

Univerza v Ljubljani
Fakulteta za *matematiko in fiziko*



Oddelek za fiziko, *matematično-fizikalna smer*

BLAŽ BAČNIK

Razvoj hitrega časovno digitalnega pretvornika

Diplomsko delo

Mentor: prof. dr. Samo Korpar

Ljubljana, 25. november 2012

Izvleček

Eden izmed možnih svetlobnih senzorjev za detektorje obročev Čerenkova bi lahko bila silicijeva fotopomnoževalka, vendar je zaradi šuma potrebno dovolj natančno določiti čas dogodkov. Načrtovan in izdelan je bil časovno-digitalni pretvornik, ki izmeri signale s časovno resolucijo pod nanosekundo. Uporabljena je bila tehnologija električnih programabilnih polj (FPGA), ki omogoča hitro prilagoditev funkcije vezja. Pretvornik je večkanalen in omogoča zajemanje podatkov preko paralelnega vhoda na računalniku s priloženim programom. Opravljena je bila tudi umeritev ter nato še preizkus s silicijevo fotopomnoževalko, ki je potrdil ustreznost naprave za omejevanje šuma pri dani aplikaciji.

Ključne besede: *časovno digitalni pretvornik, TDC, FPGA, nanosekundni, RICH, silicijeva fotopomnoževalka, SiPM*

Abstract

Silicon photomultiplier might be used as a light sensor for ring imaging Cherenkov detectors when accurate timing information for events is available. A time-to-digital converter (TDC) was designed and implemented on field-programmable gate array (FPGA) with sub-nanosecond resolution. Converter is multichannel and allows parallel port connection for data acquisition with attached computer program. Calibration was also carried out, followed by a test to confirm device was fit to limit noise from silicon photomultiplier in the given setting.

Keywords: *time-to-digital converter, TDC, FPGA, nanosecond, RICH, silicon photomultiplier, SiPM*

PACS: 06.30.Ft *Time and frequency*, 29.40.Ka *Cherenkov detectors*

Kazalo

1	Uvod	7
2	Identifikacija delcev	9
2.1	Sevanje Čerenkova	9
2.2	Detektor obročev Čerenkova	10
2.3	Silicijeva fotopomnoževalka	12
2.4	Časovna korelacija	13
3	Hitri časovni merilniki	15
3.1	Metode merjenja z visoko ločljivostjo	15
3.2	Tehnologija FPGA	17
4	Izdelava merilnika	19
4.1	Pregled delovanja	21
4.2	Skupno vezje	22
4.3	Merilni kanal	25
4.4	Bralno vezje	30
4.4.1	Branje in pisanje	32
4.5	Povezava z računalnikom	33
5	Umeritev in karakteristike merilnika	37
5.1	Statistična umeritev	37
5.2	Karakteristike naprave	39
5.2.1	Ločljivost	40
5.3	Preizkus s silicijevo fotopomnoževalko	41
6	Zaključek	47
A	Program za zajemanje podatkov	49

1 Uvod

Pri nadgradnji spektrometra Belle na trkalniku KEK v Tsukubi na Japonskem je predvidena uporaba detektorja obročev Čerenkova z aerogelom kot sevalcem. Pomembni deli detektorja predstavljajo svetlobni senzorji, ki zaznavajo fotone. Poleg fotodetektorjev tipa HAPD in MCP-PMT je bila preizkušena tudi uporaba silicijevih fotopomnoževalk. Problem pri njihovi uporabi predstavljajo temni sunki, ki so posledica termičnih ekscitacij. Z merjenjem časa prihoda fotonov lahko močno omejimo njihov vpliv in tako izboljšamo identifikacijo delcev, v tem primeru ločevanje pionov in kaonov z gibalnimi količinami do 4 GeV/c.

Za delovanje metode je potrebno zaznati tudi mesto absorpcije fotona, kar dosežemo z več senzorji. Ti so razporejeni po prostoru in vsak predstavlja svoj kanal. Ker se pričakuje pokrivanje površine več kvadratnih metrov, mora biti bralna elektronika prilagojena velikemu številu vhodov oz. kanalov. Za takšne naloge je med drugim primerna tehnologija električnih programabilnih polj, s katero je mogoče hitro prilagoditi vezje, brez stroškov ponovne izdelave. Tehnologija omogoča močno vzporedno izvajanje funkcij, ki je bistvenega pomena za načrtovano uporabo.

Diplomsko delo bo usmerjeno v izdelavo in umeritev časovno digitalnega pretvornika, ki bo zadostil potrebam za uporabo v detektorju obročev Čerenkova brez zbiralnega sistema in s silicijevimi fotopomnoževalkami kot fotodetektorji.

2 Identifikacija delcev

V fiziki osnovnih delcev izvajamo eksperimente, pri katerih so energije dovolj velike, da nastajajo novi delci. Identifikacija je postopek, s katerim določimo tip delcev, ki so se tvorili, in nam omogoča popis takih pojavov [1]. Optimalna metoda določanja zavisi od samega delca. Na primer, za identifikacijo mionov je večinoma zadosti, če ugotovimo, da so prešli skozi debel absorpcijski material z minimalnimi izgubami energije ali spremembami smeri [2]. Taka metoda je zadovoljiva zaradi zelo majhne stopnje interakcije mionov s snovjo. Po drugi strani elektroni in pozitroni zaradi svoje majhne mase odnesejo veliko energije, ki jo lahko merimo preko elektromagnetnih plazov v kalorimetrih.

Od ostalih nabitih delcev ločujemo še pione (π), kaone (K) in protone (p) ter njihove antidelce. Pri tem velja omejitev, da so dovolj dolgoživi, da sploh dosežejo naš detektor (mase in razpadni časi omenjenih delcev so podani v tabeli 1). Od električno nevtralnih pa lahko z elektromagnetnim

delec	e	μ	π	K	p
masa [MeV/c ²]	0.511	106	140	494	938
razpadni čas	–	$\sim \mu\text{s}$	$\sim 10\text{ ns}$	$\sim 10\text{ ns}$	–

Tabela 1: Mase in razpadni časi nabitih delcev, ki so dovolj dolgoživi, da dosežejo detektor.

kalorimetrom zaznamo še fotone.

Za razlikovanje pionov, kaonov in protonov z metodo odklona v magnetnem polju, poleg predznaka naboja, izmerimo tudi gibalno količino. Vendar samo ta meritev ne zadošča, saj je gibalna količina funkcija tako mase kot hitrosti. Za določitev mase, in posledično tipa delca, se najpogosteje izmeri še njegova hitrost. Dandanes so za ta namen zelo razširjene naprave, ki delujejo na osnovi sevanja Čerenkova [3].

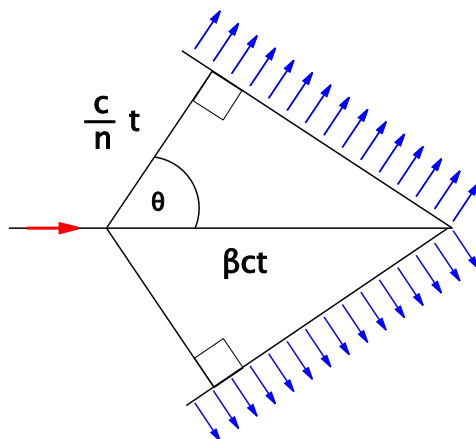
2.1 Sevanje Čerenkova

Do tega pojava pride, ko se nabit delec giblje hitreje od hitrosti svetlobe. Ker ima v snovi svetloba manjšo hitrost kot v vakuumu, delec še vedno zadosti pogojem relativistične fizike. Hitrost svetlobe v vakuumu označimo s c , razmerje fazne hitrosti svetlobe med vakuumom in optičnim medijem pa podaja lomni količnik n . Pogoj za hitrost delca v , pri kateri pride do sevanja Čerenkova, je potem $\frac{c}{n} < v < c$. Leta 1937 je to sevanje podrobno popisal P. A. Čerenkov, po katerem je pojav tudi dobil ime. Čerenkov je

uspeš eksperimentalno določiti zvezo za kot širjenja valovanja glede na smer hitrosti delca [4]

$$\cos \vartheta = \frac{1}{n\beta} \quad , \quad (2.1)$$

pri čemer je $v = \beta c$. Shematičen prikaz pojava je na sliki 1.



Slika 1: Kot θ pri sevanju Čerenkova: hipotenuza trikotnika predstavlja pot, ki jo prepotuje delec v določenem času, daljša kateta pa omejuje razdaljo, do katere se je v tem času razširila valovna fronta v mediju brez disperzije. [5]

Leto kasneje sta Frank in Tamm predstavila elektromagnetno teorijo, ki razloži ta pojav. Zvezo, ki določa spekter sevanja, po niju imenujemo *enačba Frank-Tamm* in podaja število izsevanih fotonov na energijski interval kot [4]

$$\frac{dN_{ph}}{dE} \sim \frac{370}{\text{eV cm}} L \sin^2 \vartheta(E) \quad , \quad (2.2)$$

pri čemer je L dolžina poti, na kateri delec seva. Iz enačbe $c = \lambda\nu$ lahko izpeljemo diferencial $dE = \hbar d\omega = -\frac{\hbar c}{\lambda^2} d\lambda$ (\hbar je tu Planckova konstanta). Število fotonov glede na valovno dolžino zapišemo z enačbo

$$\frac{dN_{ph}}{d\lambda} = \left| \frac{dE}{d\lambda} \right| \frac{dN_{ph}}{dE} = \frac{\hbar c}{\lambda^2} \frac{dN_{ph}}{dE} \quad .$$

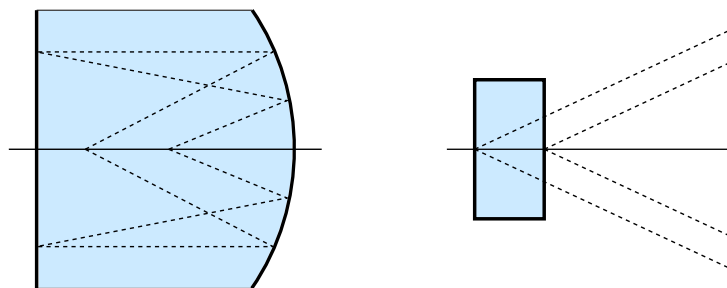
Za človeško oko ima sevanje Čerenkova karakteristično modro-vijolično barvo, ki je posledica te odvisnosti.

2.2 Detektor obročev Čerenkova

Enačba (2.1) predstavlja teoretično osnovo, na podlagi katere določimo hitrost nabitega delca. Napravo s katero jo merimo imenujemo detektor obročev Čerenkova, angl. “Ring Imaging Cherenkov detector” oz. *RICH*. Tip

detektorja, ki sta ga leta 1977 predstavila J. Seguinot in T. Ypsilantis [6], uporablja sferična zrcala, ki zberejo žarke sevanja Čerenkova v obroč na žariščni ravnini. Velikost obroča je odvisna od kota Čerenkova in posledično merilo za hitrost delca.

Na podoben princip deluje tudi detektor brez zbiralnega sistema, v angleščini poimenovan "proximity-focusing RICH". Ker nima sferičnih zrcal, je manjši in bolj primeren za vgradnjo v sisteme, kjer je prostor omejen. Hitrost se meri delcu, ki se giblje skozi medij, v katerem seva – takemu mediju pravimo sevalec. Žarki sevanja Čerenkova nato potujejo do polja fotopomoževalk, kjer nastane slika obroča (shematičen prikaz na sliki 2). Ker je



Slika 2: Razlika v delovanju detektorja obročev Čerenkova s sistemom zrcal (levo) in brez zbiralnega sistema (desno); sevalec je označen s svetlo modro barvo, pot nabitega delca s polno črto in žarki sevanja s črtkano črto.

rezolucija hitrosti σ_β

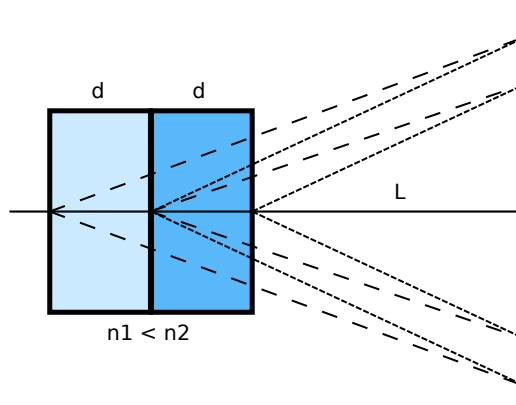
$$\frac{\sigma_\beta}{\beta} = \tan \vartheta \cdot \frac{\sigma_{\vartheta,1}}{\sqrt{N}} \quad (2.3)$$

odvisna od napake kota posameznega izmerjenega fotona sevanja $\sigma_{\vartheta,1}$ in števila izmerjenih fotonov N [3], želimo dolgo sevalno pot. Razlog se skriva v enačbi Frank-Tamm 2.2, ki podaja zvezo med sevalno potjo in številom izsevanih fotonov. Vendar daljša sevalna pot pomeni večjo negotovost točke izsevanja fotona in širino oz. razmazanost obroča. Da se izognemo slabši natančnosti pri določitvi radija, uporabimo sevalec, ki je sestavljen iz več plasti. Ob pravilni izbiri lomnega količnika tako dosežemo prekrivanje obročev, kot kaže slika 3. Pogoj za to je

$$(L + d) \tan \vartheta_1 = L \tan \vartheta_2 \quad ,$$

če ima plast sevalca širino d , razdalja med detektorsko ravnino in sevalcem pa je L . Z uporabo enačbe (2.1) za kot Čerenkova dobimo približno zvezo

$$n_2^2 = \frac{d}{L} \left(\frac{2}{\beta^2} + \frac{d}{L} \right) - \left(n_1 + \frac{d}{L} \right)^2 \quad . \quad (2.4)$$



Slika 3: Princip prekrivanja obročev z uporabo sevalcev z različnimi lomnimi količniki.

