

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČITAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

JEDNODUCHÝ TÓNOVÝ GENERÁTOR PRO FITKIT

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

ONDŘEJ MALAČKA

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

JEDNODUCHÝ TÓNOVÝ GENERÁTOR PRO FITKIT

SIMPLE TONE GENERATOR

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

ONDŘEJ MALAČKA

VEDOUCÍ PRÁCE
SUPERVISOR

ING. TOMÁŠ HERRMAN

BRNO 2007

Zadání bakalářské práce

Řešitel: **Malačka Ondřej**

Obor: Informační technologie

Téma: **Jednoduchý tónový generátor pro FITkit**

Kategorie: Vestavěné systémy

Pokyny:

1. Seznamte se s výukovým kitem obsahujícím FPGA a mikrokontroler.
2. Seznamte se s komunikačním protokolem klávesnice.
3. Navrhněte a implementujte tónový generátor ovládaný PS/2 klávesnicí připojenou ke kitu a zobrazující hrané tóny na LCD display.
4. Implementujte do generátoru možnost zahrát více tónů současně. (Současné zmáčknutí více kláves.)

Literatura:

- <http://www.fit.vutbr.cz/kit>

Při obhajobě semestrální části projektu je požadováno:

- Splnění bodů 1, 2 zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním paměťovém médiu (disketa, CD-ROM), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Herrman Tomáš, Ing.**, UPSY FIT VUT

Datum zadání: 1. listopadu 2006

Datum odevzdání: 15. května 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačových systémů a sítí
602 005, Božetěchova 2

doc. Ing. Zdeněk Kotásek, CSc.
vedoucí ústavu

**LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO**

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **Ondřej Malačka**
Id studenta: 84254
Bytem: U Tesly 1173/3, 736 01 Havířov
Narozen: 18. 02. 1985, Havířov
(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....

(dále jen "nabyvatel")

**Článek 1
Specifikace školního díla**

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
bakalářská práce

Název VŠKP: Jednoduchý tónový generátor pro FITkit
Vedoucí/školitel VŠKP: Herrman Tomáš, Ing.
Ústav: Ústav počítačových systémů
Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v:

tištěné formě	počet exemplářů: 1
elektronické formě	počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2 Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3 Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....
Nabyvatel


.....
Autor

Abstrakt

Tato bakalářská práce se zabývá generováním tónů na platformě FITkit. Cílem bakalářské práce bylo navrhnout a realizovat aplikaci pro generování tónů. Uživatelským vstupem pro zadávání tónů je PS/2 klávesnice. Pro realizování aplikace je potřeba popsat konfiguraci pro FPGA v jazyce VHDL a naprogramovat mikrokontrolér v jazyce C. Mikrokontrolér má úlohu generování tónů a ovládání LCD displeje a klávesnice. Řadiče těchto periférií jsou realizovány v FPGA. Aplikace umožňuje kromě zahrání jednoduchých tónů zahrát také několik tónů najednou. Reprodukované tóny jsou také zobrazeny na LCD displeji.

Klíčová slova

Jednoduchý tónový generátor, FITkit, FPGA, mikrokontrolér, VHDL, jazyk C, DA převodník, PS/2 klávesnice, SPI

Abstract

This bachelor's thesis deals with simple tone generation on FITkit. The main goal of this thesis was to design and to implement application for tone generation. Simple tone entry is via PS/2 keyboard. System implementation consists of two parts, description of configuration for FPGA and implementation of microcontroller in programming language C. Microcontroller is used for tone generation and taking control of LCD display and keyboard. Peripherals controllers are implemented in FPGA. It is possible not only generate simple tones, but also generate several simple tones at the same time. Generated tones are also displayed on LCD.

Keywords

Simple tone generator, FITkit, FPGA, microcontroller, VHDL, language C, DA converter, PS/2 keyboard, SPI

Citace

Ondřej Malačka: Jednoduchý tónový generátor pro FITkit, bakalářská práce, Brno, FIT VUT v Brně, 2007

Jednoduchý tónový generátor pro FITkit

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Tomáše Herrmana.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Ondřej Malačka
15.5.2007

Poděkování

Chtěl bych poděkovat vedoucímu bakalářské práce Tomáši Herrmanovi za poskytnutou pomoc při realizaci bakalářské práce.

© Ondřej Malačka, 2007.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod.....	3
2 FITkit.....	4
2.1 Úvod k FITkitu.....	4
2.2 Blokovoé schéma FITkitu.....	4
3 PS/2 klávesnice.....	5
3.1 PS/2 rozhraní.....	5
3.2 Komunikační protokol klávesnice.....	5
3.2.1 Přenos dat z klávesnice.....	6
3.2.2 Přenos dat do klávesnice.....	6
3.2.3 Kódování kláves klávesnice.....	6
4 Základy a principy generování zvuku.....	8
4.1 Zvuk.....	8
4.1.1 Důležité parametry zvukové vlny.....	9
4.2 Interference vlnění.....	9
4.3 Oktáva a oktávový systém.....	10
4.4 Ladění a hudební stupnice.....	11
4.4.1 Rovnoměrné temperované ladění.....	11
4.5 Nástroje pro generování zvuku.....	11
4.5.1 Sampler.....	12
4.5.2 Syntezátor.....	12
4.5.3 Virtuální nástroje.....	13
5 Fáze návrhu a implementace aplikace.....	14
5.1 Blokovoé schéma generátoru tónů.....	15
5.2 SPI komunikace.....	15
5.2.1 Rozhraní SPI.....	15
5.2.2 SPI na FITkitu.....	16
5.2.3 Protokol komunikace mikrokontroléru s FPGA.....	16
6 FPGA.....	17
6.1 Řadič SPI.....	17
6.1.1 Popis funkce.....	17
6.2 SPI dekodér.....	17
6.2.1 Popis funkce.....	18
6.3 Řadič PS/2 klávesnice.....	18
6.3.1 Popis funkce.....	18
6.4 Řadič přerušování.....	19

6.4.1 Popis funkce.....	19
6.5 Řadič LCD displeje.....	20
6.5.1 Popis funkce.....	20
6.6 Schéma konfigurace FPGA.....	21
7 Mikrokontrolér.....	22
7.1 Konfigurace registrů mikrokontroléru.....	22
7.1.1 DA převodník.....	22
7.1.2 Časovač.....	22
7.2 Komunikace s perifériemi.....	23
7.2.1 PS/2 klávesnice.....	23
7.2.2 LCD displej.....	23
7.3 Program mikrokontroléru.....	23
7.3.1 Zpracování vstupu z klávesnice.....	24
7.3.2 Generování tónů.....	25
8 Závěr.....	29
Literatura.....	30
Seznam příloh.....	31

1 Úvod

Tématem této bakalářské práce je vytvořit tónový generátor na platformě FITkit. FITkit je komplexní hardwarový nástroj pro tvorbu aplikací. Umožňuje návrh softwaru a hardwaru do značné míry podobným způsobem, kdy hardware je popsán jazykem VHDL a software pro mikrokontrolér je naprogramován v jazyce C. Jádrem kitu a také celé aplikace je mikrokontrolér MSP430, který řídí veškeré periférie, jejichž řadiče jsou realizovány v druhé důležité komponentě kitu a to v FPGA Spartan 3. Komunikace mezi mikrokontrolérem a FPGA je realizována pomocí synchronní sériové sběrnice SPI. Hlavním úkolem aplikace generátoru tónů je zpracovat uživatelský vstup z PS/2 klávesnice a vhodným způsobem tento klávesový vstup převést na tóny. Tóny jsou generovány mikrokontrolérem za pomoci modulu DA převodníku, který je v mikrokontroléru zabudován. Výsledné generované úrovně napětí, s frekvencí která odpovídá tónům, jsou přivedeny přes audio zesilovač na reproduktor. S pomocí reproduktoru kterého je zajištěn jejich zvukový výstup z aplikace. Kromě generování jednoduchých tónů je také implementována na principech vzorkování a interference možnost zahrát více tónů ve stejný okamžik.

Kapitola 2 podává základní informace o FITkitu a základních komponentách, které jsou na FITkitu zabudovány.

Kapitola 3 popisuje teoretické základy potřebné pro komunikaci s klávesnicí. Je zde popsán princip přenosu dat oběma směry a také formát dat, která jsou klávesnicí posílána.

Kapitola 4 popisuje teoretické základy, důležité pro generování zvuku. Je zde popsán princip vzniku zvuku, jakým způsobem lze generovat více tónů najednou, teoretické základy pro realizaci výpočtů tónů a jsou zde také popsány nástroje, kterými je zvuk generován v praxi.

Kapitola 5 se zabývá jednotlivými fázemi návrhu a implementace, je zde také popsán jakým způsobem byl vyřešen vstup aplikace, znázorněno blokové schéma celé aplikace a také popsána SPI komunikace na FITkitu.

Kapitola 6 popisuje implementaci všech řadičů realizovaných v FPGA, jejich funkci v aplikaci a také jak tyto řadiče pracují. Je zde také znázorněno schéma celé konfigurace FPGA a vzájemné propojení komponent.

Kapitola 7 se zabývá nastavením registrů mikrokontroléru tak, aby co nejlépe vyhovoval aplikaci. Tato kapitola dále popisuje jakým způsobem komunikuje mikrokontrolér s řadiči realizovanými v FPGA a také popisuje jakým způsobem byl generátor tónů naprogramován.

2 FITkit

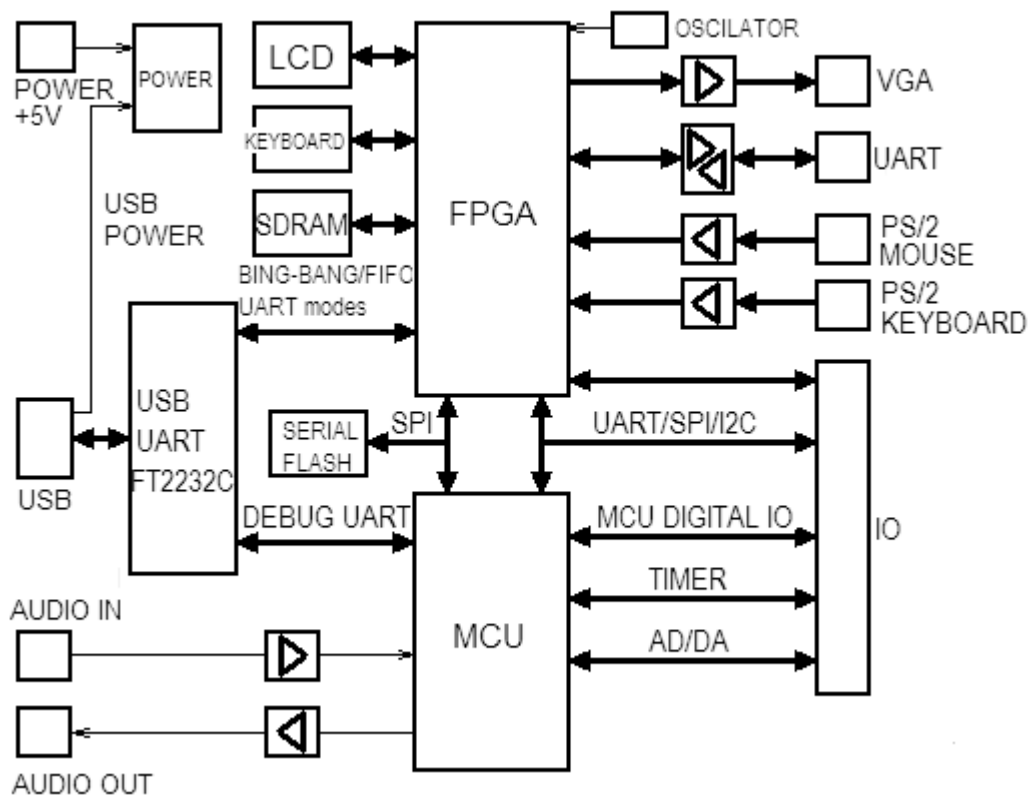
2.1 Úvod k FITkitu

Platforma FITkit vznikla za cílem umožnit studentům projekty navrhovat nejen softwarově, ale také hardwarově. FITkit je navržen tak, aby jednoduchým způsobem umožňoval popsat hardwarové komponenty s použitím reprogramovatelného hradlového pole FPGA v jazyce VHDL. Tyto komponenty jsou s pomocí rozhraní SPI ovládány výkonným mikrokontrolérem, který je naprogramovaný v jazyce C.

