

UNIVERZA V LJUBLJANI
FAKULTETA ZA MATEMATIKO IN FIZIKO
ODDELEK ZA FIZIKO
FIZIKA KONDENZIRANE SNOVI

Michel Adamič

Časovno digitalni pretvornik visoke ločljivosti na čipu Xilinx Zynq-7010

Magistrsko delo

MENTOR: izr. prof. dr. Rok Pestotnik
SOMENTOR: izr. prof. dr. Andrej Trost

Ljubljana, 2020

Zahvala

Delo, vsebovano v tem magisteriju, je močno interdisciplinarno, hkrati pa je zahtevalo nekaj izredno specifičnih znanj načrtovanja digitalnih sistemov. Sam temu nikoli ne bi bil niti približno kos, zato sem se moral pri nalogi za pomoč obrniti h kar nekaj ljudem. Najprej gre zahvala mentorju izr. prof. dr. Roku Pestotniku za predlog teme, usmerjanje pri delu in entuziastično pomoč pri eksperimentih. Najgloblje se zahvaljujem somentorju izr. prof. dr. Andreju Trostu s Fakultete za elektrotehniko, ki me je skozi vsa ta leta naučil osnov načrtovanja vezij na čipih FPGA in mi neomajno pomagal pri vseh izzivih, ki sem jih srečal na poti. Iskrena hvala Klemnu Erjavcu iz podjetja Cosylab za mentorstvo pri FPGA Akademiji in vse koristne nasvete o učinkoviti uporabi čipa Zynq. Hvala tudi doc. dr. Dušanu Poničkarju za zanimive pogovore in predloge ter prijazno možnost izposoje laboratorijske opreme. Na koncu ne gre pozabiti na Antona Potočnika, ki je z veseljem prispeval kodo za IP strežnik na Red Pitayi, Domna Cvenkla za pomoč pri izdelavi grafičnega vmesnika, ter Žiga Krajnika za matematični vložek v poglavju o naključnem šumu. Vsem najlepša hvala!

Nazadnje bi se rad zahvalil družini, študijskim kolegom in prijateljem za vso podporo tekom študija.

Izvleček

V sklopu tega magistrskega dela smo se lotili izdelave hitrega časovno-digitalnega pretvornika visoke ločljivosti na cenovno dostopni plošči Red Pitaya, ki temelji na zmogljivem polno programirljivem čipu Xilinx Zynq 7010. Časovno-digitalni pretvornik (TDC) je popolnoma digitalen in v celoti implementiran na programirljivem polju vrat (FPGA) v ravnokar omenjenem čipu, tako da za opravljanje meritev ne potrebujemo nobenih dodatnih zunanjih komponent. TDC je bil zasnovan po principu časovne interpolacije, kjer čas štejemo z uporabo sinhronega binarnega števca, za zelo natančno določitev časa prihoda med urinimi cikli pa skrbi integrirana zakasnilna linija. Posamezen kanal TDC je na voljo v obliki strojne komponente intelektualne lastnine (jedra IP) z vmesnikom AXI, kar je zelo priročno za gradnjo večkanalnih sistemov. Procesni sistem Zynq bere časovne značke aktivnih časovno-digitalnih pretvornikov in jih preko omrežja Ethernet pošilja odjemalcu, recimo osebnemu računalniku, kjer jih lahko ustrezno obdelamo in prikažemo v grafičnem uporabniškem vmesniku. V sklopu tega magisterija smo implementirali dva kanala TDC in ju podrobno okarakterizirali. Posamezen časovno-digitalni pretvornik teče na frekvenci 350 MHz in je ob mrtvem času ~ 14 ns sposoben vzorčiti do 70 milijonov značk na sekundo. Časovna ločljivost posameznega kanala dosega odličnih 11 ps in ostane zelo visoka tudi pri večjih časovnih intervalih, vsaj nekje do 100 ns. Izkazalo se je, da lahko tudi s povsem običajnimi signali s hitrostjo naraščanja 10 ns opravljamo meritve visoke ločljivosti, prav tako pa je meritev robustna na spremembe temperature. Izdelani TDC je zelo natančen in periodo zunanjega signala izmeri z relativno natančnostjo 10^{-5} v 60 stopinj širokem temperaturnem intervalu. Nazadnje smo inštrument preizkusili še v realistični postavitvi s pulznim laserjem in hitro silicijevo fotopomnoževalko (SiPM), kjer se je novi TDC odlično obnesel. Vse to dokazuje, da se da z izbiro primerne arhitekture in modernega čipa FPGA tudi na cenovno zelo dostopnih platformah, kot je Red Pitaya, implementirati izredno zmogljiv časovni merilni inštrument, ki je uporaben tako za študente kot profesionalne raziskovalce.

Ključne besede: časovno-digitalni pretvornik, TDC, hitra digitalna vezja, FPGA, Zynq, zakasnilna linija, prenosna logika, pikosekundna ločljivost

PACS: 06.30.Ft, 07.05.Hd, 07.50.-e, 84.30.Sk

Abstract

This Master's Thesis presents an implementation of a fast high-resolution time-to-digital converter (TDC) on the affordable Red Pitaya board, featuring a powerful all programmable Xilinx Zynq 7010 SoC. The design is fully digital, enabling the TDC to be implemented entirely within the Zynq FPGA, thus requiring no additional external components. The architecture of the TDC is based on the time interpolation technique which employs a coarse binary counter and a tapped delay line for fine time measurements between adjacent clock cycles. A TDC channel is packaged into an AXI-interfaced IP core, making it easy to build multichannel systems. Produced timestamps from active TDC channels are read by the Zynq processing system and sent via Ethernet to a client, for example a PC which can process and display them with a graphical user interface. A two-channel TDC system has been implemented and thoroughly characterized during this Thesis work. An individual channel runs at 350 MHz and has a dead time of ~ 14 ns, sampling up to 70 million timestamps per second. Its time resolution can reach an astounding 11 ps and remains very high even at larger time intervals of up to 100 ns. It turned out that high-resolution measurements can also be done with standard 10 ns rise time signals and with widely varying temperatures. The implemented TDC is very accurate and can measure the period of an externally applied signal with relative accuracy of 10^{-5} over a 60 degree temperature swing. Finally, the instrument was put to the test in a realistic setup with a pulsed laser and a fast silicon photomultiplier (SiPM), where the new TDC performed exceedingly well. This concludes that by choosing the right architecture and a modern FPGA, a very powerful time measurement instrument, useful both for students and professionals, can be implemented on perfectly affordable hardware platforms such as the Red Pitaya.

Keywords: time-to-digital converter, TDC, high-speed digital circuit design, FPGA, Zynq, tapped delay line, carry chain, picosecond resolution

PACS: 06.30.Ft, 07.05.Hd, 07.50.-e, 84.30.Sk

Kazalo

Seznam kratic	11
1 Uvod	13
2 Osnova izdelanega TDC	15
2.1 Splošna arhitektura TDC	15
2.2 Razvojna platforma	16
3 Načrtovanje in implementacija časovno-digitalnega pretvornika . .	19
3.1 Kanal TDC	19
3.1.1 Jedro	19
3.1.2 AXI TDC	29
3.2 Sistem TDC	31
4 Karakterizacija inštrumenta	37
4.1 Kalibracija zakasnilne linije	38
4.2 Enokanalna TDC meritev in mrtvi čas	43
4.3 Hitrost zajemanja časovnih značk	46
4.4 Medkanalna TDC meritev & časovna ločljivost	48
4.4.1 Resolucija v odvisnosti od zakasnitve	51
4.4.2 Resolucija v odvisnosti od strmine pulza	54
4.4.3 Resolucija v odvisnosti od temperature	55
4.5 Absolutna natančnost	59
4.6 Stabilnost	61
4.7 Dodatek: Generator šuma	63
4.8 Povzetek lastnosti TDCja	65
5 Fizikalni eksperiment s SiPM	67
5.1 Režim posameznih fotonov	73
5.2 Režim visoke intenzitete	76
5.2.1 Nizek prag diskriminatorja	76
5.2.2 Visok prag diskriminatorja	80
5.3 Zasičenje	83
6 Zaključek	87
Literatura	91

Seznam kratic

- ASIC** Namensko integrirano vezje (angl. *Application Specific Integrated Circuit*)
- AXI** Napredni razširljivi vmesnik (angl. *Advanced eXtensible Interface*)
- BRAM** Blokovni pomnilnik z naključnim dostopom (angl. *Block Random Access Memory*)
- CMOS** Komplementarni MOS - tehnologija izdelave integriranih vezij (angl. *Complementary Metal-Oxide-Semiconductor*)
- DMA** Direktni dostop do pomnilnika (angl. *Direct Memory Access*)
- DNL** Diferencialna nelinearnost (angl. *Differential Non-Linearity*)
- FF** Spominska celica z dinamičnim proženjem (krajše *Flip-Flop*)
- FPGA** Programirljivo polje vrat (angl. *Field Programmable Gate Array*)
- GPIO** Splošnonamenski vhod/izhod (angl. *General Purpose Input/Output*)
- HV** Visoka napetost (angl. *High Voltage*)
- IEEE** Inštitut inženirjev elektrotehnike in elektronike (angl. *Institute of Electrical and Electronics Engineers*)
- ILA** Integrirani logični analizator (angl. *Integrated Logic Analyzer*)
- INL** Integralna nelinearnost (angl. *Integral Non-Linearity*)
- IP** Intelektualna lastnina (angl. *Intellectual Property*)
- JTAG** Protokol za programiranje (angl. *Joint Test Action Group*)
- LIDAR** Svetlobno zaznavanje in merjenje (angl. *Light Detection and Ranging*)
- LSB** Najmanj pomemben bit (angl. *Least Significant Bit*)
- LUT** Vpogledna tabela (angl. *Look-Up Table*)
- LV** Nizka napetost (angl. *Low Voltage*)
- LVDS** Nizkonapetostni diferencialni prenos signalov (angl. *Low-Voltage Differential Signaling*)
- MMCM** Mešani upravljalnik ure (angl. *Mixed-Mode Clock Manager*)

NIM Modul jedrske merilne instrumentacije (angl. *Nuclear Instrumentation Module*)

PET Pozitronska emisijska tomografija (angl. *Positron Emission Tomography*)

PL Programirljiva logika (angl. *Programmable Logic*)

PLL Fazno sklenjena zanka (angl. *Phase-Locked Loop*)

PS Procesni sistem (angl. *Processing System*)

RMS Koren povprečja kvadratov (angl. *Root Mean Square*)

RTL Načrtovanje na nivoju registrov (angl. *Register Transfer Level*)

SiPM Silicijeva fotopomnoževalka (angl. *Silicon Photo-Multiplier*)

SMD Element za površinsko montažo (angl. *Surface-Mount Device*)

SoC Sistem na čipu (angl. *System on Chip*)

TCP/IP Protokol za nadzor prenosa / Internetni protokol (angl. *Transmission Control Protocol / Internet Protocol*)

TDC Časovno digitalni pretvornik (angl. *Time to Digital Converter*)

TDL Zakasnilna linija (angl. *Tapped Delay Line*)

ToF Čas preleta (angl. *Time of Flight*)

TTL Tranzistorsko-tranzistorska logika (angl. *Transistor-Transistor Logic*)

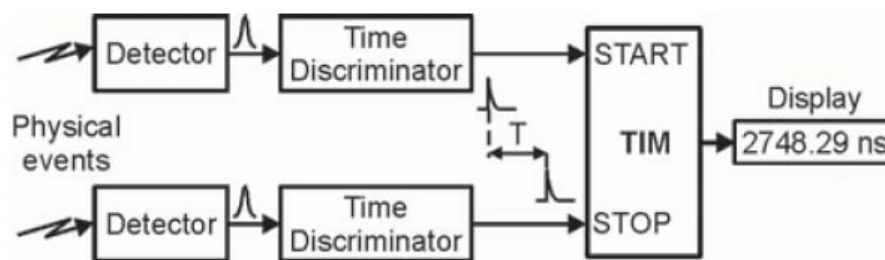
USB Univerzalno serijsko vodilo (angl. *Universal Serial Bus*)

VHDL Jezik za opis hitrih integriranih vezij (angl. *Very high speed integrated circuit Hardware Description Language*)

Poglavje 1

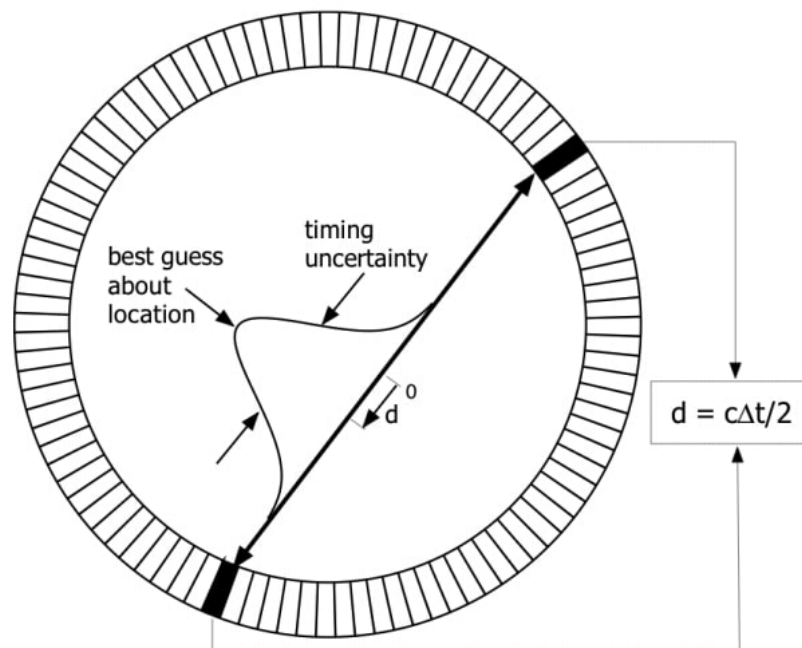
Uvod

Časovno-digitalni pretvornik ali po angleško TDC (*Time-to-Digital Converter*) je naprava za natančno merjenje časovnih intervalov med dogodki. Primer tipične izvedbe prikazuje slika 1.1, kjer z dvokanalnim TDCjem merimo časovno razliko med prihodom začetnega (START) in končnega (STOP) signala.



Slika 1.1: Meritev časovnega intervala. Slika povzeta iz [1].

Tipična aplikacija je merjenje časa preleta (ToF), kar recimo uporabljamo v visokoenergijski fiziki za identifikacijo delcev [2]. Na enak princip deluje tudi merjenje razdalje s pomočjo radarskih sunkov, laserske svetlobe (LIDAR) ali ultrazvoka [3]. Zadnje čase natančno merjenje časa preleta postaja zanimivo tudi za medicinsko fiziko, na primer v pozitronski emisijski tomografiji (PET). S pojavitvijo hitrejših fotonских detektorjev bi bilo možno poleg dozdajšnje prostorske zaznave “koincidenčnih” anihilacijskih gama žarkov natančno izmeriti tudi njihov čas prihoda, kar bi znatno izboljšalo resolucijo naprav PET [4]. Uporaba tako imenovane tehnike ToF-PET pomeni, da bi za isto ločljivost kot pri konvencionalnem PET slikanju potrebovali manj gama fotonov, kar pomeni, da lahko v pacienta vbrizgamo manjšo dozo radioaktivnega sevalca. Slika 1.2 prikazuje koncept delovanja naprave ToF-PET, kjer lahko že iz meritve zakasnitve med prihodom kolinearnih fotonov izluščimo položaj anihilacije. S pomočjo enačbe $d = ct$, kjer je c hitrost svetlobe, hitro ugotovimo, da svetloba za pot enega centimetra potrebuje približno 33 pikosekund. Z drugimi besedami, če bi radi, da je tehnika uporabna in bi merili položaje na centimeter natančno, potrebujemo časovno ločljivost reda velikosti nekaj 10 ps.



Slika 1.2: Princip delovanja naprave PET z dodano časovno informacijo. Povzeto po [4].

Ob iskanju literature hitro ugotovimo, da je bilo izdelanih že veliko TDCjev takšne ločljivosti, tako za znanstvene kot industrijske aplikacije. Za najvišje zmogljivosti se tipično uporabljajo namenska vezja (ASIC), za raziskovalne namene pa visokozmogljivi čipi FPGA (*Field Programmable Gate Arrays*), torej po slovensko električno programirljiva polja vrat. Problem je, da je večina izvedb ali zelo dragih (v primeru namenskih vezij), ali pa od načrtovalca zahteva visoko obvladovanje tehnologije FPGA in ogromno vložene truda v optimizacijo vezja za doseg visokih časovnih resolucij. Prav tako je veliko člankov že starejših, kot vemo, pa se polprevodniška tehnologija s pomočjo Moorovega zakona vsako leto radikalno izboljšuje.

V sklopu tega magistrskega dela sem želel preveriti, če lahko s svojim študentskim znanjem načrtovanja digitalnih vezij v tehnologiji FPGA in z uporabo modernega cenovno dostopnega čipa izdelam TDC visoke ločljivosti. Z mentorjema smo si zadali cilj pod 50 pikosekund, kar bi izdelan inštrument naredilo zelo uporabnega za raziskovalne namene na inštitutu in za študentske vaje na fakulteti.

V naslednjem poglavju bo predstavljena izbrana platforma za implementacijo inštrumenta in arhitektura časovno digitalnega pretvornika. Sledila bo podrobna predstavitev delovanja vsakega sklopa izdelanega vezja, v kar je bilo vložene največ truda. Nato pride obširna karakterizacija TDCja, se pravi množica fizikalnih meritev, s katerimi smo izmerili lastnosti inštrumenta. Za konec pa bom predstavil še fizikalno postavitev, s katero smo izdelani TDC preizkusili v praksi.

V želji, da bi ohranil rdečo nit besedila, osnovnih gradnikov digitalnih vezij, tehnologije čipov FPGA in načrtovanja z njimi tu ne bomo posebej obravnavali. V kolikor bralec s temi pojmi ni domač, je vljudno vabljen k branju mojega seminarja [5], ki je prosto dostopen na spletu.

Poglavje 2

Osnova izdelanega TDC

2.1 Splošna arhitektura TDC

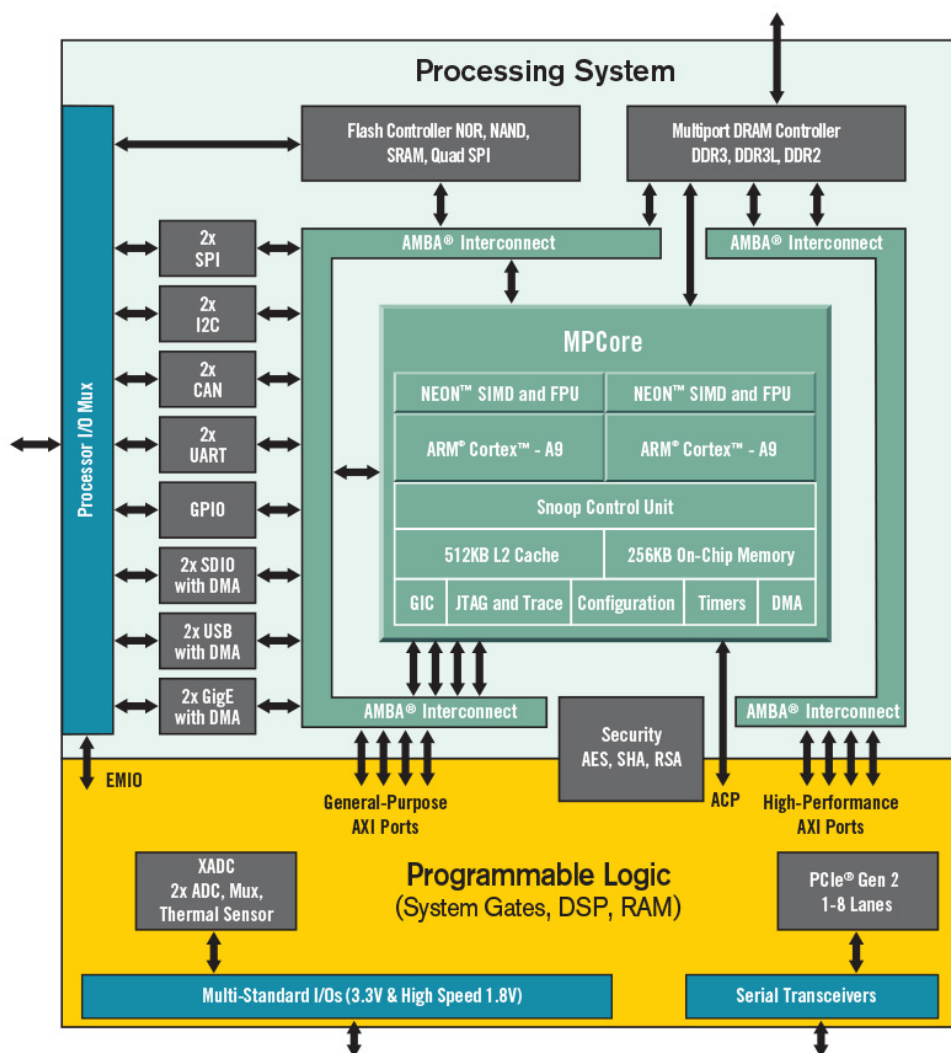
Obstaja ogromno izvedb časovno-digitalnih pretvornikov, tako analognih kot digitalnih [1]. Ker s Fakultete za elektrotehniko UL dobro poznam načrtovanje hitrih digitalnih vezij v tehnologiji FPGA, je bila izbira jasna. Vezje naj bo popolnoma digitalno in primerno za implementacijo v modernem čipu FPGA. Osnova te tehnologije je polje velikega števila programirljivih logičnih celic, ki jih lahko z uporabo ustreznega razvojnega orodja preoblikujemo in povežemo v poljubno digitalno vezje, ki je hkrati zelo zmogljivo in poceni.

