

PROHLÁŠENÍ

Předkládám tímto k posouzení a obhajobě bakalářskou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne

.....
vlastnoruční podpis

ANOTACE

Cílem této práce bylo vytvořit Camera Link[®] Frame Grabber jako FPGA bitstream a zobrazovat výstup kamery Basler ace acA2000-340kmNIR na VGA monitoru. V prvních kapitolách jsou představeny použité technologie. Poslední kapitoly se zabývají implementací cíle práce. Výsledný Frame Grabber je možné použít pro čtení obrazových dat z rozhraní Camera Link[®].

Klíčová slova

Camera Link[®], FPGA, VHDL, Avalon[®], Basler

ABSTRACT

The goal of this work was to create Camera Link[®] Frame Grabber as an FPGA bitstream and show camera output of Basler ace acA2000-340kmNIR on VGA screen. Used technologies are introduced in first chapters. Final chapters are about implementation of the work's goal. Final Frame Grabber can be used for reading image data from Camera Link[®] interface.

Keywords

Camera Link[®], FPGA, VHDL, Avalon[®], Basler

Obsah

1	Úvod	5
1.1	Cíl práce	5
1.2	Členění práce	5
2	Camera Link®	6
2.1	Úvod	6
2.2	Channel Link	6
2.3	Přenos pixelů	8
2.4	Pomocné signály	8
2.5	Kabel a konektory	9
2.6	Konfigurace zapojení	9
2.7	Ostatní standardy	11
2.7.1	GigE Vision®	11
2.7.2	USB3 Vision®	12
3	Kamera	14
3.1	Nastavení	14
3.2	Tap geometrie	15
4	Vývojové nástroje	17
4.1	FPGA	17
4.2	VHDL	18
4.3	Avalon®	18
4.4	Vývojové prostředí Quartus	19
4.5	Vývojová deska Altera DE2-115	20
4.6	CLR-HSMC	21
5	Implementace	23
5.1	Frame Grabber	24
5.1.1	Avalon® rozhraní	24
5.1.2	Synchronizace	26
5.2	Řídící program	28
5.3	SDRAM buffer	29
5.4	Konfigurace kamery	29

OBSAH

5.5	Zobrazení	30
6	Závěr	31
6.0.1	Další cíle	31
	Seznam použitých zkratek	32
	Seznam použité literatury	33

1. Úvod

Bakalářská práce se zabývá získáváním obrazu z kamery s Camera Link[®] rozhraním pomocí FPGA bitstream návrhu. Práce bude sloužit jako základ pro vytvoření složitějšího řešení strojového zpracování obrazu, kde se získaný obraz bude dále zpracovávat sekvenčním programem.

1.1 Cíl práce

Práce bude sloužit jako základ pro aplikace rychlého zpracování obrazu. Díky vysokým limitům přenosu na rozhraní Camera Link[®] jsou kamery s tímto rozhraním vhodné pro aplikace rychlého zpracování obrazu a strojového vidění. Problémem rozhraní Camera Link[®] jsou další náklady na Frame Grabber, což je prvek mezi tradičním sekvenčním počítačem a Camera Link[®] rozhraním, který poskytuje data uživatelskému programu jako stream pixelů nebo v podobě obrazových rámců v paměti RAM. Hlavní cíle této práce jsou

- vytvořit Camera Link[®] Frame Grabber v podobě FPGA bitstreamu na platformě Altera Cyclone IV E,
- ukládat obrazové rámce z kamery do paměti DRAM a
- zobrazovat obraz z kamery na VGA monitoru.

K dosažení těchto cílů byly použity vývojové nástroje a standardy společnosti Altera: vývojová deska Altera DE2-115, rozšíření CLR-HSMC, vývojové prostředí Quartus a Qsys, a návrh pomocí komponent Avalon[®]. Frame Grabber byl implementován jako Avalon[®] Stream komponenta, díky čemuž je možné provádět další zpracování obrazu v FPGA bitstreamu s minimální režií.

1.2 Členění práce

V úvodu je popsán cíl práce. V druhé kapitole je popsáno rozhraní Camera Link[®]. Ve třetí kapitole je stručně představena kamera Basler ace acA2000-340kmNIR. Ve čtvrté kapitole je popis použitých technologií při vývoji. V páté kapitole je popsána implementace Frame Grabberu. V závěru jsou shrnuté výsledky práce.

2. Camera Link[®]

Většina výrobců vysokorychlostních kamer pro strojové vidění využívá rozhraní spravované asociací AIA[5] (Automated Imaging Association). AIA vznikla v roce 1984 a její součástí je více než 330 společností zabývajících se elektronikou a softwarem pro zpracování obrazu[6]. Mezi hlavní standardy kamerových rozhraní spravovaných společnostmi AIA patří GigE Vision[®], Camera Link[®] a USB3 Vision[®].

Hlavní náplň této kapitoly bude popis rozhraní Camera Link[®], které bylo použito v této práci. Na konci kapitoly budou stručně popsány standardy GigE Vision[®] a USB3 Vision[®] pro porovnání.

2.1 Úvod

Camera Link[®][7] je standard, založený v roce 2000, pro přenos obrazu z vysokorychlostních kamer přes rozhraní Channel Link. Standard definuje 3 konfigurace zapojení, elektrické vlastnosti signálu, protokol pro přenos obrazových dat, mechanické vlastnosti zapojení a obecné sériové rozhraní pro komunikaci s kamerou. Standard definuje nejrychlejší rozhraní pro přenos dat ze všech ostatních standardů AIA, ale vyžaduje použití tzv. *Frame Grabberu* pro poskytnutí obrazových dat aplikační úrovni. Vytvořit Frame Grabber byl cíl této práce.

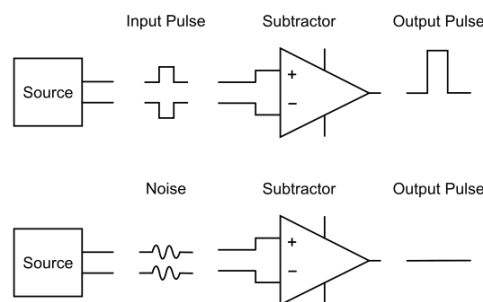


Obrázek 2.1: Logo standardu Camera Link[®]

2.2 Channel Link

Channel Link je rozhraní vytvořené společností National Semiconductor pro rychlý a efektivní přenos dat. Teoretická maximální přenosová rychlost je 6.4 Gbit/s[27]. Channel Link pro přenos dat používá LVDS[28] (Low Voltage Differential Signaling) spojení.

LVDS je standard (ANSI/TIA/EIA-644, 1996) pro sériovou komunikaci definující elektrické vlastnosti signálu. LVDS umožňuje dosáhnout vysokých rychlostí s použitím levné měděné kroucené dvojlinky. Díky tomu našel standard uplatnění ve spotřební elektronice, například pro přenos obrazu do displejů v přenosných počítačích (FPD-Link).

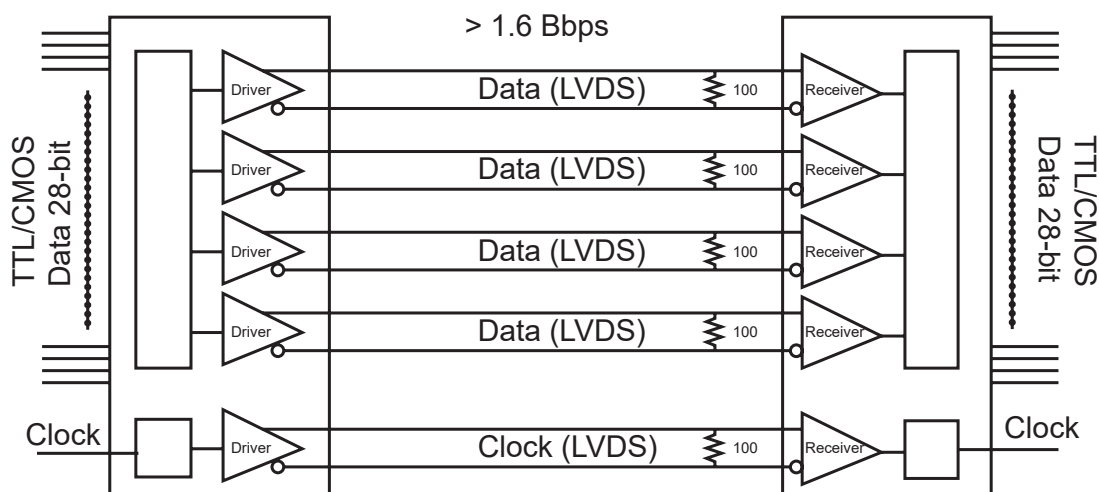


Obrázek 2.2: Odolnost diferenciálního signálu vůči šumu

LVDS používá diferenciální signál. Diferenciální signál zvyšuje odolnost vůči šumu a tak umožňuje pracovat na nižším napětí. To je znázorněno na obrázcích 2.2. LVDS dokáže pracovat s napětím 350 mV. Podle specifikace je signál odolný vůči šumu s amplitudou ± 1 mV[1]. Nízké napětí také snižuje čas náběžné a sestupné hrany čímž se zvyšuje možná přenosová rychlost. Teoretické maximum přenosové rychlosti LVDS je 1.923 Gbit/s[1]. Další výhodou diferenciálního signálu je odolnost vůči rozdílu potenciálů země na obou koncích.

LVDS přenáší data sériově, tedy bit bo bitu. Pro zvýšení objemu přenesených dat je možné použít více LVDS spojení a přenášet více bitů najednou, což je výhodné pro přenos celého pixelu nebo více pixelů v jednom hodinovém cyklu. Více spojení ale snižuje energetickou účinnost a cenu za metr kabelu.

Channel Link definuje 28 bitů širokou sběrnici a jeden port pro přenos hodin. Fyzicky pro přenos definuje 5 LVDS spojení, z toho jedno pro přenos hodin a zbylé 4 pro přenos dat (viz obrázek 2.3). Součástí čipu pro převod vstupního digitálního signálu na LVDS je fázový závěs, jehož vstup je hodinový signál (v tomto případě z kamery) a jehož výstup je hodinový signál na sedminásobné frekvenci. 28 vstupních datových bitů se rozdělí na části po 7 bitech, které se na této rychlejší frekvenci přenáší sériově po LVDS datových spojeních.