FITkit je koncipován tak, aby umožňoval použít podpůrné knihovny, vytvořené za účelem bezproblémového využití všech komponent, ať již přímo zabudovaných na kitu, nebo přes velké množství rozhraní, přes které je umožněno tyto komponenty připojit .

2.2 Blokové schéma FITkitu

Na blokovém schématu FITkitu (viz obrázek 2.1) jsou znázorněny všechny jeho důležité komponenty.



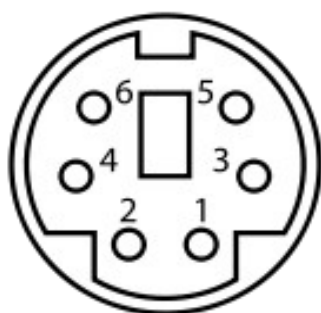
Obr. 2.1: Blokové schéma FITkitu

3 PS/2 klávesnice

3.1 PS/2 rozhraní

PS/2 rozhraní je nízkorychlostní sériové rozhraní typu master-slave. Používá se pro připojení klávesnice a myši k PC. Rozhraní je tvořeno napájením, hodinovým signálem a daty. Hodinový signál určuje dobu vzorkování dat. Hodinový signál je generován masterem a pohybuje se v rozsahu 10-30 kHz. Masterem je koncové zařízení (klávesnice, myš). Výhodou tohoto přístupu je, že si periferní zařízení samo udává tempo přenosu. Hodinový i datový signál jsou realizovány jako výstupy s otevřeným kolektorem a na logické „1“ je drží PULL-UP rezistory, z důvodu možnosti ovládnutí oběma komunikujícími stranami.

PS/2 port je šestipinový, ovšem v dnešní době se pro propojení s druhým zařízením využívají pouze 4 piny, jen u některých notebooků se používají nevyužitá 2 piny pro připojení dalšího zařízení. Protože nejsou konektory mechanicky kódovány, používá se barevné rozlišení pro myš a klávesnici. Zelený konektor a zástrčka slouží pro myš, modrofialová barva pro klávesnici. PS/2 port je zobrazen na obrázku 3.1.



Pin	Popis
1	Data
2	Nezapojeno
3	Zem
4	Napájení +5V ss.
5	Hodinový signál
6	Nezapojeno

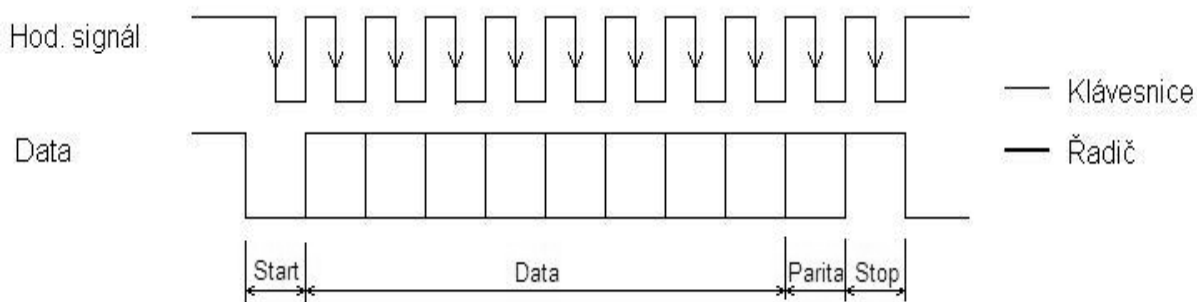
Obr. 3.1: PS/2 port

3.2 Komunikační protokol klávesnice

Komunikační protokol klávesnice je synchronní, sériový a obousměrný. Komunikuje se vždy po jedenácti bitech. Platnost dat je dána spádovou hranou hodinového signálu. Jako první je vyslán nejméně významný bit a jako poslední nejvíce významný bit. V klidovém stavu je hodinový i datový signál v logické „1“.

3.2.1 Přenos dat z klávesnice

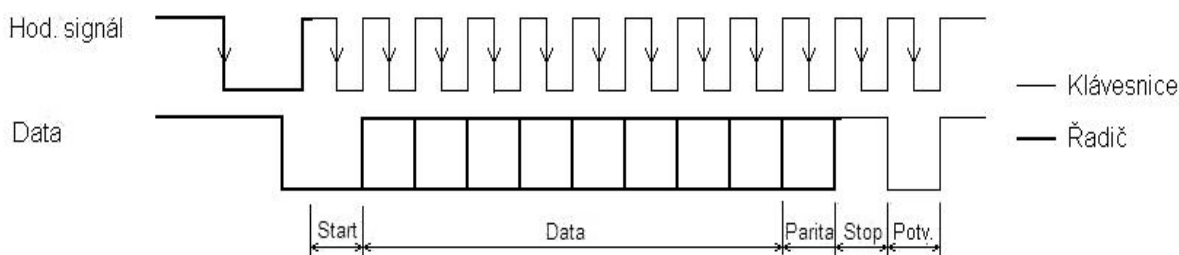
Přenos dat inicializuje klávesnice stáhnutím datového signálu do logické „0“ a započítím generování hodinového signálu. Řadič při tomto směru přenosu pouze naslouchá a na sestupnou hranu hodinového signálu vzorkuje data. Každý datový přenos je zahájen jedním start bitem (logická „0“), za nímž následuje 8 datových bitů, jeden paritní bit (lichá parita) a jeden stop bit. Časový diagram komunikace je znázorněn na obrázku 3.2.



Obr 3.2: Časový diagram komunikace při přenosu dat z klávesnice

3.2.2 Přenos dat do klávesnice

Přenos dat je inicializován řadičem, stažením hodinového signálu do logické „0“, čímž se zamezí komunikaci opačným směrem. Po době odpovídající minimálně jednomu taktu hodinového signálu, stáhne řadič také datový signál do logické „0“ a uvolní hodinový signál, který začne generovat klávesnice. Přenos pak pokračuje jedním start bitem, osmi datovými bity, paritním bitem a stop bitem, ale na rozdíl od přenosu opačným směrem, musí být úspěšný přenos ještě potvrzen koncovým zařízením. Časový diagram přenosu dat do klávesnice viz obrázek 3.3.



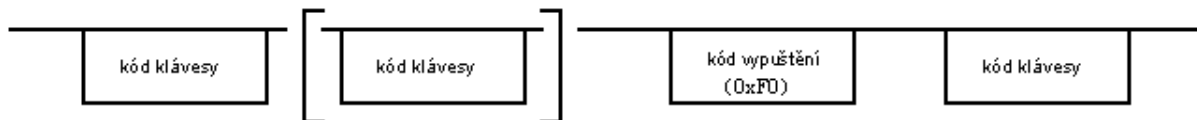
Obr 3.3: Časový diagram komunikace při přenosu dat do klávesnice

3.2.3 Kódování kláves klávesnice

Jednotlivé klávesy klávesnice lze rozdělit do tří skupin z hlediska kódování kláves. Jsou to skupina základních kláves, rozšířená skupina a speciální skupina.

Do základní skupiny se řadí 83 kláves. Při stisku klávesy ze základní skupiny se vyše kód stisknuté klávesy a po uvolnění se nejprve vyše kód vypuštění, za nímž následuje kód uvolněné

klávesy. V případě, že je klávesa stisknuta delší dobu než je nastavena doba tzv. „typematic delay“ (doba opakování), je po každém tomto intervalu vyslán znovu kód klávesy, dokud není klávesa uvolněna. Pořadí odesílaných kódů viz obrázek 3.4.



Obr 3.4: Pořadí odesílaných kódů základních kláves

V rozšířené skupině je zahrnuto 14 kláves. Tyto klávesy mají shodný kód s klávesami numerické klávesnice, ale pro rozlišení je před kód dané klávesy patřící do rozšířené skupiny vždy předřazen kód 0xF0. Pořadí odesílaných kódů viz obrázek 3.5.



Obr. 3.5: Pořadí odesílaných kódů kláves rozšířené skupiny

Speciální skupina zahrnuje všechny ostatní klávesy což jsou 'Pause', 'Print Screen', 'Caps Lock', 'Scroll Lock' a 'Num Lock'. Tyto klávesy mají specifický princip kódování a záleží na ostatních stisklých klávesách.

V této kapitole jsou použity informace získané z literatury [7].

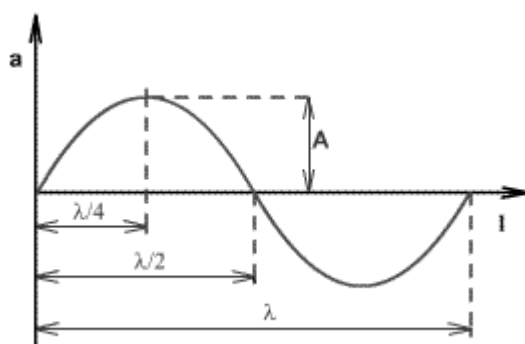
4 Základy a principy generování zvuku

Tato kapitola je zaměřena na teoretické základy pro generování zvuku. První podkapitola o zvuku je důležitá pro pochopení na jakém principu zvuk vzniká. Další podkapitola zabývající se interferencí vlnění vysvětluje princip, který je důležitý z hlediska generování několika tónů ve stejný okamžik. Podkapitoly 4.1.3 a 4.1.4 jsou teorií využitou pro výpočet tónů, které jsou výslednou aplikací generovány. Poslední podkapitola pojednává o způsobech kterými je generován zvuk v běžné praxi.

4.1 Zvuk

Zvuk obecně můžeme definovat jako mechanické kmitání, které je charakterizováno parametry pohybu částic pružného prostředí, nebo u vlnového pohybu parametry zvukového pole. Část zvuku se projevuje jako slyšitelný zvuk, což je akustické kmitání pružného prostředí v pásmu frekvencí od 20Hz do 20kHz, schopné vyvolat zvukový vjem. Tento frekvenční rozsah je ovšem silně individuální a obzvláště s věkem se horní hranice postupně snižuje. Zvuky mimo toto pásmo jsou pro nás neslyšitelné, ovšem jsme je schopni určitým způsobem vnímat a mohou na nás mít také nepříznivý vliv, obzvláště zvuky se stejnými frekvencemi jaké mají biologické procesy člověka.

