

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ

Fakulta elektrotechnická

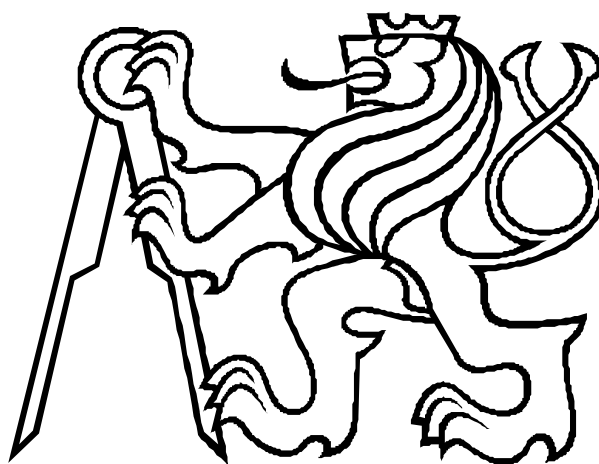
DIPLOMOVÁ PRÁCE

2005

Štěpán Martínek

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra měření



Diplomová práce

Univerzální řídicí jednotka pro CCD snímače

Autor: Štěpán Martínek
Vedoucí: Ing. Jan Fischer, CSc.

Praha, leden 2005

Abstrakt

V poslední době jsou kladeny čím dál větší požadavky na rychlost zpracování informací. V oblasti videometrie se to týká hlavně zpracování signálu či obrazu. Tato diplomová práce se zabývá možností využít programovatelné hradlové pole spolu se signálovým procesorem k řízení CCD nebo CMOS snímačů, předzpracování získaného obrazu v reálném čase a následný přenos dat do nadřazeného PC pomocí rychlého rozhraní.

Abstract

Requirements for a speed of data processing are increasing every day. Especially the speed of signal and image processing is relevant in videometry. This diploma thesis considers an option to use a field programmable gate array along with a signal processor for CCD or CMOS image sensor control and real-time image preprocessing followed by data transmission to the PC via fast interface.

Čestné prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v příloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona č. 121/2000 Sb. , o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V PRAZE DNE

.....

Štěpán Martínek

Obsah

1. ÚVOD	4
2. ROZBOR	5
2.1 Základní požadavky	5
2.2 Softwarové řízení	5
2.3 Nevýhody softwarového řízení.....	5
2.4 Videoprocessor	6
2.5 Volba rychlého rozhraní.....	6
2.6 Možnosti použití	8
3. ŘÁDKOVÁ KAMERA ULC85	9
3.1 Blokové schéma	9
3.2 Popis jednotlivých funkčních bloků ULC85.....	10
3.2.1 Řídicí procesor	10
3.2.2 Registr řízení CCD	10
3.2.3 Analogové vstupy + AFE	11
3.2.4 Rozhraní USB.....	13
3.2.5 Linka RS-232	15
3.2.6 Watchdog – dohlížecí obvod.....	16
3.2.7 Uživatelské vstupy/výstupy.....	17
3.2.8 Dekódovací logika pro periferie	18
3.2.9 Konektor pro připojení CCD snímače	19
3.2.10 Napájecí zdroj	20
3.3 Snímačová deska pro ILX551, ILX751.....	20
4. SP2EDB – VÝVOJOVÁ DESKA S FPGA SPARTAN-2E	21
4.1 Účel zařízení.....	21
4.2 Výběr hradlového pole	21
4.3 Blokové schéma	22
4.4 Popis jednotlivých funkčních bloků	23
4.4.1 Hradlové pole FPGA.....	23
4.4.2 Rychlá statická paměť RAM.....	24
4.4.3 Budiče pro registr řízení CCD snímače	24
4.4.4 Uživatelské rozhraní	26
4.4.5 Konektor pro obecné použití.....	27
4.4.6 Konektor pro externí analogovou část	27
4.4.7 USB interface.....	28
4.4.8 Sériová linka RS-232.....	29

4.4.9	Procesor pro konfiguraci FPGA	29
4.4.10	Napájecí zdroj	31
4.5	Možnosti použití SP2EDB	33
5.	ACID-IS – KOMPLEXNÍ ŘEŠENÍ	34
5.1	Spojení ULC85 a SP2EDB.....	34
5.2	Blokové schéma celé desky.....	34
5.3	Funkce jednotlivých bloků	34
5.3.1	Signálový procesor DSP.....	35
5.3.2	Budiče pro registr řízení CCD	35
5.3.3	Analog Front-End	36
5.3.4	Hradlové pole FPGA.....	36
5.3.5	Rychlá Statická paměť RAM	38
5.3.6	Obecná propojovací logika „glue logic“	38
5.3.7	Dohlížecí obvod – Watchdog.....	39
5.3.8	Sériová paměť DataFlash	40
5.3.9	Konektor pro obecné použití.....	40
5.3.10	Konektor pro připojení snímačové desky	41
5.3.11	Sériová linka RS-232.....	41
5.3.12	Komunikační rozhraní USB.....	41
5.3.13	Rozhraní Fast-Ethernet 100Base-TX	42
5.3.14	Napájecí zdroj	46
6.	PROGRAMOVÉ VYBAVENÍ.....	47
6.1	Řídicí program pro ULC85	47
6.1.1	ULC85V2 . DSP	47
6.1.2	UART . DSP	48
6.1.3	AFECFG . DSP	49
6.1.4	CCD_INTERFACE . DSP.....	49
6.1.5	ILX551T . DSP.....	49
6.1.6	PROTV2 . DSP.....	50
6.2	Přenosový protokol ProtV2.....	50
6.2.1	Linková vrstva ProtV2.....	50
6.2.2	Rámcová vrstva ProtV2.....	50
6.3	Zobrazovač pro ULC85.....	52
6.4	Řídicí program pro SP2EDB.....	53
6.4.1	Program pro řídicí procesor	53
6.5	Manažer SP2EDB pro PC	55
6.6	Řídicí program pro ACID-IS	56
6.6.1	ACID . DSP	57
6.6.2	LAN . DSP.....	57

6.6.3	OSI . DSP	58
6.6.4	USB . DSP.....	60
6.6.5	PROTV2 . DSP.....	60
6.6.6	CCD . DSP.....	60
6.6.7	SWSPI . DSP.....	60
6.6.8	FPGABOOT . DSP	61
7.	SHRNUTÍ POSTŘEHŮ, TRIKŮ A NEDOSTATKŮ	62
7.1	Jumpery na ULC85	62
7.2	Signál #BUSRQ na SP2EDB.....	62
7.3	ISP programování mikroprocesoru na SP2EDB.....	62
7.4	Obvod Power Assist na SP2EDB	62
7.5	Pin CLKUSR.....	62
7.6	EEPROM u LAN na ACID-IS	62
7.7	Čtení z LAN.....	62
7.8	Přenos přes Ethernet.....	63
7.9	Popis komunikace s PHY	63
7.10	Oddíly v DataFlash na SP2EDB.....	63
7.11	Omezení Xilinx ISE WebPack	63
8.	MOŽNOSTI A INSPIRACE PRO DALŠÍ PRÁCI	63
9.	ZÁVĚR.....	64
10.	SEZNAM POUŽITÉ LITERATURY	66
11.	SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ	67
12.	SEZNAM OBRÁZKŮ.....	69
13.	SEZNAM TABULEK.....	70
14.	PŘÍLOHY	71

1. ÚVOD

Bezkontaktní optické měření a vyhodnocování se stále častěji nasazuje do průmyslových technologických procesů, díky jeho výhodám, popularitě a modernímu trendu. Mezi výhody optických měřicích metod patří zejména malé ovlivňování měřeného procesu. Měřicí systémy, které jako zdroj informace používají obraz, se nazývají videometrické. Obor, zabývající se získáváním fyzikálních veličin z obrazu, se nazývá videometrie. Název pochází z latinských slov *video* (vidět) a *metria* (měření).

Dnes se stávají videometrické systémy stále populárnější. S vývojem nových obrazových snímačů s rozlišením překračujícím 80 milionů obrazových elementů (Fairchild Imaging) roste jejich přesnost a signálové procesory s rychlostí až 8 miliard instrukcí za sekundu (Texas Instruments) dodávají obrovské možnosti rychlého zpracování obrazu.

Právě na videometrické systémy jsou dnes kladeny čím dál větší požadavky. Mezi nejdůležitější nároky patří rychlost pořízení obrazu, následně rychlost jeho zpracování a v neposlední řadě rychlost přenosu dat do nadřazeného procesu. Požadované rychlosti lze dosáhnout více způsoby.

Jeden způsob je, že se v měřicím systému použije velmi rychlý, ale také většinou drahý mikroprocesor. Rychlost však v tomto případě nemusí být všechno. Seberyhlejší mikroprocesor stále vykonává instrukce jednu po druhé v časovém sledu. Nedokáže tedy vykonat více operací v jediném okamžiku (některé signálové procesory sice vykonávají najednou až 8 instrukcí, stále je to však omezený počet).

Výhodnějším řešením by bylo rozdělit práci mezi několik specializovaných procesorů, které by spolu navzájem komunikovaly, např. signálový procesor a videoprocessor. Signálový procesor lze použít jakýkoli vhodný, běžně vyráběný a dostupný na trhu. Videoprocessor je možné navrhnout jako zakázkový integrovaný obvod (ASIC), ale to se vyplatí až tehdy, když se předpokládá výroba statisícových sérií takového obvodu. Jestliže se ale navrhuje prototyp, je nejvhodnější použít nějaký programovatelný obvod, třeba programovatelné hradlové pole (FPGA). Teprve pokud se navržený videoprocessor v praxi osvědčí, je pak vhodné jej vyrábět jako ASIC, nebo případně videoprocessor integrovat na čip společně s CMOS obrazovým snímačem. Dohromady by tak tvořily kompaktní jednočipový videometrický systém.

Cílem této diplomové práce je zvážit právě toto druhé řešení a navrhnout univerzálně použitelné zařízení, které by umožnilo ovládat různé typy obrazových snímačů. Obrazová data ze snímače by se předzpracovávala ve videoprocessoru a do nadřazeného procesu by se přenášela pomocí rychlého rozhraní. Zařízení by tak umožnilo implementaci i složitějšího videometrického systému.

2. ROZBOR

2.1 Základní požadavky

Úkolem mé diplomové práce je navrhnout univerzální jednotku pro řízení obrazových snímačů. Jednotka by měla dokázat řídit libovolný CCD nebo CMOS snímač, přenášet z něj obrazová data, zpracovávat je a přenášet do nadřazeného PC pomocí rychlého rozhraní.

2.2 Softwarové řízení

Postup prací bude rozdělen do několika bodů. Nejprve se budu soustředit na návrh univerzální jednotky pro řádkové snímače, která bude veškeré řízení CCD a AD převodníku dělat pouze pomocí softwaru. Softwarové řízení pomocí mikroprocesoru je jednoduché a snadno se programuje. Deska tak bude schopna pro řádkové, nebo i plošné snímače generovat potřebné sekvence k řízení a získaná obrazová data přenášet do PC.

Zpracování dat bude zajišťovat signálový procesor ADSP-2185M. Jednoduché časově nenáročné zpracování zvládne signálový procesor vykonávat v reálném čase. Není proto potřeba získaná obrazová data nikam ukládat a dají se rovnou po zpracování odesílat do PC.

Běžné řádkové CCD snímače mají 2000 až 5000 a ty největší až kolem 20 000 pixelů. Počítáme-li s 8bitovým převodníkem, pak na jeden pixel vychází právě 1 byte. Celkově je tedy potřeba k uložení celého řádku 20 KB paměti. Použitý signálový procesor má na sobě 32 KB paměti RAM, takže se do ní celý řádek vejde.

U plošných snímačů je však situace jiná. Snímač velikosti standard VGA má rozlišení 640x480 pixelů, to je celkem 307 200 pixelů na snímek. Již na první pohled je jasné, že se do paměti DSP celý snímek nemůže vejít. U plošných snímačů nám tedy nezbude, než zpracovávat data v reálném čase, nebo alespoň po jednotlivých řádcích.

2.3 Nevýhody softwarového řízení

Během řízení a vyčítání snímače nelze procesorem vykonávat žádnou jinou činnost, aniž by to mělo vliv na rychlost vyčítání. Pokud chceme nasnímaná data v reálném čase upravovat, můžeme to provádět ihned po odběru každého vzorku, ovšem perioda čtení snímače se sníží právě o dobu, po kterou se data budou zpracovávat. Druhou možností je tyto činnosti od sebe oddělit. Nejprve vyčíst celý snímač nebo alespoň jeden řádek do paměti procesoru a teprve pak zpracovat data. Rychlost čtení snímače se tímto nesníží, ale zpracování udělá mezi jednotlivými snímky či řádky časové prodlevy. Celková doba pořízení snímku tedy opět vzroste.

Je sice možné použít rychlejší signálový procesor, který by dostatečně rychle zvládal vykonávat obě činnosti současně, ale takový procesor je většinou drahý a také by byla škoda plýtvat drahocenným časem rychlého signálového procesoru na generování sekvencí pro CCD snímač.

2.4 Videoprocessor

Nejlepším řešením by bylo oddělit tyto dvě činnosti ne v rámci jednoho procesoru, ale použít pro každou z nich procesor vlastní. Signálový procesor na zpracování dat a k němu další procesor nebo specializovaný obvodový řadič - videoprocessor, který by generoval sekvence pro CCD, četl data z AFE a odesílal je do DSP. Tím by se signálový procesor nemusel starat o řízení CCD a od videoprocessoru by získával pouze převedené vzorky z AD převodníku. Ve složitějším případě by mohl videoprocessor ještě videosignál předzpracovávat a do DSP by předával pouze důležité části videosignálu, nebo čistě jen získané informace, jako třeba polohy hran nebo souřadnice těžiště.

Další vyvíjená deska tedy bude obsahovat signálový procesor a videoprocessor ve formě programovatelného hradlového pole (FPGA), které kromě toho, že zvládne generovat řídicí sekvence pro CCD snímač i AFE, ještě získaná data ze snímače před odesláním do DSP upraví nebo přefiltruje. Výhodou videoprocessoru v hradlovém poli na rozdíl od mikroprocesoru je to, že může provádět spoustu operací souběžně. Na co procesor potřebuje několik instrukcí, zvládne FPGA v jednom jediném strojovém cyklu.

2.5 Volba rychlého rozhraní

Dalším bodem této diplomové práce je navrhnout a realizovat rozhraní pro rychlý přenos dat mezi nadřazeným PC a měřicím modulem. Kriteria pro výběr rozhraní jsem si stanovil takto:

1. Rozhraní musí být průmyslový standard bez licenčních poplatků.
2. Minimální rychlost by měla být kolem 100 megabitů/s, aby se v reálném čase stíhala přenášet data z AD převodníku do PC.
3. S rozhraním se musí dát jednoduše programově komunikovat jak na straně PC, tak na straně zařízení.
4. Případné specializované obvody pro konstrukci musí být sehnatelné v ČR.

Z těchto požadavků vyplývá možnost použít tyto standardní rozhraní:

USB 2.0 High-speed

Sériový poloduplexní přenos dat, stromová topologie, maximální délka kabelu 5m.

Výhody:

- Vysoká přenosová rychlost 480 Mbit/s.
- Jednoduchá kabeláž, snadno dostupné konektory.
- Velice rozšířený a perspektivní interface, dnes již v každém PC.

Nevýhody:

- Nutnost programovat vlastní ovladač pro PC.
- Složitá architektura standardu USB.

- Neexistence jednoduchého obvodového kontroléru.

FireWire nebo i.Link - IEEE1394

Sériový poloduplexní přenos dat, stromová topologie, maximální délka kabelu 4,5m. Tento interface má v podstatě velmi podobné vlastnosti jako USB, má tedy i podobné výhody a nevýhody.

Výhody:

- Vysoká přenosová rychlost 100, 200, nebo 400 Mbit/s.
- Rozhraní přímo stavěné k připojení zařízení na snímání obrazu.
- Rozšířený a snadno dostupný interface.
- Snadno dostupné obvodové kontroléry.

Nevýhody:

- Nutnost programovat vlastní ovladač pro PC.
- Složitá architektura standardu IEEE1394.

CameraLink

Vícekanálový jednosměrný paralelní, 7:1 serializovaný přenos pomocí LVDS, spojení point-to-point, volitelně 1, 2, nebo 3 paralelní kanály, délka kabelu až 10 metrů.

Výhody:

- Velmi vysoká přenosová rychlost přes 2 Gbit/s.
- Standard rozšířený mezi výrobci měřicích kamer.
- V normě definována pouze fyzická vrstva a aplikační rozhraní. (také nevýhoda)
- Nejjednodušší hardwarová implementace ze všech uvedených

Nevýhody:

- V normě definována pouze fyzická vrstva a aplikační rozhraní.
- Nutnost instalace speciální rozšiřující karty do PC s tímto rozhraním.
- Drahé a těžko dostupné kabely a konektory.
- Nutnost programovat vlastní ovladač a DLL knihovnu.

Fast Ethernet - 100Base-TX

Sériový plně duplexní přenos, stromová topologie, délka kabelu až 100 metrů.

Výhody:

- Velice rozšířený a snadno dostupný standard.
- Volně dostupné veškeré informace a dokumenty.
- Možnost přímého spojení s Internetem.
- Ze strany PC není potřeba programovat žádné ovladače.
- Jednoduchá a srozumitelná struktura komunikačních protokolů.

- Osobní zkušenosti s řadičem LAN91C111.

Nevýhody:

- Rychlost na hranici přípustnosti, 100Mbit/s.
- Malé množství univerzálních obvodových řadičů na trhu.
- Vysoká cena řadiče LAN91C111.
- Přenosový protokol musí být kompletně řešen softwarově.

Z výše uvedených alternativ jsem nakonec vyhodnotil Fast Ethernet jako ten, který půjde nejnadhěji implementovat. Tuto volbu nezapřičinily ani tak jeho výhody, ale to že se u něj nevyskytují poměrně závažné nevýhody, jako u ostatních rozhraní.

Pro realizaci jsem vybral řadič od společnosti SMSC - LAN91C111, kvůli jeho snadnému začlenění do zařízení, jednoduchosti obsluhy a hlavně kvůli osobním praktickým zkušenostem s tímto obvodem.

2.6 Možnosti použití

Vyvinuté zařízení bude mít dostatečně široké možnosti pro implementaci složitých videometrických systémů. Například jej bude možné použít v čokoládovně k počítání různě tvarovaných nebo i různě barevných bonbónů, které pojedou po běžícím pásu. Jednotka by mohla počítat kusy jednotlivých tvarů či barev a ještě varovat při výskytu defektního bonbónu.

Jinou možností je využít jednotku jen jako kontrolní bod. Jednotka s připojeným obrazovým snímačem a objektivem může kontrolovat přední skla automobilů, které pojedou na výrobní lince. V případě výskytu praskliny, rýhy, či jiného defektu ve skle, může vyslat signál a vadné sklo označit, nebo na něj aspoň upozornit.

Podobný videometrický systém je již v provozu v jedné třídírně obilí. Na klasu žita totiž může parazitovat houba - Paličkovice Nachová. Zrna obilniny se pak v důsledku napadení houbou mění v křivé lusky přilepené na klasu – tzv. námely, které jsou oproti normálnímu zrnu podstatně delší a jsou prudce jedovaté. Tyto námely se tedy musí ze žita odstraňovat a právě k tomu je systém použit. Pod koncem běžícího pásu, po kterém jedou v jedné vrstvě rozprostřená zrnka žita, je řádkový obrazový snímač s objektivem a pod ním je vodorovně umístěna sada vzduchových trysek. Videometrický systém vyhodnocuje obraz padajícího žita a vyskytne-li se v jednom místě námel, aktivuje se příslušná vzduchová tryska a nežádoucí námel odfoukne.

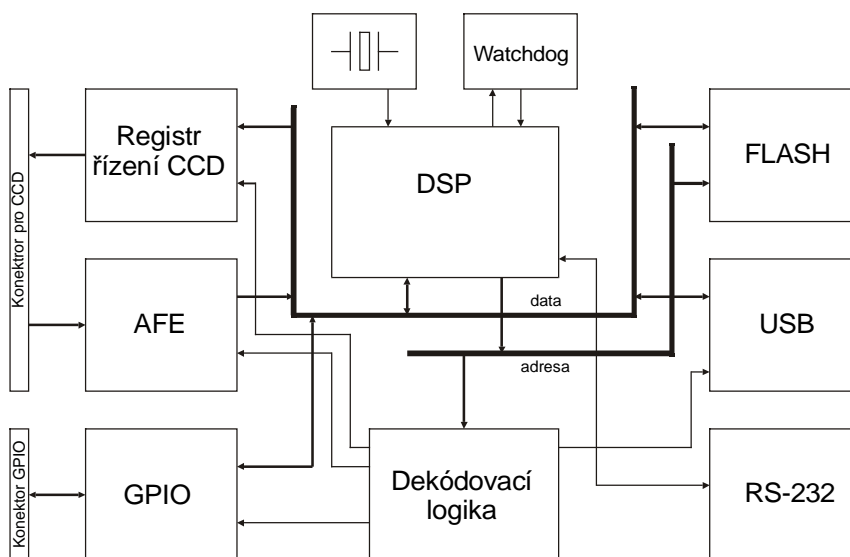
3. ŘÁDKOVÁ KAMERA ULC85

Jak již bylo řečeno, můj první úkol je návrh univerzální jednotky pro řádkové snímače, která bude dělat veškeré řízení softwarově. V návrhu jsem vycházel z řádkové kamery ULC84 v [1], kde je pro řízení CCD použit signálový procesor firmy Analog Devices ADSP-2184 a komunikace s PC probíhá prostřednictvím paralelního portu. Signálový procesor ADSP-2184 je však v dnešní době zastaralá a také příliš drahá součástka, proto jsem se rozhodl použít ve svém zapojení novější, rychlejší a levnější typ ADSP-2185M, který má také nižší napájecí napětí 3,3V.

Také paralelní port je dnes již výběhový a u nejnovějších PC se již nevyskytuje. Jako jeho náhradu jsem zvolil perspektivní rozhraní USB, které se dá na úrovni hardware realizovat velice jednoduše použitím integrovaného řadiče firmy Future Technology Devices International (FTDI).

Vyvinuté zařízení má také sloužit jako náhrada výše zmíněné řádkové kamery ULC84 ve cvičeních na katedře měření.

3.1 Blokové schéma



Obr. 3.1 Blokové schéma ULC85

3.2 Popis jednotlivých funkčních bloků ULC85

3.2.1 Řídící procesor

Jádro jednotky tvoří signálový procesor ADSP-2185MKST firmy Analog Devices. Je to novější typ s napájecím napětím 3,3 V pro sběrnice a signály a 2,5 V pro jádro. Hodinový kmitočet generuje externí 36 MHz krystal. Ten se pak uvnitř procesoru násobí pomocí PLL syntézy na 288 MHz. Vykonání jedné instrukce trvá čtyři hodinové cykly, výkon procesoru je tedy 72 MIPS.

Procesor při startu načítá svou vnitřní paměť programu z paměti FLASH o velikosti 512 KB, rozdělené na 4 stránky po 128 KB. Tento proces se také nazývá bootování. Mezi stránkami paměti se volí pomocí jumperů JP1, JP2. Zvolenou stránku je pak možné v aplikaci přeprogramovat, pokud je to povoleno jumperem JP4. V první stránce paměti FLASH je nahrána upravená verze programu „monitor“ pro procesor ADSP-2181 z vývojového kitu EZ-KIT LITE. Ten po inicializaci umožňuje po sériové lince nahrát a spustit uživatelskou aplikaci stejně, jako by se nahrála přímo z FLASH. Běh monitoru je signalizován blikáním červené LED, připojené na pin FL1. Tuto LED je pak možno využívat i ve vlastní aplikaci k různým diagnostickým účelům.