Slika 3 prikazuje primer z dvoplastnim sevalcem, vendar je plasti lahko več [7].

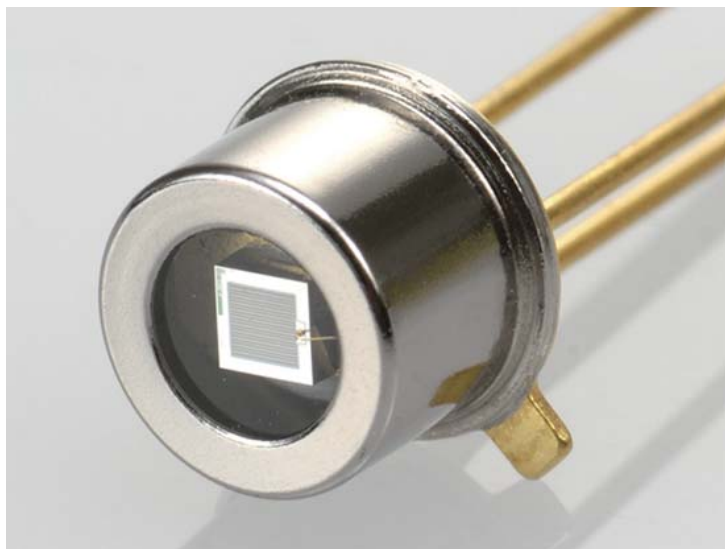
Zasnovo detektorjev brez zbiralnega sistema je omogočil napredek na področju izdelave aerogelov. Le-ti prinašajo veliko prostosti pri izbiri vrednosti lomnega količnika, saj pokrivajo področje med plini in kapljevina oz. trdninami. Prvi tak detektor je načrtovan za detektorski sistem *Belle II* [8], nadgradnjo *Belle* na trkalniku KEKB v Tsukubi na Japonskem.

2.3 Silicijeva fotopomnoževalka

Sliko obročev Čerenkova zaznamo s svetlobno občutljivim poljem, ki omogoča določitev prostorske koordinate zaznanih fotonov ter s tem radija obročev. Za polje, ki je sestavljeno iz večjega števila fotopomnoževalk, se je zaradi prisotnosti magnetnega polja pri eksperimentih pokazala želja po uporabi ti. silicijevih fotopomnoževalk (angl. “silicon” oz. “solid state photomultiplier”).

V primerjavi s klasično ima silicijeva fotopomnoževalka podoben izkoristek, a deluje pri precej nižji napetosti, le nekaj deset voltov [9]. Poleg tega se jo izdeluje z dobro razvito polprevodniško tehnologijo, kar pomeni potencialno nižjo ceno in manjše dimenzije. Vendar je glavna prednost pri uporabi v detektorjih neobčutljivost na magnetno polje, ki je v teh sistemih prisotno zaradi meritev gibalne količe. To omogoča enostavnejšo izvedbo detektorja in racionalnejšo porabo prostora.

Silicijevo fotopomnoževalko sestavlja skupek plazovnih fotodiod, povezanih v mrežo na silicijevi rezini. Plazovna fotodioda je polprevodniški element, na katerem incidenčni foton povzroči nastanek para elektron–vrzel. Ker je tok, ki ga povzroči en sam par, premajhen za merjenje, elektron oz. vr-



Slika 4: Silicijeva fotopomnoževalka

zel z električnim poljem dodatno pospešimo, da povzroči nadaljnjo tvorbo parov. Temu procesu pravimo ionizacijski plaz, saj tudi ti ponovno tvorijo nove pare. Za sproženje plazmu je dovolj en foton, kar pomeni, da takšna fotopomnoževalka lahko zaznava posamezne fotone.

Po drugi strani termične ekscitacije tudi povzročijo nastanek parov, ki jih zaznamo kot dogodke. Ampak ti dogodki niso pravi, ker jih ni povzročil foton. Rečemo jim temni dogodki, saj jih zaznamo tudi, ko je fotopomnoževalka v popolni temi in ni nikakršnih vpadnih fotonov nanjo. Pogostost temnih sunkov je pri sobni temperaturi precej velika; Dolgoshein [9] navaja red velikosti MHz/mm², a se z novejšimi fotopomnoževalkami manjša. Čeprav šum z nižanjem temperatura hitro pada, predstavlja veliko pomanjkljivost silicijevih fotopomnoževalk pri detekciji posameznih fotonov.

2.4 Časovna korelacija

Uporaba silicijevih fotopomnoževalk je mogoča za detekcijo obročev Čerenkova kljub pogostosti temnih sunkov [10]. Razloga sta dva. Prvi je, da za določitev velikosti obroča potrebujemo več fotonov, ki definirajo obroč. Tipično je njihovo število reda nekaj deset in nam en sam napačno zaznan dogodek ne pokvari slike obroča. Sledi, da napaka meritve ne bo bistveno večja, dokler bo število temnih dogodkov majhno v primerjavi s številom pravih.

Drugi razlog je čas, v katerem fotoni priletijo na detektorsko ravnino.

Do sevanja Čerenkova pride, ko je hitrost delca večja od hitrosti svetlobe v sevalcu. Ta je tipično reda svetlobne hitrosti v vakuumu, kar pomeni, da delec hitro preleti sevalec, saj je dolžina sevalca za detektor brez zbiralnega sistema le nekaj centimetrov. Obenem tudi žarki sevanja potujejo s približno enako hitrostjo, torej bo časovni interval, v katerem zaznamo fotone sevanja Čerenkova, zelo kratek. Z drugimi besedami, vsi relevantni dogodki se morajo zgoditi v tem časovnem okviru. Pogostost temnih dogodkov pa je v povprečju konstantna, medtem ko so pravi dejansko časovno precej omejeni. Tako večina temnih dogodkov pade izven pravega časovnega intervala in jih lahko zavržemo.

Časovno okno načeloma lahko določimo iz števila dogodkov – ob preletu delca zaznamo večje število fotonov kot sicer. Vendar imajo detektorji ponavadi namenske prožilne sisteme in ta podatek dobimo od zunaj. S tem omejimo vpliv temnih sunkov za ceno dodatnega določanja časa zaznave fotona. Ko upoštevamo pogostost temnih sunkov, ugotovimo, da za smiselne meritve potrebujemo časovne merilnike z ločljivostjo nekaj nanosekund [10].

3 Hitri časovni merilniki

Za zelo natančno merjenje časov uporabljamo namenske naprave, ki jim rečemo časovno digitalni pretvorniki (angl. “Time to Digital Converter” ali *TDC*). Glavne tehnične karakteristike, s katerimi ocenimo, kako dobri so takšni časovni merilniki so [11]:

merilno območje , ki nam pove, kako dolge časovne spremembe lahko merimo;

ločljivost je merilo za najmanjšo spremembo, ki jo merska naprava še zazna;

merska napaka imenujemo oceno efektivnega odmika σ ;

čas pretvorbe pove koliko časa preteče, da opravimo meritev;

mrtvi čas podaja čas za ponastavitev po končani predhodni meritvi, preden je naprava pripravljena za ponovno meritev;

nonlinearnost pa meri, koliko se resolucija spreminja po merilnem območju.

V fiziki visokih energij imamo opravka s hitrostmi blizu svetlobni. S tako hitrostjo delci preletijo več metrov v le nekaj deset nanosekundah. Vidimo, da so tudi časi, s katerimi se srečujemo, ustrezno kratki, zato potrebujemo merilnike z dovolj dobro ločljivostjo, kar precej omeji nabor primernih metod. Na srečo pa v večini teh primerov zadošča relativna meritev, torej meritev časovnih intervalov med dogodki, in se lahko izognemo težavam, ki bi jih sicer imeli z nastavljanjem pravega časa na merilniku.

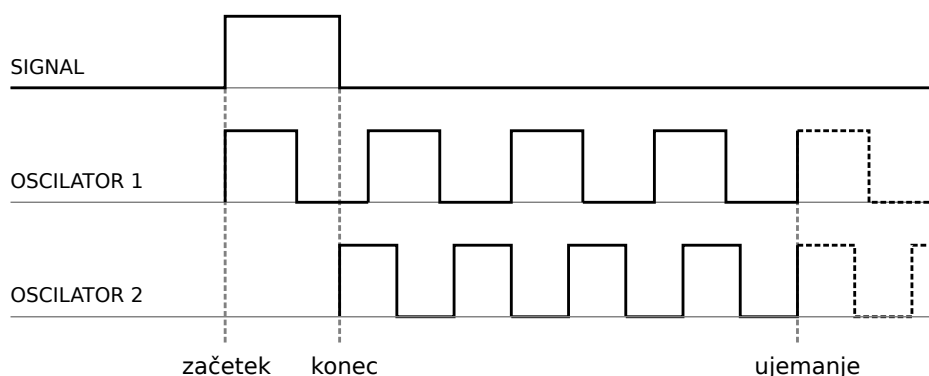
3.1 Metode merjenja z visoko ločljivostjo

Najpogosteje uporabljene metode zajemajo podaljševanje časa, meritev amplitude, Vernierjevo metodo in meritev na zakasnitveni liniji [11]. Metoda podaljševanja časa deluje tako, da za čas merjenega intervala polnimo kondenzator s tokom I_1 , nato pa ga izpraznimo s tokom I_2 . Zaradi linearne zveze med tokom in nabojem bo čas polnjenja proporcionalen času praznjenja. Če zahtevamo $I_2 < I_1$, se bo časovni interval raztegnil glede na razmerje I_1 in I_2 .

Podobno deluje meritev amplitude; najprej polnimo kondenzator, na katerem nato z analogno-digitalnim pretvornikom pomerimo napetost, ki je preko naboja odvisna od časa polnjenja. Čeprav lahko opisani metodi dosežeta veliko natančnost, nista preveč zaželeni, saj analogne komponente (kondenzator) lažje zmotijo zunanji vplivi, kot je temperatura. Poleg tega

imajo te metode daljši čas pretvorbe in jih je težje uporabiti v integriranih vezjih.

Vernierjeva metoda je dejansko digitalno podaljšanje časa z dvema oscilatorjema, katerih frekvenci sta blizu skupaj. Enega izmed njiju zaženemo ob signalu za začetek merjenja, drugega pa ob koncu. Ker frekvenci oscilatorjev nista enaki, se bo spreminjala faza med pozitivnima robovoma signala. Pozitivni rob signala je prehod iz spodnjega, neaktivnega (logična 0) v zgornje, aktivno stanje (logična 1). Zanima nas, kdaj se bosta pozitivna robova ujela, kot kaže slika 5. V tem primeru je število period n_1 in n_2 celoštevilsko.



Slika 5: Princip delovanja Vernierjeve metode

Čas med začetkom in koncem testnega signala lahko izrazimo s periodama oscilatorjev T_0^1 in T_0^2 kot

$$\begin{aligned} T &= n_1 T_0^1 - n_2 T_0^2 \\ &= n_1 (T_0^1 - T_0^2) + (n_1 - n_2) T_0^2 \quad . \end{aligned}$$

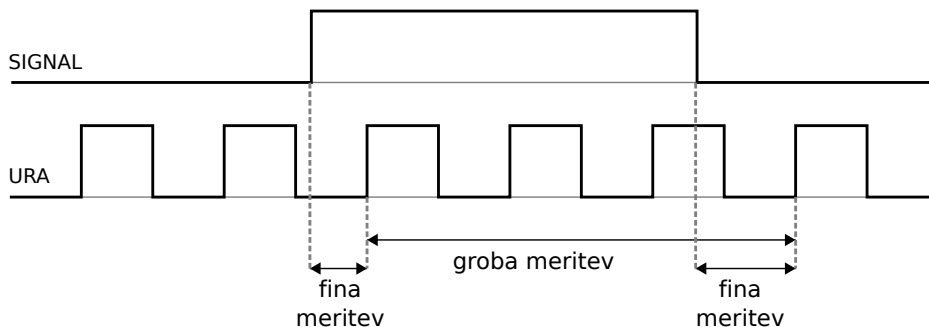
Če velja $T < T_0^1$, sta n_1 in n_2 enaka, sicer bi že prej imeli ujemanje; zadostuje uporaba le enega števca. Poleg tega drugi člen pade stran in dobimo odvisnost od razlike period, ki postane merska enota. Obenem pove tudi širino območja, ki definira ujemanje robov. Za tem večjo natančnost torej želimo čim bližji frekvenci oscilatorjev.

Delovanje zakasnitvene linije je po drugi strani konceptualno enostavno. Sestavljena je iz verige celic, pri prehodu katerih se signal zakasni. Meritev opravimo tako, da signal ob začetku intervala pošljemo skozi zakasnitveno linijo, ob koncu pa pogledamo, kako daleč (do katerega člena verige) je prišel.

Vendar je pri vseh naštetih metodah problem merilno območje, saj se tipično z njegovo velikostjo zmanjšuje natančnost metode. Še dodaten problem se pojavi pri zakasnitvenih linijah, kjer za dvakrat večje območje potrebujemo dvakrat več zakasnitvenih elementov, kar je precej potratno v primerjavi z

meritvijo s števcem. Slednji deluje po principu štetja ciklov referenčne ure in porabi samo en dodatni bit za podvojitev dolžine merilnega območja, a je ločljivost števca odvisna od frekvence ure, ki ga poganja. Za visoke frekvence (reda GHz in več) je namreč težko narediti stabilne generatorje ure in dovolj hitre števce.