Ker so programirljiva polja vrat primerna za implementacijo sinhronih digitalnih vezij, se osredotočimo na slednje. Enota za čas v sinhronih vezjih je ura, ki daje takt za delovanje vezja. Najenostavnejša metoda merjenja časa je torej štetje urinih ciklov, kar imenujemo ‘grob’ štetje [1]. Takšen pristop omogoča široko merilno območje, a bistveno premajhno časovno ločljivost. Le-ta je določena s periodo urinega signala, ki pa je v čipih FPGA tipično nekaj nanosekund. Za pikosekundne čase potrebujemo drugačno, ‘fino’ meritev, ki se ponavadi realizira z uporabo zakasnilne linije oziroma po angleško *Tapped Delay Line* - TDL [1]. Pri tem začetni pulz sproži signal, ki s končno hitrostjo potuje po verigi zakasnilnih elementov (po angl. *taps*), katerih stanje se vzorči ob prihodu končnega pulza. Ker je hitrost potovanja signala po liniji znana, lahko iz števila prepotovanih elementov razberemo časovno razliko med začetnim in končnim pulzom. Ta metoda omogoča zelo visoko ločljivost, je pa močno omejena z merilnim intervalom, ki ga lahko izmeri; daljši intervali zahtevajo daljšo linijo, kar pa hitro zasede preveč elementov na čipu in je zato neekonomično.

Pri naši izvedbi smo se odločili za tako imenovano tehniko ‘interpolacije’ [1], ki združuje obe omenjeni metodi v eno. Uporabljena je zakasnilna linija TDL visoke ločljivosti v kombinaciji s sinhronim števcem ure, ki zagotavlja široko merilno območje. Izmerjene časovne značke so torej preko fine meritve interpolirane med sosednjimi vrednostmi grobega števca, kar združuje prednosti obeh pristopov. Prav tako je metoda popolnoma digitalna in kot nalašč za uporabo v čipu FPGA, ker odlično izkorišča razpoložljive elemente znotraj čipa za izdelavo zakasnilne linije. To omogoča popolno integracijo časovno-digitalnega pretvornika na istem kosu silicija, brez potrebe po kakršnih koli zunanjih elementih. Arhitektura takšnega vezja je relativno enostavna in hkrati obljublja visoko zmogljivost, zato sem jo izbral tudi za svojo implementacijo.

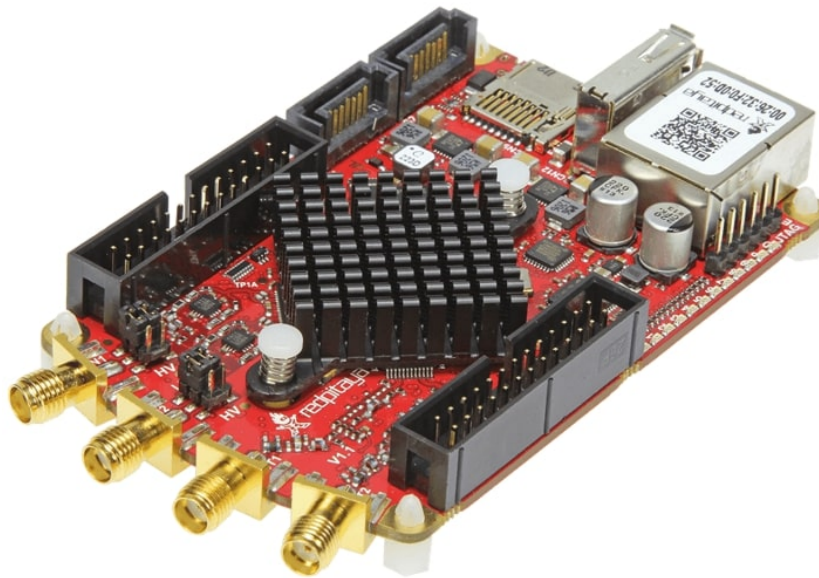
2.2 Razvojna platforma

Za izdelavo TDCja smo izbrali serijo sistemov na čipu (SoC) Zynq-7000 proizvajalca Xilinx. Gre za zelo zmogljive polno programirljive sisteme najnovejše 7. serije Xilinxove logike. Sistemi Zynq poleg polja vrat vsebujejo tudi procesni sistem z dvema jedroma ARM Cortex-A9 in vso pripadajočo periferijo (slika 2.1), kar načrtovalcu omogoča realizacijo kateregakoli digitalnega sistema. Edina omejitev je domišljija. Procesni in FPGA del sta povezana s hitrimi vodili AXI, ki omogočajo učinkovito izmenjavo podatkov. Pomemben podatek je, da je čip izdelan v sodobni 28-nm tehnologiji CMOS, kar je ključno za zmogljivost naše zakasnilne linije in cele enote TDC. Manjši tranzistorji so namreč hitrejši, in kot bomo videli, so za natančne časovne meritve potrebne čim krajše zakasnitve med posameznimi komponentami.



Slika 2.1: Splošna arhitektura Xilinxovih sistemov na čipu Zynq 7000. Čip vsebuje programirljivo logiko (PL) oziroma FPGA in procesni sistem (PS), ki sta med sabo povezana s hitrimi vodili AXI. Slika vzeta iz [6].

Obstaja veliko razvojnih plošč z omenjenim čipom, med najbolj znanimi je recimo ZedBoard [7], ali pa kar domača Red Pitaya [8]. Obojih imamo na Fakulteti za elektrotehniko na pretek, ker gre za relativno poceni vezja, ki so primerna za študijsko rabo. Naposled sem se odločil za Red Pitayo (slika 2.2), ker ima na procesnem sistemu že nameščen polno delujoč operacijski sistem Linux, kar mi je olajšalo delo pri kasnejšem programskem delu TDCja. Red Pitaya je slovenski izdelek iz Solkana in vsebuje čip Zynq-7010, ki je najmanjši in najcenejši dvojedrni predstavnik ravnokar omenjene serije 7000. Bazira na logiki Artix 7, ki več kot zadošča za to magistrsko nalogo. Sicer ni pri Red Pitayi nič posebnega in bi lahko uporabil katerokoli drugo platformo. Prav tako ni važna verzija tiskanine, ker ne uporabljam analognega dela, ki ga ponuja Red Pitaya, ampak izključno Xilinxov čip.



Slika 2.2: Slovenska Red Pitaya STEMLab 125-14. Srce, Xilinxov Zynq-7010, se skriva pod črnim hladilnikom. Slika od [8].

Poglavje 3

Načrtovanje in implementacija časovno-digitalnega pretvornika

V tem poglavju se bomo osredotočili na jedro tega magistrskega dela, torej na samo izdelavo oziroma strukturo časovno-digitalnega pretvornika. Vezje sem načrtoval v strojno opisnem jeziku VHDL v razvojnem okolju Xilinx *Vivado* 2018. Kompleksnejše digitalne sisteme, kamor sodi tudi ta, se gradi od spodaj navzgor po principu Lego kock. Predstavil ga bom po korakih, od osnovnih gradnikov posamezne enote TDC do celotnega sistema, enako kot je vezje tudi nastajalo.

3.1 Kanal TDC

Posamezno merilno enoto, ki vhodnim pulzom meri čas prihoda, definiramo kot kanal časovno-digitalnega pretvornika (TDC). Posamezen kanal predstavlja neodvisno enoto, ki se lahko skupaj z drugimi zлага v večje sisteme, kot je recimo dvokanalni na sliki 1.1. Od zdaj naprej bomo ob besedi TDC oziroma časovno-digitalni pretvornik namigovali na posamezen kanal.

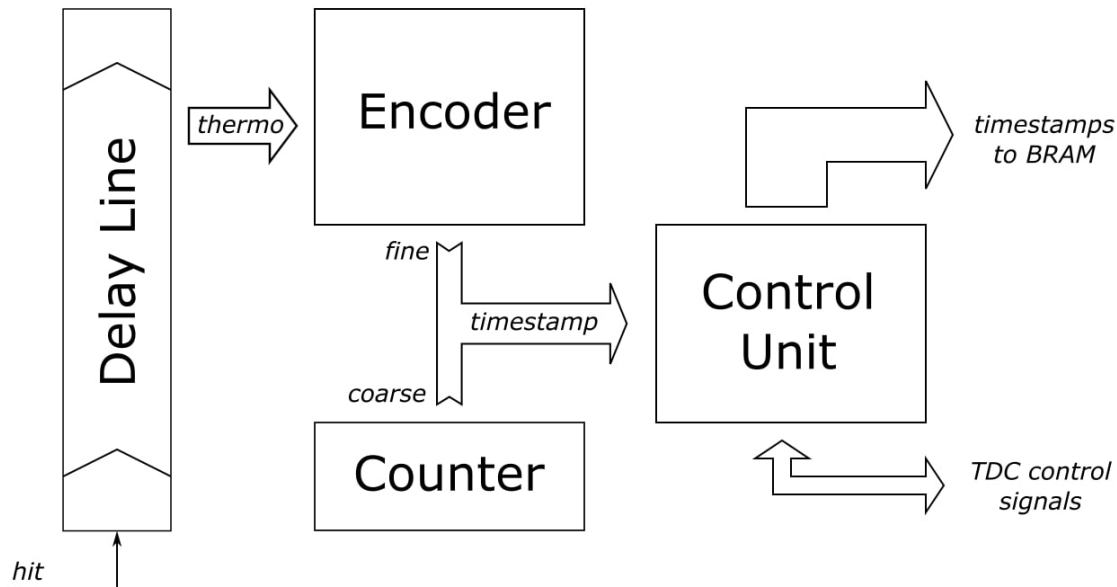
3.1.1 Jedro

Jedro posameznega časovno-digitalnega pretvornika tvori struktura iz štirih blokov, ki jo vidimo na sliki 3.1. Imamo zakasnilno linijo za fino meritev, enkoder, grobi števec in kontrolno enoto. Vezje ima en zunanji priključek, na sliki imenovan *hit*, kamor pripeljemo signal, ki ga želimo meriti.

Zakasnilna linija

Zakasnilna linija je srce TDCja, precizen merilni inštrument, zadolžen za časovne meritve najvišje ločljivosti. Struktura linije je klasičen TDL [1], kot lahko vidimo na sliki 3.2. V mirovanju je linija v stanju logične ničle, kar pomeni, da imajo vsi elementi nizek logični nivo. Ko na vhod pripeljemo pulz, torej visok nivo, se sproži potujoča pozitivna fronta (val prehajanja iz '0' v '1'), ki potuje s končno hitrostjo vzdolž zakasnilne linije in postavlja enice na *D* vhode sosednjih flip-flopov. Ob naslednji fronti ure (signal *clk*) ti flip-flopi zajamejo stanje linije in s tem dolžino poti, ki jo je fronta prepotovala. To pomeni, da s tem lahko natančno izmerimo, kdaj se je na vhodu pojavil pulz relativno na urin signal. Če je pulz prišel tik pred

Poglavje 3. Načrtovanje in implementacija časovno-digitalnega pretvornika



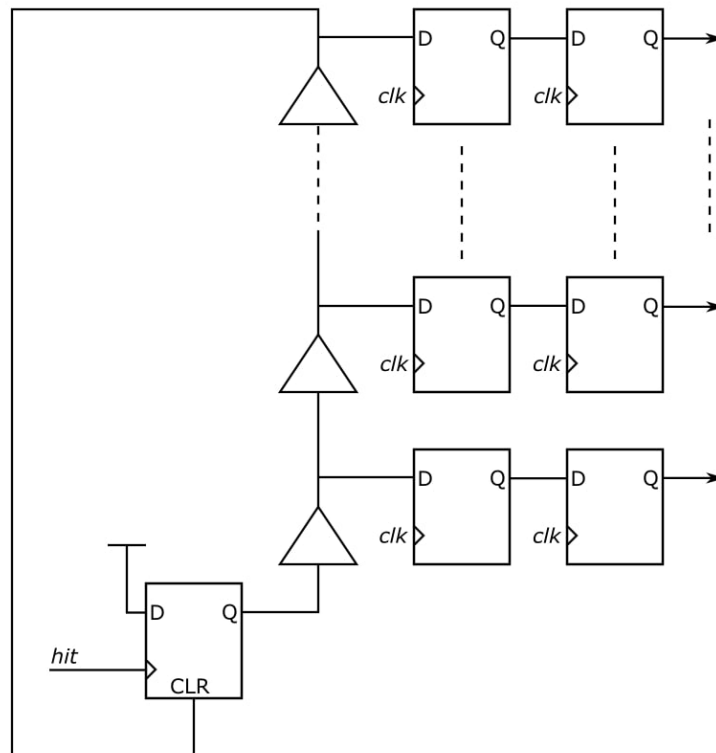
Slika 3.1: Jedro TDCja je sestavljeno iz štirih blokov. Vhodni signal *hit* pripeljemo na zakasnilno linijo (*Delay Line*), njen izhod pa na enkoder (*Encoder*), ki izračuna rezultat fine časovne meritve. Ta se združi z vrednostjo grobega števca (*Counter*) v časovno značko, ki jo kontrolna enota (*Control Unit*) vpiše v pomnilnik BRAM. Shema je namenoma v angleščini, da se imena komponent ujema z opisom v jeziku VHDL.

rastočo fronto ure, bo prepotoval le kratko pot, če pa mnogo prej, pa bo prišel skoraj do konca linije. Točno to je ta 'interpolacija' med sosednjimi urinimi cikli, ki nam da fino informacijo časa prihoda. Glede na to, da je vhodni signal asinhron, z drugimi besedami nekoreliran z uro, morajo biti flip-flopi podvojeni v izogib metastabilnemu stanju na izhodu [9].

Zakasnitev celotne linije mora biti daljša od periode ure, da lahko pokrijemo celo področje med zaporednima urinima cikloma [2, 10]. Trenutna implementacija TDCja uporablja 192 zakasnilnih elementov, s tem da so zaradi preglednosti na sliki 3.2 prikazani le trije. Linija je samoregulacijska in ne zahteva nobene zunanje ponastavitvene logike. Ko val enic doseže konec linije, se pojavi na vhodu *CLR* vhodnega flip-flopa (glej sliko 3.2), kar blokira signal *hit* in v linijo pošlje val ničel, torej negativno fronto. Ko le-ta ponastavi celo linijo nazaj na '0', se *CLR* sprosti in linija je pripravljena na zajem novega pulza. Čas, med katerim linija po proženju ignorira prihajajoče pulze, se imenuje mrtvi čas in je pomemben podatek TDCja. Izmerili ga bomo v naslednjem poglavju.

Vhodni flip-flop zagotavlja, da je linija občutljiva izključno na rastočo fronto signala *hit* in nič drugega. To tudi pomeni, da dolžina trajanja vhodnega pulza ni pomembna. 'Dogodek', ki ga TDC izmeri, je torej čas prihoda pozitivne fronte na njegov vhod. Naj še omenim, da vsak dogodek proizvede samo eno časovno značko. Linija namreč vsebuje še nekaj malega dodatne, na sliki izpuščene logike, ki poskrbi, da je zabeleženo le prvo vzorčenje linije. Ta se namreč vzorči tudi kasneje med ponastavljanjem, vendar so tiste značke neveljavne.

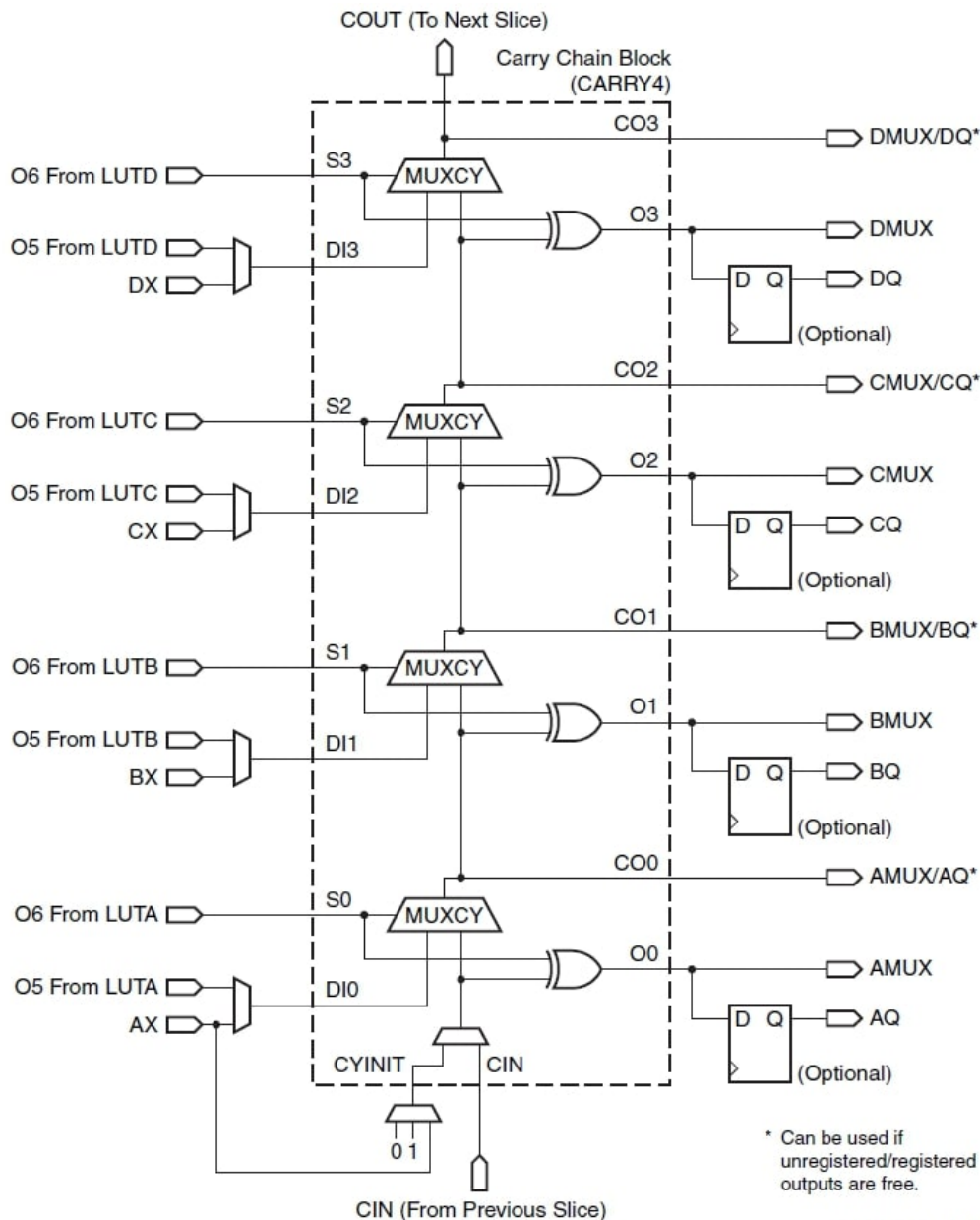
Še vedno nismo odgovorili na glavno vprašanje. Kaj so ti zakasnilni elementi,



Slika 3.2: Zakasnilna linija iz diskretnih zakasnitvenih elementov. Zaradi enostavnosti so prikazani le trije elementi.

na sliki 3.2 označeni s trikotniki? Pravzaprav so lahko karkoli. Vsaka logična vrata vedno povzročijo nek zamik; tranzistorji potrebujejo nek končen čas, da se odzovejo na spremembo vhodne vrednosti. Za namene našega merilnega inštrumenta je treba izbrati vrata s čim manjšim zamikom, ki jih lahko znotraj čipa FPGA povežemo v ravno linijo. Kandidate iščemo v logičnih celicah oziroma Xilinxovih 'rezinah' (angl. *Slice*), ki vsebujejo vpogledne tabele (LUT), prenosno logiko, multiplekserje in flip-flope [11]. Po brskanju po literaturi ugotovimo, da skoraj vse implementacije TDCjev v tehnologiji FPGA uporabljajo prenosno logiko, se pravi osnovne gradnike CARRY4. Zgradbo tega gradnika vidimo na sliki 3.3, vsebuje pa ga vsaka rezina [11]. Notri so štirje multiplekserji, na sliki označeni z *MUXCY*, ki jih lahko odlično izkoristimo kot zakasnilne elemente. Dodatna prednost uporabe prenosne logike je v geometriji; signal namreč potuje od spodaj (*CIN*) navzgor (*COUT*), kar pomeni, da lahko sosednje rezine v vertikalni smeri povežemo zaporedno v lepo, ravno linijo. To je izrednega pomena za čimbolj enakomerno razporeditev zakasnitev znotraj linije. Čipi FPGA imajo velike zakasnitve pri povezavah med rezinami, zato je kritično, da je linija ravna in neprekinjena. Za implementacijo vezja v posamezne rezine poskrbi Xilinxovo razvojno orodje, proces pa je popolnoma avtomatiziran, čeprav je v principu možno tudi naknadno ročno popravljanje. Na srečo je orodje Vivado 2018 zakasnilno linijo že avtomatsko sestavilo optimalno, tako da ročno popravljanje pri implementaciji ni bilo potrebno.

Kot zanimivost lahko omenim še, da bi lahko širjenje signala po zakasnilni liniji zakodirali tudi kot negativno fronto, torej prehod iz '1' v '0'. Vseeno sem se odločil za pozitivno fronto, ker naj bi slednja v Xilinxovih FPGAjih po prenosni logiki



Slika 3.3: Gradnik CARRY4, oziroma po slovensko prenosna logika, ki jo najdemo v logičnih celicah Xilinxovih čipov. Kot zakasnilne elemente uporabljamo multiplekserje z oznako *MUXCY*. Slika vzeta iz [11].

