäťové úrovne vstupov a výstupov pri zmene napätia na VCCIO

Napätie na VCCIO	Vstupný signál			Výstupný signál		
	2.5V	3.3V	5V	2.5V	3.3V	5V
2.5V	✓	✓	✓	✓		
3.3V	✓	✓	✓	✓	✓	✓

V tomto zariadení sa bude využívať väčšie napätie ako 2.5V, preto sú vývody VCCIO pripojené na napájacie napätie 3.3V.

Ďalším zaujímavým vývodom je OE1 (Output enable), zabezpečujúci povolenie alebo zakázanie výstupu z obvodu. Pre povolenie funkčnosti obvodu je nutné tento vývod pripojiť na napájacie napätie 3.3V. Rezistorom R41 sa nastavuje úroveň vstupného prúdu pre vývod OE.

Posledným vyhradeným vstupom je zapojenie vývodu GCLRn. Vďaka tomuto vývodu je možné nulovať každý register a to tak, že sa vývod pripojí na zem. Každý register podporuje asynchrónne prednastavenie a nulovanie. Rezistor R42 je využitý ako ochrana pred skratom pri prepojení JP2.

Zvyšné vývody sú pripojené k jednotlivým komponentom zariadenia a ich presnejšia špecifikácia pripojenia bude opísaná nižšie.

2.2.3 Konfiguračné rozhranie JTAG ISP (konektor)

JTAG je štandard definovaný normou IEEE 1149.1, tzv. Standard Test Access Port (TAP). Jedná sa o architektúru Boundary-Scan pre testovanie plošných spojov, programovanie FLASH pamätí a pod. Dáta sú prenášané sériovo.

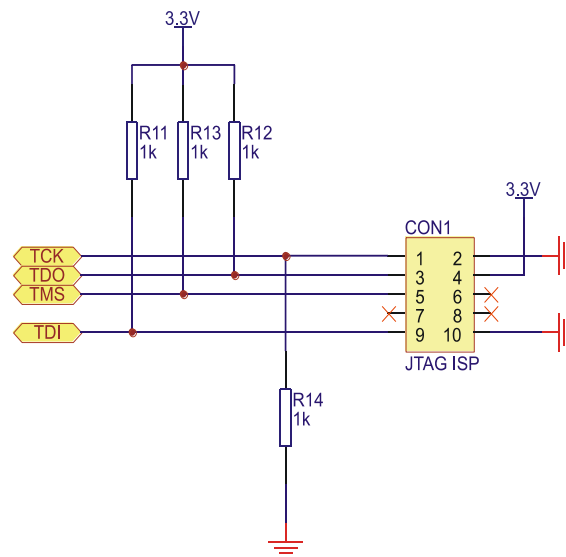
Základne signály:

- TDI (Test Data In) – dátový vstup TDI sa používa k sériovému prenosu testovacích (programovacích) dát obvodom. Hodnoty zo vstupu TDI sú načítané do príslušného dátového alebo inštrukčného registra, ktorý je zvolený aktuálnou inštrukciou. Hodnota na vstupe TDI je platná pri nábežnej hrane TCK
- TDO (Test Data Out) – dátový výstup TDO sa používa k sériovému prenosu dát. Hodnota je platná pri zostupnej hrane TCK
- TCK (Test Clock) – vstupný signál TCK je do TAP zaradený preto, aby sériová dátová cesta medzi komponentmi zariadenia mohla byť používaná nezávisle na hodinách jednotlivých komponentov. Tento spôsob taktiež dovoľuje aby posun testovacích dát prebiehal paralelne s činnosťou zariadenia
- TMS (Test Mode Select) – tento vstupný signál je dekódovaný radičom TAP a slúži k výberu testovacích operácií. Pomocou hodnôt TMS sú riadené prechody medzi stavmi v stavovom diagrame radiča. Hodnota na vstupe TMS je snímaná pri nábežnej hrane TCK

- nTRST (Test ReSeT) – tento signál je nepovinný. Pokiaľ je prítomný, potom, ak je jeho hodnota rovná log.0, musí radič TAP prejsť asynchrónne do stavu Test-Logic-Reset. Tento signál nie je povinný pretože inicializáciu radiča TAP môžeme dosiahnuť aj inak a to buď aplikáciou log.1 na signál TMS po dobu minimálne 5-tich nábežných hrán TCK alebo zapnutím napájania integrovaného obvodu

Radič TAP je synchronný konečný automat, ktorý riadi činnosť testovacej logiky vo vnútri integrovaného obvodu v závislosti na signály TMS.

Zapojenie je JTAG rozhrania je zobrazené na Obr. 12 a je realizované podľa odporúčaného zapojenia firmy Altera [24].



Obr. 12 Zapojenie konfiguračného rozhrania JTAG

Ako programátor pre rozhranie JTAG sa využíva USB Blaster vyvinutý na KEMT FEI TU v Košiciach [14], ktorý sa pripája ku konektoru CON1 (JTAG ISP).

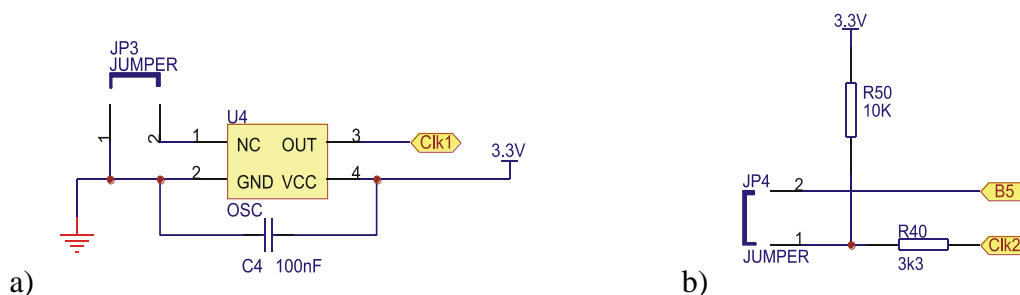
Zapojenie jednotlivých pinov na konektore a vyvodov na CPLD obvode EMP3064 [8] pre TQFP puzdro (zapojenie pinov sa pre jednotlivé puzdra líši) je uvedené v Tab. 5.

Tab. 5 Zapojenie pinov CPLD obvodu a konektora JTAG rozhrania

Číslo vývodu na konektore CON1	Označenie signálu	Názov signálu	Číslo vývodu na CPLD EMP3064
1	TCK	Clock signal	26
2	GND	Ground	–
3	TDO	Data from Device	32
4	VCC	Power supply	–
5	TMS	Jtag state machine control	7
6	–	No connect	–
7	–	No connect	–
8	–	No connect	–
9	TDI	Data to device	1
10	GND	Ground	–

2.2.4 Generátor hodinových impulzov

Ako generátor hodinových impulzov je využitý kryštálový oscilátor [17] o frekvencii 25MHz . Schéma zapojenia oscilátora je zobrazená na Obr. 13 a).



Obr. 13 Zapojenia oscilátora a) a prepajky JP4 na pripojenie k GCLK2

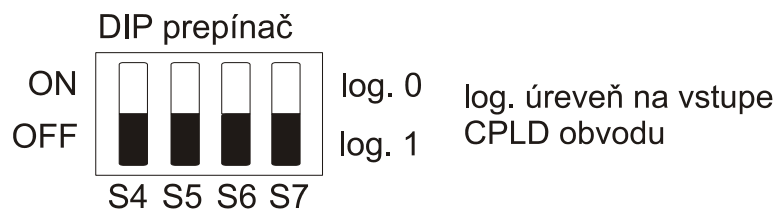
Prepojka JP3 v Obr. 13 a) je využitá na zapojenie alebo odpojenie oscilátora. Výstup oscilátora je vyvedený na vstup GCLK1 obvodu CPLD EMP3064 [8]. Frekvencia z oscilátora sa môže v obvode upraviť (napr. znížiť deličkou) a pomocou vývodu 14 nastaveného ako výstup priviesť na vývod GCLK2. Zapojenie výstupu a vstupu upraveného signálu je zobrazený na Obr. 13 b). Krátky opis zapojenia pinov je v Tab. 6.

Tab. 6 Priradenie pinov CPLD obvodu s oscilátorom a JP4

Číslo pinu na CPLD EMP3064	Zapojenie komponentu
37	Oscilátor U4
40	Pin 14 obvodu EMP3064 cez prepojkú JP4

2.2.5 Tlačidlá, prepínače

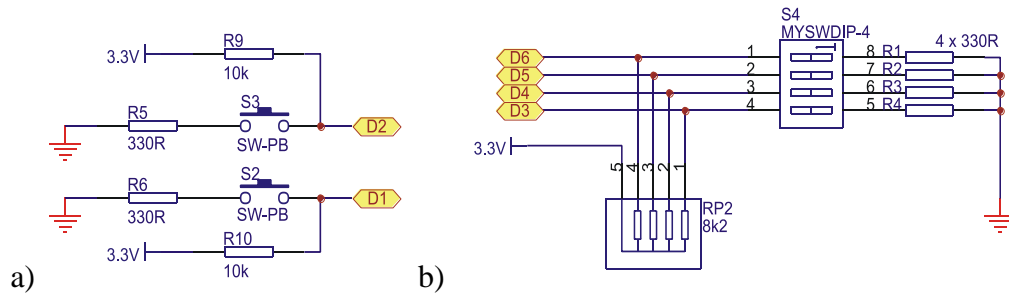
Zariadenie obsahuje 6 vstupných tlačidiel, dve s toho sú mikro-tlačidlá (S2 a S3) a štyri DIP prepínače (S4, S5, S6, S7 pozri Obr. 14).



Obr. 14 Rozdelenie jednotlivých prepínačov pre DIP prepínač

Zapojenie tlačidiel je zobrazené na Obr. 15 a) a prepínačov na Obr. 15 b).

Všetky tlačidlá pracujú v negatívnej logike to znamená, že vo vypnutom stave je na pine log.1. Táto log.1 je zabezpečená vďaka rezistorom pripojeným na napájacie napätie. Pri stlačení tlačidla sa pomocou rezistora pripojeného na zem upraví na vývode úroveň napätia na log.0. Hodnota rezistorov [20] je zvolená tak, aby úbytok na rezistore pripojenom na napájacie napätie bol nízky. Výrobca obvodu udáva štandardnú hodnotu odporu 10 K Ω . Keďže vstupný odpor obvodu CPLD EMP3064 [8] je veľký, úbytok napätia na tomto rezistore je malý. Rezistory pripojené na zem majú ochrannú funkciu. Ak by sa vývod obvodu CPLD nastavil ako výstup a tlačidlo alebo prepínač boli pripojené na zem, nastal by skrat a obvod by sa mohol poškodiť. Hodnota tohto rezistora bola zvolená tak aby výstupný prúd z obvodu CPLD nepresahoval maximálnu dovolenú hodnotu výstupného prúdu (25mA) no zároveň neprekročil hodnotu úrovne log.0. Zapojenie vývodov obvodu CPLD a prepínačov je uvedené v Tab. 7.



Obr. 15 Zobrazenie zapojenie tlačidiel a) a DIP prepínača b) na prípravku CPLD start kite

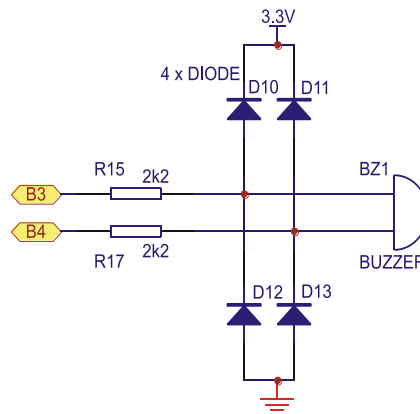
Tab. 7 Priradenie pinov CPLD obvodu tlačidlám a DIP prepínačom

Číslo pinu CPLD EMP3064	Zapojenie elementu	Číslo pinu CPLD EMP3064	Zapojenie elementu
27	S2 (mikrotlačidlo)	35	S4 (DIP prepínač)
28	S3 (mikrotlačidlo)	34	S5 (DIP prepínač)
–	–	33	S6 (DIP prepínač)
–	–	31	S7 (DIP prepínač)

2.2.6 Bzučiak

Elektroakustický menič je pre používateľa veľmi príjemným sprestením práce. Preto sa do prípravku umiestni bzučiak. Aby to však používateľ nemal veľmi jednoduché použiť sa bzučiak bez budiča. Zapojenie tohto elementu nie je zložité a je zobrazené na Obr. 16. Využitie bzučiaka bez budiča poskytuje väčšie možnosti využitia používateľom (zmena budiacej frekvencie). Budiace napätie bzučiaka je 5V ale CPLD obvod dokáže dosiahnuť na výstupe iba napätie 3.3V. Preto je potrebné obsadenie dvoch vývodov. Zapojením bzučiaka na dva vývody umožňuje dosiahnuť väčší rozkmit budiaceho napätia ako pri zapojení jedného vývodu bzučiaka na CPLD a druhého na zem. Pripojené rezistory upravujú výstupný prúd. Zapojené diódy slúžia na ochranu pinov CPLD obvodu, pretože pri manipulácii zo zariadením sa pri náhodnom klepnutí po bzučiacu dokáže vybudit' vysoké napätie, ktoré by mohlo CPLD obvod poškodiť. Dané zapojenie diód sa opíše iba pre jednu vetvu, pretože funkcia je rovnaká pre obe vetvy. Pri klasickom napätí (3.3V) pre budenie bzučiaka sú obe diódy v nepriepustnom smere, pretože na anóde diódy D10 je len o niečo menšie napätie ako na katóde, a taktiež na katóde diódy D12 je väčšie napätie ako na anóde. Pri vybudenom napätí bzučiaka (desiatky až stovky voltov) je pri kladnom napätí otvorená dióda D10 a pri

zápornom napätí je otvorená dióda D12. Zapojenie vývodov CPLD obvodu a bzučiaka je zobrazený v Tab. 8.



Obr. 16 Spôsob zapojenia bzučiaka na CPLD start kit

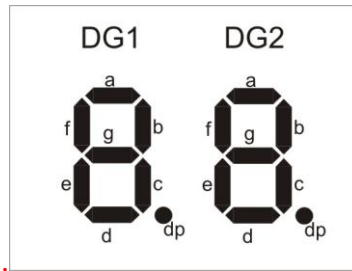
Tab. 8 Priradenie pinov CPLD obvodu bzučiaku

Zapojenie pinov na CPLD	Zapojenie elementu
12	Bzučiak (pin 1)
13	Bzučiak (pin2)

2.2.7 LED diódy, dvoj číslicový sedem-segmentový displej

Zariadenie obsahuje osem LED diód a jeden dvoj číslicový sedem-segmentový displej. Ako vidieť na Obr. 18, všetky LED diódy a sedem-segmentový displej sú zapojené v negatívnej logike. To znamená, že pri logickej úrovni log.1 na výstupe obvodu CPLD sú komponenty neaktívne (zhasnuté). Pre efektívnosť využitia pinov obvodu CPLD sa pomocou prepojky JP1 prepína medzi prvou číslovkou sedem-segmentového displeja DS1 časti DG1 a LED diódami (pozri Obr. 19). Týmto zapojením sa ušetrilo 8 I/O pinov obvodu CPLD. Predradné rezistory boli vyrátane, aby prúd pre každú vetvu bol stanovený na 1,5 mA. Ak by sa nebol obmedzil odber prúdu pre každú vetvu vzniklo by veľmi veľké zaťaženie (ak by bol odber jednej LED diódy len 4mA pri zapojení 16 diód by bol odber prúdu 64mA, čo by presiahlo spolu s ostatnými komponentmi hodnotu 100mA).

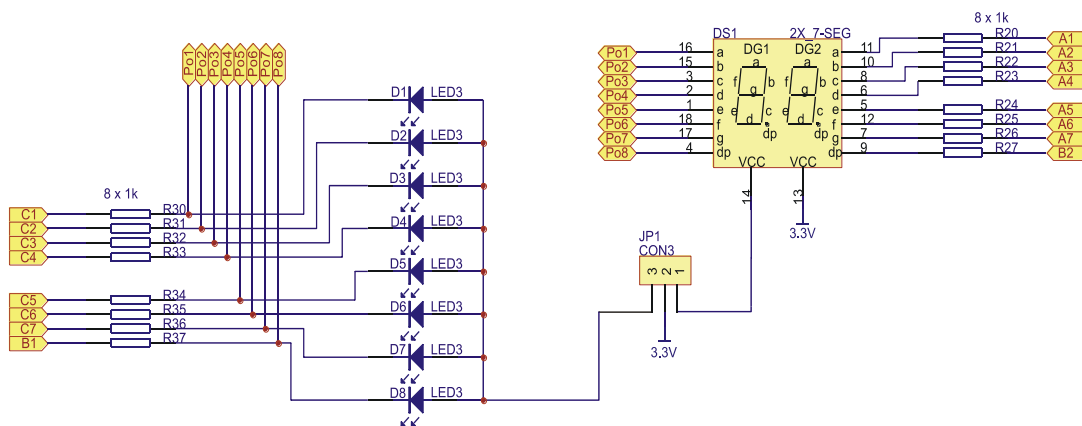
Zapojenie pinov CPLD obvodu a jednotlivých elementov je zobrazené v Tab. 9.



