

České vysoké učení technické v Praze

Fakulta elektrotechnická



DIPLOMOVÁ PRÁCE

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA ELEKTROTECHNICKÁ

Katedra mikroelektroniky

Optický senzor pro měření geometrických veličin

Diplomová práce

Vypracoval: Bc. Michal Groh

Vedoucí práce: Prof. Ing. Miroslav Husák, Csc.

Praha 2014

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. G R O H Michal**

Studijní program: Komunikace, multimédia a elektronika
Obor: Elektronika

Název tématu: **Zařízení pro měření geometrických veličin z obrazové informace**

Pokyny pro vypracování:

1. Proveďte rešerši současného stavu poznání v oblasti bezdotykových snímačů geometrických veličin s využitím řádkových obrazových sensorů.
2. Navrhněte a realizujte vzorek snímače s řádkovým senzorem CMOS určený pro bezkontaktní měření vybrané geometrické veličiny (vzdálenost, úhel, plocha apod.). Pro řešení použijte vývojovou sadu vhodného FPGA čipu. Ten doplňte o vhodný AD převodník, lineární obrazový sensor (dostupný je např. SONY ILX551A) včetně jednoduché optiky a komunikační rozhraní k PC. Navrhněte a ověřte programové vybavení pro FPGA a PC s jehož pomocí budete demonstrovat funkčnost všech částí měřicího řetězce.
3. Zjistěte a vyhodnoťte parametry realizovaného zařízení, porovnejte s obdobnými systémy, soustřeďte se především na rychlost a přesnost měření. Pokuste se alespoň kvalitativně zhodnotit vliv difrakce světla na hraně měřeného předmětu na přesnost měření.
4. Proveďte ekonomickou rozvahu pro navržený systém pro jeho případnou výrobu.

Seznam odborné literatury:

- [1] Ďaďo, S. - Fischer, J.: Optical Sensors. In Master Book on Sensors. Prague: CTU Prague a BEN, 2003, vol. 1, s. 2-1-2-132. ISBN 80-7300-129-2.
- [2] Fischer, J.: Optoelektronické senzory a videometrie. 1. vyd. Praha: ČVUT, 2002. 143 s. ISBN 80-01-02525-X.

Vedoucí: **Prof.Ing. Miroslav Husák, CSc.**

Platnost zadání: 31. 8. 2014

L.S.

Prof. Ing. Miroslav Husák, CSc.
vedoucí katedry

Prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 28. 1. 2013

Prohlášení:

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne.....

podpis.....

Poděkování:

Tímto bych chtěl poděkovat všem, kteří mi pomáhali při tvorbě této práce. Jmenovitě především svému vedoucímu Prof. Ing. Miroslavu Husákovi, Csc. za trpělivost, rady a všeobecnou pomoc s mou prací. Dále Ing. Petru Pavlatovi a Ing. Adamu Kučerovi, kteří mi poskytli neocenitelné rady v této práci. Také bych chtěl poděkovat rodině, přátelům za podporu při psaní této práce, ale i v průběhu studia.

Anotace

Diplomová práce je zaměřena na návrh optického řádkové snímače, který by měl sloužit k měření geometrických veličin. Návrh by měl obsahovat architekturu FPGA obvodu, který nahradí standardně dodávané procesory. Systém by měl komunikovat po ethernetovém rozhraní. Práce se zabývá návrhem DPS, vnitřní architektury FPGA a programovou obsluhou. Na konci bude ekonomické hodnocení pro případnou sériovou výrobu.

Klíčová slova jsou senzor, FPGA, paměť, A/D převodník, kamera, CCD.

Annotation

Diploma thesis focuses on the design of the optical line sensor, which should serve to geometrical measurements. The proposal should include architecture of FPGA circuit which replaces the standard supplied processors. The system should communicate via the Ethernet interface. The work deals with the design of PCB, FPGA internal architecture and program operation. The economic evaluation of the project for eventual mass production is given at the end of the thesis.

Keywords: sensor, FPGA , memory, A / D converter, camera, CCD .

Obsah

1	Úvod	1
2	Konkurence na trhu	2
3	Hardwarový návrh	4
3.1	Optický snímač	4
3.2	A/D převodník	6
3.3	Konektor VHDCI	8
3.4	FPGA	10
	Rodina Spartan 6	10
	Digilent Atlys	12
3.5	Objektiv	12
3.6	DPS a její návrh	13
4	Systemový návrh	15
4.1	Návrh systému a výběr základních komponent	16
4.2	RTL návrh	17
	4.2.1 CPU OpenRISC	17
	4.2.2 Wishbone	20
	4.2.3. Paměť FIFO	23
4.3	Implementace paměti DDR SDRAM	27
5	Verifikace	28
5.1	CCD Driver	29
6	Implementace	31
7	Software	33
7.1	Popis procesoru openRISC 1200 z pohledu programátora	33
7.2	sestavení programu pro obsluhu procesoru openRISC	34
7.3	Vývojové prostředí	35

7.4 Programy	36
8 Ekonomické zhodnocení	38
9 Závěr	40
10 Literatura	42
11 Přílohy	43
11.1 Osazovací a výrobní data	43
11.2 Kompletní obvodové schéma	47
11.3 Obsah přiloženého DVD	50

Seznam obrázků

OBR. 1	TYPICKÁ APLIKACE ŘÁDKOVÉ KAMERY [2].....	2
OBR. 2	PŘÍKLADY JEDNOTLIVÝCH KAMER [2]	3
OBR. 3	BLOKOVÉ SCHÉMA SENZORU ILX551A [4]	5
OBR. 4	ČASOVÝ DIAGRAM SENZORU ILX551A [4]	6
OBR. 5	BLOKOVÉ SCHÉMA PŘEVODNÍKU AD9235 [5]	7
OBR. 6	ČASOVÝ DIAGRAM PŘEVODNÍKU AD9235 [5]	8
OBR. 7	BLOKOVÉ SCHÉMA KONEKTORU VHDCI S POPISEM NÁZVU PROPOJENÍ V DESIGNU FPGA[6]	9
OBR. 8	ZÁKLADNÍ STRUKTURA BUŇKY OBVODU SPARTAN 6 PŘEVZATO [7]	11
OBR. 9	VYROBENÁ DPS SE OSAZENÝMI SOUČÁSTKAMI (TOP LAYER)	14
OBR. 10	VYROBENÁ DPS S OSAZENÝMI SOUČÁSTKAMI (BOTTON LAYER)	15
OBR. 11	BLOKOVÉ SCHÉMA SYSTÉMU DOPLNĚNÉ O BLOKY PRO CCD.....	17
OBR. 12	JÁDRO OR1200 PŘEVZATO [9]	18
OBR. 13	CPU/DSP JÁDRO, PŘEVZATO Z [10].....	19
OBR. 14	JEDNODUCHÝ CYKLUS NA SBĚRNICI WISHBONE PŘEVZATO [11].....	21
OBR. 15	ZÁKLADNÍ PROPOJENÍ SLAVE A MASTER BLOKŮ PŘEVZATO [11].....	22
OBR. 16	PROPOJENÍ SE SDÍLENOU SBĚRNICÍ Z [11].....	23
OBR. 17	PROPOJENÍ S VYUŽITÍM CROSSBAR PŘEVZATO Z [11].....	23
OBR. 18	SCHÉMA PAMĚTI FIFO Z VNITŘNÍHO DESIGNU PRO LIBOVOLNOU SBĚRNICI[12].....	24
OBR. 19	FIFO S NEZÁVISLÝM ČASOVÁNÍM: PROPOJOVACÍ SIGNÁLY [12]	25
OBR. 20	OPERACE ZÁPISU DO PAMĚTI PRO SYNCHRONNÍ HODINY [12]	27
OBR. 21	OPERACE ZÁPISU DO PAMĚTI PRO SYNCHRONNÍ HODINY [12]	27
OBR. 22	SCHÉMA DDR2 S NÁZVEM PROPOJOVACÍCH DRÁTŮ [6]	28
OBR. 23	ČTENÍ DAT ZE SENZORU.....	30
OBR. 24	CONTROL REGISTR.....	30
OBR. 25	ZÁPIS DAT DO FIFO A ČTENÍ Z FIFO	31
OBR. 26	STATUS REGISTR	31
OBR. 27	VYUŽITÍ ZDROJŮ PO SYNTÉZE [9]	32
OBR. 28	VNITŘNÍ PROPOJENÍ FPGA	33
OBR. 29	PROSTŘEDÍ ECLIPSE []	36
OBR. 30	VÝVOJOVÝ DIAGRAM PRO ODESÍLÁNÍ DAT	37
OBR. 31	NASTAVENÍ PORTU A VELIKOSTI OBRÁZKU V PROGRAMU UDPBITMAP	38
OBR. 32	VYKRESLENÍ BÍLÉHO KUSU PLASTU NA HRANÉ HNĚDÉHO STOLU	38

Seznam tabulek

TAB. 1	POPIS FUNKCÍ PINŮ OBVODU AD9235.....	8
TAB. 2	ZAPOJENÍ KONEKTORU VHDCI NA VÝVOJOVÉ DESCE ATLYS.....	9
TAB. 3	TABULKA HODNOT $\cos^4\Omega$ PŘEVZATO [1]	13
TAB. 4	POPIS SIGNÁLŮ SBĚRNICE WISHBONE, PŘEVZATO Z [6]	20
TAB. 5	POPIS FUNKCÍ JEDNOTLIVÝCH VSTUPŮ A VÝSTUPŮ PAMĚTI FIFO	26
TAB. 6	VYJÍMKY PROCESORU OPENRISC, PŘEVZATO Z [6]	34
TAB. 7	ROZPOČET PŘÍPADNÉ VÝROBY.....	40

Seznam symbolů

U [V] – elektrické napětí

f [Hz] – frekvence

t [s] – čas

R [Ω] – elektrický odpor

l [m] – délka

Seznam zkratek a symbolů

ALU	Arithmetic logic unit
ARM	Advanced RISC Machine
AVR	Alf Vegard Risc
CCD	charge-coupled device
CLK	clock
CMOS	Complementary Metal Oxide Semiconductor
DCM	Digital clock manager
DDR	Double data rate
DPS	deska plošných spojů
DSP	Digital signal processing
EPROM	Erasable programmable read-Only Memory
EEPROM	Electrically Erasable programmable read-Only Memory
FIFO	First In, First Out
FPGA	Field programmable gate array
FPU	Floating point unit
GCC	GNU Compiler Collection
GPL	General Public License
HDL	Hardware description language
MAC	Media access controller
MOS	Metal-Oxide Semiconductor
OPN	Oblast prostorového náboje
PAL	Programmable Array Logic
PIC	Programmable Interrupt Controller
PLD	Programmable logic device
PLL	Phase locked loop
PROM	Programmable Read Only memory
RAM	Random access memory
RISC	Reduced instruction set computing
RTL	Registr transfer level
RTP	Real time protocol
SHA	sample and Hold amplifier
SoC	Serial Peripheral Interface
SRAM	Static Random Access Memory
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
USB	Universal Serial Bus

VHDL	VHSIC hardware description language
VHDCI	Very High Density Cable Interconnect
WB	Wishbone

1 Úvod

V dnešní době neustále narůstá stupeň automatizace výroby a tím i nároky na kvalitu. Řada kontrol je prováděna člověkem, což může mít za následek nižší kvalitu výsledného výrobku. Část výrobních procesů je však dnes natolik rychlá, že není v lidských silách provádět vizuální kontrolu výrobků. Proto nastupují na scénu obrazové snímače a jejich kontrola vad výrobku.

Jelikož se řada kontrol provádí vizuálně, je potřeba nasadit na takovou kontrolu nějaký kamerový systém. To vede k problematice rozpoznávání a složitého zpracování obrazu. Často se nejedná jenom o rozpoznávání, ale přímo o měření geometrických veličin. V těchto složitých systémech jde také o věrohodnost výsledku z metrologického hlediska.

Použití videometrického systému v sobě skýtá mnohé výhody, jako bezproblémová implementace do stávajících linek, z důvodu bezkontaktního měření. Avšak problémy mohou být jak v malém rozlišení obrazu, nevhodném nastavení kamer a v neposlední řadě také vhodného osvětlení objektu nebo okolí. V tomto oboru často platí, že vhodné osvětlení a úhel kamery je půl úspěchu.

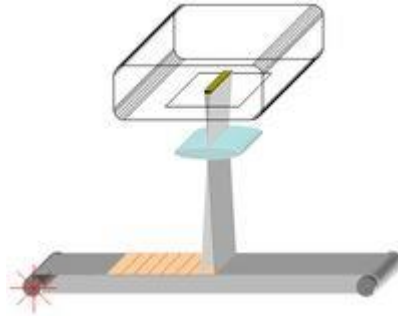
Jako základní technologie se používají v kamerách, buď CMOS, nebo CCD obrazové snímače. Velikost plochy snímače je určena podle typu použití. Snímače většinou rozlišujeme na jednořádkové a víceřádkové. Víceřádkové snímače se však dělí, také podle snímané plochy a množství použitých snímacích bodů na ploše. Snímače v dnešní době mají často vlastní předzpracování signálu, implementovanou přímo v čipu snímače, což umožňuje dnešní pokroková polovodičová technologie.

Zpracování obrazu ze snímače se často provádí mimo vlastní tělo snímače, kdy není nutné umístit výkonný čip přímo k senzoru, ale je možné využít počítač, přímo k tomu určený, který bude pouze vyčítat data přes ethernetové rozhraní, případně usb rozhraní. V dnešní době se k tomu používají často jednoduché mikroprocesory řad AVR případně výkonnější ARM.

Myšlenka umístění FPGA čipu, přímo do těla senzoru by mohla vyřešit problémy předzpracování obrazové informace a případné problémy se zahlcením kanálu dat. Takový procesor by mohl být navíc modifikován ve více verzích a umožnil by například použití dvou čipů kolmo na sebe, nebo jiné modifikace. Dnešní moderní hradlová pole umožňují programování složitých aplikací. Dnešní pokročilé stupně integrace na čip umožňují zachovat malé rozměry čipu. FPGA čipy by také umožnily použití bezdrátového přenosu informace a tím by odpadla potřeba hledání cesty kabelů ke kameře a její složitě nastavení do správně polohy.

2 Konkurence na trhu

Řádková kamera pracuje na principu skládání jednotlivých řádků snímaného obrazu. Z toho vyplývá, že je nutné, aby buď kamera, nebo snímaný objekt byly v pohybu. Takto pracuje v podstatě kancelářský scanner. Typickými aplikacemi jsou například kontrola malých součástí na páse, kontrola kvality potravin.



Obr. 1 Typická aplikace řádkové kamery [2]

V současné době je konkurence na trhu videometrických systémů poměrně velká, avšak i zde se najdou mezery. Velká nabídka průmyslových kamer je dostupná například na internetové stránce prumyslove-kamery.cz. Převládají zde však výrobky společnosti Basler, ke kterým vztáhnou svoji práci. Z nabídky je vidět množství různých typů kamer a není to jenom o jednom konkrétním typu.

Můj hlavní zájem z hlediska konkurence se jednal o jednořádkové optické kamery. Konstrukce mé kamery je založena na jednořádkovém optickém senzoru s 2048 body v jednom řádku. Konkurence nabízí na jednořádkové snímače od 1k zobrazovacích bodů až po 12k zobrazovacích bodů v černobílém provedení. Hlavními výrobce, které jsem objevil, jsou např. firmy Basler, Teledyne Dalsa.

Výhodou jednořádkových snímačů je oproti maticové CCD kameře její rychlost. Pokud bychom chtěli snímat rychle se pohybující předměty maticovým snímačem, jsou potřeba krátké expoziční časy a velmi intenzivní zábleskové světlo, aby bylo dosaženo patřičné obrazové kvality. S řádkovým snímačem lze snímat s velmi krátkým expozičním časem a s rychlostí řádově až několik tisíc řádků za sekundu. Většinou nám stačí pro nasvícení intenzivní liniové světlo, které soustředí veškerý světelný tok do úzké linky. Z hlediska optiky je řádková kamera výhodnější, protože vady a nepřesnosti objektivu se nám projeví pouze v jednom směru.

Vzhledem k rychlosti zobrazování se konkurence pohybuje u velmi rychlých jednořádkových kamer přibližně na hodnotě 140 kHz. Většina typů kamer má však hodnoty nižší a to kolem 60 kHz.

Většina kamer také komunikuje pomocí gigabitového ethernetu., nebo rozhraní Camera Link. V mém případě bude použit pouze gigabitový ethernet.

U mnou zvoleného monochromatického řádkového snímače je rychlost senzoru 1 MHz. Bohužel jak jsem zjistil u kanadského výrobce Teledyne Dalsa, jsou jejich senzory mnohonásobně rychlejší a jejich frekvence se u snímačů pohybuje v řádech desítek MHz, podle typu kamery. Rychlost 1 MHz je pouze u jejich nejnižší řady.

Kamery jsou často dodávány s možností volby A/D převodníku. U drtivé většiny je zvolená bitová hloubka 8 až 12 bitů na pixel.

Většina řádkových kamer dneška je založena na principu technologie CMOS. Tato technologie má oproti CCD výhodu v náhodném i sekvenčním čtení. Může obsahovat generátory impulsů, zesilovací obvody a rychlé převodníky. Do takových obvodů je možné také integrovat i obvody pro číslicové zpracování obrazu. Jejich velkou výhodou je také nízký proudový odběr a malé napájecí napětí. Nevýhodou oproti CCD je větší šum.

Na trhu jsou kromě monochromatických řádkových kamer také barevné řádkové kamery, u kterých je jednodušší možnost kontroly a rozpoznání vady vzhledem k obsluze. Barevné podání je pro lidské oko příjemnější než černobílé. Díky barevným sensorům, je také možnost přidání dalších aplikací, pro které můžeme kamery použít.

V dnešní době se spíše než samotné kamery dodává celý měřicí systém. Kamera tvoří pouze jednu z částí. Systém začíná volbou kamery a objektivu. Pokračuje osvětlením a volbou rozhraní pro komunikaci. A končí dodaným softwarem s možností programovacích sad pro jednotlivé aplikace. Díky tomu je možné zákazníkovi dodat přesný typ kamery pro jeho potřebu.



Obr. 2 Příklady jednotlivých kamer [2]

Z celkového pohledu si myslím, že trh je v dnešní době relativně naplněn. Většina firem nabízí komplexní řešení, čehož já nejsem v práci schopný dosáhnout. Tato řešení jsou často velmi drahá, avšak jejich výhodou je velká životnost a kvalita dosažené informace. Výhodou této práce by mohlo být použití

FPGA čipu. Při správném použití velikosti čipu by se daly ušetřit náklady spojené s výměnou senzorů a vše by mohlo mít blokovou konstrukci.

3 Hardwarový návrh

Návrh bude spočívat v jednoduché DPS, na které bude umístěn jednořádkový senzor ILX551A od společnosti Sony. Výstup senzoru bude přiveden na operační zesilovač, který zesílí výstupní signál na úroveň pro zpracování A/D převodníkem. Za operačním zesilovačem bude následovat low pass filter na 3 MHz. Z filtru přivedeme signál přímo do A/D převodníku. A/D převodník AD9235 bude mít jednotlivé výstupy přivedeny rovnou na konektor VHDCI, který je připojen na vývojový FPGA kit firmy Atlys s procesorem Spartan 6.

