



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIOELECTRONICS

IMPLEMENTACE ALGORITMU PRO MĚŘENÍ PARAMETRŮ ENERGETICKÝCH MATERIÁLŮ V OBVODU FPGA

IMPLEMENTATION OF ALGORITHM FOR ENERGETIC MATERIAL MEASUREMENT
IN FPGA DEVICE

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

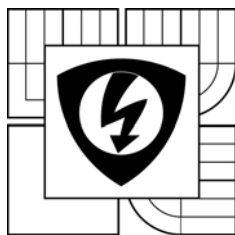
AUTOR PRÁCE
AUTHOR

Bc. JIŘÍ SLOVÁK

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Ing. ROMAN MARŠÁLEK, Ph.D.

BRNO 2014



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav radioelektroniky

Diplomová práce

magisterský navazující studijní obor
Elektronika a sdělovací technika

Student: Bc. Jiří Slovák

Ročník: 2

ID: 125631

Akademický rok: 2013/14

NÁZEV TÉMATU:

Implementace algoritmu pro měření parametrů energetických materiálů v obvodu FPGA

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s principy používaných algoritmů zpracování signálu v zařízení na měření parametrů energetických materiálů. Sestavte vlastní simulaci v prostředí MATLAB, která z měřeného jedno či vícekanálového záznamu vypočítá požadované parametry průběhu exploze (například detonační rychlost). Výsledky zpracujte do přehledné technické zprávy.

V jazyce VHDL popište a v prostředí ISE simulujte Vámi používaný algoritmus zpracování signálu pro zařízení na měření parametrů energetických materiálů. Algoritmus implementujte do dodané desky s FPGA a A/D převodníky. Činnost algoritmu ověřte na sadě záznamů změřených signálů. Výsledky zpracujte do přehledné technické zprávy.

DOPORUČENÁ LITERATURA:

[1] CHAN, E.M. Low-Cost Optoelectronic Devices to Measure Velocity of Detonation, Proceedings of SPIE Vol. 5649, 2005, p. 586-594.

[2] PROAKIS John G. Digital Communications. New York: McGraw-Hill Book, 1995.

Termín zadání: 10.2.2014

Termín odevzdání: 23.5.2014

Vedoucí práce: doc. Ing. Roman Maršálek, Ph.D.

Konzultanti diplomové práce:

doc. Ing. Tomáš Kratochvíl, Ph.D.

předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

ABSTRAKT

V textu diplomové práce je nejprve stručně pojednáno o problematice měření parametrů energetických materiálů v obecné rovině. Důraz je kladen především na popis detonační rychlosti a stručný rozbor nejčastěji používaných metod. Převážná část textu je věnována návrhu a popisu systému, jenž byl vytvořen v prostředí ISE Design Suite s využitím jazyka VHDL. Vývoj byl prováděn s ohledem na pozdější integraci do desky s FPGA a A/D převodníky. Funkčnost algoritmu detekce, implementovaného na základě modelu, byla ověřena v závěrečné části práce simulací zpracování reálných signálů optických sond.

KLÍČOVÁ SLOVA

Energetický materiál, VoD, Optická metoda, MATLAB, VHDL, FPGA

ABSTRACT

In the text of the master's thesis, it is at first briefly referred about Energetic Material Measurement topic in general. Emphasis is placed especially at the description of the Velocity of Detonation and short analysis of selected measurement method. The most significant part of the paper is dedicated to the design and description of the system that was created in ISE Design Suite environment using VHDL language. The development was performed with respect to oncoming integration into the board with FPGA and A/D converters. The operation of detection algorithm which was created based on the MATLAB model was verified in the final part of the thesis by simulation of processing of real optical probe signals.

KEYWORDS

Energetic Material, VoD, Optical Method, MATLAB, VHDL, FPGA

SLOVÁK, J. Implementace algoritmu pro měření parametrů energetických materiálů v obvodu FPGA. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2014. 44 s., 14 s. příloh. Vedoucí diplomové práce doc. Ing. Roman Maršálek, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svoji diplomovou práci na téma Implementace algoritmu pro měření parametrů energetických materiálů v obvodu FPGA jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

V Brně dne

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce doc. Ing. Romanu Maršálkovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne

.....

(podpis autora)

OBSAH

Seznam obrázků	VIII
Seznam tabulek	IX
Úvod	1
1 Detonační rychlost a metody měření	2
1.1 Detonační rychlost (VoD) a další parametry	2
1.2 Metody měření VoD	2
1.2.1 Optická metoda měření VoD	3
2 Návrh algoritmu a model systému	4
2.1 Signály optických sond	4
2.1.1 Charakter signálu a nežádoucí jevy	4
2.1.2 Definice pozice pulzu, okamžik detekce	5
2.2 Vstupní data pro návrh modelu	6
2.2.1 Parametry vstupních dat	6
2.2.2 Formát vstupních dat	7
2.3 Řetězec zpracování, systémový pohled a popis bloků	7
2.3.1 Převzorkování a filtrace signálu A/D převodníků	8
2.3.2 Nalezení okamžiku pulzu, diferenciál	10
2.3.3 Výpočet VoD	13
2.3.4 Zobrazení výstupních dat	13
3 Implementace systému v obvodu FPGA	15
3.1 Podmínky implementace, použité prostředky	15
3.1.1 HW prostředky pro zpracování	15
3.1.2 Použitá zařízení	16
3.1.3 Odhad nároků na paměť pro záznam	17
3.1.4 Vývojové prostředí	18
3.1.5 Parametrizace návrhu, soubory konstant	18
3.1.6 Speciální konstrukce jazyka VHDL, Record Types	19
3.2 Blokové schéma implementovaného systému	20
3.3 Rozhraní ADC ADS4229	21

3.3.1	Režim přenosu, LVDS	21
3.3.2	ADC jako zdroj hodinového signálu	22
3.3.3	Datové rozhraní	23
3.3.4	Konfigurační rozhraní.....	24
3.4	Blok Detector	25
3.4.1	Schéma bloku, vstupy a výstupy	25
3.4.2	Pracovní režimy bloku.....	26
3.5	Záznam signálu, blok Acquisition.....	27
3.5.1	Schéma bloku, vstupy, výstupy a komponenty	27
3.5.2	Stavový diagram.....	28
3.5.3	FIFO zásobník, Pre-Trigger okno	28
3.5.4	Režimy triggeru	30
3.5.5	Volba prahování.....	31
3.6	Použití BRAM IP Core	32
3.6.1	Popis bloku.....	32
3.6.2	Použití bloku v návrhu	33
3.7	Zpracování signálu, blok Processing	33
3.7.1	Schéma bloku, vstupy a výstupy a komponenty	33
3.7.2	Detekce prahováním.....	34
3.7.3	Detekce metodou tečen	35
3.8	Použití SW mikroprocesoru Picoblaze	36
3.8.1	Popis a vlastnosti	36
3.8.2	Připojení Picoblaze k FPGA, Vstupní a výstupní multiplex	36
3.8.3	Využití Picoblaze v projektu, přenos dat do PC	37
4	Vyhodnocení výsledků simulací	38
4.1	Nástroje simulace bloku TOP	38
4.1.1	Testování řídicích signálů.....	38
4.1.2	Testování funkce algoritmu	39
4.1.3	Waveform soubory	40
	Závěr	41
	Literatura	42
	Seznam zkratk	44
	Seznam příloh	45

SEZNAM OBRÁZKŮ

Obr. 1.1:	Princip měření VoD s použitím optovláknových sond	3
Obr. 2.1:	Příklad saturace špičky impulzu (Tetryl)	4
Obr. 2.2:	Příklad výskytu parazitních impulzů (Semtex)	5
Obr. 2.3:	Metody detekce pozice pulzu	5
Obr. 2.4:	Určení pozice pulzu metodou průsečíků (Bleskovice)	6
Obr. 2.5:	Řetězec zpracování signálu (systémový pohled)	7
Obr. 2.6:	Odhad spektra vybraného signálu optické sondy (Politol)	9
Obr. 2.7:	Výsledek filtrace kaskádou FIR a mediánového filtru	9
Obr. 2.8:	Metoda průsečíků, blokové schéma	10
Obr. 2.9:	Ukázka proložení sekvence metodou jednoduché lineární regrese	11
Obr. 2.10:	Algoritmus nalezení průsečíku dvou disktrétních přímek	12
Obr. 2.11:	Detailní zobrazení detekce pozice impulzu metodou průsečíků	14
Obr. 3.1:	Obecná struktura FPGA, převzato z [8]	15
Obr. 3.2:	Jednoduché schéma systému použitého pro implementaci	17
Obr. 3.3:	Zjednodušené blokové schéma nejvyšší úrovně designu	20
Obr. 3.4:	Příklad komplementárních signálů LVDS s offsetem V_{OCM} , převzato z [12]	21
Obr. 3.5:	Využití technologie DDR pro přenos vzorků signálu, převzato z [12]	22
Obr. 3.6:	Řetězec konverze diferenčního hodinového signálu	22
Obr. 3.7:	Řetězec konverze datového DDR signálu	23
Obr. 3.8:	Ukázka průběhů signálu SPI rozhraní ADS4229, převzato z [12]	24
Obr. 3.9:	Schematický symbol bloku Detector	25
Obr. 3.10:	Zjednodušená struktura bloku Detector	26
Obr. 3.11:	Schematický symbol bloku Acquisition	27
Obr. 3.12:	Zjednodušená struktura bloku Acquisition s popisem rozhraní BRAM	27
Obr. 3.13:	Stavový diagram bloku Acquisition	28
Obr. 3.14:	Princip kruhového zásobníku	29
Obr. 3.15:	Princip generace globálního triggeru, včetně tabulky bloku OR	30
Obr. 3.16:	Rozdíl v použití lokálního a globálního triggeru	31
Obr. 3.17:	Schematický symbol BRAM v režimu Simple Dual-Port	32
Obr. 3.18:	Schematický symbol bloku Processing	33

Obr. 3.19: Vnitřní struktura bloku Processing, včetně BRAM.....	34
Obr. 3.20: Stavový diagram detekce pozice impulsu prahováním	34
Obr. 3.21: Stavový diagram detekce impulsu metodou tečen	35
Obr. 3.22: Architektura mikroprocesoru Picoblaze, převzato z [15]	36
Obr. 3.23: Multiplexace vstupů mikroprocesoru Picoblaze [15]	37
Obr. 4.1: Řetězec simulace zpracování záznamů reálných signálů.....	39

SEZNAM TABULEK

Tab. 1.1: Přehled metod měření VoD [1], [3]	3
Tab. 2.1: Parametry záznamů, dostupných z předchozích měření, 2 varianty.....	6
Tab. 3.1: Vybrané parametry použitého FPGA [10], [11]	16
Tab. 3.2: Tabulka paměťových nároků měřeného záznamu	17
Tab. 3.3: Varianty nastavení prahu	31

ÚVOD

Energetické materiály, tedy materiály, které lze charakterizovat tím, že při jejich chemické přeměně dochází k uvolnění značného množství energie, jsou dnes využívány v mnoha odvětvích, ať už se jedná o pohonné hmoty v dopravních prostředcích, zdroje energie pro domácnosti nebo výbušné směsi, užívané pro vojenské nebo stavební účely. Nezbytným předpokladem pro využití těchto materiálů je schopnost kvantifikovat jejich účinky, čili určit jejich parametry.

Měření parametrů některých podskupin těchto materiálů může být poměrně problematické, avšak o to důležitější. Typickým příkladem je naposled zmiňovaná skupina výbušnin, jejichž analýza bude hlavní funkcí zařízení, pro které byl algoritmus navrhován. V první kapitole diplomové práce je tedy krátce pojednáno o metodách, jež jsou pro jejich analýzu používány a je objasněn význam měřeného parametru, označovaného jako VoD (Velocity of Detonation).

Kapitola druhá se zabývá návrhem algoritmu pro detekci pozic impulzů v signálech optických sond. Nejprve je pojednáno o očekávaném charakteru signálu a možných parazitních jevech, ovlivňujících výsledky měření. Objevuje se také diskuze o významu pojmu pozice impulzu. Základní principy jsou popsány prostřednictvím modelu systému, jenž byl v rámci semestrální práce vytvořen v prostředí MATLAB.

Stěžejní třetí kapitola textu obsahuje podrobný rozbor praktické realizace systému v jazyce VHDL na vývojové platformě firmy Xilinx. Text se zabývá zejména popisem jednotlivých komponent a technik, jichž bylo při implementaci využito.

V závěrečné části práce je nastíněno s jakými prostředky a s využitím jakých nástrojů byl navržený systém testován. Součástí příloh je krátké porovnání výsledků získaných pomocí modelování a simulace zpracování reálných signálů.

1 DETONAČNÍ RYCHLOST A METODY MĚŘENÍ

Jak bylo naznačeno v úvodu práce, kvantifikace účinků výbušnin, tedy podskupiny energetických materiálů, je poměrně náročný proces, který však má zásadní význam pro jejich použití. Následující kapitola slouží jako krátký úvod do problematiky měření parametrů těchto látek.

1.1 Detonační rychlost (VoD) a další parametry

Vlastnosti výbušnin, tedy látek schopných v extrémně krátkém čase uvolnit velké množství tepelné energie, jsou definovány řadou parametrů. Jmenovitě lze mluvit např. o hustotě výbušniny, detonačním tlaku, citlivosti, voděodolnosti, tepelné stabilitě či detonační rychlosti. Pro diplomovou práci je vhodné, objasnit zejména poslední ze jmenovaných parametrů.

Detonační rychlost, často označovaná zkratkou **VoD (Velocity of Detonation)**, je jedním z nejvýznamnějších parametrů výbušných materiálů. Označuje rychlost, se kterou se při detonaci šíří rázová vlna tělem výbušniny a je ovlivněna chemickým složením výbušniny. Její znalost je jedním z hlavních faktorů při volbě optimálního množství materiálu pro bezpečný a dostatečně efektivní průběh řízené exploze.

Hodnoty VoD se typicky pohybují v řádech tisíců m/s a značně tedy převyšují např. rychlost zvuku ve vakuu. Tento fakt výraznou mírou ovlivňuje používanou metodiku měření, kdy je třeba v extrémně krátkém čase zaznamenat a případně i určitým způsobem zpracovat, velké množství hodnot. Nejčastěji se jedná o intervaly od vyšších desítek mikrosekund až po milisekundy v závislosti na délce a typu výbušniny [1].

1.2 Metody měření VoD

Existuje poměrně široká škála metod, kterými lze detonační rychlost analyzovat, diplomová práce však není jejich studií a následující kapitola tedy obsahuje pouze krátký přehled pro doplnění pozadí samotného návrhu.

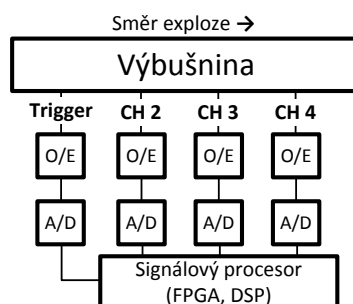
Metody měření VoD lze dělit dle různých hledisek. Existují metody, umožňující měřit rychlost spojitě v celé délce analyzovaného sloupce výbušniny, naproti tomu metody určené k měření pouze v určitých bodech např. pomocí optických sond. Odlišným způsobem je třeba přistupovat k měření tohoto parametru v otevřeném prostředí v porovnání s analýzou řízených explozí v prostoru uzavřeného charakteru. Tabulka 1.1 představuje krátký souhrn nejpoužívanějších metod se stručným popisem principu, na němž jsou založeny [1], [2].

Tab. 1.1: Přehled metod měření VoD [1], [3]

Metody	Stručný popis
Fotografické	Světlo, generované při explozi, je snímáno vysokorychlostními kamerami v reálném čase. Z grafu závislosti pozice čela vlny na čase pak, lze jednoduše vypočítat VoD.
Optické	Point to Point metody. S využitím optických vláken schopných přenést vznikající světelný signál jsou zaznamenány okamžiky průchodu rázové vlny body známých pozic. Z časů pulzů a rozdílů vzdáleností sond lze spočítat VoD.
Elektrické	Využívají změn elektrického odporu při postupné detonaci vodiče umístěného do (příp. okolo) těla nálože. Alternativou je použití ionizačních snímačů (snímače jsou se vznikem ionizace postupně vodivě spojeny).
Dautricheho	Je využito toho, že dvě vlny šířící se různou rychlostí urazí za stejnou dobu různou dráhu. Výbušnina je doplněna zápalnicí známé detonační rychlosti. VoD lze dopočítat z rozdílu vzdáleností geometrického středu mezi přípojnými body zápalnice a skutečného místa setkání obou vln.

1.2.1 Optická metoda měření VoD

Zařízení, ve kterém bude algoritmus implementován, využívá pro měření rychlosti detonace optickou metodu. Princip této metody je znázorněn na obr. 1.1



Obr. 1.1: Princip měření VoD s použitím optovláknových sond

Jak bylo naznačeno v předchozí podkapitole, optická metoda spočívá ve využití konečného počtu sond zavedených do těla výbušniny. Jednotlivé sondy jsou tvořeny optickými vlákny, schopnými zachytit a přenést světelné záblesky vznikající při explozi. Výstup těchto vláken je pak možno po vhodném optoelektronickém a následně analogově digitálním převodu analyzovat dostatečně rychlým signálovým procesorem.

Okamžik záblesku a následné destrukce vlákna se v průběhu signálu projeví ostrým maximem. Hlavním úkolem procesoru je spolehlivá detekce těchto maxim v jednotlivých kanálech a následný výpočet VoD. Dílčí kroky zpracování signálu ze sond a matematický popis výpočtu VoD jsou podrobně rozebrány v kapitolách, zabývajících se popisem algoritmu.

Z výše uvedeného popisu je zřejmé, že výstupem zařízení, implementujících tuto metodu, bude vektor hodnot detonační rychlosti, odpovídající umístění optovláknových sond [4].

2 NÁVRH ALGORITMU A MODEL SYSTÉMU

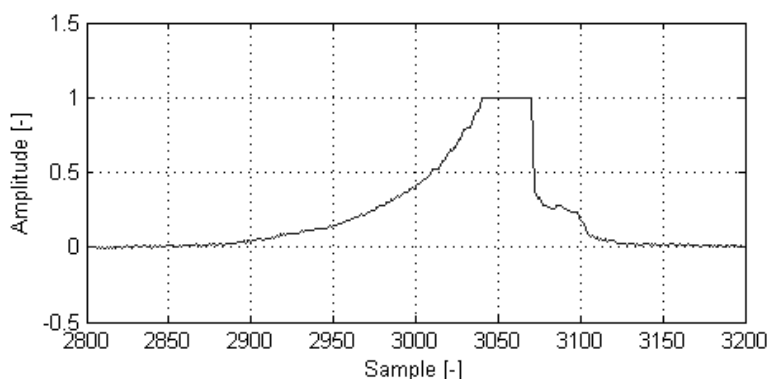
Při implementaci algoritmu do hardware je vhodné vycházet ze systémového modelu, jenž usnadní pozdější návrh a potvrdí jeho funkčnost. Kapitola druhá využívá model, vytvořený v rámci semestrálního projektu v prostředí MATLAB, pro studii výchozích podmínek pro návrh, zejména však pro teoretický rozbor samotného algoritmu. Kroky zpracování signálu jsou podrobně rozebrány v jednotlivých podkapitolách.

2.1 Signály optických sond

Pro návrh samotného číslicového systému je vhodné nejprve zvážit, jaký charakter bude mít vstupní signál, tedy signál, jenž bude systémem zpracováván. V případě použití optických sond lze vyjít z principu popisovaného v předchozí kapitole.

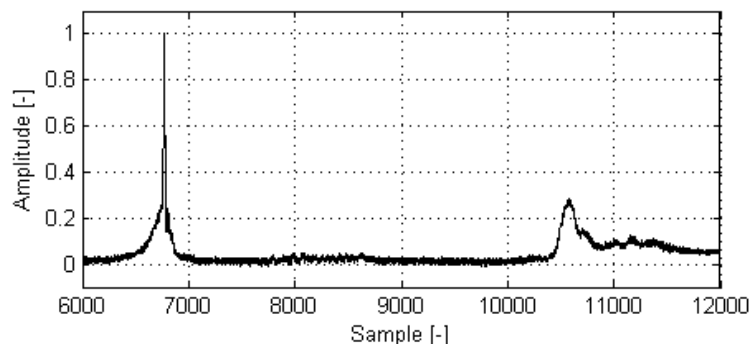
2.1.1 Charakter signálu a nežádoucí jevy

V prvé řadě bude použit vícekanálový záznam. Kanál, který jako první zareaguje na zvýšení úrovně signálu při explozi, bude zdrojem spouštěcího pulzu (Trigger), implementována však bude i možnost použít individuální trigger pro každý kanál. Vzhledem k použití optických vláken lze očekávat velmi nízké úrovně šumu v klidovém stavu, naproti tomu průchod rázové vlny (tím pádem i vznikajícího světelného signálu) pozicí umístění sondy se projeví relativně úzkým impulzem velmi vysoké amplitudy. Úroveň maxim může dosahovat saturační úrovně použitých A/D převodníků, špička pulzu tedy může být oříznuta na maximální hodnoty jejich rozsahu. Situace je znázorněna na obr. 2.1, jenž zobrazuje impuls v signálu získaném z měření prováděného na výbušnině Tetryl.



Obr. 2.1: Příklad saturace špičky impulzu (Tetryl)

Dalším možnou komplikací, jež se může v signálu vyskytnout, jsou parazitní impulzy, vznikající zejména za špičkami hledanými při destrukci optického vlákna. Průběh na obr. 2.2 znázorňuje poměrně výrazný parazitní impuls (přibližně od vzorku 10000) v signálu, měřeném při explozi výbušniny Semtex.



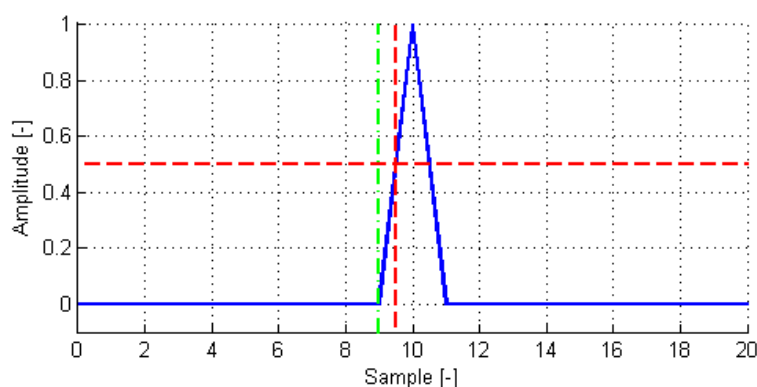
Obr. 2.2: Příklad výskytu parazitních impulzů (Semtex)

2.1.2 Definice pozice pulzu, okamžik detekce

Při zpracovávání signálu je důležité rozhodnout, jak nejlépe definovat pozici impulzu pro správné vyhodnocení naměřených dat. V průběhu návrhu bylo uvažováno několik variant.