Zaradi omenjenih razlogov se lahko poslužimo ti. metode interpolacije, pri kateri meritev opravimo s kombinacijo *grobe* in *fine* metode. Groba metoda meri daljše časovne intervale in prispeva veliko merilno območje, zaradi česar je dobro, da je čimbolj stabilna. Tem kriterijem dobro ustreza meritev s števcem. Boljšo ločljivost pa dosežemo z eno izmed ostalih, finih metod, ki sedaj ne potrebuje več velikega merilnega območja. Fino metodo uporabimo na začetku in na koncu merjenega intervala, kjer natančneje določimo oz. interpoliramo čas med dvema cikloma ure (slika 6). Potem je čas



Slika 6: Princip delovanja interpolacijske metode

$$T = T_f^z + T_g - T_f^k \quad .$$

T_f^z in T_f^k je čas fine meritve na začetku in koncu intervala, T_g pa grobe.

3.2 Tehnologija FPGA

Električno programabilno polje (angl. “Field-programmable gate array” oz. FPGA) je elektronska komponenta v obliki čipa, ki vsebuje veliko število logičnih vrat, katera lahko povežemo med seboj. Od namenskega integriranega vezja, ki mu določimo funkcijo že ob izdelavi, se FPGA razlikuje po tem, da izdelamo generični čip, ki ga je mogoče programirati tudi kasneje. Poleg logičnih vrat pogosto vsebujejo pomnilniške elemente, kot so npr. flip-flopi. Tako lahko takšna polja v veliki meri, če že ne v celoti, nadomestijo digitalne komponente na integriranem vezju. Seveda pa ne morejo nadomestiti analognih komponent.

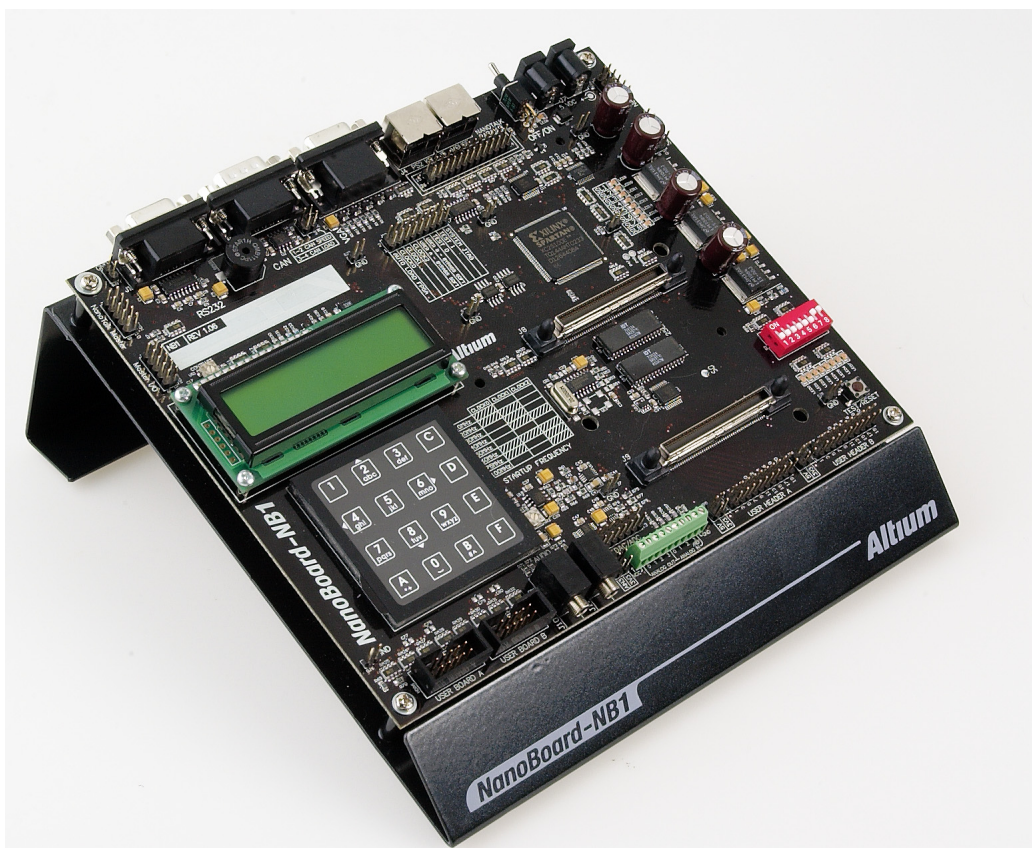
Glavna prednost čipov FPGA je hitrejši in lažji razvoj aplikacij, saj za vsako verzijo oz. popravek ni potrebna izdelava novega vezja, ampak se lahko le na novo prilagodi program. Podobno je mogoče spremeniti oz. nadgraditi njihovo delovanje na terenu, npr. v primeru poznejšega odkritja napak ali hroščev. Čeprav so namenska vezja bolj učinkovita [12] in v večini primerov cenejša za izdelavo v velikih količinah, pa so za omejene količine ponavadi cenejše izvedbe z električnimi programabilnimi polji, tudi zaradi nižjih stroškov razvoja.

Ravno zaradi velikega števila iteracij v razvoju je za časovne merilnike s finančnega vidika zanimiva uporaba čipov FPGA. V tem primeru se moramo omejiti pri izbiri metode merjenja, saj se sicer odrečemo prednostim, ki jih uporaba FPGA-jev ponuja. Zaradi analognih elementov kot so kondenzatorji in oscilatorji, ki bi jih potrebovali v vezju, večina metod merjenja z visoko ločljivostjo ni najbolj primerna. Za najbolj praktično se izkaže interpolacijska metoda z zakasnitveno linijo, ki omogoča skoraj poljubno veliko merilno območje, obenem pa se merilnik lahko v celoti implementira z elementi na čipu FPGA.

4 Izdelava merilnika

S prvenstvenim namenom ločevanja dogodkov iz silicijeve fotopomnoževalke sem zasnoval visokoločljivostni časovno digitalni pretvornik na tehnologiji FPGA, saj ta ponuja več svobode pri razvoju. Za vgradnjo v detektorske sisteme je sicer predvideno namensko integrirano vezje, vendar se za razvojne in testne namene lahko izognemo načrtovanju in izdelavi tega vezja z uporabo te razvojne platforme. To je že izdelano integrirano vezje, ki vsebuje najpogosteje uporabljene elektronske komponente in ponuja različne tipe vhodov in izhodov, kot je npr. analogni vhod preko analogno-digitalnega pretvornika.

Razvoj je tako potekal na platformi Altium NanoBoard NB1 (slika 7), saj je bila ta platforma na voljo in ima vgrajene vse potrebne elemente.



Slika 7: Razvojna platforma NanoBoard NB1

Upravljam jo s programom Altium Designer, s katerim komunicira preko paralelnih vrat računalnika. Altium Designer ponuja tudi integracijo z orodji

proizvajalcev čipov FPGA, ki so potrebni za njihovo programiranje, tako da so sheme v nadaljevanju narejene s tem programskim orodjem. Uporabljen čip je Xilinx Spartan-IIE, konkretno XC2S300E na sliki 8.



Slika 8: FPGA čip Spartan-IIE na razvojni platformi

Sam merilnik je v osnovi sestavljen iz več logičnih blokov. Srce sistema so merilni kanali, medtem ko komunikacija z zunanjim svetom poteka preko bralnega vezja, ki je neodvisno od merilnih kanalov. Skupno vezje poveže obe komponenti in poskrbi za skupne funkcije, kot je npr. povezava s fizičnimi vhodi in izhodi na razvojni platformi.

Merilni kanali delujejo po interpolacijski metodi, opisani v poglavju 3, vendar ne merijo časovnih intervalov, ampak le njihove začetke (prihod signala), saj je tak način merjenja bolj praktičen za naše potrebe. Tako namesto dolžine intervala pri grobi meritvi izmerimo trenutno stanje interne ure, s katero lahko nato določimo relativen čas glede na ostale dogodke.

Celoten sistem je zasnovan z mislijo na razširljivost, zaradi česar je dodajanje merilnih kanalov enostavno; edini del potreben spremembe v takem primeru je podsistem za izbiro kanala v bralnem vezju. Dejansko izdelana in v nadaljevanju predstavljena pa je različica z dvema kanaloma.

8. Čip FPGA

Merilnik za delovanje potrebuje enosmerno napajanje 5 V, ki ga po napajalnem kablu pripeljemo na del označen z (2). Na (4) se priklopi kabel za komunikacijo, na (3) pa najdemo vhode za signale, ki jih želimo meriti. Ti morajo biti kompatibilni z logičnimi vrednostmi po standardu LVTTTL [13]; napetost do 0,8 V predstavlja logično nič oz. stanje brez signala, 2 V in več pa logično ena oz. prisotnost signala.

Prikaz na statusnih LED diodah (6) lahko upravljamo s pomočjo kontrolnih stikal (7), a je takšna uporaba mišljena predvsem kot pomoč pri testiranju. Omogoča sicer enako funkcionalnost kot uporaba vmesnika LPT (4), vendar ni primerna za branje podatkov, saj je zaradi omejenosti z vhodi in izhodi enostavno preveč okorna. Avtomatiziranemu nadzoru in zajemanju podatkov je namenjen komunikacijski vmesnik LPT, ki je precej bolj priročen za vsakdanjo uporabo, seveda v kombinaciji z računalnikom oz. drugo kontrolno napravo.

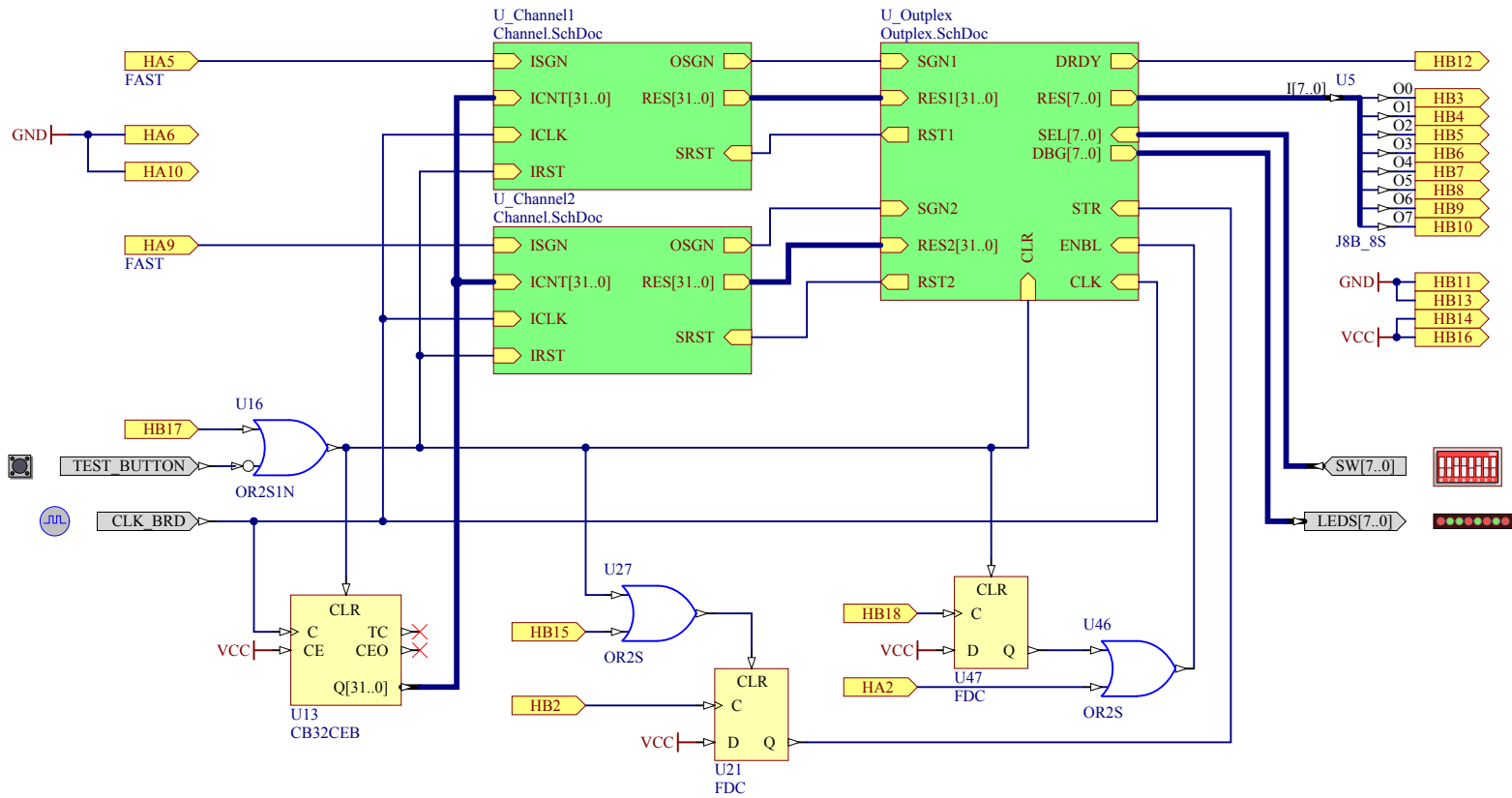
Za izvajanje meritev je potrebno napravo najprej ponastaviti, kar storimo z gumbom za ponastavitev (5) ali preko komunikacijskega vmesnika. S tem jo vrnemo v stanje, v katerem je pripravljena, da zabeleži prihod signala na enem izmed vhodnih kanalov. Ko se to dejansko zgodi, se čas zapiše v začasni pomnilnik in merilnik nam sporoči, da je na voljo nov podatek za branje. Obenem se pošlje tudi signal za ponastavitev kanalu, na katerem je do dogodka prišlo, ter se tako pripravi dotični kanal za ponovno meritev.

Pri branju nam naprava vrne 24-bitno informacijo o stanju internega števecja, ki predstavlja grobo meritev, ter 8-bitno informacijo o kanalu in fini meritvi. Če se v tem času zgodi nov dogodek, si ga naprava zapomni, dokler ima na voljo prostor v pomnilniku. Ta lahko v trenutni izvedbi shrani informacijo o 16 dogodkih, vsi nadaljnji pa se zavržejo. Ko podatek preberemo, se mesto v pomnilniku avtomatično sprosti in s tem omogoči napravi, da si zapomni naslednji dogodek. Merilnik uporablja ti. 16-mestni krožni medpomnilnik.