Obr. 17 Rozmiestnenie DG1 a DG2 na sedem-segmentovom displeji

Tab. 9 Priradenie pinov CPLD obvodu pre LED diódy a obe čísla sedem-segmentového displeja

Číslo vývodov CPLD obvodu	Zapojenie komponentu		Číslo pinu CPLD obvodu	Zapojenie komponentu DS1 – DG2
	LED	DS1 – DG1		
18	D1	a – DG1	2	a – DG2
19	D2	b – DG1	3	b – DG3
20	D3	c – DG1	5	c – DG4
21	D4	d – DG1	6	d – DG2
22	D5	e – DG1	42	d – DG2
23	D6	f – DG1	43	f – DG2
25	D7	G – DG1	44	g – DG2
8	D8	dp – DG1	10	dp– DG2



Obr. 18 Zapojenie LED a sedem-segmentového displeja na CPLD start kit

2.3 Prúdová analýza zapojenia

Napájací zdroj poskytne prúd iba 100mA. Spotreba pri využití všetkých elementov na zariadení by nemala prekročiť hodnotu 100mA. Preto sa jednotlivé obvody navrhli s čo najmenšou spotrebou. Predpokladané hodnoty prúdového zaťaženia pre jednotlivé komponenty prípravku uvedené v Tab. 10 boli získane z katalógových listov komponentov. Ako vyplýva z Tab. 10 predpokladaný obder je 82mA. Po zostrojení zariadenia sa vykonali merania, ktorými sa zistilo, že zariadenie neprekročilo hodnotu 65mA. Testovanie prebiehalo pri zapnutí maximálnych možných elementov – sedem-segmentového displeja, zapnutých tlačidiel a prepínačov do polohy ON, aktívnom oscilátore spolu s deličkou frekvencie oscilátora (v obvode CPLD) a výstupom vyvedeným na bzučiak.

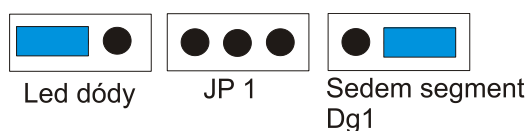
Tab. 10 Spotreba jednotlivých komponentov a celková spotreba zariadenia

Časť obvodu	Spotreba
Napájanie	4mA
CPLD EMP3064	20mA (pri $f=25\text{MHz}$)
LED, 7segment	24mA ($1.5\text{mA} \cdot 16$)
Bzučiak	11mA
Tlačidlá, prepínače	3mA
Oscilátor	20mA
Spolu	82mA

2.4 Zapojenie prepojek

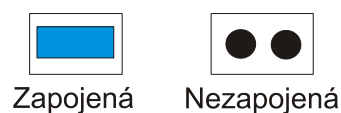
Vývojová doska obsahuje prepojky, ktorými sa ovplyvňuje činnosť zariadenia. Toto ovplyvňovanie je pre používateľa veľmi dôležité pokiaľ chce aby zariadenie fungovalo správne. Preto je v tejto kapitole opísane význam jednotlivých prepojek. Umiestnenie prepojov na prototypovom prípravku je na Obr. 6 a pre profesionálnu na Obr. 7.

- JP1 – touto prepojkou si používateľ prepína medzi zobrazovaním na LED diódach alebo na sedem-segmentovom displeji (DG1) podľa Obr. 19.



Obr. 19 Význam zapojenia prepajky JP 1

- JP2 – jej úlohou je pripojenie vývodu GCLRn obvodu CPLD na zem, pri nulovaní požadovaného registra. Štandardne je tato prepajka nezapojená (pozri Obr. 20).
- JP3 – umožňuje používateľovi vypnúť alebo zapnúť oscilátor. Pri nevyužívaní oscilátora sa odporúča zapojenie prepajky (pozri Obr. 20).
- JP4 – slúži na prepojenie vývodu 14 (ktorý slúži ako výstup z CPLD obvodu pre upravený hodinový impulz) a vývodu GCLK2 obvodu CPLD (ktorý je využitý ako druhý vstup hodinových impulzov). Pre prepojenie oboch vývodov je musí byť zapojená (pozri Obr. 20).



Obr. 20 Zobrazenie zapojenej a nezapojenej prepajky

3 Testovacie projekty zariadenia

Po úspešnej realizácii prípravku je na rade testovanie funkčnosti prípravku. Najlepší spôsob testovania je zostavenie projektov, ktoré otestujú jednotlivé komponenty prípravku. Testy preto nebudú obsiahle ani zložité. Ich hlavnou úlohou je otestovať všetky zapojenie a funkčnosť zariadenia, takže projekty budú zostavené zo základných kombinačných a sekvenčných obvodov.

3.1 Prevodník z BCD na sedem-segment

Veľmi často je potrebné zobrazovať dekadické čísla pomocou sedem-segmentoviek s LED diódami. Keďže je potrebné zobrazit' každú číslicu osobitne, nepracuje sa v binárnom kóde ale v BCD kóde (binárne kódovaná dekadická číslica). Tento kód sa pre čísla 0 až 9 rovná s binárnym kódom no pre čísla 10 až 15 si výrobca prevodníka definuje svoje znaky sám. Nevýhoda je, že pri zobrazovaní vyšších čísel ako 9 musíme zapojiť viac sedem-segmentoviek za sebou.

3.1.1 Zadané úlohy

Navrhnete prevodník kódu BCD na sedem-segmentový kód. Návrh realizujte v grafickom aj VHDL režime prostredia Altera Quartus II [22] s kompletnou časovou a funkčnou analýzou.

3.1.2 Riešenie

Pre realizáciu v grafickom editore je nutné vykonať syntézu zapojenia pozostávajúcu zo:

- Zostavenia pravdivostnej tabuľky podľa požiadaviek činnosti zapojenia
- Zostrojiti' karnaughove mapy premenných a zjednodušiť zapojenie
- Realizovať zapojenie

Ako vstupné premenné sa použijú DIP prepínače (pozri Obr. 14) kde $D = S4$, $C = S5$, $B = S6$, $A = S7$, ktorými sa zadefinuje BCD kód. Výstupy sa privedú na sedem-segmentový display DG2 (pozri Obr. 17). Zapojenie vývodov sedem-segmentového elementu a CPLD obvodu sú opísané v Tab. 9. Treba pripomenúť, že displej využíva negatívnu logiku, a preto sa do zapojenia na výstupy pripojí NOT hradlo. Nevyužitú vstup je potrebné prepnúť do stavu vysokej impedancie.

3.1.3 Syntéza

Podľa požiadavky činnosti prevodníka zostrojíme pravdivostnú tabuľku Tab. 11, v ktorej určíme pre priradené vstupy (A, B, C, D) logické hodnoty výstupov.

Tab. 11 Pravdivostná tabuľka prevodníka z BCD na sedem-segment

Dekadický ekvivalent	Vstupy – BCD kód				Výstup – sedem-segmentový display						
	D	C	B	A	a	b	c	d	e	f	g
1	0	0	0	0	1	1	1	1	1	1	0
2	0	0	0	1	0	1	1	0	0	0	0
3	0	0	1	0	1	1	0	1	1	0	1
4	0	0	1	1	1	1	1	1	0	0	1
5	0	1	0	0	0	1	1	0	0	1	1
6	0	1	0	1	1	0	1	1	0	1	1
7	0	1	1	0	1	0	1	1	1	1	1
8	0	1	1	1	1	1	1	0	0	0	0
9	1	0	0	0	1	1	1	1	1	1	1
10	1	0	0	1	1	1	1	1	0	1	1
11	1	0	1	0	0	0	0	1	1	0	1
12	1	0	1	1	0	0	1	1	0	0	1
13	1	1	0	0	0	1	0	0	0	1	1
14	1	1	0	1	0	0	1	0	1	0	1
15	1	1	1	0	1	0	0	1	1	1	1
16	1	1	1	1	0	0	0	0	0	0	0

Z pravdivostnej tabuľky prevodníka zostrojíme karnaughove mapy, ktoré sú zobrazené na Obr. 21, z ktorých určíme algebraické funkcie pre jednotlivé výstupy zapojenia.

Výstupne funkcie zapíšeme v UDNF (úplnej disjunktnej normálnej formy).

$$a = \overline{D}\overline{C}\overline{A} + \overline{D}\overline{C}B + D\overline{C}\overline{B} + CBA + \overline{D}CA$$

$$b = \overline{B}\overline{A} + \overline{D}\overline{C} + \overline{C}\overline{B} + \overline{D}BA$$

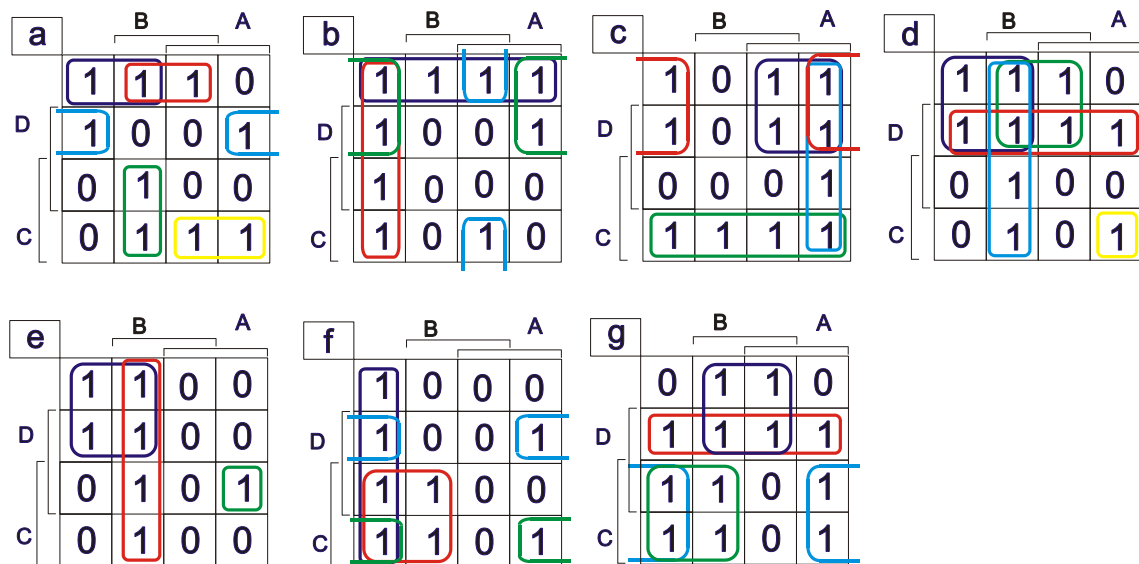
$$c = \overline{C}\overline{B} + \overline{C}A + \overline{B}A + \overline{D}C$$

$$d = \overline{C}\overline{A} + B\overline{A} + \overline{C}B + D\overline{C} + \overline{D}C\overline{B}A$$

$$e = \overline{C}A + B\overline{A} + D\overline{C}\overline{B}A$$

$$f = \overline{B}\overline{A} + C\overline{A} + D\overline{C}\overline{B} + \overline{D}C\overline{B}$$

$$g = C\overline{A} + D\overline{C} + \overline{C}B + C\overline{B}$$



Obr. 21 Karnaughove mapy pre prevodník BCD na sedem-segment

3.1.4 Schematické zapojenie obvodu

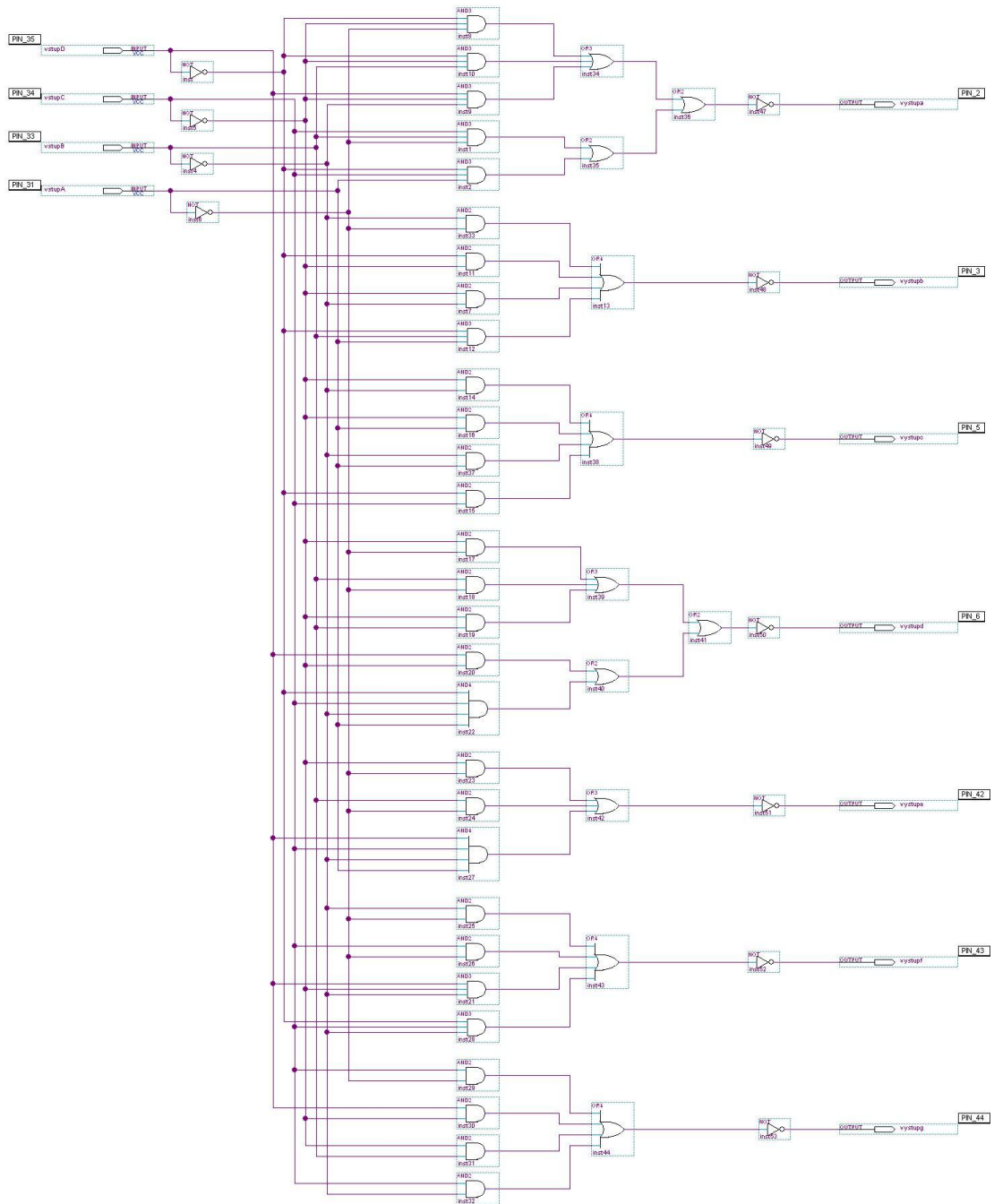
Po vytvorení nového projektu sa môže prejsť k realizácii zapojenia. Zapojenie sa realizuje z výsledných funkcií. Na Obr. 23 je zobrazená schéma zapojenia obvodu.

Pred kompiláciou je potrebné priradiť príslušné čísla pinov, ktoré sú zobrazené na Obr. 22.

