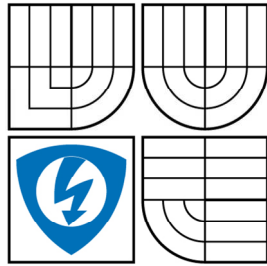


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A  
KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF CONTROL AND INSTRUMENTATION

# INTELIGENTNÍ KAMERA

AN INTELLIGENT CAMERA SYSTEM

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

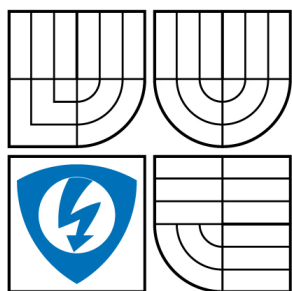
AUTOR PRÁCE  
AUTHOR

Bc. FRANTIŠEK GOGOL

VEDOUCÍ PRÁCE  
SUPERVISOR

Ing. SOBĚSLAV VALACH

BRNO 2008



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav automatizace a měřicí  
techniky

## Diplomová práce

magisterský navazující studijní obor  
Kybernetika, automatizace a měření

**Student:** Gogol František Bc.

**ID:** 89468

**Ročník:** 2

**Akademický rok:** 2007/2008

**NÁZEV TÉMATU:**

**Inteligentní kamera**

**POKYNY PRO VYPRACOVÁNÍ:**

Navrhňte systém inteligentní kamery využívající pro svoji činnost embedded strukturu tvořenou v hradlovém poli. K základní struktuře připojte CMOS snímací element. Dále navrhňte vhodnou metodu komunikace se zařízením, vhodný kanál pro přenos obrazových dat a uživatelskou aplikaci, která bude demonstrovat funkci celého zařízení.

**DOPORUČENÁ LITERATURA:**

Firemní literatura Xilinx, Micron, Texas Instruments

**Termín zadání:** 3.12.2007

**Termín odevzdání:** 26.5.2008

**Vedoucí práce:** Ing. Soběslav Valach

**prof. Ing. Pavel Jura, CSc.**

*předseda oborové rady*

**UPOZORNĚNÍ:**

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

**Vysoké učení technické v Brně**  
**Fakulta elektrotechniky a komunikačních technologií**  
**Ústav automatizace a měřicí techniky**

## **Inteligentní kamera**

Diplomová práce

Specializace studia:           Kybernetika, automatizace a měření  
Student:                        Bc. František Gogol  
Vedoucí:                        Ing. Soběslav Valach

### **Abstrakt :**

Inteligentní kamerou se rozumí kamera doplněná o řídicí, vyhodnocovací popřípadě komunikační a zobrazovací jednotku, nejčastěji tvořenou jako embedded systém. Data získaná kamerou se ihned požadovaným způsobem zpracují a výstupem je zpracovaný obraz, popřípadě jen informace o sledovaném objektu.

Tato diplomová práce se zabývá hardwarovou realizací inteligentní kamery do hradlového pole typu FPGA. Implementována architektura se skládá z řadiče kamery, řadiče paměti, řadiče IIC, VGA řadiče a vlastní výkonné jednotky. Řadič kamery obstarává komunikaci s CMOS senzorem. Řadič paměti komunikuje s DDR SDRAM pamětí. Řadič IIC tvoří rozhraní mezi PLB sběrnicí a IIC sběrnicí. VGA řadič čte data z paměti a posílá je na standardní VGA rozhraní (640x480, 60 Hz). Výkonná jednotka vyčítá obrazová data z paměti a postupně pixel po pixelu nad nimi provede požadovanou operaci. Veškeré funkce jsou řešeny hardwarově. Jednotlivé komponenty kamery byly napsány ve VHDL a Verilog jazyku.

**Klíčová slova :** FPGA, zpracování obrazu, PPC, CMOS senzory

**Brno University of Technology**  
**The Faculty of Electrical Engineering and Communication**  
**Department of Control, Measurement and Instrumentation**

## **An Intelligent camera system**

Master's Thesis

Specialisation of study:      Cybernetics, Control and Measurement  
Student:                              Bc. František Gogol  
Supervisor:                         Ing. Soběslav Valach

### **Abstract :**

An intelligent camera includes a processor, which can extract information from images without the need for an external processing unit, and interface devices used to make the results available to other devices.

This paper describes the intelligent camera design and implementation into the Field Programmable Gate Array (FPGA). The implemented architecture contains a camera controller, a memory controller, an IIC controller, a VGA controller, and an execution unit. The camera controller communicates with a CMOS chip. The memory controller communicates with a DDR SDRAM memory. The IIC controller is the interface between a PLB bus and an IIC bus. The VGA controller takes data from the memory and transform them into the VGA format (640x480, 60 Hz). The execution unit extracts the image data from the memory. These data are processed by hardware pixel by pixel, which results in a modified image. The camera units has been implemented in the VHDL and Verilog languages.

**Key words :** FPGA, Image processing, PPC, DSP, CMOS sensors

## **Bibliografická citace**

GOGOL, F. Inteligentní kamera. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 79 stran, přílohy 4. Vedoucí diplomové práce Ing. Soběslav Valach.

## **P r o h l á š e n í**

„Prohlašuji, že svou diplomovou práci na téma Inteligentní kamera jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem úmyslně neporušil autorská práva třetích osob, zejména jsem úmyslně nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.“

V Brně dne :

Podpis:

## **P o d ě k o v á n í**

Děkuji vedoucímu diplomové práce Ing. Soběslavu Valachovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce. Dále děkuji rodině za podporu při studiu na vysoké škole a kamarádům za vytvoření dobrých podmínek ke studiu.

## **OBSAH**

<b>1. ÚVOD .....</b>	<b>12</b>
<b>2. KAMERY .....</b>	<b>14</b>
2.1 Charakteristika .....	14
2.1.1 Blok snímání a digitalizace.....	14
2.1.2 Blok zpracování obrazu .....	15
2.1.3 Blok vstupů a výstupů .....	15
2.2 ROZDĚLENÍ KAMER.....	16
2.3 TECHNOLOGIE SNÍMÁNÍ .....	16
2.3.1 Technologie CCD .....	16
2.3.2 Technologie CMOS .....	19
2.3.3 CCD verzus CMOS .....	20
2.4 BAREVNÉ KÓDOVÁNÍ .....	21
2.4.1 Aditivní – RGB.....	21
2.4.2 Subtraktivní – CMY .....	22
2.5 SNÍMÁNÍ JEDNOTLIVÝCH BAREVNÝCH SLOŽEK .....	22
2.5.1 Separátní filtry na nezávislých snímačích .....	22
2.5.2 Bayer mask .....	23
<b>3. ZPRACOVÁNÍ OBRAZU.....</b>	<b>24</b>
3.1 Filtry s nízkou propustností.....	25
3.1.1 Průměrování.....	25
3.1.2 Medián .....	25
3.2 Filtry s vysokou propustností.....	26
3.2.1 Operátor invariantní vůči rotaci.....	26
3.2.2 Operátory neinvariantní vůči rotaci .....	26
<b>4. VGA ROZHRANÍ .....</b>	<b>27</b>
4.1 VGA protokol .....	27
4.2 VGA SIGNÁLY .....	30
4.2.1 Vertikální synchronizace .....	30
4.2.2 Horizontální synchronizace .....	30
4.2.3 RGB signály .....	30



4.2.4 Aktivní oblast.....	31
4.2.5 Front porch.....	31
4.2.6 Back porch.....	31
4.2.7 Doba zatemnění .....	31
<b>5. ML 403 DEVELOPMENT BOARD.....</b>	<b>32</b>
5.1 HRADLOVÉ POLE FPGA .....	33
5.1.1 IOB .....	34
5.1.2 CLB.....	34
5.1.3 BRAM .....	35
5.1.4 DCM .....	35
5.1.5 PMCD .....	36
5.1.6 XtremeDSP .....	36
5.1.7 Clock resources.....	37
5.2 PROCESOROVÉ JÁDRO POWERPC 405.....	37
5.2.1 Charakteristika.....	37
5.2.2 Architektura .....	38
5.2.3 Instrukční sada.....	39
5.2.4 Rozhraní procesoru.....	40
5.2.5 On-chip Memory (OCM).....	40
5.2.6 Processor Local Bus (PLB) .....	40
5.2.7 On-chip Peripheral Bus (OPB).....	42
5.2.8 Device Control Register Bus (DCR) .....	42
5.3 XILINX EDK.....	43
5.3.1 MHS.....	44
5.3.2 MSS .....	45
5.3.3 UCF .....	45
5.3.4 Tvorba uživatelských periférií.....	46
<b>6. OBVODOVÁ REALIZACE KAMERY .....</b>	<b>48</b>
6.1 PLATFORMA .....	48
6.1.1 Snímací element.....	48
6.1.2 Propojovací modul.....	49

6.2 ARCHITEKTURA V FPGA .....	51
6.2.1 IIC CONTROLLER.....	52
6.2.2 CAMERA CONTROLLER.....	58
6.2.3 VGA CONTROLLER .....	62
6.2.4 MEMORY CONTROLLER.....	65
6.2.5 EXECUTION UNIT .....	65
6.2.6 Konfigurace a nastavení architektury .....	70
<b>7. SHRUTÍ A DOSAŽENÉ VÝSLEDKY.....</b>	<b>72</b>
<b>8. ZÁVĚR .....</b>	<b>76</b>
<b>9. POUŽITÁ LITERATURA .....</b>	<b>77</b>

## SEZNAM OBRÁZKŮ

Obrázek 2.1: Blokové schéma inteligentní kamery .....	14
Obrázek 2.2: Princip CCD čipu .....	17
Obrázek 2.3: Full Frame CCD .....	17
Obrázek 2.4: Interline - Transfer CCD .....	18
Obrázek 2.5: Frame - Transfer CCD .....	18
Obrázek 2.6: PPS - Passive Pixel Sensor .....	20
Obrázek 2.7: APS - Active Pixel Sensor .....	20
Obrázek 2.8: Model RGB .....	21
Obrázek 2.9: Model CMY .....	22
Obrázek 2.10: Separátní filtry .....	22
Obrázek 2.11: Maska aditivního modelu .....	23
Obrázek 2.12: Maska subtraktivního modelu .....	23
Obrázek 3.1: Princip filtrace jednobarevného obrazu .....	26
Obrázek 4.1: Časování VGA signálů .....	29
Obrázek 4.2: Generování svislých barevných pruhů na obrazovce monitoru .....	29
Obrázek 5.1: Blokové schéma vývojového kitu ML 403 .....	33
Obrázek 5.2: Struktura CLB bloku .....	34
Obrázek 5.3: Struktura BRAM paměti .....	35
Obrázek 5.4: DCM blok .....	36
Obrázek 5.5: PMCD blok .....	36
Obrázek 5.6: DSP48 SLICE buňka .....	37
Obrázek 5.7: Procesorové jádro PowerPc 405 .....	38
Obrázek 5.8: Čtení přes sběrnici PLB .....	42
Obrázek 5.9: Komunikace přes sběrnici DCR .....	43
Obrázek 6.1: Schéma kamery .....	48
Obrázek 6.2: Simulační schéma v programu HyperLynx .....	49
Obrázek 6.3: Simulace signálu na vodiči bez odporu .....	50
Obrázek 6.4: Simulace signálu na vodiči s odporem .....	50
Obrázek 6.5: Skutečný průběh signálu na vodiči .....	51
Obrázek 6.6: Architektura kamery v čipu FPGA .....	52

Obrázek 6.7: Architektura IIC řadiče.....	54
Obrázek 6.8: Zápis přes IIC sběrnici .....	54
Obrázek 6.9: Čtení přes IIC sběrnici.....	55
Obrázek 6.10: Vývojový diagram řadiče přenosu .....	56
Obrázek 6.11: Vývojový diagram řadiče bajtů .....	57
Obrázek 6.12: Realizace obousměrných pinů.....	58
Obrázek 6.13: Realizace zápisu dat do fifa.....	59
Obrázek 6.14: Realizace zápisu dat do paměti přes XCL sběrnici .....	60
Obrázek 6.15: Závislost doby přenosu 64 bytů v závislosti na délce burstu .....	62
Obrázek 6.16: Zápis přes XCL sběrnici .....	62
Obrázek 6.17: Schéma VGA řadiče .....	63
Obrázek 6.18: Rozhraní poskytnuté bridgem.....	65
Obrázek 6.19: Schéma výkonné jednotky.....	66
Obrázek 6.20: Vývojový diagram řadiče filtru .....	68

## SEZNAM TABULEK

Tabulka 6.1: Vysílací registr IIC řadiče .....	55
Tabulka 6.2: Přijímací registr IIC řadiče.....	55
Tabulka 6.3: Příznakový registr IIC řadiče .....	55
Tabulka 6.4: Adresový registr řadiče kamery .....	60
Tabulka 6.5: Řídicí registr řadiče kamery .....	60
Tabulka 6.6: Uložení dat v paměti .....	61
Tabulka 6.7: Hodnoty VGA signálů .....	63
Tabulka 6.8: Adresový registr VGA .....	63
Tabulka 6.9: Řídicí registr VGA.....	64
Tabulka 6.10: Adresový registr zápisu.....	69
Tabulka 6.11: Adresový registr čtení .....	69
Tabulka 6.12: Řídicí registr výkonné jednotky .....	69
Tabulka 6.13: Offset pro jednotlivé barevné složky .....	69
Tabulka 6.14: Koeficienty filtrační masky .....	69
Tabulka 6.15: Adresový offset jednotlivých koeficientů masky .....	70
Tabulka 6.16: Přiřazení signálů jednotlivým pinům .....	71
Tabulka 7.1: Spotřeba systémových zdrojů pro jednotlivé komponenty.....	74
Tabulka 7.2: Porovnání zaplnění kompletním designem pro různá pole.....	75

## 1. ÚVOD

Počítače a počítačové technologie jsou v posledních desetiletích nepochybně nejrychleji rozvíjejícím se oborem lidské činnosti. Dnes je více počítačů v kancelářích, domácnostech než je lidí, kteří s nimi pracují. Mnoho z těchto počítačů a počítačových systémů je právě vestavěno do zařízení, ve kterých si jejich existenci zpravidla ani neuvědomujeme. Vestavěné systémy (embedded systems) jsou zabudovány v celé řadě přístrojů od domácích spotřebičů až po řídicí, monitorovací techniku průmyslových systémů, včetně inteligentních kamer.

Inteligentní kamerou se rozumí kamera doplněná o řídicí, vyhodnocovací popřípadě komunikační a zobrazovací jednotku, nejčastěji tvořenou jako embedded systém. Data získaná kamerou se ihned požadovaným způsobem zpracují a výstupem je zpracovaný obraz, popřípadě jen informace o sledovaném objektu. Tyto vlastnosti předurčují inteligentní kameru k použití při různých výrobních procesech, kde může plnit funkci automatické kontroly výrobků (např. povrchových vad, chybějících částí), bezkontaktních měření objektů nebo čtení čárových kódů.

Tato diplomová práce se zabývá hardwarovou realizací inteligentní kamery do hradlového pole typu FPGA. Z důvodu snadné dostupnosti byl za základ platformy zvolen vývojový kit ML 403 firmy XILINX. Tento kit kromě programovatelné logiky obsahuje i jeden procesor PowerPC, jež je použit k řízení jednotlivých implementovaných řadičů kamery.

V první kapitole je popsána struktura dnešních inteligentních kamer, její jednotlivé části a jakou funkcí obstarávají. Dále je zde popsáno rozdělení kamer a princip snímání digitálního obrazu. Jsou zde popsány a srovnány v dnešní době nejčastěji používané technologie snímání obrazu pomocí CCD a CMOS technologie, princip a reprezentace barevného obrazu. V následující kapitole je popsán princip filtrace obrazu a proč se používá. V další kapitole je popsáno VGA rozhraní, jež se nejčastěji používá k zobrazení obrazu na standardním monitoru, a kompletní popis signálů. V následující kapitole je stručně popsán vývojový kit ML 403. Dál je zde popsána architektura FPGA obvodu, architektura hardcore procesoru PowerPC 405 a jeho periferie jimiž může komunikovat s okolím. Na závěr kapitoly je stručně

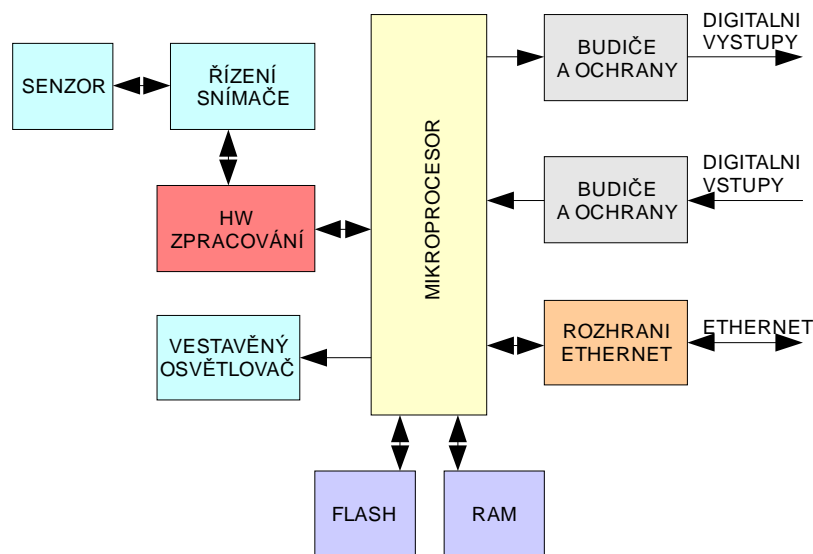
popsáno vývojové prostředí XILINX EDK, v němž byla celá práce realizována. Poslední kapitola popisuje hardwarovou realizaci kamery. Je zde popsána funkce jednotlivých navrhnutých a naprogramovaných komponent kamery, včetně důvodů proč bylo zvoleno dané řešení.

## 2. KAMERY

### 2.1 CHARAKTERISTIKA

Inteligentní kamery jsou odpovědí na požadavek přiblížit komplikované a drahé systémy strojového vidění technice běžných senzorů, např. optických. Vznik inteligentní kamery bylo umožněno díky spojení dvou prvků: polovodičového snímače obrazu a mikroprocesoru. Obecné blokové schéma inteligentní kamery je znázorněno na obrázku 2.1. Inteligentní kamera je složena z několika základních bloků:

- Blok snímání a digitalizace
- Blok zpracování obrazu
- Blok vstupů a výstupů



**Obrázek 2.1:** Blokové schéma inteligentní kamery

#### 2.1.1 Blok snímání a digitalizace

Nejdůležitější součástí kamery je polovodičový snímač - senzor, který přímo ovlivňuje kvalitu obrazu. Senzor, který může mít různou strukturu, rozlišení, citlivost a výrobní technologii viz. kapitola 2.3, převádí obvykle dopadající světlo na náboj, a ten se následně převádí na elektrický signál, který je dále upravován v kameře.



Nejčastěji dosud používaným senzorem je CCD (Complementary Charge Device). Poskytuje v principu analogový signál, který musí být digitalizován převodníkem A/D. Celý proces snímání bývá řízen u jednodušších kamer přímo výpočetní jednotkou, u výkonnějších kamer jsou snímání a digitalizace řízeny zvláštním mikropočítačem nebo hardwarovou řídicí jednotkou, která je z výpočetní části pouze synchronizována.

V poslední době došlo ke značnému pokroku v technologii senzorů CMOS (Complementary Metal Oxide Semiconductor), které jsou již schopny poskytnout obraz kvality srovnatelné se senzorem CCD. Technologie CMOS navíc umožňuje integrovat na snímací čip i převodník A/D a většinu řídicích obvodů. Obsluha senzoru CMOS je podstatně jednodušší, podobá se vybírání dat z dynamické paměti RAM.

### **2.1.2 Blok zpracování obrazu**

Zpracování obrazu vyžaduje rychlé zpracování velkého množství dat, proto se nejčastěji používají výkonné mikroprocesory a vzhledem k charakteru výpočtu často i digitální signálové procesory (DSP). K urychlení některých standardních operací potřebných při analýze obrazu, např. kódování, komprese, vyhledávání a rozpoznávání objektů v obraze, se využívají i velmi rychlé jednoúčelové konečné automaty programované v hradlových polích.

### **2.1.3 Blok vstupů a výstupů**

Právě digitální výstupy přibližují inteligentní kameru běžnému senzoru. Proto bývají i ve stejném provedení – s otevřeným kolektorem typu PNP nebo NPN, s pracovním napětím do 30 V a ochranami proti přepólování a poškození napětíovou špičkou při spínání indukční zátěže. Digitální vstupy jsou ve většině aplikací nutné pro synchronizaci sejmutí snímku se stavem procesu. Prohlížený objekt bývá nutné sejmout v určité poloze a kamera se spouští například pomocným přibližovacím čidlem nebo signálem z řídicího systému. Podobně jako standardní přibližovací senzory bývají inteligentní kamery aplikovány lokálně, na zařízení s jedním napájením, kde nehrozí velké rozdíly potenciálu. Vyrábějí se i kamery se vstupy a výstupy v úrovních signálu TTL (Tranzistor Tranzistor Logic), které pro připojení do

průmyslové aplikace potřebují ještě galvanicky oddělený převodník úrovní (většinou s externím napájením).

## 2.2 ROZDĚLENÍ KAMER

Dle technologie snímání obrazu:

- CCD
- CMOS

Dle struktury:

- Řádkové
- Plošné

Dle systému barevného kódování:

- Aditivní barevný model
- Subtraktivní barevný model

Dle principu snímání jednotlivých barevných složek:

- Použití separátních filtrů na nezávislých snímačích
- Implementace barevného filtru přímo na snímač - Bayer mask

Dle struktury:

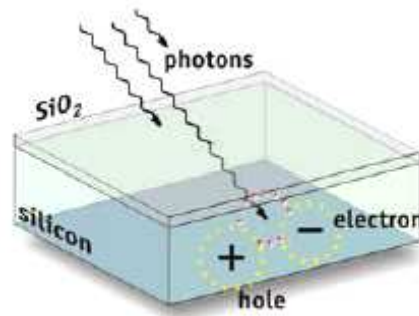
- Čtvercové
- Hexagonální
- Jiné

## 2.3 TECHNOLOGIE SNÍMÁNÍ

### 2.3.1 Technologie CCD

Zkratka CCD pochází z anglického Charge-Coupled Device, což v překladu znamená zařízení s vázanými náboji. CCD využívá podobně jako všechny ostatní světlocitlivé součástky fyzikálního jevu známého jako fotoefekt. Tento jev spočívá v tom, že částice světla foton při nárazu do atomu dokáže přemístit některý z jeho elektronů ze základního do tzv. excitovaného stavu. V polovodiči se takto uvolněný elektron může podílet na elektrické vodivosti respektive je možno ho z polovodiče odvést pomocí přiložených elektrod. U CCD je ovšem elektroda od polovodiče

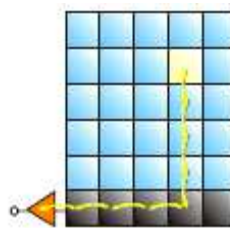
izolována tenoučkou vrstvičkou kysličníku křemičitého  $\text{SiO}_2$ , který se chová jako dokonalý izolant, takže fotoefektem uvolněné elektrony nemohou být odvedeny pryč.