Běžnou praxí je např. použití pevně nebo adaptivně vytvořeného prahu, při jehož překročení je zaznamenán příslušný časový okamžik, případně číslo aktuálního vzorku. Nevýhodou metody je nutnost vhodně definovat hodnotu prahu, značná nesourodost tvarů impulzů různých výbušnin nebo např. absence jakékoliv rezistence vůči parazitním impulzům, předcházejícím impulzu hledanému. Na obr. 2.3 je tato metoda naznačena červenými čarami, přičemž vodorovná představuje práh a svislá hledanou hodnotu.

Další možností, je zaznamenat okamžik zlomu v průběhu signálu optických sond (zelená čerchovaná čára). Použití této metody ovšem značně omezuje fakt, že v reálném signálu je zlom podstatně méně ostrý, než ve znázorněném, takřka ideálním případě a jeho pozice je tedy těžko definovatelná.



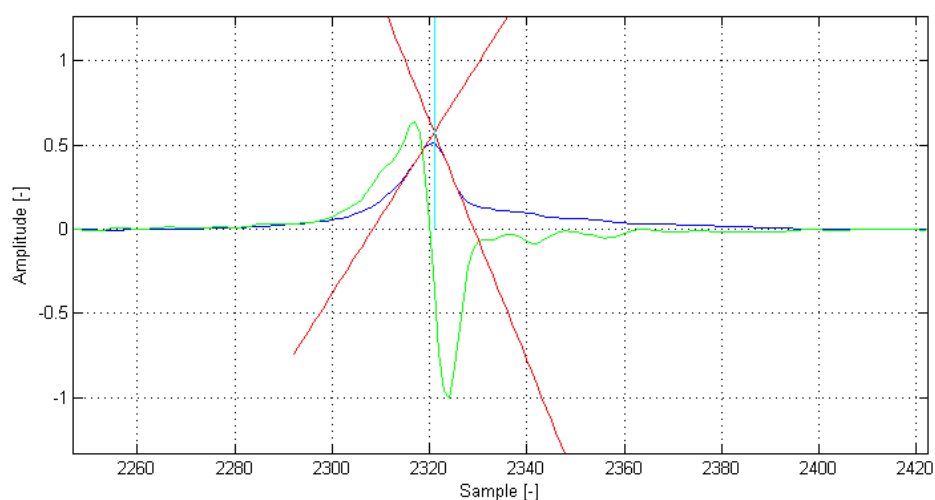
Obr. 2.3: Metody detekce pozice pulzu

Pro model algoritmu byla jako stěžejní zvolena metoda označovaná jako metoda tečen. Základem této metody je výpočet diferenciálu signálu dle vztahu

$$df(x) = \frac{f(x+\Delta x) - f(x)}{\Delta x} \quad (2.1)$$

a následné sestavení tečen, případně vhodných regresních křivek nad okamžiky jeho maxim a minim. Tyto body totiž představují místa nejstrmějšího růstu a poklesu. Průsečík těchto křivek potom označuje pozici pulzu [5].

Mezi nevýhody, lze zařadit např. nespolehlivost při výskytu úzkých špiček v počátcích impulzu, vhodnou filtrací signálu však lze tuto nevýhodu značně omezit. Použití metody je znázorněno na obr. 2.4, jenž znázorňuje reálný signál, měřený při explozi vzorku bleskovice. Modrou barvou je zobrazen filtrovaný vstupní signál, diferenciál znázorňuje zelená barva a červené úsečky jsou tečny, sestavené nad příslušnými body. Bod jejich průsečíku je znázorněn světle modrou barvou. Kompletní matematický princip metody, zejména sestavení křivek a nalezení průsečíku je popsán v následujících podkapitolách.



Obr. 2.4: Určení pozice pulzu metodou průsečíků (Bleskovice)

2.2 Vstupní data pro návrh modelu

Podkladem pro vytvoření algoritmu byla data, naměřená ve spolupráci s Fakultou chemicko-technologickou Univerzity Pardubice.

2.2.1 Parametry vstupních dat

Záznam dat byl proveden s použitím osciloskopu s parametry, uvedenými v tabulce 2.1, do souborů s příponou *.csv v číselném formátu s pohyblivou řádovou čárkou, neboli **float**. Součástí jsou i přiložené textové soubory, obsahující informace o umístění optických sond, nutné pro výpočet výsledné detonační rychlosti.

Tab. 2.1: Parametry záznamů, dostupných z předchozích měření, 2 varianty

Vzorkovací perioda [ns]	0,8	1,6
Vzorkovací frekvence [GHz]	1,25	0,625
Počet vzorků [-]	125000	62500
Délka záznamu [μs]	100	100
Počet kanálů [-]	3 + Trigger	2 nebo 3 + Trigger

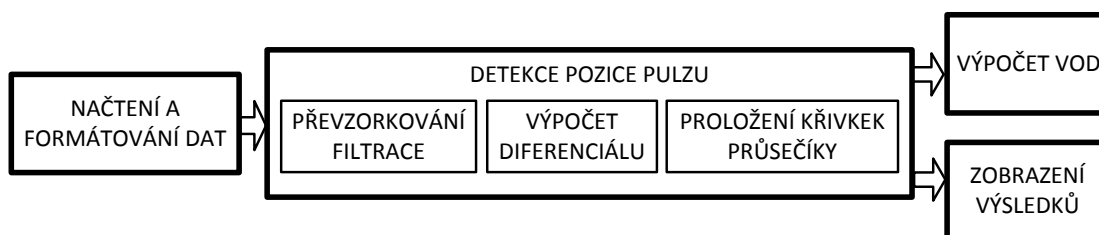
2.2.2 Formát vstupních dat

Původní varianta algoritmu pracuje se vstupními daty přímo bez jakékoliv konverze, což umožňuje snadnější návrh systému a funkcí jednotlivých bloků. Vzhledem k faktu, že ve výsledné implementaci nebude systém pracovat s formátem typu float, nýbrž s celými čísly (příp. zlomkovými s pevnou řádovou čárkou), byla však navržena také druhá varianta algoritmu. Tato varianta využívá konverze vstupních dat i koeficientů filtru k simulaci problémů, nastávajících při práci s omezeným rozsahem čísel typu **Integer** (celých čísel).

Ke konverzi typů jsou použity vestavěné funkce prostředí MATLAB. Konkrétně jsou použity funkce `int16()` a `int32()`, tedy převod na znaménková celá čísla s délkou 16 nebo 32 bitů. Vyšší počet bitů je nutno použít zejména při násobení, kdy je výsledný rozsah dán součtem délek jednotlivých součinitelů.

2.3 Řetězec zpracování, systémový pohled a popis bloků

Na základě poznatků z předchozích podkapitol a s využitím dat, jež byly k dispozici z předchozích měření, byl navržen algoritmus, jehož zjednodušené blokové schéma je zobrazeno na obrázku 2.5.



Obr. 2.5: Řetězec zpracování signálu (systémový pohled)

Toto schéma znázorňuje systémový pohled na řetězec číslicového zpracování výstupního signálu A/D převodníků. Vzhledem k faktu, že se jedná o model systému, vytvořený pro pozdější implementaci v jazyce VHDL, je algoritmus doplněn o některé bloky, které slouží pouze pro potřeby modelu a ve výsledném návrhu se neobjeví.

Základní princip algoritmu je poměrně jednoduchý. Po načtení vstupních dat pomocí bloku **Načtení a formátování dat**, jenž se u modelu výrazně liší od formy určené pro výsledný návrh, přichází na řadu samotný algoritmus detekce pozice hledaného pulzu. Detekce probíhá ve 3 základních krocích. Prvním krokem je tvarování signálu (blok **Převzorkování a filtrace**), čili snaha získat ideální průběh pro pozdější hledání minim a maxim diferenciálu signálu, jež je krokem druhým, znázorněným blokem **Výpočet diferenciálu**. Následně je v případě úspěšné detekce nalezena pozice pulzu v bloku **Proložení křivek, průsečíky**. Finálním krokem je využití získaných hodnot k výpočtu VoD a interpretaci jednotlivých mezivýsledků prostřednictvím grafů.

V následujících podkapitolách je uveden podrobný popis jednotlivých bloků, jež tento systém utváří a jež byly realizovány ve výpočetním prostředí MATLAB. Protože blok Načtení a formátování dat představuje pouze vyčtení naměřených hodnot ze vstupních *.csv souborů, není v textu podrobně rozebírán.

2.3.1 Převzorkování a filtrace signálu A/D převodníků

Po načtení signálu vhodného formátu přichází na řadu blok, jenž jej odpovídajícím způsobem upraví do tvaru, vhodného pro snadnou detekci impulsu.

Osciloskopické záznamy, měřené s parametry zobrazenými v tab. 2.1, jsou vzorkovány s několikanásobně vyšším kmitočtem, než bude použit v cílovém zařízení. První operací, jež je součástí bloku tvarování signálu, je tedy jeho převzorkování, konkrétně decimace. Ta ve zkratce znamená použití pouze každého N -tého vzorku původního signálu, přičemž N je označováno jako **decimační faktor**. V prostředí MATLAB je decimace provedena za použití vestavěné funkce `downsample()`.

Pomyslný blok tvarování signálu je doplněn také o normování vektorů hodnot, jež je prováděno dle vztahu

$$y_{norm} = \frac{y}{\max(y)} \quad (2.2)$$

a jež ulehčuje práci zejména při zobrazování mezivýsledků.

Nedílnou součástí návrhu je výběr vhodného filtru. Vzhledem k charakteru zpracovávaného signálu jsou požadavky na filtr zřejmé. Měl by být schopen eliminovat nežádoucí zvlnění a velmi úzké parazitní špičky v konvertovaném signálu optických sond, zároveň však nesmí příliš měnit samotný průběh. Zmíněné požadavky jsou protichůdné a v praxi je proto nutné volit kompromis.

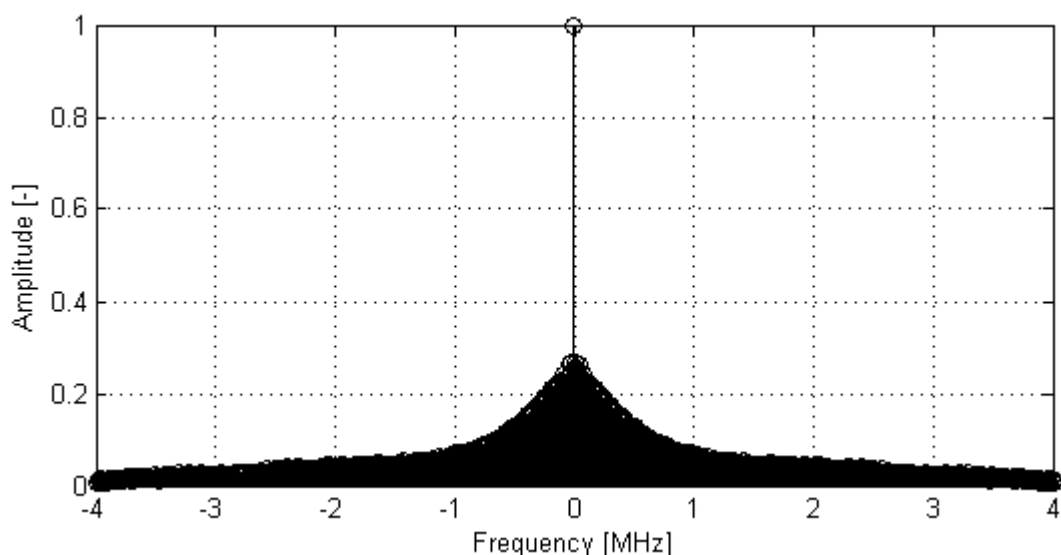
Pro odstranění zvlnění signálu bylo zvoleno použití filtru s konečnou impulzovou odezvou, tedy **FIR (Finite Impulse Response)**. FIR filtry jsou lineárními časově invariantními systémy, jež nemají obdoby v analogovém provedení. Vyznačují se zejména poměrně jednoduchou implementací, absolutní stabilitou a při vhodném návrhu mohou mít lineární fázovou charakteristiku. Pro dosažení požadované strmosti je však často nutné použít FIR filtr velmi vysokého řádu, což může u systémů citlivých na zpoždění signálu znamenat problém.

V praxi jsou FIR filtry charakterizovány vektorem koeficientů, jež představují jejich odezvu na jednotkový impuls. V případě, že se jedná o přímou realizaci, je výstup filtru definován jako konvoluční suma těchto koeficientů se vstupním signálem, daná vztahem

$$y(n) = x(n) * h(n) = \sum_{k=0}^N h(k) \cdot x(n - k), \quad (2.3)$$

kde y představuje výstupní filtrovaný signál, x vstupní signál, h vektor koeficientů filtru a N udává rád filtru. Ze vztahu (2.3) je také zřejmé, že délka filtru, tedy počet koeficientů, je rovna $N+1$ [6].

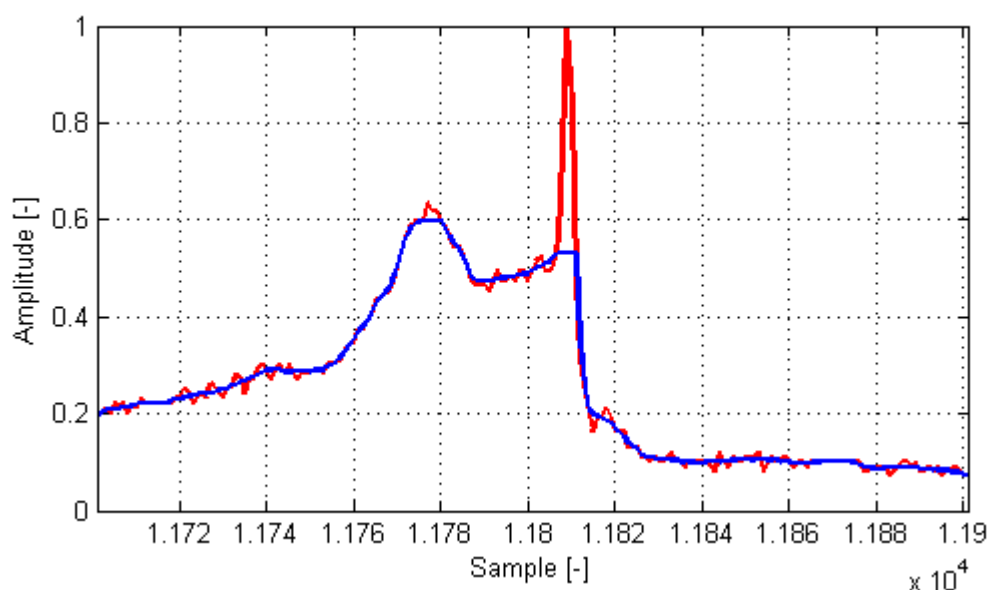
Obr. 2.6 znázorňuje odhad spektra signálu měřeného na výbušnině Politol. Je možno pozorovat, že většina energie je soustředěna na kmitočtech do nízkých jednotek MHz. K usnadnění návrhu byl vytvořen skript `filter_design`, jež umožňuje na vybraném signálu přímo porovnat filtrovaný a nefiltrovaný průběh a jejich spektra. Pro generaci koeficientů byl použit vestavěný nástroj z DSP toolboxu prostředí MATLAB, nazvaný **FDA Tool**.



Obr. 2.6: Odhad spektra vybraného signálu optické sondy (Politol)

Testováním různých variant filtrů na průbězích zpracovávaných signálů bylo ověřeno, že algoritmus nevyžaduje použití filtrů strmé charakteristiky, odstranění zvlnění lze dosáhnout i s FIR implementací velmi nízkého řádu. Ve většině případů byla pozice spolehlivě detekována i bez použití filtrace, ta však i v mírné podobě přispívá ke zlepšení přesnosti.

Druhého požadavku, tedy odstranění úzkých parazitních špiček je možno dosáhnout např. zvýšením řádu filtru, ovšem za cenu značné změny tvarů impulsu, která může mít vliv na výslednou detekovanou pozici. Z tohoto pohledu je vhodnějším řešením například použití takzvaného **mediánového filtru**, jenž využívá vlastnosti mediánu nepodléhat extrémům. Obrázek 2.7 znázorňuje průběh signálu po filtraci kaskádou FIR filtru s charakteristikou dolní propusti čtvrtého řádu a mediánového filtru pracujícího s devíti okolními vzorky.



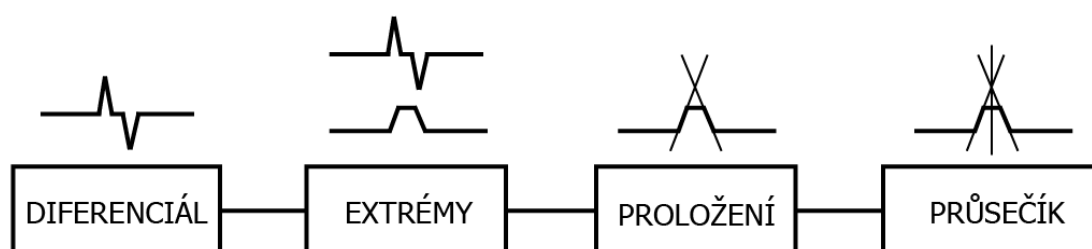
Obr. 2.7: Výsledek filtrace kaskádou FIR a mediánového filtru

Je zjevné, že filtrovaný průběh zobrazený modrou barvou dostatečně přesně kopíruje průběh signálu a parazitní špička, nacházející se u sestupné hrany impulsu, je potlačena díky vlastnosti mediánu nepodléhat extrémům.

Proces filtrace signálu je implementován funkcí `filter_signal()`. Přestože zpoždění vzniklé FIR filtrací nemá vzhledem k podstatě algoritmu vliv na výsledek, je závěr funkce doplněn o jeho kompenzaci. Tato úprava usnadňuje pozdější zobrazení a porovnání signálů. Popis filtrů, použitých pro potřeby modelu, lze nalézt jako součást příloh diplomové práce.

2.3.2 Nalezení okamžiku pulzu, diferenciál

V podkapitole 2.1.2 byl stručně popsán princip detekce impulsu, jenž byl zvolen pro model systému. Operace je jádrem celého algoritmu a následující podkapitola obsahuje detailní popis jednotlivých kroků, znázorněných na obr. 2.8.



Obr. 2.8: Metoda průsečíků, blokové schéma

Jak již bylo zmíněno, prvním krokem zpracování je výpočet **diferenciálu** dle vztahu (2.1) v celé délce vstupního vektoru hodnot. Analýzou naměřených dat bylo pozorováno, že minimum diferenciálu je ve většině případů výraznější než maximum, odpovídající nástupné hraně impulsu. Při jejich hledání je tedy nejprve nalezeno minimum v celém vektoru hodnot, následně je v předcházejícím úseku signálu hledáno příslušné maximum. V případě, že je nalezeno více stejných hodnot extrémů, je uvažována vždy pouze první hodnota.

Nalezení výše uvedených bodů umožňuje vymezit zajímavou část signálu a není již třeba pracovat s celým průběhem. Do dalšího zpracování tedy vstupují pole hodnot od polohy maxima diferenciálu po bod minima, z obou stran rozšířené o offset, který je zaveden kvůli možnosti použít regresní proložení průběhu.

Podstatou **metody průsečíků** je najít průsečík obecně dvou křivek, sestrojených nad okamžiky dříve nalezených extrémů. Významnou vlastností algoritmu by měla být nízká výpočetní náročnost, přestože následné výpočty budou probíhat jako součást **post-processingu**. Pro proložení průběhů tedy byly zvoleny pouze lineární metody.

První metodou je prosté sestrojení **tečen** nad nalezenými body na základě elementárních vztahů (2.4) až (2.6). Velkou výhodou této metody je extrémně nízká výpočetní náročnost. Pokud je směrnice k počítána ze dvou sousedních vzorků, jmenovatel odpovídající tomuto rozdílu ve vztahu (2.5) je roven jedné a výsledek (diferenciál) je tak dán pouze rozdílu odpovídajících hodnot signálu.

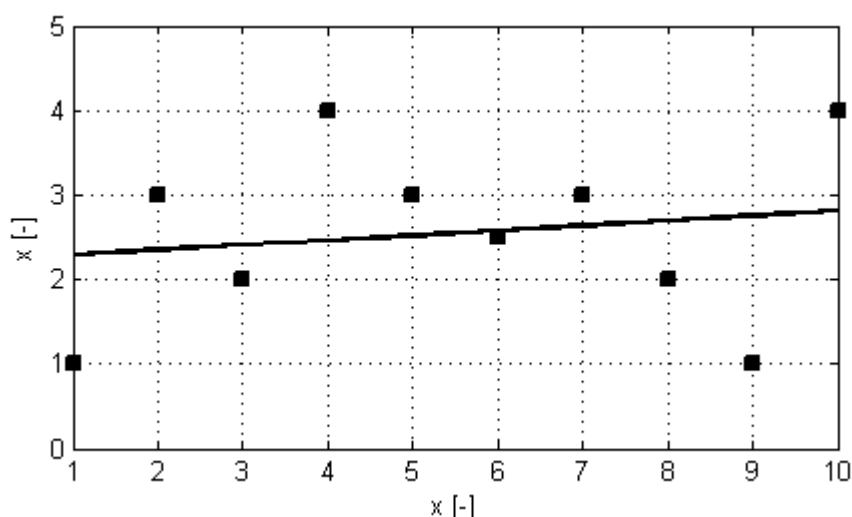
$$y(x) = k \cdot x + q \quad (2.4)$$

$$k = \frac{y(x+\Delta x) - y(x)}{\Delta x} \quad (2.5)$$

$$q = y(x) - k \cdot x \quad (2.6)$$

Tato metoda se i přes svoji jednoduchost vyznačuje dostatečnou spolehlivostí a to zejména v případech intenzivnější filtrace. Upravený signál již nevykazuje extrémně prudké změny a nehrozí tedy např. chybná orientace tečen. Pro méně filtrované signály však nemusí tečna sledovat hranu naprosto ideálně, v takovém případě lze za cenu vyšší výpočetní náročnosti použít proložení více body metodou **jednoduché lineární regrese**.

Metoda jednoduché lineární regrese je statistická metoda, jež k volbě regresní přímky využívá minimalizace tzv. reziduálního součtu čtverců, přičemž reziduum představuje vertikální vzdálenost jednotlivých bodů od hledané křivky. Obr. 2.9 zobrazuje příklad proložení deseti bodů.



