

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra telekomunikační techniky

Využití klávesnicového vstupu PS/2 a výstupu na LCD displej přípravku Spartan3E

Simona Vránová

Školitel: Ing. Pavel Lafata, Ph.D.
Srpen 2020

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Vránová** Jméno: **Simona** Osobní číslo: **474259**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra telekomunikační techniky**
Studijní program: **Elektronika a komunikace**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Využití klávesnicového vstupu PS/2 a výstupu na LCD displej přípravku Spartan3E

Název bakalářské práce anglicky:

Using PS/2 input and LCD display of Spartan3E kit

Pokyny pro vypracování:

Seznamte se s přípravkem Xilinx Spartan3E a jeho obsluhou pomocí jazyka VHDL. K přípravku připojte pomocí rozhraní PS/2 standardní klávesnici. Navrhněte a realizujte VHDL kód pro čtení stisknuté klávesy. Doplňte využití vestavěného LCD displeje s řadičem HD44780 na přípravku tak, že na tomto znakovém displeji budou jednotlivé klávesy vypisovány. Vytvořte nezbytné VHDL kódy. Výstupem tak bude jednoduché zobrazení stisknuté klávesy na LCD displeji, eventuálně upravte dle pokynů vedoucího práce.

Seznam doporučené literatury:

[1] Manuál a dokumentace k přípravku Xilinx Spartan3E, <https://store.digilentinc.com/spartan-3e-starter-boardlimited-time/> [on-line]
[2] Popis komunikace a rozhraní PS/2, <https://www.avrfreaks.net/sites/default/files/PS2%20Keyboard.pdf> [on-line]
[3] Popis řadiče LCD HD44780, <https://vyvoj.hw.cz//teorie-a-praxe/dokumentace/inteligentni-displeje-a-jejich-pripojeni-k-pc.html>

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Pavel Lafata, Ph.D., katedra telekomunikační techniky FEL

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

Datum zadání bakalářské práce: **08.01.2020**

Termín odevzdání bakalářské práce: **14.08.2020**

Platnost zadání bakalářské práce: **30.09.2021**

Ing. Pavel Lafata, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

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

III. PŘEVZETÍ ZADÁNÍ

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

Datum převzetí zadání

Podpis studentky

Poděkování

Chtěla bych poděkovat vedoucímu práce Ing. Pavlu Lafatovi za vypůjčení přípravku a všech potřebných zařízení potřebné k realizaci, za jeho vedení a pomoc při vypracování této práce.

Prohlášení

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

V Praze, 14. srpna 2020

Abstrakt

Tato bakalářská práce se zabývá návrhem VHDL kódu pro přípravek Spartan-3E. Vstupem je připojená PS/2 klávesnice. K výstupu slouží zabudovaný LCD displej s řadičem HD44780. Vytvořený program umožňuje zobrazování stisknuté klávesy na LCD displeji.

Klíčová slova: FPGA, VHDL, PS/2 protokol, PS/2 klávesnice, Spartan-3E Starter Board, LCD displej, řadič HD44780

Školitel: Ing. Pavel Lafata, Ph.D.

Abstract

This bachelor thesis deals with the design of VHDL code for Spartan-3E. The input is a connected PS/2 keyboard. The output is a built-in LCD display with HD44780 controller. The created program allows to display the pressed key on the LCD display.

Keywords: FPGA, VHDL, PS/2 protocol, PS/2 keyboard, Spartan-3E Starter Board, LCD screen, HD44780 graphics controller

Title translation: Using PS/2 input and LCD display of Spartan3E kit

Obsah

1 Úvod	1
2 Teoretická část	3
2.1 Programovatelné logické obvody .	3
2.2 FPGA.....	3
2.2.1 Architektura FPGA	4
2.2.2 Přípravek Xilinx Spartan-3E .	4
2.2.3 Návrh a realizace FPGA	6
2.3 Jazyk VHDL	6
2.3.1 Vlastnosti a struktura kódu ..	6
2.3.2 Vývojové prostředí ISE WebPACK	7
2.4 PS/2	8
2.4.1 Fyzické připojení.....	8
2.4.2 Princip komunikace	8
2.4.3 PS/2 klávesnice	9
2.5 LCD displej s řadičem HD44780	10
2.5.1 LCD displej přípravku Spartan-3E	11
2.5.2 Paměti řadiče	11
2.5.3 Příkazy řadiče displeje	12
3 Praktická část	15
3.1 Zpracování klávesnicového vstupu	15
3.1.1 Proces čtení klávesy	15
3.1.2 Filtrace vstupu	16
3.1.3 Kontrola výstupu pomocí LED	16
3.2 Zpracování výstupu na LCD ...	17
3.2.1 Ovládání displeje	18
3.2.2 Hlavní stavový automat.....	18
3.2.3 Inicializace displeje	18
3.2.4 Zápis na displej	20
3.2.5 Kontrola funkčnosti LCD displeje	21
3.3 Konečný výstup	21
3.3.1 Převod scan kódu na ASCII kód	22
3.3.2 Zobrazení na displeji	23
3.3.3 Vypisování znaku na displej .	23
3.3.4 Využití LED výstupu	24
3.3.5 Mapování portů	24
3.3.6 Výsledný projekt	24
3.3.7 Možné rozšíření	25
4 Závěr	27
Literatura	29
A Obsah příloženého CD	31

Obrázky

2.1 Základní architektura FPGA obvodu [7]	4
2.2 Vývojová deska Xilinx Spartan-3E [1]	5
2.3 Uživatelské prostředí ISE Webpack [vlastní zdroj]	7
2.4 Konektor PS/2 [1]	8
2.5 Časový průběh komunikace PS/2 klávesnice s přípravkem[2]	9
2.6 Scan kód PS/2 klávesnice [1]	9
2.7 Osmibitový interface řadiče [3] .	11
2.8 Připojení řadiče displeje k přípravku Spartan-3E [1]	11
2.9 Znakový generátor [3]	12
2.10 DD RAM adresy [1].....	13
3.1 Scan kód pro klávesu Q posílán klávesnicí [2]	16
3.2 Ukázka VHDL kódu pro vstupní filtr [zdroj vlastní]	17
3.3 Výstup LED při stisku klávesy A [zdroj vlastní]	17
3.4 Kód pro nastavení signálu LCD_RS [zdroj vlastní]	18
3.5 Stavový automat pro ovládání displeje [zdroj vlastní]	19
3.6 Zápis na displej přes 4bitové rozhraní [1]	20
3.7 Změna výstupu na displeji po inicializaci [zdroj vlastní]	21
3.8 Entita výsledného programu [zdroj vlastní]	22
3.9 Ukázka kódu pro převod scan kód na ASCII kód [zdroj vlastní]	22
3.10 Výsledný projekt [zdroj vlastní]	23
3.11 Výsledný projekt [zdroj vlastní]	25

Tabulky

2.1 Konektor PS/2 zapojení pinů....	8
2.2 Vývody displeje	10
3.1 Mapování portů	24

Kapitola 1

Úvod

Programovatelná hradlová pole FPGA jsou v současné době hojně využívány pro mnohé aplikace, lze vytvořit v podstatě libovolná číslicová zařízení. Jejich hlavní výhodou je programovatelnost až u zákazníka a jejich následná reprogramovatelnost. Přípravky obsahující FPGA obvody obsahují i další periferie.

PS/2 klávesnice jsou v dnešní době sice nahrazovány klávesnicemi se sběrnici USB, ale v některých aplikacích mohou být stále preferovány. Především díky jejich jednoduššímu ovládání.

LCD displeje s řadičem HD44780 se mezi návrháři staly standardem. Jsou variabilní v mnoha směrech. Podporují 4bitovou i 8bitovou komunikaci, lze nastavit velikost LCD displeje a další parametry.

Cílem této práce je seznámit se s přípravkem Spartan-3E, jeho obsluhou pomocí jazyka VHDL a obeznámit se s jeho periferiemi. Vytvořit VHDL kód, aby přípravek detekoval stisk klávesy ze standardní PS/2 klávesnice. Výstupem je zobrazení stisknuté klávesy na LCD displeji, který je součástí přípravku.

Tato práce je primárně rozdělena do dvou částí - teoretické a praktické. V teoretické části je představen přípravek Spartan-3E. Dále jsou popsány připojené periferie - rozhraní PS/2 port a LCD displej s řadičem HD44780. Praktická část se soustředí na realizaci programu.

Kapitola 2

Teoretická část

V této části jsou stručně popsány programovatelné logické obvody a jejich návrh a realizace pomocí jazyka VHDL. Dále je zde popsán vývojový kit Spartan-3E, který je využit pro praktickou realizaci. Také je zde popsáno rozhraní PS/2 a LCD displej s řadičem HD44780.

