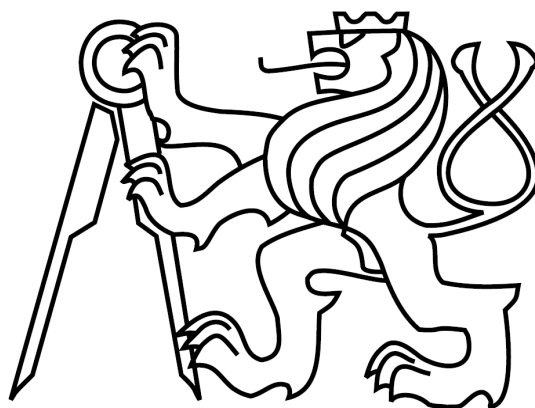


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA ELEKTROTECHNICKÁ



Diplomová práce

Rychlý osciloskop postavený s pomocí
FPGA

Bc. Jaroslav Borecký

Vedoucí práce: Ing. Pavel Kubalík, Ph.D.

Studijní program: Elektrotechnika a informatika strukturovaný
magisterský

Obor: Výpočetní technika - projektování číslicových systémů

květen 2008

Poděkování

Chci poděkovat vedoucímu mé diplomové práce Ing. Pavlovi Kubalíkovi, za odborné rady a čas který se mnou strávil při konzultacích. Dále chci poděkovat své rodině, která mě po celou dobu tvorby diplomové práce podporovala a přispěla k vymyšlení tohoto tématu.

Prohlášení

Prohlašuji, že jsem zadanou diplomovou práci zpracoval sám s přispěním vedoucího práce, konzultanta a používal jsem pouze literaturu v práci uvedenou. Dále prohlašuji, že nemám námitek proti půjčování nebo zveřejňování mé diplomové práce nebo její části se souhlasem katedry.

Datum:

.....

podpis

Abstrakt:

Tématem této práce je návrh a realizace levného jednobanového 50MHz digitálního osciloskopu, postaveného s pomocí desky Starter board s obvodem Spartan3A(3E). Pro převod dat je navrhnuto a postaveno modul s převodníkem, který lze připojit k desce s FPGA. Pomocí desky Starter board jsou naměřená data zobrazována přímo na monitor, dále tato deska obstarává ovládání modulu pomocí počítačové myši. Ovládání a komunikace s některými periferiemi je realizována pomocí embedded procesoru microblaze a dostupných IP core. Pro zobrazování a příjem dat je vytvořeno uživatelské IP core.

Abstract:

Topic of this thesis is proposal and realization of cheap one-channel 500Mhz digital oscilloscope. The oscilloscope is realized on Spartan3A(3E) Starter Board. For data conversion is proposed and realized module with convertor which can be connected to board with FPGA. Showing of measured data and controlling the module by computer mouse is realized with Starter board. Controlling and communication with some periferies is realized by embedded microprocessor microblaze and available IP cores. For showing and receiving of data is designed user IP core.

Obsah:

Prohlášení.....	5
Abstrakt:.....	7
Abstract:.....	7
1. Úvod.....	12
1.1. Pokyny pro vypracování diplomové práce.....	12
1.2. Cíle práce.....	12
1.2.1. Implementační cíl.....	12
1.2.2. Osobní cíle.....	12
2. Analýza.....	13
2.1 AD převodník.....	13
3. Návrh řešení.....	15
3.1 Modul.....	15
3.1.1 Analogová část modulu.....	15
3.1.2 Digitální část.....	15
3.1.3 Návrh schématu a plošného spoje.....	16
3.2 Vývojový kit.....	16
4. Řešení.....	17
4.1 Modul.....	17
4.1.1 Analogová část.....	17
4.1.1.1 Napájení.....	17
4.1.1.2 Vysokofrekvenční dělič.....	18
4.1.1.3 Operační zesilovač.....	20
4.1.2 Digitální část.....	21
4.1.2.1 Napájení.....	21
4.1.2.2 AD převodník.....	22
4.1.2.3 Clock buffer.....	26
4.1.2.4 Klopné obvody.....	27
4.1.2.5 Konektor.....	28
4.2 Vývojový kit.....	29
4.2.1 IP core VGA_controller.....	30
4.2.1.1 Blok Čtení z DDR.....	32
4.2.1.2 Blok Řádek – buffer.....	35
4.2.1.3 Blok VGA časování.....	37
4.2.1.4 Blok Příjem.....	39
4.2.1.4.1 Trigger.....	40
4.2.1.4.2 FSM_vstup.....	42
4.2.1.4.3 Blok BRAM.....	43
4.2.1.4.4 Blok Mux.....	43
4.2.1.4.5 Blok FSM_výstup.....	43
4.2.1.4.6 Blok Zobrazení.....	45
4.2.1.5 Blok Kurzory.....	46
4.2.1.6 Blok Myš.....	47
4.2.1.7 Blok FSL_řadič.....	48
4.2.1.8 Myš grafika.....	49
4.2.2 Microblaze.....	51
4.3. Návrh schématu a plošného spoje.....	51
5. Testování.....	53
6. Zhodnocení.....	54

<u>7. Závěr.....</u>	<u>55</u>
<u>8. Seznam použitých zkratk.....</u>	<u>56</u>
<u>9. Seznam použité literatury.....</u>	<u>56</u>
<u>10. Obsah CD.....</u>	<u>57</u>

1. Úvod

1.1. Pokyny pro vypracování diplomové práce

Navrhněte a zrealizujte levný jednokanálový kanálový 50 MHz osciloskop postavený s pomocí desky Starter board s obvodem Spartan3. Řešení bude obsahovat analogovou část zpracovávající vstupní analogový signál. Zobrazení průběhu bude umožněno pomocí monitoru připojeného přímo k desce. Pro FPGA obvod napište VHDL kód umožňující vzorkování dat pomocí převodníku, uložení dat do paměti a zobrazení průběhu na monitoru.

1.2. Cíle práce

1.2.1. Implementační cíl

Cílem mojí práce je navrhnout levný digitální osciloskop, kde hlavní část bude realizována na vývojové desce Spartan-3A Starter Kit, další částí bude přípojný modul, který bude obstarávat samotný převod analogového signálu. Tento modul by měl být kompatibilní i s vývojovým kitem Spartan-3E Starter Kit. Ovládání osciloskopu bude realizováno pomocí počítačové myši. Měřený signál bude zobrazován bez pomoci počítače přímo na monitor. Hlavní důraz je kladen na dosažení rychlého vzorkování a aby cena modulu nebyla příliš vysoká.

1.2.2. Osobní cíle

Získat nové poznatky v oblasti programování číslicových obvodů, zejména pak při návrhu systému s embedded procesorem pro FPGA (Field Programmable Gate Array). Zdokonalit se při práci s návrhovým systémem Xilinx EDK (Embedded Development Kit), návrhu číslicových obvodů v jazyce VHDL a návrhu analogově digitálního obvodu včetně jeho realizace.

2. Analýza

Již byla vyrobena spousta druhů osciloskopů s různými možnostmi, základní rozdělení je na analogové a digitální osciloskopy. V dnešní době se již vývoj věnuje více digitálním osciloskopům a jejich možnostem.

Většina rychlých a kvalitních osciloskopů je příliš drahá. Pro elektrotechniky amatéry jsou vyráběny levnější varianty, kde celý osciloskop není realizován v jednom zařízení a k jeho používání je potřeba počítače. Tyto osciloskopy jsou sice levnější, ale většinou na úkor rychlosti, či kvality měření. A pokud uživatel není majitelem notebooku, není potom toto zařízení ani moc mobilní byť má malé rozměry.

2.1 AD převodník

Pro návrh modulu je důležitou součástí výběr analogově-digitálního převodníku. Můj výběr spočíval v porovnání několika převodníků od různých výrobců, konkrétně jejich parametrů, ceny, dostupnosti a použitelnosti pro pomalejší FPGA. Analogově-digitální převodníky, mezi kterými jsem vybíral, jsou od nejznámějších výrobců z důvodu dostupnosti a kvality. Z cenových důvodů jsem vybíral pouze mezi osmibitovými převodníky, s podobnou cenou řádově desítky dolarů. Pro porovnání hodnot byly vyneseny do *tabulky 1*.

Typ	Výrobce	Rychlost vzorkování	Napájení	Interface	SNR(dB)
AD9481	Analog Devices	250Msps	Single (+3.3V)	Par(2x8)	46
AD9480	Analog Devices	250Msps	Single (+3.3V)	LVDS, Par(1x8)	47
MAX1121	Maxim	250Msps	Single (+1.8V)	LVDS, Par(1x8)	48.8
ADS831	Texas Instruments	80Msps	Single (+5V)	Par(1x8)	49

Tabulka 1: seznam vybraných převodníků

Všechny převodníky mají stejnou pipelineovou architekturu a téměř podobnou hodnotu odstupu signálu od šumu. Liší se způsobem připojení, napájecím napětím a maximální vzorkovací rychlostí. Do návrhu modulu jsem vybral převodník od Analog Devices s označením AD9481. Jako jediný má dvou portový paralelní výstup, tudíž při vyšší vzorkovací frekvenci bude FPGA obvod přijímat poloviční rychlostí než u ostatních převodníků a nemusí se navrhovat navíc žádné další specifické napájení.

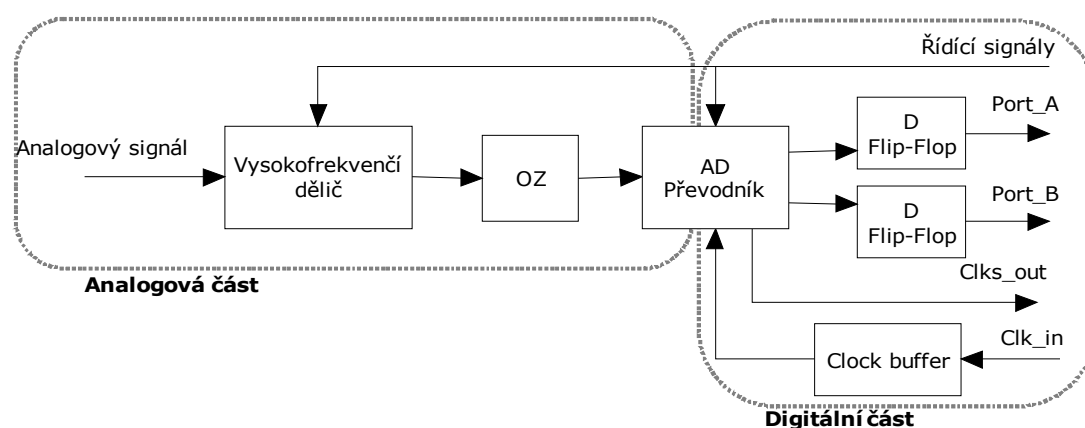
Další součástky byly vybrány podle doporučeného zapojení evaluation boardu s AD9481 převodníkem. Tyto informace se nachází v literatuře [2].

3. Návrh řešení

Celý osciloskop se dá rozdělit do dvou částí. Modul a vývojový kit, které jsou propojeny pomocí konektoru Hirose.

3.1 Modul

Jedná se o periférii, kterou lze připojit k vývojovému kitu. Modul lze opět rozdělit do dvou částí a to na analogovou a digitální část. Na *obrázku 1* je blokové schéma analogové a digitální části celého modulu.



Obrázek 1: blokové schéma modulu

3.1.1 Analogová část modulu

Analogová část má na starosti příjem a úpravu analogového vstupního signálu. Obsahuje vysokofrekvenční dělič, realizovaný pomocí několika relé, přesných rezistorů, kondenzátorů a kapacitních trimrů pro donastavení vstupní kapacity. Další prvek v analogové části je operační zesilovač, který dále upravuje signál z děliče pro AD převodník. Ten má na starosti posun signálu do pracovní úrovně převodníku a zároveň vytváří rozdílový výstup potřebný na vstupu převodníku.

3.1.2 Digitální část

Hlavním prvkem digitální části je analogově-digitální převodník, do kterého přichází upravený snímaný analogový signál z analogové části. Po převodu jsou data

zachytávány klopnými obvody typu D a poté odeslána přes konektor do vývojového kitu. Vstupní hodiny, které řídí rychlost převodu, jsou zde upraveny z jednoho signálu na dva rozdílové pomocí hodinového bufferu. To je důležité pro správnou funkci převodníku.

3.1.3 Návrh schématu a plošného spoje

Návrh schématu byl proveden tak, aby bylo možné použít čtyřvrstvého plošného spoje. Dalšími požadavky byly co nejkratší vedení datových cest pro dosažení co největší možné přenosové rychlosti. Všechny obvody kromě relátek budou v provedení smd. Návrh plošného spoje byl navrhován podle evaluation boardu uvedeném v datasheetu k převodníku.

3.2 Vývojový kit

Jedná se již o hotovou vývojovou desku od Firmy Digilent s FPGA, která obsahuje dost užitečných periférií. Tato deska má za úkol zpracovávat a ukládat data z převodníku. Ty bude následně zobrazovat přes D-sub konektor na monitor s rozlišením 800x600 pixelů. Na monitoru jsou také zobrazeny nastavovací prvky a různé kurzory. Ovládání je realizováno pomocí standardní počítačové myši přes PS/2 konektor.

Vše je realizováno pomocí FPGA, DDR2 SDRAM a Flash paměti. V FPGA je celé řízení prováděno pomocí embedded procesoru microblaze, komunikace s paměťmi bude realizována pomocí IP core dostupných v EDK. Pro příjem dat, zobrazování na monitor, příjem dat z myši a ovládání modulu je vytvořeno uživatelské IP core. To je složeno z jednotlivých částí, které jsem vytvořil v jazyce VHDL.

4. Řešení

V této části budou popsána řešení jednotlivých částí, které jsou v osciloskopu použity. Rozdělení bude zachováno podle předchozí kapitoly na modul a vývojový kit.

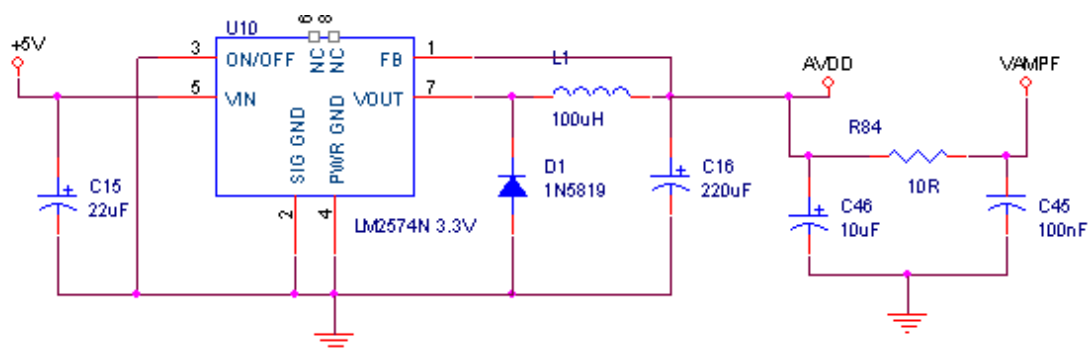
4.1 Modul

4.1.1 Analogová část

Tato část má na starosti úpravu vstupního analogového signálu. Patří sem dělič a operační zesilovač.

4.1.1.1 *Napájení*

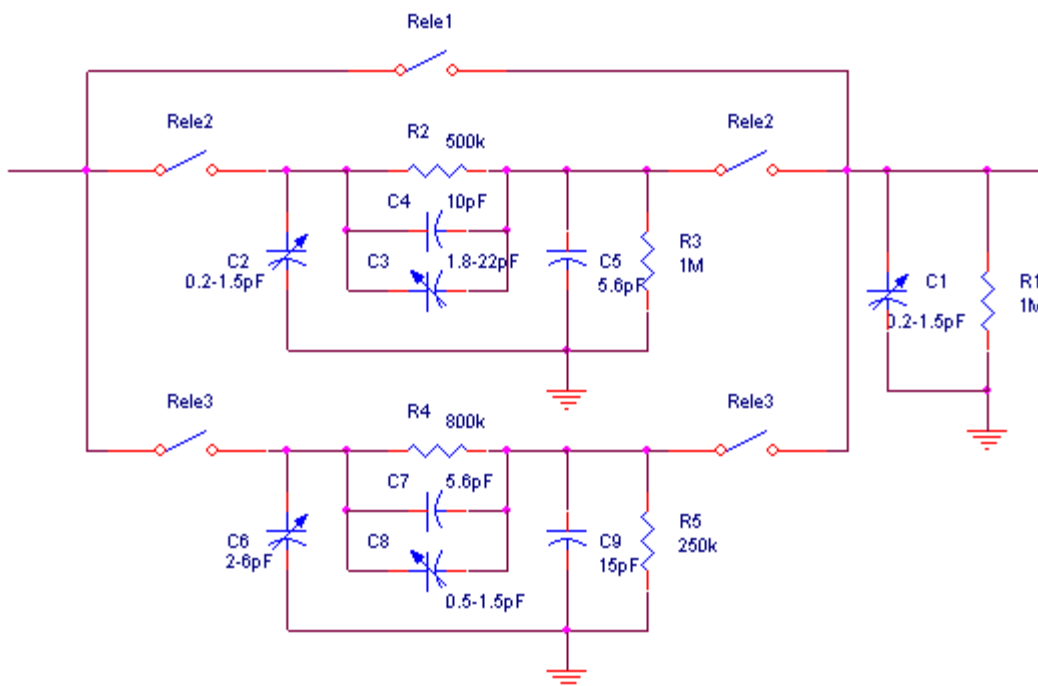
Pro napájení analogové části postačuje +5V, ty jsou přivedeny přímo z vývojového kitu. To je zde použito ke spínání relátek. Pro napájení operačního zesilovače a analogové části AD převodníku je zapotřebí +3.3V. Proto je zde použit spínací stabilizátor LM2574N 3.3V. Ten je zapojen podle doporučeného zapojení od výrobce uvedeného v literatuře[3]. Vstupní napětí je přivedeno přes elektrolitický kondenzátor o velikosti 22uF. Na výstupním pinu je připojena tlumivka o velikosti 100uH. Pokud by vstupní napájení bylo vyšší jak +7V, musela by se použít tlumivka o velikosti 330uH. Dále je zde připojena rychlá schottkyho dioda, která slouží k vybití tlumivky po odpojení vstupního napájení. Na konci je ještě připojen kondenzátor s kapacitou 220uF. Ten slouží k doplňování elektrické energie, než zpětná vazba zaznamená pokles. Zpětná vazba je připojena na výstup za cívku. Schéma zapojení je zobrazeno na *obrázku 2*. Pro napájení operačního zesilovače je napětí vedeno přes filtrační kondenzátor 10uF, rezistor 10Ω a blokovací kondenzátor 100nF. Do každého napájecího pinu AD převodníku je napájení přivedeno přes blokovací kondenzátory o kapacitě 100nF a těm předchází filtrační kondenzátor o kapacitě 10uF.