Obr. 2.9: Ukázka proložení sekvence metodou jednoduché lineární regrese

V algoritmu je jednoduchá lineární regrese implementována na základě následujících vztahů [7]

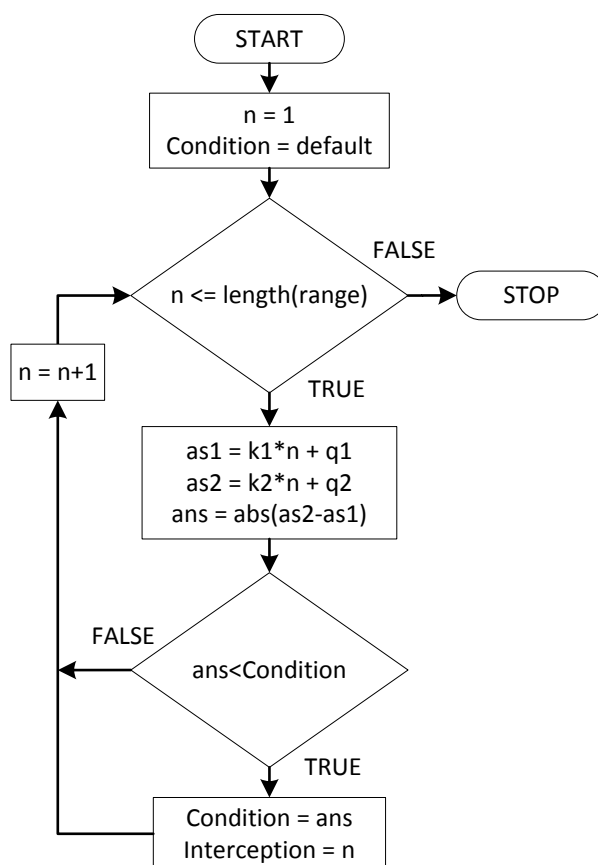
$$k = \frac{\sum_{i=1}^n (x_i \cdot y_i) - \bar{y} \cdot \sum_{i=1}^n (x_i)}{\sum_{i=1}^n (x_i^2) - \bar{x} \cdot \sum_{i=1}^n (x_i)} \quad (2.7)$$

$$q = \bar{y} - k \cdot \bar{x} \quad (2.8)$$

Jedná se tedy opět o vztahy, jež umožní vyčíslit směrnici přímky k a její průsečík s vertikální osou q . Je zřejmé, že tato metoda je výrazně výpočetně náročnější než prosté použití diferenciálu, protože vyžaduje minimálně jedno dělení (dělení konstantou pro výpočet aritmetických průměrů lze nahradit násobením převrácenou

hodnotou). Vzhledem k poměru vzorkovacího kmitočtu a běžné délky impulzu není vhodné volit pro výpočet více než 4 vzorky, počet násobení proto není příliš vysoký a díky paralelnímu zpracování bude možno jej realizovat poměrně efektivně. Výhoda metody oproti použití diferenciálu jako směrnice spočívá v lepší schopnosti sledovat hrany i při méně filtrovaném signálu.

Po úspěšném sestavení tečen je posledním krokem k určení pozice pulzu nalezení **průsečíku** diskretních přímk. Matematicky lze problém řešit porovnáním rovnic obou tečen. Výsledný vztah však vyžaduje dělení předem neznámým dělitelem, jež není možno optimalizovat např. násobením převrácenou hodnotou a je tedy nutné použití složitých a výpočetně náročných algoritmů. Pro jednodušší a efektivnější implementaci do hardware byl vytvořen jednoduchý algoritmus (Obr. 2.10), jenž hledá nejmenší rozdíl funkčních hodnot křivek. Předpokladem je zavést určitou počáteční podmínku dostatečně vysoké hodnoty.



Obr. 2.10: Algoritmus nalezení průsečíku dvou diskretních přímek

Protože je průsečík hledán pouze v dříve zvoleném úzkém rozmezí od maxima po minimum diferenciálu (případně rozšířeném o zmiňovaný offset), je k němu třeba přičíst hodnotu odpovídající počátku tohoto rozmezí v celém průběhu.

V rámci návrhu v prostředí MATLAB je celý řetězec znázorněný na obr. 2.8 součástí funkce `pulse_position()`. Samotné nalezení tečny, případně regresní přímky, je uskutečněno v rámci funkce `fit_curve()`.

2.3.3 Výpočet VoD

Na základě hodnot získaných výše popsaným zpracováním lze už poměrně snadno vypočítat výslednou detonační rychlost, přičemž v modelu je tento výpočet realizován funkcí `get_velocity()`.

Pro výpočet je využito znalosti relativní vzdálenosti jednotlivých optických sond a získaných pozic pulzů. Na základě těchto hodnot je možno využít elementární vzorec pro výpočet rychlosti pomocí dráhy s a času t (vztah (2.9)) a upravit jej do potřebné formy, kterou vyjadřuje vztah (2.10).

$$v = \frac{s}{t} \quad (2.9)$$

$$VoD = 0,001 \cdot \frac{\Delta dist}{T_{down} \cdot \Delta pos} \quad (2.10)$$

$$T_{down} = \frac{N}{F_s}, \quad (2.11)$$

kde $\Delta dist$ představuje vzdálenost sousedních sond v metrech (konstanta 0,001 představuje převod hodnoty načtené z textového souboru v milimetrech na metry), Δpos rozdíl pozic impulzů v daných kanálech a T_{down} vzorkovací periodu po decimaci vzorkovací frekvence F_s faktorem N .

Alternativní výsledek lze získat např. záměnou pozice impulzu (Δpos) za jiný příznačný okamžik signálu. V modelu je k výpočtu využito také rozdílu poloh maxim a minim diferenciálu a hodnoty takto získané je tedy možno porovnat.

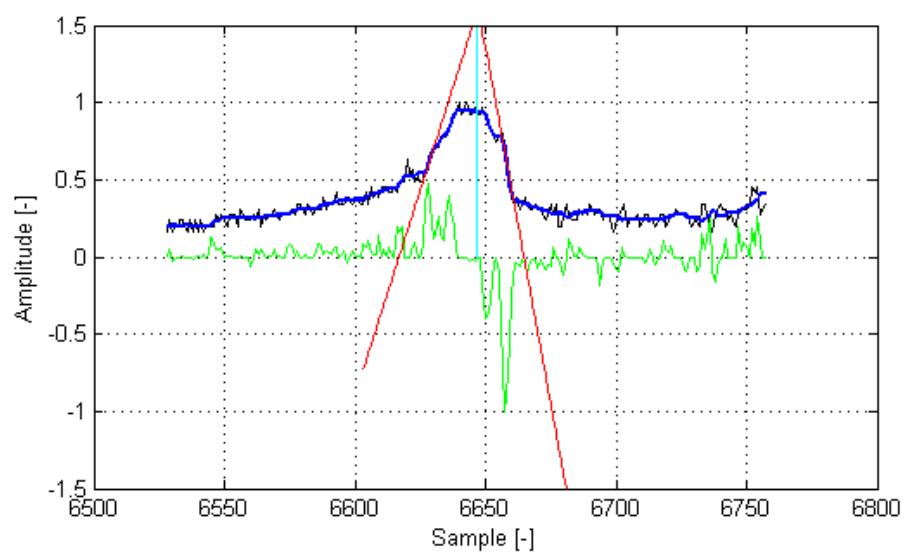
2.3.4 Zobrazení výstupních dat

Vzhledem k faktu, že model slouží k návrhu a demonstraci funkčnosti algoritmu, je kladen důraz na vizuální zobrazení důležitých průběhů.

O zobrazení důležitých dat, jež jsou předávány jako parametry jednotlivých funkcí se stará funkce `display_values()`. Ta obsahuje několik částí. V první řadě jsou zobrazeny průběhy vstupních signálů a to jak ve filtrované, tak nefiltrované podobě, včetně znázornění okamžiků, jež byly označeny za pozice impulzů.

Dalšími zobrazovanými průběhy, jsou diferenciály jednotlivých kanálů, spočítané po aplikaci filtrace. Ty umožňují vytvořit si představu o polohách jeho maxim či minim a jejich vzájemných poměrech.

Poslední a nejdůležitější grafickou interpretací je detailní zobrazení jednotlivých impulzů (před i po filtraci) včetně prvků metody průsečíků, tedy přímků prokládajících náběžné a sestupné hrany a jejich průsečíku. Příklad lze pozorovat na obr. 2.11, kde černá barva znázorňuje původní průběh, modrá jeho filtrovaný ekvivalent, zelenou je vyobrazen průběh diferenciálu v okolí impulzu a červená s azurovou zobrazují zmíněné prvky metody průsečíků.



Obr. 2.11: Detailní zobrazení detekce pozice impulsu metodou průsečíků

3 IMPLEMENTACE SYSTÉMU V OBVODU FPGA

Model, jenž byl popsán v předchozí kapitole, představuje podklad pro implementaci algoritmu detekce do cílového zařízení. V rámci diplomové práce však byl vytvářen kompletní systém, který má několik základních požadavků.

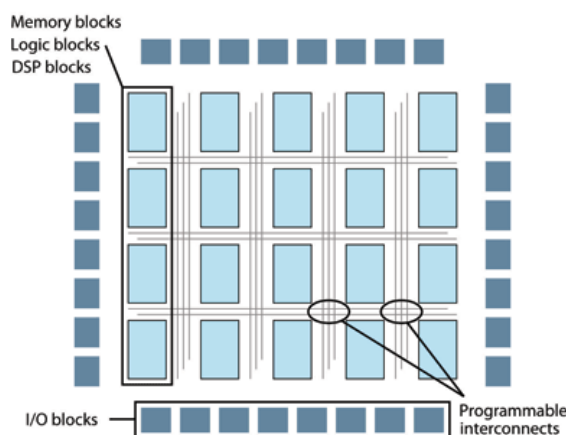
Hlavními z těchto požadavků jsou schopnost detekovat přítomnost impulsu ve vstupním signálu, zaznamenat dostatečné množství dat do vnitřní blokové paměti, detekovat v tomto množství přesné pozice impulsů a v neposlední řadě také umožnit přenos naměřených hodnot do počítače. Implementace tohoto systému je stěžejní částí diplomové práce a je podrobně popsána v kapitole třetí s důrazem na podmínky návrhu, popis navržených komponent a jejich vzájemného propojení.

3.1 Podmínky implementace, použité prostředky

3.1.1 HW prostředky pro zpracování

Jak bylo naznačeno v první kapitole textu, děje, které jsou předmětem analýzy navrhovaného systému, se vyznačují velmi krátkou dobou trvání a je proto nutné použít dostatečně výkonné prostředky pro jejich záznam a zpracování.

Nejčastěji jsou v těchto případech voleny obvody ze skupiny mikroprocesorů, označované jako digitální signálové procesory, tedy DSP (v poslední době také procesory ARM s DSP bloky). Jedná se o instrukčně založené systémy, jež je možno programovat pomocí vyšších programovacích jazyků (např. C) a je tedy možné implementovat poměrně složité algoritmy. Typická je sériová povaha realizace algoritmů (i přes určitou míru paralelizace na úrovni jádra procesoru). Výhodou je zejména relativně nízká cena těchto zařízení.



Obr. 3.1: Obecná struktura FPGA, převzato z [8]

Další nejčastěji volenou skupinou, zejména v náročnějších aplikacích, jsou logická hradlová pole označovaná jako **FPGA (Field Programmable Gate Array)**. Na obr. 3.1 je znázorněna zjednodušená struktura, jejímž základem je pole konfigurovatelných logických bloků (znázorněné světle modrou barvou), obklopené vstupně výstupními piny zařízení. Kromě univerzálních logických bloků se v obvodech objevují také

dedikované komponenty (např. DSP bloky, DCM, bloková paměť RAM a další). Požadovaná funkce zařízení je zajištěna spojením různých funkčních bloků pomocí programovatelných propojů (Programmable Interconnects). Díky paralelní povaze obvodů je možno dosahovat výrazně vyšších výpočetních výkonů při značně nižších pracovních kmitočtech. Obvody se vyznačují velmi rychlou reakcí na změnu vstupního signálu v porovnání s výše zmíněnými signálovými procesory. Typickou aplikací je např. zpracování mnoha paralelních toků dat poměrně vysoké frekvence. Paralelismus však v tomto případě znamená také značně náročnější vývoj složitějších algoritmů.

3.1.2 Použitá zařízení

Z názvu práce je zřejmé, že v projektu byl do role procesoru pro zpracování signálů zvolen právě obvod typu FPGA. Konkrétně byl zvolen model z řady **ARTIX-7** firmy Xilinx ve variantě **XC7A200T**. Volba FPGA se v tomto případě jeví jako logická, vzhledem k faktu, že zařízení bude zpracovávat velké množství paralelních kanálů, jejichž data je v případě detekce spouštěcího impulsu třeba zaznamenat do rychlé paměti. V době psaní diplomové práce, však nebylo zařízení s uvedeným obvodem k dispozici (jeho vývoj není předmětem této diplomové práce) a algoritmus byl tedy navrhován pro vývojovou desku s označením **Xilinx ML505**.

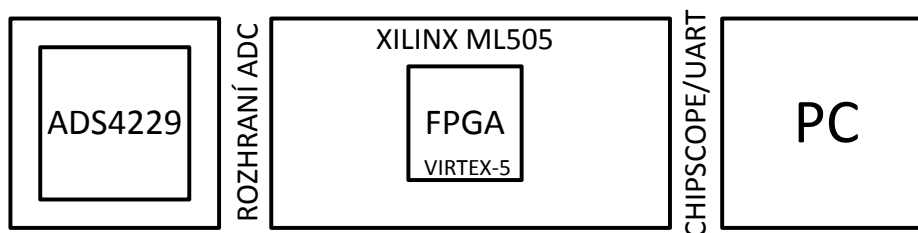
Hlavní součástí vývojové desky je starší obvod FPGA **VIRTEX-5**, osazený ve variantě **XC5VLX50T**, jehož nejdůležitější parametry jsou, spolu s parametry cílového obvodu, shrnuty v tabulce 3.1. Z porovnání parametrů obou čipů je zřejmé, že nová generace nabízí ve všech ohledech značně vyšší počet jednotlivých funkčních bloků, přestože oproti high-endovým čipům typu VIRTEX představuje řada ARTIX low-cost zařízení. Deska kromě čipu FPGA nabízí celou řadu užitečných periférií. Jmenovitě se jedná např. o DDR2 paměťové moduly, LCD displej, uživatelsky dostupné přepínače a LED diody či řada vstupně výstupních portů, které mohou posloužit k přenosu naměřených dat do počítače (např. RS-232, Ethernet a další) [9].

Tab. 3.1: Vybrané parametry použitého FPGA [10], [11]

Zařízení	Logické buňky	Slices	Distribuovaná RAM (Kb)	DSP Slices	Bloková RAM (Kb)	Uživatelské I/O
XC5VLX50T	46080	7200	480	48	2160	480
XC7A200T	215360	33650	2888	740	13140	500

Zdrojem dat pro samotné FPGA je analogově-digitální převodník značky Texas Instruments s označením **ADS4229**. Jedná se o dvoukanálový převodník schopný vzorkovat s frekvencí až **250MSPS** v bitové hloubce až **12 bitů**. Konfigurace probíhá prostřednictvím kombinace sériového rozhraní SPI a paralelních kontrolních pinů a umožňuje nastavení značného množství parametrů a módů. Přestože tvorba rozhraní mezi zmíněným ADC a FPGA není předmětem diplomové práce, je jeho funkčnost stěžejní pro funkčnost celého systému a je o něm tedy krátce pojednáno v jedné z podkapitol zabývajících se popisem jednotlivých komponent [12].

Jednoduché schéma na obr. 3.2 slouží pro lepší představu o zapojení, použitým při vývoji systému.



Obr. 3.2: Jednoduché schéma systému použitého pro implementaci

3.1.3 Odhad nároků na paměť pro záznam

Jedním z rozhodujících faktorů při volbě FPGA bylo dostatečné množství rychlé **blokové** paměti **RAM**. Pro odhad požadavků na paměť je nutno vyjít z několika známých parametrů.

V závislosti na zvolené konfiguraci použitého A/D převodníku a jeho nastavení, bude definován vzorkovací kmitočet a počet bitů na vzorek. V projektu byly zvoleny maximální hodnoty, uvedené v předchozí kapitole a vzorkování tedy probíhá na **12 bitových hladin** (počítá se však i s variantou 14 bitů), při frekvenci **250 MSPS**. Předpokládaná délka záznamu je **100 až 200 μs**, v případě delších záznamů bude signál odpovídajícím způsobem podvzorkován v rámci digitálního zpracování signálu. Posledním potřebným parametrem je počet kanálů, kterých bude ve výsledné podobě použito alespoň **8**. Během návrhu na desce ML505 však bylo počítáno s použitím pouze jednoho převodníku a tedy celkem **dvou** kanálů. Tabulka 3.2 přehledně zobrazuje očekávané nároky na paměť pro různé kombinace těchto parametrů. Pro výpočet byl použit vztah (3.1), u něhož je také uveden příklad výpočtu pro konkrétní hodnoty.

$$X = N_b \cdot M_{ch} \cdot F_s \cdot t_z = 12 \cdot 8 \cdot 250 \cdot 10^6 \cdot 100 \cdot 10^{-6} = \mathbf{2,4\ Mbit}, \quad (3.1)$$

kde N_b značí počet bitů převodníku, M_{ch} počet kanálů, F_s vzorkovací frekvenci a t_z délku záznamu. V tabulce je také uvedeno, jaká část blokové paměti RAM pro danou kombinaci je v konkrétním FPGA využita. Lze pozorovat, že při použití osmi kanálů pro záznam, je vnitřní paměť staršího obvodu VIRTEX nedostatečná.

Tab. 3.2: Tabulka paměťových nároků měřeného záznamu

Parametr	Varianta							
Počet kanálů [-]	2	2	2	2	8	8	8	8
Počet bitů ADC [-]	12	12	14	14	12	12	14	14
Vzorkovací frekvence [MSPS]	250	250	250	250	250	250	250	250
Délka záznamu [μs]	100	200	100	200	100	200	100	200
Celková paměť [Kb]	600	1200	700	1400	2400	4800	2800	5600
Využití paměti (VIRTEX-5) [%]	27,8	55,6	32,4	64,8	Více než 100%			
Využití paměti (ARTIX-7) [%]	4,6	9,1	5,3	10,7	18,3	36,5	21,3	42,6

Předimenzování systému s použitím čipu ARTIX-7 dává do budoucna možnost případného rozšíření počtu kanálů nebo délky záznamu. Je také nutno brát v úvahu, že odhad počítá s pamětí čistě pro záznam vzorků měřeného signálu. Určitou část blokové paměti je nutno vymezit pro FIFO zásobník, zajišťující dostupnost dat před okamžikem detekce, případně pro programovou paměť ROM softwarového procesoru Picoblaze, jež budou rozebrány v některé z následujících podkapitol. Tyto nároky jsou však ve srovnání s vypočtenými zanedbatelné.

3.1.4 Vývojové prostředí

Značnou výhodou plynoucí z použití obvodu FPGA jsou široké možnosti paralelního programování s využitím jazyka VHDL.

Tradičním vývojovým prostředím pro logické obvody firmy Xilinx je již dlouhou dobu komplexní nástroj nazývaný **ISE Design Suite**. Toto prostředí umožňuje provést kompletní návrh algoritmu, včetně jeho syntézy, produkující logické RTL schéma, až po mapování jednotlivých pinů a flash generovaného bitstream souboru do cílového zařízení, například prostřednictvím rozhraní JTAG. Samozřejmostí je přítomnost simulátoru iSIM, jenž je nezbytným prostředkem pro efektivní návrh a verifikaci funkce algoritmů. Vzhledem k použití čipu VIRTEX-5 bylo toto prostředí zvoleno pro tvorbu diplomové práce ve variantě **WebPack**.

Použití FPGA modelové řady ARTIX umožňuje využít také nový vývojový nástroj s označením **Xilinx Vivado**, jenž využívá modernějších algoritmů k urychlení procesu syntézy a je více soustředěn na používání předpřipravených IP bloků. Značnou výhodou prostředí Vivado je nové simulační prostředí xSIM, které umožňuje např. velmi užitečné zobrazení signálů v analogové podobě.

Systémy umožňují poměrně jednoduchou vzájemnou migraci. Při přechodu, stačí přidat původní kód do nového projektu a znovu vygenerovat IP bloky použitých komponent. Je také nutné použít nový formát Constraints File, tedy souboru, kde jsou definovány časové parametry a parametry, týkající se propojení designu s vstupně výstupními piny FPGA.

3.1.5 Parametrizace návrhu, soubory konstant

Důležitou vlastností správně navržených digitálních systémů je možnost jejich opakovaného použití (**design reuse**), tedy vlastnost, zajišťující minimální množství potřebných úprav kódu při změně některého z výchozích parametrů. Příkladem může být např. změna bitové hloubky vstupního slova, změna délky záznamu nebo dalších parametrů uvedených v tabulce 3.2.

Stejně jako v jiných programovacích jazycích, je i v jazyce VHDL možno tuto vlastnost zajistit s využitím parametrizace konstantních hodnot. Konkrétním prostředkem, jenž byl využit při vývoji diplomové práce, jsou soubory nazývané **packages**, jež je možno přiřadit na začátek každého zdrojového kódu.

Příkladem balíčku v návrhu je soubor **param_pkg.vhd**, kde se objevují deklarace nejrůznějších konstant. Konstantou je definovaná šířka vstupního slova, šířka adresního prostoru a mnoho dalších hodnot používaných napříč celým návrhem. Při deklaracích odpovídajících signálů je pak místo pevné hodnoty zvolen alias této konstanty.

Díky parametrickému přístupu znamená pro návrh změna většiny vstupních parametrů pouze nutnost upravit hodnoty konstant v balíčku a opětovné vygenerování IP komponent (např. IP blokové paměti RAM), kde není možno definovat tyto hodnoty parametricky.

3.1.6 Speciální konstrukce jazyka VHDL, Record Types

Častým problémem rozsáhlejších digitálních návrhů je velké množství obecně proměnných v systému. V jazyce VHDL hovoříme zejména o vstupních a výstupních portech komponent a vnitřních signálech. Obzvláště při strukturálním popisu znamená přidání či odebrání portu komponenty zdlouhavou úpravu v podobě editace kódu na několika různých místech. Řešením může být použití speciální konstrukce jazyka VHDL označované jako kompozitní typy, v tomto případě konkrétně **Record Types**.