3.1 Optický snímač

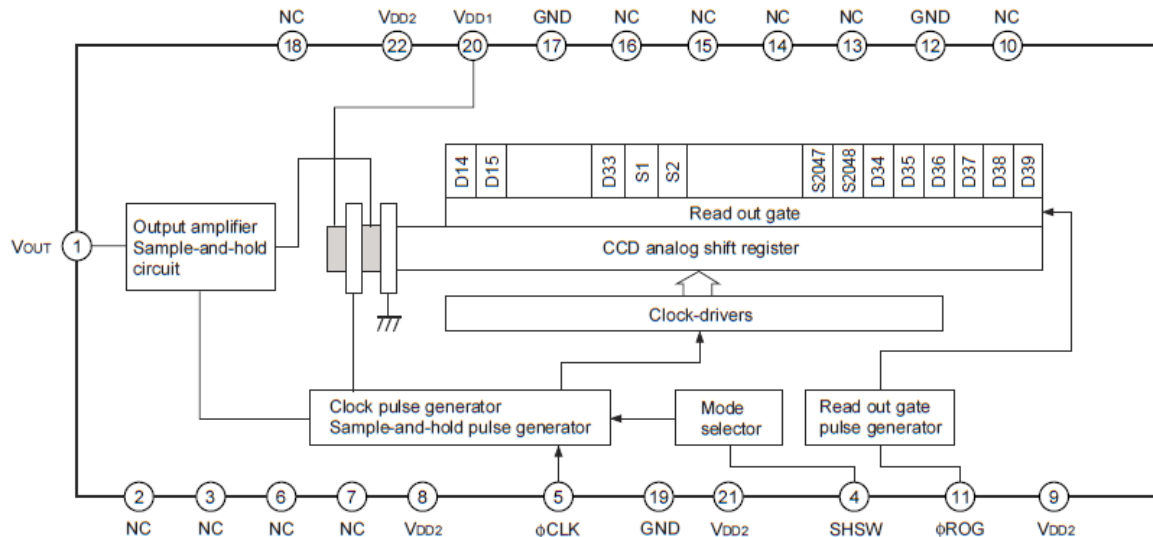
Pokud správně uspořádáme jednotlivé elementy a to fotoelektrický převodník, posuvný registr a výstupní nábojový detektor vznikne CCD snímač. Existují dva základní typy a to jednorozměrný - řádkový a dvourozměrný – plošný. V práci je použit řádkový snímač, pro více informací je potřeba nahlédnout do [1] nebo [3].

Jak bylo výše zmíněno, podíváme se na řádkové snímače. Snímač obsahuje vlastní fotoelementy, oddělovací přenosové hradlo a CCD posuvný registr pro transport nábojů. Jako detektory záření se nejčastěji používají MOS kapacitory nebo fotodiody s PN přechodem. Fotoelementy na čipu jsou od sebe odděleny tzv. stop oblastmi, které se vytvářejí difuzí nebo iontovou implantací oddělovacích oblastí P⁺. V stop oblasti je vysoká koncentrace akceptorů. Díky této koncentraci se nevytvoří pod snímacími MOS tranzistory ochuzená oblast.

Po akumulaci nábojů ve fotoelementech se přivedením kladného řídicího napětí na oddělovací přenosové hradlo pod ním vytvoří OPN a v něm kanál, jímž se náboje přenesou do příslušné potenciálové jámy posuvného CCD registru. Po přesunu náboje do registru se napětí na oddělovacím přenosovém hradle opět sníží na nulu. V detekčních elementech se opět začne akumulovat náboj a z CCD registru odchází nábojové svazky přes nábojový registr na výstupní zesilovač. Přenos dalších svazků může nastat až po vysunutí všech předešlých svazků z CCD registru ven. Pokud by se tak nestalo, docházelo by k degradaci náboje a tím zkreslení výstupního signálu.

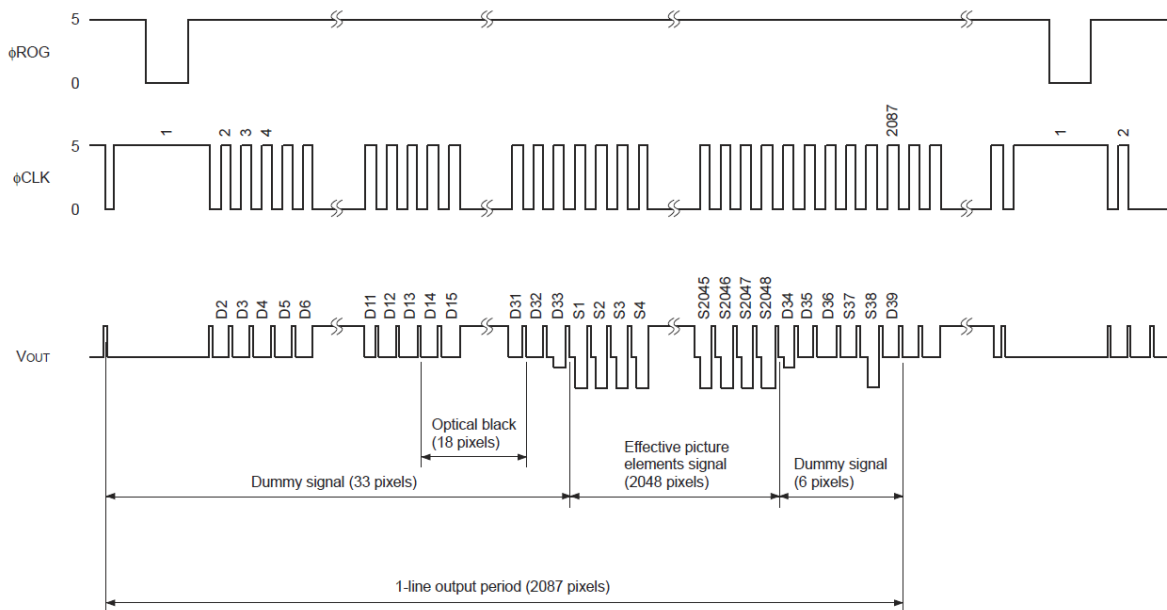
Existuje více variant řádkových CCD snímačů. Většinou se liší počtem CCD registrů a výstupních nábojových detektorů. Pro příklad je možná uvést například trilineární CCD snímač používající se pro snímání jednotlivých barevných složek signálu, z toho plyne, že máme tři výstupy.

Vybraný senzor ILX551A od společnosti Sony má pouze jeden CCD posuvný registr, jak je vidět na Obr. 3. Senzor má dvě napájecí napětí a to VDD1 5 V a VDD2 9 V. Velikost jednoho pixelu je 14 x 14 μm . Maximální hodnota hodinového signálu může být až 5MHz.



Obr. 3 Blokové schéma senzoru ilx551A [4]

Senzor je řízen pomocí dvou vstupních hodinových signálů a to CLK a ROG. Senzor funguje jako posuvný registr. Pokud pošleme impuls do vstupu CLK, na výstupu se objeví signál z prvního pixelu. Další signál znamená hodnotu druhého bodu v řádku. Signálem ROG začínáme vycitání nového řádku. Více je vidět na Obr. 4.



Obr. 4 Časový diagram senzoru ilx551A [4]

Fyzicky se na senzoru nachází 2087 bodů. Prvních 33 bodů a posledních 6 bodů jsou pixely, které nám určují hodnotu referenční černé barvy. Ty pro pozdější zpracování nebudou využívat. Celková perioda čtení řádku je rovna 2087 hodinovým signálům.

3.2 A/D převodník

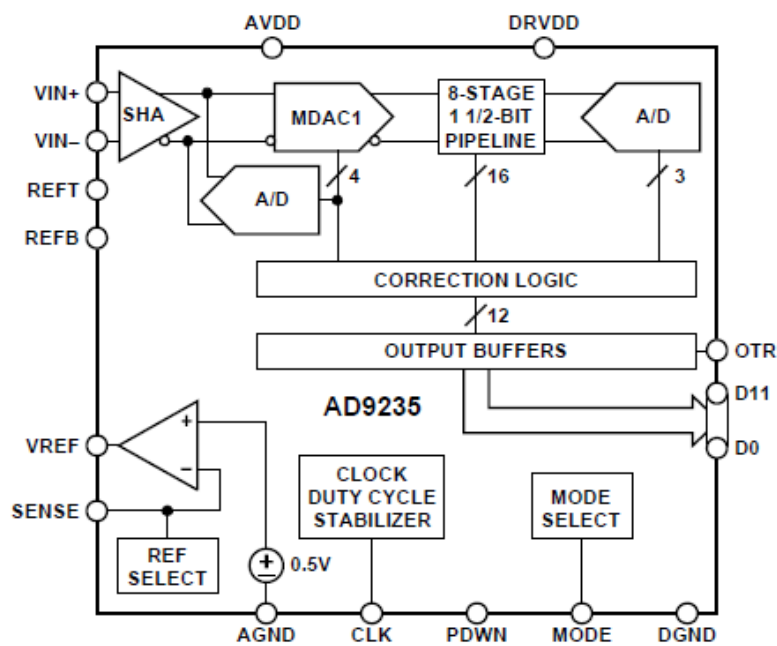
A/D převodník slouží pro transformaci vstupního spojitého analogového signálu na nespojitý digitální signál. Použití tohoto převodníku v obvodu je jednodušší pro zpracování a uchování dat obsažených v signálu. Převod spojitého signálu na diskretní je potřeba provést ve dvou krocích. Nejprve se provede vzorkování signálu a poté kvantování.

Vzorkování signálu je proces diskretizace v časové oblasti. Interval vzorkování se určuje podle vzorkovací frekvence. Chyby se při vzorkování můžeme dopustit, pokud zvolíme příliš malou vzorkovací frekvenci. Potom budeme přicházet o značné množství dat a digitální signál nebude odpovídat analogovému, nebo může nastat tzv. aliasing. Ten nastane, pokud je v původním signálu frekvence vyšší, než je polovina vzorkovací frekvence. Tento problém se dá vyřešit zařazením low pass filtru před převodník.

Kvantování je diskretizace oboru hodnot signálu. Jedná se o proces nevratný a ztrátový. U kvantování nastává největší problém s nastavením kvantizačních hladin. Nastavení je často závislé na citlivosti snímacího zařízení (v našem případě CCD snímače). Kvantizační hladiny od sebe mohou být

stejně vzdáleny, pak se jedná o lineární kvantizaci, nebo se jednotlivé vzdálenosti hladin mohou lišit, pak se jedná o nelineární kvantizaci. Pro barevné snímače se často používá kvantizace taková, aby byla každá barva zastoupena na výsledném obrázku stejným počtem pixelů.

Jako A/D převodník byl vybrán model od firmy Linear Technology, konkrétně AD9235. Jedná se o 12-bitový převodník s hodnotou 20 MSPS. Převodník je připojen na 3V napájení, přes filtrační kondenzátory. Časový signál CLK je opět vyveden z procesoru Spartan. Na převodník jsem přivedl napěťovou referenci o velikosti 1 V. Kvantizační šum se pohybuje podle výrobce na hranici 70 dB. Rozdíl výstupního napětí může být 1 V peak-peak až 2 V peak-peak.



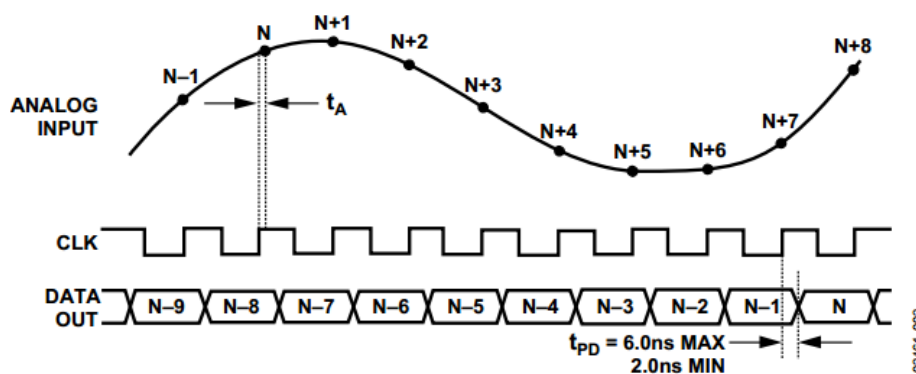
Obr. 5 Blokové schéma převodníku AD9235 [5]

Z Obr. 5. je vidět blokové funkční schéma převodníku. Vstupní obvod SHA je schopen pracovat až do vstupních frekvencí 100 MHz a může fungovat jak zapojení s jedním výstupem, nebo v diferenciálním zapojení. Pokud by vstupní signál byl mimo hodnoty, je indikován výstupem OTR. Tento výstup je vyveden přes konektor VHDCI do procesoru. Více o převodníku v [5].

Pin	Název Pinu	Funkce pinu
1	OTR	Indikátor překročení rozsahu
2	MODE	Formát dat a pracovní cyklus stabilizátoru
3	SENSE	Referenční režim hodin
4	VREF	Napěťová reference vstupu/výstupu
5	REFB	Diferenciální reference (-)
6	REFT	Diferenciální reference (+)
7,12	AVDD	Analogové napájení
8,11	AGND	Analogová zem
9	VIN+	Analogový vstupní pin (+)
10	VNI-	Analogový vstupní pin (-)
13	CLK	Vstup hodinového signálu
14	PDWN	Volba funkce Power-down
15 až 22 a 25 až 28	D0 AŽ D11	Datové digitální výstupy
23	DGND	Výstupní digitální zem
24	DRVDD	Digitální výstup ovladače napájení

Tab. 1 Popis funkcí pinů obvodu AD9235

Pro funkci výsledného obvodu je důležitý časový diagram, podle kterého budu data posílat do paměti. Protože v FPGA obvodech se tyto časové signály tvoří pomocí bloků stavového automatu je potřeba zde ukázat, z čeho jsem vycházel. Časový diagram je vidět na Obr.6. Z obrázku je vidět, že obvod reaguje na náběžnou hranu signálu.

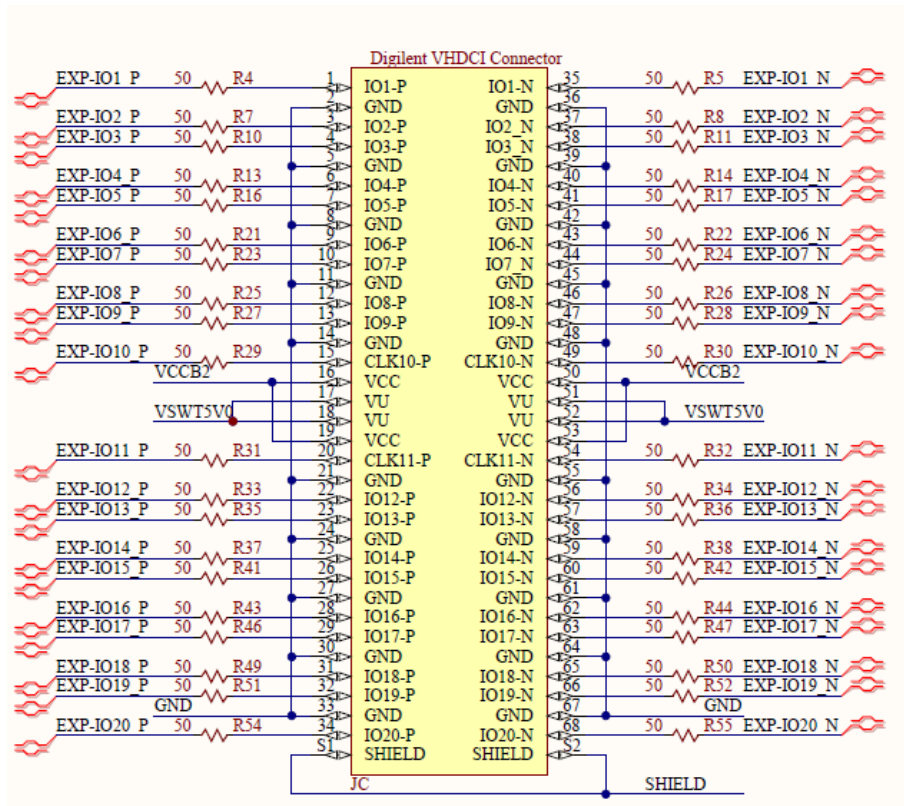


Obr. 6 Časový diagram převodníku AD9235 [5]

3.3 Konektor VHDCI

Jedná se o 68-pinový konektor. Konektor je určen pro rychlé datové přenosy, proto je každý třetí pin uzemněn. V návrhu musel být použit tento konektor, protože se nachází na vývojové desce společnosti Atlys. Konektor je přímo připojen na porty FPGA. Na Obr. 7 můžeme vidět zapojení

konektoru a v Tab. 2 je zapojení na konkrétní piny FPGA, je uvedeno zapojení pouze pro piny, které využívám k signálům.



Obr. 7 Blokové schéma konektoru VHDCI s popisem názvu propojení v designu FPGA[6]

Číslo pinu konektoru	Název pinu konektoru	Signál (Název podle schématu)	Název pinu FPGA
1	IO1-P	D0	IO_L2P_CMPCLK
3	IO2_P	D1	*IO_L5P
4	IO3_P	D2	IO_L13N_D10
6	IO4_P	D3	*IO_L15P
7	IO5_P	D4	IO_L16P
9	IO6_P	D5	*IO_L19P
10	IO7_P	D6	*IO_L20P
12	IO8_P	D7	*IO_L22P
13	IO9_P	D8	IO_L23P
15	CLK10_P	CLK AD	IO_L29P_GCLK3
16, 19	VCC	VCCIN	Nejde přímo na pin
17, 18	VU	VU	Nejde přímo na pin
20	CLK11_P	CLK S	IO_L30P_GCLK1_D13
22	IO12_P	D9	IO_L31P_GCLK31_D14
23	IO13_P	D10	*IO_L40P
25	IO14_P	D11	IO_L41P
29	IO17_P	ROG	IO_L45P
32	IO19_P	OTR	*IO_L47P
34	IO20_P	SHSW	IO_49P_D3
50, 53	VCC	VCCIN	Nejde přímo na pin
51, 52	VU	VU	Nejde přímo na pin

Tab. 2 Zapojení konektoru VHDCI na vývojové desce Atlys

3.4 FPGA

FPGA neboli programovatelná hradlová pole, jsou číslicové obvody umožňující realizaci složitých kombinačních a sekvenčních logických funkcí pouze prostřednictvím konfigurace obvodu, bez zásahu do zapojení. Historicky sahají programovatelná pole do programovatelných pamětí PROM a jejich zákaznických programovatelných pamětí EPROM a EEPROM, někdy kolem roku 1978. Přes drahé FPLA obvody vznikla koncepce PAL obvodů s programovatelnou maticí AND a pevnou maticí OR. Postupně se kvůli nárůstu fyzické velikosti obvodů začalo skládat více matic PLD obvodů do jednoho pouzdra. Dnes dostupné obvody mají příklon k hrubozrnným strukturám, kde se kromě programovatelných logických bloků vyskytují i další podpůrné bloky, které umožňují realizace složitých struktur, jako jsou např. číslicové filtry.

