

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

## ROZŠIŘUJÍCÍ MODUL S BAREVNÝM LCD DISPLE- JEM PRO PLATFORMU FITKIT

BAKALÁŘSKÁ PRÁCE

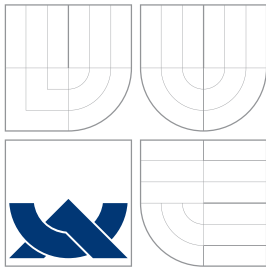
BACHELOR'S THESIS

AUTOR PRÁCE

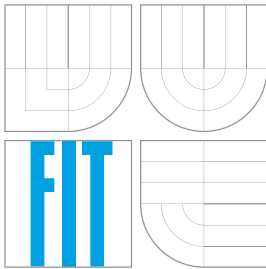
AUTHOR

PAVEL MICHAL

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

# ROZŠIŘUJÍCÍ MODUL S BAREVNÝM LCD DISPLE- JEM PRO PLATFORMU FITKIT

EXPANSION MODULE WITH COLOR LCD DISPLAY FOR FITKIT PLATFORM

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PAVEL MICHAL

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. VÁCLAV ŠIMEK

BRNO 2011

## **Abstrakt**

Cílem práce bylo vytvoření modulu s barevným LCD displejem pro platformu FITkit a jeho řadičem ve VHDL. Modul umožňuje využít všechny obrazové režimy použitého displeje a zobrazování obrazových dat z mikrokontroléru, počítače i FPGA. Práce se také zabývá rezistivním dotykovým ovládáním displeje.

## **Abstract**

The objective of this thesis was to create a module with color LCD display for FITkit platform and its controller in VHDL. The module allows you to use all display modes of used display and display image data from the microcontroller, FPGA and PC. This thesis also deals with resistive touch screen controls.

## **Klíčová slova**

Displej, dotykové ovládání, DPS, FITkit, FPGA, framebuffer, FTDI, LCD, modul, MPSSE, QVGA, řadič, SDRAM, TFT, VHDL.

## **Keywords**

Controller, display, FITkit, FPGA, framebuffer, FTDI, LCD, module, MPSSE, PCB, QVGA, SDRAM, TFT, touch control, VHDL.

## **Citace**

Pavel Michal: Rozšiřující modul s barevným LCD displejem pro platformu FITkit, bakalářská práce, Brno, FIT VUT v Brně, 2011

# Rozšiřující modul s barevným LCD displejem pro platformu FITkit

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Václava Šimka

.....  
Pavel Michal  
15. května 2011

## Poděkování

Rád bych poděkoval panu Ing. Václavu Šimkovi za odbornou pomoc, především při výrobě a ladění DPS.

© Pavel Michal, 2011.

*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

<b>1 Úvod</b>	<b>3</b>
<b>2 Technologie LCD displejů</b>	<b>4</b>
2.1 Módy kapalných krystalů	4
2.1.1 TN mód	4
2.1.2 IPS mód	4
2.1.3 VA mód	6
2.2 Podsvětlení	6
2.3 Adresace pixelů displeje	7
2.4 Rezistivní dotyková technologie se 4-mi vodiči	8
<b>3 Platforma FITkit</b>	<b>9</b>
3.1 FPGA & MCU	9
3.2 USB převodník FTDI	9
3.3 Paměť SDRAM	10
3.4 Další části platformy FITkit	10
<b>4 Koncepce rozšiřujícího modulu</b>	<b>11</b>
4.1 Displej	11
4.2 Integrovaný obvod LDS8845	12
4.3 Slot pro microSD kartu	12
4.4 Integrovaný obvod TSC2046	12
<b>5 Technická realizace</b>	<b>13</b>
5.1 Deska plošných spojů	13
5.2 Rozhraní displeje	13
5.2.1 Systémové rozhraní	13
5.2.2 Rozhraní RGB	14
5.2.3 Rozhraní VSYNC	15
5.2.4 Rozhraní obvodu TSC2046	15
5.2.5 Popis konektoru	16
5.2.6 Schéma řadiče	17
5.2.7 Ovládání	17
<b>6 Implementace systémových komponent</b>	<b>18</b>
6.1 Popis příkazů	18
6.2 SPI – komunikace s MCU	19
6.3 FTDI SPI – komunikace s PC	19

6.3.1	FTDI software	20
6.4	DISPLAY CONTROL – hlavní řídicí komponenta	20
6.5	SYSTEM INTERFACE	20
6.6	RGB INTERFACE	20
6.7	VSYNC INTERFACE	20
6.8	TOUCH CONTROL	21
6.9	TEXT GENERATOR	21
6.10	SDRAM, WRITE a READ FIFO	21
6.11	Software na PC	22
6.12	Software v MCU	22
6.13	Využití zdrojů FPGA	22
6.13.1	Výsledky syntézy řadiče	22
<b>7</b>	<b>Závěr</b>	<b>23</b>
	<b>Literatura</b>	<b>24</b>
	<b>Seznam použitých zkratk</b>	<b>25</b>
<b>A</b>	<b>Schéma modulu</b>	<b>26</b>
<b>B</b>	<b>Deska plošných spojů</b>	<b>27</b>
<b>C</b>	<b>Obsah CD</b>	<b>28</b>

# Kapitola 1

## Úvod

V dnešní době se s displeji z kapalných krystalů (LCD) setkáme na každém kroku. Barevné LCD displeje můžeme nalézt v přenosných počítačích, mobilních telefonech, MP3 přehrávačích, digitálních fotoaparátech a videokamerách, měřících a dalších zařízeních, kde je kladen důraz na jednoduchou obsluhu a přehledné zobrazení informací. Barevný LCD displej také zvyšuje hodnotu produktu a snižuje nároky na obsluhu zařízení, které takový displej obsahuje.

První barevné obrazovky počítačů, stejně jako televizory, fungovaly na principu katodové trubice (CRT) a po dlouhou dobu byly nejpoužívanější barevné zobrazovací zařízení. CRT obrazovky poskytují věrné podání barev, jsou ale velké a těžké a mají velkou spotřebu elektrického proudu, pro použití v přenosných zařízeních proto nejsou vhodné. Na konci 20. století začaly být CRT obrazovky nahrazovány displeji z kapalných krystalů. LCD displeje jsou oproti CRT obrazovkám lehké, tenké, mají menší spotřebu, jsou šetrnější k očím (nízká obnovovací frekvence CRT obrazovky způsobuje tzv. blikání obrazu, anglicky flicker) a vyzařují méně elektromagnetického záření. Na druhou stranu mohou mít LCD displeje vyšší latenci a menší pozorovací úhly ve srovnání s CRT.

Tyto vlastnosti umožňují použít LCD displeje v přenosných zařízeních. S rostoucí poptávkou a zdokonalováním technologického procesu výroby dochází ke zlevňování LCD displejů, které dnes plně nahradily CRT obrazovky. Dostupnost různých velikostí a nízká cena barevných LCD displejů pomohly k jejich rozšíření do oblastí, kde dříve vládly pouze displeje černobílé.

Tato práce je zaměřena na implementaci řadiče grafického displeje na platformě FITkit. Řadič umožňuje využívat všechny zobrazovací režimy vybraného barevného LCD displeje, zobrazovat text generovaný v FPGA, obrázky a grafická data a to jak z mikrokontroléru MSP430, který je součástí platformy FITkit, tak z PC. Připojení displeje k FITkitu je řešeno samostatnou, jednovrstvou oboustrannou deskou plošných spojů, připojenou na port X – vstupně-výstupní port FPGA. Řadič displeje je implementován v hardware (FPGA) a jeho popis je realizován v jazyce VHDL. Pro tento modul grafického displeje byl vybrán displej s rezistivní dotykovou fólií DT028TFT-TS od firmy DisplayTech, který obsahuje integrovaný řadič ILI9320 firmy ILITEK.

V následující kapitole se podíváme, jak jak fungují LCD displeje, k čemu je technologie TFT a jak funguje dotyková vrstva displejů. Třetí kapitola je věnovaná seznámení s platformou FITkit. Další kapitola popisuje, co všechno modul obsahuje. V Technické realizaci jsou popsána jednotlivá rozhraní displeje a rozhraní, se kterými bude řadič komunikovat. A poslední část s popisem implementace nám objasní, jak řadič modulu funguje. Na závěr jsou navrženy možná vylepšení modulu a další pokračování projektu.

## Kapitola 2

# Technologie LCD displejů

Kapalné krystaly byly objeveny v roce 1888 rakouským botanikem Friedrichem Reinitzerem, který experimentoval s cholesterolovou organickou kapalinou a zjistil, že po zahřátí kapalina prošla fází z mléčně zabarvené do čiré kapaliny. Toto bylo později vysvětleno jako přechod z opticky a elektricky anizotropní kapaliny do izotropní. Anizotropie, tedy rozdíl v relativní permitivitě a indexu lomu v různých natočeních molekuly kapaliny, vedla k analogii s anizotropií pevných krystalů – odtud název kapalné krystaly [6].

### 2.1 Módy kapalných krystalů

#### 2.1.1 TN mód

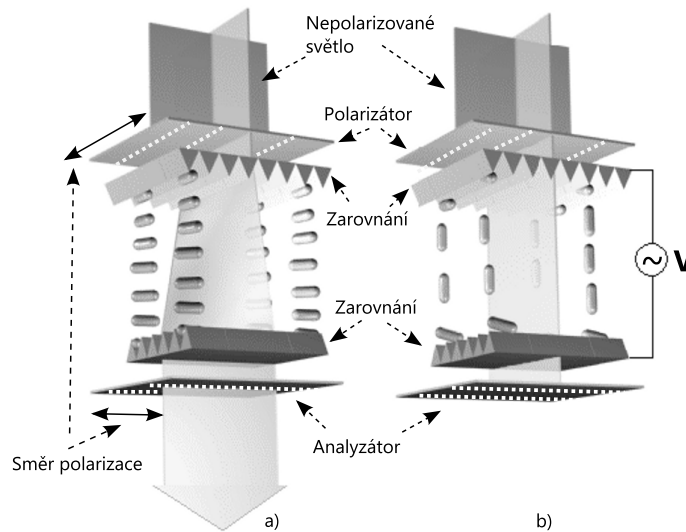
V LCD displejích se využívá kapalných krystalů v tzv. Twisted Nematic (TN) módu, ve kterém je možné regulovat množství světla propouštěného jedním pixelem. Nematické molekuly jsou poskládány podélně a rovnoběžně vůči sobě. Kapalné krystaly jsou uloženy mezi dvěma skleněnými vrstvami, jejichž vnitřní strany jsou opatřeny elektrodami z ITO, které na sobě mají polyimidový povrch s drážkami. Tyto drážky zajišťují zkroucení kapalných krystalů šroubovitě o 90 stupňů. Ke skleněné vrstvě obalující kapalné krystaly jsou z obou stran přidány polarizační filtry, světlo vstupuje na jedné straně přes polarizační filtr – tzv. polarizátor a vychází na druhé straně přes tzv. analyzátor [9].