4.2 Skupno vezje

Celotna shema vezja, po kateri programiramo čip FPGA v merilni napravi, je precej kompleksna in prikaz na eni sliki je nepripraven in nepregleden. Program, s katerim je bila shema narejena, omogoča ustvarjanje lastnih gradnikov vezja. Tako so bili določeni deli združeni v logične enote oz. bloke po svoji namembnosti. Ti so na shemah predstavljeni kot zeleni pravokotniki, z imenom ob zgornji stranici. Taki primeri na sliki 10 so bralno vezje *U_Outplex* ter merilna kanala *U_Channel1* in *U_Channel2*, ki sta sicer

identična. Konkretno implementacije teh blokov se nahajajo v njim lastnih shemah, tista na sliki 10 pa torej predstavlja osnovni blok programa, ki določa delovanje čipa FPGA. Za namene identifikacije bomo ta blok v nadaljevanju imenovali *skupno vezje*; razlog je povezovalna funkcija, ki jo opravlja.



Slika 10: Shema implementacije merilnika (skupno vezje).

Rumene puščice z oznako HAx oz. HBx so vhodi in izhodi, ki so preko razvojne platforme NanoBoard speljani do zunanjih priključkov za merilne kanale (HAx) in vmesnik LPT za komunikacijo (HBx), ki je opisan v svojem podpoglavju 4.5. Signal za prvi merilni kanal je vezan na vhod HA5, pripadajoči HA6 pa je zvezan z zemljo in lahko preko njega ozemljimo kabel za prvi kanal, če je to potrebno (npr. za koaksialni kabel). Analogno drugemu kanalu pripada par HA9-HA10.

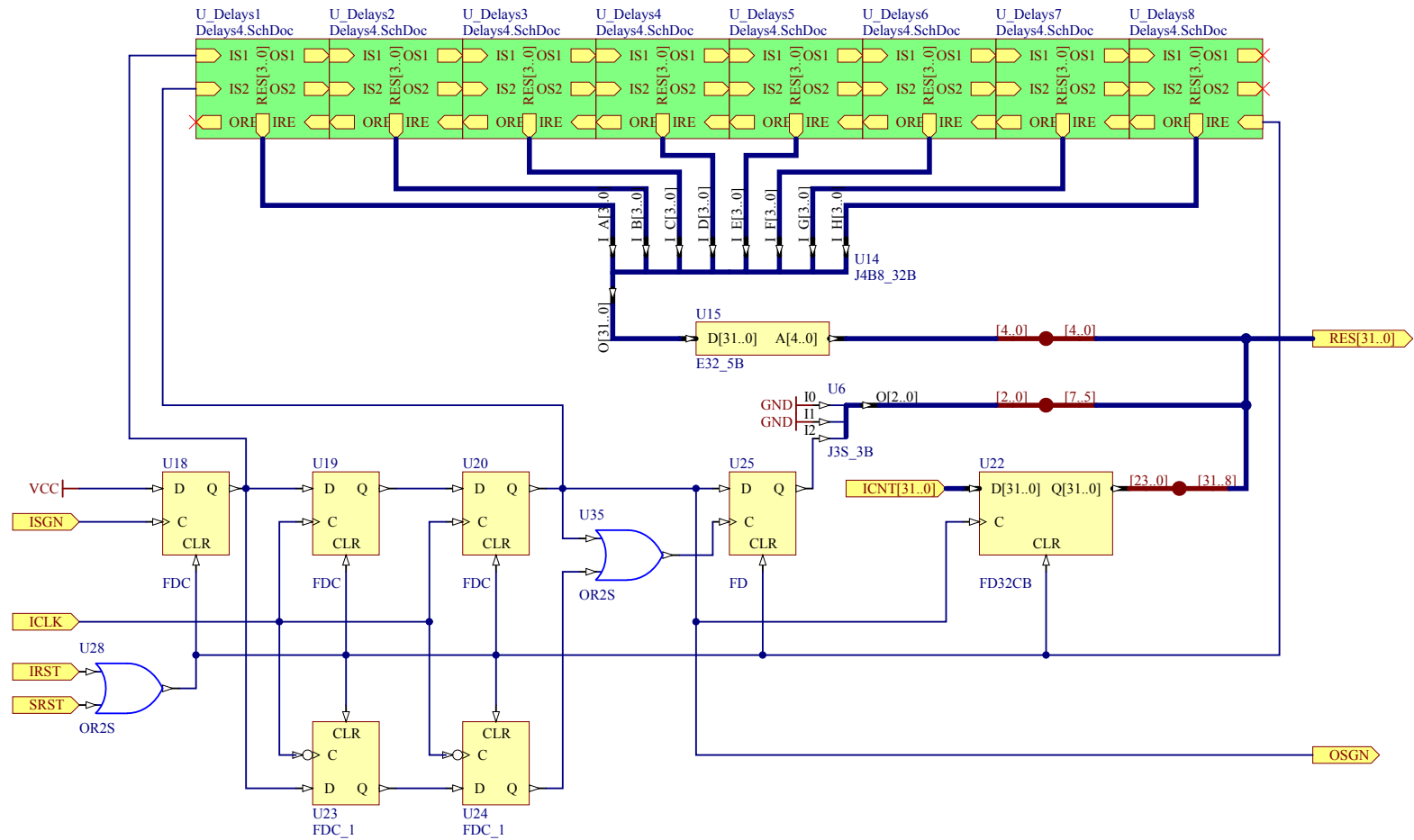
Poleg tega lahko z vhodom HA2 reguliramo, kdaj se podatki zapisujejo v interni pomnilnik (signal *ENABLE*). To pride prav v primerih, ko prihajajo dogodki v napravo hitreje, kot jih lahko beremo. Potem s signalom *ENABLE* izberemo časovno območje, ki nas zanima, merilnik pa bo v tem času izmeril toliko dogodkov, kolikor ima prostora v pomnilniku. Na ta način lahko za kratek čas merimo dogodke tako hitro, kot jih lahko merilnik izmeri, ne da bi bili omejeni s hitrostjo branja. V nasprotnem primeru, ko podatke sproti beremo, lahko nožico HA2 fizično povežemo z nožico HA0 na razvojni platformi in nam ni potrebno skrbeti za signal *ENABLE*.

Sive puščice s sličicami ob straneh tudi predstavljajo vhode in izhode, le da so ti povezani s kontrolnimi stikali (*SW*), statusnimi LED diodami (*LEDS*), gumbom za ponastavitev (*TEST_BUTTON*) in seveda sistemsko uro (*CLK_BRD*). Slednja se generira s pomočjo kvarčnega kristala na razvojni platformi in je nastavljena na frekvenco 200 MHz. Sistemska ura, poleg merilnih kanalov in bralnega vezja, poganja tudi števec, ki definira interno uro za grobo meritev. Njegov izhod je povezan na merilna kanala, katera si ob merjenem dogodku zapomnita njegovo vrednost in s tem čas dogodka. Števec je 32-biten in pri 200 MHz sistemski uri lahko meri čas dobrih 20 s, preden ponovno prične od začetka.

Omenimo še, da ima signal iz gumba negirano logiko – 0 ob pritisku, 1 sicer – in ga moramo zato obrniti, preden nam da signal za ponastavitev.

4.3 Merilni kanal

Pri uporabi merilnika za časovno korelacijo signalov iz silicijevih fotopomnoževalk zadošča določitev časa dogodka. To je razlog, da nam sedaj ni potrebno opravljati dveh finih meritev, na začetku in koncu časovnega intervala, ampak zadostuje le ena – tista ob prihodu signala. Poleg tega bi za izboljšanje ločljivosti lahko uporabili oba prehoda sistemske ure, tako pozitivni kot negativni rob, s čemer bi uri efektivno podvojili frekvenco. Vendar se izkaže, da to deluje, ko delamo grobo meritev, pri zakasnitvenih linijah pa zaradi tega izgubimo natančnost.



Slika 11: Merilni kanal, katerega naloga je sinhronizacija zakasnitvene linije (v zelenem) s sistemsko uro.

Logični blok na sliki 11, zadolžen za izvajanje meritev, sprejme na vhodu:

ISGN : signal, ki ga merimo,

ICNT : 32-bitni podatek o trenutnem času,

ICLK : sistemsko uro,

IRST, SRST : signal za ponastavitev,

na izhodu pa vrne:

OSGN : podatek o končani meritvi in

RES : izmerjeni čas dogodka (32 bitov).

Ob prihodu signala ISGN, tj. ob prehodu logične vrednosti iz 0 na 1, se ta najprej zapiše v flip-flop (U18), v izogib morebitnim problemom zaradi nihanja signala na vhodu. To je tudi znak za začetek fine meritve z zakasnitveno linijo (veriga komponent U_Delays).

Poleg tega se začne sinhronizacija s sistemsko uro; z elementom U19 znamo naslednji urin cikel. V teoriji je to dovolj in bi lahko s tem končali meritve z zakasnitveno linijo, vendar se izkaže, da je to problematično zaradi metastabilnosti flip-flopov [14]. Slednje je pojav, do katerega pride, če se podatkovni vhod flip-flopa (D) spremeni ob približno istem času, ob katerem pride signal na kontrolnem vhodu (C). Posledice metastabilnosti so daljši propagacijski časi oz. oscilacije na izhodu. Te omilimo z dvostopenjsko sinhronizacijo in tako prvemu flip-flopu sledi še drugi, U20.

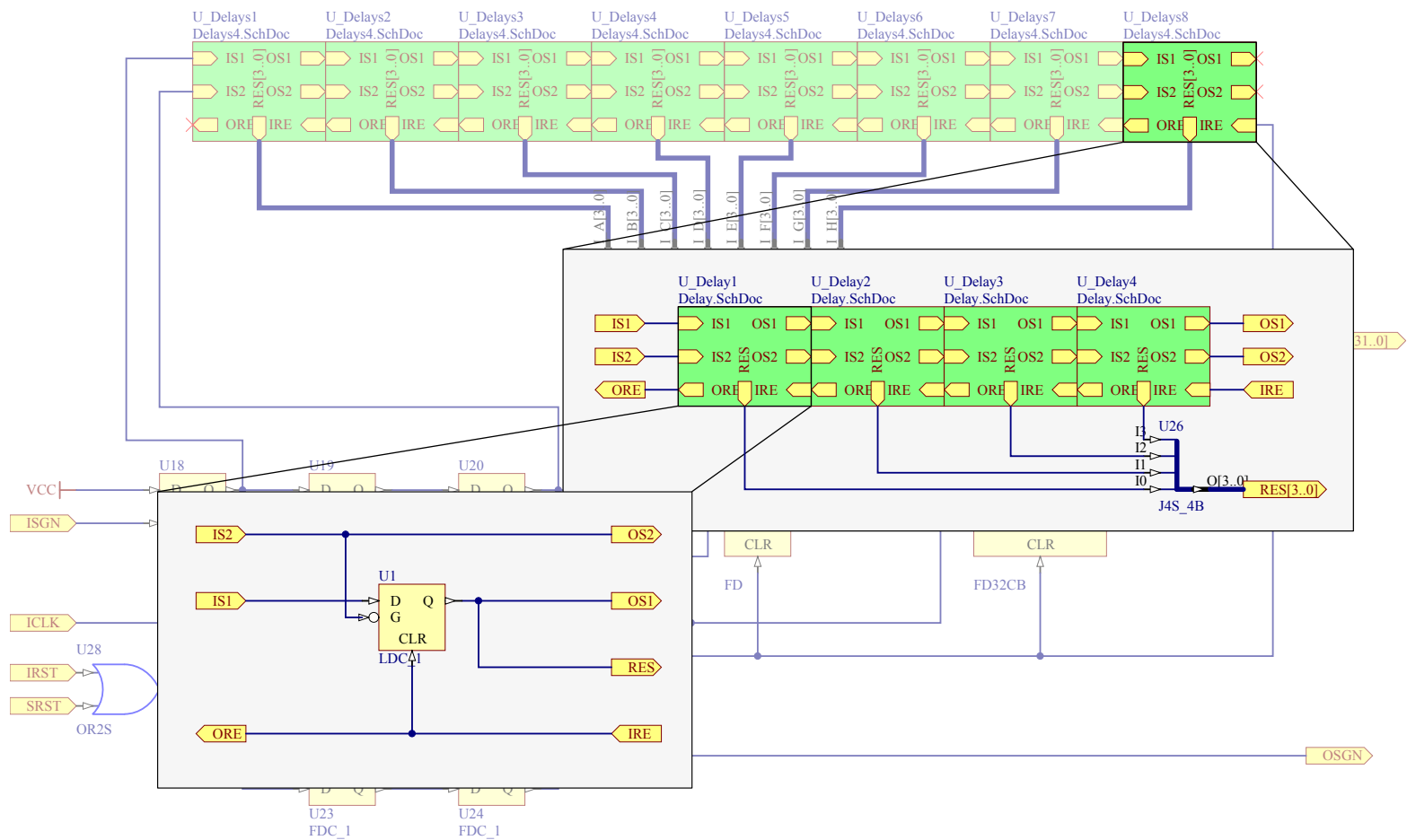
Vzporedno je postavljena še zrcalna sinhronizacijska linija (flip-flopa U23 in U24), ki se proži na negativni rob sistemske ure, preko katerega v U25 zapišemo podatek, po kateri poti je tekla sinhronizacija, in s tem definiramo *polovično vrednost* števca. To ni nič drugega kot dodaten bit ločljivosti, ki prispeva polovico merske enote števca.