Zvuk vzniká kmitáním bodů a bodových soustav. Nejjednodušším způsobem vzniku harmonického zvuku je periodický pohyb sinusového průběhu. Tímto pohybem vznikne vzruch, který se v prostoru šíří formou postupného podélného vlnění a u harmonického pohybu tak vznikne harmonická vlna viz obrázek 4.1. Tato vlna je charakterizována amplitudou a vlnovou délkou.



Obr. 4.1: Harmonická vlna

4.1.1 Důležité parametry zvukové vlny

4.1.1.1 Výška

Výška je určena frekvencí. Absolutní výšku nedokážeme určit sluchem, proto výšku porovnáváme relativně poměrem k frekvenci základního tónu. V hudební akustice je referenčním tónem komorní „a“, které má frekvenci 440Hz.

4.1.1.2 Barva

Barva závisí na spektrálním složení zvuku, tvaru kmitů a poměru amplitud alikvotních tónů. Pomocí barvy zvuku jsme schopni rozlišit daný hudební nástroj.

4.1.1.3 Intenzita zvuku

Intenzita zvuku je objektivní fyzikální veličina pro měření síly zvuku. Intenzita zvuku je dána vzorcem:

$$I = \frac{P}{S} \left[1 \frac{W}{m^2} \right]$$

Veličina P je v tomto vzorci výkon zvukového vlnění a S je plocha, kterou vlnění prochází.

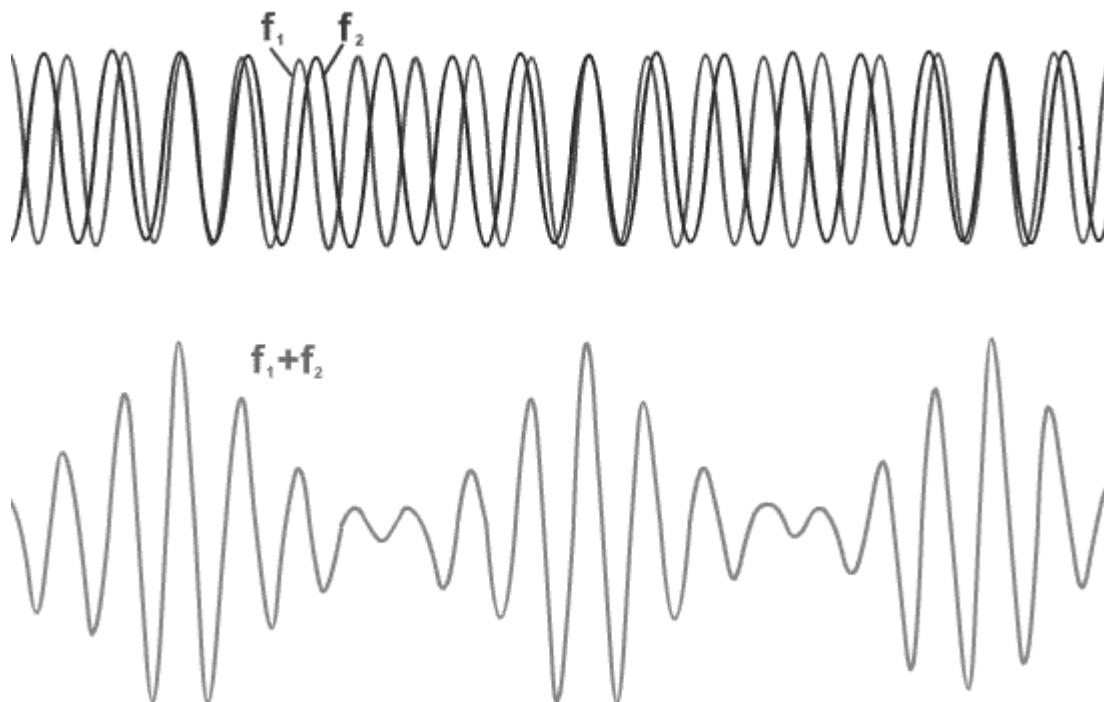
Největší citlivost ucha je při frekvencích 700Hz – 6kHz. Při těchto frekvencích vnímáme zvuky i o malé intenzitě $10^{-12} \text{ W.m}^{-2}$ – práh slyšitelnosti, zvuky o intenzitě 1 W.m^{-2} způsobují bolesti v uchu a určují práh bolesti. Jelikož je poměr mezi největší a nejmenší intenzitou značně velký – 10^{12} , tak se intenzita vyjadřuje ve speciální logaritmické stupnici. Jednotkou této stupnice je 1 bel [B]. Převod mezi touto stupnicí a intenzitou je dán vzorcem:

$$B = 10 * \log\left(\frac{I}{I_0}\right)$$

Výsledná hlasitost zvuku je vyjádřena v decibelech. Veličina I_0 je intenzita zvuku odpovídající prahu slyšení.

4.2 Interference vlnění

Pokud se v pružném prostředí šíří dvě nebo více vlnění stejného druhu, dochází k tzv. interferenci. Výsledný kmitavý pohyb hmotných bodů v prostředí je dán superpozicí kmitání vyvolaných vlněním. Posluchač není schopen rozlišit jednotlivé vlny samostatně a vnímá pouze jejich součet. Příklad interference dvou signálů viz obrázek 4.2.



Obr. 4.2: Interference vlnění dvou signálů

Na obrázku 4.2 je znázorněna interference dvou signálů s velmi blízkou frekvencí. Výsledná vlna má frekvenci rovnu aritmetickému průměru skládaných frekvencí $f = (f_1 + f_2) / 2$ a je modulována tzv. rázovou frekvencí $f_r = f_1 - f_2$. V případě, že skládané signály nemají stejnou frekvenci, nebo jejich poměr není celé číslo, se tato frekvence ve výsledném zvuku projeví jako kolísání intenzity zvuku.

4.3 Oktáva a oktávový systém

Hudební interval mezi prvním a osmým tónem diatonické (sedmitónové s intervalem celého tónu nebo půltónu mezi každými dvěma sousedními stupni) stupnice nazýváme oktávou. Oktáva tvoří základní stavební prvek naprosté většiny hudebních stupnic. Dva tóny vzdálené o oktávu mají v hudbě stejné jméno a poměr jejich frekvencí je 2:1. Jako základní tón názvosloví oktáv se používá tón C.

Jednotlivé oktávy:

- Subkontra oktáva.: od „C (16,4 Hz) do „H (tóny pod 16Hz lidské ucho nevnímá) .
- Kontra oktáva: od „C (32,7 Hz) do „H .
- Velká oktáva: od C (65,4 Hz) do H .
- Malá oktáva: od c (130,8 Hz) do h .

- Jednočárkovaná oktáva: od c' (261,5 Hz) do h' .
- Dvojitčárkovaná oktáva: od c'' (523,2 Hz) do h'' .
- Tříčárkovaná oktáva: od c''' (1046 Hz) do h''' .
- Čtyřčárkovaná oktáva: od c'''' (2093 Hz) do h'''' .
- Pětčárkovaná oktáva: od c''''' (4186 Hz) do h''''' .

4.4 Ladění a hudební stupnice

Ladění určuje přesné vzdálenosti mezi jednotlivými tóny a tím i jednotlivé frekvence tónů a poměr mezi nimi. Jednotlivým intervalům dává pevný řád hudební stupnice. V evropské hudbě se používá v dnešní době výlučně dvanáctitónová chromatická stupnice. Vytvořit takovou stupnici tak, aby všechny intervaly byly čisté ovšem není možné, jelikož poměr mezi oktávami je 2:1. Nemožnost rozdělit čistě oktávu se řeší různými systémy ladění tak, aby výsledné naladění bylo co nejlepší pro poslech. Každý systém ovšem musí v určitém hledisku ustoupit.

4.4.1 Rovnoměrné temperované ladění

Rovnoměrné temperované ladění je nepoužívanější ladění v evropské hudbě. Všechny intervaly stejného druhu jsou stejně velké, ale žádný interval kromě oktávy není úplně čistý. Modulace je možná do libovolně vzdálené tóniny bez vlivu na sluchový vjem. Oktáva je rozdělena na 12 stejně velkých intervalů - půltónů. Jelikož je poměr mezi oktávami 2:1 a je potřeba oktávu rozdělit na 12 stejných intervalů, je jeden krok mezi sousedními půltóny dán $\sqrt[12]{2}$. Frekvence tedy tvoří geometrickou posloupnost s uvedeným krokem.

4.5 Nástroje pro generování zvuku

V této podkapitole jsou popsány základní nástroje, které se používají pro generování zvuku. Vytvořená aplikace se nedá přímo zařadit do některé z kategorií. Určitým způsobem má blízko sampleru, jelikož pro generování zvuku jsou použity vzorky. Ovšem tyto vzorky přímo neodpovídají žádnému reálnému nástroji, tak zde nelze přímo zařadit. Splňuje také určité charakteristiky syntezátoru, protože lze měnit charakteristiky generovaného signálu. V aplikaci je toho důkazem možnost změnit barvu tónů za běhu. Generovaný průběh je ale pouze jedna vlna a tak nelze hovořit o syntéze.

4.5.1 Sampler

Samplery jsou nástroje, které pro generování zvuku využívají nahrané vzorky reálných nástrojů. Při přehrávání těchto vzorků je možné ovlivnit frekvenci a intenzitu zvuku, aniž by došlo ke změně barvy zvuku a je také možné vzorky přehrávat různou rychlostí. Pomocí těchto úprav je umožněno docílit změny výšky zvuku. Toto přeladování má ovšem žádoucí výsledky pouze do určitých mezí, jelikož při větším přeladění vzorků dochází i k přeladění všech rezonancí, které obsahoval nahraný nástroj a dochází tak k nepřirozené změně barvy generovaných tónů. Z těchto důvodů se používá pro každý reálný nástroj většího počtu vzorků různých výšek tónů. Při použití více vzorků je každý použit pouze pro několik sousedních tónů. V případě, že je cílem co nejvěrnější napodobení hudebního nástroje, lze použít pro každý tón vlastní vzorek. Kromě uvedeného rozdělení rozsahu nástroje na jednotlivé vzorky na tzv. frekvenční pásma, se navíc využívá dynamických pásem, kdy jsou využity různé vzorky pro různé intenzity. V praxi pak často dochází ke kombinaci obou dělení. Hlavní výhodou samplerů je velmi reálné napodobení jednotlivých tónů. Problémem je ovšem hra souvislých melodií. Jelikož nastavené parametry hraných tónů je během hry obtížné měnit a celkový zvukový vjem může být od reálného značně odlišný. Z tohoto důvodu bývají často samplery doplňovány o různé filtry, aby bylo umožněno ovlivňovat zvuk v reálném čase.

4.5.2 Syntezátor

Syntezátory jsou elektronické nástroje, které napodobují zvuk reálných hudebních nástrojů s použitím různých druhů syntézy a jejich kombinací. Zvuk těchto nástrojů je pak vytvářen kombinací a úpravou jednoduchých vlnových průběhů. Hlavní výhodou syntezátorů je možnost měnění parametrů zvuku v reálném čase. Lepších výsledků lze docílit kombinací s nahrávkami vzorků reálných nástrojů.

