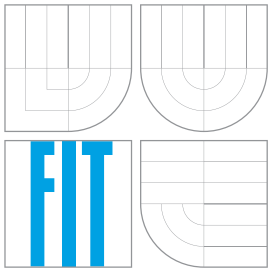


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

GENERÁTOR FUNKCÍ
POMOCÍ D/A PŘEVODNÍKU FITKITU
FUNCTION GENERATOR WITH FITKIT

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PAVEL BARTOŠ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. TOMÁŠ HERRMAN

BRNO 2007

Zadání bakalářské práce

Řešitel: **Bartoš Pavel**
Obor: Informační technologie
Téma: **Generátor funkcí pomocí D/A převodníku FITkitu**
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 funkční generátor (sinus, pila, trojúhelník, obdélník, ...) ovládaný PS/2 klávesnicí připojenou ke kitu a zobrazující frekvenci na LCD display.
4. Implementujte do generátoru možnost změny frekvence a střídý pomocí klávesnice.
5. Navrhněte a implementujte detektor signálů (sinus, pila, trojúhelník, obdélník, ...) pomocí A/D převodníku kitu.

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 00 Brno, 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: **Pavel Bartoš**
Id studenta: 84474
Bytem: Na Štěpnici 967, 562 01 Ústí nad Orlicí
Narozen: 28. 08. 1984, Ústí nad Orlicí
(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: Generátor funkcí pomocí D/A převodníku FITkitu
Vedoucí/školicel 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 práce se zabývá generováním a detekcí funkcí obdélníkového, sinusového, trojúhelníkového a pilovitého tvaru pomocí D/A převodníku FITkitu. Náplní je také popis komunikačního protokolu PS/2 klávesnice a popis ovládání LCD displeje.

Klíčová slova

FITkit, FPGA, Spartan 3, Xilinx, MCU, mikroprocesor, mikrokontrolér, MSP430F168, generátor, detektor, funkce, D/A, A/D, převodník, LCD, displej, české znaky, čeština, PS/2, klávesnice, scan kód

Abstract

This work deals with generation and detection of square, sine, triangle and saw-tooth waveforms using D/A converter of FITkit. Includes description of PS/2 communications protocol and description of LCD display.

Keywords

FITkit, FPGA, Spartan 3, Xilinx, MCU, microprocessor, microcontroller, MSP430F168, generator, detector, function, D/A, A/D, converter, LCD, display, Czech character, Czech, PS/2, keyboard, scan code

Citace

Pavel Bartoš: Generátor funkcí pomocí D/A převodníku FITkitu, bakalářská práce, Brno, FIT VUT v Brně, 2007

Generátor funkcí pomocí D/A převodníku FITkitu

Prohlášení

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

.....

Pavel Bartoš
12. května 2007

© Pavel Bartoš, 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

Úvod	3
1 FITkit	4
1.1 Mikrokontrolér	5
1.2 Programovatelné hradlové pole	5
2 PS/2 klávesnice	7
2.1 Řadič v FPGA	7
2.2 Řadič přerušení	7
2.3 Komunikační protokol	7
2.3.1 Přenos dat z periferního zařízení směrem k hostiteli	7
2.3.2 Přenos dat od hostitele směrem k perifernímu zařízení	8
2.4 Sady scan kódů	8
2.4.1 Sada 1	8
2.4.2 Sada 2	9
2.4.3 Sada 3	9
2.5 Příkazy klávesnice	9
2.5.1 Nastavení sady scan kódů	9
2.5.2 Reset klávesnice	9
2.5.3 Ovládání LED diod klávesnice	10
3 LCD displej	11
3.1 Řadič v FPGA	11
3.2 Menu	12
3.2.1 Nastavování parametrů signálu	13
3.3 České a speciální znaky	13
3.3.1 Postup nahrávání znaků do CGRAM	14
4 Generátor funkcí	15
4.1 D/A převodník	16
4.2 Časovač	16
4.3 Sinus	16
4.4 Obdélník	17
4.5 Trojúhelník	17
4.6 Píla	17

5	Detektor signálů	18
5.1	Vzorkování	18
5.2	A/D převodník a časovač	18
5.3	Postup zjištění parametrů signálu	18
5.3.1	Výpočet amplitudy	19
5.3.2	Detekce průchodů úrovní střední hodnoty	19
5.3.3	Výpočet frekvence	19
5.3.4	Výpočet střídy	20
5.4	Určení tvaru funkce	20
5.4.1	Obdélník	20
5.4.2	Pila	20
5.4.3	Trojúhelník	20
5.4.4	Sinus	21
5.5	Zobrazení na LCD	21
	Závěr	23
	Seznam použitých zdrojů	24
	Seznam použitých zkratk	25
	Seznam příloh	26
	A Seznam instrukcí LCD displeje	27
	B Tabulka SCAN kódů	28
	C Návod k použití	29

Úvod

Generátor funkcí je elektronické zařízení, které slouží ke generování periodického signálu určeného tvaru se zadanou amplitudou, frekvencí a střídou. Využívá se při opravách a návrhu elektronických systémů.

Hlavní součástí klasického generátoru je oscilátor, který vytváří periodický signál. Dnešní moderní generátory používají místo oscilátoru digitální signálový procesor (DSP) a D/A převodník. Mikroprocesor FITkitu sice není DSP, ale přesto ho lze využít pro generování nízkofrekvenčních signálů. Jen je třeba počítat s tím, že výsledný signál nebude přesně spojitý, ale zvláště při vyšších frekvencích bude docházet ke zkreslení signálu vlivem malého počtu vzorků na 1 periodu.

V první kapitole se věnuji platformě FITkit. Jsou v ní popsány součásti FITkitu, vlastnosti mikrokontroléru a podrobně popsáno programovatelné hradlové pole FPGA.

V další kapitole je popsána komunikace s PS/2 klávesnicí. Je zde zdůvodněn výběr varianty řadiče a důvod nepoužití řadiče přerušení. Dále je v kapitole popsána komunikace mezi klávesnicí a počítačem, problematika více sad scan kódů a příkazy klávesnice včetně ovládání LED diod na klávesnici.

V kapitole č. 3 se zabývám LCD displejem a jeho řadičem v FPGA. Je zde popsána i funkčnost menu, pomocí kterého se nastavují parametry generovaného signálu. Na závěr zde uvádím popis nahrávání českých znaků do LCD displeje.

Ve 4. kapitole se zabývám hlavním tématem mé práce, kterým je generátor funkcí. Je zde popsán výběr paměti, do které se budou ukládat vypočtené hodnoty funkce, nastavení D/A převodníku a časovače a způsob výpočtu hodnot jednotlivých funkcí.

V závěrečné 5. kapitole je popsán princip detekce funkcí pomocí A/D převodníku. Nejprve je popsán způsob vzorkování a ukládání hodnot. Dále se zabývám nastavením A/D převodníku a časovače, výpočtem amplitudy, frekvence a střídou. Nakonec je popsán způsob určení tvaru funkce a způsob výpisu výsledků na LCD displej.

Kapitola 1

FITkit

FITkit je platforma obsahující mikrokontrolér, programovatelné hradlové pole, 16-ti znakový jednořádkový LCD displej, maticovou klávesnici, paměť RAM, audio zesilovač, převodník USB – USART a konektory pro připojení periférií. FITkit se připojuje k počítači přes USB rozhraní, které zároveň zajišťuje přívod napájecího napětí 5 V. Po připojení externího zdroje napájecího napětí je možné s FITkitem pracovat i bez počítače.



Obrázek 1.1: FITkit

Program pro mikrokontrolér se tvoří v jazyce C a překládá se pomocí GNU překladače. Do flash paměti mikrokontroléru se přeložený program nahrává z počítače přes USB rozhraní.

Architektura pro FPGA se popisuje jazykem VHDL. Nahrání konfigurace do FPGA je možné jen prostřednictvím mikroprocesoru. Konfiguraci je možné nahrát buď do flash paměti, nebo přímo do FPGA. V prvním případě je konfigurace do FPGA nahrána z flash paměti při každém resetu mikroprocesoru. V druhém případě zůstane flash paměť nedotčena, ale je třeba počítat s tím, že při resetu procesoru, bude do FPGA nahrána konfigurace uložená v paměti flash.