Ob koncu sinhronizacije zapišemo trenutni interni čas v 32-bitni flip-flop U22 ter pošljemo signal zakasnitveni liniji za konec merjenja. Ta nam vrne podatek, do katere celice v liniji se je v tem času razširil začetni signal. Ker vsak izmed izhodov RES v liniji pove le, ali je signal prišel do pripadajoče celice, ta ti. termometerski zapis s prekodirnikom U15 pretvorimo v binarnega, ki predstavlja številko celice. Tega skupaj s podatkom o polovični vrednosti iz U25 in stanjem števca ob času meritve vrnemo na izhodu RES merilnega kanala.

Pozorni bralec lahko opazi, da se pri tem izgubi 8 bitov iz vhoda ICNT. S tem se zmanjša območje, ki ga pokriva števec, saj se le-ta prej obrne. V

splošnem to ni problematično, le merilno območje se zmanjša na ~ 80 ms. Za našo aplikacijo je takšno zmanjšanje sprejemljivo, saj precej poenostavi nadaljnjo obdelavo podatkov. V primeru potrebe po večjem merilnem območju pa je na voljo sorazmerno enostavna razširitev na 56-biten števec časa.

Sama zakasnitvena linija na sliki 12 deluje tako, da pošljemo začetni signal IS1 skozi verigo pomnilniških celic (U1). Glede na kontrolni signal G, izhod teh elementov sledi vhodu (G je 1) ali ostane nespremenjen (G je 0). V prvem primeru signal potrebuje nek določen propagacijski čas, da pride do izhoda OS1, kar uporabimo kot naš zakasnitveni element. IS2, ki je signal za konec meritve, pa postavi G na 0 in tako zamrzne stanje, da ga lahko kasneje preberemo. Linija na shemi 11 je razdeljena na osem enot, ki jih sestavljajo po štiri osnovne celice.



Slika 12: Zakasnitvena linija in celice, ki jo sestavljajo; signal IS1 se razširja skozi elemente U1, dokler ne pride IS2.

4.4 Bralno vezje

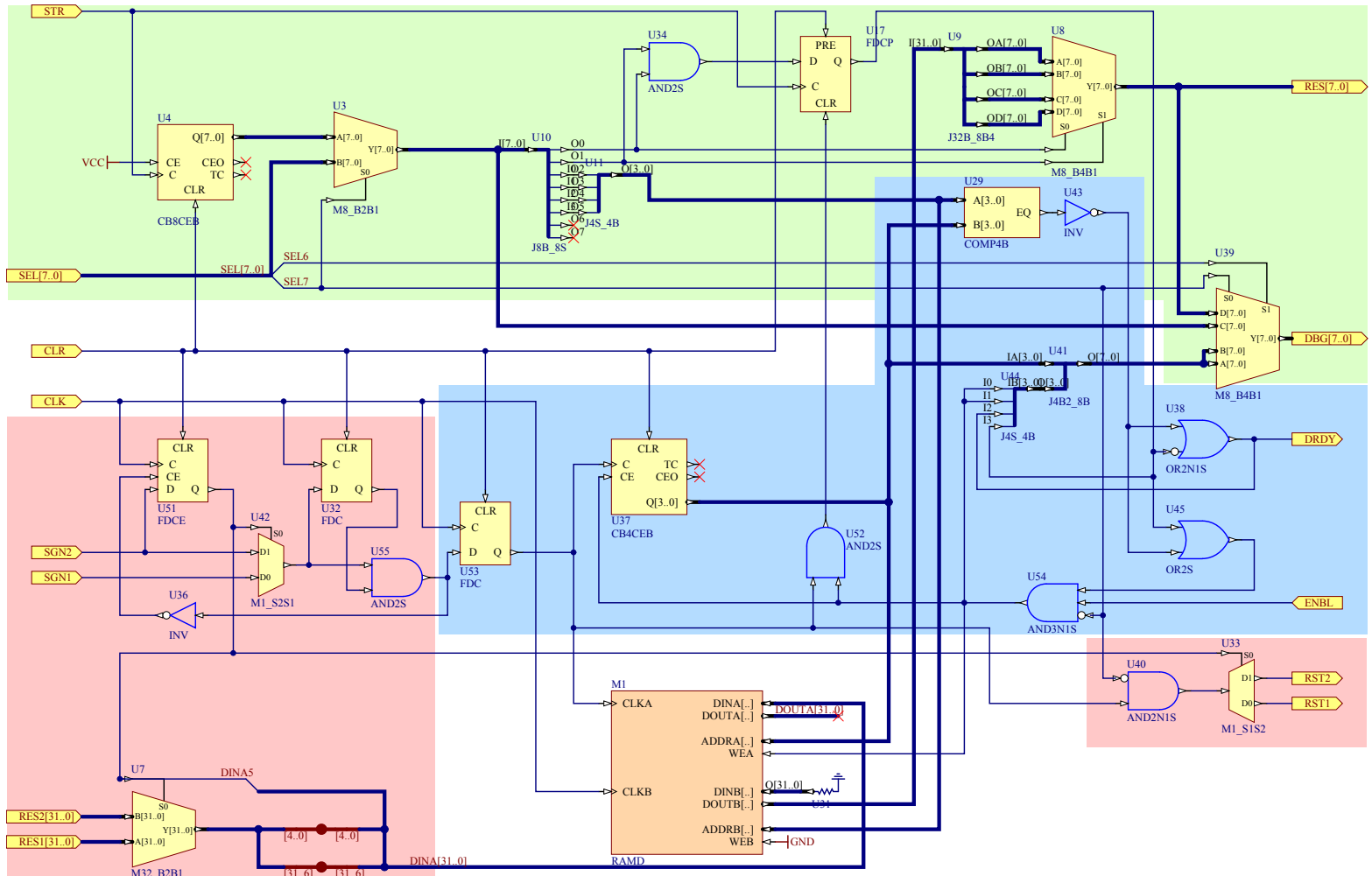
Namen tega dela vezja je uporabniku dostaviti podatke, pridobljene z merilnimi kanali. Pri tem je potrebno upoštevati določene omejitve:

- velikost podatka je lahko poljubna, podatkovno vodilo pa ima omejeno širino,
- na voljo je več kanalov, katerih podatke je potrebno zbrati,
- branje podatkov je lahko počasnejše od zajemanja.

Prva točka opisuje problem ti. serializacije. Rešitev je sistem komunikacije, kjer damo na podatkovno vodilo del podatka. Ko druga stran konča z branjem, pošlje signal, ki ga bomo imenovali *STROBE* in je znak, da damo na podatkovno vodilo naslednji del podatka. Postopek se ponovi, dokler se ne prenese celotnega podatka, ki ga lahko druga stran rekonstruira iz prebranih delov. Enostavna implementacija takega načina prenosa podatkov je s števcem, ki ga poganja *STROBE* signal. 32-bitni podatek moramo za 8-bitno vodilo razdeliti na štiri dele in torej potrebujemo 2-bitni števec *STROBE* signalov. Prednost tega pristopa je, da za naslednji podatek števca ni potrebno ponastavljati, saj se po štirih signalih sam vrne na nič. To je razlog, da smo se pri merilnih kanalih omejili na podatke velikosti $8 \cdot 2^n$ bitov.

Enako pot lahko uberemo za problem pod drugo točko, a ima tak pristop pomanjkljivost; z brezglavim branjem vseh kanalov, tudi tistih, kjer ni bilo dogodkov, le tratimo čas. Ker z večanjem števila kanalov raste tudi čas branja, bi se lahko na ta način po nepotrebnem omejili glede števila možnih kanalov. Poleg tega bi se temu vsekakor radi izognili za primere, ko je ozko grlo komunikacija z merilnikom. Boljša rešitev, ki obenem pokrije tudi primer iz zadnje točke, je uporaba začasnega pomnilnika. Vanj spravljamo dogodke in tako beremo le relevantne podatke. Pri tem smo omejeni z velikostjo, kar rešimo z uporabo krožnega medpomnilnika, vendar ne želimo, da se prepíšejo stare vrednosti, ki še niso prebrane. Na žalost to zadnje precej prispeva h kompleksnosti njegove implementacije.

Bralno vezje s sheme na sliki 13 v grobem opravlja tri funkcije: izbiro kanala, branje iz pomnilnika in pisanje v pomnilnik. Merilni kanal, na katerem čaka podatek, izberemo z elementom U51, vendar šele potem, ko smo se prepričali, da je zamenjava kanala varna. Za to poskrbimo z dodatnim flip-flopom (U32), ki zakasni signal in ne pusti zamenjave, dokler je trenutni kanal aktiven. V nasprotnem primeru bi lahko poslali signal za ponastavitev (izhod RSTx) na napačen kanal. Prave vhode in izhode nastavimo z multiplekserji U33, U42 in U7 za signale RST, SGN in RES. Zadnjemu preko bita 5 dodamo še informacijo o kanalu, na katerem je bila meritev opravljena.



Slika 13: Bralno vezje, razdeljeno po funkcionalnosti; izbira kanala (rdeče), bralni (zeleno) in pisalni del (modro).

4.4.1 Branje in pisanje

Števec U4 je namenjen štetju signalov STROBE na vhodu STR in preko najnižjih dveh bitov izvaja serializacijo pri branju (elementa U8 in U9). V tem primeru se podatek bere od najmanj do najbolj pomembnega dela, po ti. sistemu "little-endian". Ker je velikost vsakega posameznega podatka 32 bitov, lahko porabimo isti števec tudi za naslavljanje pomnilnika M1 za branje. Tako naslednji štirje biti predstavljajo naslov, na katerem se nahaja trenutni podatek, zadnja dva pa brez škode zavržemo, saj je pomnilnik le 16 mesten.

Tega ne storimo pri signalu iz kontrolnih stikal (vhod SEL) in zadnja dva bita raje porabimo za dodatne funkcije. S SEL7 izbiramo, ali želimo ročno nastaviti izhod (U3), s SEL6 pa funkcijo statusnih LED diod.

Pisanje v pomnilnik se izvede v več korakih. Po izbiri kanala preverimo, če je le-ta še vedno aktiven (U55) in v naslednjem ciklu podatek zapišemo v pomnilnik. Tako pisanje kot tudi povečanje števca U37, ki premakne kazalec za pisanje eno mesto naprej v pomnilniku, se zgodi le v primeru, ko ima pomnilnik še na razpolago prostor, kamor smemo pisati. V vsakem primeru pa se pošlje signal za ponastavitev kanalu, od koder smo ravnokar prebrali podatek.

Ob vsakem zaznanem dogodku ponastavimo tudi flip-flop U17, ki hrani informacijo o tem, ali smo nazadnje brali ali pisali. Njegovo logično vrednost bomo imenovali *RD* in dejansko pomeni, da smo ob zadnjem signalu STROBE prebrali celoten podatek. Dokler je aktiven pove, da po branju nismo več pisali v pomnilnik, saj bi ga v nasprotnem primeru ponastavili.

Skupaj s komparatorjem kazalcev za branje in pisanje U29, z logično vrednostjo *EQ*, RD podaja informacijo o tem, ali je pomnilnik poln in vanj torej ne smemo več pisati. Poleg tega to pove, če so v pomnilniku na voljo podatki za branje. Resničnostna tabela 2 navaja vrednosti RD in EQ ter njun pomen.

RD	EQ	smemo pisati	podatki na voljo
0	0	1	1
1	0	1	1
0	1	0	1
1	1	1	0

Tabela 2: Resničnostna tabela, ki pove, če smemo pisati v pomnilnik oz. če so v njem na voljo podatki.

V krožni medpomnilnik smemo pisati v dveh primerih; kadar sta kazalca

za branje in pisanje različna oz. kadar sta enaka, a je bila zadnja operacija branje. Drugi primer pomeni, da smo prebrali vse podatke iz medpomnilnika in je le-ta prazen. Matematični zapis obeh pogojev je

$$\neg EQ \vee RD \quad .$$

Podobno velja, da so podatki na voljo, ko sta kazalca različna, saj še nismo prebrali vsega, kar smo zapisali v medpomnilnik. Če sta enaka, pa smo morali nazadnje pisati, kar lahko izrazimo kot

$$\neg EQ \vee \neg RD \quad .$$

Obe gornji funkciji implementirajo vrata U45 – smemo pisati; in vrata U38 – podatki so na voljo.

4.5 Povezava z računalnikom

Zajemanje podatkov iz pretvornika poteka preko paralelnega vmesnika. Vmesnik ima 8-bitno podatkovno vodilo in omogoča tudi nadzor merilne naprave. Povežemo ga lahko s paralelnim vhodom na osebni računalniku. V tem primeru vhode in izhode HBx na shemi 10 povežemo s signali na konektorju IEEE 1284-A [15] (tabela 3).

Vhod/izhod FPGA	Signal IEEE 1284	Funkcija
HB2	nStrobe	Signal STROBE
HB3 - HB10	Data 1 - 8	Podatkovno vodilo
HB11	nAck	Ni v uporabi
HB12	Busy	Podatki na voljo
HB13	PError	Ni v uporabi
HB14	Select	Ni v uporabi
HB15	nAutoFd	Signal STROBE_ENABLE
HB16	nFault	Ni v uporabi
HB17	nInit	Ponastavitev naprave
HB18	nSelectIn	Signal ENABLE

Tabela 3: Povezava vhodov in izhodov čipa FPGA s signali na komunikacijskem kablu IEEE 1284-A.