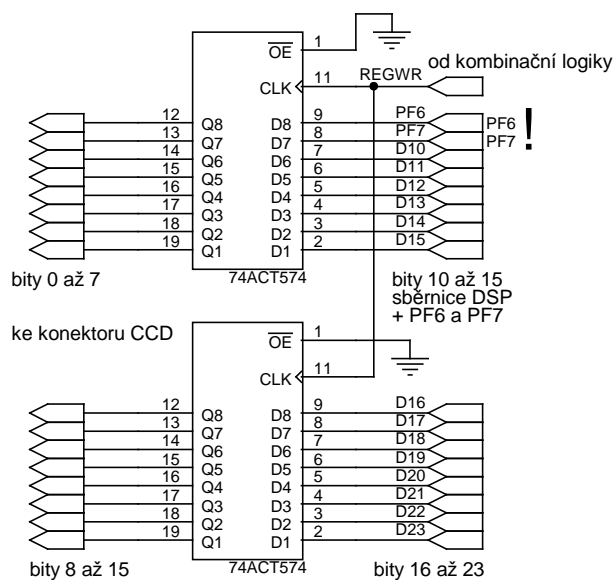
Alternativně k ADSP-2185M lze osadit jiný procesor z řady ADSP-218xM, podle zvoleného kompromisu mezi cenou a velikostí paměti RAM integrované na čipu (viz tab. 3.1).

Tab. 3.1 Porovnání procesorů řady ADSP218x

Označení	Program RAM	Data RAM
ADSP2186M	8K (24KB)	8K (16KB)
ADSP2185M	16K (48KB)	16K (32KB)
ADSP2189M	32K (96KB)	48K (96KB)
ADSP2188M	48K (144KB)	56K (112KB)

3.2.2 Registr řízení CCD

Registr řízení CCD je rychlý výstupní 16bitový registr a je primárně určen k ovládání hradel a posuvných registrů v CCD snímačích a k ovládání AFE. 14 bitů ze 16ti je připojeno přes bránu přímo na datovou sběrnici DSP a je tedy možné na nich nastavovat logické úrovně přímo jedinou instrukcí. Zbýlé dva bity jsou na bránu přivedeny z vývodů PF6 a PF7 u DSP. Zápisem na výstupní bránu se horních 14 bitů nastavuje přímo argumentem instrukce a spodní dva bity se kopírují z PF6 a PF7, které je nutno předtím nastavit. Je to poplatek za to, že můžeme jedinou instrukcí synchronně ovládat řídicí signály CCD snímače současně s řídicími signály CDSCLK1, CDSCLK2 a ADCCLK u AFE a zároveň využijeme všech 16 bitů výstupního registru.



Obr. 3.2 Zapojení výstupních budičů na ULC85

Propojení je patrné z obr. 3.2. Třetí bit (druhý počítáno od nuly) je zapojen současně na AFE jako signál ADCCLK a současně na výstupní bránu. Je to z toho důvodu, že se signál ADCCLK dá využít i na straně CCD snímače jako hodinový signál. Zápis do výstupní brány trvá buď jeden instrukční cyklus (13,9 ns), nebo dva (27,8 ns) pokud se nastaví navíc jeden čekací cyklus na DSP. Zápis na piny PFX trvá vždy jeden cyklus. Výstupní napěťové úrovně jsou TTL 5V. V případě že CCD snímač požaduje na řídicích signálech jiné napětí, je potřeba použít externí budiče.

Rychlé číslicové výstupy vyžadují použití rychlých obvodů. Když jsem ověřoval funkčnost celého zapojení na nepájivém kontaktním poli, měl jsem k dispozici obvody 74AHCT574 a 74ACT574. Obvod typu AHCT vykazoval stejné zpoždění signálu procházejícího skrz budič jako obvod typu ACT. Konkrétně to bylo 5,5 ns. AHCT však nebyl schopen vždy dostatečně rychle reagovat na zapisovací impuls a docházelo k tomu, že se data do registru vůbec nezapsala. Obvod technologie ACT takovéto problémy neměl, a proto jsem do zapojení použil právě tento typ.

3.2.3 Analogové vstupy + AFE

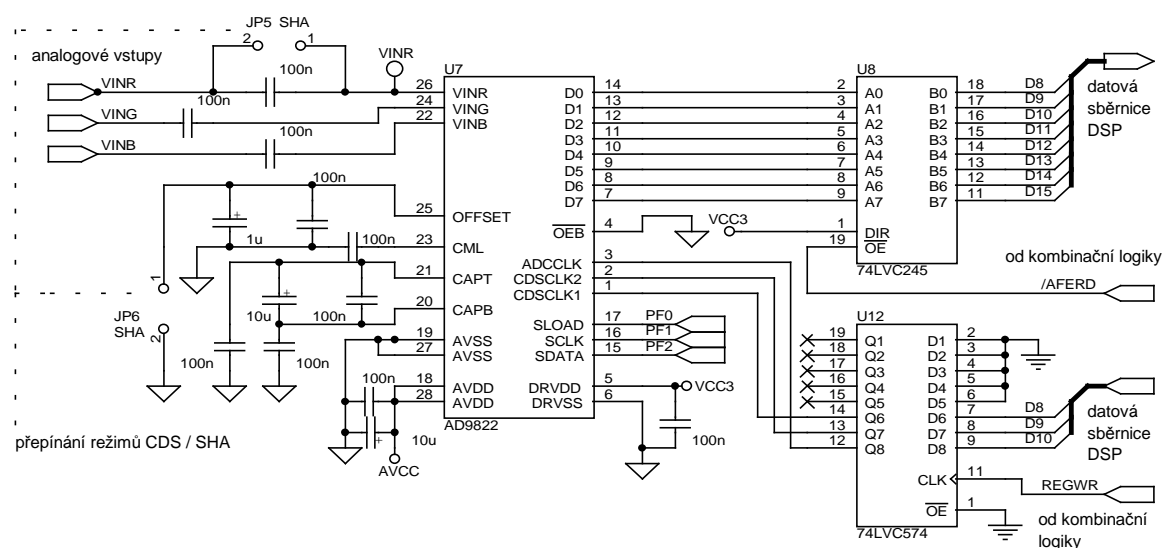
Zařízení má také možnost videosignál z obrazového snímače rovnou digitalizovat. Tři analogové vstupy jsou převáděny do číslicové podoby 14ti bitovým A/D převodníkem se sdílením času (pipe-line) pro CCD snímače AD9822, zvaným Analog Front-End (AFE).

Převodník má dva základní provozní režimy: CDS (korelované dvojité vzorkování), nebo SHA (jednoduché standardní vzorkování). Režim SHA se volí zkratováním konfigurační propojky (dále jumperu) JP6 ze strany hardwaru a vynulováním bitu 4 v konfiguračním registru AFE. V případě, že se používá režim SHA, je ještě nutné překlenout vazební kondenzátor na vstupu AFE, což se provede pomocí jumperu JP5.

AFE lze dále nastavit jako jednokanálový, nebo tříkanálový. V jednokanálovém režimu se vzorkuje pouze vstup který je vybraný registrem MUX, zatímco ve tříkanálovém režimu se navzorkují všechny 3 vstupy současně a pak se postupně převádí pomocí AD převodníku ve volitelném pořadí R-G-B, nebo B-G-R. Vstupní napěťový rozsah každého ze vstupů je 2 V špička-špička, a to v obou režimech. U vstupů je též možno nezávisle na sobě nastavovat offset v rozsahu ± 350 mV a zisk $0 \div 15$ dB.

Na obr. 3.3 je detailně zobrazeno připojení analogové části k DSP. Data z AFE jsou neustále vybavena (signál #OEB je v nule) a jsou oddělena budičem sběrnice U8, který je teprve připojen na datovou sběrnici DSP. Kdyby totiž byla datová sběrnice připojená k AFE přímo, tak by se veškerá data, která po ni putují, dostávala až dovnitř pouzdra AFE a mohla by zbytečně způsobovat rušení jeho analogové části.

Řízení AFE probíhá zápisem do registru řízení CCD (viz kapitola 3.2.2), jehož nejnižší dva bity jsou připojeny na CDSCLK1 a CDSCLK2 a zbylých 14 bitů jde přes budiče na konektor pro CCD snímač. Třetí bit (druhý, počítáno od nuly) je veden jak na konektor pro CCD snímač, tak na signál ADCCLK.

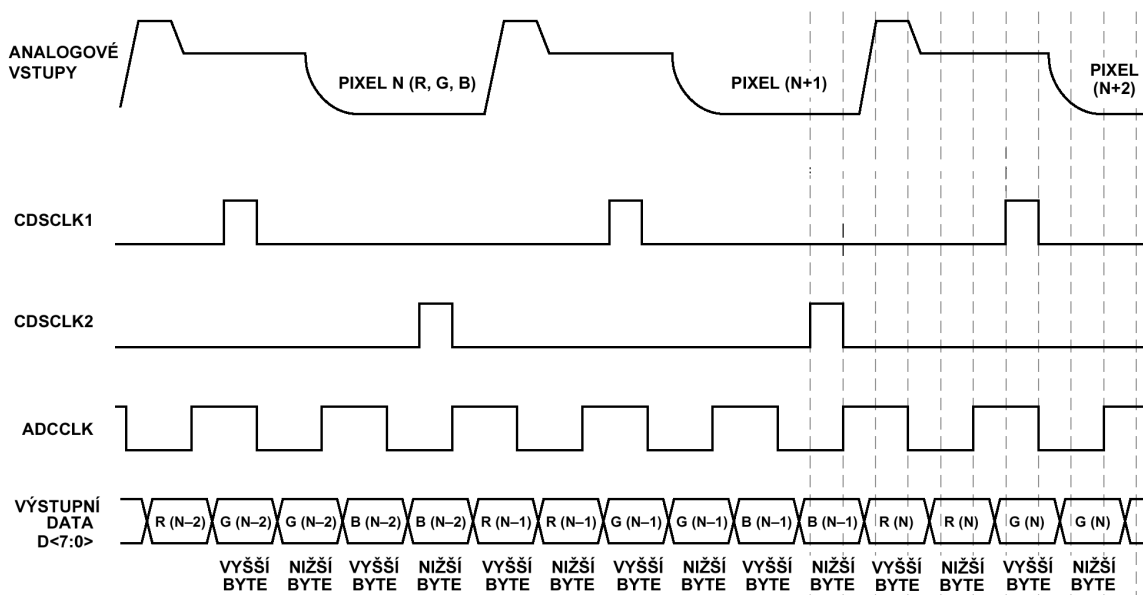


Obr. 3.3 Připojení AFE ke sběrnici procesoru ADSP-2185M

Testováním tříkanálového režimu AFE jsem zjistil, že není potřeba vyčítat všechny tři převedené hodnoty z datového registru, jak je naznačeno na obr. 3.4, ale stačí vyčíst dva nebo i jenom jeden vzorek. Interní multiplexer totiž nejspíše po příchodu impulsu na CDSCLK2 začíná s výběrem vstupů opět od začátku sekvence. Časování CDSCLK1 a 2 se vůči analogovému signálu nezmění, pouze se sníží hodinová frekvence ADCCLK ze trojnásobku na dvojnásobek frekvence pixelů.

Naskytá se tak možnost provozovat AFE v nedokumentovaném dvoukanálovém režimu, což se může hodit pro snímače se dvěma výstupy, jako je např. CCD191 od Fairchild Imaging. Tuto alternativu jsem však dále netestoval a je také možné, že jsem během

pozorování udělal nějakou chybu. Před použitím této metody tedy důrazně doporučuji vše potřebné důkladně a znovu vyzkoušet.



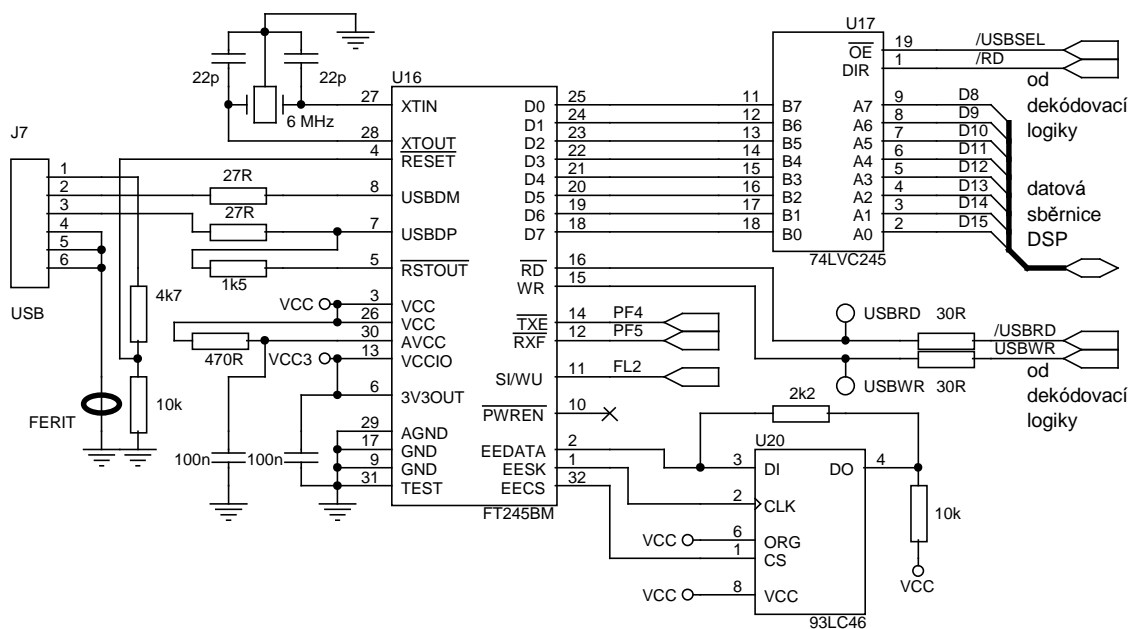
Obr. 3.4 Časování AFE pro tříkanálový režim

3.2.4 Rozhraní USB

Pro styk s nadřazeným PC je použit řadič sběrnice USB kompatibilní s USB 2.0 a USB 1.1, pracujícím ve full speed isochronním nebo bulk režimu. Využívá se např. pro přenos snímků, naměřených údajů, konfigurace, či ovládacích povelů.

Z aplikačního hlediska se USB řadič tváří jako paměť FIFO, čili k implementaci přenosu není potřeba znát nic o USB sběrnici jako takové. Řadič je z vnějšku (po sběrnici USB) nastavitelný a konfigurace se ukládá do sériové paměti EEPROM. Odchozí buffer má velikost 384 bytů, příchozí pak 128 bytů.

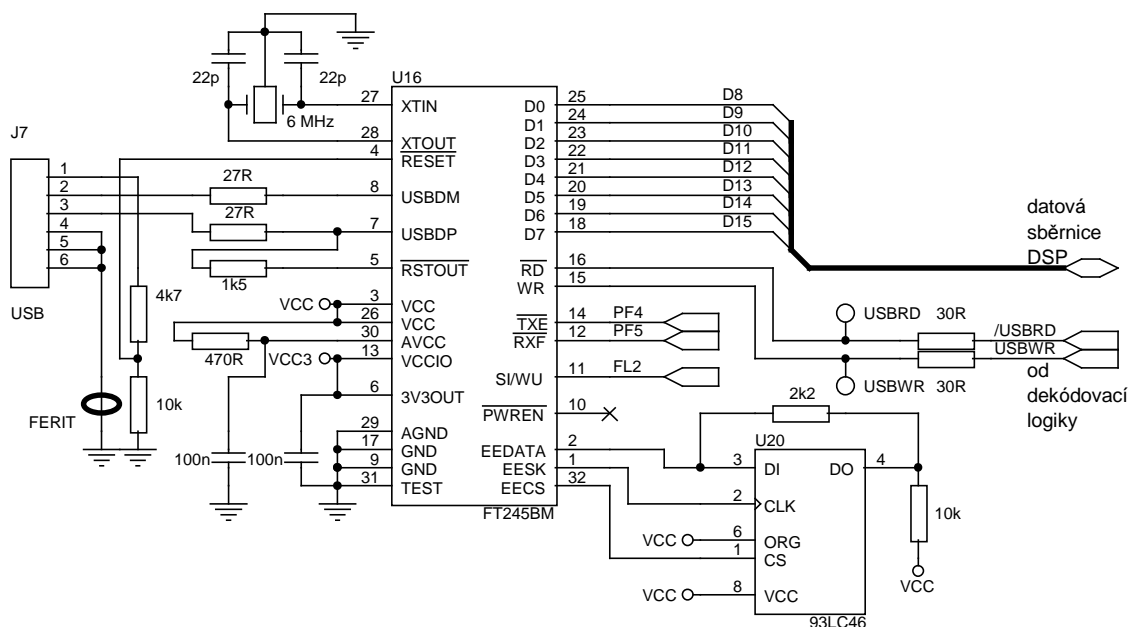
V případě že se odesílá pouze malé množství dat a buffer se nezaplní, obvod čeká na jeho zaplnění. Pokud se buffer nezaplní do doby určené TX Buffer Timeout, která je programovatelná na $1 \div 255$ ms, obvod odešle data až po uplynutí této doby. To může mít za následek snížení propustnosti. V tomto případě lze využít signál SI (send immediately – odeslat okamžitě), který je připojen na pin FL2 u DSP. Impuls vyslaný na tento pin vyvolá odeslání dat v bufferu okamžitě bez čekání na jeho zaplnění.



Obr. 3.5 Zapojení řadiče sběrnice USB na ULC85

Na obr. 3.5 je znázorněno připojení řadiče USB ke sběrnici DSP. Při testování zapojení na nepájivém kontaktním poli jsem zjistil, že řadič USB FT245BM (U16) nemá na čipu příliš silný budič datové sběrnice. Po aktivaci signálu /RD mu trvalo přibližně 100 ns, než sběrnici vybudil do požadované úrovně. Právě proto byl na datovou sběrnici hned za U16 zařazen doplňkový budič U17, který měl slabý budič na čipu FT245BM posílit.

Během oživování desky jsem však zjistil, že jsem udělal chybu v návrhu zapojení. Signál /USBSEL měl vybavovat výstup obousměrného budiče U17 ve chvíli, kdy se mělo komunikovat s USB řadičem. /USBSEL však přechází do aktivní nulové úrovně pouze na základě dekodování adresy a nezahrnuje v sobě signál /IOMS, který vybírá IO adresový prostor. Docházelo tedy k vybavení budiče i při čtení z paměti FLASH a tím pádem ke kolizi na datové sběrnici. Problém se nakonec vyřešil překlenutím budiče drátovými propojkami, protože jsem zjistil, že na DPS není budič vůbec potřeba. Datová sběrnice na DPS má totiž menší kapacitu, než měla na nepájivém poli a U16 ji zvládá vybudit sám. Koncové zapojení je na obr. 3.6.



Obr. 3.6 Zapojení řadiče USB po úpravě

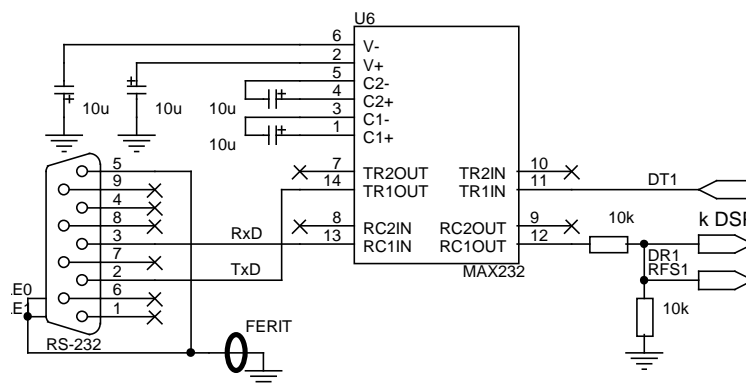
3.2.5 Linka RS-232

Signálové procesory řady ADSP-218x bohužel nemají na čipu integrovaný asynchronní sériový kanál (UART). Je tedy nutné ho částečně softwarově, částečně hardwarově emulovat (viz [9]). K emulaci je použit synchronní kanál SPORT, kterým je vybaven každý procesor řady ADSP-218x. Z aplikačního hlediska je opět vše transparentní, k dispozici jsou subrutiny na odeslání znaku, přijetí znaku a dotaz na stav přijímacího bufferu. Detailně je emulace popsána v kapitole 6.1.2.

Parametry přenosu jsou pevné – start-bit, 8 datových bitů, jeden stop-bit bez parity. Přenosová rychlost se nastavuje během kompilace zdrojového kódu i během vykonávání programu (viz kapitola 6.1.2).

Toto řešení s sebou nese i stinné stránky. Pokud je emulace sériové linky zapnutá, vykonává se periodicky přerušení od SPORTu, a to s frekvencí rovnající se dvojnásobku baudové rychlosti emulovaného sériového kanálu. Toto nepřetržité přerušování běhu procesoru může být, obzvláště v časově kritických úlohách, nežádoucí. Proto je vhodné sériovou linku používat jen v případě nutnosti, nebo emulaci během nečinnosti vypínat.

Obvodové zapojení budiče linky je na obr. 3.7.



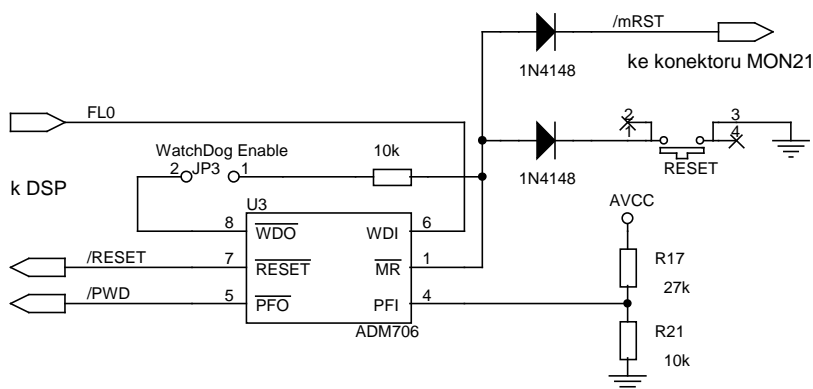
Obr. 3.7 Zapojení budiče sérové linky na ULC85

3.2.6 Watchdog – dohlížecí obvod

Dohlížecí obvod, nebo jak se mu také někdy říká „Watchdog“, má za úkol dohlížet na správnou funkci celého zařízení. Hlíďají se výpadky a poklesy napájení, nebo nekonečné zacyklení programu. Další jeho funkcí je tvarování signálu z resetovacího tlačítka.

Jestliže poklesne napájecí napětí 3,3V pod kontrolní mez (3,08V), generuje se signál RESET a drží se tak dlouho, dokud trvá pokles napětí. Po opětovném vzrůstu napětí nad kontrolní mez, se po 200ms zpoždění přestane držet DSP v resetu a obnoví se původní činnost.

Při poklesu napájecího napětí 5V pod hranici 4,625V se generuje nemaskovatelné přerušování PWD signálového procesoru, které může ještě před úplným výpadkem napájení desky, dokud jsou ještě nabitě filtrační a blokovací kondenzátory, stihnout vykonat nějakou důležitou činnost. Toto mezní napětí se nastavuje odpory R17 a R21, které tvoří napěťový dělič jak je vidět na obr. 3.8. Prahové napětí vstupu PFI je 1,25V a podle toho je potřeba dělič navrhnout.