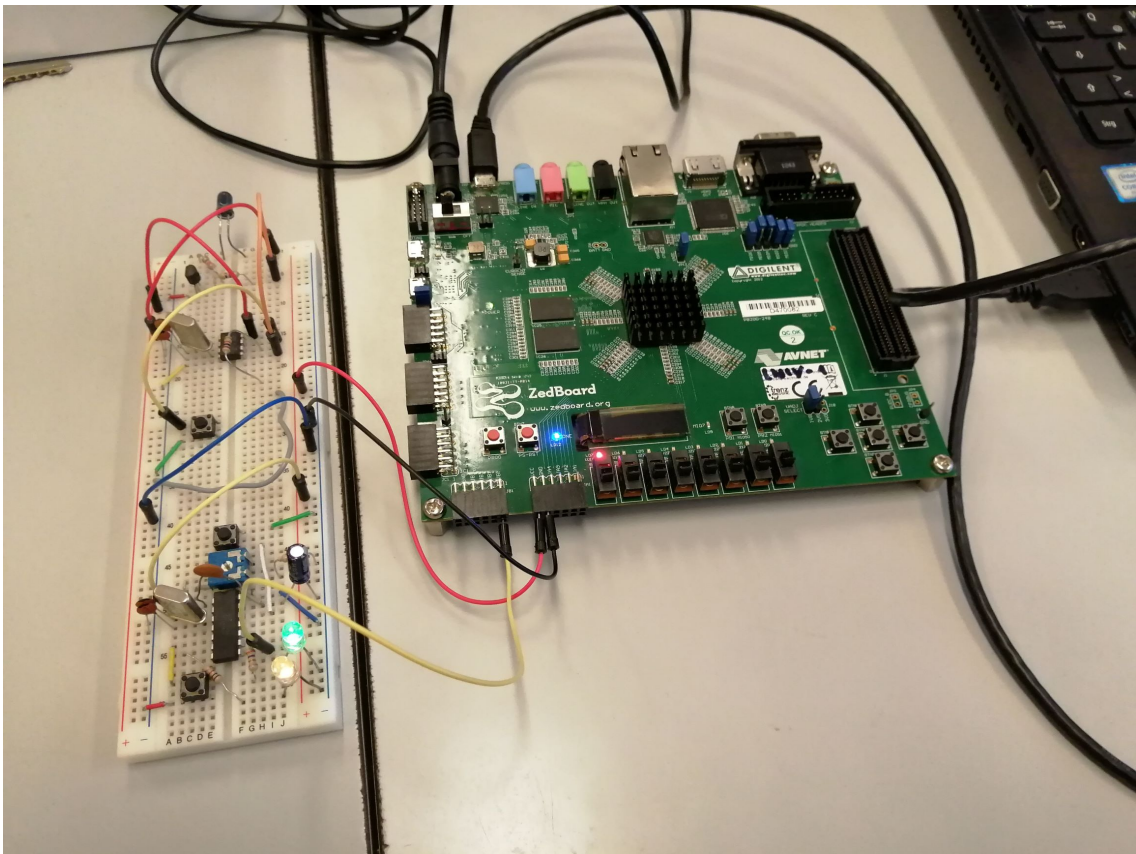
potovala hitreje od negativne. Točen razlog ni znan, zadeva je bila ugotovljena popolnoma empirično in je značilna za Xilinxove čipe, poročajo v viru [12].

Enkoder po načelu štetja enic

Izhod zakasnilne linije je 192-bitni signal v obliki termometriške kode (po analogiji s kapljevinskim termometrom), ki jo je treba pretvoriti v 8-bitno število. To je naloga naslednje komponente, enkoderja. Obstaja več pristopov k problemu, ampak nazadnje sem se odločil za metodo 'štetja enic' zaradi svoje enostavnosti in hkrati odpornosti proti tako imenovanim 'mehurčkom' v kodi [13]. Pri tej metodi samo

preštejemo, koliko enic je na izhodu linije, in to je naš rezultat. Ravnoakar omenjeni mehurčki lahko nastanejo zaradi metastabilnosti flip-flopov na liniji in neenake zakasnitve urinega signala do vsakega izmed njih (angl. *clock skew*) [12, 13, 14]. Na primer, meritev, ki bi morala izgledati kot '11110000...', se pojavi kot '11101000...', torej z mehurčkom na koncu; med enice se je vrnila ena ničla. Kljub temu, da se je položaj zadnje enice premaknil, je skupno število visokih bitov še vedno enako.

Pojav mehurčkov na izhodu zakasnilne linije ni samo hipoteza, ampak jih lahko zares opazujemo na pravem čipu. Za ta namen sem uporabil možnost, ki jo ponuja Vivado, imenuje pa se Integrirani Logični Analizator oziroma ILA. Uporabniki starejšega orodja *ISE* bodo isto stvar prepoznali pod drugačnim imenom *ChipScope*. Gre za to, da lahko svojemu vezju dodamo Xilinxovo jedro, logični analizator, ki se skupaj z našim vezjem implementira na FPGA. To nam omogoča, da v realnem času spremljamo dogajanje na internih signalih v čipu med samim delovanjem. Potrebujemo vmesnik USB-JTAG, ki ga Red Pitaya nima, zato sem za ta del uporabil ploščo ZedBoard s sestrskim čipom Zynq-7020, kar lahko vidimo na sliki 3.4.



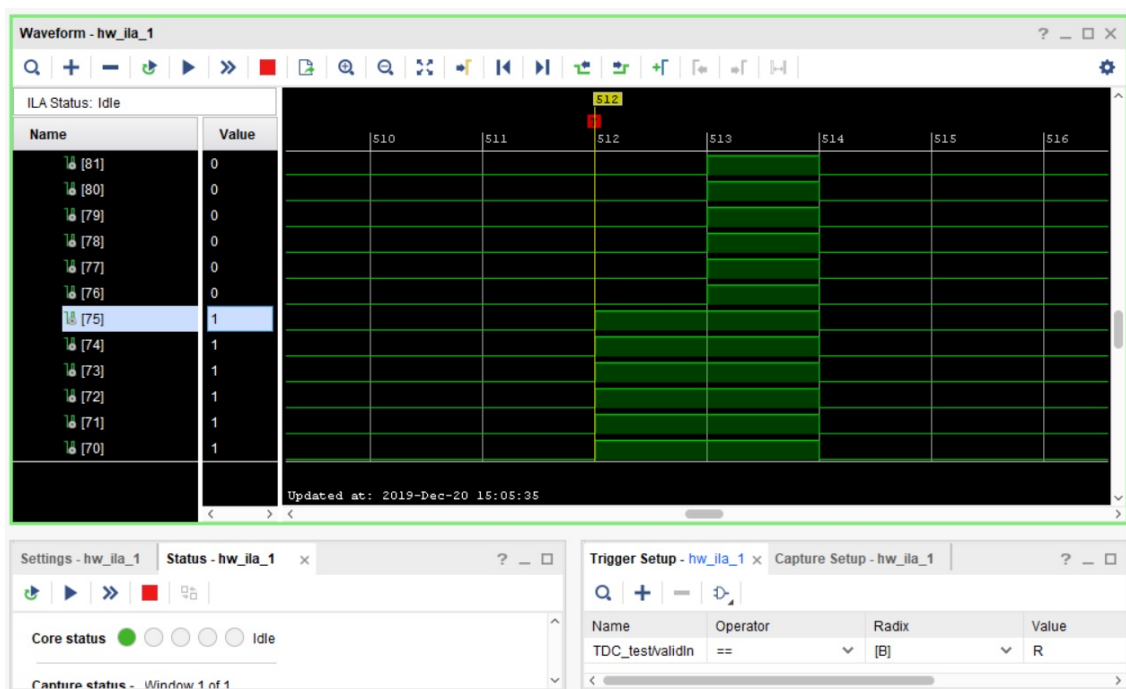
Slika 3.4: ZedBoard s čipom Zynq-7020, ki je preko vmesnika JTAG povezan z računalnikom. Levo je preizkusna ploščica z vezjem, ki na vhod TDCja pošilja asinhrono sunke.

V orodju Vivado se povežemo z implementiranim ILA, nastavimo proženje in že lahko opazujemo dogajanje na izbranem vodilu, v tem primeru izhodu iz zakasnilne linije. Slika 3.5 prikazuje rezultat, kjer smo ujeli pravilno termokodo brez mehurčkov s 76 enicami (štejemo od 0, signal se širi od spodaj navzgor). Veljavna koda je prva za navpično rumeno črto; med kasnejšimi vzorčenji se linija že ponastavlja. Zdaj je

Poglavje 3. Načrtovanje in implementacija časovno-digitalnega pretvornika

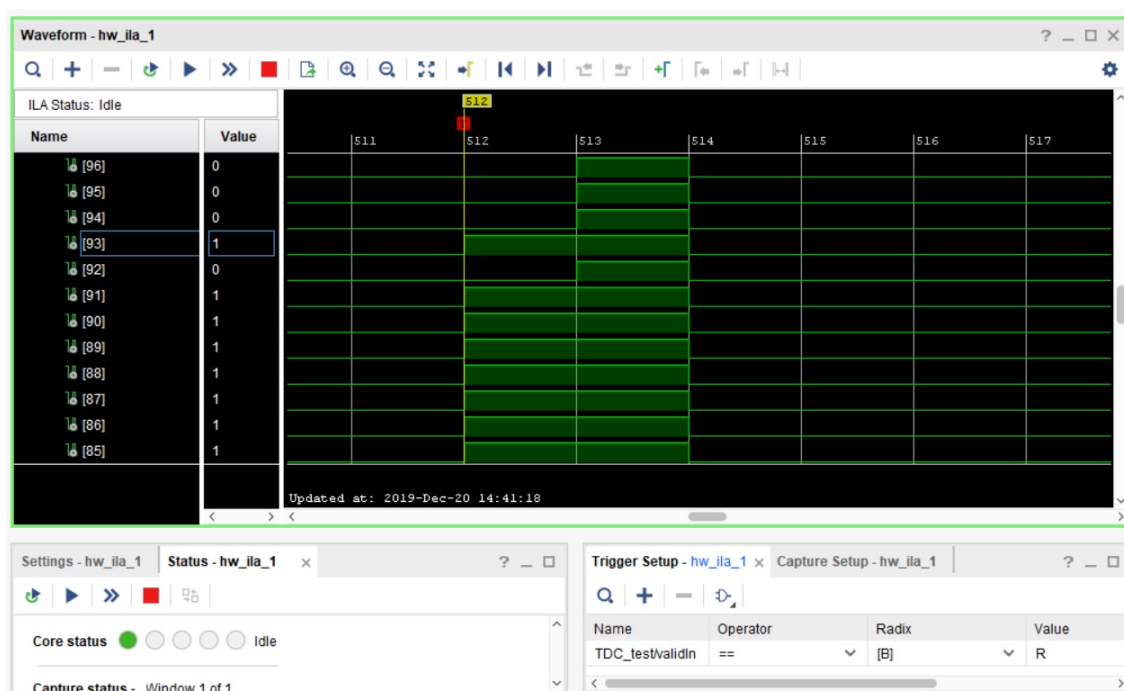
verjetno popolnoma jasno, od kod ime ‘termometerska’ koda.

Slika 3.6 je že drugačna, tokrat z mehurčkom na koncu, ki pa ne pokvari končnega rezultata na izhodu enkoderja. Kot zanimivost prilagam še sliko 3.7, kjer sem ujel grd dvojni mehur ob ponastavljanju linije, torej ob prehodu negativne fronte. Pri veljavnih kodah takšnih hujših mehurčkov ni bilo opaziti, kar je verjetno povezano z različno hitrostjo širjenja pozitivnega in negativnega roba.

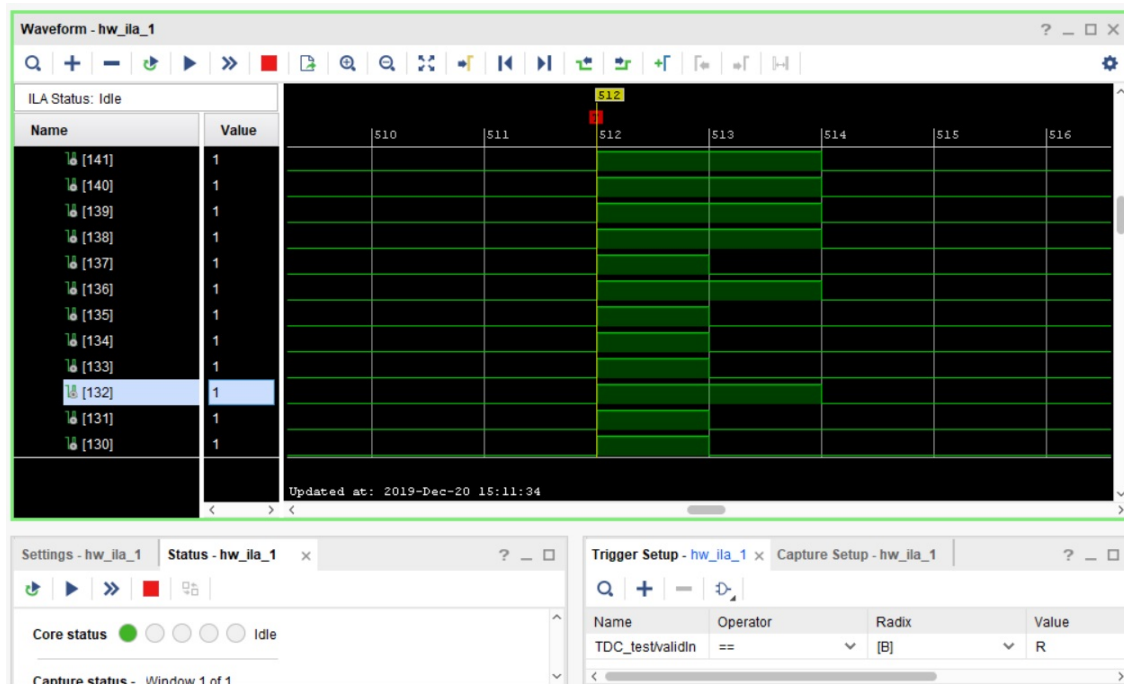


Slika 3.5: Primer pravilne termometerske kode brez mehurčkov. Posamezni stolpci so urini cikli, torej vzorčenja zakasnilne linije. Veljavna koda je prva meritev za navpično rumeno črto.

Zdaj se lahko posvetimo sami zasnovi enkoderja. Štetje ni nič drugega kot prištevanje enic, zato moramo v principu le sešteti vse bite v termokodi. Vendar hitro naletimo na težavo, saj bi naivna uporaba VHDL operacije ‘+’ v *for* zanki ustvarila izredno počasno vezje, ker bi potrebovali 192 zaporedno vezanih seštevalnikov. Potreben je drugačen pristop, s strukturo, ki jo vidimo na sliki 3.8. V prvi fazi s skupinami treh 6-vhodnih vpoglednih tabel proizvedemo 3-bitne binarne vsote števila enic na vhodu. Za zapis števila 6 namreč potrebujemo tri bite, vsaka vpogledna tabela (LUT) pa ima le en izhod, zato potrebujemo tri. Ta prva faza seštevanja je bila v 6-vhodne LUT vnešena ročno s pomočjo [15]. V naslednjem koraku je treba vse 3-bitne vsote sešteti v 8-bitni končni rezultat, kar storimo s seštevalnim drevesom (slika 3.8). Absolutno kritičnega pomena je, da jedro časovno-digitalnega pretvornika deluje na čim višji frekvenci ure [12]; nižja frekvenca namreč zahteva daljšo linijo, kar močno povečuje površino vezja. Precej neugodno, če bi radi na koncu implementirali veliko kanalov. Prav zato je vsaka stopnja seštevanja svoj del cevovoda, ki omogoča najvišjo frekvenco delovanja. Za opis takšne strukture je bilo potrebno v VHDL uporabiti rekurzijo. Slika 3.9 prikazuje del sheme RTL [5] generiranega seštevalnega drevesa, ki spominja na rusko matricjoško. Ena in ista enota, nivo registrov, ki mu sledi seštevanje, se ponavlja znotraj same sebe. Moram



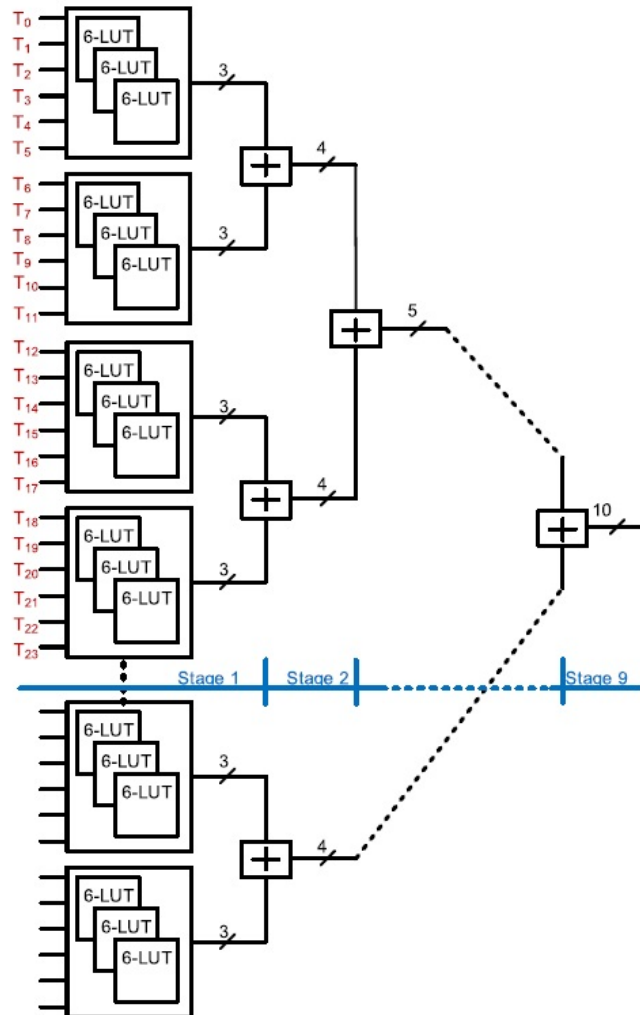
Slika 3.6: Termometrski kod z mehurčkom na koncu, ki enkoderju s štetjem enic ne predstavlja težav.



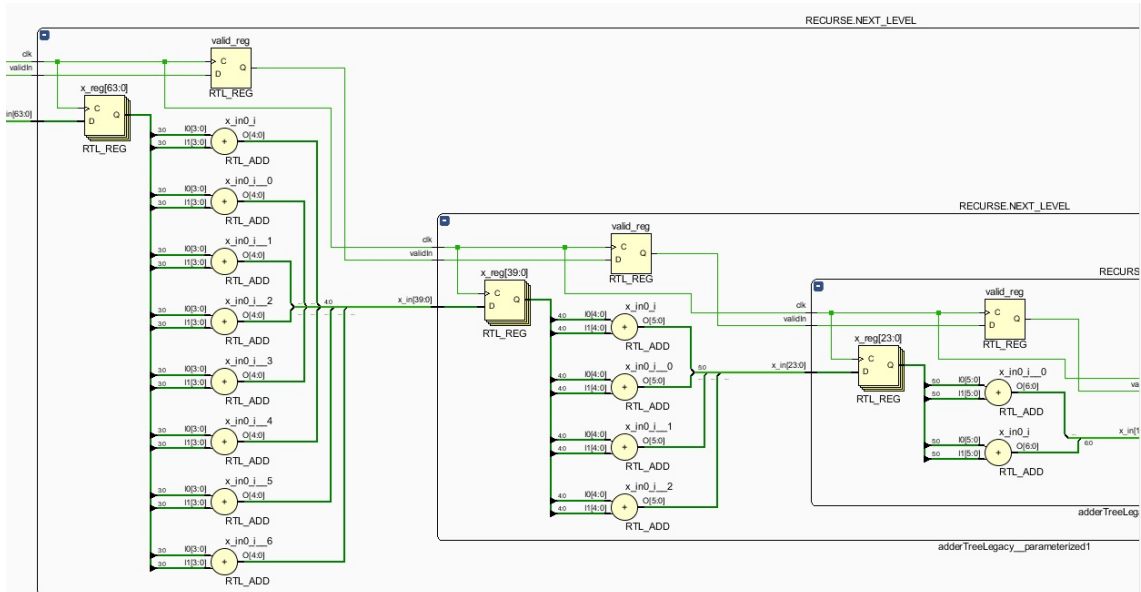
Slika 3.7: Dvojni mehurček ob prehodu negativne fronte (drugi stolpec).

Poglavje 3. Načrtovanje in implementacija časovno-digitalnega pretvornika

priznati, da sva se s prof. Trostom ob tem soočila z nemalo težavami, saj rekurzija v strojno opisnem jeziku, kot je VHDL, ni nekaj, kar bi si človek želel početi ravno vsak dan. Vseeno je drevo na koncu zelo dobro uspelo in deluje odlično.



Slika 3.8: Arhitektura enkoderja po principu hitrega seštevanja enic. Vsaka stopnja je del cevovoda, registri niso prikazani. Prvo fazo štetja opravijo vpogledne tabele, nato prevzamejo seštevalniki. Prikazani primer uporablja daljšo zakasnilno linijo od naše, zato je izhod 10-bitni. Povzeto po [13].



Slika 3.9: Del sheme RTL seštevalnega drevesa, kot jo prikaže Vivado. Rekurzija ustvari nenavadno strukturo, ki se ponavlja znotraj same sebe.