Record types jsou typy podobné strukturám známým z vyšších programovacích jazyků. Umožňují sdružovat skupiny signálů, jak je naznačeno v následující deklaraci typu. Při kombinaci různých typů je však nutno dbát na zachování syntetizovatelnosti kódu.

```
type agu_out_t is
  record
    ADDR          : std_logic_vector (ADDR_WIDTH-1 downto 0);
    ADDR_END      : std_logic;
    DV            : std_logic;
  end record;

constant AGU_OUT_INIT : agu_out_t := ((others => '0'), '0', '0');
```

Zobrazený příklad představuje deklaraci typu pro výstupní porty komponenty adresního generátoru (Address Generation Unit). Z kódu je patrné, že použití Records umožňuje vícenásobné použití běžných názvů signálů, bez nutnosti identifikace příslušné komponenty (např. signál DV). Rozlišení tedy probíhá na základě názvu celého záznamu. U signálu ADDR lze také pozorovat parametrizaci délky vektoru zmiňovanou v předchozí podkapitole. Současně s deklarací příslušného typu je vhodné vytvořit také inicializační konstantu, kterou lze využít následujícím způsobem při inicializaci odpovídajícího signálu

```
signal agu_tang_out : agu_out_t := AGU_OUT_INIT;
```

Records také významnou mírou přispívají k zprehlednění strukturálního kódu při přiřazení vnitřních signálů k portům komponenty. Následující ukázka kódu, představuje instanciaci komponenty adresního generátoru. Protože vstupní port bloku addr_gen s názvem **AGU_IN** představuje stejný datový typ, jako v předchozích příkladech popsáný signál **agu_tang_out**, lze je propojit přímo. Pro přiřazení všech signálů, zahrnutých v signálu typu **agu_out_t** tedy stačí použít pouze jeden řádek kódu. Tato vlastnost je výhodná obzvláště při opakovaném použití dané komponenty.

```

ADDR_TANGENT_GEN : addr_gen
  Port map (
    CLK      => CLK,
    INIT_R   => agu_tang_rst,
    AGU_IN   => agu_tang_in,
    AGU_OUT  => agu_tang_out
  );

```

Pro přístup k hodnotám uloženým v signálech uvnitř záznamu lze využít následující syntaxe

```

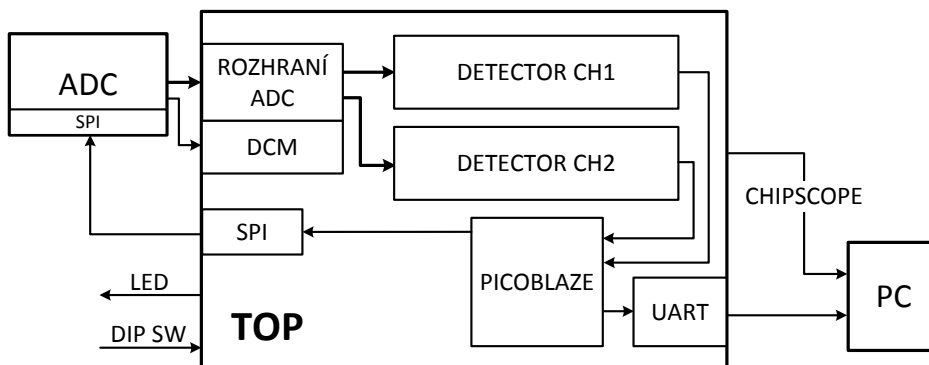
if (agu_tang_out.DV = '1') then

```

Použití Records znamená značnou úsporu potřebných úkonů při změnách na vstupně výstupních rozhraních vytvářených komponent. V kombinaci s použitím balíčků popsaných v předchozí kapitole znamená např. přidání výstupního portu změnu pouze na jednom místě kódu. V diplomové práci jsou všechny záznamy a další uživatelské typy deklarovány v balíčku `types_pkg.vhd`.

3.2 Blokové schéma implementovaného systému

Na obrázku 3.3 je možno pozorovat zjednodušené schéma nejvyšší úrovně vytvořeného systému. Systém se skládá z několika základních součástí, jež budou popsány v následujících podkapitolách.



Obr. 3.3: Zjednodušené blokové schéma nejvyšší úrovně designu

Nejprve bude popsáno rozhraní mezi analogově digitálním převodníkem a zařízením, které je dvojí. Tenkou čarou je ve schématu naznačeno konfigurační, které slouží k zajištění vhodných parametrů vstupních dat, případně řízení chodu převodníků. Tlustá znázorňuje důležitější datové rozhraní. Součástí tohoto rozhraní je také blok DCM (Digital Clock Manager), jenž zajišťuje fázovou a časovou synchronizaci hodinových signálů převodníku a je zdrojem taktovacího signálu pro celé zařízení.

Řetězec zpracování, jenž byl popsán v druhé kapitole textu, je součástí bloků **Detector** a jeho návrh je stěžejní součástí diplomové práce. Kromě samotného zpracování má blok za úkol například generaci triggeru a záznam hodnot do blokové paměti RAM. Blok je také dále dělen na několik hlavních součástí, jež budou rozebrány v nadcházejících podkapitolách.

Dalším blokem ve schématu je softwarový procesor **Picoblaze**, jenž slouží k řízení složitějších sekvenčních procedur. Příkladem je inicializace převodníků pomocí rozhraní SPI nebo řízení přenosu dat z blokové RAM do počítače. V testovacím návrhu byl pro analýzu dat použit nástroj **Chipscope**, umožňující sledovat vybrané signály za použití určité částí blokové paměti zařízení. Pro přenos dat bylo zvoleno rozhraní **UART** (standard RS-232), jež však ve finálním návrhu může být nahrazeno rozhraním, s podstatně vyšším datovým tokem.

Za zmínku stojí také použití prostředků vývojové desky pro zajištění uživatelské interakce při testování vytvořeného kódu. Kromě zobrazených LED a DIP přepínačů je k dispozici také ovladač LCD displeje.

3.3 Rozhraní ADC ADS4229

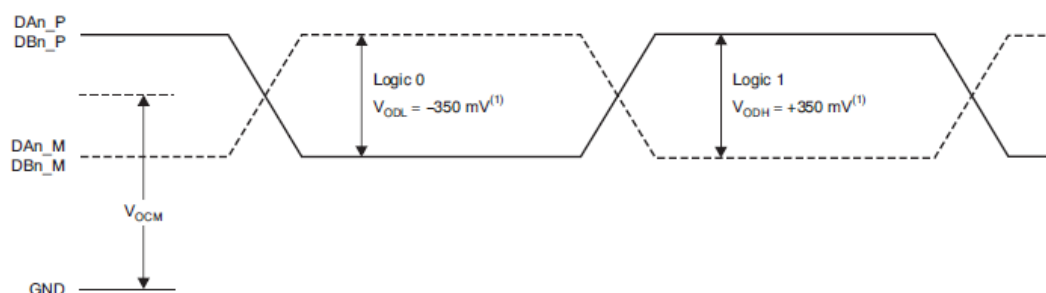
Pro správnou detekci požadovaných hodnot v naměřeném průběhu je zcela zásadní, aby byla přijímaná data interpretována vhodným způsobem. Do role A/D převodníku byl zvolen čip firmy Texas Instruments ADS4229.

3.3.1 Režim přenosu, LVDS

Důležitým krokem při návrhu komunikace mezi převodníkem a zařízením je volba přenosového režimu. ADS4229 umožňuje použít dva režimy přenosu dat.

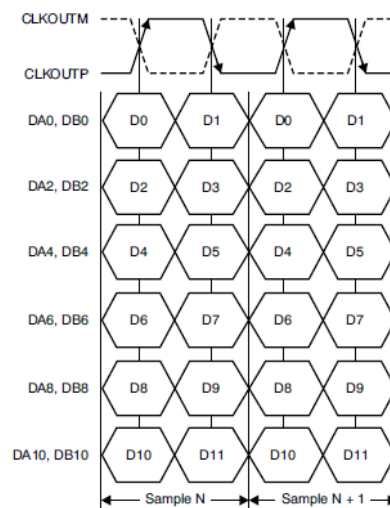
Prvním je režim CMOS, jenž představuje SDR (Single Data Rate) variantu, tedy data jsou vysílána pouze se sestupnou hranou hodinového signálu po celých bitových slovech a každému bitu připadá jeden výstupní pin. Jedná se o Single-Ended standard, výhodný obzvláště díky poměrně jednoduchosti rozhraní. Kvůli nízké odolnosti vůči šumu je však nutno použít poměrně vysoké napěťové úrovně a rychlosti přechodu mezi stavy značně limitují mezní kmitočty. Řešením je použití diferenčních standardů.

LVDS (Low Voltage Differential Signaling) představuje diferenční standard, jenž byl představen v roce 1994 a je dnes používán v rozličných oblastech výpočetní techniky. Princip spočívá v použití dvou komplementárních signálů (obrázek 3.4), přenášených párem vodičů, přičemž hledaná hodnota je dána rozdílem jejich napěťových úrovní. Diferenční povaha přenosu zajišťuje výrazně vyšší odolnost vůči šumu a interferencím. Je tedy možno použít menší rozkmit napětí a tím dosáhnout spolehlivého přenosu i na vyšších kmitočtech (řádově až jednotky GHz). Vzhledem k faktu, že realizovaný systém pracuje s vzorkovacím kmitočtem až 250MSPS, byla zvolena právě varianta LVDS.



Obr. 3.4: Příklad komplementárních signálů LVDS s offsetem V_{OCM} , převzato z [12]

Z principu LVDS je zřejmé, že použití diferenčních párů znamená dvojnásobné nároky na počet pinů pro přenos. V případě převodníku ADS4229 je situace řešena pomocí technologie **DDR** (Dual Data Rate), která umožňuje přenášet užitečná data dvojnásobnou rychlostí, za použití vzestupné i sestupné hrany hodinového signálu. Na obrázku 3.5 lze pozorovat, že pro přenos kompletního vzorku stačí opět pouze 12 pinů, tentokrát však využitých jako 6 párů a s příslušnou hranou hodinového signálu dojde k přenosu sudých nebo lichých bitů slova [12].



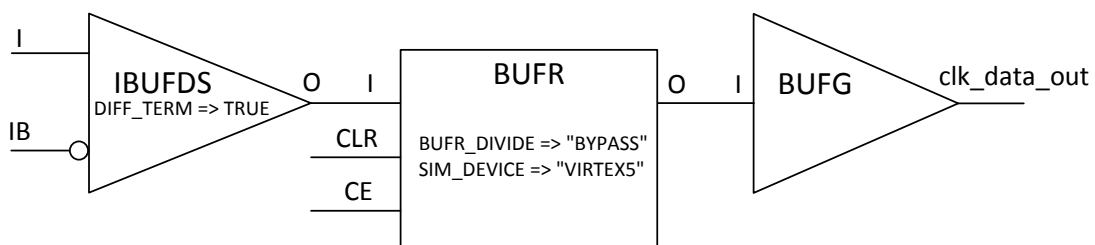
Obr. 3.5: Využití technologie DDR pro přenos vzorků signálu, převzato z [12]

3.3.2 ADC jako zdroj hodinového signálu

Systém, jenž je v rámci práce vytvářen, lze zařadit do skupiny digitálních systémů, označovaných jako **Source Synchronous Systems**. Z toho vyplývá, že kromě datových signálů je převodník také zdrojem diferenčního hodinového signálu.

Vzhledem k použití režimu LVDS je hodinový signál z převodníku přenášen pomocí diferenčního páru, v návrhu označeného jako **CLK_ADC_P** a **CLK_ADC_N**.

Obr. 3.6 zobrazuje řetězec, který slouží pro převod diferenčního hodinového signálu a jeho vhodnou úpravu pro použití v FPGA.



Obr. 3.6: Řetězec konverze diferenčního hodinového signálu

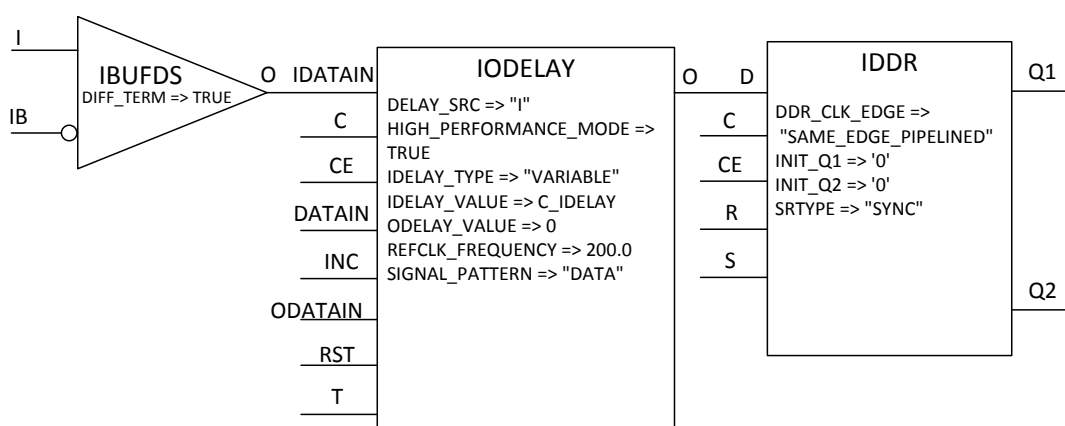
Prvním blokem je buffer **IBUFDS**, umožňující převod LVDS signálu na vstupech I a komplementárním IB na single-ended signál. Generická hodnota DIFF_TERM je použita pro nastavení varianty zakončení diferenciálního páru vodičů.

Po konverzi je signál přiveden do regionálního bufferu pro hodinový signál (**BUFR**), protože diferenční hodinový signál převodníku je připojen na piny FPGA, označené jako CC (Clock Capable). Tyto piny neumožňují přímé připojení do bufferu pro globální rozvod hodinového signálu. Generická hodnota SIM_DEVICE definuje typ použitého hradlového pole. Výsledný signál není třeba dělit, je tedy proveden bypass dělicí logiky (nastavení generické hodnoty BUFR_DIVIDE) [13].

Výstup bloku BUFG je tedy výsledným hodinovým signálem o frekvenci 250MHz, jenž je v kódu označen jako **clk_data_out**.

3.3.3 Datové rozhraní

Při zpracování datových signálů je situace podobná jako v případě hodinového signálu. Situaci znázorňuje blokové schéma na obr. 3.7.



Obr. 3.7: Řetězec konverze datového DDR signálu

Prvním krokem je opět převod diferenčního signálu na single-ended pomocí bloku **IBUFDS**. Kvůli zajištění ideálního okamžiku vzorkování signálu je na výstup zařazen blok **IDELAY**, jenž představuje konfigurovatelné zpoždění signálu. Blok má opět řadu generických parametrů.

Pomocí parametru **DELAY_SRC** lze zvolit, který z portů bude použit jako vstup bloku (na obr. 3.7 je to port **IDATAIN**). **HIGH_PERFORMANCE_MODE** umožňuje snížit jitter (nežádoucí odchylka od periodicity) za cenu mírně vyšší spotřeby. **IDELAY_TYPE** slouží k rozlišení, jestli je zpoždění fixní nebo variabilní. Dvojice **IDELAY_VALUE** a **ODELAY_VALUE** slouží k definici hodnoty zpoždění (nastavitelné v krocích od 0 po 63). Zbývající parametry představují referenční kmitočet pro blok (z externího zdroje) a informaci o tom, zda se jedná o datový nebo hodinový signál.

Posledním krokem k převodu DDR signálu do SDR podoby je použití bloku **IDDR**. Režim **SAME_EDGE_PIPELINED** pomocí přídatného registru zajišťuje, že na výstupech **Q1** a **Q2** jsou k dispozici příslušné páry sudých a lichých bitů (viz. obr. 3.5) v okamžiku stejné aktivní hrany [13].

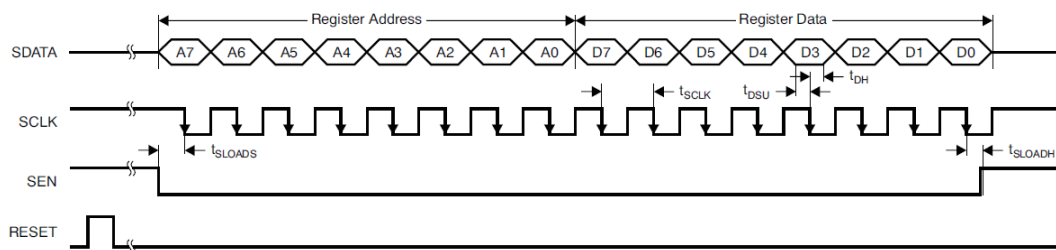
Výsledný signál představuje souvislý tok vzorků o délce 12 bitů, přenášený s frekvencí 250MHz pro každý kanál.

3.3.4 Konfigurační rozhraní

Pro konfiguraci převodníku do požadovaného režimu byla výrobcem implementována dvě základní rozhraní, paralelní a sériové.

Paralelní rozhraní představují piny, v katalogovém listu označené jako CTRL1 až CTRL3, v kombinaci s definovaným napětím na pinech SEN a SCLK. Různými konfiguracemi úrovní lze prostřednictvím tohoto rozhraní nastavovat např. režim přenosu, číselný formát výstupu nebo režim napájení. Pro aktivaci módu paralelní konfigurace je třeba držet pin RESET ve vysoké hodnotě.

Plnou kontrolu nad parametry zařízení však nabízí sériové rozhraní SPI, reprezentované piny SCLK, SEN (aktivní v nízké úrovni), SDATA, SDOUT a značným množstvím vnitřních registrů. SCLK představuje hodinový kmitočet, který je generován zdrojem konfiguračních dat (v tomto případě FPGA) a jeho hodnota může nabývat hodnot od nízkých jednotek Hz až k frekvenci 20 MHz. Obrázek 3.8 znázorňuje průběhy signálů rozhraní při odeslání jednoho rámce. Nezobrazený port SDOUT slouží ke čtení obsahu registrů po aktivaci módu READOUT [12].



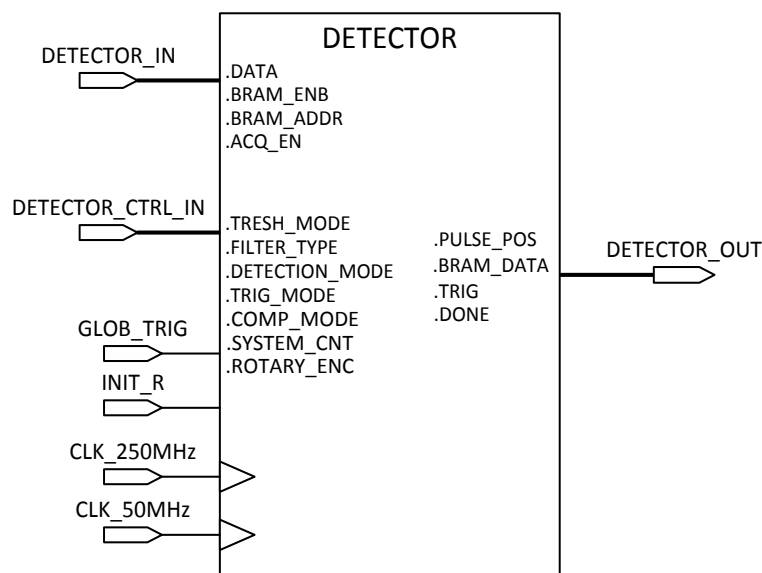
Obr. 3.8: Ukázka průběhů signálu SPI rozhraní ADS4229, převzato z [12]

Rozhraní SPI je v projektu implementováno jako jednoduchý stavový automat, řízený mikroprocesorem Picoblaze. Generace dostatečně pomalého signálu je zajištěna pomocí v určitých intervalech generovaných Clock Enable signálů.

3.4 Blok Detector

Každému kanálu výsledného návrhu přísluší jeden blok Detector. Jedná se o nejdůležitější blok z hlediska záznamu a zpracování signálu. Následující podkapitola se zabývá jeho popisem.

3.4.1 Schéma bloku, vstupy a výstupy



Obr. 3.9: Schematický symbol bloku Detector

Schematický symbol na obr. 3.9 znázorňuje vstupní a výstupní porty bloku Detector. Tlustá černá čára označuje porty kompozitního typu Records, popsaného v podkapitole 3.1.6. Názvy s tečkovou konvencí představují porty, jež jsou v jednotlivých Records zahrnuty.

Chod celého bloku je zajištěn pomocí dvojice hodinových signálů CLK_250MHz a CLK_50MHz. **CLK_250MHz** představuje rychlý hodinový signál, určený pro generaci spouštěcího pulzu a záznam hodnot do blokové paměti. Signál **CLK_50MHz** slouží pro taktování digitálního zpracování zaznamenaných hodnot a snižuje tak časové nároky designu. Jak naznačuje schematická značka, aktivní hranou je hrana vzestupná.

Vstupní port **DETECTOR_IN**, představuje signály, jež jsou unikátní pro každý blok Detector, použitý v návrhu. Port **DATA**, v něm zahrnutý, představuje tok zdrojových dat, vytvořených úpravou signálu DDR analogově digitálních převodníků, jež byla popsána v podkapitole 3.3.3.

Trojice portů **BRAM_ENB**, **BRAM_ADDR** a **BRAM_DATA** poskytuje externí přístup k obsahu blokové paměti RAM, jež je také součástí bloku Detector. Zajištění externího přístupu je nutné např. pro přenos naměřených dat do PC.

Record **DETECTOR_CTRL_IN** představuje sadu řídicích portů, jež jsou společné pro celý design. Tyto porty slouží k nastavení podmínek záznamu a zpracování. Výjimkou je port **SYSTEM_CNT**, jenž představuje systémový čítač, nutný pro určení vzájemné

pozice impulsů v režimu lokálního spouštění. Podrobný popis těchto i dalších dosud nepopsaných portů je uveden v podkapitolách, zabývajících se jednotlivými součástmi bloku Detector.

Přestože reset zařízení bude prováděn na úrovni celého obvodu FPGA, je součástí designu také port **INIT_R**, umožňující inicializaci zařízení na defaultní hodnoty a opětovné zahájení procesu detekce.

3.4.2 Pracovní režimy bloku

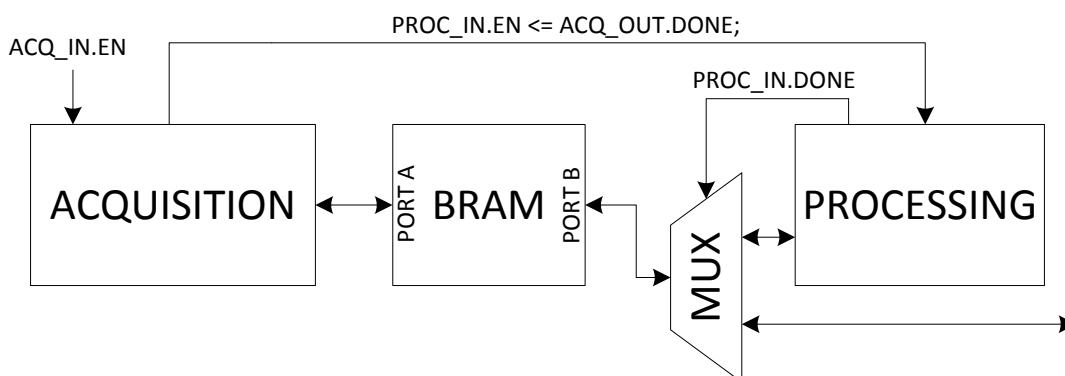
Samotný Detector je popsán převážně strukturálním kódem. Přestože blok není řízen stavovým automatem, lze jeho činnost popsat pomocí určitých pracovních režimů.