Obrázek 2.3: Channel Link schéma

Pro přenos dat je potřeba tzv. serializér, tj. čip, který převede vstupní digitální signál na Channel Link LVDS signál, a tzv. deserializér, tj. čip, který převede Channel Link LVDS signál na digitální signál. Frame Grabber používá deserializéry a kamera serializéry. Počet použitých Channel Link serializérů a deserializérů na obou stranách závisí na požadované konfiguraci Camera Link®.

Vlastnosti digitálního signálu na vstupu a výstupu jsou relevantní jen pro použité čipy, které s nimi musí být kompatibilní. Např. CLR-HSMC rozhraní k desce DE-115 podporuje konverzi z Channel Link signálů na 2.5 V a 3.3 V výstupní digitální signál.

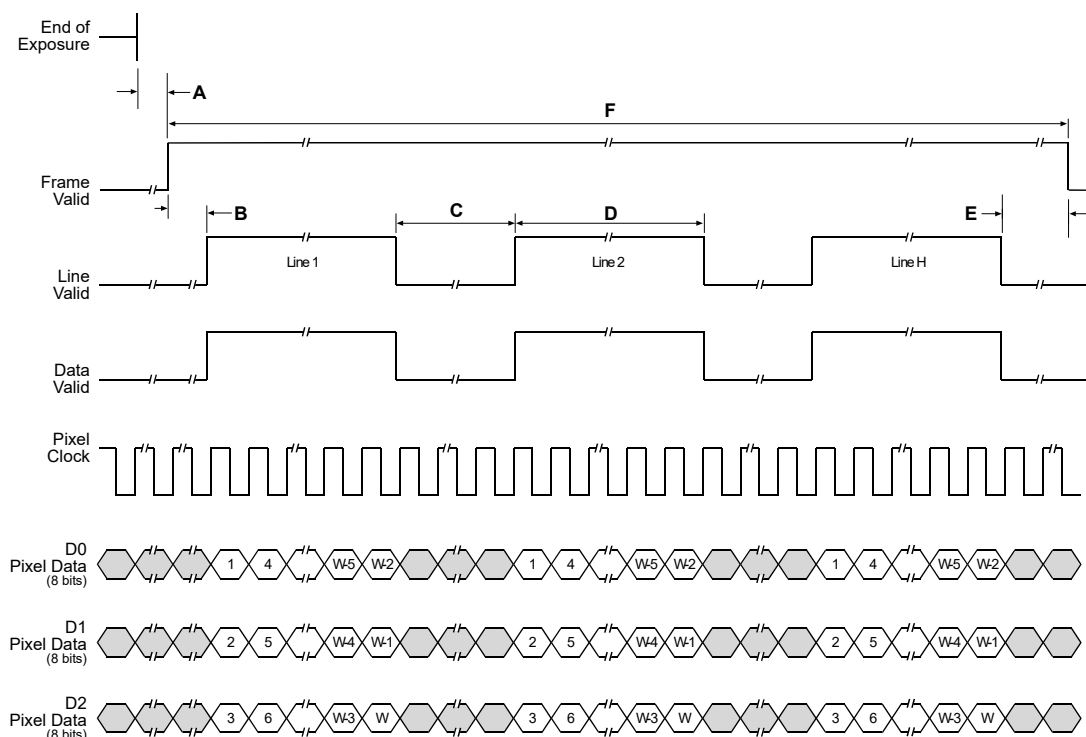
Camera Link® specifikuje použití Channel Link čipů výrobce National Semiconductor. Pár čipů DS90CR287[29] (serializér) a DS90CR288A[30] (deserializér) dokáže přenášet data paralelně na frekvencích až do 85 MHz. Pro LVDS vysílače a přijmače jsou doporučované čipy DS90LV047A[31] a DS90LV048A[32].

2.3 Přenos pixelů

Při barevné hloubce 8 bit je možné přes jeden Channel Link čip přenést tři pixely. Zbylé 3 ze 4 datových digitálních vstupů/výstupů slouží pro synchronizační signály, které pro každý hodinový cyklus signalizují stav přenosu obrazového rámce. Jeden datový vstup/výstup je nevyužitý.

- **FVAL** – Frame Valid
Indikuje, že je aktivní přenos obrazového rámce. Frame Grabber se může synchronizovat na náběžnou hranu toho signálu.
- **LVAL** – Line Valid
Indikuje, že je aktivní přenos pixelů z jedné řádky. Signál může sloužit pro inkrementaci počtu zpracovaných řádek ve Frame Grabberu.
- **DVAL** – Data Valid
Indikuje, že data na výstupních digitálních signálech jsou validní pixely. Tento signál dává kameře možnost kdykoliv pozastavit přenos.

Časový průběh synchronizačních signálů je znázorněn na obrázku 2.4.



Obrázek 2.4: Synchronizační signály

2.4 Pomocné signály

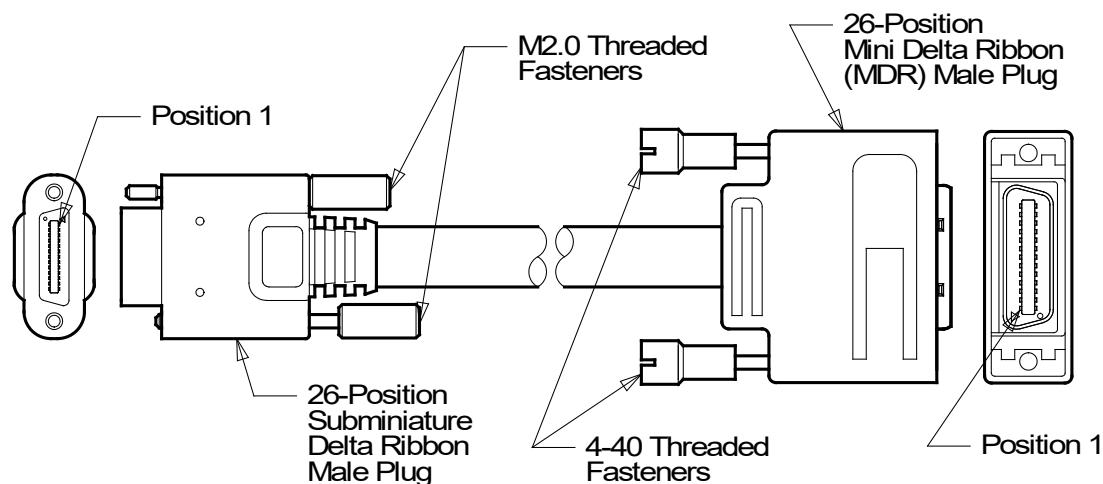
Kromě Channel Link signálů pro přenos obrazových dat Camera Link® specifikuje dalších 6 LVDS párů.

Signály CC1 až CC4 jsou obecné LVDS signály mezi Frame Grabberem a kamerou. Z pohledu kamery mohou sloužit jako vstup nebo výstup. Funkce těchto signálů není definovaná. Je na výrobci kamer, jak budou tyto signály interpretovat. Např. kamera Basler ace acA2000-340kmNIR používá signál CC1 jako externí Frame Trigger Start. Tedy Frame Grabber může tímto signálem libovolně pořizovat jednotlivé obrazové rámce.

Další dva signály slouží jako sériová linka pro obousměrnou asynchronní komunikaci s kamerou. Tyto signály jsou SerTFG (Serial to Frame Grabber) a SerTC (Serial to Camera). Camera Link® doporučuje výrobcům kamer, aby podporovali minimální rychlost 9600 baud/s. Specifikace obsahuje jen API na čtení a zápis na sériovou linku. Protokol na této lince závisí na výrobci kamery. Např. Basler používá vlastní binární protokol, jehož specifikace je volně dostupná[11].

2.5 Kabel a konektory

Camera Link® specifikuje použití kabelů s konektory MDR (Mini Delta Ribbon) a SDR (Shrunk Delta Ribbon) od výrobce 3M. Tento kabel vede 26 vodičů. Ve specifikaci Camera Link® verze 1.1 se používá 22 vodičů pro 11 diferenciálních signálů. Zbylé 4 vodiče jsou nevyužité. Ve specifikaci verze 2.0 přibyla možnost vedení napájení ve dvou párech skrz zbylé 4 vodiče. Tato verze se nazývá PoCL (Power Over Camera Link). Konektory jsou na obrázku 2.5. Konektor MDR se používá pro připojení Frame Grabberu. Menší konektor SDR vede do kamery. Camera Link® specifikace verze 1.1 specifikuje maximální délku kabelu 10 metrů.



Obrázek 2.5: MDR a SDR konektory

2.6 Konfigurace zapojení

Standard ve verzi 1.1 definuje definuje 3 konfigurace zapojení a objemu přenesených dat v jednom hodinovém cyklu. Tyto konfigurace jsou

- Base – 1 konektor, 24 datových bitů

- Medium – 2 konektory, 48 datových bitů
- Full – 2 konektory, 64 datových bitů

Standard verze 2.0 definuje další dvě konfigurace zapojení[1]

