

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Katedra telekomunikační techniky

**Generování tónů s využitím klávesnicového vstupu PS/2
přípravku Spartan3E**

květen 2020

Bakalant: Vojtěch Baštář
Vedoucí práce: Ing. Pavel Lafata, Ph.D.

Čestné prohlášení

Prohlašuji, že jsem zadanou bakalářskou práci zpracoval sám s přispěním vedoucího práce a konzultanta a používal jsem pouze literaturu v práci uvedenou. Dále prohlašuji, že nemám námitek proti půjčování nebo zveřejňování mé bakalářské práce nebo její části se souhlasem katedry.

Datum: 21. 5. 2020

.....

podpis bakalanta

Poděkování

Děkuji vedoucímu práce Ing. Pavlu Lafatovi, Ph.D. za zapůjčení přípravku, kabelů a periferních zařízení, které byly k realizaci práce potřeba, a za velice užitečné konzultace v průběhu psaní práce. Dále děkuji své sestře Tereze Baštářové za provedení korektury této práce.

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Baštář** Jméno: **Vojtěch** Osobní číslo: **474227**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra telekomunikační techniky**
Studijní program: **Elektronika a komunikace**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Generování tónů s využitím klávesnicového vstupu PS/2 přípravku Spartan3E

Název bakalářské práce anglicky:

DTMF Tone Generator Using PS/2 Keyboard and 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 kódy pro čtení stisknuté klávesy. K přípravku připojte jednoduchý reproduktor, ze kterého generujete tón s různou výškou na základě stisknuté klávesy na klávesnici. Demonstrujte princip tónové volby (DTMF) pomocí 10 vybraných kláves na klávesnici. Vytvořte nezbytné VHDL kódy. Výstupem tak bude jednoduchá demonstrace principu tónové volby na základě vstupu PS/2 klávesnice a výstupu do jednoduchého reproduktoru, 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]

Jméno a pracoviště vedoucí(ho) bakalářské práce:

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

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

Datum zadání bakalářské práce: **08.01.2020**

Termín odevzdání bakalářské práce: **22.05.2020**

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

Student bere na vědomí, že je povinen 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 studenta

Abstrakt:

Bakalářská práce se zabývá obsluhou přípravku Spartan-3E Starter Kit. K ovládání přípravku je použita připojená PS/2 klávesnice a jako výstup slouží externí reproduktor. Fungování přípravku je popsáno jazykem VHDL. Pomocí vytvořeného programu je možné nastavit přípravek jako generátor tónů vhodných k demonstraci principu DTMF v závislosti na vstupu z PS/2 klávesnice.

Klíčová slova: VHDL, FPGA, Spartan-3E Starter Kit, PS/2, DTMF

Abstract:

The bachelor thesis follows the operation of the Spartan-3E Starter Kit board. The PS/2 keyboard is used to control the board and an external speaker is used as an output. The function of the board is described via the VHDL language. Thanks to the created program the board can be used as a generator of tones suitable for the DTMF principle demonstration depending on the input from the PS/2 keyboard.

Keywords: VHDL, FPGA, Spartan-3E Starter Kit, PS/2, DTMF

Obsah

1	Úvod.....	2
2	Teoretická část.....	3
2.1	FPGA.....	3
2.1.1	Architektura FPGA.....	3
2.2	VHDL.....	5
2.2.1	Základní vlastnosti a konstrukce.....	5
2.2.2	Xilinx ISE.....	7
2.3	DTMF.....	8
2.4	Spartan-3E Starter Kit.....	9
2.4.1	Konfigurace.....	9
2.4.2	LED.....	10
2.4.3	Otočné tlačítko.....	10
2.5	Připojené periferie.....	12
2.5.1	PS/2 klávesnice a rozhraní.....	12
2.5.2	Reproduktor.....	15
3	Praktická část, postup řešení.....	16
3.1	Práce s klávesnicí.....	16
3.1.1	Čtení kláves.....	17
3.1.2	Rozpoznávání kláves.....	18
3.1.3	Držení klávesy.....	19
3.2	Práce s výstupem.....	21
3.2.1	Generování tónů.....	21
3.2.2	Míšení signálů pro DTMF.....	24
3.2.3	Využití otočného tlačítka pro nastavení frekvence střídání signálů.....	26
3.3	Projekt jako celek.....	30
3.3.1	Možná rozšíření.....	32
4	Závěr.....	33
	Zdroje.....	34
	Obsah přiloženého CD.....	36

1 Úvod

V současné době mají zařízení generující tóny velmi široké využití. S generováním tónů se setkáváme v odvětvích jako hudební průmysl nebo zdravotnictví. Jeden z typů zařízení, která mohou být využita pro generování tónů, jsou přípravky s programovatelnými hradlovými poli (FPGA přípravky). Vhodným nastavením logických obvodů a případným připojením potřebných periférií lze z použitého přípravku vytvořit například i jednodušší syntezátor. Právě možností využití FPGA přípravku jako generátoru tónů se zabývá tato práce.

Hlavním cílem práce je naprogramovat přípravek Spartan-3E Starter Kit od firmy Xilinx pomocí jazyka VHDL tak, aby bylo s využitím vstupu z PS/2 klávesnice možné demonstrovat princip tónové volby (DTMF). Pro dosažení tohoto cíle je nutné se nejprve seznámit s použitými perifériemi, potřebnými funkcemi přípravku Spartan-3E Starter Kit a také s jazykem VHDL, pomocí kterého jsou přípravek a připojené periferie obsluhovány.

V teoretické části je uvedena teorie související s použitým hardwarem a softwarem. V praktické části práce je pak popsán návrh kódu v jazyce VHDL, pomocí kterého jsou generovány tóny v závislosti vstupu z PS/2 klávesnice. Frekvence generovaných tónů jsou vybrány tak, aby byly vhodné pro demonstraci principu tónové volby. Jako zařízení generující požadované tóny je použit přípravek Xilinx UG230 Spartan-3E Starter Kit. Přípravek neobsahuje vlastní audio výstup, z tohoto důvodu je k němu kromě již zmíněné PS/2 klávesnice připojen i jednoduchý reproduktor 36CS16FN-50BD VANSONIC, který slouží jako výstup pro generované tóny.

2 Teoretická část

V následujících kapitolách je popsána teorie, jejíž znalost je užitečná pro práci s použitým hardwarem a softwarem. Konkrétně se jedná o teoretické informace o programovatelných hradlových polích (FPGA), jazyce VHDL, tónové volbě (DTMF), přípravku Spartan-3E Starter Kit a k němu připojených perifériích.

2.1 FPGA

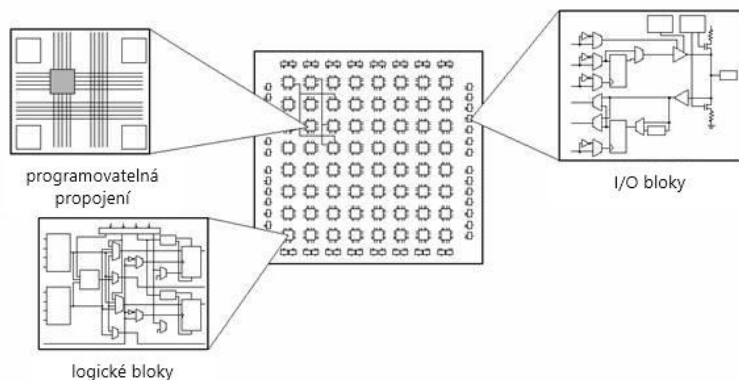
FPGA (*Field Programmable Gate Array*) neboli programovatelná hradlová pole jsou logické integrované obvody, které patří do skupiny programovatelných logických obvodů PLD (*Programmable Logic Device*). Do stejné skupiny patří například obvody SPLD (*Simple Programmable Logic Device*) nebo CPLD (*Complex Programmable Logic Device*). FPGA obvody mohou být naprogramovány koncovým uživatelem tak, aby plnily funkci, která je v danou chvíli požadována. Tím se odlišují od obvodů ASIC (*Application Specific Integrated Circuits*), u kterých je jejich funkce nastavena už výrobcem. Existují OTP (*one-time programmable*) obvody, které se dají naprogramovat jen jednou, ale mnohem častěji se využívají obvody využívající paměť SRAM (*Static Random Access Memory*), které je možné přeprogramovat s tím, jak se návrh koncového uživatele vyvíjí. [4, strana 1-4] [5, strana 3-5] [7]

2.1.1 Architektura FPGA

Pomineme-li specializované bloky, FPGA obvody se skládají z následujících tří základních stavebních prvků: [3, strana 258]

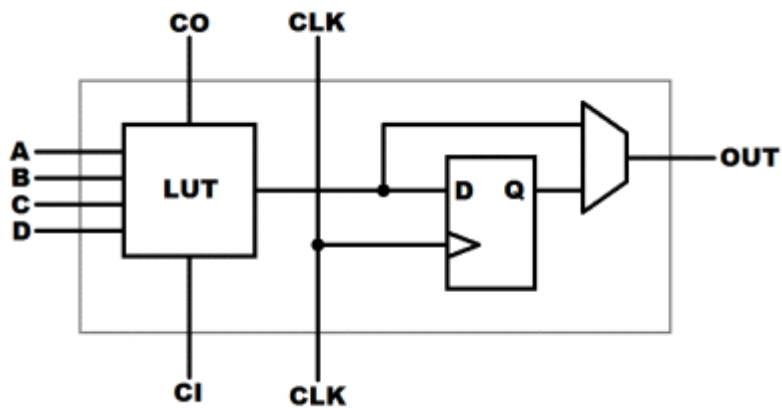
- **Programovatelné logické bloky**, jinak označované také jako logické buňky.
- **Programovatelné vstupní/výstupní bloky** (I/O bloky). Umožňují vstup a výstup signálů dovnitř a ven z FPGA. Tyto bloky obvykle obsahují registr, budič, multiplexor a ochranné obvody. [3, strana 259] [8]
- **Programovatelná propojení**, která slouží k propojení jak logických bloků mezi sebou, tak logických bloků s I/O bloky. Jsou rozmístěna rovnoměrně po celé ploše obvodu. [3, strana 259]

Zmíněné prvky jsou vyobrazeny na obrázku 2.1 včetně jejich rozmístění na FPGA obvodech.