**Obrázek 2.2:** Princip CCD čipu

Po expozici jednotlivých pixelů obrazového senzoru dochází k transportu generovaného elektrického náboje strukturou do A/D převodníku, kde dochází k digitalizaci odpovídajícího elektrického signálu. Používají se čtyři odlišné principy:

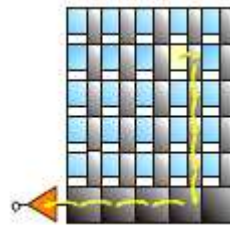
- s mechanickou uzávěrkou (Full Frame CCD)
- s meziřádkovým přenosem náboje (Interline-Transfer CCD)
- s přenosem snímků (Frame - Transfer CCD)
- s přímou adresací pixelů (Charge Injection Devices – CID)



**Obrázek 2.3:** Full Frame CCD

**Systémy s mechanickou uzávěrkou** mají velmi jednoduchou architekturu. Využívají stejné obrazové pole pro expozici fotonem, integraci náboje i jeho přenos. K řízení expozice a blokování dopadu světla na senzor během čtení je použita mechanická uzávěrka. Pixely jsou zpravidla čtvercové, takže nedochází ke zkreslení obrazu při detekci. Senzory je možné osvětlovat jak z čelní tak i zadní strany.

Zařízení s osvětlením se zadní strany mají větší kvantovou účinnost, protože není třeba, aby se světlo šířilo přes celou strukturu fotocitlivé vrstvy detektoru.



**Obrázek 2.4:** Interline - Transfer CCD

**Systémy s meziřádkovým přenosem** mají paralelní registr rozdělen na pruhy tak, že neprůhledný paměťový registr se nachází mezi sloupci pixelů. Elektronický obraz se akumuluje na exponované ploše paralelního registru. Během čtení CCD se celý obraz přesune prostřednictvím meziřádkové masky do skrytého posuvného registru. CCD s meziřádkovým přenosem vykazují relativně špatnou citlivost, protože velká část každého pixelu je pokryta neprůhlednou maskou. Ke zvýšení kvantové účinnosti lze použít mikročochky, které soustředí světlo z velké plochy na fotodiody.



**Obrázek 2.5:** Frame - Transfer CCD

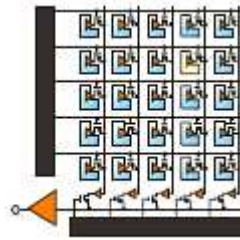
**Systémy s přenosem snímků** mají oddělené obrazové a paměťové pole. Zaznamenaný latentní obraz je nejprve přesunut z obrazového pole do paměťového pole a během další expozice obrazového pole rychle přesunut ( $\sim 300 \mu\text{s}$ ) k okraji

senzoru, kde je digitalizován. Paměťové pole má obvykle stejnou velikost jako obrazové pole a je odděleno neprůhlednou maskou jako štít proti osvětlení pixelů. Tento typ CCD umožňuje přenášet obraz spojitě bez uzávěrky a to vysokou přenosovou rychlostí. CCD s přenosem snímků mohou být také užity ve spojení s mechanickou uzávěrkou k rychlému zaznamenání dvou po sobě následujících snímků. Toho lze využít např. při záznamu dvou obrazů s rozdílnou excitační nebo emisní vlnovou délkou při některých obrazových experimentech.

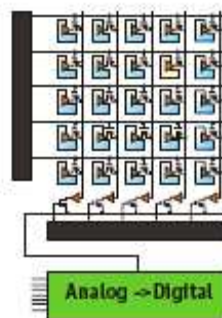
**Systémy s přímou adresací pixelů** umožňují přímý přístup k jednotlivým pixelům senzoru. Elektrický náboj je generován v pixelech sestávajících ze dvou kolmých elektrod (MOS tranzistorů), které slouží k uložení i přenosu fotogenerovaného náboje. Tento princip, využívaný jak v CCD, tak i v CMOS senzorech, umožňuje destruktivní (elektrický náboj je injektován do kontaktu) nebo nedestruktivní (náboj zůstává v pixelu a je možné provádět další expozici) čtení obrazu.

### 2.3.2 Technologie CMOS

Snímače typu CMOS (Complementary Metal Oxid Semiconductor) využívají polovodičové součástky řízené elektrickým polem a k provozu jim stačí jen jedno napájecí napětí. Proto je jejich spotřeba velmi malá. Pracují na principu fotodiody. Měří se protékající proud v okamžiku vyčítání. Tyto snímače se dělí na dva druhy. Prvním jsou tzv. pasivní CMOS (PPS - Passive Pixel Sensors), které generují elektrický náboj úměrný energii dopadajícího svazku světelných paprsků. Náboj pak jde přes zesilovač do A/D převodníku, stejně jako u CCD. V praxi však pasivní CMOS dávají díky šumu špatný obraz. Druhým typem jsou aktivní CMOS (APS - Active Pixel Sensors), u nichž je každý světlocitlivý element doplněn analytickým obvodem, který měří šum a eliminuje ho.



**Obrázek 2.6:** PPS - Passive Pixel Sensor



**Obrázek 2.7:** APS - Active Pixel Sensor

### 2.3.3 CCD verzus CMOS

Výroba CCD senzorů je poměrně drahá a časově náročná, protože vlastní záznamové prvky jsou vyráběny odlišnou technologií než ostatní podpůrné obvody (posuvné registry, A/D převodníky, apod.) a je třeba zabezpečit transport náboje přes čip bez zkrácení. Na druhé straně však dosahují vynikajících parametrů – vytvářejí velmi kvalitní nízkošumový obraz (hladinu teplotního šumu lze snížit jednak chlazením senzorů, příp. sdružováním pixelů do tzv. superpixelů) – mají vynikající světelnou citlivost, protože jejich konverzní účinnost je velmi vysoká (nedochází k ohřevu senzorů) – v současné době dosahují vyšší kvality a lepšího plošného rozlišení.

Výroba CMOS senzorů je podstatně levnější a technologicky méně náročná než u CCD (až 100×). Také provoz je v porovnání s CCD senzory podstatně levnější a tedy výhodnější. Nevýhodou CMOS senzorů je malý poměr signál/šum a nižší světelná citlivost (část energie fotonu se spotřebuje na nežádoucí ohřev CMOS senzorů). V průběhu technologického procesu je možné optimalizovat fototranzistory

na citlivost na různé vlnové délky světla, čímž lze zaznamenat informace o barvě (kterou lze zpětně rekonstruovat). CMOS senzory se vyznačují také širokým dynamickým rozsahem. Více je uvedeno v [1] a [2].

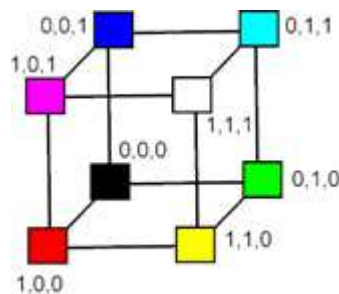
## 2.4 BAREVNÉ KÓDOVÁNÍ

Barevný model je model popisující základní barvy a určující způsob mísení těchto základních barev do barvy výsledné.

Rozlišujeme dva základní způsoby míchání barev:

- Aditivní (sčítací) míchání - přidáním nového odstínu se výsledná barva zesvětlí
- Subtraktivní (odečítací) míchání - přidáním barevného odstínu vznikne tmavší barva

### 2.4.1 Aditivní – RGB



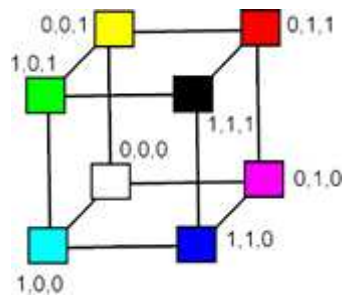
Obrázek 2.8: Model RGB

Model RGB vychází z faktu, že lidské oko obsahuje tři základní druhy buněk citlivých na barvu. Tyto buňky jsou citlivé na vlnové délky, které zhruba odpovídají červené (vlnová délka 630 nm), zelené (530 nm) a modré (450 nm) barvě. Kombinací těchto barev lze získat téměř všechny barvy barevného spektra.

RGB model je součtový a lze jej vyjádřit pomocí jednotkové krychle, kdy v počátku (0,0,0) leží černá barva a v protilehlém vrcholu (1,1,1) barva bílá - obecně lze říci, že v protilehlých vrcholech krychle leží vzájemně komplementární barvy jejichž součtem získáme bílou barvu. Barevné odstíny vznikají skládáním základních barev, jejichž intezita se udává v intervalu  $\langle 0,1 \rangle$ . V počítačové grafice se používá dělení intervalu intezity základní barvy na 256 dílů (0-255) - libovolnou barvu pak

můžeme udávat 24 (3x8) bity - barvy udávané pomocí 24 bitů označujeme jako true colors. V rámci true colors můžeme zobrazit  $256^3$ , tzn. 16 777 216 barev.

#### 2.4.2 Subtraktivní – CMY



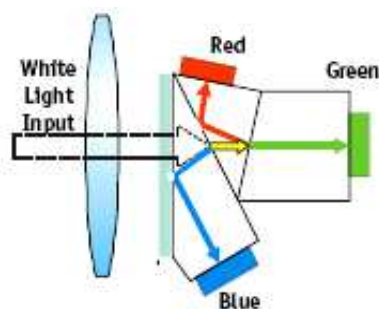
Obrázek 2.9: Model CMY

Barevný model CMY lépe odpovídá lidské zkušenosti s mícháním barev. Tento postup je typický pro míchání malířských nebo tiskařských barev - proto je tento model používán především v tiskařské technice. Podíl jednotlivé barevné složky je opět definován v rozmezí 0 až 255 nebo v procentech.

### 2.5 SNÍMÁNÍ JEDNOTLIVÝCH BAREVNÝCH SLOŽEK

Snímače jako takové barvu dopadajícího světla nerozlišují. Každá buňka registruje pouze intenzitu světla, nikoli jeho frekvenci, která udává barvu světla. Snímač samotný je tedy barvoslepý a přirozeným výstupem je obrázek ve škálách šedé.

#### 2.5.1 Separátní filtry na nezávislých snímačích

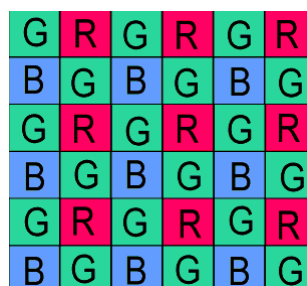


Obrázek 2.10: Separátní filtry

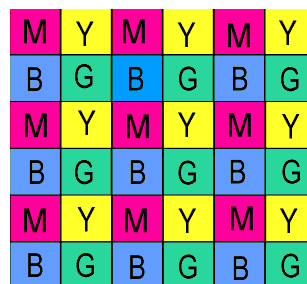


V jednom přístroji je umístěno více snímačů a před každým je jiný barevný filtr. Světlo přicházející z objektivu je pomocí optického hranolu rozloženo na jednotlivé snímače. V rámci jedné expozice je tedy možno provést snímání na všech CCD. Nejběžnější je varianta se třemi snímači, při které je před každým snímačem jeden z RGB filtrů.

### 2.5.2 Bayer mask



**Obrázek 2.11:** Maska aditivního modelu



**Obrázek 2.12:** Maska subtraktivního modelu

Základem je opět použití barevných filtrů. Aby bylo možné provést měření všech základních barev současně v průběhu jedné krátké expozice světlem, jsou na snímači současně všechny filtry uspořádané do mozaiky. Nejde o filtry celistvé, uložené přes celou plochu snímače, ale nad každou buňku je umístěn jeden miniaturní filtr, který je nad buňku napařen již při výrobě. Podrobnější popis je pak uveden v [8].

### 3. ZPRACOVÁNÍ OBRAZU

Mezi nejčastěji využívané metody zpracování obrazu patří filtrace obrazu. Filtrací obrazu rozumíme v obecné rovině operace s digitálním obrazem, které slouží ke zvýraznění určité informace. Rozlišujeme celou řadu metod filtrace obrazu od velmi jednoduchých (např. prosté průměrování) až po poměrně komplexní, sofistikované metody.

Filtrace může být využívána pro vyhlazení obrazu, potlačení šumu, zvýraznění kontrastu, detekci hran, postklasifikační zpracování obrazu a řadu dalších úloh. Vzhledem k rozsahu většiny digitálních dat je z technického hlediska nevhodné řešit podobné úlohy najednou v celém obrazu. Daný filtr je tak definován jako šablona rastrové matice (tzv. "moving window", v české literatuře se často využívá termín "kernel") - tedy pohybujícího se (plovoucího) okna. Jde o matici (většinou čtvercovou) tvořenou lichým počtem řádků a sloupců, která se při výpočtu pohybuje nad maticí originálních dat (ve směru řádků a sloupců). Nová hodnota rastrové buňky je určena na základě aritmetické operace či statistické veličiny definované filtrem a hodnot originálních dat. Do výpočtu tak vstupuje na rozdíl například od podílu obrazu množina hodnot a nikoliv pouze jedna jediná hodnota. Jde tedy o lokální a nikoliv bodové operace. Nejčastěji se využívá velikost kernelu 3x3 a 5x5.

Filtry lze rozdělit na dva základní typy:

- filtry s nízkou (low pass) propustností, které ořezávají vysoké frekvence
- filtry s vysokou (high pass) propustností, které naopak zdůrazňují vysoké frekvence

Jas v bodě  $(i,j)$  je dán lineární kombinací jasů v okolí  $O$  (velikosti  $M \times N$ ) vstupního obrazu  $h$  s váhovými koeficienty  $f$ . Pro izoplanární (nezávislá na poloze) systémy se jedná o diskrétní konvoluci:

$$g(i, j) = \sum_{m=i-\frac{M}{2}}^{i+\frac{M}{2}} \sum_{n=j-\frac{N}{2}}^{j+\frac{N}{2}} f(m-i, n-j) \cdot h(m, n) \quad (3)$$

Samotná filtrace a případné ostření probíhá ve dvou krocích. Pokud je obraz barevný, rozdělí se na tři kanály podle RGB barev, jedna barva na jeden kanál. Pokud je obraz černobílý, zůstává beze změn. Poté se aplikuje konvoluce. V případě černobílého obrazu přímo na něj, je-li obraz barevný, aplikuje se konvoluce na každý kanál samostatně. Výsledek konvoluce se přičte k původnímu obrazu a případně se tři kanály spojí zpět v jeden RGB obraz. Další typy a příklady filtrů lze nalézt v [5] a [6].

### 3.1 FILTRY S NÍZKOU PROPUSTNOSTÍ

Tento typ filtru potlačuje výraznější detaily v obraze a současně zdůrazňuje jevy s malými změnami. Vlastnosti:

- omezuje odchylky od lokálního průměru
- vyhlazuje detaily původního obrazu
- redukuje rozsah stupňů šedi

#### 3.1.1 Průměrování

Hodnota nového bodu se spočítá jako průměr z jeho okolí. Ostré změny v obraze, například hrany či šum, se rozmazou. Opakovanou aplikací filtru je rozmazání silnější. Limitně vede tato operace k jednobarevnému obrazu, jehož barva je průměrem ze všech hodnot v obraze.

$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \text{ aritmetický průměr, } h = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \text{ zvýšení váhy středu}$$

#### 3.1.2 Medián

Hlavní nevýhodou lokálně průměrujících filtrů je sklon k rozmazávání ostrých přechodů v obraze. Alternativním přístupem je nahrazení každého pixelu obrazu hodnotou mediánu z pixelu v jeho okolí. Medián získáme tak, že seřadíme všechny pixely podle jejich hodnoty a vybereme prvek, který se nachází uprostřed.

### 3.2 FILTRY S VYSOKOU PROPUSTNOSTÍ

Tento typ filtru naopak zdůrazňuje výraznější detaily a potlačuje jevy s malými změnami. Jde především o ostřicí filtry a hranové operátory. Vlastnosti:

- zvětšují detaily v ploše snímku
- jsou založeny na aproximaci druhé derivace obrazu

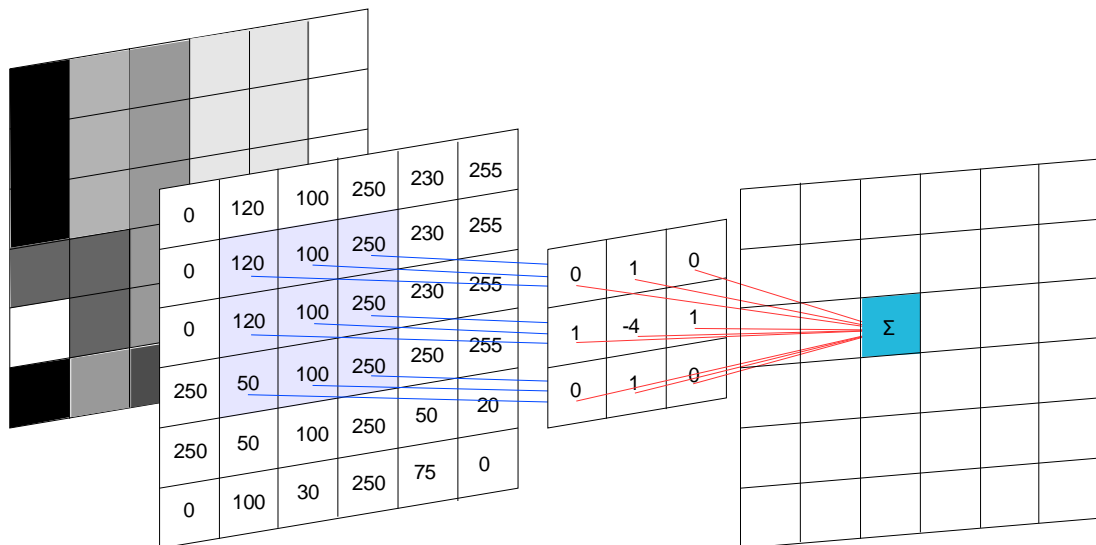
#### 3.2.1 Operátor invariantní vůči rotaci

$$h = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \longleftrightarrow \text{Laplaceův operátor}$$

#### 3.2.2 Operátory neinvariantní vůči rotaci

$$h = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \longleftrightarrow h = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \longleftrightarrow \text{Sobelův operátor}$$

$$h = \begin{bmatrix} 3 & 3 & 3 \\ 3 & 0 & 3 \\ -5 & -5 & -5 \end{bmatrix} \longleftrightarrow h = \begin{bmatrix} 3 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & -5 & 3 \end{bmatrix} \longleftrightarrow \text{Kirschův operátor}$$



Obrázek 3.1: Princip filtrace jednobarevného obrazu

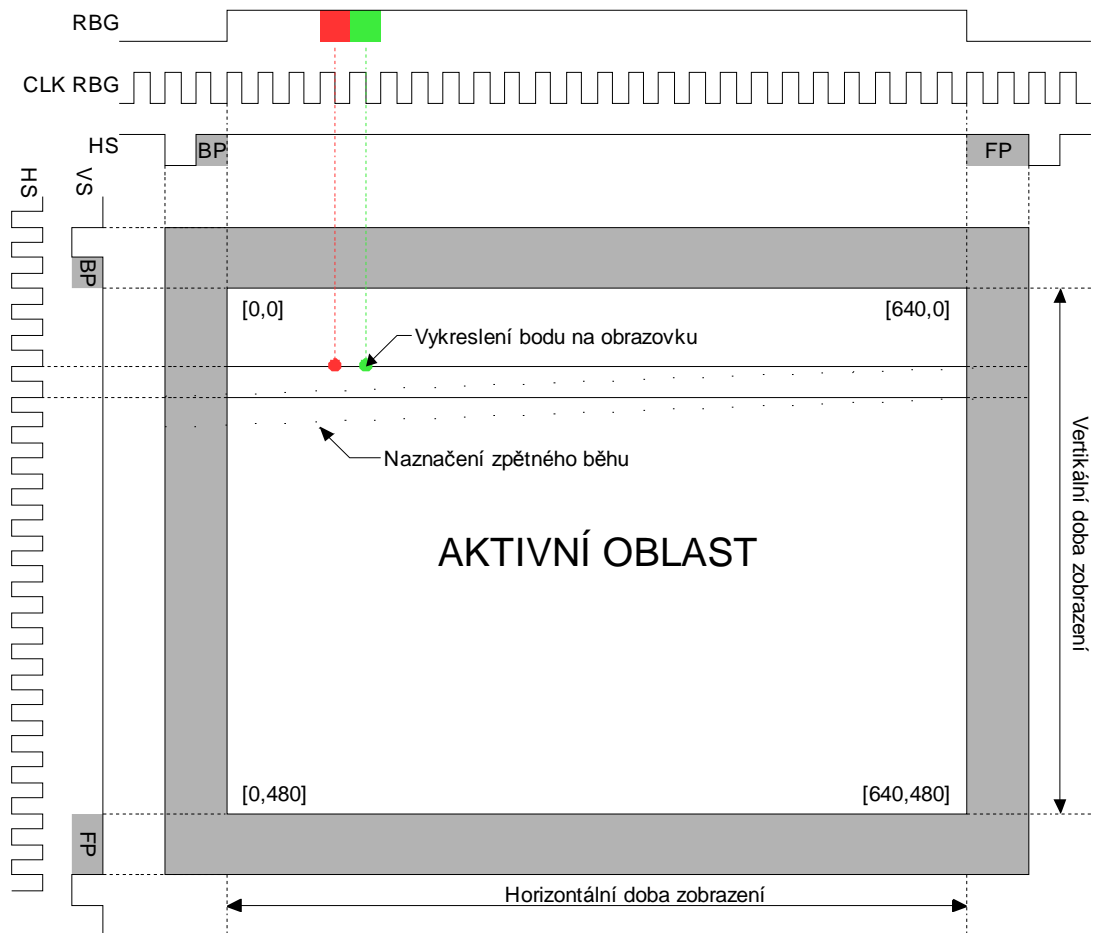
## 4. VGA ROZHŘANÍ

VGA rozhraním je dnes označováno rozhraní počítač-monitor, tedy grafický adaptér. Tento název nese toto rozhraní od té doby, kdy IBM přišla na trh s novým grafickým adaptérem VGA, který se stal průmyslovým standardem. Poprvé se objevil právě na tomto grafickém adaptéru trojřadý 15-ti pinový konektor D-SUB (DB15), který je dnes označován jako VGA konektor. Těmi nejvýznamnějšími signály, které jsou vyvedeny na piny VGA konektoru na grafickém adaptéru jsou: horizontální synchronizace (HS), vertikální synchronizace (VS) a signály tří základních barev, tj. červená (R), zelená (G), modrá (B).