- Lite – 1 konektor, 10 datových bitů
- 80 bit – 2 konektory, 80 datových bitů

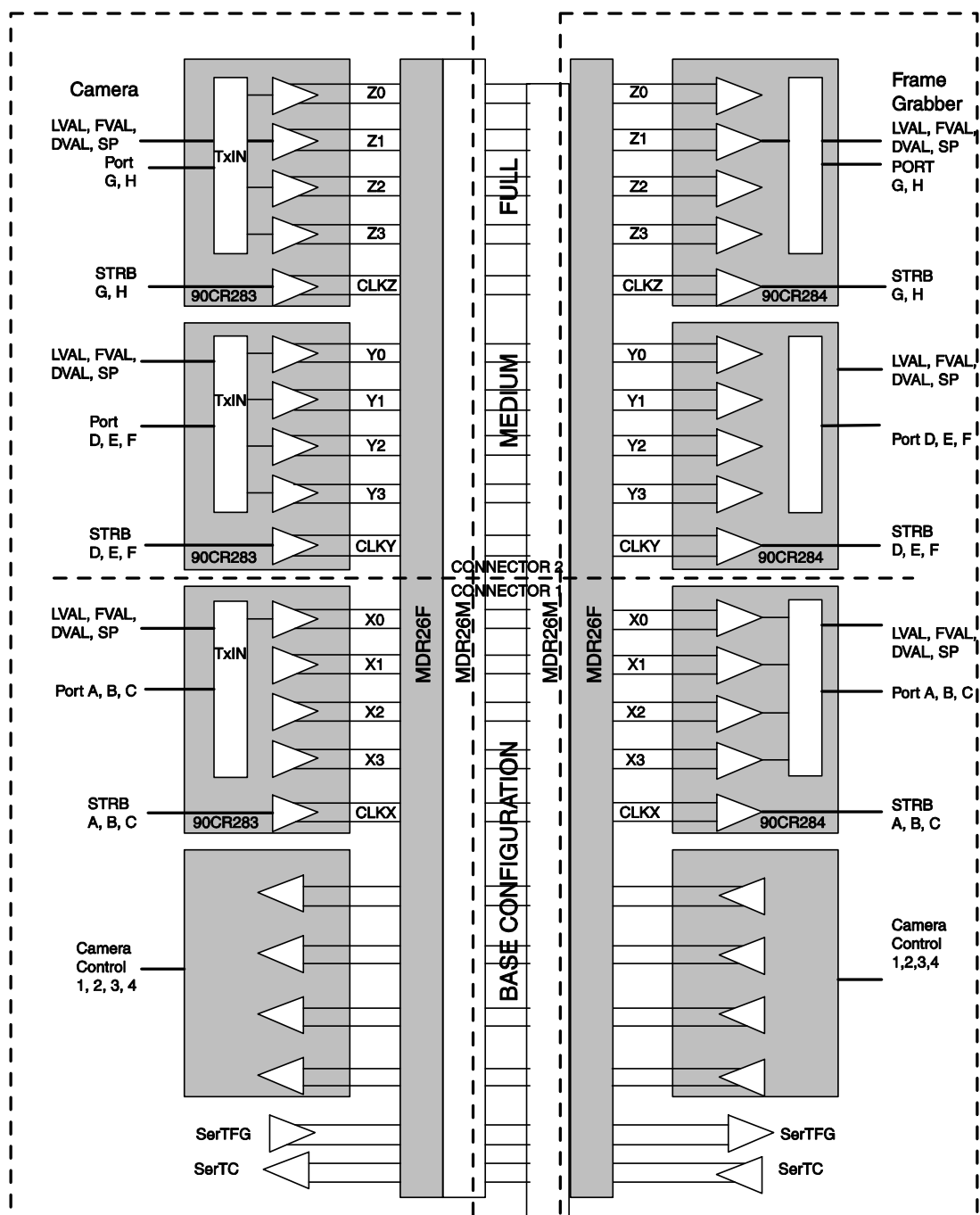
Jednotlivé konfigurace se liší také tím, kolik je potřeba Channel Link čipů (serializérů a deserializérů) na obou stranách. Základní konfigurace jsou znázorněny na obrázku 2.6. Ve všech konfiguracích je jen jedna sada pomocných signálů, které jsou přenášeny přes první kabel.

Při barevné hloubce 8 bitů je možné každý Channel Link kanál rozdělit na 3 porty, z nichž každý reprezentuje jeden pixel. V Base konfiguraci je jen jeden Channel Link kanál a je možné přenášet jen 3 osmibitové pixely na jeden hodinový cyklus. Base konfigurace plně využívá všech 11 diferenciálních párů v propojovacím kabelu (5 pro Channel Link, 6 pomocných signálů) a pro přenos více pixelů je potřeba druhý kabel pro další jeden nebo dva Channel Link kanály. Konfigurace Medium a Full se od sebe liší jen tím, že Medium používá jeden přídatný Channel Link kanál, zatímco Full používá dva. Výběr mezi Medium a Full tedy závisí na požadavcích na datový tok, na výrobní náklady na hardwarové zpracování Frame Grabberu (počet linek a Channel Link čipů na plošném spoji) a na energetické náklady na jeho provoz.

Jednotlivé konfigurace a datové toky jsou srovnány v tabulce 2.1. Maximální operační frekvence Channel Link čipů je 85 MHz. Vypočtený datový tok je násobek této maximální frekvence s počtem 8 bitových pixelů na jeden přenos.

Konfigurace	Počet pixelů v jednom přenosu	Maximální přenosová rychlost
Base	3	225 MB/s
Medium	6	510 MB/s
Full	8	680 MB/s

Tabulka 2.1: Porovnání základních Camera Link® konfigurací



Obrázek 2.6: Konfigurace zapojení Camera Link®

2.7 Ostatní standardy

2.7.1 GigE Vision®

GigE Vision®[12] je standard, založený v roce 2006, pro přenos obrazu z vysokorychlostních kamer přes síťové rozhraní Gigabit Ethernet[13][14]. Standard využívá rodinu protokolů TCP/IP[15] s transportním protokolem UDP[16]. Aplikační protokol pro přenos obrazu je definován standardem. Díky tomu je možné provozovat rozsáhlou síť kamer přes velké vzdálenosti s využitím standardních síťových prvků a existující infrastruktury.

Hlavní části standardu jsou[18]

- GigE Vision® Control Protocol (GVCP)
Protokol pro nastavování a řízení kamery.
- GigE Vision® Stream Protocol (GVSP)
Protokol pro přenos obrazu z kamery.
- GigE Vision® Device Discovery Mechanism
Automatická síťová konfigurace kamery. Obsahuje požadavek na DHCP[17] klientskou část v kameře.
- GenICam popis
Popis kamery ve standardním GenICam XML formátu.



Obrázek 2.7: Logo standardu GigE Vision®

Díky GigE Vision® Device Discovery Mechanism je možné po připojení kamery provádět veškerou správu vzdáleně. Kamery s GigE Vision® rozhraním jsou tedy vhodné pro nasazení v prostředí kde je potřeba centrální shromažďování obrazu ze vzdáleností delších jak 10 metrů a tam kde je rozmístění a počet kamer variabilní.

Protože GigE Vision® využívá UDP protokol, je možné veškeré zpracování obrazu provádět v uživatelském programu na běžném počítači s Ethernet rozhraním a s operačním systémem, který podporuje TCP/IP rodinu protokolů, např. Windows® nebo operační systém s jádrem Linux® (např. Debian). Pro uživatelský program potom stačí už jen rozhraní k transportním protokolům TCP/IP, např. Berkley sockets[19] nebo Winsock[20].

Přenosová rychlost kamer s GigE Vision® rozhraním je limitovaná maximální přenosovou rychlostí prvků v síti (routery, switche, síťová karta přijmače). Teoretické maximum prvků podporující standard Gigabit Ethernet je 125 MB/s. Praktické maximum po odečtení režie směrování dat v síti a softwarového zpracování dat ze sítě je 100 MB/s[21].

2.7.2 USB3 Vision®

USB3 Vision®[22][23] je standard, založený v roce 2013, pro přenos obrazu z vysokorychlostních kamer přes rozhraní USB 3.0 (SuperSpeed USB). Standard využívá USB *bulk transfer*[4] pro bezztrátový přenos dat.

USB rozhraní má 3 části[23]

- Control Transport Layer
Rozhraní pro nastavování a řízení kamery.
- Event Transport Layer
Rozhraní pro hlášení událostí z kamery.
- Stream Transport Layer
Rozhraní pro přenos obrazových dat z kamery.



Obrázek 2.8: Logo standardu USB3 Vision®

Rozhraní USB 3.0 je dnes běžnou součástí spotřebních počítačů a výhodou kamer s rozhraním USB3 Vision® je tedy jejich kompatibilita. Stejně jako u GigE Vision® je veškeré

zpracování obrazu monžé provádát v uživatelském programu v operačním systému, který poskytuje rozhraní pro USB *bulk transfer*. Toto rozhraní je například knihovna `libusb`[24] v operačních systémech s jádrem Linux[®].

Standard USB 3.0 explicitně nelimituje déku kabelu, ale maximální praktická délka pasivního kabelu je 3 metry[4]. Na trhu jsou dostupné i aktivní kabely s dodatečným napájením s délkou 20 metrů[25].

Teoretické maximum přenosové rychlosti USB 3.0 je 5 Gbit/s (625 MB/s)[26]. Praktická přenosová rychlost USB3 Vision[®] je 400 MB/s[21].

3. Kamera

V této práci byla použita kamera Basler ace acA2000-340kmNIR[10]. Kamera je na obrázku 3.1. Základní parametry jsou v tabulce 3.1.



Obrázek 3.1: Kamera Basler ace acA2000-340kmNIR

Rozlišení	2048 x 1088
Sensor	CMOSIS CMV2000-2E12M
Velikost pixelu	5.5 μm x 5.5 μm
Maximální rychlost	340 fps
Barevná hloubka sensoru	10 nebo 12 bitů
Frekvence přenosu (Camera Link®)	32.5/48/65/82 MHz

Tabulka 3.1: Parametry kamery Basler ace acA2000-340kmNIR

Kamera podporuje všechny základní konfigurace Camera Link®. Na obrázku 3.1 jsou vidět zdířky pro SDR konektory. Spodní zdířka je pro 12 V DC napájení. NIR v názvu značí *Near Infrared*. Jedná se o kameru s E12 variantou CMOSIS CMV2000 senzoru, který je citlivý na záření s infračervenou vlnovou délkou.

3.1 Nastavení

Pro čtení a konfiguraci kamery Basler využívá protokol přes sériovou linku Camera Link® pro čtení a zápis registrů. Tento protokol vč. detailního popisu registrů kamery jsou volně dostupné[11]. Zápisem do odpovídajících registrů je možné nastavit například rychlost hodin kamery, barevnou hloubku senzoru, tap geometrii, šířku a výšku snímků, apod.

Nastavení pro tuto práci

V této práci byla kamera použita s nastavením podle tabulky 3.2 (výsledná implementace je ale nezávislá na nastavení kamery).

Popis nastavení	Hodnota nastavení
Rozlišení	800x600
	AOI (Area of Interest), obdelník s přenášenými pixely vycentrováný na střed senzoru.
Camera Link® frekvence	82 MHz
Tap geometrie	1X4-1Y
Barevná hloubka pixelů	8 bitů
Expoziční čas	1 ms

Tabulka 3.2: Nastavení kamery

3.2 Tap geometrie

Tap geometrie je nastavení kamery, které určuje jak budou přenášeny pixely ze senzoru. Tap geometrie se označuje řetězcem ve formátu `<RegionX>X(<TapX>)-<RegionY>Y`. V jednom přenosu se přene jeden nebo více tapů, které odpovídají pixelům podle zvolené tap geometrie. Popis formátu řetězce je v tabulce 3.3.