Při nulovém napětí jsou molekuly kapalných krystalů zkroucené o 90 stupňů a přes polarizační filtry, které jsou vůči sobě také kolmo pootočený, může pronikat světlo (obrázek 2.1a). Při přivedení střídavého napětí na povrch kapalných krystalů dochází k rozbití šroubovitěho seskupení molekul až do té míry, že světlo přes kolmo pootočené polarizační filtry neprojde a výsledkem je černý bod (obrázek 2.1b). Regulaci intenzity propouštěného světla je tedy možné provádět změnou velikosti střídavého napětí. Vzhledem k tomu, že se molekuly kapalných krystalů při aplikaci střídavého napětí otáčejí kolem své kratší osy, dosahují TN displeje malých pozorovacích úhlů. Kapalné krystaly musí být řízeny střídavým napětím bez stejnosměrné složky, která by vyvolala elektrolýzu a ta by se kapalné krystaly znehodnotila. U barevných displejů je pixel rozdělen na 3 subpixely (červená, zelená, modrá).

#### 2.1.2 IPS mód

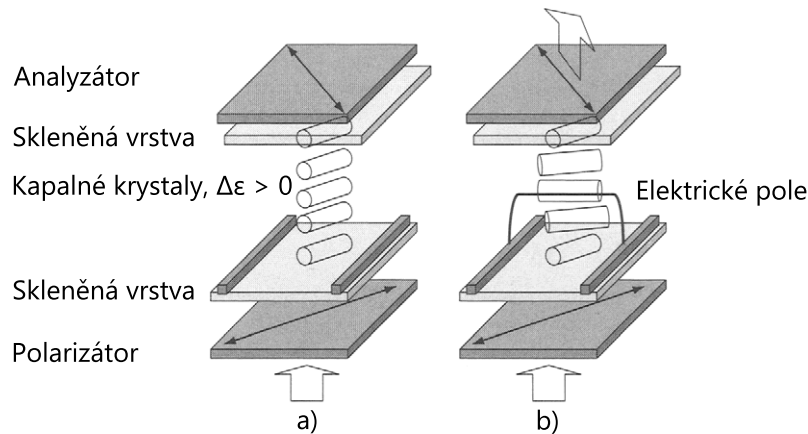
V IPS (In Plane Switching) módu je napětí regulující intenzitu procházejícího světla aplikováno mezi dvěma elektrodami na TFT buňce a elektrické pole je ke skleněným vrstvám





Obrázek 2.1: Průchod světla v TN módu. Světlo a) prochází, b) neprochází. Zdroj [9].

paralelní, narozdíl od TN módu. Postranní elektrické pole způsobuje, že se molekuly kapalných krystalů otáčí paralelně vůči obklopujícím vrstvám. Protože zůstávají molekuly kapalných krystalů v IPS módu (narozdíl od TN) vždy paralelně orientované ke skleněným vrstvám, dosahují IPS displeje širokých pozorovacích úhlů, které jsou stejné ve všech směrech. V IPS módu musí být použity kapalné krystaly s pozitivní relativní permitivitou, aby jejich molekuly dosahovaly zmíněného paralelního chování.



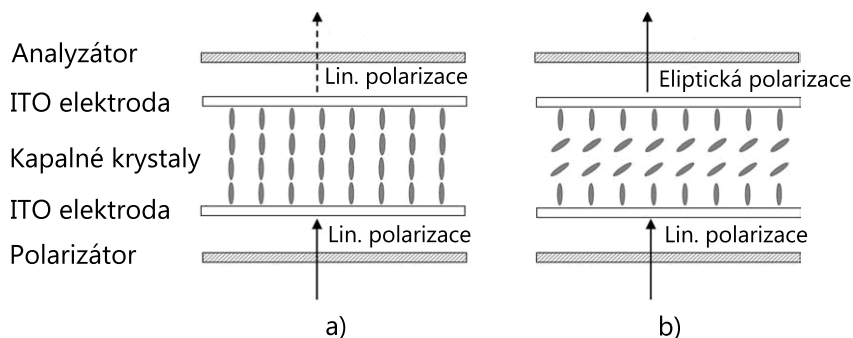
Obrázek 2.2: Průchod světla v IPS módu. Světlo a) neprochází, b) prochází. Zdroj [6].

Na obrázku 2.2 je ukázán princip IPS módu. Zarovnávací vrstva molekul je vytvořena pod úhlem 45 stupňů vůči elektrodám na TFT vrstvě. Světlo do buňky vstupuje polarizátorem a prochází molekulami kapalných krystalů. Pokud však na molekuly nepůsobí elektrické pole, nezmění se polarizace světla a to neprojde přes analyzátor, který je vůči polarizátoru pootočen o 90 stupňů. Se zvyšujícím se napětím na elektrodách se vlivem působícího elektrického pole postupně otáčí molekuly kapalných krystalů až o 45 stupňů. To způsobí polarizaci procházejícího světla, které může projít analyzátozem. Stejně jako v TN módu je použito střídavé napětí. IPS displeje potřebují oproti TN displejům silnější podsvi-

cení, protože množství procházejícího světla je snižováno neprůhlednými elektrodami TFT buněk a to až o 50 % [6].

### 2.1.3 VA mód

V tomto módu, jak napovídá název (Vertical Alignment), jsou molekuly kapalných krystalů v klidovém stavu orientovány kolmo k obklopujícím skleněným vrstvám. U VA módu nedochází v takové míře k úniku světla přes kapalně krystaly v zavřeném stavu jako u TN nebo IPS módu, protože jsou molekuly orientovány kolmo, což zvyšuje kontrast displeje. VA buňku můžeme vidět na obrázku 2.3. Ve stavu, kdy kapalně krystaly nepropouští světlo, není na elektrody přivedeno střídavé napětí a světlo projde polarizátorem s nezměněnou polarizací k analyzátoru, kde se zastaví. Při přivedení napětí na elektrody se molekuly kapalných krystalů uprostřed uprostřed buňky začnou otáčet do vodorovné polohy, což způsobí polarizaci procházejícího paprsku světla, který pak může projít analyzátozem [11]. Pro VA mód se používají kapalně krystaly s negativní relativní permitivitou a nevyužívá se zarovnání molekul drážkami. Samotné VA displeje postrádají kontrolu nad rotací vrstev molekul při aplikaci elektrického pole. Proto se v praxi používají multidoménové VA (MVA) displeje pro zvýšení symetričnosti pozorovacích úhlů [6].



Obrázek 2.3: Průchod světla ve VA módu. Světlo a) neprochází, b) prochází. Zdroj [11].

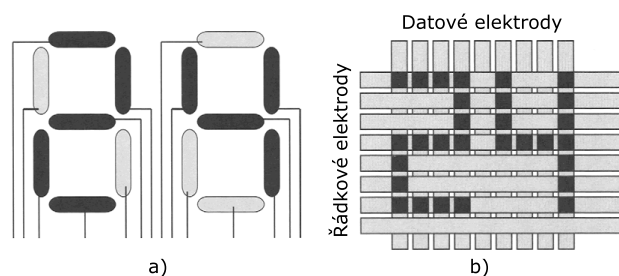
## 2.2 Podsvětlení

Pro zobrazení informací na displeji musí být kapalně krystaly podsvětleny, protože samy o sobě světlo nevyzařují, fungují pouze jako regulátor propuštěného světla. Podsvětlení může být reflexní, transmisivní a transreflexní. Reflexní podsvětlení nalezneme například v kalkulačkách a v náramkových hodinkách, pro zobrazení informací je využito okolní světlo a ve tmě na něm nic nevidíme. Barevné displeje s reflexním podsvětlením se vyznačují malou sytostí barev, na druhou stranu je reflexní podsvětlení výhodné pro zařízení napájená z baterií. Transmisivní podsvětlení je podsvětlení umělé, může jím být fluorescenční výbojka nebo LED diody. Takto podsvětlený displej má pochopitelně vyšší spotřebu než displej s reflexním podsvětlením a je špatně čitelný na přímém slunci. Kombinací těchto dvou druhů podsvětlení dostáváme transreflexní podsvětlení, které je kompromisem obou přístupů a jsou jím vybaveny například mobilní telefony [11].

## 2.3 Adresace pixelů displeje

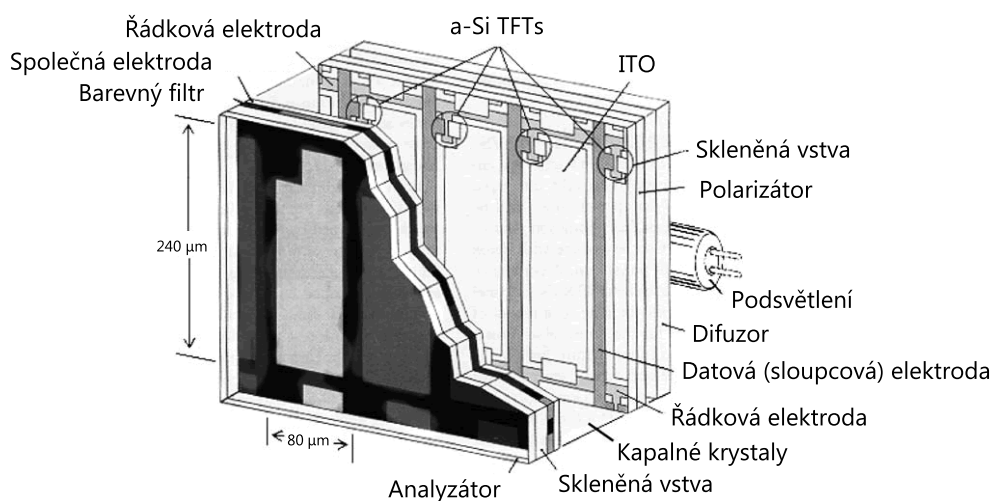
Displeje dělíme podle způsobu adresace pixelů na segmentové, displeje s pasivní a aktivní maticí. Pro zobrazení omezených alfanumerických informací, jako například hodiny nebo číslice v kalkulačce, se používají segmentové displeje. Displej je rozdělen na segmenty (typicky na 7) a ty jsou adresovány přímo (obrázek 2.4a).