1.1 Mikrokontrolér

FITkit obsahuje nízkopříkonový mikrokontrolér MSP430F168 od firmy Texas Instruments. Mikrokontrolér je vybaven:

- 16-ti bitovou RISC architekturou.
- 48 kB + 256 B flash paměti. Paměť je rozdělena na hlavní a informační sekci, které se liší způsobem adresování. Hlavní sekce má 96 segmentů o velikosti 512 bytů a informační sekce má 2 segmenty o velikosti 128 bytů.
- 2 kB paměti RAM.
- dvěma 16-ti bitovými časovači/čítači (označeny A a B). Oba mají několik jednotek záchytu hrany a výstupních komparátorů.
- analogovým komparátorem.
- 12-ti bitovým A/D převodníkem s 8 kanály a s vlastním generátorem referenčního napětí (1,5 V nebo 2,5 V).
- 12-ti bitovým D/A převodníkem s 2 nezávislými kanály. Velikost referenčního napětí je sdílena s A/D převodníkem.
- 2 sériovými rozhraními USART0 a USART1. První z nich se používá k propojení s počítačem přes převodník USB – USART. Druhý je vyveden do FPGA a na konektor JP9 a jeho využití závisí na programátorovi.

1.2 Programovatelné hradlové pole[7, 8]

Programovatelné hradlové pole umístěné na FITkitu je řady Spartan 3 od firmy Xilinx. Tento reprogramovatelný hardware lze neomezeně modifikovat.

Všechna FPGA firmy Xilinx se konfiguruje pomocí statické paměti RAM. To znamená, že po připojení napájecího napětí je nutné vždy znovu nahrát konfiguraci do FPGA. Výhodou tohoto řešení je téměř nekonečná reprogramovatelnost FPGA a také rychlost. Konfigurační propojky pracující na principu paměti RAM jsou totiž rychlejší než přepínače založené na principu EEPROM.

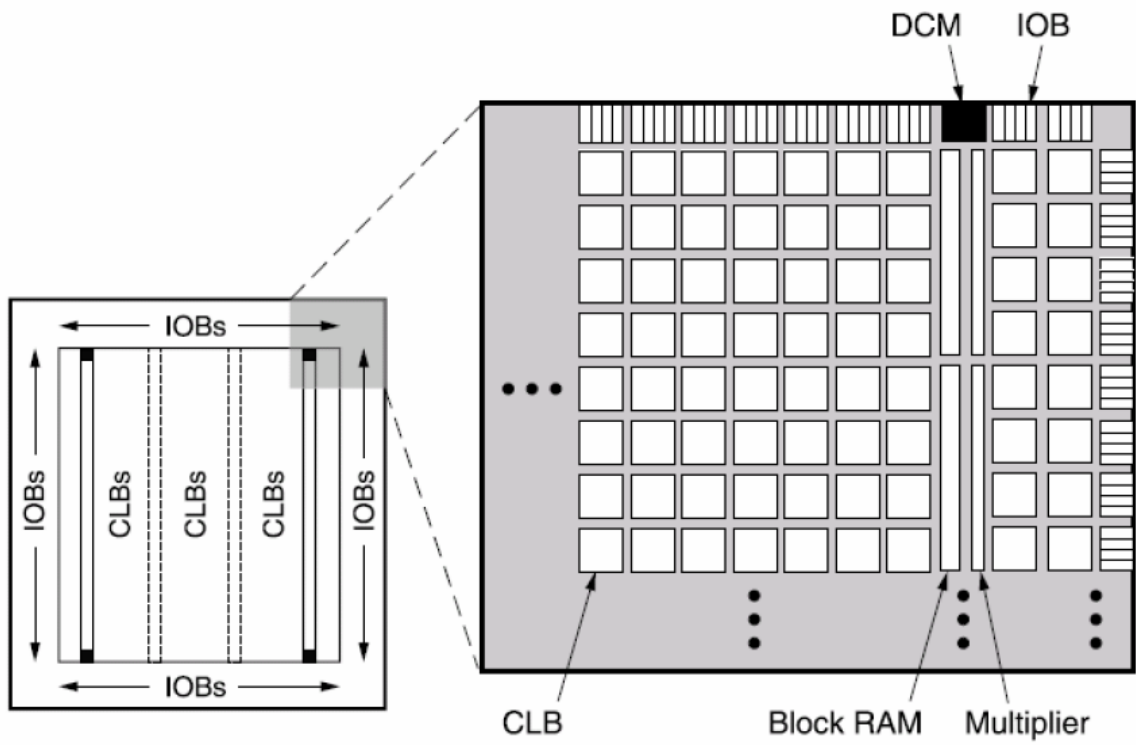
Interní struktura obvodu Spartan 3 je znázorněna na obrázku 1.2.

Obvod obsahuje vstupně-výstupní buňky (IOB), konfigurovatelné logické bloky (CLB), blokovou paměť RAM (Block RAM), násobičky a obvody pro řízení hodinového signálu (DCM).

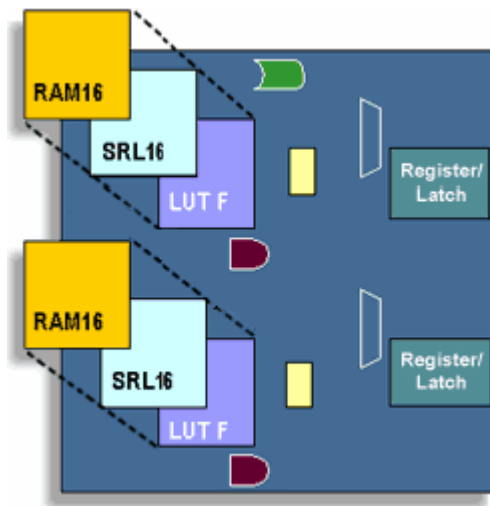
Každý pin (buňka IOB) FPGA může být konfigurován jako vstup, výstup nebo obojí. Dva sousední piny lze zapojit jako diferenciální pár. Podporováno je velké množství standardů (LVTTTL, LVCMOS, HSTTL, PCI-X, AGP a další).

Každý konfigurovatelný logický blok (CLB) obsahuje čtyři menší logické elementy (slice) a 2 nezávislé *carry* řetězce pro konstrukci rychlých sčítaček.

Slice (viz obr. 1.3) obsahuje 2 funkční generátory, které mohou být použity jako logické hradlo, nebo jako paměť o velikosti 16x1 bitů, a nebo jako posuvný registr. Slice ještě obsahuje 2 registry, multiplexory a *carry* logiku. Pomocí multiplexorů lze například vytvořit paměť o libovolné šířce dat.



Obrázek 1.2: Vnitřní struktura FPGA Spartan 3



Obrázek 1.3: Vnitřní struktura slice

Kapitola 2

PS/2 klávesnice

2.1 Řadič v FPGA

Komunikaci mezi klávesnicí a FITkitem obstarává řadič naprogramovaný v FPGA, který převádí sériovou komunikaci PS/2 protokolu na paralelní a naopak. Použil jsem již hotový řadič z SVN FITkitu. Existují 2 varianty tohoto řadiče. Varianta *half*, která umožňuje přenos pouze z klávesnice do FITkitu, a varianta *full*, pomocí které lze komunikovat obousměrně. Protože potřebuji pracovat s LED diodami klávesnice a nastavit určitou sadu SCAN kódů, musel jsem použít variantu *full*. Podrobný popis řadiče viz [10].

2.2 Řadič přerušení

Řadič klávesnice generuje při příjmu dat z klávesnice signál, který určuje v daném čase platnost dat na sběrnici. Tento signál lze využít pro generování žádosti o přerušení. Pokoušel jsem se použít řadič přerušení z SVN FITkitu[6], ale nepodařilo se mi ho zapojit. Na pinu X(0) se nevyskytoval žádný signál od řadiče přerušení, a to ani při použití ukázkové aplikace.

Protože externí přerušení potřebuji pouze od klávesnice, tak jsem na pin X(0) připojil signál DATA_VLD z řadiče klávesnice a řadič přerušení jsem odstranil. Po propojení pinu X(0) a mikrokontroléru vodičem funguje přerušení bez problémů.

2.3 Komunikační protokol

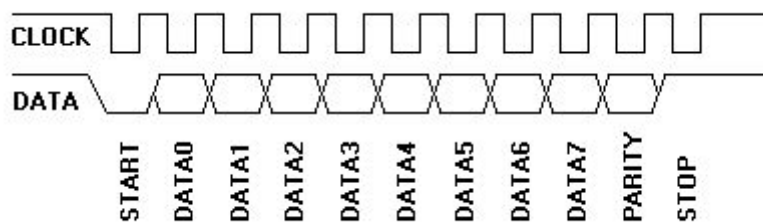
Komunikace je synchronní osmibitová se *start* bitem (log. 0), *paritním* bitem (lichá parita) a *1 stop* bitem (log. 1). Datové bity se posílají od nejméně významného po nejvíce významný bit.

Komunikace na PS/2 rozhraní je řízena příslušnou periferií, v tomto případě klávesnicí, protože se předpokládá, že periferie bude data posílat častěji.