Obrázek 2: schéma spínaného zdroje

4.1.1.2 Vysokofrekvenční dělič

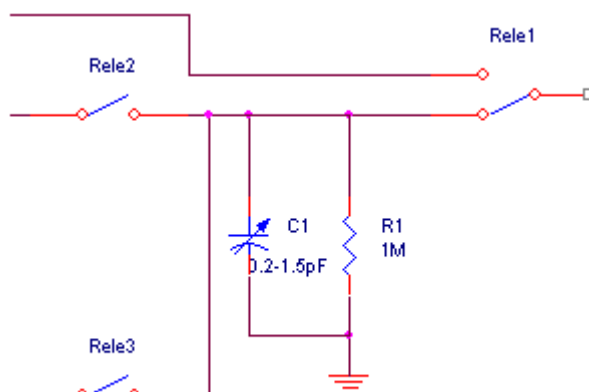
V poloze 1:1, realizováno sepnutím Relé 1, slouží C1 k nastavení přesné jmenovité vstupní kapacity. V poloze 2:1, sepnuté Relé 2, se k trvale zapojenému rezistoru R1 zapojí paralelně rezistor R3, takže na výstupní straně vznikne 500k Ω . Kvůli tomuto důsledku musí mít sériový rezistor R2 také hodnotu 500k Ω . K nastavení kapacity by stačil kondenzátor C3, ale jelikož nastavení děliče musí být provedeno s přesností větší než 1% a musí zůstat stabilní. Proto nesmí mít C3 příliš velký rozsah nastavení, zapojí se tedy paralelně malý trimr C3 s pevným kondenzátorem C4. C5 vědomě zvětšuje z výstupní strany zátěžovou kapacitu děliče, proto aby byla celkové sériová kapacita větší a lépe nastavitelná. Vstupní kapacita děliče je ve všech polohách pod 1:1 menší, neboť sériová(podélná) kapacita se vzrůstajícím dělicím poměrem klesá. K nastavení slouží tedy C2, pomocí něj se i pro poměr 2:1 nastaví jmenovitá hodnota 15pF na vstupu přístroje. Všechny následující další polohy děliče jsou navrženy stejně. Pro poměr 5:1 a vstupní odpor 1M Ω potřebujeme 800k Ω + 200k Ω . Jelikož R1 = 1M Ω , je R5 = 250k Ω , jejich paralelní zapojení dá potřebných 200k Ω . Pro nastavení podélné kapacity je k C7 paralelně připojen nejmenší dostupný trimr 0.2 ... 1.5 pF, pomocí něj je možno vyrovnat výrobní tolerance C8, C9. Vstupní kapacita se pomocí C6 nenastavuje, neboť trimr pro sériovou kapacitu již dále nelze zmenšovat, zvyšují se nutně s rostoucím dělicím poměrem přidavné kondenzátory na výstupní straně(C5, C9). Tím se ve spojení s indukčnostmi přívodů tvoří parazitní kmitavé obvody, které u širokopásmových přístrojů vedou ke zkreslení impulsů. Z tohoto důvodu se u širokopásmových oscilokopů usiluje o minimalizaci vstupní kapacity. Tyto informace byly čerpány z literatury [1].



Obrázek 3: schéma jednostupňového vysokofrekvenčního děliče

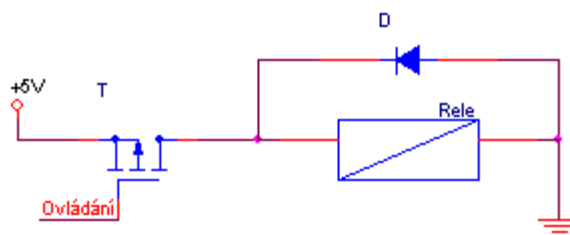
Dělicí poměr lze zvýšit řazením těchto děličů do řetězce. Na vývojovém modulu jsou tyto děliče dva, kde druhý lze pomocí nulového rezistoru přemostit a vyřadit tak z provozu v případě, že by se navzájem špatně ovlivňovaly a signál by byl moc zkreslen. Aby si tyto děliče neovlivňovali pevný odpor na výstupní straně, tak je spínací Relé 1 na druhém děliči nahrazeno relém s přepínáním dvou vstupů. S tím že jeden vstup je připojen na vstup děliče a druhý na výstup za rezistor R1. Na obrázku 4 je zobrazena tato část, ostatní relátka jsou připojena jako na prvním děliči. Vstup prvního děliče je připojen na konektor BNC pro připojení sondy.

Všechny relátka jsou realizována pomocí jazýčkových relátek.



Obrázek 4: úprava druhého děliče

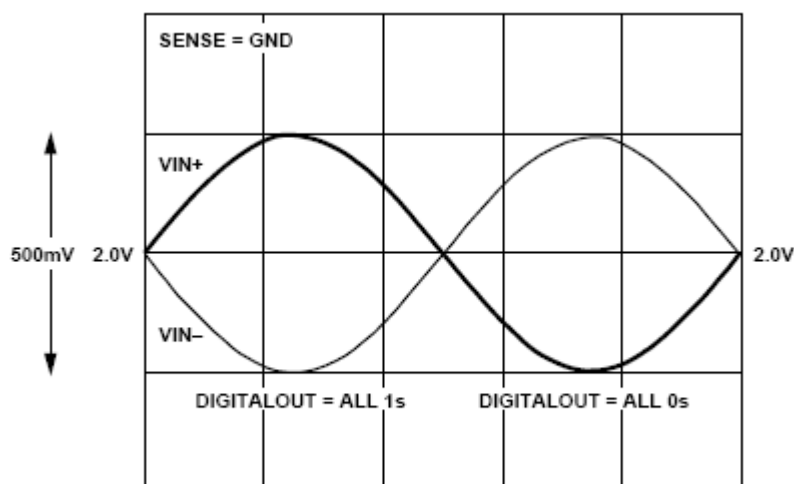
Spínání relátek je realizováno pomocí tranzistorů MOSFET s P kanály. Zapojení je jednoduché, na drain(D) je připojena jedna část cívky relé, druhá část je připojena na zem. K source(S) je připojeno napětí +5V a na poslední vývod tranzistoru gate(G) je připojen ovládací signál z FPGA. Pokud je na ovládacím signálu log „1“ je tranzistor rozepnut, sepne jeli přivedena log „0“. Kvůli vybíjení cívky relé je ke každému relátku připojena paralelně dioda anodou k zemi, jak je zobrazeno na *obrázku 5*. Takto jsou ovládána všechna relé v návrhu.



Obrázek 5: spínání relé

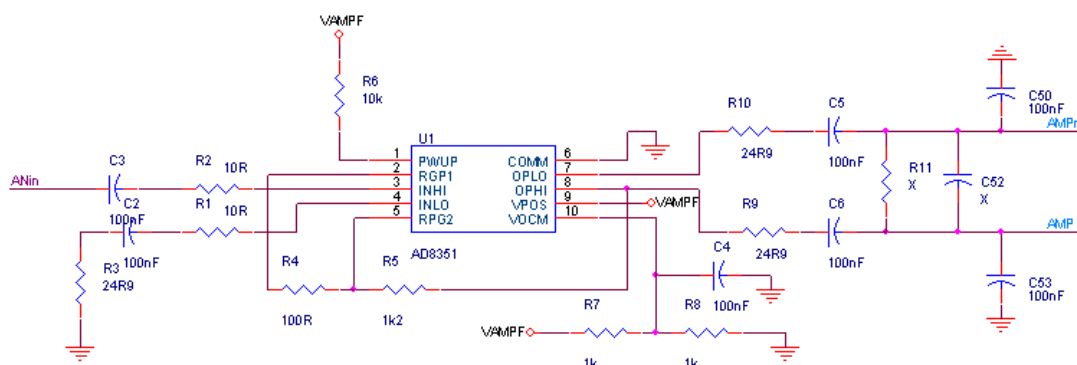
4.1.1.3 Operační zesilovač

Operační zesilovač AD8351 má za úkol upravený signál pomocí děliče, dále upravit pro AD převodník a zároveň slouží jako oddělovací člen. Úprava spočívá v tom, že vstupní signál o velikosti maximálně 1Vp-p se středem na zemi, posune o určité napětí a střed na výstupu nebude na zemi, ale na 2V a zároveň vytváří rozdílový výstupní signál, který potřebuje AD převodník k rychlému převodu. Příklad výstupního signálu je zobrazen na *obrázku 6*.



Obrázek 6: vstupní signál pro AD převodník

Schéma bylo vytvořeno podle literatury[2], kde je doporučené zapojení vyobrazeno i s návrhem plošného spoje pro evaluation board. *Obrázek 7* ukazuje schéma použitého při návrhu modulu. Signál z děliče je přiveden přes kondenzátor 100nF a rezistor 10Ω na rozdílový kladný vstup INHI. Druhý vstup INLO je uzemněn přes rezistor 10Ω, kondenzátor 100nF a další rezistor 25Ω. Výstupy OPHI a OPLO jsou do AD převodníku přivedeny přes 25Ω rezistor a 100nF kondenzátory, jeden je v sérii s rezistorem a druhý přiveden na zem. Rezistor a kondenzátor s hodnotou X nejsou osazeny, ale dle potřeby úpravy signálu je možné je osadit. Na pin PWUP je přes 10kΩ rezistor přivedeno napájení, tím je zajištěno, že je OZ aktivní. Výstupní signál je z OPHI zpětnou vazbou přiveden přes rezistor 1k2 na pin RPG2 (Gain Resistor Input) a dále přes 100Ω rezistor na RGP1. COMM je připojen na zem a VPOS na napájení. Na VOVM je přivedeno napětí z děliče s blokovacím kondenzátorem, které udává posun vstupního signálu.



Obrázek 7: schéma zapojení OZ

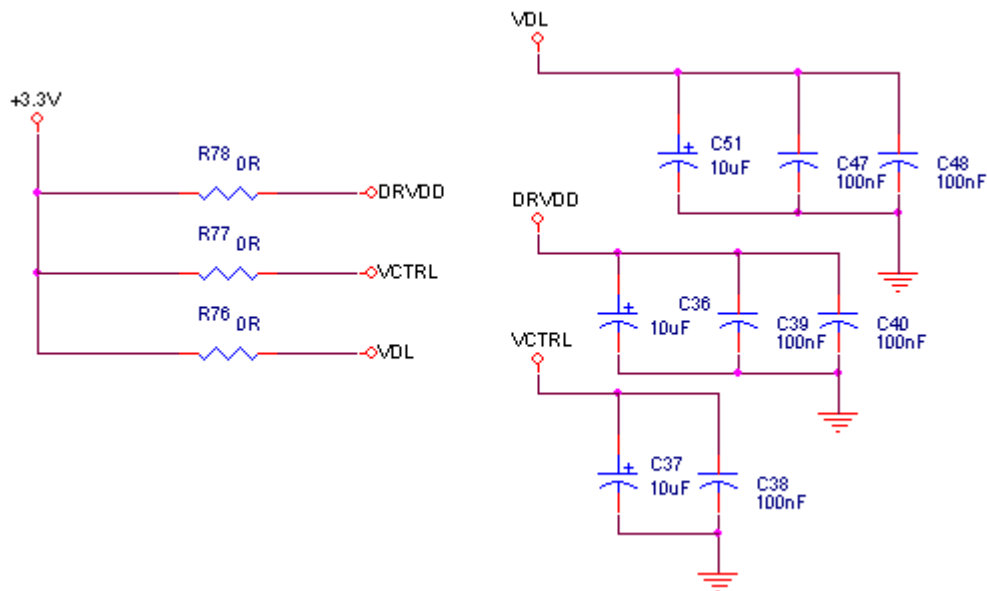
4.1.2 Digitální část

Digitální část má na starosti převod analogového signálu na digitální a jeho distribuci do FPGA. Nachází se zde AD převodník, clock buffer a klopné obvody typu D reagující na náběžnou hranu.

4.1.2.1 Napájení

Pro napájení celé digitální části postačuje napětí +3.3V, které je přivedeno přímo z vývojového kitu. To je přes nulové rezistory dále rozděleno na DRVDD, VDL a VCTRL. Rezistory jsou zde pro ožívování při připojování součástek. Za

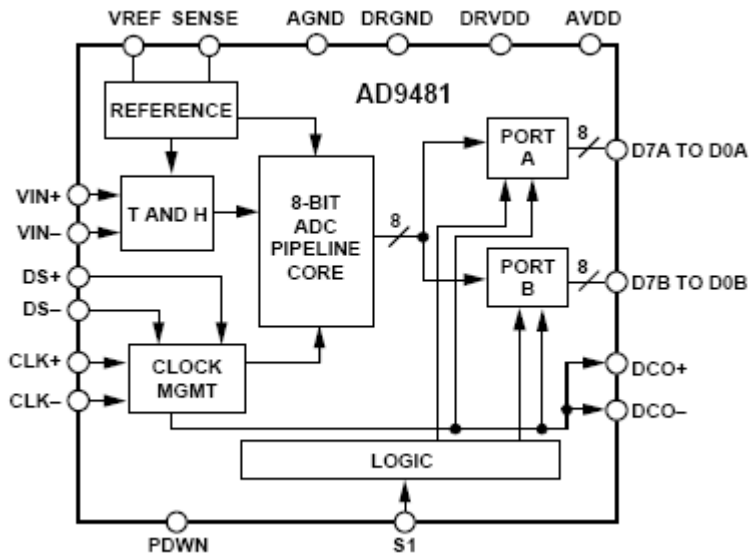
rezistory jsou připojeny filtrační kondenzátory s kapacitou 10 μ F a na piny součástek jsou dále přivedeny přes blokovací kondenzátory s kapacitou 100nF. Jak je vyobrazeno na *obrázku 8*, DRVDD je napájení pro AD převodník, přes VDL jsou napájeny klopné obvody typu D a napětí VCTRL slouží pro obvod upravující vstupní hodiny pro AD převodník.



Obrázek 8: napájení digitální části s filtračními a blokovacími kondenzátory

4.1.2.2 AD převodník

Hlavním obvodem modulu je AD převodník AD9481 od firmy Analog Devices. Jedná se o pipelineový převodník, který obsahuje interní referenční napětí, track-and-hold obvod. Výstup je realizován pomocí dvou digitálních portů kompatibilních s TTL/CMOS. Na *obrázku 9* je zobrazeno blokové schéma, kde jsou jednotlivé části vidět. Všechny informace čerpány z literatury[2].



Obrázek 9: blokové schéma obvodu AD9481

Popis vstupů a výstupů:

- VIN+ a VIN- vstup pro diferenciální analogový signál
- DS+ a DS- řídicí diferenciální vstup, slouží pro spouštění převodu, využívá se například pro synchronizaci více AD převodníků, aby na výstupních portech byly hodnoty vysílány s určitým posuvem
- Clk+ a Clk- vstupní diferenciální hodiny
- PDWN řídicí vstup, který připojením tohoto signálu na log „1“ se přivede převodník do low power stavu
- S1 řídicí vstup, podle napěťové úrovně na tomto pinu se nastaví kódování výstupu a zapne nebo vypne duty cycle stabilizátor. V *tabulce 2* jsou zobrazeny napěťové úrovně a jejich vliv na výstup a stabilizátor.