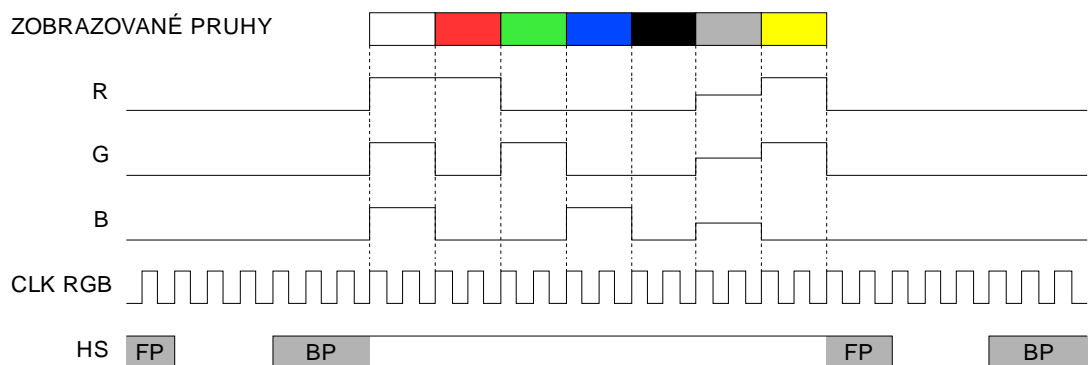
### 4.1 VGA PROTOKOL

Časování VGA signálů specifikovala, vydala a autorská práva vlastní asociace VESA. Kompletní informace o časování lze nalézt v dokumentaci VESA dostupné bohužel pouze za úplatu. Nejvíce se lze asi dozvědět z [9] a pak také z [11]. Pro popis časování byl zvolen režim VGA, grafické rozlišení 640 x 480 bodů, frekvence vykreslování bodů 25 MHz, řádkový kmitočet 31,25 KHz, snímkový kmitočet 60 Hz. U CRT monitoru je obraz na obrazovce vytvářen amplitudově modulovaným elektronovým paprskem pohybujícím se po stínítku zleva doprava a odshora dolů, viz obrázek 4.1. Při horizontálním vychylování paprsek svým přímočarým pohybem zleva doprava vykresluje jednotlivé řádky, jedná se o tzv. řádkový aktivní běh. Jakmile paprsek dospěje za pravý okraj obrazu, vrací se rychle zpět, pak se jedná o tzv. řádkový zpětný běh. Při zpětném běhu se informace na obrazovce nezobrazuje. Jelikož je paprsek podroben také vertikálnímu vychylování, pohybuje se paprsek i ve svislém směru, ale mnohem pomaleji než ve vodorovném směru. Proto jsou ve skutečnosti řádky mírně skloněny. Pohybuje-li se paprsek odshora dolů, jedná se o tzv. snímkový aktivní běh. Vrací-li se paprsek zpět nahoru, pak se jedná o tzv. snímkový zpětný běh. Vytvořený obraz na stínítku je pak ve skutečnosti obraz časově rozvinutý v body, kde barva bodu je daná třemi amplitudově modulovanými signály RGB, které jsou přerušovány zatemňovacími impulsy. Během doby zatemnění má amplituda signálů RGB takovou úroveň, že se

bod na obrazovce nerozsvítí. Tyto úseky jsou vyznačeny šedou barvou na obrázku 4.1. Během kratší doby zatemnění probíhá řádkový zpětný běh a během delší doby zatemnění probíhá snímkový zpětný běh, který je tedy pomalejší. Během každého snímkového zpětného běhu pokračuje řádkové vychylování, proto se paprsek při dokončení snímku nevrací přímou cestou z pravého spodního rohu obrazovky nahoru do levého horního rohu, ale klikatou cestou. Do šířky vertikálního synchronizačního impulsu vejde několik period horizontální synchronizace. Navíc snímkový zpětný běh nemusí nutně začínat až po dokončení posledního řádku v pravém spodním rohu, ale např. už uprostřed posledního zatemněného řádku (vždy mimo aktivní oblast) a končit uprostřed horního zatemněného řádku. Pak už následuje normální řádkový běh, zleva doprava atd. Správné časování signálů HS a VS zajišťuje, aby se video data (RGB) dostala na správnou souřadnici bodu (pixelu) v každém taktu vzorkovacích hodin (CLK\_RGB). Způsob vytváření obrazu pomocí těchto VGA signálů je demonstrován na příkladu generování svislých barevných pruhů znázorněný na obrázku 4.2.



Obrázek 4.1: Časování VGA signálů



Obrázek 4.2: Generování svislých barevných pruhů na obrazovce monitoru

## 4.2 VGA SIGNÁLY

### 4.2.1 Vertikální synchronizace

Signál vertikální synchronizace VS určuje frekvenci, s jakou jsou snímky na obrazovce obnovovány. Perioda tohoto signálu tak určuje dobu, za kterou je jeden celý snímek nebo půlsnímek vykreslen na obrazovku. To zda je to celý snímek nebo jen půlsnímek záleží na tom, jedná-li se o neprokládané nebo prokládané řádkování. Náběžná nebo sestupná hrana pozitivního nebo negativního synchronizačního impulsu v signálu určuje začátek každého nového snímku nebo půlsnímku, viz. obrázek 4.1. Zde jsou vyobrazeny pouze časové průběhy signálů pro neprokládané řádkování s negativní synchronizací. Snímková frekvence se dnes běžně pohybuje v rozsahu 56 až 120 Hz.

### 4.2.2 Horizontální synchronizace

Signál horizontální synchronizace HS určuje frekvenci, se kterou jsou vykreslovány jednotlivé řádky na obrazovce. Tento signál má velmi podobné vlastnosti a řídí se podobnými pravidly jako předchozí synchronizační signál VS. Řádková frekvence je o několik řádů vyšší než snímková a běžně se dnes pohybuje v rozsahu 31,5 až 100 kHz. Synchronizačního impulsu v signálu HS v tomto případě udává začátek každého nového řádku, viz. obrázek 4.1.

### 4.2.3 RGB signály

Analogové signály RGB přenáší obrazová data pro každý bod na obrazovce zvlášť. Na obr.4.1 je zobrazen pomocný vzorkovací signál označený CLK\_RGB, který každým taktem definuje čas, kdy je informace o barvě z analogového signálu RGB předána danému bodu na obrazovku. Signál CLK\_RGB se ale nepřenáší z grafického adaptéru do monitoru. Frekvence vzorkování barevného analogového signálu, respektive frekvence s jakou jsou vykreslovány jednotlivé body na obrazovce se dnes běžně pohybuje v rozsahu od 25 do 315MHz. Jelikož je barva pro jeden bod přenášena pomocí tří barevných signálů (R,G,B), je výsledná barva dána jejich součtem. Jedná se o tzv. aditivní (součtové) mísení barev, kde při současném dopadu dvou nebo tří světelných paprsků na plochu vzniká nová barva. Intenzita



zrakového vjemu se sčítá, jas se zvětšuje. Při plné intenzitě jednotlivých složek barev RGB tak vzniká na stínítku barva bílá.

#### **4.2.4 Aktivní oblast**

Aktivní oblast odpovídá rozměrům obrazu zobrazeném na stínítku obrazovky, v tomto případě 640 x 480 bodů. Aktivní oblast se nazývá proto, že elektronový paprsek při průchodu touto oblastí vykresluje viditelně jednotlivé řádky, respektive celé snímky na obrazovku. Mimo tuto oblast je elektronový paprsek zatemněn.

#### **4.2.5 Front porch**

Front porch (FP) se nazývá časový interval mezi koncem vykreslování obrazu a začátkem synchronizačního impulsu. Během tohoto intervalu se žádná informace na obrazovce nezobrazuje, tvoří tak okraj obrazu, přesněji jeho pravý a spodní okraj. Tento interval je v čase před započítáním zpětného běhu.

#### **4.2.6 Back porch**

Back porch (BP) se nazývá časový interval mezi koncem synchronizačního impulsu a začátkem vykreslování obrazu. Během tohoto intervalu se žádná informace na obrazovce nezobrazuje, tvoří tak okraj obrazu, přesněji jeho levý a horní okraj. Tento interval obvykle v čase následuje až po dokončení zpětného běhu. Interval Back porch mívá zpravidla o dost delší dobu trvání než Front porch.

#### **4.2.7 Doba zatemnění**

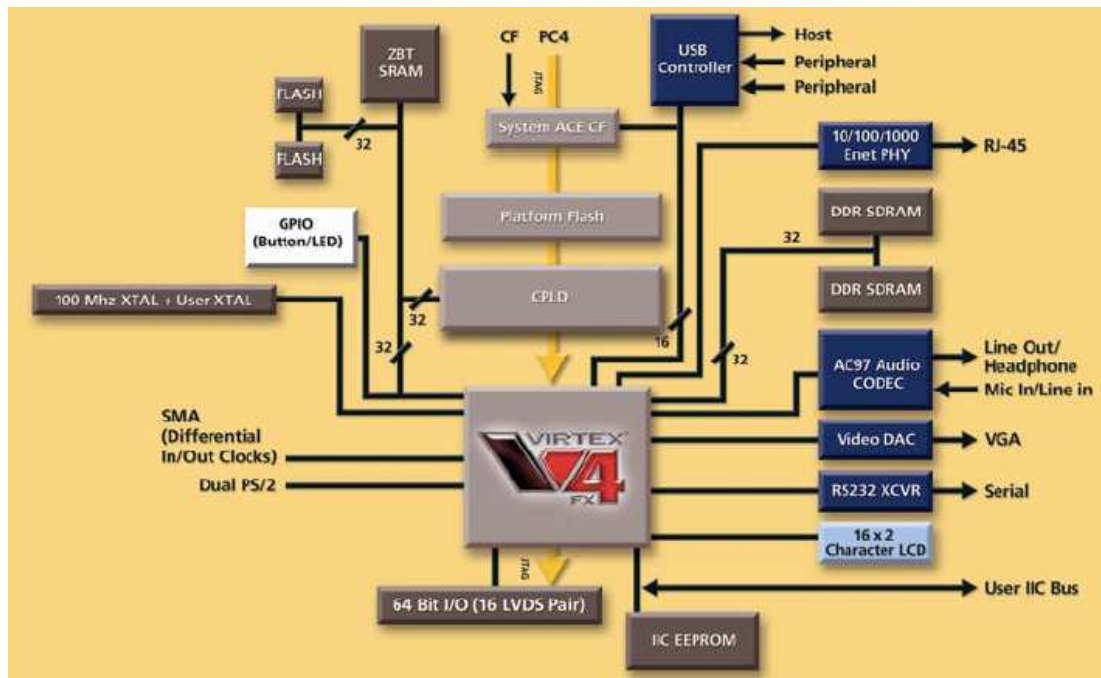
Dobu zatemnění tvoří dohromady intervaly FP, BP a synchronizační impuls jak naznačuje obr.4.1. Během doby zatemnění probíhá tzv. zpětný běh a příprava elektronového paprsku k vykreslování nového řádku či snímku po jeho návratu.

## 5. ML 403 DEVELOPMENT BOARD

Použitý vývojový kit Xilinx Virtex-4 ML403 Evaluation Platform je z řady Xilinx ML40x (ML401, ML402, ML403, ML405), které se liší především typem použitého FPGA obvodu z řady Virtex-4. Řada Virtex-4 je dále rozdělena do tří skupin. Obvody označené LX obsahují hodně logických a vstupně výstupních bloků, obvody SX mají hodně DSP bloků a větší množství paměti a FX, které obsahují procesorová jádra PowerPC. Kit ML403 je osazen FPGA obvodem ze skupiny FX. Podrobný popis je v [12], kompletní schéma je v [13] a další informace pak v [15] a [16].

Kit se skládá z těchto hlavních částí:

- XC4VFX12-FF668-10C - VIRTEX-4 FX FPGA obsahující jedno procesorové jádro POWER PPC 405
- 64 MB DDR SDRAM paměť
- RS – 232 sériový port
- 10/100/1000 ETHERNET rozhraní
- JTAG - Joint Test Action Group – konfigurační port
- 4 MB flash paměť
- System ACE configuration chip



**Obrázek 5.1:** Blokové schéma vývojového kitu ML 403

## 5.1 HRADLOVÉ POLE FPGA

Jelikož je použito pole VIRTEX-4, je popis zaměřen pouze na tento obvod. Konstrukce jiných FPGA obvodů je do značné míry podobná.

Hradlové pole je složeno z několika základních funkčních elementů, které lze podle potřeby nakonfigurovat aby byla zajištěna funkčnost a maximální výkon implementovaného designu. Kompletní popis pak je uveden v [17], [19], [20] a [21].

VIRTEX-4 obsahuje tyto bloky:

- IOB – bloky vstupů a výstupů
- CLB – bloky základní logiky
- BRAM – blokové RAM paměti
- DCM – bloky pro operace s hodinovým signálem
- PMCD – bloky pro generování hodinových signálů
- XtremeDSP – bloky pro rychlé zpracování signálů
- Clock resources – rozvody hodin

Pole s označením FX navíc obsahuje:

- PowerPC 405 – hardcore procesor

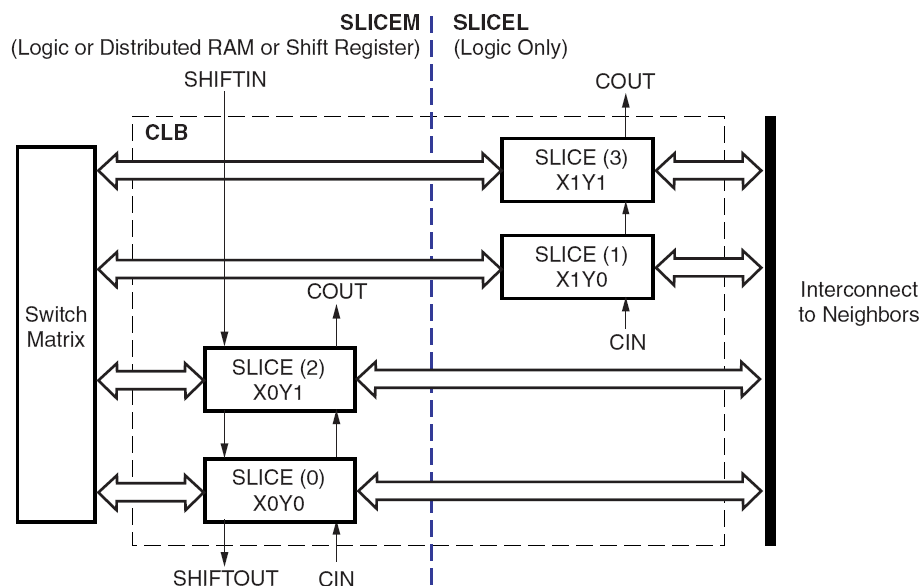
- Tri-Mode Ethernet MAC – řadič ethernetu
- RocketIO Transceiver – rychlé sériové rozhraní

### 5.1.1 IOB

Bloky vstupů a výstupů poskytují rozhraní mezi pinem obvodu a vnitřní logikou. Každý pin lze nakonfigurovat jako vstup, výstup, nebo třístavový výstup. Dál lze nastavit každému pinu jeden z vstupně-výstupních standartů. Pro single-ended zapojení to jsou LVCMOS, LVTTL, HSTL, SSTL, GTL, PCI a pro diferenciální to jsou LVDS, LDT, LVPECL, BLVDS, HSTL a SSTL. U výstupního pinu lze nadefinovat omezení výstupního proudu.

### 5.1.2 CLB

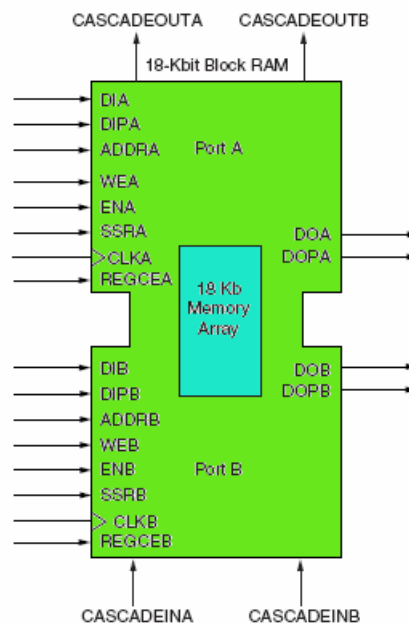
Každý CLB blok je připojen na hlavní propojovací matici. Jeden CLB blok se skládá ze čtyř SLICE bloků, SLICE bloky tvoří uvnitř CLB dva páry, jak vidět na obrázku III. Každý SLICE blok tvoří dva funkční generátory F a G, dvě paměťové buňky, několik multiplexerů a logických hradel a propojovací logika. Bloky s označením SLICEM navíc můžeme nakonfigurovat jako 16 bitový posuvný registr nebo jako 16 bitovou rozvrženou paměť.



Obrázek 5.2: Struktura CLB bloku

### 5.1.3 BRAM

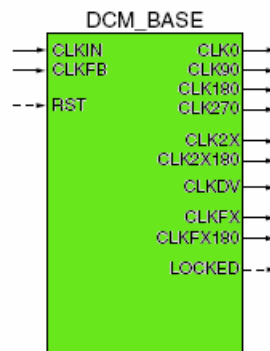
Jedná se o konfigurovatelný paměťový modul. Základ modulu tvoří 18 kb dvouportový RAM blok. Oba porty jsou symetrické a nezávislé a sdílejí pouze uložená data. Každý port může mít nezávisle na druhém nastaven jiný rozměr v rozsahu 16Kx1, 8Kx2, do 512x36. Dva BRAM moduly můžeme zapojit kaskádně bez nutnosti použití další logiky a získat tak paměť o velikosti 36 kb. Obsah paměti může být vymazán nebo nahrán konfiguračním bitstreamem. Nejčastěji je tento modul využíván pro tvorbu zásobníku typu FIFO. U FIFO může být nastavena velikost zásobníku na 4K při bitové šířce 4 bity, nebo 2Kx9, 1Kx18 a 512x36. Zapisující i čtecí porty mají stejnou šířku dat, ale frekvence zapisovacích a čtecích hodin může být naprosto nesourodá (asynchronní FIFO).



Obrázek 5.3: Struktura BRAM paměti

### 5.1.4 DCM

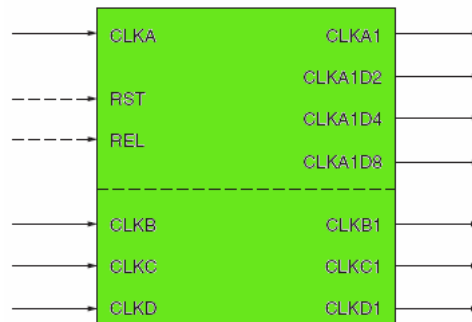
Jedná se o násobičku hodinového signálu. Vstupním signálem CLKIN. Na výstupu jsou hodiny se stejnou frekvencí jako na vstupu (CLK0), hodiny s fázovým posunem o 90° (CLK90), 180° (CLK180) nebo 270° (CLK270). Dál jsou na výstupu hodiny násobené konstantou (CLKFX). Konstanta násobení je určena poměrem dvou celých čísel M (multiplier) a D (divisor).



Obrázek 5.4: DCM blok

### 5.1.5 PMCD

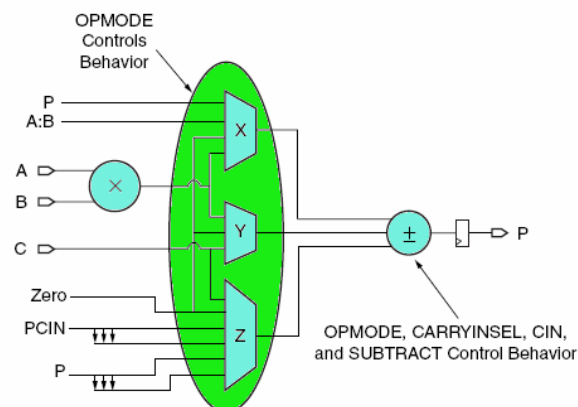
Jedná se o modul na jehož výstupu je čtveřice signálu jenž vůči sobě nejsou fázově posunuty. Jsou to signály s frekvencí stejnou jak vstupní, s frekvencí poloviční, čtvrtinovou a osminovou oproti vstupnímu signálu.



Obrázek 5.5: PMCD blok

### 5.1.6 XtremeDSP

Každý blok XtremeDSP obsahuje dvě buňky DSP48 SLICE. Umožňuje provádět široké množství funkcí jež vyžadují násobení, násobení následováno sčítáním, porovnávání velkých dat, popřípadě velké čítače. DSP48 SLICE buňky je možno propojovat a použít k implementaci složitých matematických operací, DSP filtrů. Každá buňka DSP48 SLICE obsahuje 18bit x 18bit násobičku, 48bitovou třívstupovou sčítačku a další logické obvody. Kompletní popis bloku je v [18].



**Obrázek 5.6:** DSP48 SLICE buňka

### 5.1.7 Clock resources

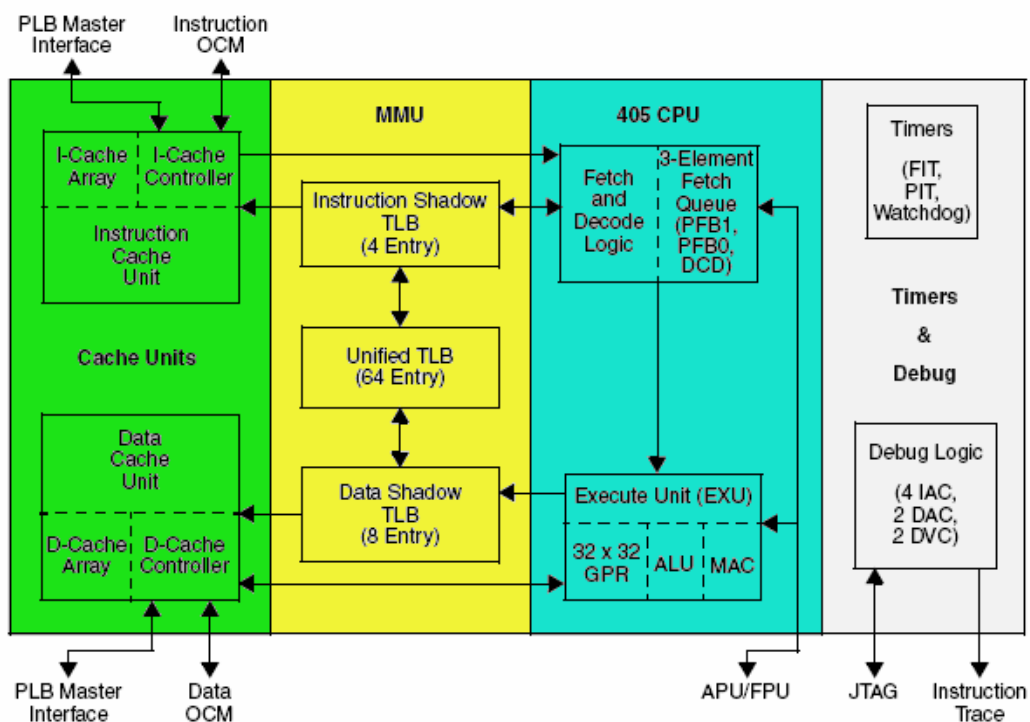
Každý čip obsahuje globální a regionální rozvod hodin. Čip je rozdělen na regiony. Jejich počet je závislý na velikosti čipu od 8 u nejmenšího až po 24 u největšího. V každém čipu je 32 globálních rozvodů hodin jenž sou zdrojem hodin pro základní bloky čipu (CLB, BRAM, DCM) a jednotlivé regiony. Každý region má dva vstupy pro hodiny, jenž můžou být v rámci regionu poděleny celým číslem v rozmezí 1 až 8.

## 5.2 PROCESOROVÉ JÁDRO POWERPC 405

### 5.2.1 Charakteristika

PowerPc 405 je vestavěný 32-bitový procesor s RISC architekturou od firmy IBM, splňuje specifikaci UISA (User Instruction Set Architecture) a částečně také splňuje i ostatní specifikace (VEA – Virtual Enviroment Architectrue, OEA – Operating System Architecture). Procesor může pracovat ve dvou režimech jak uživatelském tak i chráněném, podporuje virtuální paměťový prostor, nepodporuje 64-bitové operace a floating point. Jeho pracovní frekvence může činit až 400 MHz. Kompletní popis jádra je v [25], další informace pak jsou [26] a dané jádro v FPGA pak je popsáno v [23] a [24].