Obr. 2.1: Části FPGA [6]

Programovatelné logické bloky jsou tvořeny dvěma základními komponentami, kterými jsou klopné obvody a vyhledávací tabulky LUT (*lookup tables*). Klopné obvody slouží k přepínání a ukládání logických stavů mezi jednotlivými cykly hodinového signálu. V podstatě se jedná o binární posuvné registry. LUT tabulky fungují jako generátory logických funkcí. Veškerá kombinační logika (AND, OR, NAND a další hradla) jsou implementována v paměti LUT jako pravdivostní tabulky. Každá LUT tabulka má tedy v paměti uložený seznam výstupů pro všechny možné kombinace vstupů. Zjednodušené schéma logického bloku je na obrázku 2.2. [3, strana 284-285] [6] [7]



Obr. 2.2: Zjednodušené schéma logického bloku [7]

Kromě výše zmíněných základních prvků jsou v současné době součástí FPGA obvodů také různé specializované bloky. Paměťové bloky, uspořádané na obvodu ve sloupcích, mohou být stejně jako ostatní zmíněné bloky uživatelem nastaveny podle potřeby. Je možné je nakonfigurovat tak, aby plnily například funkci posuvného registru nebo paměť ROM a RAM. Stejně tak jsou do sloupců uspořádány i násobičky a sčítačky. Jejich umístěním na obvod se výrazně sníží počet využívaných LUT tabulek a klopných obvodů u matematických operací. Dalším typem specializovaného bloku, který byl do FPGA přidán, jsou fázové závěsy (PLL – *Phase Locked Loop*) případně závěsy zpoždění (DLL – *Delay Locked Loop*). Tyto bloky jsou na FPGA využívány zejména k násobení a dělení hodinového signálu, tedy k vytvoření nových kmitočtů. [3, strana 286] [6] [7]

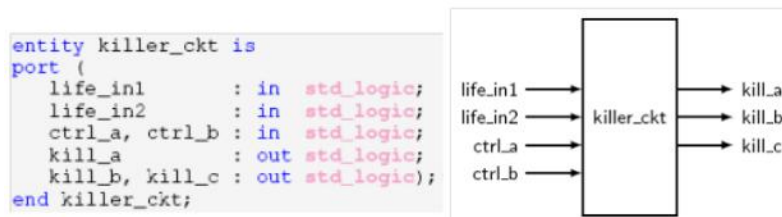
2.2 VHDL

Jedním z hlavních a nejčastěji používaných jazyků určených pro návrh FPGA obvodů je jazyk VHDL. Tento jazyk byl použit i v této práci. VHDL je zkratkou pro *VHSIC Hardware Description Language*, jedná se tedy, jak ze zkratky vyplývá, o jazyk popisující hardware. VHSIC je zkratkou pro *Very High Speed Integrated Circuits*. Jazyk VHDL je určen pro popis chování (návrh) elektrických obvodů nebo systémů a jejich následnou simulaci. Takto popsané číslicové systémy jsou většinou realizovány v hardwaru a je tedy nutné psát kód tak, aby po dokončení mohl projít syntézou. [3, strana 20] [9, strana 1] [10, strana 1]

2.2.1 Základní vlastnosti a konstrukce

VHDL se od běžných programovacích jazyků liší tím, že se jedná takzvaný paralelní jazyk. To znamená, že veškeré instrukce jsou defaultně prováděny ve stejnou chvíli, tedy paralelně, a to nezávisle na množství příkazů. Paralelní způsob chování jazyka VHDL v zásadě odpovídá kombinačním logickým obvodům. Pro realizaci sekvenčních logických obvodů je potřeba zapsat sekvenční příkazy v jazyce VHDL, k čemuž slouží tzv. proces (struktura a funkce procesu je blíže popsána dále v této kapitole). [9, strana 1] [10, strana 1]

Základní jednotkou každého návrhu v jazyce VHDL je **entita**, která v zásadě odpovídá výslednému realizovanému logickému obvodu. V entitě jsou vypsány vstupní a výstupní piny (porty), se kterými se v kódu následně pracuje. Entita představuje jakési pouzdro uzavírající celý popisovaný systém a určuje, kolik má daný systém vstupů/výstupů a jakého jsou typu. [3, strana 27-28] [9, strana 2-4] [10, strana 15-17] [11, strana 18-21]



Obr. 2.3: Příklad deklarace entity a odpovídající vnější náhled [11]

Na entitu úzce navazuje **architektura**. Zatímco entita popisuje vnější reprezentaci popisovaného obvodu, architektura určuje, co daný obvod vlastně dělá. Jinými slovy, architektura popisuje chování entity. Jedna entita přitom může mít více architektur. Architektury mohou obsahovat tři různé typy popisu dané entity:

- **Data-flow:** Je popisován tok dat mezi vstupy a výstupy vestavěných komponent jazyka VHDL, jako jsou například operátory AND, OR nebo XOR. Obvody jsou sestavovány paralelními příkazy. [11, strana 51-53][12, strana 4]
- **Behaviorální:** Na rozdíl od data-flow popisu neposkytuje detaily o tom, jak je návrh implementován na hardware. Tímto popisem se modeluje, jak budou výstupy obvodu reagovat na vstupy, tedy jak se bude daný systém chovat. Behaviorálně popsané obvody mohou být realizovány v paralelním i sekvenčním prostředí. Behaviorální popis byl použit i v této práci. [11, strana 51, 53] [12, strana 4]
- **Strukturální:** Tímto stylem se propojují již existující VHDL modely (vytvořené pomocí data-flow nebo behaviorálního popisu). Strukturální styl tedy umožňuje popis zapojení logických obvodů, ale naopak neumožňuje popis logických funkcí, které určují fungování těchto celků. [11, strana 51] [12, strana 4]

Architektura má dvě části. První část, kterou architektura nutně obsahovat nemusí, je část deklarační. V této části jsou deklarovány veškeré signály a konstanty. Druhou částí je samotný kód, ve kterém se pracuje se zmíněnými vstupními/výstupními porty a také případně s deklarovanými signály a konstantami. Příklad architektury hradla NAND je na následující obrázku. [3, strana 28] [9, strana 2, 4-5] [10, strana 17] [11, strana 23]

```

ARCHITECTURE myarch OF nand_gate IS
BEGIN
  x <= a NAND b;
END myarch;

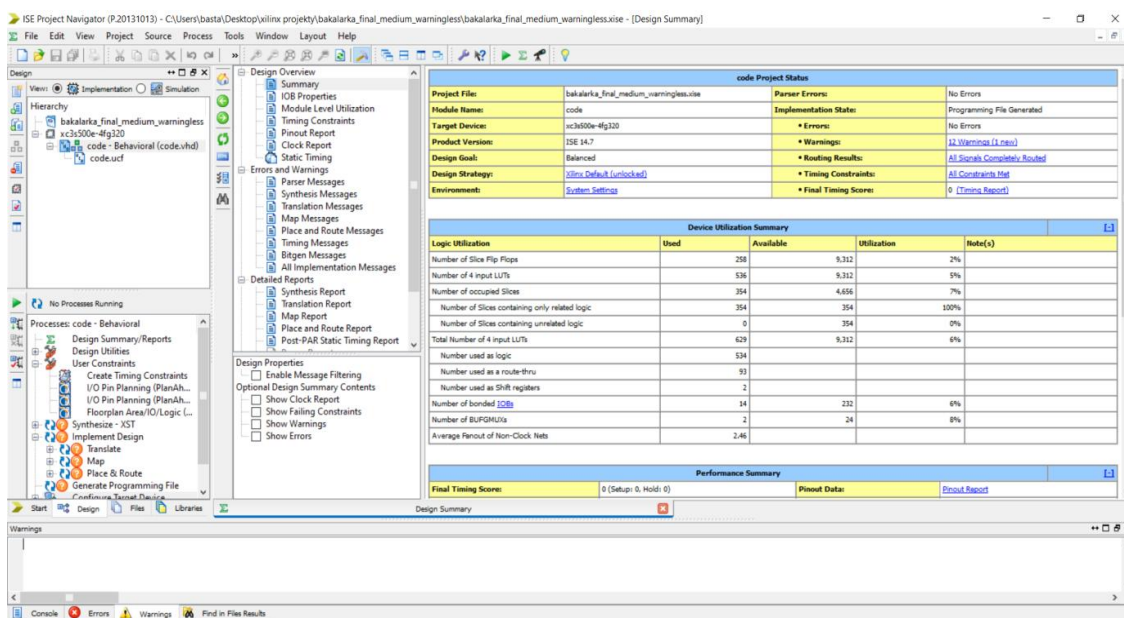
```

Obr. 2.4: Příklad architektury hradla NAND [10]

Důležitou součástí VHDL kódu jsou **procesy**. Procesy jsou umístěny v architektuře daného návrhu a může jich zde být libovolný počet. Zásadní vlastností procesů je, jak už bylo zmíněno, že příkazy umístěné uvnitř procesu jsou prováděny sekvenčně. Zároveň však platí, že procesy jsou příkazem uvnitř architektury a všechny procesy tak probíhají současně (paralelně). Syntaxe deklarace procesu je obdobná architektuře. Obsahuje deklarační část, ve které jsou umístěné proměnné využívané v daném procesu, a následně část kódu, kde se pracuje s těmito proměnnými a signály. Navíc je nutné deklarovat takzvaný citlivostní seznam, ve kterém jsou vypsány signály, při jejichž změně je proces spuštěn. [3, strana 47-48] [9, strana 3, 9-10] [10, strana 91-92] [11, strana 53-55]

2.2.2 Xilinx ISE

K tvorbě kódu jsem použil program Xilinx ISE. Jedná se o softwarovou pomůcku od firmy Xilinx, která je určena pro simulaci a syntézu návrhů v jazyce VHDL nebo Verilog. Tento software je určený pro práci s FPGA obvody od firmy Xilinx (jako například v této práci použitý Spartan-3E) a nemůže být tedy využíván k práci se zařízeními od jiných firem. Program Xilinx ISE byl použit při tvorbě této práce. Uživatelské rozhraní je vidět na následujícím obrázku. [8] [13]



Obr. 2.5: Uživatelské rozhraní Xilinx ISE

2.3 DTMF

DTMF (*Dual-tone multi-frequency*), v češtině tónová volba, je systém signalizačních tónů využívaný zejména v telefonii. Nejznámějším využitím je přenos volby telefonního čísla. U DTMF je každý symbol reprezentován kombinací dvou různých tónů. Tyto tóny jsou rozděleny na dvě skupiny, a to na tóny s nízkou frekvencí a tóny s vysokou frekvencí. Každý signál tónové volby pak využívá po jednom tónu z obou skupin. Tím se zabrání chybnému vyhodnocení harmonických frekvencí jakožto součástí signálu. Osm možných frekvencí (čtyři v každé skupině) je zvoleno tak, aby od sebe byly dostatečně vzdálené a bylo tak možné je bez problémů rozlišit. Klávesy využívané pro tónovou volbu a jim odpovídající frekvence jsou uvedeny v následující tabulce.