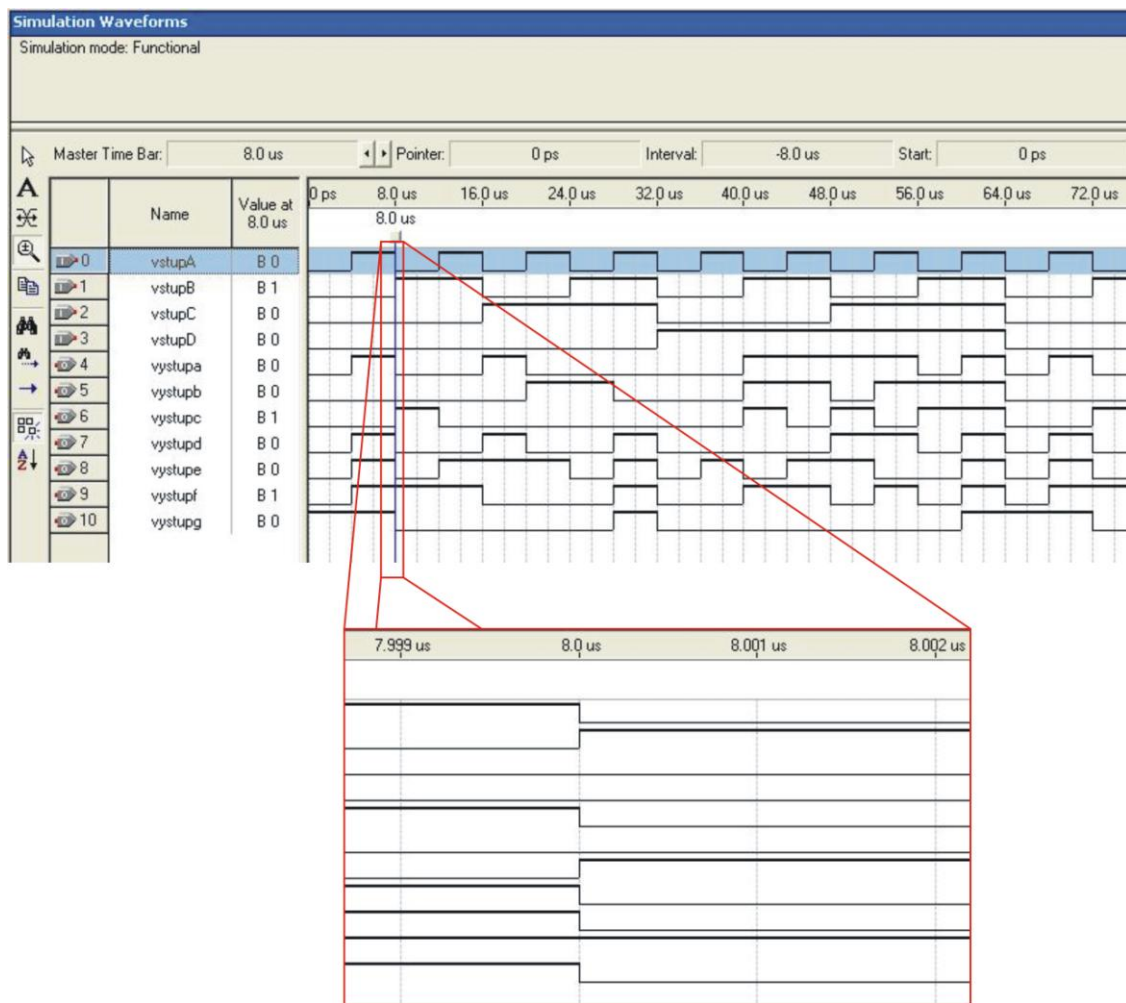
	From	To	Assignment Name	Value	Enabled
1		TCK	Location	PIN_26	Yes
2		TDI	Location	PIN_1	Yes
3		TDO	Location	PIN_32	Yes
4		TMS	Location	PIN_7	Yes
5		vstupA	Location	PIN_31	Yes
6		vstupB	Location	PIN_33	Yes
7		vstupC	Location	PIN_34	Yes
8		vstupD	Location	PIN_35	Yes
9		vystupa	Location	PIN_2	Yes
10		vystupb	Location	PIN_3	Yes
11		vystupc	Location	PIN_5	Yes
12		vystupd	Location	PIN_6	Yes
13		vystupe	Location	PIN_42	Yes
14		vystupf	Location	PIN_43	Yes
15		vystupg	Location	PIN_44	Yes
16	<<new>>	<<new>>	<<new>>		

Obr. 22 Priradenie príslušných pinov v projekt prevodník

Po úspešnej kompilácii sa môže prístupit' k funkčnej a časovej simulácii obvodu.

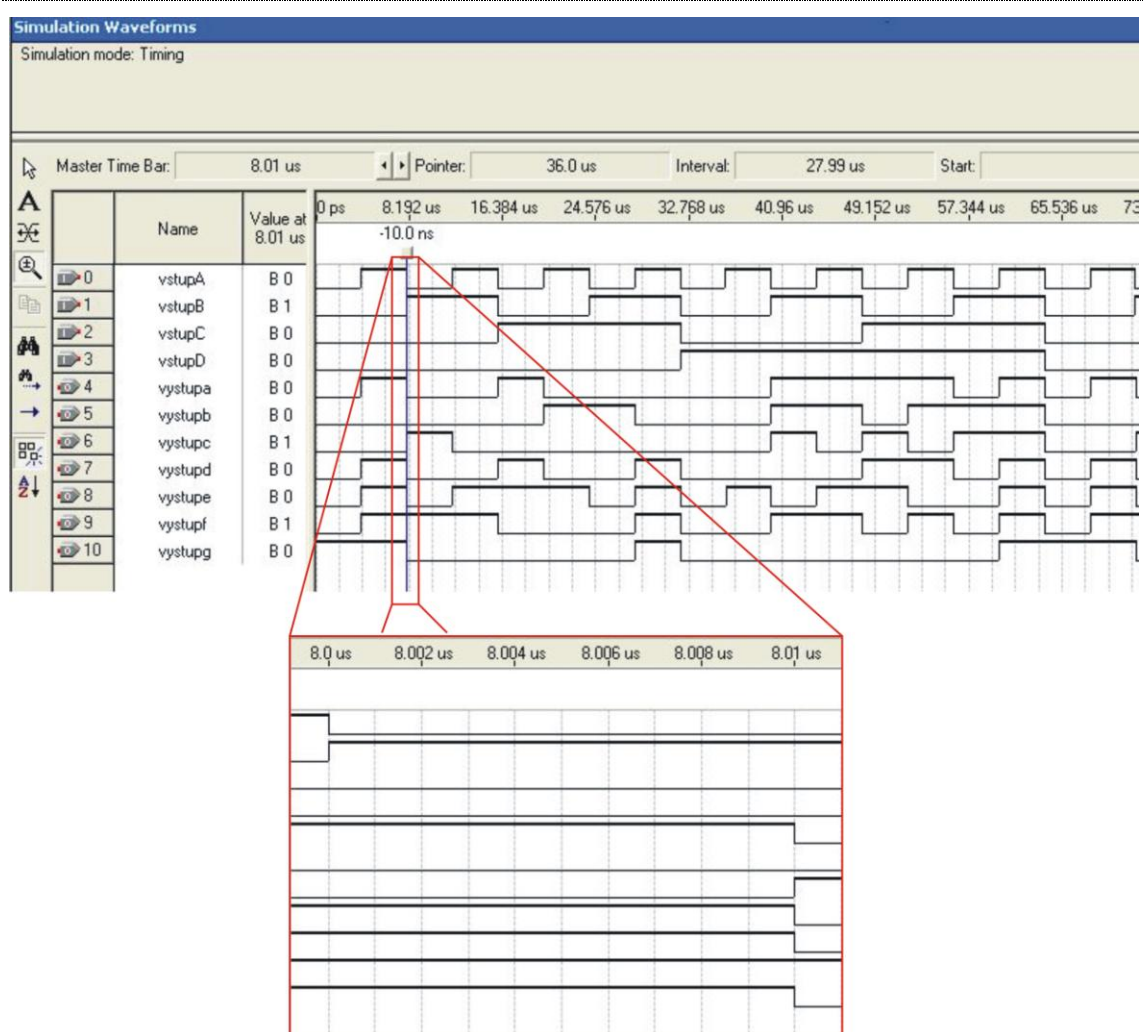


Obr. 23 Schéma zapojenia prevodníka BCD na sedem-segment



Obr. 24 Výsledok funkčnej simulácie pre zapojenie prevodníka BCD na sedem-segment

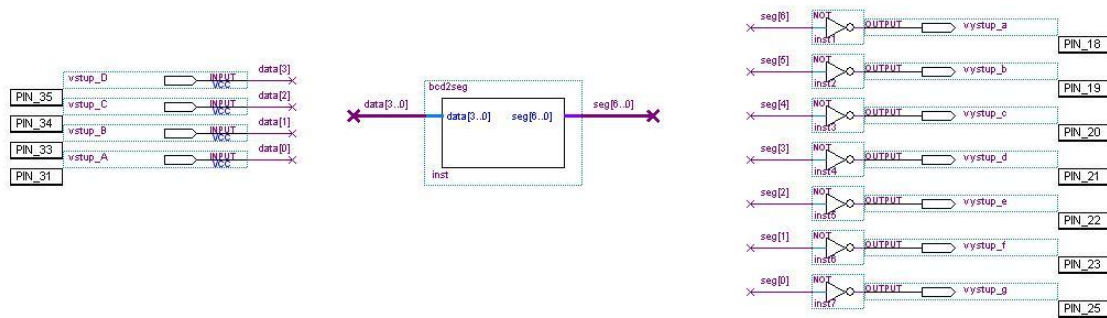
Obr. 24 zobrazuje výsledok funkčnej simulácie so zvýrazneným zobrazením prepínania signálov. Ako vidno funkčná simulácia nepočíta z oneskoreniami takže, zmena signálu na vstupe sa ihneď prejaví na výstupe. Vďaka tomu trvá funkčná simulácia oveľa kratšie ako časová simulácia. V reálnom obvode sa však musí počítať s časovým oneskorením, takže je potrebné vykonať i časovú simuláciu. Časová simulácia je zobrazená so zvýrazneným prepínaním signálu je zobrazená na Obr. 25. Na zvýraznenej časti je vidieť oneskorenie zmeny signálu zo vstupu na výstup obvodu.



Obr. 25 Výsledok časovej simulácie pre prevodník BCD na sedem-segment

3.1.5 Riešenie pomocou VHDL

Riešenie pomocou opisného jazyka VHDL je oveľa efektívnejšie ako grafické riešenie. Riešenie úlohy pozostáva z opisu požadovanej činnosti obvodu. Po napísaní kódu sa môže prísť k vytvoreniu schematickej značky, ktorá sa využije v grafickom editore. Zapojenie schematickej značky je uvedené v schéme zapojenia na Obr. 26.



Obr. 26 Zapojenie prevodníka z BCD na sedem-segment vytvoreného vo VHDL

Ukážka kódu vo VHDL jazyku pre prevodník.

```

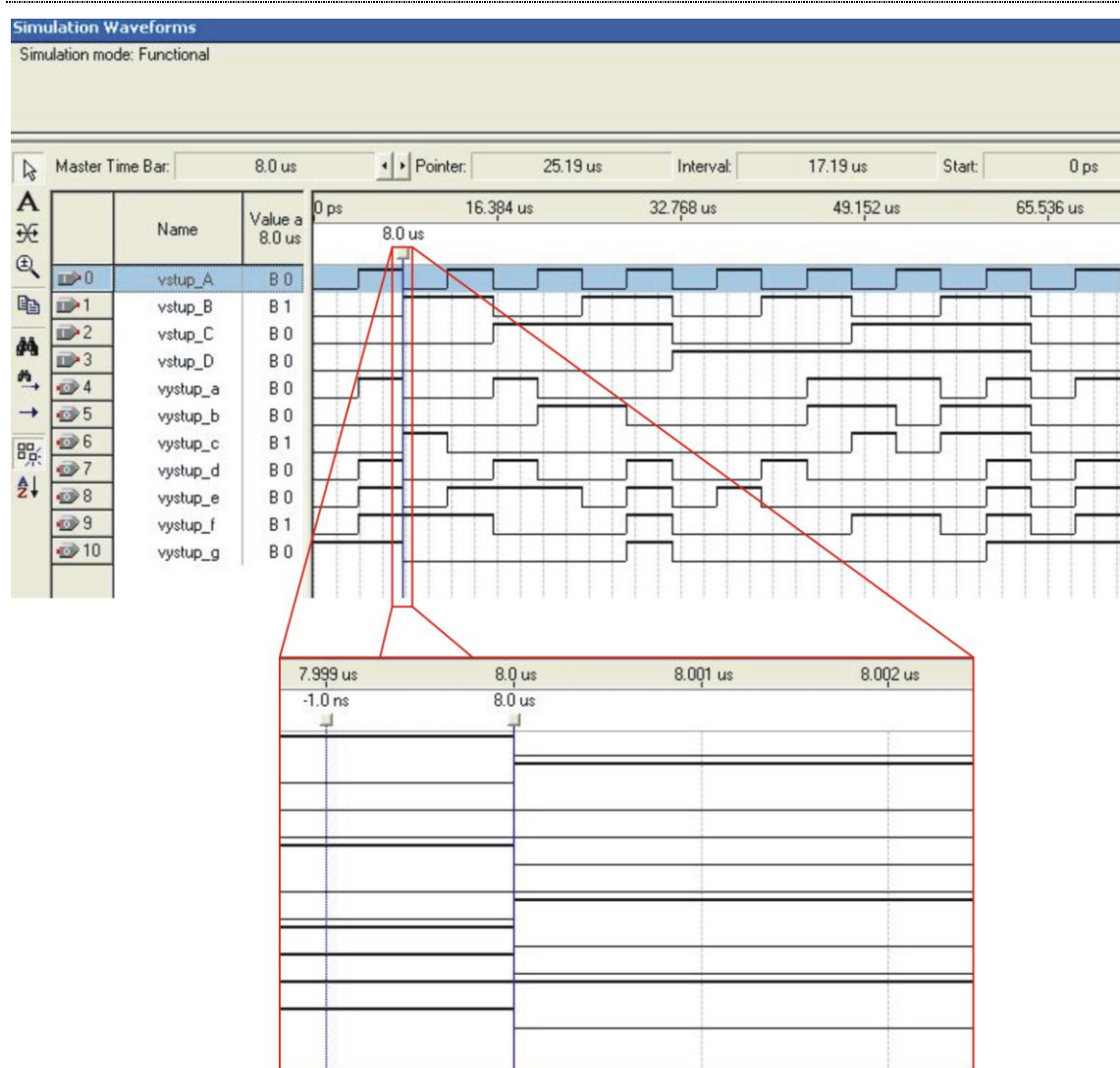
library ieee;                                -- standardna kniznica organizacie IEEE
use ieee.std_logic_1164.all;                 -- kniznica datovych typov

entity bcd2seg is                             -- zadefinovanie entity
port(
  data : in std_logic_vector (3 downto 0);   -- vektor vstupov
  seg : out std_logic_vector (6 downto 0)    -- vektor vystupov
);
end bcd2seg;                                 -- ukoncenie entity

architecture a_segment of bcd2seg is         -- zadefinovanie architektury entity
begin
  process (data) begin                       -- zistovanie vstupneho vektora
    case data is
      when "0000" => seg <= "1111110";     -- 0
      when "0001" => seg <= "0110000";     -- 1
      when "0010" => seg <= "1101101";     -- 2
      when "0011" => seg <= "1111001";     -- 3
      when "0100" => seg <= "0110011";     -- 4
      when "0101" => seg <= "1011011";     -- 5
      when "0110" => seg <= "1011111";     -- 6    priradenie
      when "0111" => seg <= "1110000";     -- 7    vystupnych
      when "1000" => seg <= "1111111";     -- 8    vektorov
      when "1001" => seg <= "1111011";     -- 9
      when "1010" => seg <= "1110111";     -- A
      when "1011" => seg <= "0011111";     -- b
      when "1100" => seg <= "0001101";     -- c
      when "1101" => seg <= "0111101";     -- d
      when "1110" => seg <= "1001111";     -- E
      when others => seg <= "0000000";     --
    end case;
  end process;
end a_segment;

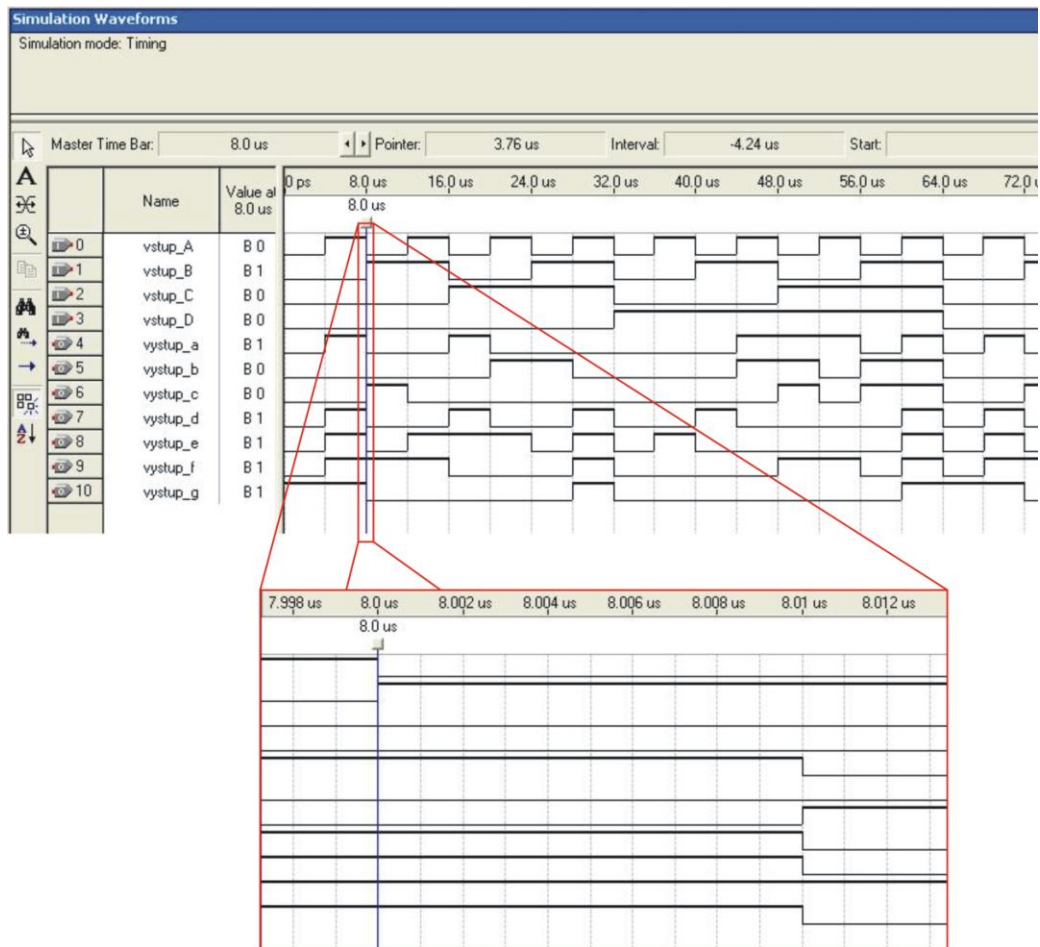
```