Grobi števec

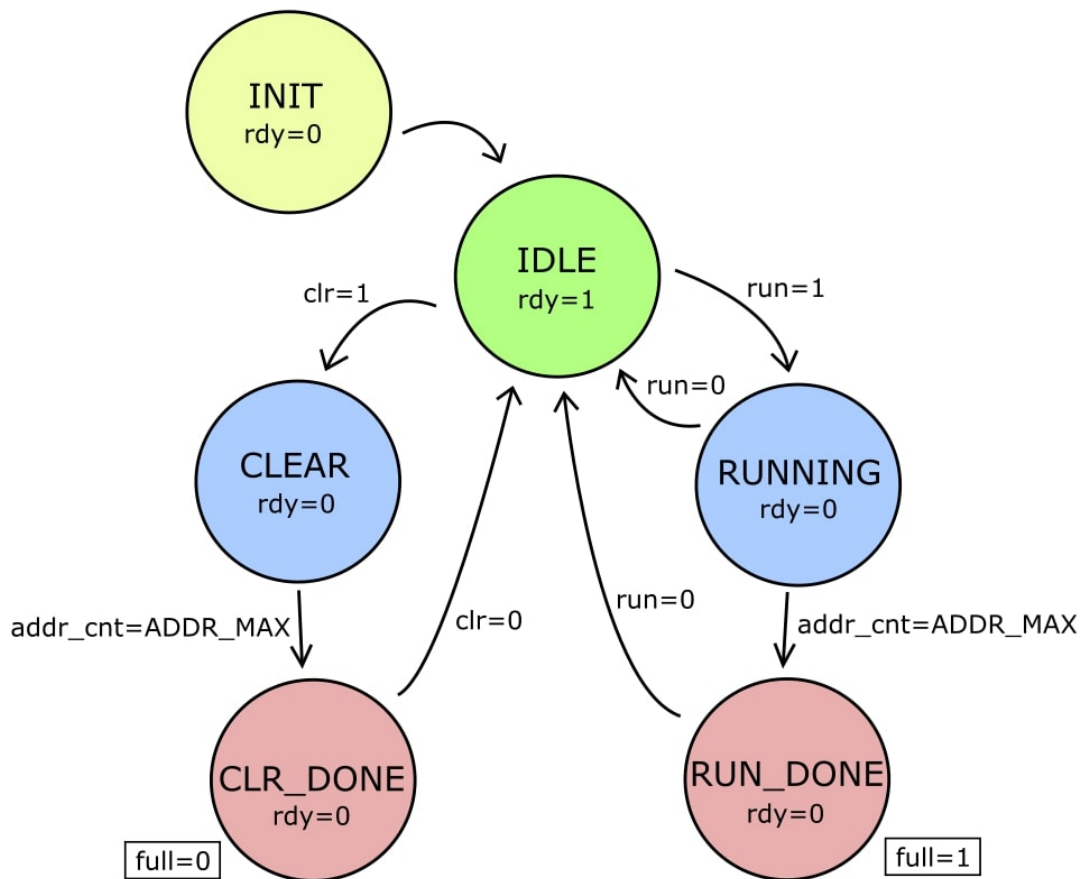
Izhod enkoderja je 8-bitna fina časovna meritev TDCja. Skupaj s 24-bitnim izhodom binarnega števca tvorita 32-bitno časovno značko, ki se pošlje kontrolni enoti (glej sliko 3.1). 24-bitni števec je povsem običajni binarni števec, ki ob vsaki fronti ure prišteje +1. Števec je vedno aktiven in začne takoj ob prenosu konfiguracije na FPGA. Ker ima vsak kanal TDC svoj števec, je na ta način poskrbljeno, da so vsi sinhronizirani med sabo, sicer časovne značke ne bi bile primerljive.

Kontrolna enota

Zadnji blok jedra TDC je kontrolna enota, ki upravlja delovanje kanala ter zbira časovne značke in jih pošilja dalje v shrambo na integrirani blokovni RAM pomnilnik (BRAM) [16]. Realizirana je v obliki sinhronnega avtomata (stroja stanj) s šestimi stanji in štirimi kontrolnimi signali. Preko signalov *run* in *clr* TDC sprejema ukaze, preko *rdy* in *full* pa javlja svoj status. Slika 3.10 prikazuje diagram prehajanja stanj kontrolnega avtomata.

Ob zagonu se avtomat znajde v stanju INIT in že takoj ob naslednjem ciklu skoči v IDLE, kjer čaka ukaze (zastavica *rdy* = 1 signalizira, da je TDC kanal na voljo). Če želimo pognati TDC, postavimo bit *run* = 1 in avtomat skoči v stanje RUNNING. V tem stanju zajemamo časovne značke pulzov na vходу časovno-digitalnega pretvornika, ki se pošiljajo v BRAM, na katerem je prostora za 2048 značk na kanal. TDC je aktiven dokler ne ustavimo zajemanja z *run* = 0 ali pa dokler se BRAM do konca ne zapolni; v slednjem primeru se avtomat premakne v RUN_DONE in postavi zastavico *full* = 1, ki signalizira, da je kanal poln. Preden lahko nadaljujemo, je treba v vsakem primeru resetirati *run* nazaj na '0', da se avtomat vrne v IDLE.

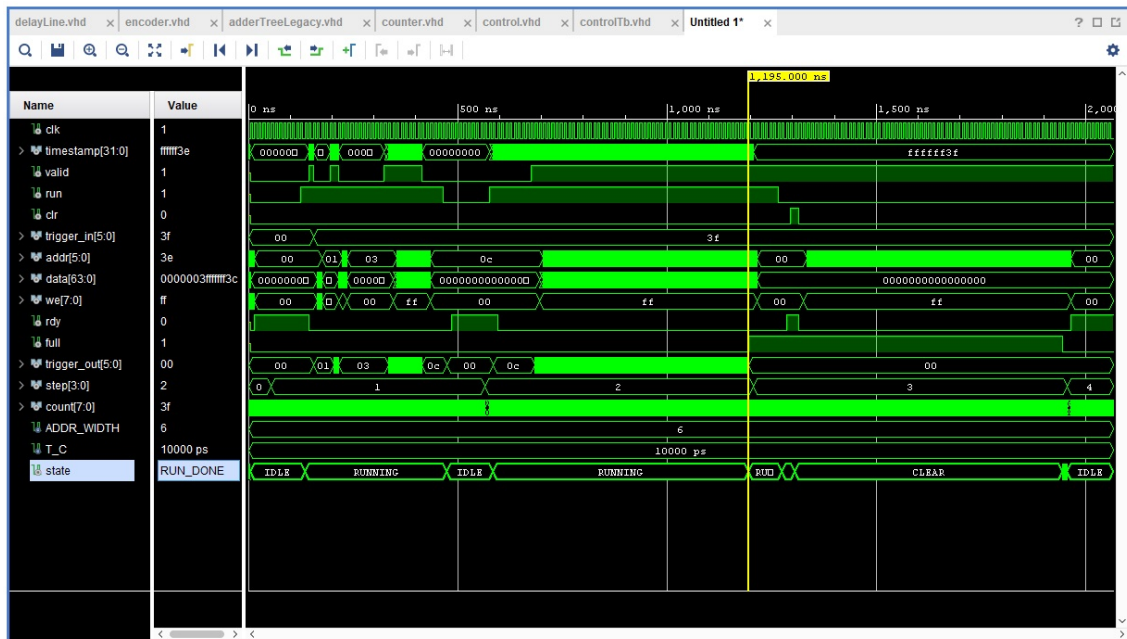
Ob postavitvi bita *clr* na '1' se sproži brisanje BRAMa. Avtomat skoči v stanje CLEAR in začne vpisovati ničle v vseh 2048 mest, ki so na voljo. Ko konča, gre v



Slika 3.10: Diagram prehajanja stanj kontrolnega avtomata.

CLR_DONE in ponastavi zastavico *full* nazaj na '0'. Tako kot pri *run* moramo tudi tu resetirati *clr*, da se vrnemo v IDLE. V primeru napake, če bi se avtomat znašel v kakšnem nedefiniranem stanju (*others*), se kontrolna enota resetira preko stanja INIT.

Pravilno delovanje kontrolne enote lahko preverimo v simulaciji, izsek iz katere lahko vidimo na sliki 3.11.



Slika 3.11: Simulacija delovanja kontrolnega avtomata v orodju Vivado.

3.1.2 AXI TDC

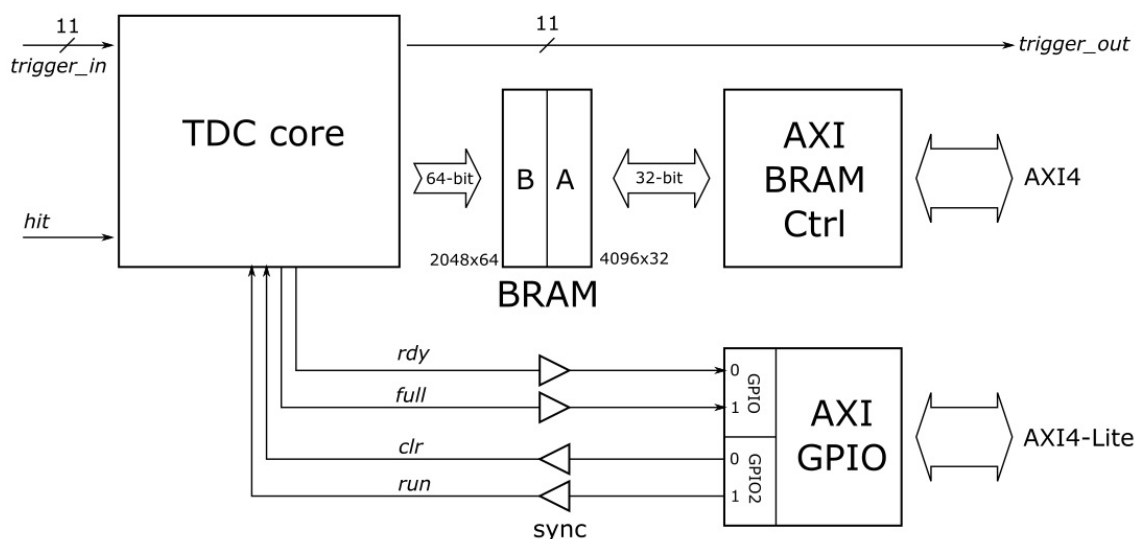
Na naslednji stopnji je jedru časovno-digitalnega pretvornika potrebno dodati vmesnik AXI, da bomo TDC lahko vključili v sistem. AXI oziroma *Advanced eXtensible Interface* je standard za komunikacijo komponent na čipu, ki sta ga skupaj razvila ARM in Xilinx, in se na čipu Zynq uporablja za povezovanje strojnih komponent s procesnim sistemom [17]. Radi bi, da procesor lahko bere časovne značke, shranjene v BRAMu, in upravlja s posameznim kanalom TDC preko prej omenjenih kontrolnih signalov.

Z načrtovanjem vezja v jeziku VHDL smo več ali manj zaključili, zato se selimo nivo višje, na nivo povezovanja večjih in kompleksnejših komponent. Xilinx v svojem okolju Vivado ponuja brezplačna jedra IP, oziroma *Intellectual Property cores*, ki jih najprej pravilno nastavimo in nato vključimo v svoj sistem. Na ta način našemu jedru TDC dodamo tri Xilinxove komponente: blokovi RAM in njegov vmesnik AXI ter vmesnik AXI za kontrolne signale. Rezultat je shema na sliki 3.12.

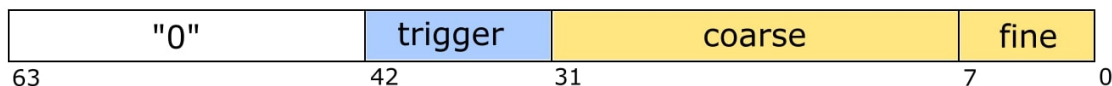
Vezju smo dodali še signal *trigger*, ki bo prišel prav kasneje, ko bomo gradili večkanalne sisteme, na samo delovanje TDCja pa ne vpliva. Gre za to, da 32-bitni časovni znački pripnemo še 11-bitni naslov *trigger_in* (kar ustreza 2048 besedam v BRAMu), naslov značke, ki jo trenutno vpisujemo, pa pošljemo na *trigger_out*. Na ta način lahko neodvisne TDC kanale povezujemo v vzročno-posledična razmerja, kar potrebujemo za medkanalne meritve, kot je tista na sliki 1.1. Poimenovanje *trigger* je nekoliko neposrečeno, ker ta signal ničesar ne proži, ampak znački samo dodaja informacijo nekega drugega kanala.

Značke, ki se vpisujejo v blokovi RAM, je bilo treba torej razširiti na 64 bitov, njihovo strukturo pa prikazuje slika 3.13. Blok RAM oziroma BRAM je pomnilnik, ki je tipično na voljo na Xilinxovih čipih FPGA v kosih po 32 kilobitov, za naše potrebe 2048 64-bitnih značk pa smo torej porabili 4 takšne bloke [16]. Pomnilniku dodamo še komponento *AXI BRAM Controller* [18], ki BRAM poveže na

Poglavje 3. Načrtovanje in implementacija časovno-digitalnega pretvornika



Slika 3.12: Shema kanala TDC s pomnilnikom BRAM in vmesniki AXI. Urin signal je zaradi preglednosti izpuščen.

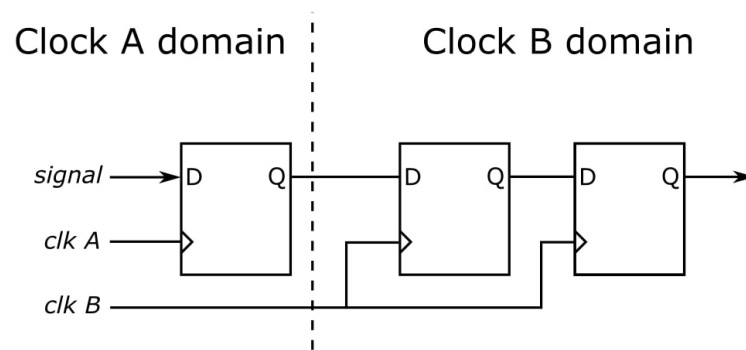


Slika 3.13: Sestava časovne značke, ki se vpiše v BRAM. Vsebuje 8-bitno fino vrednost, 24-bitno vrednost grobega števca in 11-bitni 'trigger'. Zgornjih 21 bitov je praznih.

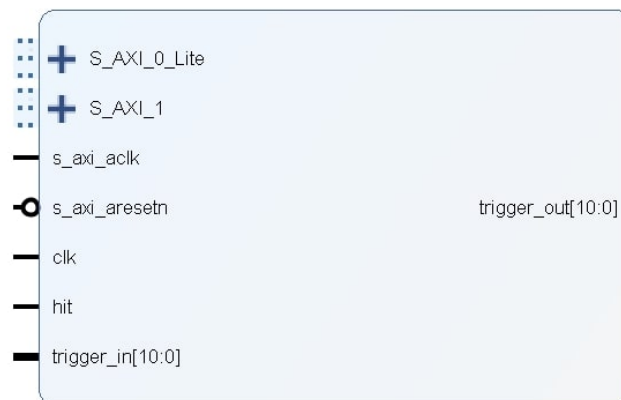
podatkovno vodilo AXI4. Ena ključnih prednosti pomnilnika BRAM je v tem, da je mogoče do podatkov dostopati prek dveh neodvisnih 'vrat', ki jih označimo z 'A' in 'B'. Kontrolna enota v vrata B vpišuje 64-bitne značke (glej sliko 3.12), medtem ko je vodilo AXI 32-bitno. To ni težava, ker vrata A lahko nastavimo kot 4096 32-bitnih celic, za branje ene 64-bitne značke pa bosta z AXI strani torej potrebna dva urina cikla.

Štiri kontrolne signale *rdy*, *full*, *clr* in *run* povežemo na komponento *AXI GPIO* [19], ki se poveže na vodilo AXI4-Lite. Gre za 'oskubljeno' verzijo vodila AXI4, ki je primerna za manj zahteven 'promet'. Na tem mestu je treba poudariti, da med jedrom TDC in vmesniki AXI pride do tako imenovanega prečkanja časovnih domen iz ene frekvence ure na drugo, na kar je treba pri načrtovanju digitalnih vezij zelo dobro paziti. Jedro TDC deluje na cca. 350 MHz, medtem ko vodilo AXI tiktaka na precej nižjih 100 MHz. Za BRAM to ni problem, ker so vrata A in B med seboj neodvisna in omogočajo dostop do podatkov z različnima frekvenca. Drugače je pri kontrolnih linijah, na katere moramo vstaviti sinhronizatorje (na sliki 3.12 so označeni z malimi trikotniki). Zgradbo takšnega sinhronizatorja vidimo na sliki 3.14. Gre enostavno za podvojen flip-flop na strani nove ure. Takšen pristop deluje le za enobitne signale, kakršni so tudi naši štirje.

Tako, pa smo prišli do konca izdelave posameznega kanala TDC. Časovno-digitalni pretvornik z vmesnikom AXI na sliki 3.12 zapakiramo v svoje jedro IP, ki



Slika 3.14: Vezje za varen prehod iz ure A na uro B. Prvi flip-flop na strani B lahko postane metastabilen, zato mora biti podvojen.



Slika 3.15: AXI TDC smo zapakirali v novo jedro IP, ki ga bomo uporabili v naslednji fazi načrtovanja.

ga bomo uporabili na naslednji stopnji: gradnji celotnega sistema časovno-digitalnih pretvornikov. Na sliki 3.15 lahko vidimo, kako izgleda naša novo ustvarjena komponenta. Ima vhod za uro TDC in prihajajoče pulze, vhodno/izhodni 'trigger' in dva vmesnika AXI; AXI4 za branje značk ter AXI4-Lite za upravljanje kanala.

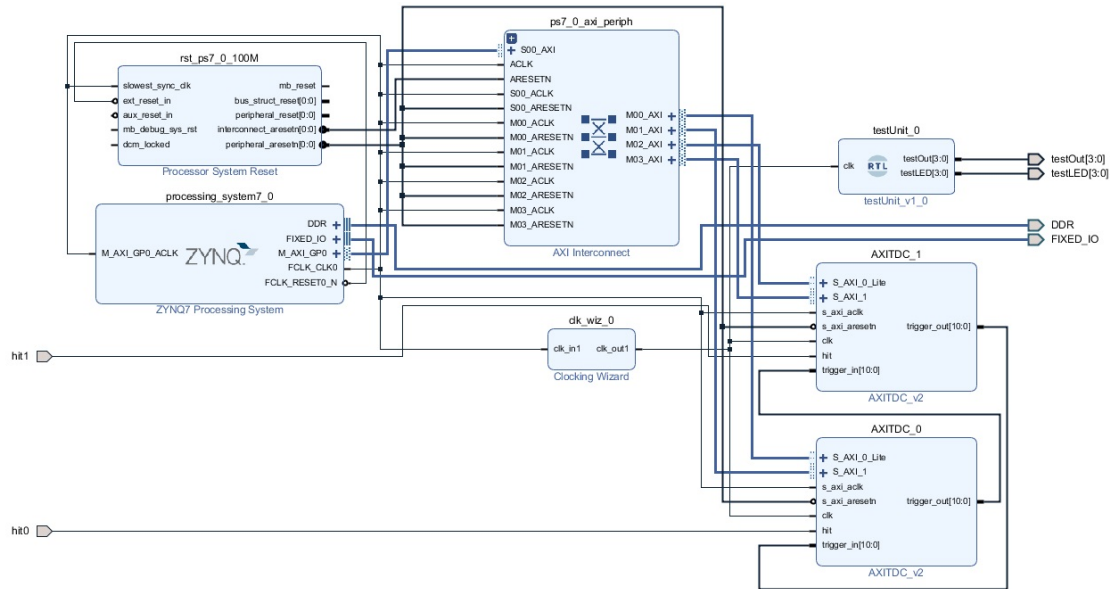
Kot sem rekel na začetku; Lego kocke. Lepo po vrsti, od osnovnih gradnikov do celotnega sistema. To je vsa umetnost načrtovanja digitalnih sistemov, naj so še tako kompleksni.

3.2 Sistem TDC

Selimo se na najvišji nivo načrtovanja našega sistema na čipu Zynq, kjer bomo ustvarjene kanale TDC povezali v delujoč digitalni sistem. V orodju Vivado izberemo 'Block Design', ki nam omogoča grafično povezovanje in konfiguriranje jeder IP. Izdelava večkanalnega sistema časovno-digitalnih pretvornikov je zelo preprosta; v shemo dodamo več jeder AXI TDC. Če v načrt vključimo še blok 'Zynq Processing System', orodju sporočimo, da bi radi strojno opremo na FPGAju povezali s procesnim sistemom čipa (slika 2.1). To avtomatsko doda še vse ostale komponente, ki

Poglavje 3. Načrtovanje in implementacija časovno-digitalnega pretvornika

so za to potrebne, vključno z logiko za AXI povezljivost (blok ‘AXI Interconnect’). Čarovnik poskrbi, da se vsa vodila AXI pravilno povežejo. Rezultat je sistem na sliki 3.16, v katerega smo dodali dva kanala TDC in še eno majceno komponento za testiranje, ki v tem primeru ni pomembna.



Slika 3.16: Blokovna shema dvokanalnega sistema TDC, kot jo prikaže Vivado IP Integrator. Generiranje takšnega načrta je visoko avtomatizirano in od načrtovalca ne zahteva skoraj nobenega truda. Dodati je treba le nekaj osnovnih jeder IP.

Oba kanala povežemo med sabo preko 11-bitne linije *trigger*, da ju bomo lahko uporabljali v medkanalnih meritvah. Ta signal sicer lahko ostane nepovezan in ne vpliva na osnovno delovanje časovno-digitalnega pretvornika. Priključki AXI4 in AXI4-Lite se avtomatsko povežejo na AXI Interconnect, merilna vhoda vsakega kanala pa vodimo na zunanje priključke čipa (signala *hit0* in *hit1*). Ostane nam še ura. AXI logika deluje na frekvenci 100 MHz, ki jo zagotavlja izhod procesnega sistema *FCLK_CLK0*. Ker pa naši jedri TDC potrebujejo precej višjo frekvenco, moramo vmes vstaviti komponento MMCM oziroma *Mixed-Mode Clock Manager* [20], vsebovano v bloku ‘Clocking Wizard’. MMCM temelji na fazno vklenjeni zanki (PLL), ki je sposobna iz 100 MHz ure sintetizirati 350 MHz, ki jih vodimo na *clk* vhod časovno-digitalnih pretvornikov. Ta frekvenca je bila določena tako, da sem vezje poskušal implementirati pri različnih hitrostih in 350 MHz je bila najvišja vrednost, pri kateri so bile še izpolnjene časovne zahteve.

S tem, ko smo enoti TDC preko vodila AXI povezali na procesni sistem, sta naša časovno-digitalna pretvornika postala periferni enoti procesorja ARM na čipu Zynq. Še preden pa lahko dostopamo do njiju, jima je treba dodeliti naslove oziroma ju umestiti v naslovni prostor. To storimo v zavihku ‘Address Editor’ programa Vivado, kot prikazuje slika 3.17. Najmanjša enota naslovnega prostora, ki jo lahko dodelimo, so štirje kilobajti (4K), kar priredimo kontrolnim signalom. Velikost BRAM, po drugi strani, je 4096x32 bitov (slika 3.12), torej 4096x4 bajtov (1 bajt = 8 bitov), se pravi 16 kilobajtov.