Pro zobrazení jednoduché grafiky s nepříliš rychlým překreslováním se používají displeje s pasivní maticí. Obraz o rozměrech  $m \times n$  pixelů je adresován  $m$  vodiči pro výběr řádku a  $n$  pro výběr sloupce. Tomuto způsobu adresace, kdy je v daný okamžik aktivní pouze jeden řádek a pixely jsou nastaveny na hodnoty sloupcových vodičů, se nazývá multiplexované řízení. Výhodou je nízký počet potřebných vodičů, nevýhodou je však ovlivňování průměrné hodnoty pixelů na ostatních řádcích a pomalá odezva displeje (obrázek 2.4b).



Obrázek 2.4: a) Segmentový displej, b) displej s pasivní maticí. Zdroj [6].

Displeje s aktivní maticí poskytují vyšší kvalitu zobrazení, vyšší rozlišení i velikost zobrazované plochy, větší kontrast, větší pozorovací úhly a kvalitněji zobrazují video. Každý subpixel displeje s aktivní maticí obsahuje tranzistor a kondenzátor, kterým lze přesně řídit a uchovat napětí na každém subpixelu a odpadá tak problém multiplexovaného řízení. Displej s aktivní maticí se svým způsobem podobá paměti DRAM, která obdobně uchovává data. Displeje s aktivní maticí označujeme TFT (obrázek 2.5). Tento „tranzistorový film“ vzniká vyleptáním do tenké křemíkové vrstvy, která se nechá narůst na skle za teploty několika stovek stupňů Celsia. Nejrozšířenější typy TFT displejů jsou a-Si TFT a poly-Si TFT, což je dáno jejich méně náročnou výrobou za nižších teplot (300–500 °C) [6].

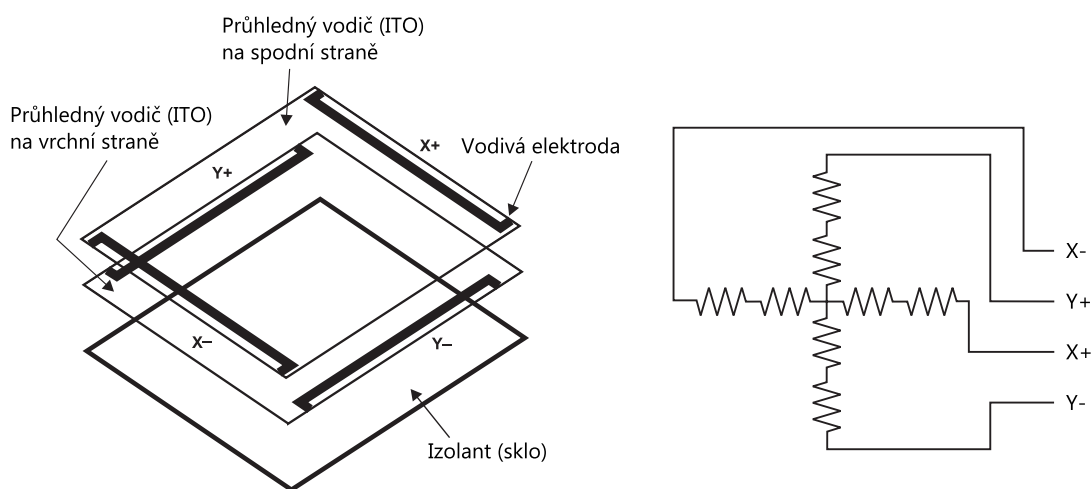


Obrázek 2.5: Struktura pixelu TFT displeje s transmisivním podsvícením. Zdroj [11].

## 2.4 Rezistivní dotyková technologie se 4-mi vodiči

Dotykové ovládání displeje představuje efektivní a přirozený způsob ovládání. Rezistivní dotyková fólie je nejdostupnější možností, jak dotykově ovládat displej. Ačkoliv tato fólie snižuje oproti jiným způsobům dotykového ovládání čitelnost displeje, rezistivní dotyková fólie je velmi odolná a je možné ji poškodit pouze velmi ostrým předmětem. Dotykový panel s rezistivní technologií se skládá ze dvou průhledných vrstev, které mají na vnitřní straně vodivé elektrody, mezi kterými je rezistivní plocha tvořená ITO (obrázek 2.6). Tyto vrstvy jsou odděleny průhlednými body a celý dotykový panel je umístěn na pevném podkladu, většinou přímo na skle displeje [10]. Při dotyku se vrchní vrstva vodivě spojí se spodní. Pokud je přivedeno napětí na jednu vrstvu, po vyvolání tlaku a spojení vrstev je vytvořen dělič napětí. Pozici dotyku zjistíme, když přivedeme napětí na vrstvu ve směru osy  $y$  a A/D převodníkem přečteme hodnotu vytvořenou děličem napětí. To stejné uděláme na vrstvě ve směru osy  $x$ . Hodnota napětí na takto vzniklém děliči napětí je lineární vůči pozici dotyku.

Abychom zajistili, že přečtené hodnoty byly validní, potřebujeme detekovat stav, kdy na displeji došlo k doteku. To lze zajistit pomocí pull-up rezistoru (s odporem větším než je celkový odpor dotykové plochy) připojeného na  $Y+$  a připojením  $X-$  na zem. Pokud na dotykovou vrstvu nepůsobí tlak, logická hodnota na  $Y+$  je 1, pokud dojde k dotyku, logická hodnota se změní na 0. Čtení  $x$ -ové a  $y$ -ové souřadnice bodu dotyku probíhá ve dvou krocích. Nejprve je přivedeno napětí na  $Y+$ , vodič označen  $Y-$  je připojen na zem a napětí na  $X+$  je změřeno. Poměr napětí na  $X+$  a napájecího napětí se rovná poměru souřadnice  $y$  vůči šířce obrazovky. Souřadnice  $x$  je získána připojením  $X+$  na napájecí napětí,  $X-$  na zem a na  $Y+$  změříme napětí. Poměr napětí na  $Y+$  a napájecího napětí je roven poměru  $x$ -ové souřadnice ku délce obrazovky. [7].



Obrázek 2.6: Rezistivní dotyková technologie se 4-mi vodiči. Zdroj [10].

## Kapitola 3

# Platforma FITkit

FITkit vznikl jako platforma pro výuku na FIT VUT. Je dostupný pro všechny studenty FIT a ti si tak mohou v praxi vyzkoušet návrh vestavěných systémů (Embedded Systems), jak během výuky, tak i během řešení samostatných projektů. V současné době je poslední verze platformy 2.0, která je zároveň využita pro realizaci této práce.

### 3.1 FPGA & MCU

FPGA jsou programovatelnými obvody, které mohou měnit svoji funkci podle aktuální konfigurace. Jsou složeny z mnoha konfigurovatelných logických bloků (CLB), které jsou propojeny programovatelnými propojeními. CLB je základní logická jednotka FPGA, skládající se z přepínací matice, multiplexorů a flip-flop klopných obvodů. CLB může implementovat kombinační logiku, posuvné registry nebo RAM [2].

FITkit je osazen programovatelným hradlovým polem (FPGA) XC3S50 firmy Xilinx, které obsahuje ekvivalentních 50 tisíc logických hradel a 4 blokové RAM o celkové kapacitě 72 Kb. Dále FITkit obsahuje 8 bitový mikrokontrolér MSP430F2617, který doplňuje funkčnost a řídí činnost FPGA. Komunikace mezi MCU a FPGA probíhá přes rozhraní SPI. MCU může být však použit i samostatně a nezávisle na FPGA a slouží také pro nahrání konfigurace do FPGA.

### 3.2 USB převodník FTDI

Dalším důležitým prvkem je USB převodník FT2232D od firmy FTDI, který zajišťuje komunikaci MCU i FPGA s PC. Převodník obsahuje 2 komunikační kanály (kanál A – FPGA, B – MCU) a několik režimů přenosů dat. Pro posílání uživatelských příkazů a programování FITkitu se používá sériová linka na kanále připojeném do MCU. Ta dosahuje rychlosti až 1 Mbit/s, což pro větší objem dat, jako jsou nekomprimovaná obrazová data displeje, nemusí dostačovat. Vyšší propustnosti dosahuje režim MPSSE, který zjednodušuje implementaci synchronní sériové komunikace pro rozhraní JTAG, I<sup>2</sup>C a SPI. Režim MPSSE je dostupný na kanále připojeném k FPGA, nevýhodou režimu je obsazení obou komunikačních kanálů. V módu SPI je však možné dosáhnout rychlosti až 5,6 Mbit/s, čehož využijeme pro přenos obrazových dat z PC do FPGA. Je třeba mít na paměti, že pro využití vyšší rychlosti synchronního přenosu přes USB je potřeba organizovat data do větších bloků (512 B) z důvodu režie protokolu USB (paketový přenos).

### 3.3 Paměť SDRAM

Pro rozhraní RGB a VSYNC, kdy je obrazovka periodicky překreslována, je použita SDRAM paměť K4S640832N-LC75 od firmy Samsung s kapacitou 8 MB. Tato paměť se nachází přímo na FITkitu a je připojena k FPGA. Paměť je rozdělena do 4 banků, každý s kapacitou  $2M \times 8$  bitů, maximální frekvence synchronizačního signálu je 133 MHz. Protože se jedná o dynamickou paměť, je potřeba obnovovat uchovávaná data tak, aby celá paměť byla obnovena nejpozději do 64 ms. Obnovovat dynamickou RAM je možné speciálním příkazem a nebo příkazem pro čtení. V tomto projektu je obnovování paměti zajištěno periodickým čtením oblasti dat, která je posílána k zobrazení na displeji. Musíme tedy vzít na vědomí fakt, že data z ostatních částí SDRAM nejsou obnovována a není tedy zajištěna jejich konzistence.

### 3.4 Další části platformy FITkit

FITkit obsahuje další části, které však nejsou v tomto projektu využity. Jedná se o konektory periferních zařízení (VGA, PS/2, RS-232), maticovou klávesnici  $4 \times 4$ , znakový LCD displej  $16 \times 2$ , piezo reproduktorek a zvukový kodek.

## Kapitola 4

# Koncepce rozšiřujícího modulu

Součástí práce bylo vytvoření modulu grafického displeje pro platformu FITkit. Modul je umístěn na desce plošných spojů o rozměrech  $6 \times 8$  cm a skládá se z těchto částí:

- Displej DT028TFT-TS s dotykovou fólií,
- integrovaný obvod LDS8845 pro řízení podsvětlení displeje,
- slot pro microSD kartu (v tomto projektu není používán),
- integrovaný obvod TSC2046, řadič dotykové fólie.