2.3.1 Přenos dat z periferního zařízení směrem k hostiteli

Zapojení datového i hodinového signálu je s otevřeným kolektorem. Protože je mezi každý z těchto vodičů a napětí 5 V vložen pull-up rezistor, je klidový stav sběrnice v log. 1. Když chce periferní zařízení poslat data, nejdříve zkontroluje, zda je sběrnice v klidovém stavu. Pokud tomu tak není, znamená to, že hostitel chce poslat data do periferního zařízení a periferie musí s odesláním dat počkat do doby, až hostitel dokončí přenos.

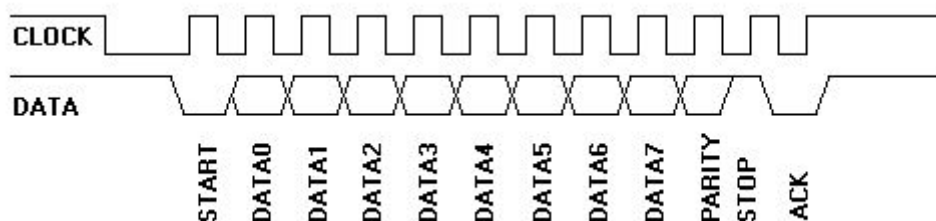
Poté periferie začne generovat hodinový signál a vkládat bity na datový vodič, přičemž jsou data platná, pokud je hodinový signál v log. 0, a ke změně dat dochází v log. 1 hodinového signálu. Viz obr. 2.1.



Obrázek 2.1: Schéma komunikace z periferie k hostiteli

2.3.2 Přenos dat od hostitele směrem k perifernímu zařízení

Pokud hostitel potřebuje poslat data do periferie, stáhne hodinový signál do log. 0, aby periferie nemohla začít vysílat data. Poté stáhne do log. 0 i datový vodič, čímž vytvoří *start* bit, a uvolní hodinový vodič. Periferie začne generovat hodinový signál a hostitel začne vkládat data na datový vodič. Na rozdíl od komunikace z periferie směrem k hostiteli, jsou data platná při log. 1 hodinového signálu a mění se při log. 0 hodinového signálu. Viz obr. 2.2.



Obrázek 2.2: Schéma komunikace od hostitele k perifernímu zařízení

Podrobněji viz [10, 3]

2.4 Sady scan kódů

Při stisku nebo uvolnění klávesy klávesnice generuje scan kód, který jednoznačně identifikuje klávesu a její stav (stisknuta, uvolněna). Existují 3 způsoby tvorby tohoto kódu, tzv. *sady* scan kódů[1].

2.4.1 Sada 1

Tato sada pochází z klávesnic počítačů XT. Při stisku klávesy se odešle její kód (tzv. *make* kód) podle tabulky B.1 na straně 28. U některých kláves (na některých klávesnicích

označeny šedou barvou) tvoří make kód 2 byty, přičemž první má hodnotu $e0_h$. Např. klávesa Enter má make kód $1c_h$ a klávesa Enter na numerické části klávesnice má make kód $e0\ 1c_h$.

Při uvolnění klávesy je generován tzv. *break* kód, který se od make kódu liší jen přičtením hodnoty 80_h . V případě 2 bytových kódů se nejdříve odešle hodnota $e0_h$ a poté teprve kód klávesy s přičtenou hodnotou 80_h . Pro klávesu Enter je tedy break kód $9c_h$ a pro Enter na numerické části klávesnice $e0\ 9c_h$.

Výjimku tvoří klávesa *pause*, která generuje make kód $e1\ 1d\ 45\ e1\ 9d\ c5_h$. Break kód při uvolnění negeneruje.

2.4.2 Sada 2

Později byla vyvinuta nová sada scan kódů, ale způsobovala problémy u starých programů. Proto se výstup z klávesnice konvertoval pomocí mikroprocesoru 8042 zpět na sadu 1. Toto řešení se používá dodnes.

Sada 2 se od sady 1 liší v jiném číslování kláves a tím i v jiných make kódech. Liší se také v break kódech, kdy místo přičítání 80_h se před kódem klávesy odešle byte $f0_h$. Proto může být v této sadě make kód větší než 127. V případě 2 bytových kódů se příznak uvolnění $f0_h$ odesílá až po prvním bytu ($e0_h$). Make kód klávesy Enter je $5a_h$, break kód $f0\ 5a_h$. Make kód numerického Enteru je $e0\ 5a_h$ a break kód $e0\ f0\ 5a_h$.

Výjimku opět tvoří klávesa *pause*, která má make kód $e1\ 14\ 77\ e1\ f0\ 14\ f0\ 77_h$ a break kód nemá.

2.4.3 Sada 3

Tato sada byla vytvořena pro PS/2 klávesnice, ale téměř nikdo ji nepoužívá. Některé klávesnice ji ani nepodporují.

2.5 Příkazy klávesnice

Do klávesnice lze odesílat různé příkazy. Například lze požádat o opětovné poslání posledního bytu, nastavit sadu scan kódů, rozsvěcet LED diody, nastavit opakování kláves při dlouhém stisku nebo klávesnici resetovat. Kompletní seznam příkazů lze nalézt v [2].

2.5.1 Nastavení sady scan kódů

Sadu scan kódů lze u většiny dnešních klávesnic nastavit. Provede se to odesláním bytu $f0_h$, na který klávesnice odpoví bytem *ack* (fa_h), a bytu 1_h , 2_h nebo 3_h dle požadované sady, na který klávesnice opět odpoví bytem *ack*.

Aktuálně nastavenou sadu lze zjistit odesláním bytu $f0_h$ a bytu 0_h . Klávesnice na oba byty odpoví bytem *ack* a poté pošle byte 43_h , 41_h nebo $3f_h$, což odpovídá sadám 1, 2 nebo 3.

Pro svoji práci jsem zvolil sadu 1, protože je jednodušší ji implementovat.

2.5.2 Reset klávesnice

Klávesnice provede reset po připojení napájecího napětí nebo po přijetí příkazu k provedení resetu. Po resetu se provede otestování klávesnice, tzv. BAT (Basic Assurance Test). Pokud

je tento test úspěšný, klávesnice odešle byte \mathbf{aa}_h (BAT successful), v případě neúspěchu odešle \mathbf{fc}_h (error).

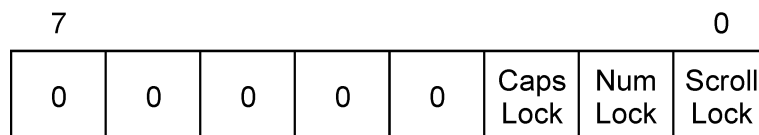
Po resetu jsou nastaveny tyto hodnoty:

- Nastavena sada scan kódů č. 2.
- Prodleva před opakováním klávesy nastavena na 500 ms.
- Rychlost opakování klávesy 10,9 úhozů za vteřinu.

2.5.3 Ovládání LED diod klávesnice

LED diody na klávesnici lze libovolně rozsvěcet a zhasínat. Ovšem je třeba mít na paměti, že při svítící LED Num Lock má na některých klávesnicích např. klávesa Home jiný scan kód než když LED nesvítí. (Při svítící LED a stisku klávesy Home se odešle kód stisku klávesy Shift, poté kód stisku a uvolnění klávesy 7 na numerické klávesnici a nakonec kód uvolnění klávesy Shift. Toto jsem pozoroval u 3 klávesnic, které jsem k FITkitu připojil, ale podle [1] má mít klávesa Home svůj vlastní scan kód.)

Ovládání LED diod se provádí odesláním bytu \mathbf{ed}_h , následovaného bytem obsahujícím požadované stavy LED diod, viz obr. 2.3. Zapsání 1 na pozici požadované LED znamená rozsvícení, 0 znamená zhasnutí LED.