Podle způsobu konfigurace rozeznáváme dva základní typy FPGA a to volatilní a nevolatilní. Volatilní obvody využívají SRAM paměťové buňky. Jejich výhodou je jednoduchá rekonfigurace systému, která je možná za určitých okolností i za běhu. Nevýhodou je nutnost nahrání konfigurace při každém spuštění systému, což zvyšuje energetickou náročnost obvodu. Nevolatilní obvody nejčastěji používají flash paměťové buňky integrované přímo na čipu. Výhodou je okamžité spuštění, nemožnost kopírování, nižší spotřeba. Nevýhodou je však nemožnost rekonfigurace a vyšší cena oproti volatilní konfiguraci.

Obvody FPGA se skládají z programovatelných bloků. Nejčastěji umožňují realizovat logickou funkci pro několik vstupních bitových proměnných pomocí look-up tabulky následované paměťovým prvkem. Většinou registrem typu flip-flop. Jednotlivé programovatelné bloky jsou propojeny sítí spojů přes programovatelné propojovací matice. Moderní FPGA obvody obsahují také další prvky jako bloky pamětí, speciální rozvody hodinového signálu a resetu, hardwarové násobičky, bloky pro řízení hodin, atd.

Rodina Spartan 6

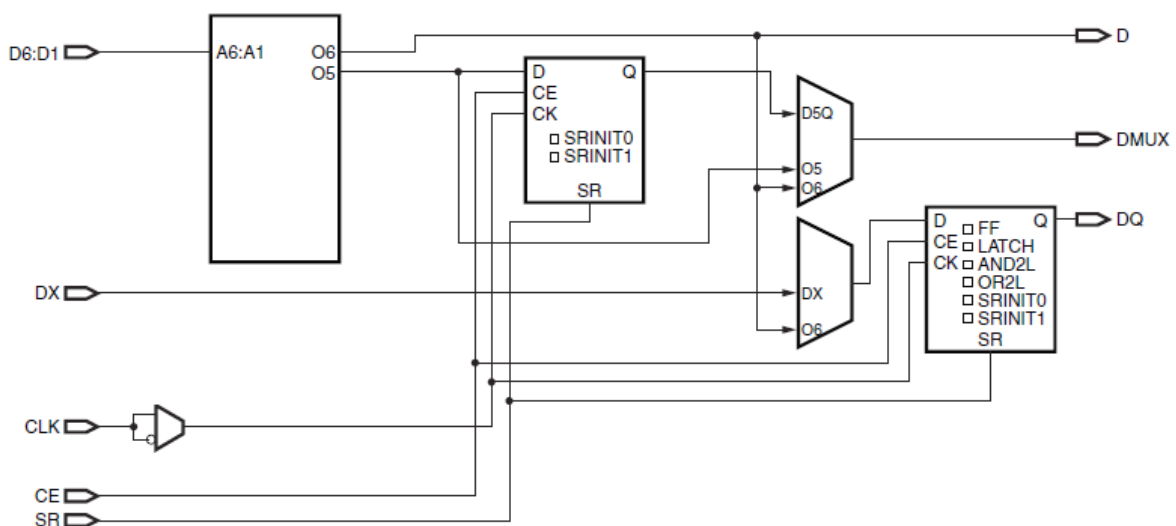
Tato práce byla implementována na obvodu Xilinx XC6SLX45-CSG324 z rodiny Spartan 6, který je přítomný na vývojové desce, proto zde bude popsán podrobně.

Spartan 6 je řadou obvodů firmy Xilinx pro méně až středně náročné aplikace, kde je brána i výsledná cena celého systému. Řada Spartan vždy navazuje a přenáší novinky obsažené v dřívější generaci Hi-end obvodů řady Virtex, v tomto konkrétním případě Virtex 5.

Obvod XC6SLX45-CSG324 obsahuje:

- 43 661 logických buněk
- 54 576 flip-flopů
- 58 DSP48A1 jednotek
- 2088 Kb blokové paměti
- 4 DCM a 2 PLL
- 2 řadiče pro externí dynamickou paměť SDRAM, DDR nebo DDR2
- 218 uživatelských vstupně výstupních pinů

Základní CLB buňka je u tohoto obvodu tvořena z jedné lookup tabulky s 6 vstupy a dvěma D flip-flopy. Datová cesta z tabulky je vedena přes multiplexory, což umožňuje řídit chování buňky. Schéma je na Obr. 8. Celou buňku tvoří čtyři takovéto struktury.



Obr. 8 základní struktura buňky obvodu Spartan 6 převzato [7]

Významným prvkem jsou jednotky DSP48A1. Jednotka obsahuje dvě nezávislé osmnáctibitové násobičky a čtyřicetiosmibitový akumulátor. Bloky DCM a PLL slouží ke generování a úpravě hodinového signálu, umožňují jeho násobení, dělení a fázový posun o násobek 45°. Tyto bloky je nutné použít vždy, pokud jsou v návrhu jiné hodiny, než jsou dostupné z vnějšího zdroje. Vlastní úprava implementovaná uživatelem není vhodná, jelikož FPGA využívá rozvod hodin po speciálních strukturách zaručující synchronní provoz na celém hodinovém stromě a definované zpoždění.

Detailní popis jednotlivých bloků lze nalézt v dokumentaci [].

Digilent Atlys

Pro implementaci systému byla použita vývojová deska Atlys společnosti Digilent.

Na desce najdeme:

- FPGA Spartan 6 XC6SLX45-CSG324
- 128Mbyte DDR2 paměť
- 10/100/1000 Ethernet PHY Marvel 88E111
- 16Mbyte x4 SPI Flash
- 100MHz CMOS oscilátor
- 8 LED, 6 tlačítek, 8 přepínačů
- Integrovaný JTAG USB adaptér Digilent
- 2 HDMI vstupy, 2 HDMI výstupy
- Zvukový kodek AC97
- USB na UART převodník
- 48 I/O pin na konektoru pro rozšiřující desku

3.5 Objektiv

Pro výběr objektivu je nutné definovat základní vstupní podmínky:

- Formát CCD snímače v kameře
- Požadované zvětšení objektivu
- Vzdálenost objektu od objektivu
- Přípustná změna zvětšení se změnou vzdálenosti předmětu od objektivu CCD kamery
- Požadované rozlišovací schopnosti
- Přípustné geometrické zkreslení objektivu
- Přípustný pokles intenzity ozáření CCD snímače v okrajích zorného pole ($\cos^4\omega$ pokles)

Při návrhu nemusí mít situace pouze jedno řešení, ale řešení může být parametrické. Pokud např. není pevně dána vzdálenost objektu od kamery, je možné dosáhnout požadovaného zvětšení a tím i pokrytí CCD snímače obrazem snímaného objektu volbou objektivu s různou ohniskovou vzdáleností při současné změně vzdálenosti od objektu.

Pokud je však dána buď vzdálenost objektivu od objektu, nebo naopak ohnisková vzdálenost objektivu při pevně daném požadovaném zvětšení, má situace jedno řešení. Vzhledem k tomu, že se objektivy vyrábějí s odstupňovanou řadou ohniskových vzdáleností, bývá řešení často kompromisem.

Pro snímání polí o velkém rozměru jsou potřeba objektivy s malou ohniskovou vzdáleností. Pak se musí počítat s poklesem $\cos^4\omega$. V Tab. 1 je vidět jak souvisí pokles intenzity ozáření snímače se

čtvrtou mocninou úhlu ω . Tento pokles by se měl ve videometrickém systému programově korigovat. Pokud by snímáný objekt neměl vždy stejnou polohu, je potřeba zohlednit změnu zvětšení objektivu. V tomto případě je pak lepší zvolit raději větší předmětovou vzdálenost, kdy s větší předmětovou vzdáleností klesá potřebný úhel obrazového pole objektivu.

ω	5°	10°	15°	20°	25°	30°	35°	40°	45°
$\text{Cos}^4\omega$	0,98	0,94	0,87	0,78	0,67	0,56	0,45	0,34	0,25

Tab. 3 Tabulka hodnot $\text{cos}^4\omega$ převzato [1]

Pokud by nastala situace, kdy je nutné dosáhnout určitého zvětšení při dané předmětové vzdálenosti a není k dispozici objektiv s ohniskovou vzdáleností, jaká odpovídá daným vstupním podmínkám, je možno použít objektivy s proměnou ohniskovou vzdáleností. Jedná se například o zoomovací objektivy.

V případě, kdy se měří malé objekty, nebo malý objekt svou vzdálenost při měření mění, lze použít telecentrický objektiv. Telecentrický objektiv obsahuje dvě spojné čočky, které jsou umístěny tak, že obrazové ohnisko přední čočky splývá s předmětovým ohniskem druhé čočky.

Na trhu se příliš neobjevují objektivy pro řádkové snímače. Jediné, kde se dají koupit, jsou přímo firmy zabývající se průmyslovými kamerami. Proto byl koupen jednoduchý objektiv pro maticový CCD snímač, avšak s tímto objektivem nebylo dosaženo optimálních výsledků.

3.6 DPS a její návrh

DPS neboli deska plošných spojů, je vyrobena jako dvouvrstvá ve velikosti 10x10 cm. Deska je koncipována tak, aby na jedné straně byl pouze vlastní senzor pro snímání obrazu, a na druhé straně se nachází veškerá elektronika. Rád bych zde popsal, jak jde signál ze senzoru, jak je vedeno napájení a jak jsou přivedeny signály na FPGA. Schéma desky se nachází v přílohách. Deska byla navržena v programu Altium Designer 14 a vyrobena ve firmě Pragoboard.

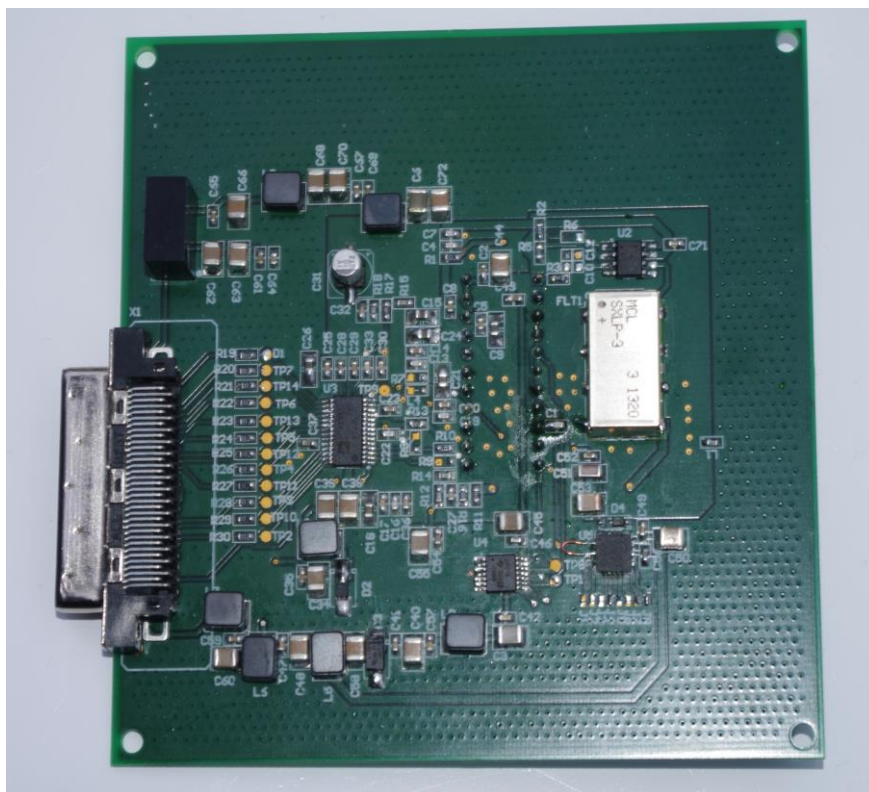
Signál ze senzoru je veden přes operační zesilovač AD8007, zapojený jako napěťový sledovač. Z operačního zesilovače pokračuje signál přes filtr SXLP-3. Tento filtr funguje jako dolní propust s mezním kmitočtem 3 MHz. Tím dochází k filtraci nežádoucích vyšších kmitočtů, které by nám zkreslovaly signál při převodu z analogového na digitální. Z filtru signál pokračuje na vstup A/D převodníku VIN-. A/D převodník je zapojen podle katalogového zapojení převzatého z datasheetu.

Bližší popis A/D převodníku je v kapitole 3.2. Výstupy A/D převodníku jsou přivedeny na konektor VHDCI. Více je vidět ve schématu v příloze.

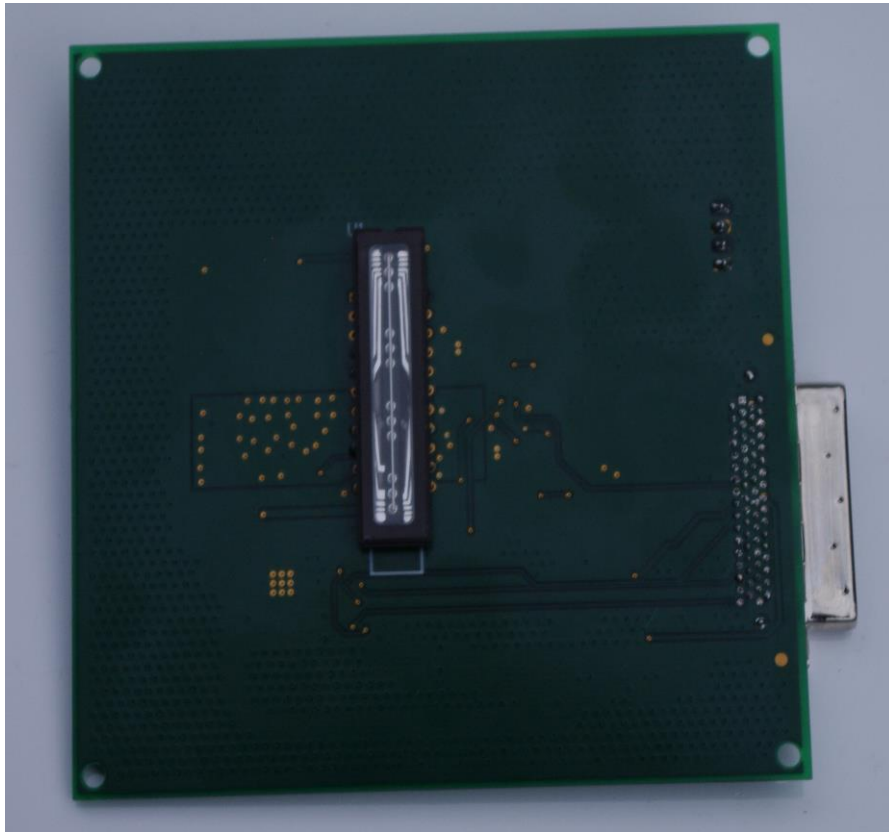
Na desce se nachází několik napájecích napětí. Ta jsou přivedena přes konektor VHDCI a to VCCIN ve velikosti 3,3 V a VU ve velikosti 5 V. Signál VU je veden na IE0509SH, což je DC/DC zvyšující měnič z 5 V na 9 V. 9 V napájení slouží jako druhé napájecí napětí pro senzor ILX551A a pro napájení operačního zesilovače AD8007. Signál VU je dále přiveden na obvod typu LDO TPS7A4700RGWT, který slouží pro vytvoření přesného napětí 3,3V, díky přesně definovaným vnitřním odporům. Toto napětí slouží jako analogové napájení A/D převodníku. Napájecí napětí VU je přes filtrační část dále ve schématu vedeno jako VCC_5V. Tento napájecí signál slouží k napájení senzoru.

Signál VCCIN je přes filtrační část dále veden ve schématu jako VCC_3,3V. Toto napájecí napětí je použito k digitálnímu napájení A/D převodníku.

Pro desky s A/D převodníky je důležité oddělit analogový a digitální signál. Proto navržená deska má dvě země označené jako AGND, což je zem pro analogovou část obvodu a pak GND což je zem pro digitální část obvodu. Obě země jsou na desce rozlity z obou stran. Pro spojení obou zemí je použito velké množství prokůvů. Země jsou spojeny přes cívku L1.



Obr. 9 Vyrobená DPS se osazenými součástkami (TOP Layer)



Obr. 10 Vyrobená DPS s osazenými součástkami (BOTTON Layer)

Toto je pouze návrh pro vývojovou desku. Pro sériovou výrobu by se musela navrhnout deska nová, která by byla i s procesorem a pamětí. To by znamenalo návrh několikvrstvé DPS. Navrhovaná deska by nemohla mít standardní prokovy, vzhledem k přítomnosti FPGA na desce, by deska musela být vyrobena technologií microwire. Tato technologie značně prodražuje desku. V návrhu by také nebyl přítomen VHDCI konektor, který by byl nahrazen ethernetovým konektorem případně USB. To vše je zohledněno v kapitole 7.

Problém by mohl také nastat ve volbě velikosti standardních smd rezistorů a kondenzátorů. Vzhledem k ručnímu osazení DPS na prototypu jsem zvolil nejmenší velikost pouzdra 0603. To už by nebylo možné na desce pro sériovou výrobu, kdy by bylo potřeba dosáhnout co nejmenších rozměrů. Proto bych volil většinu pasivních součástek ve velikosti 0405.

4 Systémový návrh

Navrhovaný systém musí být schopen předaná data z kamery sestavit do jednotlivých paketů a poslat je směrem k dalšímu zpracování a vykreslení obrazu. Vzhledem k složitosti tohoto procesu a pro lepší variabilitu systému bude tato obsluha řešena pomocí softwarového procesorového jádra

implementovaného v logice FPGA. Komunikace bude probíhat na implementaci systému z diplomové práce [7].

4.1 Návrh systému a výběr základních komponent

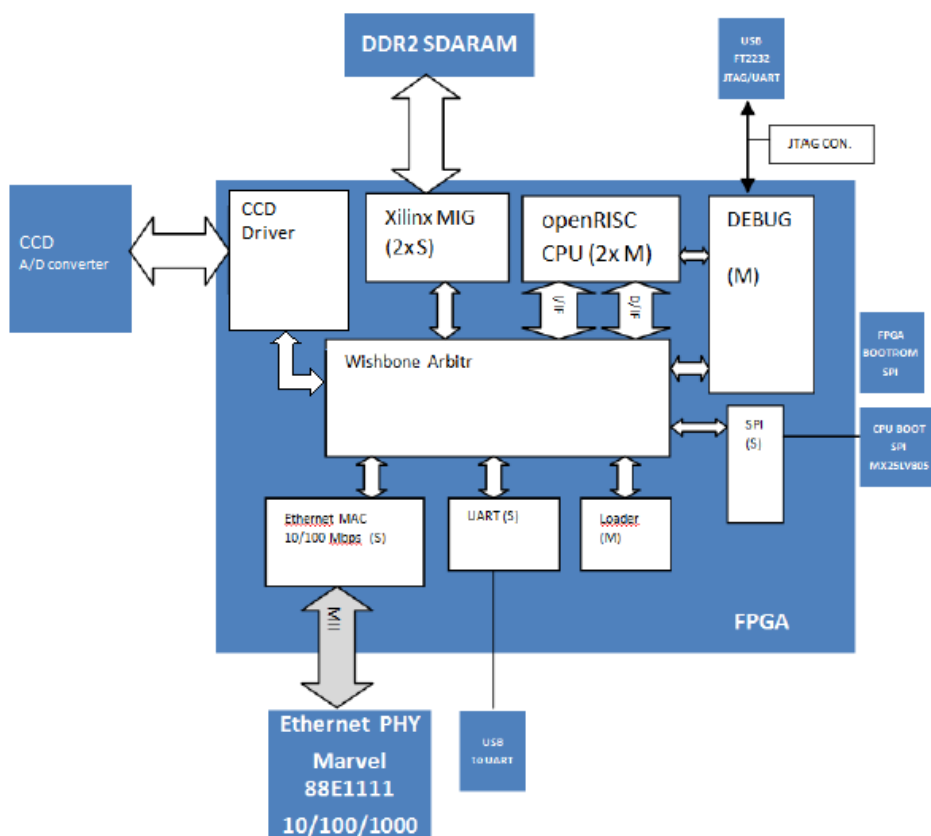
Vzhledem k navázání návrhu na již stávající aplikace bylo vybráno procesorové jádro OpenRISC [7] v implementaci OR1200. Jedná se o návrh procesoru, optimalizovaný pro FPGA, soustředěný kolem komunity *opencores.org* [8]. Jádro je 32 bitové s pětistupňovým pipeline. Obsahuje pouze základní periférie, jako Tick timer. K systému se připojuje přes dva kanály sběrnice Wishbone. Výhodou tohoto jádra je zjednodušené ladění (Debug), existence simulátoru a MMU. K dispozici je také port kompilátoru GCC, základních knihoven pro C a Linux. Více informací je možné získat v [7].