## 5.2.2 Architektura



**Obrázek 5.7:** Procesorové jádro PowerPc 405

Procesor používá zřetězené zpracování instrukcí v pěti stupních. Obsahuje instrukční a datovou cache jejichž velikost je 16 kB. Jedná se o dvoucestné paměti, každá cesta obsahuje 256 řádků po 32 bajtech. Obsluha probíhá pomocí ICU (Instruction Cache Unit) a DCU (Data Cache Unit) jednotek, které načítají data z externích pamětí skrze sběrnici PLB (Processor Local Bus). ICU je schopna zasílat 1-2 instrukce do dekodování jednotky v každém taktu, nepoužité instrukce jsou zahozeny (pro skoky). DCU pracuje při zápisu v režimu Write-through nebo Write-back (lze konfigurovat ze strany uživatele).

Cílem jednotky pro stupně fetch a decode je zachovat plynulý tok instrukcí do jednotky provádění (Execution Unit). Součástí jednotky je fronta předzpracovaných instrukcí, do které je možné zasílat až dvě instrukce v jednom taktu. Celkem má fronta tři položky: dvě předzpracované a jednu dekodovanou pro odeslání do EU. Uvnitř jednotky je implementována statická predikce skoků: pro skoky s negativním



podmínkou se předpokládá, že se provedou (způsob predikce lze změnit v assembleru nebo nastavením v době kompilace). Pokud skončí predikce neúspěšně, odstraní jednotka instrukce načtené v frontě předzpracovaných instrukcí.

Jednotka provádění instrukcí (Execution Unit) obsahuje registrové pole 32 x 32 bitových registrů pro obecné použití, aritmeticko-logickou jednotku a jednotku MAC (multiply and accumulate) operaci využívanou v DSP aplikacích. Registrové pole obsahuje celkem 3 čtecí porty a 2 zápisové porty, 2 čtecí a 1 zápisový port je určen pro běžné provádění aritmeticko-logických instrukcí, zbývající 1 čtecí a 1 zápisový port je použit pro operace Load/Store pro načítání/zápis dat. Oba tyto typy operací lze pak provádět paralelně.

PowerPC umožňuje adresovat paměťový prostor až o velikosti 4GB. Memory Management Unit (MMU) zajišťuje ochranu tohoto prostoru tzn. jednotlivé aplikace si nemohou vzájemně zasahovat do paměťového prostoru, stejně tak nemohou zasahovat do prostoru operačního systému. Pro efektivní práci s paměťovým prostorem podporuje režim stránkování, kdy se provádí překlad mezi logickou a fyzickou adresou. Tento stránkový režim lze nezávisle aktivovat pro paměť instrukční a paměť dat. Podporuje následující velikosti stránek: 1 KB, 4 KB, 16 KB, 64 KB, 256 KB, 1 MB, 4 MB a 16 MB. Pro rychlý překlad adres obsahuje PowerPC plně asociativní cache paměť o velikosti 64 položek. Pro zvýšení výkonu lze některé z položek trvale rezervovat pro často používané oblasti paměti.

### **5.2.3 Instrukční sada**

- Aritmetické operace: ADD, SUB, MUL, DIV, CMP (22 instrukcí)
- Multiply and Accumulate operace: MAC (18 instrukcí)
- Logické operace: AND, NAND, OR, NOR, XOR, EQV (17 instrukcí)
- Instrukce posunu a rotace: RLW, SR (7 instrukcí)
- Instrukce pro načítání/zápis dat: ST[B,H,W], L[B,H,W] (38 instrukcí)
- Instrukce skoku: B, BC, BCCTR, BCRL (4 instrukce)
- Podmíněné logické instrukce: CRAND, CROR, CRXOR, atd. (9 instrukcí)
- Instrukce pro řízení procesoru (12 instrukcí)
- Instrukce pro řízení cache paměti (13 instrukcí)

- Instrukce pro stránkování (5 instrukcí)

#### 5.2.4 Rozhraní procesoru

Připojení jednotlivých komponent k PowerPC procesoru vychází ze specifikace CoreConnect od firmy IBM:

- On-chip Memory (OCM)
- Processor Local Bus (PLB)
- On-chip Peripheral Bus (OPB)
- Device Control Register Bus (DCR)

#### 5.2.5 On-chip Memory (OCM)

Součástí procesoru jsou dva fyzicky odělené řadiče (instrukční ISOCM a datový DSOCM). Pracovní kmitočet lze definovat pro každý radič nezávisle. Poměr mezi kmitočtem jádra a kmitočtem rozhraní OCM může být 1:1, 2:1, 3:1 nebo 4:1. PowerPC vyčleňuje pro každý OCM řadič 16MB paměťového prostoru, přímo z OCM lze adresovat 64K a 128K adresového prostoru. Jedno čtení z paměti trvá dva takty – v prvním taktu je vystavena adresa, v druhém taktu se čtou data. Tím, že ISOCM má šířku dat 64 bitu, poskytuje přístup do instrukční paměti s propustností jedna instrukce za jeden hodinový cyklus. Datová propustnost OCM je opozdní horší, neboť ta je oproti datové PLB poloviční. Čtení i zápis sice trvají podobně jako v případě instrukční větve dva takty, avšak šířka datové sběrnice je poloviční.

#### 5.2.6 Processor Local Bus (PLB)

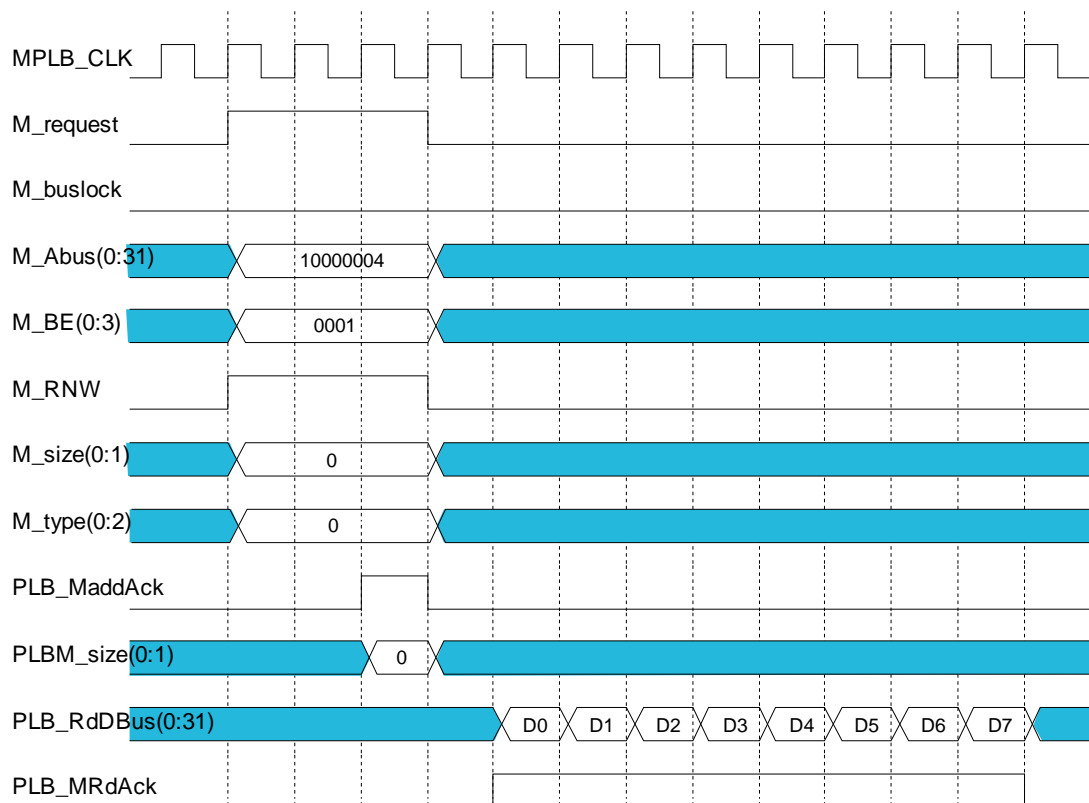
Sběrnice PLB je komunikačně nejsložitějším a nejvyspělejším rozhraním procesorového jádra. Jedná se o sběrnici dovolující řetězení a tím zpracování několika transakcí současně s cílem co nejvíce překrýt komunikační režii. Aby se dosáhlo vyšší výkonnosti, obsahuje procesorové jádro dvě nezávislé PLB sběrnice – datovou DSPLB (Data-Side PLB) napojenou na radič datové cache (DCU) a instrukční ISPLB (Instruction-Side PLB) sběrnici připojenou k řadiči instrukční cache (ICU). Protože PLB vedoucí z procesorového jádra je řízena řadiči cache a umožňuje připojit libovolný počet periférií, označuje se jako PLB master. Na úrovni

FPGA čipu lze obě dedikované PLB sloučit a vytvořit sběrnici sdílenou. K tomu je však zapotřebí PLB arbitr.

ISPLB dovoluje ICU (Instruction Cache Unit) číst instrukce z jakéhokoliv paměťového zařízení, které je k této sběrnici připojeno. Neumožňuje však jejich zápis. Zápis do instrukční paměti je nutné řešit sdílenou sběrnici pomocí DSPLB. Instrukční PLB má 30 bitovou adresní sběrnici a 64 bitovou datovou sběrnici. Rozhraní je navrženo tak, aby mohlo být připojeno jako PLB master k 64 bitové nebo 32 bitové sběrnici PLB. Sběrnice je schopna dosáhnout propustnosti 64/32 bitu za takt PLBCLK.

DSPLB dovoluje DCU (Data Cache Unit) číst a zapisovat data z jakéhokoliv paměťového zařízení připojeného na tuto sběrnici. Datová PLB obsahuje 32 bitovou adresní část sběrnice a dvě 64 bitové datové sběrnice, jednu pro směr z procesoru (zápis) a druhou pro směr opačný (čtení). Rozhraní je navrženo tak, aby mohlo být připojeno jako PLB master k 64 bitové nebo 32 bitové sběrnici PLB. Sběrnice je schopna dosáhnout propustnosti 64/32 bitu za takt PLBCLK.

Přenos přes sběrnici PLB je zobrazen na obrázku 5.8. PLB master vystaví na adresovou sběrnici  $M\_Abus(0:31)$  adresu ze které chce číst, současně s tímto nastaví signál  $M\_request$  a chce-li číst tak nastaví signál  $M\_RNW$  do logické „1“, v případě zápisu je ponechán v logické „0“. Používáme-li burst přenos určuje signál  $M\_BE(0:3)$  délku přeneseného burstu. Signál  $M\_type(0:2)$  určuje typ přenosu, hodnota 0 indikuje standardní paměťový přenos. Signálem  $M\_size(0:1)$  je určena šířka datové sběrnice. Signálem  $PLB\_Msize(0:1)$  říká slave mastru šířku své adresové sběrnice. Akceptuje-li slave adresu potvrdí jej signálem  $PLB\_MaddAck$ . Při odpovědi jsou na  $PLB\_RdDBus(0:31)$  vystavena postupně požadována data. Signálem  $PLB\_MRdAck$  je potvrzena platnost dat na datové sběrnici. Pro přehlednost nejsou uvedeny všechny signály. Kompletní popis signálů je uveden v [27].



Obrázek 5.8: Čtení přes sběrnici PLB

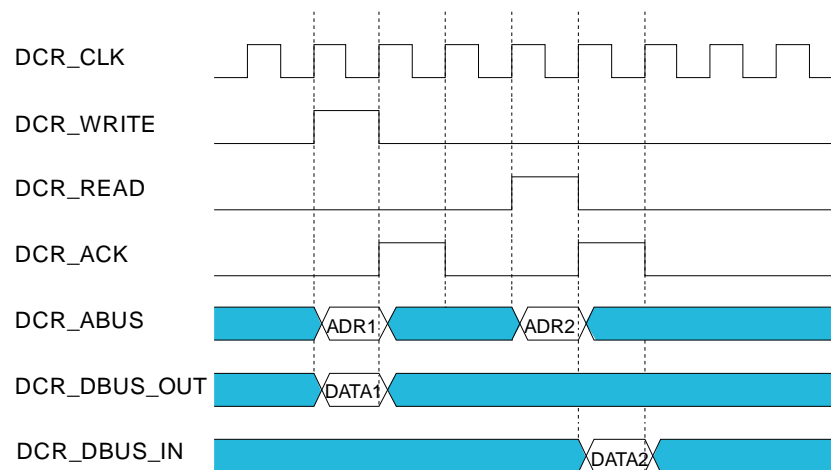
### 5.2.7 On-chip Peripheral Bus (OPB)

Určena pro pomalejší periférie, od PLB je oddělena bridžem. Adresní sběrnici může mít až 64 bitů, datovou sběrnici má 32 bitovou nebo 64 bitovou. sběrnice s podporou „Master-Slave“ komunikace, zařízení typu „Master“ může být na sběrnici více, přidělení mezi nimi potom řeší centrální arbit. V současnosti se tato sběrnice dostává do ústraní.

### 5.2.8 Device Control Register Bus (DCR)

Rozhraní DCR poskytuje prostředek, kterým dovoluje procesoru řídit a inicializovat části systému, které nejsou součástí procesorového jádra. Na interní DCR rozhraní jsou připojeny oba externí OCM řadiče (datový i instrukční), ke kterým by jinak nebylo možné z procesoru přistupovat a konfigurovat je. Jedná se vlastně o 32-bitový paralelní port, pomocí kterého je možné číst a měnit obsah registru externích zařízení. Jednotlivá zařízení a jejich registry je možné adresovat pomocí 10-bitové adresy. Protože je DCR vyvedeno i z procesorového bloku (externí

DCR), je možné je s výhodou využít k ovládání externích periférií umístěných uvnitř FPGA čipu. Na DCR je možno připojit libovolný počet zařízení. Sdílení se provede tak, že se datové vstupy a výstupy jednotlivých zařízení zapojí do řetězce. Přístup na rozhraní ze strany procesoru je umožněn pomocí specifických instrukcí mtdcr (move to dcr) a mfdcr (move from dcr). Komunikace je zobrazena na obrázku 5.9. Chceme-li do daného registru zapsat hodnotu je nastaven signal write a současně na adresovou sběrnici je vystavena adresa a na výstupní datovou sběrnici data. Proběhl-li zápis úspěšně je potvrzen signálem ack. Čtení dat probíhá obdobným způsobem.



**Obrázek 5.9:** Komunikace přes sběrnici DCR

### 5.3 XILINX EDK

Softwarový balík Xilinx Embedded Development Kit, zkráceně EDK (bude tak i nazýváno v dalším textu) umožňuje návrh komplexních vestavěných systémů pro FPGA obvody Xilinx. Řídícím prvkem těchto systémů je buď softcore procesor MicroBlaze nebo procesor PowerPC. Pro použití jádra PowerPC je nutné mít takový FPGA obvod, který toto jádro obsahuje. Tyto obvody jsou u řady Xilinx Virtex-4 označovány zkratkou FX, jak již bylo zmíněno v kapitole o vývojovém kitu ML403. Vzhledem k použitému kitu, respektive FPGA obvodu, kterým je osazen, je v mé práci použit procesor PowerPC.

Jednou z hlavních součástí balíku EDK je aplikace Xilinx Platform Studio (XPS), která slouží k návrhu vestavěného systému zejména po stránce hardwaru.

Software je možné navrhovat také, ale XPS poskytuje pouze základní funkce. Pro návrh softwaru slouží specializovaná aplikace Xilinx Platform Studio SDK, která umožňuje použít při návrhu aplikace obvyklé funkce pro ladění programu. V následující části textu je uveden základní návod pro práci s XPS.

Konfigurace projektu je uložena ve třech hlavních souborech. Tyto soubory jsou uloženy v adresáři projektu a mají přípony mhs, mss a ucf. Úpravy v nich lze provádět buď přímou editací těchto souborů nebo pomocí funkcí XPS.

### 5.3.1 MHS

V souboru MHS je uložena hardwarová konfigurace projektu. Pro názornost je uvedeno, jak vypadají záznamy v jednotlivých souborech, pokud je součástí projektu jádro `plbv46_dvi_cntlr`, které zprostředkovává čtení dat a následně zobrazení na tft monitoru.

```
BEGIN plbv46_dvi_cntlr
    PARAMETER INSTANCE = plbv46_dvi_cntlr_0
    PARAMETER HW_VER = 1.00.a
    PARAMETER C_DCR_BASEADDR = 0b0000000000
    PARAMETER C_DCR_HIGHADDR = 0b0000000001
    PARAMETER C_DEFAULT_TFT_BASE_ADDR = 0b00000011111
    PARAMETER C_BASEADDR = 0xc9800000
    PARAMETER C_HIGHADDR = 0xc980ffff
    BUS_INTERFACE MPLB = plb
    BUS_INTERFACE SPLB = plb
    BUS_INTERFACE SDCR = dcr_v29_0
    PORT SYS_dcrClk = sys_clk_s
    PORT SYS_tftClk = CLK_25MHZ
    PORT tft_lcd_b = tft_lcd_b & nc_tft_b0
    PORT tft_lcd_clk = tft_lcd_clk
    PORT tft_lcd_g = tft_lcd_g & nc_tft_g0
    PORT tft_lcd_hsync = tft_lcd_hsync
    PORT tft_lcd_r = tft_lcd_r & nc_tft_r0
    PORT tft_lcd_vsync = tft_lcd_vsync
END
```

Na začátku je uveden název IP jádra, dále název instance tohoto jádra a jeho verze. Dále již následují položky, nastavující jednotlivé parametry jádra. V tomto případě adresy DCR registrů, defaulte nastavená adresa ze které kontrolér začíná číst a další parametry. Řádek `BUS_INTERFACE MPLB = plb` znamená, že jádro je k procesoru připojené pomocí sběrnice plb, obdobně pro další sběrnice. Nakonec je

ještě uvedena definice vstupních a výstupních portů. V souboru MHS kromě definic jader, která jsou součástí projektu dále jsou uvedeny definice externích portů. Tyto porty jsou napojeny přímo na piny FPGA obvodu a na jejich definici navazuje definice v UCF souboru. Přidání externího portu je možné opět provést přímým zápisem do MHS souboru. Definice portů pro tft controler potom vypadá následovně:

```
PORT tft_lcd_b = tft_lcd_b, VEC = [5:1], DIR = 0
PORT tft_lcd_clk = tft_lcd_clk, DIR = 0
PORT tft_lcd_g = tft_lcd_g, VEC = [5:1], DIR = 0
PORT tft_lcd_hsync = tft_lcd_hsync, DIR = 0
PORT tft_lcd_r = tft_lcd_r, VEC = [5:1], DIR = 0
PORT tft_lcd_vsync = tft_lcd_vsync, DIR = 0
```

### 5.3.2 MSS

Tento soubor obsahuje informace o softwarové konfiguraci projektu.

```
BEGIN DRIVER
  PARAMETER DRIVER_NAME = generic
  PARAMETER DRIVER_VER = 1.00.a
  PARAMETER HW_INSTANCE = plbv46_dvi_cntlr_0
END
```

Určuje nám, že k instanci jménem `plbv46_dvi_cntlr_0`, bude přiřazen implicitní ovladač. Dále jsou v souboru MSS uvedeny definice knihoven, které chceme přidat do projektu.

### 5.3.3 UCF

Zde se k externím portů přiřazují konkrétní piny FPGA obvodu. Rozdílem oproti předchozím dvěma souborům je, že XPS neposkytuje funkce pro práci s UCF soubory pomocí grafického uživatelského rozhraní. Jedinou možností je tedy v tomto případě přímá editace souboru.

```
NET tft_lcd_b<1> LOC = C5 | DRIVE = 8 | SLEW = FAST;
NET tft_lcd_b<2> LOC = C7 | DRIVE = 8 | SLEW = FAST;
NET tft_lcd_b<3> LOC = B7 | DRIVE = 8 | SLEW = FAST;
NET tft_lcd_b<4> LOC = G8 | DRIVE = 8 | SLEW = FAST;
NET tft_lcd_b<5> LOC = F8 | DRIVE = 8 | SLEW = FAST;
NET tft_lcd_g<1> LOC = E4 | DRIVE = 8 | SLEW = FAST;
NET tft_lcd_g<2> LOC = D3 | DRIVE = 8 | SLEW = FAST;
NET tft_lcd_g<3> LOC = H7 | DRIVE = 8 | SLEW = FAST;
NET tft_lcd_g<4> LOC = H8 | DRIVE = 8 | SLEW = FAST;
NET tft_lcd_g<5> LOC = C1 | DRIVE = 8 | SLEW = FAST;
```

```
NET tft_lcd_r<1> LOC = C2 | DRIVE = 8 | SLEW = FAST;
NET tft_lcd_r<2> LOC = G7 | DRIVE = 8 | SLEW = FAST;
NET tft_lcd_r<3> LOC = F7 | DRIVE = 8 | SLEW = FAST;
NET tft_lcd_r<4> LOC = E5 | DRIVE = 8 | SLEW = FAST;
NET tft_lcd_r<5> LOC = E6 | DRIVE = 8 | SLEW = FAST;
NET tft_lcd_vsync LOC = A8 | DRIVE = 8 | SLEW = FAST;
NET tft_lcd_hsync LOC = C10 | DRIVE = 8 | SLEW = FAST;
NET tft_lcd_clk LOC = AF8 | IOSTANDARD = LVDCI_33 |
DRIVE = 8 | SLEW = FAST;
```

### 5.3.4 Tvorba uživatelských periférií

Kromě periférií integrovaných v XPS je možné do systému přidávat i vlastní periférie napsané v jazyce VHDL nebo Verilog. Tuto periférii můžeme připojit k procesoru pomocí sběrnice PLB nebo pomocí FSL (Fast Simplex Link). Při vytváření periférie je možné využít průvodce, kterého spustíme přes položku v menu **Hardware®Create or Import Peripheral...** . Periférii je možné vytvořit i bez použití průvodce. Struktura adresáře se soubory periférie je následující :

#### **\data**

Obsahuje soubor **jmeno\_periferie\_verze.mpd** (Microprocessor Peripheral Definition), ve kterém je definice rozhraní z pohledu periférie (parametry, porty). Potom je zde soubor **jmeno\_periferie\_verze.pao** (Peripheral Analyze Order), který obsahuje seznam souborů nutných pro syntézu periférie. Dále může obsahovat soubor **jmeno\_periferie\_verze.tcl**, obsahující skript v jazyce Tcl, což je programovací jazyk založený na Lispu. Tyto skripty slouží například ke kontrole správnosti uživatelem nastavovaných parametrů.

#### **\hdl\vhdl**

Zde jsou soubory ve VHDL, které zajišťují vlastní funkčnost periférie.

#### **\hdl\verilog**

Obsahuje soubory ve Verilogu. Periférie může být napsána v samotném VHDL, v samotném Verilogu a nebo může být složena z obou typů souborů.

#### **\doc**

Může obsahovat datasheet k periférii **jmeno\_periferie.pdf**.