RegionX	Počet regionů v horizontální dimenzi. Např. 2X znamená, že horizontální dimenze senzoru bude rozdělena na dvě části. Má-li senzor pixelovou šířku W , v prvním tapu se budou přenášet pixely od sloupce 1 do sloupce $W/2$ a v druhém tapu pixely od sloupce $W/2 + 1$ do sloupce W .
TapX	Počet tapů v horizontální dimenzi přenesených v jednom přenosu pro každý region. Např. 1X2 znamená, že v jednom přenosu budou 2 horizontální pixely. Má-li senzor pixelovou šířku W , jedna řádka kamery se přene ve $W/2$ přenosech. Není-li uveden parametr TapX, jeho implicitní hodnota je 1.
RegionY	Počet regionů ve vertikální dimenzi. Např. 1X-2Y znamená, že vertikální dimenze senzoru bude rozdělena na dvě části. Má-li senzor pixelovou výšku H , v prvním tapu se budou přenášet pixely ve všech sloupcích od řádku 1 do řádku $H/2$ a v druhém tapu se budou přenášet pixely ve všech sloupcích od řádku $H/2 + 1$ do řádku H .

Tabulka 3.3: Formát tap geometrie

Počet tapů v jednom přenosu je tedy roven parametru TapX. Kamera Basler ace acA2000-340kmNIR podporuje tap geometrie 1X2-1Y, 1X3-1Y, 1X4-1Y, 1X6-1Y, 1X8-1Y a 1X10-1Y. Horizontální pixely v jednom přenosu nemohou přesahovat na další řádku, tedy šířka přenášeného obrazu musí být dělitelná počtem tapů. Konfigurace tap geometrie závisí na konfiguraci Camera Link®. Např. pro tap geometrii 1X4-1Y pixelů je nutná Medium Camera Link® konfigurace. Na obrázku 3.2 je znázorněn přenos pixelů v tap geometrii 1X4-1Y.



Obrázek 3.2: Tap geometrie 1X4-1Y

4. Vývojové nástroje

4.1 FPGA

FPGA (Field-Programmable Gate Array) je čip navržený pro snadnou rekonfiguraci po výrobě. FPGA čipy obsahují bloky implementující logické funkce, registry a konfigurovatelné propojení mezi nimi díky čemuž může FPGA čip provádět libovolnou logickou kombinační a sekvenční funkci.

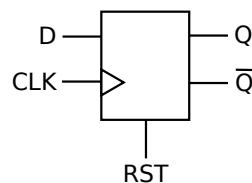
Logické bloky v FPGA čipu jsou obvykle slovníky LUT bloky (Look-Up Table) s určitým počtem vstupů a jedním výstupem. Čipy Altera Cyclone IV E mají 4 vstupové LUT bloky a pomocí nich je možné implementovat jakoukoliv logickou funkci o 4 proměnných. Například kombinační obvod sestávající z bloků AND, NOR a NOT implementující 3 parametrou logickou funkci

$$(A \text{ AND } B) \text{ NOR } (\text{NOT } C)$$

by měl následující konfiguraci jednoho LUT bloku (poslední vstup je nevyužitý, tzv. Don't Care bit)

A	B	C	Výstup
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Další bloky v FPGA čipu jsou registry, jednobitové paměti, které jsou implementovány hodinami řízenými klopnými obvody D. Klopný obvod D má vstup D a výstup Q. Pokud



Obrázek 4.1: Klopný obvod D

na D přivedeme opačnou logickou hodnotu, stav (a výstup) klopného obvodu se změní jen při náběžné hraně hodinového signálu CLK (clock). Logická 1 na signálu RST (reset) nastaví a bude držet stav klopného obvodu v logické 0 dokud se signál RST nevrátí zpět do 0. Vstup RST je asynchronní, nezávisí na hodinovém signálu CLK. Čipy společnosti Altera, např. použitý čip Altera Cyclone IV E, obsahují klopné obvody D s asynchronním reset signálem, ale např. registry v čipech společnosti Xilinx mají synchronní reset.

4.2 VHDL

VHDL je HDL jazyk (Hardware Description Language) pro popis návrhu digitálního hardware[3]. Jazyk byl původně navržen pro simulaci ASIC obvodů, ale postupem času vznikly nástroje pro automatickou syntézu definic pro tvorbu elektrických obvodů. Jazyk byl vytvořen Ministerstvem obrany USA a v roce 1987 vznikl jeho první standard IEEE 1076-1987 (VHDL-1987). V této práci byla použita verze jazyka VHDL-2008.

S rozšířením FPGA čipů se jazyk VHDL (a další HDL jazyky) začal používat i pro syntézu FPGA bitstreamů. Druhý rozšířený HDL jazyk Verilog. VHDL má oproti Verilogu silnější statický typový systém, díky čemuž je možné odhalit chyby při překladu a syntéze výsledného bitstreamu.

Základní funkční prvky VHDL návrhů jsou přiřazení signálů a procesy. Proces je smyčka, která se myšlenkově vykonává pořád dokola. Výsledný logický obvod závisí na překladači. VHDL nemá oproti Verilogu speciální syntax pro definici registrů, ty se musí definovat pomocí speciálního konstrukturu v procesu, kdy se logika přiřazení signálů provede za podmínky, že nastala náběžná hrana hodin. Takto například vypadá výše popsáný registr definovaný v jazyce VHDL:

```
process (clk, rst)
begin
    if rst = '0' then
        q <= '0';
    elsif rising_edge(clk) then
        q <= d;
    end if;
end process;
```

Operace `q <= d` přiřadí hodnotu signálu `d` do signálu `q`. V závorkách procesu je uveden seznam signálů, po jejichž změně se proces vykoná. Ve větvení je vidět, že změna signálu RST se projeví ihned, a změna na vstupu D se projeví jen při náběžné hraně hodin (`rising_edge(clk)`).

Zapouzdření je dosaženo pomocí tzv. entit. Entita obsahuje rozhraní a architekturu, což je implementace rozhraní. V rozhraní jsou definovány vstupní a výstupní logické signály a v architektuře jsou pomocné signály použité jen uvnitř jedné architektury, přiřazení signálů kombinováním jiných signálů a procesy, které definují složitější přiřazení signálů a registry.

Více podrobností o jazyce VHDL je v literatuře [2] a [3].

4.3 Avalon[®]

Avalon[®] je standard definovaný společností Altera pro snadné propojování komponent. Komponenty jsou například VHDL entity (nebo Verilog ekvivalent) doplněné souborem s metadaty, který dále popisuje komponentu a signály rozhraní. Komponentu je potom možno distribuovat jako uzavřený systém, který se jen zapojí do systému s Avalon[®] sběrnici.

Avalon definuje několik typů rozhraní. Každá komponenta má jedno nebo více rozhraní. Propojovat lze jen rozhraní stejného typu. Dále je možné propojovat jen rozhraní, které mají stejný hodinový signál. Komponenta může definovat jeden nebo více hodinových vstupů a každé její rozhraní je řízeno právě jedním hodinovým signálem.

V dalších podsekcích budou zmíněny hlavní typy rozhraní použitých v této práci.

Avalon-ST

Avalon-ST (Avalon Stream Interface) je rozhraní proudového typu, kde se data přenášejí paralelně. Rozhraní je podobné např. rozhraní Camera Link[®]. Komponenty definují Avalon-ST vstupy a výstupy a v systému se za sebe řetězí. Toto rozhraní je ideální pro zpracování obrazu, kde algoritmy nepracují s celým obrazovým rámcem. Rozhraní podporuje tzv. backpressure pomocí signálu `ready`, kde přijímací strana může dočasně pozastavit přenos odesílatele. Rozhraní dále podporuje přenos paketů, kde jsou data jednoho paketu vymezeny signály `startofpacket` a `endofpacket`. V případě backpressure a paketového přenosu musí tyto signály podporovat výstupní i vstupní strana.

Avalon-MM

Avalon-MM (Avalon Memory Mapped Interface) je rozhraní typu master-slave s adresní a datovou sběrnicí. Rozhraní je podobné např. rozhraní I²C. Propojení master a slave komponent reprezentuje jednu paměťovou sběrnicí, kde komponenty typu master mohou číst a zapisovat do adresního prostoru komponent typu slave. Adresní prostor slave komponent může být implementován libovolně. Sběrnice je vhodná např. pro paměti a GPIO.

Ostatní rozhraní

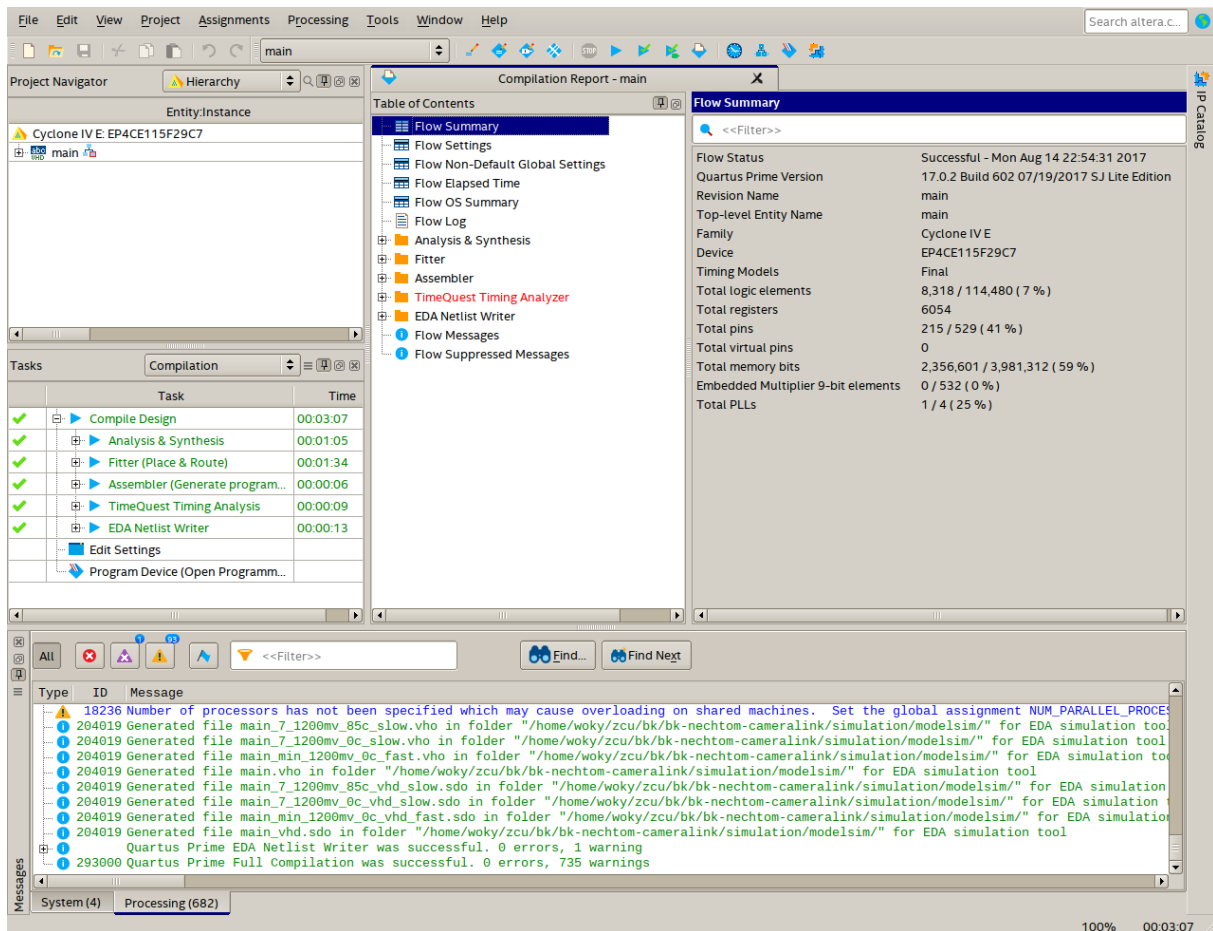
Avalon-Conduit rozhraní slouží pro vyvedení signálů mimo systém Avalon[®]. Avalon-Interrupt rozhraní slouží pro signály přerušení. Využívají ho například vstupní GPIO piny pro signalizaci změny logické úrovně. Avalon-Clock a Avalon-Reset rozhraní jsou pro rozvod hodin a přerušení.