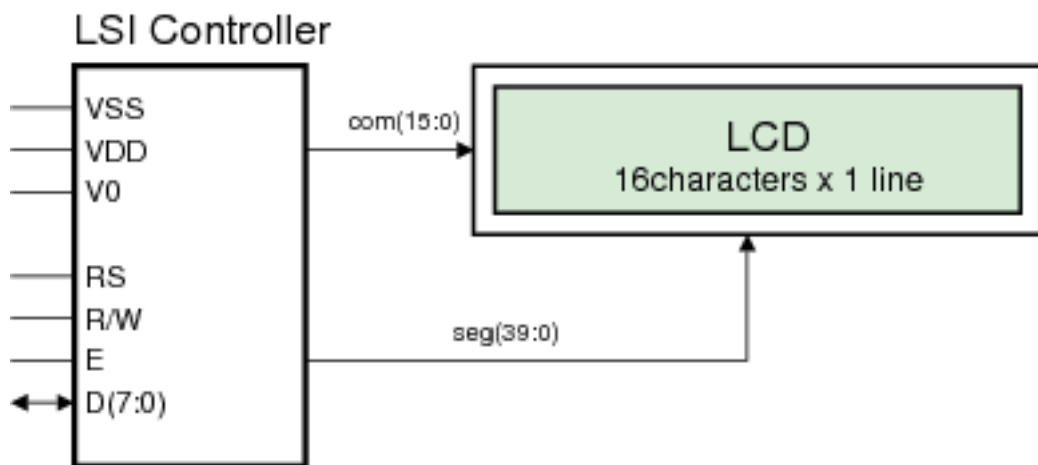
Obrázek 2.3: Struktura bytu pro nastavení stavů LED na klávesnici

Kapitola 3

LCD displej

Součástí FITkitu je 16-ti znakový jednořádkový LCD displej připojený k FPGA. Displej se skládá z řadiče (generátoru znaků) a z vlastního LCD prvku (viz obr. 3.1). V řadiči jsou 3 paměti:

- ROM – Obsahuje bodové předlohy znaků.
- CGRAM – Lze do ní nahrát 8 vlastních znaků, které jsou poté dostupné pod kódy 0–7 a ještě jednou pod kódy 8–15. Více v sekci 3.3.
- DDRAM – Uchovává aktuální stav displeje (zobrazené znaky).



Obrázek 3.1: Řadič LCD displeje

3.1 Řadič v FPGA

Převod dat z vnitřní sběrnice FITkitu na sběrnici LCD displeje provádí řadič umístěný v FPGA. Použil jsem LCD řadič z SVN FITkitu. Tento řadič umožňuje pouze zápis do paměti v řadiči LCD displeje. LCD displej sice umožňuje číst *Busy flag* a data z DDRAM paměti, ale tato data nejsou potřeba.

Komunikace s displejem může být po 8-bitové sběrnici (DB0–DB7) nebo 4-bitové sběrnici (DB4–DB7). 4-bitovou sběrnici je vhodné použít při nedostatku vývodů na procesoru. Komunikace ale bude dvakrát pomalejší, protože se data posílají ve 2 cyklech. LCD displej je k FPGA připojen pomocí 8-bitové sběrnice.

Mimo tuto datovou sběrnici existují ještě 3 řídicí signály, kterými se LCD displej ovládá. Jsou to signály:

- RS (Register select) – Tímto signálem se rozlišují data a instrukce. Pokud je tento signál v log. 0, na datové sběrnici je instrukce, v opačném případě jsou to data.
- R/W (Read/Write) – Určuje směr přenosu. Pokud je v log. 0, znamená to, že zapisujeme data do LCD displeje, v případě log. 1 data čteme. Řadič v FPGA je schopen pouze zapisovat, takže tento signál je trvale v log. 0.
- E (Enable) – Tímto signálem se potvrzuje platnost dat na datové sběrnici. Data musí být platná vždy při sestupné hraně tohoto signálu.

Displej se ovládá pomocí instrukcí (tab. A na straně 27), které se posílají po datové sběrnici. Data a instrukce se rozlišují signálem RS.

Po zapnutí displeje je nutné provést základní inicializaci displeje, která spočívá v nastavení počtu bitů datové komunikace, směru psaní znaků, posunu řádku nebo pozice kurzoru a blikání kurzoru. Po této inicializaci je displej připraven na komunikaci.

3.2 Menu

Pro zjednodušení nastavování parametrů generovaného signálu, jsem naprogramoval jednoduché menu, ve kterém se lze pohybovat kurzorovými klávesami připojené PS/2 klávesnice. Do menu lze vstoupit ze základního režimu stiskem klávesy Enter.

Struktura menu:

1. Zapnutí / vypnutí generování funkce
2. Průběh
 - 2.1 Sinus
 - 2.2 Obdélník
 - 2.3 Trojúhelník
 - 2.4 Pila
 - 2.5 Inverzní pila
3. Frekvence
4. Amplituda
5. Střída
6. Detektor

Aktuální pozice v menu je vyznačena číslem a názvem položky. Směry pohybu, které jsou v dané chvíli přípustné, jsou vyznačeny šipkami na okrajích displeje, viz obr. 3.2.



Obrázek 3.2: Ukázka možných směrů pohybu

3.2.1 Nastavování parametrů signálu

Průběh funkce

Pro nastavení druhu funkce (průběhu) generátoru je třeba jen vybrat v menu příslušný průběh a stisknout klávesu Enter. Na cca 0,5s se zobrazí nápis OK a volba je uložena.

Parametry funkce

Při nastavování frekvence, amplitudy a střídy je třeba zadat číselnou hodnotu. Po vybrání požadovaného parametru funkce a stisku klávesy Enter se na LCD zobrazí výzva k zadání číselné hodnoty (viz obr. 3.3). Tu lze zadat buď přímo klávesami 0–9, nebo kurzorovými klávesami (šipkami) nahoru/dolů. Mezi jednotlivými číslicemi se lze pohybovat pomocí kláves vlevo/vpravo. Protože program průběžně vyhodnocuje zadané číslice, není možné zadat číslo mimo povolený rozsah. Po stisku klávesy Enter se zobrazí nápis OK a hodnota je uložena.

Nastavená střída signálu se použije pouze u funkcí obdélník a trojúhelník.



Obrázek 3.3: Ukázka zadávání hodnot

3.3 České a speciální znaky

Aby popisky v menu mohly být v češtině, je třeba české znaky vytvořit a nahrát do CGRAM displeje. Bohužel českých znaků s diakritikou je více, než se jich do CGRAM vejde. Navíc ještě používám speciální znaky znázorňující průběh funkce. Proto program musí zajistit, aby v CGRAM byly vždy jen právě zobrazované české a speciální znaky. Toto zajišťuje pole 8 bytů (kapacita paměti CGRAM), pomocí kterého se převádí ordinální hodnota českého znaku na adresu znaku v paměti CGRAM. Toto pole je při každém smazání displeje vynulováno.

Pokud se znak, který chceme zobrazit na displej, zatím nenachází v tomto poli, je třeba tento znak nahrát do CGRAM a učinit o tom záznam do tohoto pole, aby při dalším výskytu

tohoto znaku tento se již nemusel znovu nahrávat do CGRAM.

3.3.1 Postup nahrávání znaků do CGRAM

Velikost znaků, které zapisujeme do paměti, je 8x8 pixelů, ale zobrazí se jen 5x8 pixelů. Data se zapisují po řádcích shora dolů. Viz obr. 3.4. Před každým zápisem je třeba nastavit adresu, která se skládá z adresy znaku a čísla řádku viz tab. 3.1.

7	6	5	4	3	2	1	0
0	1	adr. znaku			č. řádku		

Tabulka 3.1: Nastavení adresy CGRAM

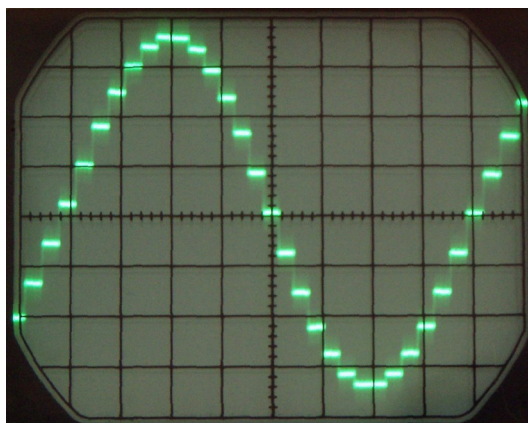
			■		0x02
		■			0x04
■				■	0x11
				■	0x11
	■	■	■	■	0x0f
				■	0x01
	■	■	■		0x0e
					0x00

Obrázek 3.4: Příklad českého znaku “ý” včetně kódu pro každý řádek

Kapitola 4

Generátor funkcí

Hlavním tématem mé práce je generátor funkcí, což je zařízení, které na svém výstupu generuje periodický signál se zadanou amplitudou, frekvencí a střídou. V případě použití mikroprocesoru a D/A převodníku, nebude výsledný signál spojitý, ale bude generován po skocích tak, jak budou posílány hodnoty výstupního napětí do D/A převodníku (viz obr. 4.1).



Obrázek 4.1: Výstup D/A převodníku zobrazený na osciloskopu

Počítání hodnot za běhu generátoru by bylo příliš výpočetně náročné. Proto je třeba před samotným generováním průběhu vypočítat hodnoty jedné periody funkce a ty uložit do paměti.