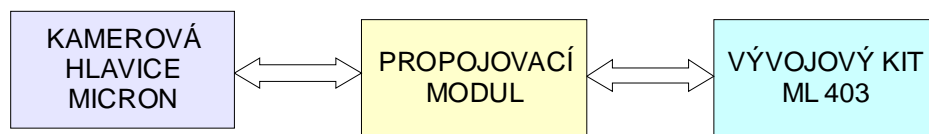


Vytvořenou periferii nakopírujeme do adresáře **pcores** v projektovém adresáři, aby se objevila na záložce **IP Catalog** a bylo ji možné přidat do systému.

## 6. OBVODOVÁ REALIZACE KAMERY

### 6.1 PLATFORMA

Základní schéma kamery je znázorněno na obrázku 6.1.



**Obrázek 6.1:** Schéma kamery

Celá platforma je založena na vývojovém kitu ML403 od firmy XILINX jenž je osazen hradlovým polem FPGA řady VIRTEX 4 FX (XC4VFX12), které obsahuje procesorové jádro PowerPC 405, 36 blokových pamětí (BRAM) o velikosti 18 kB, 320 vstupně výstupních pinů, 5472 logických elementů. Popis tohoto modulu je uveden v kapitole 5. Tento kit byl zvolen z důvodů snadné dostupnosti. Propojení kamerové hlavice a kitu ML 403 je realizováno propojovacím modulem vlastní výroby.

#### 6.1.1 Snímací element

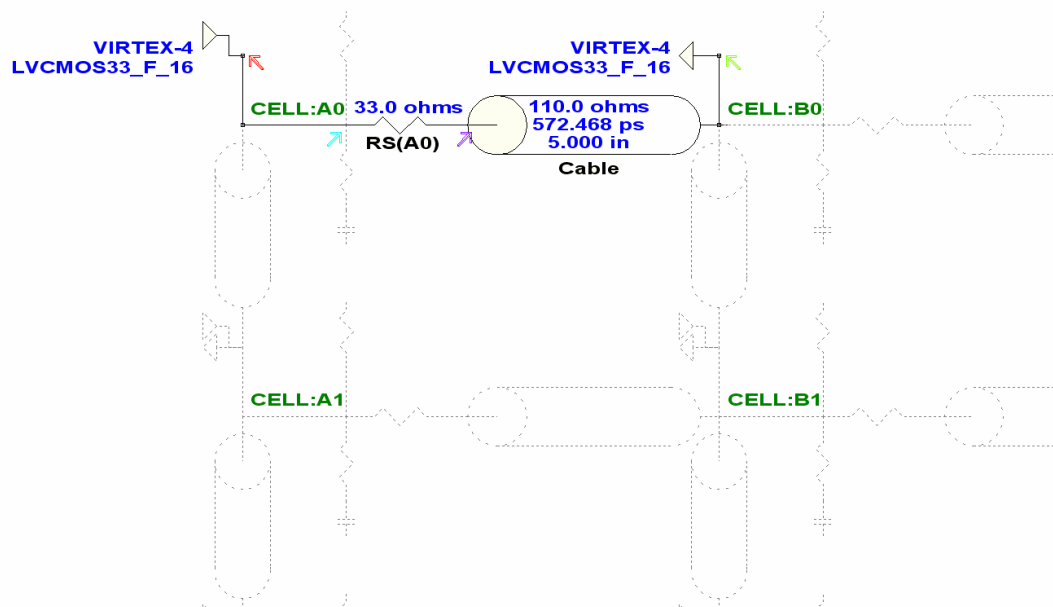
Jako snímací element je použita kamerová hlavice firmy Micron, která obsahuje CMOS čip MT9V125 s rozlišením 640x480 pixelů se snímací frekvencí 25 snímků za sekundu. Velkou výhodou tohoto čipu je, že poskytuje několik výstupních formátů dat a rozhraní. Je použito výstupu RGB 565, který generuje obrazová data v bitové hloubce 5 bitů červená, 6 bitů zelená a 5 bitů modrá barva. Všechny datové a řídicí signály jsou přivedeny přes propojovací modul do funkčního bloku řadiče kamery.

Po resetu je potřeba provést inicializaci čipu a provést konfiguraci příslušných registrů. To je provedeno přes standardní IIC rozhraní. Nejprve je potřeba nastavit požadovaný výstupní formát dat (RGB 565) zapsáním hodnoty do registru a poté je nutné provést další nastavení registrů jež výrobce doporučuje v [29]. Lze rovněž nastavit pořadí vysílaných dat, čímž lze otáčet obraz jak vertikálně

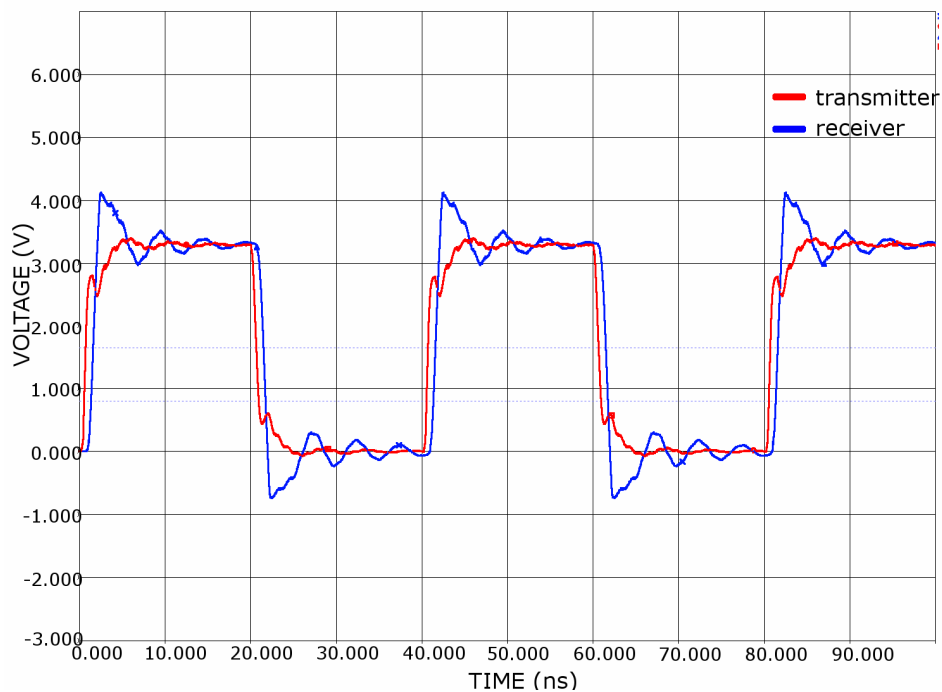
tak i horizontálně. Také je možno u čipu nastavit generování testovacích obrazců, jež můžou být horizontální pruhy, popřípadě vertikální pruhy, tak posloupnost definovaných snímků. Této vlastnosti bylo využito pro ověření funkčnosti kamery. Kompletní popis čipu je v [28].

### 6.1.2 Propojovací modul

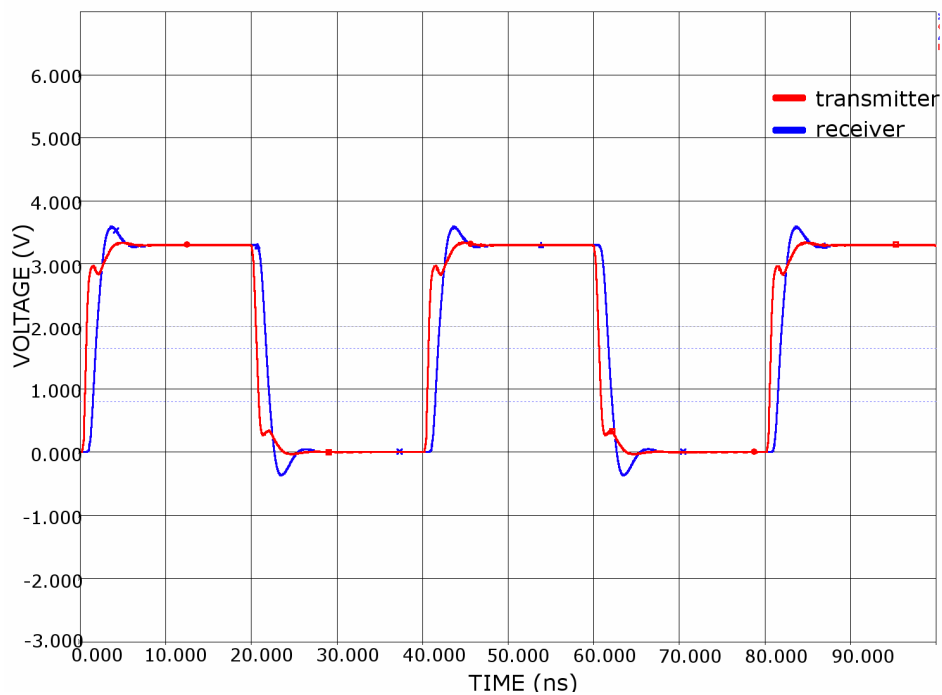
Slouží k propojení všech datových a řídicích signálů mezi kamerovou hlavicí a vývojovou deskou ML 403. Z důvodu impedančně nepřizpůsobeného zakončení může docházet k odrazům na vedení, proto je potřeba tento jev potlačit. Více je uvedeno v [31]. K potlačení tohoto jevu jsou do signálových cest sériově zapojeny odpory. K určení neoptimálnější velikosti těchto odporů bylo využito vývojového prostředí HyperLynx od firmy MENTOR. Simulací signálu o frekvenci 25 MHz (frekvence hodinového signálu) vyšlo nejlépe zapojení se sériově zapojenými odpory o velikosti 33R. Dál je modul osazen jedním blokovacím kondenzátorem, který má za úkol pokrýt spotřebu kamerové hlavice v době nárazového odběru (klopení několika hradel současně), proto je zapojen co nejbližže jejího konektoru. Schéma a seznam součástek tohoto modulu je uveden v příloze.



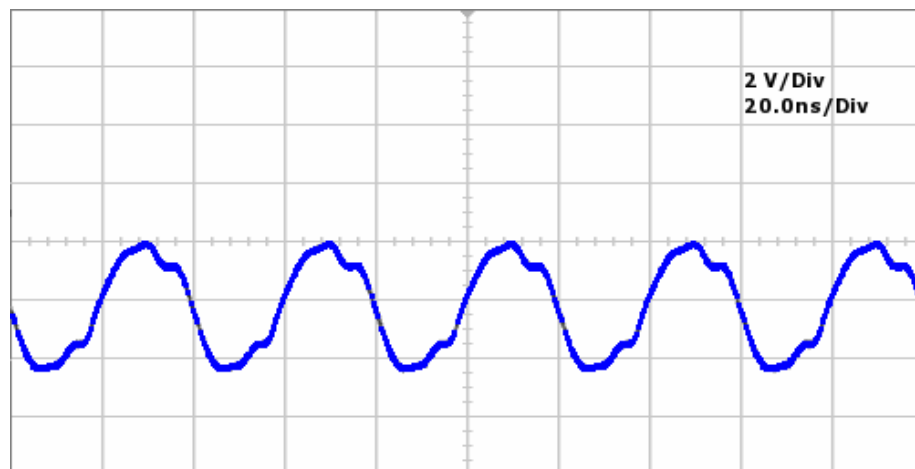
**Obrázek 6.2:** Simulační schéma v programu HyperLynx



**Obrázek 6.3:** Simulace signálu na vodiči bez odporu



**Obrázek 6.4:** Simulace signálu na vodiči s odporem

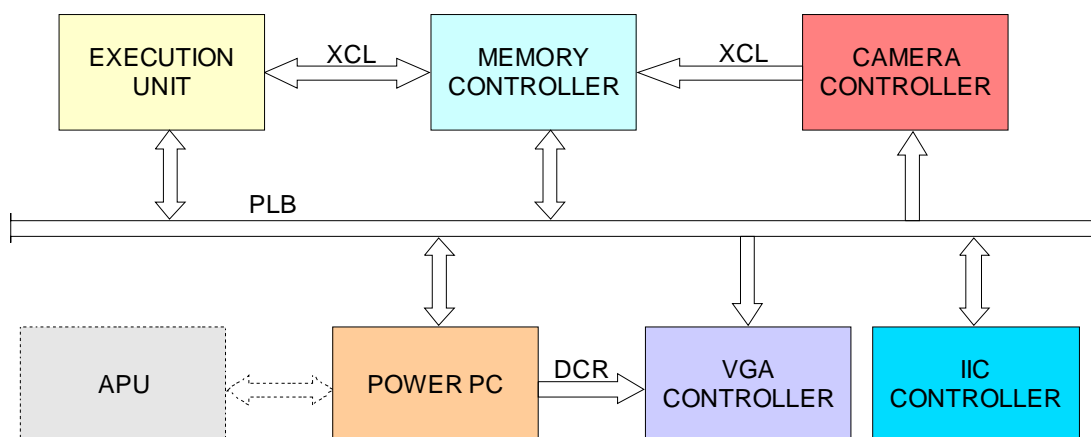


**Obrázek 6.5:** Skutečný průběh signálu na vodiči

## 6.2 ARCHITEKTURA V FPGA

Architektura inteligentní kamery je znázorněna na obrázku 6.6. Základním komunikačním uzlem celé architektury je sběrnice typu PLB verze 46. Byla vybrána z důvodu dostatečné datové propustnosti jenž činí 800 MB/s při použité datové šířce 64 bit a taktovací frekvenci 100 MHz. Ke komunikaci mezi pamětí a některými samostatnými jednotkami byla použita sběrnice XCL z důvodů menší komunikační režie, ale s dostačující přenosovou rychlostí 400 MB/s při frekvenci 100 MHz. Dále byla použita sběrnice typu DCR, která obsluhuje řadič VGA a byla zvolena z důvodů nízké náročnosti na systémové zdroje. Řízení jednotlivých bloků obstarává procesor PowerPC 405.

Jako zdroj hodinového signálu je použit oscilátor generující signál s frekvencí 100 MHz. Tento signál vstupuje do DCM bloku. Jeho výstupem je signál 100 MHz, na kterém pracují veškeré vnitřní sběrnice a funkční bloky kamery a dále je generován signál 25 MHz, který je určen pro VGA řadič a také je využit jako zdroj hodinového signálu pro snímací jednotku. Poslední generovanou skupinou signálů jsou hodiny s frekvencí 200 MHz a 100 MHz ale o 90 stupňů posunuté, které slouží pro řízení DDR SDRAM paměti.



**Obrázek 6.6:** Architektura kamery v čipu FPGA

### 6.2.1 IIC CONTROLLER

Jak již bylo zmíněno po resetu je potřeba provést inicializaci a konfiguraci CMOS čipu, která probíhá přes standardní IIC rozhraní. Za tímto účelem je do architektury zakomponován IIC řadič, který slouží jako interface mezi IIC sběrnici a PLB sběrnici.

IIC sběrnice je obousměrná složená pouze ze dvou vodičů - "data SDA (serial data)" a "hodiny SCL (serial clock)". Na jednu sběrnici může být připojeno více integrovaných obvodů. V základní verzi jsou obvody adresovány 7bitově. To umožňuje připojení 128 čipů s různou adresou na jednu společnou sběrnici. Frekvence hodin IIC sběrnice v základní verzi je 100kHz. Ve vylepšených verzích to může být 400kHz nebo 1MHz. Oba vodiče musí být implicitně v logické jedničce a to je zajištěno pull-up rezistory. Jejich odpory mají hodnotu v řádech jednotek kiloohmů. Čím je vyšší komunikační frekvence, tím musí být nižší hodnoty těchto odporů. Tyto rezistory jsou součástí kamerové hlavičky a tudíž je nebylo potřeba osazovat.

Princip přenosu je následující: Jeden obvod je nastaven jako MASTER a všechny ostatní obvody jsou SLAVE. Master při jakémkoli přenosu generuje hodinový signál na vodiči SCL. Když master vysílá, přijímají všechny ostatní obvody a pouze podle adresy určují, zda jsou data určena jim. Čip, který chce vyslat/přijmout data musí nejprve definovat adresu čipu, s kterým chce komunikovat

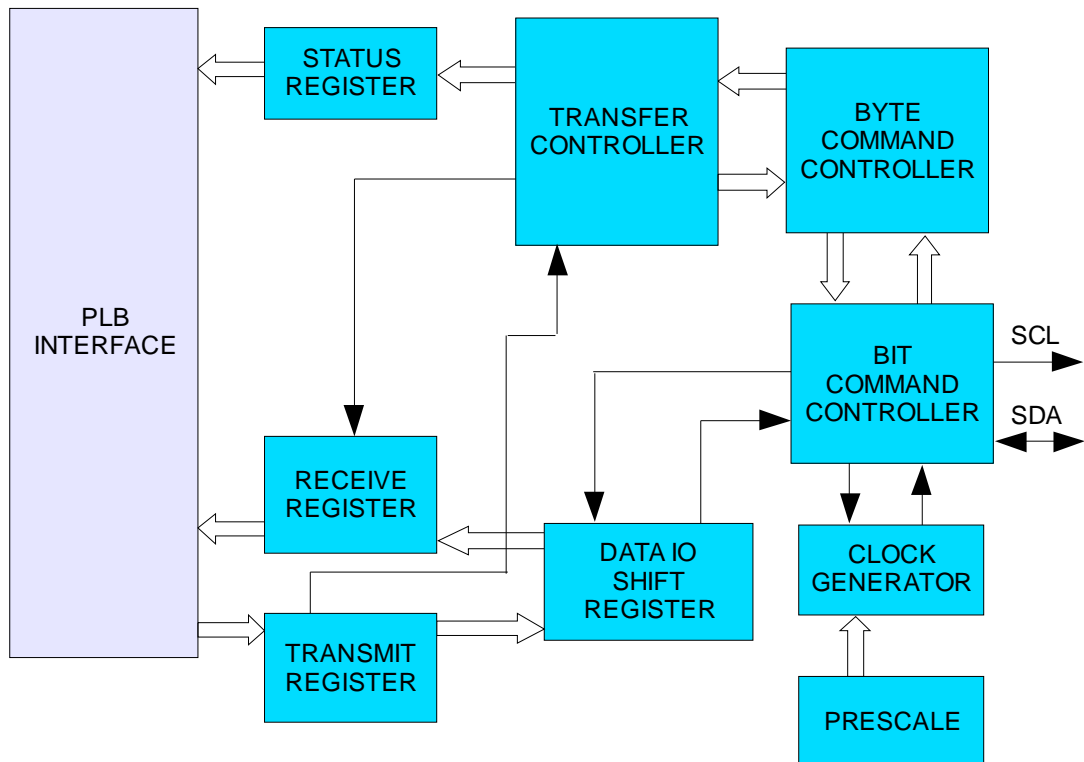
a zda půjde o příjem nebo vysílání - tedy o čtení nebo zápis. To určuje R/W (read/write) bit, který je součástí adresy.

Přenos je zajištěn kombinací následujících celků:

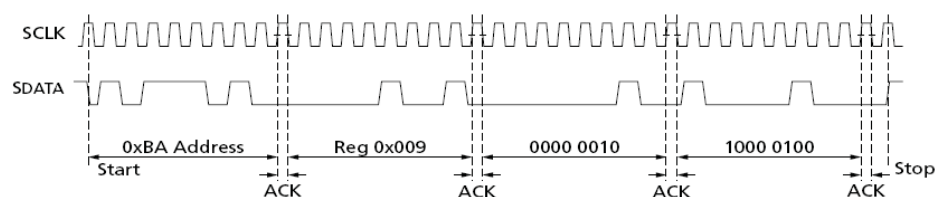
- **stav klidu** - Je zajištěn logickými jedničkami na obou vodičích, master tedy negeneruje hodinový signál a neprobíhá žádný přenos. Logické jedničky jsou na obou vodičích zajištěny pull-up, takže klidový stav nastane i pokud jsou výstupy obvodu master ve stavu vysoké impedance.
- **start bit** - Zahajuje přenos nebo jeho další část. Je vygenerován tak, že se změní úroveň SDA z 1 na 0 zatímco je SCL v logické 1.
- **stop bit** - Ukončuje přenos. Je vygenerován podobně jako start bit. Logická úroveň SDA se změní z 0 na 1 zatímco je SCL v logické 1. Stop bit může být generován pouze po "nepotvrzení přenosu", tedy pouze po přijmutí Ack v logické 1.
- **přenos dat** - Data jsou přenášena po 1 bajtu tedy 8 po sobě jdoucích bitů od nejvyššího po nejnižší. Při přenosu dat se může logická úroveň na SDA měnit pouze pokud je SCL v logické 0. Při každém pulzu na SCL je přenesen jeden bit.
- **potvrzující bit Ack** (acknowledge) - Tento bit slouží k potvrzení správného přijmutí dat. Ack bit se odesílá stejným způsobem jako by se odesílal devátý bit dat, ale s tím rozdílem, že ho generuje čip, který přijímal (přijímač) a nikoliv ten který data odesílal. Pokud přenos proběhl v pořádku tak odešle logickou 0. Logická 0 potvrzujícího bitu znamená rovněž to, že je přijímač připraven na příjem dalšího byte, který následuje okamžitě po něm při dalším pulzu na SCL. Pokud přenos selhal odešle logickou 1. Nebo pokud má dojít k ukončení přenosu, tak "neodešle nic". Pull-up rezistor pak zajistí, že bude na SDA logická 1 a Ack bit (v logické 0) odešle vysílač.

Blokové schéma řadiče je znázorněno na obrázku 6.7. Chceme-li zapisovat nebo číst z IIC sběrnice je potřeba zapsat příslušnou hodnotu do vysílacího registru. K indikaci stavů slouží příznakový registr. Je-li požadováno čtení ze sběrnice IIC je přečtená hodnota uložena do přijímacího registru. Na základě komunikačního protokolu pro zápis, který je znázorněn na obrázku 6.8, nebo čtení, který je na obrázku 6.9, řídí řadič přenosu posloupnost vysílaných bajtů a předává je do řadiče

bajtů. Ten na základě komunikačních celků řídí řadič bitů, jenž řídí logické úrovně na sběrnici. Rychlost IIC sběrnice je nastavena na 100 kHz a je odvozena z předděliče. Nastavení předděliče určuje generická konstanta a tím můžeme změnit rychlost IIC sběrnice. Vývojové diagramy obou řadičů jsou na obrázcích 6.10 a 6.11.