Mezi způsoby generování zvuku syntezátory patří:

- Aditivní syntéza.
- Subtraktivní syntéza.
- Frekvenční modulace.
- Fázová modulace.
- Granulární syntéza.
- Wavetable syntéza.
- Fyzikální modelování.

Jednotlivé typy syntézy jsou vhodné pro různé hudební nástroje.

4.5.3 Virtuální nástroje

Tyto nástroje jsou založeny na matematické modelování, kdy na rozdíl od principů použitých v syntezátorech a samplech se snaží vytvořit pomocí matematických výrazů matematický model reálného nástroje. Kvalita výsledného zvuku je závislá na kvalitě vytvořeného modelu a pokud je reálný nástroj namodelován dobře, může velmi důvěryhodně po všech stránkách napodobovat reálný nástroj.

V této kapitole jsou některé informace čerpány z literatury [1],[8] a [9].

5 Fáze návrhu a implementace aplikace

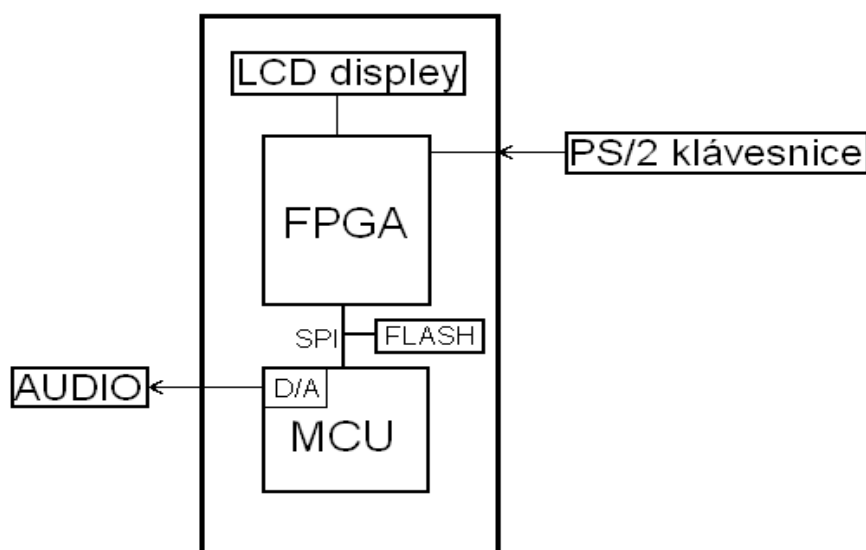
Tato kapitola popisuje jednotlivé fáze návrhu a implementace aplikace generátoru tónů.

Zadáním byla jako uživatelský vstup určena klávesnice, připojena přes PS/2 rozhraní. Základem je, aby uživatel měl jednoduchým způsobem možnost zadávat hrané tóny. Z tohoto důvodu, byl jako vzor návrhu vstupu tónů zvolený hudební nástroj klavír a to proto, že princip hraní na něj se dá na klávesnici realizovat nejlépe ze všech hudebních nástrojů. Jednotlivé tóny jsou tedy rozmístěny na klávesnici tak, aby co nejvíce napodobovaly reálný nástroj (viz příloha [1]).

Nejprve bylo nutné vytvořit řadiče jednotlivých částí aplikace v FPGA. Pro komunikaci mezi hradlovým polem a mikrokontrolérem, které jsou propojeny rozhraním SPI, bylo nutné vytvořit řadiče, který převede SPI protokol na interní sériovou sběrnici. Aby bylo možné adresovat jednotlivé řadiče v FPGA a také pro převedení sériového protokolu na paralelní rozhraní daného řadiče, je další realizovanou komponentou v FPGA SPI dekodér. Hlavní funkcí výsledné konfigurace FPGA je čtení příchozích dat z klávesnice a zapisování dat do řadiče LCD displeje. K těmto účelům byly vytvořeny řadiče těchto periférií a pro možnost čtení dat z řadiče klávesnice jen v okamžiku načtených dat také řadič přerušení, který vyvolá přerušení v mikrokontroléru. K vytvoření výsledné konfigurace FPGA bylo nutné vytvořit vzájemnou komunikaci mezi komponentami.

Základní funkcí mikrokontroléru v aplikaci je generování tónů podle aktuálně stisknutých kláves. Mikrokontrolér musí umět generovat také více tónů ve stejný okamžik. K vyřešení tohoto problému bylo nutné vybrat jedno ze dvou možných řešení. Realizované řešení je založeno na použití časovače pro počítání doby jednoho cyklu výpočtů a generování vzorku signálu, které odpovídají tomuto času. Druhé nabízené řešení by bylo použití přerušení od časovače. V každém přerušení by byl vypočítán vzorek a ten generován s pomocí DA převodníku. Toto řešení by sice mělo tu výhodu, že by neustále nezatěžovalo mikrokontrolér, ovšem bylo by nutné přesně stanovit dobu ve které budou vzorky generovány a tím předem určit maximální možný počet současně generovaných tónů a jejich kvalitu. Realizované řešení umožňuje dynamické změny kvality podle náročností výpočtu způsobeného hraním více tónů zároveň.

5.1 Blokové schéma generátoru tónů



Obr. 5.1: *Blokové schéma aplikace*

Na obrázku 5.1 je znázorněno blokové schéma, zobrazující důležité komponenty FITkitu a připojených periférií, z hlediska realizace generování tónů. Vstupem pro zadávání hraných tónů je PS/2 klávesnice, která je připojena k FITkitu na piny programovatelného hradlového pole FPGA. Na piny hradlového pole je také připojen LCD displej, který zobrazuje hrané tóny. Generování jednotlivých tónů je realizováno pomocí mikrokontroléru prostřednictvím DA převodníku, který je jeho součástí a na jehož výstup je připojen přes audio rozhraní reproduktor, který slouží k výsledné reprodukci tónů.

5.2 SPI komunikace

5.2.1 Rozhraní SPI

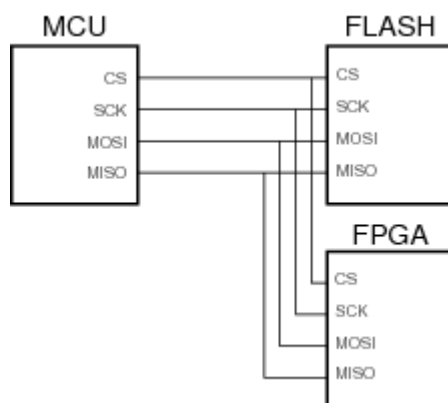
Rozhraní SPI je synchronní sériové rozhraní, které se používá k připojení periférií k mikrokontroléru, případně k vzájemnému propojení mikrokontrolérů. Toto zařízení tedy umožňuje kromě přímého propojení mezi dvěma zařízeními také vytvořit určitou sběrnici k propojení více zařízení, kdy ovšem v daný okamžik mohou komunikovat pouze dvě zařízení mezi sebou. Toto rozhraní je plně duplexní, což znamená, že umožňuje v každém okamžiku komunikovat oběma směry. Každé zařízení, které je připojeno na sběrnici SPI může být v jednom ze dvou režimů. Prvním režimem je master a v tomto režimu se může nacházet pouze jedno zařízení. Zařízení v tomto režimu generuje hodinový signál a rozhoduje se kterým zařízením v režimu slave se bude komunikovat. V režimu master většinou bývá mikrokontrolér.

Rozhraní SPI je tvořeno těmito čtyřmi signály:

- MOSI (master out slave in) – výstup dat z master zařízení.
- MISO (master in slave out) – výstup dat ze slave zařízení.
- CS (chip select) – signál sloužící pro výběr slave zařízení pro komunikaci z master zařízením.
- SCK (clock) – hodinový signál generovaný zařízením v režimu master.

5.2.2 SPI na FITkitu

Rozhraní SPI realizuje na FITkitu komunikaci mikrokontroléru s řadiči periférií realizovanými v FPGA a s paměti FLASH, ve které je uložen program mikrokontroléru a konfiguraci FPGA.



Obr.5.2: SPI rozhraní

Paměť FLASH a FPGA sdílejí všechny signály. Rozlišení adresovaného zařízení je realizováno na úrovni přenášených bitů. Maximální frekvence je 3.6864MHz.

5.2.3 Protokol komunikace mikrokontroléru s FPGA

Pro komunikaci s libovolným množstvím zařízení v FPGA je potřeba mít k dispozici vytvořenou interní sběrnici a protokol dle jakého se bude komunikace řídit. V FPGA můžeme komunikaci dle daného protokolu rozdělit na 3 části:

- Odeslání operačního kódu. Operační kód je osmice bitů identifikující FPGA a nesoucí informace pro SPI řadič v FPGA.
- Adresa cílového zařízení. Adresu tvoří libovolný počet bitů určujících zařízení v FPGA, se kterým proběhne komunikace.
- Přenos libovolného počtu bitů dat.

V této kapitole jsou některé informace čerpány z literatury [2].

6 FPGA

6.1 Řadič SPI

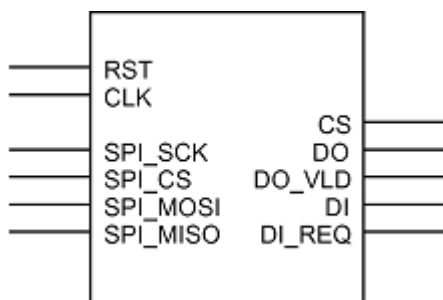
Řadič SPI je komponenta pro převod SPI protokolu na interní sériovou sběrnici a pro rozlišení, zda je transakce určena FPGA.

6.1.1 Popis funkce

Komunikace je zahájena nastavením signálu SPI_CS na úroveň logické „0“. Po zahájení je vzorkován hodinový signál SPI sběrnice SPI_SCK hlavním hodinovým signálem CLK. Při vzestupné hraně signálu SPI_SCK je čten bit signálu SPI_MOSI, při spádové hraně je zapisován bit do signálu SPI_MISO.

První čtyři načtené bity určují, zda má být komunikace navázána s FPGA. V případě, že první čtyři bity odpovídají hodnotě identifikující FPGA (0001), je aktivován signál CS a je povoleno generování signálů DO_VLD a DI_REQ. Výstupem dat z mikrokontroléru je signál DO, jehož platnost určuje signál DO_VLD a vstupem do mikrokontroléru je signál DI. Jelikož je SPI rozhraní typu master-slave a přenos je řízen mikrokontrolérem, je pro určení doby kdy je nutné vystavit další bit dat na DI použit signál DI_REQ.

V případě, že se první čtyři bity neshodují s identifikací FPGA, není navázána komunikace, signál CS je neaktivní a signály SPI_MISO a DI musí být ve stavu vysoké impedance.



Obr. 6.1: Řadič SPI sběrnice

6.2 SPI dekodér

SPI dekodér je komponenta pro převod interní sériové komunikace na paralelní rozhraní pro komunikaci s připojenou periférií.