V návrhu musí být zohledněn také paměťový prostor pro program a pro data ze senzoru. Pro program bude využita externí paměť flash. Pro data ze senzoru bude použita paměť typu FIFO. Paměť FIFO se jeví jako nejlepší volba pro dočasné uložení dat ze senzoru.

Na Obr.11 je vidět hrubá představa kompletního systému, který je potřeba navrhnout. Část systému je převzata z [7]. Hlavní doplnění spočívá v bloku CCD Driver, který řídí vyčítání dat ze senzoru a jeho následnou distribuci. V bloku je přítomna paměť FIFO a stavový automat pro řízení hodin. Blíže bude celý systém popsán v následujících kapitolách.

Jádro obsahuje vstupy pro řízené napájení, které bude v projektu vytvořeno pro napájení desky s optickým senzorem.

Více o provedení návrhu komunikace je možné získat v [7].



Obr. 11 Blokové schéma systému doplněné o bloky pro CCD

4.2 RTL návrh

Uvedeme si zde popisy jednotlivých bloků navrhovaného systému.

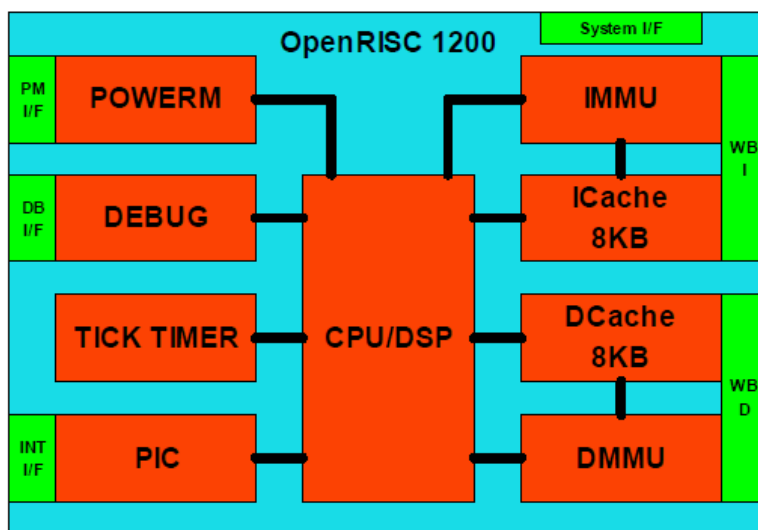
4.2.1 CPU OpenRISC

Procesor OR1200 je implementací architektury OpenRisc, která je vyvíjena kolem projektu opencores. K jádru jsou dodávány zdrojové kódy v jazyce Verilog, překladač GCC, port knihovny uClib, základní GNU nástroje a simulátor orklsim s podporou GNU Debuggeru. Díky tomu je implementace jednodušší.

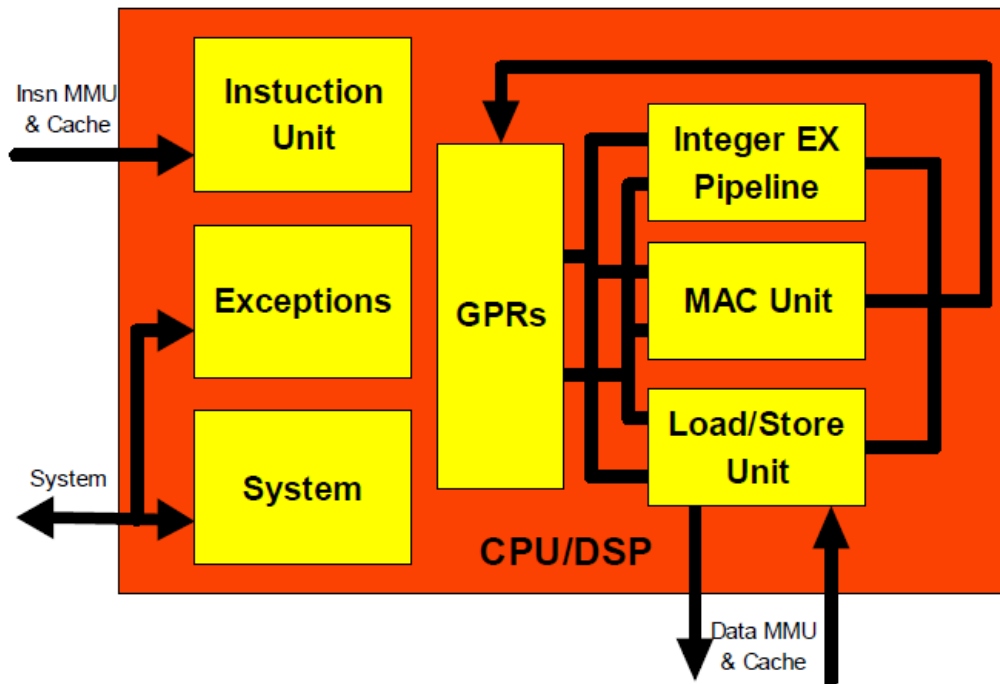
Procesor OR1200 může být nakonfigurován jako 32 bitový s podporou MMU umožňující využití virtuální paměti, čímž je mimo jiné vhodný pro běh multitaskových operačních systémů, například linuxu. Procesor má oddělenou datovou a instrukční sběrnici. Pro ASIC implementace je obsažena

podpora režimů se sníženou spotřebou. Instrukční sada procesoru je typ RISC, s možností rozšíření o vektorové operace. Využita je pětistupňová pipeline. Díky tomu je možné většinu instrukcí vykonávat v jednom cyklu.

Jádro je přednostně navrženo pro užití na FPGA, a implementace byla otestována na obvodech všech hlavních výrobců (Xilinx, Altera, atd.). V minulosti byly také provedeny zkušební implementace jako obvodu typu ASIC, do budoucna je počítáno se sériovou výrobou. V dosavadních ASIC implementacích s využitím 180 nm výrobního procesu, bylo dosaženo při taktovací frekvenci 250 MHz, teoretického výkonu až 250 DMIP. Při implementaci na nejlepších dostupných FPGA je možné dosáhnout až k 100 DMIPS. Proto se procesor jeví jako perspektivní pro řadu aplikací. Topologie jádra je znázorněna na Obr. 12 obsahuje vlastní CPU jádro, jednotky Wishbone rozhraní v režimu master pro připojení datové a instrukční sběrnice, volitelnou datovou a instrukční cache, programovatelný řadič přerušení (až 32 externích kanálů), jednotky správy napájení, jednotku Debug pro ladění aplikací a tick timer.



Obr. 12 Jádro OR1200 převzato [9]



Obr. 13 CPU/DSP jádro, převzato z [10]

Na Obr. 13 je znázorněno základní jádro implementující architekturu openRISC. Základem je Instruction Unit, která je zodpovědná za načítání instrukcí z paměťového podsystému, pipeling, správa fází při zpracování instrukce, vyvolání výjimky a provádění skoků. Další podstatnou částí je soubor 32 pracovních registrů.

Neméně důležitou částí je Integer Execution Pipeline, jedná se v podstatě o ALU. Implementuje operace porovnání, logické, aritmetické, posuny a rotace. Většinu podporovaných operací provádí v jednom cyklu.

Blok Load/Store Unit se stará o přenos dat mezi pracovními registry, vnitřní sběrnici modulu a paměťovým subsystémem. Také se stará o zarovnání a implementaci adresových režimů. Dokáže pracovat částečně nezávisle na zbytku jádra.

Další části jsou MAC jednotka, určená především pro DSP operace a pracující s 32x32 bit operandy a 48 bitů širokým akumulátorem. Dále Systém Unit pro řízení a konfiguraci jádra prostřednictvím SFR. Správu výjimek má na starosti blok Exceptions. Je možné také implementovat blok pro práci s plovoucí řádkovou čárkou.

Implementace Cache je provedena nezávisle pro data a instrukce, jako jednocestná přímo mapovaná. Velikost je konfigurovatelná od 512B po 8KB. Je možné jí také zakázat.

4.2.2 Wishbone

Již dříve bylo uvedeno, že k propojení systému bude použita sběrnice Wishbone. Sběrnice byla použita v předchozí práci, a proto bylo nutné provést implementaci na této otevřené sběrnici. Mezi její výhody patří:

- Šířka 8 až 64 bitů
- Podpora paralelního, kaskádního a point to point zapojení
- Široká podpora v open source projektech
- Optimalizace pro využití v SoC a FPGA
- Licencována jako GPL

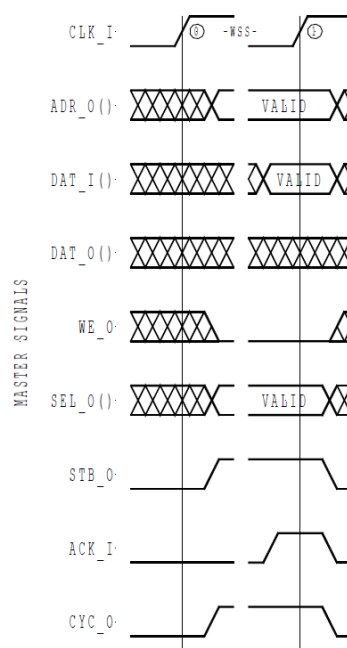
Sběrnice je především určena pro interní propojení modulů systémů implementovaných na FPGA a ASIC. Sběrnice umožňuje připojení zařízení s menším datovým slovem, než je její nominální datová šířka, a signalizaci platných dat. Od verze v3 je rozšířena o podporu blokových přenosů. Jsou rozlišovány dva typy zařízení – Master (zařízení, které může přistupovat ke sběrnici a ovládat podřízená zařízení) a Slave (řízená zařízení). Zařízení jsou mezi sebou obvykle propojena pomocí přepínače, který se stará o rozpoznávání adresy, propojení sběrnice a arbitrace přístupu. Signály, která sběrnice využívá, jsou vidět v Tab. 2

Společné signály	Šířka [b]	Popis	Směr
DAT_I	N	data	I
DAT_O	N	data	O
CLK_I	-	hodiny	I
RST_I	-	reset rozhraní	I
TAGN_I	K	uživatelsky def.	I
TAGN_O	K	uživatelsky def.	O
Signály master			
STB_O	1	začátek rámce	O
CYC_O	1	začátek cyklu	O
SEL_O	N/8	platná data	O
WE_O	1	čtení/zápis	O
ADR_O	N	adresová sběrnice	O
RTY:O	1	opakování cyklu	I
ERR_I	1	indikace chyby	I
Signály slave			
STB_I	1	začátek rámce	I
CYC_I	1	začátek cyklu	I
SEL_I	N/8	platná data	I
WE_I	1	čtení/zápis	I
ADR_I	B	vstup adresy	I
RTY_I	1	opakování cyklu	O
ERR_I	1	indikace chyby	O

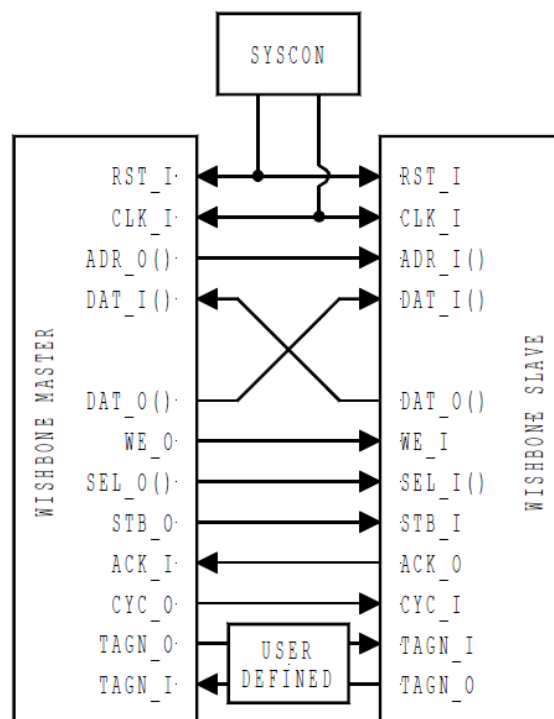
Tab. 4 Popis signálů sběrnice Wishbone, převzato z [7]

Master je zodpovědný za řízení sběrnice, generuje adresu a požadavek o její přidělení. Jednoduchý cyklus probíhá následujícím způsobem, více Obr. 14

1. Master nastaví adresu (`adr_o`), příznak platných dat datového slova (`sel_o`), příznak zápisu ('1' = zápis, '0' = čtení), příznak začátek rámce (`stb_o`) a nastavením příznaku začátku cyklu (`cyc_o`) požádá o přidělení sběrnice.
2. Je-li přítomen arbitr-switch sběrnice, dochází k dekódování adresy a propojení na danou slave periferii.
3. Periferie je aktivní příznakem cyklu `cyc_i`, '1' se začíná se zpracování dat.
4. Slave dokončil zpracování, dle výsledku nastavuje výstup (výstupní data, chyba, požadavek pro opakování cyklu) a nastavením signálu indikuje master dokončení požadavku.
5. Master nastavuje příznak cyklu a začátek rámce do '0', tím uvolňuje sběrnici, arbitr-switch ruší propojení.



Obr. 14 Jednoduchý cyklus na sběrnici Wishbone převzato [11]

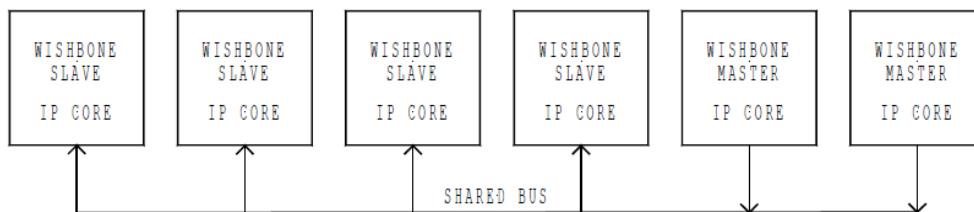


Obr. 15 Základní propojení Slave a Master bloků převzato [11]

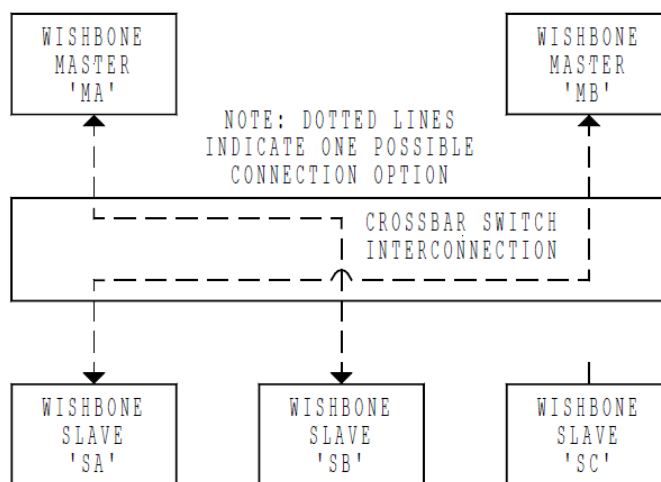
Na Obr. 15 je ukázáno základní propojení Master a Slave jednotek bez použití arbitr-switch. Jedná se o propojení, které se vytvoří po přidělení sběrnice a dekodování adresy.

V praxi je na sběrnici připojeno více zařízení obou typů. Proto je potřeba využít zařízení, které umožňuje jejich vzájemné propojení definovaným způsobem, řízení a arbitraci přístupu. Toto můžeme řešit třemi následujícími způsoby.

1. Jako multiplexor, který umožňuje propojení vždy je jednoho master zařízení s jedním slave zařízením v jednom okamžiku. Obr. 16
2. Jako tzv. Crossbar, Obr. 17, který umožňuje více paralelních spojení. Vzhledem k absenci třístavových budičů ve struktuře obvodů FPGA je nutné zajistit, aby nedocházelo ke kolizím mezi výstupy (tzn. nelze je přímo spojit). Je výhodné, aby bylo možné řídit priority přístupu jednotlivých master zařízení ke sběrnici. Také je výhodné zajistit, aby nemohlo dojít k zablokování sběrnice chybou některého připojeného zařízení, což by znemožnilo dohledání případné chyby. Propojovací matice dále zajišťuje dekodování adres a následný přístup k slave zařízením.
3. Poslední možností je kaskádní zapojení jednotlivých bloků za sebe, při čemž každá obsahuje rozhraní pro slave a master, tento typ propojení není ve většině dostupných jader použit.



Obr. 16 Propojení se sdílenou sběrnici z [11]

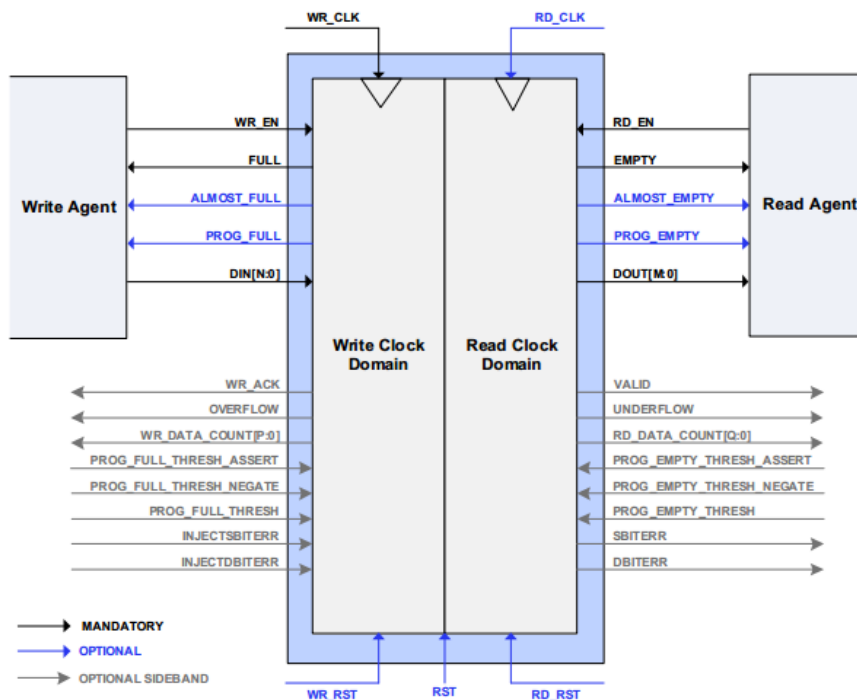


Obr. 17 Propojení s využitím Crossbar převzato z [11]

V revizi „b3” byla specifikace rozšířena o podporu datových přenosů, to znamená, že v jednom cyklu je možné přenést více datových slov, a to bez uvolnění sběrnice.