Pre porovnanie činnosti sa ešte uvedú výsledky funkčnej (Obr. 27) a časovej simulácie (Obr. 28) pre prevodník kódu vytvoreného pomocou VHDL jazyka.



Obr. 27 Výsledok funkčnej simulácie prevodníka z BCD na sedem-segment vytvoreného vo VHDL

Funkčná simulácia pre prevodník vytvorený vo VHDL je zobrazená na Obr. 27. Časová simulácia pre prevodník vytvorený vo VHDL je zobrazená na Obr. 28.



Obr. 28 Výsledok časovej simulácie prevodníka z BCD na sedem-segment vytvoreného vo VHDL

3.1.6 Výsledok úlohy

Ako je vidieť na časovej a funkčnej simulácii (pri porovnaní s pravdivostnou tabuľkou), úloha je realizovaná úspešne v grafickom i VHDL režime. Touto úlohou sa otestovalo zapojenie sedem-segmentového displeja, pričom grafické riešenie úlohy testuje zapojenie druhej číslice DG2 a zapojenie realizované VHDL kódom testuje prvú číslicu DG1. Ako vstup sa využil DIP prepínač, takže sa tiež otestovalo i zapojenie jednotlivých prepínačov. Pri porovnaní oboch úloh je zreteľne vidieť, že riešenie VHDL kódom je oveľa efektívnejšie. Ak by bolo potrebné zmeniť výstupné funkcie prevodníka musel by sa pri prvom riešení opakovať celý náročný proces syntézy. Pri zmene výstupných funkcií prevodníka pri druhej úlohe je oveľa jednoduchšie, pretože stačí iba upraviť kód a celú syntézu za nás vykoná program Altera Quartus II. Výsledky simulácii vyšli pre oba spôsoby riešenia rovnaké no pri zložitejších úlohách by časové simulácie by mohli byť rôzne. Archivovaný projekt pre obe riešenia a obrázky zo simulácií sú uložené v Prílohe B.

3.2 Demultiplexor

Je to kombinačný logický obvod, ktorý slúži ako elektronický prepínač. V prípade n adresných vstupov má 2^n dátových výstupov a 1 dátový vstup X . Informácia privedená na dátový vstup sa dostane na ten výstup, ktorého adresu určuje kombinácia adresných vstupov. Ostatné výstupy budú mať hodnotu "log 0". Demultiplexory sa využívajú ako generátory časových impulzov, na prevod sériových údajov na paralelné a na výber obvodov pomocou adresných vstupov.

3.2.1 Zadanie úlohy

Realizujte zapojenie demultiplexora z jedným dátovým vstupom a ôsmymi výstupmi. Zapojenie realizujte v grafickom i VHDL režime prostredia Altera Quartus II [22] s kompletnou časovou a funkčnou simuláciou.

3.2.2 Riešenie

Riešenie úlohy bude pozostávať s vytvorenia pravdivostnej tabuľky a zostrojenia obvodu. Pre túto úlohu sa môže vynechať vytváranie karnaughových máp, pretože zapojenie má v danom čase iba jeden výstup, ktorý je presne definovaný adresovými vstupmi.

Keďže zadaná úloha má obsahovať osem výstupom, na zaadresovanie každého výstupu je potrebné mať tri adresové vstupy. Pomocou jednotlivých prepínačov (S5, S6, S7) DIP prepínača (pozri Obr. 14) sa budú zadávať adresy výstupov (A_2 , A_1 , A_0). Vstupné dáta sa zadávajú pomocou prepínača S2. Tieto tlačidlá pracujú v negatívnej logike. Výstupy demultiplexora sa pripoja na LED diódy. Je potrebné pripomenúť zapojenie prepojky JP1 (pozri Obr. 19), aby sa výstup zobrazil na LED diódach (pre zapojenie pinov CPLD obvodu a LED diód pozri Tab. 9). Taktiež je potrebné spomenúť, že LED diódy majú negatívnu logiku a preto je potrebné do riešenia pridať na výstupy NOT hradlá.

3.2.3 Syntéza

Podľa požiadavky úlohy zostrojíme pravdivostnú tabuľku Tab. 12. Ako vyplýva z pravdivostnej tabuľky nie je potrebné zostrojovať karnaughove mapy, čím sa syntéza danej úlohy zjednoduší.

Tab. 12 Pravdivostná tabuľka demultiplexra

Adresné vstupy				Výstup							
X	A2	A1	A0	D8	D7	D6	D5	D4	D3	D2	D1
1	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
Zvyšné kombinácie				sú samé nuly							

Priamo z pravdivostnej tabuľky (Tab. 12) môžeme vypísať funkcie v UNDF tvare.

$$D1 = X \cdot \overline{A_2} \cdot \overline{A_1} \cdot \overline{A_0}$$

$$D2 = X \cdot \overline{A_2} \cdot \overline{A_1} \cdot A_0$$

$$D3 = X \cdot \overline{A_2} \cdot A_1 \cdot \overline{A_0}$$

$$D4 = X \cdot \overline{A_2} \cdot A_1 \cdot A_0$$

$$D5 = X \cdot A_2 \cdot \overline{A_1} \cdot \overline{A_0}$$

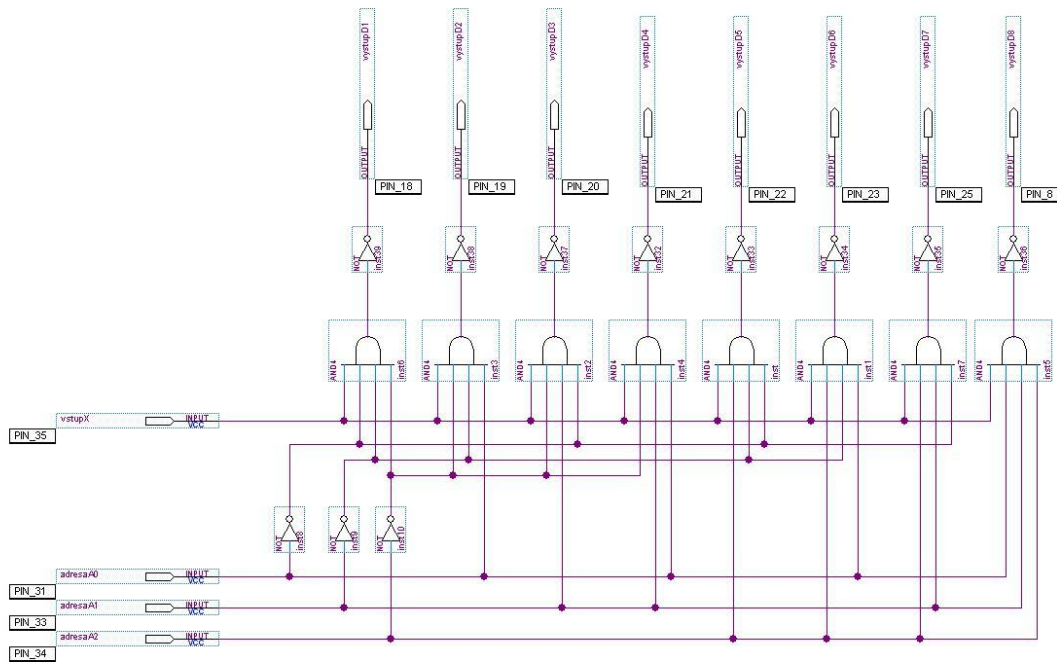
$$D6 = X \cdot A_2 \cdot \overline{A_1} \cdot A_0$$

$$D7 = X \cdot A_2 \cdot A_1 \cdot \overline{A_0}$$

$$D8 = X \cdot A_2 \cdot A_1 \cdot A_0$$

3.2.4 Zapojenie obvodu

Vďaka uvedeným funkciám si po otvorení nového projektu zostrojíme zapojenie ktoré je zobrazené na Obr. 29.



Obr. 29 Schematické zapojenie demultiplexra

Po nakreslení schémy zapojenia obvodu je potrebné priradiť výstupy zapojenia jednotlivým vývodom obvodu, tak ako to zobrazuje Obr. 30.

	From	To	Assignment Name	Value	Enabled
1		adresaA0	Location	PIN_31	Yes
2		adresaA1	Location	PIN_33	Yes
3		TCK	Location	PIN_26	Yes
4		TD1	Location	PIN_1	Yes
5		TDO	Location	PIN_32	Yes
6		TMS	Location	PIN_7	Yes
7		adresaA2	Location	PIN_34	Yes
8		vystupD1	Location	PIN_18	Yes
9		vystupD2	Location	PIN_19	Yes
10		vystupD3	Location	PIN_20	Yes
11		vystupD4	Location	PIN_21	Yes
12		vystupD5	Location	PIN_22	Yes
13		vystupD6	Location	PIN_23	Yes
14		vystupD7	Location	PIN_25	Yes
15		vystupD8	Location	PIN_8	Yes
16		vstupX	Location	PIN_35	Yes
17	<<new>>	<<new>>	<<new>>		

Obr. 30 Priradenie pinov v projekte demultiplexor

Následne sa môže prístupit' ku kompilácii zapojenia a ak všetko prebehlo bez chýb je čas prístupit' k funkčnej a časovej simulácii. Z dôsledku veľkosti obrázkov sú výstupy funkčnej a časovej simulácie zobrazené v Prílohe B1 a v Prílohe B2.

3.2.5 Riešenie pomocou VHDL

Po napísaní kódu opisujúceho funkciu zadanej úlohy vytvoríme schematickú značku, ktorej zapojenie je zobrazené na Obr. 31. Pred kompiláciou sa ešte prideli jednotlivým výstupom a vstupom príslušné číslo pinov (pozri Prílohu B3) a môže sa prejsť ku kompilácii.

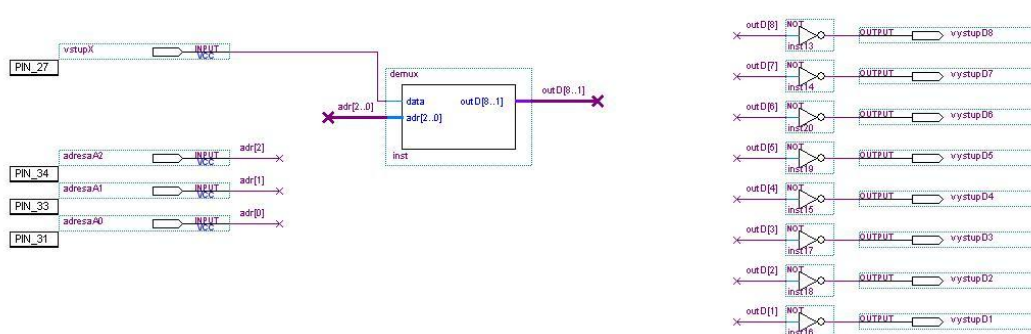
```

library ieee;
use ieee.std_logic_1164.all;

entity demux is
port(
  data : in std_logic;
  adr : in std_logic_vector (2 downto 0);
  outD : out std_logic_vector (8 downto 1)
);
end demux;

architecture demux_arch of demux is
begin
  process (adr, data) begin
    if data = '1' then
      case adr is
        when "000" => outD[0] <= "00000001"; -- D1
        when "001" => outD[1] <= "00000010"; -- D2
        when "010" => outD[2] <= "00000100"; -- D3
        when "011" => outD[3] <= "00001000"; -- D4
        when "100" => outD[4] <= "00010000"; -- D5
        when "101" => outD[5] <= "00100000"; -- D6
        when "110" => outD[6] <= "01000000"; -- D7
        when "111" => outD[7] <= "10000000"; -- D8
        when others => outD[0:7] <= "00000000"; --
      end case;
    else outD[0:7] <= "00000000";
    end if;
  end process;
end demux_arch;

```



Obr. 31 Zapojenie demultiplexra vytvoreného vo VHDL

	From	To	Assignment Name	Value	Enabled
1		adresaA0	Location	PIN_31	Yes
2		adresaA1	Location	PIN_33	Yes
3		adresaA2	Location	PIN_34	Yes
4		TCK	Location	PIN_26	Yes
5		TDI	Location	PIN_1	Yes
6		TDO	Location	PIN_32	Yes
7		TMS	Location	PIN_7	Yes
8		vstupX	Location	PIN_27	Yes
9		vystupD1	Location	PIN_18	Yes
10		vystupD2	Location	PIN_19	Yes
11		vystupD3	Location	PIN_20	Yes
12		vystupD4	Location	PIN_21	Yes
13		vystupD5	Location	PIN_22	Yes
14		vystupD6	Location	PIN_23	Yes
15		vystupD7	Location	PIN_25	Yes
16		vystupD8	Location	PIN_8	Yes
17	<<new>>	<<new>>	<<new>>		

Obr. 32 Zapojenie vývodov demultiplexra riešeného pomocou VHDL k jednotlivým pinom CPLD obvodu

Po úspešnej kompilácii sa vykoná funkčná a časová simulácia. Výsledky funkčnej a časovej simulácie sú zobrazené v Prílohe B3 a v Prílohe B4.

3.2.6 Výsledok úlohy

Po porovnaní simulácií s pravdivostnou tabuľkou sa overila funkčnosť oboch riešení.

Úloha bola realizovaná a odskúšaná na navrhutej vývojovej doske. Vďaka tejto úlohe sa otestovali zapojenia jednotlivých LED diód. Znova sa ukázala výhoda riešiť úlohu pomocou opisného jazyka VHDL, aj keď v praxi sa využíva taktiež riešenie grafické, kde sa zložitá úloha rozčlení na jednoduchšie časti, ktoré sa popíšu pomocou VHDL jazyka. Z týchto častí sa vytvoria schematické značky, ktoré sa v grafickom editore spájajú do výsledného zložitejšieho zapojenia.

3.3 Multiplexor

Je to kombinačný logický obvod, ktorý slúži ako elektronický prepínač. V prípade n adresných vstupov má 2^n údajových vstupov a 1 výstup. Na údajový výstup sa prenesú údaje z toho vstupu, ktorého adresu určuje kombinácia adresných vstupov. Multiplexory sa využívajú ako generátory časových impulzov, na prevod paralelných údajov na sériové a na výber obvodov pomocou adresy na adresných vstupoch.