Pro komunikaci s modulem byl vytvořen řadič v FGPA. S tímto řadičem je možné displej ovládat z MCU přes SPI nebo z PC přes USB převodník FTDI2232D. Samotný displej podporuje tyto 4 rozhraní: *systémové rozhraní*, *RGB*, *VSYNC* a *SPI*. Systémové rozhraní – displej je ovládán zápisem hodnot do registrů displeje. V tomto režimu se provádí inicializace displeje, rozhraní umožňuje přístup do integrované GRAM (paměť pro grafická data) displeje pro vykreslování statických obrázků a lze použít funkce integrovaného řadiče ILI9320 jako jsou prokládání řádků, zmenšení rozměrů obrázku, rolování obrazovky (scroll) a tzv. „Partial display mode“, což je režim, kdy je vykreslována pouze část plochy displeje. RGB – tento režim je podobný známému VGA rozhraní, řadič generuje signály VSYNC, HSYNC, DOTCLK a ENABLE. V režimu RGB interface lze na displeji zobrazit video, animovanou grafiku nebo pohyblivé objekty generované v FPGA. Režim VSYNC používá signály VSYNC, který udává frekvenci překreslení displeje a signál WR pro vzorkování hodnot pixelů. Nejprve jsou nahrány data do interní GRAM displeje a poté je podle signálu VSYNC displej překreslen. Rozhraní SPI není v tomto projektu použito. Vizualizace modulu vytvořená v programu Eagle3D je na obrázku 4.1.

### 4.1 Displej

Displej DT028TFT-TS s úhlopříčkou 2.8 palců a transmisivním podsvícením poskytuje rozlišení  $240 \times 320$  pixelů s 18-ti nebo 16-ti bitovou barevnou hloubkou a využívá technologii TFT. Displej má poměrně malé pozorovací úhly (80 stupňů ve vertikálním i horizontálním směru) a dlouhou odezvu (kolem 35 ms bílá-černá)[4]. Displej je dostupný za cenu kolem 25 EUR (distributor Mouser Electronics, květen 2011). Integrovaný řadič displeje ILI9320 obsahuje 172 800 B grafické RAM paměti a je možné ovládat ho pomocí 4 typů rozhraní: systémové, RGB, VSYNC a SPI [1]. Součástí displeje je rezistivní dotyková fólie se čtyřmi vodiči X+, Y+, X-, Y-.

## 4.2 Integrovaný obvod LDS8845

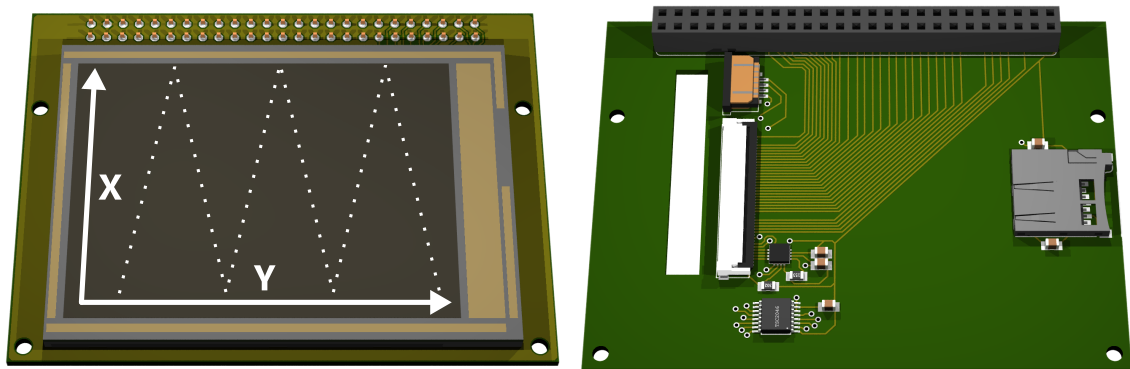
Obvod LDS8845 slouží k řízení podsvětlení displeje, které se skládá ze 4 bílých LED diod. Jejich jas je možné přes zmíněný obvod regulovat pomocí PWM.

## 4.3 Slot pro microSD kartu

Paměti typu flash představují v dnešní době levný velkokapacitní druh nevolatilní paměti. Paměťové karty s pamětí typu flash jsou rozšířené v přenosných zařízeních (MP3 přehrávače, mobilní telefony, digitální fotoaparáty, ...) a čtečky těchto karet najdeme téměř v každém notebooku. Mezi nejrozšířenější paměťové karty patří typ SD – Secure Digital. Tento typ paměťových karet zároveň obsahuje integrované SPI rozhraní, které zjednodušuje připojení takovéto paměťové karty k vestavěnému systému. Modul barevného displeje obsahuje slot pro microSD kartu, která je zatím nejmenší SD kartou a dá se pořídit kolem 100 Kč s kapacitou 1 GB (duben 2011). I když paměťová karta v tomto projektu využita není, v modulu s barevným displejem může být využita (po vytvoření příslušného řadiče) pro zobrazování obrázků či videa.

## 4.4 Integrovaný obvod TSC2046

Integrovaný obvod TSC2046 od firmy Texas Instruments zjednodušuje komunikaci s rezistivní dotykovou fólií displeje. Místo použití A/D převodníku (analog to digital converter) pro zjištění bodu dotyku je rezistivní fólie ovládána obvodem TSC2046, který je vybaven rozhraním SPI. Díky tomu můžeme informaci o dotyku zpracovávat v FPGA, které A/D převodník neobsahuje.



Obrázek 4.1: Vizualizace modulu s grafickým displejem.



## Kapitola 5

# Technická realizace

Před samotným návrhem je dobré shrnout požadavky na modul: „*Modul s barevným displejem pro platformu FITkit by měl být schopen pracovat se všemi obrazovými režimy použitého displeje, měl by umět zobrazovat grafiku z MCU i PC. Implementace by měla být provedena v hardware kvůli rychlejšímu a přesnějšímu zpracování signálů. Měla by být také možnost vykreslovat text v hardware*“. Časy uvedené na časových diagramech jsou minimální možné (maximální rychlost).

### 5.1 Deska plošných spojů

Aby bylo možné displej vůbec k platformě FITkit připojit, musela být nejprve vytvořena deska plošných spojů (DPS). DPS byla navržena v programu Eagle 5.10 od firmy Cadsoft jako jednovrstvá oboustranná deska a byl kladen důraz na co největší využití desky. Při návrhu byla vytvořena knihovna součástek (umístěna na CD), kterou je možné použít při návrhu nové desky. Schéma zapojení modulu je možné prohlédnout si na obrázku [A.1](#). Mimo samotného displeje a obvodu řídicího jeho podsvícení, byl modul doplněn o obvod řídicí dotykovou vrstvu displeje a slot pro microSD kartu, který lze využít v dalších projektech. Celý modul je napájen z FITkitu, který může dodávat proud až 500 mA (do spotřeby je nutné zahrnout také FITkit). Maximální odebíraný proud modulem je asi 235 mA a klidový proud (bez microSD karty) kolem 150 mA (údaje z datasheetu). Samotná výroba desky byla zajištěna vedoucím práce na UPSY FIT.

### 5.2 Rozhraní displeje

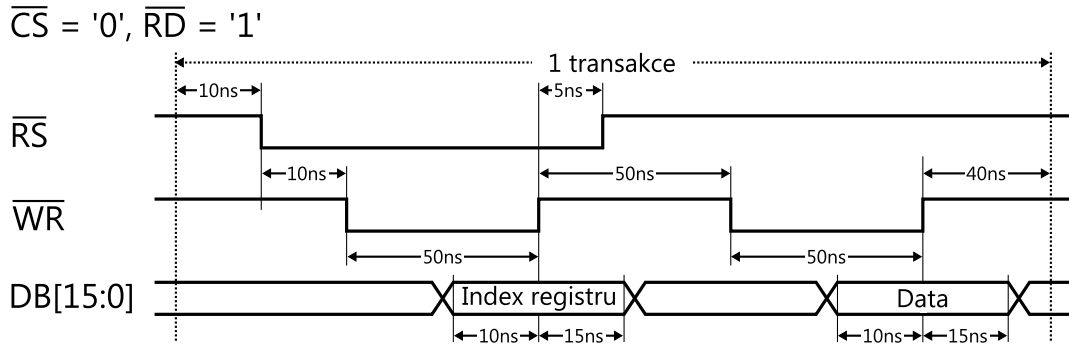
Displej DT028TFT-TS nabízí několik způsobů, jak na něm zobrazovat informace. Je vybaven rozhraními: systémové, RGB, VSYNC a SPI. Rozhraní SPI je sériová obdoba systémového rozhraní (umožňuje zápis do registrů displeje a vykreslování obrazu je řízeno interní synchronizací), a je nahrazeno rychlejším, paralelním systémovým rozhráním.

#### 5.2.1 Systémové rozhraní

Systémové rozhraní vychází z paralelního rozhraní Intel i80 a používá se pro přístup do registrů displeje (obrázek [5.1](#)). Používá vodiče: RS, CS, WR, RD a 16 datových bitů. Signálům RD a CS můžeme nastavit neměnnou logickou hodnotu, protože do registrů displeje budeme pouze zapisovat (signál RD nastavíme na log. 1) a systémové rozhraní bude aktivní

neustále, signál CS tedy bude nastaven do log 0, což nám zjednoduší implementaci. Při zápisu do registru displeje musíme nejprve nastavit index registru (1 B hodnota) na datovou část sys. rozhraní a uvedením signálu RS do log. 0 oznámit, že bude nastaven index registru. Poté můžeme zapisovat 2 B data jejich vystavením na datovou část sys. rozhraní a uvedením signálu WR do log. 0 signalizovat jejich přítomnost.

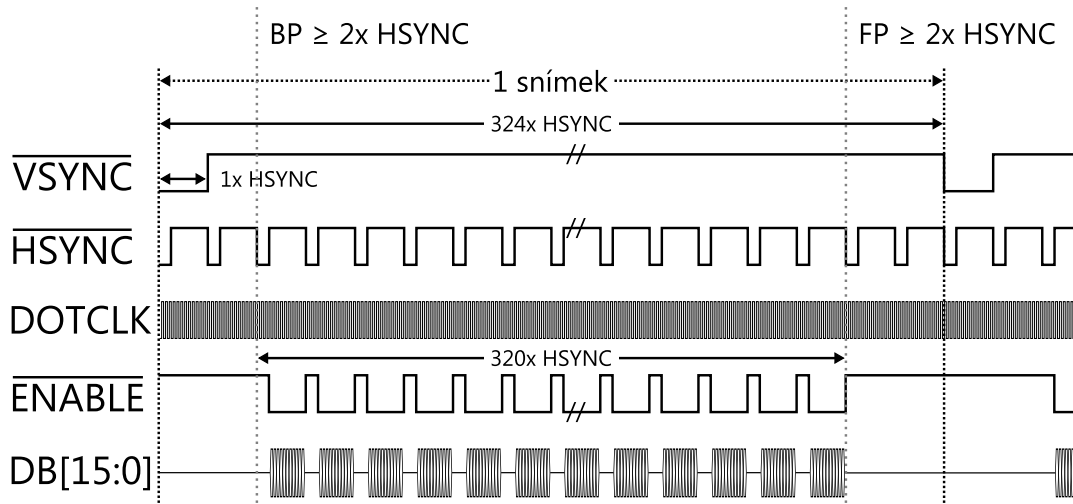
Přes systémové rozhraní můžeme zapisovat do grafické paměti displeje (GRAM), zápisem hodnoty do registru s indexem 0x22. Po zápisu je automaticky inkrementována adresa na další pixel v GRAM. Při sekvenčním zápisu do GRAM nemusíme nastavovat před každým zápisem index registru, pokud je jeho aktuální hodnota 0x22.