4.2.3. Paměť FIFO

Paměť FIFO neboli first-in first-out je nutné použít z důvodu odesílání paketů ke zpracování. Pro moje použití bude stačit paměť, vytvořená jako blok přímo v FPGA. Při tvorbě paměti FIFO se přímo nabízí použití designu společnosti Xilinx, jediné čím je paměť omezena je velikostí paměti v použitém FPGA, který umožňuje velikost vnitřní paměti pouze 116kB. Pro moje použití je tato velikost ovšem dostatečná.

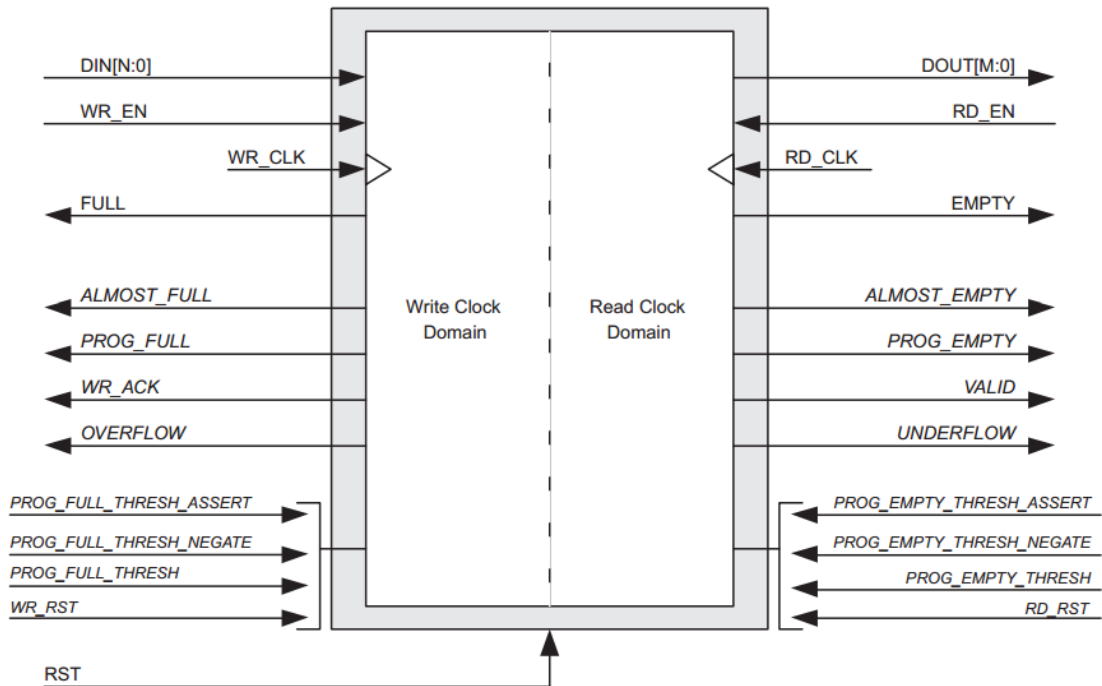


Obr. 18 Schéma paměti FIFO z vnitřního designu pro libovolnou sběrnici[12]

- Datová šířka 1-1024 bitů
- Symetrický nebo nesymetrický poměr stran (poměr čtení ku zápisu od 01:08 až po 08:01)
- Možnost synchronního nebo asynchronního resetu
- Volitelný typ paměti (blok RAM, distribuovaná RAM, posuvný registr nebo vestavěná FIFO)
- Možnost pracovat v normálním režimu nebo FWFT režimu (First-Word Fall-Through)
- Příznaky pro indikaci plné a prázdné paměti
- Možnost nadefinovat příznaky pro plnou nebo prázdnou paměť

Velikou výhodou vygenerované FIFO paměti je nesymetrický poměr dat pro zápis a čtení z paměti. Tím dosáhneme automatické konverze dat. Poměr mezi vstupním signálem a výstupním signálem může dosáhnout 08:01 nebo až 01:08. Tohoto poměru dosáhneme, pokud nastavíme nezávislé hodiny pro čtení a pro zápis.

Na Obr. 19 jsou vidět důležité signály, které je potřeba definovat v designu celého systému. Jednotlivý popis a funkce těchto signálů je popsána v Tab. 5. V tabulce je uvedena pouze část signálů a jejich zkrácený popis. Kompletní informace jsou k dispozici v [12].



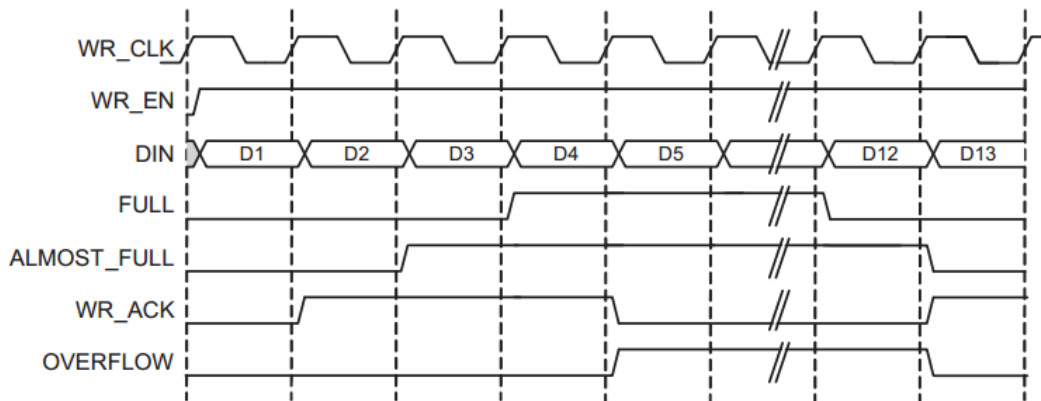
Obr. 19 FIFO s nezávislým časováním: propojovací signály [12]

Jméno	Typ výstupu	Popis
RST	IN	Reset: asynchronní reset signálu, který inicializuje všechny vnitřní ukazatele a výstupní registry
Zápisové signály		
WR_CLK	IN	Write clock: Hodinový signál pro zápis dat.
DIN [N:0]	IN	Data Input: vstup dat pro zápis po sběrnici
WR_EN	IN	Write Enable: Pokud není FIFO naplněná, tento vstup určuje, že se data mohou do paměti stále ukládat.
FULL	OUT	Full Flag: Pokud je FIFO plná, signál bude v log. 1 a zápis od paměti bude pozastaven, dokud se v paměti neuvolní místo.
WR_RST	IN	Write Reset: Po proběhnutí inicializuje všechny ukazatele a příznaky zápisových hodin.
ALMOST_FULL	OUT	Almost Full: Detekuje situaci, kdy po další zápisu se paměť naplní.
PROG_FULL	OUT	Programmable Full: Detekuje úplné zaplnění paměti, není možnost zapsání dalších dat
WR_ACK	OUT	Write Acknowledge: Signál nám detekuje, že data byla do paměti zapsána. Jedná se o jeden cyklus.
OVERFLOW	OUT	Overflow: Signál nám detekuj, že data nebyla do paměti zapsána, protože je FIFO paměť naplněna.

Signály čtení		
RD_RST	IN	Read reset: Vstup hodinového signálu, inicializuje všechny ukazatele, příznaky a výstupní registry.
RD_CLK	IN	Read Clock: Hodinový signál pro čtení.
DOUT[M:0]	OUT	Data Output: výstup dat z paměti po sběrnici
RD_EN	IN	Read Enable: pokud jsou ve FIFO data, pak nám signál určuje, že mohou číst data za paměti
EMPTY	OUT	Empty Flag: pokud je paměť prázdná je signál v log. 1 a paměť je zablokována pro čtení.
ALMOST_EMPTY	OUT	Almost Empty Flag: Určuje, že ve FIFO je pouze poslední slovo na čtení, takže je paměť téměř prázdná
PROG_EMPTY	OUT	Programmable Empty: signál udává, pokud je v paměti počet slov menší nebo rovno velikosti výstupu.
VALID	OUT	Valid: data jsou k dispozici na výstupu sběrnice
UNDERFLOW	OUT	Underflow: označuje, že požadavek na čtení během předchozího taktu byl odmítnut, protože je paměť prázdná

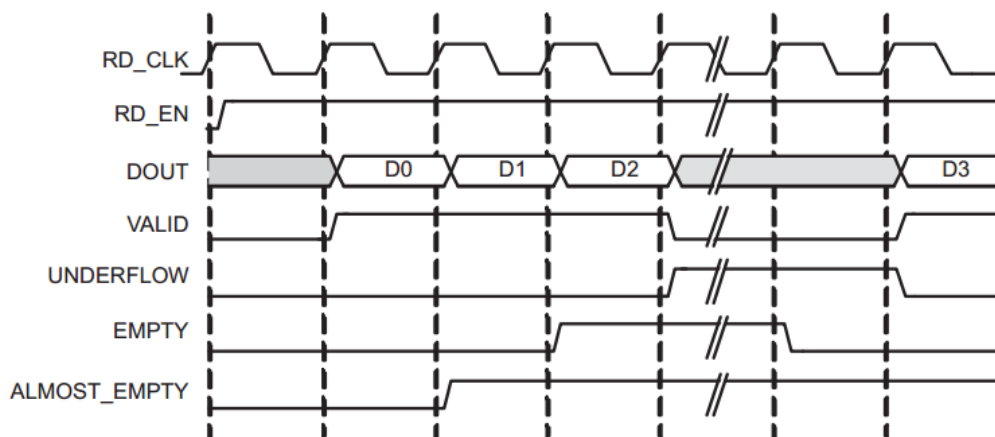
Tab. 5 Popis funkcí jednotlivých vstupů a výstupů paměti FIFO

Zápis do paměti FIFO probíhá podle Obr. 20. Pokud pošleme signál WR_EN do logické 1 tak na následující náběžnou hranu WR_CLK dojde k zápisu do paměti. Pokud není FIFO plná tak překlopením signálu WR_ACK so logické 1 se potvrdí zápis dat do paměti. Pokud nastane situace, kdy je paměť před naplněním a vejde se do ní poslední příchozí paket, tak signál ALMOST_FULL přejde do stavu logická 1. Pokud je paměť zaplněna přejde signál FULL do stavu logická 1. Pokud bychom chtěli poslat další paket, spadne signál WR_ACK do stavu logická 0 a signál OVERFLOW přepne do stavu logická 1, čímž oznamuje přetečení paměti. Jakmile bude prováděna jedna nebo více žádostí o čtení paměti, signál FULL se opět dostane do stavu logická 0 a umožní další zapsání do paměti. Signály WR_ACK a OVERFLOW se překlopí opět do stavu pro možný zápis dat.



Obr. 20 Operace zápisu do paměti pro synchronní hodiny [12]

Pro čtení paměti je důležité, aby ve FIFO byl alespoň jeden paket, což způsob čtení je patrný z Obr. 21. Pokud skočí signál RD_EN do logické 1. K operaci čtení může dojít následující hodinový puls signálu RD_CLK. Zároveň signál VALID přejde do stavu logická 1. Pokud je signál VALID v log. 1 znamená to, že jsou data v paměti přítomna. Pokud je v paměti poslední paket signál EMPTY přejde do stavu log. 1. Pokud se bude uživatel nadále snažit o čtení paměti, je požadavek na čtení ignorován. Pokud nejsou v paměti data je v log. 1 signál UNDERFLOW a signál VALID v log. 0.



Obr. 21 Operace zápisu do paměti pro synchronní hodiny [12]

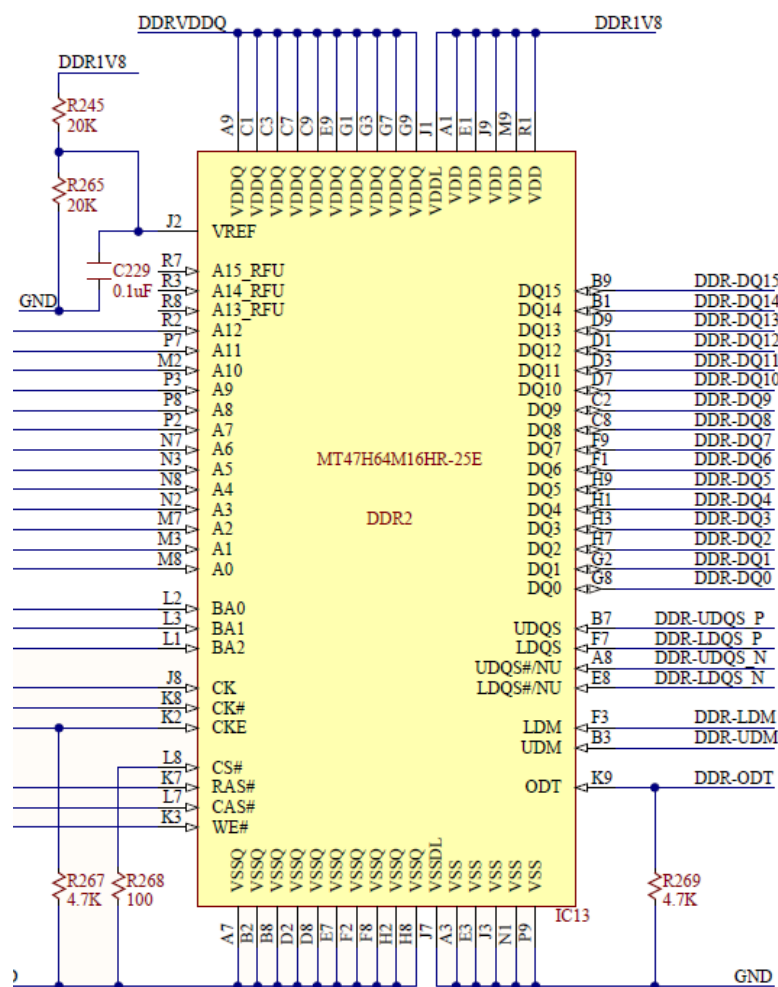
4.3 Implementace paměti DDR SDRAM

Protože ve vybraném FPGA lze využít pouze 128 KB pro celý použitý design, bylo potřeba použít externí paměť DDR2 integrovanou na desce vývojového kitu. Paměť má označení Micron MT47H64M16-25E. Vzhledem k tomu, že ovládání tohoto typu paměti je značně komplikované a neobejde se bez hardwarové podpory, nabízelo se využití integrovaného paměťového řadiče v obvodu. Řadič byl nakonec vytvořen pomocí Xilinx interface generátoru. Toto řešení má velkou nevýhodu v nepřenositelnosti mezi různými FPGA, pro jiné výrobky Xilinx je možnost jádro pouze přegenerovat.

V případě jiných FPGA zcela nahradit. Údaje pro správnou konfiguraci je třeba získat z [13]. Výhodou tohoto řešení je rychlá a snadná konfigurace.

Pro připojení k systému přes Wishbone sběrnici je nutné vytvořit blok, který zajistí obsluhu sběrnice a řadič paměti. Tento blok byl navržen tak, že vnější rozhraní pro připojení do systému jsou dva nezávislé kanály typu wishbone slave.

Na Obr.22 je možné vidět zapojení paměti na vývojové desce. Vzhledem k množství propojení na desce od procesoru k paměti je nutné si hlídat správný zápis propojení. Snadno se v tomto textovém zápisu udělá chyba a špatně se tato chyba hledá.



Obr. 22 Schéma DDR2 s názvem propojovacích drátů [6]

5 Verifikace

Verifikace je nezbytnou součástí vývoje číslicového systému. Díky ní můžeme snadněji zjistit, jak se obvod chová a odhalit tak jeho chyby. Většinou se verifikace provádí na dvou úrovních. Jedná se

o RTL verifikaci, která ověřuje pouze chování systému daného vlastním zápisem kódu HDL, ale při zanedbání reálií jako jsou propojení a jednotlivé elementy. Druhou částí je verifikace na tzv. hradlové úrovni, která bere tyto jevy v úvahu a více se přibližuje chování fyzického obvodu.

Verifikace celého systému je velmi problematická, z důvodu velké složitosti. Proto se verifikace většinou provádí po jednotlivých blocích systému. V jednotlivých blocích budou však verifikovány pouze některé základní stavy, jako časování, reset, vyčítání dat, zápis do FIFO, ad. Verifikování pouze některých stavů je z důvodu velké časové náročnosti takového procesu.

Více o verifikaci je k dispozici v [12]

Stručně si zde popíšeme jednotlivé části Obr. 11, hlavní popis však bude spočívat v části CCD Driver, který je nosnou částí této práce.

5.1 CCD Driver

Tato část celého systému se skládá z FIFO paměti, stavového automatu CCD, tři registrů a Wishbone arbitru. Pro mou práci se dalo využít již implementovaný design FIFO paměti, který nabízí přímo firma Xilinx v simulátoru ISE Webpack. O celé řízení tohoto bloku se stará stavový automat CCD, který je zodpovědný za vyčítání dat ze senzoru, následovanou správnou funkcí A/D převodníku s uložením a vyčtením dat do a z paměti FIFO. Pro správnou funkci bylo nutné nastudovat jednotlivé waveformy, které se nachází v datasheetech. Určitý problém se vyskytl při zápisu a čtení dat z paměti. Maximální frekvence senzoru je 1 MHz, kdežto sběrnice běží na 50 MHz. Z toho plyne, že vyčítání dat může být mnohem rychlejší než jejich zápis.

Pro odzkoušení funkce CCD Driveru byl vytvořen Testbench, který je implementován v souboru CCD_TB.vhd. Výsledky jsou vidět jako časové průběhy.

První co bylo potřeba odzkoušet a simulovat, bylo vyčítání dat ze senzoru a poslání binárních dat z A/D převodníku na vstup paměti FIFO. V designu byla nastavena pro vstup jako 8 bitová. Pro výsledný designe však bude již upravena na 12-ti bitovou. Této velikosti bylo dosaženo pouhým oříznutím vrchních 4 bitů. Díky tomuto zásahu se však systém lépe ladil. Celý systém se rozbíhá po globálním resetu označeném jako `rst_i`.