3.3.1 Zadanie úlohy

Realizujte zapojenie multiplexora zo štyrmi dátovými vstupmi a jedným dátovým výstupom. Zapojenie realizujte v grafickom i VHDL režime prostredia Altera Quartus II s kompletnou časovou a funkčnou simuláciou.

3.3.2 Riešenie

Riešenie pozostáva zo zostrojenia pravdivostnej tabuľky, ktorá obsahuje vstupné adresné dáta a požadovaný výstup. Keďže je potrebné adresovať štyri dátové vstupy postačia na adresovanie dva adresné vstupy, ktorých hodnoty sa budú zadávať pomocou DIP prepínača a to prepínačmi $S6 = A1$ a $S7 = A0$ (pozri Obr. 14). Dátové vstupy sa odoberú z výstupov deličky. Jeden výstup z deličky sa privedie na výstupný pin (pin 14) CPLD obvodu a pomocou prepojky JP4 sa privedie na vstupný pin (GCLK2) CPLD obvodu. Je nutné pripomenúť na prepojenie prepojky JP3, ktorá umožňuje zapnutie a vypnutie oscilátora. Pri nezapojení tejto prepojky nebude zapojenie fungovať pri žiadnej kombinácii adresných vstupov. Výstup z multiplexora sa pripojí na vstup bzučiaka. Kód deličky sa nachádza v Prílohe C4.

3.3.3 Syntéza

Pre danú úlohu sa zostrojí pravdivostná tabuľka Tab. 13.

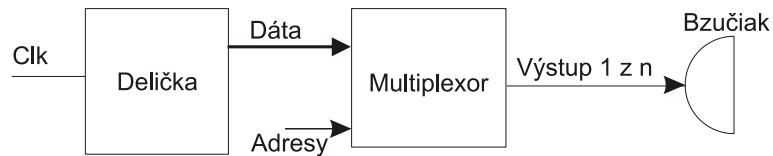
Tab. 13 Pravdivostná tabuľka 4-vstupového multiplexora

Adresné vstupy		Výstup Z
A1	A0	
0	0	Data1
0	1	Data2
1	0	Data3
1	1	Data4

Z pravdivostnej tabuľky sa získa výstupná funkcia Z.

$$Z = D_1 \cdot \overline{A_1} \cdot \overline{A_0} + D_2 \cdot \overline{A_1} \cdot A_0 + D_3 \cdot A_1 \cdot \overline{A_0} + D_4 \cdot A_1 \cdot A_0$$

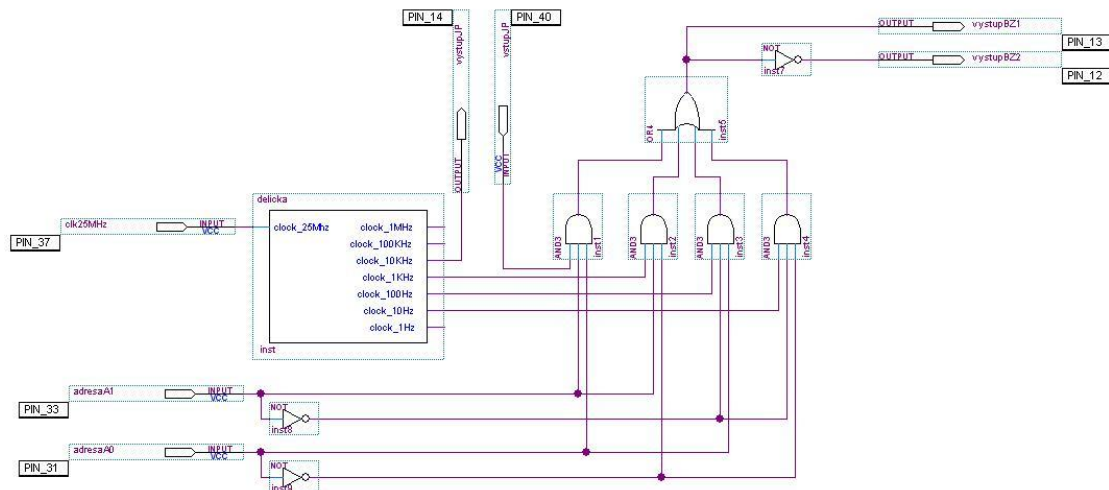
Na základe tejto funkcie sa navrhne obvod, ktorý danú funkciu realizuje. Bloková schéma takéhoto návrhu je zobrazená na Obr. 33.



Obr. 33 Bloková schéma zapojenia multiplexora

Prvý blok z názvom Delička má za úlohu vydeliť vstupnú frekvenciu 25MHz na frekvencie počuteľné človekom. Blok multiplexor umožní výber jednej zo štyroch frekvencií, ktoré sa privedú na vstupy bzučiaka.

Delička frekvencie je realizovaná pomocou opisného jazyka VHDL Kód deličky je uvedený v Prílohe C4. Multiplexor je realizovaný pomocou hradíel zapojených podľa výstupnej funkcie. Celková schéma je zobrazená na Obr. 34.

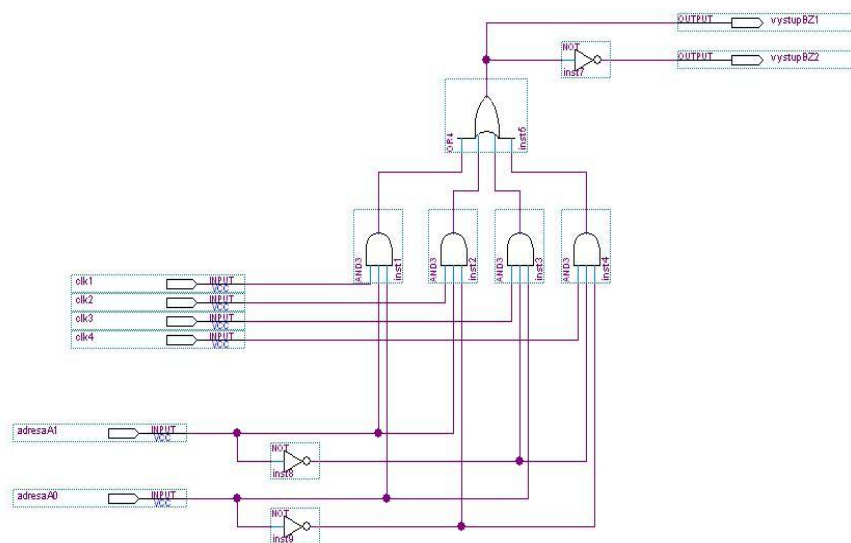


Obr. 34 Zapojenie Multiplexora

Vstupom a výstupom sa priradia jednotlivé vývody tak, ako to zobrazuje Obr. 35. A môže sa pristúpiť ku kompilácii zapojenia. Po úspešnej kompilácii je na rade časová a funkčná simulácia. Pre toto zapojenie by však simulácia trvala veľmi dlho a preto sa musí zapojenie upraviť odobratím deličky zo zapojenia. Príklad zapojenia pripraveného na simuláciu zobrazuje Obr. 36.

	From	To	Assignment Name	Value	Enabled
1		adresaA0	Location	PIN_31	Yes
2		adresaA1	Location	PIN_33	Yes
3		clk25MHz	Location	PIN_37	Yes
4		TCK	Location	PIN_26	Yes
5		TDI	Location	PIN_1	Yes
6		TDO	Location	PIN_32	Yes
7		TMS	Location	PIN_7	Yes
8		vstupJP	Location	PIN_40	Yes
9		vystupBZ2	Location	PIN_12	Yes
10		vystupJP	Location	PIN_14	Yes
11		vystupBZ1	Location	PIN_13	Yes
12	<<new>>	<<new>>	<<new>>		

Obr. 35 Zapojenie vstupov a výstupov multiplexra k jednotlivými vývodmi CPLD obvodu.



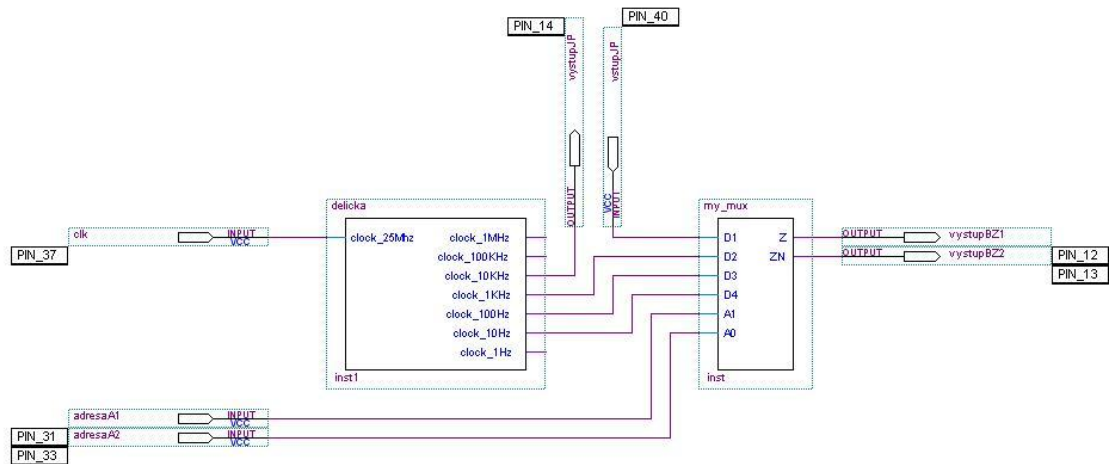
Obr. 36 Zapojenie multiplexra pre simulácie

Po takejto úprave zapojenia sa môže prejsť k simulácii. Výsledky funkčnej simulácie zobrazuje Príloha B5 a časovej simulácie zobrazuje Príloha B6.

3.3.4 Riešenie pomocou VHDL

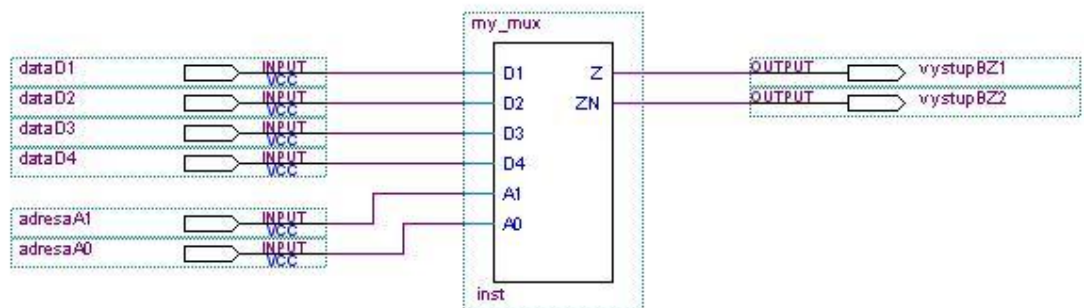
Riešenie pomocou VHDL bude veľmi podobné blokovej schéme na Obr. 33.

Pomocou VHDL kódu sa vytvorí blok multiplexra. Tento blok sa zapojí tak ako to zobrazuje Obr. 37.



Obr. 37 Zapojenie multiplexra riešeného pomocou jazyka VHDL.

Aj v tomto zapojení sa nachádza delička a preto je pre simuláciu opäť potrebné odstrániť tento blok. Ako vidieť na Obr. 38 zapojenie sa značne zjednoduší. Po takejto úprave zostáva ešte priradiť jednotlivým vstupom a výstupom vývody z CPLD obvodu a môže sa prejsť k simulácii.



Obr. 38 Upravené zapojenie multiplexra riešeného pomocou VHDL pre simuláciu

	From	To	Assignment Name	Value	Enabled
1		adresaA1	Location	PIN_31	Yes
2		adresaA2	Location	PIN_33	Yes
3		clk	Location	PIN_37	Yes
4		TCK	Location	PIN_26	Yes
5		TDI	Location	PIN_1	Yes
6		TDO	Location	PIN_32	Yes
7		TMS	Location	PIN_7	Yes
8		vstupJP	Location	PIN_40	Yes
9		vystupBZ1	Location	PIN_12	Yes
10		vystupBZ2	Location	PIN_13	Yes
11		vystupJP	Location	PIN_14	Yes
12	<<new>>	<<new>>	<<new>>		

Obr. 39 Zapojenie vstupov a výstupov multiplexra k jednotlivým vývodom CPLD obvodu

Výsledky simulácii sú zobrazené v Prílohe B7 (funkčná simulácia) a v Prílohe B8 (časová simulácia).

3.3.5 Výsledok úlohy

Úloha mala overiť zapojenie oscilátora a bzučiaka na CPLD start kite. Po skontrolovaní simulácii sa oba riešenia otestovali na prípravku. Otestovali sa všetky možné kombinácie adres vstupov, zapnutie a vypnutie oscilátora a taktiež zapojenie prepojkyJP4. Jednotlivé komponenty prípravku fungovali správne.

4 Záver

Úlohy zadania tejto diplomovej práce boli splnené. Úlohou bolo navrhnuť a technicky realizovať prípravok na precvičovanie základných kombinačných a sekvenčných obvodov pomocou obvodu Altera EPM3064A. Kritériom návrhu bolo navrhnuť prípravok čo najmenších rozmerov s minimom externých zariadení a s čo najnižšou koncovou cenou. To bolo rozhodujúce pre výber súčiastok a návrh obvodu. Tento prípravok sa podarilo úspešne realizovať s čo najmenším počtom externých zariadení. Externé zariadenia, ktoré prípravok využíva sú najnutnejšie. Jedná sa o konfiguračný adaptér USB Blaster JTAG vyvinutý na KEMT FEI TU v Košiciach a o zariadenie napájania, ktoré pozostáva iba z USB kábla. Nevýhoda tohto riešenia pozostáva v tom, že používateľ využíva až dva USB porty súčasne. Obsadenie dvoch portov je však ekonomicky i rozmerovo prijateľnejšie ako využitie externého napájacieho transformátora.

Prípravok bol otestovaný testovacími projektmi. Tie boli navrhnuté tak, aby otestovali všetky komponenty nachádzajúce sa na prípravku. Testovacie príklady boli realizované v grafickom aj VHDL režime prostredia Altera Quartus. Všetky testovacie príklady sú k dispozícii v prílohe kde sú umiestnené kompletne archivácie projektov pre oba riešenia spolu s funkčnou a časovou simuláciou. Vďaka tomu sa môžu využiť aj ako demonštračné príklady, pri prvom stretnutí používateľa s prípravkom. Výsledky testov potvrdili, že prípravok je realizovaný úspešne.

Do budúcnosti by bolo vhodné rozšíriť prípravok o viac programových projektov s funkčnou a časovou simuláciou v grafickom i VHDL režime prostredia Altera Quartus.