Úroveň napětí na S1	Formát dat	Duty cycle stabilizátor
$(0.9 \times AVDD)$ až $AVDD$	Offset binary	Vypnut
$(\frac{2}{3} \times AVDD) \pm (\frac{AVDD}{10})$	Offset binary	Zapnut
$(\frac{1}{3} \times AVDD) \pm (0.1 \times AVDD)$	Twos complement	Zapnut
AGND až $(0.1 \times AVDD)$	Twos complement	Vypnut

Tabulka 2: napěťových úrovní na S1

- DCO+ a DCO- výstupní diferenciální hodiny

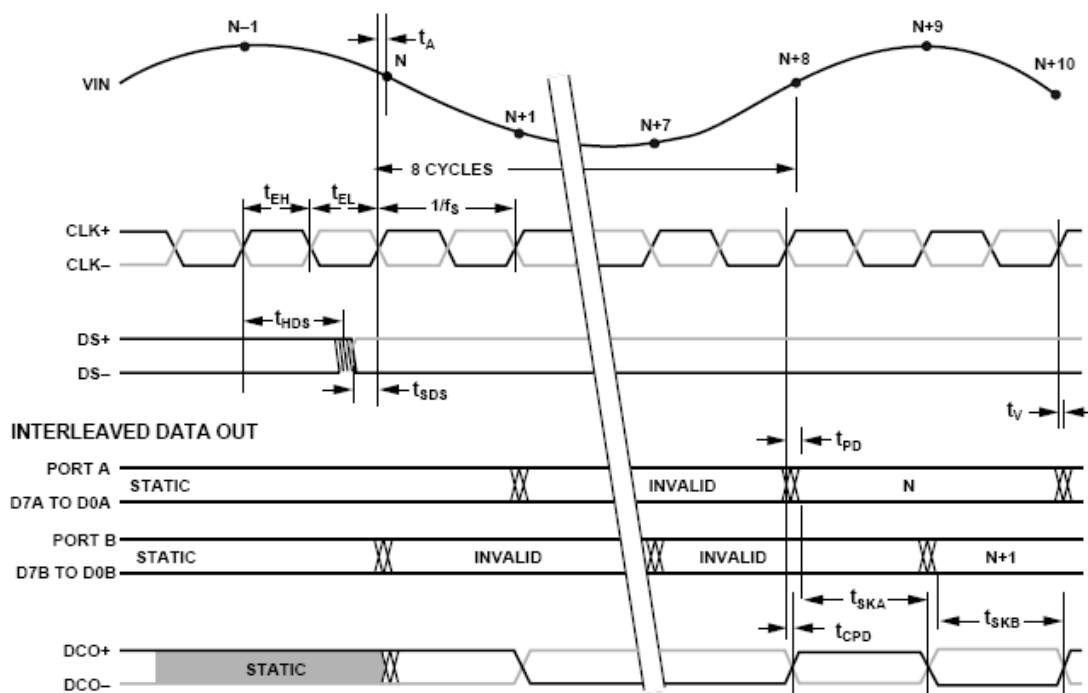
- S3 řídicí vstup, pokud je na něj přivedena log. „1“ jsou aktivovány výstupní hodiny, pokud je přivedena log. „0“ výstupní hodiny stojí
- DA a DB výstupní datové porty, v *tabulce 3* jsou zobrazeny výstupy podle nastavení kódování v závislosti na vstupním signálu

Kód	(VIN+) - (VIN-)	Offset binary	Twos complement
255	> +0.512V	1111 1111	0111 1111
255	+0.512V	1111 1111	0111 1111
255	+0.508V	1111 1110	0111 1110
...
129	+0.04V	1000 0001	0000 0001
128	0.0V	1000 0000	0000 0000
127	-0.04V	0111 1111	1111 1111
...
2	-0.504V	0000 0010	1000 0010
1	-0.508V	0000 0001	1000 0001
0	-0.512V	0000 0000	1000 0000
0	< -0.512V	0000 0000	1000 0000

Tabulka 3: výstupní hodnoty podle vstupního signálu

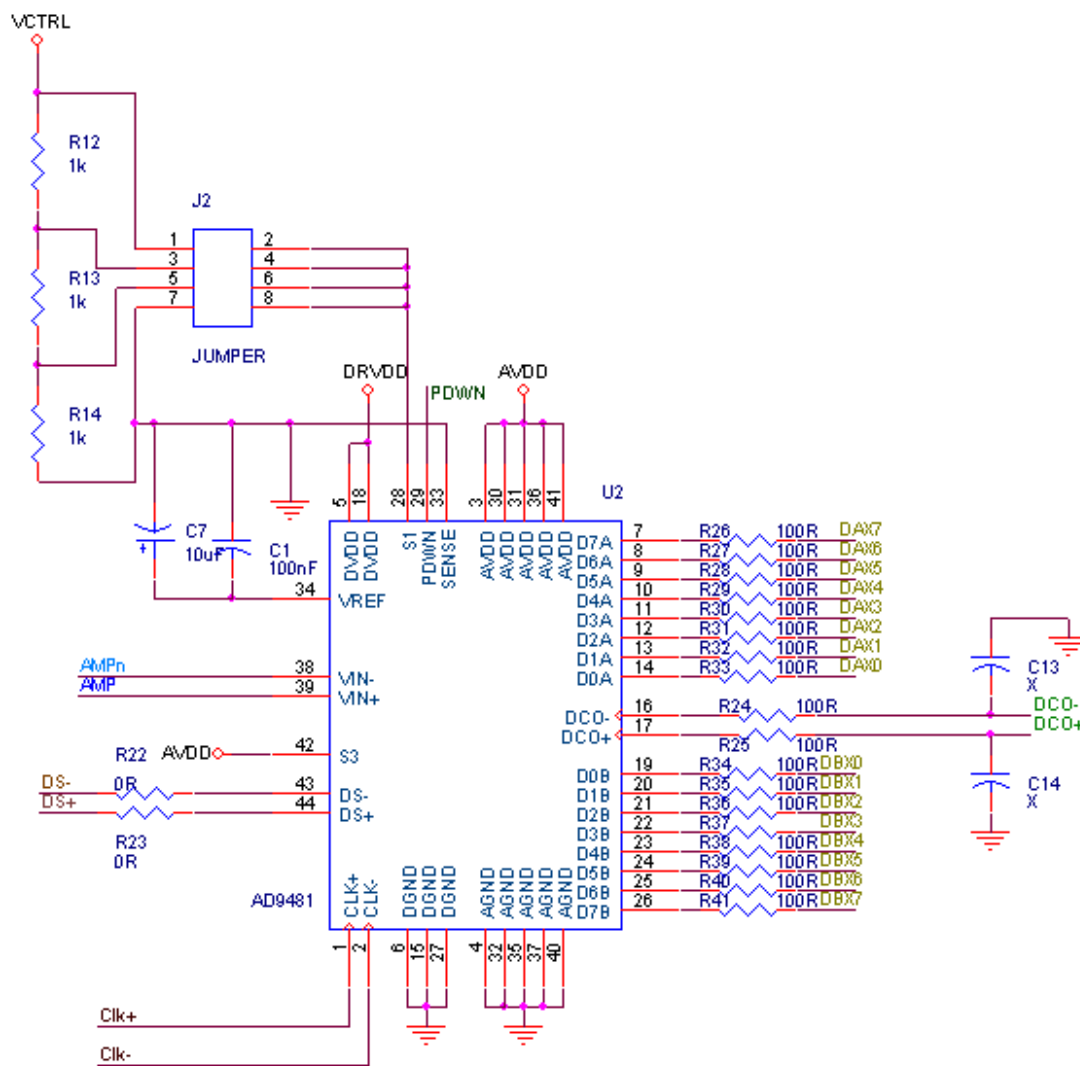
- AVDD analogové napájení
- DRVDD digitální napájení
- AGND analogová zem
- DRGND digitální zem
- SENSE řídicí vstup, pokud je připojen na AVDD reference je řízena externě, pokud na zem, tak je použito interní referenční napětí, pokud je jiná hodnota dá se naprogramovat jiné referenční napětí v závislosti na VREF
- VREF vstup pro externí referenci, pokud není využita, jsou na tento pin připojeny kondenzátory s kapacitami 10uF a 100nF vůči zemi

Na *obrázku 10* je vidět časový průběh převodu dat. Převod je započat aktivováním DS+ a DS-, následující takt se načte do převodníku první hodnota. Ta je ale na výstup distribuována až po osmi taktech, což je hloubka pipeline. Data jsou zapsána na PORTA souběžně s náběžnou hranou DCO+, na PORTB souběžně s náběžnou hranou DCO-.



Obrázek 10: časový průběh převodu převsato z [2]

Použité zapojení pinů je zobrazeno na obrázku 11. SENSE je připojeno na zem, aby byla použita interní reference, proto jsou na VREF připojeny kondenzátory 10uF a 100nF. PDWN je připojeno přímo k FPGA, k S1 je připojen konektor pomocí kterého se jumprem dá nastavit úroveň napětí odpovídající údajům v tabulce 3. Na VIN+ a VIN- je přiveden upravený signál z OZ. S3 je připojen na napájení proto, aby výstupní hodiny byly aktivní. DS+ a DS- jsou přes nulové rezistory přivedeny na FPGA. Rezistory jsou zde z důvodu testování, neboť tento diferenciální vstup je defaultně nastaven jako aktivní. Clk+ a Clk- jsou připojeny na hodinový buffer. Porty DA a DB jsou přes 100Ω připojeny na vstupy klopných obvodů typu D, rezistory jsou zde z důvodu zabránění odrazů. DCO+ a DCO- jsou připojeny přímo na vstupní piny hodin FPGA a zároveň na hodiny klopných obvodů. AVDD je připojeno na analogové napájení a DVDD je připojeno na digitální napájení.

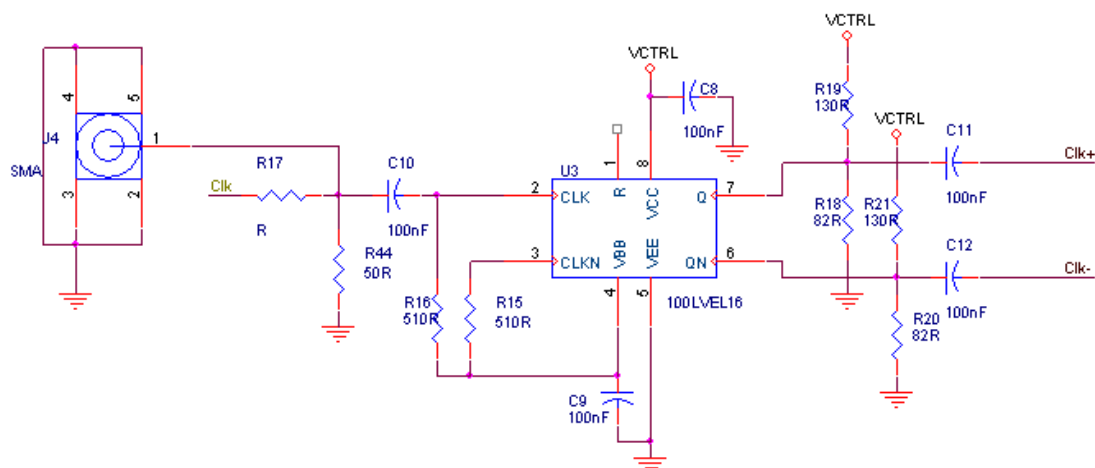


Obrázek 11: schéma zapojení AD převodníku

4.1.2.3 Clock buffer

Pro převod vstupních single hodin z FPGA na diferenciální hodiny, které jsou potřeba pro správný chod převodníku, je použit obvod MC100-LVEL16D. Doporučené zapojení bylo čerpáno z literatury[2]. Vstup hodin může být přiveden buď přímo z FPGA, nebo z konektoru SMA. Pokud chceme použít hodinový signál z FPGA stačí osadit nulový rezistor a neosazovat 50Ω rezistor. Ten je zde pokud bychom chtěli použít vstup z SMA konektoru, má totiž za úkol impedančně přizpůsobit vstup, na který jsou připojovány koaxiální kabely. Hodiny jsou na vstup obvodu přivedeny přes kondenzátor s kapacitou 100nF, ten propouští pouze střídavý signál. Na VCC je přivedeno napájení přes blokovací kondenzátor 100nF. VEE je přivedeno na zem. Na výstupní referenční napětí je vůči zemi připojen kondenzátor

100nF. Reference je dále vedena přes rezistor 510Ω na vstup hodin Clk a ClkN. Výstupní hodiny Q a QN jsou připojeny pře rezistory 130Ω na napájení a přes další rezistory 82Ω na zem, to je pro rychlé přepínání změny hodin, neboť převodník detekuje změnu hodin již při rozdílu 200mV mezi vodiči Clk+ a Clk-. Výstupní hodiny Q a QN jsou do převodníku dále vedeny přes kondenzátory s kapacitou 100nF. Použité schéma je vyobrazeno na *obrázku 12*.

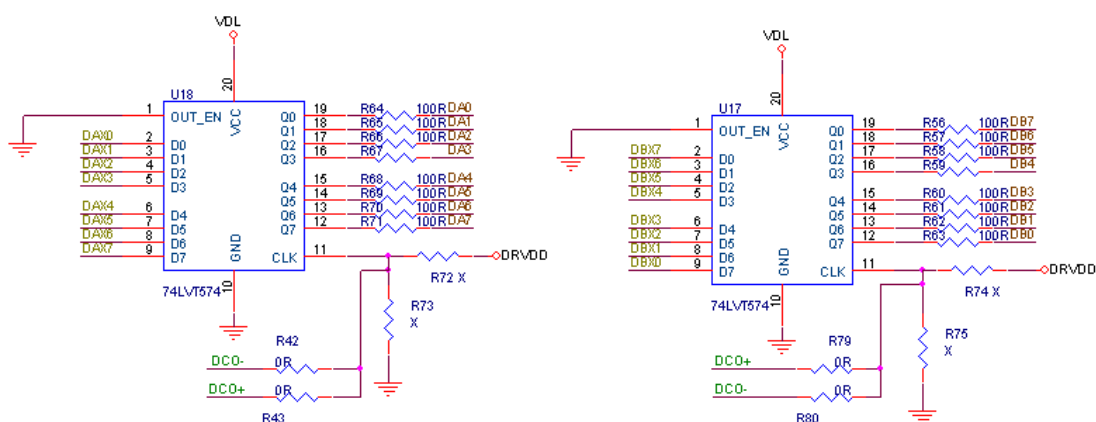


Obrázek 12: schéma zapojení obvodu pro převod hodin

4.1.2.4 Klopné obvody

Pro posílení digitálního signálu z AD převodníku byly použity dva obvody 74LVT574. Jedná se o klopné obvody typu D reagující na náběžnou hranu hodin. Maximální vstupní frekvence těchto obvodů je 150MHz což postačuje, neboť i při maximální vzorkovací frekvenci půjdou z převodníku hodiny s maximální frekvencí 125MHz. Na *obrázku 13* je zobrazeno schéma zapojení. Vstup řídicího signálu OEnon je přiveden na zem, proto aby se data neustále s každou náběžnou hranou hodin propagovala na výstup. Vstupy D jsou připojeny na rezistory, které jsou připojeny na výstupní porty AD převodníku. Hodiny obvodu připojeného na výstup portu A jsou připojeny na DCO- a druhý obvod má hodiny připojeny na DCO+. Takto je zaručeno, že načtená data z převodníku jsou již ustálená na správné hodnotě a jsou konstantní po potřebnou dobu načítání dat klopných obvodů. Jedná se o časy t_s (setup time) a t_h (hold time). V návrhu je zahrnuta možnost i opačného připojení hodin, stačí dle potřeby osadit nulové odpory. Data jsou na výstup propagována s určitým zpožděním, proto se data z portu A načítají při náběžné hraně DCO+ a z portu B při náběžné hraně DCO-. Takže se data dostanou do FPGA se zpožděním 9

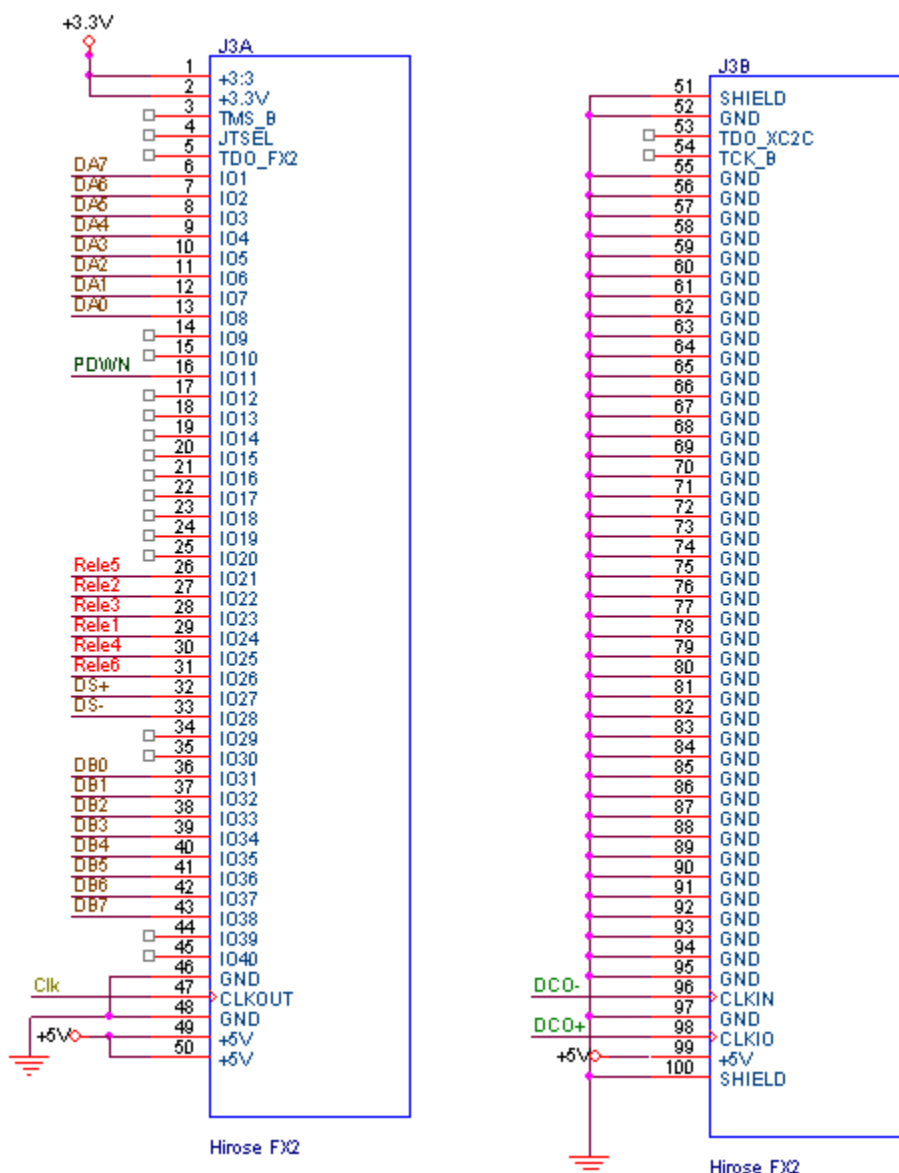
cyklů, z čehož 8 cyklů trvá převod. Obvody jsou napájeny 3.3V. Výstupy Q jsou přes 100Ω rezistory připojeny přímo na FPGA, jsou zde z důvodu minimalizace odrazů na vodičích plošného spoje.