Po inicializaci zařízení a aktivaci signálu ACQ_EN (viz. obr. 3.9) je systém v režimu zaznamenávání signálu, řízeném prostřednictvím bloku **Acquisition**, jemuž je věnována podkapitola 3.5. V případě detekce impulsu ve vstupním signálu je spuštěn záznam do blokové paměti RAM až do okamžiku, kdy je uloženo požadované množství vzorků. Po dokončení záznamu je aktivován blok **Processing**, který slouží k detekci přesné pozice impulsu zpracováním naměřených dat, podle zvoleného algoritmu. Zjednodušená struktura bloku je zobrazena na obr. 3.10.

Kromě uvedených bloků je součástí Detectoru také **multiplexor**, jenž slouží k řízení přístupu k portu pro čtení obsahu blokové paměti. K přepnutí dochází v okamžiku dokončení digitálního zpracování signálu.

Poslední součástí je proces, který slouží pro výpočet pozice impulsu na základě hodnoty systémového čítače a pozice detekované v záznamu. Proces je aktivován po dokončení digitálního zpracování.

Aktivace řídicího signálu INIT_R se zařízení navrácí do režimu zaznamenávání.

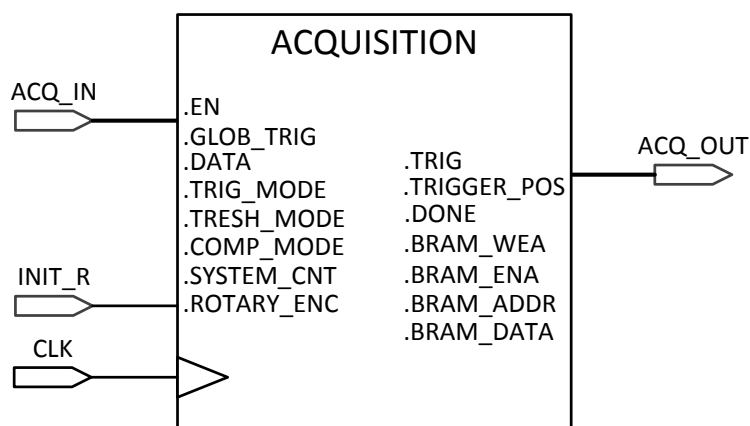


Obr. 3.10: Zjednodušená struktura bloku Detector

3.5 Záznam signálu, blok Acquisition

Jak již bylo několikrát zmíněno, blok Acquisition slouží k detekci impulsu a záznamu požadovaného počtu vzorků. Díky vysokému taktovacímu kmitočtu (250MHz) se jedná o nejnáročnější část systému z hlediska časování.

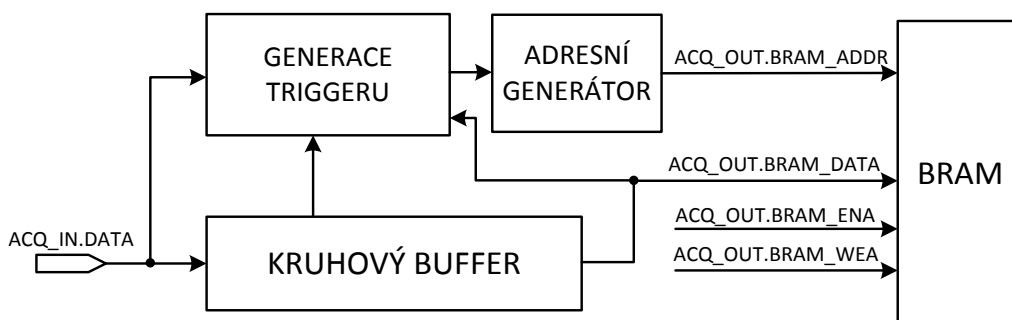
3.5.1 Schéma bloku, vstupy, výstupy a komponenty



Obr. 3.11: Schematický symbol bloku Acquisition

Ze schematického symbolu na obr. 3.11 je zřejmé, že do bloku vstupuje řada řídicích signálů pro nastavení parametrů měření. Hodinový signál CLK s aktivní vzestupnou hranou je propojen s portem CLK_250MHz bloku Detector. Vzhledem k faktu, že z bloku je přistupováno k portu blokové RAM, určenému pro záznam, jsou signály rozhraní paměti doplněny o port BRAM_WEA, jenž umožňuje povolení zápisu.

Zjednodušená vnitřní struktura bloku je společně s blokem BRAM zobrazena na obr. 3.12. Základem je kruhový buffer předem definované délky, zajišťující možnost zaznamenat hodnoty i před příchodem impulsu. Generace triggeru je zajištěna pomocí bloku MEAN_ACQ na základě vstupního a zpožděných vzorků signálu převodníku. Adresní generátor je aktivován v případě přechodu do režimu záznamu a umožňuje řídit přístup k paměťovým pozicím BRAM. Uvedené komponenty jsou podrobněji rozebrány v následujících podkapitolách.



Obr. 3.12: Zjednodušená struktura bloku Acquisition s popisem rozhraní BRAM

3.5.2 Stavový diagram

Na rozdíl od bloku Detector je chod bloku Acquisition řízen pomocí jednoduchého stavového automatu zobrazeného na obr. 3.13.

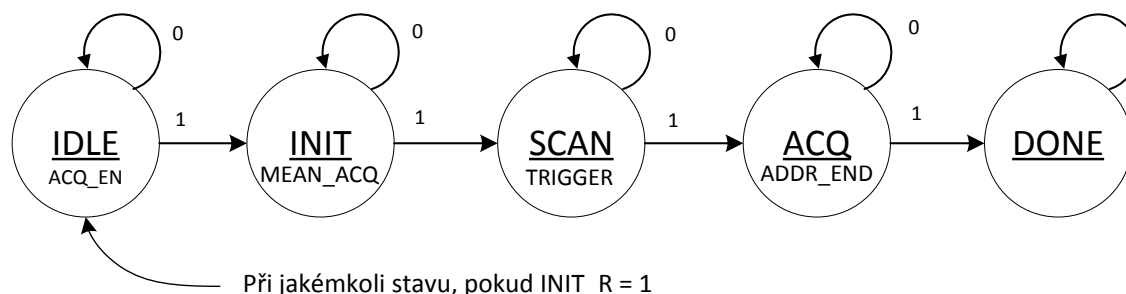
Po startu zařízení se blok nachází ve stavu **IDLE**. Jedná se o klidový stav, ve kterém je kruhový buffer deaktivován a nedochází k detekci spouštěcího impulsu. Při nastavení vysoké úrovně na portu ACQ_EN v bloku Detector dochází k zahájení inicializace a následně spuštění režimu skenování.

Ve stavu inicializace (**INIT**) je proveden výpočet průměru ze vzorků procházejících kruhovým bufferem a případné nastavení hladiny komparace dle aktuálně zvoleného módu (viz. podkapitola 3.5.5). Po dokončení uvedených operací je aktivován řídicí signál bloku MEAN_ACQ a systém přechází do stavu skenování.

Stav **SCAN** představuje komparaci aktuální hodnoty vzorku nebo průměru z několika vstupních hodnot (pro větší odolnost vůči náhodným špičkám) s hodnotou prahu nastavenou v předchozím kroku. V případě překročení prahu je nastavena vysoká úroveň na portu TRIGGER.

Aktivací uvedeného portu systém přechází do stavu **ACQ**. V tomto stavu je aktivován adresní generátor a je proveden záznam hodnot do příslušného bloku paměti. Při dokončení záznamu je generován signál informující o dosažení poslední adresy (ADDR_END) a systém přechází do stavu **DONE**.

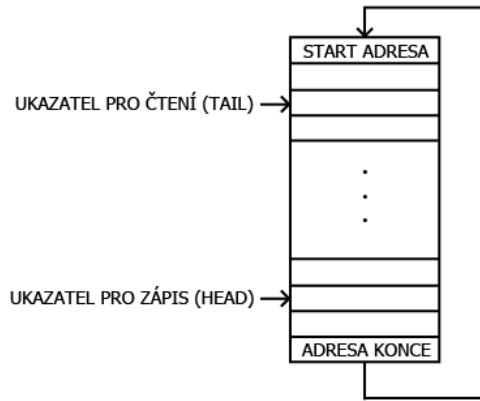
Ve stavu **DONE** je opět deaktivován kruhový buffer a systém v něm setrvává až do příchodu impulsu na portu INIT_R.



Obr. 3.13: Stavový diagram bloku Acquisition

3.5.3 FIFO zásobník, Pre-Trigger okno

Při záznamu signálů optických sond je vhodné, aby byly hodnoty zaznamenány s určitým předstihem a k dispozici tedy byl i signál před pozicí triggeru. V návrhu je tato funkce zajištěna pomocí jednoduchého synchronního FIFO (First In, First Out) zásobníku, jenž je implementován jako kruhový buffer, znázorněný na obr. 3.14. Základní princip spočívá v kruhovém adresování určitého paměťového bloku. V případě, že ukazatel pro čtení nebo pro zápis dosáhne poslední vyhrazené adresy, vrací se na nejnižší pozici.



Obr. 3.14: Princip kruhového zásobníku

Při volbě délky bufferu, je nutno brát v úvahu celkový počet vzorků jednoho kanálu, jenž se pohybuje v řádu deseti tisíců. Výhodné je volit jeho délku, jako mocninu dvou, což umožňuje ušetřit logiku zajišťující kruhovou indexaci (v případě přetečení dojde k vynulování příslušného indexu). V návrhu je délka nastavitelná pomocí parametru `WINDOW_WIDTH`, jenž nepřímo ovlivňuje délku okna dle vztahu

$$WINDOW = 2^{WINDOW_WIDTH} \quad (3.2)$$

přičemž defaultně je nastavena hodnota `WINDOW_WIDTH = 10`, odpovídající délce 1024 vzorků. Celkové nároky na paměť kruhového bufferu jednoho kanálu jsou tedy pro případ délky slova 12 bitů

$$CBUFF_BITS_{TOTAL} = 12 \cdot 1024 = 12288 \text{ bit} \quad (3.3)$$

Buffer je realizován pomocí dvouportové paměti BRAM, jež se skládá z bloků předem definované délky. V Případě čipu VIRTEX-5 se jedná o bloky délky 36kbit, jež je možno v případě potřeby rozdělit na dva bloky o délce 18kbit. Při výpočtech je nutno brát v úvahu, že hodnoty, uvedené v dokumentacích firmy Xilinx, jsou počítány s faktorem 1024 pro převod mezi bity a kilobity, celkový počet bitů v jednom bloku je tedy 18432. Fyzické bloky podporují pouze předdefinované šířky bitového slova (např. 9, 18 a další), tudíž není vždy možno využít paměť kompletně. Pro zvolenou kombinaci délky slova a počtu prvků bufferu je při implementaci kódu použit právě jeden blok velikosti 18kbit. S použitím vztahu (3.4) je možno vyčíslit procentuální využití tohoto bloku. V reálné implementaci pro obvod ARTIX-7 obsahující přes 13mbit paměti je však hodnota, odpovídající nevyužité části (přibližně 6kbit), zanedbatelná [13].

$$CBUFF_USG = 100 \cdot \frac{12288}{18432} \cong 66,6 \% \quad (3.4)$$

Kromě popsaného bufferu, je součástí návrhu i krátký kruhový buffer, zajišťující dostupnost vzorků zpožděných o definovanou hodnotu. Pomocí aktuálního a zpožděného vzorku je pak možno vypočítat krátkodobý průměr (viz. vztah (3.5)), jenž může při komparaci nahradit prostou hodnotu aktuálního vzorku (vyšší odolnost vůči náhlým změnám). Vzhledem k malému počtu vzorků (maximálně 16) je při syntéze

buffer realizován pomocí dedikované paměti RAM, tedy s použitím standardních logických buněk.

$$MEAN = \frac{SUM - SHORT_BUFF_{OLD} + SHORT_BUFF_{NEW}}{SHORT_BUFFER_LENGTH} \quad (3.5)$$

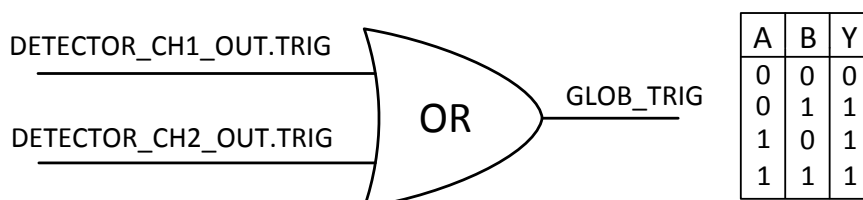
SUM představuje krátkodobou sumu, $SHORT_BUFF_{OLD}$ představuje nejstarší vzorek v bufferu (jeho výstup), $SHORT_BUFF_{NEW}$ aktuální vzorek a $SHORT_BUFFER_LENGTH$ jeho délku.

3.5.4 Režimy triggeru

Důležitou funkcí bloku Acquisition, konkrétně funkcí komponenty **MEAN_ACQ**, je generace spouštěcího pulzu (Trigger). Protože každému kanálu přísluší vlastní blok Acquisition, je možno implementovat dva základní režimy triggerování. Jedná se o režim lokální a globální. Pro volbu mezi režimy, lze použít společný kontrolní signál **detector_ctrl_in.TRIG_MODE** v bloku TOP.

Oba zmíněné způsoby vycházejí z použití signálu TRIG, jenž je nastaven do vysoké úrovně na základě porovnání hodnoty vzorku (případně průměru z několika posledních hodnot) a zvoleného prahu.

V režimu **globálního triggeru** je záznam zahájen při detekci aktivní úrovně signálu GLOB_TRIG. Signál je generován na základě lokálních triggerů všech použitých kanálů v modulu TOP. Princip generace signálu GLOB_TRIG s použitím logického bloku OR je naznačen na obr. 3.15.



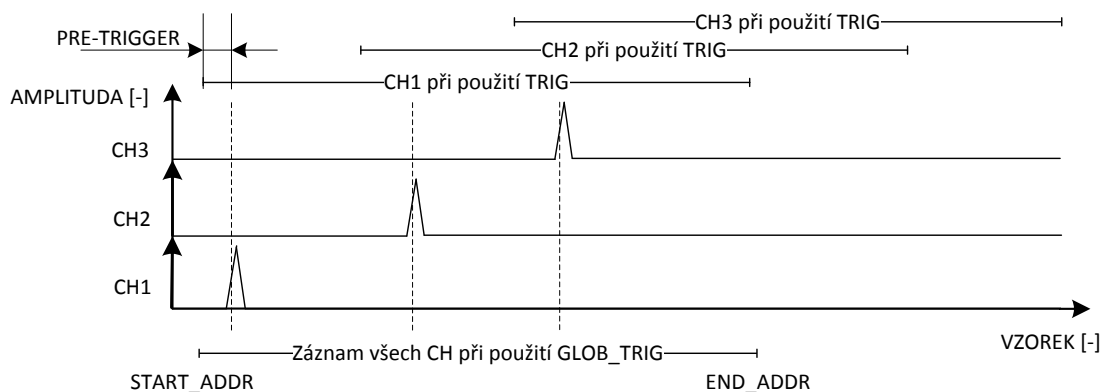
Obr. 3.15: Princip generace globálního triggeru, včetně tabulky bloku OR

Druhá varianta, tedy režim **lokální**, využívá příslušný signál TRIG a zahájení záznamu v jednotlivých blocích Detector tedy probíhá nezávisle na ostatních kanálech. Aby bylo možno definovat vzájemné pozice nezávislých spouštěcích impulzů, je při jejich detekci uložena také hodnota systémového čítače, jenž je spuštěn spolu s prvním aktivním triggerem. Uložení a předání této hodnoty blokům vyšší úrovně je provedeno prostřednictvím signálu TRIGGER_POS (viz. obr. 3.11).

Pomocí lokální metody je možno lépe využít vyhrazenou paměť, obzvláště při použití velkého množství kanálů. Situace je naznačena na obr. 3.16. Čárkovane jsou naznačeny **lokální trigger** jednotlivých kanálů. **Pre-trigger** představuje úsek, předcházející spouštěcím impulzům, zaznamenaný díky použití kruhového zásobníku. V případě použití **globálního triggeru** je záznam všech kanálů spuštěn s příchodem prvního detekovaného impulzu. Je zřejmé, že s rostoucí vzdáleností impulzů od okamžiku triggeru klesá délka užitečné části záznamu z hlediska pozdějšího zpracování (za užitečný lze považovat úsek začínající pre-triggerem daného kanálu), kvůli pevně

dané velikosti blokové paměti. Naproti tomu při použití **lokálních triggerů** daných kanálů je celá paměť jednotlivých kanálů vyhrazena pro užitečný signál.

S použitím lokální metody však také roste riziko, že signál v některém z kanálů nebude zaznamenán (např. příliš nízká úroveň špičky impulsu).



Obr. 3.16: Rozdíl v použití lokálního a globálního triggeru

3.5.5 Volba prahování

Pro záznam požadovaných hodnot do blokové paměti je stěžejní, navrhnout vhodnou metodu určení prahu pro komparaci. Pro testovací účely byly návrhu implementovány tři varianty prahování, mezi kterými je možno volit prostřednictvím řídicího signálu **detector_ctrl_in.TRESH_MODE**.

Tabulka 3.3 obsahuje popis jednotlivých variant. Manuální nastavení hodnoty lze provést pomocí rotačního enkodéru, přičemž aktuální úroveň prahu je zobrazována na LCD displeji desky ML505. Otáčením v obou směrech lze nastavovat hodnoty v rozsahu 0 až 4095, tedy maximum odpovídající šířce slova 12 bitů, jež jsou přenášeny prostřednictvím signálu ROTARY_ENC (viz. obr. 3.9 a 3.11). Zbývající dvě varianty nastaví práh na základě průměru, počítaného nad obsahem dlouhého kruhového bufferu, podle uvedeného vztahu.

Tab. 3.3: Varianty nastavení prahu

TRESH_MODE	TRESHOLD
"00"	Hodnota rotačního enkodéru
"01"	$ROZSAH - (ROZSAH - PRŮMĚR_{WINDOW})/4$
"10"	$ROZSAH - (ROZSAH - PRŮMĚR_{WINDOW})/2$
"11"	Rezervováno

Samotné prahování probíhá prostou komparací hodnoty signálu TRESHOLD, s hodnotou aktuálního vzorku nebo krátkého průměru, v závislosti na hodnotě řídicího signálu **COMP_MODE**.

3.6 Použití BRAM IP Core

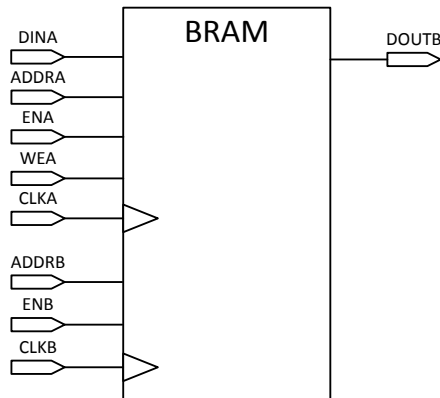
Součástí každého bloku Detector je paměť BRAM předem zvolené velikosti, určené na základě požadované délky záznamu. Tuto paměť tvoří dedikované bloky, jejichž popisem se zabývá následující podkapitola.

3.6.1 Popis bloku

Základními paměťovými prvky čipu VIRTEX-5 jsou bloky velikosti 36kb (RAMB36). Těchto bloků je v zařízení k dispozici 60, což odpovídá hodnotě 2160kbit paměti, uvedené v tab. 3.1. Bloky RAMB36 je možno dále dělit do dvou nezávislých bloků velikosti 18kb. Paměť obsahuje dva plnohodnotné nezávislé porty a na základě jejich konfigurace ji lze provozovat ve třech základních režimech.

Režimy **Single-Port** a **True Dual-Port** využívají příslušné porty pro zápis i čtení obsahu paměti, přičemž mezi operacemi je přepínáno pomocí vyhrazených signálů. Při použití těchto režimů je třeba dbát na zamezení možnosti vzniku kolize dat (např. pokus o zápis obou portů na stejnou adresu) a definici chování výstupního portu v průběhu zápisu.

Při implementaci systému, však bylo zvoleno použití režimu **Simple Dual-Port**. Tento režim využívá jeden z portů výhradně pro zápis a druhý pro čtení zaznamenaných dat. Na obr. 3.17 je zobrazen symbol bloku BRAM, jenž byl generován jako IP (Intellectual Property) Core firmy Xilinx s názvem **Block RAM Generator** [14].



Obr. 3.17: Schematický symbol BRAM v režimu Simple Dual-Port

Z popisu vstupních a výstupních portů vyplývá, že port A je určen pro zápis. Zdrojem vstupních dat je port **DINA**. K aktivaci portu dochází při nastavení vysoké úrovně na portu **ENA**. Záznam na adresu, definovanou pomocí hodnoty na portu **ADDR A**, je však zahájen až při aktivaci zápisu, nastavením portu **WEA** (Write Enable A).

V případě portu B je situace podobná. Port je však určen pouze pro čtení dat a není tedy třeba využívat WE. Výstupní data jsou dostupná na portu **DOUT B**.

Dvojice signálů **CLKA** a **CLKB** naznačuje, že funkce obou popsaných portů může být asynchronní. V návrhu je port A připojen k rychlému signálu CLK_250MHz a port B k signálu CLK_50MHz, vhodném pro časově náročnější operace (např. výpočty v bloku processing).

Pro zlepšení časových parametrů při čtení blokové paměti, je při generaci příslušného IP Core možno zvolit použití registrů na jejich výstupech. Při zpracování těchto dat je třeba počítat se zpožděním validity signálu.

3.6.2 Použití bloku v návrhu

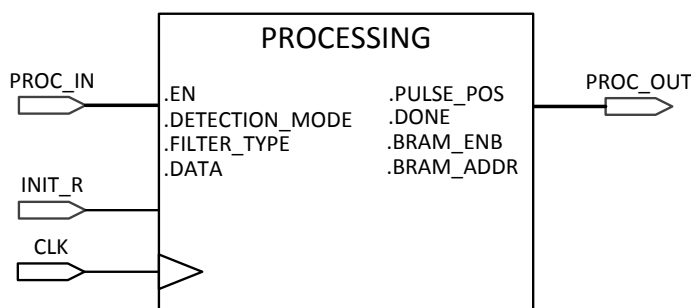
Jak bylo zmíněno, v návrhu jsou bloky paměti umístěny do bloku Detector. Na obr. 3.10 bylo naznačeno, jaké komponenty k jejím portům přistupují, přičemž přístup z různých bloků systému je zajištěn pomocí zobrazeného multiplexoru.

Počet vzorků, jenž má být uložen do paměti byl při generaci IP Core pro testovací návrh zvolen na 26500 (vhodné využití bloků), na základě délky dostupných záznamů uvedené v tab. 2.1, rozšířených o okrajové hodnoty. Při změně této hodnoty je třeba změnit v balíčku `param_pkg.vhd` také parametry **RAM_DEPTH**, jenž slouží pro generaci potřebného rozsahu adres při čtení paměti a **ADDR_WIDTH**, jenž vyjadřuje bitovou šířku adresního vstupu portu.