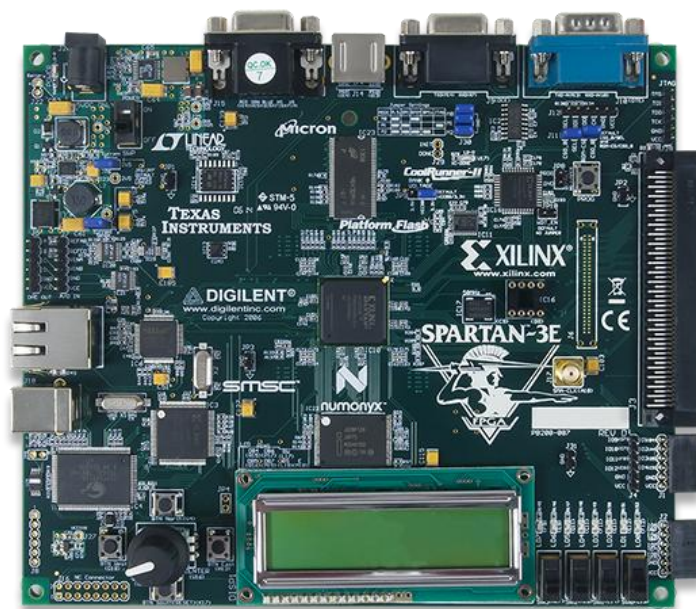
<i>f</i> [Hz]	1209	1336	1477	1633
697	1	2	3	A
770	4	5	6	B
852	7	8	9	C
941	*	0	#	D

Tab. 2.1: Klávesy a frekvence používané u DTMF

Klávesy A, B, C a D byly původně používány jako součást amerického armádního telefonního systému, ale v současnosti se už téměř nevyužívají. Délka DTMF tónů závisí na konkrétním využívaném systému. V současnosti využívané dekodéry pracují s tóny o délce pár desítek milisekund a stejně dlouhými mezerami mezi nimi. [14] [15] [16]

2.4 Spartan-3E Starter Kit

Hlavní částí přípravku Spartan-3E Starter Kit od společnosti Digilent je FPGA XC3S500E z rodiny Spartan-3E, který obsahuje více než 10 000 logických bloků, až 232 I/O bloků, 32-bitový RISC procesor a DDR paměť. Kromě toho jsou v přípravku zabudovány porty PS/2 a VGA nebo 2 devíti pinové RS232 porty pro DTE i DCE roli. Součástí je i osm LED, čtyři přepínače, pět tlačítek včetně jednoho tlačítka otočného, LCD displej a hodinový oscilátor pracující s frekvencí 50 MHz. K nahrání programu do přípravku slouží USB port typu B, propojený na rozhraní JTAG. Spartan-3E Starter Kit je plně kompatibilní s vývojovým softwarem Xilinx ISE a všemi jeho součástmi. [1, strana 12] [17, strana 2]



Obr. 2.6: Přípravek Spartan-3E Starter Kit [2]

2.4.1 Konfigurace

Spartan-3E Starter Kit nabízí několik konfiguračních možností v závislosti na nastavení jumperu J30. V této práci byla používána varianta nahrání návrhu přímo na FPGA přes JTAG pomocí zabudovaného USB rozhraní. Jinou možností je využití nejdříve mezikroku v podobě naprogramování některé ze tří zabudovaných Flash PROM pamětí a až následně nahrát návrh na FPGA s pomocí některého ze zabudovaných rozhraní (konkrétně SPI nebo BPI). [1, strana 26-27]

Další jumpery jsou určeny k nakonfigurování FPGA pomocí SPI rozhraní (jumper JP10, J11 a J12) nebo k nastavení napájecího napětí všech I/O pinů (jumper JP9). Toto napájecí napětí může být nastaveno na 3,3 V nebo 2,5 V. Stejným napětím je však napájen i 50 MHz oscilátor (zdroj hodinového signálu), který vyžaduje napájecí napětí 3,3 V a při přepnutí na 2,5 V by nemusel fungovat správně. [1, strana 22, 90]

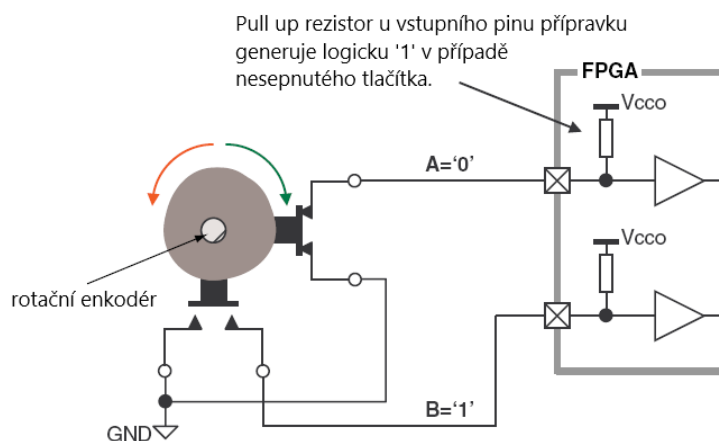
2.4.2 LED

Spartan-3E Starter Kit obsahuje osm LED, které jsou umístěny nad posuvnými spínači v pravém dolním rohu přípravku (viz obrázek 2.6 v kapitole 2.4). LED jsou značeny LED0 až LED7, přičemž LED7 je úplně vlevo. Každá LED má jednu stranu připojenou na zem. Druhá strana je přes rezistor s odporem $390\ \Omega$ připojena k výstupnímu pinu přípravku. Je-li na tento pin přivedena logická 1, LED se rozsvítí. [1, strana 19-20]

2.4.3 Otočné tlačítko

Otočné (rotační) tlačítko je na přípravku umístěno v levém dolním rohu mezi čtyřmi tlačítky (viz obrázek 2.6 v kapitole 2.4). Kromě otáčení je možné pomocí otočného tlačítka vyhodnotit i jeho stisknutí. [1, strana 17]

Rotací otočného tlačítka jsou spínány a rozepínány dvě tlačítka (spínače). Tyto spínače jsou ovládány pomocí signálů (výstupů rotačního tlačítka) A a B. Pomineme-li možnost vyhodnocení stisknutí, má tedy rotační tlačítko dva výstupy připojené ke vstupním pinům přípravku. Zapojení otočného tlačítka je naznačeno na následujícím obrázku. [1, strana 17-18]



Obr. 2.7: Zapojení otočného tlačítka k přípravku [1]

Hodnota signálů A a B je závislá na směru otáčení tlačítka. Podle toho, kterým směrem je s tlačítkem otáčeno, je tedy určeno pořadí, v jakém se oba spínače sepínají a rozepínají. V případě, kdy je tlačítko v klidu a k žádnému otáčení nedochází, jsou oba spínače sepnuté. Sepnutím spínače se kontakt připojí na zem, což odpovídá hodnotě logická 0 na vstupním pinu přípravku. Logická 1 se tedy na pinu objevuje v případě rozepnutí spínače, a to díky pull up rezistoru, který je umístěn u obou spínačů (viz obrázek 2.7). [1, strana 17-18]

Při práci s otočným tlačítkem je nutné zohlednit vliv mechanických kontaktů na signály A i B. Na vstupech do přípravku vzniká šum (rychlé náhodné překmitávání mezi logickými hodnotami), které výrazně ovlivňuje vyhodnocení otáčení tlačítka. Více o vyhodnocení otáčení a vlivu mechanického šumu pojednává kapitola 4.3.4. [1, strana 17-18]

2.5 Připojené periferie

V následujících kapitolách jsou popsány parametry periferních zařízení, která jsou k přípravku připojena, a částečně i teorie související s jejich obsluhou. Těmito periferiemi jsou PS/2 klávesnice a reproduktor 36CS16FN-50BD VANSONIC.

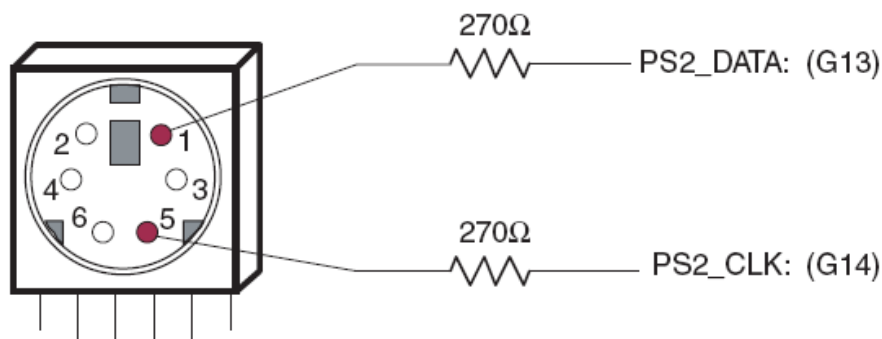
2.5.1 PS/2 klávesnice a rozhraní

Přípravek (konkrétně tóny, které jsou přípravkem generovány) je ovládán pomocí PS/2 klávesnice. V následujících kapitolách budou popsány náležitosti komunikace s PS/2 klávesnicí, stejně jako základní informace o PS/2 konektoru. Teorie popsaná v těchto kapitolách je částečně opakována a následně rozvíjena v kapitolách popisujících praktické řešení.

2.5.1.1 PS/2 konektor

Klávesnice je připojena k PS/2 portu, který je umístěn uprostřed horního okraje přípravku Spartan-3E Starter Kit (viz obrázek 2.6). Port je označen jako J14. [1, strana 61]

PS/2 konektory mohou mít pět nebo šest pinů s tím, že oba typy jsou z elektronického hlediska totožné a jediný rozdíl je v rozmístění pinů. V tomto projektu a obecně častěji používaným typem konektoru je konektor s šesti piny. Rozmístění pinů na PS/2 konektoru je vyobrazeno na obrázku 2.8, signály odpovídající jednotlivým pinům pak v tabulce 2.2. Na obrázku 2.8 je vyobrazen konektor typu „samice“, tedy konektor na přípravku Spartan-3E Starter Kit. Konektor na straně PS/2 klávesnice je typu „samec“ a oproti obrázku má zrcadlově otočené rozmístění pinů. Z uvedených pinů je k FPGA připojen jen datový a hodinový pin. Signály z těchto dvou pinů jsou díky zapojení přes pull up rezistory v případě neaktivity (z klávesnice nejsou vysílána žádná data) udržovány ve stavu logická 1, podobně jako tomu je u otočného tlačítka (viz kapitola 2.4.3). [1, strana 61] [2, strana 1-2]



Obr. 2.8: Scan kódy použité PS/2 klávesnice [1]

číslo pinu	signál
1	datový signál
2	nezapojeno
3	GND (země)
4	+5V napájecí napětí
5	hodinový signál
6	nezapojeno

Tab. 2.2: Signály odpovídající pinům PS/2 konektoru [1]

PS/2 klávesnice vyžaduje pro svoje správné fungování napájecí napětí v rozmezí 4,5 V až 5,5 V. To je zajištěno připojením zemnicího a napájecího pinu klávesnice, konkrétně pinů 3 a 4 (viz obrázek 2.8), k zemi a napájecímu napětí 5 V na přípravku. Spartan-3E Starter Kit sice nepatří mezi přípravky tolerující 5 V, ale je schopen komunikovat s 5 V zařízeními s využitím proud-omezujících rezistorů. [1, strana 65] [2, strana 2-3]

2.5.1.2 Scan kód

Každá klávesa má svůj unikátní scan kód, který má podobu dvou čísel v šestnáctkové soustavě. Pomocí scan kódu je možné rozpoznat, jaká klávesa byla právě stisknuta. Scan kódy všech kláves použité PS/2 klávesnice jsou uvedeny na obrázku 2.9. [1, strana 62-63] [2, strana 8-9]

ESC 76	F1 05	F2 06	F3 04	F4 0C	F5 03	F6 0B	F7 83	F8 0A	F9 01	F10 09	F11 78	F12 07	↑ E0 75	
~ 0E	1! 16	2@ 1E	3# 26	4\$ 25	5% 2E	6^ 36	7& 3D	8* 3E	9(46	0) 45	-_ 4E	=+ 55	Back Space ← 66	→ E0 74
TAB 0D	Q 15	W 1D	E 24	R 2D	T 2C	Y 35	U 3C	I 43	O 44	P 4D	[{ 54]} 5B	\ 5D	← E0 6B
CapsLock 58	A 1C	S 1B	D 23	F 2B	G 34	H 33	J 3B	K 42	L 4B	;; 4C	'" 52	Enter ↵ 5A	↓ E0 72	
↵ Shift 12	Z 1Z	X 22	C 21	V 2A	B 32	N 31	M 3A	,< 41	>. 49	/? 4A	↵ Shift 59			
Ctrl 14	Alt 11	Space 29						Alt E0 11	Ctrl E0 14					