2.1 Programovatelné logické obvody

Číslicové programovatelné obvody se souhrnně označují PLD (Programmable Logic Device). PLD umožňují návrh a realizaci logických funkcí pomocí programovatelných logických obvodů a hradel. Lze vytvářet specifické integrované obvody, speciální obvody pro konkrétní aplikace a běžné logické funkce a obvody s minimálními náklady. [4, strana 93]

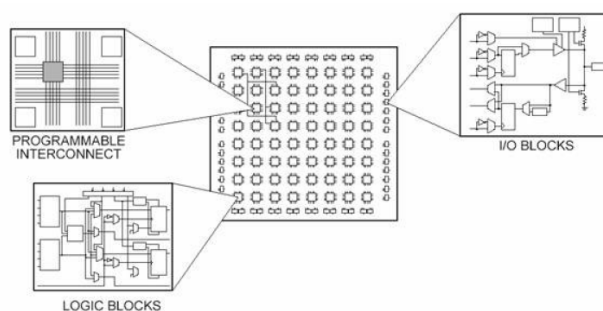
Podle vnitřní struktury se dělí na 3 skupiny. Klasické PLD, také označované jako Simple PLD, které dokáží realizovat jen jednodušší funkce a jsou zároveň historicky prvními elektronicky reprogramovatelnými obvody. Další skupinou jsou CPLD, komplexní PLD, u kterých se oproti SPLD objevuje globální propojovací matice. Poslední skupinu tvoří obvody FPGA (Field Programmable Gate Array). [9] [4, strana 94]

2.2 FPGA

Jak již bylo zmíněno v předchozí části, FPGA spadá do skupiny programovatelných logických obvodů. Jedná se o speciální integrovaný obvod s velmi velkou hustotou integrace. Je založen na myšlence využití paměti a paměťových obvodů pro definování logických funkcí.[4, strana 95-96] [5] [6, strana 35]

Využití FPGA je výhodné zvláště pro aplikace, kde jsou zpracovány velké datové toky a pro výpočetně náročné algoritmy. Také jsou využívány během návrhu zákaznického integrovaného obvodu. Dalším využitím je situace, kde je potřeba speciální hardware, ale série je moc mála, takže se nevyplatí návrh zákaznického integrovaného obvodu.[6, strana 38-40]

Hlavními konkurenty FPGA jsou mikroprocesory, CPLD a ASIC (Application Specific Integrated Circuit). Mikroprocesory jsou flexibilnější, ale



Obrázek 2.1: Základní architektura FPGA obvodu [7]

nenabízí paralelní řešení problémů. CPLD lze použít jen na jednodušší funkce, jsou však rychlejší a levnější. ASIC jsou dražší a nejsou reprogramovatelné. Ve výsledku jsou rychlejší a mají menší spotřebu oproti FPGA. [8] [5]

Hlavními výrobci obvodů FPGA jsou firmy Xilinx, Altera, Atmel, Actel a Lattice. [9] [5] Tato práce se zaměřuje na produkt firmy Xilinx - Spartan-3E.

2.2.1 Architektura FPGA

Všechna FPGA se skládá z několika základních prvků:

- konfigurovatelné logické bloky (CLB) - základní část FPGA provádějící logické operace
- vstupně/výstupní (I/O) bloky - zajišťují spojení vnitřku FPGA a okolním světem
- propojovací matice spojů - soubor všech signálových cest a propojů

Tato základní architektura je na obrázku 2.1.

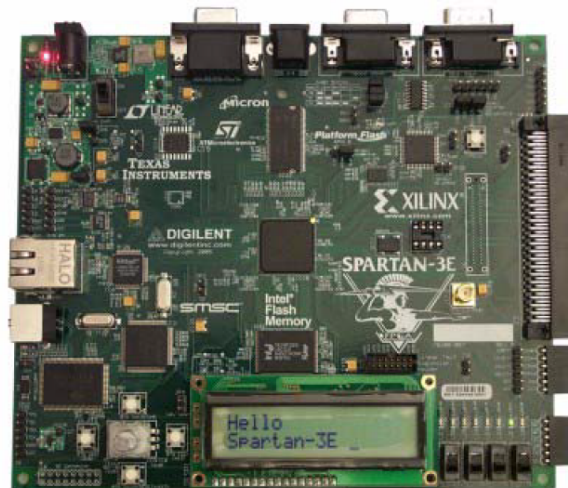
Logické bloky jsou složeny z logických buněk, označovaných LC (Logic Cells). Například u přípravku Spartan-3E jsou součástí logických buněk klopné obvody typu D, multiplexory a LUT (Look-Up Table) - 16bitové paměti schopné realizovat všechny logické funkce čtyř proměnných. [6, strana 41] [5] [7]

Dále mohou obsahovat další specializované bloky jako jsou například generátory hodin, paměti RAM, celé dedikované bloky pro aritmetické operace, dodatečné paměti flash atd. [6, strana 41]

2.2.2 Přípravek Xilinx Spartan-3E

Přípravek Spartan-3E (obrázek 2.2) je výrobkem firmy Xilinx. Jeho hlavní částí je FPGA obvod s označením XC3S500E. Základní architektura tohoto FPGA se skládá z šesti základních funkčních bloků:

- vstupně/výstupní bloky
- konfigurovatelné logické bloky



Obrázek 2.2: Vývojová deska Xilinx Spartan-3E [1]

- programovatelná propojovací matice
- blok paměti RAM
- blok násobičky
- taktovací obvod DCM (Digital Clock Manager)

Kromě FPGA obvodu obsahuje přípravek ještě paměť a taktovací obvody. Do této kategorie patří 50 MHz vnitřní oscilátor. Dále obsahuje velké množství periférií, mezi které patří například:

- LCD displej, 16 znaků, 2 řádky
- PS/2 port pro myš a klávesnici
- 8 indikačních LED diod
- VGA displej port
- 4 přepínače - posuvné
- 4 spínače - tlačítka
- vestavěné rozhraní pro naprogramování, ladění a debugging pomocí USB

Přípravek Spartan-3E na rozdíl od typického FPGA obsahuje tři různé konfigurační paměti, které musí fungovat pospolu. [1] [13] [4, strana 110-113]

■ 2.2.3 Návrh a realizace FPGA

Celý proces návrhu a realizace FPGA se skládá z několika kroků. Pro popis návrhu logických obvodů FPGA se používají HDL jazyky, jako jsou VHDL a Verilog. [7] Tento textový popis se syntetizuje a výsledkem je netlist obsahující obecné logické prvky. Následně se musí tento netlist optimalizovat pro konkrétní FPGA zařízení. Během návrhu se využívá simulátor pro ověření funkčnosti. Posledním krokem je nahrání programu do konkrétního koncového zařízení. [9] [13, strana 15-17]

■ 2.3 Jazyk VHDL

Zkratka VHDL znamená Very-High-Speed Integrated Circuits Hardware Description Language. To je v překladu jazyk pro popis velmi rychlých integrovaných obvodů. Účelem je popis a simulace rozsáhlých návrhů číselných obvodů. Dále také psaní testovacích programů. Jazyk VHDL se stal standardem pod označením IEEE.

Jazyk VHDL je úzce spjat s hardwarem. Při programování se musí brát v úvahu hardwarová omezení daného programovatelného pole. S tím také přichází podmínka syntetizovatelnosti. To znamená, že překladač musí být schopen vytvořit zapojení jednotlivých hradel a logických členů. [4, strana 96-109] [11, strana 1-9] [10, strana 1-10]

■ 2.3.1 Vlastnosti a struktura kódu

V jazyce VHDL rozlišujeme 3 způsoby modelování logických obvodů:

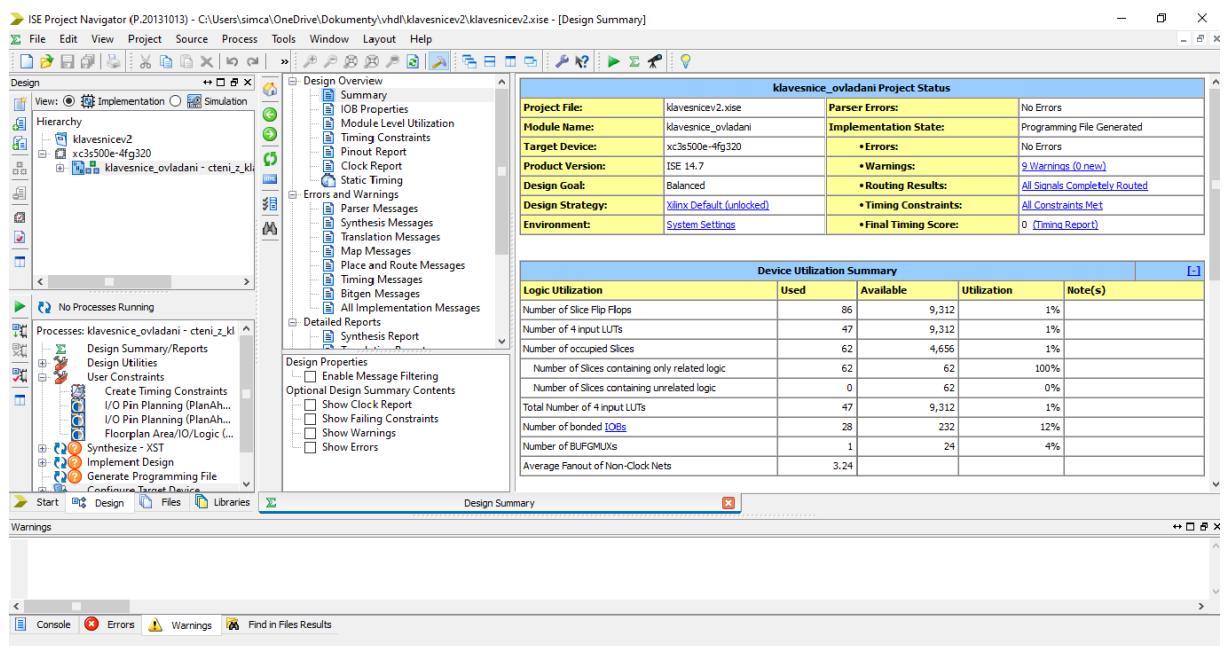
- behaviorální - model chování obvodu, nejvyšší úroveň abstrakce
- Dataflow popis (RTL) - modelování činnosti obvodu pomocí Booleových rovnic
- strukturální - modelování zapojení obvodu (nakreslení schématu zapojení)

Způsoby modelování se mohou v jazyce VHDL kombinovat. Behaviorální popis je ideální pro tvoření složitějších struktur.

Prostředí jazyka VHDL je defaultně paralelní, stejně jako FPGA obvody. To znamená, že jsou všechny příkazy prováděny najednou. Sekvenční prostředí se musí specifikovat pomocí příkazu *process*, poté jsou příkazy zpracovány jeden po druhém. [4]

VHDL modul se skládá z následujících položek:

- hlavička
- deklarace knihoven
- deklarace entity
- deklarace architektury



Obrázek 2.3: Uživatelské prostředí ISE Webpack [vlastní zdroj]

Hlavička obsahuje libovolný text a komentáře. Knihovny obsahují veškeré funkce, příkazy a operace v jazyce VHDL. Pro jejich použití se musí deklarovat daná knihovna a její balíčky. Například knihovna `std_logic_1164` zpřístupňuje datové typy, konverzní funkce a operace pro vícehodnotovou logiku.

V rámci entity se definuje její název a porty (vstupy a výstupy). Samotná entita představuje výsledný obvod. Každá entita má architekturu, která popisuje její chování, činnost nebo zapojení. Jedna entita může mít více architektur. [10] [11]

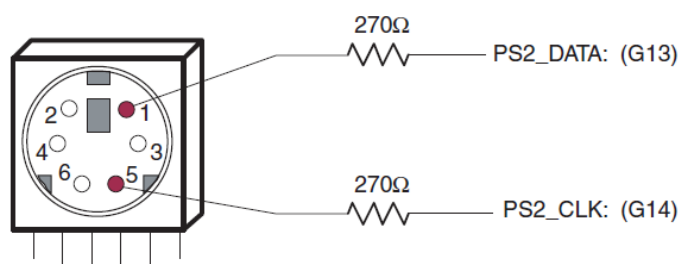
Veškeré dostupné struktury jazyka VHDL lze dohledat v literatuře [12].

2.3.2 Vývojové prostředí ISE WebPACK

Pro vývoj FPGA aplikací je potřeba systému, který obsahuje dva nástroje. Nástroj pro syntézu HDL kódu na obecný netlist a druhý nástroj, který obecný netlist převede na netlist pro specifické FPGA, aby bylo dosaženo optimálního rozložení a propojení jednotlivých logických obvodů. Vývojové prostředí ISE WebPACK obsahuje oba tyto nástroje. Je to návrhové prostředí od firmy Xilinx, které je dostupné zadarmo ke stažení.

Prostředí ISE (Integrated Software Environment) obsahuje vlastní návrhový systém, nástroj pro simulace nebo konfigurátor rozložení jednotlivých logických bloků a jejich propojování na vstupně/výstupní periferie. [13, strana 17-18] [4, strana 113-114] [9]

Realizace návrhu probíhá v několika krocích. Nejprve se navrhne program v editoru schémat, nebo VHDL jazyce. Poté se definují a přiřadí vstupní a výstupní porty a signály programu k odpovídající části přípravku, pomocí nástroje ISE PlanAhead. Následuje syntéza pomocí nástroje XST (Xilinx



Obrázek 2.4: Konektor PS/2 [1]

PS/2 DIN Pin	Signál
1	DATA
2	Rezervován
3	Uzemnění
4	Napájení (+5V)
5	Hodinový signál PS/2
6	Rezervován

Tabulka 2.1: Konektor PS/2 zapojení pinů

Synthesis Technology). Během celého procesu lze kontrolovat zapojení například pomocí simulací. Posledním krokem je nahrání programu do přípravku většinou pomocí rozhraní JTAG v programu iMPACT. [13, strana 15-18] [4, strana 113-114]

2.4 PS/2

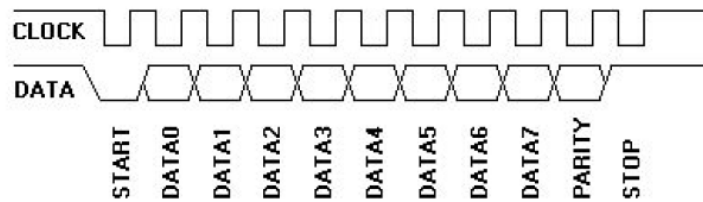
Konektor PS/2 je standardní 6-pin mini-DIN konektor, kterým se připojují k počítači zařízení jako klávesnice nebo myš. Zkratka PS/2 pochází ze slovního spojení Personal System/2. V dnešní době je konektor vytlačován USB. [2]

2.4.1 Fyzické připojení

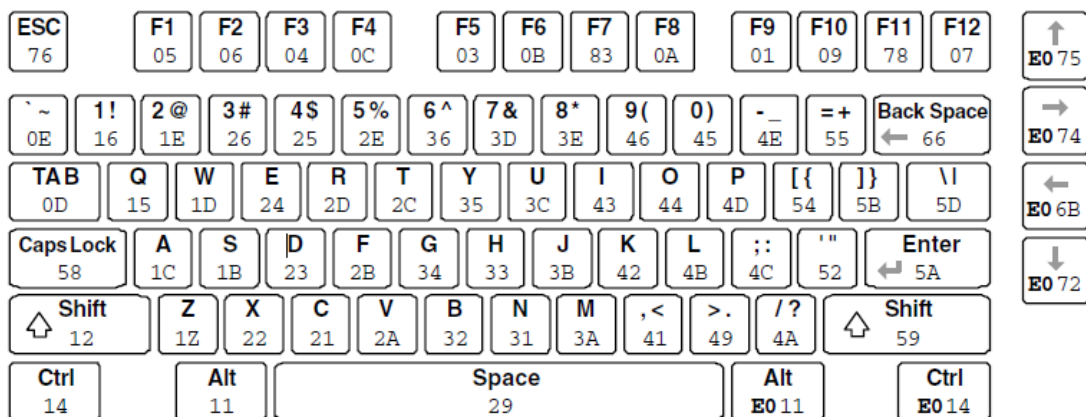
PS/2 port má dva typy konektorů: 5-pin DIN a 6-pin mini-DIN. Oba typy jsou podobné, hlavní rozdíl je v rozložení pinů. Přípravek Spartan-3E má typ 6-pin mini-DIN. Klávesnice s tímto konektorem jsou označovány jako PS/2 klávesnice. Rozložení pinů je zobrazeno na obrázku 2.4 a jejich význam vypsán v tabulce 2.1. PS/2 port je napájen 5V. [2] [1, strana 61-62, 65]

2.4.2 Princip komunikace

Komunikace mezi klávesnicí a přípravkem je obousměrná. Klávesnice generuje hodinový signál. V komunikačním protokolu se rozlišují tři stavy. Klávesnice



Obrázek 2.5: Časový průběh komunikace PS/2 klávesnice s přípravkem[2]



UG230_c8_03_021806

Obrázek 2.6: Scan kód PS/2 klávesnice [1]

může odesílat data pouze ve stavu *idle*, kdy oba signály hodiny i dat jsou na úrovni logické '1'. Data posílaná klávesnicí jsou čtena na sestupné hraně hodinového signálu. [2] Komunikace klávesnice s přípravkem probíhá v 11-bitových rámcích. Tyto rámce se skládají z následujících bitů:

- 1 start bit - vždy '0'
- 8 data bitů - nejméně významný bit (LSB) první
- 1 parity bit - detekce chyby
- 1 stop bit - vždy '1'

Sekvence bitů při komunikaci klávesnice s přípravkem je vidět na obrázku 2.5. [2]

2.4.3 PS/2 klávesnice

Klasické PS/2 klávesnice mají 101, nebo 104 kláves. Klávesnice je tedy velká matice kláves, které jsou monitorovány procesorem na klávesnici. Tento procesor kontroluje stisk/puštění kláves a posílá příslušná data přípravku.