Obr. 3.8 Zapojení dohlížecího obvodu na ULC85

Další zabezpečení spočívá ve sledování běhu procesoru. Na signálu WDI, který je připojen na pin FL0 u DSP, musí být nepřetržitě překlápěna logická úroveň. V případě že se

WDI ocitne ve stejném stavu po dobu delší než 1,6 s, generuje se resetovací impuls pro DSP. Aby se toto zajistilo, vloží se do vhodných míst programu, kde se běh neustále opakuje, instrukce pro aktivaci a deaktivaci signálu FL0. Běží-li program tak jak má, překlápí se signál WDI podle instrukcí v programu. Pokud se ale běh programu někde zacyklí v nekonečné smyčce, přestane se signál WDI překlápět a procesor se do 1,6 s resetuje. Aby však toto zabezpečení fungovalo, musí se nejprve povolit propojením jumperu JP3.

Zkušenosti ukazují, že není vhodné používat instrukci pro překlopení stavu FL0 (`toggle FL0`), protože může dojít k zacyklení právě přes tuto jednu instrukci a celé zabezpečení pak postrádá smysl.

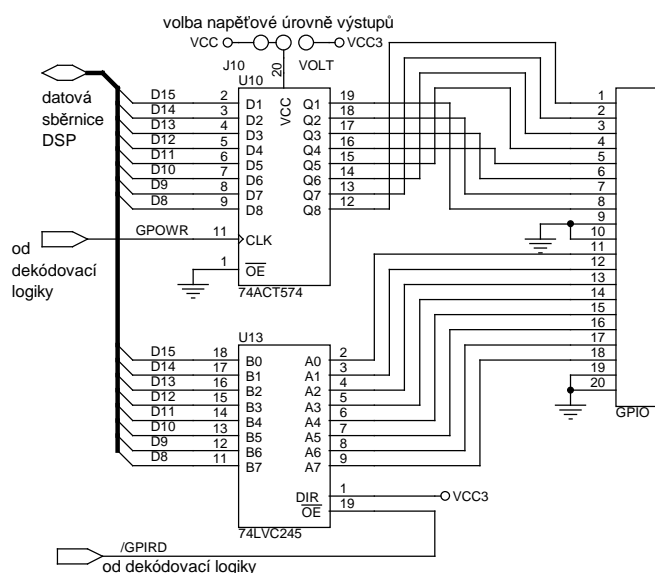
3.2.7 Uživatelské vstupy/výstupy

Uživatelské vstupy a výstupy (dále jen GPIO) tvoří další část komunikačního rozhraní s okolím. Na datovou sběrnici jsou připojeny stejným způsobem jako registr řízení CCD (viz kapitola 3.2.2), čili jejich rychlost je s ním srovnatelná. Alternativně lze GPIO použít jako další řídicí signály pro ovládání a čtení CCD, nebo k připojení CMOS snímače.

Vstupy ani výstupy nejsou galvanicky oddělené od napájení na desce, nejsou ani chráněny proti přepětí nebo elektrostatickému výboji. Vstupy jsou 5V tolerantní a kompatibilní s TTL, LVTTTL a CMOS. Rozhodovací úroveň je 1,8V.

Výstupy jsou volitelně buď HCMOS 5V, nebo LVTTTL 3,3V. Mezi těmito dvěma napěťovými úrovněmi se volí osazením U10 jako 74LVC574 pro 3,3V a 74ACT574 pro 5V. Propojkou J10, která je implicitně propojená na 5V přímo na DPS, je ještě nutné nastavit správné napájecí napětí pro zvolený obvod.

Úplné zapojení GPIO je na obr. 3.9.



Obr. 3.9 Uživatelské vstupy a výstupy (GPIO) na ULC85

3.2.8 Dekódovací logika pro periferie

Vnější periferie jsou připojeny do IO adresového prostoru procesoru na adresy podle tab. 3.2. Dekódování adres se děje pomocí standardních logických obvodů typu LVC pro 3,3V logiku.

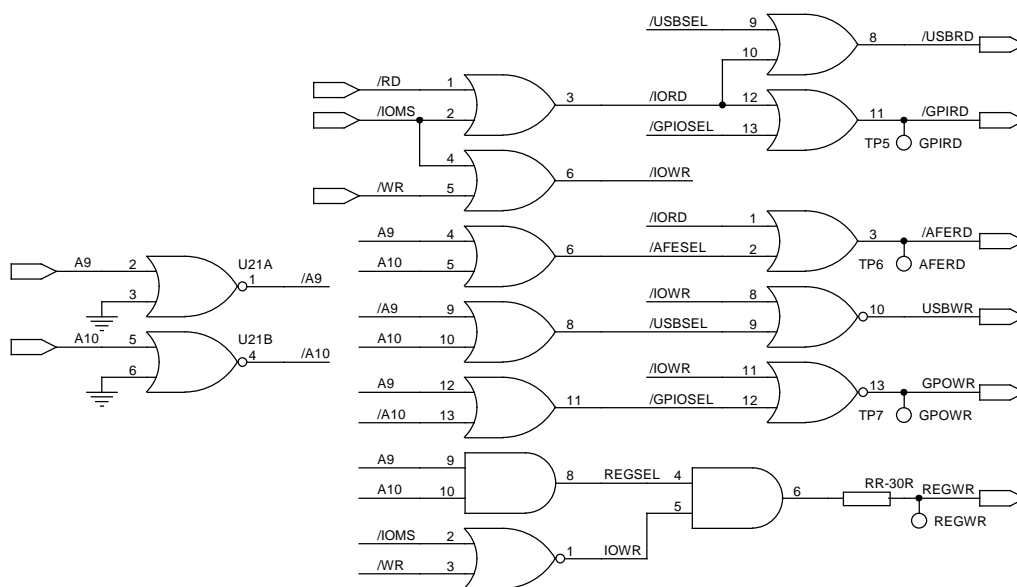
Počet hradel, přes které musí adresové a řídicí signály projít, než se dostanou k vlastní periférii, je optimalizován tak, aby rychlé periferie měly tuto hloubku co nejnižší. Registr pro řízení CCD a AFE mají tuto hloubku 2 hradla a ostatní periferie u kterých nezáleží na rychlosti, jako je USB a GPIO, mají hloubku 3 hradla.

K dekódování jsou použity pouze adresové vodiče A9 a A10, zatímco na A0 až A8 nezáleží. To způsobuje, že se periferie dá adresovat v celém rozsahu 512 adres, pro které jsou A9 a A10 konstantní. Kromě jednoduchosti dekódovací logiky to má ještě jednu výhodu. U adresových prostorů dle tab. 3.2 se dá u DSP nezávisle na sobě nastavit rychlost přístupu pomocí čekacích cyklů.

Tab. 3.2 Mapování periferií ULC85 do IO prostoru DSP

Periferie	Adresa	Směr
Analog Front-End	0x000 – 0x1FF	Čtení
USB interface	0x200 – 0x3FF	Čtení / Zápis
Uživatelské vstupy/výstupy (GPIO)	0x400 – 0x5FF	Čtení / Zápis
Registr pro řízení CCD	0x600 – 0x7FF	Zápis

Pro úplnost uvádím na obr. 3.10 detailní zapojení dekódovací logiky tak, jak je na DPS.



Obr. 3.10 Detailní zapojení dekódovací logiky ULC85

3.2.9 Konektor pro připojení CCD snímače

Konektor pro připojení CCD snímače je 40 pinová dvouřadá lišta pro konektory typu PFH. Zapojení konektoru pro připojení snímačové desky je na obr. 3.11. Piny OUT0 až OUT15 jsou rychlé číslicové výstupy z registru pro řízení CCD v úrovni 5V TTL. OUT0 a OUT1 jsou určené pro ovládání časové nekritických signálů, protože k jejich změně jsou potřeba dvě instrukce procesoru, zatímco ke změně ostatních stačí instrukce jedna (viz kapitola 3.2.2).

VCC3,3 a VCC5 jsou výstupy napájení 3,3V a 5V. Z těchto pinů je možno napájet případnou logiku na snímačové desce. Některé CCD snímače vyžadují ještě jiné napájecí napětí, než je 5V. Z tohoto důvodu vedou na konektor pro snímač signály VCCA a VCCB, což jsou uživatelské napájecí linky, přivedené přímo z napájecího konektoru ULC85. Těchto signálů lze také využít pro napájení ULC85 ze snímačové desky, když se v napájecím konektoru např. VCCB propojí s +5V. Také je možné ULC85 napájet přímo do signálu VCC5, ale to příliš nedoporučuji, protože tento signál je připojen k analogovému napájení a to je spojené s digitálním jen přes tenký spoj, protože se nepředpokládá z analogového napájení příliš velký odběr.

VINR, VING, VINB jsou analogové vstupy pro červenou, modrou a zelenou složku videosignálu, vedeného do AD převodníku. Dle jejich koncových písmen lze určit, který k jaké barvě náleží (**R**ed, **G**reen, **B**lue). Signál VINR lze jako jediný provozovat v obou režimech CDS i SHA.

Všechny ostatní piny jsou připojené na nulový potenciál GND.

GND	1	2	OUT0
OUT1	3	4	GND
OUT2	5	6	OUT3
GND	7	8	OUT4
OUT5	9	10	GND
OUT6	11	12	OUT7
GND	13	14	OUT8
OUT9	15	16	GND
OUT10	17	18	OUT11
GND	19	20	OUT12
OUT13	21	22	GND
OUT14	23	24	OUT15
GND	25	26	GND
VCC3,3	27	28	VCC5
VCCA	29	30	VCCB
GND	31	32	GND
VINB	33	34	GND
VING	35	36	GND
VINR	37	38	GND
GND	39	40	GND

Obr. 3.11 Zapojení konektoru pro CCD snímač na ULC85

4.SP2EDB – VÝVOJOVÁ DESKA S FPGA SPARTAN-2E

4.1 Účel zařízení

Deska je primárně určena k tomu, abych se na ni naučil pracovat s hradlovým polem FPGA. Bylo také potřeba ověřit algoritmus načítání konfigurace do FPGA a ověřit správnost zapojení zvláštního napájecího zdroje pro napájení jádra FPGA. Deska je ale také navržena tak, aby se k ní dal připojit CCD snímač stejně, jako k ULC85, a daly se na ni odzkoušet i algoritmy řízení CCD, generované přímo hradlovým polem.

4.2 Výběr hradlového pole

Následoval výběr vhodného hradlového pole. V různých pracích na katedře měření byla použita hradlová pole značek Altera a Xilinx, proto jsem se rozhodl nehledat mezi jinými výrobci a zaměřil jsem se právě na tyto dva. Porovnání různých typů je v tab. 4.1. LE značí počet logických elementů, EH počet ekvivalentních hradel, I/O počet vstupně/výstupních uživatelských pinů, Pouzdro značí nejběžnější pouzdro, V_{cc} je napájecí napětí a Or. Cena je hrubá orientační cena jednoho kusu získaná z různých, převážně zahraničních zdrojů tam, kde se jí podařilo zjistit.

Tab. 4.1 Porovnání hradlových polí značek Altera a Xilinx

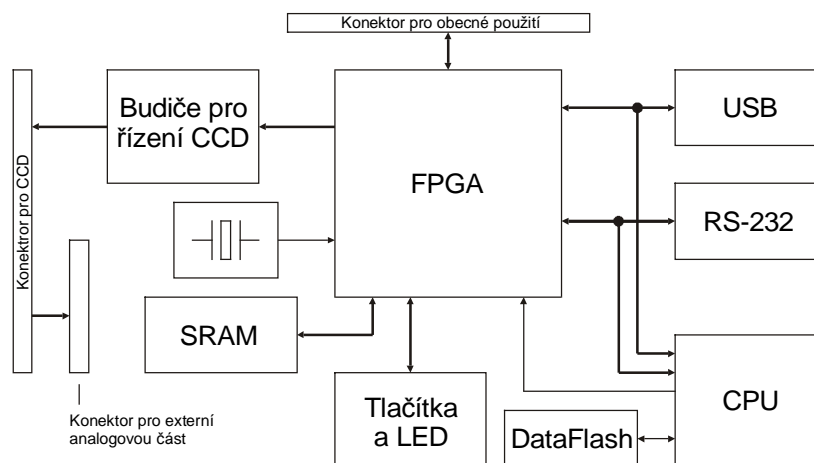
Výrobce	Typ	LE	EH	I/O	Pouzdro	V _{cc}	Or. cena
Altera	EP1K10-3	576	10 000	66	TQFP100	2,5	324
Altera	EP1K10-3	576	10 000	92	TQFP144	2,5	379
Altera	EP1K10-3	576	10 000	120	PQFP208	2,5	433
Altera	EP1K30-3	1728	30 000	102	TQFP144	2,5	541
Altera	EP1K30-3	1728	30 000	147	PQFP208	2,5	622
Altera	EP1K50-3	2880	50 000	102	TQFP144	2,5	758
Altera	EP1K50-3	2880	50 000	147	PQFP208	2,5	839
Altera	EP1K100-3	4992	100 000	147	PQFP208	2,5	1163
Altera	EP1C3	2910	N/A	65	TQFP100	1,5	551
Altera	EP1C3	2910	N/A	104	TQFP144	1,5	629
Altera	EP1C6	5980	N/A	98	TQFP144	1,5	1023
Altera	EP1C6	5980	N/A	185	PQFP240	1,5	1102
Xilinx	XC2S50	1728	50 000	92	TQFP144	2,5	477
Xilinx	XC2S50	1728	50 000	140	PQFP208	2,5	
Xilinx	XC2S100	2700	100 000	140	PQFP208	2,5	
Xilinx	XC2S200	5292	200 000	140	PQFP208	2,5	1119
Xilinx	XC2S50E	1728	50 000	146	PQFP208	1,8	
Xilinx	XC2S100E	2700	100 000	146	PQFP208	1,8	
Xilinx	XC2S200E	5292	200 000	146	PQFP208	1,8	
Xilinx	XC2S300E	6912	300 000	146	PQFP208	1,8	1530

Z uvedených alternativ jsem nakonec zvolil hradlové pole značky **Xilinx Spartan-IIE, XC2S300E**. Hlavním důvodem byl vysoký počet logických elementů (nejvyšší v tabulce) a také to, že jsem s programovatelnými obvody firmy Xilinx měl v té době již jisté zkušenosti.

Pro značku Xilinx jsem se rozhodl také kvůli vývojovému prostředí. ISE WebPack, vytvořené přímo firmou Xilinx, poskytuje licenci permanentní, zatímco konkurenční Quartus-II pro hradlová pole firmy Altera, má licenci pouze časově omezenou a po jejím vypršení se musí výrobce žádat o novou. Hrozí tak nebezpečí, že když třeba firma Altera zkrachuje, nebo spíše když se rozhodne přestat zdarma poskytovat licence, celé vyvinuté zařízení by rázem ztratilo na použitelnosti. Také po stručném seznámení se s prostředím Quartus-II, jsem měl pocit, že vývojové prostředí Xilinx ISE je přeci jenom na první pohled srozumitelnější a nechybí ani příklady, předlohy a šablony pro psaní v jazyce VHDL i Verilog, které jsem v Quartusu nenašel.

Již jsem se zmínil o programovacích jazycích VHDL a Verilog. Správně bych měl mluvit o popisovacích nebo konfiguračních (descriptive) jazycích, protože výsledkem překladu zdrojového textu není program, ale konfigurace – popis. Všechny jazyky k popisu hardwaru jsou souhrnně označovány zkratkou HDL (Hardware Description Language). Psaní v jakémkoli HDL se výrazně liší od psaní v programovacím jazyce a je také, alespoň z počátku, složitější se v něm orientovat. Z tohoto důvodu, a také proto, že Spartan-IIE je v laboratoři nová součástka, jsem vyvinul vývojovou desku s tímto hradlovým polem, SP2EDB.

4.3 Blokové schéma

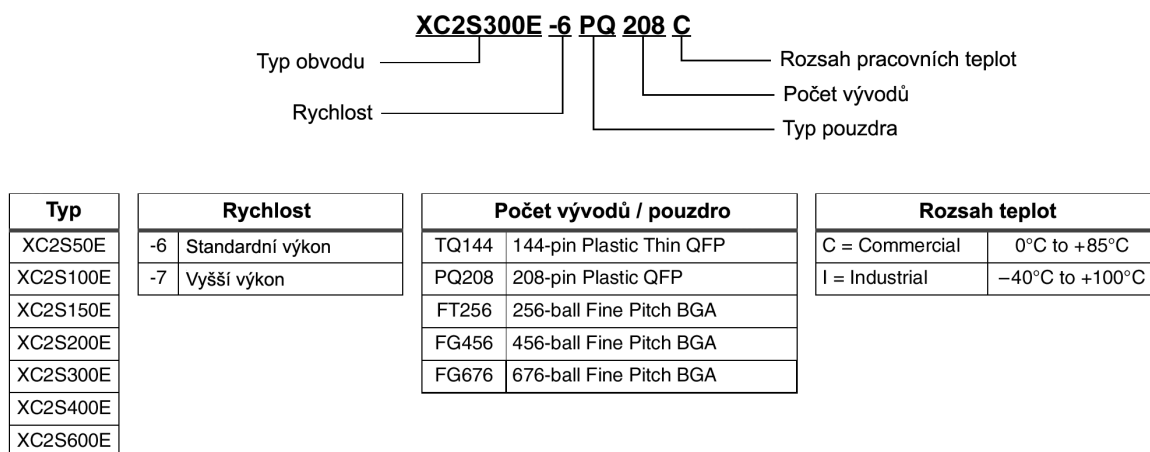


Obr. 4.1 Blokové schéma SP2EDB

4.4 Popis jednotlivých funkčních bloků

4.4.1 Hradlové pole FPGA

Jak již bylo popsáno v kapitole 4.2, použité hradlové pole je Xilinx Spartan-IIE, s úplným typovým označením XC2S300E-6PQ208C. Základní parametry jsou: 1,8V napájecí napětí jádra, 3,3V napětí vstupních a výstupních pinů, 6912 logických elementů, až 300 000 systémových hradel, 146 uživatelských pinů a 8 KB integrované paměti SRAM. Značení obvodů Spartan-IIE je popsáno na obr. 4.2.



Obr. 4.2 Značení obvodů Spartan-IIE

Hradlová pole FPGA mají konfiguraci uloženou ve vnitřní paměti RAM, která po odpojení napájecího napětí ztratí svůj obsah a je tedy nutné ji při každém zapnutí napájení do hradlového pole znovu nahrát. Existují celkem čtyři způsoby, jak zavést konfiguraci do obvodu Spartan-IIE, které se vybírají piny M0, M1 a M2 přímo na pouzdru FPGA. Detailně je konfigurování popsáno v [11] na straně 12.

První způsob se jmenuje „Slave Serial“ a ten je také použit na desce SP2EDB. V tomto režimu se konfigurace nahrává do hradlového pole sériově pomocí dvou signálů CCLK (hodiny) a DIN (data). Oba tyto signály se generují zvnějšku a hradlové pole je podřízené (proto také „slave“).

Druhý způsob je „Master Serial“. Od prvního režimu se liší pouze tím, že signál CCLK je generovaný interním oscilátorem uvnitř FPGA. Tento režim je primárně určen na konfigurování FPGA ze sériových pamětí PROM.

Třetí způsob, jak nakonfigurovat hradlové pole Spartan-IIE, je použít režim „Slave Parallel“. U dřívějších modelů byl tento režim znám také pod názvem „SelectMAP“ a je to nejrychlejší způsob, jak hradlové pole nakonfigurovat. Data se do FPGA odesílají po 8 bitech po signálech D0 až D7 a jako řídicí a stavové signály slouží #CS, #WRITE a #BUSY.

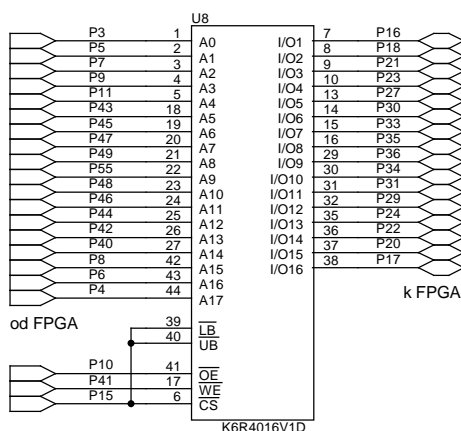
Čtvrtou možností je použit režim „Boundary-Scan“ přes standardní rozhraní IEEE 1149.1 (JTAG). U tohoto režimu jsou k zavádění konfigurace použity piny TMS, TCK, TDI a TDO. Ve skutečnosti je tento způsob konfigurace možný vždy, nezávisle na nastavení pinů M0 až M2. Nastavením režimu konfigurace „Boundary-Scan“ se jednoduše zakážou všechny ostatní režimy.

Pro uživatelské potřeby jsou do hradlového pole zavedeny dva pevné hodinové signály: 40 MHz na pin GCK2 a 8 MHz na pin GCK1. Na pin GCK3 vede navíc signál z pinu 39 konektoru pro připojení CCD snímače, který má sloužit na generování uživatelského hodinového signálu pro FPGA ze snímačové desky.

4.4.2 Rychlá statická paměť RAM

Hradlové pole XC2S300E má na čipu integrovánou paměť RAM o velikosti 8KB. To však pro náročnější výpočty a algoritmy zpracování obrazu nestačí, proto byla do zapojení přidána externí rychlá statická paměť RAM od firmy Samsung Electronics, K6R4016V1D o velikosti 256Kx16 bitů a přístupovou dobou 10 ns. Ke hradlovému poli je připojena přímo, podle obr. 4.3.

V případě že se paměť RAM nepoužívá, je vhodné nastavit alespoň na řídicí signály paměti #CS, #WE a #OE přímo úroveň H, nebo jen odpory pull-up, aby paměť nebyla stále ve vybraném stavu.

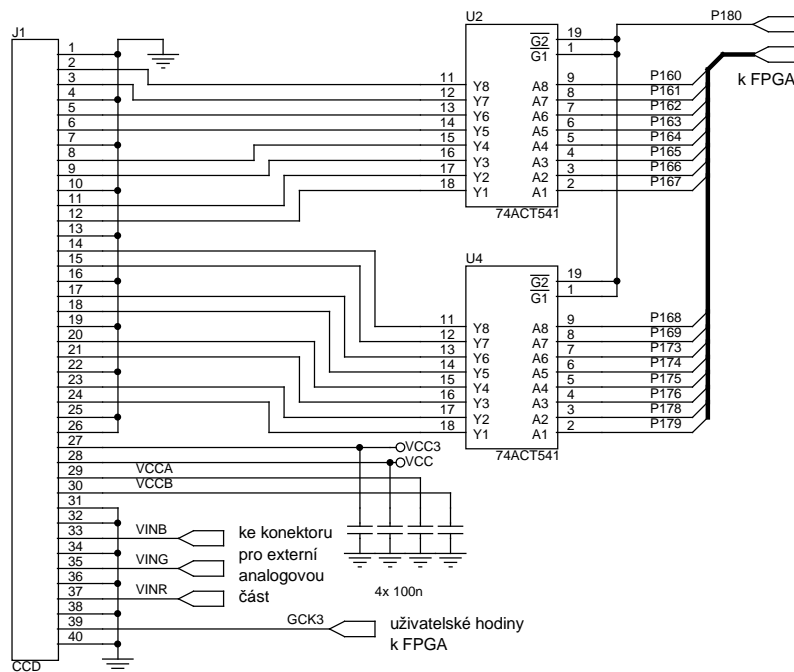


Obr. 4.3 Připojení SRAM ke hradlovému poli FPGA

4.4.3 Budiče pro registr řízení CCD snímače

Na desce SP2EDB je stejný konektor pro připojení snímačové desky, jako na ULC85, jsou tedy potřeba i 5V budiče pro registr řízení CCD. Jelikož je registr umístěn do hradlového pole, nejsou zvenčí potřeba budiče s pamětí (latch), ale stačí obyčejné jednosměrné budiče sběrnice 74ACT541.