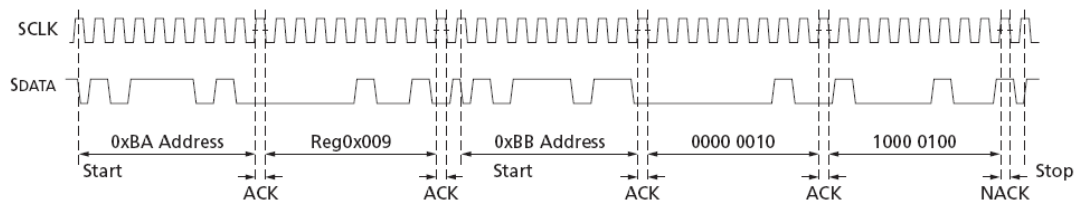


Obrázek 6.7: Architektura IIC řadiče



Obrázek 6.8: Zápis přes IIC sběrnici





Obrázek 6.9: Čtení přes IIC sběrnici

Tabulka 6.1: Vysílací registr IIC řadiče

TRANSMIT REGISTER		BASE_ADRESS + 0x00
BIT #	PŘÍSTUP	POPIS
31:25	W	SLAVE ADRESA
24	W	„0“ – ZÁPIS „1“ – ČTENÍ
23:16	W	ADRESA REGISTRU
15:0	W	ZAPISOVANÁ HODNOTA

RESET VALUE 0x00000000

Tabulka 6.2: Přijímací registr IIC řadiče

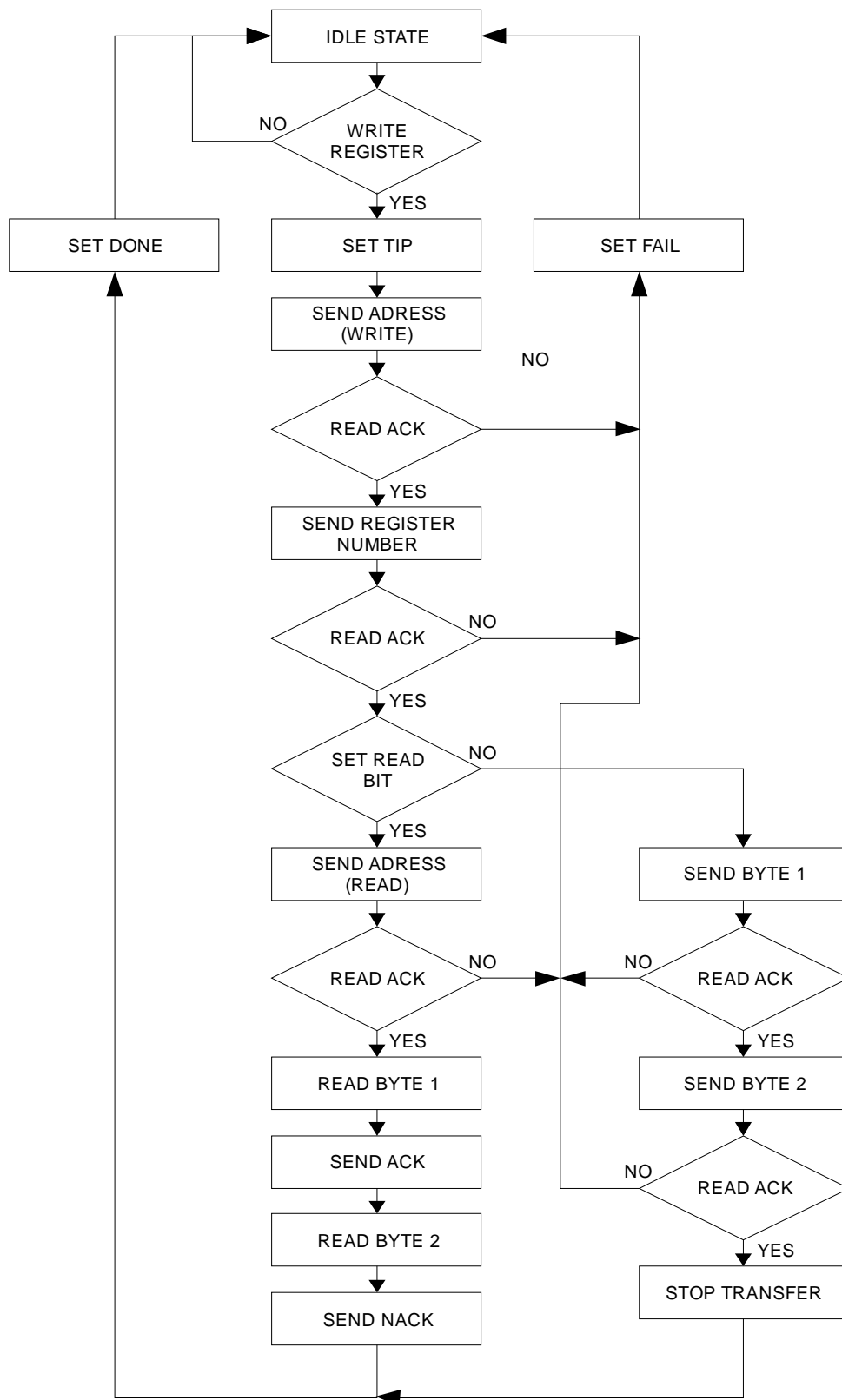
RECEIVE REGISTER		BASE_ADRESS + 0x04
BIT #	PŘÍSTUP	POPIS
31:16	R	REZERVOVANO
15:0	R	PŘIJATÁ DATA

RESET VALUE 0x00000000

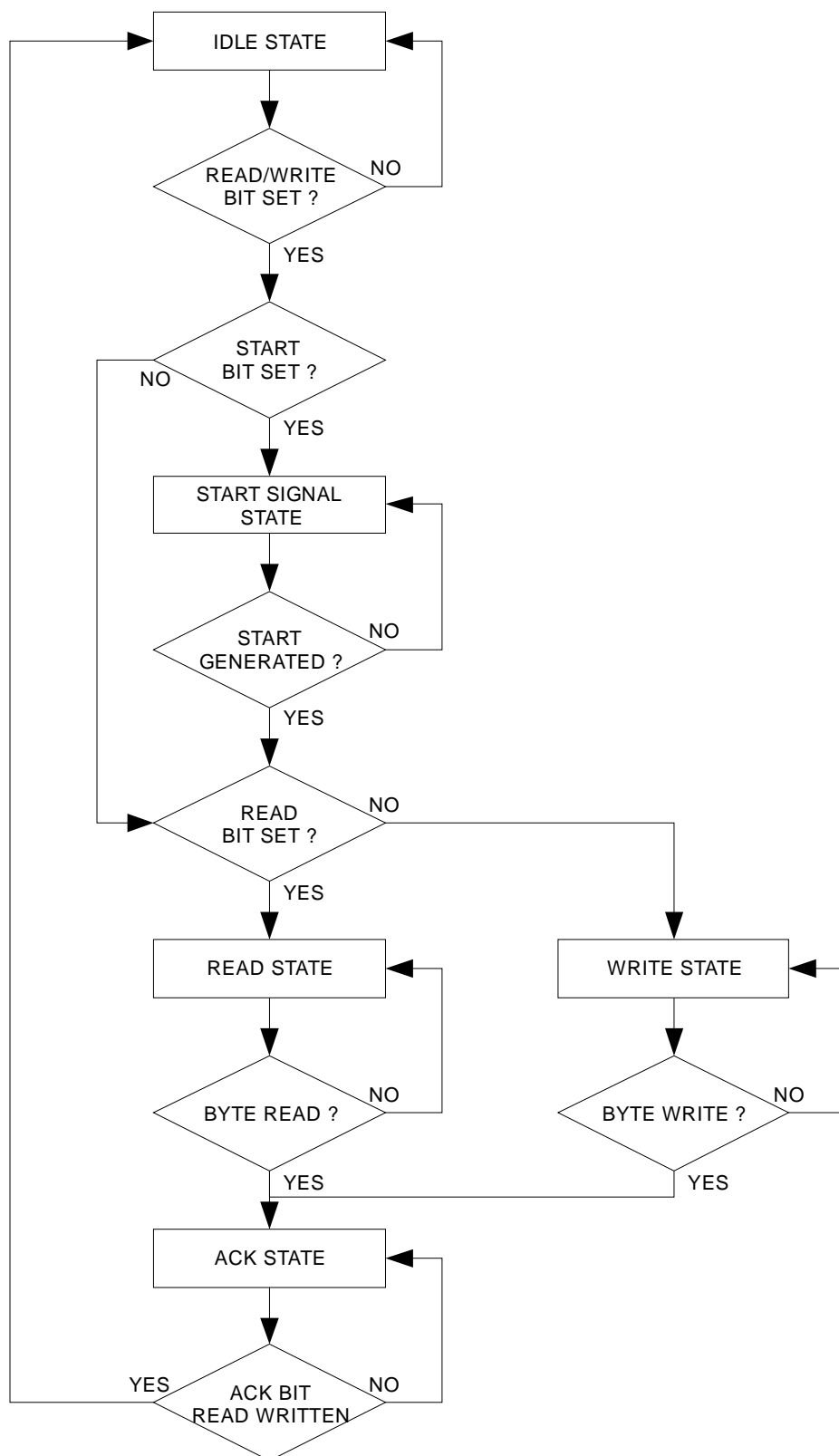
Tabulka 6.3: Příznakový registr IIC řadiče

STATUS REGISTER		BASE_ADRESS + 0x08
BIT #	PŘÍSTUP	POPIS
31:3	R	REZERVOVANO
2	R	FAIL – selhal prenos
1	R	TIP – probíha prenos
0	R	DONE – hotovo

RESET VALUE 0x00000000



Obrázek 6.10: Vývojový diagram řadiče přenosu



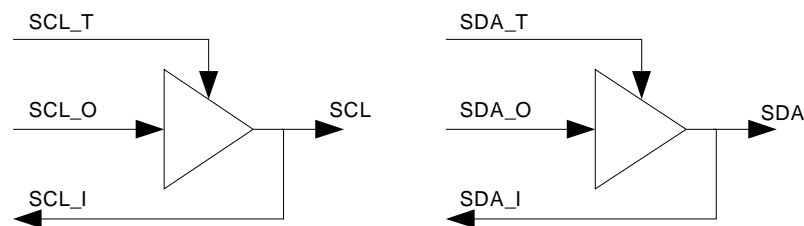
Obrázek 6.11: Vývojový diagram řadiče bajtů

Signály SCL i SDA jsou obousměrné, proto je nutné nadefinovat třístavový budič sběrnice. To je provedeno v mpd souboru této entity následujícím způsobem:

```
PORT Sda = "", DIR = IO, THREE_STATE = TRUE, TRI_I =
Sda_I, TRI_O = Sda_O, TRI_T = Sda_T, PERMIT = BASE_USER,
IO_IF = iic_0, IO_IS = Serial_Data
PORT Sda_I = "", DIR = I
PORT Sda_O = "", DIR = O
PORT Sda_T = "", DIR = O
```

```
PORT Scl = "", DIR = IO, THREE_STATE = TRUE, TRI_I =
Scl_I, TRI_O = Scl_O, TRI_T = Scl_T, PERMIT = BASE_USER,
IO_IF = iic_0, IO_IS = Serial_Clock
PORT Scl_I = "", DIR = I
PORT Scl_O = "", DIR = O
PORT Scl_T = "", DIR = O
```

Signál `_O` je výstupní, `_I` je vstupní a signálem `_T` je určeno zda-li je port vstupní nebo výstupní.



**Obrázek 6.12:** Realizace obousměrných pinů

## 6.2.2 CAMERA CONTROLLER

Řadič kamery obstarává komunikaci se snímací jednotkou. Do řadiče vstupují a vystupují všechny datové a řídicí signály kamerové hlavičky.

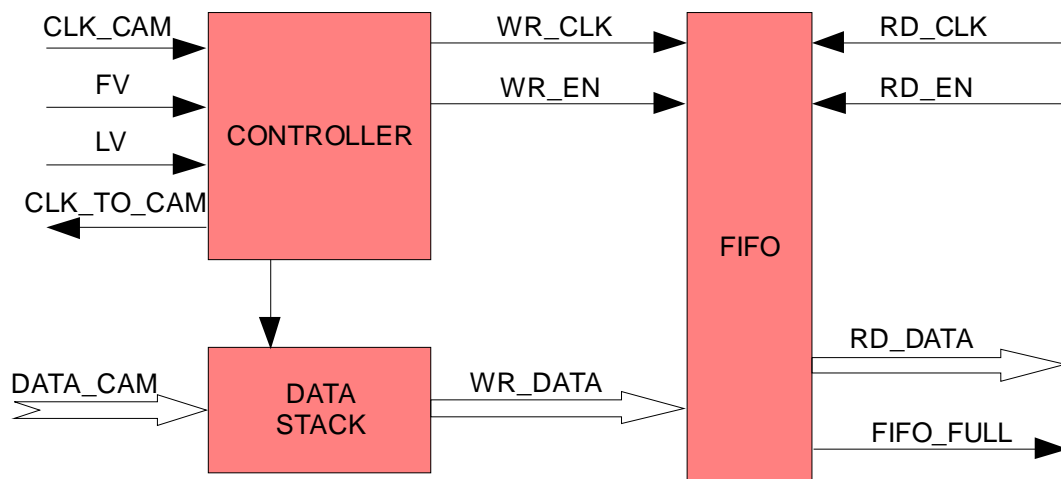
Výstupní signály z řadiče:

- **RESET** – Slouží k externímu resetu CMOS čipu.
- **CLK\_TO\_CAM** – Vstupní hodinový signál do čipu s frekvencí 25 MHz.

Vstupní signály do řadiče:

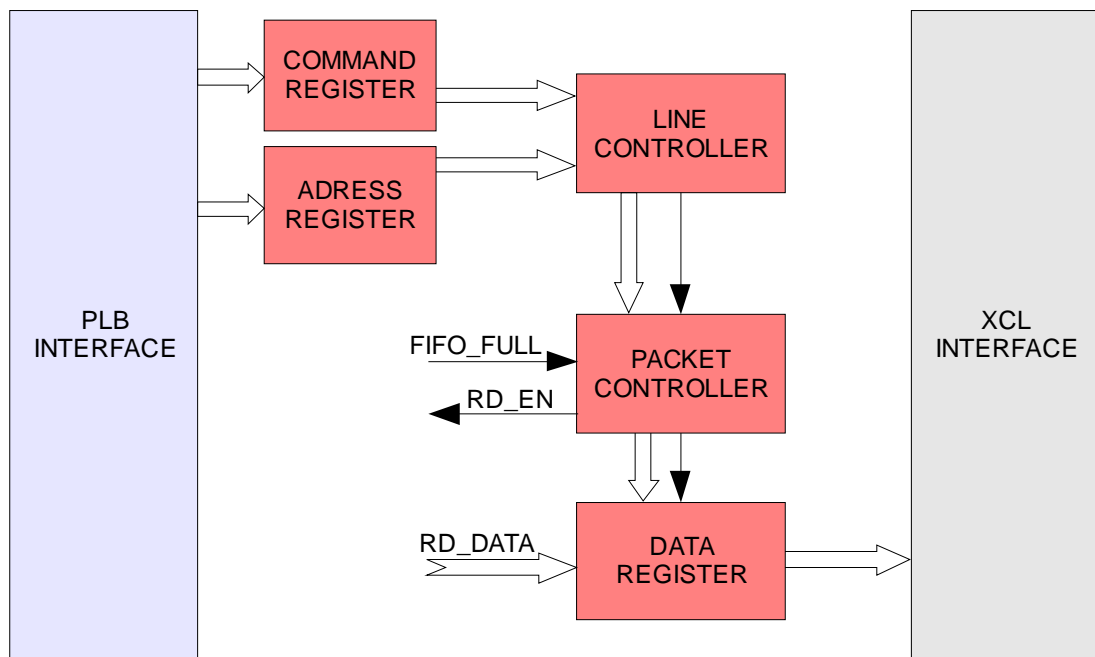
- **CLK\_CAM** – Výstupní hodinový signál z čipu s frekvencí 25 MHz, ale je oproti signálu CLK\_TO\_CAM fázově opožděn.

- **FV** – Indikuje platnost snímku.
- **LV** – Indikuje platnost řádku
- **DATA\_CAM** – Výstupní obrazová data čipu s datovou šířkou až 10 bitů. Při použitím výstupním formátu RGB 565 je využito jen horních 8 bitů.



**Obrázek 6.13:** Realizace zápisu dat do fifa

Do snímací jednotky je potřeba vysílat hodinový signál s frekvencí 25 MHz. Čip vysílá v prvním bajtu 5 bitů červené barvy a horní 3 bity zelené barvy, v následujícím bajtu spodní 3 bity zelené barvy a 5 bitů modré barvy. Data jsou platná jen tehdy jsou-li aktivní současně signály FV a LV. Tyto data jsou složena a jako 16 bitová jsou zapsána do fifa. Zápis do fifa je rychlostí 12,5 MHz a vyčítání dat je rychlostí 100 MHz, navíc jsou zapisovací a čtecí hodiny vůči sobě fázově posunuty, z tohoto důvodu je použito asynchronního fifa i za cenu vyšších systémových zdrojů. K vytvoření fifa bylo využito nástroje IP COREGEN. Dojde-li k naplnění fifa na hodnotu 320 je nastaven signál fifo\_full a spuštěno vyčítání a zápis těchto dat do paměti. Přenos trvá tak dlouho dokud se fifo nevyprázdní a celý proces se opakuje.



**Obrázek 6.14:** Realizace zápisu dat do paměti přes XCL sběrnici

**Tabulka 6.4:** Adresový registr řadiče kamery

ADDRESS REGISTER		BASE_ADDRESS + 0x00
BIT #	PŘÍSTUP	POPIS
31:0	R/W	ADRESA ZAPISU
RESET VALUE 0x00000000		

**Tabulka 6.5:** Řídicí registr řadiče kamery

COMMAND REGISTER		BASE_ADDRESS + 0x08
BIT #	PŘÍSTUP	POPIS
31	W	„0“ – STOP „1“ – POVOLENO
30:0	R/W	REZERVOVANO
RESET VALUE 0x00000000		

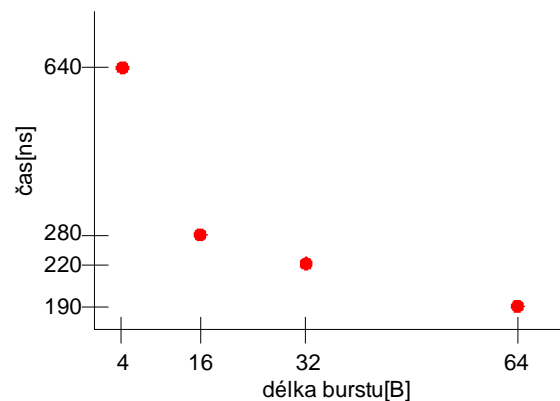
Ovládání řadiče je přes sběrnici PLB, takto lze nastavit cílovou adresu ukládání dat, spustit nebo zastavit přenos dat. Adresa ukládání dat je uložena v adresovém registru, řízení je pak umožněno přes řídicí registr. Kamerový senzor vysílá nejprve posloupnost lichých a pak sudých řádků snímku, proto je nutné je přeskládat. Toto má na starosti řadič řádků, který řídí adresy zápisů. Při dostatečném naplnění zásobníku jsou data vyčítaná, vhodně uspořádá do paketů a

přes sběrnici XCL zapisovaná do paměti. Při tomto řešení je dosaženo snížení datového toku, o adresovou režii, než kdyby se zapisoval každý pixel samostatně. V paměti je jeden pixel reprezentován 4 bajty.

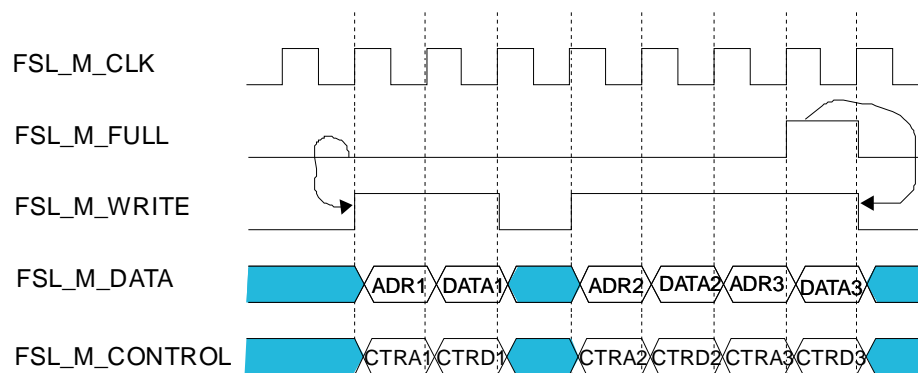
**Tabulka 6.6:** Uložení dat v paměti

BIT #	COLOR
31:24	
23:16	RED
15:8	GREEN
7:0	BLUE

XCL sběrnice je jednosměrný point to point kanál mezi dvěma komunikujícími jednotkami, z nichž jedna je typu master a druhá typu slave. Princip zápisu přes sběrnici je znázorněn na obrázku 6.16. Chceme-li zapisovat je nutné aby byl signal FSL\_M\_FULL v logické „0“. Při zápisu je nutné nastavit signál FSL\_M\_WRITE do logické „1“, a vystavit data na FSL\_M\_DATA. Jelikož jsou data i adresa 4 bajtové signálem FSL\_M\_CONTROL je určeno zdali zapíšeme jen jeden bajt, nebo všechny čtyři jež sou součástí této adresy. Zapisovat data můžeme dvěma způsoby, buďto jednoduchým způsobem, avšak komunikačně náročnějším, kde je nejprve na datovou sběrnici vystavena adresa a v dalším taktu vysílaná data, nebo využít paketového přenosu kdy vystavíme adresu a pak přesně definovanou posloupnost dat, která jsou ukládána na následující adresy. Tohoto způsobu bylo využito i zde, kde při zápisu vyšleme adresu a následujících 16 taktů data. Tím byla zkrácena doba pro zápis dat téměř na čtvrtinu než kdybychom zapisovali prvním způsobem. Časovou náročnost pro zapsání 64 bajtů dat v závislosti na typu přenosu ukazuje obrázek 6.15.