Nejprve jsem zamýšlel použít pro uchování těchto hodnot externí RAM připojenou k FPGA. Při vlastní realizaci jsem zjistil, že bylo potřeba použít 16-ti bitového adresování na SPI rozhraní MCU–FPGA, což mělo za důsledek zpomalení komunikace na této sběrnici. I při nastavení maximální frekvence SPI rozhraní a použití assembleru v kritické části kódu, bylo možné generovat jen cca 5000 vzorků za vteřinu. Další omezení spočívalo v tom, že při přenosu dat k dalším periferiím připojeným k FPGA (LCD, klávesnice) docházelo k výpadkům generování signálu, protože nebylo možné včas přenést data z RAM do MCU.

Proto jsem se rozhodl použít paměť RAM integrovanou v čipu MCU. Tím jsem zvýšil rychlost generování vzorků na více než 20 kHz. Tato paměť má velikost jen 2 kB a používá se pro uložení proměnných a pro zásobník. Překladač mi dovolil alokovat pro potřebu uložení hodnot funkce maximálně 1 kB této paměti (512 16-ti bitových hodnot), což je

při maximálně 20 tisících vzorcích za vteřinu dostatečná hodnota i pro generování funkcí s nízkou frekvencí. Navíc se přenos hodnot z paměti do D/A převodníku zjednodušil jen na 1 instrukci a není třeba programovat v assembleru.

4.1 D/A převodník

D/A převodník je 12-ti bitový a má dva nezávislé kanály. Výsledný signál je generován oběma kanály zároveň.

Referenční napětí se používá stejné jako u A/D převodníku. Lze vybrat napětí 1,5 V nebo 2,5 V. Z důvodu lepší měřitelnosti výstupního signálu, jsem zvolil vyšší napětí, tedy 2,5 V. D/A převodník by měl být schopen generovat napětí třikrát větší, ale nikdy se mi toho nepodařilo dosáhnout.

Rozsah 0 – 4095 bitů odpovídá rozsahu 0 – 2,5 V. Na 1 V výstupního napětí tedy připadá $4096 \div 2,5 = 1638$ bitů. Tato hodnota je potřebná pro přepočítání amplitudy zadané ve voltech na velikost výstupního napětí zadávaného D/A převodníku. Výstupní zesilovač FITkitu má napěťové zesílení $A_U = 1$, takže velikost výstupního napětí neovlivní.

4.2 Časovač

D/A převod je spouštěn vzestupnou hranou výstupní jednotky č. 1 časovače/čítače A. Tento časovač zároveň v polovině periody vyvolává žádost o přerušení, během kterého se naplní datové registry D/A převodníku novými hodnotami.

Nastavení časovače je závislé na požadované frekvenci signálu a počtu vzorků na 1 periodu. Výsledná frekvence přerušení časovače je dána frekvencí oscilátoru připojeného k jednotce časovače, předděličkou a hodnotou, při které se časovač vynuluje. Časovač může čítat hodinové signály SMCLK (7,3728 MHz) nebo ACLK (32,768 kHz). Ostatní hodnoty se vypočítají pro oba tyto hodinové signály a použije se varianta, která bude mít menší odchylku od požadované frekvence. V případě stejné odchylky se použije hodinový signál ACLK, protože je jeho frekvence přesnější.

4.3 Sinus

Vypočítávání hodnot funkce sinus pomocí Taylorova rozvoje by bylo příliš náročné, jednodušší variantou je mít uloženy hodnoty funkce v několika bodech, jejichž počet závisí na požadované přesnosti, a hodnoty mezi těmito body vypočítat lineární interpolací. Stačí mít uloženy jen hodnoty v 1. kvadrantu funkce, protože hodnoty v ostatních kvadrantech lze snadno odvodit. Do paměti na uložení vzorků se vejde 512 vzorků, proto stačí mít uloženo 128 vzorků a nemusíme ani interpolovat.

Celkový počet vzorků zjistíme tak, že maximální frekvenci převodu vzorků v D/A převodníku (20 kHz) vydělíme požadovanou frekvencí signálu. Pokud je požadovaná frekvence menší než 40 Hz, je vypočtený počet vzorků větší než je velikost paměti pro jejich uložení (512 vzorků), počet vzorků se proto nastaví na velikost této paměti. Tak je zaručena maximální možná kvalita výstupního signálu.

Aby bylo možné převést hodnoty funkce sinus z tabulky uložené v paměti programu na hodnoty s požadovanou amplitudou, je třeba určit koeficient, kterým když vynásobíme hodnoty z tabulky, dostaneme funkci sinus s požadovanou amplitudou. Tento koeficient definuji jako poměr mezi požadovanou amplitudou a maximální hodnotou z tabulky. Protože

ale mikroprocesor neumí pracovat s čísly v plovoucí desetinné čárce, nelze tento koeficient vypočítat. Proto se tabulková hodnota musí nejdříve vynásobit čitatelem a následně vydělit jmenovatelem tohoto poměru, přičemž se musí počítat s 32 bitovými proměnnými. Pokud vzorek náleží do 3. nebo 4. kvadrantu, je ještě vynásoben -1 .

V tabulce hodnot funkce sinus je uloženo 128 hodnot z intervalu $\langle 0; \frac{\pi}{2} \rangle$. Všechny se použijí jen v případě, že celkový počet prvků je roven 512. V opačném případě se z tabulky vyberou rovnoměrně jen některé prvky. Pokud se vzorek nachází v 2. nebo 4. kvadrantu, musí se index upravit.

Nyní se již vzorky uloží do paměti, z které se budou po spuštění časovače kopírovat do D/A převodníku.

4.4 Obdélník

Abychom zjistili počet hodnot (dále značeno H) v log. 1 a log. 0, nejprve spočítáme největší společný dělitel střídy a hodnoty $100 - \text{střída}$. K výpočtu je použit Euklidův algoritmus[9]. Např. pokud střída je 50 %, největší společný dělitel je 50 a v log. 1 i log. 0 tedy bude jen jedna hodnota. Při střídě 30 % budou v log. 1 tři hodnoty a v log. 0 sedm hodnot apod.

V případě, že počet hodnot a frekvence bude tak vysoká, že nebude platit vztah $H \cdot f \leq f_{max}$, kde H je celkový počet hodnot, f je frekvence a f_{max} je maximální frekvence 20 kHz, musí se celkový počet hodnot H snížit tak, aby tento vztah platil a přitom zůstal alespoň přibližně zachován poměr střídy.

Nyní již stačí jen amplitudu zadanou ve voltech přepočítat na 11ti bitovou kladnou a zápornou amplitudu a uložit je na příslušná místa do paměti.

4.5 Trojúhelník

Zjištění celkového počtu vzorků je stejné jako u sinusového průběhu. Tento celkový počet je následně rozdělen podle střídy na část, ve které bude signál stoupat, a na část, ve které bude signál klesat. Poté se již jen generují a ukládají lineárně rostoucí či klesající vzorky v rozmezí od záporné amplitudy do kladné amplitudy.

4.6 Pila

Signál tvaru pily je generován stejně jako signál tvaru trojúhelníku. Pouze je předtím nastavena střída na 0 % nebo 100 %, podle požadované orientace náběhové hrany.

Kapitola 5

Detektor signálů

5.1 Vzorkování

Určení parametrů detekovaného signálu nelze provádět v reálném čase, protože se při analýze signálu potřebujeme pohybovat v čase. Je tedy třeba část vstupního signálu navzorkovat, uložit do paměti a následně zpracovat. Vzorkovací frekvenci jsem zvolil 20 kHz, protože velikost paměti je omezená a je tedy třeba zvolit kompromis mezi detekcí signálů vyšší frekvence a detekcí signálů nízké frekvence tak, aby byla v paměti uložena alespoň jedna celá perioda. K ukládání hodnot se přímo nabízí použít prostoru v paměti MCU jinak používaného k ukládání hodnot generovaného signálu. Do tohoto prostoru lze uložit maximálně 512 vzorků. Minimální detekovatelnou frekvenci lze spočítat podle rovnice $f_{min} = 2 \cdot \frac{f_{vzork}}{d}$ (kde d je počet vzorků (512), $f_{vzork} = 20$ kHz) a je rovna 78,125 Hz. Násobení 2 se v rovnici vyskytuje z důvodu, že nelze zaručit, že bude začátek periody na začátku pole vzorků.

5.2 A/D převodník a časovač

A/D převod je spouštěn vzestupnou hranou na výstupní jednotce časovače A, do kterého je přiveden vstupní signál o frekvenci $f_{osc} = 7,3728$ MHz. Protože je potřeba vzorkovací frekvence 20 kHz, časovač bude čítat od nuly do hodnoty 369. Tato hodnota je přibližným výsledkem výrazu:

$$\frac{f_{osc}}{f_{vzork}} = \frac{7,3728 \cdot 10^6}{20 \cdot 10^3} = 368,64 \doteq 369$$

Jakmile hodnota čítače dosáhne hodnoty o 1 menší než tato hodnota, generuje se na výstupní jednotce vzestupná hrana, která spustí A/D převod. Při hodnotě 369 je čítač automaticky vynulován a výstupní jednotka je resetována.