4.4 Vývojové prostředí Quartus

Pro překlad VHDL zdrojových souborů a syntézu FPGA bitstreamu bylo použito vývojové prostředí Quartus. Sada software Quartus obsahuje překladač, syntetizér, editoru VHDL kódu a další nástroje pro vývoj pro čipy společnosti Altera. Na obrázku 4.2 je snímek obrazovky po úspěšné syntéze bitstreamu.

Qsys

Program Qsys slouží k navržení Avalon[®] systému popsaného v předchozí sekci. Na obrázku 4.3 je snímek obrazovky programu se systémem navrženým pro tuto práci. V levém horním



Obrázek 4.2: Vývojové prostředí Quartus

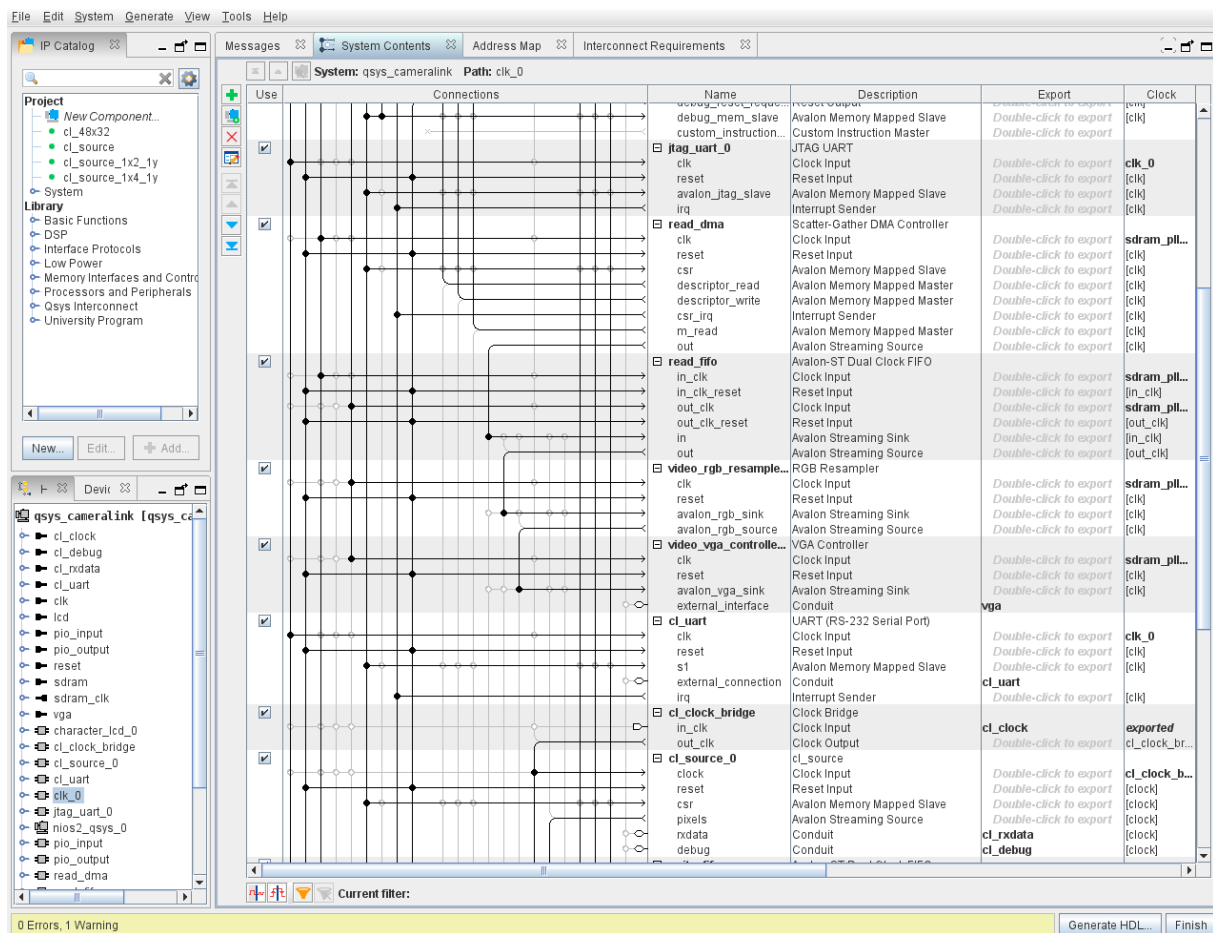
rohu je knihovna komponent dodaných společností altera (Library) a seznam vlastních komponent (Project). Ve sloupci Export jsou signály, které budou vyvedeny mimo systém Avalon®. Jedná se typicky o signály Avalon-Conduit rozhraní, které se v hlavní entitě VHDL napojí na periferie desky.

4.5 Vývojová deska Altera DE2-115

Vývojová deska Altera DE2-115[8] obsahuje FPGA čip spolu s mnoha dalšími perifériemi (obrázek 4.4). Jádrem desky je FPGA čip Altera Cyclone IV E, konkrétně typ EP4CE115, na který se nahrává výsledný FPGA bitstream. Periferie desky, které byly použité v této práci jsou:

- Paměť SDRAM – pro ukládání obrazových rámců
- USB Basler Port – pro nahrávání FPGA bitstreamu
- VGA port – pro zobrazování obrazu na VGA monitoru
- RS-232 port – pro konfiguraci kamery v počáteční fázi vývoje
- HSMC konektor – pro připojení CLR-HSMC Camera Link® rozhraní

KAPITOLA 4. VÝVOJOVÉ NÁSTROJE



Obrázek 4.3: Avalon[®] systém v programu Qsys

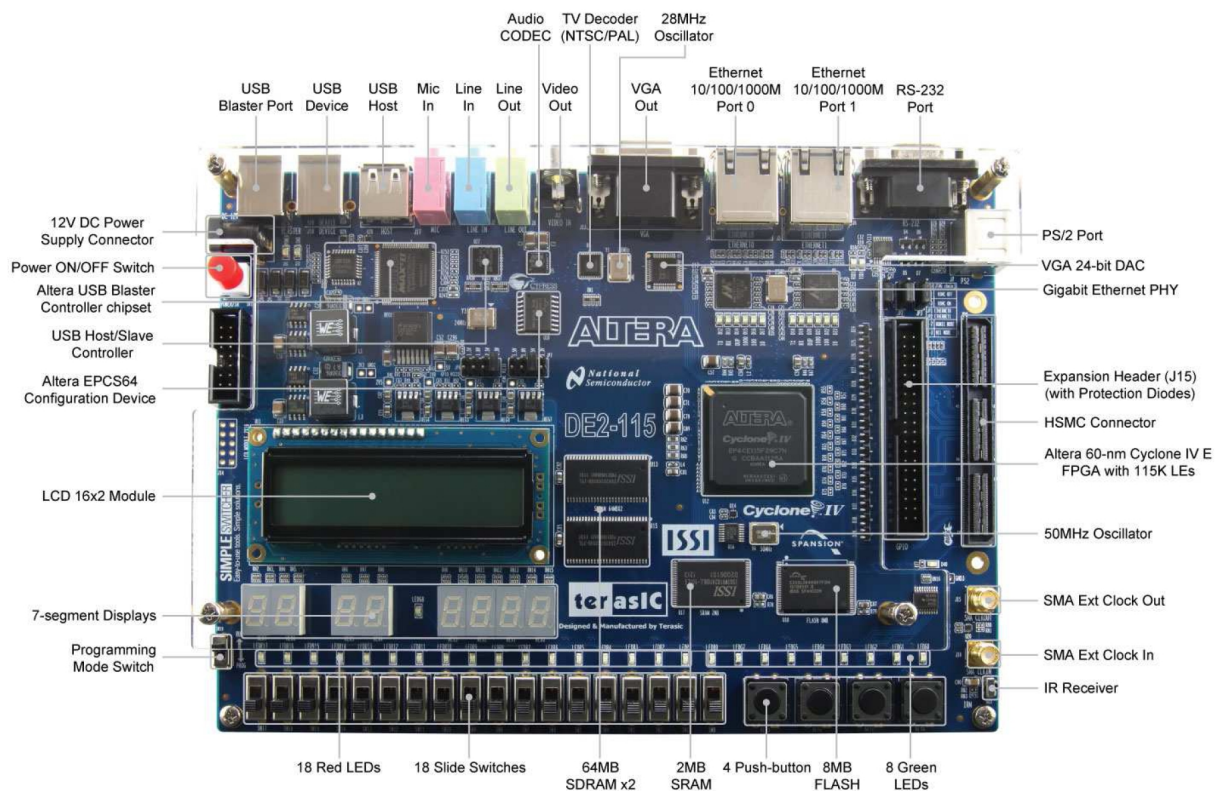
- 50MHz oscilátor – jako zdroj hodin
- Tlačítka – pro reset bitstreamu
- Přepínače – pro volbu výstupu z kamery (obraz ze senzoru a testovací rámce)
- Zelené LED diody – pro indikaci stavu inicializace

Deska obsahuje 2 paměťové čipy DRAM IS42S16320D společnosti Integrated Silicon Solution, Inc. každý o velikosti 64 MB.