Obr. 2.9: Scan kódy použité PS/2 klávesnice [1]

Je-li klávesa držena, je po krátké době opakovaně odeslán scan kód (přibližně každých 100 ms) držené klávesy až do okamžiku, kdy je klávesa puštěna. Doba, po kterou při držení klávesy nejsou odesílána žádná data, se nazývá typematic delay a standardně trvá v rozmezí 0,25 s až 1 s. [1 strana 62] [2, strana 9-10]

Ve chvíli, kdy je klávesa uvolněna, je namísto scan kódu odeslán break kód, který má hodnotu F0. Break kód je odeslán vždy po uvolnění klávesy, ať už byla klávesa držena nebo jen stisknuta. [2, strana 9-10]

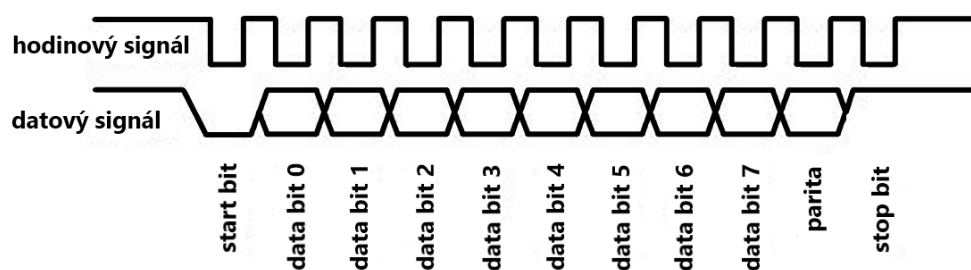
2.5.1.3 Komunikace s klávesnicí

Klávesnice je schopna komunikovat dvěma směry, a to ve směru od hostitele (v našem případě se jedná o přípravek Spartan-3E Starter Kit) ke klávesnici nebo od klávesnice k hostiteli. Komunikace je vždy řízena hodinovým signálem PS/2 klávesnice, přičemž komunikace je pokaždé zahajována přepnutím hodinového signálu do stavu logická 0. [1, strana 61-62] [2, strana 3-4]

Komunikace ve směru klávesnice-hostitel probíhá pomocí 11-ti bitových slov odesílaných po datové lince z klávesnice. Tato slova se skládají z následujících bitů:

- **Jeden start bit**, který má vždy hodnotu logická 0.
- **Osm datových bitů**, z nichž první je LSB (least significant bit).
- **Jeden bit parity**, který je využíván k detekci chyb a má hodnotu závislou na hodnotě datových bitů.
- **Jeden stop bit**, který má vždy hodnotu logická 1.

Jednotlivé bity jsou posílány vždy při přepnutí hodinového signálu na hodnotu logická 1, přičemž hodinový signál klávesnice pracuje s frekvencí 10 až 16,7 kHz. Průběh komunikace, tedy průběh odeslání všech jedenácti bitů, je vyobrazen na obrázku 2.10.



Obr. 2.10: Komunikace klávesnice s přípravkem [vlastní zpracování podle 2]

Osm datových bitů dohromady tvoří informaci o tom, jaká klávesa byla právě stisknuta, a to v podobě scan kódu (viz předchozí kapitola) dané klávesy. Pro určení scan kódu je potřeba převést datové bity z dvojkové soustavy do šestnáctkové soustavy. [1, strana 63] [2, strana 4-5]

Komunikace v opačném směru, tedy ve směru hostitel-klávesnice, má několik odlišností. Chce-li hostitel zahájit komunikaci, musí nejprve uvést hodinový a datový signál do takzvaného stavu „Request-to-send“, přičemž klávesnice v pravidelných intervalech kontroluje, zda k tomuto stavu nedošlo. Klávesnice opět generuje hodinový signál, stejně jako u opačného směru, bity jsou však nyní odesílány po datové lince od hostitele. Za každým 11-ti bitovým slovem je navíc jeden „acknowledgement“ bit. Detaily komunikace ve směru hostitel-klávesnice zde nejsou uvedeny, neboť v této práci nebyla využívána. [2, strana 5-6]

2.5.2 Reproduktor

Spartan-3E Starter Kit nemá na rozdíl od jiných FPGA přípravků (například Nexys 4) zabudovaný vlastní audio výstup. Funkci zdroje zvuku tedy plní externě připojený reproduktor 36CS16FN-50BD VANSONIC, který je vyfocen na obrázku 2.11. Reproduktor je k přípravku připojen přes šesti-pinový konektor J1 (konkrétně k pinu IO4), který se nachází v dolní části pravého okraje přípravku (viz obrázek 2.6). Reproduktor má podle dokumentace normální jmenovitý výkon 0,5 W a jmenovitý odpor 16 Ω . Reproduktor by měl být schopný generovat tóny o frekvencích v rozsahu 300-6500 Hz. Tento rozsah pokrývá všech osm frekvencí využívaných u tónové volby, takže lze použítý reproduktor považovat za vhodný k demonstraci DTMF. [18]



Obr. 2.11: Reproduktor 36CS16FN-50BD VANSONIC [18]

3 Praktická část, postup řešení

V následujících kapitolách bude popsána praktická část bakalářské práce, tedy samotný návrh VHDL kódu a postup řešení jednotlivých problémů, které v průběhu tvorby kódu nastaly. V kapitolách je kromě průběhů použitých procesů stručně popsána také teorie, která s problematikou daných procesů úzce souvisí.

Jak už je zmíněno v úvodu, finálním cílem práce je vytvoření generátoru tónů, pomocí kterého je možné demonstrovat princip tónové volby (DTMF). Celý návrh je možné rozdělit na dva samostatné úkoly:

- rozpoznávání stisknuté klávesy na PS/2 klávesnici,
- generování tónu o určité frekvenci.

V použitém řešení bylo postupováno právě tímto způsobem a problematika těchto úkolů byla řešena odděleně. Kromě zmíněných dvou částí bylo do řešení přidáno ještě využití otočného tlačítka pro nastavení rychlosti střídání signálů na výstupu. Po vyřešení dílčích úkolů bylo jednodušší dát dohromady celkový kód.

Kód je rozdělen do několika procesů. Kromě těchto vstupních (načítání z klávesnice) a výstupních (generování tónů) procesů jsou používány pomocné funkce, které slouží k nastavování frekvencí signálů, ze kterých jsou skládány tóny DTMF (viz kapitola 3.3).

V průběhu vytváření jednotlivých částí kódu jsem pomocí testování konkrétních situací objevil několik problémů v chování přípravku, které bylo nutné řešit vhodnou úpravou kódu. Většina těchto problémů byla způsobena specifickými vlastnostmi PS/2 klávesnice. Jiné problémy souvisely s nedostatečným ošetřením všech situací, které mohly v průběhu programu nastat. V následujících kapitolách je popsáno jen finální řešení všech vyvstanuvších problémů.

3.1 Práce s klávesnicí

Jak bylo zmíněno v kapitole 2.5.1, k FPGA je připojen datový a hodinový pin PS/2 klávesnice. Výstupy těchto pinů mají defaultně hodnotu logická 1. Hodnota výstupů se začne měnit v případě, kdy dochází k přenosu dat z klávesnice. V této kapitole bude popsán způsob vyhodnocení komunikace ve směru z klávesnice do přípravku Spartan-3E Starter Kit. Pro ověření správného fungování čtení kláves byly využity zelené LED na přípravku.

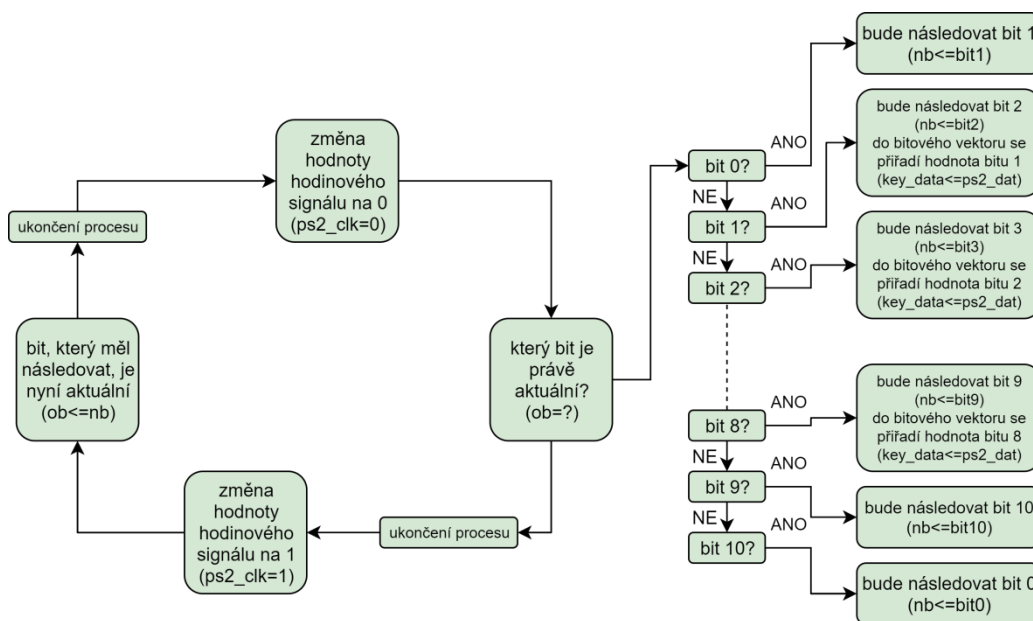
3.1.1 Čtení kláves

V citlivostním seznamu procesu (viz kapitola 2.2.1), pomocí kterého je rozpoznávána stisknutá klávesa, je datový a hodinový signál PS/2 klávesnice. Proces je tedy spuštěn vždy, když se změní hodnota jednoho z těchto dvou vstupních signálů. Tyto signály jsou označeny jako ps2_clk (hodinový signál), a ps2_data (datový signál).

Jak je vidět na obrázku 2.10 v kapitole 2.5.1.3, klávesnice pošle další bit vždy, když je přítomna vzestupná hrana hodinového signálu a přepnutí hodinového signálu na sestupnou hranu naopak značí, že jsou přijatá data již platná. V první fázi procesu se tedy detekuje přepnutí hodinového signálu z klávesnice na logickou hodnotu 1. Je-li tato podmínka splněna, dojde k posunutí na následující bit (viz diagram procesu).

Jediná další varianta, která může po spuštění procesu nastat, je přepnutí hodinového signálu na logickou hodnotu 0. V tomto případě jsou čteny hodnoty bitů (viz diagram procesu). Start, stop a parity bit, který není v práci uvažován s ohledem na jednoduchost komunikace, jsou přeskočeny, protože s jejich hodnotami v následujících procesech není potřeba pracovat, a datové bity (2. až 9. bit každého 11-ti bitového slova) jsou ukládány do bitového vektoru. Ten v sobě má po ukončení procesu uloženou hodnotu scan kód stisknuté klávesy ve dvojkové soustavě.