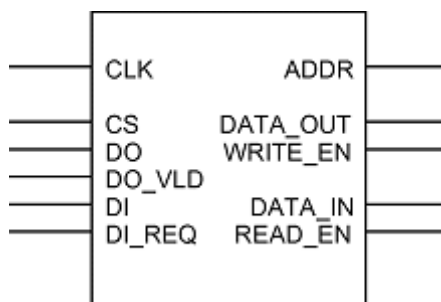
V aplikaci jsou realizovány SPI dekodéry pro LCD displej, PS/2 klávesnici a řadič přerušení. Nastavení generických parametrů určujících šířku adresy, dat, adresní sběrnice připojeného zařízení a bázovou adresu zařízení znázorňuje tabulka 6.2.

zařízení	ADDR_WIDTH	DATA_WIDTH	ADDR_OUT_WIDTH	BASE_ADDR
PS/2 Klávesnice	8	8	1	0x02
LCD displej	8	8	1	0x00
Řadič přerušení	8	8	1	0x80

Tab. 6.2: Generické parametry

6.2.1 Popis funkce

SPI dekodér je aktivován signálem CS. Dekodér nejprve přijme čtyři nejméně významné bity operačního znaku, podle kterých zjistí, zda se bude jednat o čtení, zápis nebo případně obojí. Další fází je příjem určitého počtu adresních bitů. Nejvyšší bity určují cílové zařízení. Komunikace pokračuje pouze se zařízením, jehož bázová adresa se s těmito bity shoduje. Zápis dat do zařízení je vždy realizován v okamžiku, kdy je přijat správný počet bitů daný parametrem DATA_WIDTH a jen v případě, že je nastaven režim zápisu. V okamžiku kdy jsou data vystavena na sběrnici DATA_OUT je generován signál WRITE_EN. Čtení probíhá odlišným způsobem. Po vygenerování signálu READ_EN jsou data vystavena na sběrnici DATA_IN v dalším taktu hodinové signálu.



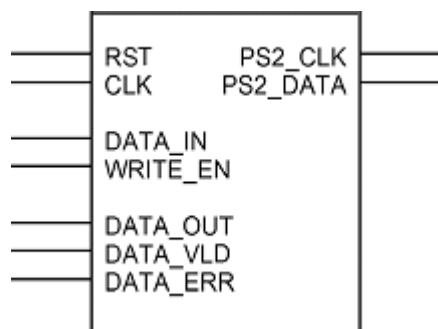
Obr. 6.3: Dekodér SPI sběrnice

6.3 Řadič PS/2 klávesnice

Řadič PS/2 je komponenta realizující převod sériového PS/2 protokolu na paralelní protokol klávesnice.

6.3.1 Popis funkce

Řadič komunikuje s klávesnicí dle protokolu popsaného v kapitole 3.2. Po příjmu dat z klávesnice, oznámí mikrokontroléru signálem DATA_VLD a tyto data vystaví na sběrnici DATA_OUT, odkud si je mikrokontrolér načte. V případě komunikace opačným směrem musí mikrokontrolér nejprve aktivovat signál WRITE_EN pro povolení zápisu a v dalším taktu vystaví data na sběrnici DATA_IN. Řadič tyto data zpracuje a odešle klávesnici dle komunikačního protokolu klávesnice.



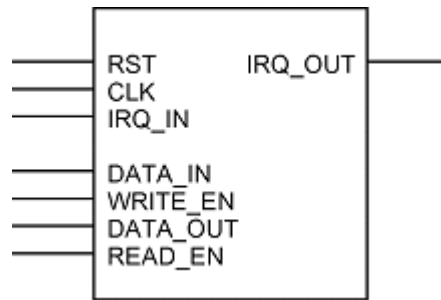
Obr. 6.4: Řadič PS/2 klávesnice

6.4 Řadič přerušení

Z některých komponent není potřeba data číst neustále, většinou z důvodu velkých intervalů mezi daty nebo jejich nepravidelnosti. U těchto komponent by bylo nevýhodné neustále zjišťovat stav jejich řadiče. Z tohoto důvodu je výhodné použít řadič přerušení. V aplikaci je použita klávesnice, u které je výhodné tohoto řadiče využít. Řadič je implementován v FPGA a v případě příchodu požadavku na přerušení, které je vyžádáno vždy v době, kdy jsou připravena v řadiči data z klávesnice, generuje žádost o přerušení pro mikrokontrolér. Mikrokontrolér si následně přečte vektor přerušení z řadiče uvnitř FPGA a podle jeho hodnoty zjistí jeho zdroj. V aplikaci je jediným zdrojem přerušení řadič klávesnice a v důsledku přerušení přečte mikrokontrolér data z řadiče klávesnice.

6.4.1 Popis funkce

Při restartu FPGA dojde k nastavení masky přerušení zakazující veškerá přerušení. Po této inicializaci je potřeba tuto masku změnit na takovou, která povoluje přerušení vyvolané řadičem klávesnice. V případě detekce žádosti o přerušení na vstupu IRQ_IN je generována žádost směrem do mikrokontroléru signálem IRQ_OUT. Mikrokontrolér si po zpracování žádosti o přerušení přečte přes sběrnici SPI vektor přerušení a tento vektor následně vynuluje. V důsledku této události je deaktivován signál IRQ_OUT.



Obr. 6.5: Řadič přerušení

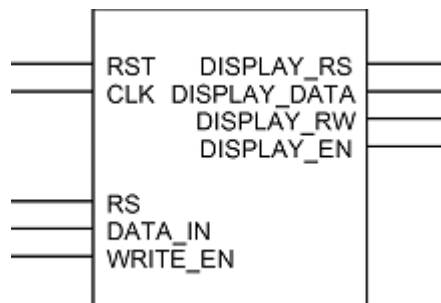
6.5 Řadič LCD displeje

Šestnáctiznakový LCD displej je zobrazovací prvek zabudovaný přímo na FITkitu. Tento displej obsahuje vlastní řadič, který realizuje zobrazování dat na LCD prvku. Řadič obsahuje tři paměti. Jedna z těchto pamětí je typu ROM a obsahuje bodové předlohy znaků, které je možno zobrazovat na LCD. Kromě této paměti obsahuje ještě další dvě paměti RAM pro definování vlastních znaků a ukládání aktuálního stavu zobrazovaných znaků.

V FPGA je potřeba realizovat řadič, který bude umět komunikovat s řadičem LCD.

6.5.1 Popis funkce

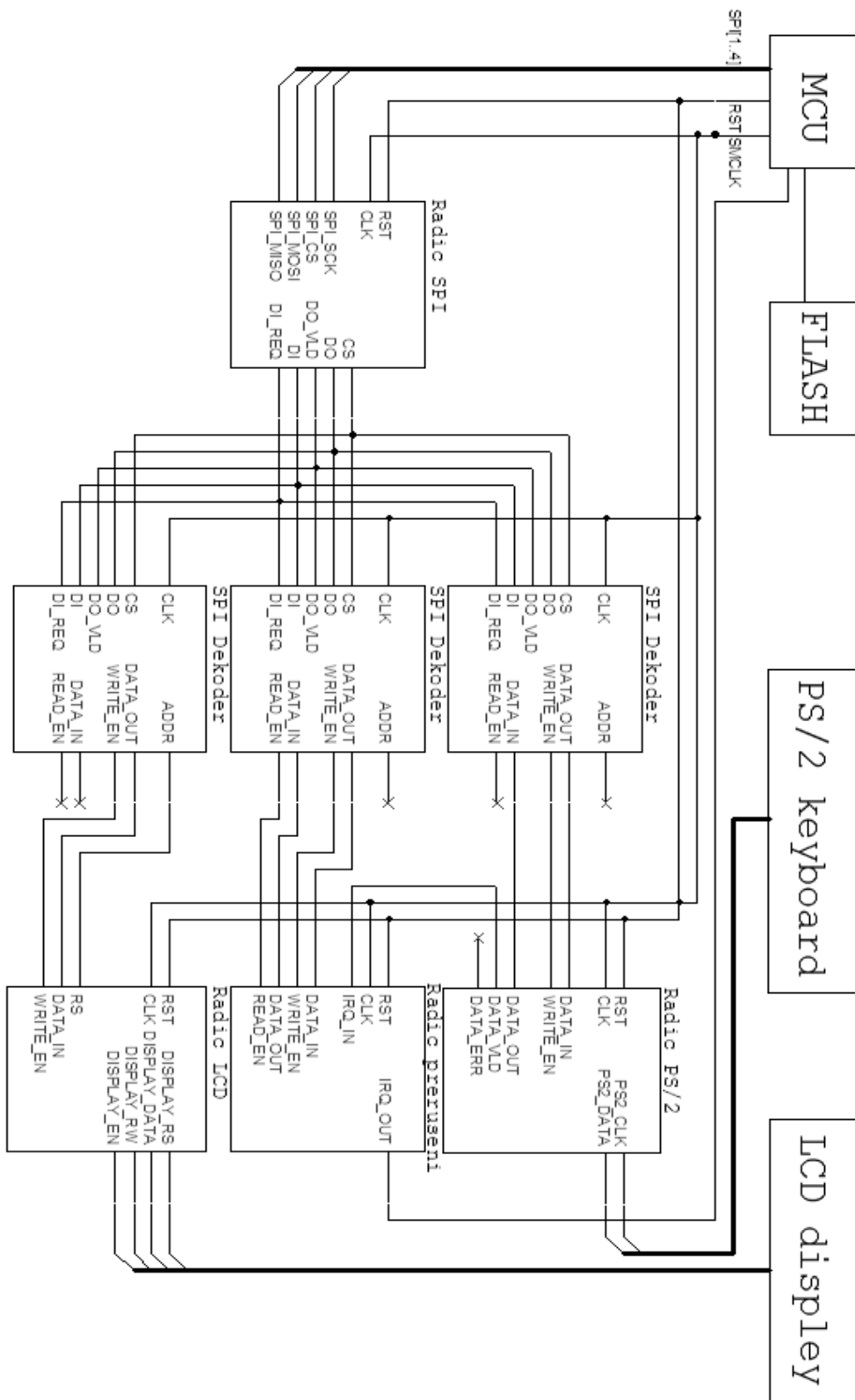
Data pro LCD displej jsou posílána mikrokontrolérem přes sběrnici SPI a interní sběrnici realizovanou v FPGA do řadiče LCD displeje, který je rovněž realizován v FPGA. Přichozí data mohou znamenat kromě konkrétních dat také instrukci pro ovládání displeje. Pro rozlišení instrukcí od dat je použit signál RS. Plné funkčnosti LCD displeje lze dosáhnout, aniž by bylo potřeba číst aktuální stav. Z tohoto důvodu, obsahuje řadič realizovaný v FPGA pouze piny pro sběrnici vstupních dat z mikrokontroléru DATA_INT a signál, který povolí zapsání dat do LCD označený WRITE_EN.



Obr. 6.6: Řadič LCD displeje

V této kapitole jsou některé informace čerpány z literatury [2],[3],[4] a [5].

6.6 Schéma konfigurace FPGA



Obr. 6.7: Schéma konfigurace FPGA

7 Mikrokontrolér

7.1 Konfigurace registrů mikrokontroléru

Pro generování tónů je zapotřebí mít možnost ovlivňovat analogové napětí na výstupu a toto napětí měnit v určitých pravidelných intervalech, které jsou dány konkrétními tóny. Z tohoto důvodu je zapotřebí mít možnost vhodně nastavit a použít moduly mikrokontroléru, které tyto požadavky umožní realizovat. Těmito moduly jsou DA převodník a časovač.