Senzor se spíná jedním hodinovým cyklem do nuly. Po tomto pulzu musí do nuly spadnout signál ROG, označeném jako `ccd_rog`, na dobu mezi 500 až 1000 ns. Tím dosáhneme náběžné hrany, na kterou nám začne reagovat signál CLK, označeném jako `ccd_clock`. Tím dojde k funkci hlavních hodin a

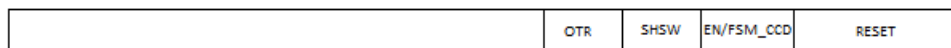
vyčítání dat ze senzoru. Prvních 33 pixelů, jsou tzv. dummy. Ty si přes čítač do 33 odečteme a budeme je ignorovat. Následuje vyčítání všech 2048 pixelů, kdy na toto vyčtení je nastaven další čítač počítající do 2048. Zbylých 6 pixelů opět ignoruji. Tuto funkci hodin je možné vidět na simulaci na obrázku Obr. 23



Obr. 23 Čtení dat ze senzoru

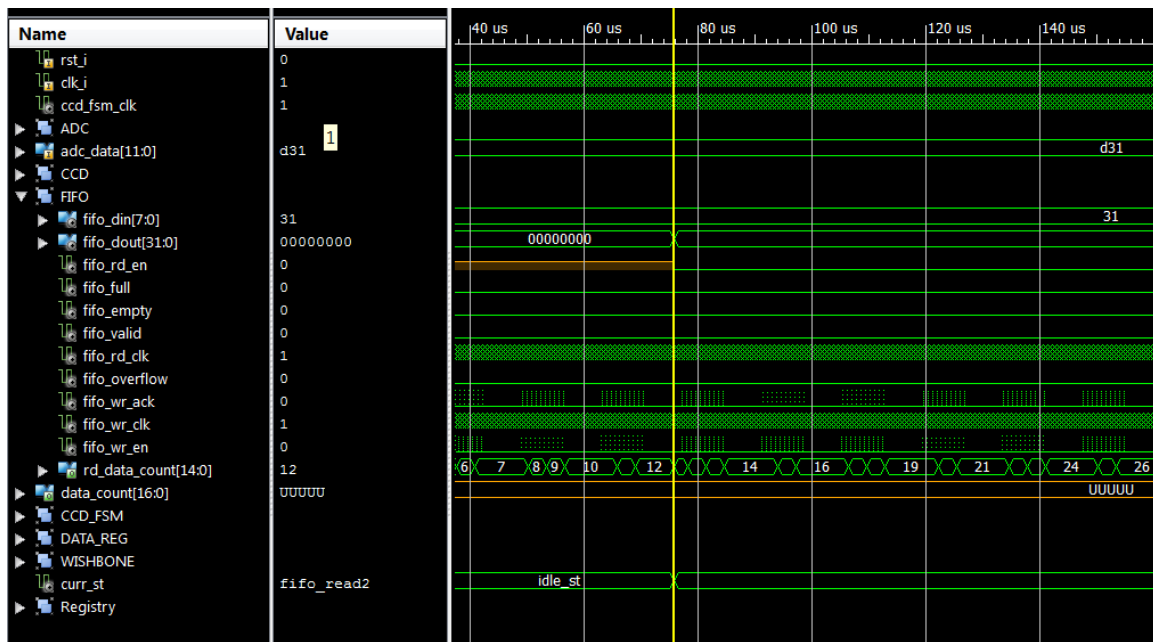
Pro lepší názornost je počet dummy pixelů snížen na 5 a počet obrazových pixelů na 10. Tato změna je pouze z důvodu simulace. Jinak by simulace byla příliš nepřehledná. Přidán je také hodinový signál A/D převodníku, který reaguje na hodinový signál senzoru. V části CCD_FSM je také dobře vidět počítání čítačů pro dummy pixely a pro obrazové pixely.

Signály, které rozhodují o nastavení CCD snímače (SHSW) a o nastavení A/D převodníku (OTR), jsou umístěny v registru CONTROL. Na prvním bitu tohoto registru se nachází globální reset. Druhý bit zapíná stavový automat CCD. Schéma registru je vidět na Obr. 24



Obr. 24 CONTROL registr

Po získání dat z CCD musíme vyřešit jejich zápis do paměti FIFO. Tato část simulace musí probíhat tak, že hodiny pro zápis běží na stejné frekvenci jako hodiny CCD a to 1 MHz. Zápis proběhne následující hodinový cyklus po fifo_wr_en do jedničky. Pokud zapíše prvních osm bitů správně, do registru FIFO_DATA se přičte jednička až do čísla 2048. Jakmile v registru dosáhnou této hodnoty, vím, že se vyčetl jeden řádek. Registr vynuluje a může začít snímání nového řádku.



Obr. 25 Zápís dat do FIFO a čtení z FIFO

Paměť FIFO a jednotlivé stavy všech důležitých signálů pro zápis dat a jejich čtení je uloženo v registru STATUS. Tento registr má velikost 32 bitů. Využíváno je pouze prvních 14 bitů. Schéma tohoto registru je možné vidět na Obr. 26

	UNDERFLOW	VALID	PROG_EMPTY	ALMOST_EMPTY	EMPTY	RD_EN	RD_RST	OVERFLOW	WR_ACK	PROG_FULL	ALMOST_FULL	WR_RST	FULL	WR_EN
--	-----------	-------	------------	--------------	-------	-------	--------	----------	--------	-----------	-------------	--------	------	-------

Obr. 26 STATUS registr

6 Implementace

Implementace byla prováděna v prostředí Xilinx ISE 14.7 ve verzi webpack, na FPGA Xilinx Spartan 6 XC6SLX45 pro pouzdro CbSG324.

Po spuštění syntézy bylo vygenerováno velké množství varování, většina se však týkala nezapojených pinů a nepoužitých signálů. Nezapojené piny se nacházeli především v procesoru, což je dáno jeho konfigurovatelnou podstatou. Seznam byl velmi obsáhlý a nebudu jej zde uvádět.

Po syntéze je design implementován na FPGA s využitím zdrojů dle Obr. 27

Device Utilization Summary			
Slice Logic Utilization	Used	Available	Utilization
Number of Slice Registers	5,413	54,576	9%
Number of Slice LUTs	11,246	27,288	41%
Number used as Memory	1,841	6,408	28%
Number of bonded IOBs	109	218	50%
Number of RAMB16BWERs	65	116	56%
Number of BUFG/BUFGMUXs	9	16	56%
Number of DSP48A1s	4	58	6%

Obr. 27 Využití zdrojů po syntéze [9]

Dosažitelné časování po syntéze:

Timing Summary:

Speed Grade: -3

Minimum period: 65.760ns (Maximum Frequency: 15.207MHz)

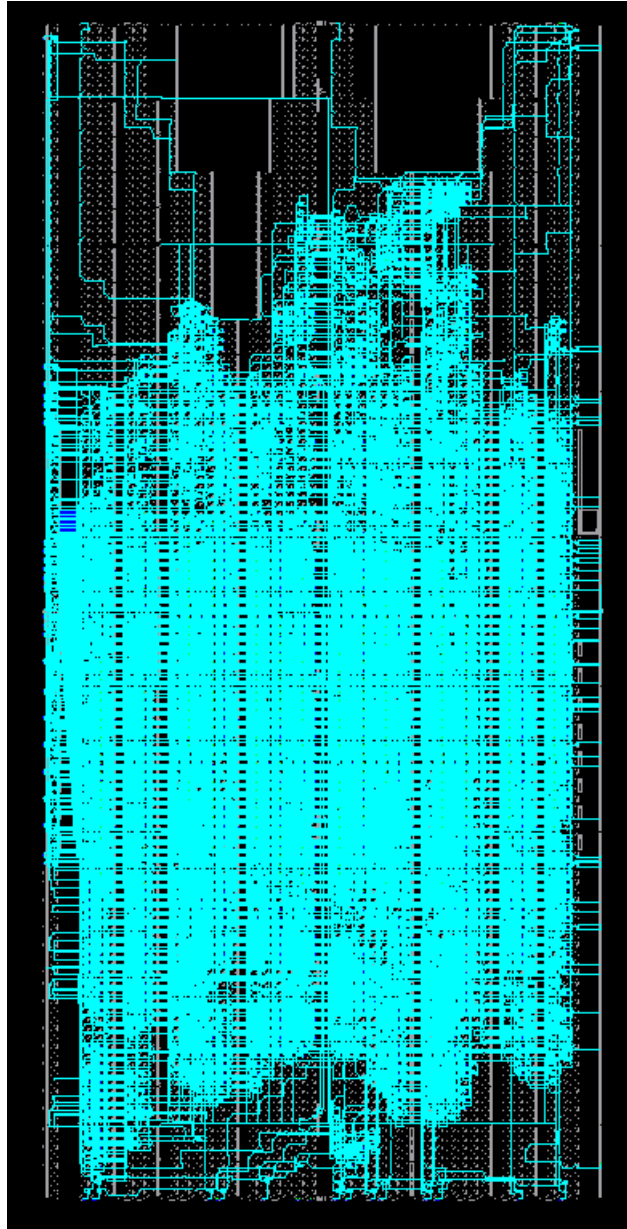
Maximum combinational path delay: No path found

Využití logiky po PaR je následující:

Slice Logic Utilization:

Number of Slice Registers: 5,413 out of 54,576 9%
 Number used as Flip Flops: 5,351
 Number used as Latches: 59
 Number used as Latch-thrus: 0
 Number used as AND/OR logics: 3
Number of Slice LUTs: 11,246 out of 27,288 41%
 Number used as logic: 9,299 out of 27,288 34%
 Number using O6 output only: 7,795
 Number using O5 output only: 250
 Number using O5 and O6: 1,254
 Number used as ROM: 0
Number used as Memory: 1,841 out of 6,408 28%
 Number used as Dual Port RAM: 1,832
 Number using O6 output only: 1,420
 Number using O5 output only: 0
 Number using O5 and O6: 412
 Number used as Single Port RAM: 0
 Number used as Shift Register: 9
 Number using O6 output only: 9
 Number using O5 output only: 0
 Number using O5 and O6: 0
Number used exclusively as route-thrus: 106
Number with same-slice register load: 64
Number with same-slice carry load: 11
Number with other load: 31

Zde ukáží, jak vypadá rozmístění propojení vnitřní architektury FPGA. Na Obr. 28 je vidět, že propojení v FPGA je velmi náročné a systém programování pomocí jazyka VHDL značně ulehčuje práci. Toto schéma pomáhá ladit systém pro větší rychlost u složitějších systémů, kdy umístění jednotlivých prvků rozhoduje o rychlosti celého systému.



Obr. 28 Vnitřní propojení FPGA

7 Software

7.1 Popis procesoru openRISC 1200 z pohledu programátora

Procesor obsahuje 32 třicetidvoubitových registrů pro všeobecné použití, nad nimiž lze provádět operace. Registry jsou implementovány jako pole přímo v jádře procesoru, a proto k nim lze přistupovat velmi rychle. Některé mají specifický význam, viz [10].

Procesor dále obsahuje soubor speciálních funkčních registrů (SFR), které poskytují jednak informace o aktuálním stavu procesoru a jeho konfiguraci, a také umožňují ovládat jeho funkce – jako cache, interní časovač, atd. K přístupu k registrům jsou použity speciální instrukce, viz [10]

Podstatnou část je nutné věnovat výjimkám, jejich soupis je uveden v Tab. 6. Jejich obsluhu je třeba řešit na úrovni speciální části programu, která je umístěna na daných adresách a z pravidla psána v jazyce symbolických adres. Zvláště významná je výjimka reset, která je vyvolána při resetování procesoru. Zde je nutné provést základní konfiguraci procesoru a systémů, při čemž je nutné mít na paměti, že dokud nebude provedena alespoň inicializace paměti a stacku, procesor není schopen provést ani skok (resp. Není schopen správně uložit návratovou adresu). Detailní popis výjimek je uveden v dokumentaci [12].

Výjimka	Vektor	Popis
Reset	0x100	Externí reset
Bus error	0x200	Chyba na sběrnici
Data page error	0x300	Chyba virtuální paměti
Ins page error	0x400	Chyba virtuální paměti
Timer	0x500	Přerušení od časovače
Aligment	0x600	Chyba zarovnání dat
Neplatná instrukce	0x700	Neplatná instrukce
Externí přerušení	0x800	Přerušení od PIC
DTLB miss	0x900	Data cache miss
ILTB miss	0xA00	Instruction cache miss
Systém call	0xC00	Systémové volání
Debug	0xD00	Debugger, pro realizaci break pointu
Reserved	0xF00	Nevyužito

Tab. 6 Výjimky procesoru openRISC, převzato z [6]

7.2 sestavení programu pro obsluhu procesoru openRISC

K přeložení zdrojového kódu do spustitelné binární formy je nutné program přeložit a sestavit. K tomu je potřeba použít nástroje z balíku GNU toolchain [] pro openRISC. Balík obsahuje překladač C CGG pro OR32, linker pro sestavení spustitelného souboru, knihovnu jazyka C, a nástroje pro práci s hotovými objekty. Pro překlad je nutné mít další nástroje unixového prostředí, a celý 38 toolchain je rovněž možné využívat pouze pod unixovým prostředím. Pod Windows je možné toolchain využít pod cygwinem, což ale vyžaduje jeho novou kompilaci a přináší jisté problémy.

Překlad programu je řízen pomocí makefile, které je nutné udržovat ručně. Protože část práce byla přebrána a pouze po jednotlivých částech upravena, musel jsem využít již připravené prostředí. Stažení, kompilace, instalace a propojení všech nástrojů není úplně jednoduché. Na doprovodném DVD je přidán kompletní obraz systému (Linux Mint) s nainstalovanými a nakonfigurovanými nástroji tak, aby byl hned k dispozici. DVD obsahuje diskový obraz ve formátu vdi pro vizualizační nástroj Virtualbox.

Mimo běžného kódu v jazyce C jsou potřebné ještě další dvě součásti pro úspěšné sestavení programu, a to speciální zaváděcí program a skript pro linker, který definuje organizaci paměti a umístění jednotlivých částí programu. Zaváděcí program je malý kus kódu, umístěný v přesné oblasti

paměti. Slouží k provedení nejnnutnějších operací pro spuštění procesoru, jako je inicializace stacku, či nastavení kritických SFR. Slouží také k obsluze výjimek. Tato část je obvykle psaná v jazyce symbolických adres, a teprve po jejím ukončení je volána funkce main().

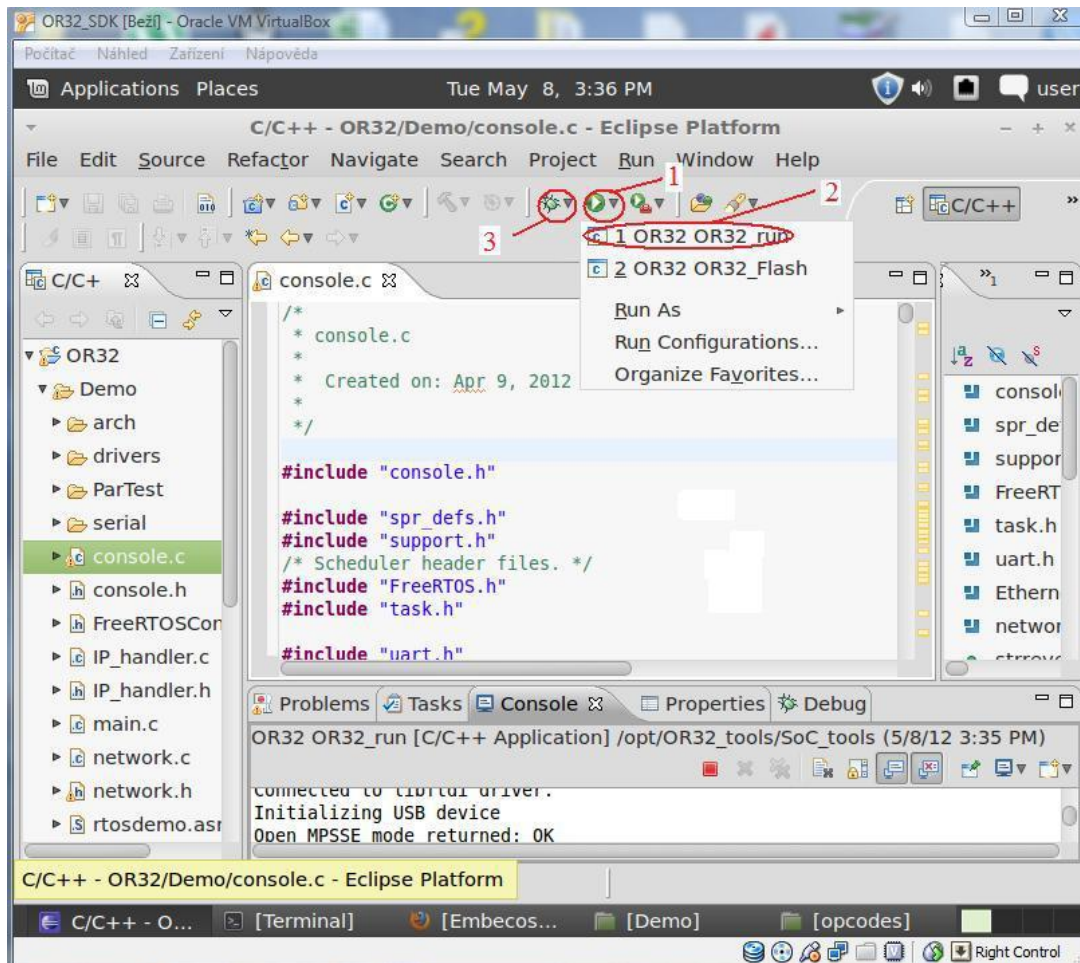
7.3 Vývojové prostředí

Vývojové prostředí muselo být rozloženo na dvě části a to implementace mé části práce do již stávajícího projektu a pak napsání části pro vykreslení obrazu spustitelný pro Windows.

Platforma OR32 není příliš rozšířena a toto prostředí je velmi složité pro přípravu, proto je na doprovodném DVD umístěn nakonfigurovaný a připravený systém, který vše potřebné obsahuje. Diskový obraz systému je na doprovodném DVD uložen ve složce /SDK v komprimovaném stavu. Jedná se o diskový obraz pro volně šiřitelné vizualizační prostředí Virtualbox[.].

Systém byl z části propojen s vývojovým prostředím Eclipse CDT, což umožnilo zjednodušený vývoj softwaru. Bohužel jsem musel tuto část dodržet a moji část kódu, pro posílání dat ze senzoru, implementovat ve stejném prostředí.

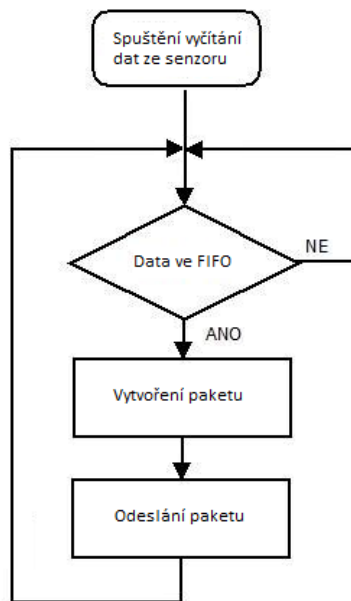
Eclipse musí volat kompilační skript (makefile) pro překlad, ovšem ten je nutné vytvářet a upravovat ručně. Na Obr.29 je otevření projektu v prostředí Eclipse. Je možné ho přímo z prostředí přeložit a spustit (číslo 1 a 2), nahrát do paměti flash, nebo spustit ladění v desce přes JTAG (číslo 3).



Obr. 29 Prostředí Eclipse [7]

7.4 Programy

Jak bylo zmíněno výše, tak první program slouží k posílání dat ze senzoru. Jeho jednoduchá funkce je uvedena na obrázku Obr. 30. Jediné co kontrolují je přítomnost dat ve FIFO paměti.