Na obrázku 3.1 je pomocí diagramu vyobrazen průběh vyhodnocení stisknutí klávesy výše uvedeným procesem. Je-li stisknuta například klávesa „A“, v bitovém vektoru bude po přečtení všech jedenácti bitů uložena hodnota 0001 1100, přičemž jednotlivé bity jsou přenášeny od LSB po MSB, tedy v opačném pořadí, než jak je zde uvedena výsledná binární hodnota. Tato hodnota po převedení do šestnáctkové soustavy odpovídá číslu 1C, což je scan kód klávesy „A“. Se scan kódem se následně pracuje v procesu pro generování tónů.



Obr. 3.1: Průběh procesu určeného pro čtení stisknuté klávesy

3.1.2 Rozpoznávání kláves

Prvním výsledkem, kterého bylo při tvorbě kódu dosaženo, bylo rozpoznání stisknutí jakékoliv klávesy přípravkem. Tedy zajištění reakce přípravku Spartan-3E na přenos dat z PS/2 klávesnice. S tím úzce souvisí další krok, jímž bylo rozpoznání konkrétní klávesy. Rozpoznávání konkrétní klávesy je realizováno pomocí porovnání hodnoty vektoru obsahujícího scan kód stisknuté klávesy (v kódu pojmenován `key_data`) se scan kódem klávesy požadované (například 1C, má-li přípravek reagovat na stisknutí klávesy „A“).

V této fázi přípravek sice správně rozsvítil požadovanou LED v závislosti na stisknutí klávesy, ale LED zůstala rozsvícena, i když nebyla stisknuta požadovaná klávesa, a to do chvíle, než byla stisknuta jiná klávesa. Příčinou tohoto chování bylo, že podmínka pro rozsvícení LED (například stisknutí klávesy „A“) byla plněna při každém průběhu procesu, neboť vektor obsahující scan kód stisknuté klávesy nebyl nikdy nulován.

Tento problém není vyřešen v procesu čtení z klávesnice, neboť nulovat scan kód ve stejném procesu, ve kterém je ukládán by mohlo být problematické. Problém s vynulováním výstupu je tedy řešen až ve „výstupním“ procesu, kde dochází ke kontrole podmínek pro rozsvícení LED a generování tónů.

Jak je uvedeno v kapitole 2.5.1.3, klávesnice odešle namísto scan kódu takzvaný break kód v případě uvolnění stisknuté klávesy. Break kód má hodnotu F0. Samotná kontrola odeslání break kódu by však popisovaný problém nevyřešila, neboť klávesnice po break kódu odešle do přípravku opět scan kód uvolněné klávesy. [1, strana 62]

Pro řešení problému je použita pomocná proměnná help typu std_logic a čítač n, jenž je inkrementován s každou náběžnou hranou hodinového signálu přípravku. Proměnná help má defaultně hodnotu logická 0 a na hodnotu 1 se přepne v případě, že byla právě uvolněna nějaká klávesa. Čítač n slouží k překlenutí opětovného odeslání scan kódu uvolněné klávesnice. Ve chvíli, kdy čítač dosáhne své maximální hodnoty, je proměnná help opět přepnuta na hodnotu logická 0. Maximální hodnota n je 5000000, což vzhledem k frekvenci hodinového signálu odpovídá době 100 ms. Úryvek kódu, ve kterém se s proměnnou help a čítačem n pracuje, je na následujícím obrázku.

```
if (key_data = x"F0") then -- je-li poslan break code, nastavi se pomocna promena "help" na hodnotu 1
    help := '1'; -- plati-li help=1, tak je vystup prerusen
    n := 0; -- promenna n kontroluje, jak dlouho byl prerusen vystup
end if;

if (help = '1' and n /= 5000000) then
    n := n + 1;
end if;

if (n = 5000000) then -- po 100ms je vystup obnoven (help=0)
    help := '0';
end if;
```

Obr. 3.2: Úryvek kódu s pomocnými proměnnými „help“ a „n“

Má-li proměnná help hodnotu logická 1, je vypnut výstup přípravku (LED jsou zhasnuté a není generován žádný tón).

3.1.3 Držení klávesy

Na problém uvedený v předchozí kapitole a jeho řešení úzce navazuje problematika držení klávesy. Problematickým je zejména v kapitole 2.5.1.2 zmíněný typematic delay klávesnice, který má v tomto konkrétním případě délku přibližně 0,5 s. Tato prodleva mezi prvním odeslaným 11-ti bitovým slovem a těmi následujícími má za následek krátké přerušení výstupu přípravku. Podle [1] posílá PS/2 klávesnice v případě držení klávesy scan kód každých přibližně 100 ms. Tím se však v řešení není potřeba zabývat, neboť typematic delay je několikanásobně delší. Podaří-li se překlenout typematic delay, bude tím vyřešena i periodicky se opakující doba 100 ms, kdy hodinový a datový signál klávesnice zůstává přepnut na horní hranu.

Čítač counter je upraven tak, aby počítal až do hodnoty 35000000 (0,7 s), což s velkou rezervou stačí na překlenutí požadované doby. Tento čítač je závislý jak na hodnotě hodinového signálu klávesnice, tak na pomocné proměnné help, která byla zmíněna v předchozí kapitole. Counter je inkrementován s každou náběžnou hranou hodinového signálu přípravku v případě, kdy je hodinový signál klávesnice v klidovém stavu (ps2_clk = '1'). Čítač je vynulován v případě zahájení přenosu dat z klávesnice (ps2_clk = '0') a maximální hodnota je mu naopak přiřazena v případě, kdy byla uvolněna stisknutá klávesa (help = '1'). Na obrázku 3.3 je úryvek kódu, ve kterém se pracuje s čítačem counter. V případě, kdy counter dosáhne své maximální hodnoty, je vypnut výstup přípravku.

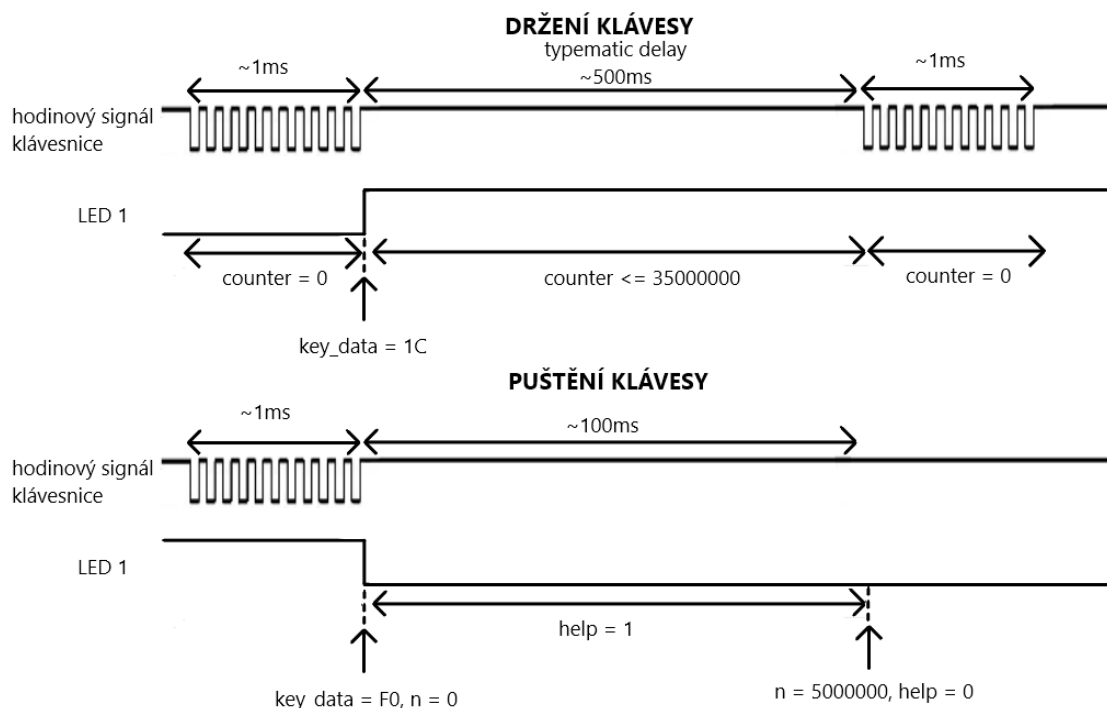
```

if (ps2_clk = '1' and counter /= 35000000) then -- kontrola přenosu z klávesnice; v případě klidu se ukončí výstup
  counter := counter + 1;
elsif (help = '1') then -- přiřazení hodnoty, která ukončí výstup
  counter := 35000000;
elsif (ps2_clk = '0') then -- byl-li zahájen (probiha-li) přenos z klávesnice, pomocný counter se vynuluje
  counter := 0;
end if;

```

Obr. 3.3: Úryvek kódu s pomocnou proměnnou „counter“

Na následujícím obrázku je ilustrováno řešení problematiky držení klávesy pro příklad, kdy by se po stisknutí klávesy „A“ měla rozsvítit LED1.



Obr. 3.4: Ilustrace řešení problému způsobeného držením klávesy

3.2 Práce s výstupem

Přípravek Spartan-3E Starter Kit nemá zabudovaný žádný audio výstup. Jako zdroj generovaných tónů je tedy využít externí reproduktor připojený k jednomu z pinů přípravku (viz kapitola 2.5.2). V následujících kapitolách je popsán způsob řešení generování tónů vhodných pro demonstraci principu tónové volby. Součástí řešení je i nastavení LED na přípravku.

3.2.1 Generování tónů

Pro generování tónu je potřeba získat logickou proměnnou, která bude periodicky měnit svou hodnotu. Rychlost, kterou se bude měnit hodnota proměnné, bude odpovídat frekvenci generovaného tónu. Tato proměnná následně může být přivedena na pin, ke kterému je připojen jednoduchý externí reproduktor. Signál, který vstupuje do reproduktoru, je posloupnost pravidelně se střídajících jedniček a nul. Nejedná se tedy o sinusový signál, ale o obdélníkový signál.

Přípravek Spartan-3E Starter Kit obsahuje oscilátor, který generuje hodinový signál o frekvenci 50 MHz. Tento hodinový signál je využít pro generování požadovaných tónů. Frekvenci hodinového signálu je nutné vydělit tak, abychom získali frekvenci požadovanou. Zmíněné dělení frekvence lze realizovat několika způsoby.