Obrázek 5.1: Časový diagram systémového rozhraní při zápisu.

### 5.2.2 Rozhraní RGB

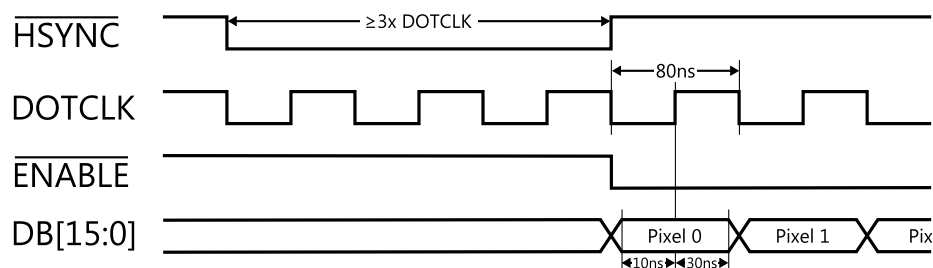
Rozhraní RGB obsahuje signály podobné VGA rozhraní – do displeje však nepřenášíme pouze řádkovou a snímkovou synchronizaci, ale i signál pro vzorkování jednotlivých pixelů. Nalezneme zde signály HSYNC, VSYNC, DOTCLK a ENABLE + hodnotu pixelu na datové části rozhraní (obrázek 5.2).



Obrázek 5.2: Rozhraní RGB, přehled signálů.

Pro správné zobrazení musí signály RGB rozhraní splňovat několik podmínek. Minimální délka periody signálu HSYNC v log. 0 je  $3 \times$  délka periody DOTCLK. Minimální délka

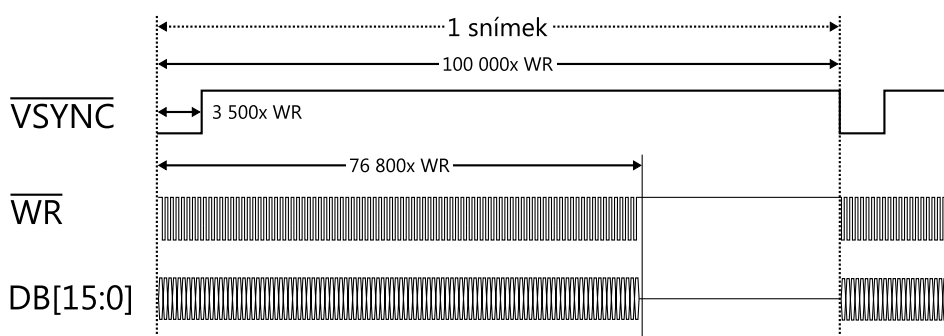
periody signálu VSYNC v log. 0 je 1 perioda signálu HSYNC. Signál ENABLE, který pokud je aktivní říká displej že může s nástupnou hranou signálu DOTCLK vzorkovat hodnoty pixelů vystavené na datové části rozhraní. Signál ENABLE není aktivní pokud je aktivní signál HSYNC nebo probíhá tzv. Back Porch (BP) či Front Porch (FP). BP a FP jsou použity pro nastavení časování rozhraní [4]. Hodnoty BP a FP uvádí počty řádků (násobky HSYNC), musí být dohromady menší než 16 a každý z nich musí být větší nebo roven 2 řádkům. Časový diagram zápisu pixelu přes rozhraní RGB je uveden na obrázku 5.3.



Obrázek 5.3: Detailní pohled na časování signálů rozhraní RGB.

### 5.2.3 Rozhraní VSYNC

Rozhraní VSYNC je určeno podobně jako RGB rozhraní pro zobrazení videa na displeji, je však jednodušší a neobsahuje tolik synchronizačních signálů (obrázek 5.4). Jedná se vlastně o systémové rozhraní doplněné o synchronizaci VSYNC udávající frekvenci překreslení displeje. Obrazová data jsou nejprve přes systémové rozhraní zapsána do GRAM a se sestupnou hranou signálu VSYNC je obrazovka překreslena. U VSYNC rozhraní je důležité dodržet minimální frekvenci 5,7 MHz pro zápis do GRAM, aby překreslování celé obrazovky bylo plynulé při frekvenci překreslení 60 Hz. Pro signál WR a datovou část DB[15:0] platí stejné časování jako u stejných signálů systémového rozhraní.

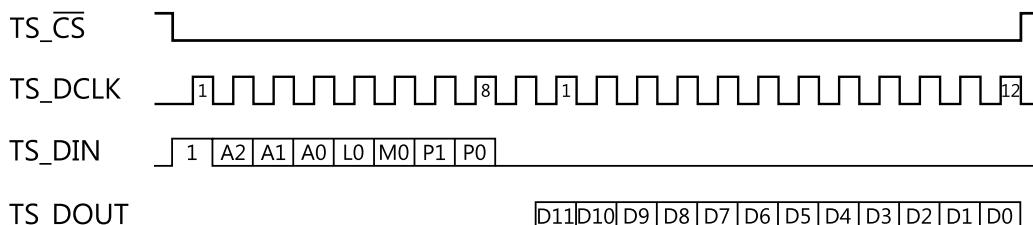


Obrázek 5.4: Signály VSYNC rozhraní.

### 5.2.4 Rozhraní obvodu TSC2046

Integrovaný obvod TSC2046 umožňuje dva způsoby měření pozice na dotykové vrstvě displeje a také měřit velikost tlaku dotyku a teplotu. První způsob měření – „sigle-ended“ mód měří klasicky velikost napětí mezi elektrodou dotykového panelu a zemí a převádí ji na číslo. Tento způsob není příliš přesný pro měření pozice dotyku na displeji, může být

zatížen množstvím chyb a používá se spíše jen pro měření napětí baterie, teploty a napětí na pomocném vstupu. Druhý způsob „differential-reference“ měří poměr napětí přímo na dotykové ploše, je méně náchylný na chyby a používá se pro měření pozice dotyku [5]. Při dotyku obrazovky obvod TSC2046 nastaví signál PENIRQ do log. 0 a FPGA na to zareaguje zasláním příkazu po SPI pro změření pozice bodu dotyku na ose x a y. V projektu nebudeme měřit teplotu ani sílu tlaku na displej. Formát komunikace s obvodem TSC2046 je uveden na obrázku 5.5.

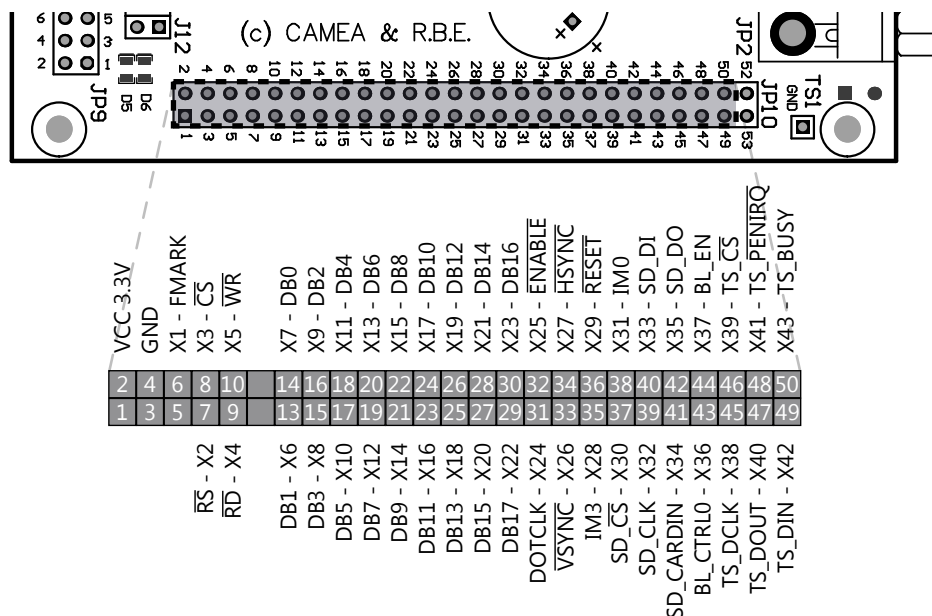


Obrázek 5.5: Formát komunikace s obvodem TSC2046.

Maximální frekvence signálu TS\_DCLK je 2,5 MHz. První bit signálu TS\_DIN je tzv. Start bit a má vždy hodnotu '1'. Bity A2-A0 definují, co se bude měřit (osa x, y, případně tlak na dotykovou plochu), bit L0 definuje přesnost měření (8/12 bitů), bit M0 mód měření (single-ended, differential) a P1-P0 určují režim napájení obvodu TSC2046.

### 5.2.5 Popis konektoru

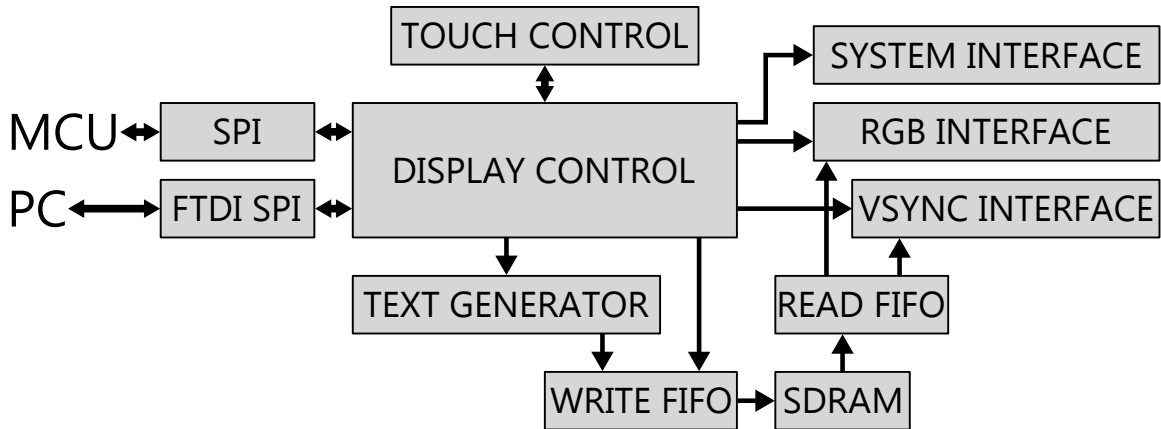
Modul se připojuje k FITkitu pomocí portu JP10 (v FPGA adresovaný jako signálový vektor X) a obsadí piny 1-50 (obrázek 5.6). Aby mohly být využity všechny piny portu JP10 je nutné mít rozpojenou propojku J6. Po vytvoření desky plošných spojů došlo ke změně návrhu, proto je v zdrojových souborech pro výrobu desky trochu jiné rozmístění signálů na konektoru. Jedná se pouze o piny 45-50 řadiče dotykové fólie.