Číslo vývodu	Označení	Význam
1	V_{ss}	GND
2	V_{cc}	napájení (+5V)
3	V_{ee}	nastavení kontrastu
4	RS	instrukce/data
5	R/W	zápis/čtení
6	E	aktivace displeje
7	DB0	data, bit 0 (nejnižší)
8	DB1	data, bit 1
9	DB2	data, bit 2
10	DB3	data, bit 3
11	DB4	data, bit 4
12	DB5	data, bit 5
13	DB6	data, bit 6
14	DB7	data, bit 7 (nejvyšší)

Tabulka 2.2: Vývody displeje

Přípravek obsahuje "keyboard controller", který dekóduje přijatá data. Celá komunikace probíhá na IBM protokolu. [2]

Informace, které se posílají, se nazývají scan kód (obrázek 2.6). Rozlišujeme dva typy scan kódů: make kód a break kód. Make kód je vyslán při stisku a držení klávesy. Break kód je poslán při puštění klávesy. Každá klávesa má unikátní set make kódu a break kódu. Důležité je, že neexistuje žádná přímá spojitost mezi make kódem a ACII kódem. Make kód reprezentuje klávesu nikoliv písmeno napsané na ni. [2]

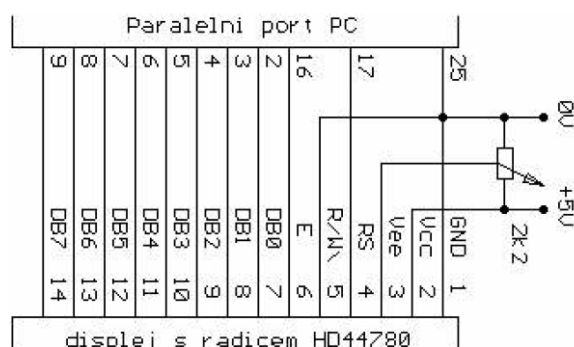
Většina kláves má make kód 1 byte dlouhý. Některé speciální klávesy jsou dlouhé 2 byty, například klávesa pravé šipky, kde make kód je "E0, 74". Break kód je pak pro většinu kláves 2 byty dlouhý, pro speciální klávesy pak 3 byty. [2]

To znamená, že při stisku klávesy "A" je vyslán make kód "1C", po jejím puštění je vyslán break kód "F0, 1C". Break kód přímo vyplývá z make kódu. Pro klávesu pravé šipky by pak byl break kód "E0, F0, 74". [2]

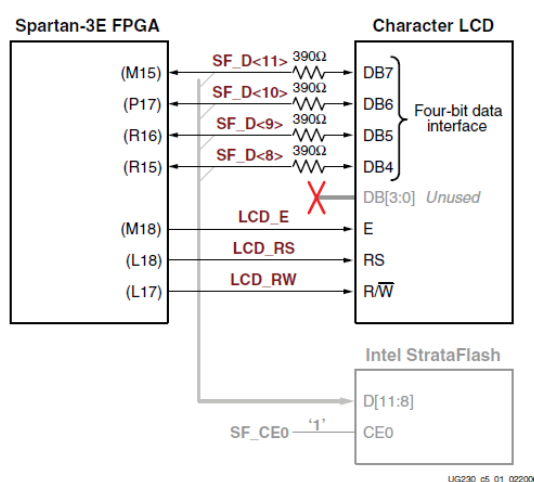
2.5 LCD displej s řadičem HD44780

LCD displeje se vyskytují ve velké řadě aplikací. Velkou výhodou displejů s řadičem HD44780 je jejich rozšířenost, snadné ovládání, možnosti definice až osmi znaků, což umožňuje nastavit češtinu nebo různé efekty, různorodost ve výběru velikosti. Jejich specifikace se zavádí jako průmyslový standard. [2]

Řadič obsahuje 14 vývodů, význam jednotlivých vývodů je vypsán v tabulce



Obrázek 2.7: Osmibitový interface řadiče [3]



Obrázek 2.8: Připojení řadiče displeje k přípravku Spartan-3E [1]

2.2. Komunikační protokoly probíhají po osmibitové nebo čtyřbitové sběrnici. Osmibitové připojení k PC je zobrazeno na obrázku 2.7. [3]

2.5.1 LCD displej přípravku Spartan-3E

LCD displej na přípravku Spartan-3E obsahuje 16 znaků na řádek a dva řádky. Tento displej je založen na řadiči HD44780 od společnosti Hitachi. Samotné připojení řadiče k přípravku je vidět obrázku 2.8. Jak je vidět, displej je připojen čtyřmi datovými vodiči a dalšími 3 řídicími signály. [1, strana 41-42]

2.5.2 Paměti řadiče

Řadič obsahuje tři vnitřní paměti, každá má svůj účel. K těmto pamětím lze přistupovat pouze po inicializaci displeje.

DD RAM ukládá kód znaku, který má být zobrazen na displeji. Na obrázku 2.10 jsou defaultní adresy 32 lokací znaků na displeji.

	0	0	0	0	0	0	1	1	1	1	1	1
	0	0	0	1	1	1	1	0	0	1	1	1
	0	1	1	0	0	1	1	1	1	0	0	1
	0	0	1	0	1	0	1	0	1	0	1	0
xxxx0000		0	@	P	`	P	-	9	3	α	ρ	
xxxx0001		!	1	A	Q	a	q	。	ア	チ	4	ä
xxxx0010		"	2	B	R	b	r	「	イ	ツ	×	β
xxxx0011		#	3	C	S	c	s	」	ウ	フ	ε	ω
xxxx0100		\$	4	D	T	d	t	、	エ	ト	μ	Ω
xxxx0101		%	5	E	U	e	u	・	オ	ナ	1	ü
xxxx0110		&	6	F	V	f	v	ヲ	カ	ニ	ヨ	Σ
xxxx0111		'	7	G	W	g	w	フ	キ	ヲ	ラ	π
xxxx1000		(8	H	X	h	x	イ	ク	ネ	リ	⊗
xxxx1001)	9	I	Y	i	y	ウ	ケ	ル	リ	4
xxxx1010		*	:	J	Z	j	z	エ	コ	ハ	レ	i
xxxx1011		+	;	K	[k	[オ	サ	ヒ	ロ	×
xxxx1100		,	<	L	¥	l	l	ヲ	シ	フ	ワ	φ
xxxx1101		-	=	M]	m]	ユ	ズ	ハ	ン	÷
xxxx1110		.	>	N	^	n	^	ヨ	セ	ホ	ッ	ñ
xxxx1111		/	?	O	_	o	_	ッ	ツ	マ	°	ö

Obrázek 2.9: Znakový generátor [3]

CG ROM (Character Generator ROM) je paměť obsahující předdefinované znaky displeje - znakový generátor (obrázek 2.9).

CG RAM (Character Generator RAM) přináší místo pro vytvoření osmi vlastních znaků.

[1, strana 43-46]

■ 2.5.3 Příkazy řadiče displeje

Displej se ovládá pomocí souboru příkazů, které jsou vypsány v [3] a v manuálu k přípravku Spartan-3E [1, strana 46-50]. Jedná se o tyto příkazy:

- smazání displeje
- nastavení kurzoru na začátek
- nastavení vstupního režimu

1	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10
2	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

Obrázek 2.10: DD RAM adresy [1]

- zapnutí/vypnutí displeje, kurzoru a jeho blikání
- nastavení pohybu kurzoru/displeje
- nastavení interface
- nastavení pozice v CG RAM
- nastavení pozice v DD RAM
- zapíše do DD RAM nebo CG RAM
- čte z DD RAM nebo CG RAM

Kapitola 3

Praktická část

Následující část popisuje postup realizace praktického řešení zadání. Tedy vytvoření VHDL kódu pro čtení stisknuté klávesy a její zobrazení na LCD displeji, následné nahrání syntetizovaného kódu do přípravku Spartan-3E a kontrola výstupu.