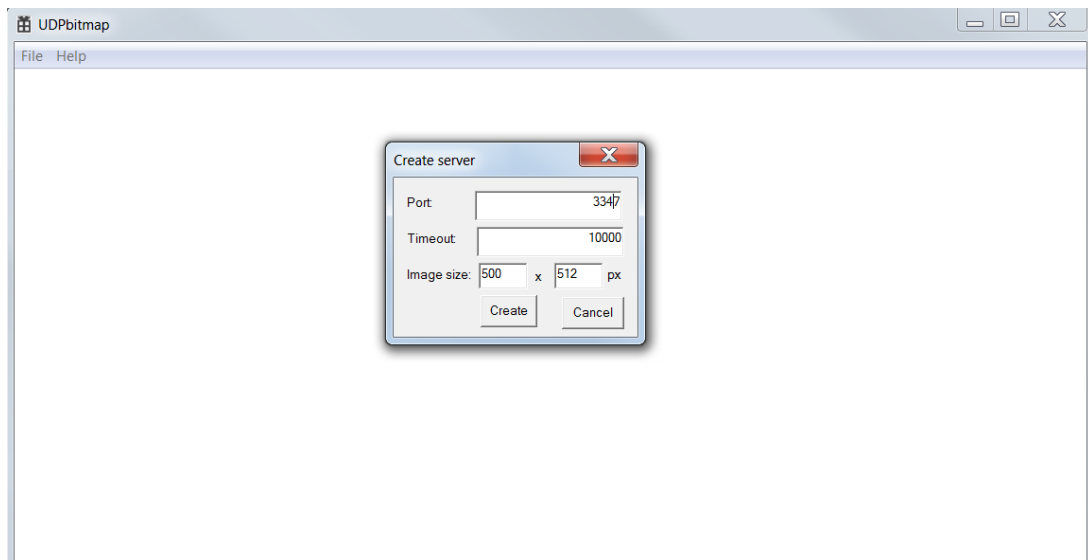


Obr. 30 Vývojový diagram pro odesílání dat

V tomto programu není potřeba příliš velká obsluha, ta je řešena v hardwaru a zde se pouze mohou nastavit jednotlivé registry.

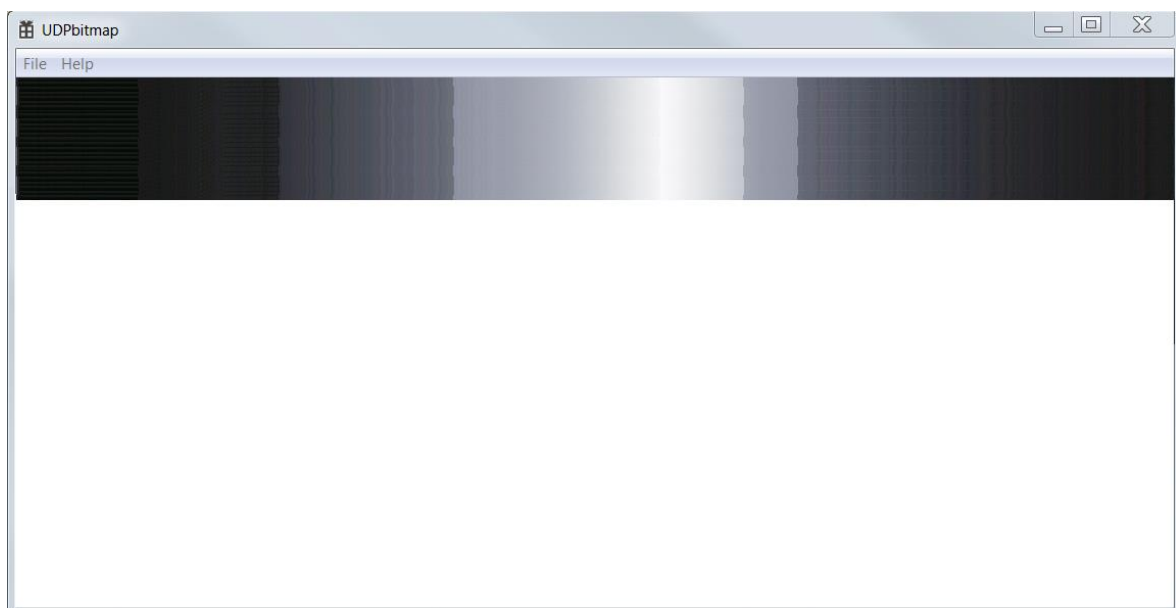
Druhý program řeší příjem dat přes UDP protokol a jejich vykreslení. Program nazvaný UDPbitmap.exe se nachází na doprovodném DVD. Pro správnou funkci programu je potřeba nainstalovat knihovny ze souboru vcredist_x86.exe.

Pro příjem dat je potřeba zadat port, na kterém bude aplikace přijímat data. Můžeme vybrat rozměry obrázku (šířka x výška), který chceme zobrazit. Vykreslení obrázku je do zadaných rozměrů. Pokud přijde méně dat než je šířka řádku, zbytek řádku není přepsán. Pokud přijde více dat, jsou data nad rozměry řádku ignorovány. Předpoklad je, že jeden řádek bude jeden paket, podle toho si program získá délku řádku. Na Obr. 31 je vidět volba portu a velikost vykreslení řádku a na Obr.32 jsou vykreslena data ze senzoru.



Obr. 31 Nastavení portu a velikosti obrázku v programu UDPbitmap

Na následující obrázku si ukážeme vykreslení bílého kusu plastu. Bohužel zobrazení je statické, ale jsou přibližně vidět okraje, kde by se mohl předmět nacházet. Bohužel zvolené optika nedovoluje kvalitnější vykreslení. Problémem může být i ve snímání obrazu nebo ve vykreslovacím problému. Nejsvětlejší místo je osvětleno čelovou svítilnou.



Obr. 32 Vykreslení bílého kusu plastu na hrané hnědého stolu

8 Ekonomické zhodnocení

Provedení ekonomického rozboru je zde značně složité. Vývojový kit od společnosti Atlys a mnou navržená vývojové deska mají části, které by ve výrobě nebyly použity. V kapitole 3.6 je zmínka o tom jak by přibližně vypadal výsledný produkt a co by bylo potřeba v tomto produktu mít. Proto

náklady na vývoj zahrnu do jedné variabilní položky, která bude zohledňovat přibližnou cenu, která je potřeba na různé revize desek, změny součástek, atd.

Počet kusů jsem v kalkulaci ceny omezil na 1000, ve kterých bude rozpočítána cena vývoje. Vzhledem k rozsáhlému použití produktu nebudou do konečné ceny započítávány ceny objektivu, je to z důvodu toho, že na každou aplikaci je potřeba jiný objektiv.

	cena za jednotku	ks	cena
počet kusů v sérii		1000	
příprava filmových podkladů pro výrobu DPS	750	1	750
příprava předloh pro strojní osazování	1560	1	1560
vývoj FPGA	1000	250	250000
vývoj hardware	500	150	75000
Vývoj firmware	800	150	120000
vývojový kit	cca 7000	1	7000
fixní náklady na kus			454,31
výroba DPS	1000	1	1000
osazování desky - strojové SMD	1000	1	1000
osazování desky - THT ruční	15	1	15
FPGA Spartan 6	2917	1	2917
AD převodník AD9235	240	1	240
DDR2 MT47H64M16HR-25E	125	1	125
Senzor ILX555A	360	1	360
filtr MCL SXLP-3	300	1	300
AD8007	40	1	40
TPS7A4700RGWT	57	1	57
IE0509SH	157	1	157
SN74LVC07APW	13	1	13
rezistor 0603	0,5	50	25
kondenzátory keramické 0603	0,5	50	25
kondenzátory keramické 1206	2	20	40
kondenzátory keramické 1210	5	20	100
kondenzátory elektrolytické SMD	3	1	3
TVS 6V	4	1	4
TVS 6,3V	2	1	2
High speed dioda	6	1	6
TVS 3,3V	8	1	8
Induktor 1812	20	8	160
konektory	500	1	500
variabilní náklady na kus			7 097,00

celkové náklady na výrobu a vývoj jedné řádkové kamery			7 551,31
--	--	--	----------

Tab. 7 Rozpočet případné výroby

V Tab. 7 jsou vidět přibližné náklady na výrobu jedné kamery. Nejsem si jistý výslednými náklady. Jak jsem psal výše, je těžké tyto náklady odhadnout a myslím si, že reálné náklady budou 2x až 3x vyšší. Důvod vyšších nákladů spočívá v úplně jiných součástkách, které budou na desce potřeba, než se na desce momentálně nachází. Samozřejmě součástky na desce zůstanou, avšak přibudou součástky kolem FPGA a paměti. Proto jsem v kalkulacích opatrný.

Cena součástek pro výrobu této testovací desky je přibližně 2500,- Kč s DPH. Deska byla vyrobena pouze dvouvrstvou technologií za 650,- Kč.

Po konečném rozboru možností trhu a přibližných cen jsem určil výslednou cenu na 7551,- Kč bez DPH.

9 Závěr

V diplomové práci byl navržen systém skládající se ze tří částí. První část byl návrh jednoduché DPS s příslušným řádkovým snímačem. Výsledná deska je sice větších rozměrů než je potřeba, ale je plně funkční a data ze senzoru a A/D převodníku nejsou na výstupu nijak zkreslena. Je to dáno konstrukcí desky a vysokým potlačením rušivých kmitočtů, které by mohly signál rušit, a mohlo by docházet ke špatné interpretaci dat.

Druhá část se skládá z návrhu SoC na FPGA Spartan 6. V této části byl implementován již stávající design pro komunikaci po ethernetu a usb. Systém byl zvolen z důvodu velké časové náročnosti návrhu a mé malé zkušenosti s návrhem SoC. Bohužel se toto řešení ukázalo jako velice nedostatečné. Ideálním řešením je začít s čistým návrhem a postupovat od vyčítání dat ze senzoru až po komunikaci po UDP. Toto řešení by si však vyžádalo velkou časovou náročnost, protože výsledná implementace je velmi složitá.

Třetí částí, kterou diplomová práce obsahuje, je samotný softwarový návrh. Pro komunikaci byl opět z části použit stávající programová podpora pro procesor openRISC. Toto řešení také není zcela ideální, avšak tato část je provázána s hardwarovou částí FPGA a nebylo jí možné příliš upravovat. Výsledné vykreslení je už otázkou jednoduchého softwaru.

Vzhledem ke konkurenci si myslím, že tento návrh není moc použitelný pro sériovou výrobu. První změna, která by musela nastat, je v použitém snímači. Snímač je konstrukčně zastaralý a jeho

použití pro účely průmyslové kamery nedostatečný, zvláště tam, kde je potřeba rychlé vyčítání. Ideální by bylo použití moderního CMOS snímače pracujícího na 50 MHz.

Při implementaci rychlejšího snímače by muselo dojít k úpravě FPGA, avšak za cenu značné časové úspory. To je právě velká výhoda tohoto návrhu při totožné DPS a pouzdru snímače se dá celé řízení vyčítání změnit během pár dnů práce k jinému senzoru. Pouze přehráním jádra FPGA dosáhneme značné finanční a časové úspory.

Pokud se systém má navrhovat pro sériovou výrobu, byla by potřeba u FPGA a softwaru začít od začátku a snažit se nevyužít již existující části. Velkou výhodou by bylo, aby systém byl přenositelný mezi jádry různých typů FPGA. Poté by byla navržena DPS s přítomností FPGA a ethernetového rozhraní. Jednalo by se o velice elegantní a funkční řešení.

Nejvíce času by bylo potřeba věnovat návrhu optiky, vzhledem k mé malé zkušenosti s tímto návrhem se mi úplně nepodařilo správně zvolit použitý objektiv. Samostatný návrh se nepovedl úplně zrealizovat, protože výsledky s tímto návrhem byly nepoužitelné. Proto zde návrh není uveden.

Závěrem bych zhodnotil, že systém je funkční v omezené míře studentské práce a návrhu pomocí vývojového kitu. Po zkušenostech s návrhem bych již dnes postupoval jinak, protože určité části se tímto způsobem v praxi řešit nedají.

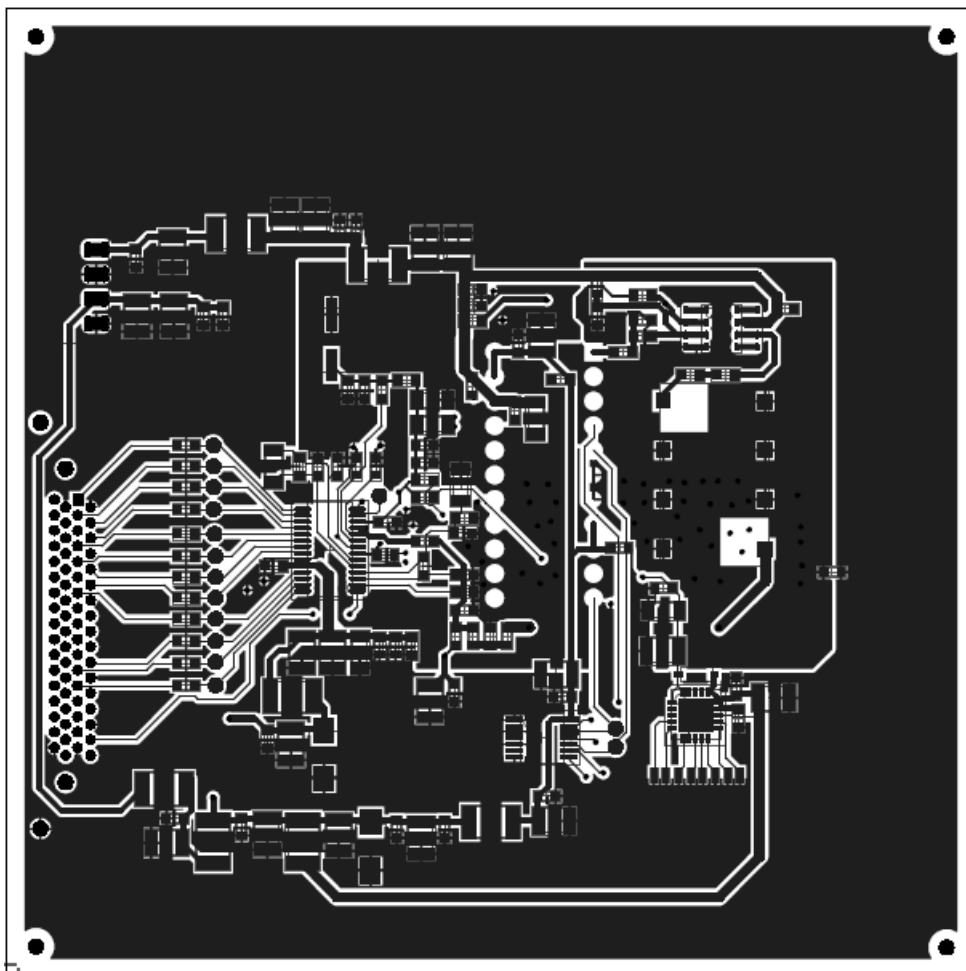
10 Literatura

- [1] Fisher, J.: Optoelektronické senzory a videometrie. 1.vyd. Praha: ČVUT , 2002 [cit. 2014-05-08].. 143 s. ISBN 80-01-02525-X
- [2] *Řádkové kamery - úvod do problematiky* [online]. 2011 [cit. 2014-05-08]. Dostupné z: <http://www.prumyslove-kamery.cz/clanky-a-aktuality/clanky?pg=443>
- [3] Ďaďo, S. – Fisher, J: Optical Sensors. In Master Book on sensors. Prague: CTU Prague a BEN, 2003 [cit. 2014-05-08],. vol. 1, s. 2-1-2-132. ISBN 80-7300-129-2.
- [4] Sony. ILX551A 2048-pixel CCD Linear Sensor.[online]. [cit. 2014-05-08]. Dostupné z: <http://pdf1.alldatasheet.com/datasheetpdf/view/47512/SONY/ILX551A.html>
- [5] ANALOG DEVICES. *AD9235*. 2012. Dostupné z: http://www.analog.com/static/imported-files/data_sheets/AD9235.pdf
- [6] DIGILENT. *Atlys schematic*. 2010. Dostupné z: <http://www.digilentinc.com/Products/Detail.cfm?NavPath=2,400,836&Prod=ATLYS>
- [7] PAVLATA, Petr. *Hardwarový akcelerátor pro BCI aplikace*. Praha, 2011. Diplomová práce. ČVUT.
- [8] **OpenCores project**. OpenCores.org. [Online] <http://opencores.org/>.
- [9] The FreeRTOS Project. [Online] <http://www.freertos.org/>.
- [10] **Damjan, Lampret**. OpenRISC 1200 IP Core Specification. [Online] http://www.da.isy.liu.se/courses/tsea44/OpenRISC/or1200_spec.pdf.
- [11] **OpenCores**. WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP cores. [Online] 2010. http://cdn.opencores.org/downloads/wbspec_b4.pdf.
- [12] XILINX. *LogiCORE IP FIFO Generator v9.3*. Dostupné z: http://www.xilinx.com/support/documentation/ip_documentation/fifo_generator/v9_3/pg057-fifo-generator.pdf
- [13] ŠŤASTNÝ, Jakub. *FPGA prakticky*. 1. vydání. Praha: BEN - technická literatura, 2011. ISBN 978-80-7300-261-9.
- [14] GNU Toolchain . <http://opencores.org>. [Online] Dostupné z: http://opencores.org/openrisc,gnu_toolchain.