Na rozdíl od ULC85, je zde k dispozici všech 16 bitů v plné rychlosti, až co zvládne FPGA. Ze snímačové desky také lze vést uživatelský hodinový signál CLKUSR (viz obr. 4.5) do hradlového pole (pin 39), pokud je potřeba snímač taktovat jiným kmitočtem, než který je na desce k dispozici.



Obr. 4.4 Zapojení výstupních registrů a konektor pro CCD na SP2EDB

GND	1	2	OUT0
OUT1	3	4	GND
OUT2	5	6	OUT3
GND	7	8	OUT4
OUT5	9	10	GND
OUT6	11	12	OUT7
GND	13	14	OUT8
OUT9	15	16	GND
OUT10	17	18	OUT11
GND	19	20	OUT12
OUT13	21	22	GND
OUT14	23	24	OUT15
GND	25	26	GND
VCC3,3	27	28	VCC5
VCCA	29	30	VCCB
GND	31	32	GND
VINB	33	34	GND
VING	35	36	GND
VINR	37	38	GND
CLKUSR	39	40	GND

Obr. 4.5 Zapojení konektoru pro snímačovou desku na SP2EDB

4.4.4 Uživatelské rozhraní

Pro naučení se základních principů psaní v jazyce HDL a následné ověření pomocí vizuální kontroly, jsem desku doplnil čtyřmi tlačítky a osmi různobarevnými LED, bez předem definované funkce. Tlačítka i LED jsou zapojeny k FPGA podle schématu na obr. 4.6.

Mezi tlačítka a FPGA jsem z bezpečnostních důvodů zařadil odpory. Kdyby totiž uživatel omylem nastavil pin s tlačítkem jako výstup a objevila by se na něm úroveň L, byl by tento pin po stisku tlačítka zkratován na +3,3V a to by mohlo způsobit i zničení FPGA.

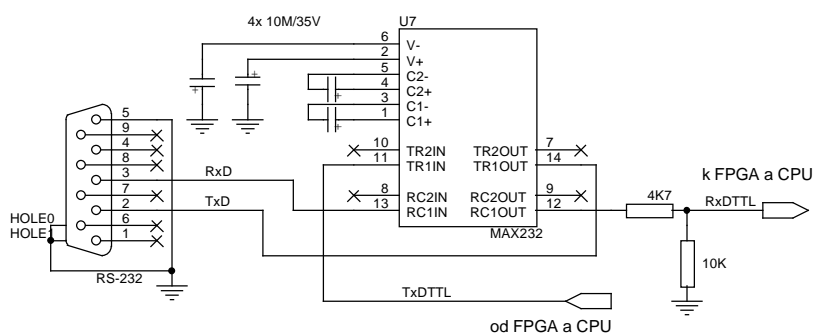
Ke správné funkci tlačítek i LED diod je potřeba správně nastavit vlastnosti odpovídajících pinů na FPGA. Piny pro LED diody je vhodné nastavit jako pomalé výstupní s maximální „silou“ (drive strength) 24 mA, aby nedocházelo ke zbytečným ztrátám na spínacích prvcích uvnitř FPGA. Nepoužitým pinům LED diod je vhodné nastavit pull-up odpor, aby skrz ně zbytečně netekl proud.

Naopak piny s tlačítky musí být nastaveny jako vstupní s pull-down odporem, neboť se spínají do +3,3V. Nepoužité piny tlačítek není potřeba nijak zvlášť ošetřovat.

4.4.8 Sériová linka RS-232

Alternativně k USB lze použít ke komunikaci linku RS-232. Je s rozhraním USB naprosto rovnocenná a nemusí se mezi nimi nijak přepínat. Dokonce se dají používat obě současně, i když to postrádá logický smysl. SP2EDB totiž nerozlišuje mezi tím, ze které linky jí přijde požadavek na nějakou činnost. Prostě se příkaz vykoná a odpověď na něj se odešle na tu samou linku, ze které byl požadavek přijat.

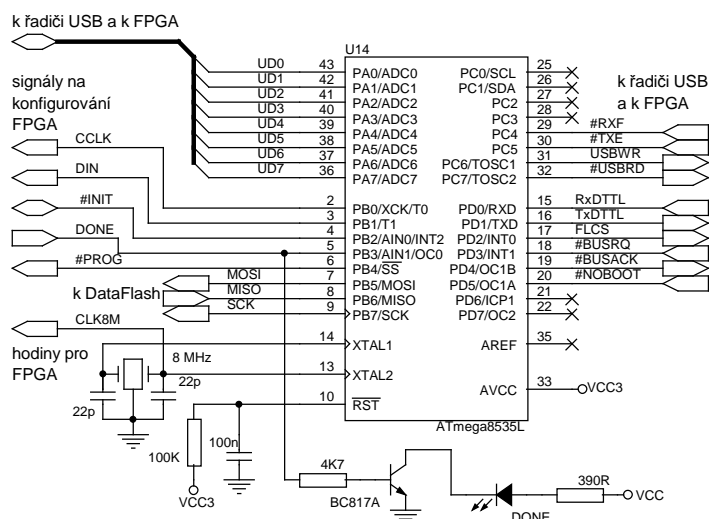
Obvodové zapojení je na obr. 4.10. Oba signály pro příjem i odesílání jsou vedeny současně na FPGA i řídicí procesor. Jak již bylo popsáno dříve, řídicí procesor se může od všech signálů odpojit a přenechat tak veškeré řízení na FPGA. Tato možnost bude dále popsána v kapitole 4.4.9.



Obr. 4.10 Zapojení budiče sériové linky RS-232 na SP2EDB

4.4.9 Procesor pro konfiguraci FPGA

Procesor pro řízení a konfigurování FPGA jsem zvolil z rodiny AVR od firmy Atmel, ATmega8535L-8JC. Hlavním důvodem, proč jsem použil právě tento procesor, bylo to, že splňoval všechny mé požadavky na počet pinů, napájecí napětí a rychlost, a také jeho okamžitá dostupnost v jedné pražské prodejně.



Obr. 4.11 Zapojení řídicího procesoru na SP2EDB

Na desce slouží tento procesor k jedinému účelu. Má za úkol komunikovat s PC přes USB nebo sériovou linku a přenášet soubory s konfigurací buď do sériové FLASH paměti, nebo přímo do FPGA.

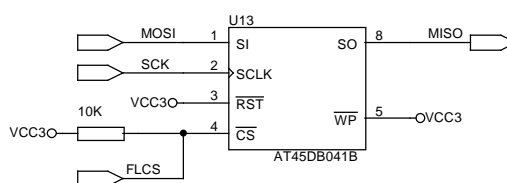
Hned po zapnutí napájení procesor prohledá tabulku rozdělení DataFlash, která je uložena v její první stránce a hledá, jestli v ní není aktivní blok, určený k automatickému zavedení do FPGA. Pokud je takový blok nalezen, odešle se jako konfigurace do hradlového pole. Poté je procesor připraven ke komunikaci s PC. Vlastní konfigurování FPGA pak probíhá pomocí signálů CCLK, DIN, #INIT, #PROG a DONE. Podrobně je algoritmus konfigurace popsán v [11] na straně 12. Proběhne-li konfigurace bez chyby, rozsvítí se LED s názvem DONE.

Po úspěšném nakonfigurování může hradlové pole zažádat řídicí procesor o uvolnění všech signálů sdílených mezi FPGA a procesorem. Docílí se toho to nastavením signálu #BUSRQ do úrovně L. Jakmile procesor zjistí signál v úrovni L, nastaví všechny své sdílené piny do stavu vysoké impedance a odpoví nastavením signálu #BUSACK také do úrovně L. Opačný postup také funguje. Když hradlové pole již nepotřebuje přístup ke sdíleným prostředkům, nastaví signál #BUSRQ zpět do úrovně H a procesor obnoví svou činnost. Po dobu, co je řídicí procesor odpojen, s ním však nelze nijak komunikovat.

Zvlášť proto upozorňuji na možnost vzniku chyby s tímto pinem. Vývojové prostředí pro hradlová pole, Xilinx ISE, má totiž implicitně nastaveno, aby na nepoužité piny v návrhu nastavil pull-down odpor. Tento pull-down je však silnější, než pull-up v řídicím procesoru, takže se po konfiguraci objeví na pinu #BUSRQ úroveň L. Čili pokud FPGA vůbec nebude potřebovat přístup ke sdíleným prostředkům, je vhodné tento pin držet v úrovni H, aby se neztratila možnost komunikace s řídicím procesorem. Docílí se toho buď přiřazením pevné úrovně H tomuto výstupnímu pinu, nebo nastavením prostředí ISE tak, aby hradlové pole na nepoužité piny nastavilo pull-up odpory místo pull-down. Osobně bych dal přednost

druhému řešení, protože jak budiče pro CCD, tak SRAM a diody LED jsou aktivní v nulové úrovni a všechny tyto periferie by byly, díky pull-down odporům, aktivní.

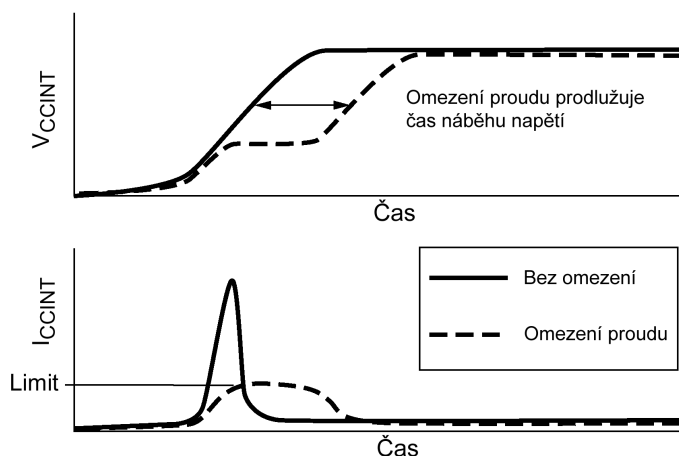
V případě že uživatel nahraje do DataFlash konfigurační blok a nastaví jej jako aktivní, dojde k jeho automatickému zavedení do FPGA při zapnutí napájení. Pokud bude v této konfiguraci úmyslně, nebo i nedopatřením držen pin #BUSRQ v úrovni L, komunikaci s řídicím procesorem se vůbec nepodaří navázat a vypnout tak automatické nahrávání konfigurace při stratu. Pro tento případ je na procesoru pin s názvem #NOBOOT. Přidržením tohoto pinu v úrovni L (spojením se zemí) při zapnutí napájení nedojde k automatickému zavedení jakéhokoli bloku z DataFlash a procesor je ihned připraven ke komunikaci. Tento pin však na DPS nikam nevede a v případě že je potřeba jej použít, musí se přímo na pin 20 na pouzdro procesoru připojit vodič spojený se zemí.



Obr. 4.12 Připojení sériové paměti DataFlash k CPU na SP2EDB

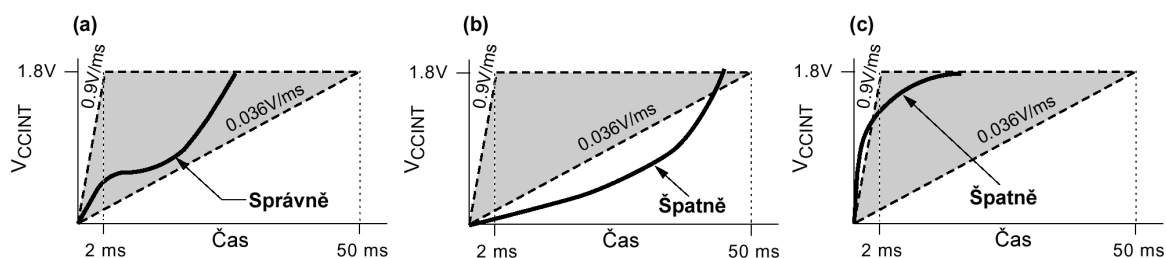
4.4.10 Napájecí zdroj

Konfigurace programovatelného hradlového pole je uložena v paměťových buňkách RAM, které určují propojení jednotlivých signálů uvnitř. Paměťové buňky jsou však po zapnutí napájení v náhodném stavu a vnitřní propojení tedy také. To způsobuje že se FPGA, při náběhu napájecího napětí V_{CCINT} kolem 0,6 až 0,8 V, chová jako velmi malá impedance 0,3 až 6 Ω a pokud není zdroj napětí proudově omezen, odebírá až několik ampérů do té doby, než se všechny paměťové buňky vynulují. Potom napětí V_{CCINT} vzroste na svých jmenovitých 1,8 V. Proudová špička je zobrazena na obr. 4.13. Omezením proudu se špička zmenší, ale také se prodlouží doba náběhu napětí.



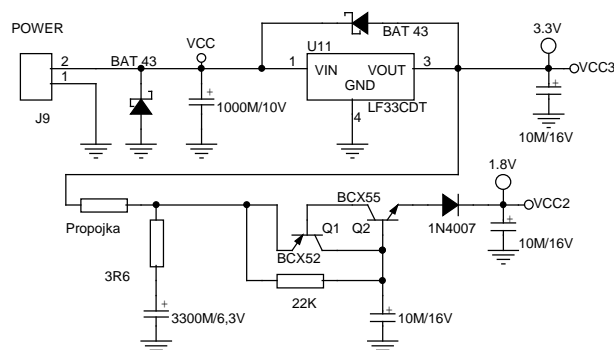
Obr. 4.13 Proudová špička při zapnutí napájení FPGA

V dokumentaci o hradlových polích Spartan-IIE ([11] a [13]) se uvádí minimální doba náběhu napětí V_{CCINT} 2 ms a maximální doba 50 ms. K rychlejšímu vzrůstu napětí (obr. 4.14c) by totiž bylo potřeba proudové špičky o velikosti až několika ampérů a to by hradlové pole nemuselo vydržet. Naopak náběh mnohem pomalejší než 50 ms (obr. 4.14b) by způsoboval, že by napětí zůstávalo příliš dlouho v kritické oblasti 0,6 až 0,8 V, ve které je vysoký proudový odběr a hrozilo by poškození FPGA tepelnými účinky. Správný průběh napětí V_{CCINT} je na obr. 4.14a.



Obr. 4.14 Doba náběhu napětí na FPGA Spartan-IIE

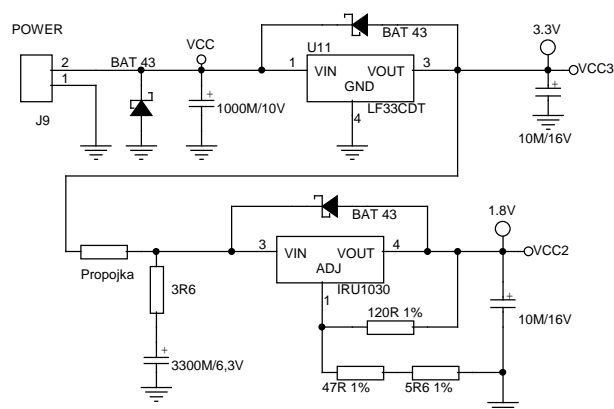
K zajištění správného průběhu napětí V_{CCINT} slouží obvod s názvem „Power Assist“ (viz [12]). Obvod zajišťuje opožděné sepnutí napájení V_{CCINT} až ve chvíli, kdy jsou filtrační kondenzátory ve zdroji úplně nabitě a jsou tak schopné dodat potřebnou proudovou špičku. Schéma obvodu je na obr. 4.15. Jako spínací prvek je použit tyristor, udělaný ze dvou tranzistorů Q1 a Q2. Časovou prodlevu určuje RC článek, zapojený do řídicí elektrody tyristoru. Jakmile napětí na kondenzátoru překročí přibližně 0,6V, tyristor se otevře. Odpor $3,6\Omega$ v sérii s filtračním kondenzátorem slouží k omezení proudu a ke zpomalení náběhu napětí.



Obr. 4.15 Napájecí zdroj a obvod „Power Assist“ na SP2EDB

Když jsem však takto realizovaný obvod vyzkoušel v praxi, zjistil jsem že na tyristoru není konstantní úbytek napětí, jak jsem původně předpokládal. Na výstupu obvodu bylo naprázdno 2,4V a při zatížení proudem 200 mA kleslo napětí na 1,2V. Obvod se tedy chová jako velmi měkký zdroj napětí a pro napájení FPGA je nepoužitelný.

Zdroj na DPS jsem nakonec upravil podle schématu na obr. 4.16. Ke stabilizaci 1,8V jsem použil lineární regulovatelný stabilizátor s nízkým úbytkem napětí od firmy International Rectifier, IRU1030. S takto zapojeným zdrojem jsem neměl nejmenší problémy a spolehlivě funguje. Jako alternativu za IRU1030 lze také použít obvod LM1084, vyráběný firmou National Semiconductor.



Obr. 4.16 Napájecí zdroj SP2EDB po úpravě

4.5 Možnosti použití SP2EDB

Vývojová deska SP2EDB je velice univerzální nástroj pro výuku práce s hradlovým polem FPGA. Použitím k řízení CCD snímače její možnosti však zdaleka nekončí. Velikost hradlového pole a schopnost autonomní funkce (automatická konfigurace z DataFlash) umožňují do SP2EDB naprogramovat např. 8, 16 nebo 32bitový logický analyzátor se vzorkovací frekvencí až 160 MHz, nebo generátor sekvencí s taktom až 160 MHz. Velikost

hradlového pole umožňuje realizaci tak složitých zařízení, jako je např. MP3 dekodér, nebo hardwarový emulátor jednočipového mikrokontroléru AVR.

5. ACID-IS – KOMPLEXNÍ ŘEŠENÍ

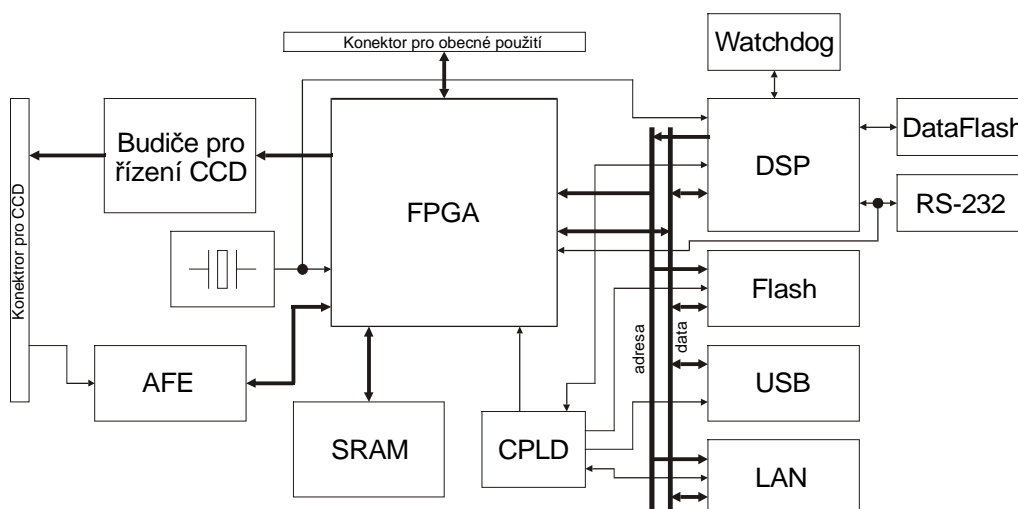
(All-purpose Control and Interface Device for Image Sensors)

5.1 Spojení ULC85 a SP2EDB

Finálním produktem a také komplexním řešením zadání diplomové práce je zařízení, které v sobě kombinuje výhody obou desek ULC85 a SP2EDB. Zapojení signálového procesoru, paralelní FLASH paměti, dohlížecího obvodu, USB klienta a AD převodníku bylo úspěšně odzkoušeno a odladěno na ULC85. SP2EDB zase posloužila jako výtečný diagnostický prostředek na odzkoušení zapojení a konfigurace hradlového pole FPGA, ovládání a čtení sériové paměti DataFlash firmy Atmel, ale také odhalila chyby a nesrovnalosti v katalogových zapojeních nebo v aplikačních poznámkách (viz kapitola 7).

Kombinace obou uvedených zařízení je navíc doplněna o řadič rozhraní Fast Ethernet, jak již bylo uvedeno v kapitole 2.5.

5.2 Blokové schéma celé desky



Obr. 5.1 Blokové schéma ACID-IS

5.3 Funkce jednotlivých bloků

Většina funkčních bloků jednotky ACID-IS je převzatá z desek ULC85 a SP2EDB. Budu zde proto popisovat pouze odlišnosti od původní implementace jednotlivých bloků. Kompletní schéma lze nalézt v příloze.

5.3.1 *Signálový procesor DSP*

Signálový procesor, použitý na této desce, je ze stejné rodiny jako na ULC85, typ ADSP-2188NKST-320. Od procesoru na ULC85 se liší napájecím napětím, které je zde 1,8V a taktovacím kmitočtem, který je 40 MHz (interně tedy 320 MHz). Dále má k dispozici 48K slov paměti programu a 56K slov paměti dat integrované na čipu.

Paralelní FLASH paměť, ze které DSP načítá při startu paměť programu, má velikost 512 KB a je opět rozdělena na 4 stránky po 128 KB. Adresové vodiče A17 a A18 jsou vedeny skrz obvod CPLD (viz kapitola 5.3.6), kde dochází k vlastnímu překladu adres. Na této desce nedochází k žádnému zrcadlení paměti jako na ULC85, pouze se stránky mapují do adresového prostoru DSP v jiném pořadí. Jak je toto mapování závislé na propojkách J14 a J15 je popsáno v tab. 5.1.

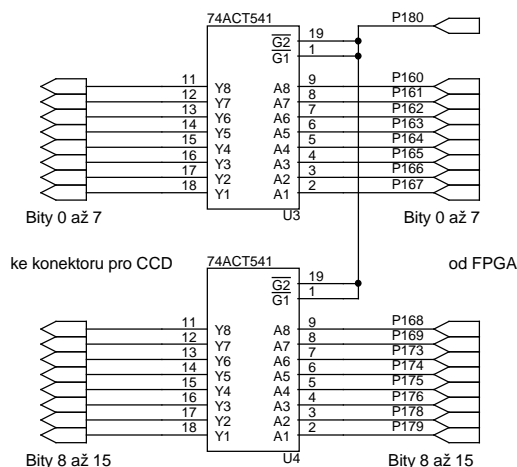
Tab. 5.1 Mapování paměti FLASH do adresového prostoru DSP na ACID-IS

J14 - BANK1	J15 - BANK0	Pořadí stránek ve FLASH, jak je vidí DSP
0	0	0 - 1 - 2 - 3
0	1	1 - 2 - 3 - 0
1	0	2 - 3 - 0 - 1
1	1	3 - 0 - 1 - 2

V nulté stránce paměti FLASH je opět nahrán program „monitor“, který umožňuje po sériové lince nahrát a spustit aplikaci pro DSP. Procesor na této desce je však taktován kmitočtem 40 MHz, nelze proto přímo použít monitor z ULC85, kde je takt procesoru jen 36MHz. Spolu s hodinovým kmitočtem procesoru se totiž mění i přenosová rychlost sériové linky. Musel jsem tedy monitor upravit na nový hodinový kmitočet. Nikde jsem ale nenašel fungující zdrojové texty monitoru, tak jsem použil fungující monitor z ULC85, jen jsem v něm upravil konstantu pro sériovou linku. Nově upravený monitor pro hodinový kmitočet 40 MHz je k dispozici na příloženém CD v adresáři s podklady pro DSP.