Samotné řešení se dá rozdělit do více kroků. V prvním bylo potřeba vytvořit kód pro čtení stisku klávesy. Druhým krokem byla realizace zobrazování libovolných znaků na displeji. Posledním úkolem bylo propojení kódů pro vstup a výstup. Takto je dělena i praktická část, tedy do podkapitol:

- Zpracování klávesnicového vstupu
- Zpracování výstupu na LCD
- Konečný výstup

V jednotlivých podkapitolách bude vysvětlena hlavní myšlenka postupu, případně poukázáno na řešení různých problémů, které nastali při tvorbě. Dále je poukázáno na možné rozšíření řešení.

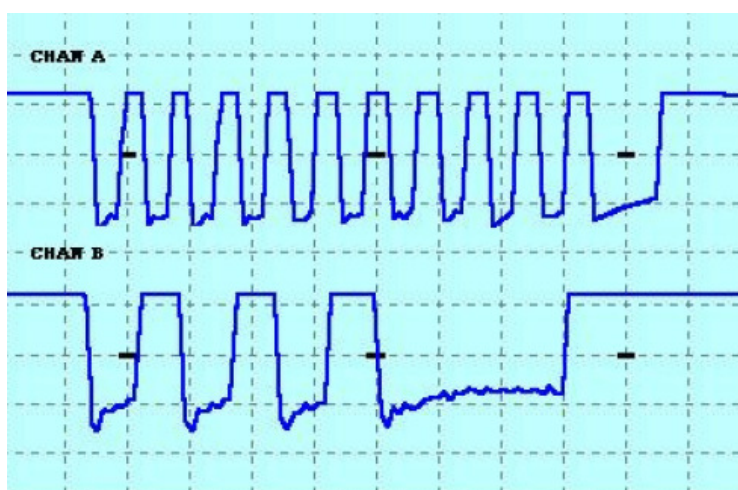
3.1 Zpracování klávesnicového vstupu

Pro vstup je využitý port PS/2, který přípravek Spartan-3E obsahuje. Klávesnice je k přípravku připojena. Vstupem je časový a datový signál, které jsou oba generovány klávesnicí. Výstupem je scan kód, který je v dalších částech upraven pro následné použití na výstup displeje.

3.1.1 Proces čtení klávesy

Princip komunikace klávesnice s přípravkem byl popsán v teoretické části (2.4.2). Posílání dat je řízeno hodinovým signálem generovaným samotnou klávesnicí. Komunikace probíhá vždy po 11 bitech. Data jsou platná na sestupná hraně hodinového signálu.

Načítání stisknuté klávesnice probíhá v rámci jednoho procesu. Výstupem tohoto procesu je scan kód stisknuté klávesy a signál označující konec vysílání. Scan kód je pak dále využitý na výstupu.



Obrázek 3.1: Scan kód pro klávesu Q posílán klávesnicí [2]

V rámci procesu jsou postupně na sestupné hraně hodinového signálu načítáno 11 bitů. Druhý načítaný bit je prvním bitem scan kódu. Tento bit je uložen do vektoru na pozici nula, bity jsou posílány LSB první. Postupně se takto načte všech osm bitů do vektoru scan kódu. Následují ještě dva bity - parity a stop. Při načtení stop bitu se nastaví signál, označující ukončení načítání klávesy, na logickou 1. Na začátku procesu je tento signál opět nastaven na hodnotu logické 0.

3.1.2 Filtrace vstupu

Na obrázku 3.1 je vidět časový průběh datového a hodinového signálu posílaným klávesnicí. Kanál A obsahuje hodinový signál, kanál B datový signál. Jak je vidět vstup může být poměrně rušený. Toto rušení může způsobit špatné zachycení hrany hodinového signálu nebo špatné detekování datových bitů.

Řešením tohoto problému je implementace filtru na vstupu. K tomuto slouží jednoduchý 8bitový posuvný registr. Do registru načítáme aktuální hodnotu signálu vždy na vzestupné hraně hodinového signálu, který je generovaný 50 MHz hodinového generátoru přípravku Spartan-3E. K překlopení z logické 1 na logickou 0 dochází až když má registr hodnotu $x"00"$. Opačně pokud má registr hodnotu $x"FF"$. Ukázka realizace ve VHDL je na obrázku 3.2.

3.1.3 Kontrola výstupu pomocí LED

Výstupem klávesnicového vstupu je scan kód. Pro kontrolu správnosti scan kódu jsou využity LED výstupy přípravku Spartan-3E. Scan kód je 8bitový vektor a přípravek obsahuje 8 LED. Načítaný scan kód je zobrazen binárně na LED a poté je porovnán s očekávanými hodnotami pro scan kód. Například při stisku klávesy A, se posílá scan kód "00011100". Na přípravku se rozsvítí LED, jak je vidět na obrázku 3.3.

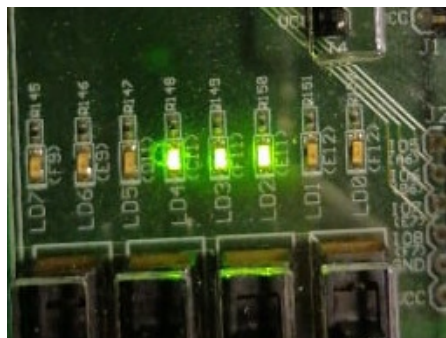
Díky této kontrole byl nalezen problém s načítáním bitů do vektoru pro


```

filtrace : process(clk)
begin
  if(clk'event and clk='1') then
    ps2c_filter(7) <= ps2_clk;
    ps2c_filter(6 downto 0) <= ps2c_filter(7 downto 1);
    ps2d_filter(7) <= ps2_data;
    ps2d_filter(6 downto 0) <= ps2d_filter(7 downto 1);
    if(ps2c_filter=x"FF") then
      ps2cf <= '1';
    elsif(ps2c_filter=x"00") then
      ps2cf <= '0';
    end if;
    if(ps2d_filter=x"FF") then
      ps2df <= '1';
    elsif(ps2d_filter=x"00") then
      ps2df <= '0';
    end if;
  end if;
end process filtrace;

```

Obrázek 3.2: Ukázka VHDL kódu pro vstupní filtr [zdroj vlastní]



Obrázek 3.3: Výstup LED při stisku klávesy A [zdroj vlastní]

scan kód. Bity ve vektoru scan kódu byly načítány vždy s posunem o jednu pozici. Tento problém byl vyřešen přidáním druhé podmínky na začátku načítání čtení klávesy. Touto podmínkou je detekce start bitu. To znamená přidání porovnání prvního s hodnotou očekávanou pro start bit, logická 0.

3.2 Zpracování výstupu na LCD

Přípravek Spartan-3E obsahuje zabudovaný LCD displej s řadičem DH44780. Tento displej je použit jako výstup k projektu. V následujících kapitolách je popsáno základní nastavení a ovládání displeje.

```
with cur_stav select
  LCD_RS <= '0' when function_set | entry_mode | displej_on | clear | nastav_adr,
  '1' when others;
```

Obrázek 3.4: Kód pro nastavení signálu LCD_RS [zdroj vlastní]

3.2.1 Ovládání displeje

Displej ovládáme pomocí tří řídicích signálů a čtyř datových. Dále k jeho ovládání je potřeba hodinový signál a signál SF_CE0. Celkem je tedy pro ovládání displeje potřeba 9 signálů.

Důležité je vědět, co jaký signál reprezentuje. Například signál SF_CE0 vypíná intel StrataFlash a má konstantní hodnotu logické 1. Datové signály jsou sdíleny právě se zmiňovanou StrataFlash, proto musí být vypnuta. Dále řídicí signál LCD_RW určuje zda displej čte, nebo píše. Jelikož je po celou dobu potřeba pouze psaní, nastaví se hodnota na logickou nulu. Další signály se mění podle stavu, ve kterém se displej nachází. Například signál LCD_RS je ve stavu logická 1, když se posílají data. Při posílání instrukcí je signál roven logické 0. Kód pro toto přiřazení je vidět na obrázku 3.4.

Displej je ovládán pomocí tří stavových automatů popsaných níže v odstavcích 3.2.2, 3.2.3 a 3.2.4.