7.1.1 DA převodník

Součástí mikrokontroléru jsou dva dvanáctibitové digitálně analogové převodníky. Každý z těchto převodníků obsahuje dva hlavní registry.

Prvním registrem je DAC12_XCTL. Pro aplikaci generátoru tónů je nastaven takto:

- Referenčním napětím je zvoleno napětí V_{ref+} . Toto napětí je přivedeno z modulu AD převodníku, ve kterém je zabudován generátor interního napětí. S pomocí registrů AD převodníku lze nastavit hodnotu referenčního napětí, kdy pro aplikaci bylo zvoleno napětí 2,5V.
- Výstup DA převodníku je aktualizován vždy při zápisu do jeho datového registru.
- Je zapnuta kalibrace offsetu, aby bylo zabráněno generování nesprávných hodnot.
- Vstupní a výstupní zesilovače jsou nastaveny na maximální rychlost.
- Je zvoleno dvanáctibitové rozlišení převodníku, především z důvodu možnosti generování více tónů najednou.

Druhým registrem je DAC12_XDAT. Je to datový registr do nějž se zapisují data, která se po zápisu načtou na výstup převodníku.

7.1.2 Časovač

Mikrokontrolér obsahuje dva moduly časovače. V aplikaci je využit pouze jeden tento modul s označením „Timer_A“.

Registr pro nastavení použitého časovače má označení TACTL. Tento registr je v aplikaci nastaven tak, aby časovač čítal s frekvencí 7,3278MHz a čítal ve smyčce od nuly do maximální hodnoty. Registr ve kterém je uchován stav aktuální hodnoty čítače má označení TAR. Po inicializaci čítače se v aplikaci pracuje pouze s tímto registrem, jehož hodnota je čtena a nulována podle potřeby.

7.2 Komunikace s perifériemi

V této podkapitole je popsána komunikace mezi mikrokontrolérem a řadiči realizovanými v FPGA. Mikrokontrolér komunikuje s těmito řadiči prostřednictvím řadiče SPI sběrnice a dekodéru pro konkrétní řadič .

7.2.1 PS/2 klávesnice

Stisknutí klávesy je událost, o které předem nevíme kdy nastane. Mikrokontrolér by byl příliš vytížen, kdyby musel neustále zjišťovat, zda tato událost nastala. Z tohoto důvodu, je pro oznámení o připravených datech v řadiči klávesnice mikrokontrolér informován prostřednictvím řadiče přerušení.

Při příchodu žádosti o přerušení z FPGA mikrokontrolér nejprve přečte vektor přerušení z řadiče realizovaného v FPGA a pro možnost generování dalšího přerušení jej po načtení vynuluje. V aplikaci je přerušení využito pouze u klávesnice a vektor přerušení vždy pouze informuje o přerušení z klávesnice. Poté, co mikrokontrolér zjistí, že jsou připravena data, spustí rutinu, která tato data načte a zpracuje. Příchozí data odpovídají scan kódu některé z kláves, nebo kódu informujícím o uvolnění klávesy. Načtení dat z řadiče klávesnice v FPGA probíhá pomocí sběrnice SPI. Pro zaadresování umístění těchto dat musí mikrokontrolér nejprve poslat jeden bajt obsahující adresu FPGA, která je rozpoznána v řadiči SPI a povolení čtení. Po této adrese následuje bajt s adresou konkrétního řadiče, která je pevně nastavena v dekodéru pro řadič klávesnice. V případě že jsou tyto adresy správné, jsou tato data přes dekodér a řadič SPI poslána do mikrokontroléru. Jednotlivé scan kódy jsou popsány v příloze [2].

7.2.2 LCD displej

Komunikace s řadičem LCD displeje je realizovaná pouze jednosměrně. Pro zobrazování hraných tónů v aplikaci je to dostačující. Jediným problémem při hraní více tónů najednou může být to, že není možné zobrazit všechny hrané tóny, protože displej má pouze 16 znaků a některé tóny jsou označeny až pěti znaky. Posílaná data mají vždy 3 bajty. První bajt obsahuje adresu FPGA a pro možnost zápisu dat do řadiče displeje také bit povolující zápis, který musí být povolen. Druhý bajt obsahuje básovou adresu displeje a jeden bit (RS) určující typ dat. V posledním bajtu jsou obsažena již konkrétní data.

7.3 Program mikrokontroléru

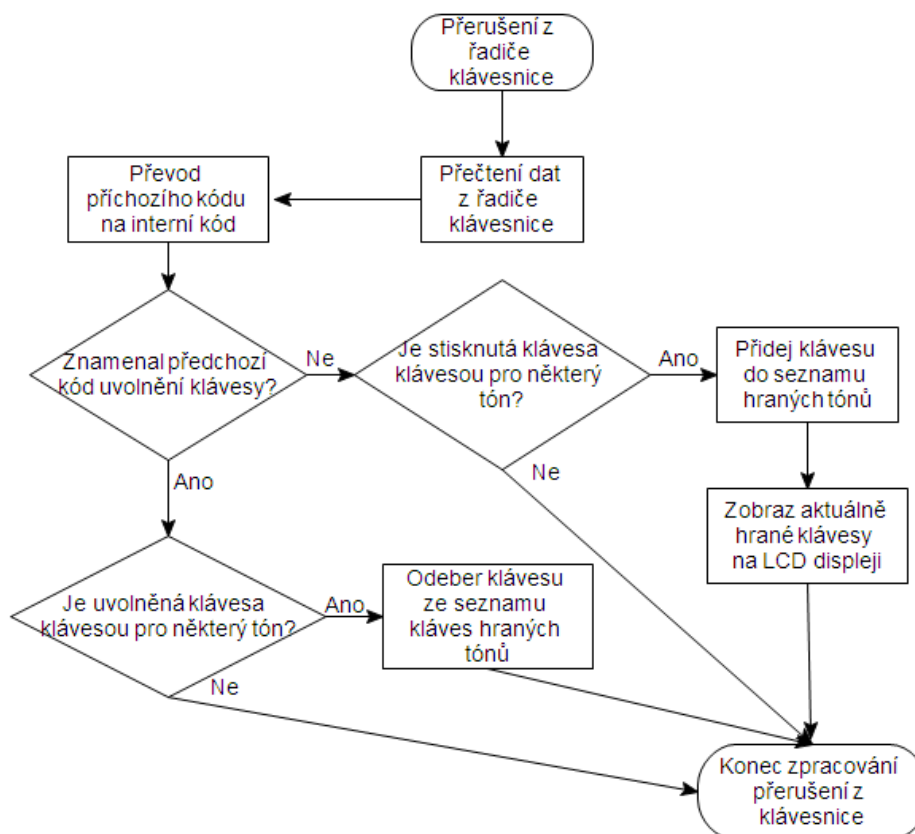
Hlavní činnost mikrokontroléru v aplikaci se skládá ze dvou úloh. První úlohou je zpracování příchozích přerušení, generovaných z hradlového pole, vyvolaných klavesnicí. Druhou úlohou je generovat určitou hladinu napětí, s frekvencí odpovídající stisknutému tlačítku a s požadovaným průběhem určujícím barvu tónů.

7.3.1 Zpracování vstupu z klávesnice

Z důvodu realizace více úloh v mikrokontroléru zároveň je nutné, aby čtení dat z řadiče klávesnice neprobíhalo neustále a nečekalo se, až se v řadiči nějaká data objeví, ale aby se data načetly pouze v okamžiku až jsou v řadiči připravena. Z tohoto důvodu je načítání realizováno v obslužné rutině přerušení generovaného z FPGA. Načítání dat je realizováno do malého bufferu, aby mohlo být uloženo více scan kódů, pro případ, že by se nestihly zpracovat příchozí data do doby dalšího přerušení.

Při příchodu přerušení z klávesnice je důležité nejprve na toto přerušení reagovat přečtením dat přes sběrnici SPI z řadiče klávesnice realizovaném v FPGA. Tyto data mohou být různého charakteru z hlediska aplikace a následné zpracování je pro každý typ dat jiný. Příchozí data mohou znamenat kód klávesy, která má reprodukovat tón a nebo změnit parametry tónů, kód znamenající uvolnění klávesy a nebo kód který je z hlediska aplikace nevýznamný. V případě kódu kláves které odpovídají některému z tónu je také důležité, zda je klávesa stisknuta nebo uvolněna.

Postupné zpracování jednotlivých příchozích kódů a jejich zpracování je znázorněno diagramem viz obrázek 7.1.



Obr. 7.1: Diagram zpracování vstupu z klávesnice

V programu, jak je znázorněno zjednodušeně v diagramu na obrázku 7.1, je uchováván seznam všech aktuálně stisklých kláves. Do tohoto seznamu se ukládají pouze ty klávesy, které v něm

ještě nejsou. Kromě tohoto seznamu je ukládán také seznam názvů tónů aktuálně stisklých kláves z důvodu jejich zobrazování na LCD displeji. Zobrazení tónů z tohoto seznamu je vždy realizováno při změně stavu některé z kláves.

7.3.2 Generování tónů

Tato podkapitola popisuje všechny principy a výpočty použité pro generování tónů mikrokontrolérem. Toto generování je realizováno v nekonečné smyčce ihned po inicializaci všech modulů k tomu potřebných. Výpočty spojené s generováním probíhají neustále z toho důvodu, že není předem známa jejich délka a také kvůli realizaci generování různého počtu tónů najednou. Tyto principy se snaží umožnit generovat tóny s takovou kvalitou, která odpovídá celkové délce výpočtu. Z praktického hlediska to znamená, že čím více tónů je generováno, tím více klesá jejich kvalita.

7.3.2.1 Výpočty spojené s výškou tónů

Pro realizaci generování tónů s odpovídající výškou, je důležité vypočítat frekvence jednotlivých tónů a tomu odpovídající počet taktů mikrokontroléru pro jednu periodu generovaného signálu.

Základní referenční tón komorní „a“ má frekvenci 440Hz a odvíjí se od něj všechny ostatní tóny. Podle rovnoměrného temperovaného ladění, které se používá v evropské hudbě, je jeden krok mezi půltóny $\sqrt[12]{2}$. Pomocí těchto dvou údajů lze vypočítat frekvence všech potřebných tónů.

K výpočtu počtu taktů mikrokontroléru pro jednotlivé tóny, je důležitým parametrem frekvence těchto taktů, která je rovna 7,3728MHz. Výsledný počet taktů je pak dán podílem $7372800/f$, kde f je frekvence tónu pro který se výpočet počítá. Všechny vypočtené hodnoty jsou obsaženy v tabulce 7.2.