5.3.2 *Budiče pro registr řízení CCD*

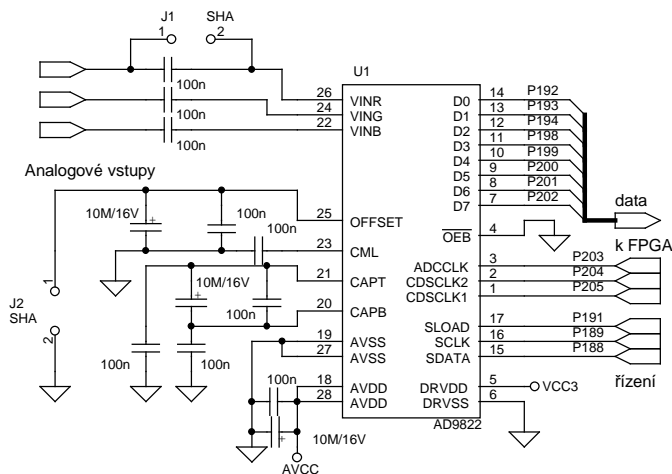
Zapojení výstupních budičů je naprosto totožné se zapojením v modulu SP2EDB (viz kapitola 4.4.3), včetně čísel pinů na FPGA. Jen pro úplnost uvádím zapojení na obr. 5.2.



Obr. 5.2 Zapojení výstupních budičů registru řízení CCD na ACID-IS

5.3.3 Analog Front-End

Použitý převodník a zapojení jeho analogové části je opět shodné jako na ULC85 (viz kapitola 3.2.3). Digitální výstup, tvořený signály D0 až D7, je veden přímo na FPGA, kde se počítá s jeho dalším zpracováním. Řídicí signály i konfigurační rozhraní AFE vedou taktéž přímo na FPGA. Čísla pinů na hradlovém poli jsou na obr. 5.3 u každého signálu uvedena.

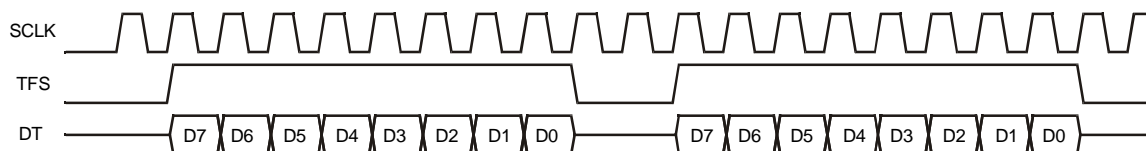


Obr. 5.3 Zapojení AD převodníku (AFE) na ACID-IS

5.3.4 Hradlové pole FPGA

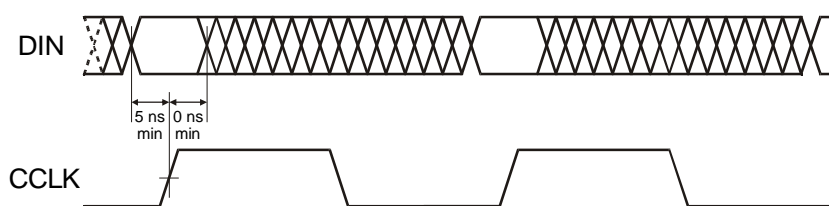
Je zde použit stejný typ hradlového pole Spartan-IIIE, XC2S300E-6PQ208C, jako na desce SP2EDB (viz kapitola 4.4.1). Jeho konfigurování v módu „Slave Serial“ zajišťuje

signálový procesor prostřednictvím sériového kanálu SPORT. Časový diagram signálů kanálu SPORT, je-li nastaven podle Fig. 5-17 v [4], je na obr. 5.4.



Obr. 5.4 Časový diagram odesílání pomocí SPORTu signálového procesoru

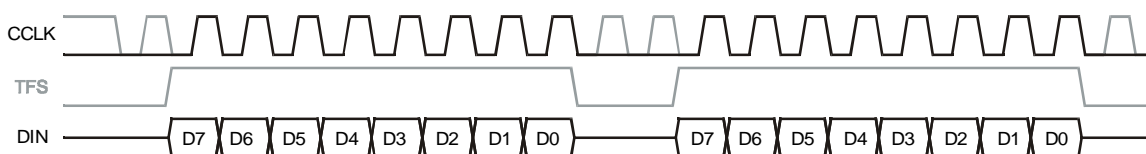
Hradlové pole však očekává platná data při náběžné hraně signálu CCLK, viz obr. 5.5. Bylo tedy potřeba hodiny ze SPORTu SCLK invertovat a ještě k tomu hradlovat signálem TFS, protože SCLK běží neustále, i když nejsou vysílána žádná data.



Obr. 5.5 Časový diagram konfigurace FPGA

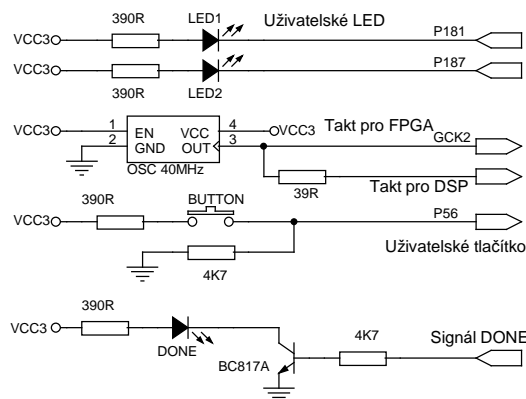
Časový průběh výsledných signálů pro konfiguraci FPGA jsou na obr. 5.6. Zobrazen je také signál TFS, kterým se v CPLD hradluje hodinový signál z SPORTu. Logická rovnice pro signál CCLK je:

$$CCLK = \overline{SCLK} \cdot TFS$$



Obr. 5.6 Signály pro konfiguraci FPGA, upravené v CPLD

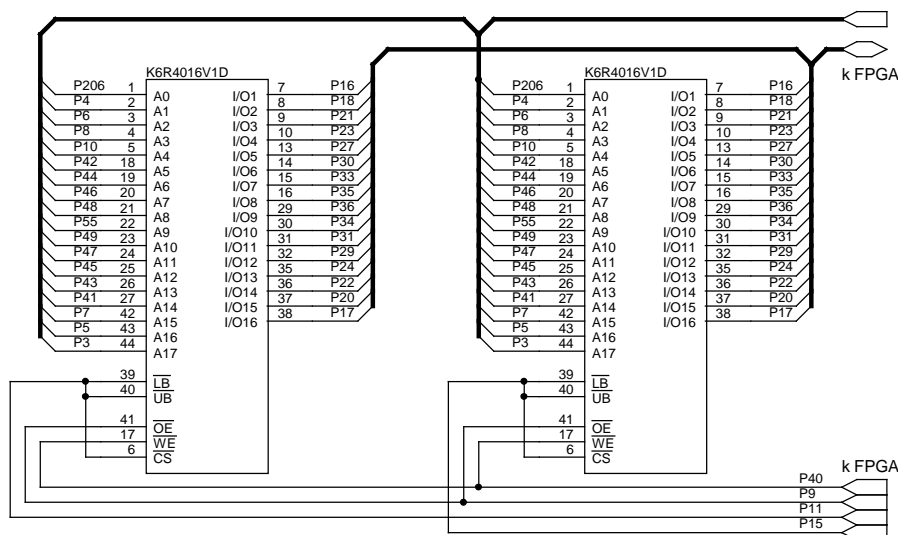
Přímo k FPGA jsou také připojeny dvě LED, které jsou fyzicky umístěny blízko AFE a konektoru pro CCD snímač. Lze je tedy využít pro jednoduchou signalizaci. Na pin 56 hradlového pole je připojeno uživatelské tlačítko, které lze rovněž využít k libovolnému účelu. Všechny doplňkové obvody pro FPGA jsou na obr. 5.7.



Obr. 5.7 Zapojení pomocných obvodů u FPGA na ACID-IS

5.3.5 Rychlá Statická paměť RAM

Paměť RAM je na této desce k FPGA připojena stejným způsobem, jako na SP2EDB. Liší se pouze propojení pinů, protože jsou zde paměti dvě se společnou adresovou i datovou sběrnicí, ale oddělenými signály chip-select #SCS0 a #SCS1. Celkem lze tedy adresovat 1MB, tedy 1024KB paměti v organizaci 512Kx16 bitů. Zapojení paměti včetně čísel pinů na FPGA je na obr. 5.8.

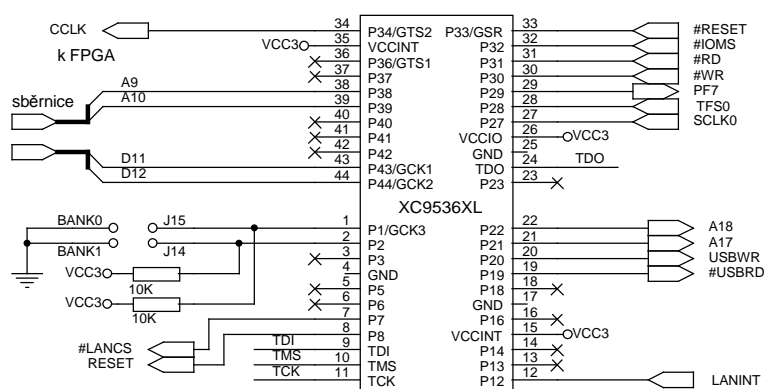


Obr. 5.8 Zapojení statické paměti RAM na ACID-IS

5.3.6 Obecná propojovací logika „glue logic“

Místo složité a pevné propojovací logiky, použité na ULC85, je na této desce použit programovatelný logický obvod CPLD od firmy Xilinx, XC9536XL-5VQ44C. Obsahuje 36 programovatelných makrobuňek s možností nahradit až 800 logických hradel. Zpoždění

logických signálů přes CPLD výrobce uvádí jednotné, 5 ns pin-pin. Zapojení CPLD je na obr. 5.9.



Obr. 5.9 Zapojení obvodu CPLD na ACID-IS

Obvod se programuje v systému přes standardní rozhraní IEEE1149.1 (JTAG), přímo z vývojového prostředí Xilinx ISE.

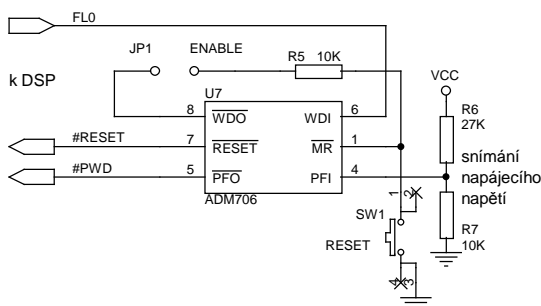
Vnitřní zapojení v obvodu CPLD je popsáno pouze několika řádky textu v jazyce VHDL a implementace zabírá v obvodu jen asi 6% jeho celkové kapacity. Menší CPLD obvod se však nevyrobí a GAL nebo PALCE jsem použít nechtěl kvůli jejich vysoké spotřebě.

Příklad zdrojového textu v jazyce VHDL pro popis kombinační logiky uvnitř CPLD:

```
Reset <= not nReset;
CCLK <= (not SCLK0) and TFS0;
PF7 <= not LANINT;
USBWR <= not ((not AIN(9)) or AIN(10) or nIOMS or nWR);
nUSBRD <= (not AIN(9)) or AIN(10) or nIOMS or nRD;
AOUT <= D + (not BANK);
nLANCS <= AIN(9) or AIN(10) or nIOMS;
```

5.3.7 Dohlížecí obvod – Watchdog

Dohlížecí obvod je zapojen stejně jako na ULC85 (viz kapitola 3.2.6), má zde i totožnou funkci.

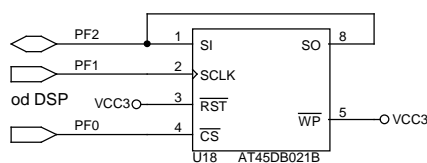


Obr. 5.10 Dohlížecí obvod – Watchdog na ACID-IS

5.3.8 Sériová paměť DataFlash

DataFlash je zde zapojená úsporným způsobem. Jelikož během komunikace nikdy nedochází k současnému příjmu a vysílání dat, mohl jsem piny SI a SO spojit (viz obr. 5.11). Sdružené signály jsou vedeny na pin PF2, který je buzen obousměrně.

Rozhraní SPI, kterým se paměť ovládá, je v signálovém procesoru kompletně softwarově emulováno. Podrobně to bude popsáno v kapitole 6.6.7, včetně časových průběhů.



Obr. 5.11 Připojení sériové paměti DataFlash k DSP na ACID-IS

5.3.9 Konektor pro obecné použití

Na uživatelský konektor se zapojením na obr. 5.12, je vyvedeno 33 nezapojených pinů hradlového pole. Jedenáct párů signálů na tomto konektoru se dá využít jako přenosové linky standardu LVPECL (viz [11], str. 7). Hradlové pole sice podporuje i LVDS, ale k použití tohoto standardu by celé napájení vstupně výstupní části FPGA muselo být místo na stávajících 3,3V připojeno na 2,5V. To však deska neumožňuje.

Konektor je primárně určen k přímému připojení CMOS obrazového snímače. Obsahuje dostatečný počet uživatelských signálů pro připojení jak řádkových, tak i plošných CMOS snímačů s vnitřní digitalizací videosignálu.

2	P151	3	P149	4	P148	5	P146	6	GND	7	P140	8	P138	9	P136	10	P134	11	GND	12	P129	13	P126	14	P125	15	P122	16	GND	17	P116	18	P114	19	P113	20	P111	21	VCC3,3	22	VCC3,3		
1	P152	3	P150	5	GND	7	P147	9	P145	11	P141	13	P139	15	GND	17	P135	19	P133	21	P132	23	P127	25	GND	27	P123	29	P121	31	P120	33	P115	35	GND	37	P112	39	P110	41	P109	42	

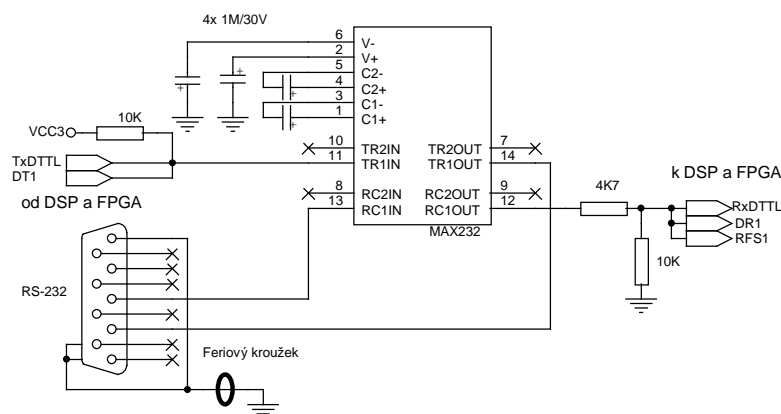
Obr. 5.12 Konektor pro obecné použití na ACID-IS

5.3.10 Konektor pro připojení snímačové desky

Konektor je zapojen stejně jako na SP2EDB (viz kapitola 4.4.3), včetně signálu CLKUSR na pinu č. 39.

5.3.11 Sériová linka RS-232

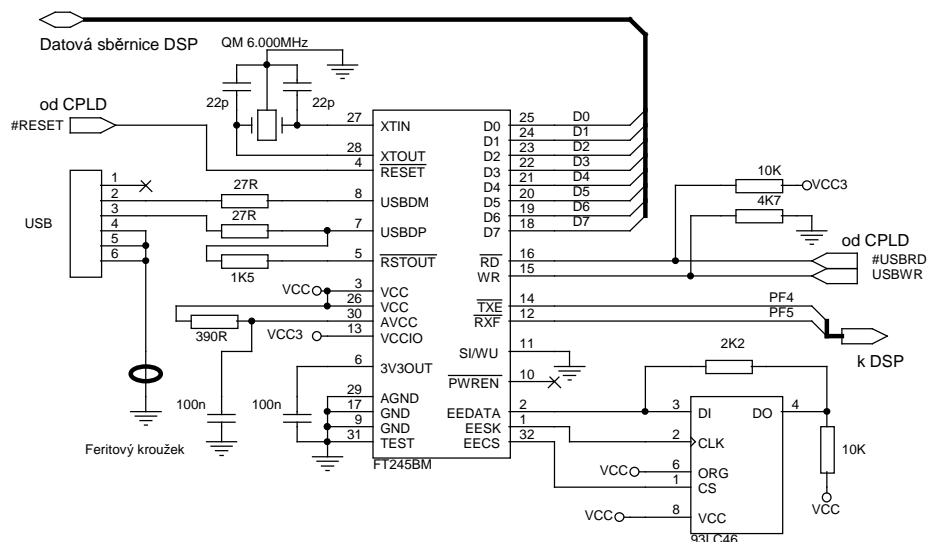
Budič sériové linky RS-232 je společný pro DSP i FPGA. Uživatel má tak možnost řídit sériové linky buď emulovat jako na ULC85 (viz kapitola 3.2.5), nebo do hradlového pole naprogramovat hardwarový řadič, který by běžel nezávisle na procesoru a šetřil by tak jeho čas.



Obr. 5.13 Zapojení budiče sériové linky RS-232 na ACID-IS

5.3.12 Komunikační rozhraní USB

Řadič rozhraní USB je, na rozdíl od SP2EDB, přístupný pouze ze signálového procesoru a je zapojený podobně, jako na ULC85, pouze signál #RESET je vedený od resetovacího tlačítka.



Obr. 5.14 Zapojení řadiče rozhraní USB na ACID-IS

5.3.13 Rozhraní Fast-Ethernet 100Base-TX

Novinkou na tomto zařízení je síťové rozhraní Fast-Ethernet. Jako obvodový řadič jsem použil LAN91C111 od firmy SMSC. V tomto obvodu je integrován jak řadič linkové vrstvy (MAC), tak i budič fyzické vrstvy rozhraní (PHY). Je proto potřeba minimum vnějších součástek. Obvod bývá v anglické literatuře označován jako NIC (Network Interface Controller). Tento název je však v Českém jazyce zavádějící, a proto jsem se rozhodl obvodový řadič označovat jako LAN.

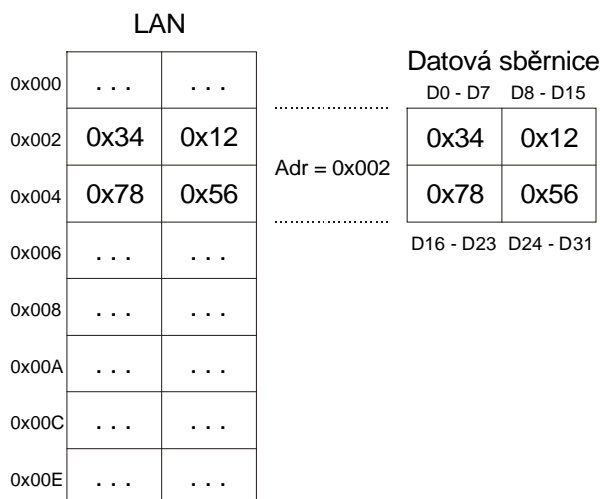
Registry řadiče jsou napávány do IO adresového prostoru signálového procesoru, kde obsazují adresy 0x000 až 0x1FF, přičemž skutečný rozsah adres je jen 0x000 až 0x007 a ten se dále zrcadlí v celém, výše zmíněném rozsahu. Všechny registry jsou 16bitové, proto je adresová sběrnice obvodu optimalizovaná právě pro adresování po 16 bitech.

Signálový procesor má však paměť i IO prostor adresovaný po 16bitových slovech, zatímco LAN má registry adresované po bytech. DSP tedy uloží dvě 16bitová slova na adresy 0x000 a 0x001, ale LAN musí tato dvě slova uložit na adresy 0x000 a 0x002, viz obr. 5.15.



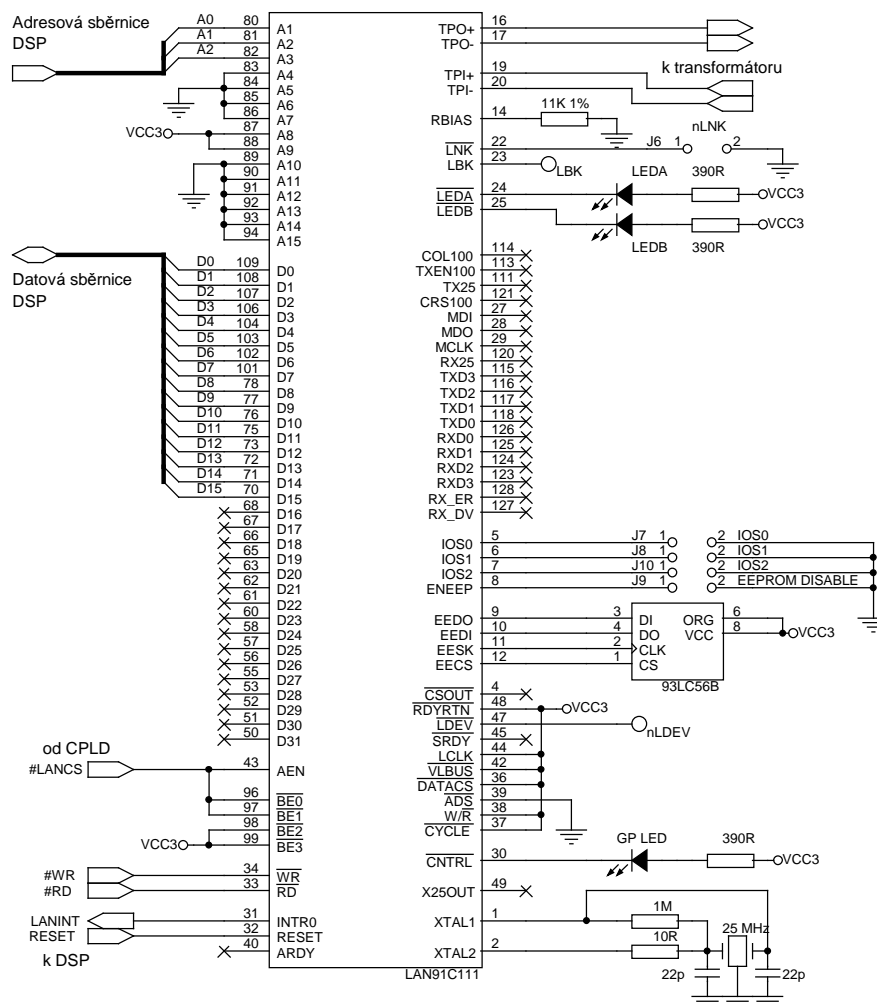
Obr. 5.15 Překlad adres mezi DSP a LAN

Proto jsou také adresové vodiče od DSP A0 až A2, zapojeny na signály A1 až A3 u řadiče LAN, aby docházelo k potřebnému „násobení adresy dvěma“. Signál A0 se u LAN vůbec nevyskytuje, nejmenší adresovatelný blok je 1 slovo (16 bitů). Na sběrnici se ale vždy zpřístupňují současně 4 byty (32 bitů), počínaje nastavenou adresou (viz obr. 5.16). Přístup k jednotlivým bytům 32bitového bloku se pak povoluje pomocí výběrových signálů #BE0 až #BE3. V mém zapojení se však používá výhradně 16bitový přístup, čili signály #BE2 a #BE3 jsou trvale v log. 1 a signály #BE0 a #BE1 se současně aktivují spolu s výběrem obvodu AEN. Druhá polovina datové sběrnice, tedy signály D16 až D31, nejsou použity.