3.2.2 Hlavní stavový automat

Displej se vždy nachází v jednom z následujících 9 stavů: *idle*, *function_set*, *entry_mode*, *displej_on*, *clear*, *pauza*, *nastav_adr*, *znak_přijat* a *hotovo*. Přecházení mezi jednotlivými stavy je znázorněno na obrázku 3.5.

Podle toho v jakém stavu se displej nachází, se mění hodnoty signálu LCD_RS. Pokud se předává instrukce má hodnotu logická 0, při přenosu dat má hodnotu logická 1.

Datové signály pak přenáší instrukce/data ve všech stavech kromě *init*, *pauza*, *hotovo*. Samotné posílání je pak řešeno samostatným stavovým automatem, který popsán v odstavci 3.2.4.

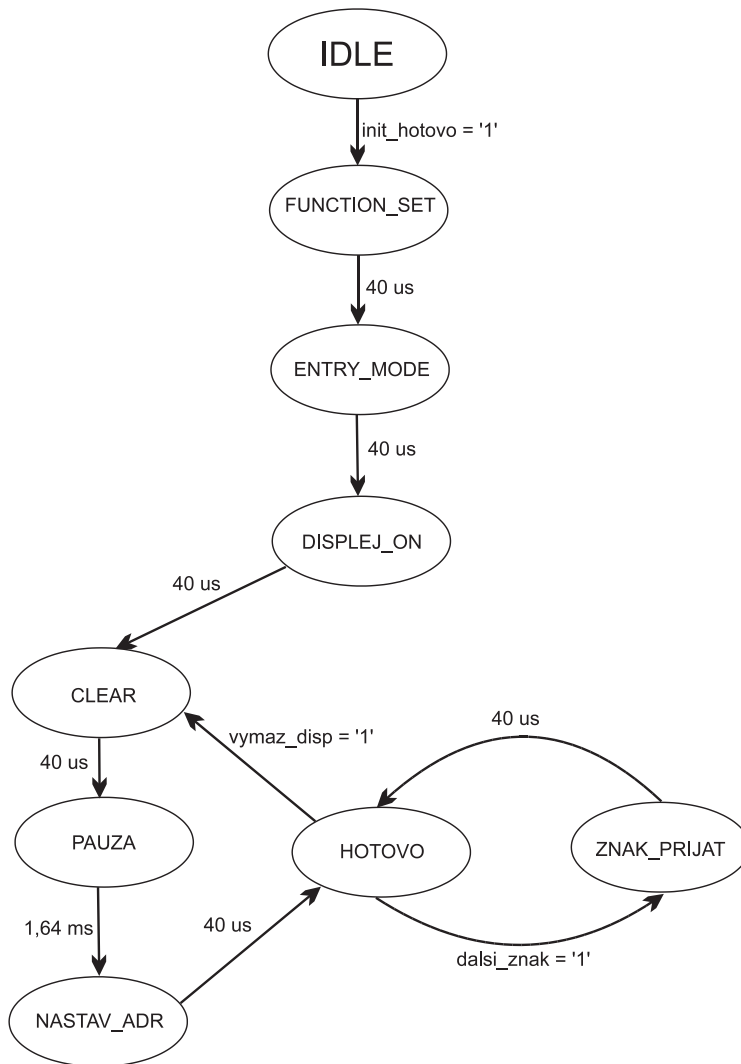
Unikátním stavem je stav *init*, který je také řešen samostatným stavovým automatem a blíže popsán v následující části 3.2.3.

Stavy *function_set*, *entry_mode*, *displej_on* nastaví základní parametry displeje jako počet řádku, font písma, posun pozice v DD RAM a rozsvícení displeje. Po tomto prvotním nastavení je potřeba displej vymazat. Doba na provedení vymazání displeje je 1,42 ms. Následuje nastavení adresy pro zápis.

Po předchozích krocích je displej připraven k zobrazování. Když od vstupu dostane pokyn k zápisu, přepne se do stavu *znak_přijat* a vypíše znak uložený v příslušném signálu.

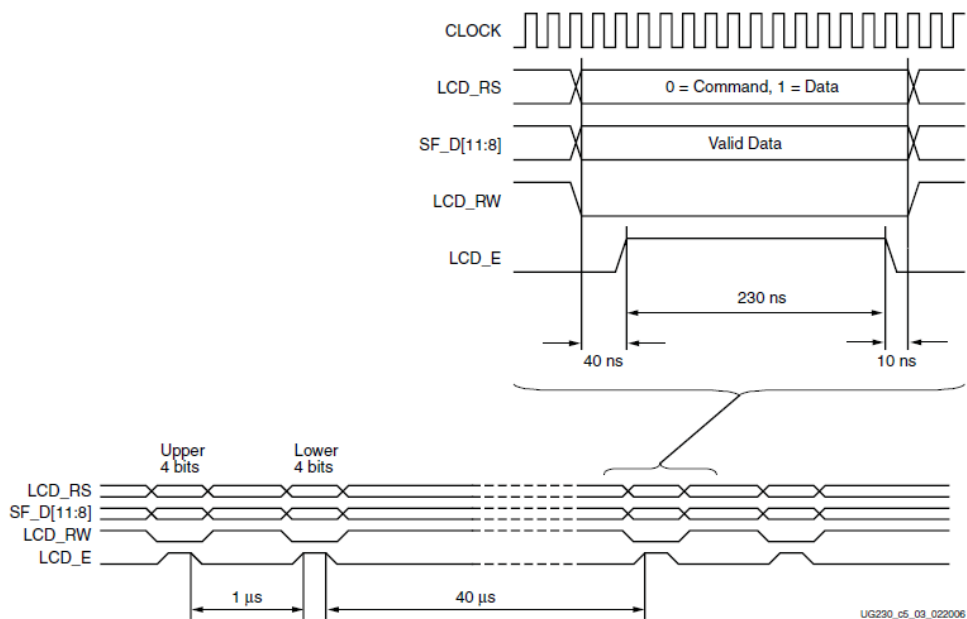
3.2.3 Inicializace displeje

Po zapnutí displeje je nutné nastavit 4bitové rozhraní. To se nastavuje pomocí sekvence příkazů a časových intervalů. Tato sekvence se skládá z devíti kroků, konkrétně:



Obrázek 3.5: Stavový automat pro ovládání displeje [zdroj vlastní]

1. časový interval 15 ms (tomu odpovídá 750 000 hodinových cyklů)
2. nastavení LCD_E := '1' a posláání datové hodnoty 0x3
3. časový interval 4,1 ms (tomu odpovídá 205 000 hodinových cyklů)
4. nastavení LCD_E := '1' a posláání datové hodnoty 0x3
5. časový interval 100 μ s (tomu odpovídá 5 000 hodinových cyklů)
6. nastavení LCD_E := '1' a posláání datové hodnoty 0x2
7. časový interval 40 μ s (tomu odpovídá 2 000 hodinových cyklů)
8. nastavení LCD_E := '1' a posláání datové hodnoty 0x3
9. časový interval 40 μ s (tomu odpovídá 2 000 hodinových cyklů)



Obrázek 3.6: Zápis na displej přes 4bitové rozhraní [1]

Popis postupu inicializace je popsán v datasheetu k přípravku Spartan-3E [1].

Po dokončení inicializace 4bitové komunikace se změní hodnota signalizující konec inicializace a pokračuje nastavení displeje, jak je popsáno v odstavci 3.2.2.

■ 3.2.4 Zápis na displej

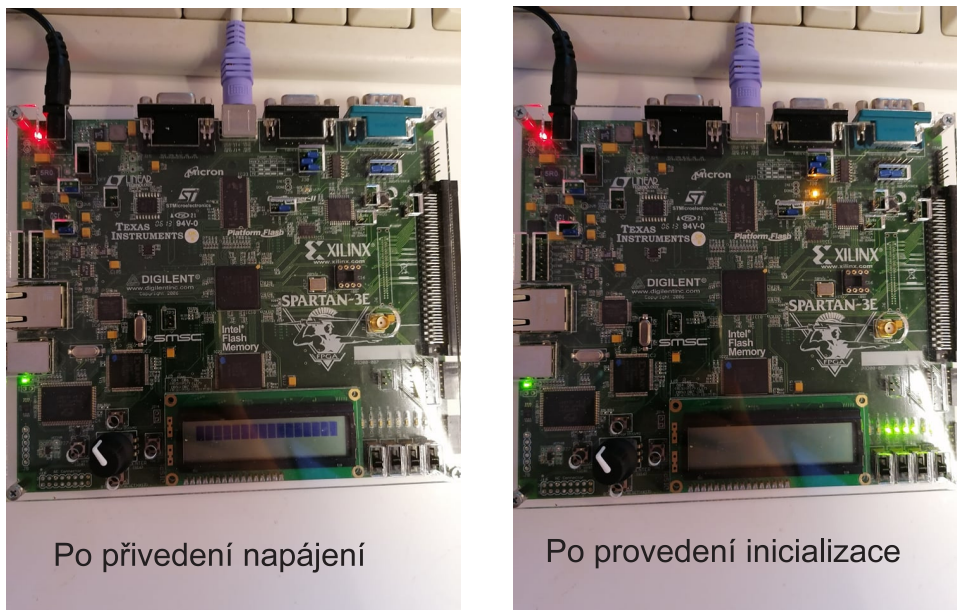
Zápis na displej přes 4bitové rozhraní je na rozdíl od 8bitového rozhraní trochu komplikovanější. Celý zápis probíhá ve dvojici 4bitových slov. Důležité je také časování posílání bitů.