Obrázek 5.6: Připojení signálů modulu k FITkitu.

### 5.2.6 Schéma řadiče

Pro ovládání modulu byl navržen řadič v FPGA podle dataflow schématu (obrázek 5.7). Jednotlivé bloky (komponenty) označují entity ve VHDL. Při návrhu jsem vycházel z doporučení firmy Altera [3]. Návrh je modulární, lze tedy jednotlivé komponenty zaměnit za jiné implementace nebo tyto komponenty použít v jiných projektech.



Obrázek 5.7: Schéma řadiče modulu grafického displeje.

### 5.2.7 Ovládání

Pro komunikaci s interním řadičem ILI9320 je použita sada registrů s indexy v rozsahu 0x00 až 0x98. Registry označené 0xA0–0xFF nejsou interním řadičem využity a proto je můžeme použít pro ovládání řadiče v FPGA. V řadiči modulu bude potřeba implementovat tyto příkazy:

- zapnutí/vypnutí RGB režimu,
- zapnutí/vypnutí VSYNC režimu,
- sekvenční zápis do GRAM displeje,
- zápis do RAM řadiče modulu,
- nastavení adresy RAM řadiče modulu,
- vytisknutí znaku na obrazovku (různá velikost, orientace písmene a pozice, barva, barva pozadí).

## Kapitola 6

# Implementace systémových komponent

Řadič modulu je implementován v FPGA a je ovládán přes SPI rozhraní příkazy z MCU nebo PC. Sada příkazů, jimiž je možné řadič ovládat je uvedena na obrázku 6.1.

Index	Příkaz	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0xA0	Index 0x22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xA1	GRAM write	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0xA2	RGB on/off	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R0
0xA3	VSYNC on/off	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	V0
0xA4	RAM write	0	0	0	0	0	0	0	0	D7	D6	D5	D4	D3	D2	D1	D0
0xA5	RAM x	0	0	0	0	0	0	0	0	X7	X6	X5	X4	X3	X2	X1	X0
0xA6	RAM y	0	0	0	0	0	0	0	Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
0xA7	RGB effect on/off	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xB0	Generate char	0	0	B0	F0	R1	R0	S1	S0	0	C6	C5	C4	C3	C2	C1	C0
0xB1	Char x offset	0	0	0	0	0	0	0	0	X7	X6	X5	X4	X3	X2	X1	X0
0xB2	Char y offset	0	0	0	0	0	0	0	Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
0xB3	Char color	0	0	0	0	0	0	0	0	C7	C6	C5	C4	C3	C2	C1	C0
0xB4	Char bg color	0	0	0	0	0	0	0	0	C7	C6	C5	C4	C3	C2	C1	C0

Obrázek 6.1: Příkazy podporované řadičem.

### 6.1 Popis příkazů

Příkaz **Index 0x22** – pouze nastaví index registru na hodnotu 0x22 na systémové sběrnici. **GRAM write** – sekvenční zápis hodnoty D[15:0] do GRAM displeje po nastavení indexu registru na hodnotu 0x22. **RGB on/off** – pokud je bit R0 nastaven na 1, RGB rozhraní je zapnuto, jinak je vypnuto. **VSYNC on/off** – pokud je bit V0 nastaven na 1, VSYNC rozhraní je zapnuto, jinak vypnuto. RAM je adresována sloupci a řádky, podobně jako obrazovka. **RAM write** – zápis D[7:0] do RAM, **RAM X** – nastavení adresy v RAM pro další zápis, hodnota 0-239 (pozice na řádku), **RAM Y** – nastavení adresy v RAM pro další zápis, hodnota 0-319 (výběr sloupce). **Generate char** – vykreslí znak na obrazovce, C[6:0] hodnota znaku, S[1:0] velikost, R0 rotace podle x, R1 rotace podle y, F0 převrátit zrcadlově,

B0 vykreslit pozadí. **Char x offset** – začátek vykreslování písmene na ose x (0-239), **Char y offset** – začátek vykreslování písmene na ose y (0-319), **Char color** – 8b barva písmene, **Char bg color** – 8b barva pozadí.

## 6.2 SPI – komunikace s MCU

Pro komunikaci s MCU byl použit SPI řadič s SPI dekodérem, přičemž oba jsou dostupné v svn repozitáři FITkitu. V FPGA je vždy jedna instance SPI řadiče a libovolné množství SPI dekodérů. SPI řadič má za úkol rozeznat, zda bude MCU komunikovat s FPGA nebo flash pamětí a pokud je rozpoznán přenos s FPGA, začne SPI řadič převádět komunikaci na interní sériovou sběrnici, která propojuje SPI řadič s SPI dekodéry. SPI dekodéry poté převádí sériovou komunikaci na paralelní o zadané šířce. Počet SPI dekodérů závisí na počtu komponent v FPGA, které potřebují komunikovat s MCU. Každý takový dekodér je určen adresovým prostorem, šířkou adresy a šířkou dat. Při přenosu je možné data zároveň číst i zapisovat. Přenos mezi MCU a FPGA by měl probíhat rychlostí 460,8 kB/s, sdílení SPI rozhraní flash pamětí s FPGA a možnost použití více dekodérů přináší jistou režii. Na zápis jednoho bytu dat tak potřebujeme přenést po SPI byty tři. První byte tvoří operační kód, který udává zda jsou data určena pro FPGA nebo flash pamět, dále jsou zde 2 bity informující SPI řadič, zda bude probíhat čtení a/nebo zápis. Druhý (případně více) bytů tvoří adresa pro určení dekodéru. Za ní už následují data pro komponentu v FPGA [12]. Jak je vidět na obrázku s příkazy řadiče 6.1, komunikace s řadičem displeje se skládá z 1 B indexu a 2 B dat. Pro přenos jednoho příkazu je tedy potřeba přenést z MCU 5 B. Pro vykreslení celé plochy displeje je potřeba přenést  $230 \times 240 = 76800$  příkazů, což nám dává teoreticky rychlost překreslení jedné obrazovky z MCU za 0,83 s. V praxi však dosáhneme překreslení celé obrazovky z MCU za necelé 3 s při jedné barvě (měřeno stopkami). Pokud použijeme barevný přechod, který se v MCU generuje, musíme si na výsledek překreslení celé obrazovky počkat přes 6 sekund (měřeno stopkami).

## 6.3 FTDI SPI – komunikace s PC

Komunikaci s PC zajišťuje integrovaný obvod FT2232D v módu MPSSE s rozhraním SPI (signály TCK, TDI, TDO, CS). Abychom mohli číst data s nástupnou hranou TCK, je dobré si tento externí signál navzorkovat. FPGA má totiž pouze omezený počet vstupů pro externí hodinové signály. Při vzorkování signálu musíme dát pozor na jeho maximální frekvenci, aby navzorkovaný signál odpovídal původnímu. Maximální frekvence je u signálu TCK 6 MHz (frekvence hlavního hodinového signálu CLK je 75 MHz). Spolu se signálem TCK vzorkujeme zároveň také signály TDI a CS, aby nedošlo k posunu hodnot vůči nástupné hraně sig. TCK. Při vytváření instance modulu obsluhující komunikaci s obvodem FTDI2232 je možné zvolit šířku dat jednoho příkazu. V ukázkové aplikaci je šířka nastavena na 16 bitů, kvůli rychlejšímu zobrazení obrázků z PC. K těmto 16-ti bitům je přidán kód příkazu A1, což je zápis do GRAM displeje. Jak již bylo uvedeno, přenos přes USB je realizován v blocích o velikosti 512 B, takže vykreslení celé obrazovky, pokud odesíláme jednotlivé příkazy, trvá velmi dlouho (více než 10 s). Proto je demonstrační aplikace pro PC použita pouze pro přenos obrázků. Obrázek přes celou plochu displeje je vykreslen za přibližně 800 ms (měřeno programově) při přenášení 2 B dat na 1 pixel.

### 6.3.1 FTDI software

Na straně PC je použita knihovna libFTDI (libusb) a libftdispi. Jedná se o multiplatformní opensource knihovny. Pro ověření funkčnosti komunikace PC – FPGA byla vytvořena ukázková aplikace, tzv. slideshow, kde jsou na displeji modulu zobrazovány obrázky z PC. Během testování se však několikrát objevila chyba, která se projevuje deformovaným obrázkem a zacyklením USB paketů. Chybu se zatím nepodařilo záměrně vyvolat a její odladění bude vyžadovat další testování. Alternativou může být komunikace pomocí FIFO režimu obvodu FTDI, který má poskytovat vyšší rychlost (1 MB/s), ale v době testování tohoto režimu jsem neměl k dispozici logický analyzátor pro ověření průběhů signálů.

## 6.4 DISPLAY CONTROL – hlavní řídicí komponenta

Do této komponenty přichází data z MCU a PC. Na základě indexu příkazu je rozhodnuto, zda se jedná o příkazy interního řadiče displeje a jsou tak odeslány na systémové rozhraní, nebo se jedná o příkazy pro řadič modulu v FPGA (obrázek 6.1). Přes DISPLAY CONTROL komponentu je vybíráno právě aktivní rozhraní (RGB/VSYNC/systémové rozhraní), probíhá zde zápis do paměti RAM a je odtud ovládána komponenta pro generování znaků. Dále je generován signál resetu displeje, ovládáno podsvětlení, šířka datové části sys. rozhraní a generován signál DOTCLK pro rozhraní RGB.

## 6.5 SYSTEM INTERFACE

Systémové rozhraní je implementováno stavovým automatem podle časového diagramu na obrázku 5.1. Většina příkazů je tvořena indexem registru a hodnotou, což odpovídá časovému diagramu na zmíněném obrázku. Je také implementována možnost samotného nastavení indexu registru a sekvenční zápis hodnot na systémovém rozhraní. To je potřeba pro správnou funkci RGB i VSYNC rozhraní, kde je při jejich zapnutí nutné nastavit pouze index registru pro zápis do GRAM.