Na koncu poženemo implementacijo, ki celotno vezje preslika na logične celice v čipu FPGA in jih poveže med sabo. Rezultat je prikazan na sliki 3.18, kjer

Cell	Slave Interface	Base Name	Offset Address	Range	High Address
processing_system7_0					
Data (32 address bits : 0x40000000 [1G])					
AXITDC_0	S_AXI_1	mem1	0x43C1_0000	16K	0x43C1_3FFF
AXITDC_0	S_AXI_0_Lite	reg0	0x43C0_0000	4K	0x43C0_0FFF
AXITDC_1	S_AXI_1	mem1	0x43C3_0000	16K	0x43C3_3FFF
AXITDC_1	S_AXI_0_Lite	reg0	0x43C2_0000	4K	0x43C2_0FFF

Slika 3.17: Izsek iz zavihka, kjer komponentama AXI TDC priredimo naslove za dostop. Vsaka izmed obeh enot ima dva naslova; enega za BRAM (*mem1*) in drugega za kontrolne signale (*reg0*).

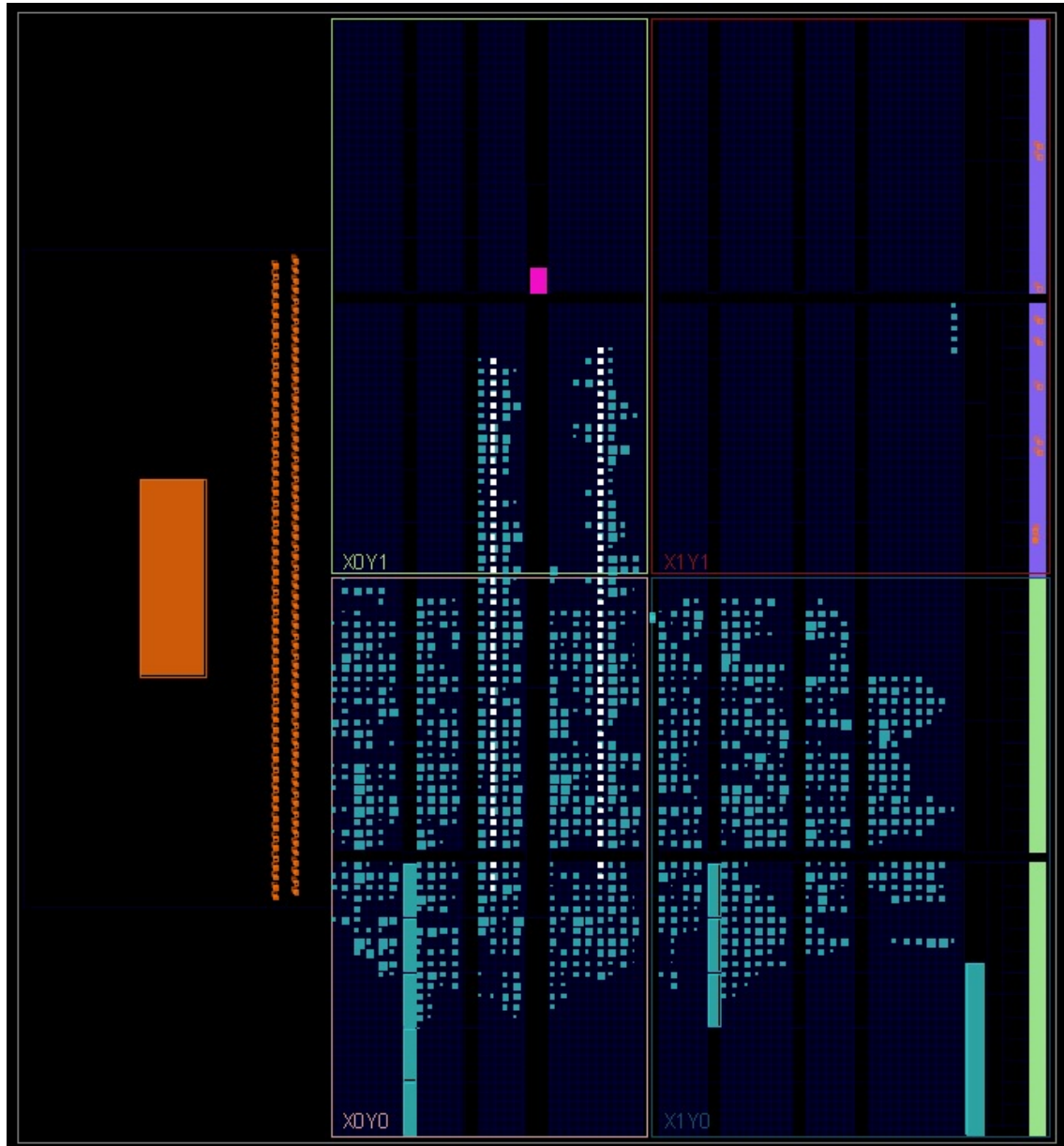
lahko vidimo strukturo čipa Zynq 7010 in uporabljene elemente. Z oranžno označen blok na levi je procesni sistem, ostalo pa je programirljiva logika, torej FPGA. Uporabljene celice so pobarvane modro. Z belo sem označil tudi obe zakasnilni liniji. Lepo se vidi, da sta obe ravni in neprekinjeni, kakor smo si tudi želeli.

Tabela 3.1 prikazuje zasedenost vezja po implementaciji. Poraba logičnih gradnikov je razdeljena po komponentah, ki smo jih predstavili tekom tega poglavja; najprej golega jedra TDC, nato komponente AXI TDC in nazadnje celotnega sistema. Za primerjavo so prikazani podatki za enokanalni in dvokanalni sistem TDC. Golo jedro TDC zasede zelo malo, komajda 2 %. Nekaj podobnega lahko rečemo za

Tabela 3.1: Poročilo o zasedenosti čipa

	LUT (17600)		Flip-flopi (35200)		BRAM (60)	
Jedro TDC	261	1.5 %	784	2.2 %	0	0 %
AXI-TDC IP	549	3.1 %	1084	3.1 %	4	6.7 %
1-kanalni sistem	~1400	8.0 %	~2100	6.0 %	4	6.7 %
2-kanalni sistem	2575	14.6 %	4185	11.9 %	8	13.3 %

komponento AXI TDC, ki lepo izravna uporabo vpoglednih tabel in flip-floпов na cca. 3 %. Vendar pa se situacija močno spremeni, ko zgradimo celoten enokanalni sistem; ob tem se poraba strmo poveča na 8 % LUT in 6 % FF, predvsem zaradi logike za AXI povezljivost. Upali bi, da z dodajanjem novih kanalov zasedenost za logiko AXI ostaja približno enaka, a na žalost ni tako. Dvokanalni sistem namreč zasede skoraj dvakrat toliko logike kot enokanalni, kar pomeni, da vsak kanal s sabo pripelje enako znatno količino zahtev po AXI povezljivosti. Čeprav je pristop preslikave komponent TDC v naslovni prostor, ki smo ga ubrali tu, najbolj enostaven, verjetno ni najbolj primeren za gradnjo velikih sistemov z mnogimi kanali. V



Slika 3.18: Implementacija vezja na čipu Zynq-7010. Poleg zakasnilnih linij je dobro vidnih tudi 8 uporabljenih blokov BRAM (4 na kanal) in enota MMCM desno spodaj.

takšnem primeru bi bilo bolj smiselno, če bi TDCji direktno pretakali podatke v sistemski pomnilnik preko DMA (angl. za ‘Direct Memory Access’), kar bi omogočalo tudi veliko hitrejši prenos podatkov iz časovno-digitalnih pretvornikov na procesni sistem. Vseeno se za takšno rešitev nisem odločil zaradi bistveno večje kompleksnosti izdelave in potencialne nekompatibilnosti z operacijskim sistemom, ki se trenutno uporablja na Red Pitayi.

Po implementaciji ustvarimo datoteko za programiranje oziroma konfiguracijo čipa FPGA. Naše vezje je s tem pripravljeno na uporabo.

Programska oprema

Red Pitayo poganja nekoliko prilagojen operacijski sistem Linux, ki teče na centralno procesni enoti čipa Zynq. Da lahko uporabljamo dodano strojno opremo, sem v jeziku C napisal program *TDCserver*, ki preko systemskega klica *mmap* dostopa do obeh kanalov TDC na naslovih, ki smo jih izbrali prej (slika 3.17). Program lahko upravlja s kanaloma in bere časovne značke, prav tako pa ima možnost branja temperature čipa, saj ima le-ta integriran termometer. Vsebuje tudi TCP/IP strežnik, ki prebrane značke pošilja odjemalcu preko Etherneta. Za strežniški del programa sem lahko uporabil kodo Antona Potočnika [21], za kar sem mu neskončno hvaležen.

Na drugi strani Ethernet kabla je osebni računalnik z grafičnim vmesnikom oziroma TCP/IP odjemalcem, ki se uporablja za nastavljanje časovno-digitalnih pretvornikov in ustrezno prikazovanje izmerjenih časovnih značk. Grafični vmesnik je spisan v App Designerju programa MATLAB R2017a, preizkušen pa je z operacijskim sistemom Windows 10. Več o delovanju in uporabi MATLAB aplikacije bo predstavljeno v naslednjem poglavju.

Prvo verzijo svojega časovno-digitalnega pretvornika in nekaj osnovnih meritev z njim sem predstavil oktobra 2019 na konferenci Austrochip na Dunaju, članek pa je na voljo v IEEE Xplore [22].

Poglavje 4

Karakterizacija inštrumenta

V tem poglavju se bomo posvetili karakterizaciji izdelanega merilnega inštrumenta. Sproti bo predstavljen tudi grafični vmesnik v MATLABu in kako se ga uporablja za opravljanje posameznih meritev. Najprej pa na hitro še, kako se Red Pitayo s TDCjem usposobi za delovanje.

Vhodna digitalna signala za prvi in drugi kanal sta vezana na navadna LV-CMOS33 FPGA priključka M14 in M15, ki na tiskanini vodita do pinov 17 in 18 na razširitvenem konektorju E1. Paziti je treba, da nanje pripeljemo signale ustreznih napetostnih nivojev, sicer lahko uničimo čip. Kot lahko preberemo v podatkovnem listu [23], je priporočeno območje vhodnih napetostih za LVCMOS33 od -0,20 do +3,50 V, logična enica pa zahteva nivo vsaj 2,00 V.

Pred vklopom TDCja je treba najprej znižati frekvenco na izhodu procesnega sistema *FCLK_CLK0*, ki ga naše vezje uporablja kot urin signal. Privzeta vrednost ob zagonu operacijskega sistema je 125 MHz, kar je potrebno znižati na načrtovanih 100 MHz, ker logika AXI povezljivosti hitreje ne zmore. To storimo tako, da poženemo skripto *PLclock*, ki vsebuje nekaj sistemskih ukazov:

```
#!/bin/bash

devcfg=/sys/devices/soc0/amba/f8007000.devcfg
test -d $devcfg/fclk/fclk0 || echo fclk0 > $devcfg/fclk_export
echo 1 > $devcfg/fclk/fclk0/enable
echo 100000000 > $devcfg/fclk/fclk0/set_rate
```

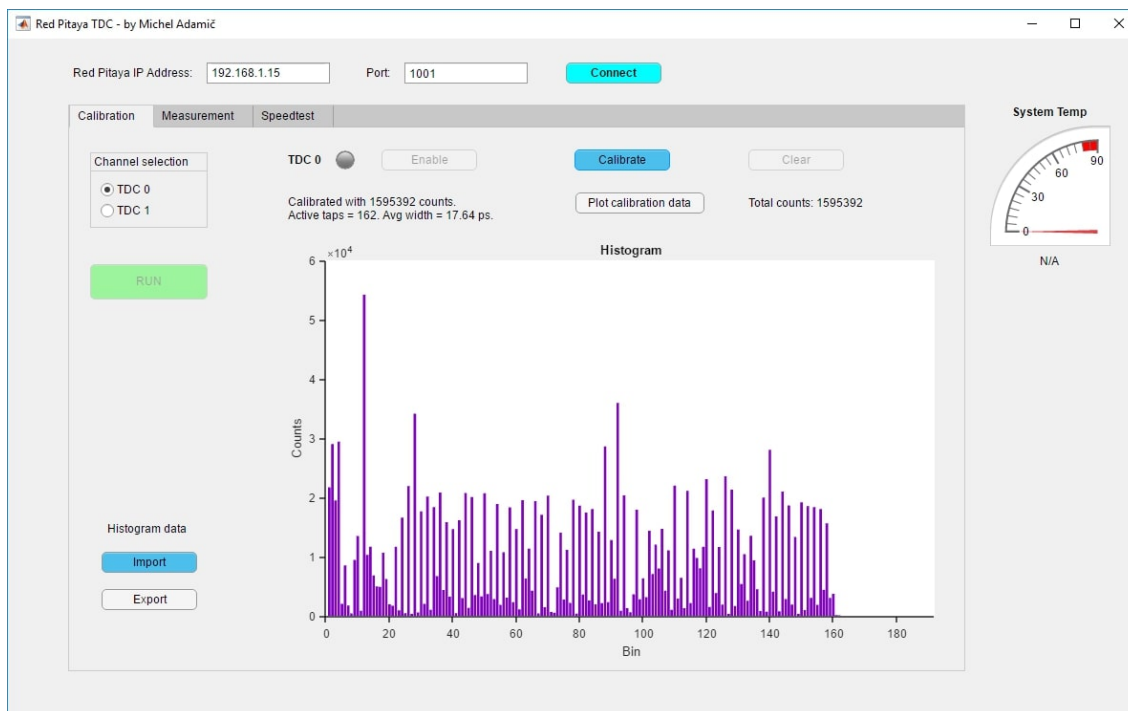
Nato na FPGA naložimo konfiguracijo, ki na čipu ustvari naše vezje s časovno-digitalnimi pretvorniki:

```
cat TDCsystem_wrapper3.bit > /dev/xdevcfg
```

Kot vidimo, je ukaz presenetljivo enostaven, ker sistem Linux vsebino naložene konfiguracije FPGA vidi kot nekakšno datoteko v mapi */dev*. Nazadnje samo še poženemo program *TDCserver* in s strani Red Pitaye je vse pripravljeno. Na prek Etherneta povezanem osebem računalniku v MATLAB aplikaciji kliknemo na gumb 'Connect' in če smo pravilno vnesli naslov IP ter vrata strežnika, lahko že beremo temperaturo čipa Zynq.

4.1 Kalibracija zakasnilne linije

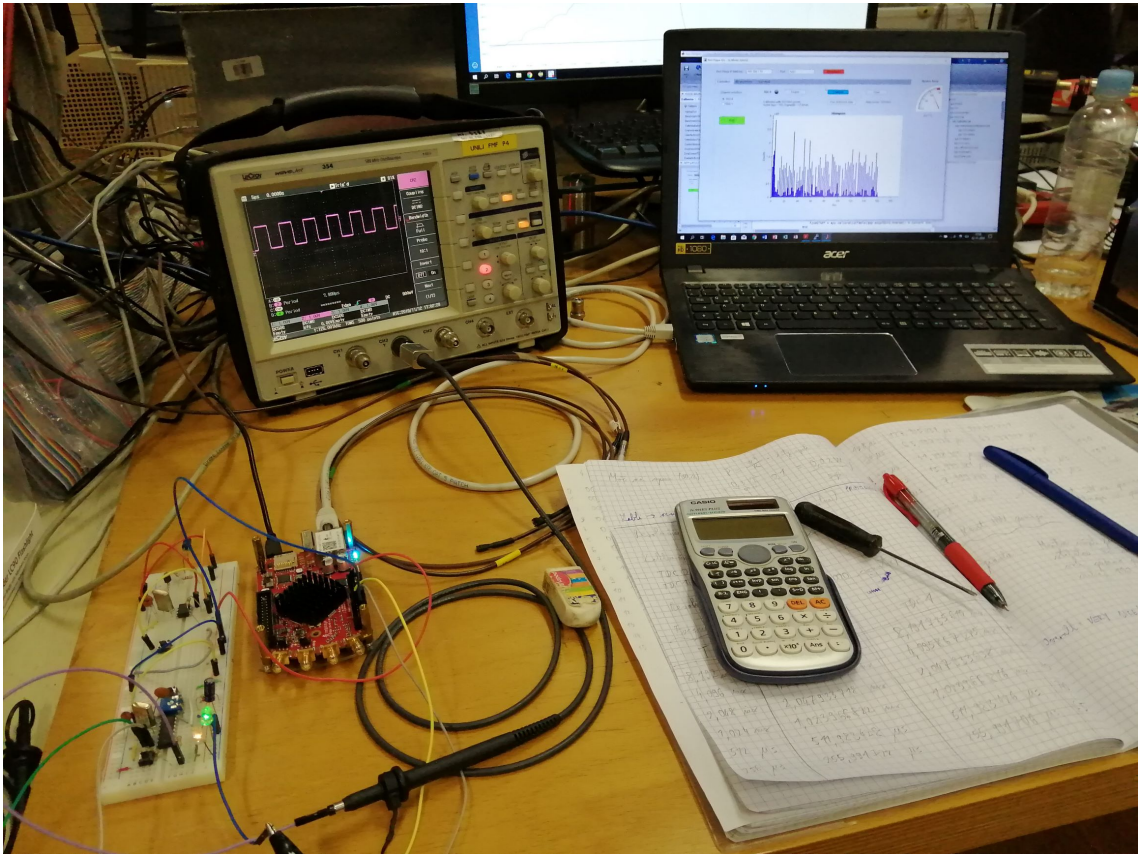
Prva stvar, ki jo moramo narediti, preden lahko začnemo uporabljati TDC, je kalibracija zakasnilne linije v posameznem kanalu. To opravimo v prvem zavihku MATLAB aplikacije, imenovanem ‘Calibration’, kar vidimo na sliki 4.1.



Slika 4.1: MATLAB aplikacija z odprtim zavihkom za kalibracijo. Opomba: na sliki povezava z Red Pitayo ni več vzpostavljena, zato so nekatere funkcije onemogočene, prav tako pa ni podatka o temperaturi sistema.

Potrebno je izmeriti zakasnitev vsakega elementa v zakasnilni liniji, da bomo lahko potem časovne značke pretvarjali v fizičen čas v pikosekundah. Idealno bi pričakovali, da so vsi zakasnitveni elementi enaki, a realnost je zelo daleč od tega, predvsem zaradi neenakomernih zakasnitev v povezavah znotraj čipa in neenakomerne zakasnitve urinega signala do vzorčevalnih flip-flopov. Širina posameznih elementov (‘binov’) oziroma njihov časovni zamik se izmeri s tako imenovanim statističnim testom (angl. *statistical code density test*) [1, 12, 14], pri katerem v TDC pošljamo pulze, ki niso korelirani z uro sistema Red Pitaya. Na sliki 4.2 vidimo testno postavitev, kjer sem kot vir pulzov uporabil mikrokrmilnik AVR, ki generira kvadratni signal. Važno je, da ima generator svojo uro, ki ni v sorodu s tisto na Red Pitayi. V grafičnem vmesniku omogočimo kanale TDC, ki jih je potrebno kalibrirati, in poženemo zajemanje z gumbom ‘RUN’. Ob tem dobimo naključno razporejene zadetke vzdolž linije, pri čemer je večja verjetnost, da zadanemo ‘širše’ zakasnilne elemente, torej tiste z večjo zakasnitvijo. Histogram zadetkov posameznih zakasnitvenih elementov lahko v realnem času spremljamo v oknu grafičnega vmesnika (slika 4.1). Ko zberemo dovolj statistike, kliknemo na gumb ‘Calibrate’ in MATLAB ustvari kalibracijsko tabelo za ta kanal.

Širina W_i posameznega zakasnitvenega elementa i , merjena v pikosekundah, je



Slika 4.2: Primer generiranja sunkov za kalibracijo inštrumenta. Vir ni pomemben, važno je le, da sunki ne izvirajo iz Red Pitaye.

enostavno [12]

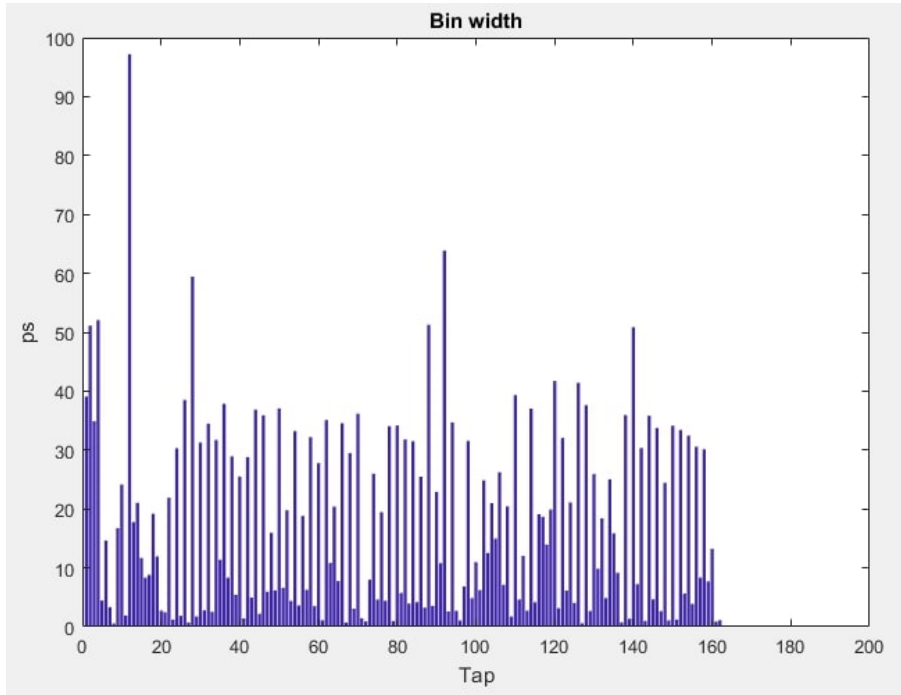
$$W_i = \frac{N_i}{N} \cdot T, \quad (4.1)$$

kjer je N število vseh nabranih sunkov, N_i število sunkov v elementu i in T perioda ure jedra TDC. Pretvorba iz fine meritve i v pikosekunde je potem [12]:

$$T_i = \sum_{k=1}^{i-1} W_k + \frac{W_i}{2}, \quad (4.2)$$

kjer seštejemo širine vseh prejšnjih zakasnitvenih elementov in polovico i -tega, da dobimo čas T_i , ki ga shranimo v kalibracijsko vpogledno tabelo. Tabelo za vsak kanal hrani MATLAB aplikacija. Na slikah 4.3 in 4.4 lahko vidimo rezultat kalibracije kanala TDC0 pri sobni temperaturi.