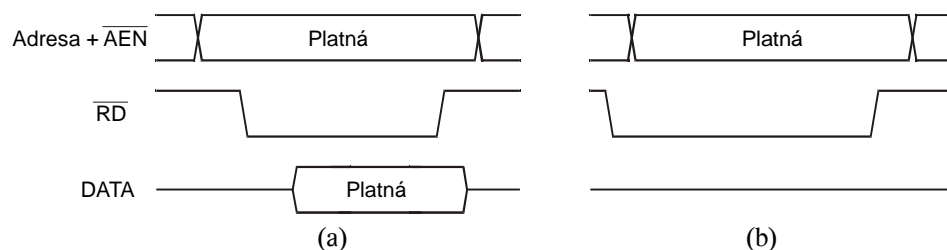
Obr. 5.16 Adresování řadiče Ethernetu

Obvod je původně určen pro sběrnici ISA osobního počítače IBM PC, proto je v něm integrován dekodér adresy. Platná adresa obvodu se vyhodnocuje z adresových signálů A4 až A15. Běžné síťové karty do sběrnice ISA mají základní adresu 0x300 a na tuto adresu je také LAN implicitně nastaven. Na adresové sběrnici vypadá adresa 0x300 jako 0000001100000000. Dvě jedničky na deváté a desáté pozici zprava odpovídají adresovým vodičům A8 a A9. Tato adresa, jak si lze všimnout na obr. 5.17, je u obvodu natvrdo nastavena. Proměnné jsou pouze vodiče A1 až A3, které vybírají registry obvodu.



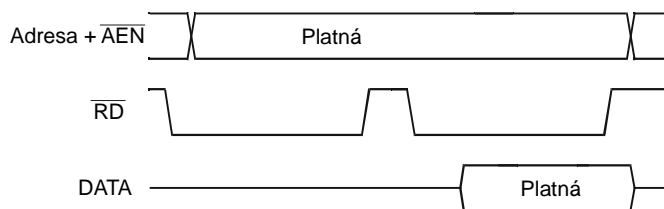
Obr. 5.17 Zapojení řadiče Fast-Ethernet na ACID-IS

V dokumentaci [15] se uvádí předstih mj. signálu #AEN před aktivním signálem #RD minimálně 10 ns (obr. 5.18a). Jistou anomálií tohoto obvodu je jeho chování při čtení z jeho registrů, když přejde signál #RD do log. 0 dříve, než signál #AEN. Data se na sběrnici totiž neobjeví 10 ns po aktivaci signálu #AEN, jak by se dalo očekávat. Data se na sběrnici neobjeví vůbec (obr. 5.18b). Signál #RD tedy nejspíše reaguje na hranu a ne na úroveň, jak bývá zvykem.



Obr. 5.18 Zjednodušený časový diagram čtení z LAN

Signálový procesor generuje #IOMS, #RD a adresu téměř současně, ale adresa spolu s #RD prochází ještě dekódovací logikou, kde se o 5 ns zpozdí. Na obvod LAN tedy dorazí spádová hrana signálu #RD dříve, než spádová hrana signálu #AEN (obr. 5.18b) a čtecí cyklus vůbec nezačne. Tento problém jsem nakonec vyřešil dvojitým, těsně navazujícím čtením z obvodu (obr. 5.19). DSP při dvou po sobě jdoucích instrukcích čtení z IO portu nechává aktivní signál #IOMS, pouze na krátký okamžik deaktivuje signál #RD. Obvod LAN vyhodnotí jako platnou až druhou spádovou hranu #RD a teprve s ní začne čtecí cyklus. Tento jev však paradoxně nenastává u zápisu, pouze u čtení.



Obr. 5.19 Vynucení čtecího cyklu dvojitým čtením z LAN

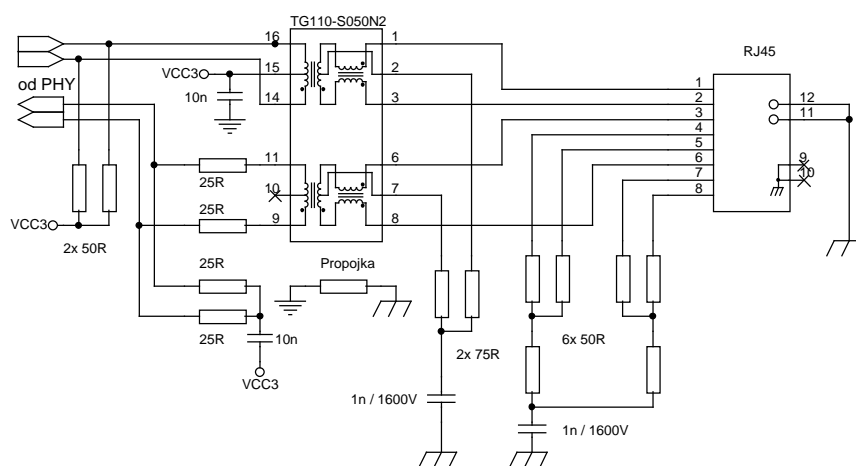
Obvod LAN dokáže generovat přerušení signálovému procesoru, pokud dojde k takové události na radiči, u které je generování přerušení povoleno. Na DSP je přerušení obsluženo jako IRQ2.

Pin #CNTRL na obvodu LAN slouží jako obecný výstupní signál, nastavitelný pomocí konfiguračního registru obvodu, a je na něj připojena signalizační LED. Během ožívování desky jsem tuto LED využil jako první bod, kde se dá ověřit správná komunikace DSP s radičem. Ve finálním návrhu slouží jako signalizace zapnutého/vypnutého fyzického rozhraní Ethernet.

Podobně jako #CNTRL, existuje na obvodu obecný vstupní pin s názvem #LNK a je na něj připojen jumper. Jeho stav se dá zjistit přečtením EPH Status registru (viz [15]).

Základní nastavení obvodu, jako je fyzická (MAC) adresa, báze adresa na sběrnici a konfigurační registr, lze uložit do vnější sériové paměti EEPROM. Lze v ní mít uloženo dokonce až osm různých konfigurací najednou a volit mezi nimi pomocí jumperů IOS0 až IOS2. MAC adresa je však v paměti uložena pouze jednou, nezávisle na nastavení propojek IOS. V případě že nechceme z nějakého důvodu externí paměť EEPROM používat, zakáže se jumperem EEPROM DISABLE.

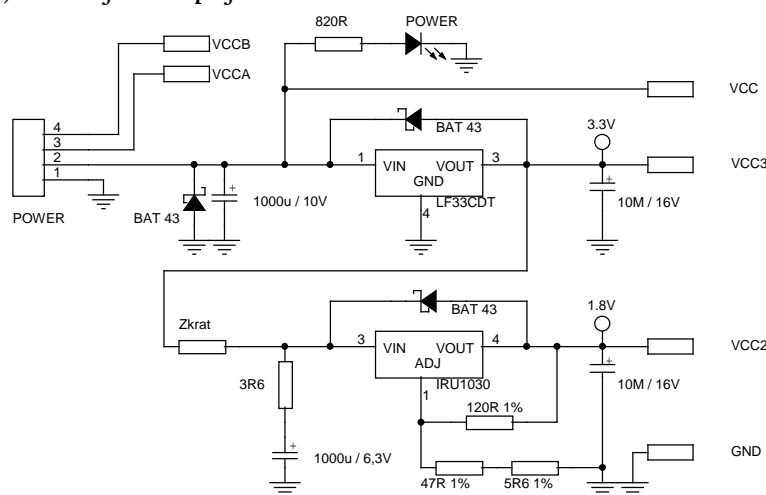
Obvod LAN má možnost zprostředkovat programování paměti EEPROM řídicímu procesoru pomocí algoritmu popsáném v [16]. Algoritmus jsem zkoušel, ale do paměti se mi zapsat nepodařilo. Není to ale zas tak velká komplikace. Externí paměť jsem zakázal jumperem EEPROM DISABLE a správné hodnoty do registrů, které se jinak načítají z EEPROM, zapisují procesorem při inicializaci obslužného programu. Výsledek je totožný.



Obr. 5.20 Zapojení analogové části fyzické vrstvy Ethernetu na ACID-IS

5.3.14 Napájecí zdroj

Napájecí zdroj je převzat z desky SP2EDB po úpravě, protože na ni bez problémů funguje. Neměl jsem tedy žádný důvod vymýšlet jiné zapojení. Napájecí konektor je stejný jako na ULC85, včetně jeho zapojení.

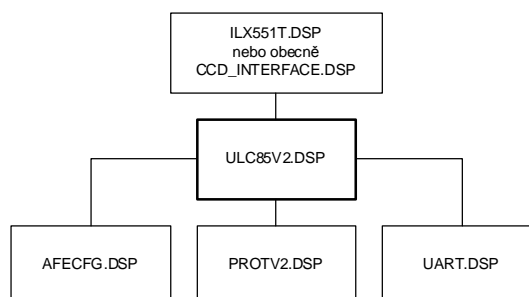


Obr. 5.21 Zapojení napájecího zdroje na ACID-IS

6. PROGRAMOVÉ VYBAVENÍ

6.1 Řídicí program pro ULC85

Program pro signálový procesor je sestaven z několika nezávislých programových modulů. Jejich hierarchie je na obr. 6.1. Ty samé moduly je pak možno znovu použít v dalších verzích, nebo v obslužném programu jiného zařízení. Následuje jejich stručný popis. Podrobnější popisy lze najít v samotných zdrojových kódech.



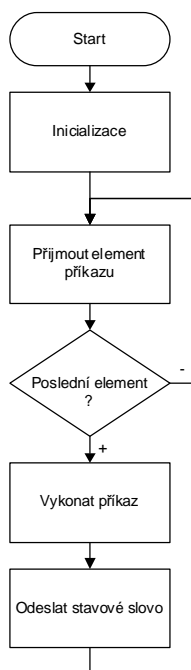
Obr. 6.1 Hierarchie programových modulů ULC85

6.1.1 ULC85V2.DSP

Hlavní modul obslužného programu. Zajišťuje inicializaci a nastavení všech periférií, komunikaci po USB a stavový automat pro zpracování příkazů ULC85. Podporovány jsou tyto příkazy:

- **Identify** (povinný příkaz s číslem 0)
Slouží k jednoznačné identifikaci připojeného zařízení.
- **Single-cycle Request**
Požadavek na jednorázový cyklus řízení CCD.
- **Long Download**
Pokus o odeslání dlouhého bloku dat (>128 slov) do ULC, které je napřed dotázáno, jestli data chce.
- **Long Upload**
Pokus o přijetí bloku dat z ULC, které je napřed dotázáno, zda může nebo chce data odeslat.
- **Short Download**
Odeslání krátkého bloku dat do ULC (≤ 128 slov) bez dotazu.
- **Set Parameter**
Nastavení obecného parametru ULC.
- **Read Parameter**
Přečtení obecného parametru ULC.

Podrobně jsou všechny příkazy popsány na konci zdrojového textu Ulc85v2.dsp v adresáři \Sources\DSP\ULC85 na příloženém CD.



Obr. 6.2 Vývojový diagram základního modulu ULC85

6.1.2 *UART.DSP*

Modul pro softwarovou emulaci asynchronní sériové linky UART. Implementace je částečně převzatá z [9] a částečně doopravená, protože původní program nefungoval.

Emulace je zčásti softwarová, ale využívá činnost synchronního sériového portu SPORT, který patří mezi základní periferie ADSP218x. Parametry emulované sériové linky jsou: 8 datových bitů, 1 start bit, 1 stop bit, žádná parita. To je celkem deset vysílaných nebo přijímaných bitů.

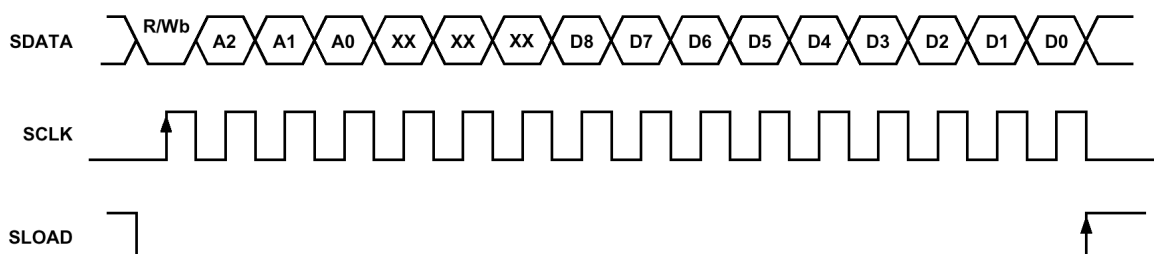
Začátek rámce UART je vždy identifikován start bitem, který má hodnotu log. 0. Toho je využito pro startování příjmu. Signál pro příjem dat DR0 je spojen se signálem pro startování přenosu RFS0, který reaguje právě na sestupnou hranu start bitu. Tím se spustí příjem dvou 15bitových slov, které dohromady tvoří jeden 30bitový rámec SPORT. Protože SPORT je synchronní se svými hodinami, můžou nastat dvě extrémní situace, že start bit nastane těsně před, nebo těsně po hodinovém impulsu. Celý přijatý 30bitový rámec může být tedy o jeden bit posunutý, a to je právě důvod, proč se přijímá 30 bitů namísto 10, takže hodinová frekvence SPORTu musí být nastavena na trojnásobek baudové rychlosti asynchronní linky. Každý bit rámce UART je tedy reprezentován třemi bity v rámci SPORT. Dojde-li totiž k posunutí o jeden bit dopředu nebo dozadu, prostřední ze tří bitů zůstane

vždy nezměněn. Konečný rámec UART se tedy poskládá z deseti prostředních bitů jednotlivých trojic v UART rámci.

Odesílání po sériové lince je jednodušší, než příjem. Není totiž potřeba žádná synchronizace, stačí pouze vygenerovat rámec pro SPORT tak, že se datový byte doplní na začátku start bitem a na konci stop bitem a celý desetibitový SPORT rámec se odešle s hodinovou frekvencí rovnou baudové rychlosti. Avšak nutnost přepínání rychlostí při odesílání a příjmu znemožňuje současné odesílání a příjem. Hodinová frekvence pro přenos po SPORTu se totiž nastavuje globálně pro příjem i odesílání. Aby se nemusela rychlost přepínat, je odesílaný rámec pro SPORT také „roztažen“ na 30 bitů a odesílá se také trojnásobnou rychlostí. Takto emulovaný sériový kanál splňuje požadavek na plně duplexní provoz.

6.1.3 AFECFG.DSP

Modul je určen pro konfiguraci AFE po softwarově emulované sériové lince. Časový průběh pro zápis do konfiguračních registrů AFE je na obr. 6.3. Čtení z registrů není implementováno. Podrobný popis jednotlivých podprogramů je uveden přímo ve zdrojovém kódu.



Obr. 6.3 Časový průběh zápisu do registru v AFE

6.1.4 CCD_INTERFACE.DSP

Šablona modulu pro řízení konkrétního typu obrazového snímače. Do této šablony se doplní podprogramy na řízení obrazového snímače, správy přenosu dat mezi ULC a PC a nastavování obecných parametrů, jako je doba integrace náboje, rychlost závěrky, zisk a offset AFE, atd. Podrobný popis požadavků na jednotlivé podprogramy je uveden přímo ve zdrojovém kódu.

6.1.5 ILX551T.DSP

Vzorový modul s konkrétní implementací podprogramů pro řádkový CCD snímač ILX551B od firmy Sony. Obsahuje řízení snímače ve standardním režimu i v triviálním módu FIR s obdélníkovou impulsní odezvou. Je možno nastavovat všechny konfigurační registry AFE a dobu integrace. Podporuje také zkušební rychlejší metodu přenosu dat do PC – „Burst“, při které se data odesílají neustále bez předešlého požadavku z PC.

6.1.6 *PROTV2.DSP*

Modul obsluhy univerzálního přenosového protokolu „ProtocolV2“, vyvinutého pro komunikaci mezi PC a měřicím modulem. Přenosovému protokolu je věnována celá kapitola 6.2. Modul zajišťuje přenos na úrovni jeho rámcové vrstvy.

6.2 Přenosový protokol ProtV2

Přenosový protokol je navržen tak, aby ho bylo možno použít na jakékoli přenosové médium. Od první chvíle, kdy byl ProtV2 vyvíjen, bylo počítáno s jeho nasazením na linky RS-232, USB a Ethernet a je pro ně také optimalizován. Protože je na DSP obtížné pracovat s jednotlivými byty, je navíc celý protokol zarovnaný na 16bitová slova.

Dvouvrstvá architektura Master – Slave (Master je většinou nadřazené PC a slave je podřízené zařízení) umožňuje implementaci základních subrutin pro přenos do zvláštní knihovny nebo modulu a jeho použití v několika různých programech. Dvě vrstvy protokolu jsou nazvány „linková“ a „rámcová“. Komunikace probíhá vždy stylem dotaz – odpověď, přičemž dotaz vysílá vždy master.

6.2.1 *Linková vrstva ProtV2*

Linková vrstva je závislá na konkrétním přenosovém médiu a je bytově orientovaná. Tvoří ji subrutiny `SendByte`, `GetByte` a `IsByteIn`, které nejsou obsaženy v modulu `ProtV2.DSP`. `SendByte` má odeslat jeden byte po lince. `GetByte` má přijmout s čekáním jeden byte z linky a `IsByteIn` má vrátit příznak, zda v příchozím kanálu čeká byte na přijetí. Tyto tři subrutiny volá modul `ProtV2` a jejich konkrétní implementace závisí na použitém přenosovém médiu.

6.2.2 *Rámcová vrstva ProtV2*

Do rámcové vrstvy patří pět základních elementárních rámců, přes které probíhá veškerá komunikace. Jsou to Příkaz, Datové slovo, Datový paket, Stavové slovo a Escape. Každý z elementů má pevně danou délku, kromě datového paketu, který má proměnnou délku.

6.2.2.1 *Příkaz*

Příkaz se používá k odeslání obecného požadavku nebo instrukci podřízenému zařízení a může ho vysílat pouze master. Jeho délka je vždy 4 byty (2 slova) a má strukturu dle tab. 6.1.

Tab. 6.1 Struktura příkazu ProtV2

Offset	Data	Popis
+00	0x00	doplněk do sudé velikosti (padding)
+01	(CN)	číslo příkazu 0x00 ÷ 0x7F
+02	(LO)	parametr příkazu nižší byte

+03	(HI)	parametr příkazu, vyšší byte
-----	------	------------------------------

6.2.2.2 Datové slovo

Datové slovo se odesílá tehdy, je-li potřeba předat obecný 16bitový parametr. Jedná se v podstatě o *příkaz* s rezervovaným číslem 0x80 a jeho parametr je právě datové slovo. Přesná struktura je popsána v tab. 6.2.

Tab. 6.2 Struktura datového slova ProtV2

Offset	Data	Popis
+00	0x00	doplněk do sudé velikosti (padding)
+01	0x80	0x80 - DataWord
+02	(LO)	Nižší byte datového slova
+03	(HI)	Vyšší byte datového slova

6.2.2.3 Datový paket

Datový paket slouží, obdobně jako *datové slovo*, k přenosu obecných dat. Velikost *datového paketu* může být libovolná v rozmezí 1 až 65535 slov, tedy 2 až 131070 bytů. Struktura *datového paketu* je popsána v tab. 6.3.

Tab. 6.3 Struktura datového paketu ProtV2

Offset	Data	Popis
+00	0x00	doplněk do sudé velikosti (padding)
+01	0x81	0x81 – DataPacket
+02	(LO)	Nižší byte délky paketu N (ve slovech)
+03	(HI)	Vyšší byte délky paketu N (ve slovech)
+04	(LO)	První datové slovo, nižší byte
+05	(HI)	První datové slovo, vyšší byte
..	..	
+N*2+2	(LO)	Poslední datové slovo, nižší byte
+N*2+3	(HI)	Poslední datové slovo, vyšší byte
+N*2+4	0x00	Nepoužito, lze využít na kontrolní součet
+N*2+5	0x00	Nepoužito, lze využít na kontrolní součet

6.2.2.4 Stavové slovo

Stavové slovo zpravidla odesílá slave, vždy po dokončení činnosti, která byla iniciována vysláním *příkazu* do zařízení. Slave tím dává nadřazenému zařízení zprávu o úspěchu nebo neúspěchu akce vysláním chybového kódu. Struktura rámce se stavovým slovem je v tab. 6.4.

Tab. 6.4 Struktura stavového slova ProtV2

Offset	Data	Popis
+00	0x00	doplňek do sudé velikosti (padding)
+01	0x82	0x82 - Status
+02	(LO)	Nižší byte stavového slova
+03	(HI)	Vyšší byte stavového slova

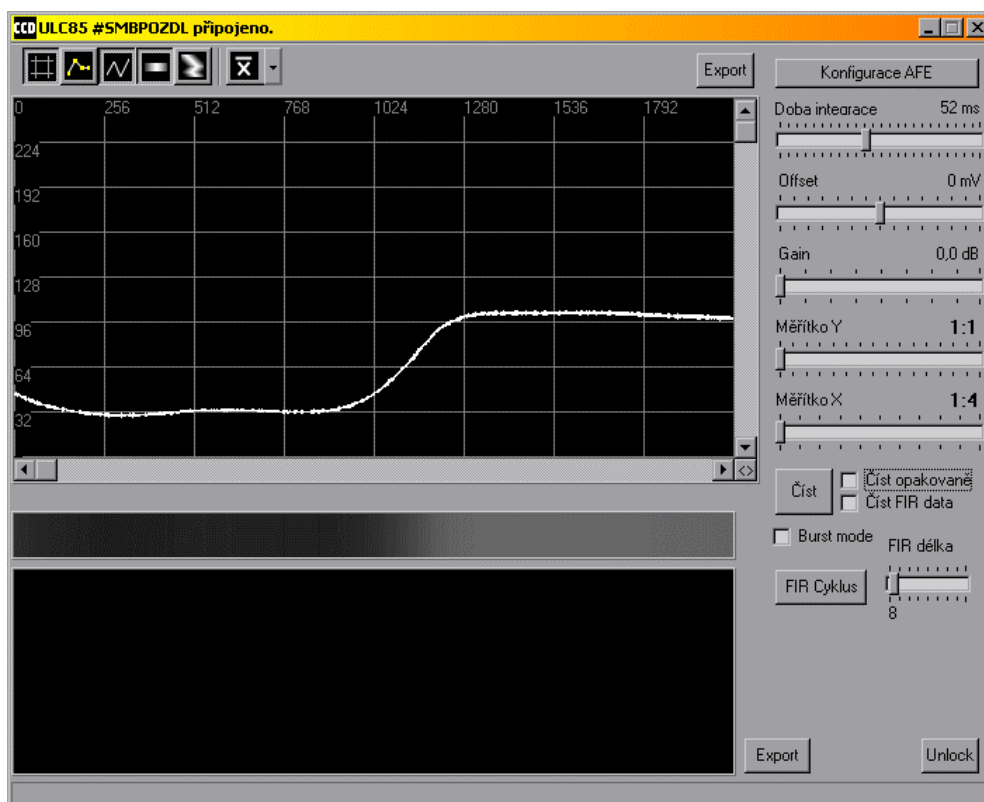
6.3 Zobrazovač pro ULC85

Pro testovací účely jsem napsal grafický zobrazovací software pro ovládání ULC85 s připojeným řádkovým CCD snímačem SONY ILX551B. Grafické uživatelské rozhraní je napsané v jazyce Delphi 6 a zkompilevané pro platformu WIN32. Program v zásadě umožňuje:

- § Zobrazit graficky a opticky průběh jasů na CCD snímači
- § Zaznamenávat změny jasového profilu v čase
- § Zvětšovat/zmenšovat výřez zobrazeného grafu
- § Průměrovat naměřená data v čase pomocí klouzavého průměru
- § Nastavovat parametry vstupních obvodů AFE a dobu integrace v reálném čase
- § Exportovat naměřená data do aplikace Microsoft Excel
- § Kurzorem přesně odečítat hodnoty z grafu

Na obr. 6.4 je náhled na okno ovládacího programu na PC. Těsně pod titulkovým pruhem je nástrojová lišta obsahující tlačítka na skrytí/zobrazení mřížky, zvýraznění bodů, zobrazení čar/bodů, zapnutí/vypnutí vizuálního zobrazení jasového profilu, zapnutí/vypnutí záznamu jasového profilu v čase a nastavení průměrování v čase. Pod nástrojovou lištou se nachází graf se zobrazeným jasovým průběhem. Levým tlačítkem myši lze graf posouvat tažením, pravým tlačítkem se zobrazí kurzor. Pod grafem je vizuální zobrazení jasového profilu a pod ním jeho záznam v čase. U grafu jsou pak ovládací prvky. Tlačítka na čtení a export dat, dále pak posuvníky k nastavení integrační doby, offsetu, zisku (gain) vstupního zesilovače převodníku a zvětšení grafu.