## 6.6 RGB INTERFACE

RGB rozhraní je možné použít jak pro zobrazení statických obrázků, tak pro pohyblivou grafiku. Pomocí čítačů jsou jako násobky signálu DOTCLK vytvářeny signály HSYNC, VSYNC a ENABLE. Díky řádkové (HSYNC) a sloupcové (DOTCLK) synchronizaci víme přesnou pozici, na které se bude bod vykreslovat. Můžeme tak vytvářet v FPGA grafické objekty jako je v ukázkové aplikaci čtverec, který se pohybuje po obrazovce a neovlivňuje data vykreslovaná z RAM. Dalším využitím by mohl být například hardwarový kurzor. Jeden snímek je vykreslen za 10,5 ms, takže za 1 sekundu je vykresleno asi 95 snímků (měřeno v simulaci).

## 6.7 VSYNC INTERFACE

VSYNC je další rozhraní, které lze na modulu s barevným displejem využít. Díky jeho jednodušší implementaci je možné ho použít v aplikacích, kde je potřeba pouze zobrazovat kontinuálně data z RAM. Je možné tak oproti implementaci RGB rozhraní ušetřit zdroje



v FPGA. Při frekvenci hlavního hodinového signálu CLK 75 MHz je jeden celoobrazovkový snímek vykreslován 13,56 ms, to je asi 73 snímků za sekundu (měřeno v simulaci).

## 6.8 TOUCH CONTROL

Dotyková fólie je řízena obvodem TSC2046, s kterým komunikuje FPGA přes SPI. Při dotyku fólie obvod TSC2046 nastaví signál PENIRQ do log. 0, na který FPGA zareaguje vyvoláním měření pozice bodu dotyku. Jsou realizovány 2 přenosy – změření pozice na ose  $x$  a  $y$ . Hodnota bodu dotyku je uložena v registru FPGA a může být pomocí přerušení přenesena do MCU nebo PC (není použito). Ačkoliv je řadič obvodu TSC2046 úspěšně od-simulován, nemůže být prakticky vyzkoušen kvůli zřejmě polámanému kontaktu z dotykové fólie. Na desce plošných spojů se také zřejmě objevil zkrat na signálu DIN vedoucí k obvodu TSC2046. Obě závady by měly být opravitelné, avšak z nedostatku času na ně nedošlo.

## 6.9 TEXT GENERATOR

Komponenta pro generování textu využívá blokovou ROM, kde je uloženo 128 znaků, přičemž každý znak se skládá z  $8 \times 16$  bitů (bodů). Generovaný znak je možné otočit v ose  $x$ ,  $y$ , překlomit, nastavit velikost ve 4 úrovních:  $8 \times 16$ ,  $16 \times 32$ ,  $32 \times 64$  a  $64 \times 128$  bodů. Lze také nastavit barvu znaku a barvu pozadí, které může být volitelně vykresleno. Možnost vykreslovat pozadí znaku lze také využít při zaplnění obrazovky jednou barvou. Nastavíme barvu pozadí znaku na požadovanou barvu obrazovky a zvolíme znak mezery (index znaku je 0x0) s největší velikostí. Pro vyplnění obrazovky je potřeba vytisknout 10 znaků, takže překreslení celé obrazovky jednou barvou trvá asi 80 ms (podle zdrojového kódu). Body, které je potřeba pro zobrazení znaku vykreslit jsou vloženy do fronty pro zápis do SDRAM (WRITE FIFO). Znaková ROM byla převzata z knihy [8].

## 6.10 SDRAM, WRITE a READ FIFO

Paměť SDRAM je v řadiči použita jako framebuffer – paměť, ve které je skládán snímek pro vykreslení na displeji. Snímek z framebufferu může být vykreslen vícekrát a MCU či PC tak nemusí generovat každý snímek zvlášť. Pro implementaci byl použit řadič SDRAM paměti, který je dostupný v svn repozitáři FITkitu. Tento řadič je však určen pro čtení a zápis 8 bitových hodnot bez možnosti sekvenčního čtení/zápisu. Přečtení 16 bitové hodnoty pro jeden pixel proto trvá příliš dlouho. Sice byla vyzkoušena verze s 16-ti bitovým čtením a zápisem do RAM v RGB režimu s 67 snímky za sekundu, kdy je blikání blikání obrazu znatelné jen při podrobném zkoumání ale ve VSYNC režimu bylo při 53 snímcích za sekundu blikání viditelné (maximální dosažené rychlosti překreslení při rychlosti 75 MHz hlavního hodinového signálu CLK). V RGB a VSYNC režimu je proto použita pouze 8 bitová barevná hloubka pro jeden pixel. Protože je potřeba do RAM paměti zapisovat i číst, což není možné provádět zároveň, byly vytvořeny 2 fronty s blokovými RAM. Ze zápisové fronty se zapisuje do RAM, když zrovna není obsazena jinou operací. Při obsazené RAM se zapisují data do fronty s hloubkou 1024 položek, přičemž každá položka se skládá z 17 bitů adresy a 8 bitů dat. Čtení ze zápisové fronty probíhá asynchronně. Druhá fronta slouží pro čtení z RAM a přenos obrazových dat na rozhraní RGB a VSYNC. Fronta má hloubku 2048 položek s délkou 8 bitů, používá synchronní čtení a je asynchronně resetována při vypnutí rozhraní RGB nebo VSYNC. Zápisová fronta je tvořena dvěma blokovými RAM o kapacitě

2×18 kb, čtecí používá jednu blokovou RAM o kapacitě 18 kb. Obě fronty byly vygenerovány v programu Xilinx ISE za použití nástroje Xilinx CORE Generator.

## 6.11 Software na PC

Jak již bylo uvedeno v části implementace FTDI software, pro komunikaci s FPGA na FITkitu byla použita knihovna libFTDI a libftdispi. Pro načítání obrázků byla použita opensource knihovna FreeImage. Ta umožňuje načítat nejpoužívanější formáty obrázků a pracovat s nimi. V aplikaci jsou načítány obrázky formátu PNG s rozměry 240 × 320 pixelů a je u nich upravena barevná hloubka na 16 bitů. Poté je obrázek načten do pole, které je po blocích odesláno a zobrazeno na displeji. Program doplňuje shellový skript a dohromady vytváří tzv. slideshow aplikaci. Pro spuštění aplikace je potřeba mít rozpojenou propojku J8 a po ukončení aplikace je nutné restartovat FTDI převodník, buď propojkou J7 nebo vytažením USB kabelu.

## 6.12 Software v MCU

V MCU je implementována inicializace displeje, přepínání grafických režimů a několik příkazů ukazující možnosti řadiče modulu (ukázková aplikace).

## 6.13 Využití zdrojů FPGA

Výsledný design se do FPGA XC3S50 vešel docela těsně. Při použití modulu v jiném projektu bude nutné použít pouze nezbytně nutné komponenty nebo použít FPGA s vyšší kapacitou. Další možnosti může být optimalizace VHDL kódu.

### 6.13.1 Výsledky syntézy řadiče

Number of BUFGMUXs	6 out of 8	75%
Number of DCMs	1 out of 2	50%
Number of External IOBs	90 out of 124	72%
Number of LOCed IOBs	90 out of 90	100%
Number of RAMB16s	4 out of 4	100%
Number of Slices	766 out of 768	99%
Number of SLICEMs	2 out of 384	1%

## Kapitola 7

# Závěr

Výsledkem práce je hardwarový modul rozšiřující FITkit o barevný displej. Na displeji je možné zobrazovat informace z mikrokontroléru, FPGA i počítače. Pro plnohodnotné využití displeje je však omezujícím faktorem rychlost přenosu obrazových dat. Ale i tak může displej zjednodušit ovládání a použití FITkitu jako samostatného systému. Po připojení dotykové vrstvy displeje se možnosti použití ještě zvýší.

Při práci na projektu jsem se seznámil s vývojem, simulací a laděním aplikací pro FPGA. Cennou zkušeností pro mě byl také návrh, výroba a ladění desky plošných spojů. Při návrhu byl kladen důraz na další využitelnost, jak desky modulu, tak komponent řadiče, který je do jisté míry univerzální – například by mělo být možné připojit displej s vyšším rozlišením a podobným ovládáním.

Vylepšením projektu může být vytvoření řadiče SDRAM paměti, který bude podporovat sekvenční zápis dvou 8 bitových hodnot, aby mohla být v RGB a VSYNC režimu využita 16 bitová barevná hloubka. Dále může být vylepšena komunikace s PC využitím FIFO režimu USB převodníku FTDI2232D, ten by měl umožňovat přenosovou rychlost až 1 MB/s. Pro zrychlení komunikace mikrokontroléru s FPGA je možné vytvořit paletu barev. Další možností je komprese přenosu dat, například zadání počtu opakování zvolené barvy. Je také možné zvýšit počet framebufferů a přepínat mezi nimi. Uživatel by tak neviděl, jak se obrázek postupně načítá, protože by po přepnutí framebufferu bylo zobrazení obrázku okamžité.

Pro další vývoj projektu bude vhodné použít FITkit osazený větším programovatelným hradlovým polem XC3S400. Možné pokračování projektu je například implementace soft procesoru v FPGA, na kterém bude implementováno generování grafických primitiv a jednoduchý 3D engine. Jinou možností je implementace hry na displeji, případně portování hry určené pro operační systém DOS. S využitím již dokončených projektů lze například vytvořit osciloskop s logickým analyzátozem, který bude zobrazovat průběhy signálů na displeji nebo zobrazování obrázků a videí z microSD karty.