Obrázek 13: schéma zapojení kolpných obvodů

4.1.2.5 Konektor

Na obou vývojových kitech je použit 100 pinový konektor Hirose, aby byla dosažena kompatibilita mezi deskami musí se dát pozor, které piny použít, neboť na desce Spartan-3E starter kit jsou některé IO piny sdílené s jinými konektory či LEDkami, nebo některé piny můžou sloužit jen jako vstupní. Na obrázku 14 je nakresleno schéma zapojení konektoru na modulu. Označení použitého konektoru je HIROSE FX2-100S-1.27DS. Polovina pinů je připojena na zem, na krajích je připojeno napětí +5V a +3.3V u desky se Spartanem-3E se musí nastavit jumper tak, aby výstupní napětí bylo opravdu +3.3V. Data z portu A jsou přivedeny na IO1 až IO8, data z portu B jsou přivedeny na IO31 až IO38. PDWN je přiveden na pin IO11. Ovládání spínání relátek je na pinech IO21 až IO26. Na pinech IO27 a IO28 jsou připojeny řídicí signály DS+ a DS-. Piny IO13 až IO20 jsou schválně neosazeny, neboť jsou na desce se Spartanem-3E sdílené s LEDkami. Vstupní hodiny pro modul jsou přivedeny z pinu CLKOUT. Výstupní hodiny DCO+ jsou přivedeny na pin CLKIO a DCO- na pin CLKIN. Zbylé piny jsou nevyužity.



Obrázek 14: schéma připojení konektoru Hirose

4.2 Vývojový kit

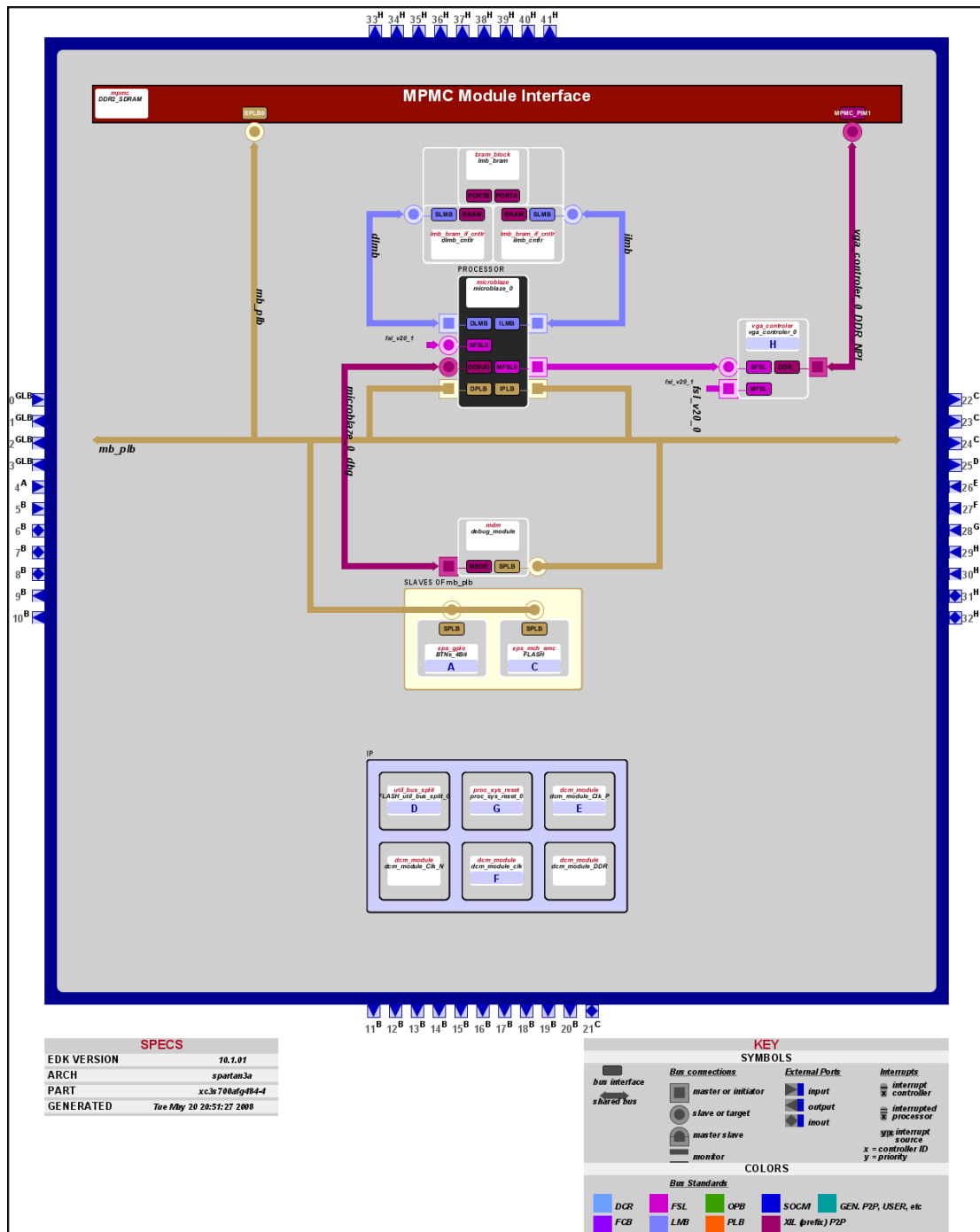
Vývojový kit obsahuje spoustu různých periférií a konektorů pro různá zařízení. Veškeré informace o vývojovém kitu čerpány z literatury [6]. Pro osciloskop jsou využity konektory: PS/2 pro počítačovou myš, D-sub pro připojení monitoru a hirose pro připojení modulu. Z periférií je využito: DDR2 SDRAM paměť pro ukládání zobrazované obrazovky, paralel NOR flash PROM pro uložení bitmap zobrazovaných prvků a Xilinx Platform flash pro nahrání bitstreamu do FPGA. Pro vytvoření návrhu do FPGA bylo použito vývojového prostředí EDK od firmy Xilinx. V tomto prostředí je k dispozici embedded procesor microblaze a spousta IP core pro komunikaci s perifériemi. Na obrázku 15 je vidět blokové schéma použitých jader a

jednotlivé sběrnice, kterými jsou propojeny jádra s procesorem. Pro komunikaci s modulem, zobrazování dat a komunikace s myší bylo vytvořeno uživatelské jádro s názvem `VGA_controller`. To je propojeno s procesorem pomocí sběrnice FSL(Fast Simplex Link), s jádrem MPMC(Multi-Port Memory Controller) je propojen pomocí NPI(Native Port Interface) sběrnice. MPMC jádro slouží pro rychlou komunikaci s DDR2 SDRAM pamětí, výhodou tohoto jádra je více portová komunikace s možností výběru druhu sběrnice. Je tedy možno k němu připojit další jádra nebo procesory. Pro komunikaci s flash pamětí slouží `XPS_MCH_EMC`(XPS Multi-Channel External Memory Controller) jádro. Další jádro, které je na *obrázku15*, vidět je MDM(Microblaze Debug module) to slouží k odlaďování návrhu, či programu. Tyto jádra jsou k procesoru připojeny pomocí sběrnice PLB(Processor Local Bus). Aby procesor mohl zpracovávat data a instrukce je k němu připojena paměť BRAM(Block Random Access Memory), ta je připojena přes LMB(Local Memory Bus) sběrnici, ta se dělí na instrukční a datovou. Další jádra v návrhu jsou: Utility Bus Split to užívá EMC ke komunikaci s flash pamětí, další jádro je Processor systém Reset Module. To slouží k generování resetu pro sběrnice, procesor, periferie, je řízen externím resetem a zároveň lock signálem z DCM(Digital Clock Manager) jader, ty jsou použity 4 mají za úkol upravovat vstupní hodiny a generovat vnitřní hodiny s konkrétní potřebnou frekvencí. Pro DDR2 paměť jsou generovány dvoje hodiny o frekvenci 100MHz, které jsou vůči sobě fázově posunutý o 90 stupňů. Další generované hodiny jsou o frekvenci 40MHz, ty jsou potřebný ke generování VGA signálů, externí hodiny 50MHz jsou použity pro sběrnice, procesor a další jádra. Hodiny přivedené z přípojného modulu jsou připojeny také k DCM pro možnost úpravy.

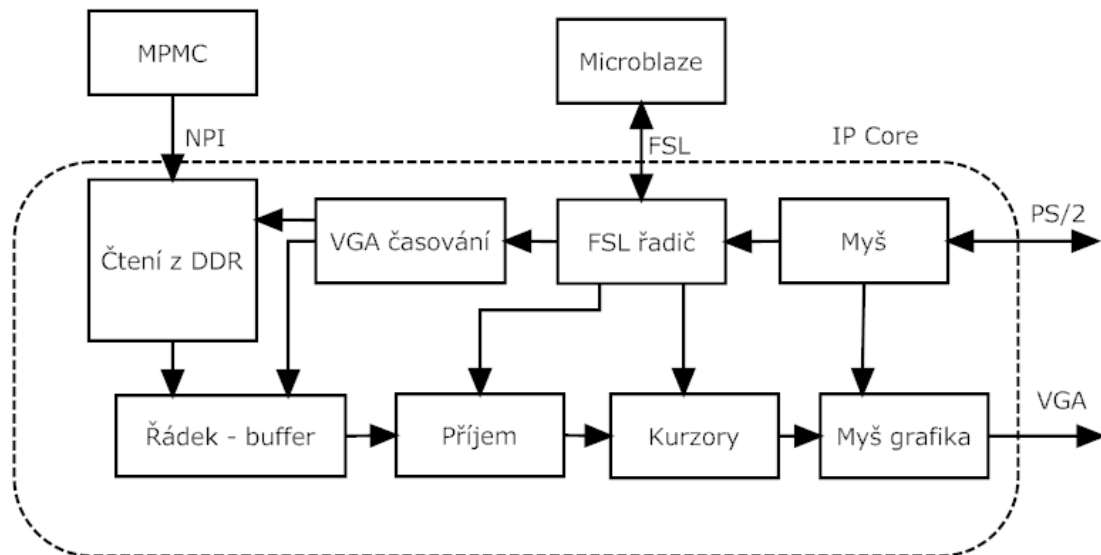
4.2.1 IP core `VGA_controller`

Jedná se o vytvořené jádro s možností připojení k procesru a MPMC jádru. Na *obrázku 16* je znázorněno blokové schéma struktury tohoto jádra a rozdělení na podbloky. Blok s názvem čtení z DDR načítá data k zobrazení přes sběrnici NPI a ukládá je do bufferu řádku, což je paměť BRAM. Ty jsou dále vyčítány pomocí VGA časování, které generuje všechny potřebné signály pro zobrazování na monitor. Z bufferu řádku jsou data dále distribuována přes bloky příjem, kurzory a myš grafika. Příjem zpracovává data z externího modulu, které následně zobrazuje na monitor, podle nastavení a zobrazovacích signálů. Blok kurzory zobrazuje v měřícím poli

kurzory pro měření nebo nastavení hodnot triggeru. Blok myš má na starosti komunikaci s myší. Získaná a upravená data posílá do FSL řadiče a bloku myš grafika. Myš grafika slouží k zobrazování kurzoru myši. FSL řadič přijímá řídicí signály pro jednotlivé bloky, nebo externí modul, pokud je zmáčknuto tlačítko posílá procesoru souřadnice kurzoru.



Obrázek 15: blokové schéma propojení jader s procesorem



Obrázek 16: blokové schéma VGA_controlleru

4.2.1.1 Blok Čtení z DDR

Blok čtení z DDR komunikuje s jádrem MPMC přes sběrnici NPI, z ní jsou využity pouze signály pro čtení, zapisovací výstupní signály jsou nepřipojeny a vstupní signály jsou připojeny na log. „0“. Data jsou dále připravena pro zápis do bufferu řádku. Všechny informace o této sběrnici jsou čerpány z literatury[4]. Princip čtení dat přes NPI sběrnici je následující, nastaví se adresa a další potřebné řídicí signály. Pošle se žádost o příjem adresy, v ten samý takt hodin může být přijmuto potvrzení a začne se vyřizovat žádost. Vyžádaná data jsou k dispozici je-li shozen signál empty, vyčítání dat začne nastavením signálu pop. Data mohou být přijmuta s různou latencí, podle nastavení řídicích signálů se posílá různá délka a počet dat.

Popis použitých signálů pro čtení:

Addr – startovní adresa konkrétní žádosti

AddrReq – nastavením toho signálů se posílá žádost o zpracování adresy
včetně nastavení dalších řídicích signálů

RNW – pokud je tento signál nastaven na log. „0“ jedná se o žádost zápisu,
pokud je nastaven na log. „1“ jedná se o žádost čtení

Size – indikuje typ přenosu při žádosti

- 0x0 = přenos jednoho slova (pouze při 32 bitovém NPI)
- 0x0 = přenos dvou slov (pouze při 64 bitovém NPI)
- 0x1 = přenos 4 slov řádku cache

- 0x2 = přenos 8 slov řádku cache
- 0x3 = přenos 16 slov v pořadí za sebou (burst)
- 0x4 = přenos 32 slov v pořadí za sebou (burst)
- 0x5 = přenos 64 slov v pořadí za sebou (burst) (pouze při 64 bitovém NPI a použití SRL FIFO paměti)

InitDone – log. „1“ indikuje, že inicializace je kompletní paměti FIFO se mohou použít

AddrAck – potvrzení přijetí žádosti o zpracování adresy a řídicích signálů, tento signál je aktivní v jednom taktu

RdFIFO_Pop – nastavením signálu se žádá o vyčtení dat z FIFO, nesmí být nastaven jeli InitDone = 0 a pokud RdFIFO_Empty = 1, tento signál musí být shozen jeli čtecí FIFO vyprázdněna

RdFIFO_Flush – resetuje čtecí FIFO paměť, nesmí být nastaven dokud není signál RdFIFO_Empty = 1

RdFIFO_Data – výstup požadovaných dat

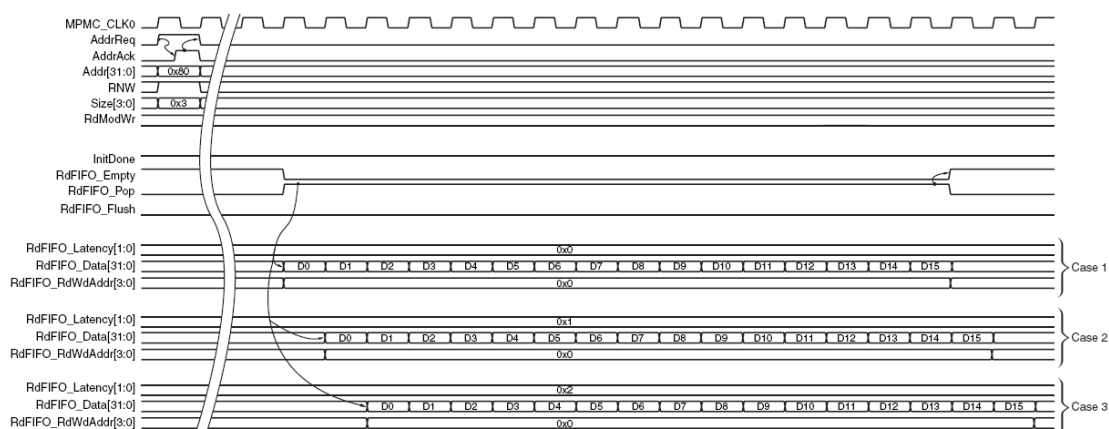
RdFIFO_RdWAddr – udává adresy dat při posílání řádku z cache, nultá adresa indikuje data pro které byla podána žádost

RdFIFO_Empty – pokud je tento signál nastaven, znamená to že nejsou k dispozici žádná data, jeli shozen indikuje že jsou data připravena ve FIFO paměti

RdFIFO_Latency – může nabývat hodnot 0, 1, 2

- 0 = data z FIFO jsou platná v tom samém taktu
- 1 = data z FIFO jsou platná v následujícím taktu
- 2 = data z FIFO jsou platná po dvou taktech