Tlačítko FIR Cyklus a posuvník vedle něho slouží k ovládání zkušebně naprogramovaného řízení snímače v módu FIR.

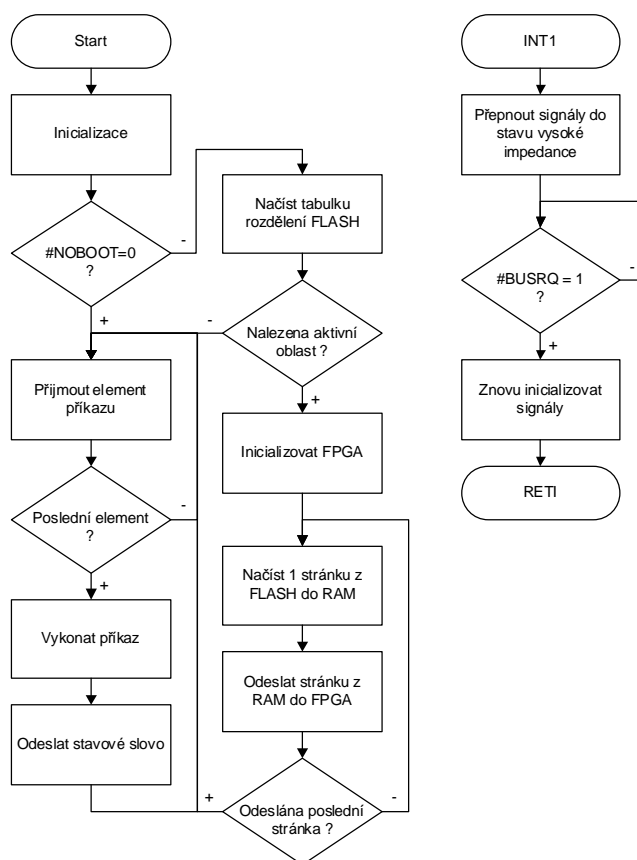


Obr. 6.4 Zobrazovač pro ULC85

6.4 Řídicí program pro SP2EDB

6.4.1 Program pro řídicí procesor

Obslužný program pro desku SP2EDB je celý napsán v jazyce assembler pro jednočipové mikrokontroléry Atmel AVR. Slouží k obsluze sériové paměti DataFlash, komunikaci s PC po USB nebo RS-232 prostřednictvím protokolu ProtV2, známého z ULC85, a ke konfigurování hradlového pole. Zjednodušený vývojový diagram programu je na obr. 6.5.



Obr. 6.5 Vývojový diagram programu pro SP2EDB

Po zapnutí napájení program inicializuje všechny své proměnné a nastaví vstupní a výstupní piny procesoru do výchozích stavů. Poté ovzorkuje pin #NOBOOT a v případě že není v log. 0, pokusí se zavést do FPGA první aktivní oddíl s konfigurací ve FLASH.

Následně program přejde do hlavní smyčky, kde se čeká na příjem elementů ProtV2. Poté se vykonávají příkazy podle přijatých elementů a po dokončení příkazu se odešle stavové slovo.

Podporované příkazy jsou:

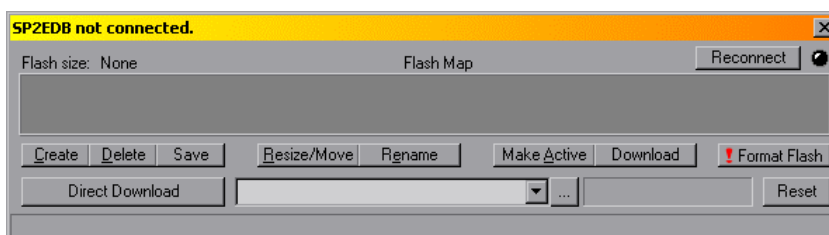
- **Identify** (povinný příkaz s číslem 0)
Jednoznačná identifikace připojeného zařízení.
- **Download FPGA**
Přímá konfigurace FPGA z komunikační linky.
- **Program Flash Page**
Naprogramování jedné stránky (256 bytů) v paměti FLASH.
- **Read Flash Page**
Přečtení jedné stránky v paměti FLASH.

- **Get SP2EDB Status**
Zjištění stavu FPGA a FLASH.

Podrobný popis struktury jednotlivých příkazů je uveden na konci zdrojového textu, který je na příloženém CD v adresáři \Sources\AVR\SP2EDB.

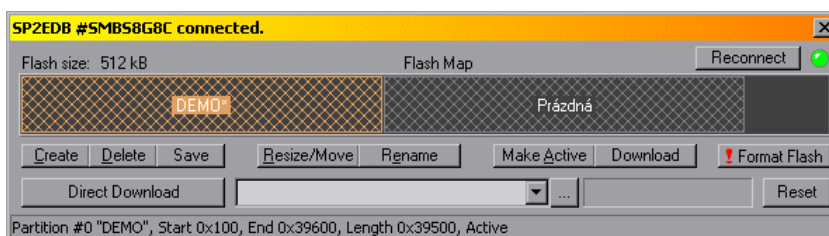
6.5 Manažer SP2EDB pro PC

K ovládání a správě desky SP2EDB byl napsán obslužný program SP2DL v jazyce Delphi 6. Po jeho spuštění se objeví okno podle obr. 6.6. Pokud je deska již připojena k PC pomocí USB kabelu, program ji automaticky najde a okno se změní podle obr. 6.7.



Obr. 6.6 Okno programu SP2DL

Připojíme-li desku když je program již spuštěn, je třeba kliknout na tlačítko **Reconnect**. To provede manuální pokus o znovunavázání spojení. Naváže-li se spojení úspěšně, rozsvítí se virtuální LED v pravém horním rohu.

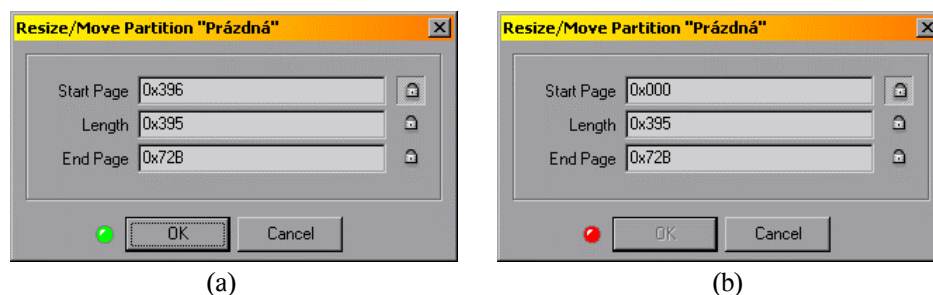


Obr. 6.7 Spojení navázáno, zobrazena mapa paměti FLASH

Poté se zobrazí mapa paměti FLASH, kde je možno vizuálně kontrolovat, kolik paměti je obsazeno a kolik jí zbývá. Na obr. 6.7 jsou vidět dva oddíly (partition) s názvy „DEMO“ a „Prázdná“. Oddíl DEMO je zobrazen oranžově a za názvem má zobrazenou hvězdičku *, zatímco prázdný oddíl je zobrazen šedivou barvou. To proto, že první oddíl je nastaven jako aktivní a tudíž při zapnutí napájení se automaticky zavede do FPGA.

Ovládání programu je intuitivní a myslím že není potřeba detailně popisovat všechny funkce. Zmíním se pouze o těch, které by nemusely být na první pohled jasné.

Kliknutím na tlačítko **Create** se vytvoří nový oddíl, zabírající celý první volný blok paměti. Ten je potom možno zmenšit na požadovanou velikost, nebo ho přesunout. Toho se docílí tlačítkem **Resize/Move**. Po jeho stisknutí se objeví okno podle obr. 6.8a.



Obr. 6.8 Okno změna velikosti/přesun oddílu

Jsou v něm tři měnitelné údaje. Počáteční stránka, velikost (v počtu stránek) a koncová stránka. Každý z údajů si lze zamknout kliknutím na ikonku zámku. Zamknutý údaj se nebude měnit. Chceme-li například posunout oddíl na začátek paměti bez změny velikosti, zamkneme políčko **Length** (velikost) a změníme údaj **Start Page**. Údaj **End Page** se bude automaticky měnit. Je-li v údajích chyba, změní se barva virtuální LED na červenou a změny se nedají potvrdit tlačítkem OK (obr. 6.8b).

Položka **End Page** je ve skutečnosti poslední stránka +1, čili číslo první další volné stránky. Hodnoty do zadávacích polí je možno psát v desítkové i šestnáctkové soustavě. Před číslo v šestnáctkové soustavě je potřeba napsat symbol 0x.

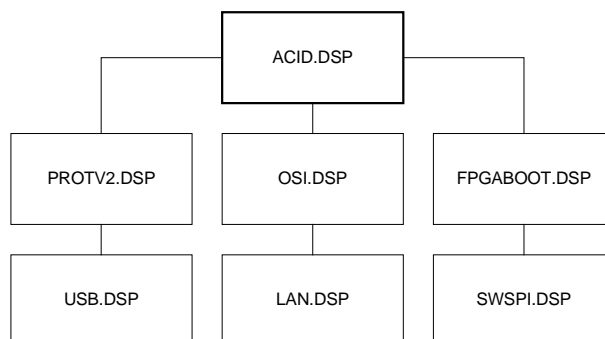
Přesunutím oddílu se však nepřesouvají data v něm uložená. Oddíl je v podstatě jenom rezervování místa v paměti a slouží jen jako informace pro řídicí procesor od jaké adresy a kolik stránek má odesílat do FPGA. Pokud se oddíl přesune, musí se do něj znovu uložit data.

Soubor s konfigurací se ukládá do vybraného oddílu tlačítkem **Download**. Také je možné soubor odeslat přímo do hradlového pole tlačítkem **Direct Download**. Požadovaný soubor je však nejprve potřeba nalistovat tlačítkem, vedle rozbalovacího seznamu historie, označeného [...].

6.6 Řídicí program pro ACID-IS

Obslužný program pro signálový procesor byl z části převzat z ULC85, protože je na ACID-IS z pohledu DSP mnoho věcí podobných. Na obr. 6.9 je naznačena hierarchická struktura programových modulů tak, jak spolu vzájemně komunikují. Např. modul PROTV2 byl převzat kompletně beze změny, protože je napsaný velice univerzálně.

Zdrojové texty od všech modulů lze nalézt v adresáři \Sources\DSP\ACID, na příloženém CD. Nyní si popíšeme jednotlivé moduly.



Obr. 6.9 Hierarchie programových modulů v ACID-IS

6.6.1 *ACID.DSP*

Hlavní modul. Obsahuje v sobě vazby a propojení na všechny ostatní moduly. Při startu inicializuje všechny interní a externí periferie, nastaví výchozí hodnoty proměnných a zavede konfiguraci z paměti FLASH do hradlového pole, pokud tam je.

Další funkce je obdobná jako u ULC85 nebo SP2EDB, neboť zde je také implementován přenosový protokol ProtV2. V hlavní smyčce se čeká na příchod elementů protokolu a pomocí stavového automatu se příkazy dekodují a zpracovávají. Specializované příkazy související se snímačem CCD se vykonávají mimo hlavní modul. Pro ně je vyhrazen modul CCD.DSP (viz kapitola 6.6.6)

6.6.2 *LAN.DSP*

Tento modul slouží jako ovladač rozhraní Ethernet. Obsahuje potřebné subrutiny pro základní ovládání obvodu, jako je:

- Inicializace obvodu
- Přejít do/z úsporného režimu
- Přijetí a odeslání paketu
- Čekání a dotaz na příchozí paket
- Komunikace s obvodem pro fyzickou vrstvu (PHY)

Modul však zajišťuje komunikaci jen na úrovni linkové vrstvy modelu OSI (viz kapitola 6.6.3). Podrobný popis práce s tímto ovladačem je popsán přímo ve zdrojovém kódu. Nyní zde v hrubých rysech naznačím algoritmy odesílání a příjmu paketu. Detailní popis lze nalézt v [15], kapitole 10.

Obvod LAN91C111 má v sobě integrováno 8KB paměti RAM, která tvoří buffer pro odchozí a příchozí pakety. Tato paměť je rozdělena na 4 bloky, každý po 2KB. Každý blok je určen pro jeden paket. Velikost bufferu je dostatečná, neboť maximální povolená délka

rámce na síti Ethernet je 1518 bytů. Jednotlivé bloky se dynamicky alokují a uvolňují podle potřeby buď automaticky, nebo manuálně.

Při odesílání paketu se nejprve pro odchozí paket alokuje jeden blok paměti v bufferu. Číslo alokovaného paketu se uloží do registru Packet Number a Pointer Register se nastaví na zápis do odchozího bufferu. Poté se zkopírují data paketu do bufferu přes datový registr a odešle se příkaz na zařazení paketu do fronty k odeslání. Pak již stačí počkat na dokončení odesílání a alokovanou paměť opět uvolnit. V ovladači je použito automatické uvolňování paměti pomocí bitu AUTO RELEASE v Control registru. Nastavení tohoto bitu způsobí, že se paměť po úspěšném odeslání paketu uvolní sama.

Příjem paketu je jednodušší, protože si obvod řídí alokaci paměti sám. Pro přijetí paketu je potřeba jen povolit příjem v registru RCR. Po příchodu paketu si obvod sám alokuje jeden blok paměti a paket do něj zapíše. Stačí čekat na přerušeni nebo na příznak došlého paketu v INTERRUPT registru, nastavit Pointer registr na čtení z příchozího bufferu a přes datový registr vyčíst data příštího paketu. Nakonec je potřeba manuálně uvolnit alokovanou paměť v bufferu.

Ještě bych zde chtěl zmínit komunikaci s obvodem fyzické vrstvy (PHY). V různých manuálech se totiž popis navzájem odlišuje. Komunikace s probíhá sériově a řídí se pomocí čtyř bitů MDOE, MCLK, MDI a MDO v MI registru v LAN. Sériové vysílání a příjem se musí řídit kompletně softwarově. Vzorový program pro komunikaci, napsaný v jazyce C a uvedený v [16] v kapitole 7.3.1, jsem ověřil a podle něho napsal komunikaci s PHY v jazyce assembler pro DSP.

6.6.3 OSI. DSP

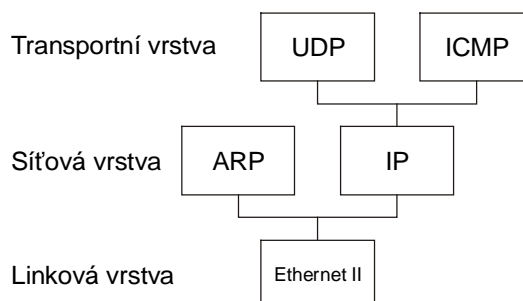
V tomto modulu je obsažena komunikace po Ethernetu pomocí standardních protokolů rodiny TCP/IP, ze síťové a transportní vrstvy referenčního modelu OSI. OSI je zkratka pro Open Systems Interconnection. Jedná se o zavedený standard, který rozčleňuje celou síťovou komunikaci do sedmi vrstev a pro každou z nich definuje její funkci.

Tab. 6.5 Vrstvy referenčního modelu OSI

Aplikační	Poskytuje služby aplikaci uživatele
Prezentační	Provádí potřebný převod a překlad dat, šifrování a dešifrování, případně komprimaci
Relační	Řídí spojení a jeho průběh
Transportní	Zajišťuje komunikaci koncových bodů
Síťová	Rozhoduje o koncovém směrování a adresování
Linková	Přenáší bloky dat (rámce)
Fyzická	Fyzický přenos dat v čase, kódování signálu

Protokoly používané tímto modulem jsou ARP, IP, ICMP a UDP. Tyto internetové protokoly se přenáší po Ethernetu zapouzdřené do rámců s názvem Ethernet II (viz [18]).

Obsluha všech těchto protokolů není v žádném případě úplná. Důraz byl kladen hlavně na rychlost a také na to, aby obsluha nezabírala v paměti procesoru příliš mnoho místa. Podporovány jsou proto jen ty nejnужnější části protokolů a celá komunikace má tím pádem mnohá omezení, které si popíšeme v průběhu kapitoly.



Obr. 6.10 Hierarchie protokolů v modelu OSI

Protokol ARP (viz [19]) slouží k překladu logické síťové adresy na fyzickou. Ačkoli logických i fyzických adres je několik druhů, v tomto modulu je implementována pouze možnost odpovídat na dotazy ARP, které žádají o překlad logické adresy IP na fyzickou adresu MAC. Lidově řečeno, když PC vyšle oběžník (paket určený všem zařízením) s dotazem „Kdo má IP adresu 10.0.0.9?“, deska ACID dokáže odpovědět „IP adresu 10.0.0.9 mám já a mám MAC adresu 00-90-C2-C1-CA-E5.“

Protokol IP (viz [21]) sám o sobě není určen pro přenos uživatelských dat. Je určen pro zapouzdření protokolů vyšších vrstev, které používají logické adresování pomocí IP adres a musí mít před sebou jeho hlavičku. Zajišťuje také fragmentaci dlouhých datagramů na pakety a na cílové straně je odpovědný za jejich opětovné složení. Všecky ostatní protokoly vyšších vrstev běží právě nad protokolem IP.

Také implementace tohoto protokolu je omezená. Nejsou podporovány hlavičky IP větší, než 20 bytů (které obsahují Options). Není podporováno skládání fragmentovaných datagramů a nevyhodnocuje se pole Time To Live ani Type Of Service (pro vysvětlení pojmů, viz [21]). Naprostá většina paketů v síti Internet však těmto omezením vyhovuje.

Další z řady je protokol ICMP (viz [20]), který je již nad protokolem IP a je rovněž implementovaný jen částečně. Pomocí tohoto protokolu totiž většina zařízení včetně PC zjišťuje dostupnost koncového bodu příkazem zvaným „ping“. Například příkaz „ping 10.0.0.9“ požádá koncové zařízení s logickou adresou 10.0.0.9 pomocí protokolu ICMP o opakování paketu, tzv. „Echo request“. Koncové zařízení pouze v hlavičce zamění příjemce a odesilatele, změní typ na „Echo reply“ a paket zase odešle zpátky. Tím si hostitelské zařízení ověří, že koncové zařízení funguje a je schopno komunikovat.

Pro programátora je přístupný pouze protokol UDP (viz [22]). Dotazy protokolů ARP a ICMP jsou automaticky modulem odpovězeny, bez nutnosti programového zásahu. Protokol UDP je nespojový, tzn. že se nemusí před zahájením komunikace navazovat žádné

spojení. Není u něho však také garantováno doručení paketu. Konzistenci přenášených dat si musí hlídat aplikační software. Popis komunikace pomocí protokolu UDP je popsán v samotném zdrojovém textu modulu.

Původně jsem chtěl osvědčený protokol ProtV2 použít i na Ethernetu nad protokolem UDP. Komunikace po Ethernetu v rámci vrstev OSI však probíhá jinak, než jsem si v době návrhu protokolu myslel. ProtV2 je optimalizovaný na spojení se zařízením s bytově orientovaným přenosem. Ethernet je však striktně orientovaný na paketový přenos a odesílání po jednom bytu by se muselo emulovat pomocí dalšího odesílacího buferu. Nejen že by se tím celá komunikace znatelně zpomalila, ale také by se zvýšily nároky na paměť procesoru. Navíc se souběžně s mojí diplomovou prací na katedře měření vyvíjí nový komunikační standard, který by měl být v budoucnu použit na všech zařízeních, určených k přenosu obrazu. Proto jsem od použití neperspektivního protokolu ProtV2 na UDP ustoupil a ponechal komunikaci přes UDP volně v rukou programátora.

6.6.4 USB.DSP

Modul obsahuje pouze subrutiny na odeslání znaku, přijetí znaku a dotaz na stav přijímacího bufferu řadiče USB. Tyto subrutiny jsou volány z programového modulu PROTV2 (viz kapitola 6.6.5).

6.6.5 PROTV2.DSP

Modul slouží pro komunikaci pomocí protokolu ProtV2 a je kompletně převzatý z desky ULC85. Popis modulu lze nalézt v kapitole 6.1.6.

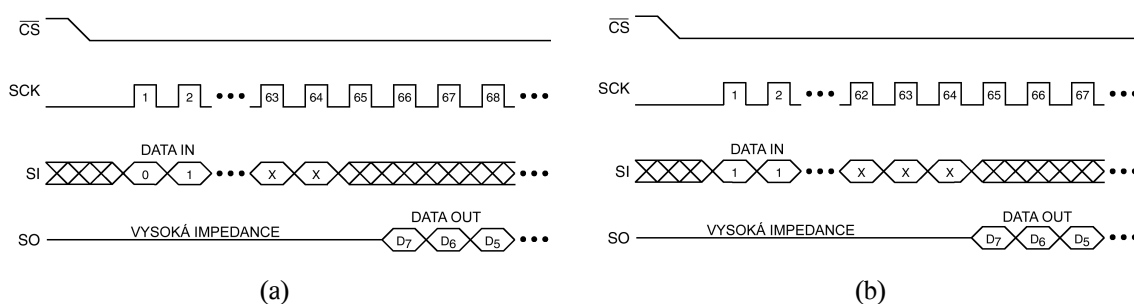
6.6.6 CCD.DSP

Toto je uživatelem programovaný modul pro řízení konkrétního CCD snímače. Má velmi podobné aplikační rozhraní, jako modul CCD_INTERFACE z desky ULC85 (viz kapitola 6.1.4). Tento modul však již nemusí řídit CCD snímač sám, ale pouze odesílá příkazy a nastavení do videoprocessoru v hradlovém poli. Detailní popis lze najít přímo ve zdrojovém textu modulu.

6.6.7 SWSPI.DSP

Signálový procesor ADSP-2188 nemá integrovaný SPI interface, proto je nutné ho pro komunikaci s DataFlash softwarově emulovat. Tuto emulaci, spolu se specifickými funkcemi k obsluze sériové paměti, zajišťuje právě tento modul.

Zatímco na SP2EDB byla paměť DataFlash používána v SPI módu 0, protože to přímo umožňoval řadič SPI integrovaný v procesoru, k softwarové emulaci se více hodí režim „Inactive Clock Polarity Low“ (dále jen ICPL). DataFlash navíc umí pracovat v režimech SPI mód 3 a „Inactive Clock Polarity High“, které však nejsou na žádné z desek využity. Všechny tyto režimy jsou vysvětleny v [14]. ICPL se od SPI módu 0 liší pouze v přechodu mezi příjmem a vysíláním dat. Lépe je to vidět na obr. 6.11.