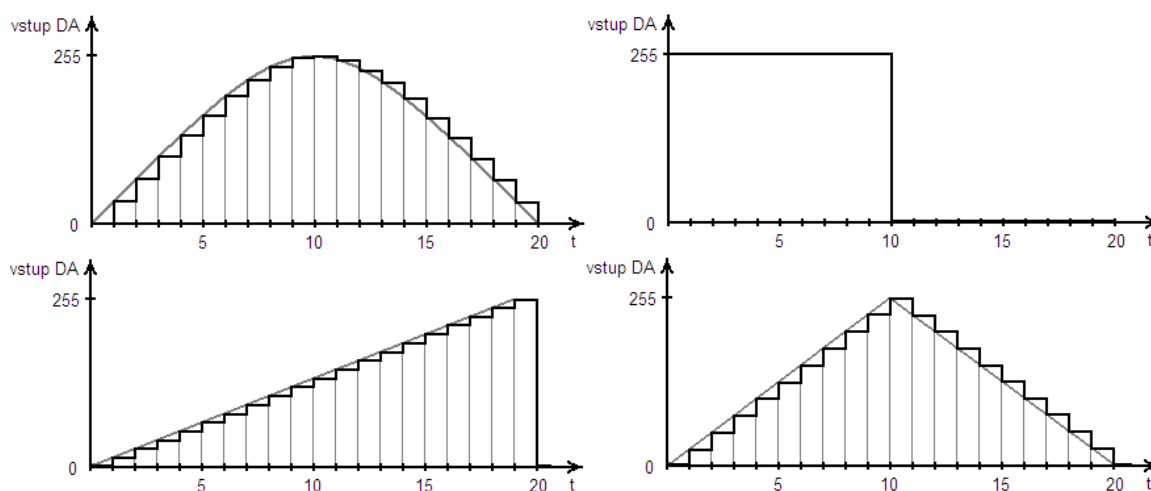
tón	f[Hz]	taktů MCU	klávesa	tón	f[Hz]	taktů MCU	klávesa	tón	f[Hz]	taktů MCU	klávesa
c	131	56361	q	c'	262	28181	i	c''	523	14090	c
cis	139	53198	2	cis'	277	26599	9	cis''	554	13300	f
d	147	50212	w	d'	294	25106	o	d''	587	12553	v
dis	156	47394	3	dis'	311	23697	0	dis''	622	11849	g
e	165	44734	e	e'	323	22367	p	e''	659	11184	b
f	175	42223	r	f'	349	21112	[f''	698	10556	n
fis	185	39854	5	fis'	370	19927	=	fis''	740	9963	j
g	196	37617	t	g'	392	18808]	g''	784	9404	m
gis	208	35505	6	gis'	415	17753	a	gis''	831	8876	k
a	220	33512	y	a'	440	16756	z	a''	880	8378	,
ais	233	31632	7	ais'	466	15816	s	ais''	932	7908	l
h	247	29856	u	h'	494	14928	x	h''	988	7464	.
								c'''	1047	7045	/

Tab. 7.2: Vypočtené hodnoty pro výšky tónů

7.3.2.2 Princip realizace barvy tónů

Napodobení reálných nástrojů by bylo velmi složité, jelikož náročnost na výkon mikroprocesoru by byla velmi velká. Z tohoto důvodu je barva tónů v aplikaci ovlivňována pouze tvarem signálu. Základním průběhem, který by byl pro generování nejvhodnější je průběh sinusový. Zde ovšem nastává problém v tom, že použitý DA převodník, který je součástí mikrokontroléru na kitu je schopen generovat pouze kladné napětí. Jednou z variant, která se nabízela bylo posunout tento průběh celý do kladných hodnot. Toto řešení ovšem při praktické realizaci mělo při testování úplně jiné zvukové vjemy, než původní průběh a to značně horších kvalit. Z tohoto důvodu je jako základní průběh použita pouze kladná půlvlna sinusového průběhu. Tento signál není jediný, který je možné generovat. Kromě sinusového průběhu jsou implementovány navíc další základní signály a to obdélníkový, pilový a trojúhelníkový.

Implementace barvy tónů je realizována s pomocí vzorků těchto průběhů. Pro určení počtu vzorků jedné periody generovaných signálů bylo nutné zvážit jejich výslednou kvalitu, ale rovněž by bylo zbytečné, aby těchto vzorků bylo příliš mnoho a časová vzdálenost mezi nimi byla menší, než by trval výpočet jednoho vzorku, což by se nejvíce projevilo u tónu s nejvyšší frekvencí. Generované průběhy jsou zobrazeny na obrázku 7.3.



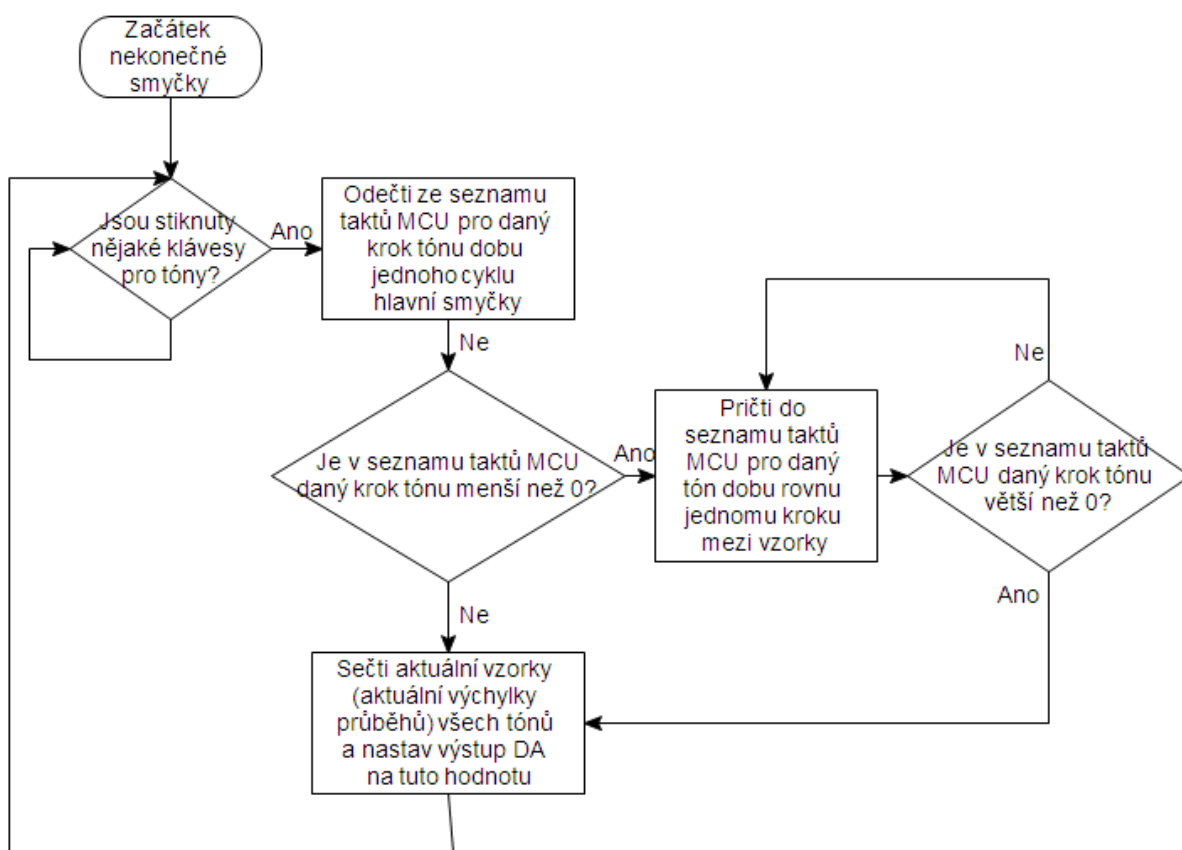
Obr. 7.3: Časové průběhy generovaných tónů

Na obrázku 7.3 je vždy u daného signálu znázorněn ideální průběh a průběh, který přibližně odpovídá generovanému signálu v případě, že mikrokontrolér stihne během kroku jednoho vzorku vypočítat další. Přibližně je to z důvodu možnosti realizování hry více tónů najednou, kdy tato doba nelze přesně stanovit, jelikož je pro každý tón jiná, ovšem k výpočtu všech tónů musí dojít ve stejném čase. Detailněji je tato problematika popsána v následující podkapitole 7.3.2.3.

7.3.2.3 Princip generování

Zvuk jako mechanický druh vlnění se skládá interferencí. Při interferenci se vlnění skládají superpozicí, což v případě aplikace, kdy je zvuk generován v jednom místě, znamená součet výchylek všech vzorků aktuálně hraných tónů. Aby byly změny průběhu jednotlivých tónů zachyceny co nejdříve, jsou hlavní výpočty umístěny v nekonečné hlavní smyčce. Druhým možným řešením se nabízelo použití přerušení od časovače s periodou přerušení daného krokem mezi vzorky. Toto řešení se obzvláště pro řešení situace hry více tónů ve stejnou dobu značně zkomplikovalo a proto od něj bylo upuštěno.

Princip hlavní smyčky generující hrané tóny je znázorněn na diagramu viz obrázek 7.4.



Obr. 7.4: Diagram znázorňující princip generování tónů

V paměti je uchováván seznam, který obsahuje pro každý aktuálně hraný tón počet taktů mikrokontroléru mezi dvěma vzorky generovaného tónu. Tento seznam je uložen dvakrát. Je to z toho důvodu, že od jednoho seznamu se odečítá již uběhlý počet taktů a až se vynuluje, je znovu načten z druhého seznamu, který je statický. Cyklus výpočtu vždy probíhá, pokud jsou stisknuty některé klávesy. V každém cyklu je pomocí časovače zjištěn počet strojových cyklů minulého cyklu a tento počet je odečten od počtu taktů mezi vzorky v seznamu u všech tónů. V případě, že u některého tónu již tato doba uběhla, jinak řečeno je tato hodnota menší nebo rovna nule v seznamu taktů pro tento

tón, je nastavena aktuální výchylka pro daný tón na další vzorek v pořadí. Také je v tomto případě v seznamu taktů mikrokontroléru u tohoto tónu doba navýšena o hodnotu v druhém seznamu, uchovávaného původní hodnoty. V této situaci se ovšem může stát, že doba jednoho cyklu je větší, než doba jednoho kroku. Kdyby tato situace nebyla ošetřena, došlo by k zkreslení generovaného signálu a zvukový vjem by byl neodpovídající požadovanému tónu. Z tohoto důvodu, pokud nastane tato situace, není použit následující vzorek, ale vzorek o tolik kroků dále, aby časově odpovídal.

V této kapitole jsou použity informace z literatury [6].

8 Závěr

Vytvoření aplikace generátoru tónů se skládalo z naprogramování dvou komponent. První touto komponentou bylo FPGA, kde bylo nutné naprogramovat jeho konfiguraci. Základní knihovna FITkitu obsahuje sadu řadičů, které značně zjednodušily návrh konfigurace FPGA a také její realizaci.

Druhá komponenta, kterou bylo nutné naprogramovat byl mikrokontrolér. Je zde opět využita knihovna FITkitu a to především pro komunikaci s FPGA přes sběrnici SPI a pro ovládání LCD displeje jsou rovněž využity funkce této knihovny. Zbývající problematika obnášející vyřešení a realizaci generování tónů a přijímání dat z klávesnice, musela být zcela vyřešena s použitím jen základních knihoven mikrokontroléru. Pro zpracování dat z klávesnice byl vytvořen zvlášť modul, který je možné použít i v jiných aplikacích.