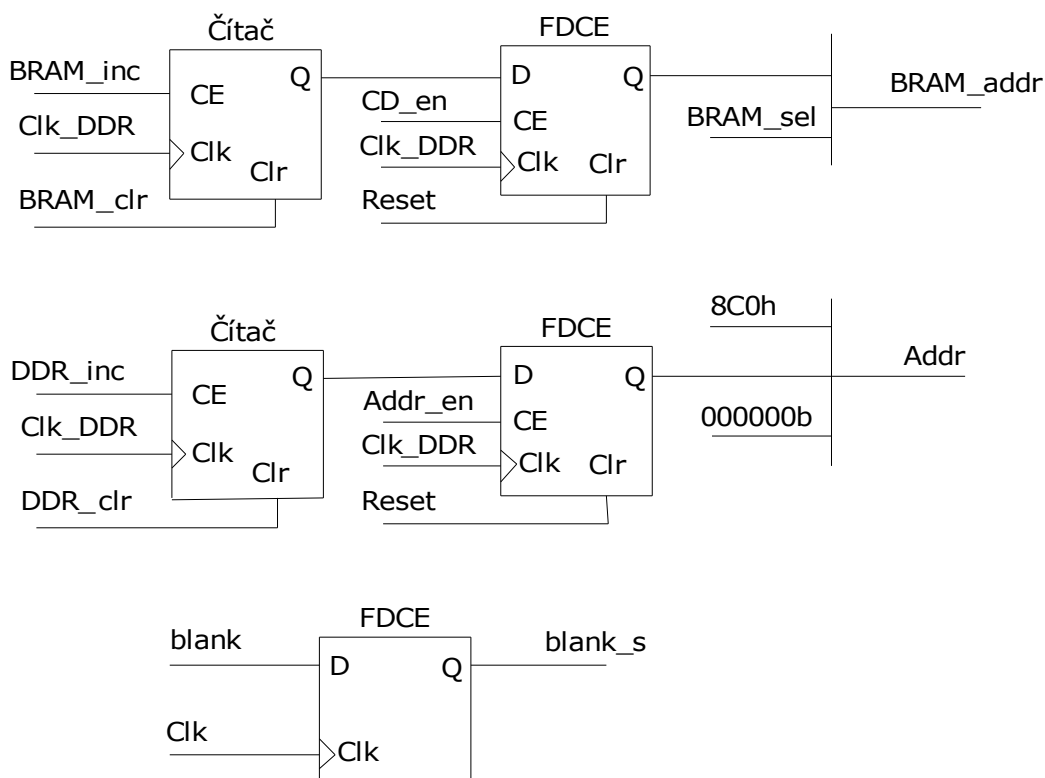
V této aplikaci je použito čtení šestnácti slov, neboť se musí do bufferu řádku načíst vždy celý řádek, což je 800 pixelů po 16 bitech. Ale NPI sběrnice podporuje nejmenší šířku dat 32 bitů, takže se načtou vždy dva pixely. Z tohoto důvodu se čte po šestnácti 32 bitových slovech, neboť adresa musí být dělitelná daným počtem slov beze zbytku, jinak budou další data načtena ve špatném pořadí. Na *obrázku 17* je vidět časový průběh použité metody čtení dat.



Obrázek 17: časový průběh čtení 16 slov z NPI sběrnice převzato a upraveno z [4]

Čtení dat je realizováno 16 stavovým řídicím automatem, který je vyobrazen na obrázku 18. Datová část je zobrazena na obrázku 19. Jelikož čtení dat musí být stejně rychlé jako frekvence samotných DDR, je vše navrženo pro takt hodin 100MHz. Automat ovládá řídicí signály NPI sběrnice pro čtení, řídicí signály pro zápis do BRAM a zároveň ovládá datovou část. Princip automatu je následující, po resetu se dostane do stavu kdy se čeká dokud není InitDone = 1, poté se žádá o zpracování adresy a řídicích signálů, které jsou nastaveny pro čtení 16 slov. Po potvrzení žádosti se čeká dokud nebudou data připravena v čtecí FIFO paměti. Jsou-li připravena započne čtení, během toho se zjistí s jakou latencí jsou data vysílána a inkrementují se čítače adresy pro BRAM a DDR. Podle stavu latence se přejde do dalšího stavu, až je nakonec načteno první slovo a přejde se do stavů, kdy se již latence neuvažuje. V dalších osmi stavech se načítají další slova a zapisují se do BRAM, kde horní část adresy je tvořena čítačem a spodní 3 bity jsou generovány automatem. Pokud je přijato a zapsáno 16 slov přejde se do stavu Konec Dat, kde se zjistí zda byl načten celý řádek. V tomto stavu se načte nová adresa, která je tvořena z dvanácti bitů horní adresy pomocí které přistupuje procesor do paměti. V mém případě se jedná o adresu 8C0h, další část je tvořena čítačem a posledních 6 bitů je nulových, neboť adresa ukazuje na jednotlivé byty. Úprava adresy je obsažena v datové části. Pokud je načten celý řádek přejde se na stav Čekací smyčka, zde se nuluje čítač adresy BRAM a pokud je čítač adresy DDR na hodnotě 15000 vynuluje se i on. Z tohoto stavu se přejde do stavu Čeká náběh pokud je blank_s = 0, jinak se přejde do stavu Čeká spád. Tento stav tu je z důvodu čekání na začátek vyčítání dat z bufferu, pokud už je načten a zapsán celý řádek do bufferu. Pokud je čtení započato,

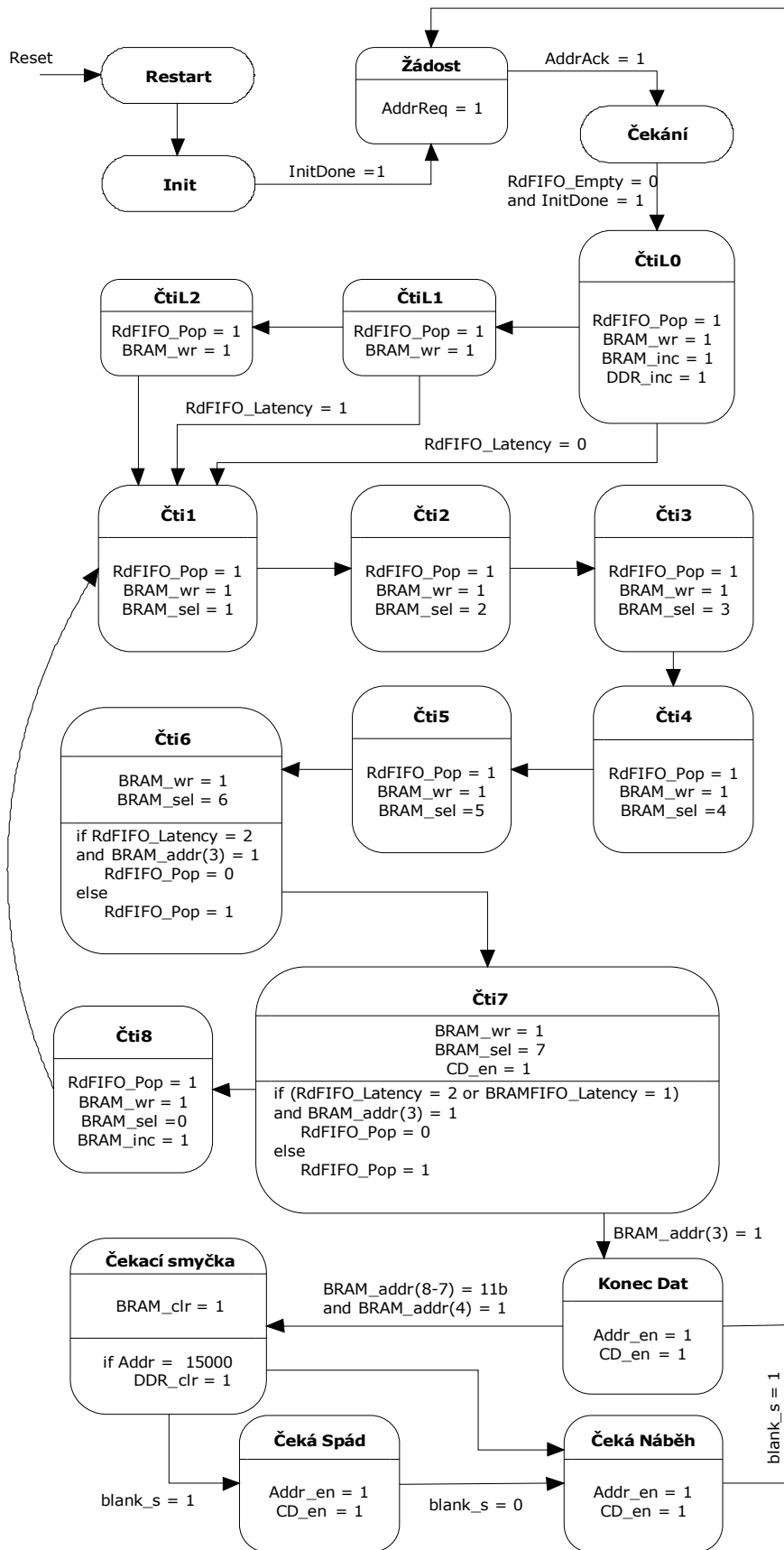
je indikováno shovením signálu blank_s na nulu a přejde se na stav Čeká náběh. Jedná se o stav, kdy se čeká dokud nejsou vyčteny všechny data z bufferu a může být započato načítání dat se zápisem nového řádku. To je indikováno nastavením blank_s na log. „1“ a přejde se do stavu Žádost. Signál blank je přes klopný obvod typu D snímán frekvencí 50MHz, neboť je sám generován 40MHz a dále je porovnáván frekvencí 100MHz. Pokud by se tento obvod vynechal, EDK by hlásilo chyby v časování.



Obrázek 19: datová část čtení z DDR

4.2.1.2 Blok Řádek – buffer

Jedná se o paměť BRAM, která je zároveň redukcí z 32 bitů na 16 bitů. Je použita komponenta RAMB16_S18_S36, kde port A je 16 bitový a port B je 32 bitový. Port B je tedy použit pro zápis dat z NPI sběrnice a Port A slouží pouze pro čtení RGB hodnoty jednoho pixelu. Ale jelikož vývojový kit podporuje pouze 4 bity na jeden kanál barvy, využívá se tedy jen spodních 12 bitů.



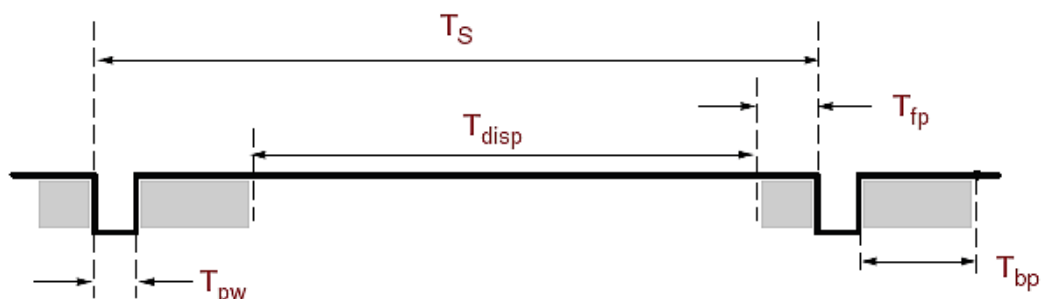
Obrázek 18: řídicí automat Čtení dat z DDR

4.2.1.3 Blok VGA časování

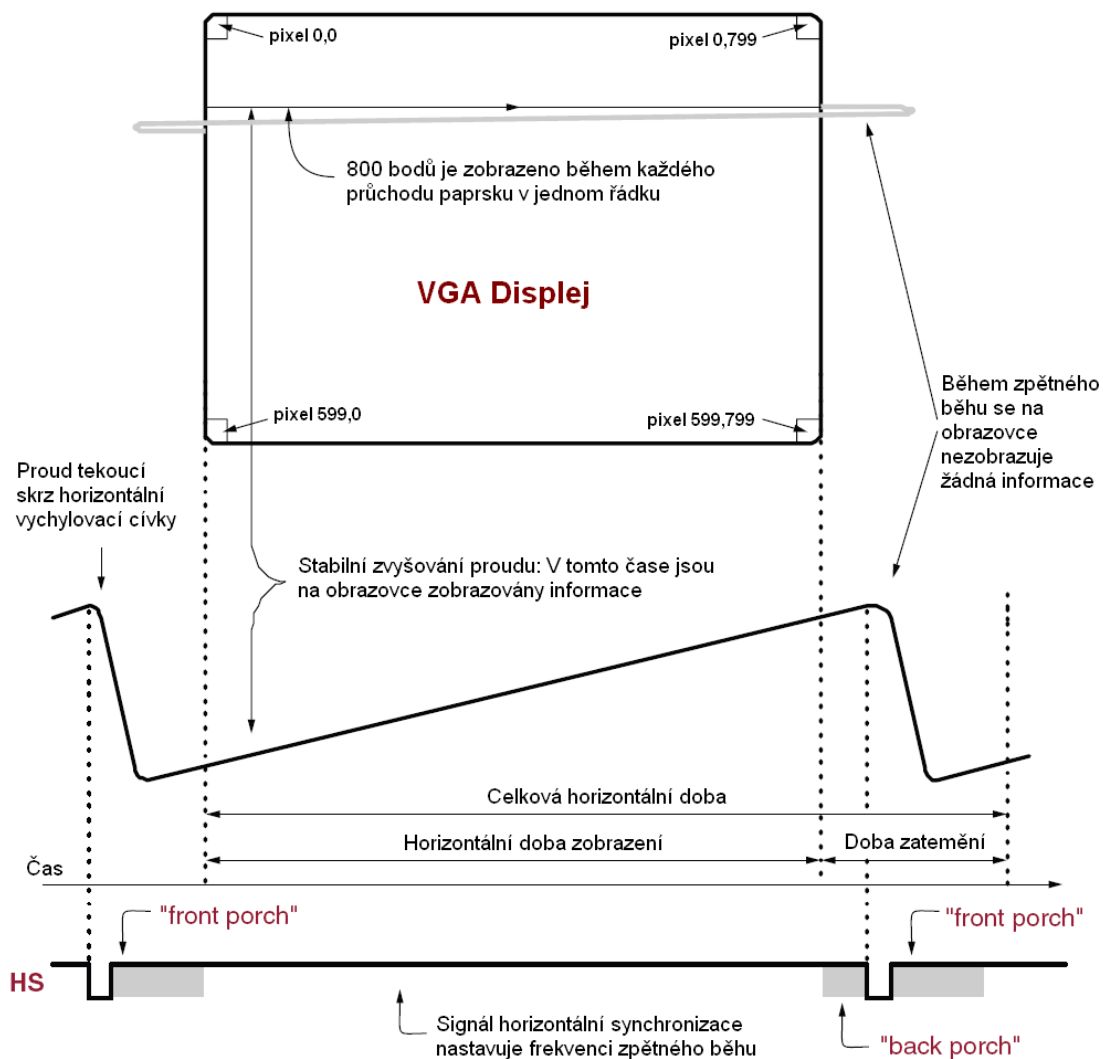
Pro aplikaci byl vybrán režim SVGA, grafické rozlišení 800 x 600 bodů, frekvence vykreslování bodů 40MHz, řádkový kmitočet 37.9kHz, snímkový kmitočet 60Hz. U starších CRT monitorů je obraz vytvářen elektronovým paprskem pohybujícím se po stínítku zleva doprava a odshora dolů. Během horizontálního vychylování vykresluje paprsek svým pohybem zleva doprava vždy jeden řádek. Když se paprsek dostane za pravý okraj obrazu, vrací se rychle zpět. Během tohoto zpětného běhu se na obrazovce nezobrazují žádné informace. Paprsek je vychylován i ve vertikálním směru, ale mnohem pomaleji oproti horizontálnímu vychylování. Jde-li paprsek směrem dolů, je na stínítku vytvářen obraz, kde barva bodu je tvořena signály RGB, které jsou přerušovány zatemňovacími impulsy. Vrací-li se paprsek doleva nahoru nezobrazují se žádné informace. Během zatemnění (nezobrazují se žádné informace) mají signály RGB takovou hodnotu, že se bod na obrazovce nerozsvítí. Průběh je zobrazen na *obrázku 20*. V *tabulce 4* jsou zobrazeny parametry časování potřebné k generování synchronizačních pulzů, jako je na *obrázku 19*.