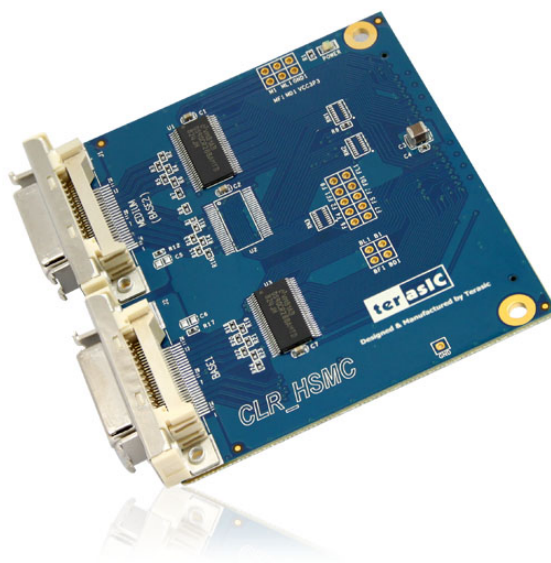
4.6 CLR-HSMC

Pro konverzi signálů z rozhraní Camera Link[®] na digitální signál přivedený do FPGA čipu byla použita rozšiřující deska CLR-HSMC[9]. Rozšíření CLR-HSMC je k desce Altera DE2-115 připojeno přes rozhraní HSMC (High Speed Mezzanine Card), definované společností Altera. Rozšíření má dva MDR Camera Link[®] konektory, ale jen dva čipy Channel Link. Nejrychlejší konfigurace Camera Link[®], která jde s touto deskou dosáhnout je tedy konfigurace Medium.

KAPITOLA 4. VÝVOJOVÉ NÁSTROJE



Obrázek 4.4: Vývojová deska Altera DE2-115

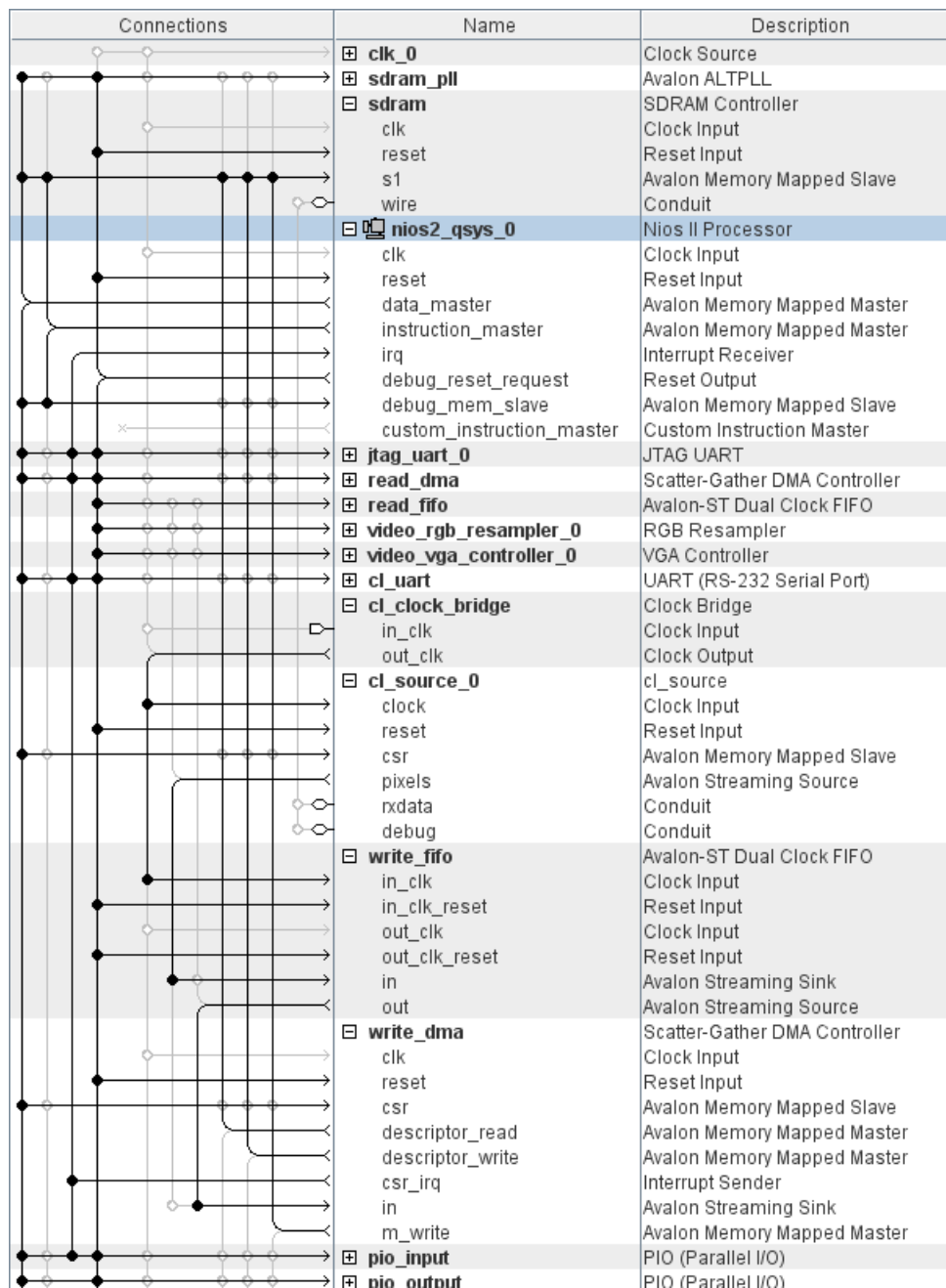


Obrázek 4.5: Rozšíření CLR-HSMC

5. Implementace

Práce byla pojata jako demonstrace použití kamery s Camera Link[®] rozhraním. Tedy výsledný FPGA bitstream obsahuje Frame Grabber, zápis rámců do paměti a zobrazení obrazu na VGA monitoru.

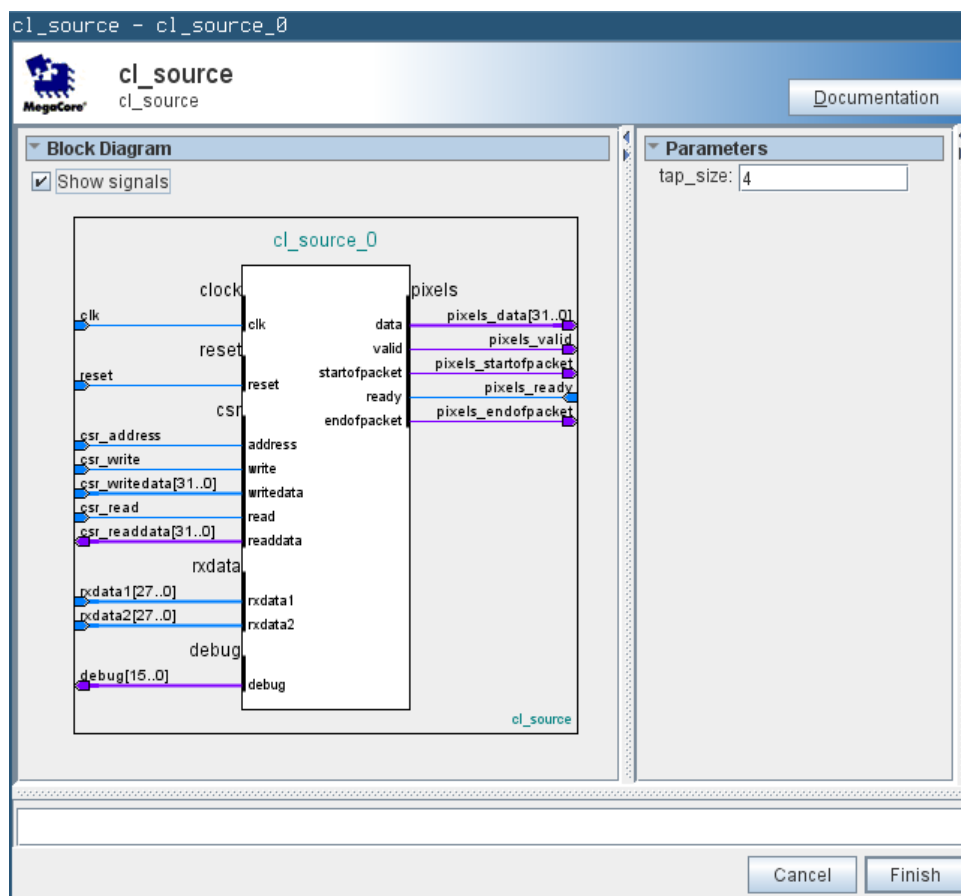
Většina logiky byla implementována jako Avalon[®] systém (qsys_cameraLink). Zapisovací část je na obrázku 5.1. Proces získávání obrazu je nastartován procesorem Nios II/e, který inicializuje periferie a spustí Frame Grabber. Data z frame grabberu potom jdou přes frontu do SGDMA komponenty, která je zapisuje do paměti.



Obrázek 5.1: Zapisovací část Avalon[®] systému

5.1 Frame Grabber

Frame Grabber byl implementován jako generická Avalon[®] komponenta. Na obrázku 5.2 je konfigurační okno komponenty v programu Qsys. Parametr `tap_size` specifikuje očekávanou tap geometrii (viz sekce 3.2). Hodnota 4 představuje tap geometrii 1X4-1Y.



Obrázek 5.2: Frame Grabber Avalon[®] komponenta

5.1.1 Avalon[®] rozhraní

clock

Hodinový signál. Komponenta očekává hodinový signál Camera Link[®] z kamery. Na frekvenci těchto hodin pracují všechny ostatní rozhraní.

reset

Reset komponenty.

csr

Configuration/Status Registers. Komponenta vystavuje Avalon-MM rozhraní pro zapisování konfigurace a čtení stavu. Adresový prostor je rozdělen na prostor pro zápis a prostor

KAPITOLA 5. IMPLEMENTACE

pro čtení. Tedy zapsaná konfigurace nelze číst, a do stavového registru nelze zapisovat.

V následujících tabulkách je popis registrových map.

Adresa	Název	Popis
0x0	status	Status Frame Grabberu. První (LSB) bit je 1, když Frame Grabber běží (čte data z kamery a posílá pakety na Avalon-ST výstup), jinak je 0.