Ti signali so na računalniku dostopni na določenih naslovih, ki so odvisni od uporabljenih paralelnih vrat. Posamezna vrata identificiramo z osnovnim naslovom, *BASEADDR*. Njegove tipične vrednosti so 0x378 in 0x278, kjer

notacija števil z 0x predstavlja heksadecimalni zapis. Točno na naslovu BASEADDR se nahaja podatkovni register, kateremu sledita najprej statusni in nato še kontrolni. Vsi trije so 8 bitni, zvezo med njihovi biti in posameznimi signali pa podaja tabela 4 [16]. Kot nakazujejo že njihova imena,

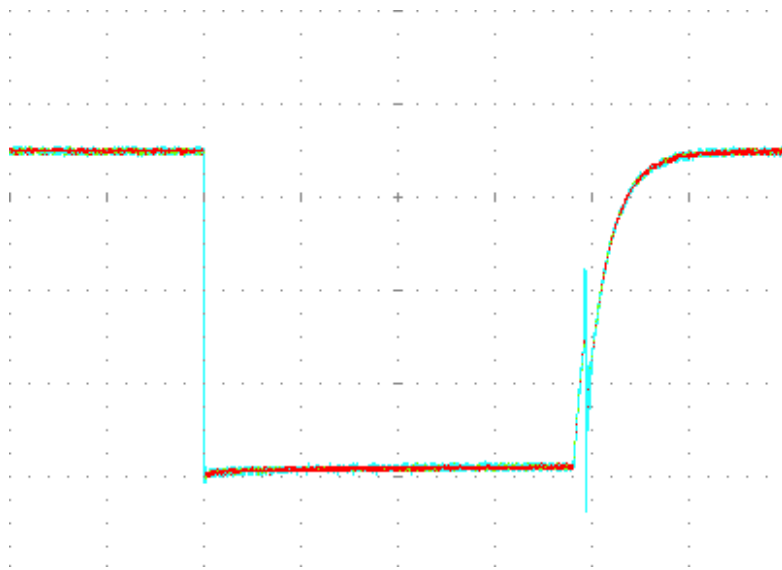
Register	podatkovni	statusni	kontrolni
Bit 7	Data 8	\neg Busy	x
Bit 6	Data 7	nAck	x
Bit 5	Data 6	PError	x
Bit 4	Data 5	Select	x
Bit 3	Data 4	nFault	\neg nSelectIn
Bit 2	Data 3	x	nInit
Bit 1	Data 2	x	\neg nAutoFd
Bit 0	Data 1	x	\neg nStrobe
Naslov	BASEADDR	BASEADDR + 1	BASEADDR + 2

Tabela 4: Signali v registrih paralelnih vrat; x pomeni brez povezave.

iz statusnega registra beremo status naprav, s kontrolnim jih upravljamo, preko podatkovnega pa prenašamo podatke. Privzeta funkcija podatkovnega registra je pošiljanje podatkov iz računalnika, saj je bil vmesnik LPT tipično namenjen komunikaciji s tiskalniki. Sprejem podatkov na podatkovnem registru je tako potrebno še dodatno nastaviti, kar v večini primerov dosežemo z nastavljanjem bita 5 v kontrolnem registru. Poleg tega so nekateri signali na fizičnih vratih dejansko invertirani; pri statusnem registru je tak le Busy (bit 7), medtem ko je invertiran celoten kontrolni register, z izjemo signala nInit (bit 2). V tabeli 4 smo jih označili z znakom \neg pred imenom.

To pomeni, da, dokler je bit 7 v statusnem registru (Busy) aktiven, še nismo uspeli izmeriti nobenega dogodka in vsebina podatkovnega registra ni definirana. Ko postane omenjeni bit 0, so podatki o meritvah na voljo in lahko začnemo z branjem iz podatkovnega registra. S signalom STROBE nato napravi povemo, da pošlje nov del podatkov na podatkovno vodilo. Vendar enostavna signalizacija ne zadostuje, ker se ob spremembi vrednosti bita na vratih lahko na komunikacijskem kablu pojavi oscilacija signala (slika 14). Ta se sicer hitro izniha, vendar je števec na merilniku dovolj občutljiv, da jo zazna kot večkratni signal. V izogib omenjenemu problemu se za aktivacijo signala STROBE najprej nastavi bit 1 v kontrolnem registru za STROBE_ENABLE, šele nato se pošlje STROBE z nastavljanjem bita 0 na 0 (nStrobe je namreč invertiran).

Bit 2 kontrolnega registra za ponastavitev naprave pa takšne dvostopenj-



Slika 14: Oscilacija na komunikacijskem kablu ob spremembi logičnega stanja.

ske signalizacije ne potrebuje, saj je pomembno le njegovo stanje in ne število prehodov med njimi. Primer izvorne kode programa za uporabo vmesnika po tej shemi je priložen v dodatku A.

5 Umeritev in karakteristike merilnika

Pri implementaciji zakasnitvenih linij v merilnih kanalih smo uporabili identične osnovne celice. Naivno bi potem pričakovali, da bodo tudi zakasnitve pri prehodu signala skozi celice enake. Izkaže se, da temu ni tako. Glavni razlog je interna struktura čipov FPGA, ki so sestavljeni iz logičnih blokov. Zakasnitve signala ob prehodu med takimi bloki so lahko nekajkrat večje kot znotraj bloka [17]. Obenem so lahko izvor variabilne zakasnitve sami logični elementi in njihova postavitve, saj orodja za programiranje FPGA izvajajo določeno optimizacijo, nad katero nimamo veliko vpliva. Včasih ima lahko celo invertiran signal drugačno zakasnitev kot neinvertiran.

Različno dolžino zakasnitev in posledično širino predalčka, v katerega meritev pade, imenujemo *diferencialna nelinearnost* [11]

$$l_i = \frac{p_i - p_0}{p_0}$$

in je definirana preko verjetnosti p_i , da naključni dogodek pade v i -ti predalček. p_0 je verjetnost v primeru N enako širokih predalčkov in je $\frac{1}{N}$. Za vsak predalček definiramo tudi *integralno nelinearnost* kot

$$L_j = \sum_{i=1}^j \frac{l_i}{N} \quad ,$$

ki je merilo za napako i -tega predalčka zaradi nelinearnosti. Napaka meritve v zakasnitveni liniji je torej lahko precejšnja, saj je seštevek nelinearnosti vsake izmed predhodnih zakasnitvenih celic oz. predalčka, ki ga ta celica definira.

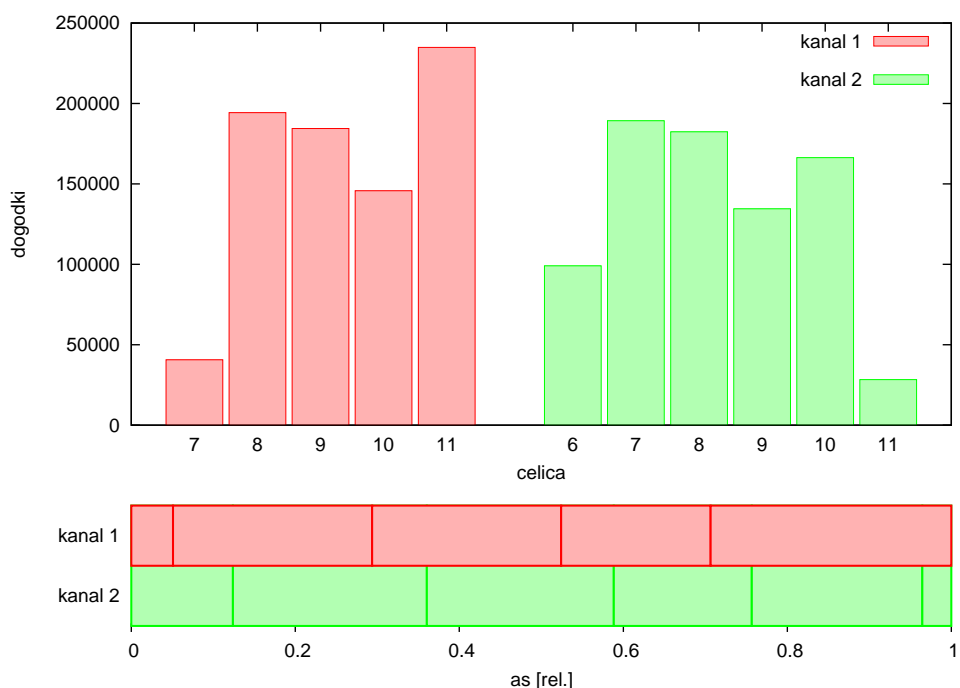
Integralno nelinearnost lahko zmanjšamo, če izmerimo dolžino zakasnitve na vsaki celici linije in s tem podatkom umerimo merilni kanal. Uporabnost tega pristopa omeji dejstvo, da je zakasnitev odvisna tudi od zunanjih dejavnikov, predvsem temperature in napajalne napetosti [17]. Tako ne moremo umeriti merilnika enkrat za vselej, ampak je za optimalno meritev načeloma potrebno nenehno umerjanje. Kljub temu je opazna bistvena razlika že ob enkratni umeritvi, saj velik del nelinearnosti prispevajo prehodi med logičnimi bloki znotraj FPGA čipa, ki so prisotni ne glede na zunanje vplive.

5.1 Statistična umeritev

Zakasnitev na celicah se lahko enostavno določi z meritvijo porazdelitve naključnih dogodkov. Ker je verjetnost za naključni dogodek konstantna v času, bo število dogodkov za vsako celico sorazmerno relativni zakasnitvi na njej.

Načeloma niti ne potrebujemo pravih naključnih dogodkov, dovolj je že, da ti niso korelirani s sistemsko uro merilnika. To pomeni, da v večini primerov za umerjanje lahko dejansko uporabimo kar samo meritev. To je sicer zelo praktično, a se moramo zavedati, da pada napaka s korenem števila dogodkov, torej je za dobro meritev potrebno zajeti zadostno količino podatkov. Seveda pa tak pristop ne upošteva nihanj temperature in ostalih zunanjih dejavnikov v času merjenja.

Očitno je, da je potrebno narediti umeritev za vsak kanal posebej, saj ima vsak svojo zakasnitveno linijo. Sam postopek ni zapleten, ko enkrat že imamo porazdelitev dogodkov po celicah in s tem relativne zakasnitve (slika 15). Najboljšo oceno časa dogodka dobimo, če vsaki zakasnitveni celici



Slika 15: Število zaznanih dogodkov po zakasnitvenih celicah (zgoraj) in delitev urinega cikla na manjše intervale za vsakega izmed kanalov (spodaj).

pripišemo vsoto zakasnitev celic pred njo in polovico lastne zakasnitve. Tako dobimo umeritveni koeficient

$$C_i = \sum_{j=1}^{i-1} p_j + \frac{p_i}{2}$$

za i -to celico. Dejanske zakasnitve dobimo, ko relativne pomnožimo z dolžino časovnega intervala t_0 , ki ga zakasnitvena linija meri. Tako nam meritev na

i -ti celici pove, da je signal prišel pred $C_i \cdot t_0$ časa. Za dani merilnik je $t_0 = 5$ ns.

Poleg meritve zakasnitvene linije dobimo iz naprave tudi podatek o polovičnem času systemske ure oz. podvojene frekvence, a se izkaže, da nam ne koristi veliko. Glavni razlog je, da nam razdeli časovni interval na dva dela, potem ko smo ga že fineje razdelili z zakasnitveno linijo. Resda razdeli eno izmed celic in s tem efektivno poveča njihovo število za ena, vendar nam ločljivost omejuje najdaljša zakasnitev v liniji. V tem primeru ta ostaja enaka, saj vedno razdelimo le sredinsko celico, ki pa tu ni najšibkejši člen.

Načeloma bi še vedno lahko uporabili podvojeno frekvenco za zmanjšanje časa trajanja meritve. Zaradi razlik v razširjanju med normalnim in invertiranim signalom pa v bistvu izgubimo na ločljivosti in se nam to ne izplača. Temu bi se verjetno dalo izogniti z ločeno zakasnitveno linijo za invertirano logiko, ampak večje in kompleksnejše vezje težko upraviči takšno minimalno prednost.

5.2 Karakteristike naprave

V poglavju 3 smo našli nekatere lastnosti, pomembne za časovne merilnike, ki bi jih sedaj radi določili tudi za predstavljeno napravo. Verjetno najpomembnejša lastnost je ločljivost, ki jo za časovno digitalne pretvornike z metodo interpolacije določimo preko merske napake σ [11]. Razlog je, da zaradi nelinearnosti zakasnitvena linija z 10 elementi nima nujno merske napake sorazmerne z $\frac{1}{10}t_0$, ampak je lahko večja, saj je natančnost meritve odvisna od celice, s katero je bil dogodek zaznan.

Ker smo merilne kanale umerili, nelinearnost pogojuje najdaljša zakasnitev v liniji. Pri gornjem umerjanju lahko vidimo, da znaša 1,47 ~~nanosekunde~~. Merilno območje pri naši implementaciji zavisi od velikosti števca in je dolgo dobrih 83 ms. Ena izmed prednosti metode interpolacije je, da razširitev merilnega območja ni problematična in se izvede z razširitvijo števca, ki ga lahko enostavno zaporedno povežemo z drugim.