Symbol	Parametr	Vertikální sync			Horizontální sync	
		Čas	Takty	Řádky	Čas	Takty
T_S	Doba sync pulsu	16.579ms	663 168	628	26.4us	1056
T_{DISP}	Doba zobrazení	15.84ms	633 600	600	20.0us	800
T_{PW}	Šířka pulsu	0.106ms	4 224	4	3.2us	128
T_{FP}	Front porch	0.026ms	1 056	1	1.0us	40
T_{BP}	Back porch	0.607ms	24 288	23	2.2us	88

Tabulka 4: parametry časování



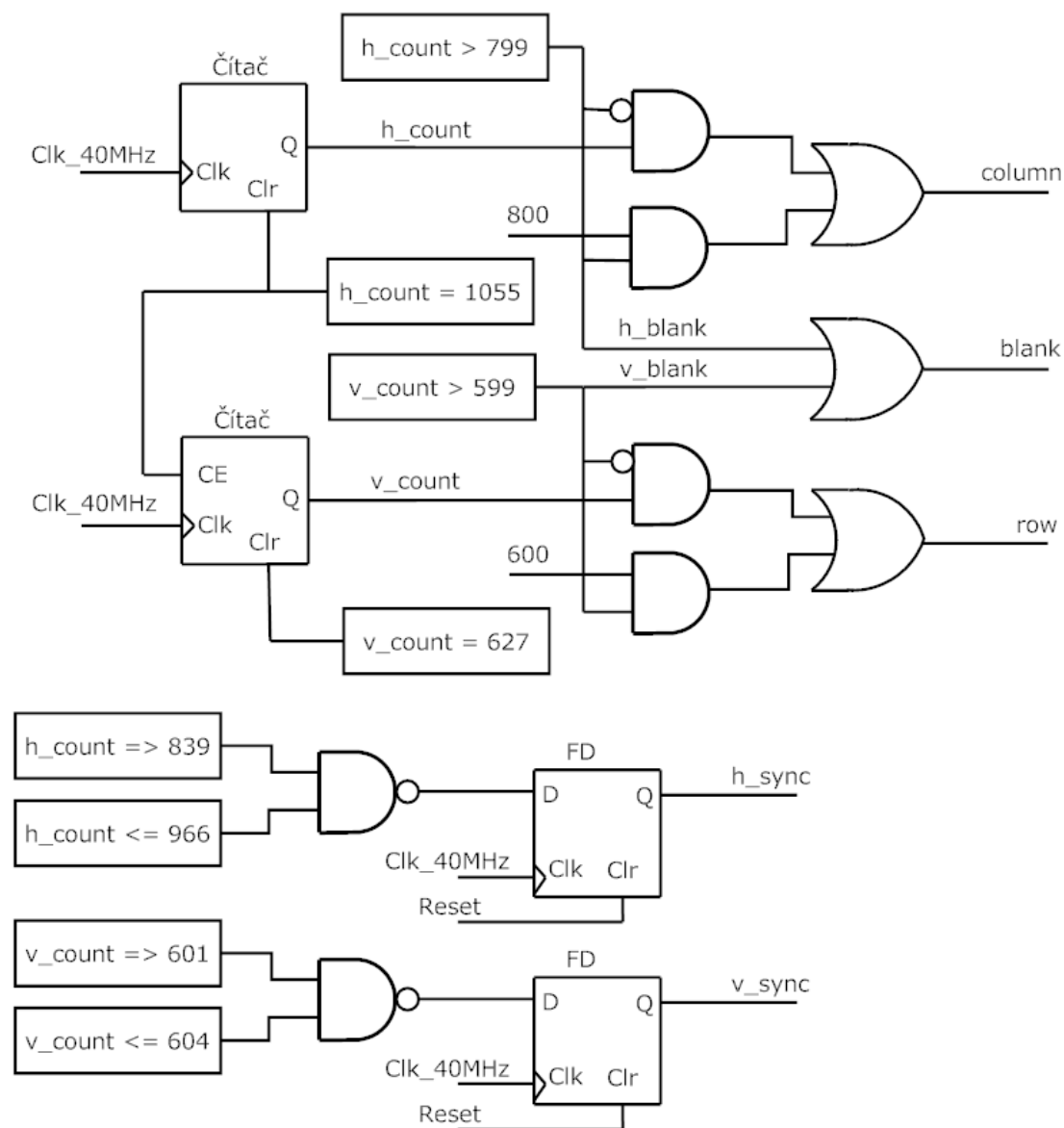
Obrázek 19: časový průběh synchronizačních signálů



Obrázek 20: průběh zobrazování na displeji

Princip generování synchronizačních pulsů a zatemnění (blank) je realizován pomocí dvou čítačů, běžící na hodinové frekvenci 40MHz. Jeden čítá neustále hodnotu h_count a je nulován dosáhne-li hodnoty 1055. Dosáhne-li h_count hodnoty větší jak 799, je generován signál h_blank a hodnota Column je nastavena na hodnotu 800, jinak obsahuje stejnou hodnotu jako h_count . Druhý čítač čítá hodnotu v_count pouze, když h_count je roven 1055. Nuluje se, je-li v_count roven 627. Při hodnotě v_count větší jak 599 je generován signál v_blank a hodnota Row je nastavena na 600, jinak má stejnou hodnotu jako v_count . Signál blank je nastaven podle signálu h_blank , nebo v_blank . Puls horizontální synchronizace je generován je-li hodnota $h_count \Rightarrow 839$ a zároveň je $h_count \leq 966$. Stejně tak vertikální synchronizační puls je generován, je-li hodnota $v_count \Rightarrow 601$ a zároveň je $v_count \leq 604$. Jelikož

se začíná zobrazováním, hodnota Column může vyčítat RGB hodnoty bodů z bufferu řádku. Na *obrázku 21* je zobrazen návrh VGA časování.

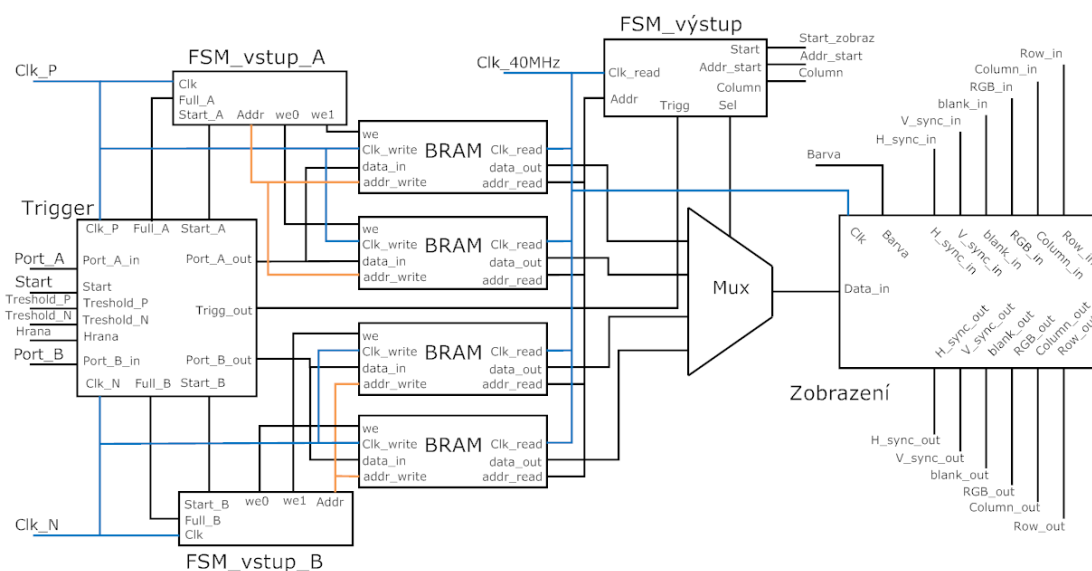


Obrázek 21: schéma VGA časování

4.2.1.4 Blok Příjem

Tento blok je rozdělen do dalších podbloků, jak je zobrazeno na *obrázku 22*. Jak je z blokového schématu patrné obsahuje trigger, dva FSM_vstupní bloky, čtyři paměti BRAM, FSM_výstup, multiplexor a blok zobrazení. Trigger spouští ukládání dat z AD převodníku do paměti BRAM, sám je spouštěn podle nastavení. Horní dvě paměti běží na hodinách ClkP(DCO+) a dolní dvě paměti jsou připojeny k hodinám

ClkN(DCO-). Vyčítání dat z paměti je realizováno pomocí bloku FSM_výstup, ten je nastavený pro všechny paměti se stejnou adresou a vyčítání je uskutečněno výběrem dat multiplexorem. Z důvodu zobrazování běží vyčíslní dat z paměti na frekvenci 40MHz. Data zmultiplexoru se dále dostávají do bloku zobrazení, ten má na starosti aby se dané body zobrazily ve správném poli.

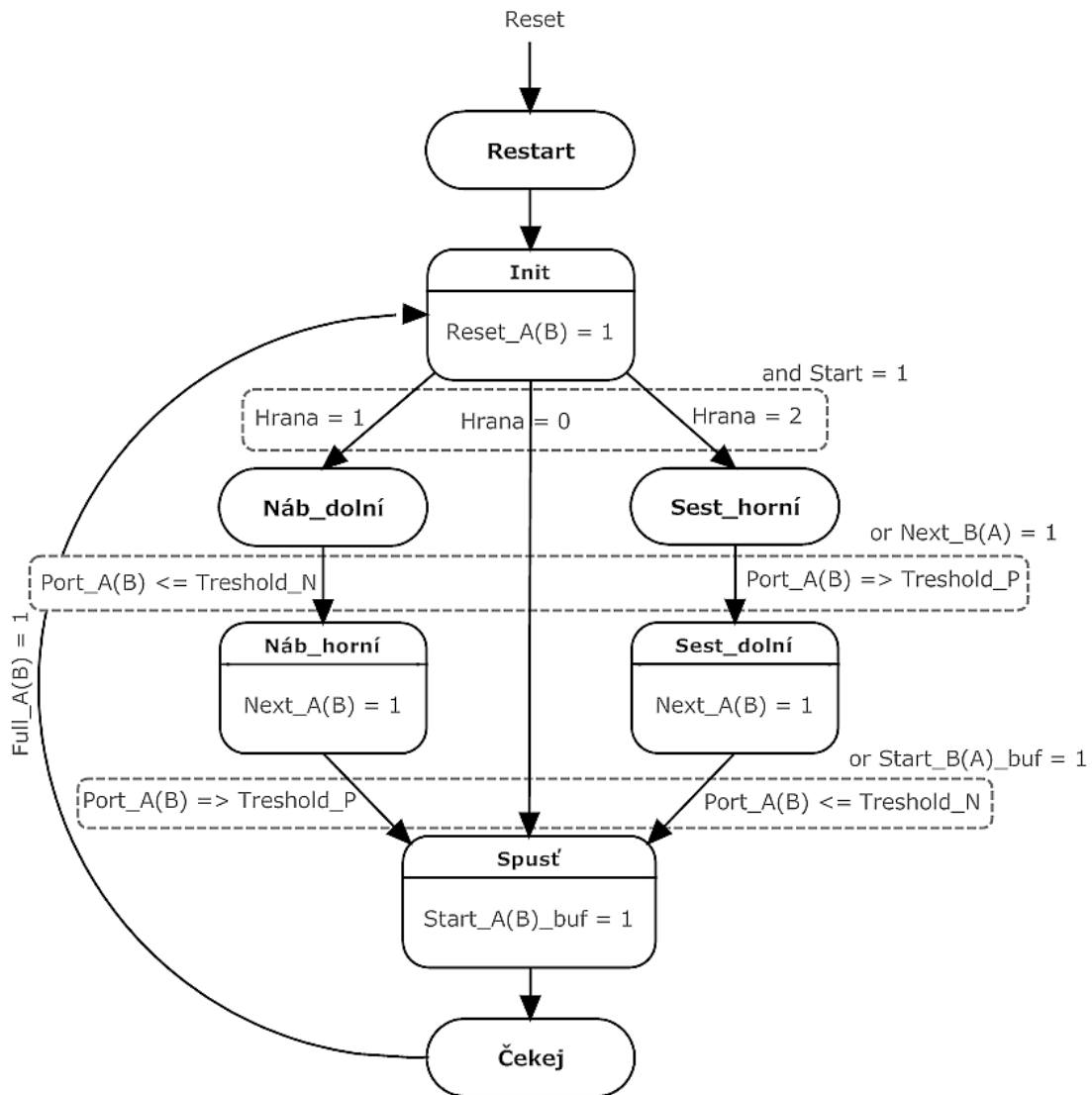


Obrázek 22: blokové schéma příjmu

4.2.1.4.1 Trigger

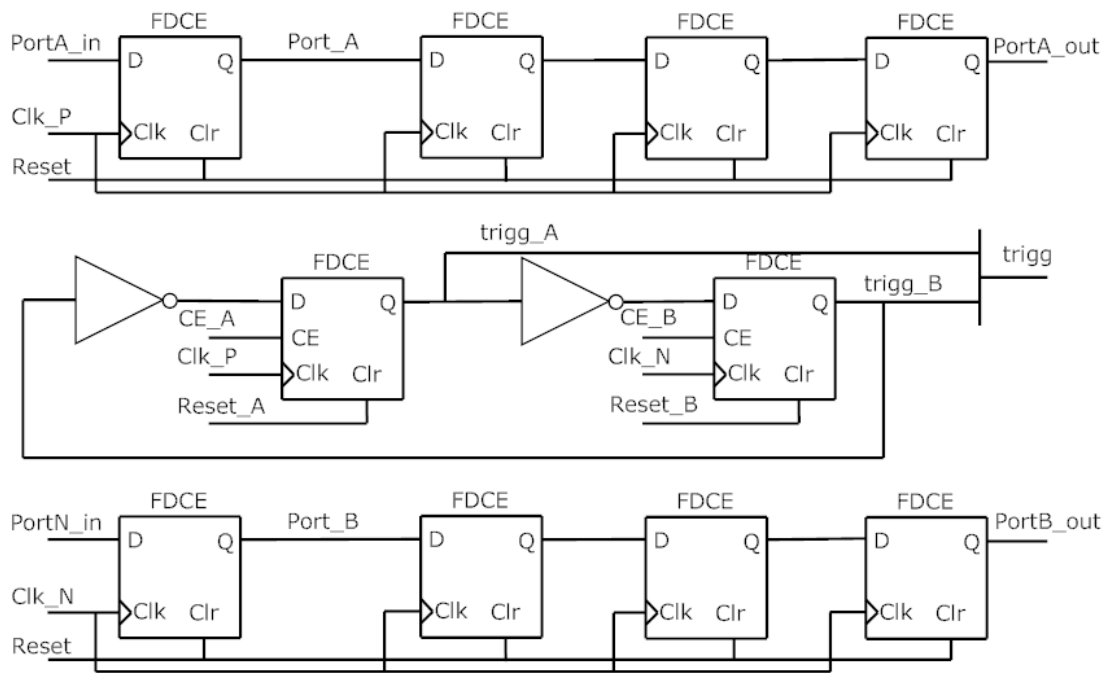
Trigger má za úkol podle nastavení hlídat zda proběhla náběžná, nebo sestupná hrana, nebo je vypnut a data jsou do paměti ukládána neustále. Realizován je dvěma automaty, na obrázku 23 je jeden zobrazen a hodnoty v závorkách odpovídají druhému automatu. Datová část je zobrazena na obrázku 24. Funkce automatu je následující. Po resetu přejde do stavu Init, kde se podle nastavení typu reakce rozhodne do jakého stavu dál a jeli spuštěn pokračuje do dalších stavů. Pokud jde o reakci na náběžnou hranu přejde do stavu Náb_dolní, odsud se dostane do stavu Náb_horní, je-li splněna podmínka $\text{Port_A(B)} \leq \text{Threshold_N}$, nebo nepřešel-li do tohoto stavu druhý automat. Odsud se dostane do stavu Spusť splněním podmínky $\text{Port_A(B)} \geq \text{Threshold_P}$, nebo jestli se do tohoto stavu dostal druhý automat. Pokud reaguje na sestupnou hranu dostane se automat do stavu Sest_horní. Pro přechod do stavu Sest_dolní musí být splněna podmínka $\text{Port_A(B)} \leq \text{Threshold_N}$, nebo dostane-li se do tohoto stavu druhý automat. Do stavu Spusť se dostane opět splněním

podmínky $\text{Port}_a(B) \leq \text{Threshold}_N$, nebo dostane-li se do tohoto stavu druhý automat. Jestliže trigger nereaguje na žádnou hranu přejde rovnou do stavu Spust', zde se spustí ukládání do paměti BRAM. V následujícím stavu se čeká dokud není paměť BRAM naplněna, pokud je plná přechází na stav Init.



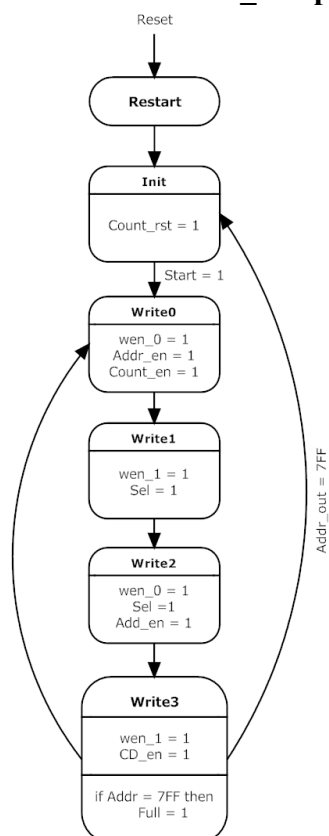
Obrázek 23: řídicí automat triggeru

V datové části jsou vstupní porty z externího modulu přivedeny na klopné obvody typu D pro zachycení dat. Ty dále pokračují do dalších třech klopných obvodů a to z důvodu zpoždění dat proto, aby se zapsala hodnota, která vyvolala spuštění ukládání dat. Aby se začalo vyčítat z paměti, do které byla uložena první hodnota, je zde zapojení, které podle spuštění určí, který port zahájil ukládání, aktivováním jednoho ze dvou signálů. $\text{CE}_A(B)$ je rovno $\text{Start}_A(B)_buf$ a výstupní hodnota je nulována ve stavu Init.