**Obrázek 6.15:** Závislost doby přenosu 64 bytů v závislosti na délce burstu



**Obrázek 6.16:** Zápis přes XCL sběrnici

### 6.2.3 VGA CONTROLLER

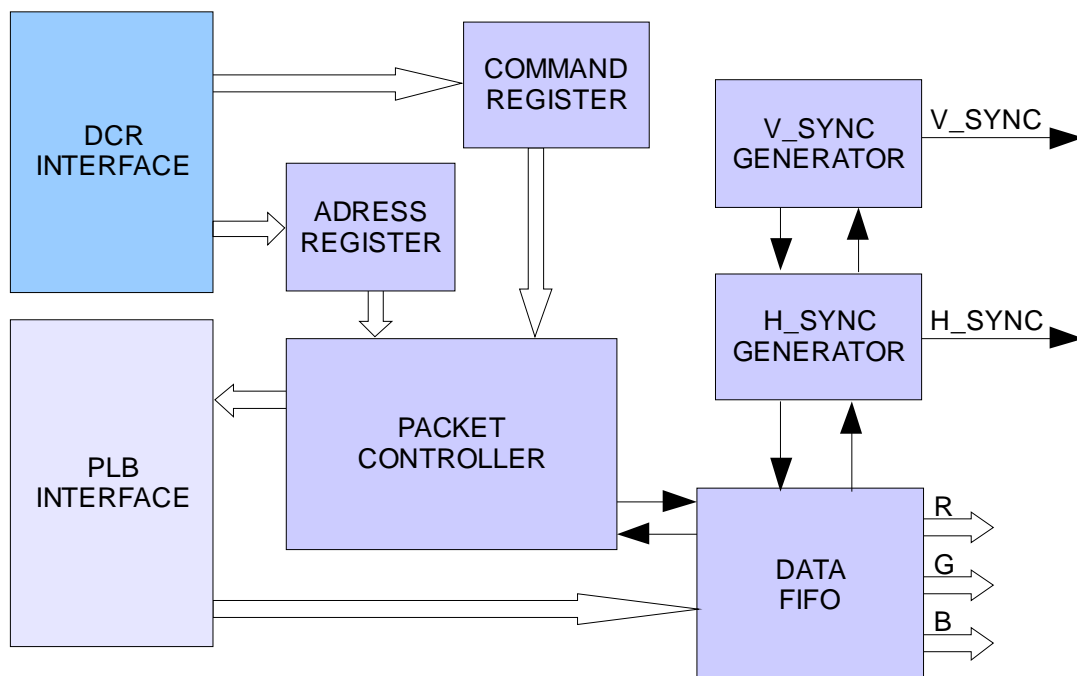
Pro zobrazení zpracovaných dat slouží VGA řadič, který čte přes sběrnici PLB data z paměti. Jeho výstupem je VGA rozhraní s rozlišením 640x480 pixelů se snímkovou frekvencí 60 Hz, k němuž je možno připojit standardní monitor. Při daném rozlišení a snímkové frekvenci je potřebná frekvence obrázkových bodů 25 MHz, vypočet viz rovnice (6). Ovládání řadiče je umožněno přes rozhraní DCR, tím lze nastavit, která oblast v paměti je čtena a zobrazena. Schéma řadiče je znázorněno na obrázku 6.17. Přes PLB sběrnici je natažen celý řádek do BRAM paměti a poté postupně vyslán přes VGA rozhraní na monitor. Při čtení je opět využito paketového přenosu dat o délce 32 dvojslov. Při tomto rozlišení je vyžadován datový tok na sběrnici PLB 55 MB/s, což sběrnice s velkou rezervou poskytuje.

$$(640 + 16 + 96 + 48) \cdot (480 + 12 + 2 + 31) \cdot 60 = 25\text{MHz} \quad (6)$$



**Tabulka 6.7:** Hodnoty VGA signálů

OZNAČENÍ	HODNOTA	POPIS
H_DISP	640	horizontální rozlišení
H_FPORCH	16	front porch pro horizontální směr
H_SYN	96	délka horizontálního synchronizačního pulzu
H_BPORCH	48	back porch pro horizontální směr
V_DISP	480	vertikální rozlišení
V_FPORCH	12	front porch pro vertikální směr
V_SYN	2	délka vertikálního synchronizačního pulzu
V_BPORCH	31	back porch pro vertikální směr



**Obrázek 6.17:** Schéma VGA řadiče

**Tabulka 6.8:** Adresový registr VGA

ADDRESS REGISTER		BASE_ADRESS + 0x00
BIT #	PŘÍSTUP	POPIS
31:21	R/W	ADRESA ČTENÍ
20:0	R/W	REZERVOVANO

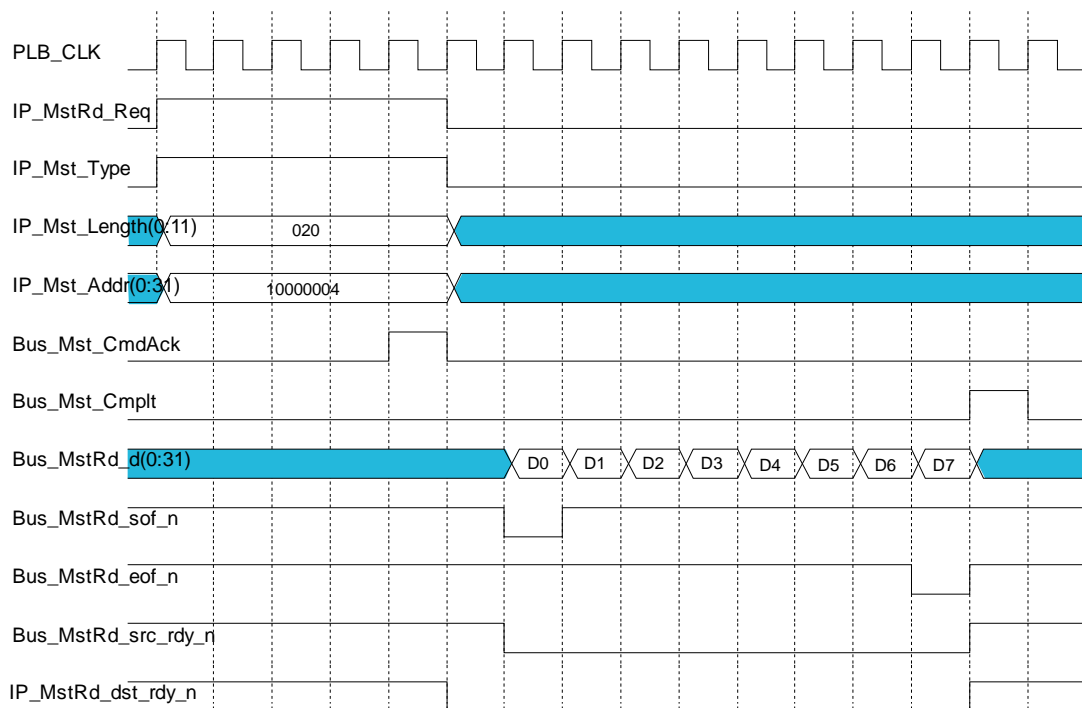
RESET VALUE 0x03E

**Tabulka 6.9:** Řídicí registr VGA

COMMAND REGISTER		BASE_ADRESS + 0x01
BIT #	PŘÍSTUP	POPIS
31:2	R/W	REZERVOVANO
1	R/W	„0“ – STOP
		„1“ – POVOLENO
0	R/W	REZERVOVANO

RESET VALUE 0x0002

Řadič je na sběrnici PLB připojen jako master. Jelikož je sběrnice PLB dosti komplikovaná a v případě připojení více masterů je řešena arbitráž bylo pro připojení využito bridge. Ten umožňuje připojení přes jednodušší rozhraní. Jeho kompletní popis lze nalézt v [36]. Na obrázku 6.18 je zobrazeno čtení z paměti v burst režimu. Současně jsou poslány do bridge signály IP\_MstRd\_Req - požadavek na čtení, IP\_Mst\_Type – je-li v logické „0“ jednoduchý přenos a v logické „1“ burst přenos. Délku burst přenosu určuje IP\_Mst\_Length(0:11). Adresu na které začínáme číst určuje signál IP\_Mst\_Addr(0:31). Tyto signály je potřeba držet tak dlouho dokud nám není zpětně poslán signál Bus\_Mst\_CmdAck. Ukončení přenosu je oznámeno přes Bus\_Mst\_Cmplt. Signál Bus\_MstRd\_sof\_n indikuje začátek a signál Bus\_MstRd\_eof\_n konec posloupnosti požadovaných dat. Oba signály jsou aktivní v logické „0“. Požadována data jsou vystavena na Bus\_MstRd\_d(0:31), platnost těchto dat určuje Bus\_MstRd\_src\_rdy\_n. Nastavením signálu IP\_MstRd\_dst\_rdy\_n je potvrzena schopnost přijímat data. Oba signály jsou aktivní v logické „0“.



**Obrázek 6.18:** Rozhraní poskytnuté bridgem

## 6.2.4 MEMORY CONTROLLER

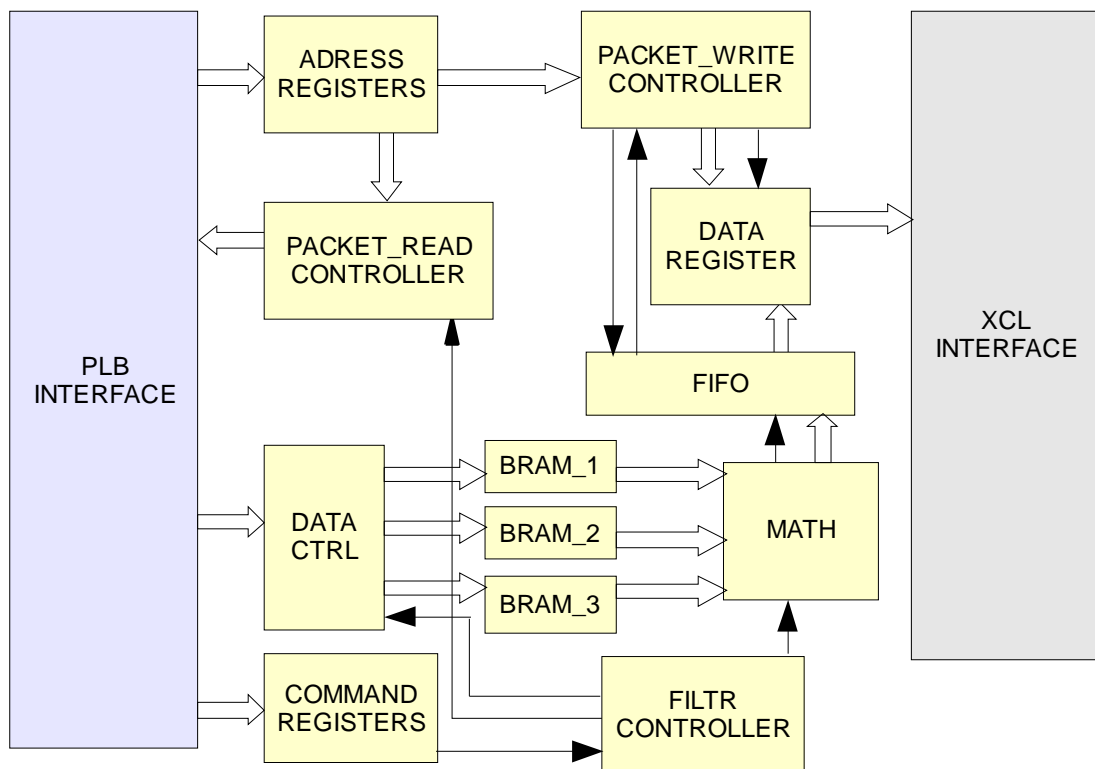
Ke komunikaci s DDR SDRAM paměti slouží řadič paměti. Je využito standardní komponenty MPMC controller jež je součástí vývojového prostředí EDK. Na desce je použita paměť o velikosti 64 MB což je plně dostačující pro uložení obrazových dat (jeden snímek zabírá 0,9 MB paměti). U komponenty byly nastaveny čtyři komunikační porty s adresovým rozsahem 0x00000000 až 0x03FFFFFF. První i druhý port je připojen na sběrnici PLB a je typu slave. Třetí a čtvrtý port je nakonfigurován a připojen na sběrnici XCL a je také typu slave. U těchto portů je nastaven přenos po burstech o velikosti 64 bajtů. Třetí port slouží pro zápis dat z CMOS čipu a čtvrtý port pak pro zápis zpracovaných dat. Popis této komponenty je uveden v [35].

## 6.2.5 EXECUTION UNIT

Výkonná jednotka zajišťuje požadované operace s obrazem, což jsou filtrace, konvoluční filtry a jiné operace nad obrazem. Při filtraci jsou data z paměti postupně vyčítaná, je nad nimi pixel po pixelu provedena matematická operace a poté jsou

zapsána do paměti. Je možno provádět operace nad obrazem přičtení nebo odečtení offsetu a to jak ke všem nebo jen k určitým barevným složkám obrazu, vynásobení obrazu nebo sečtení původního a filtrovaného obrazu. Při použitém čipu je potřeba zpracovávat obrazové body minimálně s rychlostí 7,7 MHz výpočet viz rovnice (7).

$$640 \cdot 480 \cdot 25 = 7,7 \text{ MHz} \quad (7)$$



**Obrázek 6.19:** Schéma výkonné jednotky

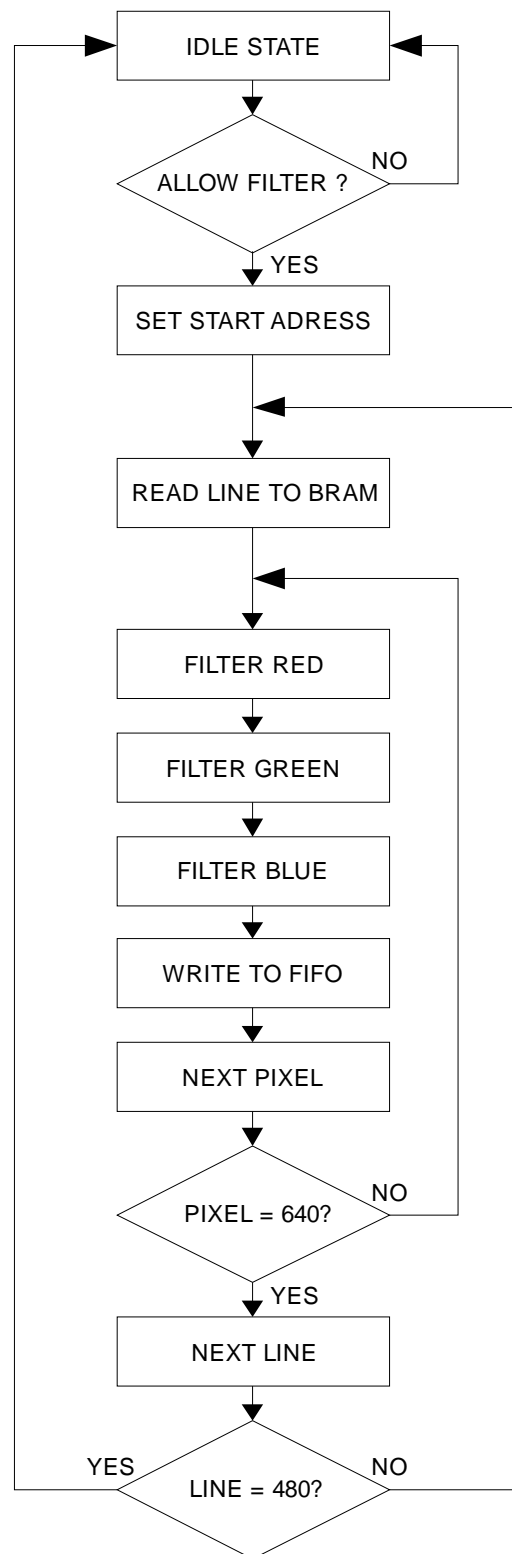
Celé řízení jednotky obstarává řadič filtru, jeho vývojový diagram je na obrázku 6.20. Pomocí adresových registrů lze nastavit oblast čtení i oblast zápisu obrazových dat. Řízení jednotky je pomocí řídicího registru, kterým lze celou jednotku spustit, zastavit a dále nastavit prolínání obou obrazů. Při čtení obrazových dat je vždy celý řádek natažen do samostatné BRAM paměti a následně zpracován. Data z paměti jsou vyčítána přes sběrnici PLB. Z důvodu použití filtrační masky o rozměru 3x3 je potřeba 3 BRAM paměti. Je-li skončena filtrace jednoho řádku,

dojde k přepnutí pořadí BRAM a na místo nejstaršího řádku je natažen nový řádek obrazu.

Jádro jež zajišťuje filtraci, běží na trojnásobné frekvenci než je frekvence vlastního zpracování obrazu, jelikož jednotlivé barevné složky obrazu jsou zpracovány postupně. Toto řešení je zvoleno z důvodů snížení potřebného počtu násobiček na jednu třetinu (9 oproti 27 – maska 3x3). Aby nedocházelo ke zmenšení rozměru filtrovaného obrazu jsou krajní pixely využity pro roztažení obrazu. Hodnoty masky lze libovolně měnit zapsáním hodnoty do příslušného konfiguračního registru filtru. Velikost konfiguračních hodnot je nastavena na 8 bitů, jelikož každá barva je také reprezentována 8 bity, maximální velikost čísla, které vznikne vynásobením konfigurační hodnoty a hodnoty pixelu může být maximálně 16 bitů. Použité násobičky mají vstupní šířku dat 18 bitů a výsledek může mít datovou šířku až 36 bitů.

Další typy matematických operací, které se mají provést nad obrazem závisí od nastavení příslušných řídicích a konfiguračních registrů.

Zpracované obrazové body jsou zapsány do fifo. Zápis do fifo je rychlostí 12.5 MHz a vyčítání dat je rychlostí 100 MHz, zapisovací a čtecí hodiny nejsou vůči sobě fázově posunuty, z tohoto důvodu je použito synchronní fifo. Zápis do paměti probíhá obdobným způsobem jako u řadiče kamery.



**Obrázek 6.20:** Vývojový diagram řadiče filtru

**Tabulka 6.10:** Adresový registr zápisu

ADDRESS REGISTER		BASE_ADDRESS + 0x00
BIT #	PŘÍSTUP	POPIS
31:0	R/W	ADRESA ZAPISU
RESET VALUE 0x00000000		

**Tabulka 6.11:** Adresový registr čtení

ADDRESS REGISTER		BASE_ADDRESS + 0x04
BIT #	PŘÍSTUP	POPIS
31:0	R/W	ADRESA ČTENÍ
RESET VALUE 0x00000000		

**Tabulka 6.12:** Řídicí registr výkonné jednotky

COMMAND REGISTER		BASE_ADDRESS + 0x08
BIT #	PŘÍSTUP	POPIS
31	R/W	„0“ – STOP
		„1“ – POVOLENO
30:29	R/W	„01“ – JEN FILTROVANY OBRAZ
		„10“ – JEN PUVODNI OBRAZ
		„11“ – PROLOZENI OBRAZU
28:0	R/W	REZERVOVANO
RESET VALUE 0x00000000		

**Tabulka 6.13:** Offset pro jednotlivé barevné složky

OFFSET REGISTER		BASE_ADDRESS + 0x0C
BIT #	PŘÍSTUP	POPIS
31:24	R/W	CELKOVÝ OFFSET
23:16	R/W	RED OFFSET
15:8	R/W	GREEN OFFSET
7:0	R/W	BLUE OFFSET
RESET VALUE 0x00000000		

**Tabulka 6.14:** Koefficienty filtrační masky

FILTER REGISTERS		BASE_ADDRESS + 0x20 + OFFSET_ADDRESS
BIT #	PŘÍSTUP	POPIS
31:24	R/W	REZERVOVANO
23:16	R/W	RED
15:8	R/W	GREEN
7:0	R/W	BLUE
RESET VALUE 0x00000000		

**Tabulka 6.15:** Adresový offset jednotlivých koeficientů masky

0x00	0x04	0x08
0x0C	0x10	0x14
0x18	0x1C	0x20

### 6.2.6 Konfigurace a nastavení architektury

Jak již bylo výše zmíněno páteřní sběrnici celé architektury je sběrnice PLB. Ta umožňuje adresovat prostor o velikosti 4 GB. Nejmenší adresový prostor jež může být jedné komponentě přiřazen je 256 B. Adresové prostory jednotlivých komponent připojených na sběrnici PLB:

DDR SDRAM paměť	0x00000000	-	0x03ffffff
Řadič IIC sběrnice	0x0cb00000	-	0x0cb000ff
Řadič kamery	0x0ca00000	-	0x0ca000ff
Výkonná jednotka	0x0cc00000	-	0x0cc000ff

Konfigurace VGA řadiče je přes sběrnici DCR. Adresová sběrnice je široká pouze 10 bitů a proto adresovatelný prostor je jen 1 kB. Adresový prostor tohoto řadiče na DCR sběrnici je:

VGA řadič	0x000	-	0x001
-----------	-------	---	-------

Jelikož, je k architektuře připojen externí CMOS čip, jenž komunikuje s řadičem kamery prostřednictvím vstupních a výstupních portů, je nutné tyto porty nakonfigurovat. Pro všechny signály byl zvolen vstupně-výstupní standart logických úrovní LVTTTL, který má definovanou vstupní úroveň pro logickou „0“ v rozmezí - 0,2 V až 0,8 V a pro logickou „1“ v rozmezí 2,0 V až 3,45 V. Na výstupu jsou zaručeny napěťové úrovně pro logickou „0“ menší než 0,4 V a pro logickou „1“ větší než 2,4 V. U výstupních pinů bylo nastaveno omezení proudu na 8 mA. Přiřazení signálu jednotlivým pinům je uvedeno v tabulce 6.16.



**Tabulka 6.16:** Přiřazení signálů jednotlivým pinům

SIGNÁL	TYP	PIN
FV	IN	V20
CLK_CAM	IN	AC25
DATA_CAM 0	IN	AC24
DATA_CAM 1	IN	W25
DATA_CAM 2	IN	AB24
DATA_CAM 3	IN	Y24
DATA_CAM 4	IN	AB23
DATA_CAM 5	IN	W26
DATA_CAM 6	IN	Y26
DATA_CAM 7	IN	Y25
DATA_CAM_LSB 0	IN	AA26
DATA_CAM_LSB 1	IN	AA23
CLK_TO_CAM	OUT	AC21
SDA	IN/OUT	AB26
SCL	IN/OUT	AC23
RST	OUT	AB25
LV	IN	AD23

## 7. SHRNUTÍ A DOSAŽENÉ VÝSLEDKY

Jako snímací element byl použit CMOS čip MT9V125 s rozlišením 640x480 s frekvencí snímků 25 Hz. Byl zvolen jelikož je poměrně snadno dostupný, má dostatečný dynamický rozsah, má automatické řízení doby expozice a automatickou korekci chvění, lze nastavit zesílení u jednotlivých barevných složek tak i celkové zesílení.

Kvůli propojení kamerové hlavičky a vývojového kitu ML 403 bylo nutno navrhnout a realizovat vlastní propojovací modul. Jelikož frekvence signálu na vodičích je až 25 MHz projevuje se zde již impedanční nepřizpůsobení, které má za následek odrazy na vedení. Proto byla provedena simulace. Na základě této simulace byly do signálových cest vloženy sériové odpory, jenž eliminují tyto odrazy na vedení. Kompletní popis modulu je uveden v kapitole 6.1.2 a schéma včetně seznamu použitých součástek je uvedeno v příloze 1.

Jako výstup kamery bylo zvoleno VGA rozhraní s rozlišením 640x480 60 Hz, jednak protože je stále nejpoužívanějším rozhraním mezi grafickým adaptérem a monitorem ale také pro svůj jednoduchý výstupní formát a je vyvedeno na použitém vývojovém kitu ML 403.

Při podrobném rozboru úlohy se jevílo jako neoptimálnější začít s vytvářením VGA řadiče. Jak bylo popsáno výše pro jeho řízení je použita standardní DCR sběrnice z důvodů nízkých nároků na systémové zdroje, ovšem přístup k takto připojeným registrům je pouze pomocí specifických instrukcí viz kapitola 5.2.8. Zobrazená data jsou čtena přes PLB sběrnici. Jelikož je tato sběrnice dosti komplikovaná byl použit mezi sběrnici a samostatným jádrem jednotky bridge, který za nás řeší přístup na PLB sběrnici. Při použitém VGA formátu je požadován datový tok 55MB/s což tato sběrnice bohatě stačí přenést. Systémové nároky na tento řadič uvádí tabulka 7.1.

Následně byl vytvořen řadič IIC sběrnice z důvodu možnosti inicializovat a konfigurovat dle potřeby CMOS čip. Jedná se o jednotku, která je na sběrnici IIC master a na sběrnici PLB slave. Jednotka byla simulována v prostředí modelsim. Výstupní analýza této jednotky je uvedena v příloze 2. Frekvenci na sběrnici IIC lze

nastavit pomocí generické konstanty při překladu řadiče, zde je frekvence nastavena na 100 kHz. Jedná se o komponentu jež spotřebovala nejmenší počet systémových zdrojů jak lze vidět v tabulce 7.1.

Pro obsluhu externí paměti byla použita standardní komponenta MPMC controller od firmy XILINX. Pro daný účel byla jen vhodně nakonfigurována.