Čas pretvorbe je enak času sinhronizacije in je torej odvisen od systemske ure; v najboljšem primeru traja en, v najslabšem pa dva urina cikla, torej 5–10 ns. Mrtvi čas je po drugi strani odvisen od bralnega vezja in posameznega kanala. Tu je ozko grlo bralno vezje, ki zbira podatke z več kanalov. Če prihajajo dogodki le na en kanal, si morajo slediti v razmaku vsaj 30 ns (6 ciklov ure) za prvi in 40 ns (8 ciklov) za drugi kanal, če želimo zaznati vse. Do razlike pride zaradi preklapljanja med kanali in se ujema s številom ciklov, ki jih vezje potrebuje, da zapiše podatek. Ta čas je sicer lahko daljši, če mora kanal počakati na vrsto. V praksi je prej ozko grlo komunikacija z zunanjim svetom in čakanje ni kritično; v nasprotnem primeru bi bilo mogoče dodati

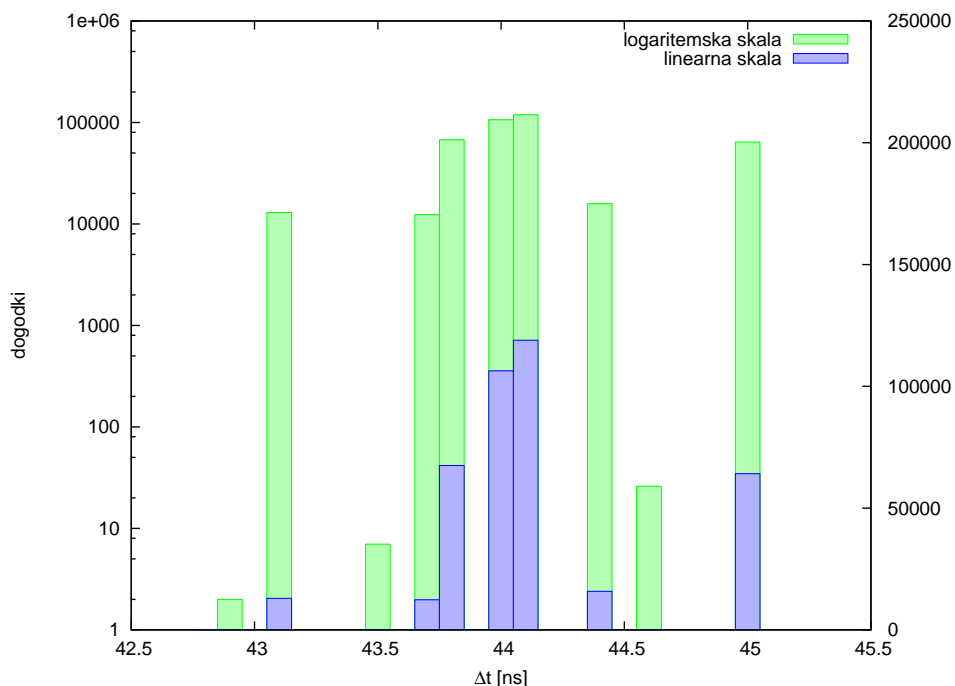
medpomnilnik direktno na kanal in tako doseči čas 30 ns oz. 6 ciklov v vseh primerih.

5.2.1 Ločljivost

Za namen določanja ločljivosti oz. merske napake sem meril dva generirana dogodka v razmaku 44,2 ns, ki sem ju približno vsakih 40 μ s poslal na prvi kanal, z zakasnitvijo 138,7 ns pa še na drugega. Na skoraj 400.000 dogodkih sem izmeril časovne razlike, predstavljene v tabeli 5 in na sliki 16. Razlika

	Časovna razlika [ns]	σ
med dogodkoma na kanalu 1	44,34	0,59
med dogodkoma na kanalu 2	44,13	0,43
med 1. dogodkom na kanalih	138,07	0,46
med 2. dogodkom na kanalih	137,86	0,43

Tabela 5: Izmerjene časovne razlike med testnimi dogodki



Slika 16: Čas med dvema zaporednima dogodkoma na drugem kanalu

med dogodkoma se za oba kanala dobro ujema z vrednostjo, izmerjeno z osciloskopom, kar pomeni, da je ločljivost merilnika pod 0,60 ns.

Luknje med diskretnimi vrednostmi so sicer posledica dejstva, da smo grafirali vrednosti, razdrobljene v predalčke velikosti $0,1\text{ ns}$, ki so manjši od velikosti posameznih zakasnitev. Grupiranje po predalčkih je potrebno, ker v splošnem dobimo 36 diskretnih vrednosti (kvadrat šestih za ta kanal), dejansko pa še več zaradi omejene natančnosti pri računanju. Visoke vrednosti na robu intervala lahko pripišemo celicam z dolgo zakasnitvijo. Pri teh je verjetnost, da na njih izmerimo dogodek, največja in obenem dajo največjo napako oz. odstopanje.

Napaka na posamezni celici je $\frac{1}{\sqrt{12}}p_i \cdot t_0$, saj je verjetnostna porazdelitev za izmerek znotraj celice enakomerna, zunaj pa enaka 0. Efektivno napako celotne zakasnitvene linije σ_l lahko tako zapišemo z vsoto po celicah:

$$\begin{aligned}\sigma_l^2 &= \int p(x)x^2 dx - \left(\int p(x)x dx \right)^2 \\ &= \sum_i p_i \frac{1}{p_i} \left[\int p_i x^2 dx - \left(\int p_i x dx \right)^2 \right] \\ &= \sum_i p_i \cdot \sigma_i^2 \quad .\end{aligned}$$

Podobno obliko ima tudi izraz za napako meritve med kanaloma, kjer je potrebno dodatno upoštevati, da se meritev izvaja na dveh linijah. Ker sta liniji neodvisni, napaki na posameznih celicah seštevamo v kvadratu, vsoto pa obtežimo z verjetnostjo, da je bila meritev opravljena na danih celicah;

$$\sigma_k^2 = \sum_i \sum_j p_i^I p_j^{II} \left(\frac{(p_i^I)^2}{12} + \frac{(p_j^{II})^2}{12} \right) \quad .$$

Iz tabele 6 razberemo, da se v tem primeru izračunana vrednost (0,45) lepo ujema z izmerjeno (0,43 oz. 0,46 iz tabele 5).

Pri merjenju razlike med dogodkom na obeh kanalih, tj. časovni razliki med kanaloma ($\approx 138\text{ ns}$), ta odstopa od prave vrednosti $138,7\text{ ns}$ več, kot je efektivni odmik meritve σ_k . Sklepamo lahko, da je med kanaloma slabo nanosekundo časovne razlike, ki je verjetno posledica postavitve merilnih kanalov znotraj čipa FPGA in različnih poti, po katerih so speljani vhodi NanoBoard-a na čip. Takšna relativna razlika nas morda niti ne moti; če je potrebno, pa s tem podatkom umerimo meritve iz obeh kanalov.

5.3 Preizkus s silicijevo fotopomnoževalko

Motivacija za izdelavo časovnega merilnika je bilo ločevanje med signali pravih in temnih dogodkov iz silicijeve fotopomnoževalke. V primeru, da je intenziteta vpadne svetlobe velika, na izhodu dobimo signal, ki ustreza večjemu

i	σ_i^I	σ_i^{II}
6	-	0.07994
7	0.03281	0.15274
8	0.15680	0.14717
9	0.14878	0.10857
10	0.11758	0.13423
11	0.18951	0.02285
linija (skupno)	0.34780	0.29152
med kanaloma	0.45382	

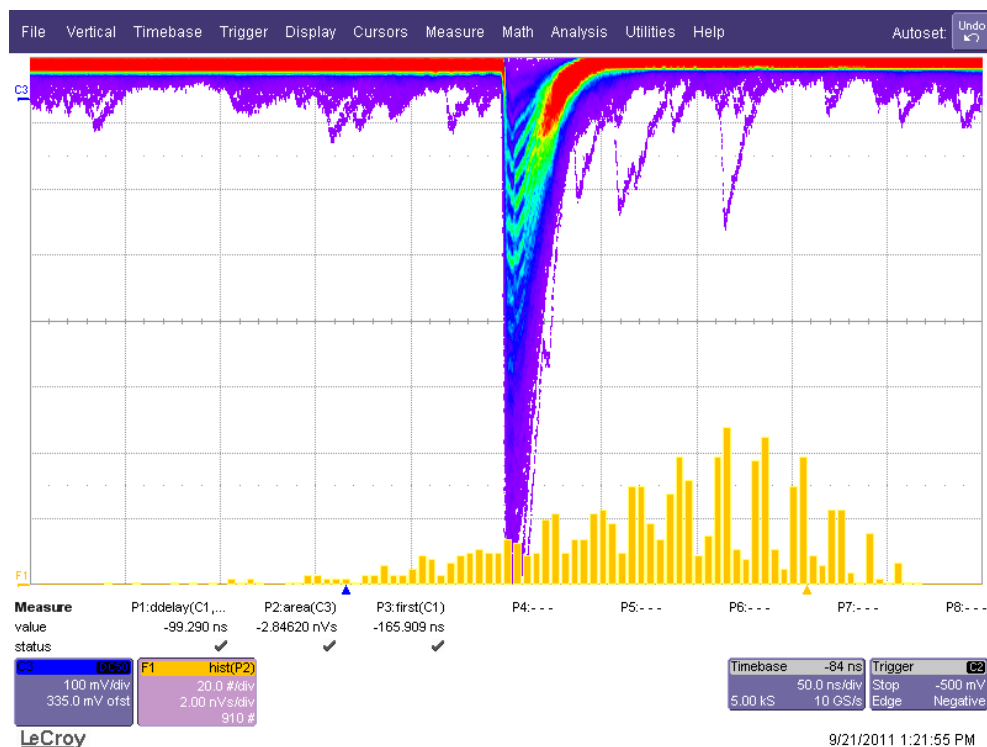
Tabela 6: Tabela efektivnih napak zakasnitvenih linij

številu zaznanih fotonov (slika 17). Ta se dobro loči od šumnega signala, kjer večinoma zaznamo le napetost, ki ustreza enemu fotonu, in le redko dvema ali večim.

V detektorskih sistemih pa smo omejeni z velikostjo in posledično številom fotonov sevanja Čerenkova po enačbi (2.2). Dolžina poti skozi sevalec je tipično izbrana tako, da delec izseva le nekaj deset fotonov [18]. Ti so porazdeljeni po obroču in na fotopomnoževalki večinoma dobimo enofotonske signale, kvalitativno enake temnim sunkom. Pomembna razlika med obema tipoma pa je dobra korelacija pravih dogodkov s sprožilnim signalom, ki pri šumu ni prisotna.

Za test sevanje Čerenkova kot svetlobni izvor zamenjamo z laserjem. Intenziteto pulzov laserske svetlobe, s katerimi svetimo na silicijevo fotopomnoževalko, prilagodimo tako, da iz fotopomnoževalke le redko dobimo večfotonski signal. Ker imamo nizko povprečje in je število absorbiranih fotonov naključno, po Poissonovi porazdelitvi mnogokrat sploh ne bomo zaznali laserskega pulza. S tem simuliramo enako obnašanje kot pri detekciji sevanja Čerenkova. Ojačan signal iz fotopomnoževalke na sliki 18 preko diskriminatorja pripeljemo na časovni merilnik. Enako storimo s sprožilnim signalom iz laserja, ki ga vodimo na drug kanal merilnika, da z njim definiramo časovno izhodišče in gledamo, kako so dogodki iz fotopomnoževalke odvisni od njega.

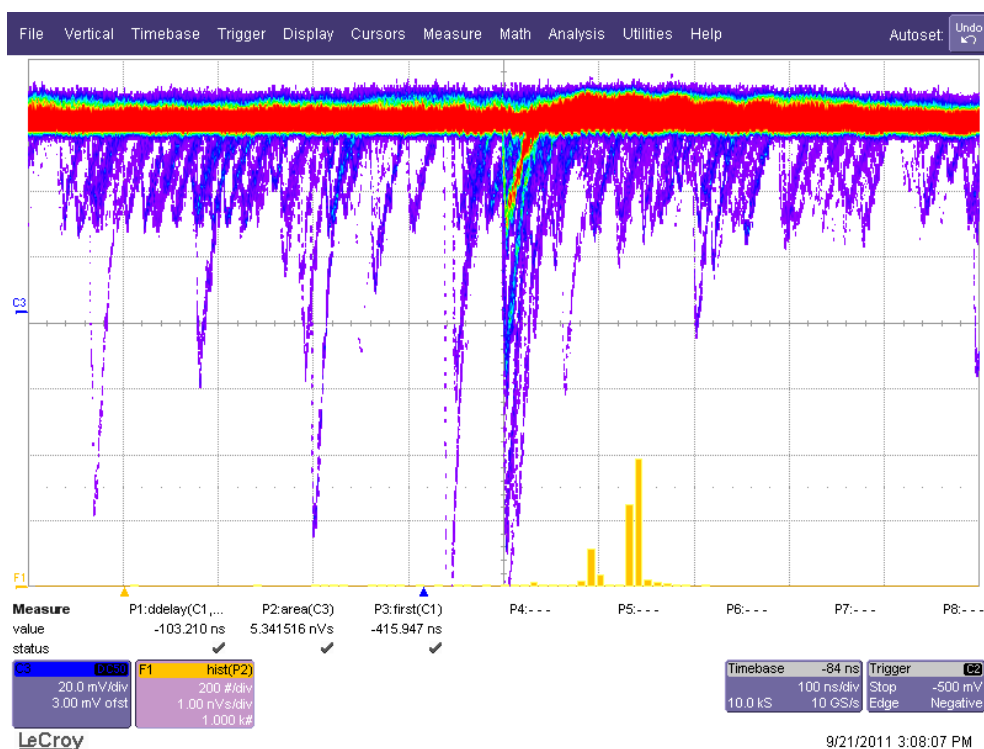
Iz slike 19 je lepo razvidna korelacija pravih dogodkov v obliki dobro definiranih pokončnih črt. Te so razmaknjene za $10 \mu s$, saj smo frekvenco laserskih pulzov nastavili na 100 kHz. Temni dogodki so po drugi strani raztreseni po celotnem intervalu in jih predstavlja signal med črtami, ki ga imenujemo kar *ozadje*, po analogiji s sevanjem. Ker takšni naključni dogodki niso korelirani, jih je ob poljubnem času precej manj kot je dogodkov ob vrhovih – na levi sliki je logaritemska skala. Kljub temu, da je število vseh temnih dogodkov večje od pravih.



Slika 17: Signal iz silicijeve fotopomnoževalke za laserski pulz povprečno ~ 5 zaznanih fotonov (zgoraj) in histogram naboja sunka, kjer so vidni vrhovi, ki ustrezajo številu sproženih celic (spodaj). Drugi vrh z desne ustreza enemu zaznanemu fotonu, njihovo število pa raste v levo – razlog je negativen predznak vhodnega signala.