Zoznam použitej literatúry

- [1] Configuring Altera FPGAs, Altera Corporation, Rev.1.0, 8/2005, 8 s.
www.altera.com
- [2] Configuration File Formats, Altera Corporation, Rev. v2.2, 4/2007, 8 s.
www.altera.com
- [3] ByteBlaster II Download Cable, Altera Corporation, Rev. 1.3, 4/2008, P.N. UG-BBII81204-1.3, www.altera.com
- [4] USB Blaster Download Cable User Guide, Altera Corporation, Rev. 2.4, 4/2008, P.N. UG-USB81204-2.4, www.altera.com
- [5] ByteBlasterMV Download Cable User Guide, Altera Corporation, Rev. 1.0, 8/2004, P.N. UG-BBMV81204-1.0, www.altera.com
- [6] EthernetBlaster Communications Cable User Guide, Altera Corporation, Rev. 1.0, 12/2004, P.N. UG-120904-1.0, www.altera.com
- [7] MasterBlaster Serial/USB Communications Cable, Altera Corporation, Rev. 1.0, 4/2004, P.N. UG-MB81204-1.0, www.altera.com
- [8] MAX 3000A Programmable Logic Device Family Data Sheet, Altera Corporation, Rev. 3.5, 7/2006, P.N. DS-MAX3000A-3.5, www.altera.com
- [9] www.altera.com
- [10] Very low drop voltage regulators with inhibit, STMicroelectronics, Rev. 21, 2/2008, 49 s. lf33c.pdf, www.st.com
- [11] Altera Product Catalog 2007, Altera Corporation, 2007, product-catalog.pdf.,60 s.
- [12] Altera Enhanced configuration Devices, Altera Corporation, Rev. 2.3, 4/2007, 22 s., www.altera.com
- [13] www.elcom.sk
- [14] www.kemt.fei.tuke.sk
- [15] Raychem Circuit Protection Products, Tyco Electronics, Rev. D, 8/2007, 1 s.
www.tycoelectronics.com
- [16] SMD CHIP LED LAMP, Kingbright Elec. Co., Rev. V.10, 7/2007, 4 s.,
www.kingbright.com
- [17] High Stability Oscillator with Stop function, Jauch Rev. 3, 8/2007, 2 s.,
www.jauch.de
- [18] Dual digit numeric display, Kingbright Elec. Co., Rev. V.13, 5/2007, 4 s.,
www.kingbright.com
- [19] Fast Si-Rectifiers, Diodec Semiconductor AG, Rev. 1, 8/2005, 2 s.,
www.diodec.com

-
- [20] General purpose chip resistors RC 206, Yageo Corporation, Rev. 2, 9/2004, 10 s., www.yageo.com
 - [21] Pinker, Jiří – Poupa, Martin: Číslicové systémy a jazyk VHDL. Praha: BEN, 2006, 352 s., ISBN 80-7300-198-5
 - [22] Introduction to the Quartus II software, Altera Corporation, Rev. 7.2, 10/2007, 268 s., www.altera.com
 - [23] FT232R USB UART IC Data Sheet, Future Technology Devices International Ltd., Rev. 1.04, 7/2005, 29 s., www.ftdichip.com
 - [24] In-System Programmability in MAX Devices, Altera Corporation, Rev. 1.5, 9/2005, 11 s., www.altera.com

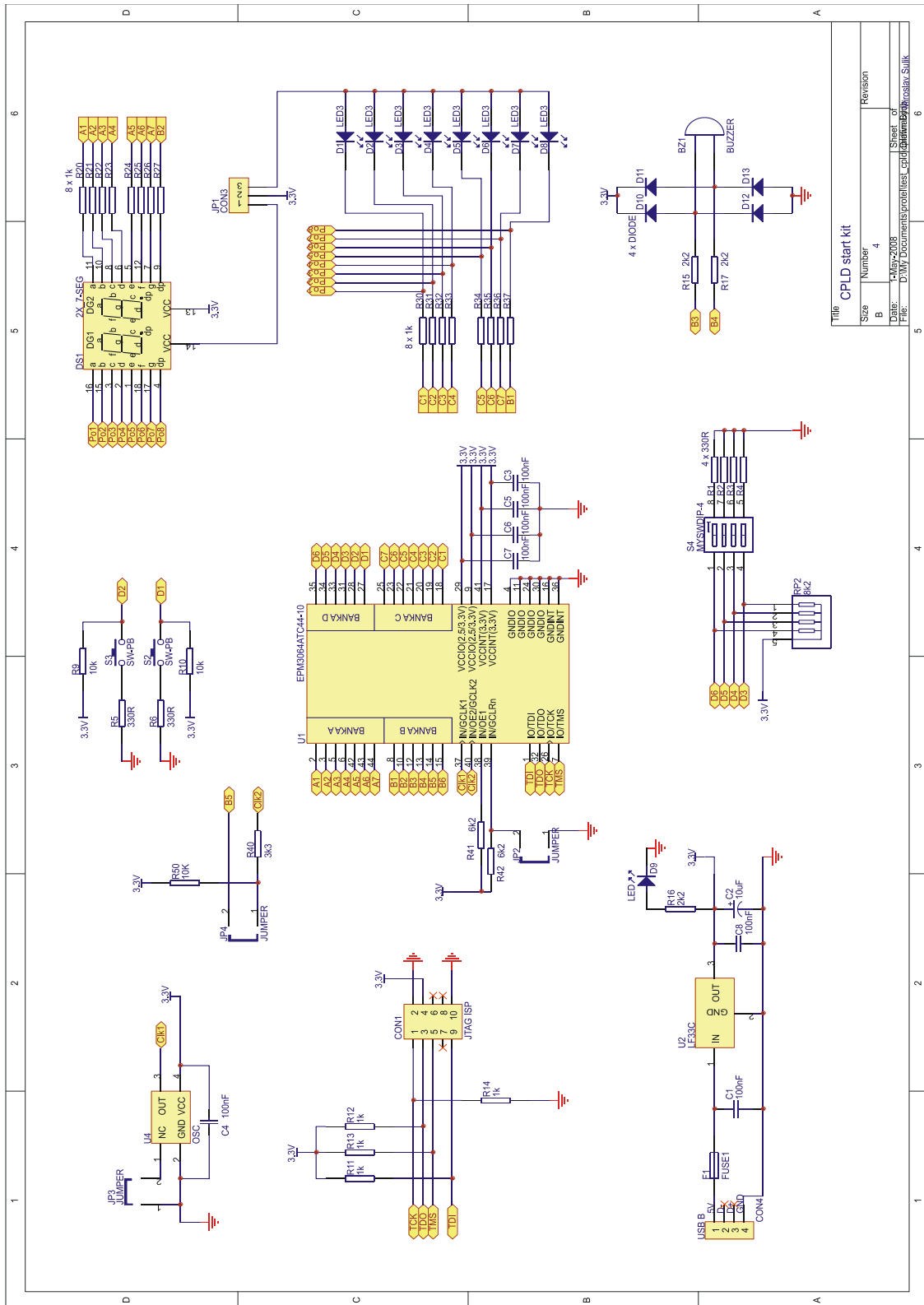
Prílohy

- Príloha A: CPLD start kit
- Schéma CPLD start kit
 - CPLD start kit zo strany súčiastok
 - CPLD start kit zo strany spojov
 - Zapojenie vývodov CPLD obvodu k jednotlivým komponentom
- Príloha B: Výsledky simulácii
- Demultiplexra
 - Multiplexra
- Príloha C: VHDL kódy
- Prevodník
 - Multiplexer
 - Demultiplexer
- Príloha D: CD médium – diplomová práca v elektronickej podobe, prílohy v elektronickej podobe.
- Gerber dáta
 - Protel dáta
 - Schema
 - Doska plošných spojov
 - Quartus projekty
 - Prevodník BCD na sedem-segment
 - Demultiplexor
 - Multiplexor

PRILOHA A

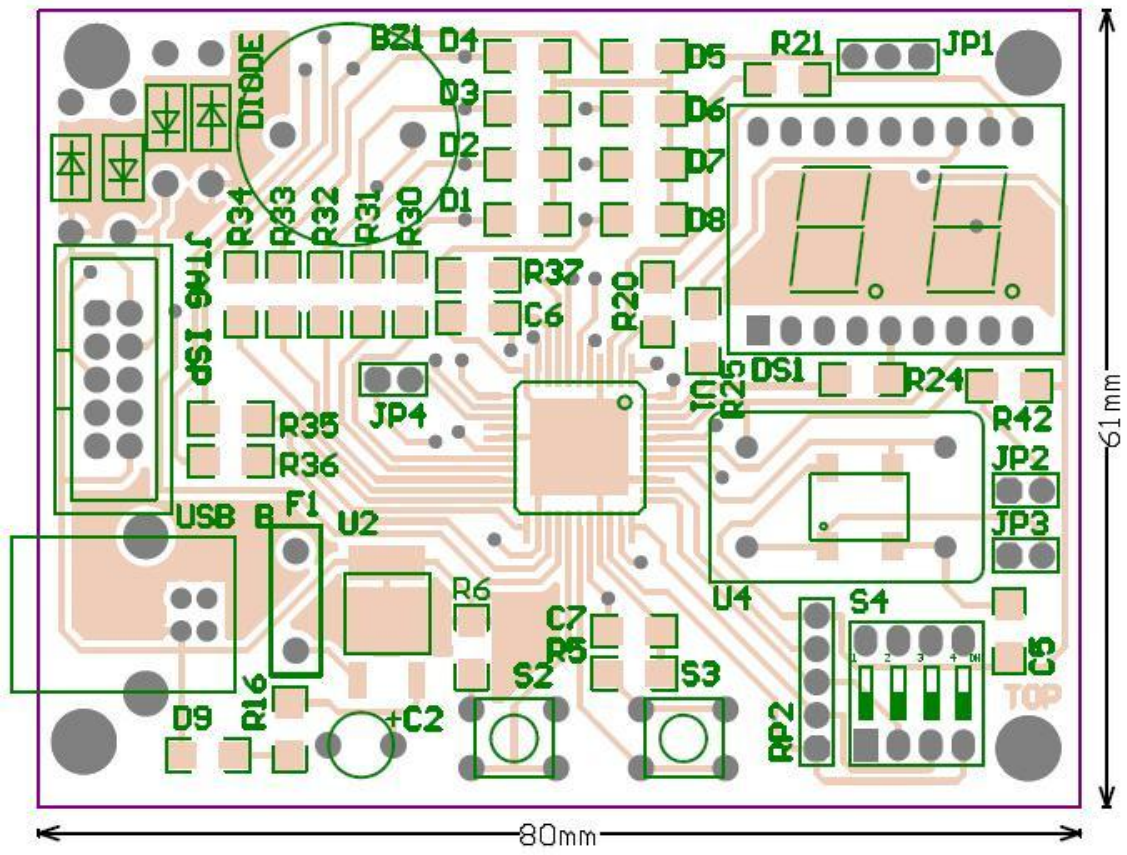
CPLD start kit

A1.Schéma CPLD start kit

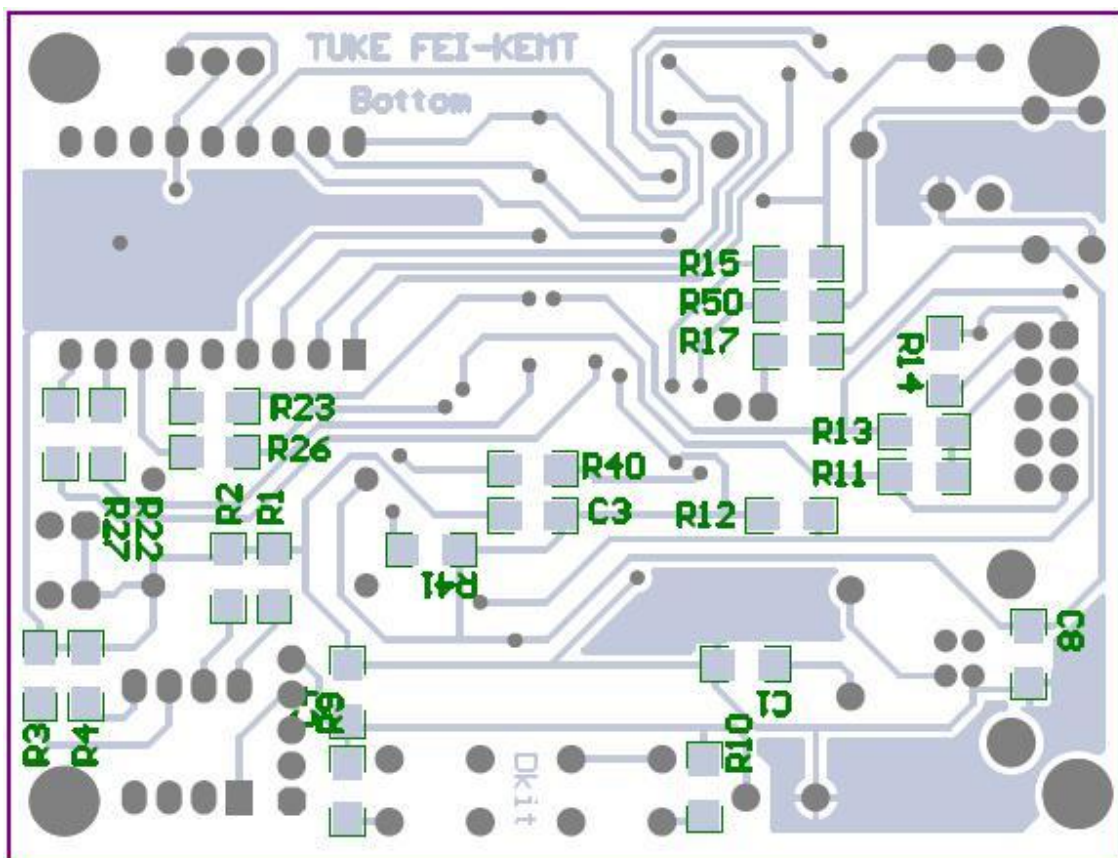


Title		CPLD start kit	
Size	Number	Revision	
B	4		
Date:	1-May-2008	Sheet of	6
File: D:\MY Documents\projet\test_cpld\cpld\startkit\cpld_startkit.sch			