Tabulka 5.1: Registry pro čtení

Adresa	Název	Popis									
0x0	control	Zápis hodnoty 1 spustí Frame Grabber.									
0x4	resolution	Rozlišení obrazu z kamery: <table border="1"><thead><tr><th>Bit</th><th>Název</th><th>Popis</th></tr></thead><tbody><tr><td>31..16</td><td>width</td><td>Počet sloupců.</td></tr><tr><td>15..0</td><td>height</td><td>Počet řádků.</td></tr></tbody></table>	Bit	Název	Popis	31..16	width	Počet sloupců.	15..0	height	Počet řádků.
Bit	Název	Popis									
31..16	width	Počet sloupců.									
15..0	height	Počet řádků.									

Tabulka 5.2: Registry pro zápis

Řídící program by měl nejprve zapsat rozlišení kamery do registru **resolution** a následně spustit Frame Grabber zápisem 1 do registru **control**.

pixels

Frame Grabber vystavuje pixely obrazu na výstupní rozhraní Avalon-ST. Rozhraní podporuje paketový přenos. Začátek obrazového rámce indikuje logická 1 na signálu **pixels_startofpacket**, konec logická 1 na signálu **pixels_endofpacket**. Paketový přenos je využit v komponentě SGDMA, která zapisuje data do paměti SDRAM. Rozhraní také podporuje pozastavení přenosu vstupním signálem **pixels_ready**. Pozastavením ve většině případů přejde komponenta dočasně do stavu kde neodesílá data, protože přenos z kamery nejde pozastavit (viz dále).

rxdata

Camera Link[®] signály. Toto je Avalon-Conduit rozhraní vyvedené do hlavní entity, kde je napojené na signály z CLR-HSMC modulu.

debug

Informace o stavu komponenty vyvedené do hlavní entity. Avalon-Conduit rozhraní určené čistě jen pro usnadnění vývoje komponenty.

5.1.2 Synchronizace

Výstupní Avalon-ST rozhraní podporuje pozastavení přenosu pro případ, že další zřetězené komponenty nebudou stíhat datový tok (např. zápis do paměti). Ale protože komponenta běží z hodin Camera Link[®], které není možné pozastavit, musí se pozastavit čtení pixelů a po znovuzahájení začít od nového obrazového rámce.

Stavová komponenta se může nacházet v jednom ze čtyř stavů: **desync3**, **sync**, **desync1** a **desync2**.

desync3

Počáteční stav. V tomto stavu komponenta čeká na zápis 1 do konfiguračního registru **control**, na to, až Camera Link[®] signál **FVAL** (Frame Valid, viz 2.3) bude v nule a na to, až komponenta napojená na Avalon-ST výstup nastaví signál **pixels_ready** na jedničku. Pokud nastanou všechny tyto události najedou, znamená to, že kamera dokončila přenos posledního snímku, čeká se na přenos dalšího snímku, který bude indikován náběžnou hranou **FVAL** a další komponenta je schopná přijímat obrazové rámce. V tomto případě Frame Grabber přejde do stavu **sync**.

sync

V tomto stavu je aktivní přenos pixelů a komponenta napojená na Avalon-ST výstup je připravena zpracovávat výstupní data. Kamera může pozastavovat přenos během přenášení obrazu shoením signálu **DVAL** (viz 2.3) do nuly. Frame Grabber pracuje jen když je tento signál v jedničce. Tuto skutečnost indikuje další Avalon-ST komponentě signálem **pixels_valid**. Pokud je v tomto stavu signál **pixels_ready** v nule, znamená to, že další Avalon-ST komponenta není schopná dále zpracovávat výstupní data a Frame Grabber přejde do stavu **desync3** pokud zrovna začal přenos nového obrazového rámce, jinak do stavu **desync1**. Pokud je signál **pixels_ready** v jedničce, inkrementuje se čítač sloupců a řádků podle aktuální pozice a nastaveného rozlišení. Tyto čítače slouží pro indikování začátku a konce obrazového rámce (paketu, signály **pixels_startofpacket** a **pixels_endofpacket**).

desync1 a desync2

Pokud další Avalon-ST komponenta nebyla schopná zpracovat data před koncem přenosu obrazového rámce, znamená to, že nedostala indikaci konce paketu (signál **pixel_endofpacket**). V okamžiku, kdy bude schopná přijímat další data už ale pixely z kamery nebudou navazovat na poslední zpracované pixely (Camera Link[®] hodiny z kamery nelze pozastavit). Tento stav může nastat například pokud je na konci Avalon-ST kaskády komponenta která zapisuje na Ethernet, kde může být rychlost přenosu proměnlivá. Tento stav ale není žádoucí a v reálném nasazení by ve většině případů neměl nastat. Pro usnadnění vývoje aplikace zpracování obrazu je ale vhodné aby byl definovaný. V této práci se jen ukončí přenos momentálně přenášeného rámce a přenos začne od začátku nového rámce až bude další komponenta schopná zpracovávat výstup.

Jak bylo popsáno v předchozí sekci, pokud zrovna začal přenos nového obrazového rámce a další komponenta není schopná přijímat data, přejde se do počátečního stavu,

protože komponenta zatím nedostala signál `pixels_startofpacket` a není tedy potřeba posílat signál `pixels_endofpacket`. Pokud probíhá přenos obrazového rámce, přejde se do stavu `desync1`, kde se čeká dokud je signál `pixels_ready` v nule. Ze stavu `desync1` se přejde do stavu `desync2` na jeden hodinový cyklus, ve kterém je signál `pixels_endofpacket` v jedničce. Ze stavu `desync2` se přejde do počátečního stavu (`desync3`). Účel těchto stavů je tedy jen oznámit další komponentě, že přenos skončil.

Stavový automat

V následujícím výpisu je VHDL kód implementující popsané stavové přechody.

```
process (clk, reset)
    constant row0: unsigned(row'range) := to_unsigned(0, row'length);
    constant col0: unsigned(col'range) := to_unsigned(0, col'length);
begin
    if reset = '1' then
        status_go <= '0';
        state <= desync3;
        row <= row0;
        col <= col0;
    elsif rising_edge(clk) then
        case state is
            when desync1 =>
                if pixels_ready = '1' then
                    state <= desync2;
                end if;
            when desync2 =>
                state <= desync3;
                row <= row0;
                col <= col0;
            when desync3 =>
                if config_go = '1' and fval = '0' and pixels_ready = '1' then
                    status_go <= '1';
                    state <= sync;
                end if;
            when sync =>
                if rxval = '1' or config_go = '0' then
                    if pixels_ready = '1' and config_go = '1' then
                        if col = frame_w - tap_size_u then
                            col <= col0;
                            if row = frame_h - 1 then
                                row <= row0;
                            else
                                row <= row + 1;
                            end if;
                        else
                            col <= col + tap_size_u;
                        end if;
                    end if;
                end if;
            end case;
        end if;
    end process;
```

```
else
    status_go <= '0';
    if col = col0 and row = row0 then
        -- SOP wasn't sent
        state <= desync3;
    else
        -- send EOP when sink becomes ready
        state <= desync1;
    end if;
end if;
end if;
end case;
end if;
end process;
```

Zdrojové kódy Frame Grabberu jsou v adresáři `cl_source`.

5.2 Řídící program

Pro snadnou inicializaci a konfiguraci komponent byl použit procesor Nios II/e. V Avalon[®] systému je napojený jako Avalon-MM master na všechny komponenty, které inicializuje. Procesor umožňuje řídit funkci výsledného bitstreamu vykonáváním skevenčního programu, který je uložený v paměti. Altera poskytuje překladač, standardní knihovnu jazyka C, funkce pro přístup k Avalon[®] komponentám. Knihovna Altera částečně emuluje POSIX API. Komponenty jsou v programu reprezentovány souborovými deskriptory, které se získají otevřením souborů zařízení v pseudo souborovém systému `/dev`. Většina dodávaných komponent nabízí pro svou konfiguraci dodatečné C API.

Program po startu

1. otevře Camera Link[®] sériovou linku a provede inicializaci kamery,
2. alokuje paměť pro ukládání obrazových rámců
3. inicializuje SGDMA komponentu pro zápis obrazu do paměti
4. inicializuje a spustí Frame Grabber,
5. inicializuje tlačítka a přepínače
6. inicializuje SGDMA komponentu pro čtení obrazu z paměti (VGA monitor)

Potom vkročí do nekonečné smyčky, kde reaguje na změnu přepínačů a mění výstup z kamery.

Zdrojový kód programu je v souboru `nios_app/src/main.c`.

5.3 SDRAM buffer

Pro zápis do paměti DRAM byla použita Avalon[®] komponenta SGDMA (Scatter-Gather Direct Memory Acces). Tato komponenta byla použita v módu *Stream To Memory*. Její datová rozhraní jsou Avalon-ST vstup a Avalon-MM master.

5.4 Konfigurace kamery

Pro konfiguraci kamery byl implementován binární protokol pro přístup do registrů Basler kamery. Rozhraní protokolu pro jazyk C je v souboru `basler_serial.h` a implementace v souboru `basler_serial.c` v adresáři `basler_serial`. Rozhraní definuje jednoduché API pro čtení a zápis registrů v paměťovém prostoru Basler kamery:

```
ssize_t basler_read(int fd, unsigned addr, size_t len, void *buf);
ssize_t basler_write(int fd, unsigned addr, size_t len, void *buf);
ssize_t basler_write_dword(int fd, unsigned addr, uint32_t val);
```

API vyžaduje otevřený souborový deskriptor sériové linky k němuž přistupuje přes POSIX API funkce pro zápis a čtení. API se používá v řídicím programu, ale lze ho použít i na klasickém operačním systému s POSIX API.