Na obrázku 3.6 je zobrazena celá sekvence zápisu. Jak je vidět nejprve se posílá vyšší 4bitové slovo po té následuje mezera $1 \mu\text{s}$, pak je poslána druhá čtveřice bitů. Zápis je ukončen pauzou $40 \mu\text{s}$.

Druhá část obrázku 3.6 zobrazuje detail posílání jednoho 4bitového slova. Datové signály musí být nastaveny 40 ns před přepnutím signálu LCD_E do stavu logická 1. Slovo posílané přes datové signály a signál LCD_E zůstávají konstantní po dalších 230 ns.

Například zaslání písmene A by probíhalo v těchto krocích:

- nastavení pro datové signály slovo "0100"
- vyčkání 40 ns
- nastavení LCD_E na logickou 1
- vyčkání 230 ns
- interval $1 \mu\text{s}$ a nastavení LCD_E na logickou 0



Obrázek 3.7: Změna výstupu na displeji po inicializaci [zdroj vlastní]

- nastavení pro datové signály slovo "0001"
- vyčkání 40 ns
- nastavení LCD_E na logickou 1
- vyčkání 230 ns
- interval 40 μ s a nastavení LCD_E na logickou 0

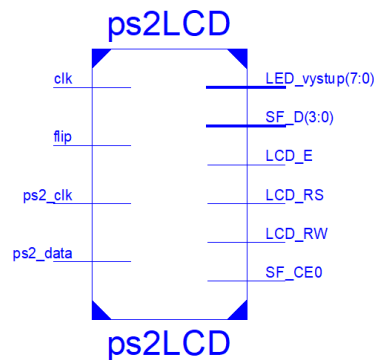
Stavový automat pro zápis přesně kopíruje tyto kroky. Obsahuje pět stavů a je spuštěn vždy v příslušném stavu hlavního stavového automatu, kde je potřeba zaslání dat/instrukcí.

■ 3.2.5 Kontrola funkčnosti LCD displeje

Po syntéze VHDL kódu a naprogramování přípravku proběhne inicializace displeje. Po inicializaci a nastavení se změní displej a je prázdný (obrázek 3.7). V tuto chvíli lze předpokládat, že vše proběhlo v pořádku. pomocí LED lze zjistit v jakém stavu se displej nachází. Pro další kontrolu byly na vstup displeje posílány ASCII kódy pro různé znaky a kontrola zda se znak zobrazuje správně.

■ 3.3 Konečný výstup

Tato část se zabývá samotným propojením vstupu z klávesnice na displej. V následujících odstavcích jsou popsány provedené změny a doplnění v kódu pro správnou funkčnost konečného výstupu.



Obrázek 3.8: Entita výsledného programu [zdroj vlastní]

```

--Převodní tabulka pro přiřazení ASCII kódu k scan kódu klávesy
with scan_kod select
    znak <=
        "00110000" when "01000101", --0
        "00110001" when "00010110", --1
        "00110010" when "00011110", --2
        "00110011" when "00100110", --3
        "00110100" when "00100101", --4
        "00110101" when "00101110", --5
        "00110110" when "00110110", --6
        "00110111" when "00111101", --7
        "00111000" when "00111110", --8
        "00111001" when "01000110", --9

        "01000001" when "00011100", --A

```

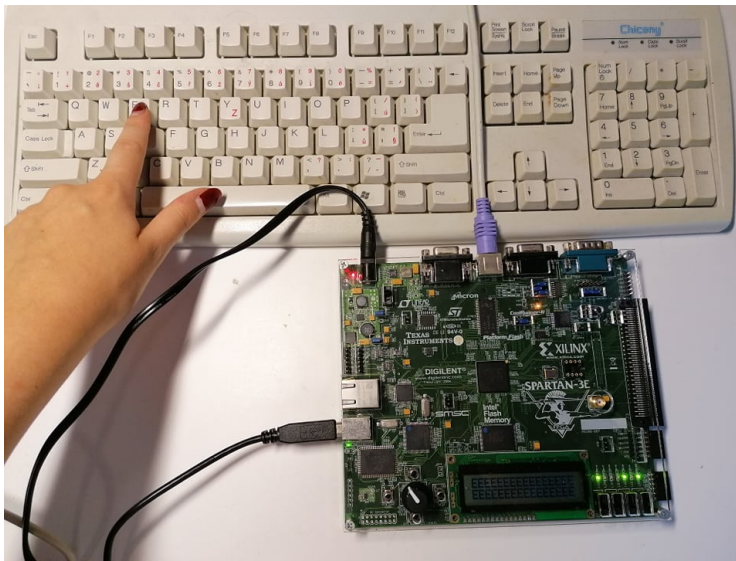
Obrázek 3.9: Ukázka kódu pro převod scan kód na ASCII kód [zdroj vlastní]

Entita, která reprezentuje výsledný program, je zobrazena na obrázku 3.8. Zde jsou vidět jednotlivé vstupy a výstupy.

3.3.1 Převod scan kódu na ASCII kód

Výstupem z klávesnice je scan kód, který reprezentuje pouze fyzickou pozici klávesy. Pro vstup k zobrazení na displeji je potřeba převést scan kód na ASCII kód, který odpovídá písmenu na klávesnici. Pro tento účel byl vytvořen signál znak, do kterého se uloží ASCII hodnota písmene. Převod je realizován pomocí konstrukce "with - select" (obrázek 3.9).

To znamená, že se ke každé hodnotě scan kódu přiřadí hodnota ASCII kódu. Takto byly přiřazeny základní znaky: abeceda, číselné znaky a další základní znaky. Pokud načítání proběhne špatně, nebo je stisknutá nedefinovaná klávesa, je na displeji vypsán znak "*".



Obrázek 3.10: Výsledný projekt [zdroj vlastní]

3.3.2 Zobrazení na displeji

Pro zobrazení znaků na displej je nutné, aby byl displej nejprve inicializován a nastaven. Tato kontrola pro výstup je prováděna v samostatném procesu. Jeho výstupem jsou proměnné, které určují, zda lze na displej zapisovat a zda je načten další znak pro zobrazení. Tento proces obsahuje dvě podmínkové konstrukce.

První podmínkou je kontrola stavu, ve kterém se displej nachází. Pokud se displej nachází ve stavu *hotovo*, přechází se ke kontrole druhé podmínky. Pokud je displej v jakémkoli jiném stavu, jsou signály *dalsi_znak* a *muzu_zapisovat* v hodnotách, které zabraňují zápisu na displej (logická 0).

Druhá podmínka kontroluje, zda je ze vstupu vysíláno. Signál řídící vysílání *zobrazeni* je nastavován, až v procesu konečného výstupu. Pokud vysíláme je signál *dalsi_znak* nastaven na hodnotu logická 1. Tímto se spustí další stav displeje *znak_prijat*, který znak vypíše na displej. Zároveň je signál *muzu_zapisovat* na hodnotu logická 0. V případě, že nevysíláme, zůstává displej ve stavu *hotovo* a signál *muzu_zapisovat* je nastaven na hodnotu, která umožňuje zápis a čeká na vstup.

3.3.3 Vypisování znaku na displej

Na předchozí proces úzce navazuje konečný výstup. V procesu výstupu je rozhodováno, kdy se mění hodnota signálu *zobraz*, který řídí výpis na displej. První podmínkou možnosti výpisu je hodnota signálu *muzu_zapisovat*, který indikuje připravenost displeje přijímat data. Druhou podmínkou je dokončení načítání ze vstupu.