A2.CPLD start kit zo strany súčiastok



A3.CPLD start kit zo strany spojov

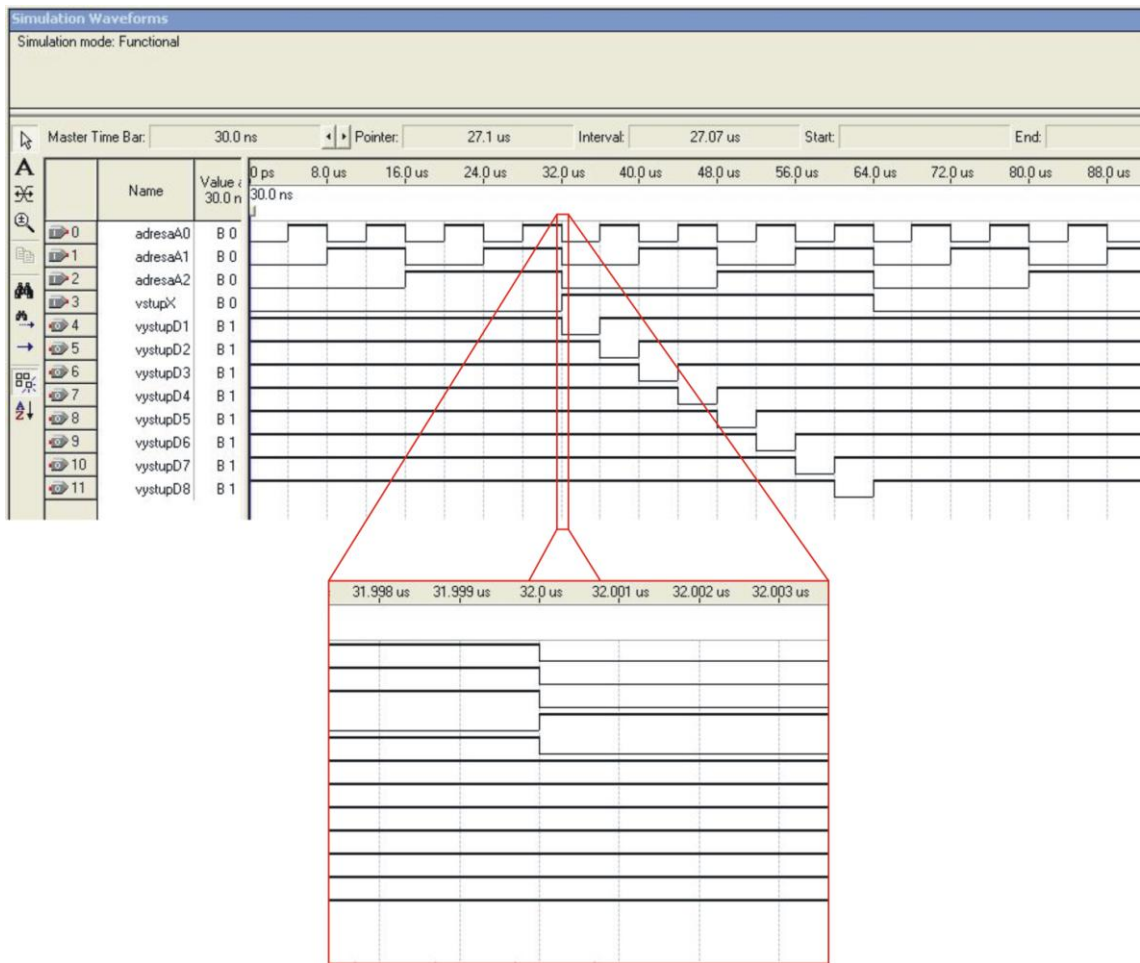


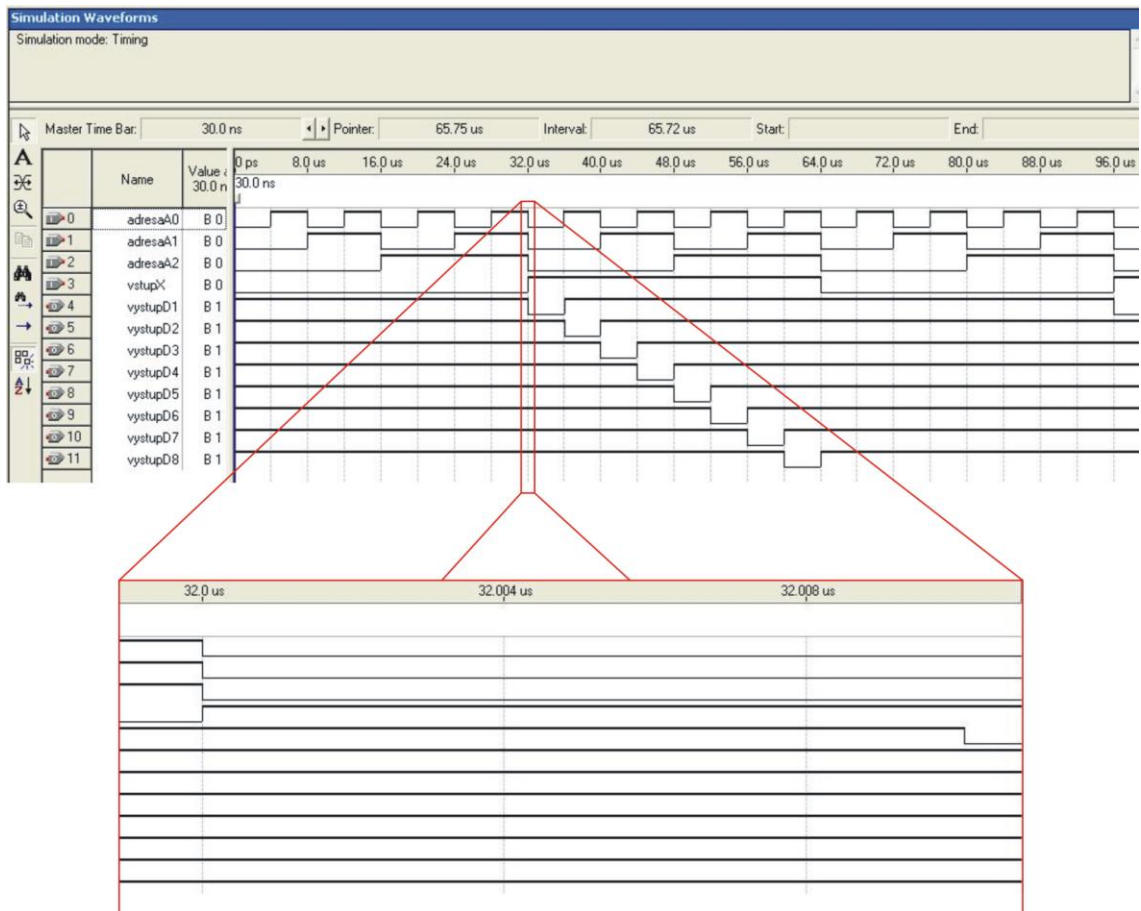
A4.Zapojenie vývodov CPLD obvodu k jednotlivým komponentom

Číslo vývodu CPLD EPM3064	Pripojený komponent	Číslo vývodu CPLD EPM3064	Pripojený komponent
1	TDI (JTAG)	23	DS1 DG1 f
2	DS1-DG2 a	24	GNDIO
3	DS1-DG2 b	25	DS1 DG1 g
4	GNDIO	26	TCK (JTAG)
5	DS1-DG2 c	27	S2
6	DS1-DG2 d	28	S3
7	TMS (JTAG)	29	VCCIO
8	D8, DS1 DG1 dp	30	GNDIO
9	VCCIO	31	S7 (DIP switch)
10	DS1-DG2 dp	32	TDO (JTAG)
11	GNDIO	33	S6 (DIP switch)
12	BZ1	34	S5 (DIP switch)
13	BZ1	35	S4 (DIP switch)
14	JP4 - (GCLK2)	36	GNDINT
15	–	37	U4 (oscilator)
16	GNDINT	38	OE1
17	VCCINT	39	JP2 (GCLRn)
18	DS1 DG1 a	40	JP4
19	DS1 DG1 b	41	VCCINT
20	DS1 DG1 c	42	DS1-DG2 e
21	DS1 DG1 d	43	DS1-DG2 f
22	DS1 DG1 e	44	DS1-DG2 g

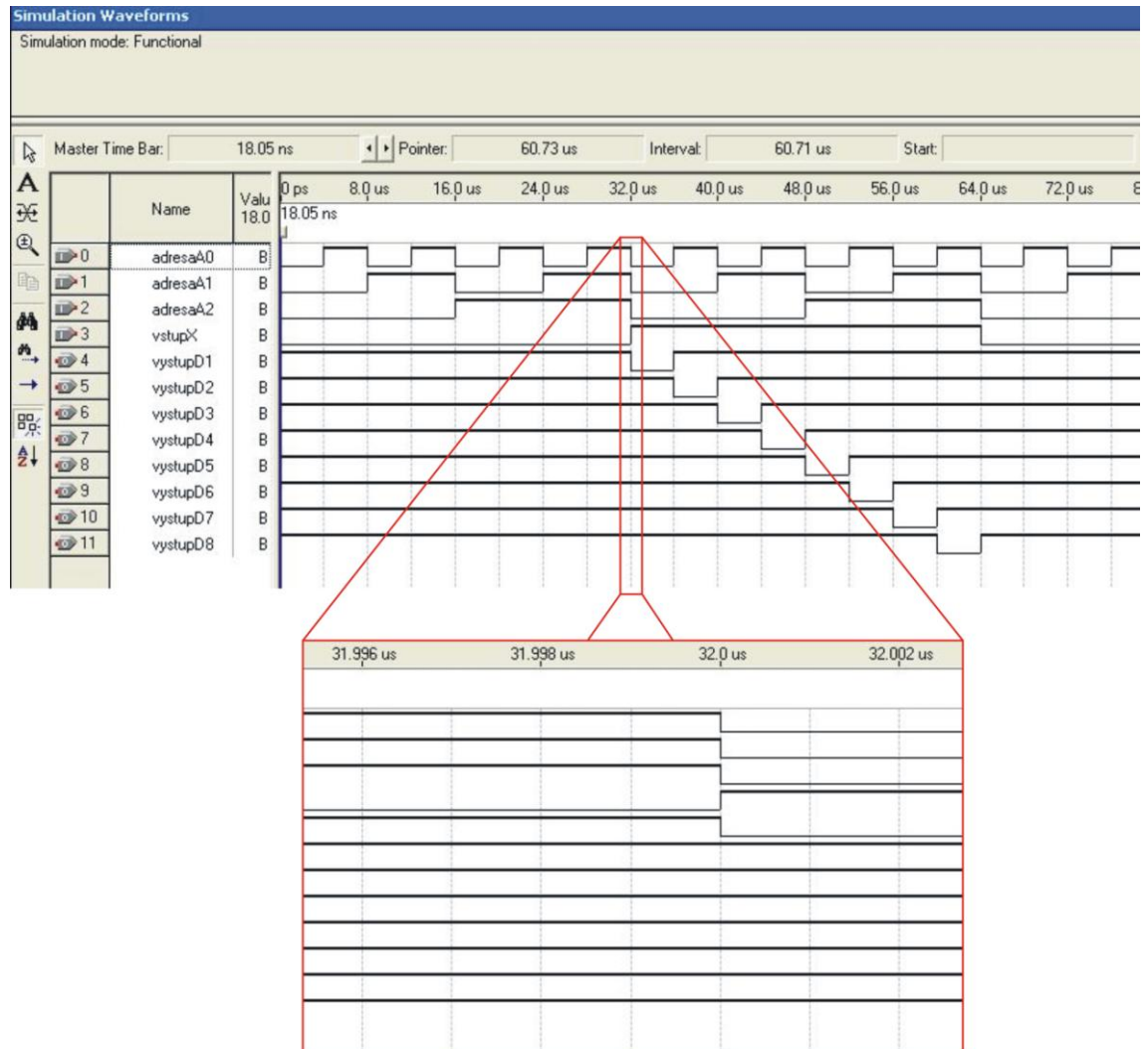
PRÍLOHA B
Výsledky simulácií

B1. Výsledok funkčnej simulácie demultiplexra riešeného syntézou

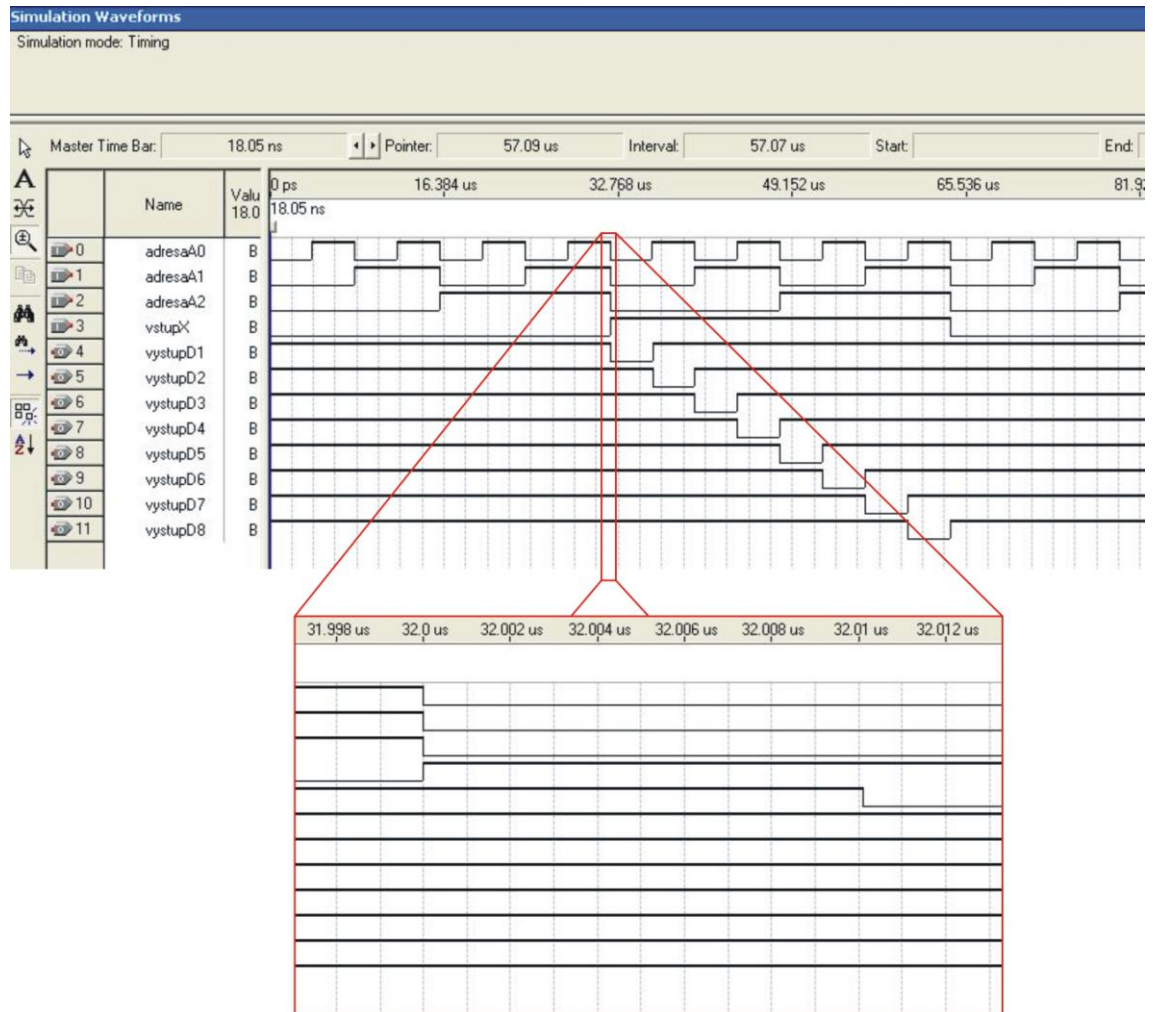


B2. Výsledok časovej simulácie demultiplexra riešeného syntézou

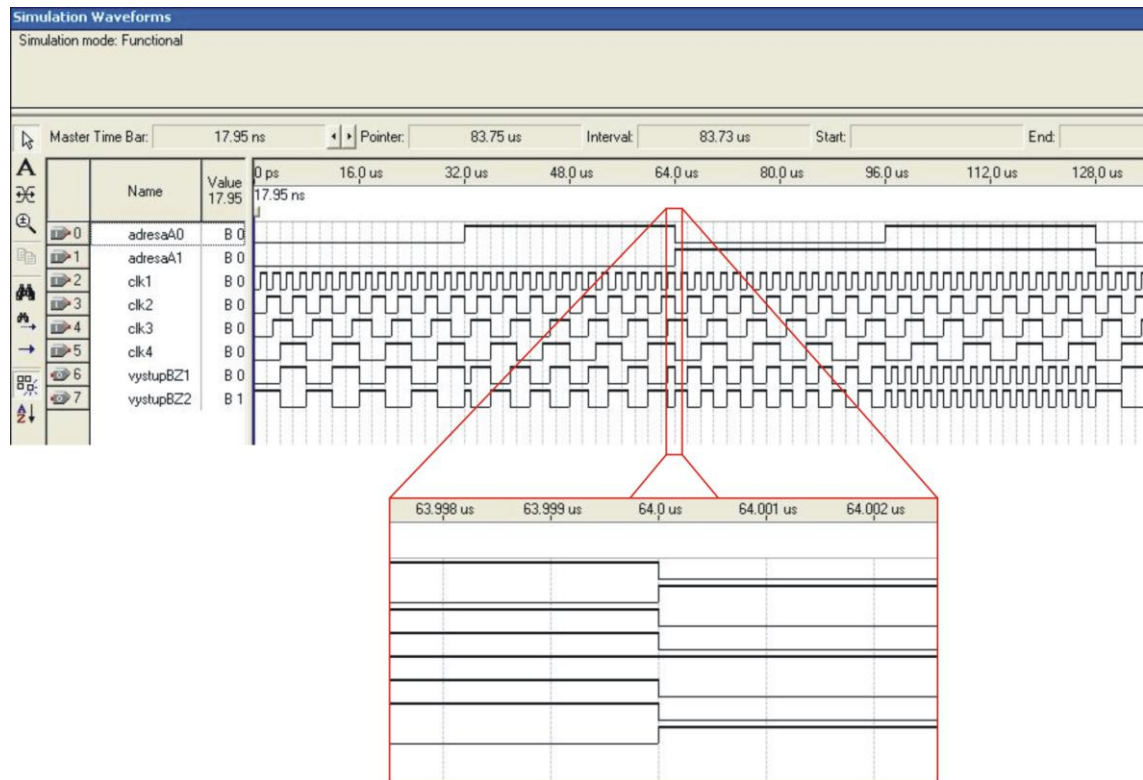
B3. Výsledok funkčnej simulácie demultiplexra riešeného pomocou opisného jazyka VHDL