Ihned po skončení A/D převodu je generována žádost o přerušení, která musí být co nejdříve obsloužena. Jinak by mohlo dojít k přepsání hodnoty v registru A/D převodníku a to by mělo za důsledek ztrátu kontinuity vzorků. Tato událost generuje vlastní žádost o přerušení a pokud k tomuto během vzorkování dojde, musí se signál začít vzorkovat znovu.

5.3 Postup zjištění parametrů signálu

Před testováním tvaru signálu se nejprve zjistí obecné informace o signálu. Jedním průchodem polem navzorkovaných hodnot se zjistí maximální a minimální hodnota. Z nich lze

snadno spočítat střední hodnotu signálu.

V dalším průchodu se spočítá průměrná hodnota vzorků větších a menších než střední hodnota. Tyto hodnoty se dále použijí jen k detekci obdélníkového signálu, kde odstraní případný náhodný výkyv napětí u maxima či minima.

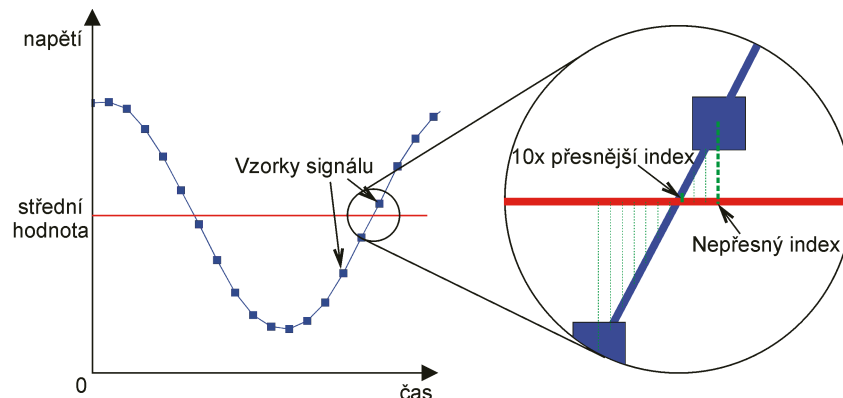
5.3.1 Výpočet amplitudy

Velikost amplitudy signálu se počítá jako rozdíl maximální nalezené hodnoty a střední hodnoty, v případě obdélníkového signálu se místo maximální hodnoty použije průměrná hodnota vzorků ve stavu log. 1. Následně je tento rozdíl převeden na napětí.

5.3.2 Detekce průchodů úrovní střední hodnoty

K určení začátku a konce periody signálu je třeba zjistit polohy průchodů signálu střední hodnotou (dále v textu jsou tyto průchody nazývány *hranami*, i když se o hrany jedná pouze v případě obdélníkového signálu). Tyto údaje později mj. umožní výpočet frekvence a střídý. Hrany signálu se nejjednodušeji zjistí porovnáváním hodnot po sobě jdoucích vzorků se střední hodnotou signálu. Jakmile je vzorek s indexem i menší než střední hodnota a zároveň vzorek s indexem $i + 1$ větší než střední hodnota, zaznamená se tento index i do seznamu vzestupných hran. Stejným způsobem se zaznamenají i indexy sestupných hran.

Při vyšších frekvencích vzorkovaného signálu připadá na jednu periodu méně vzorků, než stačí k dostatečně přesnému určení polohy hrany. Pomocí lineární interpolace je poloha hrany uložena desetkrát přesněji. Viz obr. 5.1.



Obrázek 5.1: Znázornění indexu průchodu signálu střední hodnotou před a po zpřesnění pomocí lineární interpolace

5.3.3 Výpočet frekvence

Frekvenci navzorkovaného signálu lze určit z detekovaných hran podle vzorce:

$$f = \frac{f_{vzork}}{\frac{h_n - h_1}{n}}$$

kde h jsou polohy hran a n je počet hran. Aby se dosáhlo vyšší přesnosti, spočítá se frekvence s použitím vzestupných a sestupných hran zvlášť a výsledek se určí pomocí aritmetického průměru.

5.3.4 Výpočet střídý

Střída znamená poměr časů, ve kterých je periodický signál nad a pod úrovní střední hodnoty signálu. Pokud je střída uvedena ve tvaru 1 : 1, znamená to, že kladná i záporná část průběhu trvají stejnou dobu. Střídu lze uvádět i v procentech. V tomto případě údaj znamená dobu trvání kladné úrovně vůči periodě signálu. V mé práci střídu vždy uvádím v procentech.

Obdélník

Výpočet střídý obdélníkového signálu je v mé práci prováděn pomocí již dříve detekovaných hran. Jde o poměr mezi počtem vzorků mezi vzestupnou a sestupnou hranou a počtem vzorků mezi 2 vzestupnými hranami.

Trojúhelník

U trojúhelníkového průběhu nejde o střídu v pravém slova smyslu, ale jde o poměr mezi částí signálu, kde úroveň stoupá, a částí, kde klesá. Výpočet se provádí obdobně jako u obdélníkového signálu s tím rozdílem, že se nepoužívají hrany, ale v každé periodě se nalezne lokální maximum a minimum a spočítá se poměr mezi dobou od maxima do minima a dobou celé periody.

5.4 Určení tvaru funkce

Určování tvaru funkce probíhá postupně podle náročnosti výpočtů v tomto pořadí: obdélník, pila, trojúhelník a sinus.

5.4.1 Obdélník

Rozhodnutí, zda je navzorkovaný signál obdélníkový, se provádí tak, že se spočítá průměrná odchylka vzorků od již dříve spočítaných mezí signálu (viz kap. 5.3, odst. 2). Pokud je průměrná odchylka menší než stanovená konstanta, je signál obdélníkový.

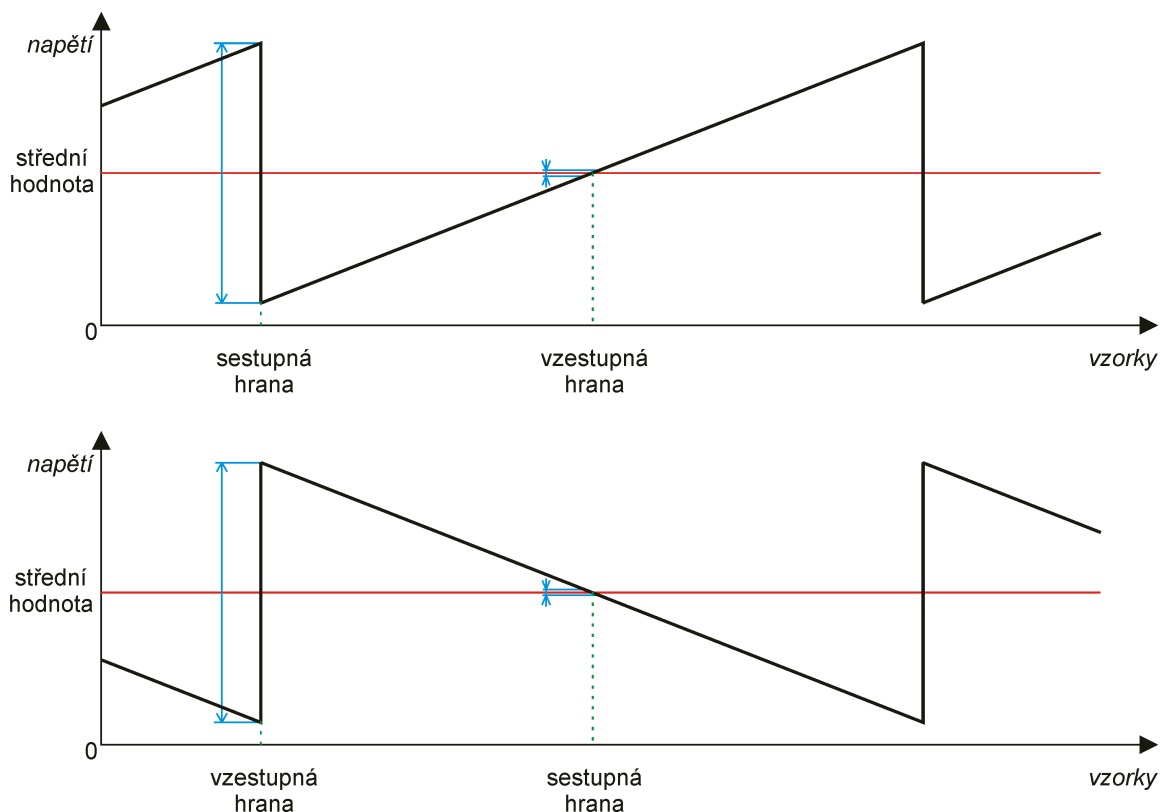
5.4.2 Pila

Při rozpoznávání pilovitého průběhu je třeba nejdříve zjistit orientaci náběhové hrany. To se nejjednodušeji zjistí porovnáním rozdílu hodnot v okolí průchodu signálu jeho střední hodnotou, viz obr. 5.2.