Desno je povečava prvega vrha, kjer lahko vidimo, da le-ta ni oster, ampak je malo razmazan. To je posledica treh dejavnikov; najprej je zaradi ločljivosti merilnika pričakovati nihanje izmerjene vrednosti v intervalu slabih treh nanosekund. Poleg tega se čas z diskriminatorjem zaznanega pulza glede na prožilni signal lahko razlikuje tudi za več kot nanosekundo. Končno prispeva nek del še postopek razvrščanja vrednosti po predalčkih, ki v tem primeru predstavljajo 1 ns intervale, saj manjšanje velikosti predalčkov daleč pod ločljivost pač ni smiselno. Prvi vrh tako določimo pri 101,8 ns, z efektivnim odmikom 1,3 ns.

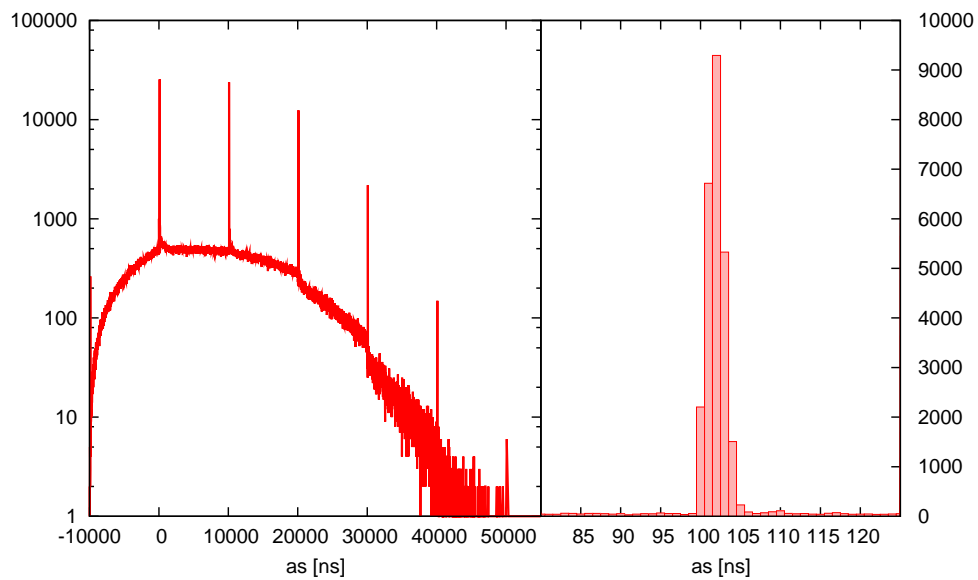
Vrhov je več, ker smo vsem zaznanim dogodkom iz ene meritve odšteli čas prvega prožilnega signala. Merilna naprava je zmožna hraniti le do 16 dogodkov naenkrat, kar v tem primeru predstavlja eno meritev. Pogostost temnih dogodkov je reda 100 kHz, toliko kot laserskih pulzov. Na sliki bi zato v povprečju pričakovali nekako 8 vrhov. Ker jih je manj, lahko sklepamo, da



Slika 18: Signal iz silicijeve fotopomnoževalke za laserski pulz povprečno ~ 1 zaznanega fotona (zgoraj) in histogram naboja sunka (spodaj); vrh na desni ustreza nič fotonom, torej brez zaznanega signala, prvi vrh levo pa enofotonskemu.

je temnih dogodkov več. Meritev pokaže dejansko vrednost $491,8 \pm 0,7$ kHz.

Velikost vrhov s časom upada zaradi omejenosti s pomnilnikom na merilniku. Vrednosti ob negativnem času so posledica dogodkov, ki so se zgodili, preden smo izmerili prvi prožilni signal. Število teh z negativnim časom tudi pada, saj jih z vedno večjo verjetnostjo izmerimo po prožilnem signalu (verjetnost je konvolucija enakomerne porazdelitve šuma in porazdelitve za naključni prihod prvega prožilnega signala).



Slika 19: Število zaznanih dogodkov glede na prožilni signal; desno je povečava prvega vrha signala na levi.

6 Zaključek



Možnosti za izboljšave izdelanega časovnega merilnika je več. Poleg že omejenih razširitev merilnega območja in števila kanalov ter morda večjega pomnilnika, je najpomembnejša večja ločljivost. To je najenostavneje doseči z uporabo boljšega in hitrejšega čipa FPGA, a tudi pri najboljših čipih samo s tem ne pridobimo več kot velikostnega reda. Alternativa je uporaba Vernierjeve metode, npr. z dvema zakasnitvenima linijama, vendar na račun daljšega časa trajanja meritve. Še dodatna izboljšava, ki bi jo lahko uporabili tako na naši izvedbi kot v kombinaciji z Vernierjevo metodo, so ti. prožilniki valovnih spojev, ki ob prihodu signala skozi zakasnitveno linijo pošljejo prilagojen signal z dodatnim prehodom [17]. To omogoči natančnejšo meritev na celicah z največjo zakasnitvijo, saj lahko prehod izmerimo še na drugi celici, kjer je zakasnitev krajša. Takšni merilniki dosežejo ločljivost vse do 10 ps.

Ob tem se je potrebno zavedati, da takšne izboljšave povečajo kompleksnost merilnika in potrebo po številu vrat v čipu FPGA. Pri detektorskih sistemih z majhnimi fotopomnoževalkami pokrivamo sorazmerno velike površine in je zato potrebno veliko število merilnih kanalov. V takih primerih je verjetno bolj racionalna uporaba enostavnejših merilnikov, kakršen je bil predstavljen v tem delu.

~~Zanj smo izmerili~~ ločljivost boljšo od 0,5 ns na kanal ter s tem izpolnili cilj za ustreznost časovno digitalnega pretvornika za uporabo v spektrometrih, kjer se določa hitrost delcev v detektorjih Čerenkovih obročev s pomočjo silicijevih fotopomnoževalk. V tem primeru je pogoj določitev časa dogodka na vsaj 10 ns natančno [19], kar smo eksperimentalno preverili s testom na silicijevi fotopomnoževalki. Čeprav je bil razvoj podrejen zadostitvi zahtev za takšno specifično uporabo, pa ni nobene ovire, da naprave ne bi mogli uporabiti tudi za druge aplikacije, ki potrebujejo merjenje časa z nanosekundno ločljivostjo.

A Program za zajemanje podatkov

Sledi izvorna koda programa, s katerim so bili zajeti podatki iz izdelanega časovno-digitalnega pretvornika. Napisana je v programskem jeziku C ter uporablja nekatere funkcije razvojnega okolja LabWindows/CVI za komunikacijo z operacijskim sistemom. Funkcije so bile pripravne za uporabo, vendar je program moč uporabljati brez njih, če se popravi glavna rutina `WinMain` in definira funkciji `inp` ter `outp`, ki komunicirata s paralelnim vhom.

```
#include <ansi_c.h>
#include <utility.h>
#include <cvirte.h>
#include <stdio.h>

/* Control:
    0x08 start
    0x04 reset (non-inverted)
    0x02 strobe enable (read clock)
    0x01 strobe (read clock)
    Status:
    0x80 data ready (inverted)
*/

#define WRITE 1
#define BASEADDR 0x378

unsigned char ctrlReg = 0x29; // b10 1001

int event_wait ()
{
    unsigned char istat;
    int i;

    //printf("Wait for data...\n");
    for (i = 0; i < 1000; i++)
    {
        istat = inp (BASEADDR + 1);
        if (!(istat & 0x80)) return 1;
        Delay(0.01);
    }
}
```

```

    if (WRITE) {
        printf("x");
    } else {
        printf("Error: Data_NOT_ready!\n");
        getchar();
    }

    return 0;
}

int event_print (FILE *datafile)
{
    unsigned char idata;
    unsigned long result;
    int i;

    // read data (4 bytes)
    result = 0;
    for (i = 0; i < 4; i++) {
        idata = inp (BASEADDR);

        // strobe for next byte
        outp(BASEADDR + 2, ctrlReg | 0x02); // b11
        outp(BASEADDR + 2, ctrlReg & !0x01); // b10
        outp(BASEADDR + 2, ctrlReg); // b01

        result |= idata << (8*i);
    }

    if (result & 0x40) {
        if (WRITE) {
            printf("!");
        } else {
            printf("Channel_%u:_%lu_%u_%u_",
                ((result & 0x20) >> 5) + 1,
                (result >> 8) & 0xffff,
                result & 0x80,
                result & 0x1f);
            printf("Error: Corrupt_data!\n");
        }
    }
}

```

```

        getchar();
    }

    return 0;
}

if (WRITE)
    fprintf(datafile, "%u:_%lu_%u_%u\n",
            ((result & 0x20) >> 5) + 1,
            (result >> 8) & 0xffffffff,
            result & 0x80,
            result & 0x1f);
else
    printf("Channel_%u:_%lu_%u_%u\n",
           ((result & 0x20) >> 5) + 1,
           (result >> 8) & 0xffffffff,
           result & 0x80,
           result & 0x1f);

return 1;
}

int __stdcall WinMain (HINSTANCE hInstance, HINSTANCE hPrevInstance,
                      LPSTR lpszCmdLine, int nCmdShow)
{
    FILE *datafile = NULL;
    int i, j, k;

    if (InitCVIRTE (hInstance, 0, 0) == 0)
        return -1;    /* out of memory */

    if (WRITE)
        datafile = fopen("data.txt", "w");

    //outp(0x37a, 0x0c); return 0; // reset

    for (i=1; i<=100000; )
    {
        outp(BASEADDR + 2, ctrlReg | 0x04); // init
        outp(BASEADDR + 2, ctrlReg);
    }
}

```

```

//outp(BASEADDR + 2, ctrlReg & !0x08); // go
//outp(BASEADDR + 2, ctrlReg);

for (j=1; j<=16; j++)
{
    event_wait();
    event_print(datafile);
}

if (WRITE)
    fprintf(datafile, "\n");
else
    printf("\n");

if (WRITE) {
    if (i % 5000 == 1)
        printf("%2u:~", i / 5000 + 1);
    if (i % 100 == 0)
        printf(".");
    if (i % 5000 == 0)
        printf("\n");
};
i++;
}

if (WRITE)
    fclose(datafile);

printf("Done!\n");
getchar();

return 0;
}

```

Literatura

- [1] F. Halzen and A.D. Martin. *Quarks and leptons: an introductory course in modern particle physics*. Wiley, 1984.
- [2] D. Denisov. Detection of muons. *Academic Lecture*, 2000. http://www-ppd.fnal.gov/EPPOffice-w/Academic_Lectures/Denisov_%20Lecture.pdf.
- [3] T. Ypsilantis and J. Seguinot. Theory of ring imaging cherenkov counters. *Nucl. Instr. and Meth. A*, 343(1):30 – 51, 1994.
- [4] J.V. Jelley. *Cerenkov Radiation And Its Applications*. Pergamon Press, 1958.
- [5] Arpad Horvath. Licensed under: <http://creativecommons.org/licenses/by/3.0/>.
- [6] J. Seguinot and T. Ypsilantis. Photo-ionization and cherenkov ring imaging. *Nucl. Instr. and Meth.*, 142(3):377 – 391, 1977.
- [7] S. Korpar et al. Rich with multiple aerogel layers of different refractive index. *Nucl. Instr. and Meth. A*, 553(1-2):64 – 69, 2005.
- [8] T. Abe et al. Belle II Technical Design Report. *ArXiv e-prints*, November 2010.
- [9] B. Dolgoshein et al. Status report on silicon photomultiplier development and its applications. *Nucl. Instr. and Meth. A*, 563(2):368 – 376, 2006. TRDs for the Third Millenium - Proceedings of the 3rd Workshop on Advanced Transition Radiation Detectors for Accelerators and Space Applications.
- [10] S. Korpar et al. Silicon photomultiplier as a detector of cherenkov photons. *Nucl. Instr. and Meth. A*, 595(1):161 – 164, 2008.
- [11] Józef Kalisz. Review of methods for time interval measurements with picosecond resolution. *Metrologia*, 41(1):17, 2004.
- [12] Ian Kuon and Jonathan Rose. Measuring the gap between fpgas and asics. In *Proceedings of the 2006 ACM/SIGDA 14th international symposium on Field programmable gate arrays*, FPGA '06, pages 21–30, New York, NY, USA, 2006. ACM.

- [13] Xilinx, Inc. Spartan-ii fpga family data sheet. June 2008. http://www.xilinx.com/support/documentation/data_sheets/ds077.pdf.
- [14] Thomas J. Chaney and Charles E. Molnar. Anomalous behavior of synchronizer and arbiter circuits. *IEEE Transactions on Computers*, C-22(4):421–422, April 1973.
- [15] Ieee standard signaling method for a bidirectional parallel peripheral interface for personal computers. *IEEE Std 1284-2000*, pages i–100, 2000.
- [16] Jan Louise Axelson. *Parallel Port Complete: Programming, Interfacing and Using the PC's Parallel Printer Port*. Lakeview Research, 1997.
- [17] Jinyuan Wu and Zonghan Shi. The 10-ps wave union tdc: Improving fpga tdc resolution beyond its cell delay. *2008 IEEE Nuclear Science Symposium Conference Record*, 1(630):3440–3446, 2008.
- [18] S Amato et al. *LHCb RICH: Technical Design Report*. Technical Design Report LHCb. CERN, Geneva, 2000.
- [19] R. Pestotnik et al. Silicon photomultiplier as a position sensitive detector of cherenkov photons. *Nucl. Instr. and Meth. A*, 581(1–2):457 – 460, 2007.

Izjava o avtorstvu

Izjavljam, da je to diplomsko delo moje avtorsko delo.

Ljubljana, 25. november 2012

Blaž Bačnik