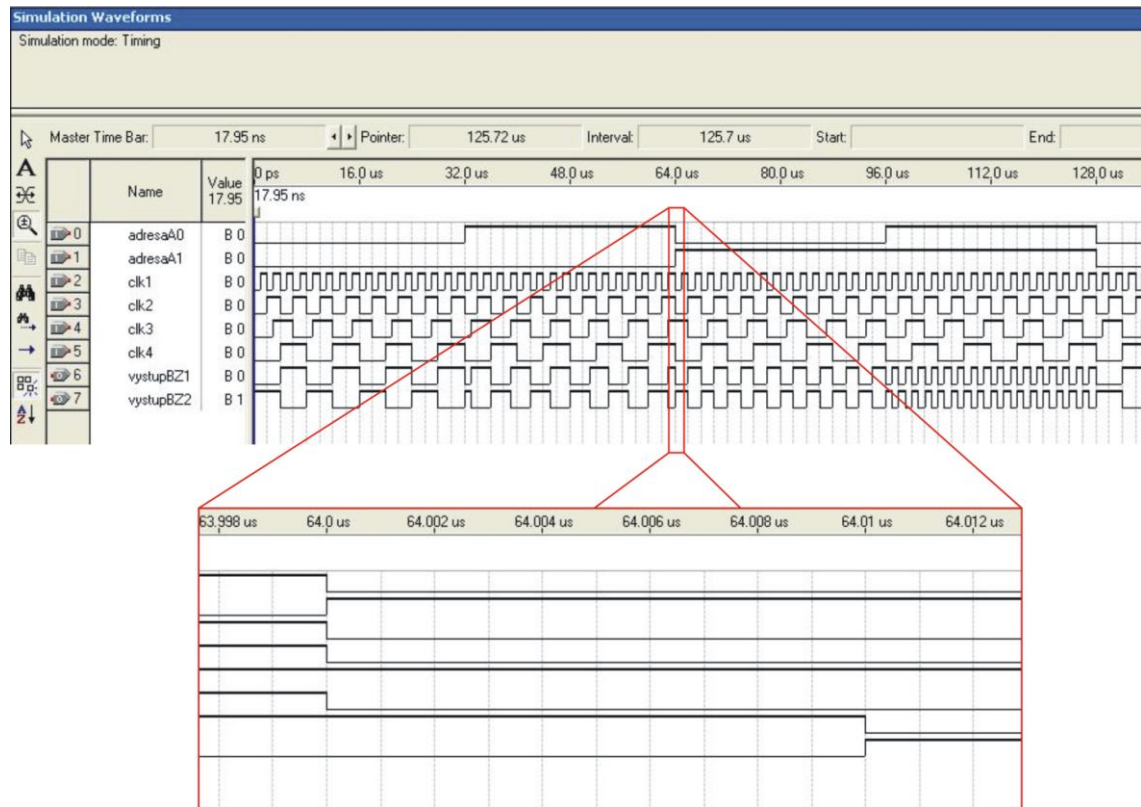


B4. Výsledok časovej simulácie demultiplexra riešeného pomocou opisného jazyka VHDL

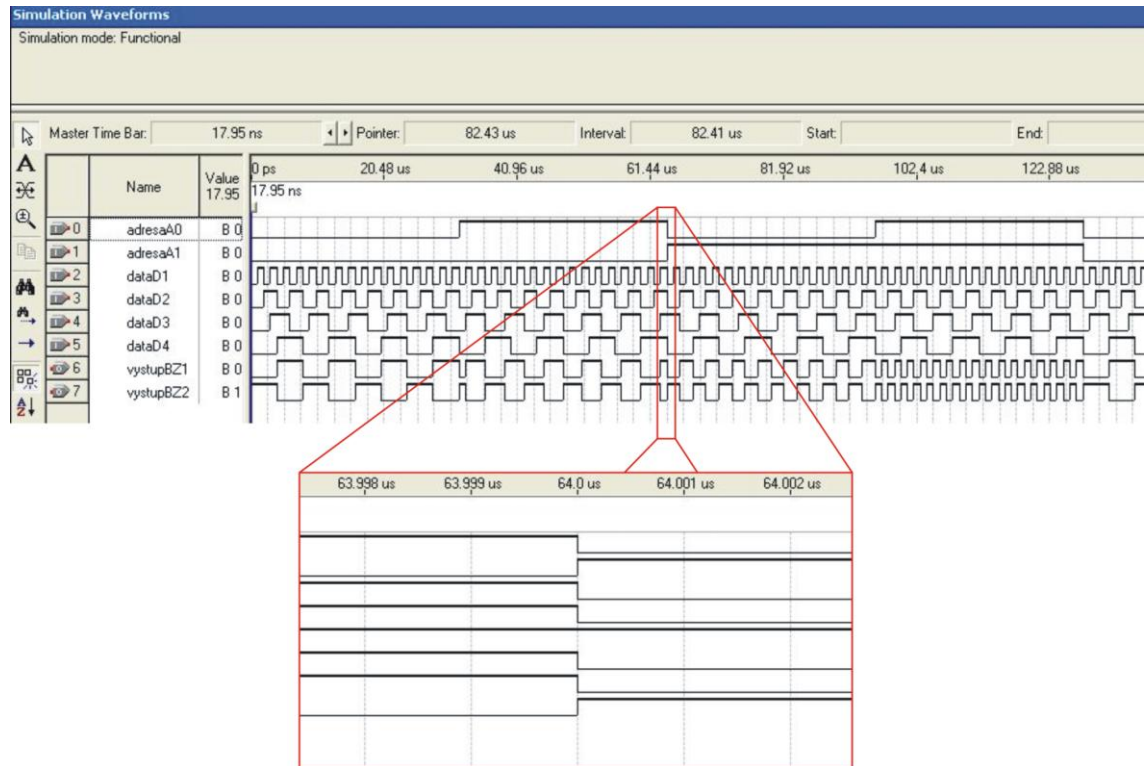


B5. Výsledok funkčnej simulácie multiplexra riešeného syntézou

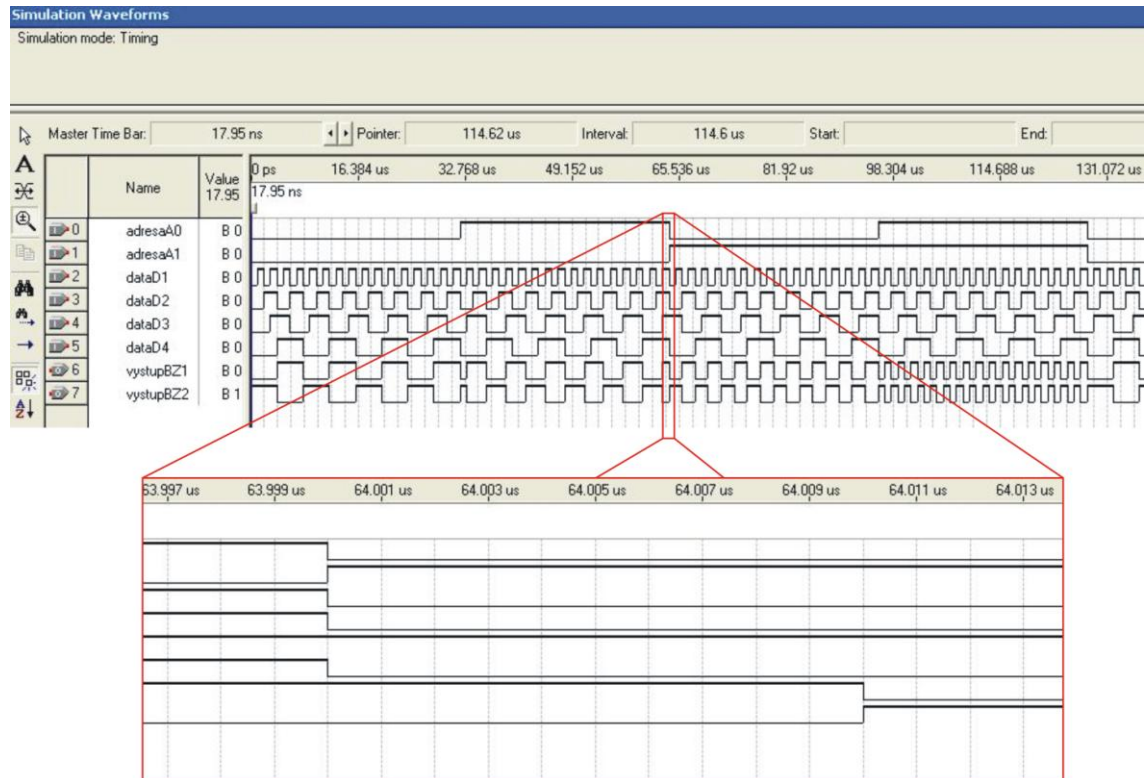


B6. Výsledok časovej simulácie multiplexra riešeného syntézou

B7. Výsledok funkčnej simulácie multiplexra riešeného pomocou opisného jazyka VHDL



B8. Výsledok časovej simulácie multiplexra riešeného pomocou opisného jazyka VHDL



PRÍLOHA C
VHDL KÓDY

C1. Kód prevodníka z BCD na sedem-segment

```

-- sedemsegmentovy display

--      +----PREVODNIK----+
--      |                    |
--      --- INPUT A  OUTPUT a ---
--      --- INPUT B  OUTPUT b ---
--      --- INPUT C  OUTPUT c ---
--      --- INPUT D  OUTPUT d ---
--      |            OUTPUT e ---
--      |            OUTPUT f ---
--      |            OUTPUT g ---
--      |                    |
--      +-----+-----+

library ieee;
use ieee.std_logic_1164.all;

entity bcd2seg is
port(
  data : in std_logic_vector (3 downto 0);
  seg : out std_logic_vector (6 downto 0)
);
end bcd2seg;

architecture a_segment of bcd2seg is
begin
  process (data) begin
    case data is
      when "0000" => seg <= "1111110";
      when "0001" => seg <= "0110000";
      when "0010" => seg <= "1101101";
      when "0011" => seg <= "1111001";
      when "0100" => seg <= "0110011";
      when "0101" => seg <= "1011011";
      when "0110" => seg <= "1011111";
      when "0111" => seg <= "1110000";
      when "1000" => seg <= "1111111";
      when "1001" => seg <= "1111011";
      when "1010" => seg <= "1110111";
      when "1011" => seg <= "0011111";
      when "1100" => seg <= "0001101";
      when "1101" => seg <= "0111101";
      when "1110" => seg <= "1001111";
      when others => seg <= "0000000";
    end case;
  end process;
end a_segment;

-- standardna kniznica organizacie IEEE
-- kniznica datovych typov

-- zedefinovanie entity
-- vektor vstupov
-- vektor vystupov

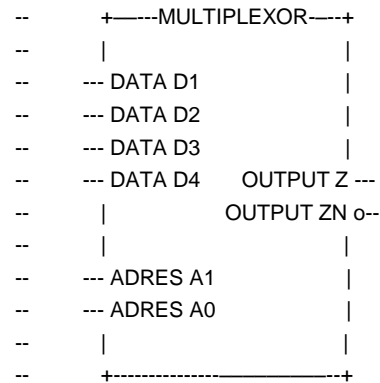
-- ukoncenie entity

-- zedefinovanie architektury entity

-- zistovanie vstupneho vektora
-- 0
-- 1
-- 2
-- 3
-- 4
-- 5
-- 6   priradenie
-- 7   vystupnych
-- 8   vektorov
-- A
-- b
-- c
-- d
-- E
--

```


C3. Kód multiplexora



```

library ieee;
use ieee.std_logic_1164.all;

entity my_mux is
  port (
    D1,D2,D3,D4 : in std_logic;           -- vstupne data
    A1, A0 : in std_logic;               -- adresne vstupy
    Z, ZN : out std_logic;               -- priamy a negovany vystup niektorej premenej
  );
end my_mux;

architecture mux_arch of my_mux is

  SIGNAL N : std_logic;
  SIGNAL V :std_logic_vector (1 DOWNT0 0); -- pomocny signal pre adresovanie
  -- vsupov

begin
  V(1) <= A1;                             --vytvorenie vektora
  V(0) <= A0;                             --vytvorenie vektora

  WITH V SELECT
    Z <= D1 WHEN "00",                    -- vyber podľa vektora V
    D2 WHEN "01",
    D3 when "10",
    D4 when "11",
    NULL WHEN OTHERS;                    -- priradenie vystupu podľa vyberu

  ZN <= not Z;                             -- vytvorenie negovaneho vystupu
end mux_arch;

```

C4. Kód Deličky

```

--      +----Delicka----+
--      |                  |
--      |          OUT f1 ---
--      |          OUT f2 ---
--      |          OUT f3 ---
--      |  IN clk  OUT f4 ---
--      |          OUT f5 ---
--      |          OUT f6 ---
--      |          OUT f7 ---
--      |                  |
--      |                  |
--      +-----+

```

```

library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_ARITH.all;
use IEEE.STD_LOGIC_UNSIGNED.all;

```

```

ENTITY delicka IS

```

```

    PORT

```

```

    (
        clock_25Mhz      : IN  STD_LOGIC;      -- zdefinovanie vstupov a vystupov
        clock_1MHz       : OUT STD_LOGIC;
        clock_100KHz     : OUT STD_LOGIC;
        clock_10KHz      : OUT STD_LOGIC;
        clock_1KHz       : OUT STD_LOGIC;
        clock_100Hz      : OUT STD_LOGIC;
        clock_10Hz       : OUT STD_LOGIC;
        clock_1Hz        : OUT STD_LOGIC);

```

```

END delicka;

```

```

ARCHITECTURE a OF delicka IS
-- vytvorenie architektury a zdefinovanie vnutornych
-- signalov

```

```

    SIGNAL count_1Mhz: STD_LOGIC_VECTOR(4 DOWNTO 0);
    SIGNAL count_100Khz, count_10Khz, count_1Khz : STD_LOGIC_VECTOR(2 DOWNTO 0);
    SIGNAL count_100hz, count_10hz, count_1hz : STD_LOGIC_VECTOR(2 DOWNTO 0);
    SIGNAL clock_1Mhz_int, clock_100Khz_int, clock_10Khz_int, clock_1Khz_int: STD_LOGIC;
    SIGNAL clock_100hz_int, clock_10Hz_int, clock_1Hz_int : STD_LOGIC;

```

```

BEGIN

```

```

    PROCESS
--prve delenie, ziskava sa frekvencia 1MHz
    BEGIN

```

```

-- Divide by 25

```

```

        WAIT UNTIL clock_25Mhz'EVENT and clock_25Mhz = '1';
        IF count_1Mhz < 24 THEN
            count_1Mhz <= count_1Mhz + 1;
        ELSE
            count_1Mhz <= "00000";
        END IF;
        IF count_1Mhz < 12 THEN
            clock_1Mhz_int <= '0';
        ELSE

```

```

        clock_1Mhz_int <= '1';
    END IF;

-- Ripple clocks are used in this code to save prescaler hardware
-- Sync all clock prescaler outputs back to master clock signal
    clock_1Mhz <= clock_1Mhz_int;
    clock_100Khz <= clock_100Khz_int;
    clock_10Khz <= clock_10Khz_int;
    clock_1Khz <= clock_1Khz_int;
    clock_100hz <= clock_100hz_int;
    clock_10hz <= clock_10hz_int;
    clock_1hz <= clock_1hz_int;
END PROCESS;

-- Divide by 10
PROCESS                                     -- druhe delenie, ziskava sa 100kHz
BEGIN
    WAIT UNTIL clock_1Mhz_int'EVENT and clock_1Mhz_int = '1';
    IF count_100Khz /= 4 THEN
        count_100Khz <= count_100Khz + 1;
    ELSE
        count_100khz <= "000";
        clock_100Khz_int <= NOT clock_100Khz_int;
    END IF;
END PROCESS;

-- Divide by 10
PROCESS                                     -- tretie delenie, ziskava sa 10kHz
BEGIN
    WAIT UNTIL clock_100Khz_int'EVENT and clock_100Khz_int = '1';
    IF count_10Khz /= 4 THEN
        count_10Khz <= count_10Khz + 1;
    ELSE
        count_10khz <= "000";
        clock_10Khz_int <= NOT clock_10Khz_int;
    END IF;
END PROCESS;

-- Divide by 10
PROCESS                                     -- stvrte delenie, ziskava sa 1kHz
BEGIN
    WAIT UNTIL clock_10Khz_int'EVENT and clock_10Khz_int = '1';
    IF count_1Khz /= 4 THEN
        count_1Khz <= count_1Khz + 1;
    ELSE
        count_1khz <= "000";
        clock_1Khz_int <= NOT clock_1Khz_int;
    END IF;
END PROCESS;

-- Divide by 10
PROCESS                                     -- piate delenie, ziskava sa 100Hz
BEGIN
    WAIT UNTIL clock_1Khz_int'EVENT and clock_1Khz_int = '1';
    IF count_100hz /= 4 THEN
        count_100hz <= count_100hz + 1;
    ELSE

```

```
        count_100hz <= "000";
        clock_100hz_int <= NOT clock_100hz_int;
    END IF;
END PROCESS;

-- Divide by 10
PROCESS                                -- sieste delenie, ziskava sa 10Hz
BEGIN
    WAIT UNTIL clock_100hz_int'EVENT and clock_100hz_int = '1';
    IF count_10hz /= 4 THEN
        count_10hz <= count_10hz + 1;
    ELSE
        count_10hz <= "000";
        clock_10hz_int <= NOT clock_10hz_int;
    END IF;
END PROCESS;

-- Divide by 10
PROCESS                                -- siedme delenie, ziskava sa 1Hz
BEGIN
    WAIT UNTIL clock_10hz_int'EVENT and clock_10hz_int = '1';
    IF count_1hz /= 4 THEN
        count_1hz <= count_1hz + 1;
    ELSE
        count_1hz <= "000";
        clock_1hz_int <= NOT clock_1hz_int;
    END IF;
END PROCESS;

END a;
```