Jednou z možností je využití bitového vektoru, v tomto případě konkrétně šestnáctibitového, který funguje jako čítač. Čítač počítá každé přepnutí hodinového signálu na hodnotu logická 1. Vezmeme-li tedy hodinový signál o frekvenci 50 MHz, takzvaný least significant bit (LSB), označíme ho například jako čítač (0), se bude měnit s frekvencí 25 MHz. Bit čítač (1) se pak bude měnit s frekvencí 12,5 MHz a tak dále s tím, že každý další bit se bude měnit s frekvencí 2x menší než bit předchozí. Most significant bit (MSB) čítač (15), který je přiveden na vstup reproduktoru, se pak bude měnit s frekvencí 763 Hz. Pro frekvence, kterými se mění hodnoty jednotlivých bitů, platí obecně vztah:

$$f_B = \frac{f_{clock}}{2^{B+1}}, \quad (1)$$

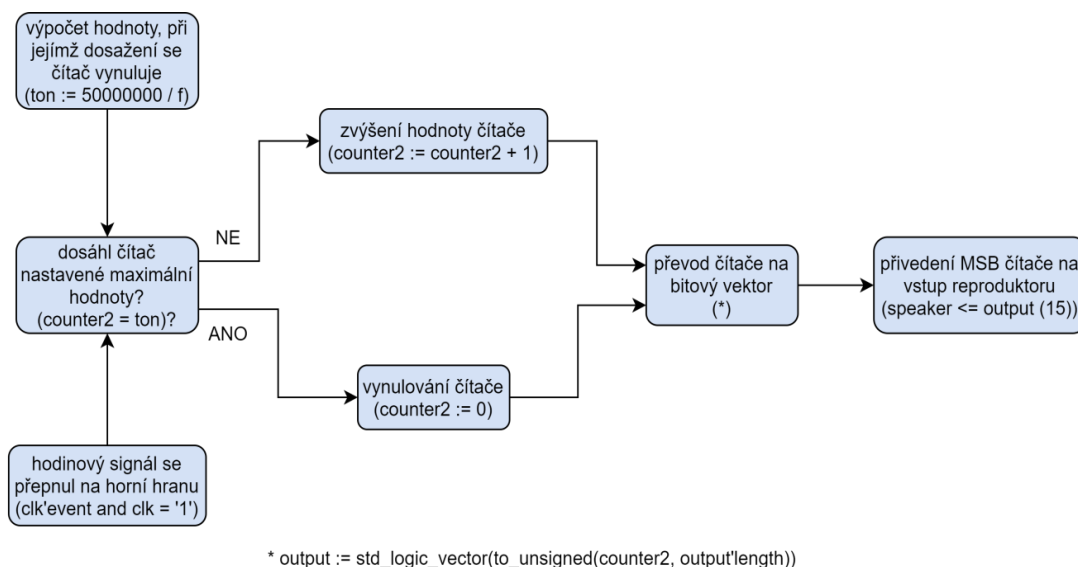
kde $f_{clock}[Hz]$ je frekvence hodinového signálu a $B[-]$ je číslo daného bitu čítače (0 až 15).

Čítač je schopný počítat od 0 maximálně do 65535. Dosáhne-li čítač této hodnoty, automaticky se vynuluje a začne počítat opět od nuly. Pro dosažení odlišných frekvencí, než je zmiňovaná frekvence MSB 763 Hz, je potřeba nastavit čítač tak, aby se vynuloval dříve, než dosáhne své maximální hodnoty 65535. Úpravou vztahu (1) získáme vzorec pro výpočet hodnoty, při jejímž dosažení se bude čítač nulovat:

$$ton = \frac{f_{clock}}{f_{tónu}}, \quad (2)$$

kde $f_{tónu} [Hz]$ ve vzorci (2) odpovídá frekvenci generovaného tónu.

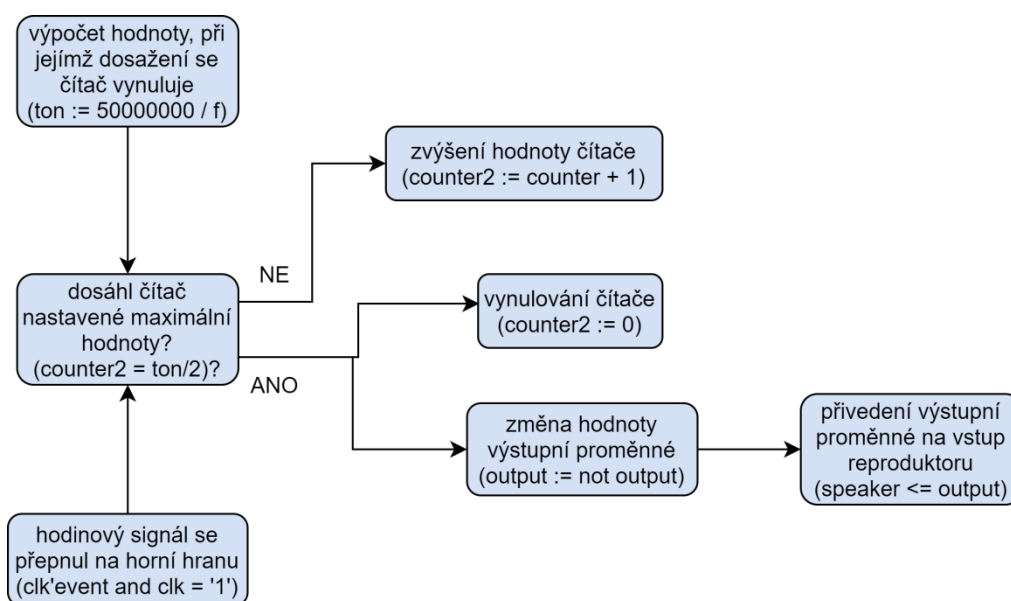
Diagram znázorňující generování tónů s využitím šestnáctibitového vektoru je na obrázku 3.5. Čítač, v diagramu označen jako counter2, je typu integer a následně převáděn na bitový vektor, jehož MSB je připojen na vstup reproduktoru.



Obr. 3.5: Průběh generování tónu požadované frekvence

Použitím tohoto postupu však způsobíme menší komplikaci, která je demonstrována v následujícím příkladu. Chceme-li dosáhnout frekvence 876 Hz, bude čítač nulován vždy, když dosáhne hodnoty 57077. Frekvence signálu vstupujícího do reproduktoru nyní skutečně bude 876 Hz, avšak střída nyní nebude 50 %, jako by tomu bylo, kdyby se čítač nuloval při dosažení své maximální hodnoty. MSB čítače (15. bit) má hodnotu logická 0 pro prvních 32767 přepnutí hodinového signálu na horní hranu, stejně jako tomu bylo bez nastavení frekvence. Logická 1 je však u MSB jen pro hodnoty čítače od 32768 až 57077, což je jen 24309 přepnutí hodinového signálu. To odpovídá střídě přibližně 43 %. Z tohoto důvodu bylo výše uvedené řešení ve finální verzi upraveno.

Čítač je částečně přenastaven a střídy 50 % je dosaženo dělením výstupu dvěma. Použijeme-li stejný příklad jako v minulém odstavci, bude to znamenat, že čítač nebude počítat do 57077, ale jen do poloviční hodnoty, tedy do 28538. Frekvence je tedy v tomto případě dělena nejprve číslem 28538 a následně dvěma. Průběh generování tónu s využitím této úpravy je na následujícím diagramu.



Obr. 3.6: Průběh generování tónu požadované frekvence se střídou 50%

Jak je vidět z diagramu, čítač se nuluje a inkrementuje za stejných podmínek jako před úpravou s tím rozdílem, že maximální hodnota je nyní $\text{ton}/2$, přičemž ton získáme opět ze vzorce (2). Při dosažení maximální hodnoty se zároveň změní hodnota výstupu, což je hlavní změna oproti převodu na šestnáctibitový vektor. Tóny generované se střídou 50 % jsou pouhým poslechem nerozeznatelné od tónů s neupravenou střídou.

3.2.2 Míšení signálů pro DTMF

Jak už bylo několikrát uvedeno, hlavním cílem projektu je demonstrace principu tónové volby (DTMF) pomocí deseti tónů (kláves). DTMF funguje na principu současného vysílání dvou sinusových signálů o různých frekvencích (viz kapitola 2.3). V této práci byly pro demonstraci DTMF využity frekvence odpovídající klávesám 0 až 9. Tyto kmitočty jsou uvedeny v tabulce 3.1.

f [Hz]	1209	1336	1477
697	1	2	3
770	4	5	6
852	7	8	9
941	*	0	#

Tab. 3.1: Frekvence používané u DTMF

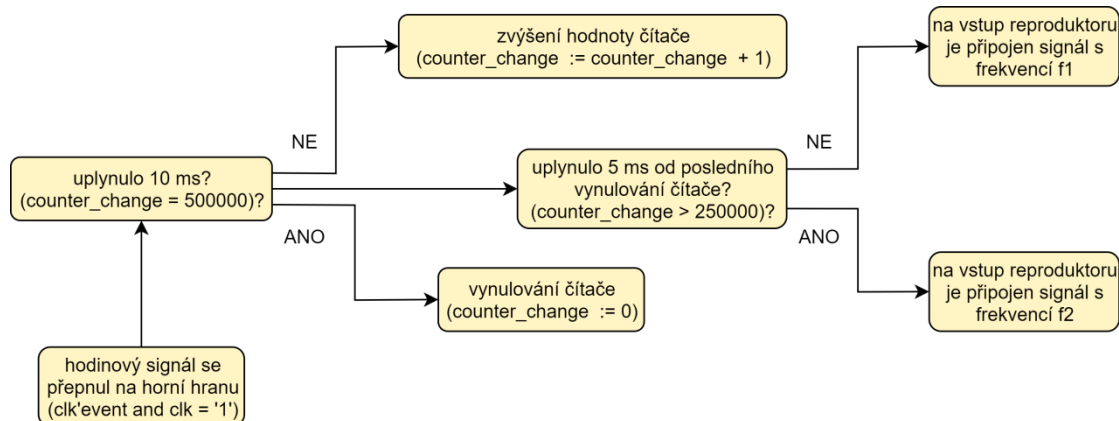
Do vstupu reproduktoru není možné posílat dva signály najednou. Je tedy potřeba přijít na způsob, jak pomocí jednoho výstupního signálu simulovat signály dva. V průběhu vývoje byly testovány a porovnávány následující tři varianty:

- **generování jednoho signálu**, jehož frekvence je součtem frekvencí sloupce a řádku (viz tabulka 3.1),
- **logický součet dvou signálů**,
- **střídání dvou signálů na vstupu reproduktoru**.

Výsledkem generování jednoho signálu, jehož frekvence je součtem frekvencí signálů, které by měly být v superpozici, jsou tóny o poměrně vysoké frekvenci, což by nebyl zásadní problém. Při použití frekvencí uvedených v tabulce 3.1 však dochází k potížím s rozlišením jednotlivých tónů. Je tedy nutné nadefinovat jiné frekvence řádkům i sloupcům tak, aby jejich výsledné součty byly dostatečně daleko od sebe. Pro demonstraci principu DTMF by tato metoda mohla být dostačující, ale určitě ne ideální.

Výstupem logického součtu neboli logické funkce OR je 1 právě tehdy, když má alespoň jeden ze sčítaných bitů hodnotu 1. Na vstupu do reproduktoru tedy bude logická hodnota 0, jen když oba sčítané signály budou mít v danou chvíli nulovou hodnotu. Pro použití této metody je potřeba generovat dva signály namísto jednoho, čehož je možné dosáhnout použitím dvou čítačů popsaných v předchozí kapitole. Na vstup do reproduktoru je pak přiveden logický součet výstupů obou signálů (na obrázku 3.6 v kapitole 3.2.1 označeno jako output). Při testování této metody byly použity hodnoty z tabulky 1 a poslechem byly porovnávány generované tóny s tóny DTMF. U pěti kláves (konkrétně „0“, „1“, „2“, „6“ a „7“) bylo dosaženo tónů, které znějí stejně nebo alespoň velmi podobně jako tóny při stisknutí tlačítek u DTMF. U ostatních kláves byl generovaný tón více či méně mimo. Metoda logického součtu, ač opět ne ideální, se zdá být vhodná pro demonstraci DTMF nebo alespoň vhodnější, než metoda sčítání frekvencí popsaná v předchozím odstavci.