3.7 Zpracování signálu, blok Processing

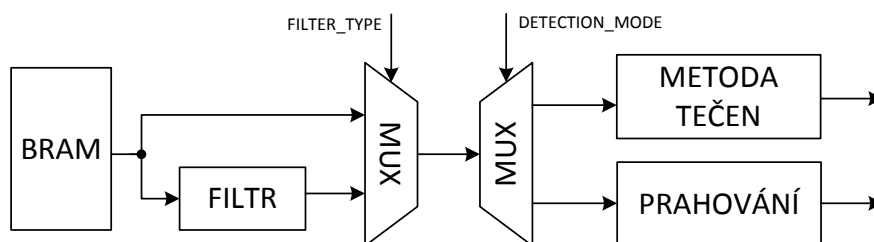
V případě úspěšného provedení záznamu hodnot následuje krok zpracování signálu, jenž spočívá v nalezení přesných vzájemných pozic impulsů v jednotlivých kanálech zařízení. Detekce těchto pozic je stěžejním krokem pro určení rychlosti detonace.

3.7.1 Schéma bloku, vstupy a výstupy a komponenty



Obr. 3.18: Schematický symbol bloku Processing

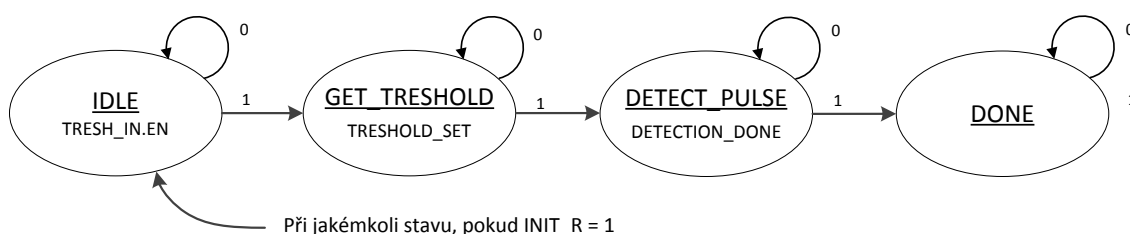
Samotný proces zpracování je aktivován při ukončení záznamu pomocí signálu **EN**, jenž je navázán na výstup **DONE** bloku Acquisition. Trojice signálů **DATA**, **BRAM_ENB** a **BRAM_ADDR** představuje rozhraní pro přístup k zaznamenaným datům. Funkce řídicích signálů **DETECTION_MODE** a **FILTER_TYPE** je naznačena na obr. 3.19. První jmenovaný umožňuje zařadit do datové cesty výstupu paměti BRAM blok, určený pro filtraci měřeného signálu (blok filtrace nebyl v testovacím návrhu implementován). Druhý z řídicích signálů umožňuje zvolit, která metoda detekce impulsu bude při zpracování použita. Výstup **PULSE_POS** představuje výslednou hodnotu detekce, jež je předána do vyšších úrovní návrhu pro další zpracování. Taktovacím kmitočtem pro celý blok zpracování je kmitočet s hodnotou 50MHz připojený na vstup **CLK**, jenž je použit i jako zdroj hodinového signálu pro čtecí port BRAM.



Obr. 3.19: Vnitřní struktura bloku Processing, včetně BRAM

3.7.2 Detekce prahování

První metodou pro přesnou detekci pozice impulsu je použití jednoduchého prahování, jež se v mnohém podobá komparaci, použité při záznamu. Výhodou post-processingu je dostupnost kompletního naměřeného průběhu a tudíž možnost, nastavit práh podle jeho rozsahu.



Obr. 3.20: Stavový diagram detekce pozice impulsu prahováním

Stavový diagram, znázorněný na obr. 3.20 vyjadřuje jednotlivé kroky detekce, prováděné uvnitř bloku TRESHOLD_DET.

Celý proces je zahájen nastavením vysoké úrovně aktivačního signálu **EN**, jenž je propojen s aktivačním signálem bloku Processing.

Ve stavu **GET_TRESHOLD** jsou provedeny dvě základní operace. První je nalezení maxima v zaznamenaném průběhu. Druhou je výpočet aritmetického průměru klidové úrovně záznamu dle vztahu (3.6). Výpočet je prováděn nad počátečními prvky záznamu, jejichž počet je definován parametrem **TH_WINDOW** (defaultní hodnota je 512 vzorků). Aby průměr nebyl příliš ovlivněn vysokými hodnotami v okolí impulsu, je vhodné volit hodnotu **TH_WINDOW** na základě délky kruhového bufferu (např. jako polovinu jeho délky).

$$MEAN = \frac{SUM_{TH_WINDOW}}{TH_WINDOW} \quad (3.6)$$

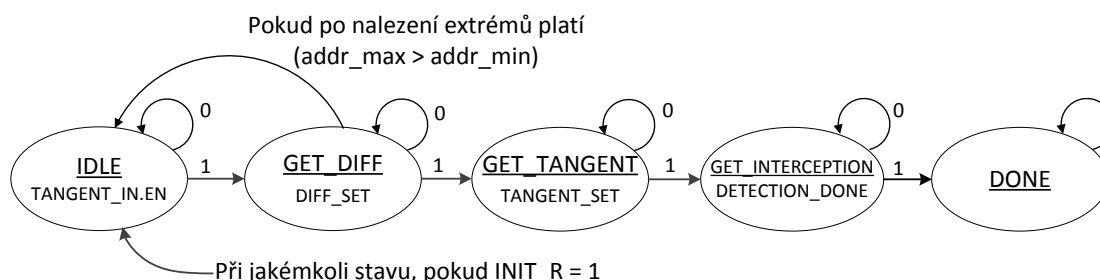
kde *MEAN* představuje průměr a SUM_{TH_WINDOW} sumu prvních *TH_WINDOW* vzorků záznamu. Dělení je v algoritmu implementováno jako bitový posuv o bitovou šířku zvoleného okna a délka tedy musí být mocninou dvou. Před přechodem do stavu **DETECT_PULSE** je ze získaných hodnot nastavena hodnota prahu dle vztahu (3.7), přičemž *TRESHOLD* představuje samotný práh a *MAX* nalezené maximum záznamu.

$$TRESHOLD = MAX - \frac{(MAX - MEAN)}{4} \quad (3.7)$$

Ve stavu DETECT_PULSE je opět procházen obsah paměti a při prvním překročení nastaveného prahu je zaznamenána adresa daného vzorku. Ta představuje jeho nalezenou pozici.

3.7.3 Detekce metodou tečen

Detekce metodou tečen představuje implementaci algoritmu, jenž byl popsán v druhé kapitole textu.



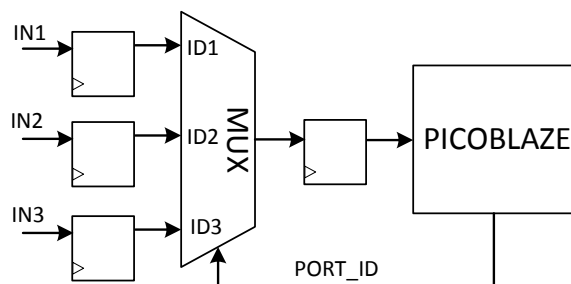
Obr. 3.21: Stavový diagram detekce impulsu metodou tečen

Stavový diagram metody tečen kopíruje jednotlivé kroky zobrazené na blokovém schématu 2.8.

Po aktivaci bloku pomocí signálu EN přechází systém do stavu **GET_DIFF**. V tomto stavu je procházen záznam signálu a ve všech bodech je vypočítána hodnota diferenciálu dle vztahu (2.1). Rozdíl Δx ve jmenovateli lze z výpočtu vynechat, protože pro výpočet jsou použity dva sousední vzorky signálu. V okamžicích **nejvyšší** a **nejnižší** hodnoty **diferenciálu** je třeba zaznamenat hodnoty, jež jsou potřebné pro další výpočty. Jedná se o hodnotu diferenciálu (**směrnice tečny k**), adresu jemu odpovídající (**poloha extrému v ose x**) a hodnotu vzorku v daném bodě (**funkční hodnota y**). Pokud je adresa nalezeného maxima diferenciálu (**addr_max**) vyšší než adresa odpovídající minima (**addr_min**), je třeba celý cyklus opakovat. Maximum je však dále hledáno pouze v úseku od počátečního bodu po adresu minima (viz. podkapitola 2.3.2). Po provedení výpočtu nad všemi hodnotami v paměti je aktivován signál **DIFF_SET** a systém přechází do dalšího stavu.

Ve stavu **GET_TANGENT** nedochází k manipulaci s blokovanou pamětí, ale na základě zaznamenaných hodnot v daných extrémech jsou vypočítány rovnice tečen s využitím vztahu (2.6). Je také vymezen rozsah, ve kterém bude hledán průsečík těchto tečen. Ten je volen jako rozdíl poloh extrémů, z obou stran rozšířených o volitelný offset (defaultně 5 vzorků, nastavitelný prostřednictvím `param_pkg.vhd`).

Po provedení výpočtů přechází systém do finální fáze detekce, tedy do stavu **GET_INTERCEPTION**. Ten je implementací algoritmu, znázorněného na obr. 2.10, jenž umožňuje nalézt průsečík dvou diskretních křivek bez nutnosti dělení. Aby nebylo nutno násobit relativně velkými čísly, tedy adresami vzorků, jež představují polohy pulzů, je výpočet proveden v počátku souřadné soustavy. Po detekci průsečíku je jeho poloha přičtena počátku dříve zvoleného rozsahu a řídicí signál **DETECTION_DONE** převede systém do stavu **DONE**.



Obr. 3.23: Multiplexace vstupů mikroprocesoru Picoblaze [15]

3.8.3 Využití Picoblaze v projektu, přenos dat do PC

V rámci systému, do kterého bude kód navržený v rámci diplomové práce integrován, je pomocí Picoblaze řízena řada důležitých procesů.

Procesor zajišťuje sériovou konfiguraci analogově-digitálního převodníku, prováděnou prostřednictvím rozhraní SPI ve spojení s SPI driverem, jenž byl také navržen v rámci diplomové práce. S využitím definovatelných sekvencí v registrech převodníku, je možno otestovat, zda jsou vstupní data interpretována správným způsobem.

Pomocí procesoru je možno generovat krátké resetovací impulzy pro různé komponenty napříč celým systémem.

Sekvenční přístup a snadnou generaci definovatelných zpoždění lze využít při ovládání periférií desky ML505. Příkladem může být použití LCD displeje, nastavení prahu pomocí rotačního enkodéru či snadné ošetření zákmitů tlačítek.

Picoblaze také umožňuje poměrně jednoduše implementovat rozhraní UART, umožňující přenést data z blokové paměti RAM do PC. V testovací fázi vývoje, kdy je stěžejní zejména korektní záznam naměřených hodnot, však lze využít pro zachycení a uložení dat nástroj **Chipscope**, jenž je součástí plné verze prostředí Xilinx ISE Design Suite.

Prostřednictvím Chipscope lze zaznamenat definovatelný počet vzorků libovolných vnitřních signálů zařízení, v reakci na změnu signálu, jenž je označen za trigger. Jako médium pro záznam dat využívá Chipscope vnitřní blokové paměti analyzovaného FPGA.

4 VYHODNOCENÍ VÝSLEDKŮ SIMULACÍ

Kapitola čtvrtá se zabývá popisem simulačních technik, jež byly využity pro verifikaci funkce vytvořeného systému. Srovnání výsledků simulací s výstupem modelu je uvedeno jako součást příloh práce.

4.1 Nástroje simulace bloku TOP

Základním nástrojem pro simulaci v jazyce VHDL jsou soubory, označované jako **Testbench**. Tyto soubory umožňují použít speciální konstrukce jazyka, jež nejsou syntetizovatelné v reálném HW, ale poskytují možnost řídit časové parametry signálů. Přestože během vývoje systému byly vytvořeny soubory Testbench pro mnohé z vyvíjených komponent, v práci je popsán pouze soubor, analyzující kompletní systém, tedy `top_tb.vhd`.

Výše zmíněný Testbench bloku TOP, byl navrhován tak, aby kromě ověření funkčnosti implementovaného algoritmu umožnil také testovat reakci systému na změny řídicích signálů. Základní struktura testovacích souborů je tvořena několika základními součástmi.

Samotný blok TOP, je v kódu reprezentován jako komponenta, jejíž porty jsou připojeny k vnitřním signálům. Kód dále obsahuje procesy, určené pro generaci hodinových signálů pomocí konstrukcí, umožňujících zařadit do procesu definovatelné zpoždění. Změny řídicích signálů jsou obvykle prováděny prostřednictvím procesu, označovaného jako **Stimulus Process**. Podle toho jestli jsou testovány řídicí signály nebo algoritmus jsou použity některé ze součástí, popsanych v následujících podkapitolách (nepoužité součásti je v kódu třeba označit jako komentáře).

4.1.1 Testování řídicích signálů

V případě testování řídicích signálů a přechodů mezi jednotlivými režimy zařízení je pro generaci vstupních dat využito dvou procesů. První proces představuje generátor pseudonáhodných posloupností, jenž je implementován s použitím funkce **uniform**, jak je naznačeno v následujícím úseku kódu, představujícím sekvenční proces, řízený vzestupnou hranou hodinového signálu.

```
rand_gen : process (clk)
    variable seed1, seed2 : positive := 1;
    variable rand : real;
    variable range_of_rand : real := 40.0;

begin
    if rising_edge(CLK) then
        uniform(seed1, seed2, rand);
        sample_CH1_in <= std_logic_vector(to_unsigned(integer
            (rand*range_of_rand)*gen_pulse, SAMPLE_WIDTH));
    end if;
end process;
```

Samotná funkce `uniform` je součástí slohy `IEEE.math_real`. Vstupní hodnoty **seed1** a **seed2** slouží k prvotní inicializaci pseudonáhodného generátoru. Na základě jejich inicializačních hodnot je rozlišena výstupní pseudonáhodná posloupnost, jež je generována jako reálné číslo v rozsahu 0 až 1. Pro získání hodnot v požadovaném rozsahu stačí výstupní hodnotu **rand** vynásobit maximální hodnotou tohoto rozsahu (**range_of_rand**) [16].

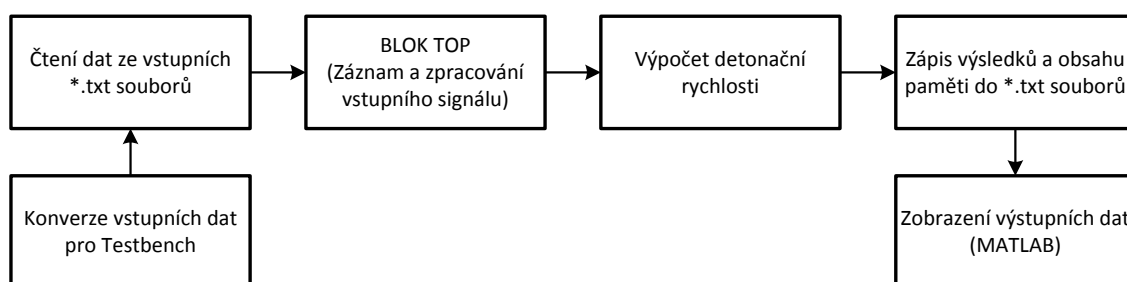
Signál, jenž je generován výše uvedeným způsobem, je dále násoben hodnotou signálu **gen_pulse**, jenž představuje impuls, generovaný separátním procesem v požadovaném čase simulace. Ke generaci impulsu libovolného tvaru lze využít několika příkazů **wait**, jak je naznačeno v následujícím úseku kódu.

```
wait for 100us; gen_pulse <= 20;
wait for 8ns;   gen_pulse <= 40;
wait for 8ns;   gen_pulse <= 80;
wait for 8ns;   gen_pulse <= 100;
wait for 8ns;   gen_pulse <= 100;
wait for 8ns;   gen_pulse <= 20;
wait for 8ns;   gen_pulse <= 1;
```

Správnou funkci systému, lze otestovat různými kombinacemi řídicích signálů nastavených v procesu Stimulus (např. `INIT_R` nebo signálů, jež jsou součástí `Record Detector_ctrl_in`, viz. podkapitola 3.4.1). K vyhodnocení výsledků slouží tzv. **Waveform** soubory, jež jsou popsány v podkapitole 4.1.3.

4.1.2 Testování funkce algoritmu

Testování funkčnosti algoritmu bylo provedeno s využitím záznamů reálných signálů, měřených ve spolupráci s Fakultou chemicko-technologickou Univerzity Pardubice. Pro návrh byla použita převážně data měřená pomocí osciloskopu, jejichž parametry jsou shrnuty v tabulce 2.1. V závěrečné části vývoje byla funkčnost algoritmu ověřena také na datech, naměřených přímo s využitím převodníku ADS4229 a zaznamenaných pomocí nástroje **Chipscope** (zpráva z měření a jeho vyhodnocení je součástí příloh). Celý řetězec, jenž je pro testování algoritmu použit, je zobrazen na obr. 4.1.



Obr. 4.1: Řetězec simulace zpracování záznamů reálných signálů

Aby bylo použití záznamů v testovacím kódu co nejjednodušší, je data nejprve nutno upravit do vhodné formy. Kód je navržen pro práci s textovými soubory, jež mají hodnoty vzorků uloženy na jednotlivých řádcích. Pro konverzi osciloskopických záznamů byl v prostředí MATLAB vytvořen skript s názvem `csv_to_TB.m`. K převodu záznamů z Chipscope, jež byly exportovány s příponou `*.prn`, lze využít skript

prn_to_TB.m, jenž je založen na skriptu **Data_input_PRN.m** (autorem je Ing. Michal Kubíček, Ph.D.), doplněném o vytvoření textového souboru ve formátu vhodném pro simulaci.

Čtení dat z generovaných textových souborů je implementováno prostřednictvím procesů **read_File_CHx**, kde *x* představuje číslo kanálu. V deklarační části procesu je třeba uvést kompletní cestu k umístění vybraného textového souboru na disku počítače. Samotné čtení probíhá prostřednictvím funkcí **readline** (přečtení řádku souboru) a **read** (přečtení hodnoty na daném řádku), jež jsou součástí slohy **std.textio**. Hodnoty vzorků, vyčtené z textových souborů jsou přivedeny přímo na datové vstupy bloků Detector (**detector_in.DATA**) a simulují tedy výstupní data převodníků po úpravě, popsané v podkapitole 3.3.3.

Pro výpočet detonační rychlosti dle vztahu (2.10) je třeba do proměnných **x_chy** (*y* odpovídá číslu kanálu) v procesu **get_VOD** zadat vzájemné pozice vláken, odpovídající jednotlivým kanálům, v milimetrech. Výsledná detonační rychlost je dostupná jako hodnota signálu **detected_VOD**.

Po dokončení výše uvedeného výpočtu je zahájen proces uložení obsahu blokové RAM do textového souboru. K vytvoření souboru a záznamu hodnot je využito funkcí **write** a **writeline**. Pro řízení přístupu k BRAM je použit jednoduchý adresní generátor, čítající hodnoty od nuly po nejvyšší hodnotu adresního rozsahu. Výstupem simulace je *x* textových souborů s názvy **OUTPUT_CHx.txt** (obsah BRAM) podle počtu použitých kanálů a soubor **OUTPUT_RESULTS.txt**, jenž obsahuje hodnoty nalezených pozic impulsů a výslednou detonační rychlost. Výchozím úložištěm pro uvedené soubory je kořenová složka projektu.

K zobrazení obsahu výstupních souborů lze využít skript **show_data.m**. Při spuštění je nutné zvolit soubor s názvem **OUTPUT_CH1.txt**, ostatní soubory jsou načteny automaticky.

4.1.3 Waveform soubory

Pro analýzu průběhu libovolného signálu či komponenty návrhu je při simulaci možno využít tzv. Waveform souborů. Jedná se o soubory s příponou *.wcfg, prostřednictvím kterých je uživateli umožněno uložit aktuální nastavení sledovaných signálů v okně simulace a není tedy nutné opakovaně nastavovat formát interpretace, barevné rozlišení signálů či jejich uspořádání.

V návrhu bylo vytvořeno několik Waveform souborů, pro snadnou analýzu nejdůležitějších bloků. Soubory jsou přiloženy spolu se zdrojovými kódy testovacích souborů.

ZÁVĚR

V rámci diplomové práce bylo nejprve stručně pojednáno o principech, používaných v zařízeních na měření parametrů energetických materiálů. Důraz byl kladen na popis detonační rychlosti a principu optické metody.

Na základě získaných poznatků a s využitím záznamů reálných signálů byl v prostředí MATLAB vytvořen funkční model, umožňující detekovat přesnou pozici impulsu v jedno nebo vícekanálovém záznamu signálů optických sond. Model byl navržen ve variantě, pracující s čísly s plovoucí řádovou čárkou i v celočíselné variantě, umožňující simulovat problémy, související s omezeným číselným rozsahem hardwarové implementace. S využitím modelu byl navržen vhodný typ filtrace pro zvýšení přesnosti detekce.

Stěžejní třetí kapitola textu, poskytuje podrobný popis plně implementovatelného systému, jehož návrh byl hlavní náplní diplomové práce a jenž byl navrhován pro použití na vývojové platformě ML505 s čipem VIRTEX-5 firmy Xilinx, s ohledem na pozdější implementaci do desky s novějším čipem ARTIX-7.

V rámci návrhu systému byl na základě dříve vytvořeného modelu úspěšně implementován algoritmus detekce (Metoda tečen), jehož funkčnost byla ověřena na sadě záznamů, dostupných z měření reálných signálů optických sond. Popis a vyhodnocení jednotlivých měření jsou uvedeny jako součást příloh včetně porovnání s hodnotami získanými s použitím modelu.

V době dokončení diplomové práce byl projekt, jehož součástí je návrh uvedeného systému, ve fázi testování. Ve finálním návrhu bude možno dosáhnout poměrně velkého zjednodušení kódu odstraněním parametrů, jež byly pro testovací účely navrženy ve více variantách.