Iz grafa širine zakasnitvenih elementov je takoj jasno, da je zakasnitev daleč od enakomerne. Nekateri elementi imajo skoraj ničeln zamik, drugi pa se zelo široki; najde se celo en s skoraj 100 ps zamikom (slika 4.3). Vseeno jih je večina nekako pod 40 ps, kar je spodbudno. Opazimo še, da je aktivnih le 162 elementov, kar pomeni, da zgornjih 30 (do 192) signal nikoli ne doseže. 162 elementov torej ustreza polni urini periodi. Grafični vmesnik nam pove tudi, da povprečna širina zakasnitvenega elementa ob tem znaša 17,64 ps, kar nam v grobem že daje namig za časovno ločljivost TDCja. Zanimivo je narisati histogram porazdelitve širine elementov, ki ga



Slika 4.3: Izmerjene širine zakasnitvenih elementov TDC kanala 0, kot jih izriše aplikacija v MATLABu. Na osi x je številka zakasnitvenega elementa, na y pa njegova širina oziroma zakasnitev v pikosekundah.

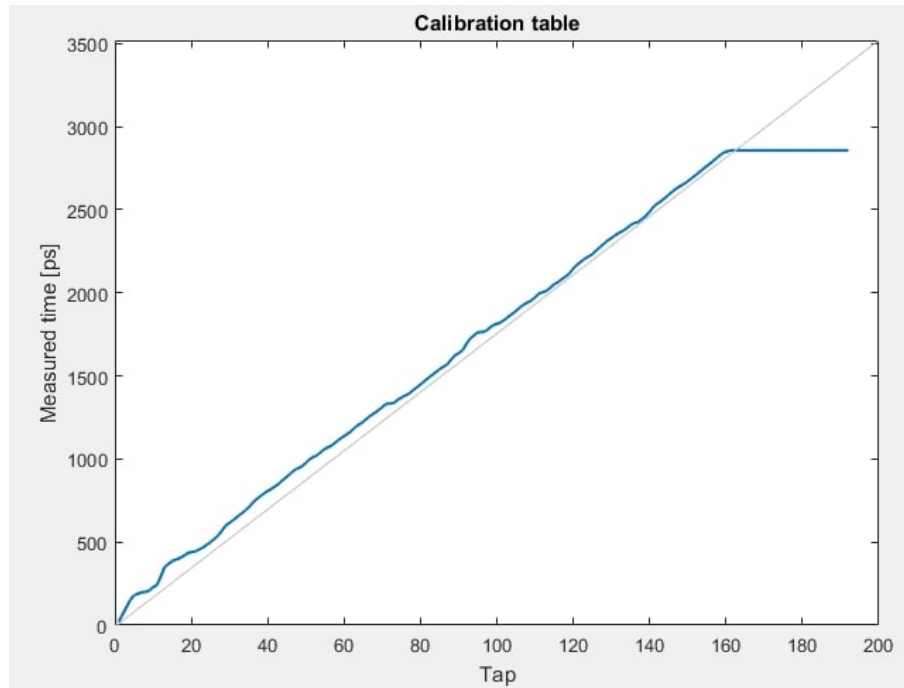
vidimo na sliki 4.5. Nekateri viri, npr. [12], v zmogljivejših čipih poročajo o zelo veliki količini ‘mrtvih’ oziroma ničelnih elementov, kar pri nas očitno ni problem.

Modra krivulja na sliki 4.4 predstavlja izmerjeno kalibracijsko tabelo, ki jo aplikacija potrebuje za fine meritve s časovno-digitalnim pretvornikom. Vsaka časovna značka namreč vsebuje 8-bitno fino vrednost od 1 do 192, ki jo prek te tabele pretvorimo v fizični čas v pikosekundah. Kot že omenjeno, je v tem primeru aktivnih 162 elementov. Če bi bil zamik popolnoma enakomeren in s tem linija idealna, bi na sliki 4.4 dobili ravno črto, ki je ponazorjena s sivo premico. Odstopanje od te premice je nelinearnost časovno-digitalnega pretvornika, ki jo ovrednotimo z diferencialno (DNL) in integralno nelinearnostjo (INL). Diferencialna nelinearnost i -tega zakasnitvenega elementa je definirana kot [14]

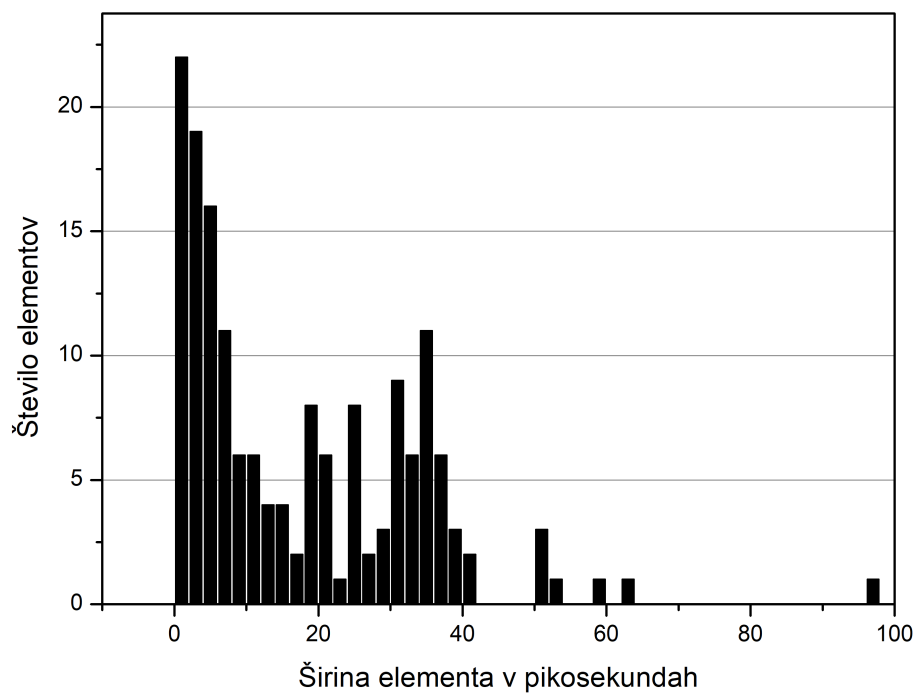
$$DNL_i = \frac{N_i}{\bar{N}} - 1, \quad (4.3)$$

kjer je $\bar{N} = \frac{N}{A}$ povprečno število sunkov na element (N je število vseh sunkov, A pa število aktivnih elementov, v našem primeru 162). Omenimo še, da enačba 4.3 velja le za aktivne elemente. DNL je torej merilo za odstopanje posameznega aktivnega elementa od povprečja. Integralna nelinearnost, po drugi strani, pa je kumulativna vrednost, torej seštevek diferencialnih nelinearnosti [14] in ponazarja odstopanje kalibracije od idealne. Na sliki 4.4 jo prepoznamo kot navpično razdaljo med modro in sivo krivuljo.

$$INL_i = \sum_{k=1}^i DNL_k. \quad (4.4)$$



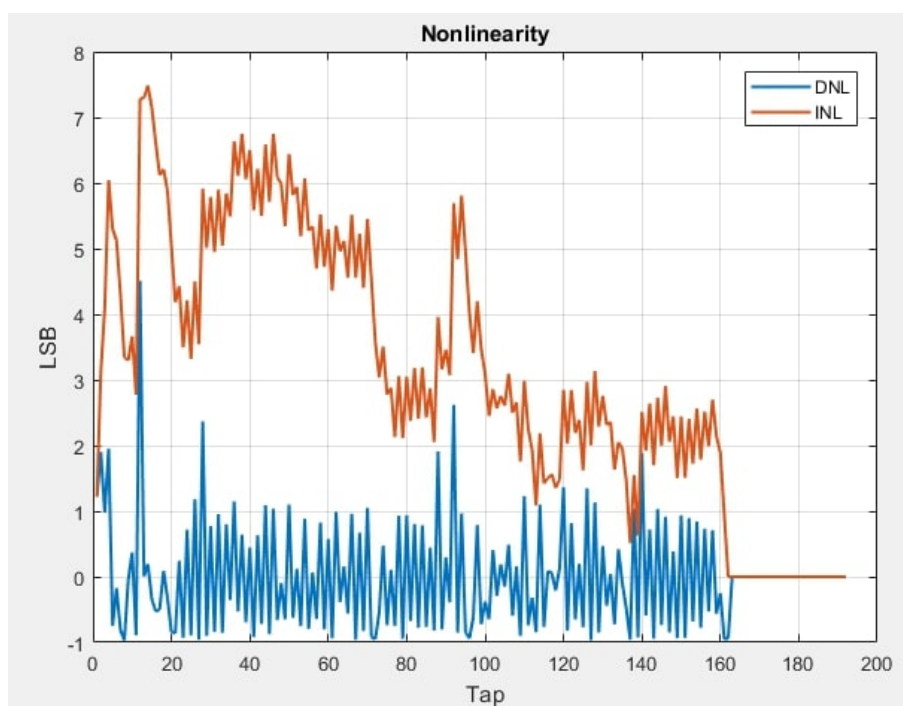
Slika 4.4: Pripadajoča kalibracijska tabela (funkcija) TDC kanala 0, ki jo izračuna MATLAB. Na osi x je izmerjena fina vrednost (številka zakasnitvenega elementa) od 1 do 192, na y pa pripadajoči čas v pikosekundah. Elementi nad 162 niso aktivni, zato je krivulja tam ravna. Siva premica predstavlja idealen primer, če bi bili vsi zakasnitveni elementi enaki.



Slika 4.5: Histogram porazdelitve širine zakasnitvenih elementov TDC kanala 0. Zelo majhnih elementov je sicer največ, a še vedno primerljivo število z ostalimi.

Poglavje 4. Karakterizacija inštrumenta

Slika 4.6 prikazuje DNL in INL za kalibriran TDC kanal 0. Tipično se podaja v enotah enega zakasnitvenega elementa, torej LSB ('Least Significant Bit'), ki ustreza povprečni širini posameznega elementa. Kot vidimo, nelinearnost na začetku linije hitro poskoči, kar je očitno tudi na sliki 4.4. DNL se giblje nekje med -1 in $+1$, medtem ko INL zaradi zelo širokih elementov na začetku linije (slika 4.3) zraste preko $+7$ LSB. Grafe na slikah 4.3, 4.4 in 4.6 v aplikaciji prikažemo s klikom na gumb 'Plot calibration data'. Grafični vmesnik omogoča tudi izvoz kalibracije z gumbom 'Export' v obliki .mat datoteke, ki jo lahko kasneje zopet uvozimo z 'Import', da nam ni treba vsakič znova nabirati sunkov za kalibracijo. Če želimo počistiti histogram in pognati novo zajemanje sunkov, imamo na voljo gumb 'Clear'.

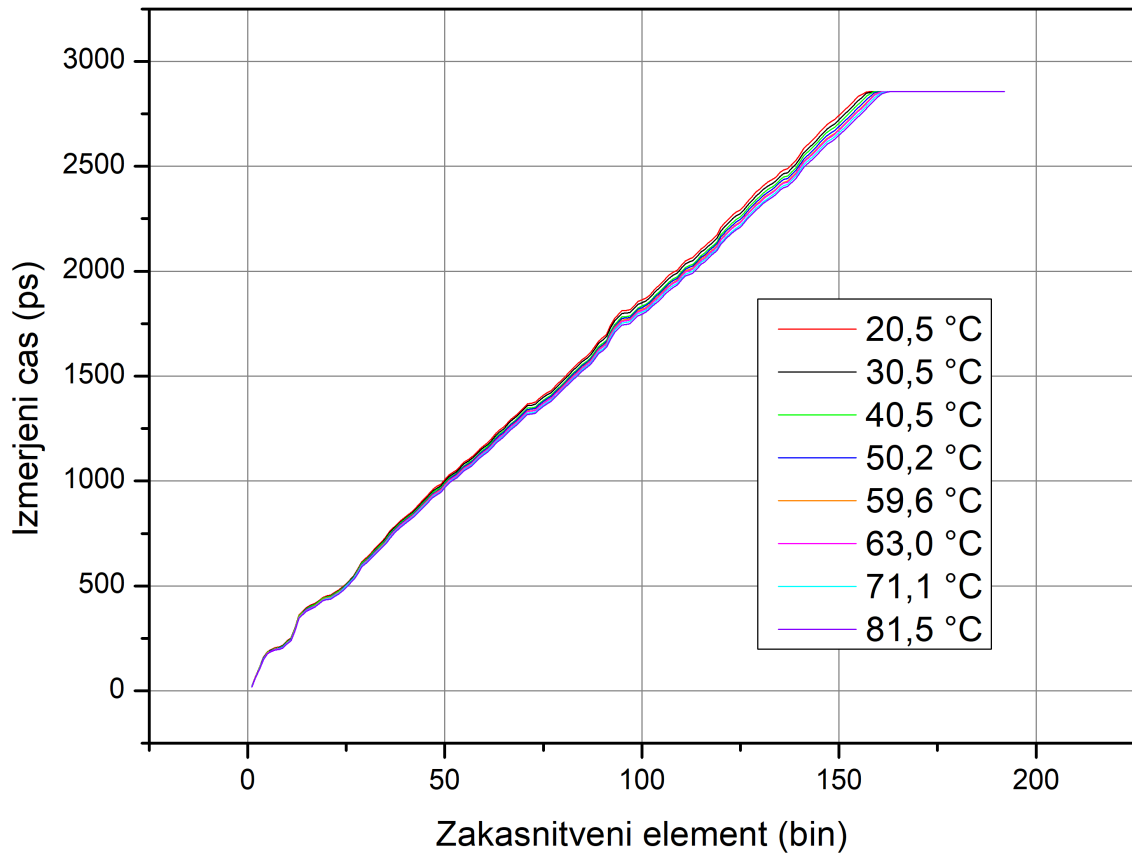


Slika 4.6: Diferencialna (DNL) in integralna nelinearnost (INL) kalibriranega kanala.

Viri [24, 25, 26] poročajo o občutljivosti zakasnilne linije na spremembe temperature in napajalne napetosti, zato je potrebno kalibracijo redno posodablјati, če pride do spremembe zunanjih pogojev. Ravno zato je na desnem robu okna MATLAB vmesnika indikator temperature čipa Zynq. Ob zagonu sistema je treba najprej počakati kakšnih 10 minut, da se temperatura ustali, preden izvajamo kalibracijo. Za občutek, pri sobni temperaturi se čip segreje na obratovalno temperaturo približno 60 °C.

Občutljivost linije na variacije temperature smo se odločili preveriti, tako da smo Red Pitayo zaprli v temperaturno komoro. Slika 4.7 prikazuje kalibracijsko funkcijo kanala TDC0 v odvisnosti od temperature čipa. Vidimo, da so razlike relativno mile, čeprav je na koncu med 20 °C in 80 °C vseeno 100 ps vertikalne razlike, kar ni malo. Opazimo še, da pri višji temperaturi signal prepotuje dlje; pri 20 °C je aktivnih 159 zakasnitvenih elementov, pri 80 °C pa 164. To ima po mojem mnenju dve možni razlagi. Prva je, da se pri višji temperaturi poveča mobilnost nosilcev v polprevodniku, zato so tranzistorji hitrejši in zamik se zmanjša; signal torej znotraj periode ure pride dlje. Druga razlaga pa je, da se nekoliko zmanjša tudi frekvenca

kristala, kar bi podaljšalo periodo ure.



Slika 4.7: Odvisnost kalibracije od temperature čipa za kanal TDC0.

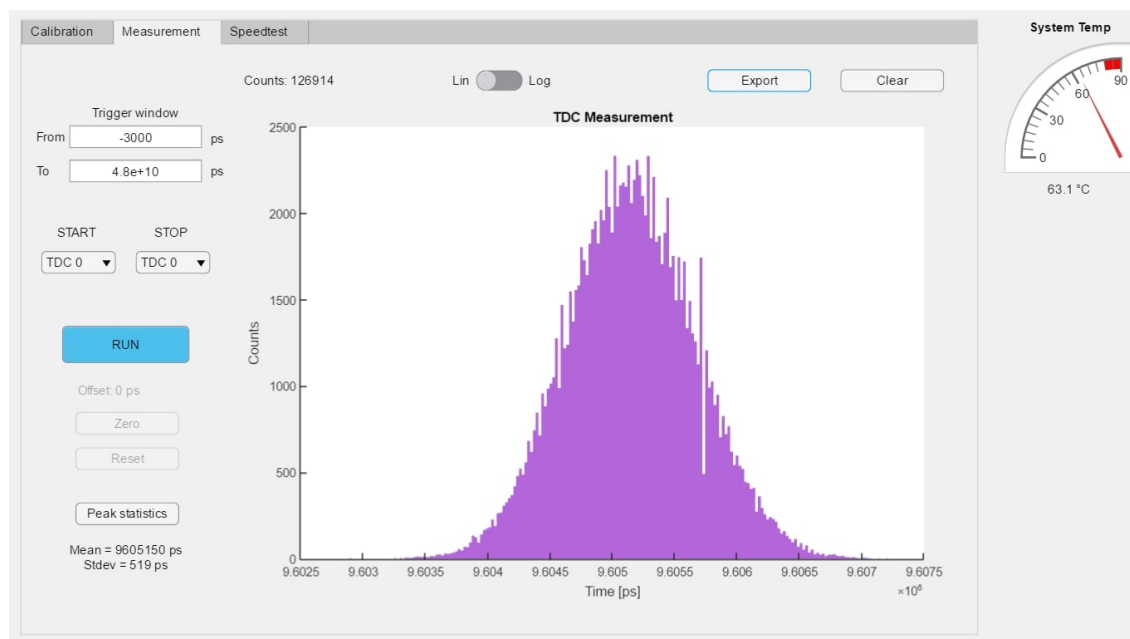
Drugi kanal časovno-digitalnega pretvornika, TDC1, se obnaša zelo podobno, zato ga ne bomo posebej izpostavljali.

4.2 Enokanalna TDC meritev in mrtvi čas

Ko so zakasnilne linije kalibrirane, lahko začnemo opravljati časovne meritve. Odpremo naslednji zavihek 'Measurement', ki je prikazan na sliki 4.8. Najprej moramo na levi izbrati časovno-digitalne pretvornike, s katerimi bomo izvedli meritev – kanala START in STOP. Če sta kanala različna, gre za **medkanalno** meritev, če pa v obeh menijih izberemo isti kanal, tako kot na sliki 4.8, gre za **enokanalno** meritev. V tem poglavju se bomo osredotočili na slednje, torej meritve s samo enim TDC kanalom. V tem primeru MATLAB prenaša značke z izbranega kanala in prikazuje časovne razmake med zaporednimi značkami. Na ta način lahko vidimo vse časovne razmake med pulzi, ki se pojavljalo na vhodu TDCja. Če je signal periodičen, to rezultira v posameznih vrhovih, kot je ta na sliki 4.8.

Med ostalimi funkcijami, ki jih ponuja vmesnik, je izbira območja občutljivosti, kar vnesemo levo zgoraj pod 'Trigger window'. Značke, ki bodo izven tega okna, ne bodo prikazane, kar lahko uporabimo, če želimo posebej izmeriti enega izmed več vrhov. Maksimalno območje sega od -3 nanosekund do $+48$ milisekund, kar je omejeno s strani strojne opreme. Za grobe meritve namreč uporabljamo 24-bitni

Poglavje 4. Karakterizacija inštrumenta



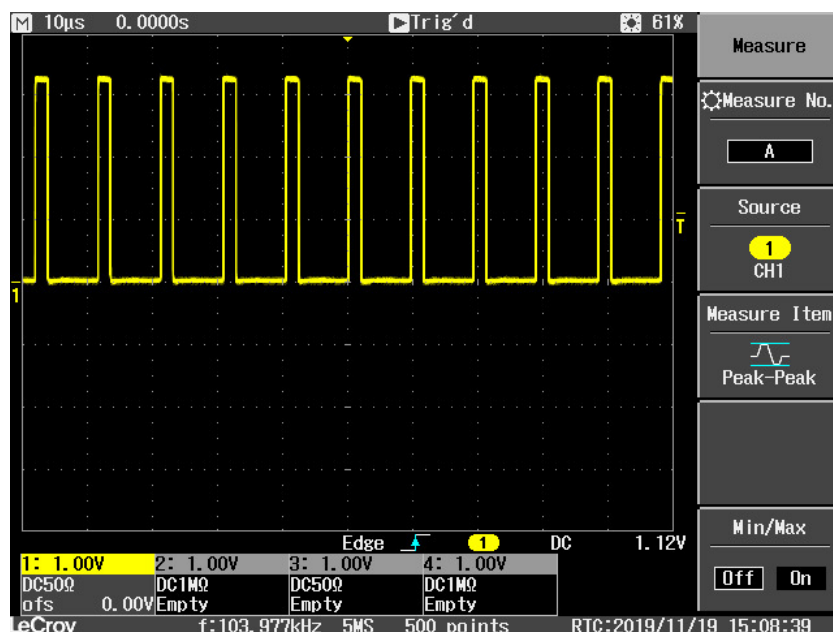
Slika 4.8: Primer enokanalne TDC meritve, s katero izmerimo periodo signala.