Pro střídání dvou signálů o různých frekvencích na vstupu reproduktoru je stejně jako u metody logického součtu potřeba použít dva čítače. Zásadní rozdíl je v přidání dalšího pomocného čítače, který zajišťuje přepínání signálů na vstupu do reproduktoru. Tento čítač (v kódu a v diagramu pojmenován counter_change) počítá stejně jako ostatní čítače přepnutí hodinového signálu přípravku na horní hranu. Testováno bylo nejprve střídání po 50 ms. Tato perioda se ukázala být příliš dlouhá, neboť výstup nezněl jako jeden tón, ale jako dva periodicky se střídající tóny. Následně tedy bylo testováno střídání po výrazně kratší době, konkrétně po 2,5 ms a 5 ms. Právě při střídání po 5 ms (perioda 10 ms) se zdají být všechny generované tóny totožné nebo alespoň velmi blízké tónům DTMF. Princip fungování této metody je naznačen na obrázku 3.7. Periodě střídání 10 ms odpovídá nastavení maximální hodnoty čítače counter_change na 500000. To, který ze dvou generovaných signálů je připojen na vstup reproduktoru, závisí na tom, je-li hodnota counter_change větší než 250000 nebo není.



Obr. 3.7: Fungování metody střídání signálů po 5 ms

Metoda s použitím periody střídání dvou signálů se zdá být z uvedených tří možností tou nejvhodnější pro demonstraci principu tónové volby.

K nastavování frekvencí generovaných signálů jsou využívány funkce `frequency_low` a `frequency_high`. Účelem těchto funkcí je výpočet hodnoty čísla ton ze vztahu (2). Toto číslo je zároveň návratovou hodnotou funkcí. Vstupními parametry obou funkcí je číslo odpovídající stisknuté klávese. Při stisknutí klávesy „1“ je tedy vstupní hodnota také 1, pro klávesu „2“ je vstupní hodnota 2 a tak dále až ke klávese „0“, pro kterou je vstupní hodnota 11. Funkce `frequency_low` pracuje s celočíselným dělením vstupního parametru a funkce `frequency_high` se zbytkem po celočíselném dělení vstupního parametru. V tabulkách 3.2 a 3.3 jsou uvedeny frekvence a návratové hodnoty ton získané ze zmíněných pomocných funkcí v závislosti na vstupní hodnotě X.

$\frac{X-1}{3}$	$f_{\text{tónu}}$	$\text{ton} = \frac{f_{\text{clock}}}{f_{\text{tónu}}}$
0	697	71736
1	770	64935
2	852	58685
3	941	53134

Tab. 3.2: Frekvence a návratové hodnoty N pro vstupní hodnoty X funkce `frequency_low`

$X \bmod 3$	$f_{\text{tónu}}$	$\text{ton} = \frac{f_{\text{clock}}}{f_{\text{tónu}}}$
1	1209	41356
2	1336	37425
0	1477	33852

Tab. 3.3: Frekvence a návratové hodnoty N pro vstupní hodnoty X funkce `frequency_high`

3.2.3 Využití otočného tlačítka pro nastavení frekvence střídání signálů

Jak už bylo zmíněno v předchozí kapitole, pro demonstraci DTMF je použita metoda střídání dvou signálů o různých frekvencích na vstupu reproduktoru. Nejvhodnější periodou, s jakou se dané signály střídají, se zdá být 10 ms, ale i s jinými periodami bylo v průběhu testování dosahováno zajímavých výsledků. Nabízí se tedy rozšíření kódu, které by umožňovalo přepínat mezi několika periodami. K tomuto účelu bylo využito otočné tlačítko na přípravku Spartan-3E Starter Kit.

Otočné tlačítko je nastaveno tak, aby s jeho pomocí bylo možné přepínat mezi celkem patnácti módy. Periody střídání signálů odpovídající jednotlivým módům jsou uvedeny v tabulce 3.4. Číslo aktuálně použitého módu (jeho binární hodnota) je zobrazováno pomocí čtyř levých LED na přípravku (LED4 až LED7). Defaultně je nastaven mód 8.

mód	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
T [ms]	3	3,5	4	4,5	5	6	7,5	10	12,5	15	20	25	37,5	50	75

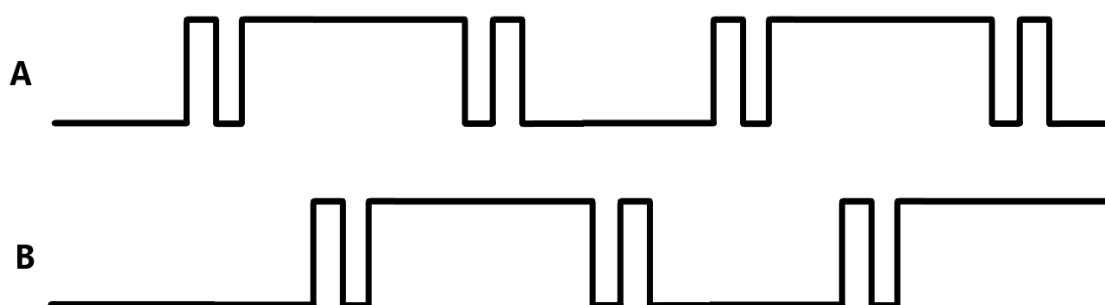
Tab. 3.4: Módy a jim odpovídající periody střídání signálů

Rotací otočného tlačítka jsou ovládány dva spínače (signály A a B). Ideálně by spínače fungovaly tak, že se sepnutý spínač připojí na zem, což má za následek přepnutí jemu odpovídajícího signálu na logickou 0. Stejně tak by se po rozepnutí spínače měl signál přepnout na hodnotu logická 1 (viz kapitola 2.4.3). Příklad ideálního průběhu signálů A a B v případě otáčení po směru hodinových ručiček je vidět na obrázku 3.8.



Obr. 3.8: Ideální průběh signálů A, B při otáčení doprava [vlastní zpracování podle 1]

Reálně však vlivem mechanických kontaktů dochází k zašumění signálů A, B. Úroveň signálů se po přepnutí ještě několikrát velmi rychle změní, než se ustálí na správné hodnotě. Důsledkem toho může být, že přípravek chybně vyhodnotí náhodné kmity jako další otočení tlačítkem, a to i v opačném směru. Tento problém je vyobrazen na obrázku 3.9. [1, strana 18-19] [19]



Obr. 3.9: Zašumění vzniklé mechanickými kontakty při otáčení doprava
[vlastní zpracování podle 1]

V procesech pro vyhodnocování otočného tlačítka bylo vycházeno z [19]. V prvním procesu se vyhodnocuje, jestli došlo k otáčení tlačítkem a v případě, že ano, tak v jakém směru. Do pomocných proměnných se ukládají hodnoty signálů A a B. Nastane-li například situace, kdy má signál A hodnotu 1 a signál B hodnotu 0, v procesu bude vyhodnoceno, že směr otáčení je po směru hodinových ručiček. Nyní však záleží na tom, jestli k otáčení vůbec dochází. To je vyhodnoceno podle toho, jestli budou mít oba signály v následujícím stavu logickou hodnotu 0 (k otáčení nedochází) nebo 1 (k otáčení dochází). Tato část kódu je na obrázku 21.

```

case vysledek_otaceni is
  when "00" => detekce_otaceni <= '0';
                smer_otaceni <= smer_otaceni;
  when "01" => detekce_otaceni <= detekce_otaceni;
                smer_otaceni <= '0';
  when "10" => detekce_otaceni <= detekce_otaceni;
                smer_otaceni <= '1';
  when "11" => detekce_otaceni <= '1';
                smer_otaceni <= smer_otaceni;
  when others => detekce_otaceni <= detekce_otaceni;
                smer_otaceni <= smer_otaceni;
end case;

```

Obr. 3.10: Úryvek kódu s vyhodnocením směru otáčení

Následuje proces, ve kterém dochází k eliminaci dříve zmíněných nežádoucích zákmitů. V podstatě se tedy v tomto procesu znovu kontroluje, zda skutečně došlo k otočení tlačítkem. Pro vyhodnocení se využije porovnání předchozí a současné hodnoty signálu, ve kterém je uložena informace o tom, že by mělo/nemělo docházet k otáčení. Platí-li, že předchozí hodnota signálu je 0 a současná 1, bude přípravkem vyhodnoceno, že k otočení skutečně došlo.

V posledním procesu určeném k vyhodnocení otočného tlačítka se pracuje s výstupem. Došlo-li k otočení, mění se index módu v závislosti na směru. V případě otočení po směru hodinových ručiček je index navýšen o 1, v případě otočení proti směru je tomu naopak. Při dosažení maximálního/minimálního indexu se index nebude měnit do té doby, než uživatel otočí tlačítkem na druhou stranu. Z indexu 15 se tedy nedá skočit na index 1 a naopak.

Nastavený index je následně zobrazen pomocí LED4 až LED7, jak už bylo zmíněno. Pro nastavení odpovídající frekvence přepínání signálů pak slouží integerové pole, ve kterém je uloženo 15 hodnot, které jsou přiřazovány jako maximální dosažitelná hodnota čítače určeného k přepínání signálů (viz kapitola 3.2.2).

3.3 Projekt jako celek

Program není po jeho nahrání na přípravek potřeba spouštět žádným posuvným spínačem. Tóny jsou generovány hned po stisknutí odpovídající klávesy a výstup se vrací do klidového stavu hned po jejím puštění.