Na obrázku 3.10 je vidět výstup displeje po ošetření základních podmínek pro výstup. Po stisku klávesy, se znak vypíše přes celý displej. Vektor scan

Port	ps2_data	ps2_clk	LED[7]	LED[6]	LED[5]	LED[4]	LED[3]
PIN	G13	G14	F9	E9	D11	C11	F11
Port	LED[2]	LED[1]	LED[0]	SF_CE0	LCD_E	LCD_RS	LCD_RW
PIN	E11	E12	F12	D16	M18	L18	L17
Port	SF_D[3]	SF_D[2]	SF_D[1]	SF_D[0]	clk	flip	
PIN	M15	P17	R16	R15	C9	L13	

Tabulka 3.1: Mapování portů

kódu totiž uchovává svoji hodnotu do stisku další klávesy. Pomocí dvou pomocných vektorů, do kterých se ukládá nová a předchozí hodnota scan kódu, se vyřeší tento problém opakovaného vysílání znaku. Další podmínkou tedy je, že k výpisu dochází pouze při změně scan kódu.

Break kód signalizuje konec výpisu. Podmínkou detekce break kódu je, že aktuální ani předchozí hodnota ve vektoru se nesmí rovnat hodnotě "F0" v šestnáctkové soustavě. Po té se na displej vypíše vždy pouze jeden znak při stisku klávesy.

Vymazání displeje Tento displej má na řádce 16 viditelných znaků. Další znaky na řádce jsou zapamatovány, ale nejsou již viditelné. Pomocí signálu *pocet_znaku* se kontroluje počet již vypsaných znaků na displej. Pokud je vypsáno maximum znaků, nastaví se vymazání displeje. Tímto krokem se hlavní stavový automat přesune do stavu *clear*, jak je vidět na obrázku 3.5.

3.3.4 Využití LED výstupu

LED výstupy byly primárně využívány pro kontrolu funkčnosti designu v průběhu návrhu. Ve finální verzi jsou využity spolu s posuvným tlačítkem pro zobrazování binární hodnoty právě zobrazovaného písmene/znaku. Pomocí posuvného tlačítka na přípravku Spartan-3E lze nastavit, zda zobrazit ASCII kód písmene, nebo scan kód klávesy.

3.3.5 Mapování portů

Jedním z posledních kroků je přiřazení pinů z přípravku k vstupním a výstupním portům entity. Jednotlivé piny jsou vypsány v tabulce 3.1. Toto přiřazování je prováděno v programu PlanAhead.

3.3.6 Výsledný projekt

Výsledkem celého projektu je naprogramovaný přípravek Spartan-3E. K přípravku je připojena PS/2 klávesnice a při stisku klávesy se daná klávesa zobrazuje na LCD displeji, jako je zobrazeno na obrázku 3.11.

Po nahrání programu na přípravek dojde nejprve k inicializaci displeje a jeho prvotnímu nastavení. Když je displej připraven, může začít načítání vstupů z klávesnice. Po stisku klávesy je daná klávesa zobrazena na displeji.



Obrázek 3.11: Výsledný projekt [zdroj vlastní]

Binární hodnota v ASCII, nebo scan kódu, je ukázána pomocí LED. Po popsání řádku je displej vymazán a lze opět zapisovat.

3.3.7 Možné rozšíření

PS/2 vstup by se dal rozšířit o rozpoznávání stisku dvou kláves na jednu, například při stisku klávesy *Shift*. Dále by se dal vstup rozšířit o rozpoznání kláves s rozšířenou funkcí. Vstup z klávesnice by se také dal použít k ovládání dalších periférií přípravku Spartan-3E.

LCD displej by se dal doplnit o znakovou sadu pro český jazyk, nebo vytvořit jiné znaky. Dále by se dalo implementovat reakci displeje na stisk kláves s rozšířenou funkcí. Při stisku klávesy *Delete* by se vymazal poslední znak atd.

Kapitola 4

Závěr

Cílem této práce bylo napsat VHDL kód pro čtení stisku klávesy PS/2 klávesnice a její následné zobrazení na LCD displeji, který je součástí přípravku Spartan-3E.

Pro splnění zadání bylo nejprve potřeba prostudovat teoretické fungování PS/2 klávesnice a LCD displeje s radičem HD44780. Dále bylo potřeba seznámit se s přípravkem Spartan-3E a jeho ovládání pomocí jazyka VHDL a prostředí ISE.

V praktické části jsou vysvětleny jednotlivé části kódu. Je řešen vstup z klávesnice a výstup na displej nejprve samostatně, poté jako celek. Je popsáno fungování a řešení jednotlivých problémů. Praktická část obsahuje i diagramy, úryvky kódu a foto výstupu pro lepší vysvětlení funkce. V závěru praktické části je popsáno fungování programu jako celku.

Výsledný program zpracovává základní vstup z PS/2 klávesnice a zobrazení stisknuté klávesy na vestavěném LCD displeji. Kód lze použít i nezávisle jen pro načítání kláves z PS/2 vstupu, nebo jen výstup na displej. Program by šel rozšířit například o znakovou sadu pro český jazyk, nebo využití kláves s rozšířenou funkcí.



Literatura

- [1] *Spartan-3E Starter Kit Board User Guide..* [online]. Xilinx, Inc., 2006. [cit. 2020-7-13]. Dostupné z: <https://store.digilentinc.com/spartan-3e-starter-board-limited-time/>
- [2] CHAPWESKE, Adam. The PS/2 Mouse/Keyboard Protocol. In: *Computer-Engineering* [online]. 2007. [cit. 2020-7-13]. Dostupné z: <https://www.avrfreaks.net/sites/default/files/PS2%20Keyboard.pdf>
- [3] DRESLER, Tomáš. Inteligentní displeje a jejich připojení k PC. In: *HW server* [online], 2000. [cit. 2020-7-13]. Dostupné z: <https://vyvoj.hw.cz/teorie-a-praxe/dokumentace/inteligentni-displeje-a-jejich-pripojeni-k-pc.html>
- [4] LAFATA, Pavel, Petr HAMPL a Michal PRAVDA. *Digitální technika*. Praha: České vysoké učení technické, 2011. ISBN 978-80-01-04914-3.
- [5] DANĚK, Martin. Programovatelná hradlová pole – FPGA. *Automa* [online]. 2006, vol. 2.[cit. 2020-7-18]. ISSN 1210-9592. Dostupné z: https://automa.cz/cz/casopis-clanky/programovatelná-hradlová-pole-fpga-2006_02_30930_672/
- [6] ŠŤASTNÝ, Jakub. *FPGA prakticky: realizace číslicových systémů pro programovatelná hradlová pole*. Praha: BEN - technická literatura, 2010. ISBN 978-80-7300-261-9.
- [7] *FPGA Fundamentals*. [online]. NI,2020. [cit. 2020-7-13]. Dostupné z: <https://www.ni.com/cs-cz/innovations/white-papers/08/fpga-fundamentals.html>
- [8] KEIM, Robert. What Is an FPGA? An Introduction to Programmable Logic. In: *EETech Media, LLC*. [online]. 2018. [cit. 2020-7-13]. Dostupné z: <https://www.allaboutcircuits.com/technical-articles/what-is-an-fpga-introduction-to-programmable-logic-fpga-vs-microcontroller/>
- [9] PECH, Jan. Nebojte se FPGA. In: *HW server* [online]. 2002 [cit. 2020-7-16]. Dostupné z: <https://vyvoj.hw.cz/teorie-a-praxe/dokumentace/nebojte-se-fpga.html>

- [10] KAFIG, William. *VHDL 101: everything you need to know to get started*. Oxford: Newnes, 2011. ISBN 978-1-85617-704-7.
- [11] DOUŠA, Jiří. *Jazyk VHDL*. Praha: ČVUT, 2003. ISBN 80-01-02670-1.
- [12] ASHENDEN, Peter J. *The student's guide to VHDL. 2nd ed.* Burlington: Morgan Kaufmann, c2008. ISBN 978-1-55860-865-8.
- [13] CHU, Pong P. *FPGA prototyping by VHDL examples: Xilinx Spartan-3 version*. Hoboken: Wiley-Interscience, c2008. ISBN 978-0-470-18531-5.
- [14] PINKER, Jiří a Martin POUPA. *Číslicové systémy a jazyk VHDL*. Praha: BEN - technická literatura, 2006. ISBN 80-7300-198-5.
- [15] *Manuál k řadiči displeje HD44780*. [online]. SparkFun Electronics, 1998. [cit. 2020-7-19]. Dostupné z: <https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>



Příloha A

Obsah přiloženého CD

Na přiloženém CD jsou vypáleny následující soubory:

- Pdf soubor obsahující text bakalářské práce
- Finální VHDL kód v textové formě
- Složka s finálním VHDL kód a ostatními soubory vygenerovanými programem Xilinx ISE
- Stažené pdf soubory s online materiály
- Vytvořené obrázky a fotografie