LITERATURA

- [1] National Institute of Rock Mechanics. Evaluation of explosives performance through in-the-hole detonation velocity measurement [online]. 2001 [cit. 2013-12-15]. Dostupné z: <http://arblast.osmre.gov/downloads/International/India%20papers/VOD-S_T-final.pdf>
- [2] TETE, Aruna D., A.Y. DESHMUKH a R.R. YERPUDE. Velocity of detonation (VOD) measurement techniques practical approach. *International Journal of Engineering and Technology* [online]. 2013 [cit. 2013-12-15]. Dostupné z: <<http://www.sciencepubco.com/index.php/ijet/article/download/1023/817>>
- [3] MEYER, Rudolf, Josef KÖHLER a Axel HOMBURG. Explosives, 6th, Completely Revised Edition [online]. Germany: Wiley-Vch, 2007. ISBN 978-3-527-31656-4. Dostupné z: <<http://miningandblasting.files.wordpress.com/2009/09/explosives-6th-edition-by-meyer-kohler-and-homburg-2007.pdf>>
- [4] CHAN, Edwin M., Vivian LEE, Samuel P. MICKAN a Phil J. DAVIES. Low-Cost Optoelectronic Devices to Measure Velocity of Detonation [online]. 2005 [cit. 2013-12-15]. Dostupné z: <http://www.eleceng.adelaide.edu.au/thz/publications/chan_2005_proc_spie.pdf>
- [5] KRUPKOVÁ, Vlasta a Petr FUCHS. Matematika 1 [online]. [cit. 2013-12-15]. Dostupné z: <<http://www.umat.feec.vutbr.cz/~krupkova/hlavní.pdf>>
- [6] MEYER-BAESE, Uwe. Digital Signal Processing with Field Programmable Gate Arrays [online]. Germany: Springer, 2007 [cit. 2013-12-15]. ISBN 978-3-540-72612-8. Dostupné z: <Digital Signal Processing with FPGAs%2C 3rd Edition.pdf>
- [7] SELTMAN, Howard J. Experimental Design and Analysis [online]. 2013 [cit. 2013-12-15]. Dostupné z: <<http://www.stat.cmu.edu/~hseltman/309/Book/Book.pdf>>
- [8] The evolution of FPGA coprocessing. STRICKLAND, MIKE. ALTERA. *Hearst Electronic Products* [online]. San Jose, CA, 2010 [cit. 2014-05-02]. Dostupné z: <http://www.electronicproducts.com/Digital_ICs/Standard_and_Programmable_Logic/The_evolution_of_FPGA_coprocessing.aspx>
- [9] XILINX, Inc. *ML505/ML506/M L507 Evaluation Platform: User Guide (UG347, v3.1.2)* [online]. 2011 [cit. 2014-05-05]. Dostupné z: <http://www.xilinx.com/support/documentation/boards_and_kits/ug347.pdf>
- [10] XILINX, Inc. *Virtex-5 Family Overview: Product Specification (DS100, v5.0)* [online]. 2009 [cit. 2014-05-10]. Dostupné z: <http://www.xilinx.com/support/documentation/data_sheets/ds100.pdf>
- [11] XILINX, Inc. *7 Series FPGAs Overview: Product Specification (DS180, v1.15)* [online]. 2014 [cit. 2014-05-10]. Dostupné z: <http://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf>
- [12] Texas Instruments. *ADS4229: Dual-Channel, 12-Bit, 250-MSPS Ultralow-Power AD* [online]. 2011 [cit. 2014-05-10]. Dostupné z: <<http://www.ti.com/lit/ds/symlink/ads4229.pdf>>
- [13] XILINX, Inc. *Virtex-5 FPGA: User Guide (UG190, v5.4)* [online]. 2012 [cit. 2014-05-07]. Dostupné z: <http://www.xilinx.com/support/documentation/user_guides/ug190.pdf>Xilinx. Virtex-5>

- [14] XILINX, Inc. *LogiCORE IP Block Memory Generator v7.3: Product Guide (PG058)* [online]. 2012 [cit. 2014-05-12]. Dostupné z:
<http://www.xilinx.com/support/documentation/ip_documentation/blk_mem_gen/v7_3/pg058-blk-mem-gen.pdf >
- [15] XILINX, Inc. *PicoBlaze 8-bit Embedded Microcontroller: User Guide (UG129)* [online]. 2011 [cit. 2014-05-17]. Dostupné z:
<http://www.xilinx.com/support/documentation/ip_documentation/ug129.pdf>
- [16] Generating random numbers in a VHDL testbench. VHDL Coding Tips and Tricks [online]. 2013 [cit. 2014-05-17]. Dostupné z:
<<http://vhdlguru.blogspot.cz/2013/08/generating-random-numbers-in-vhdl.html>>

SEZNAM ZKRATEK

A/D – Analog-to-Digital

ADC – Analog-to-Digital Converter

ARM – Advanced RISC Machine

DCM – Digital Clock Manager

DSP – Digital Signal Processing (Processor)

DDR – Dual Data Rate

FIFO – First-In First-Out

FIR – Finite Impulse Response

FPGA – Field Programmable Gate Array

IP – Intellectual Property

LED – Light Emitting Diode

LVDS – Low Voltage Differential Signaling

JTAG – Joint Test Action Group

RAM – Random Access Memory

ROM – Read Only Memory

RTL – Register Transfer Level

SDR – Single Data Rate

SPI – Serial Peripheral Interface

UART – Universal Asynchronous Receiver and Transmitter

VHDL – VHSIC (Very High Speed Integrated Circuit) Hardware Description Language

VoD – Velocity of Detonation

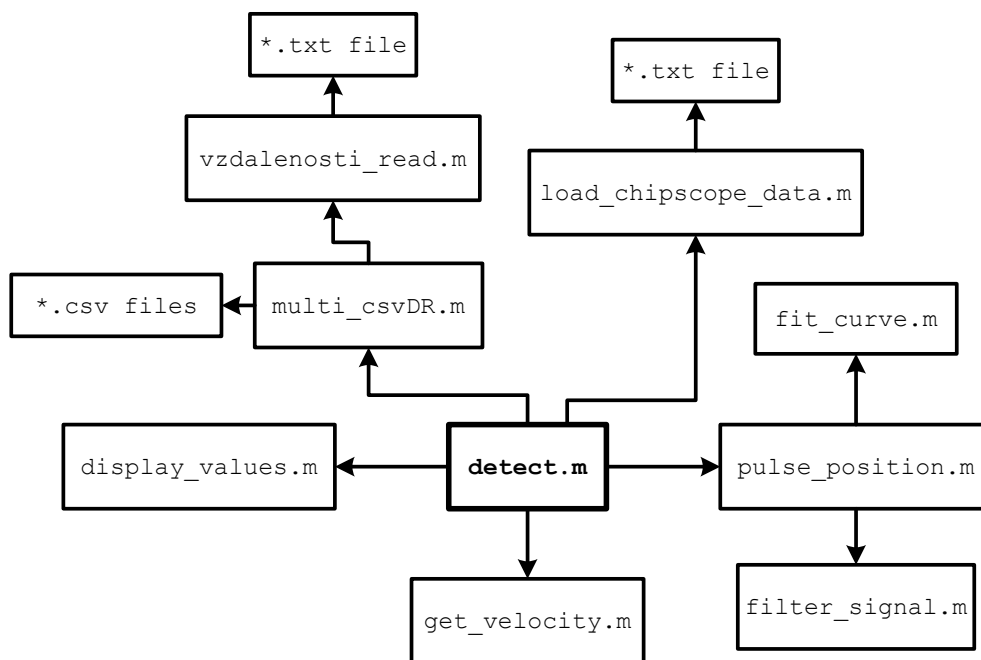
SEZNAM PŘÍLOH

A	Souborová struktura modelu	46
B	Analýza dostupných dat za použití různých konfigurací modelu	46
C	Popis a vyhodnocení záznamů pořízených pomocí ADS4229 a Chipscope	49
C.1	Měření s použitím skleněných vláken.....	50
C.2	Měření s použitím plastových vláken	54
C.3	Tabulka detekovaných rychlostí, srovnání s výsledky simulace	57
D	Obsah DVD	58

A SOUBOROVÁ STRUKTURA MODELU

Na obr. A.1 je zobrazen diagram souborové struktury vytvořeného modelu. Šipky směřují od volající funkce (souboru) k volané. Popis jednotlivých funkcí je uveden jako součást jednotlivých matlabovských souborů a je možné zobrazit ho např. příkazem **help** v prostředí MATLAB, následovaným názvem zkoumané funkce.

Struktura verze modelu, pracující s celočíselnými čísly, je shodná se strukturou zobrazenou níže, název funkcí je doplněn příponou **_INT** (např. **detect_INT.m**).



Obr. A.1: Diagram souborové struktury modelu

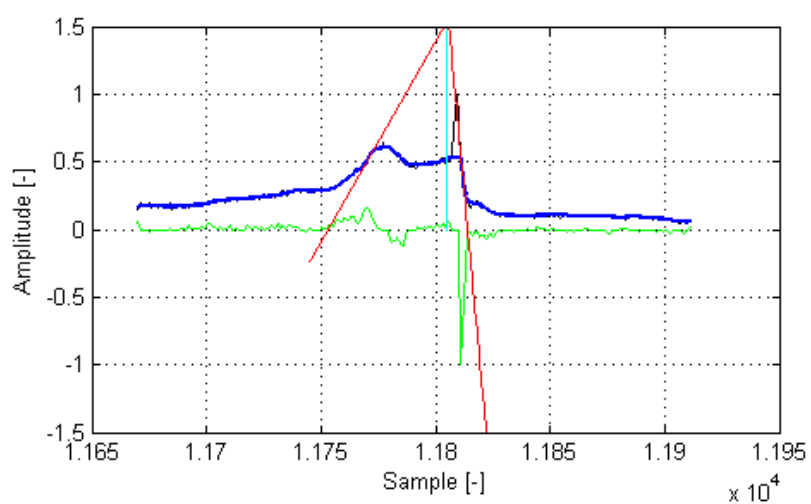
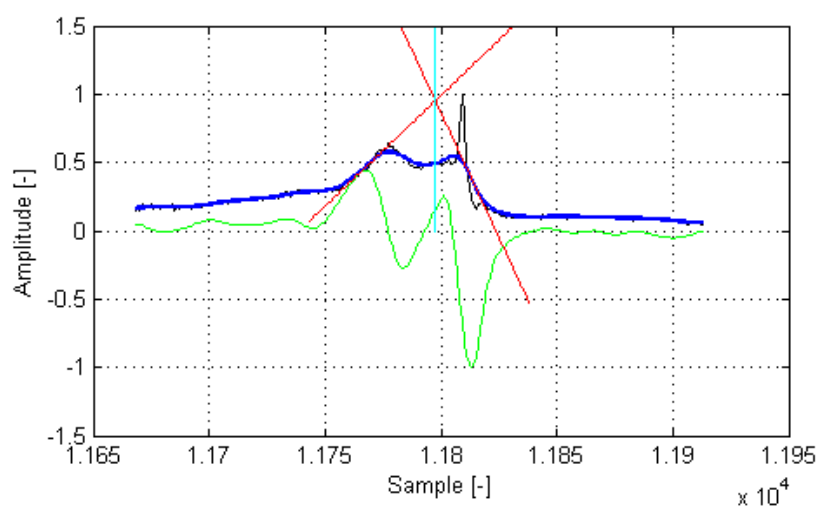
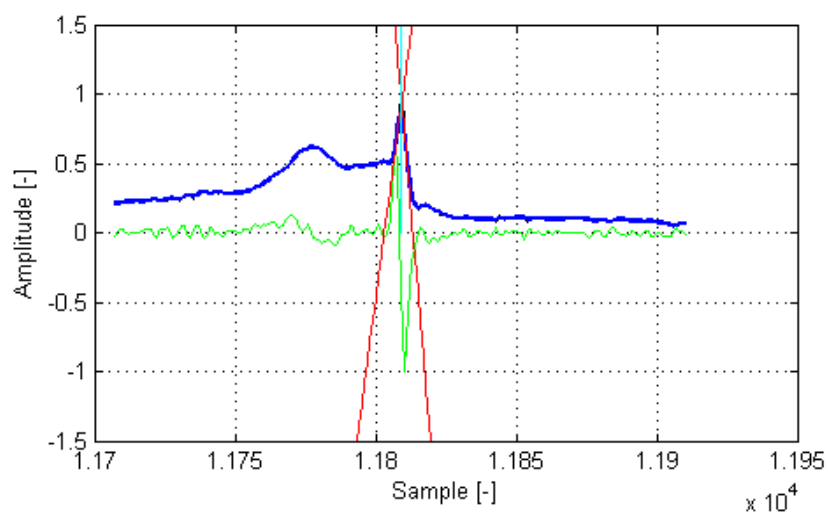
B ANALÝZA DOSTUPNÝCH DAT ZA POUŽITÍ RŮZNÝCH KONFIGURACÍ MODELU

Tab. B.1: Popis variant filtrace v modelu, použitých pro testovací účely

Použité varianty FIR filtrů		
Parametr	Varianta	
	Mírný	Ostrý
Typ okna	Hamming	Hamming
Vzorkovací frekvence	250 MHz	250 MHz
Mezní frekvence	15 MHz	5 MHz
Řád	4	24
Použitý mediánový filtr		
Mediánový filtr realizován pomocí funkce medfilt1()		
Pro medián volen počet okolních vzorků 9		

Tab. B.2: Porovnání detekovaných hodnot při různých konfiguracích modelu

Tabulka hodnot detekované VoD v závislosti na konfiguraci							
Soubor s měřením (výbušnina)	Princip detekce	Varianta Filtru					
		Mírný		Ostrý		Mírný + Mediánový	
		VoD _{P21}	VoD _{P32}	VoD _{P21}	VoD _{P32}	VoD _{P21}	VoD _{P32}
A-IX-1-78g	A	5806	6203	5794	6213	5806	6203
	B	5783	6219	5781	6219	5783	6219
	C	5796	6215	5792	6216	5796	6215
Bleskovice	A	6778	6616	6783	6613	6778	6616
	B	6804	6608	6804	6605	6804	6608
	C	6770	6605	6770	6611	6770	6605
Np2_04	A	4908	4198	4908	4177	4908	4198
	B	4851	4265	4846	4271	4851	4265
	C	4923	4159	4963	4067	4923	4159
Np10_07	A	4791	5421	4794	5421	4791	5421
	B	4822	5421	4808	5421	4822	5421
	C	4799	5421	4797	5424	4799	5421
PI-Np10	A	7962	7700	8011	7659	7962	7662
	B	7957	7672	8034	7682	7957	7677
	C	7975	7721	7993	7596	7975	7598
Politol	A	2358	1897	2364	1898	2358	1898
	B	2370	1889	2368	1905	2370	1904
	C	2360	1902	2359	1901	2359	1902
Semtex	A	8016	7095	7984	7100	8011	7100
	B	8016	7100	7921	7100	8011	7100
	C	8016	7095	8005	7100	8011	7100
Tetryl-74g	A	5924	6090	5922	6090	5922	6092
	B	5879	6105	5907	6074	5879	6105
	C	5924	6087	5929	6089	5924	6087
Všechny hodnoty uvedeny v m/s. Principy detekce: A - podle průsečíku B - maximum diferenciálu C - minimum diferenciálu							



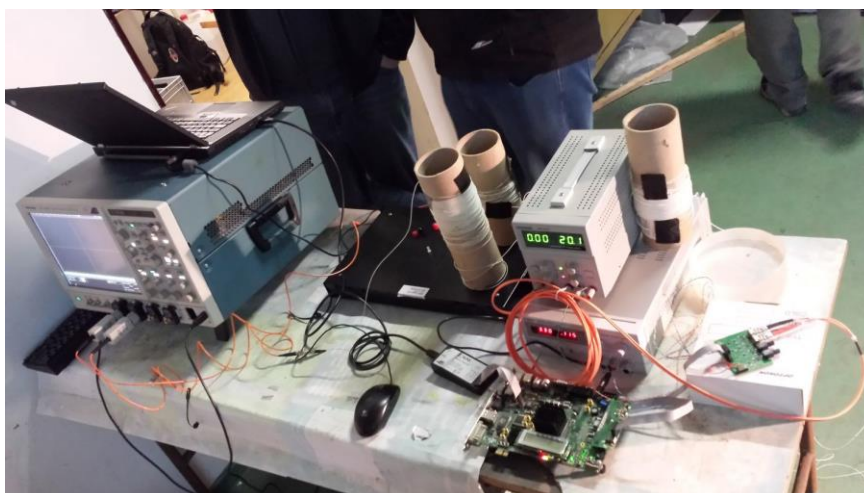
Obr. B.1: Srovnání vlivu filtrů na tvar pulzu Politolu. Shora mírný, ostrý, mírný + mediánový filtr

C POPIS A VYHODNOCENÍ ZÁZNAMŮ POŘÍZENÝCH POMOCÍ ADS4229 A CHIPSOCPE

Dne 16.5.2014 byla provedena sada měření s využitím desky s převodníkem ADS4229, propojeným s vývojovou deskou ML505. Testy byly provedeny v detonační komoře chemicko-technologické fakulty Univerzity Pardubice. Kód VHDL, jenž byl použit pro konfiguraci čipu VIRTEX-5, byl redukován pouze na bloky rozhraní A/D převodníku, procesoru Picoblaze (ovládání periférií desky) a jednoduchý proces, generující trigger při překročení hodnoty, nastavené rotačním enkodérem.

Záznam obou kanálů převodníku byl proveden pomocí nástroje Chipscope v délce 32768 vzorků na kanál. Současně s použitím Chipscope byl také proveden záznam s využitím Osciloskopu Tektronix.

Během měření byly testovány obě varianty optických vláken, tedy plastové i skleněné, přičemž při použití skleněné varianty byl signál rozdělen splitterem pro osciloskopický záznam.



Obr. C.1: Fotografie měřicí aparatury



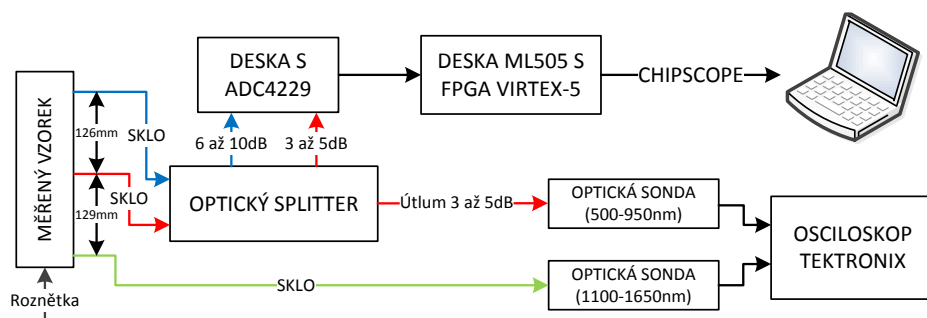
Obr. C.2: Příprava výbušniny Semtex pro měření

C.1 Měření s použitím skleněných vláken

Následující část příloh obsahuje vyhodnocení měření na jednotlivých výbušninách, zaznamenaných s využitím nástroje **Chipscope**. ΔX v tabulkách hodnot představuje rozdíl poloh obou kanálů (tzn. vzdálenost mezi dvěma použitými optickými vlákny). Barvy signálů zobrazených v detailech pulzů neodpovídají barvám ve schématech měření.

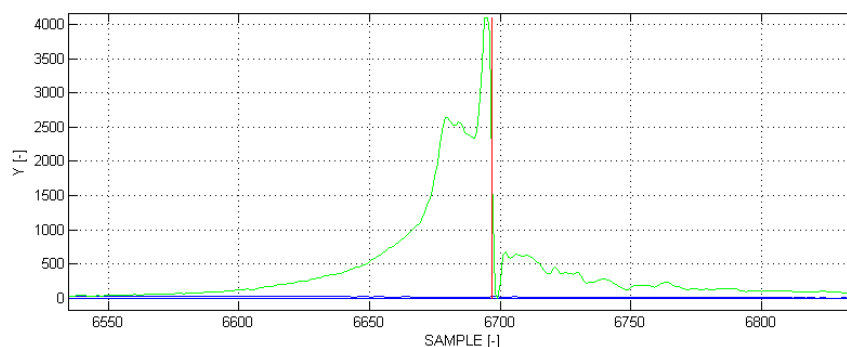
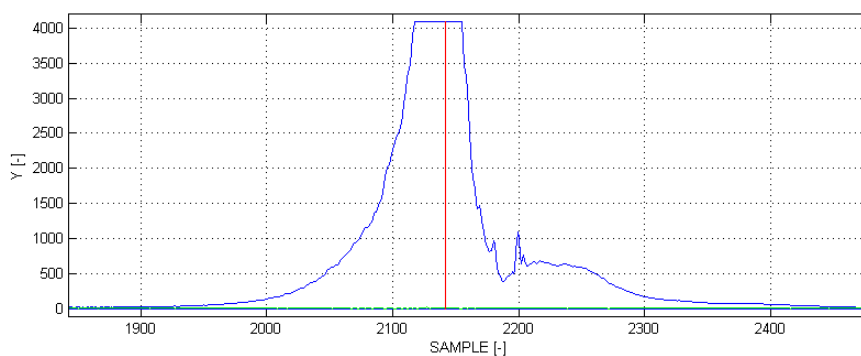
1. BLESKOVICE

Pozn. Analyzovaný průběh odpovídá druhému odpalu Bleskovice.