Konfigurace kamery v této práci se provádí následovně:

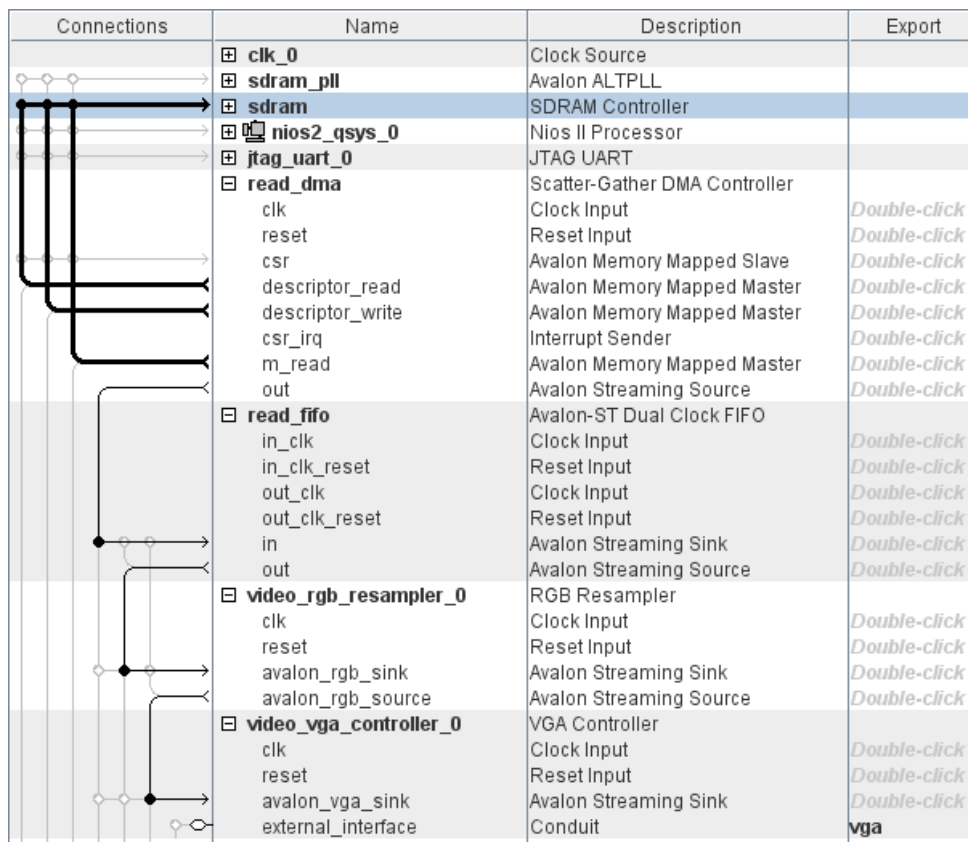
```
// pixel clock = 82 MHz
if (basler_write_dword(fd, 0x00000740 + 0x4, 23) < 0)
    return -1;
// sensor bit depth = 8 bits per pixel
if (basler_write_dword(fd, 0x00030500 + 0x4, 8) < 0)
    return -1;
// tap geometry = 1x4-1y (medium, 4 pixels)
if (basler_write_dword(fd, 0x00000720 + 0x4, 9) < 0)
    return -1;
// image width
if (basler_write_dword(fd, 0x00030200 + 0x4, FRAME_W) < 0)
    return -1;
// image height
if (basler_write_dword(fd, 0x00030220 + 0x4, FRAME_H) < 0)
    return -1;
// center x
if (basler_write_dword(fd, 0x00030280 + 0x4, 1) < 0)
    return -1;
// center y
if (basler_write_dword(fd, 0x000302A0 + 0x4, 1) < 0)
    return -1;
// time base = 1us
if (basler_write_dword(fd, 0x00040440 + 0x4, 1000) < 0)
    return -1;
// exposure time raw = 1 ms
```

```
if (basler_write_dword(fd, 0x00040460 + 0x4, 5000) < 0)
    return -1;
```

Registry kamer Basler mají 4 bytovou hlavičku, za kterou následuje hodnota. V dokumentaci (3.1) jsou adresy registrů uvedeny od hlavičky. V programu se tedy používají tyto *kanonické* adresy a přičte se k nim offset hodnoty.

5.5 Zobrazení

Pro demonstraci funkčnosti byl použit standardní VGA monitor s rozlišením 800x600. Pro převod na VGA signál byly použity komponenty *Altera University Program Video Out IP Cores*. Zapojení zobrazovací části Avalon® systému je na obrázku 5.3.



Obrázek 5.3: Zobrazovací část Avalon® systému

6. Závěr

Práce ukázala, že s pomocí FPGA lze přenášet obraz z kamery s Camera Link[®] v konfiguraci Medium. Práce je připravena na rozšíření pro další aplikace. Zpracování obrazu se může provádět na frekvenci přenosu rozhraní Camera Link[®] pouhým přidáním Avalon-ST komponenty do kaskády. Zpracování obrazu nad celým rámcem je také možné, protože výsledný bitstream používá paměť SDRAM jako buffer. To se může výhodně využít v hardware, kde FPGA a procesor sdílejí paměť.

Všechny body zadání byly splněny. Při zpracování byly analyzovány a vyzkoušeny různé možnosti implementace. Zadání neobsahovalo požadavek na ukládání obrazu do paměti DRAM, ale použití paměti DRAM usnadnilo požadavek na zobrazení na VGA monitor, neboť rychlost rozhraní Camera Link[®] je mnohem větší než rychlost rozhraní VGA. Tato implementace bude mít také přínos pro další aplikace, jako je zpracování obrazu nebo ukládání videa.

Významný výsledek této práce je modularita Frame Grabberu a možnost snadné změny různých aspektů výsledného FPGA bitstreamu, díky použití standardu Avalon[®]. Nevýhoda tohoto přístupu je závislost na čípech a nástrojích společnosti Altera. Autor byl obeznámen s plány do budoucna a proto zvolil tento přístup.

Další praktický výsledek je knihovna a API pro komunikaci s kamerami Basler s rozhraním Camera Link[®]. Knihovna byla zpočátku použita v operačním systému Linux, vyvedením signálů sériové linky na RS-232 port a propojením s PC pomocí FTDI USB kabelu. Později byla veškerá konfigurace přenesena do programu pro procesor Nios II/e na čipu FPGA.

6.0.1 Další cíle

Frame Grabber momentálně neupozorňuje pokud další prvek v Avalon-ST kaskádě pozastaví příjem čímž se ztratí část nebo i více celých obrazových rámců. Možné řešení by bylo rozšířit Frame Grabber o Avalon-Interrupt.

Frame Grabber byl vyzkoušen jen v Medium Camera Link[®] konfiguraci s tap geometrií 1X4-1Y. Pro vyšší rychlosti by bylo potřeba použít Camera Link[®] přijmač kompatibilní s konfigurací Full nebo 80 bit.

Seznam použitých zkratek

AIA Automated Imaging Association

FPGA Field-Programmable Gate Array

USB Universal Serial Bus

MAC Media Access Control

LVDS Low Voltage Differential Signaling

ASIC Application Specific Integrated Circuit

VHDL VHSIC Hardware Description Language

VHSIC Very High Speed Integrated Circuit

HDL Hardware Description Language

SDRAM Synchronous Dynamic Random-Access Memory

Seznam použité literatury

- [1] *Camera Link 2.0 - Specifications of the Camera Link Interface Standard for Digital Cameras and Frame Grabbers, version 2.0*, AIA, 2012.
- [2] KARIM, CHEN. *Digital Design - Basic Concepts and Principles*. CRC Press, 2007. ISBN 978-1-420061-31-4.
- [3] ASHENDEN *The Designer Guide to VHDL, 3rd Edition*. Elsevier, 2008. ISBN 978-0-080568-85-0.
- [4] Jan AXELSON. *USB Complete, 5th Edition*. Lakeview Research LLC, 2015. ISBN 978-1-931448-29-1.

Elektronické zdroje

- [5] AIA Vision Standards. <https://www.visiononline.org/vision-standards.cfm>
- [6] About AIA. <https://www.visiononline.org/vision/about-aia>
- [7] Camera Link — The Only Real-Time Machine Vision Protocol – AIA <https://www.visiononline.org/vision-standards-details.cfm?type=6>
- [8] Altera DE2-115 Development and Education Board – Terasic <https://www.terasic.com.tw/cgi-bin/page/archive.pl?No=502>
- [9] CLR-HSMC Camera Link Receiver Daughter Card – Terasic <https://www.terasic.com.tw/cgi-bin/page/archive.pl?No=588>
- [10] Basler ace acA2000-340kmNIR – Basler <https://www.baslerweb.com/en/products/cameras/area-scan-cameras/ace/aca2000-340kmnir/>
- [11] REGISTER STRUCTURE AND ACCESS METHODS FOR CAMERA LINK CAMERAS – Basler <https://www.baslerweb.com/en/support/downloads/document-downloads/basler-racer-camera-link-register-structure-and-access-methods/>
- [12] GigE Vision® – AIA <https://www.visiononline.org/vision-standards-details.cfm?type=5>
- [13] IEEE 802.3™: Ethernet – IEEE <http://standards.ieee.org/about/get/802/802.3.html>
- [14] Gigabit Ethernet – Wikipedia https://en.wikipedia.org/wiki/Gigabit_Ethernet
- [15] RFC 1180: A TCP/IP Tutorial – IETF <https://tools.ietf.org/html/rfc1180>
- [16] RFC 768: User Datagram Protocol – IETF <https://tools.ietf.org/html/rfc768>
- [17] RFC 2131: Dynamic Host Configuration Protocol – IETF <https://tools.ietf.org/html/rfc2131>

SEZNAM POUŽITÉ LITERATURY

- [18] Acquiring from GigE Vision Cameras with Vision Acquisition Software – National Instruments <http://www.ni.com/white-paper/5651/en/>
- [19] Beej's Guide to Network Programming — Using Internet Sockets <https://beej.us/guide/bgnet/output/html/multipage/index.html>
- [20] Windows Sockets 2 – MSDN [https://msdn.microsoft.com/en-us/library/windows/desktop/ms740673\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/ms740673(v=vs.85).aspx)
- [21] Choosing the Right Camera Bus – National Instruments <http://www.ni.com/white-paper/5386/en>
- [22] USB3 Vision™ – AIA <https://www.visiononline.org/vision-standards-details.cfm?type=11>
- [23] USB3 Vision™ version 1.0.1 – AIA https://www.visiononline.org/userAssets/aiaUploads/file/USB3_Vision_Specification_v1-0-1.pdf
- [24] libusb <http://libusb.info/>
- [25] Cable USB 3.0, Ext. A female / A male, 20 m - Data Cables – Basler <http://libusb.info/>
- [26] SuperSpeed USB – USB.org <http://www.usb.org/developers/ssusb/>
- [27] National Semiconductor Channel Link Design Guide – Texas Instruments <http://www.ti.com/lit/ug/snla167/snla167.pdf>
- [28] An Overview of LVDS Technology – Texas Instruments <http://www.ti.com/lit/an/snla165/snla165.pdf>
- [29] DS90CR287 – Texas Instruments <http://www.ti.com/product/DS90CR287>
- [30] DS90CR288A – Texas Instruments <http://www.ti.com/product/DS90CR288A>
- [31] DS90LV047A – Texas Instruments <http://www.ti.com/product/DS90LV047A>
- [32] DS90LV048A – Texas Instruments <http://www.ti.com/product/DS90LV048A>