Poté již stačí pouze zkontrolovat linearitu mezi minimem a maximem. Stejně jako u obdélníkového signálu se počítá průměrná odchylka od teoretické přímky spojující toto minimum a maximum. Pokud je průměrná odchylka menší než stanovená mez, signál má tvar pily.

5.4.3 Trojúhelník

Trojúhelníkový průběh se rozpoznává velice podobně jako pilovitý. Linearita se kontroluje jak mezi minimem a následujícím maximem, tak i mezi maximem a následujícím minimem.



Obrázek 5.2: Zjištění orientace náběhové hrany pilovitého signálu porovnáním rozdílu napětí v okolí střední hodnoty

5.4.4 Sinus

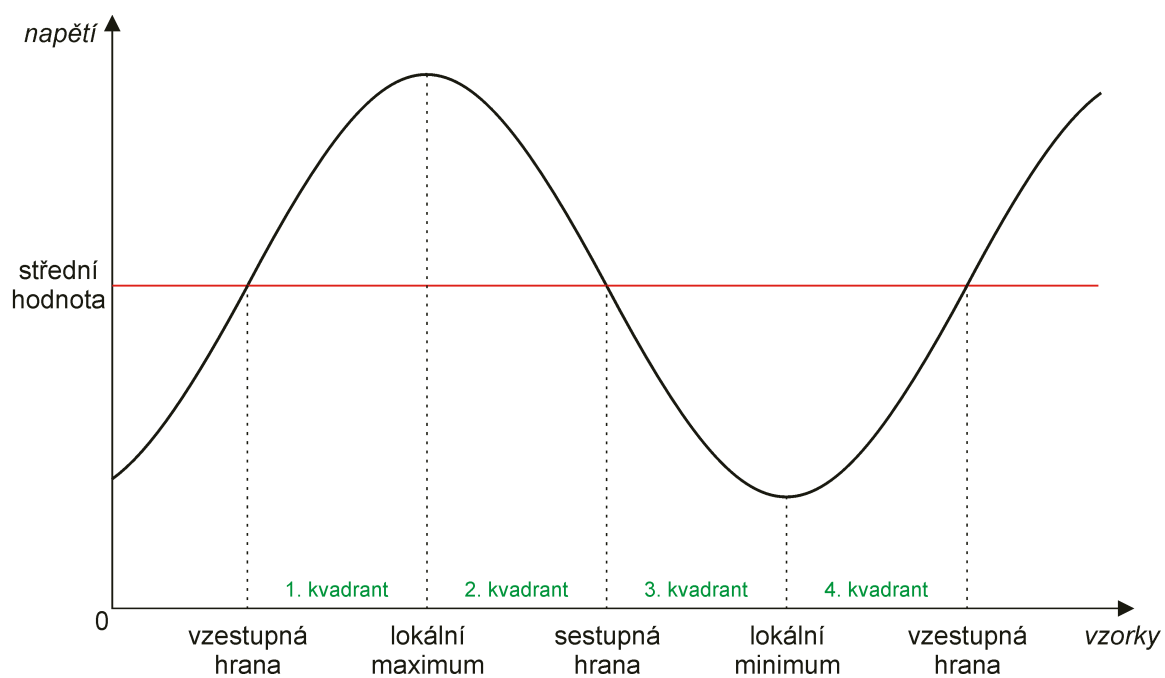
Perioda funkce sinus se před rozpoznáváním musí rozdělit na 4 kvadranty. K tomu je třeba znát indexy vzestupných a sestupných hran a indexy lokálních maxim a minim, viz obr. 5.3.

V programu je vložena tabulka s hodnotami funkce sinus v 1. kvadrantu. Jedná se o 128 hodnot s 11ti bitovou přesností. Tyto hodnoty se musí převést na rozsah (amplituda, perioda) navzorkovaného signálu a upravit podle aktuálního kvadrantu funkce. Toto je poměrně výpočetně náročné, protože se převod musí provádět pro každý vzorek a musí se použít datový typ long (32 bitů).

Jakmile známe teoretickou hodnotu funkce, můžeme již snadno spočítat odchylku vzorku od této hodnoty. Po zkontrolování několika period a porovnání průměrné odchylky s empirickou konstantou program rozhodne o tom, zda se jedná o funkci sinus či ne.

5.5 Zobrazení na LCD

Po úspěšném rozpoznání tvaru signálu je na LCD displeji naznačen jeho tvar. Po něm je vypsána frekvence, amplituda a střída (viz obr. 5.4). Pokud není rozpoznán žádný tvar, zobrazí se na displeji nápis ŠUM.



Obrázek 5.3: Zobrazení hran, lokálního maxima a minima a kvadrantů



Obrázek 5.4: Ukázka detekovaného signálu (obdělák, frekvence 1011 Hz, amplituda 107 mV, střída 49%)

Závěr

Cílem této práce bylo pomocí FITkitu vytvořit generátor a detektor funkcí ovládaný PS/2 klávesnicí.

Generátor funkcí pracuje do frekvence cca 2 kHz bez problémů. Nad touto frekvencí začínají být funkce sinus, trojúhelník a pila již příliš zkreslené. Generování obdélníkového průběhu je bez problémů až do 10 kHz. V případě dalšího vývoje bych se zaměřil na zvýšení maximální generovatelné frekvence.

Detektor signálu pracuje spolehlivě za podmínky dostatečně silného vstupního signálu. Signál s amplitudou menší než 100 mV nelze s jistotou detekovat, protože se již nachází v pásmu šumu. Také je možné, že při vyšší frekvenci bude funkce sinus vyhodnocena jako trojúhelník, protože v jedné periodě signálu bude příliš málo vzorků na rozlišení těchto dvou funkcí. V případě dalšího vývoje bych se zaměřil na zvýšení vzorkovací frekvence a na možnost detekce signálů s nízkou frekvencí (nižší než 80 Hz).

V rámci zadání jsem vytvořil i modul pro komunikaci s PS/2 klávesnicí, který převádí scan kódy na znaky. Zmínit se musím i o modulu ovládajícím LCD displej, který je schopen pracovat i s českými znaky. Tyto 2 moduly jsem se snažil naprogramovat tak, aby byly snadno opětovně použitelné i v dalších projektech.

Vlastní přínos shledávám v prohloubení mých znalostí v oblasti vestavěných systémů a periferních zařízení. Také jsem se prakticky seznámil s hradlovým polem FPGA a s komunikačním protokolem PS/2 klávesnice. Nelze opomenout ani získání praktických zkušeností s DTP programem \LaTeX při sazbě této technické zprávy.

Seznam použitých zdrojů

- [1] Brouwer, A.: Keyboard scancodes. [online], rev. 1.2e, [cit. 6. 3. 2007].
URL <http://www.win.tue.nl/~aeb/linux/kbd/scancodes.html>
- [2] Chapweske, A.: The PS/2 Keyboard Interface. [online], rev. 3. 1. 2004, [cit. 6. 3. 2007].
URL <http://www.computer-engineering.org/ps2keyboard/>
- [3] Chapweske, A.: The PS/2 Mouse/Keyboard Protocol. [online], rev. 3. 9. 2005, [cit. 6. 3. 2007].
URL <http://www.computer-engineering.org/ps2protocol/>
- [4] Data Image Corporation: LCD Module Specification. [online], rev. 5. 6. 2000, [cit. 28. 4. 2007].
URL <http://merlin.fit.vutbr.cz/FITkit/doc/hardware/datasheet/CM1610NR-J2.pdf>
- [5] Markovič, J.: FITkit Tutoriál. [online], [cit. 8. 5. 2007].
URL <http://merlin.fit.vutbr.cz/FITkit/docs/navody/20060210a.html>
- [6] Markovič, J.: Řadič přerušení. [online], [cit. 7. 5. 2007].
URL <http://merlin.fit.vutbr.cz/FITkit/docs/firmware/20060404int.html>
- [7] Martínek, T.: Technologie FPGA. [online], [cit. 6. 5. 2007].
URL https://www.fit.vutbr.cz/study/courses/PCS/private/prednasky/03_technologie_fpga/fpga_techn.pdf
- [8] Pech, J.: Programovatelné logické obvody. [online], [cit. 6. 5. 2007].
URL <http://www.sweb.cz/fpga/>
- [9] Příspěvatelé Wikipedie: Euklidův algoritmus. Wikipedie: Otevřená encyklopedie. [online], verze 1310464 ze dne 16. 3. 2007, [cit. 28. 4. 2007].
URL http://cs.wikipedia.org/w/index.php?title=Euklid%C5%AFv_algoritmus&oldid=1310464
- [10] Vašíček, Z.: Řadič PS/2. [online], [cit. 6. 3. 2007].
URL <http://merlin.fit.vutbr.cz/FITkit/docs/firmware/ps2ctrl.html>