Výsledná aplikace generátoru tónů zvládá generovat jednotlivé tóny bez jakýchkoliv problémů. Generování více tónů najednou je problematičtější, jelikož obnáší složitější výpočty a větší požadavky na rychlost výpočtů mikrokontroléru, která je omezená. Generování je realizováno tak, aby všechny základní tóny měli stejnou kvalitu a kvalita začne klesat až v případě, že mikrokontrolér přestane stíhat výpočet vzorků všech tónů a z tohoto důvodu dochází ke snížení vzorkovací frekvence. Z těchto důvodů nelze přesně určit možný počet tónů, který aplikace zvládá zároveň generovat a například tóny nižších frekvencí zvládá generovat o několik více než u vyšších frekvencí. Aplikace umožňuje změnit barvu tónů za běhu na jednu ze čtyř předdefinovaných možností. Tato změna je realizována změnou tvaru generovaných signálů. Z těchto čtyř signálů je po stránce sluchového vjemu nejpříjemnější signál jehož průběh odpovídá kladné části sinusové vlny.

Z pohledu dalšího vývoje projektu, by mohl být navržena a realizován systém úpravy hlasitosti tónů různých frekvencí, protože člověk nevnímá hlasitost všech tónů stejně. Komplikovanějším vylepšením, by mohla být implementace jiného principu realizace barvy tónů tak, aby generátor více odpovídal některému z reálných hudebních nástrojů. Toto vylepšení by mohlo být například realizováno na principech virtuálních nástrojů, kdy je vytvořen matematický model daného nástroje.

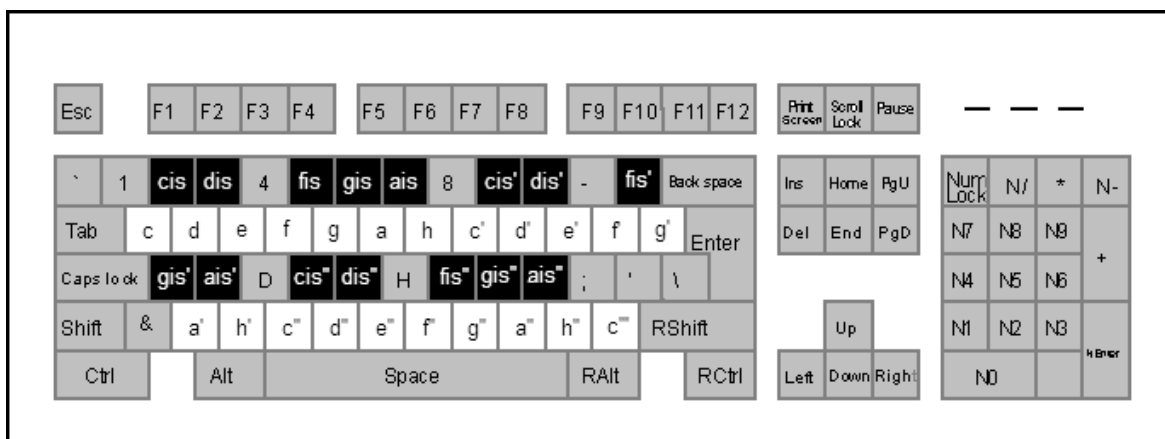
Literatura

- [1] VAŠÍČEK, Zdeněk. *Firmware/ Propojovací systém FITkitu* [online]. c2006 [cit. 2007-05-03].
Dostupné na URL: <<http://merlin.fit.vutbr.cz/FITkit/docs/firmware/spifitkit.html>>
- [2] VAŠÍČEK, Zdeněk. *Firmware/ LCD displej* [online]. c2006 [cit. 2007-05-03].
Dostupné na URL: <<http://merlin.fit.vutbr.cz/FITkit/docs/firmware/20060227lcd.html>>
- [3] VAŠÍČEK, Zdeněk. *Firmware/ Řadič PS/2* [online]. c2006 [cit. 2007-05-03].
Dostupné na URL: <<http://merlin.fit.vutbr.cz/FITkit/docs/firmware/ps2ctrl.html>>
- [4] MARKOVIČ, Ján. *Firmware/ Řadič přerušeni* [online]. c2007 [cit. 2007-05-03].
Dostupné na URL: <<http://merlin.fit.vutbr.cz/FITkit/docs/firmware/20060404int.html>>
- [5] Manuál mikrokontrolérů řady MSP430x1xx [cit. 2007-05-03].
Dostupné na URL: <<http://focus.ti.com/lit/ug/slau049f/slau049f.pdf>>
- [6] SIMANDL, Petr. *Klávesnice pro PC* [online]. 3.6.1999 [cit. 2007-05-03].
Dostupné na URL: <<http://www.hw.cz/Teorie-a-praxe/Dokumentace/ART1537-Klavesnice-pro-PC.html>>
- [7] ČERMÁKOVÁ, Magdalena. *Akustika pro studenty středních škol* [diplomová práce]. Brno, 2002. 90 s. Přírodovědecká fakulta. Masarykova univerzita v Brně. Katedra obecné fyziky.
Dokument dostupný na URL: <<http://www.physics.muni.cz/~cerm/dipla-magda.pdf>>
- [8] BERNAT, Petr. *Akustika* [online]. Poslední revize 27.2.2007 [cit. 2007-05-03].
Dostupné URL: <http://home.vsb.cz/petr.bernat/texty/varhany/anatomie/pistaly_akustika.htm>
- [9] GUŠTAR, Milan. *Audio - nahrávání "klasických" hudebních nástrojů* [online]. Poslední revize 24.8.2005 [cit. 2007-05-03]. Dostupné na URL: <<http://www.muzikus.cz/pro-muzikanty-serialy/Muzete-si-sami-natocit-CD~24~srpen~2005/>>

Seznam příloh

- [1] Rozložení tónů na klávesnici
- [2] Scan kódy kláves
- [3] CD se zdrojovými kódy a manuálem použití

Příloha 1 – Rozložení tónů na klávesnici



Příloha 2 – Scan kódy kláves

** Hlavní klávesy **

Klávesa	Vyslaný kód při stisku	Vyslaný kód při uvolnění
`	0x0E	0xF0 0x0E
1	0x16	0xF0 0x16
2	0x1E	0xF0 0x1E
3	0x26	0xF0 0x26
4	0x25	0xF0 0x25
5	0x2E	0xF0 0x2E
6	0x36	0xF0 0x36
7	0x3D	0xF0 0x3D
8	0x3E	0xF0 0x3E
9	0x46	0xF0 0x46
0	0x45	0xF0 0x45
-	0x4E	0xF0 0x4E
=	0x55	0xF0 0x55
Backspace	0x66	0xF0 0x66
Tabelátor	0x0D	0xF0 0x0D
Q	0x15	0xF0 0x15
W	0x1D	0xF0 0x1D
E	0x24	0xF0 0x24
R	0x2D	0xF0 0x2D
T	0x2C	0xF0 0x2C
Y	0x35	0xF0 0x35
U	0x3C	0xF0 0x3C
I	0x43	0xF0 0x43

O	0x44	0xF0 0x44
P	0x4D	0xF0 0x4D
[0x54	0xF0 0x54
]	0x5B	0xF0 0x5B
\	0x5D	0xF0 0x5D
Caps Lock	0x58	0xF0 0x58
A	0x1C	0xF0 0x1C
S	0x1B	0xF0 0x1B
D	0x23	0xF0 0x23
F	0x2B	0xF0 0x2B
G	0x34	0xF0 0x34
H	0x33	0xF0 0x33
J	0x3B	0xF0 0x3B
K	0x42	0xF0 0x42
L	0x4B	0xF0 0x4B
;	0x4C	0xF0 0x4C
'	0x52	0xF0 0x52
Enter	0x5A	0xF0 0x5A
Levý Shift	0x12	0xF0 0x12
Z	0x1A	0xF0 0x1A
X	0x22	0xF0 0x22
C	0x21	0xF0 0x21
V	0x2A	0xF0 0x2A
B	0x32	0xF0 0x32
N	0x31	0xF0 0x31
M	0x3A	0xF0 0x3A
,	0x41	0xF0 0x41
.	0x49	0xF0 0x49
/	0x4A	0xF0 0x4A
Pravý Shift	0x59	0xF0 0x59
Levý Ctrl	0x14	0xF0 0x14
Levý Alt	0x11	0xF0 0x11
Mezera	0x29	0xF0 0x29
Pravý Alt	0xE0 0x11	0xE0 0xF0 0x11
Pravý Ctrl	0xE0 0x14	0xE0 0xF0 0x14
ESC	0x76	0xF0 0x76

** Funkční klávesy **

Klávesa	Vyslaný kód při stisku	Vyslaný kód při uvolnění
F1	0x05	0xF0 0x05
F2	0x06	0xF0 0x06
F3	0x04	0xF0 0x04
F4	0x0C	0xF0 0x0C
F5	0x03	0xF0 0x03
F6	0x0B	0xF0 0x0B
F7	0x83	0xF0 0x83
F8	0x0A	0xF0 0x0A
F9	0x01	0xF0 0x01
F10	0x09	0xF0 0x09
F11	0x78	0xF0 0x78
F12	0x07	0xF0 0x07

** Numerická část **

Klávesa	Vyslaný kód při stisku	Vyslaný kód při uvolnění
NumLock	0x77	0xF0 0x77
-	0x7B	0xF0 0x7B
/	0xE0 0x4A	0xE0 0xF0 0x4A
.	0x71	0xF0 0x71
*	0x7C	0xF0 0x7C
+	0x79	0xF0 0x79
Enter	0xE0 0x5A	0xE0 0xF0 0x5A
0	0x70	0xF0 0x70
1	0x69	0xF0 0x69
2	0x72	0xF0 0x72
3	0x7A	0xF0 0x7A
4	0x6B	0xF0 0x6B
5	0x73	0xF0 0x73
6	0x74	0xF0 0x74
7	0x6C	0xF0 0x6C
8	0x75	0xF0 0x75
9	0x7D	0xF0 0x7D

** Pomocná část **

Klávesa	Vyslaný kód při stisku	Vyslaný kód při uvolnění
Ins	0xE0 0x70	0xE0 0xF0 0x70
Home	0xE0 0x6C	0xE0 0xF0 0x6C
Pg Up	0xE0 0x7D	0xE0 0xF0 0x7D
Del	0xE0 0x71	0xE0 0xF0 0x71
End	0xE0 0x69	0xE0 0xF0 0x69
Pg Dn	0xE0 0x7A	0xE0 0xF0 0x7A
Šipka nahoru	0xE0 0x75	0xE0 0xF0 0x75
Šipka doleva	0xE0 0x6B	0xE0 0xF0 0x6B
Šipka doprava	0xE0 0x74	0xE0 0xF0 0x74
Šipka dolů	0xE0 0x72	0xE0 0xF0 0x72
PrtSc	0xE0 0x12 0xE0 0x7C	0xE0 0xF0 0x7C 0xE0 0xF0
0x12		
Ctl-PrtSc	0xE0 0x7C	0xE0 0xF0 0x7C
Alt-PrtSc	0x84	0xE0 0x84
ScrLk	0x7E	0xF0 0x7E
Ctrl-Break	0xE0 0x7E 0xE0 0xF0 0x7E	nic