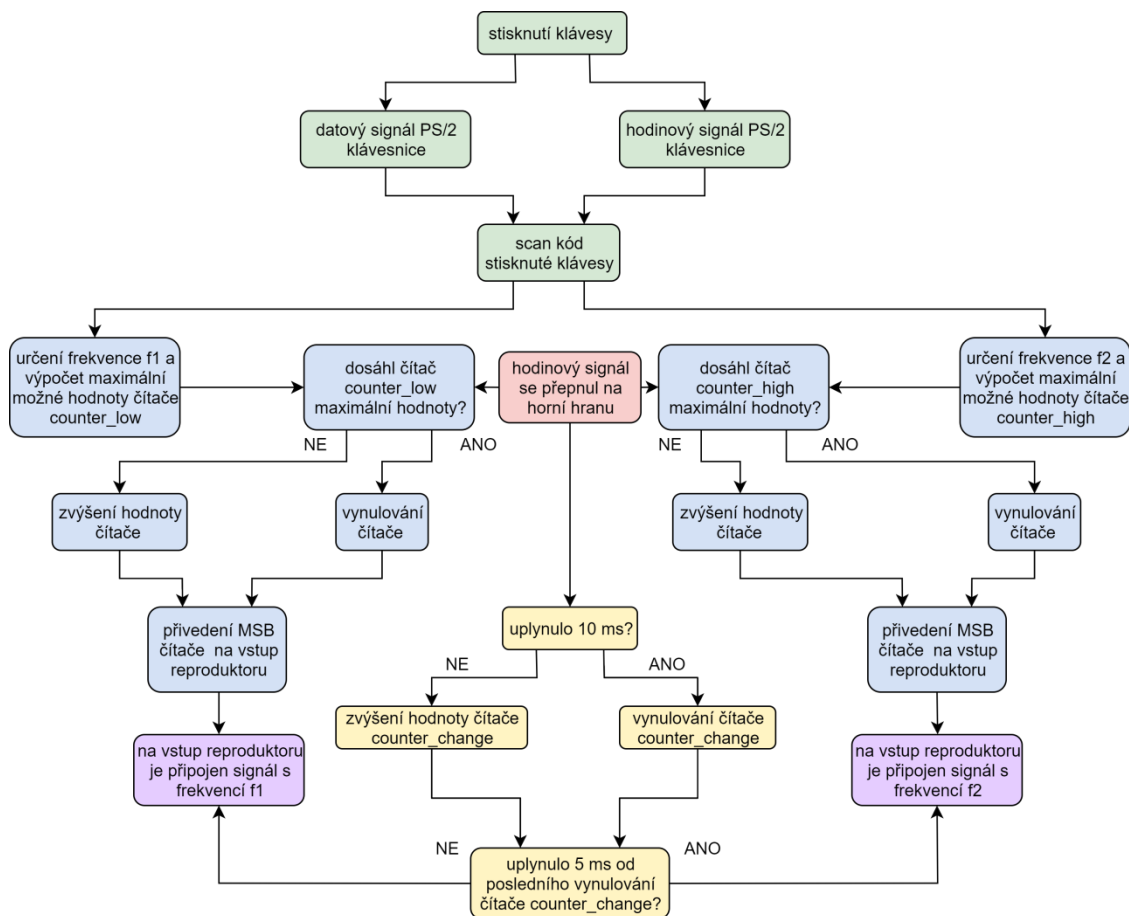
Pro generování tónů jsou využity klávesy „1“ až „0“ v horní řadě klávesnice. Je-li stisknuta jiná klávesa, výstup přípravku je přepnut na nulu (LED nesvítí a není generován tón).

Jako kontrola správného vyhodnocení stisknuté klávesy fungují LED0 až LED3 na přípravku. LED se rozsvěcují tak, aby zobrazovaly hodnotu stisknuté klávesy ve dvojkové soustavě s tím, že klávesa „0“ se zobrazuje pomocí LED jako číslo 10 ve dvojkové soustavě. Je-li tedy stisknuta například klávesa „3“, rozsvítí se LED0 a LED1. Ostatní čtyři LED, tedy LED4 až LED7, slouží jako indikace nastaveného módu otočného tlačítka.

Jako vstupní piny přípravku jsou nastaveny datový a hodinový pin PS/2 konektoru (v kódu ps2_data a ps2_clk), ke kterému je připojena PS/2 klávesnice, signály A a B otočného tlačítka (v kódu otaceni_smer1 a otaceni_smer2) a hodinový signál přípravku (v kódu clk). Výstupními piny jsou pak již zmiňované LED0 až LED7, které jsou v deklaraci entity rozděleny na dva čtyřbitové vektory (v kódu stav a klavesa_dvojkove) a vstup do externího reproduktoru (v kódu speaker).

Pro demonstraci DTMF je využívána metoda střídání dvou signálů na vstup reproduktoru s tím, že defaultně je nastavena perioda střídání 10 ms. Tuto periodu lze nastavit na celkem 15 různých délek, přičemž nejkratší možná perioda je 3 ms a nejdelší možná je 75 ms.

Průběh celého programu s délkou periody střídání signálů na vstupu do reproduktoru nastavenou na 10 ms je vyobrazen na následujícím diagramu. Barvy jednotlivých částí programu odpovídají barvám použitým u digramů v předchozích kapitolách.



Obr. 3.11: Průběh celého programu

Na následujícím obrázku je vyfoceno zapojení celého návrhu, tedy propojení přípravku Spartan-3E Starter Kit s použitými periferiemi. Na fotografii je vidět příklad rozsvícení LED v případě, kdy je stisknuta klávesa „5“ a je využíván mód číslo 8 otočného tlačítka.



Obr. 3.12: *Propojení přípravku s periferiemi*

3.3.1 Možná rozšíření

Dalším zajímavým rozšířením souvisejícím s generováním tónů by mohlo být přidání možnosti přepínání mezi metodami vhodnými pro demonstraci DTMF. Účelem tohoto rozšíření by bylo hlavně přidání možnosti porovnat jednotlivé metody bez nutnosti nahrávání jiných programů na přípravek. K přepínání by se daly využít posuvné spínače obsažené na přípravku.

Dále by mohlo být užitečné využít LCD displej. Například pro zobrazování frekvencí obou signálů, které by byly v danou chvíli používány pro generování tónu, nebo pro zobrazení periody, s jakou se generované signály na výstupu střídají.

4 Závěr

Cílem práce bylo demonstrovat princip tónové volby (DTMF) v závislosti na vstupu z PS/2 klávesnice pomocí přípravku Spartan-3E Starter Kit a k němu připojeného reproduktoru.

Přípravek je schopný generovat požadovaných deset tónů v závislosti na stisknutí jedné z určených deseti kláves na PS/2 klávesnici. Frekvence těchto tónů je totožná nebo minimálně velmi blízká tónům používaných u tónové volby, z čehož lze usoudit, že použité řešení je skutečně vhodné k demonstraci principu DTMF. K řešení bylo přidáno i rozšíření o otočné tlačítko použitého přípravku, pomocí kterého lze generované tóny upravovat.

Pro splnění tohoto cíle bylo nejprve potřeba naučit se používat jazyk VHDL na dostatečně pokročilé úrovni. Dále bylo potřeba seznámit se s funkcemi přípravku Spartan-3E Starter Kit, fungováním komunikace mezi PS/2 klávesnicí a přípravkem a s principem DTMF.

V praktické části je popsán způsob řešení jednotlivých problémů. Celou problematiku návrhu lze rozdělit na komunikaci s klávesnicí a generování tónů. Kromě popisu kódu je v praktické části, pro snazší pochopení návrhu, formou diagramů vyobrazen průběh některých částí programu. V závěru praktické části je shrnut program jako celek a jsou uvedena možná rozšíření.

Práce ukazuje, jakým způsobem lze generovat tóny pomocí přípravku Spartan-3E Starter Kit a jednoduchého reproduktoru. Úpravou kódu by bylo možné využít přípravek kromě demonstrace principu tónové volby například i jako jednoduchý syntezátor.

Zdroje

- [1] *Spartan-3E Starter Kit Board User Guide* [online]. 2006 [cit. 2020-03-2]. Dostupné z: <https://store.digilentinc.com/spartan-3e-starter-board-limited-time/>
- [2] CHAPWESKE, Adam. *The PS/2 Mouse/Keyboard Protocol* [online]. 2007 [cit. 2020-03-4]. Dostupné z: <https://www.avrfreaks.net/sites/default/files/PS2%20Keyboard.pdf>
- [3] PINKER, Jiří a Martin POUPA. *Číslicové systémy a jazyk VHDL*. Praha: BEN - technická literatura, 2006. ISBN 80-730-0198-5.
- [4] MAXFIELD, Clive. *The design warrior's guide to FPGAs: devices, tools, and flows*. Boston: Newnes/Elsevier, c2004. ISBN 0-7506-7604-3.
- [5] MOORE, Andrew a Ron WILSON. *FPGAs For Dummies*. 2. spec. Intel vyd. Hoboken: John Wiley & Sons, 2017. ISBN 978-1-119-39049-7.
- [6] *FPGA Fundamentals* [online]. 2019 [cit. 2020-3-30]. Dostupné z: <https://www.ni.com/cs-cz/innovations/white-papers/08/fpga-fundamentals.html>
- [7] *What is an FPGA?* [online]. [cit. 2020-3-30]. Dostupné z: <https://breadboardgremlins.wordpress.com/what-is-an-fpga/>
- [8] *Nebojte FPGA* [online]. [cit. 2020-3-30]. Dostupné z: <https://vyvoj.hw.cz/teorie-a-praxe/dokumentace/nebojte-se-fpga.html>
- [9] PERRY, Douglas L. *VHDL: programming by example*. 4th ed. New York: McGraw-Hill, c2002. ISBN 0-07-140070-2.
- [10] PEDRONI, Volnei A. *Circuit design with VHDL*. Cambridge, Mass.: MIT Press, c2004. ISBN 0-262-16224-5.
- [11] MEALY, Brian a Fabrizio TAPPERO. *Free Range VHDL*. c2018.
- [12] ŠUSTA, Michal. *Návrh obvodů ve VHDL modelovacími styly "dataflow" a "structural"*. verze 1.1. Praha, 2019. Katedra řídicí techniky ČVUT FEL.
- [13] Xilinx ISE. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2010 [cit. 2020-04-3]. Dostupné z: https://en.wikipedia.org/wiki/Xilinx_ISE
- [14] Dual Tone Multi Frequency (DTMF). In: *Signal Identification Guide Wiki* [online]. 2015 [cit. 2020-04-8]. Dostupné z: [https://www.sigidwiki.com/wiki/Dual_Tone_Multi_Frequency_\(DTMF\)](https://www.sigidwiki.com/wiki/Dual_Tone_Multi_Frequency_(DTMF))

- [15] *DTMF (Dual-Tone Multi-Frequency)* [online]. [cit. 2020-04-8]. Dostupné z: <https://www.mediacollege.com/audio/tone/dtmf.html>
- [16] *Working Of DTMF (Dual Tone Multi Frequency) Technology and Its Applications* [online]. [cit. 2020-04-8]. Dostupné z: <https://www.efxkits.us/dtmf-dual-tone-multi-frequency-technology-working-applications/>
- [17] *Spartan-3E FPGA Family Data Sheet* [online]. 2018 [cit. 2020-03-15]. Dostupné z: https://www.xilinx.com/support/documentation/data_sheets/ds312.pdf
- [18] *36CS16FN Datasheet* [online]. [cit. 2020-04-6]. Dostupné z: <https://en.maritex.com.pl/acoustics/speakers/rounds/vec-36cs16fn-50bd.html>
- [19] CHAPMAN, Ken. XILINX LTD. *Rotary Encoder Interface for Spartan-3E Starter Kit* [online]. 2006 [cit. 2020-04-10]. Dostupné z: https://www.mikrocontroller.net/attachment/56824/s3esk_rotary_encoder_interface.pdf

Obsah přiloženého CD

Na CD, které je přiloženo k tištěné verzi, jsou vypáleny následující soubory související s touto prací:

- Pdf soubor, který obsahuje **vlastní text bakalářské práce**.
- Stažené pdf soubory s **materiály, ze kterých bylo v průběhu psaní práce čerpáno**.
- Vytvořené **obrázky a diagramy**, které jsou v práci obsažené.
- **Finální VHDL kód** jak ve formě textového souboru, tak ve formě souboru vhd společně s ostatními soubory vygenerovanými programem Xilinx ISE.