števec, ki deluje pri 350 MHz. Takšen števec lahko šteje do $2^{24} - 1 = 16777215$, kar pri frekvenci 350 MHz prešteje v 47,9 ms, potem pa začne od začetka. Značke, ki so oddaljene več kot 48 ms, so torej izven merilnega območja TDCja in bodo izmerjene narobe. Primer: signal s periodo 50 ms bo prezracaljen preko ničle in prikazan kot signal s periodo $50 - 48 = 2$ ms. Nekaj podobnega se zgodi pri analogno-digitalnih pretvornikih, če gremo preko Nyquistove meje.

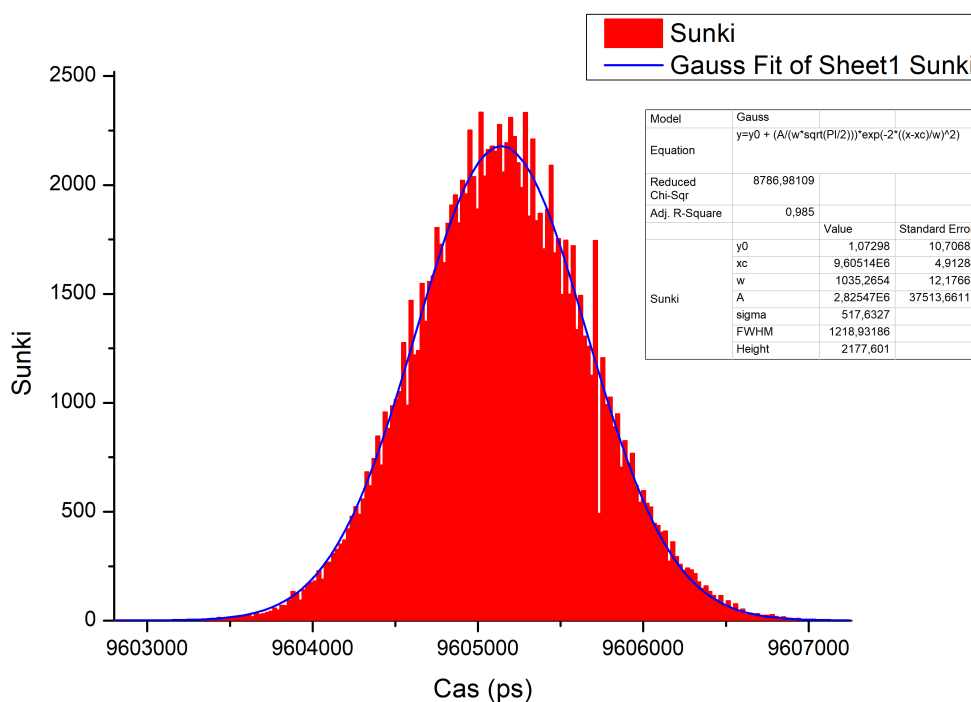
Meritev poženemo z gumbom 'RUN'. Največje število značk, ki jih lahko nabere, je 1024000, potem pa se meritev avtomatsko zaustavi. Nad grafom lahko preklapljam med linearno in logaritemsko skalo, levo spodaj z gumbom 'Peak statistics' pa prikažemo povprečje in raztros izmerjenih značk. Raztros ('Stdev') računa standardni odklon oziroma vrednost RMS, torej koren povprečja kvadratnih odstopanj. Ta gumb je uporaben samo, če imamo en vrh brez ozadja, ker program ne išče in prilagaja vrhov, ampak vzame vse izmerke v oknu. Histogram lahko izvozimo v datoteko `.txt` s klikom na gumb 'Export', 'Clear' pa počisti meritve.

Meritev na sliki 4.8 je bila izvedena s 300 MHz pulznim generatorjem HP8130A, ki je generiral signal na sliki 4.9. Signal je periodičen s frekvenco okoli 104 kHz, torej s periodo približno 9,6 μ s.

Kot lahko vidimo na sliki 4.8, TDC0 izmeri en vrh pri 9,605150 μ s RMS širine 519 ps. Kot bomo videli kasneje, je to krepko nad resolucijo časovno-digitalnega pretvornika, zato lahko raztros pripišemo izključno tresenju oziroma frekvenčni stabilnosti generatorja. Opazimo, da je porazdelitev zelo Gaussovska, kar preverimo s prilagajanjem v Originu (slika 4.10). Program za prilagojeno krivuljo javi povprečje 9,605140 μ s in širino $\sigma = 518$ ps, kar se odlično ujema s tem, kar je prikazal grafični vmesnik TDC. Pri Gaussovih porazdelitvah sta vrednosti RMS in σ enaki.



Slika 4.9: Testni signal frekvence približno 104 kHz, izmerjen s 500 MHz digitalnim osciloskopom LeCroy waveJet 354.



Slika 4.10: Prilagajanje Gaussovske funkcije izmerjenemu histogramu v programu Origin.

Mrtvi čas

V tem načinu merjenja se da ugotoviti izjemno pomemben parameter izdelanega inštrumenta – mrtvi čas. Mrtvi čas oziroma po angleško *dead time* TDCja je čas, ki je potreben, da se sprožena zakasnila linija ponastavi in pripravi na sprejem

novega pulza. Sunki na posameznem kanalu morajo biti razmaknjeni vsaj za mrtvi čas, sicer nekaterih ne bomo zaznali. Določanje mrtvega časa je sila preprosto; s TDCjem merimo signal, ki mu višamo frekvenco, dokler meritve niso več prave.

Z generatorjem HP8130A sem na ta način določil mrtvi čas za oba kanala. TDC0 pravilno izmeri periodo signala vse do 14 ns, ko pa generator nastavimo na 13 ns, se pojavi vrh pri 26 ns, kar je ravno dvakratnik prave periode. To pomeni, da kanal vsako drugo značko izpusti. TDC1 se odreže podobno; 13 ns *večinoma* meri prav, 12 ns pa prikaže že kot 24 ns. Najkrajši varen razmik med značkami za naš časovno-digitalni pretvornik je torej **14 ns**, kar ustreza frekvenci ~ 70 MHz. Maksimalna frekvenca vzorčenja je torej 70 milijonov značk na sekundo (70 MS/s).

Zanimivo je to izraziti s periodami ure jedra TDC. Slednja znaša približno 2,86 ns, kar v 14 ns nanese na slabih 5 urinih ciklov. To deluje povsem smiselno. Zakasnilna linija namreč potrebuje dobro urino periodo za propagacijo pozitivne fronte, potem še kakšno periodo, da se signal vrne daleč nazaj na vhodni flip-flop, nato spet dobro periodo za propagacijo negativne fronte (reset) in še približno periodo, da se signal vrne in sprosti vhodni flip-flop. Vsega skupaj torej dobre 4 cikle, kar se ujema z našo meritvijo mrtvega časa.

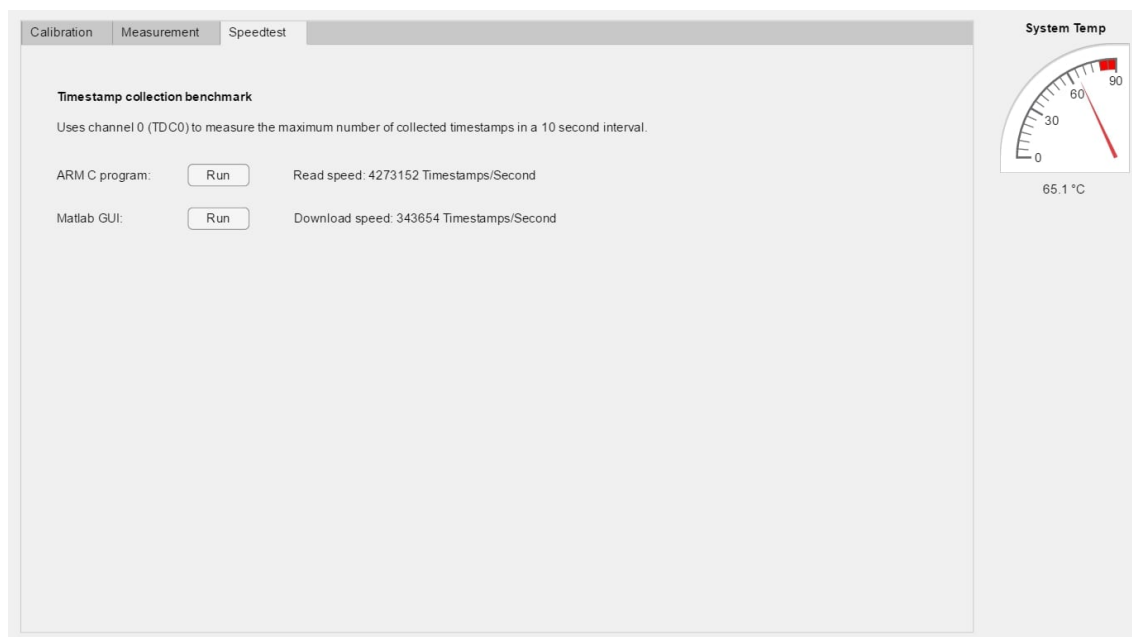
4.3 Hitrost zajemanja časovnih značk

Če TDC na strojnem nivoju lahko zaznava 70 milijonov značk na sekundo, to še zdaleč ne pomeni, da jih lahko dejansko toliko preberemo. Takoj, ko se 2048-mestni BRAM zapolni, je zgodbe konec in kanal se mora ustaviti in počakati, da nekdo te značke pride prebrat. Možni ozki grli sta dve; najprej pri potovanju značk znotraj čipa od BRAMA preko vodil AXI do procesorja ARM, potem pa pri pošiljanju teh značk iz Red Pitaye na osebni računalnik preko Ethernet povezave. Meritev teh dveh hitrosti opravimo v tretjem zavihku aplikacije, imenovanem ‘Speedtest’ (slika 4.11). Ob kliku na gumb ‘Run’ za določitev posamezne hitrosti meritev traja 10 sekund, izpiše pa se povprečno število prenešenih značk na sekundo.

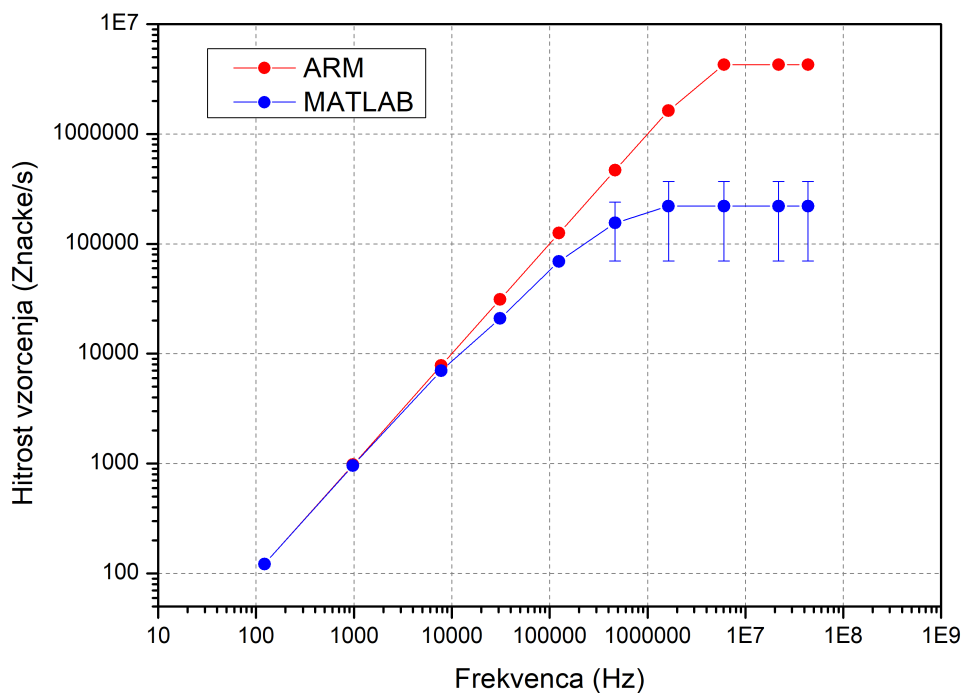
Meritev sem opravil pri različnih frekvencah sunkov na vhodu kanala TDC0, da sem lahko videl, kje se posamezen prehod zasiči. Rezultate prikazuje graf na sliki 4.12. Vidimo, da je do nekaj kHz odziv še linearen, potem pa začne postajati MATLABova hitrost prenosa omejujoča in se naposled zasiči pri reda velikosti 100 kHz (modra krivulja). Izmerki na koncu so precej negotovi, ker je hitrost prenosa odvisna od tega, kaj v tistem trenutku v ozadju počne Windows 10. Po drugi strani je komunikacija na čipu Zynq pričakovano mnogo hitrejša (rdeča krivulja) in se zasiči šele pri 4,27 MS/s.

Dejanska hitrost prenosa značk pri TDC meritvah je seveda pogojena z najožjim grlom, se pravi z modro krivuljo, a zagotovo je tudi ta veliko preveč optimistična. Zavihek ‘Speedtest’ namreč testira izključno samo hitrost prenosa, brez kakršnekoli obdelave značk, prav tako pa le ob aktivnem kanalu TDC0. Ob dodatnem procesiranju v MATLABu in hkrati aktivnih več kanalih so dejanske hitrosti v resnici še bistveno nižje. Ker je za prenos podatkov iz posameznih kanalov zadolžena programska koda, ki se izvaja zaporedno, pričakujemo, da hitrost prenosa na kanal pada obratno s številom aktivnih kanalov; ob dveh delujočih kanalih bi torej pričakovali polovično hitrost in tako naprej.

4.3. Hitrost zajemanja časovnih značk



Slika 4.11: Zavihek za merjenje maksimalne hitrosti prenosa značk med kanalom TDC0 ter C programom na ARMu in med C strežnikom ter MATLAB aplikacijo na drugi strani Ethernet kabla.



Slika 4.12: Rezultati meritev hitrosti prenosa za obe poti časovnih značk. Rdeča krivulja je za prenos znotraj čipa preko vodila AXI, modra pa med obema računalnikoma preko Ethernet kabla.

4.4 Medkanalna TDC meritev & časovna ločljivost

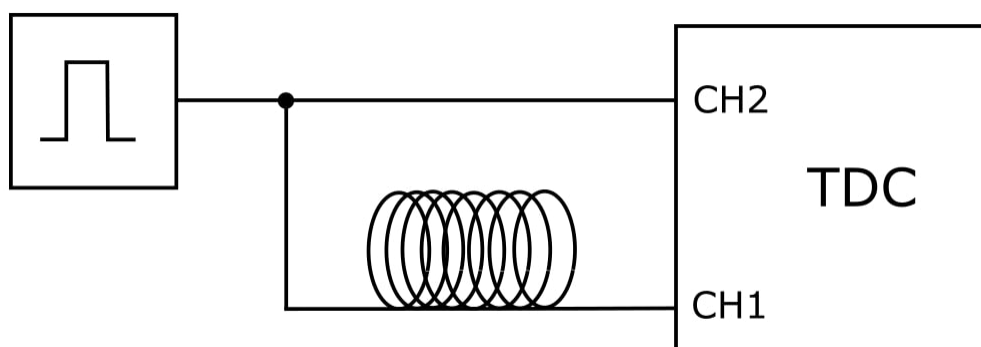
Drug način merjenja časovnih značk se opravlja med dvema kanaloma TDC, tako kot na sliki 1.1. V tem primeru moramo v menijih START in STOP na levi strani zavihka ‘Measurement’ izbrati različna kanala. Meritev deluje tako, da MATLAB računa časovno razliko oziroma časovni interval med značkami na STOP in START kanalu. Z drugimi besedami, STOP minus START. Medkanalne meritve so možne le v parih, ki so na čipu povezani preko *trigger* signala, tako kot na sliki 3.16. Vir ‘trigger’ signala moramo izbrati kot kanal START. V našem primeru smo oba kanala TDC0 in TDC1 na čipu povezali enega za drugim, zato ju lahko poljubno obračamo naokoli v MATLAB aplikaciji. Signal *trigger* namreč potrebujemo zato, da lahko vzročno-posledično povežemo značke na različnih kanalih. Brez njega izgubimo informacijo, katere značke STOP pripadajo kateri znački START.

Mogoče je samoumevno, ampak pulzi STOP morajo načeloma priti za pulzom START. Izjemoma lahko končni pulz prehití začetni pulz za največ en urin cikla, torej 2,86 ns. Zaradi tega gre ‘Trigger window’ v grafičnem vmesniku od -3 ns, ne pa od 0 (slika 4.8). Če pulz STOP prehití START za več kot urin cikla, bo dodeljen prejšnji START znački.

Poglejmo si postavitev na sliki 4.13. Signal pulznega generatorja razdelimo na dva kabla in enega peljemo na kanal START (na sliki CH2), drugega pa nekoliko zakasnimo in peljemo na kanal STOP (na sliki CH1). Izmerjen časovni interval med prihodom signala je torej razlika med obema značkama:

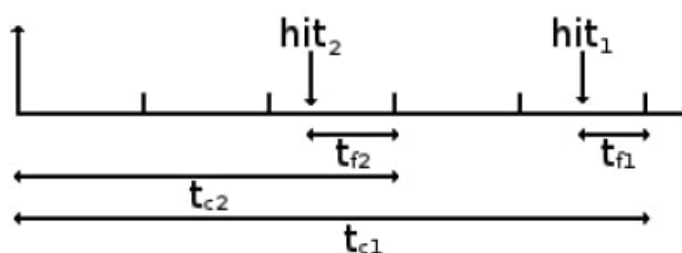
$$\Delta t = t_1 - t_2 = (t_{c1} - t_{f1}) - (t_{c2} - t_{f2}) , \quad (4.5)$$

kjer je posamezen t_c groba meritev, t_f pa fina. Enačbo 4.5 si lahko vizualiziramo s pomočjo slike 4.14, kjer se lepo vidi princip časovne interpolacije TDCja. Grobi čas je enostavno vrednost števca, pomnožena s periodo ure, medtem ko je fini čas odčitán iz kalibracijske tabele glede na izmerjeno kodo zakasnilne linije. Paziti je treba na minus znak pred t_f , ker meri kako dolgo *pred* urino fronto je prišel pulz – glej sliko 4.14.



Slika 4.13: Testna postavitev za določanje ločljivosti časovno-digitalnega pretvornika.

S postavitvijo na sliki 4.13 izmerimo najbolj pomemben parameter časovno-digitalnega pretvornika; njegovo časovno ločljivost. Ideja je v tem, da oba kanala



Slika 4.14: Časovna interpolacija je kombinacija grobe in fine meritve časa. Črtice na časovni osi so urini cikli.

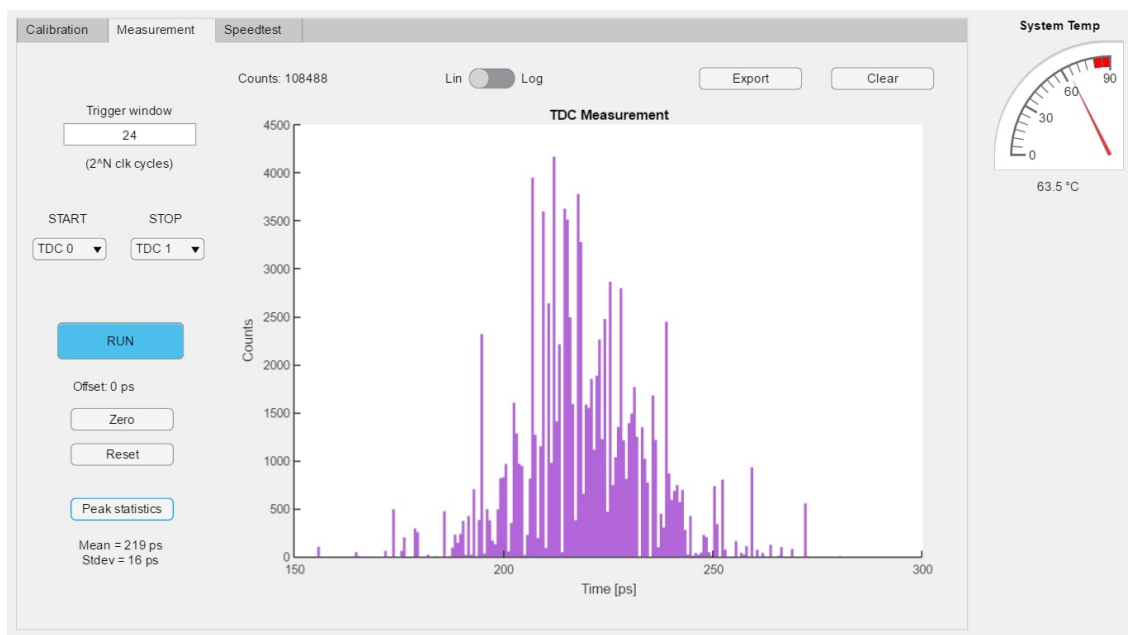
merita *isti* pulz, zato ne vidita tresenja oziroma neidealnosti generatorja. Po mnogih pulzih bomo dobili porazdelitev zakasnitev, katere raztros je posledica izključno neidealnosti merilnega inštrumenta, torej samega TDCja.

Slika 4.15 prikazuje takšno postavitve z Red Pitayo in generatorjem HP8130A, ki generira isti 104 kHz signal kot prej. Ker uporabljamo dolge kable, morajo biti na koncu linije obvezno prisotni 50 Ω zaključitveni upori, sicer merimo samo odboje.



Slika 4.15: Določanje ločljivosti izdelanega TDCja na Red Pitayu z generatorjem HP8130A, katerega izhod razdelimo na dva kabla.