Obrázek 24: datová část triggeru

4.2.1.4.2 FSM_vstup

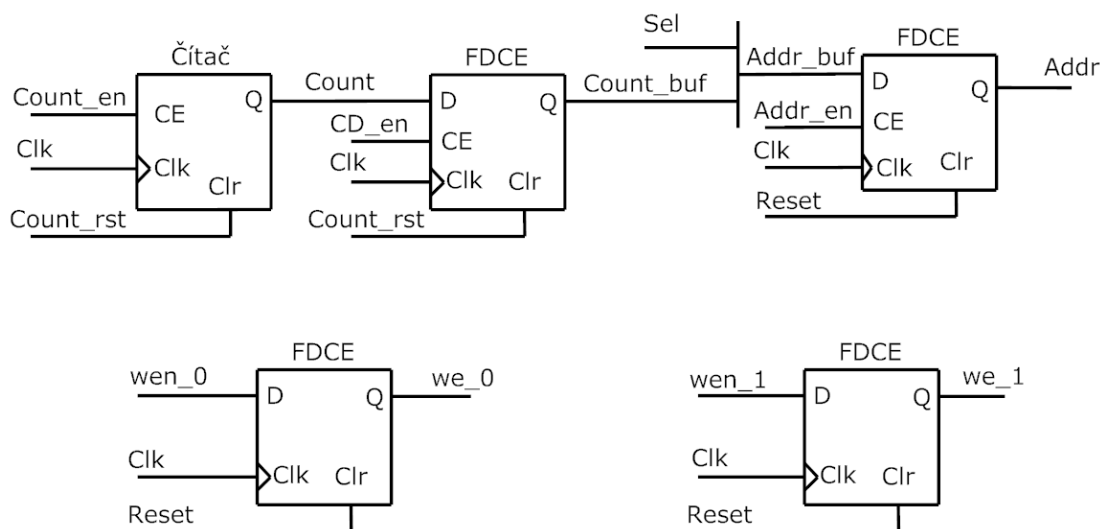


Obrázek 25: řídicí automat

FSM_vstup

Bloky FSM_vstup_A a FSM_vstup_B jsou realizovány stejným automatem, jen mají jiné vstupní a výstupní signály. Na obrázku 25 je zobrazen. Je navrhnut tak, aby stíhal i vyšší frekvence. Po resetu přejde do stavu Init, zde se nuluje číta. Do dalšího stavu se dostane tak, že od triggeru dostane impuls, aby začal. Dostane se do stavu Write0, zde se zapíše hodnota do paměti BRAM připojené na we_0, v tomto stavu je také distribuována adresa od čítače, která je složena z hodnoty čítače a spodní bit je tvořen signálem Sel. Zároveň je povoleno čítači inkrementovat hodnotu. V dalším stavu Write1 se zapíše do paměti připojené na signál we_1, zároveň je změněna hodnota Sel na

jedna. Ve stavu Write2 se opět zapisuje do první paměti a je distribuována adresa, která je o jedna větší než předcházející. Ve stavu Write3 se zapisuje do druhé paměti, povolí se distribuce hodnoty z čítače pro tvorbu adresy. Pokud je v tomto stavu adresa rovna hodnotě 7FF, nastaví se signál Full a skočí se na stav Init, jestliže je hodnota menší pokračuje se stavem Write0. V datové části na *obrázku 26* je zobrazen obvod pro generování adresy a klopné obvody pro signály povolení zápisu.



Obrázek 26: datová část FSM_vstup

4.2.1.4.3 Blok BRAM

Paměť BRAM je realizována komponentou RAMB16_S9_S9, která má oba porty o šířce 8 bitů. Port A je určen pro zápis a port B pro čtení dat.

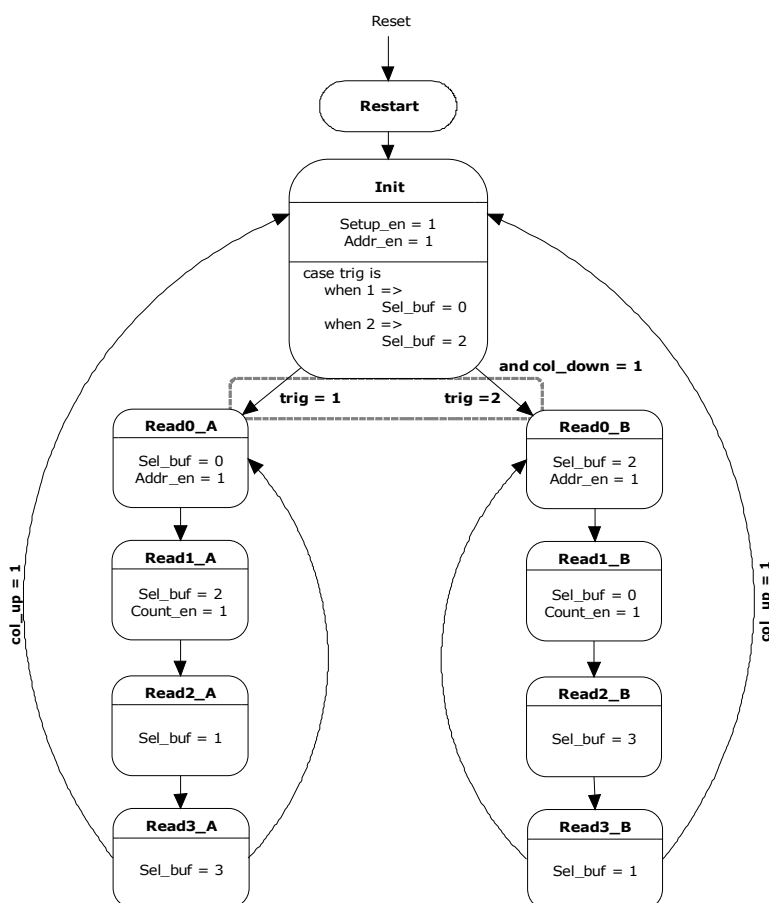
4.2.1.4.4 Blok Mux

Jedná se o klasický multiplexor, který podle signálu Sel z bloku FS_výstup vybírá data ze všech čtyř pamětí s uloženými naměřenými daty.

4.2.1.4.5 Blok FSM_výstup

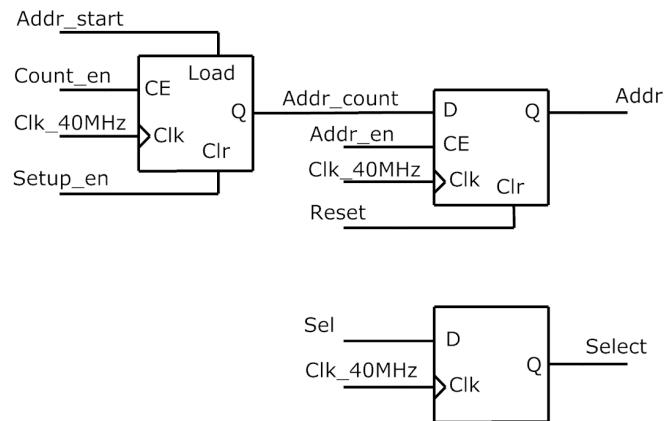
Tento blok má na starosti čtení dat z pamětí BRAM pro zobrazení na monitor. Řídící automat je zobrazen na *obrázku 27*. Který po resetu přejde do stavu Init, v něm se načte Addr_start jako počáteční hodnota čítače a zároveň je tato adresa distribuována na výstup. Další stav je vybrán podle hodnoty trig, která určuje do jaké

dvojice paměti se začalo ukládat jako první a zároveň se podle trig určí hodnota Sel, která ovládá multiplexor. Do určeného stavu hodnotou trig se dostane pouze, pokud je col_down aktivní. Ve stavu Read0_A je distribuována adresa na výstup a zároveň je Sel nastaven na nulu. V dalším stavu Read1_A je povoleno čítači inkrementovat hodnotu a Sel je nastaven na 2. Ve stavu Read2_A je pouze nastaven Sel na 1. Stav Read3_A změní hodnotu Sel na 3 a pokud je nastaven signál col_up skočí na Init, pokud ne přejde se na stav Read0_A. Stavy Read0_B až Read3_B jsou podobné jako stavy s označením A, jediný rozdíl je v nastavení signálu Sel. Col_down je nastaven, jeli Column roven sloupci odpovídající levému okraji zobrazovaného pole. Podobně je nastaven col_up jen s tím rozdílem, že se jedná o pravý okraj zobrazovaného pole.



Obrázek 27: řídicí automat FSM_výstup

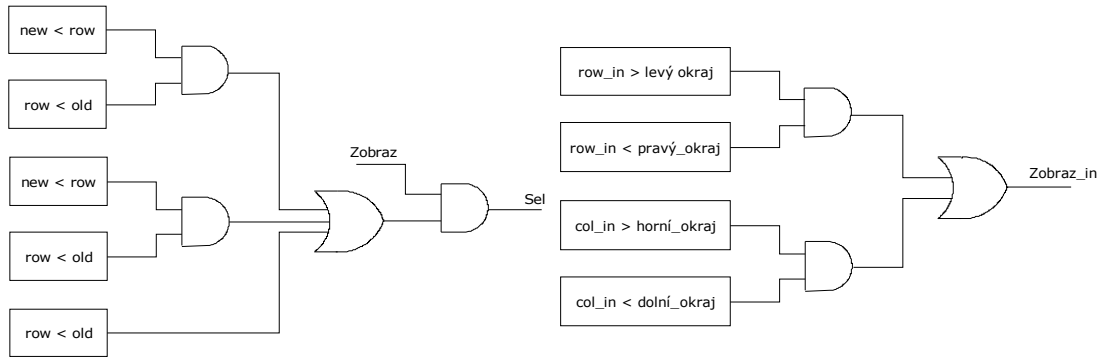
Na obrázku 28 se nachází datová část, která obsahuje čítač, jenž si jako počáteční hodnotu nastavuje Addr_start a klopný obvod jenž distribuuje adresu pro paměti. Druhý klopný obvod předává hodnotu Sel multiplexoru, ten podle ní vybírá data z paměti.



Obrázek 28: datová část FSM_výstup

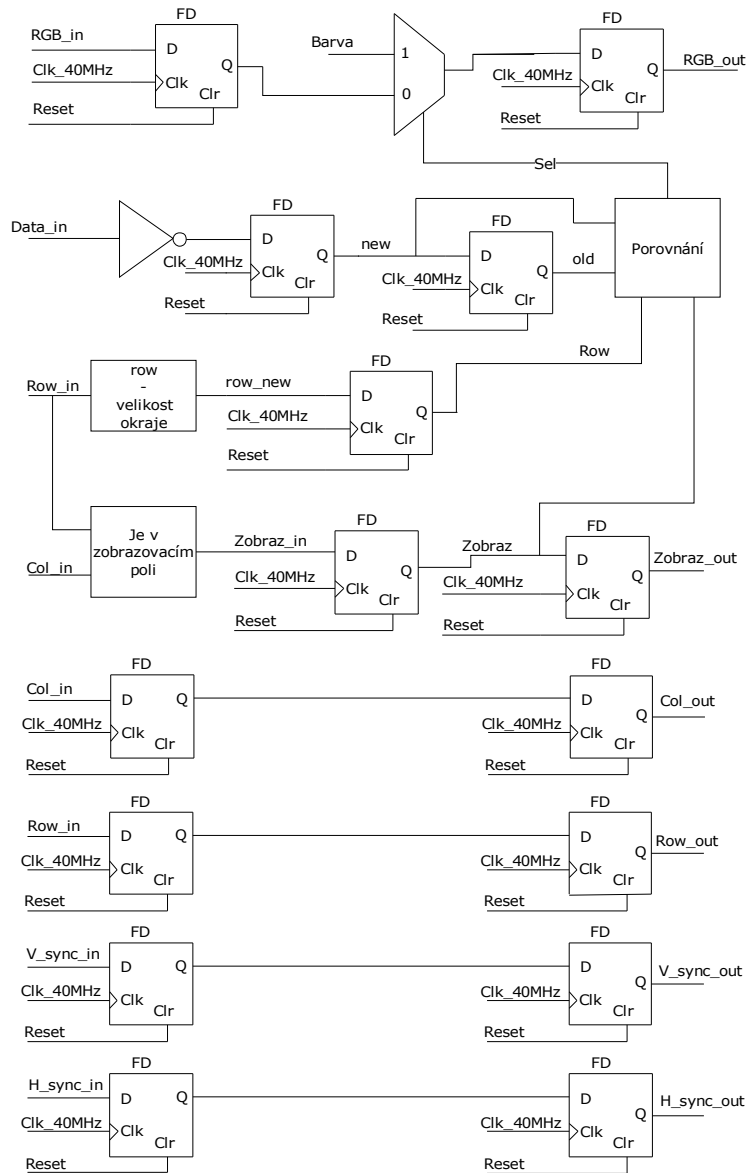
4.2.1.4.6 Blok Zobrazení

Tato část má na starosti zobrazování naměřených dat v určeném poli. Celý obvod je znázorněn na *obrázku 31*. Návrh je vytvořen pro data s kódováním offset binary, který se nastavuje na externím modulu pomocí jumperu. Vstupní data jsou invertorem znegována, neboť hodnota Row, s kterou se porovnávají má na horním okraji obrazovky nulovou hodnotu a směrem dolů, roste. Takže nyní bude mít vstupní napětí +0.512V binární hodnotu 00000000b. Takto upravená data jsou posílána do dvou klopných obvodů, takže máme vždy k dispozici starou a novou hodnotu. Ty jsou následně porovnávány s hodnotou Row, která je takt dopředu upravena tak, že se od původní hodnoty Row odečte poloha horního zobrazovacího pole, ale jelikož se zobrazují hodnoty od 0.5V do -0.5V je poloha okraje brána menší o 3. Celé porovnávání je zobrazeno na *obrázku 29*. Zároveň se dopředu generuje signál zobraz, jenž udává, že se může zobrazovat, neboť souřadnice vychází do zobrazovacího pole. Generování signálu zobraz je demonstrováno na *obrázku 30*. Podle výsledku porovnání a signálu zobraz se přepíná multiplexor, který buď přepošle hodnotu RGB, nebo místo ní přepošle hodnotu Barva. Aby se celý obraz zobrazoval korektně, jsou signály H_sync, V_sync, blank, Row, Column, RGB vedeny přes dva klopné obvody. Signál RGB má v cestě mezi klopnými obvody již zmiňovaný multiplexor.



Obrázek 29: porovnávací část

Obrázek 30: generování signálu zobraz



Obrázek 31: návrh obvodu zobrazení

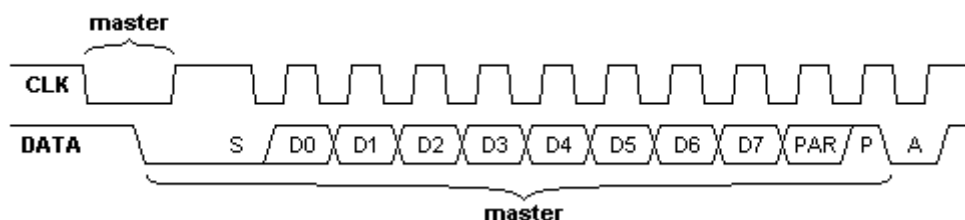
4.2.1.5 *Blok Kurzory*

Tento blok slouží k zobrazování třech kurzorů. Jmenovitě pro časovou osu, napěťovou osu a úroveň treshold pro trigger. Jsou zde tři multiplexory, které propouští buď hodnotu RGB, nebo Barvu konkrétního páru kurzorů. Ovládány jsou signálem generovaným porovnáním nastavených hodnot procesorem s hodnotami Row či Column podle orientace kurzorů. Tento signál je poslán do multiplexoru pouze pokud je aktivní i signál Zobraz, který je přiveden z bloku Příjem. Z posledního multiplexoru jsou hodnoty RGB dále distribuovány přes klopné obvody typu D, stejně jako ostatní potřebné signály k zobrazování.

4.2.1.6 *Blok Myš*

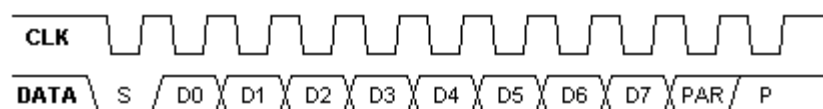
Myš je připojena přes konektor PS/2. Jedná se o synchronní obousměrnou komunikaci master-slave. Hodiny vždy generuje slave, master je nadřazen a pomocí signálu hodin povoluje, zakazuje, nebo přerušuje přenos dat. Hodinový kmitočet se pohybuje v rozmezí cca 10-16kHz. Oba signály (data i hodiny) jsou řešeny s otevřeným kolektorem.

Přenos master – slave: master musí vyslat požadavek na přenos dat. To se provede stáhnutím hodin a log. „0“ nejméně na 100us, tím se zablokuje případný přenos opačným směrem. Poté master stáhne na log. „0“ i data a čeká dalších 5us. Pak master uvolní hodiny, na tuto změnu slave zareaguje nejdéle do 10ms tak, že začne generovat hodiny. Slave vzorkuje data přesně uprostřed log. „1“ úrovně hodin, master tedy musí měnit hodnotu během log. „0“ úrovně hodin. Na *obrázku 32* je znázorněn přenos. Nejprve jde start bit, pak následují data, lichá parita, stop bit a následně je na data slavem zapsán potvrzovací bit ack.



Obrázek 32: přenos master - slave

Přenos slave – master, pokud jsou obě linky hodin a data v úrovni log. „1“ po dobu nejméně 50us může slave samovolně zahájit přenos. Datový rámec obsahuje celkem 11 bitů. Přenos je zahájen start bitem, následuje 8bitů dat řazených od LSB, další bit je parita a nakonec je stop bit. Na *obrázku 33* je zobrazen přenos od myši.