Seznam použitých zkratek

A/D	Analog / Digital – Analogově digitální převodník
CGRAM	Character Generator RAM – Paměť uživatelských znaků
D/A	Digital / Analog – Digitálně analogový převodník
DDRAM	Display Data RAM – Paměť zobrazených znaků
DSP	Digitální Signálový Procesor
EEPROM	Electrically Erasable Programmable Read-Only Memory – Elektronicky mazatelná paměť
FPGA	Field-Programmable Gate Array – Programovatelné hradlové pole
GNU	Svobodný software
LCD	Liquid Crystal Display – Displej z tekutých krystalů
LED	Light Emitting Diode – Svítivá dioda
MCU	Micro-Controller Unit – Mikrokontrolér
RAM	Random Access Memory – Paměť s libovolným (náhodným) přístupem
RISC	Reduced Instruction Set Computer – Procesor s redukovanou instrukční sadou
ROM	Read Only Memory – Paměť jen pro čtení
SPI	Serial Peripheral Interface – Sériové rozhraní
USART	Universal Synchronous Asynchronous Receiver Transmitter – Sériové rozhraní
USB	Universal Serial Bus – Univerzální sériová sběrnice
VHDL	VHSIC Hardware Description Language – Jazyk pro popis el. obvodů

Seznam příloh

- A Seznam instrukcí LCD displeje
- B Tabulka SCAN kódů
- C Návod k použití

Příloha A

Seznam instrukcí LCD displeje

Instrukce	Kód										Doba
	RS	RW	7	6	5	4	3	2	1	0	
Vymazání displeje	0	0	0	0	0	0	0	0	0	1	1,53 ms
Návrat na počáteční pozici	0	0	0	0	0	0	0	0	1	*	1,53 ms
Nastavení módu vstupu dat	0	0	0	0	0	0	0	1	ID	S	39 μ s
Nastavení módu displeje	0	0	0	0	0	0	1	D	C	B	39 μ s
Posuv kurzoru nebo displeje	0	0	0	0	0	1	SC	RL	*	*	39 μ s
Nastavení funkce displeje	0	0	0	0	1	DL	N	F	*	*	39 μ s
Nastavení adresy CGRAM	0	0	0	1	A	A	A	A	A	A	39 μ s
Nastavení adresy DDRAM	0	0	1	A	A	A	A	A	A	A	39 μ s
Čtení adresy a bitu <i>Busy flag</i>	0	1	BF	A	A	A	A	A	A	A	
Zápis dat do CGRAM nebo do DDRAM	1	0	D	D	D	D	D	D	D	D	43 μ s
Čtení dat z CGRAM nebo z DDRAM	1	1	D	D	D	D	D	D	D	D	43 μ s

Tabulka A.1: Instrukce LCD displeje. Převzato z [4, str. 12].

Příloha B

Tabulka SCAN kódů

Kód	Klávesa	Kód	Klávesa	Kód	Klávesa	Kód	Klávesa
29	‘ ~	3a	CapsLock	39	space	4e	KP-+
2	1 !	1e	A	e0-38	RAlt	e0-1c	KP-Enter
3	2	1f	S	e0-1d	RCtrl	1	Esc
4	3 #	20	D	e0-52	Insert	3b	F1
5	4 \$	21	F	e0-53	Delete	3c	F2
6	5 %	22	G	e0-47	Home	3d	F3
7	6 ^	23	H	e0-4f	End	3e	F4
8	7 &	24	J	e0-49	PgUp	3f	F5
9	8 *	25	K	e0-51	PgDn	40	F6
0a	9 (26	L	e0-4b	Left	41	F7
0b	0)	27	; :	e0-48	Up	42	F8
0c	- _	28	“	e0-50	Down	43	F9
0d	= +	0	non-US-1	e0-4d	Right	44	F10
0e	Backspace	1c	Enter	45	NumLock	57	F11
0f	Tab	2a	LShift	47	KP-7 / Home	58	F12
10	Q	2c	Z	4b	KP-4 / Left	e0-37	PrtScr
11	W	2d	X	4f	KP-1 / End	54	Alt+SysRq
12	E	2e	C	e0-35	KP-/	46	ScrollLock
13	R	2f	V	48	KP-8 / Up	e1-1d-45	Pause
14	T	30	B	4c	KP-5	e0-46	Ctrl+Break
15	Y	31	N	50	KP-2 / Down	e0-5b	LWin
16	U	32	M	52	KP-0 / Ins	e0-5c	RWin
17	I	33	, <	37	KP-*	e0-5d	Menu
18	O	34	. >	49	KP-9 / PgUp	e0-5f	Sleep
19	P	35	/ ?	4d	KP-6 / Right	e0-5e	Power
1a	[{	36	RShift	51	KP-3 / PgDn	e0-63	Wake
1b] }	1d	LCtrl	53	KP-. / Del		
2b	\\	38	LAlt	4a	KP- -		

Tabulka B.1: Seznam SCAN kódů sady 1

Příloha C

Návod k použití

Naprogramování FITkitu

Nejdříve zkompilujte a poté nahrajte do mikrokontroléru program ze složky `sw`. Postupujte podle [5, odst. 1.1].

Následně zkompilujte architekturu pro FPGA (postup [5, odst. 1.2]). Po zkompilování nahrajte architekturu do FPGA (postup [5, odst. 1.3]).

Připojení klávesnice

Připojte PS/2 klávesnici do spodního (JP3) PS/2 konektoru. Propojte vodičem pin č. 5 na konektoru JP10 s pinem č. 26 na konektoru JP9.

Audio konektory

Výstup generátoru je přiveden na oba kanály spodního audio konektoru (JP7). Signál k detektoru signálu se připojuje na pravý kanál horního audio konektoru (JP6).

Pro správnou funkci je třeba mít uzavřeny propojky J13–J16 a propojku J19.

Vstup do menu

Po spuštění FITkitu se program nachází v základním stavu. Na displeji je zobrazeno “Výstup neaktivní”. Do menu, ve kterém lze nastavit parametry funkce, kterou chcete generovat, se vstupuje stisknutím klávesy `Enter` na připojené PS/2 klávesnici.

Pohyb v menu

V menu se lze mezi jednotlivými položkami pohybovat kurzorovými klávesami. Možné pohyby jsou vždy zobrazeny šipkami na okraji displeje.

Potvrzení volby se provádí klávesou `Enter`.

Zpět do základního režimu se dostanete stisknutím klávesy `Esc`.

Nastavení parametrů generovaného signálu

Nastavení tvaru generovaného signálu provedete zobrazením položky menu č. 2.1–2.5 a stiskem klávesy `Enter`. Na displeji se na 0,5s objeví nápis `OK` a volba je uložena.

Nastavení frekvence, amplitudy a střídání se provádí zobrazením příslušné položky menu a stiskem klávesy `Enter`. Na displeji se zobrazí výzva k zadání hodnoty. Hodnotu je možné

zadat buď přímo číselnými klávesami, nebo zvyšovat či snižovat číslici o 1 kurzorovými klávesami Nahoru a Dolů. Kurzor lze mezi číslicemi přesouvat klávesami Vlevo a Vpravo. Hodnotu potvrďte klávesou Enter nebo stornujte klávesou Esc.

Spuštění a zastavení generování

Spuštění a zastavení generování nastaveného signálu se provádí pomocí položky menu č. 1 nebo stiskem klávesy Tab. Generování je také zastaveno vždy při změně parametrů generované funkce.

Na generování signálu upozorňuje na klávesnici svítící LED dioda Scroll Lock.

Detekce signálu

Detektor signálů se spouští stisknutím klávesy Enter nad položkou menu č. 6. Pokud je poté na vstupu A/D převodníku rozeznán známý signál, na LCD displeji se naznačí jeho průběh a vypíše se frekvence v Hz, amplituda v mV a střída v %.

Pokud není rozeznán žádný signál, na displeji se zobrazí nápis ŠUM.