# Literatura

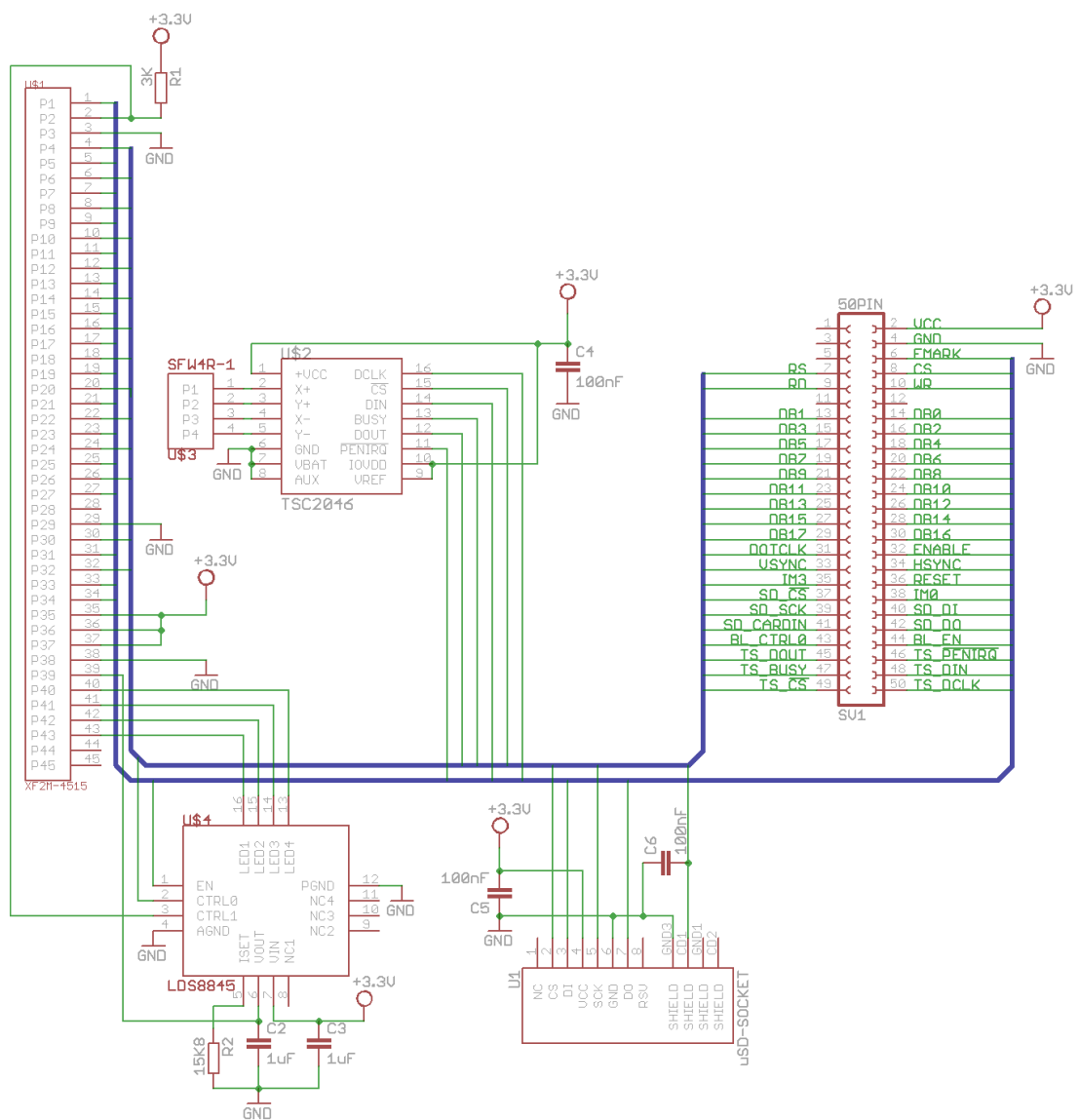
- [1] Datasheet ILI9320 v0.55 [online].  
[http://www.displaytech-us.com/pdf/application/TFT\\_DriverIC/ILI9320DS\\_V0.55.pdf](http://www.displaytech-us.com/pdf/application/TFT_DriverIC/ILI9320DS_V0.55.pdf), [cit. 2011-04-25].
- [2] Xilinx web: Getting started [online].  
<http://www.xilinx.com/company/gettingstarted/index.htm>, [cit. 2011-04-29].
- [3] Using FPGAs to Render Graphics and Drive LCD Interfaces [online].  
<http://www.altera.com/literature/wp/wp-01100-graphic-lcd-display.pdf>, [cit. 2011-05-06].
- [4] Datasheet DT028TFT-TS v. 5.0 [online].  
<http://www.mouser.com/catalog/specsheets/sdt028tft-v50.pdf>, [cit. 2011-05-08].
- [5] Datasheet TSC2046 rev. G [online]. <http://www.ti.com/lit/gpn/tsc2046>, [cit. 2011-05-08].
- [6] den Boer, W.: *Active Matrix Liquid Crystal Displays*. Newnes, 2005, ISBN 0750678135, 265 s.
- [7] Brenner, N.; Sullivan, S.: 4-Wire and 8-Wire Resistive Touch-Screen Controller Using the MSP430 (Rev. A) [online]. <http://www.ti.com/litv/pdf/sl3aa384a>, [cit. 2011-05-08].
- [8] Chu, P. P.: *FPGA Prototyping by VHDL Examples: Xilinx Spartan-3 Version*. Wiley, 2008, ISBN 9780470185315, 468 s.
- [9] Cristaldi, D. J.; Pennisi, S.; Pulvirenti, F.: *Liquid Crystal Display Drivers*. Springer Netherlands, 2009, ISBN 978-90-481-2255-4, 295 s.
- [10] Downs, R.: Using resistive touch screens for human/machine interface [online].  
<http://focus.ti.com/lit/an/slyt209a/slyt209a.pdf>, [cit. 2011-05-12].
- [11] Ge, Z.; Wu, S.-T.: *Liquid Crystal Display Drivers*. Springer Netherlands, 2010, ISBN 978-0-470-74373-7, 246 s.
- [12] Vašíček, Z.: Internetové stránky platformy FITkit [online]. [www.fit.vutbr.cz/kit](http://www.fit.vutbr.cz/kit), [cit. 2011-04-28].

# Seznam použitých zkratek

**A/D** – analog-digital  
**a-Si** – amorphous Silicon  
**BP, FP** – Back Porch, Front Porch  
**CD** – Compact Disk  
**CLB** – Configurable Logic Block  
**CRT** – Cathode Ray Tube  
**DPS** – deska plošných spojů  
**DRAM** – Dynamic Random Access Memory  
**FIFO** – First In First Out  
**FPGA** – Field Programmable Gate Array  
**GRAM** – Graphics Random Access Memory  
**I<sup>2</sup>C** – Inter-Integrated Circuit  
**IPS** – In Plane Switching  
**ITO** – Indium Tin Oxide  
**JTAG** – Joint Test Action Group  
**LCD** – Liquid Crystal Display  
**MCU** – Microcontroller Unit  
**MP3** – MPEG (Moving Picture Experts Group) layer III audio  
**MPSSE** – Multi-Protocol Synchronous Serial Engine  
**PC** – Personal Computer  
**poly-Si** – poly-crystalline Silicon  
**PWM** – Pulse Width Modulation  
**ROM** – Read Only Memory  
**SDRAM** – Synchronous Dynamic Random Access Memory  
**SPI** – Serial Peripheral Interface  
**TFT** – Thin Film Transistor  
**TN** – Twisted Nematic  
**USB** – Universal Serial Bus  
**VA** – Vertical Alignment  
**VGA** – Video Graphics Array  
**VHDL** – Very High Speed Integrated Circuit Hardware Description Language

# Dodatek A

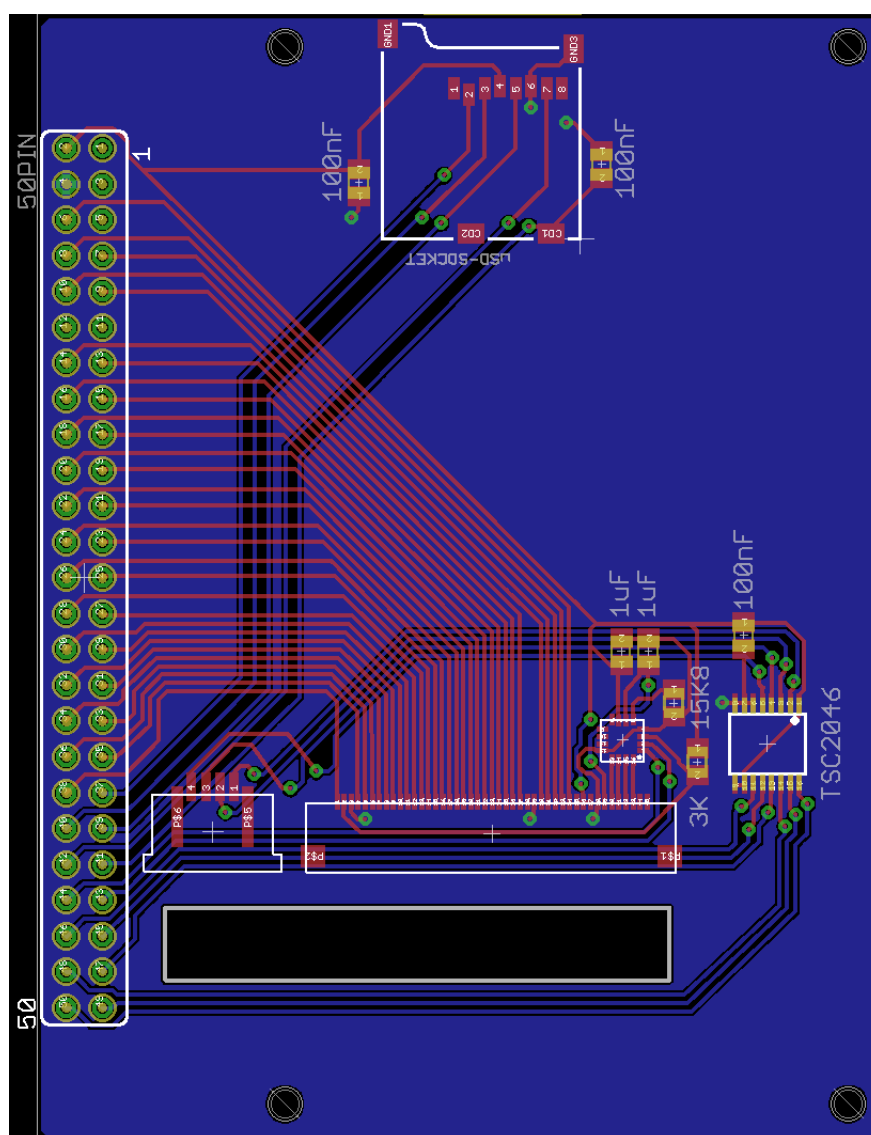
## Schéma modulu



Obrázek A.1: Schéma modulu.

## Dodatek B

# Deska plošných spojů



Obrázek B.1: Deska plošných spojů.

## Dodatek C

### Obsah CD

Soubor	Popis
datasheets.zip	archiv použitých datasheetů
app-qdevkit.zip	vlastní implementace, s adresářovou strukturou pro program QDevKit
doc.zip	zdrojové soubory této technické zprávy
eagle.zip	podklady pro výrobu desky plošných spojů, obsahuje knihovnu součástek
LICENCE	licenční podmínky
README	uživatelská příručka