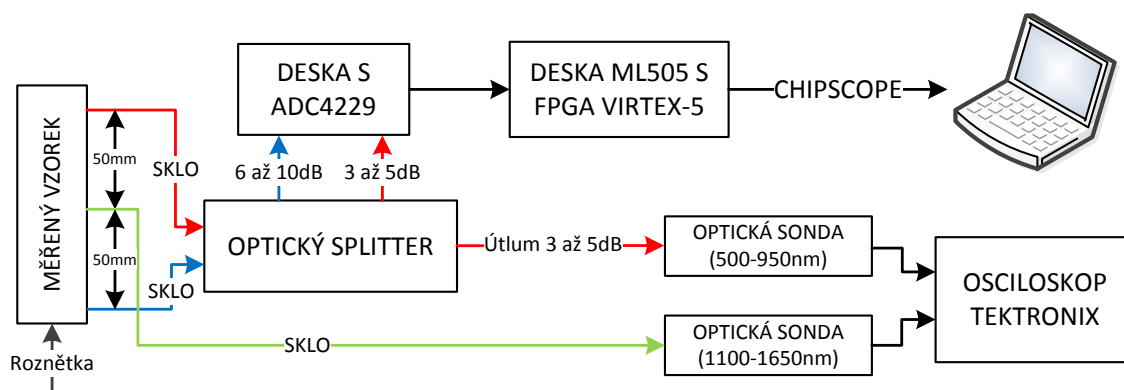
Obr. C.3: Schéma zapojení při měření parametrů exploze Bleskovice

TRESHOLD [-]	500	ΔX [mm]	126
VoD_{TANG} [$m \cdot s^{-1}$]	6915	VoD_{TRESH} [$m \cdot s^{-1}$]	6873



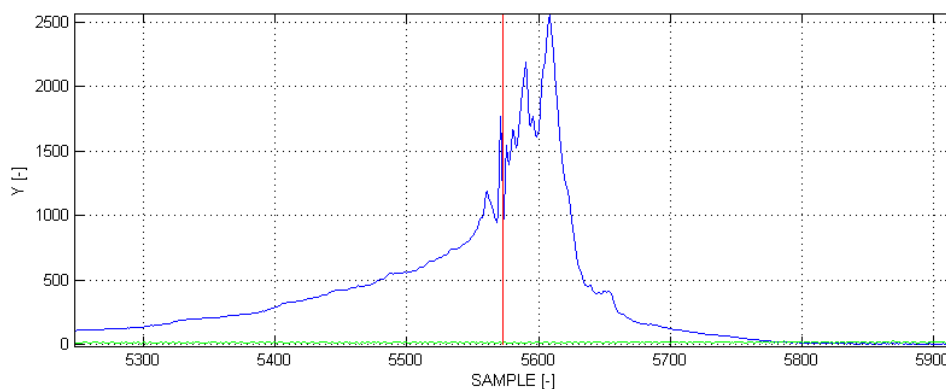
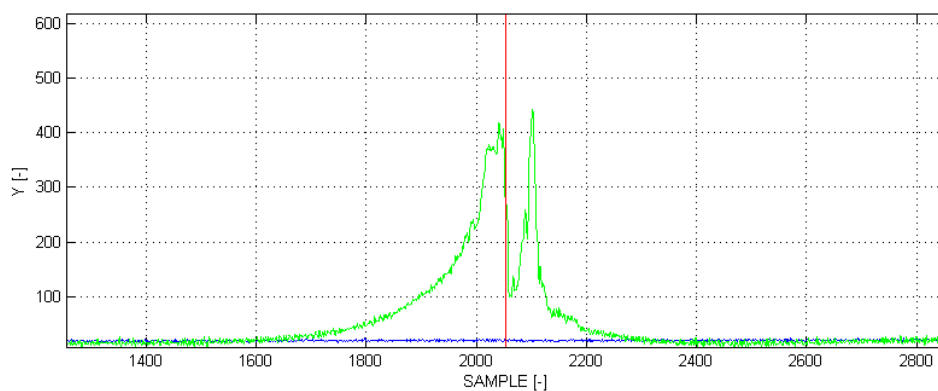
Obr. C.4: Bleskovice. Měřeno skleněnými vlákny. Tvar impulzů a vyznačení detekované pozice (Metoda tečen)

2. SEMTEX



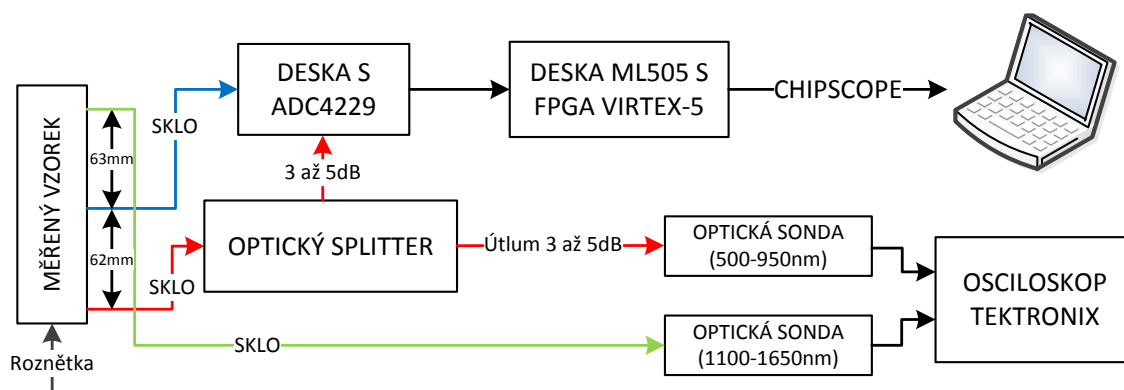
Obr. C.5: Schéma zapojení při měření parametrů exploze výbušniny SEMTEX

TRESHOLD [-]	500	ΔX [mm]	100
VoD_{TANG} [m.s ⁻¹]	7104	VoD_{TRESH} [m.s ⁻¹]	6998



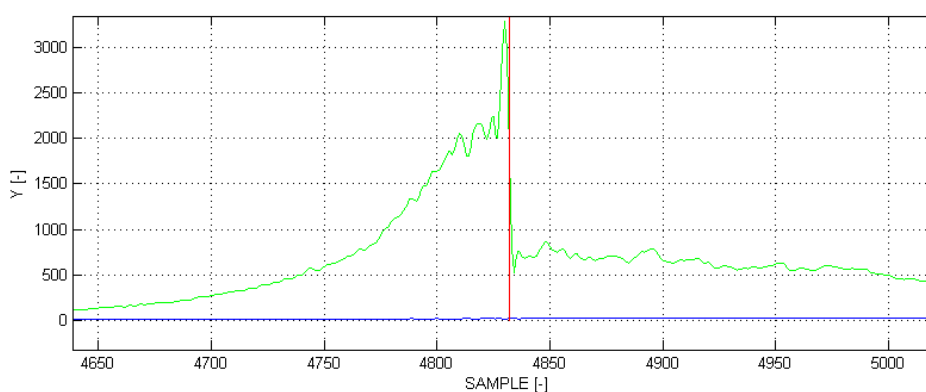
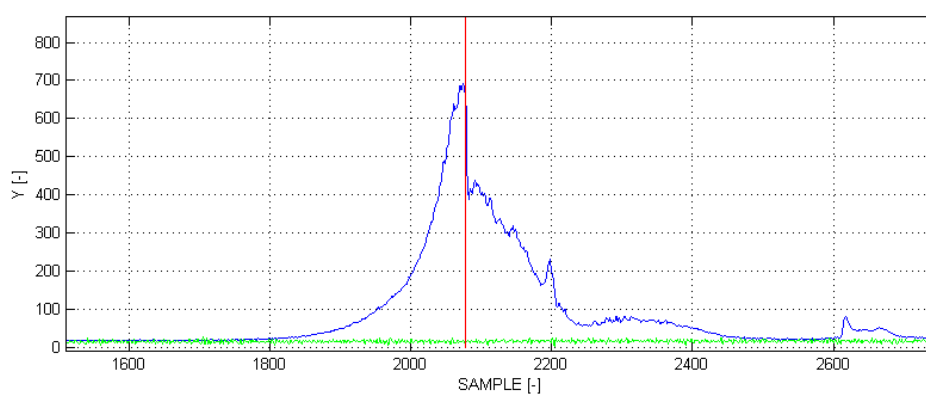
Obr. C.6: Semtex. Měřeno skleněnými vlákny. Tvar impulzů a vyznačení detekované pozice (Metoda tečen)

3. EMULZE PULSAR



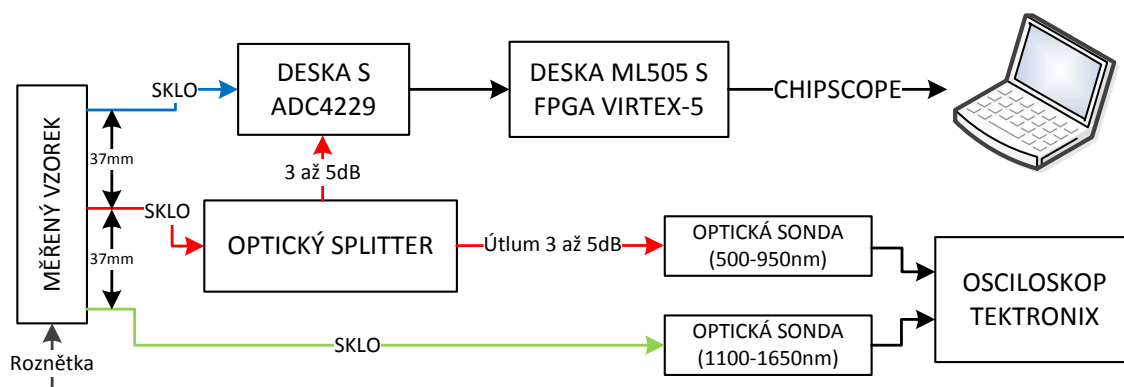
Obr. C.7: Schéma zapojení při měření parametrů exploze emulze Pulsar

TRESHOLD [-]	200	ΔX [mm]	62
VoD_{TANG} [$m.s^{-1}$]	5628	VoD_{TRESH} [$m.s^{-1}$]	5581



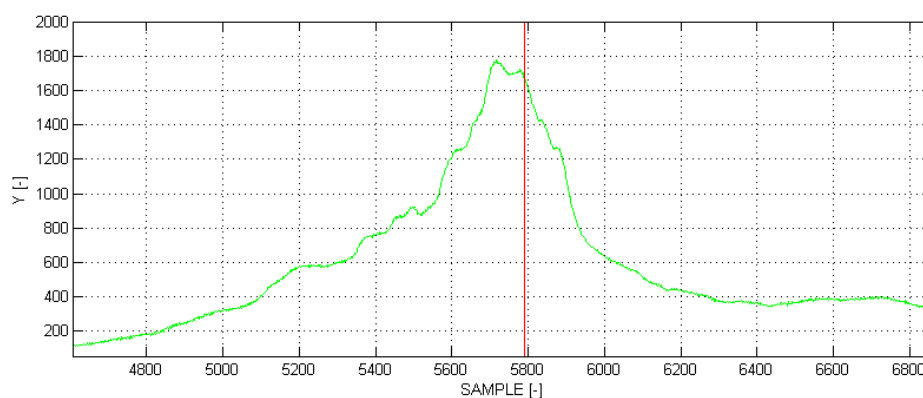
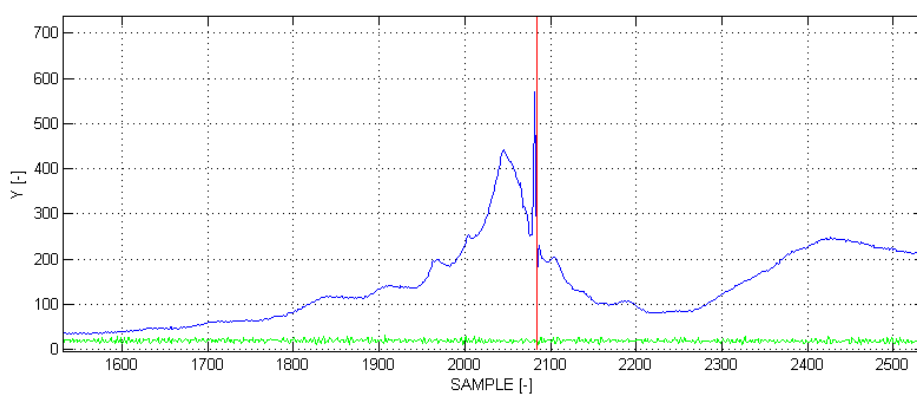
Obr. C.8: Emulze Pulsar. Měřeno skleněnými vlákny. Tvar impulzů a vyznačení detekované pozice (Metoda tečen)

4. T/H 50/50



Obr. C.9: Schéma zapojení při měření parametrů exploze směsi T/H 50/50

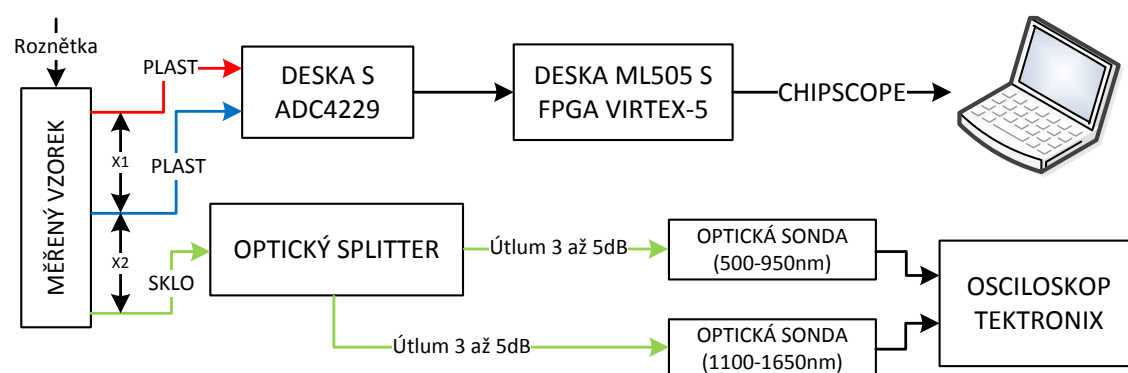
TRESHOLD [-]	200	ΔX [mm]	37
VoD_{TANG} [m.s ⁻¹]	2495	VoD_{TRESH} [m.s ⁻¹]	2566



Obr. C.10: T/H 50/50. Měřeno skleněnými vlákny. Tvar impulzů a vyznačení detekované pozice (Metoda tečen)

C.2 Měření s použitím plastových vláken

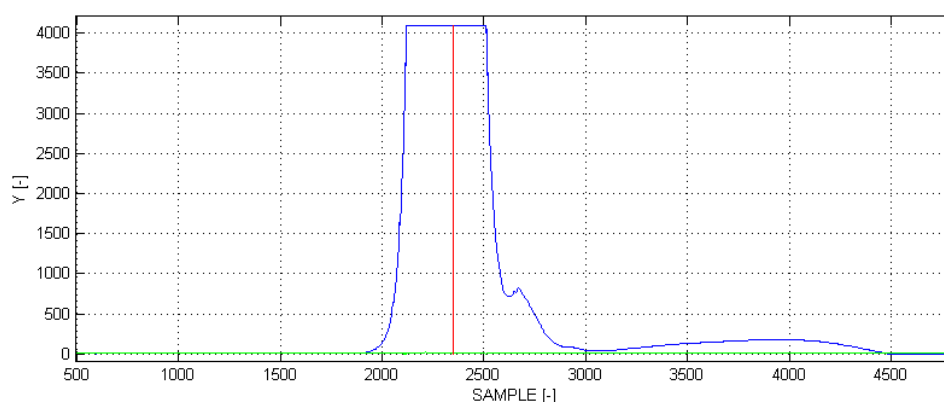
V případě testování měření pomocí plastových vláken, bylo schéma zapojení ve všech případech stejné (viz. obr. C.11). Z průběhů je zřejmé, že absence utlumovacích článků, jež v předchozích měřeních představoval optický splitter (umožňuje dělit pouze signál skleněných vláken), způsobuje značnou saturaci výstupu převodníků. Pro výraznou podobnost průběhů přesaturovaných signálů, je u většiny vzorků zobrazen detail pouze jednoho z impulsů.



Obr. C.11: Obecné schéma zapojení při měření parametrů výbušin s plastovými vlákny

1. BLESKOVICE

TRESHOLD [-]	1000	ΔX [mm]	144
VoD_{TANG} [m.s ⁻¹]	6903	VoD_{TRESH} [m.s ⁻¹]	6820

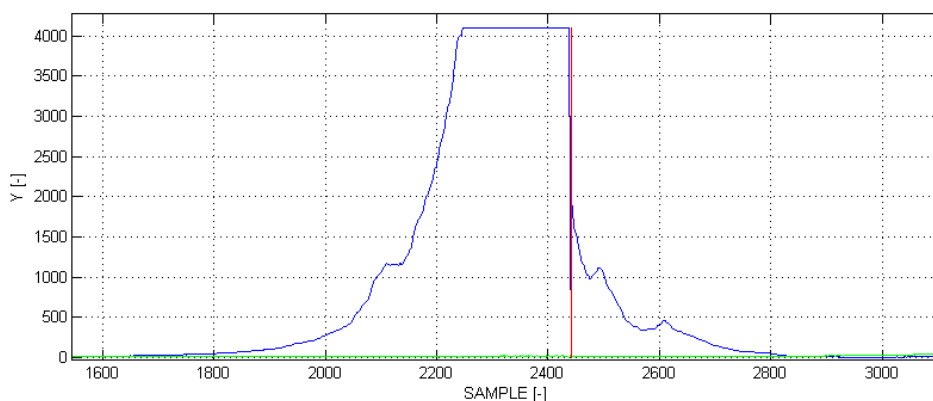


Obr. C.12: Bleskovice. Měřeno plastovými vlákny. Tvar impulsu a vyznačení detekované pozice (Metoda tečen)

2. SEMTEX

Pozn. Lze pozorovat, že ostrý zákmit v oblasti sestupné hrany impulzu způsobuje značné zkreslení při detekci pozice. Řešením může být použití filtrace, případně detekce s použitím metody prahování.

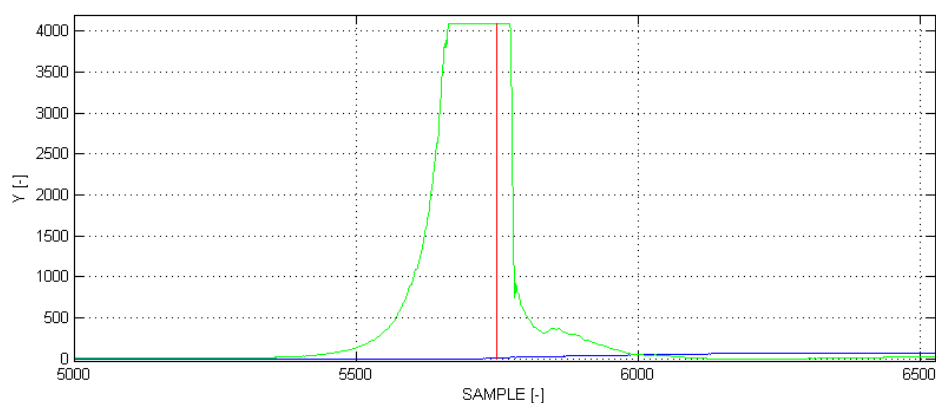
TRESHOLD [-]	500	ΔX [mm]	41
VoD_{TANG} [m.s ⁻¹]	7487	VoD_{TRESH} [m.s ⁻¹]	7700



Obr. C.13: Semtex. Měřeno plastovými vlákny. Tvar impulzu a vyznačení detekované pozice (Metoda tečen)

3. EMULZE PULSAR

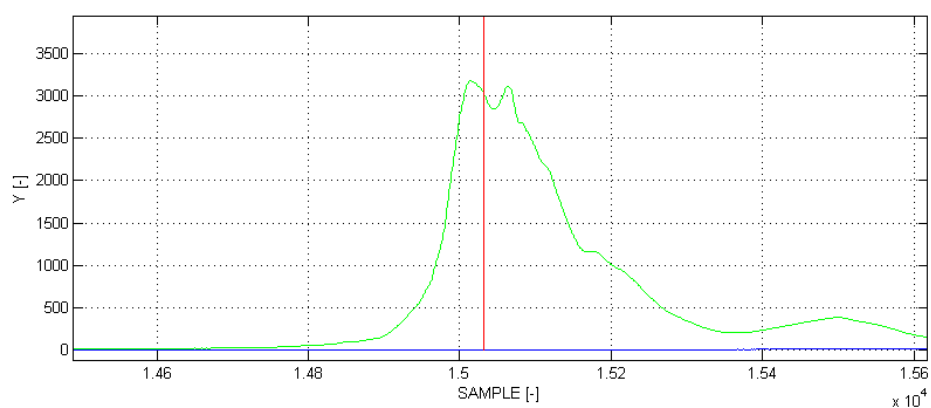
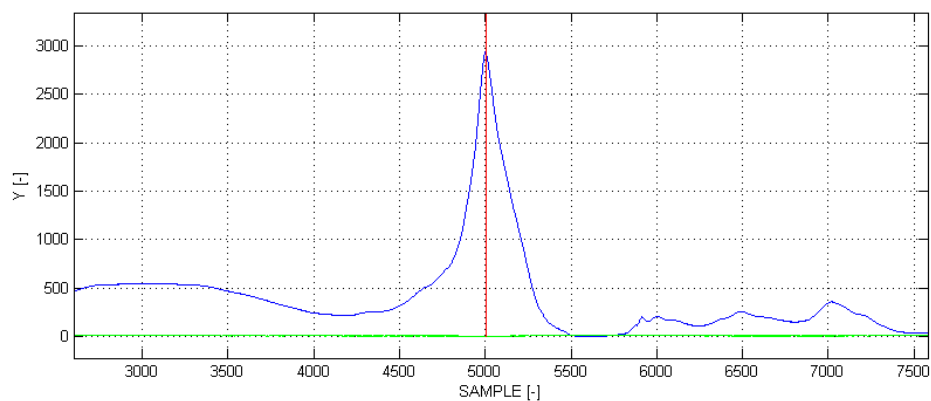
TRESHOLD [-]	300	ΔX [mm]	72
VoD_{TANG} [m.s ⁻¹]	5333	VoD_{TRESH} [m.s ⁻¹]	5138



Obr. C.14: Emulze Pulsar. Měřeno plastovými vlákny. Tvar impulzu a vyznačení detekované pozice (Metoda tečen)

4. T/H 50/50

TRESHOLD [-]	300	ΔX [mm]	58
VoD_{TANG} [m.s ⁻¹]	2517	VoD_{TRESH} [m.s ⁻¹]	2515



Obr. C.15: T/H 50/50. Měřeno plastovými vlákny. Tvar impulzu a vyznačení detekované pozice (Metoda tečen)

C.3 Tabulka detekovaných rychlostí, srovnání s výsledky simulace

V tabulce C.1 lze pozorovat, že pro metodu tečen ve většině případů není odchylka rychlosti, určené simulací HW, od výstupní hodnoty modelu větší než 10% detekované hodnoty. Největším problémem z hlediska detekce je analýza impulzů výbušniny Semtex. Vzhledem k značnému množství zákmitů a parazitních špiček je při analýze vhodné použít filtraci.

Všechny výstupní hodnoty modelu, byly určeny s využitím metody tečen. Uvedené varianty filtrace odpovídají filtrům, popsaným v tabulce B.1.

Tab. C.1: Souhrnná tabulka detekovaných rychlostí, včetně porovnání s výstupem modelu

VZOREK	Typ kabelu	Detekovaná VOD (ms^{-1})		Výstup modelu s filtrací (ms^{-1})	
		Prahování	Tečny	Mírný + Medián	Ostrý
BLESKOVICE	SKLO	6873	6915	6918	6925
	PLAST	6820	6903	6857	6869
SEMTEX	SKLO	6998	7104	7018	7129
	PLAST	7700	7487	7493	7369
EMULZE PULSAR	SKLO	5581	5628	5626	5626
	PLAST	5138	5333	5330	5296
T/H 50/50	SKLO	2566	2495	2492	2491
	PLAST	2515	2517	2518	2519

D OBSAH DVD

Kromě **elektronické verze** diplomové práce ve formátu pdf, je součástí přiloženého DVD několik složek následujícího obsahu:

- **DESIGN** – obsahem je projekt, vytvořený v prostředí ISE Design Suite s názvem VOD_Detect, jenž představuje navržený systém. Všechny **zdrojové kódy** jsou uloženy ve složce **Sources**.
- **FOTO** – složka obsahuje fotografie z měření prováděného 16.5.2014
- **MODEL** – složka obsahuje model systému vytvořený v prostředí MATLAB
- **SKRIPTY** – skripty, jež byly vytvořeny v průběhu návrhu systému, např. skripty pro převod dat do formátu vhodného pro simulaci
- **ZAZNAMY** – sada záznamů z měření uskutečněných ve spolupráci s Fakultou chemicko-technologickou Univerzity Pardubice