Po-té byl naprogramován a do designu přidán řadič kamery, který zpracovává výstupní data z CMOS čipu a řídí jejich zápis do paměti. Tato data jsou ukládaná do zásobníku fifo. Jedná se o asynchronní fifo proto se jako neoptimálnější jevílo implementovat jej do BRAM paměti. Při šířce dat 16 bitů může být velikost jeho fronty 1024 položek (velikost BRAM je 18kbit), což je plně dostačující jelikož vyprázdnění fifa začíná při naplnění na hodnotu 320 (při jednom burstu se vyčte 16 položek, což vystačí na 20 přenosů a také je to polovina řádku) jak lze vidět na simulacích, které jsou uvedeny v příloze 3. Design zabírá počet systémových prostředků, který je taktéž uveden v tabulce 7.1.

Nejsložitější komponentou jež byla v rámci této práce naprogramována je výkonná jednotka. Na její realizaci byly spotřebovány čtyři BRAM paměti. Obraz lze projíždět filtrační maskou o rozměru 3x3, proto jsou nataženy nejdříve tři řádky do samostatných BRAM pamětí z důvodů rychlého přístupu k datům. Následně je obraz projížděn zadanou filtrační maskou. Hodnoty ve filtrační masce lze měnit zapsáním hodnot do příslušných konfiguračních registrů viz. kapitola 6.2.5. Kdyby byla filtrační maska zadána pevně bez možnosti změny, došlo by při syntéze k vyoptimalizování a nebylo by potřeba násobiček.

**Tabulka 7.1:** Spotřeba systémových zdrojů pro jednotlivé komponenty

		VGA ŘADIČ	IIC ŘADIČ	ŘADIČ KAMERY	VÝKONNÁ JEDNOTKA	ŘADIČ PAMĚTI
Slice	Počet	537	340	536	981	2485
	%	9	6	9	17	45
Flip Flop	Počet	753	441	715	806	3190
	%	6	4	6	7	29
4 input LUT	Počet	703	496	703	1264	3800
	%	6	4	6	11	34
RAMB16	Počet	1	0	1	4	13
	%	2	0	2	11	36
BUFR	Počet	0	0	1	0	0
	%	0	0	6	0	0
DSP48	Počet	0	0	0	9	0
	%	0	0	0	28	0
IO	Počet	630	287	393	548	7520

Použité hradlové pole XC4VFX12 je designem jenž obsahoval nezbytné komponenty což je procesor Power PC, programovací a ladicí rozhraní JTAGPPC, blok generování potřebných hodin, řadič plb sběrnice, řadič DDR SDRAM paměti, VGA řadič, výkonnou jednotku, řadič kamery, řadič IIC sběrnice, téměř zaplněno a přidání dalších komponent již není téměř možné. V případě potřeby implementovat další jednotky je nutné již použít větší hradlové pole, kde tento design zabírá jen část pole a proto je možné zde přidat další jednotky. Procentuální zaplnění jednotlivých typů polí uvádí tabulka 7.2.

**Tabulka 7.2:** Porovnání zaplnění kompletním designem pro různá pole

		XC4VFX12	XC4VFX20	XC4VFX60	XC4VFX100
Slice	Počet	5400	6321	6321	6318
	%	98	73	25	15
	Počet v poli	5472	8544	25280	42176
Flip Flop	Počet	7047	7054	7054	7054
	%	64	41	13	8
	Počet v poli	10944	17088	50560	84352
4 input LUT	Počet	9030	9119	9117	9117
	%	82	53	18	10
	Počet v poli	10944	17088	50560	84352
RAMB16	Počet	27	23	23	23
	%	75	33	9	6
	Počet v poli	36	68	232	376
GCLK	Počet	4	4	4	4
	%	12	12	12	12
	Počet v poli	32	32	32	32
DCM	Počet	1	1	1	1
	%	25	25	8	8
	Počet v poli	4	4	12	12
BUFR	Počet	1	1	1	1
	%	6	6	3	2
	Počet v poli	16	16	32	40
PPC405	Počet	1	1	1	1
	%	100	100	50	50
	Počet v poli	1	1	2	2
JTAGPPC	Počet	1	1	1	1
	%	100	100	100	100
	Počet v poli	1	1	1	1
DSP48	Počet	9	9	9	9
	%	28	28	7	5
	Počet v poli	32	32	128	160

Jelikož je design implementován do pole typu FPGA, je po připojení napájení nutné vždy nahrát znovu konfiguraci. Konfiguraci lze provést jednak přes jtag jak tomu bylo zde, nebo je potřeba design nahrát do paměti odkud je vždy po připojení napájení natažen do paměti.

Skutečná realizace kamery je zobrazena na snímcích v příloze 4.

## 8. ZÁVĚR

V rámci této práce byla navržena a odzkoušena jednoduchá architektura inteligentní kamery vytvořená v hradlovém poli VIRTEX 4 (XC4VFX12), navržen a realizován propojovací modul. Jednotlivé komponenty byly napsány a simulovány v jazyku VHDL a Verilog. Rozšířením práce je snaha implementovat pomocnou výpočetní jednotku APU, která by zvýšila možnosti zpracování obrazových dat a umožnila implementovat nové funkce. Bude-li jednotka implementována je nutné již větší hradlové pole XC4VFX20. Dále je snaha o naprogramování komponent v jazyce impulse C.

## 9. POUŽITÁ LITERATURA

- [1] DALSA.: *Image Sensor Architectures for Digital Cinematography*, [on-line]  
[http://www.dalsa.com/dc/documents/  
Image\\_Sensor\\_Architecture\\_Whitepaper\\_Digital\\_Cinema\\_00218-00\\_03-70.pdf](http://www.dalsa.com/dc/documents/Image_Sensor_Architecture_Whitepaper_Digital_Cinema_00218-00_03-70.pdf)
- [2] ZMEŠKAL, O.: *Principy digitálního záznamu obrazu*, [on-line]  
[http://www.fch.vutbr.cz/lectures/imagesci/download/stud02\\_hrakra01.pdf](http://www.fch.vutbr.cz/lectures/imagesci/download/stud02_hrakra01.pdf)
- [3] VALACH, S.: *Inteligentní průmyslové kamery – přehled trhu*, [on-line]  
<http://www.automatizace.cz/article.php?a=389>
- [4] HAVLE, O.: *Smart kamery pro strojové vidění*, [on-line]  
<http://www.automatizace.cz/article.php?a=1588>
- [5] *Zpracování obrazových dat*, [on-line]  
[http://gama.fsv.cvut.cz/wiki/index.php/Zpracov%C3%A1n%C3%AD\\_obrazov%C3%BDch\\_dat\\_-\\_cvi%C4%8Den%C3%AD\\_%C4%8D.5](http://gama.fsv.cvut.cz/wiki/index.php/Zpracov%C3%A1n%C3%AD_obrazov%C3%BDch_dat_-_cvi%C4%8Den%C3%AD_%C4%8D.5)
- [6] HLAVÁČ, V.: *Hledání hran*, [on-line]  
<http://cmp.felk.cvut.cz/~hlavac/TeachPresCz/11DigZprObr/22DetekceHran.pdf>
- [7] HLAVÁČ, V.: *Techniky pořízení obrazu*, [on-line]  
<http://cmp.felk.cvut.cz/~hlavac/TeachPresCz/11DigZprObr/06TechnikyPorizKamery.pdf>
- [8] *Bayer filter*, [on-line]  
[http://en.wikipedia.org/wiki/Bayer\\_filter](http://en.wikipedia.org/wiki/Bayer_filter)
- [9] ENGDAHL, T.: *Calculator for video timings*, [on-line]  
<http://www.tkk.fi/Misc/Electronics/faq/vga2rgb/calc.html>
- [10] ČAPKA, L.: *VGA rozhraní*, [on-line]  
<http://merlin.fit.vutbr.cz/FITkit/docs/firmware/20060931vga.html>
- [11] HRADECKÝ, D.: *VGA řadič na FPGA*, [on-line]  
[http://amber.feld.cvut.cz/fpga/teaching/fpga/VGA\\_FPGA.pdf](http://amber.feld.cvut.cz/fpga/teaching/fpga/VGA_FPGA.pdf)
- [12] XILINX. *ML403 User Guide*, [on-line]  
[http://www.xilinx.com/support/documentation/boards\\_and\\_kits/ug080.pdf](http://www.xilinx.com/support/documentation/boards_and_kits/ug080.pdf)

- [13] XILINX. *ML403 Schematics*, [on-line]  
[http://www.xilinx.com/support/documentation/boards\\_and\\_kits/ml401\\_2\\_3\\_schematics.pdf](http://www.xilinx.com/support/documentation/boards_and_kits/ml401_2_3_schematics.pdf)
- [14] XILINX. *ML40x EDK Procesor Reference Design*, [on-line]  
[http://www.xilinx.com/support/documentation/boards\\_and\\_kits/ug082.pdf](http://www.xilinx.com/support/documentation/boards_and_kits/ug082.pdf)
- [15] XILINX. *ML403 Evaluation Platform*, [on-line]  
<http://www.xilinx.com/products/boards/ml401/docs/ML403.pdf>
- [16] XILINX. *ML40x Getting Started Tutorial*, [on-line]  
[http://www.xilinx.com/support/documentation/boards\\_and\\_kits/ug083.pdf](http://www.xilinx.com/support/documentation/boards_and_kits/ug083.pdf)
- [17] XILINX. *Virtex-4 User Guide*, [on-line]  
[http://www.xilinx.com/support/documentation/user\\_guides/ug070.pdf](http://www.xilinx.com/support/documentation/user_guides/ug070.pdf)
- [18] XILINX. *XtremeDSP for Virtex-4*, [on-line]  
[http://www.xilinx.com/support/documentation/user\\_guides/ug073.pdf](http://www.xilinx.com/support/documentation/user_guides/ug073.pdf)
- [19] XILINX. *Virtex-4 Packaging and Pinout Specification*, [on-line]  
[http://www.xilinx.com/support/documentation/user\\_guides/ug075.pdf](http://www.xilinx.com/support/documentation/user_guides/ug075.pdf)
- [20] XILINX. *Virtex-4 Overview*, [on-line]  
[http://www.xilinx.com/support/documentation/data\\_sheets/ds112.pdf](http://www.xilinx.com/support/documentation/data_sheets/ds112.pdf)
- [21] XILINX. *Virtex-4 Configuration User Guide*, [on-line]  
[http://www.xilinx.com/support/documentation/user\\_guides/ug071.pdf](http://www.xilinx.com/support/documentation/user_guides/ug071.pdf)
- [22] XILINX. *Xilinx Memory Interface Generator*, [on-line]  
[http://www.xilinx.com/support/documentation/user\\_guides/ug086.pdf](http://www.xilinx.com/support/documentation/user_guides/ug086.pdf)
- [23] XILINX. *PowerPC 405 Processor Block Reference Guide*, [on-line]  
[http://www.xilinx.com/support/documentation/user\\_guides/ug018.pdf](http://www.xilinx.com/support/documentation/user_guides/ug018.pdf)
- [24] XILINX. *PowerPC Processor Reference Guide*, [on-line]  
[http://www.xilinx.com/support/documentation/user\\_guides/ug011.pdf](http://www.xilinx.com/support/documentation/user_guides/ug011.pdf)
- [25] IBM. *PPC405 Embedded Processor Core*, [on-line]  
[http://www-01.ibm.com/chips/techlib/techlib.nsf/techdocs/A07CE56994E69BFE0025731C005C9BFE/\\$file/ppc405S\\_um\\_01\\_pub.pdf](http://www-01.ibm.com/chips/techlib/techlib.nsf/techdocs/A07CE56994E69BFE0025731C005C9BFE/$file/ppc405S_um_01_pub.pdf)
- [26] IBM. *Power Architecture PowerPC 405*, [on-line]  
<http://www-03.ibm.com/technology/ges/semiconductor/power/licensing/cores/ppc405.html>

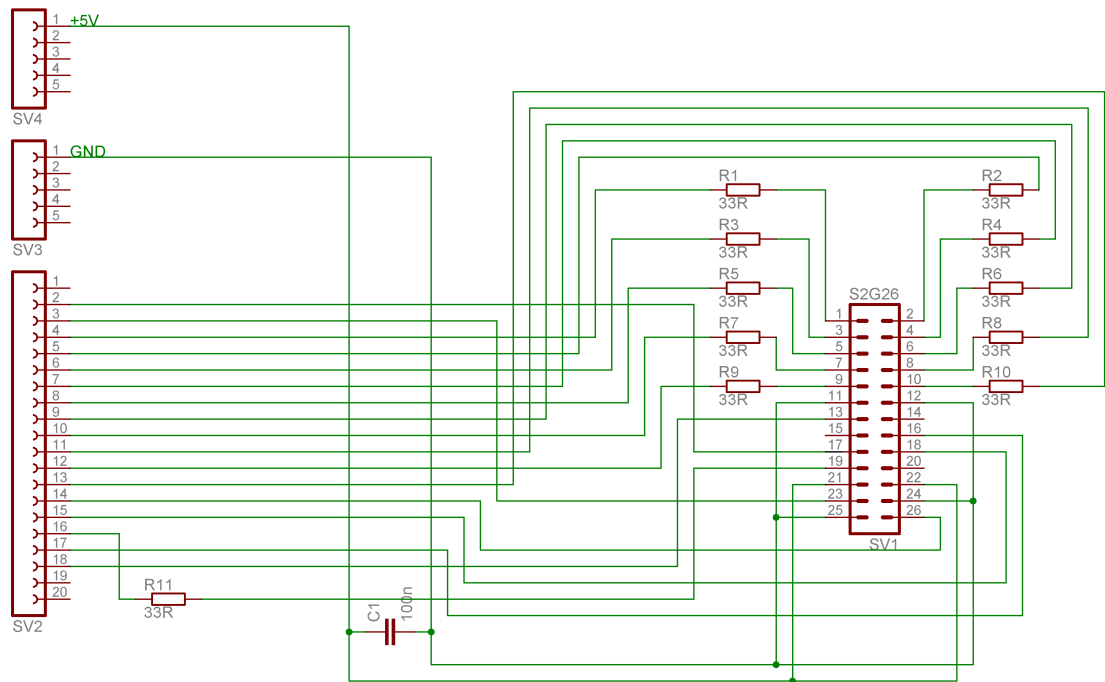


- [27] IBM. *128-Bit Procesor Local Bus*, [on-line]  
[http://www-01.ibm.com/chips/techlib/techlib.nsf/techdocs/3BBB27E5BCC165BA87256A2B0064FFB4/\\$file/PlbBus\\_as\\_01\\_pub.pdf](http://www-01.ibm.com/chips/techlib/techlib.nsf/techdocs/3BBB27E5BCC165BA87256A2B0064FFB4/$file/PlbBus_as_01_pub.pdf)
- [28] MICRON. *MT9V125: SOC VGA Digital Image Sensor*,
- [29] MICRON. *MT9V125: SOC VGA Digital Image Sensor Errata*,
- [30] MICRON. *354SOC Headboard*,
- [31] RAŠEK, P.: *Odrazy na vedení*, [on-line]  
<http://rfprop.com/tep/odrazy%20na%20vedeni/odrazy%20na%20vedeni.htm>
- [32] OLEJÁR, M.: *Stručný popis sběrnice I2C*, [on-line]  
[http://hw.cz/design/i2c\\_pic/index.html](http://hw.cz/design/i2c_pic/index.html)
- [33] HERVEILLE, R.: *I2C - Master Core Specification*, [on.line]  
[http://www.opencores.org/tmp/cvsget\\_cache/i2c/doc/i2c\\_specs.pdf](http://www.opencores.org/tmp/cvsget_cache/i2c/doc/i2c_specs.pdf)
- [34] IMPULSE. *CoDeveloper User Guide version 3.0*,
- [35] XILINX. *Multi-PPort Memory Controller*, [on-line]  
[http://www.xilinx.com/support/documentation/ip\\_documentation/mpmc.pdf](http://www.xilinx.com/support/documentation/ip_documentation/mpmc.pdf)
- [36] XILINX. *PLBV46 Master Burst*,

## **SEZNAM PŘÍLOH**

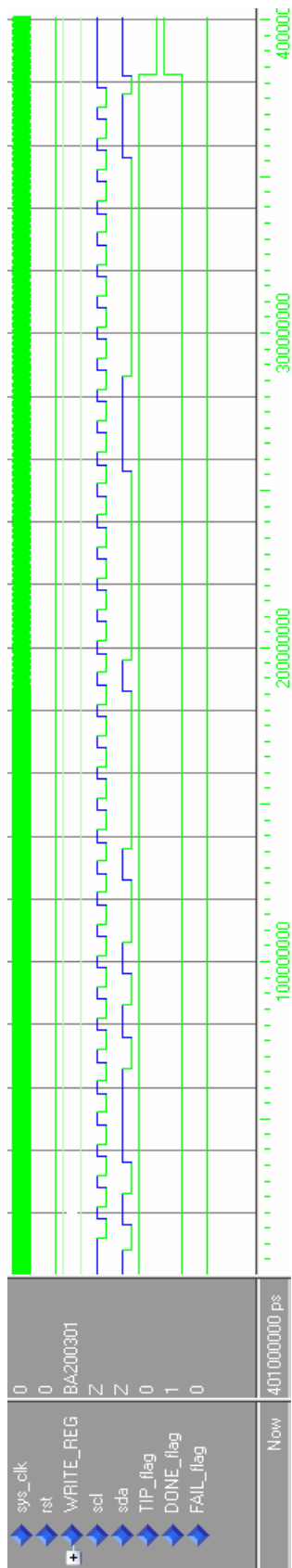
- Příloha 1 Schéma propojovacího modulu
- Příloha 2 Simulace IIC řadiče
- Příloha 3 Simulace řadiče kamery
- Příloha 4 Snímky realizace kamery

## Příloha 1: Schéma propojovacího modulu

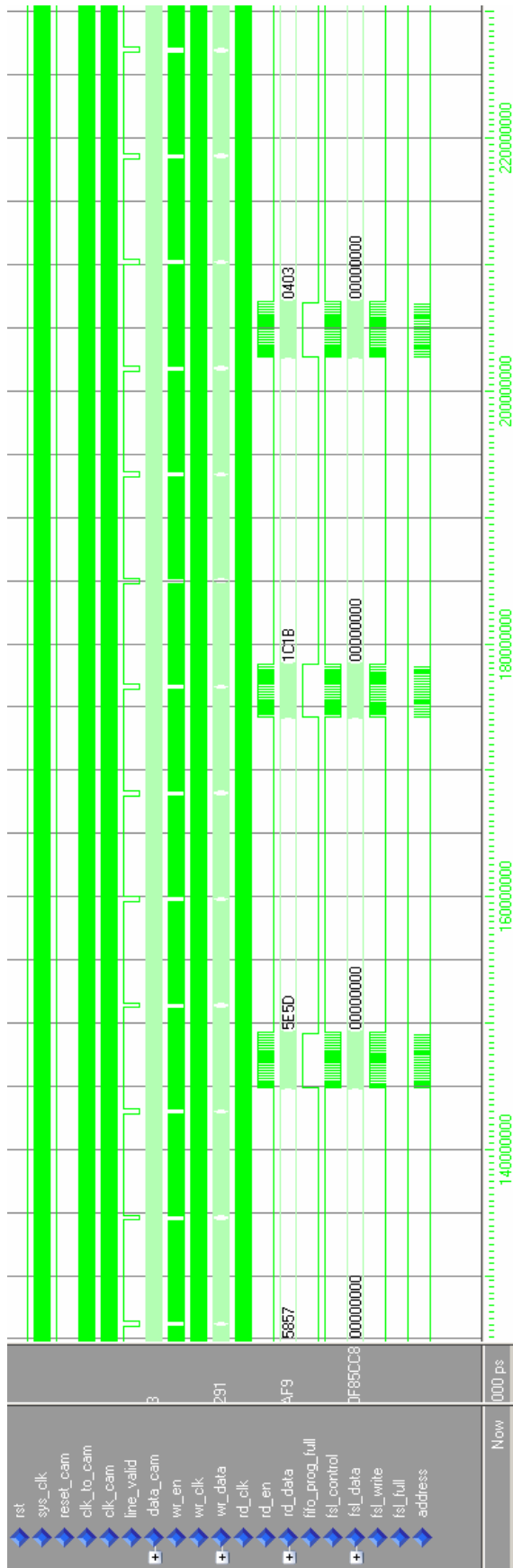


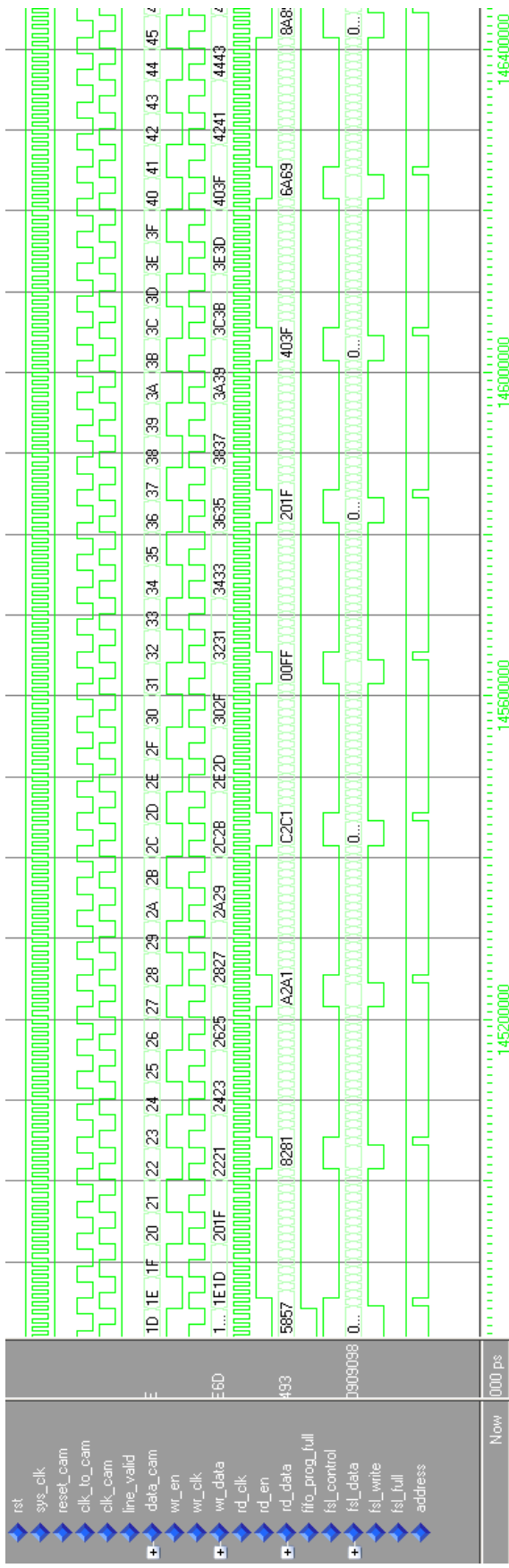
OZNAČENÍ	TYP
R 1	33R
R 2	33R
R 3	33R
R 4	33R
R 5	33R
R 6	33R
R 7	33R
R 8	33R
R 9	33R
R 10	33R
R 11	33R
C 1	100n
SV 1	S2G26
SV 2	BL20G
SV 3	BL05G
SV 4	BL05G

## Příloha 2: Simulace IIC řadiče



### Příloha 3: Simulace řadiče kamery





## Příloha 4: Snímky realizace kamery