Obr. 6.11 Porovnání režimů Inactive Clock Polarity Low (a) a SPI mód 0 (b)

V režimu SPI mód 0 (obr. 6.11b) datový výstup SO přechází ze stavu vysoké impedance se spádovou hranou hodinového impulsu posledních zapisovaných dat. Pro softwarovou emulaci se však více hodí režim ICPL (obr. 6.11a), kde jsou první vyslaná data až po dalším hodinovém taktu. Umožňuje to lépe oddělit subrutiny pro odeslání a přijetí bytu z SPI. Programově se mezi těmito dvěma režimy volí instrukčním kódem, odeslaným do DataFlash.

6.6.8 FPGABOOT.DSP

Tento modul odesílá do hradlového pole konfiguraci uloženou v DataFlash. Data se do FPGA odesílají vždy, i když v paměti není nic nahráno. Hradlové pole má na začátku konfiguračního souboru synchronizační sekvenci a na konci kontrolní součet. Pokud je v zapisovaných datech nenajde, konfigurace se nezahájí.

7. SHRNU TÍ POSTŘEHŮ, TRIKŮ A NEDOSTATKŮ

7.1 Jumpery na ULC85

Monitor pro DSP je v paměti FLASH nahrán od adresy 0 a k jeho zavedení do DSP je potřeba oba jumpery BANK na ULC85 propojit.

7.2 Signál #BUSRQ na SP2EDB

Viz kapitola 4.4.9. Signál hradlového pole #BUSRQ musí být držen v log. 1 pro správnou funkci řídicího procesoru. Log. 0 na tomto signálu odpojuje procesor od sběrnic sdílených s FPGA.

7.3 ISP programování mikroprocesoru na SP2EDB

Řídicí procesor je zaletovaný přímo na DPS a programuje se pomocí ISP konektoru. Při programování přes ISP může uživatel nechtěně zamknout procesor tak, že jej nebude možno dále programovat pomocí ISP. Je proto třeba věnovat případně změně firmware zvláštní pozornost.

7.4 Obvod Power Assist na SP2EDB

Viz kapitola 4.4.10. V návrhu desky SP2EDB jsem počítal s obvodem zvaným Power Assist pro řízení průběhu napájecího napětí FPGA při zapnutí desky. Ukázalo se, že žádný takový obvod není potřeba. Stačí běžný lineární stabilizátor.

7.5 Pin CLKUSR

Pin CLKUSR slouží jako vstup uživatelských hodin do FPGA ze snímačové desky. Najít ho lze jako pin č. 39 na konektoru pro snímačovou desku na deskách SP2EDB a ACID-IS. Na desce ULC85 je tento pin připojen na zem (GND). Pokud snímačová deska počítá s využitím tohoto signálu a nemá možnost jeho odpojení, nelze ji pak připojit na ULC85.

7.6 EEPROM u LAN na ACID-IS

Viz kapitola 5.3.13. EEPROM u řadiče LAN na desce ACID-IS se mi nepodařilo pomocí DSP z neznámých důvodů naprogramovat. Výchozí hodnoty se do registrů v LAN zapisují pomocí softwaru.

7.7 Čtení z LAN

Registry v LAN se čtou pomocí dvou instrukcí čtení z portu těsně za sebou. Podrobně je to vysvětleno v kapitole 5.3.13.

7.8 Přenos přes Ethernet

Signálový procesor pracuje pouze se slovy o délce 16 bitů. Z tohoto důvodu není podporován přenos lichého počtu bytů přes Ethernet.

7.9 Popis komunikace s PHY

V různých manuálech obvodu LAN91C111 se uvádí algoritmy komunikace s PHY, které se navzájem odlišují. Vzorový program uvedený v [16] je vyzkoušený a funguje.

7.10 Oddíly v DataFlash na SP2EDB

Viz kapitola 6.5. Přesouváním oddílů v paměti FLASH na desce SP2EDB se nepřesouvají data v oddílech. Oddíly slouží jen jako informace pro řídicí procesor, kde jsou uloženy konfigurace pro FPGA a jak jsou dlouhé.

7.11 Omezení Xilinx ISE WebPack

Vývojové prostředí Xilinx ISE, volná verze WebPack, dokáže provádět syntézu pouze pro hradlová pole do velikosti 300 000 ekv. hradel. FPGA použité na desce SP2EDB a ACID-IS je tedy největší možné, které volná verze vývojového prostředí podporuje.

8. MOŽNOSTI A INSPIRACE PRO DALŠÍ PRÁCI

V době návrhu zařízení jsem měl pouze strohé informace o standardu CameraLink, proto jsem se ho rozhodl neimplementovat. Dodatečně by se však toto rozhraní dalo doplnit jako externí modul připojený na konektor pro CMOS snímač. Minimální verze tohoto rozhraní však obsahuje 35 řídicích a komunikačních vodičů, ale konektor pro CMOS snímač jich má pouze 32. Některé z nich je však možno nezapojit.

Komunikace přes Ethernet pomocí protokolu UDP je z principu nespolehlivá a doručení dat se musí hlídat programově. Možností pro vylepšení by byla implementace protokolu TCP, který má zajištěný přenos dat. Obsluha tohoto protokolu je však poměrně složitá a nejsem si jist, zda by nezabírala v paměti DSP příliš mnoho místa.

Drobné zrychlení přenosu přes Ethernet by také zajistila podpora předčasného příjmu/odeslání (Early Receive/Early Transmit). Jedná se o signalizaci příchodu paketu ještě před dokončením jeho příjmu a započítání odeslání paketu před dokončením jeho zápisu do odchozího bufferu. Obvodový radič obě tyto metody hardwarově podporuje.

Další možnou prací by mohl být návrh univerzálního videoprocessoru, který by umožňoval připojení libovolného plošného či řádkového CCD nebo CMOS snímače, které by dokázal řídit ve standardním i v nestandardních režimech. Dále by umožňoval základní předzpracování videosignálu, jako je třeba průměrování v čase, nebo filtraci signálu pomocí číslicového filtru v reálném čase. Práce na návrhu videoprocessoru by však byla takového rozsahu, že by sama mohla být zadáním diplomové práce.

9. ZÁVĚR

Účelem práce zde nebyla ani tak hloubková a detailní analýza a použití složitých algoritmů pro řízení CCD a zpracování obrazu. Po dohodě s vedoucím dipl. práce jsem se soustředil hlavně na návrh takového zařízení, které plně umožní implementaci takových algoritmů.

Průběh mé diplomové práce se dá popsat několika body. Prvním úkolem bylo navrhnout a zapojit zařízení, kterým by se dal ovládat libovolný CCD snímač pomocí signálového procesoru.

Tímto zařízením je univerzální řádková kamera ULC85, která je řízena signálovým procesorem ADSP-2185M s výpočetním výkonem 72 MIPS. Program se uchovává v paměti FLASH o velikosti 512 KB. Pro řízení CCD snímače je k dispozici 16 programovatelných výstupních signálů v úrovni 5V TTL. Vstupní analogovou část tvoří speciální rychlý 14bitový AD převodník pro CCD snímače s rychlostí až 15 MSa/s. Jednotka je také vybavena 8bitovým vstupně/výstupním portem pro obecné použití. Pro komunikaci a přenos obrazových dat je použito rozhraní USB. Sériová linka RS-232 je programově emulovaná a je primárně určena jen pro účely ladění. V případě potřeby je však také možné ji použít ke komunikaci.

ULC85 je plně funkční a je pro ni napsaná vzorová aplikace k řízení CCD snímače Sony ILX551B ve standardním režimu a v jednoduchém módu FIR s obdélníkovou impulzní odezvou. Ve standardním módu jsem dosáhl rychlosti vyčítání 3,6 milionu pixelů za sekundu. Čistě softwarové řízení snímače zajišťuje velkou flexibilitu desky. Pouze změnou jednoho programového modulu je možné desku používat s libovolným CCD snímačem.

Další část práce byla vývojová deska SP2EDB, která byla vyvinuta ve velmi krátkém čase (asi 14 dní). Během užívání desky jsem však zjistil, že některé její části by se daly vyřešit lépe. Pokud by se dělala verze 2 této desky, určitě bych na ní udělal resetovací tlačítko pro řídicí procesor, aby se nemuselo vypínat napájení pro znovunačtení konfigurace. Další signál, který by si býval zasloužil tlačítko je #NOBOOT, kterým se při startu zakazuje automatické načtení konfigurace z paměti FLASH.

Jinak je tato deska velice univerzální prostředek, na kterém lze ladit návrhy pro hradlové pole a již v době psaní tohoto textu je takto využívána v laboratoři videometrie. Na desce je také konektor pro připojení CCD snímače, který je kompatibilní s konektorem na ULC85. Pomocí této desky se dají Díky možnosti její snadné rekonfigurace je však její použití nekončí jen v oblasti videometrie. Snadno se do ní dá naprogramovat např. logický analyzátor, nebo po připojení rychlého AD převodníku, i digitální osciloskop či spektrální analyzátor. Pro mne tato deska posloužila k ověření všech potřebných zapojení pro hradlové pole, pro vyzkoušení algoritmu konfigurace a obsluhy sériové paměti DataFlash.

Do 512KB paměti FLASH této desky se vejdu dva různé konfigurační soubory pro FPGA, což dává možnost rychle přepínat mezi dvěma verzemi konfigurace. Přímé načtení konfiguračního souboru do hradlového pole po lince USB trvá přibližně 4 sekundy. Načtení

tohoto souboru při startu z FLASH do FPGA trvá asi 2 sekundy. Uložení konfigurace přes USB do FLASH trvá asi 30 sekund.

Posledním úkolem mé práce byl návrh zařízení, které kompletně splňuje požadavky zadání. Jednotka ACID-IS v sobě kombinuje desky ULC85 a SP2EDB. Obsahuje nejvyšší model signálového procesoru řady ADSP-218x s výpočetním výkonem 80 milionů instrukcí za sekundu. Dále programovatelné hradlové pole rodiny Spartan-IIe, kde 6912 logických elementů dokáže nahradit až 300 000 hradel standardní logiky. K němu je navíc připojena rychlá statická paměť RAM o velikosti 1024 KB a dobou přístupu 10 ns. A protože jsou v zapojení přítomny analogové i rychlé digitální obvody, je celá jednotka navržena na čtyřvrstvé desce plošných spojů.

O komunikaci s okolím se starají linky USB, RS-232 a Fast-Ethernet. Rozhraní Fast-Ethernet komunikuje pomocí základně implementovaných internetových protokolů rodiny TCP/IP, jako je ARP, IP, ICMP a UDP. Komunikací protokolem UDP jsem dosáhl maximální rychlosti 5200 KB/s, ale optimalizací přenosu by se mělo dát dosáhnout rychlosti přibližně dvojnásobné. Protokol UDP má navíc minimální režii 2,8% přenosové kapacity. Přenos velkou rychlostí přes UDP velmi zatěžuje síť a také procesor PC. Pokud se procesor v nadřazeném PC zatíží jinou činností, dochází ke ztrácení paketů a přenosová rychlost prudce klesá. Když jsem např. zatížil PC přibližně na 80%, ztrácelo se až 95% paketů a rychlost klesala na stovky KB/s. Ztrácení paketů je však otázka softwaru na PC. Použitím vlastního protokolu síťové vrstvy modelu OSI a pomocí přímého přístupu k ovladači síťové karty by se dal přenos optimalizovat.

10. SEZNAM POUŽITÉ LITERATURY

- [1] Radil, T.: *Řádková kamera ULC84*, Diplomová práce ČVUT FEL – K338, 2002
- [2] Kazda, M.: *Modul s ADSP2185MKST*, Semestrální práce ČVUT FEL – K338, 2002
- [3] Fischer, J.: *Optoelektronické senzory a videometrie*, Vydavatelství ČVUT, 2002
- [4] Firemní: *ADSP-218x DSP Hardware Reference*, Analog Devices, 2001
- [5] Firemní: *ADSP-218x DSP Instruction Set Reference*, Analog Devices, 2001
- [6] Firemní: *+3V Voltage Monitoring μ P Supervisory Circuits*, katalogový list ADM706P/R/S/T, Analog Devices, <http://www.analog.com>
- [7] Firemní: *Complete 14-Bit CCD/CIS Signal Processor*, katalogový list AD9822, Analog Devices, <http://www.analog.com>
- [8] Firemní: *FT245BM USB FIFO*, katalogový list DS245B, Future Technology Devices International, 2002, <http://www.ftdichip.com>
- [9] L. D., G. G.: *Simulating an RS-232 UART using the Synchronous Serial Ports on the ADSP-21xx Family DSPs*, aplikační poznámka EE-60, Analog Devices, 1999, <http://www.analog.com>
- [10] Forstner, P.: *Input and Output Characteristics of Digital Integrated Circuits*, aplikační poznámka SDYA010, Texas Instruments, 1996, <http://www.ti.com>
- [11] Firemní: *Spartan-II 1.8V FPGA Family*, Katalogový list DS077, Xilinx, 2003, <http://www.xilinx.com>
- [12] Goldblatt, K., Rinck, J., Sanders, H.: *Power-Assist Circuits for the Spartan-II and Spartan-IIe Families*, aplikační poznámka XAPP451, Xilinx, 2001
- [13] Goldblatt, K.: *Power-On Requirements for the Spartan-II and Spartan-IIe Families*, aplikační poznámka XAPP450, Xilinx, 2001
- [14] Firemní: *4-megabit 2.5-volt or 2.7-volt DataFlash*, Katalogový list AT45DB041B, Atmel, 2004
- [15] Firemní: *LAN91C111 10/100 Non-PCI Ethernet Single Chip MAC + PHY*, Katalogový list, SMSC, 2004
- [16] Firemní: *SMSC LAN91C111 32/16/8-Bit Three-In-One Fast Ethernet Controller*, aplikační poznámka 9.6, SMSC, 2004
- [17] Osterloh, H.: *TCP/IP Kompletní průvodce*, SoftPress, 2003
- [18] Hornig, Ch.: *A Standard for the Transmission of IP Datagrams over Ethernet Networks*, Network Working Group RFC894, 1984
- [19] Plummer, D.: *An Ethernet Address Resolution Protocol*, Network Working Group RFC826, 1982
- [20] Postel, J.: *Internet Control Message Protocol*, Network Working Group RFC792, 1981

- [21] Postel, J.: *Internet Protocol*, Network Working Group RFC791, 1981
[22] Postel, J.: *User Datagram Protocol*, Network Working Group RFC768, 1980

11. SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ

ACID-IS	(All-purpose Control and Interface Device for Image Sensors) víceúčelová ovládací a komunikační jednotka pro obrazové snímače
AFE	(Analog Front-End) analogově číslicový převodník pro snímače CCD
ARP	(Address Resolution Protocol) protokol pro překlad adres v síti Internet
ASIC	(Application Specific Integrated Circuit) zakázkový integrovaný obvod
CCD	(Charge Coupled Device) nábojově vázaný prvek
CDS	(Correlated Double Sampling) korelované dvojité vzorkování
CMOS	(Complementary Metal Oxide Silicon) technologie výroby polovodičových součástek
CPLD	(Complex Programmable Logic Device) programovatelný logický obvod s konfigurací v paměti EEPROM
DPS	Deska plošných spojů
DSP	(Digital Signal Processor) signálový procesor
EEPROM	(Electrically Erasable Programmable Read-Only Memory) elektricky mazatelná programovatelná paměť pouze ke čtení
FIFO	(First In First Out) typ dvoubránové paměti RAM - fronta
FIR	(Finite Impulse Response) filtr s konečnou impulsní odezvou, nebo režim řízení CCD snímače
FPGA	(Field Programmable Gate Array) programovatelné hradlové pole s konfigurací v paměti RAM
GAL	(Generic Array Logic) programovatelný logický obvod
GPIO	(General Purpose Input Output) vstupy a výstupy s obecným využitím
HCMOS	(High-CMOS) technologie výroby polovodičových součástek
HDL	(Hardware Description Language) jazyk pro popis hardwaru
ICMP	(Internet Control Message Protocol) internetový řídicí protokol
ICPL	(Inactive Clock Polarity Low) režim rozhraní SPI
IP	(Internet Protocol) internetový komunikační protokol
JTAG	(Joint Test Action Group) port pro diagnostiku a programování logických obvodů

LAN	(Local Area Network) lokální síť, v tomto dokumentu také řadič Ethernetu
LED	(Light Emitting Diode) světlo emitující dioda
LVDS	(Low Voltage Differential Swing) standard pro přenos dat po symetrické lince
LVPECL	(Low Voltage Pseudo Emitter-Coupled Logic) standard pro přenos dat po symetrické lince
LVTTL	(Low Voltage Transistor-Transistor Logic) technologie výroby polovodičových součástek
MAC	(Media Access Control) řízení na úrovni linkové vrstvy sítě, nebo také hardwarová adresa v síti
MIPS	(Millions Instructions Per Second) miliony instrukcí za sekundu
MSa/s	(Mega Samples Per Second) miliony vzorků za sekundu
NIC	(Network Interface Controller) Radič síťového rozhraní Ethernet
OSI	(Open Systems Interconnection) popis referenčního modelu Internetu
PC	(Personal Computer) osobní počítač
PHY	(PHYsical layer) obvod řízení fyzické vrstvy Ethernetu
RAM	(Random Access Memory) paměť s náhodným přístupem
SHA	(Sample and Hold) jednoduché vzorkování
SP2EDB	(Spartan-2E Development Board) vývojová deska pro Spratan-2E
SPI	(Serial Peripheral Interface) Sériové rozhraní vyvinuté firmou Motorola
SPORT	(Serial PORT) Synchronní sériové rozhraní uvnitř ADSP-218x
TCP	(Transmission Control Protocol) protokol pro zajištěný přenos dat po Internetu
TDI	(Time Delayed Integration) režim řízení CCD snímače
UART	(Universal Asynchronous Receiver/Transmitter) řadič sériové linky
UDP	(User Datagram Protocol) protokol pro nezajištěný přenos dat po Internetu
ULC	(Universal Line-scan Camera) univerzální řádková kamera
USB	(Universal Serial Bus) univerzální sériová sběrnice
VHDL	(Very high speed integrated circuits HDL)
#	Před názvem signálu značí jeho negativní logiku. Tedy # <i>PROG</i> je to samé jako \overline{PROG} .
0x12A	Symbol 0x předchází číslu v šestnáctkové (hexadecimální) soustavě.

12. SEZNAM OBRÁZKŮ

Obr. 3.1 Blokové schéma ULC85.....	9
Obr. 3.2 Zapojení výstupních budičů na ULC85.....	11
Obr. 3.3 Připojení AFE ke sběrnici procesoru ADSP-2185M	12
Obr. 3.4 Časování AFE pro tříkanálový režim	13
Obr. 3.5 Zapojení řadiče sběrnice USB na ULC85	14
Obr. 3.6 Zapojení řadiče USB po úpravě	15
Obr. 3.7 Zapojení budiče sérové linky na ULC85.....	16
Obr. 3.8 Zapojení dohlížecího obvodu na ULC85.....	16
Obr. 3.9 Uživatelské vstupy a výstupy (GPIO) na ULC85	17
Obr. 3.10 Detailní zapojení dekódovací logiky ULC85	18
Obr. 3.11 Zapojení konektoru pro CCD snímač na ULC85	19
Obr. 3.12 Napájecí zdroj ULC85.....	20
Obr. 3.13 Zapojení snímačové desky pro CCD snímač ILX551/751	20
Obr. 4.1 Blokové schéma SP2EDB	22
Obr. 4.2 Značení obvodů Spartan-IIe.....	23
Obr. 4.3 Připojení SRAM ke hradlovému poli FPGA	24
Obr. 4.4 Zapojení výstupních registrů a konektor pro CCD na SP2EDB.....	25
Obr. 4.5 Zapojení konektoru pro snímačovou desku na SP2EDB	26
Obr. 4.6 Zapojení tlačítek a signalizačních LED na SP2EDB.....	27
Obr. 4.7 Zapojení konektoru pro obecné použití na SP2EDB	27
Obr. 4.8 Zapojení konektoru pro připojení externí analogové části na SP2EDB	28
Obr. 4.9 Rozhraní USB na SP2EDB.....	28
Obr. 4.10 Zapojení budiče sériové linky RS-232 na SP2EDB.....	29
Obr. 4.11 Zapojení řídicího procesoru na SP2EDB	30
Obr. 4.12 Připojení sériové paměti DataFlash k CPU na SP2EDB	31
Obr. 4.13 Proudová špička při zapnutí napájení FPGA.....	32
Obr. 4.14 Doba náběhu napětí na FPGA Spartan-IIe	32
Obr. 4.15 Napájecí zdroj a obvod ‚Power Assist‘ na SP2EDB.....	33
Obr. 4.16 Napájecí zdroj SP2EDB po úpravě	33
Obr. 5.1 Blokové schéma ACID-IS.....	34
Obr. 5.2 Zapojení výstupních budičů registru řízení CCD na ACID-IS.....	36
Obr. 5.3 Zapojení AD převodníku (AFE) na ACID-IS.....	36
Obr. 5.4 Časový diagram odesílání pomocí SPORTu signálového procesoru.....	37
Obr. 5.5 Časový diagram konfigurace FPGA	37
Obr. 5.6 Signály pro konfiguraci FPGA, upravené v CPLD	37
Obr. 5.7 Zapojení pomocných obvodů u FPGA na ACID-IS	38
Obr. 5.8 Zapojení statické paměti RAM na ACID-IS	38
Obr. 5.9 Zapojení obvodu CPLD na ACID-IS	39
Obr. 5.10 Dohlížecí obvod – Watchdog na ACID-IS.....	40

Obr. 5.11 Připojení sériové paměti DataFlash k DSP na ACID-IS.....	40
Obr. 5.12 Konektor pro obecné použití na ACID-IS.....	41
Obr. 5.13 Zapojení budiče sériové linky RS-232 na ACID-IS	41
Obr. 5.14 Zapojení řadiče rozhraní USB na ACID-IS	42
Obr. 5.15 Překlad adres mezi DSP a LAN.....	42
Obr. 5.16 Adresování řadiče Ethernetu.....	43
Obr. 5.17 Zapojení řadiče Fast-Ethernet na ACID-IS	44
Obr. 5.18 Zjednodušený časový diagram čtení z LAN	45
Obr. 5.19 Vynucení čtecího cyklu dvojitým čtením z LAN	45
Obr. 5.20 Zapojení analogové části fyzické vrstvy Ethernetu na ACID-IS	46
Obr. 5.21 Zapojení napájecího zdroje na ACID-IS.....	46
Obr. 6.1 Hierarchie programových modulů ULC85	47
Obr. 6.2 Vývojový diagram základního modulu ULC85	48
Obr. 6.3 Časový průběh zápisu do registru v AFE.....	49
Obr. 6.4 Vývojový diagram programu pro SP2EDB	54
Obr. 6.5 Okno programu SP2DL	55
Obr. 6.6 Spojení navázáno, zobrazena mapa paměti Flash.....	55
Obr. 6.7 Okno změna velikosti/přesun oddílu	56
Obr. 6.8 Hierarchie programových modulů v ACID-IS.....	57
Obr. 6.9 Hierarchie protokolů v modelu OSI	59
Obr. 6.10 Porovnání režimů Inactive Clock Polarity Low (a) a SPI mód 0 (b)	61

13. SEZNAM TABULEK

Tab. 3.1 Porovnání procesorů řady ADSP218x.....	10
Tab. 3.2 Mapování periférií ULC85 do IO prostoru DSP.....	18
Tab. 4.1 Porovnání hradlových polí značek Altera a Xilinx	21
Tab. 5.1 Mapování paměti FLASH do adresového prostoru DSP na ACID-IS.....	35
Tab. 6.1 Struktura příkazu ProtV2.....	50
Tab. 6.2 Struktura datového slova ProtV2	51
Tab. 6.3 Struktura datového paketu ProtV2.....	51
Tab. 6.4 Struktura stavového slova ProtV2	52
Tab. 6.5 Vrstvy referenčního modelu OSI	58