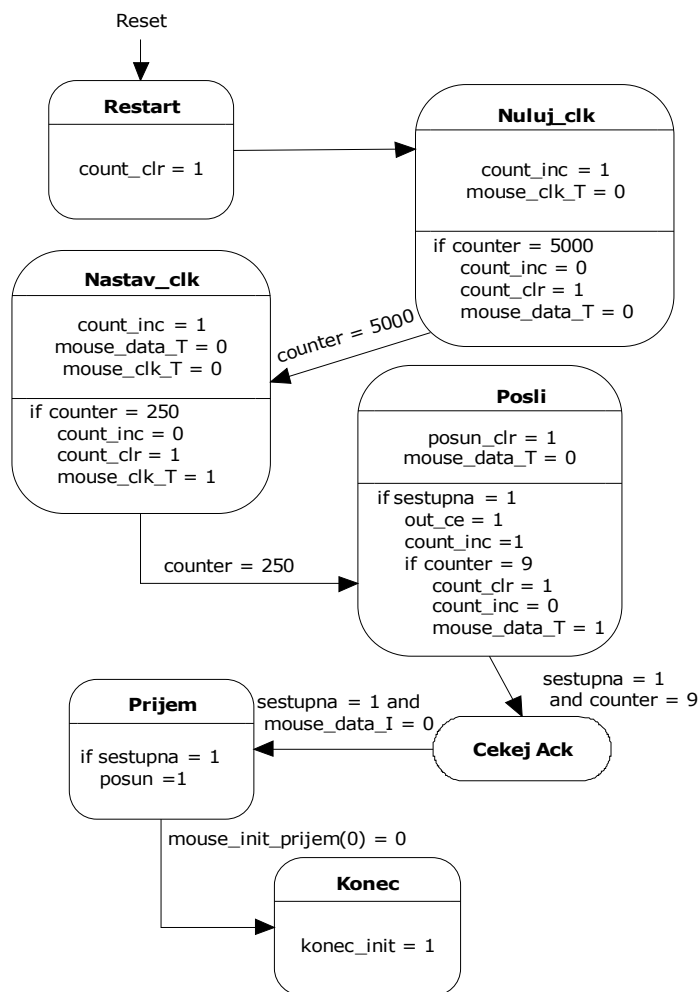


Obrázek 33: přenos slave – master

Příjem v FPGA je realizován pomocí automatu zobrazeného na *obrázku 34*. Po resetu je automat ve stavu Nuluj_clk, kde se stáhnou hodiny na log. „0“ a čeká se přibližně 100us. Pak se přejde do stavu Nastav_clk, kde se stáhnou i data, čeká se 5us a povolí se hodiny. Až myš začne generovat hodiny, pošlou se v dalším stavu data s hodnotou čtení ID zařízení. Po poslání se čeká, až zařízení potvrdí příjem. Následně je ve stavu Příjem, přijato 11 bitů. Jakmile je přijato přejde se do stavu Konec, kde se spustí příjem dat od myši. To je řešeno v datové části na *obrázku 35*. Pokud je nastaven Konec_init, jsou neustále přijímána data. Princip je následující, do 32 bitového signálu se postupně se sestupnou hranou hodin, posouvá datový signál od zařízení. Pokud se na nulté pozici objeví log. „0“ je indikován konec přenosu a přijatá data jsou upravena a poslána dál. Vezmou se pouze tři osmi bitová čísla. Nultý byte na pozici osm až jedna obsahuje bitové příznaky jednotlivých tlačítek btn_left, btn_right a btn_middle. Další byte na pozici 19 až 12 obsahuje relativní změnu polohy myši v ose x, poslední byte na pozici 32 až 23 obsahuje relativní změnu polohy myši v ose y. Tyto hodnoty jsou dále zpracovány v jiné části, kde generují přerušení pro procesor a vypočítávají pozice pro kurzor. V datové části je ukázán obvod příjmu dat během inicializace zařízení. Dále je tu detekce sestupné hrany hodin generované myší, čítač použitý pro vysílání dat, nebo detekce uplynutí konkrétní doby.

4.2.1.7 Blok FSL_řadič

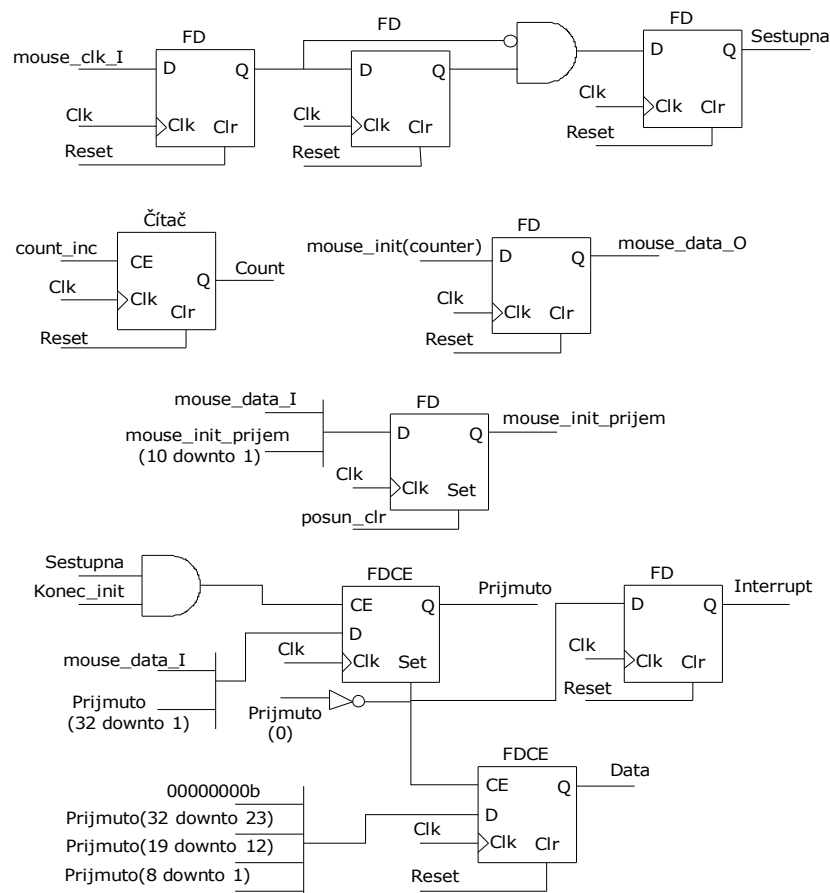
Slouží ke komunikaci s procesorem, ten pomocí FSL sběrnice nastavuje jednotlivé hodnoty pro chod jádra, nebo řídicí signály externího modulu. Pro příjem a vysílání je použita metoda vygenerovaná prostředím EDK. FSL řadič na žádost od procesoru, mu přeposílá pozici kurzoru. Přijímáno je 32 bitů z čehož spodních 24 je pro data a zbytek ovládá klopné obvody typu D. Takto je rozšířeno dostatečné množství signálů potřebných pro ovládání a nastavení jednotlivých bloků.



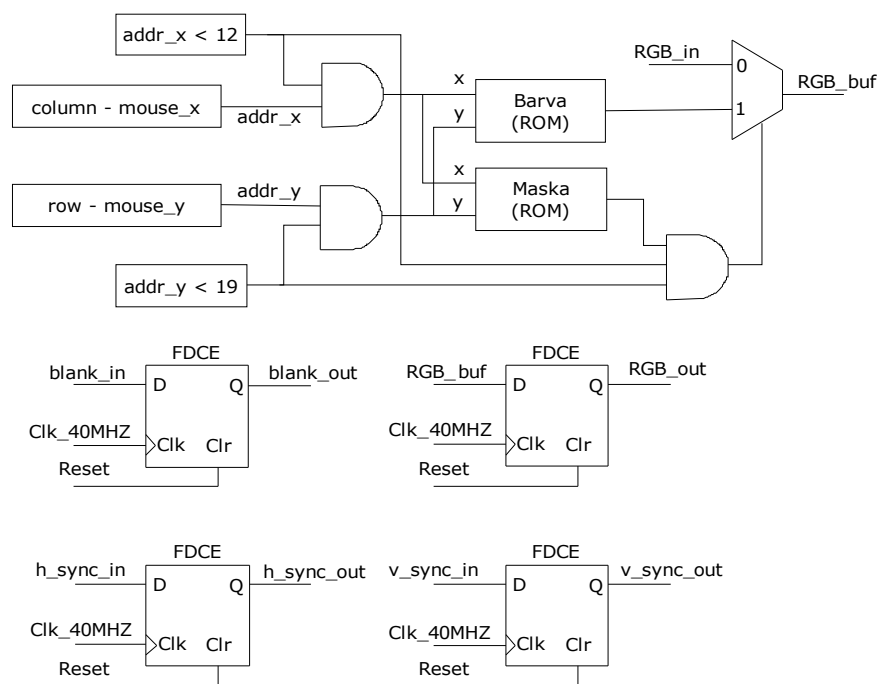
Obrázek 34: řídicí automat myši

4.2.1.8 Myš grafika

Jedná se o část, kde je zobrazován kurzor myši. Na *obrázku 36* je znázorněno použité schéma. Základem jsou dvě paměti ROM, jedna obsahuje data pro maskování a druhá obsahuje barvy jednotlivých bodů. Od hodnoty Column je odečtena pozice x kurzoru, pokud je tato hodnota menší jak 12 (šířka kurzoru), je použita jako adresa x. Podobně se od hodnoty Row odečte pozice y a pokud je tato hodnota menší jak 19 (výška kurzoru), je použita jako adresa y. Do jednoho vstupu multiplexoru je přivedena RGB hodnota a do druhého vstupu hodnota z paměti s barvou pro dané adresy x a y. Paměť maska generuje signál, zda se má zobrazovat kurzor, obraz pod kurzorem. Všechny potřebné VGA signály jsou vedeny přes klopné obvody typu D. Jelikož je toto poslední blok, který pracuje s daty pro zobrazení, tak se následně podle signálu blank určuje, zda se pošle RGB signál, nebo černá barva.



Obrázek 35: datová část myši



Obrázek 36: schéma myš grafika

4.2.2 Microblaze

Tento embedded procesor má na starosti veškeré nastavování a zobrazování pozadí spolu s dalšími potřebnými bitmapami, které jsou vyčteny z paměti flash do DDR paměti. Pozadí je v DDR dále upravováno dle změn nastavení uživatelem, či posuvu signálu a kurzorů na osách. V paměti flash jsou uloženy bitmapy, které jsou upraveny a připraveny k přímému zobrazování bez dalších změn.

Pozadí vypadá následovně, v horní části je mřížka o rozměrech 10x20, kde jedno políčko má rozměr 25x25 pixelů. Okolo mřížky jsou u os hodnoty daného času a napětí. V levé části jsou posuvníky pro kurzory měření napětí, hodnoty jsou zobrazovány pod posuvníky. Posuvníky pro kurzory, které měří hodnoty času jsou pod mřížkou a hodnoty jsou zobrazeny napravo od posuvníků. Na pravé straně od mřížky jsou posuvníky pro nastavení hodnot treshold pro úroveň triggeru. Ve spodní části se nachází posuvník pro posun signálu, dále tu jsou přepínače pro změnu hodnot rozsahu vstupního napětí 100mV/dílek, 200mV/dílek, 400mV/dílek, 500mV/dílek, 1V/dílek a 2.5V/dílek. Dále se zde nachází tlačítko pro spouštění snímání a pro zapnutí a nastavení triggeru. Možnost vypnutí a zapnutí zobrazování jednotlivých párů kurzorů.

4.3. Návrh schématu a plošného spoje

Schéma i plošný spoj je navržen ve vývojovém prostředí ORCAD. Plošný spoj je navržen se čtyřmi vrstvami, kde krajní vrstvy mají obsahují datové cesty a jsou na ně připájené součástky. Ve vnitřních vrstvách je obsaženo napájení a zem rozlité po celé ploše plošného spoje.

Součástky pro napájení jsou pouze v horní vrstvě a dále jsou jednotlivé napájení do ostatních obvodů distribuovány vnitřní vrstvou. Každý napájený obvod, má u svých nožiček připojených k napájení, blokovací kondenzátor.

Plošné spoje byly vyrobeny firmou PRAGOBOARD. Tato firma vyrábí plošné spoje na profesionální úrovni a opět mě nezklamala, neboť výsledné desky vypadají opravdu dobře.

Při zapojování a oživování jednotlivých součástí jsem přišel na některé nedostatky, které jsem v návrhu popletl. Jedná se o nepřipojený napájecí pin, ale to jsem vyřešil propojením se sousední nožičkou, na kterou je přivedeno napájení. Další chyba je, že vybíjecí diody u relátek mají být připojeny paralelně na cívky jednotlivých relátek a ne tranzistorů.

Na přiloženém CD je schéma, kde jsou již tyto chyby opraveny.

5. Testování

Externí modul byl proměřen a odzkoušen pomocí jednoduchého obvodu napsaného ve VHDL. Všechny signály probíhající na této desce byly zároveň zobrazeny na osciloskopu a vypadaly celkem slušně i při vyšším maximálním kmitočtu.

Všechny jednotlivé části jádra VGA_controller, byly odsimulovány pomocí programu modelsim. Každá z těchto částí je funkční, některé části byly spojeny do celku a taktéž odsimulovány. Části čtení z DDR, buffer řádku, VGA_časování, Myš, Myš grafika a FSL_řadič byly odzkoušeny na vývojovém kitu a jsou plně funkční. Dokonce pomocí těchto částí, po menších úpravách, vznikla plně funkční hra arkanoid, ovládaná myší a řízena procesorem microblaze, která byla představena jako ukázková aplikace na Dni Otevřených Dveří. Otestování jádra jako celku na přípravku nebylo provedeno, neboť EDK mělo problémy s mapováním hodin a proroutáním některých cest za splnění určitých časových podmínek. Některé tyto problémy se mi povedlo vyřešit, ale většinou nefungovalo komunikování procesoru s DDR pamětí přes PLB sběrnici a procesor se vždy zasekl. Z tohoto důvodu se mi nepovedlo odzkoušet osciloskop jako celek.

6. Zhodnocení

V této práci se podařilo vyrobit externí modul, který převádí analogový signál na digitální, s možností přepínání rozsahů vstupního signálu. Přípravek zvládá i vyšší vzorkovací frekvenci, než bylo zadáno a je kompatibilní s vývojovými deskami SPARTAN-3A Starter Kit a SPARTAN-3E Starter kit. Cena tohoto modulu je vypočtena v *tabulce 5*. Pokud se nepřipočte cena vývojových desek jedná se o pěknou cenu, která může konkurovat nejlevnějším modelům připojitelných PC, s tím že s tímto modulem lze dosáhnout vyšší vzorkovací frekvence. Tato cena by byla mnohem menší kdyby se nejednalo o kusovou výrobu a pak by byl rozdíl mezi prodávanými osciloskopy mnohem větší.

Jednotlivé části	Cena
AD9481	1540,-
MC100LVEL16D	273,-
AD8351	257,-
Ostatní součástky	cca 1600,-
Plošný spoj	1995,-
Celkem	5665,-

Tabulka 5: přehled cen částí modulu

Bylo vytvořeno jádro, které se dá připojit k embedded procesoru a pomocí MPMC k DDR pamětem. Bohužel vznikly problémy s mapováním více hodin a proroutováním některých cest. Jednotlivé části tohoto jádra jsou funkční.

Bylo vytvořeno grafické rozhraní, ale nebylo řádně odzkoušeno z důvodu již zmiňovaných problémů. Pro tvorbu grafického prostředí byl vytvořen program, který převádí obrázky do formátu, jenž dokáže vytvořené jádro zobrazit bez dalších nutných úprav.

7. Závěr

Výsledkem této diplomové práce je fungující modul, který převádí analogový signál na digitální. Ten je možné připojit k vývojovým deskám SPARTAN-3A Starter Kit a SPARTAN-3E Starter kit. Pro vstupní měřený signál je možno měnit rozsahy zobrazovaného napětí. Zobrazení je realizováno pomocí FPGA, DDR paměti a paměti flash. Nastavování a ovládání je intuitivní, neboť je k tomu použita počítačová myš a grafické prostředí. Bohužel zařízení jako celek nelze použít, neboť se vyskytly problémy při generování bitstreamu pro obvod FPGA a to při mapování rozložení cest hodin, nebo proroutování některých sběrnic k procesoru. Nicméně všechny vytvořené části jsou funkční.

V rámci této diplomové práce bylo vyrobeno funkční schéma. Z tohoto schématu byly vytvořeny podklady pro výrobu čtyřvrstvého plošného spoje, ten byl následně osazen a otestován. Dále bylo vytvořeno rozsáhlé jádro pro embedded procesor, grafické uživatelské prostředí a program pro microblaze, který přepisuje bitmapy, nebo ovládá externí modul.

8. Seznam použitých zkratk

FPGA	Field Programmable Gate Array
EDK	Embedded Development Kit
PLB	Procesor Local Bus
FSL	Fast Simplex Link
NPI	Native Port Interface
MPMC	Multi Port Memory Controller
BRAM	Block Random Access Memory

9. Seznam použité literatury

- [1] Artur Seibt: Osciloskopy od A do Z. Vydavatelství HEL, ISBN 80-86167-11-9

- [2] Datasheet analogove-digitálního převodníku AD9481
http://www.analog.com/UploadedFiles/Data_Sheets/AD9481.pdf

- [3] Datasheet pulsního stabilizátoru LM2574N 3.3V
<http://www.national.com/ds.cgi/LM/LM2574.pdf>

- [4] Datasheet IP core MPMC
http://www.xilinx.com/support/documentation/ip_documentation/mpmc.pdf

- [5] Časování VGA monitorů
http://www.epanorama.net/documents/pc/vga_timing.html

- [6] Datasheet k vývojovému kitu SPARTAN-3A Starter Kit
http://www.xilinx.com/support/documentation/boards_and_kits/ug330.pdf

10. Obsah CD

1. Text diplomové práce ve formátu pdf
2. Použitá literatura
3. Projekt z návrhového prostředí EDK
4. Projekt z návrhového prostředí Orcad