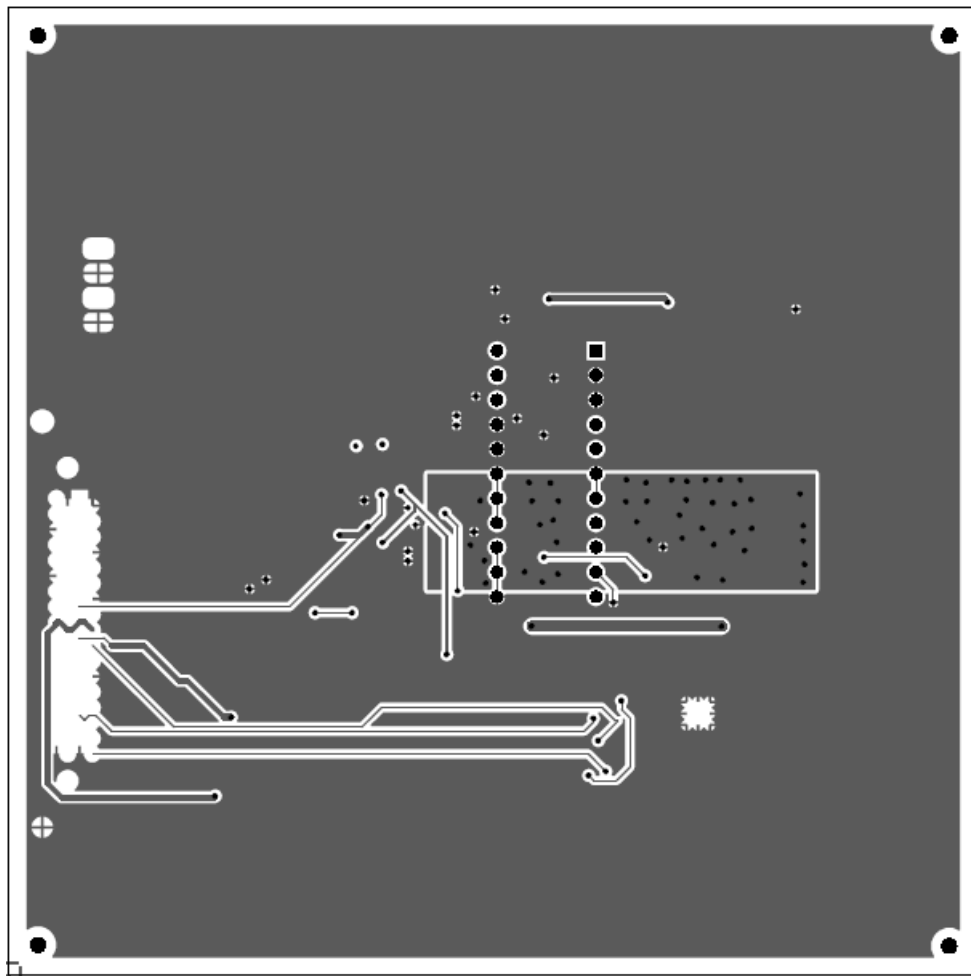
11 Přílohy

11.1 Osazovací a výrobní data

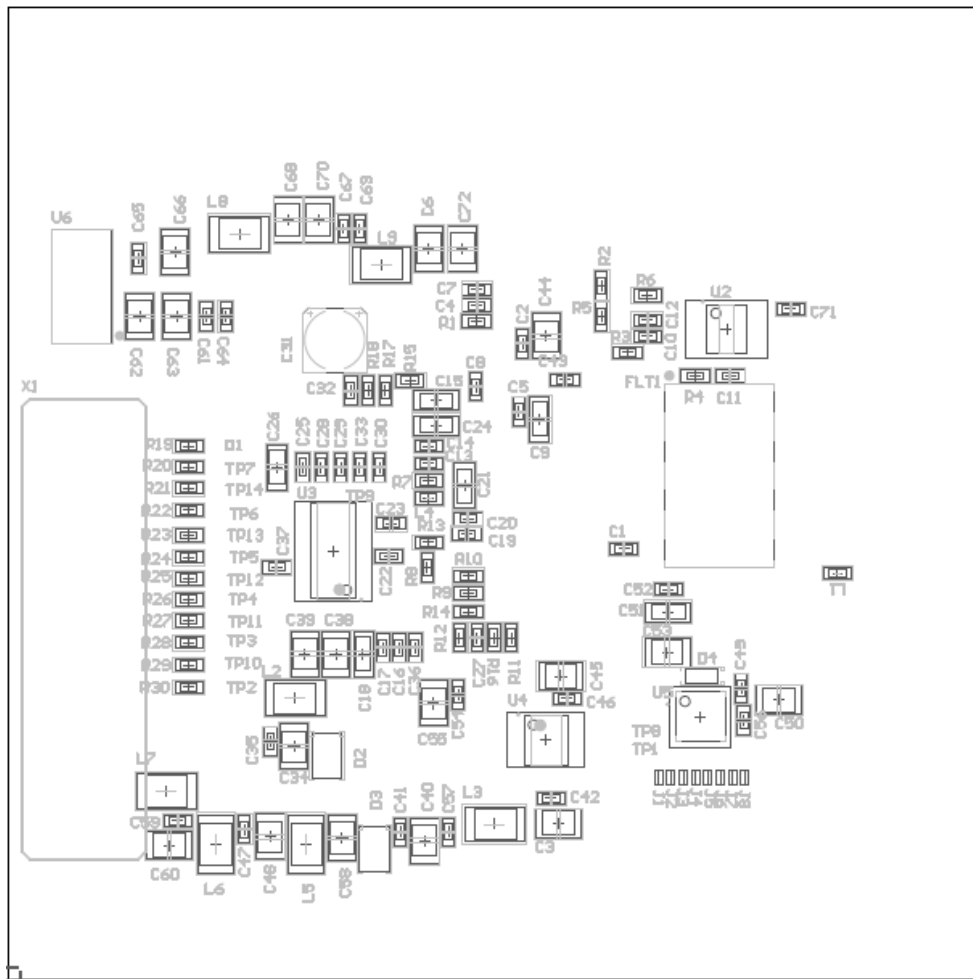
Deska plošných spojů, výrobní data, top, pohled shora, měřítko 1:1



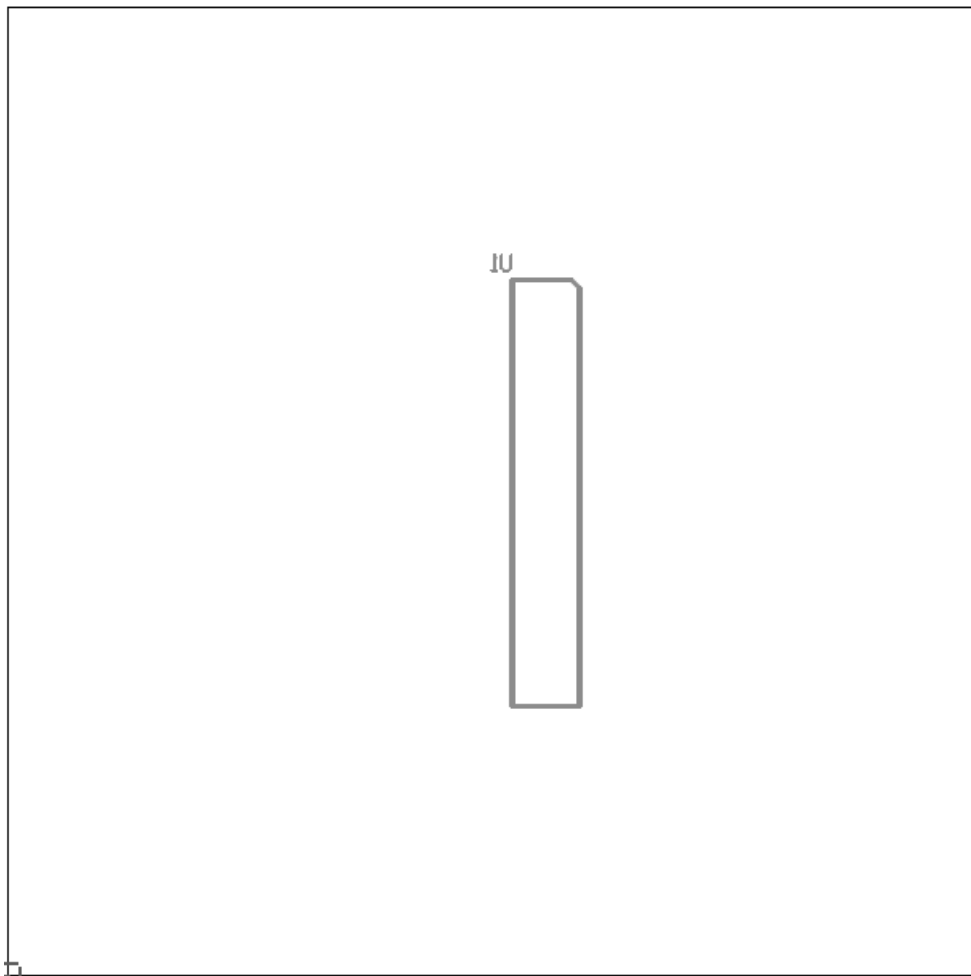
Deska plošných spojů, výrobní data, botton, pohled shora, měřítko 1:1



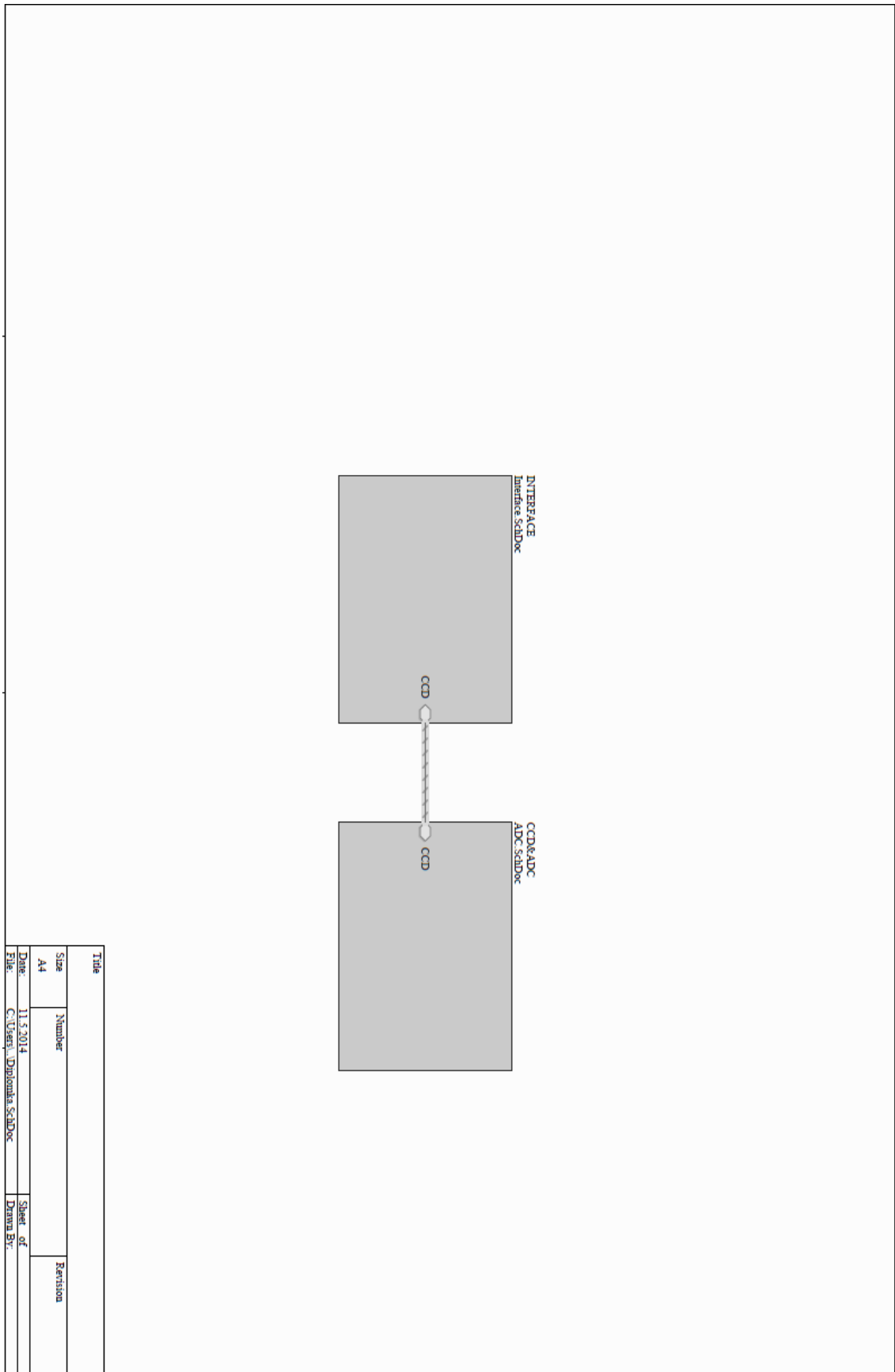
Deska plošných spojů, osazovací data, assemblytop, pohled shora, měřítko 1:1

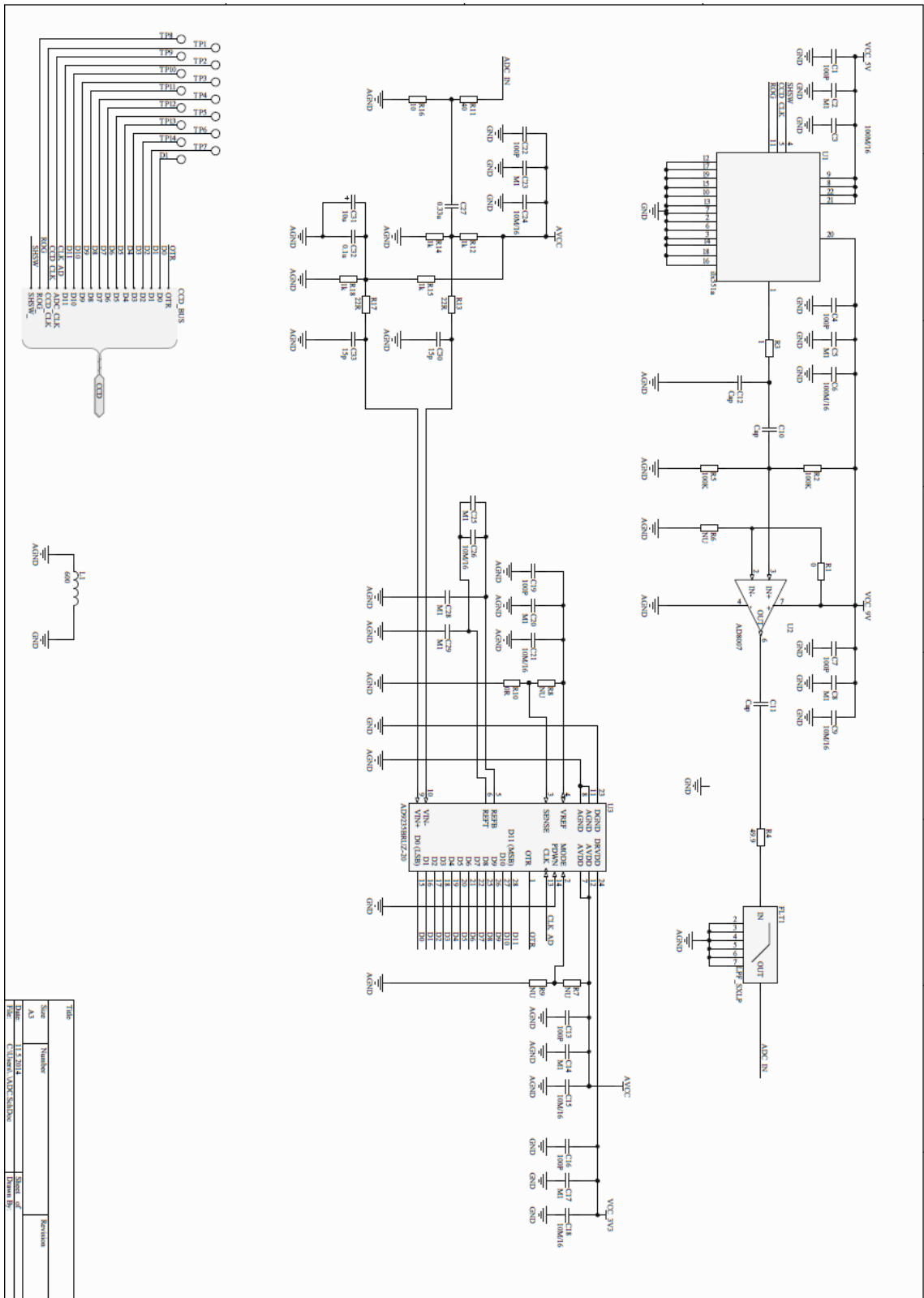


Deska plošných spojů, osazovací data, assemblybotton, pohled shora, měřítko 1:1

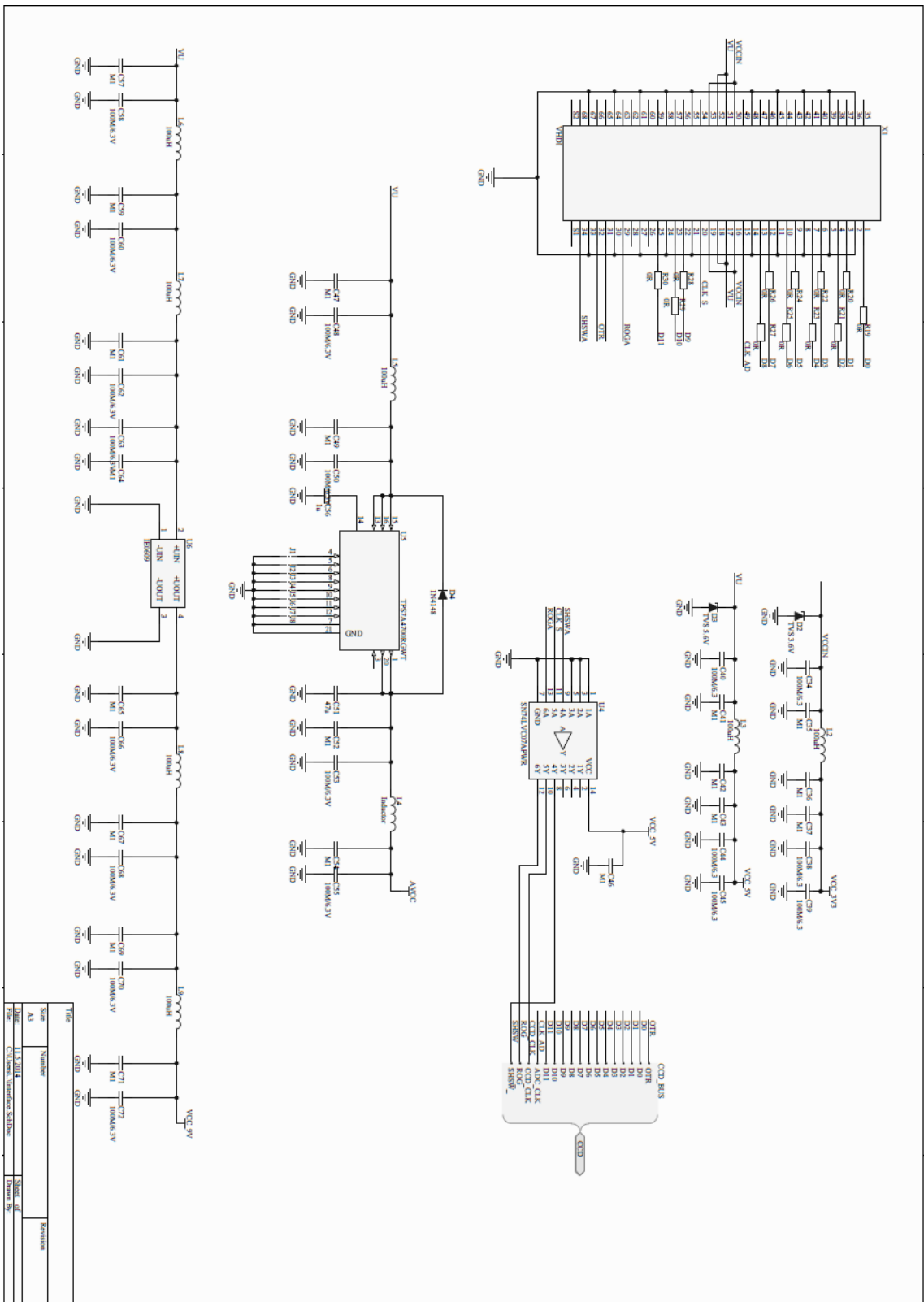


11.2 Kompletní obvodové schéma





Title		Revision	
Size	Number	Sheet No.	Revision
A1		1	1
DATE	FILE	DESIGNER	CHECKER
2023.11.14	AD9258BRLZ20	YANG	YANG



11.3 Obsah přiloženého DVD

Adresář /pdf – dokumentace k použitým součástkám

Adresář /sw – Linuxové SDK systém pro vývoj aplikací, sw pro vykreslení dat

Adresář /hw - návrh hardware, v podobě pro Altium Designer 14

Adresář /fw – HDL zdrojové kódy

Adresář /doc - tato práce ve formátu pdf