Na sliki 4.16 vidimo rezultat meritve pri sobni temperaturi (T čipa $\approx 63,5$ $^{\circ}\text{C}$) po dobrih 100 tisoč sunkih. Raztros RMS je pičlih 16 ps, porazdelitev pa niti približno ni več Gaussova tako kot na sliki 4.8. Vzrok je neidealnost časovno-digitalnih



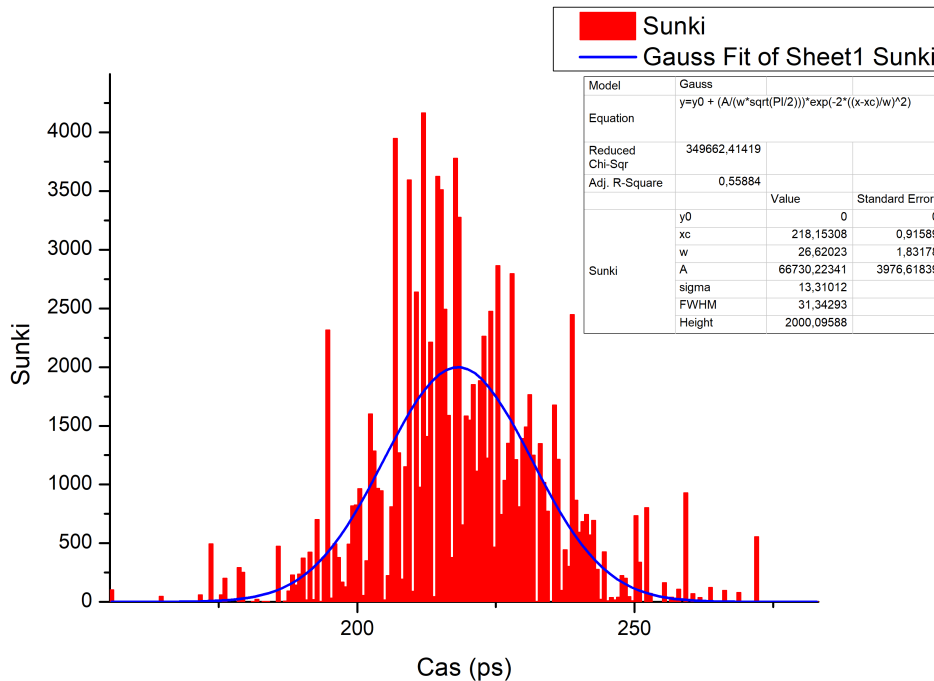
Slika 4.16: Zavihek ‘Measurement’ v medkanalnem režimu, kjer določamo resolucijo TDCja. Opomba: na sliki je nekoliko starejša različica vmesnika, zato ‘Trigger window’ izgleda drugače.

pretvornikov oziroma njunih zakasnilnih linij. Vsa tista nelinearnost, vsi tisti ultra široki zakasnitveni elementi s slike 4.3 zdaj udarijo ven in popačijo meritev. Smo na sami meji zmogljivosti TDCja.

Vseeno lahko v Originu izmerjeni porazdelitvi prilagodimo Gaussovo funkcijo, kar vidimo na sliki 4.17. Kot pričakovano je ujemanje zelo slabo, zanimivo pa je primerjati izračunane parametre. MATLAB kot povprečje oziroma izmerjeni čas vrne 219 ps, Gauss pa 218 ps, kar je praktično isto. Drugače je pri raztrosu; aplikacija javi RMS 16 ps, medtem ko je širina Gaussove funkcije le $\sigma = 13 \pm 1$ ps. Tipično bomo raztros vedno podajali kot vrednost RMS, ki velja splošno, σ pa naj bo rezervirana za Gaussove porazdelitve.

Izmerjenih 16 pikosekund je *medkanalna* resolucija, h kateri prispevata tako TDC0 kot TDC1. Če predpostavimo, da sta kanala enaka, lahko torej RMS ločljivost posameznega kanala ocenimo na $16/\sqrt{2} \approx 11$ ps. To je nadvse impresivna številka, ki je krepko preseгла naša pričakovanja pred začetkom izdelave TDCja, hkrati pa več kot izpolnjuje zastavljeni cilj resolucije pod 50 ps.

Če se na hitro vrnemo na sliko 4.16, vidimo, da imamo levo z gumbom ‘Zero’ možnost od izmerkov odšteti njihovo povprečje oziroma ‘offset’, ki je posledica merilne postavitve. Tako lahko po svoje definiramo časovno razliko nič in merimo od tam dalje, kar pogosto pride prav. Če želimo izmerke kadarkoli povrniti nazaj na njihove originalne vrednosti, kliknemo ‘Reset’. Ta del grafičnega vmesnika je na voljo le pri medkanalnih meritvah, pri enokanalnih (slika 4.8) pa odštevanje povprečja ni fizikalno in so zato omenjena polja onemogočena.



Slika 4.17: Prilagajanje Gaussovske funkcije izmerjeni porazdelitvi v programu Origin.

4.4.1 Resolucija v odvisnosti od zakasnitve

Ravnokar izmerjena ločljivost je bila dosežena pri časovnem intervalu ~ 200 ps, ki pade znotraj istega urinega cikla in je torej izmerjen izključno z uporabo zakasnilne linije, se pravi brez binarnega števca. Sklepamo, da se pri večjih intervalih, ko vmes preteče več ciklov ure, resolucija pokvari zaradi tresenja ure, kar bomo preverili v tem razdelku. Postavitev ostane ista kot na slikah 4.13 oziroma 4.15, s tem da na linijo STOP dodamo nastavljivo zakasnitev. Za zakasnitve do 64 ns uporabimo ‘generator zamika’ EG&G-ESN DL6001, za daljše intervale pa kolut kabla – oboje lahko vidimo na sliki 4.18. Škatla DL6001 ni nič drugega kot skupek različno dolgih kablov, ki so med sabo povezani s stikali.

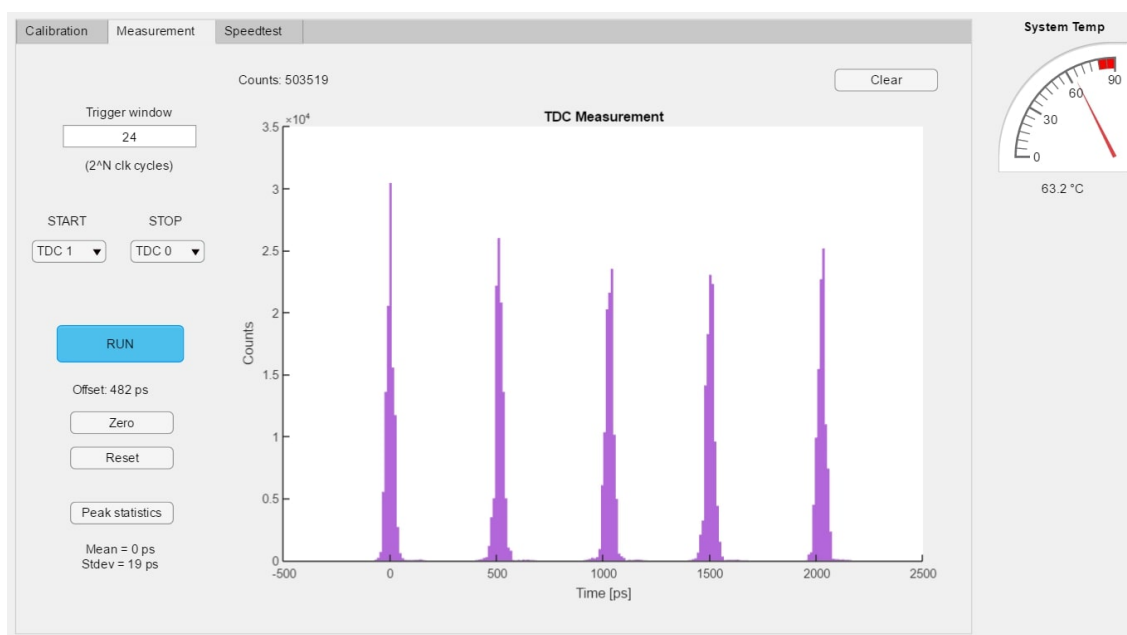
Z nastavljivo zakasnitvijo lahko hitro preverimo, če medkanalne meritve delujejo tako, kot bi morale. Na sliki 4.19 je nekaj vrhov, ki smo jih ustvarili z diskretnim spreminjanjem zakasnitve od 0 do 2 ns v korakih po 0,5 ns. S škatlo DL6001 gremo lahko do 64 ns, z dodatnimi koluti kabla pa še višje. Graf na sliki 4.20 potrjuje pravilno delovanje TDCja skozi celotno območje merjenih zakasnitev.

Najbolj nas zanima, kakšna je ločljivost RMS meritev na sliki 4.20, odgovor pa nam ponuja graf na sliki 4.21. Vpliv dodatnih urin ciklov je presenetljivo majhen, saj do zakasnitve 10 ns medkanalna ločljivost ostane pod 20 ps, do 50 ns pa še vedno pod 25 ps, kar je zelo primerljivo z začetno resolucijo 17 ps. Situacija se drastično poslabša šele, ko dodamo kolute kabla, ki je precej tanjši od tistega v škatli DL6001. Posledično je pri 256 ns atenuacija že tako huda, da lahko komaj še presežemo prag za logično enico na vhodih FPGA, signal pa zelo počasi narašča in nima definiranega roba. Glavni krivec za slabšo ločljivost je torej šum, ne pa ura TDCja. Z modro označenih meritev na sliki 4.21 zato ne gre jemati preveč zares, ker bi jih bilo treba ponoviti z debelejšim kablom oziroma z nečim, ki ima manjšo atenuacijo signala.

Poglavje 4. Karakterizacija inštrumenta

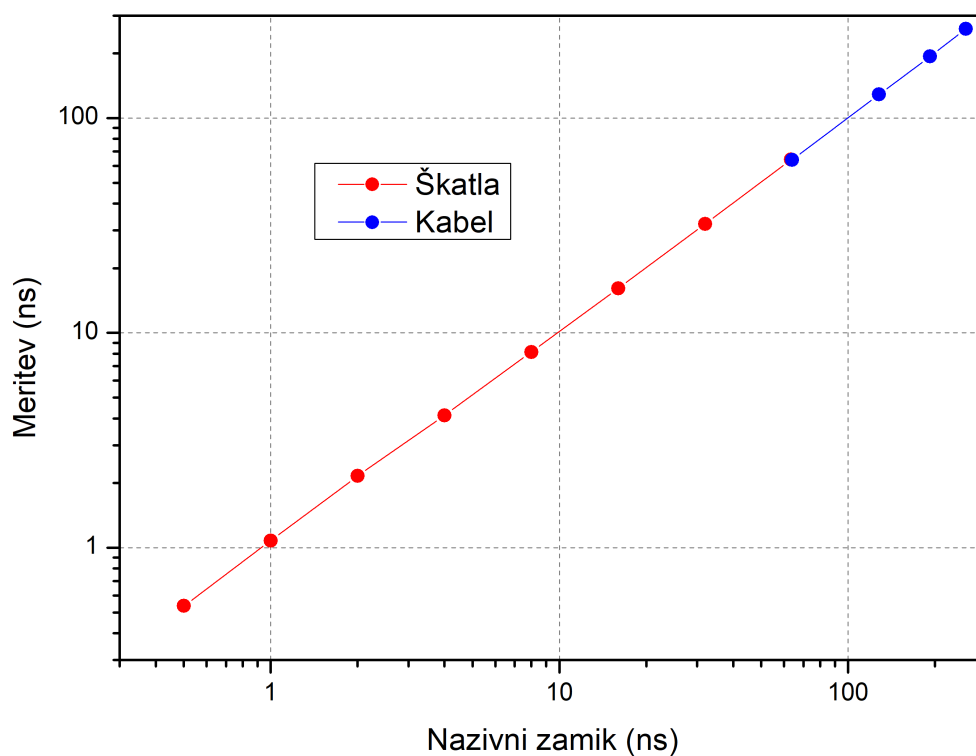


Slika 4.18: Dodatno zakasnitev ustvarimo zelo enostavno: v linijo dodamo dolge kable.

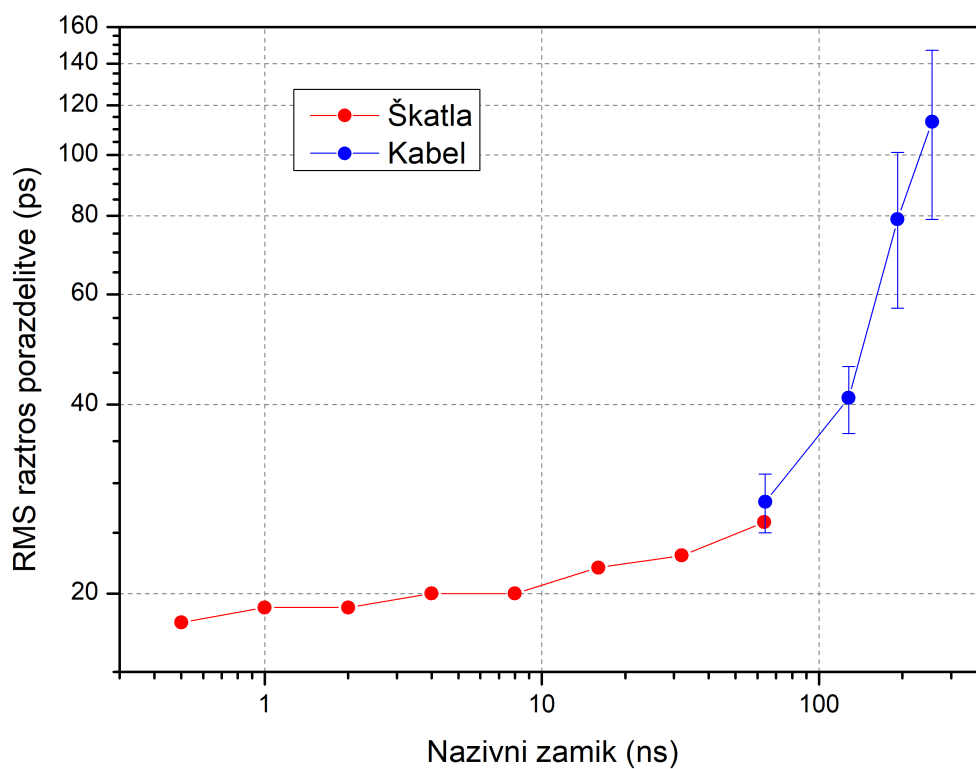


Slika 4.19: Rezultat meritve, med katero smo diskretno spreminjali zamik med kanaloma od 0 do 2 ns. Še pred tem smo odšteli offset, ki pri ničelni nazivni zakasnitvi znaša 482 ps.

4.4. Medkanalna TDC meritev & časovna ločljivost



Slika 4.20: Meritve časovnega intervala od 0,5 do 63,5 ns. Rdeče zamike smo ustvarili s škatlo, modre pa z dodatnimi koluti kabla.

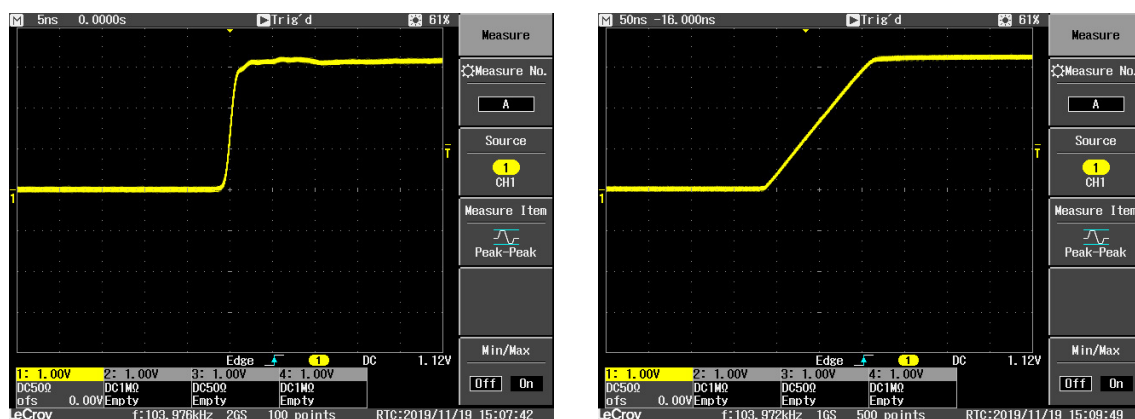


Slika 4.21: Izmerjena odvisnost RMS ločljivosti meritev v odvisnosti od časovnega intervala. Modri izmerki po moji oceni niso več kredibilni, predvsem tisti nad 100 ns.

O podobni neobčutljivosti resolucije časovno-digitalnih pretvornikov na intervale do nekaj 10 ns poročajo tudi v delih [12, 13].

4.4.2 Resolucija v odvisnosti od strmine pulza

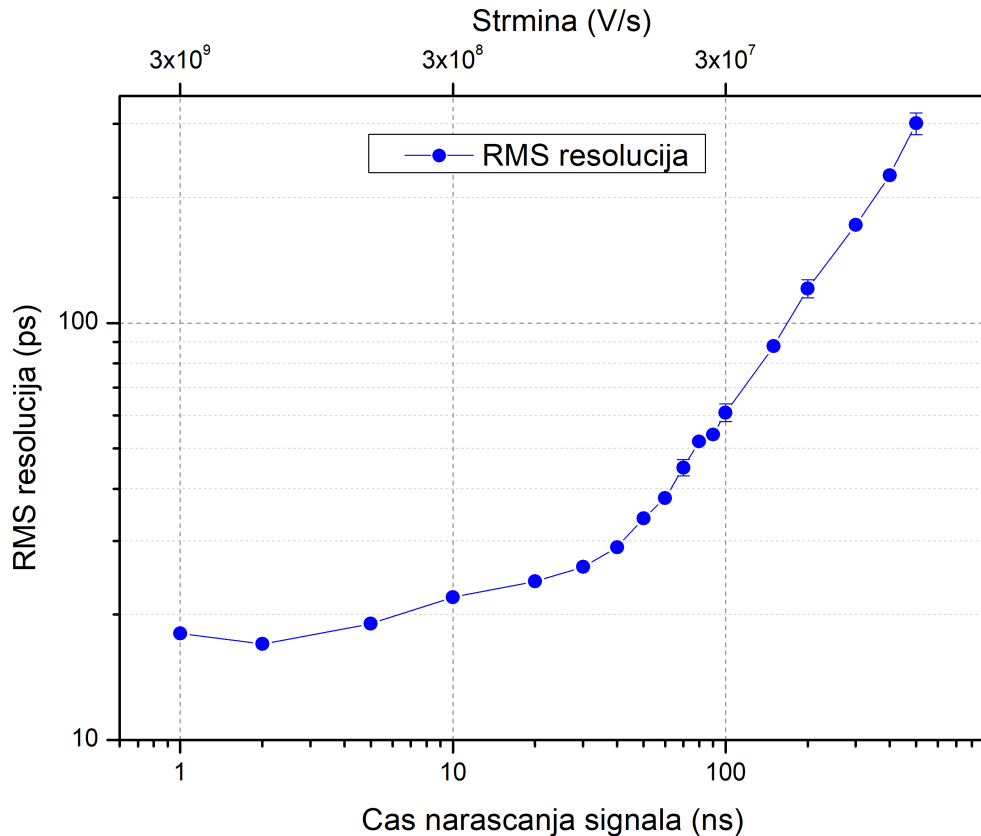
Za zelo natančne časovne meritve je pomembno, da je ‘dogodek’ dobro definiran. TDC je občutljiv na prehod iz logične ničle na enico, torej morajo imeti signali čim bolj oster rob oziroma čim krajši čas naraščanja (angl. *rise time*). Meritve ločljivosti v prejšnjih razdelkih so bile opravljene s signali s časom naraščanja ~ 1 ns, zanima pa nas, kako se to spremeni pri bolj položnih, počasnejših signalih. Na srečo uporabljeni generator HP8130A omogoča natančno nastavljanje tega parametra od 1 ns navzgor. Na sliki 4.22 lahko vidimo fronto istega signala, enkrat z zeli hitrim naraščanjem znotraj 1 ns in drugič zelo počasi v 100 ns. Časovne meritve s signali, ki počasneje naraščajo, so bolj občutljive na šum, zato z večanjem časa naraščanja pričakujemo slabšanje resolucije.



Slika 4.22: Hitra fronta s časom naraščanja ~ 1 ns (levo) in počasna fronta s časom naraščanja 100 ns (desno). Posneto z digitalnim osciloskopom LeCroy waveJet 354.

Meritve so bile opravljene z isto postavitvijo kot na slikah 4.13 oziroma 4.15, s tem da sem tokrat na generatorju spreminjal rampo signala od 1 ns do 500 ns. Izmerjeno odvisnost RMS ločljivosti prikazuje graf na sliki 4.23.

Do nekaj 10 ns je resolucija zelo visoka, potem pa se začne naglo slabšati. To je precej širši interval, kot smo pričakovali, saj se digitalni signali z 10 ns časom naraščanja smatrajo že kot dokaj počasne oziroma standardne. Kljub temu pri 10 ns še vedno dosegamo ločljivost 22 ps napram začetnim 17 ps. To pomeni, da lahko tudi s povprečno hitro logiko opravljamo zelo natančne časovne meritve.



Slika 4.23: Izmerjena odvisnost medkanalne RMS ločljivosti od časa naraščanja vhodnega signala.

4.4.3 Resolucija v odvisnosti od temperature

Še zadnja točka pri karakterizaciji medkanalnih meritev je odvisnost izmerjenega časovnega intervala in ločljivosti od temperature TDCja. Pri kalibraciji zakasnilne linije smo omenili in tudi izmerili, da je le-ta občutljiva na temperaturne spremembe, zato moramo preveriti, kako to vpliva na dejanske meritve. Za to sem uporabil temperaturno komoro kitajskega proizvajalca Shanghai Jianheng (na sliki 4.24), kamor sem zaprl svojo Red Pitayo (slika 4.25). Preostanek postavitve je bil enak kot doslej, torej po vzoru sheme na sliki 4.13, z vsemi ostalimi komponentami razen TDCja zunaj na sobni temperaturi.

Meritve sem opravljal pri temperaturi komore od -5 °C do $+60\text{ °C}$, kar ustreza temperaturi čipa od 20 °C do 85 °C . Odvisnost temperature silicija od temperature okolice je pričakovano premica, kar lahko vidimo na grafu na sliki 4.26. V temperaturni komori je čip le 25 stopinj toplejši od zraka, ker je vzpostavljena cirkulacija z ventilatorjem. Če zrak miruje, se Red Pitaya bolj segreje, na sobni temperaturi nad 60 °C .

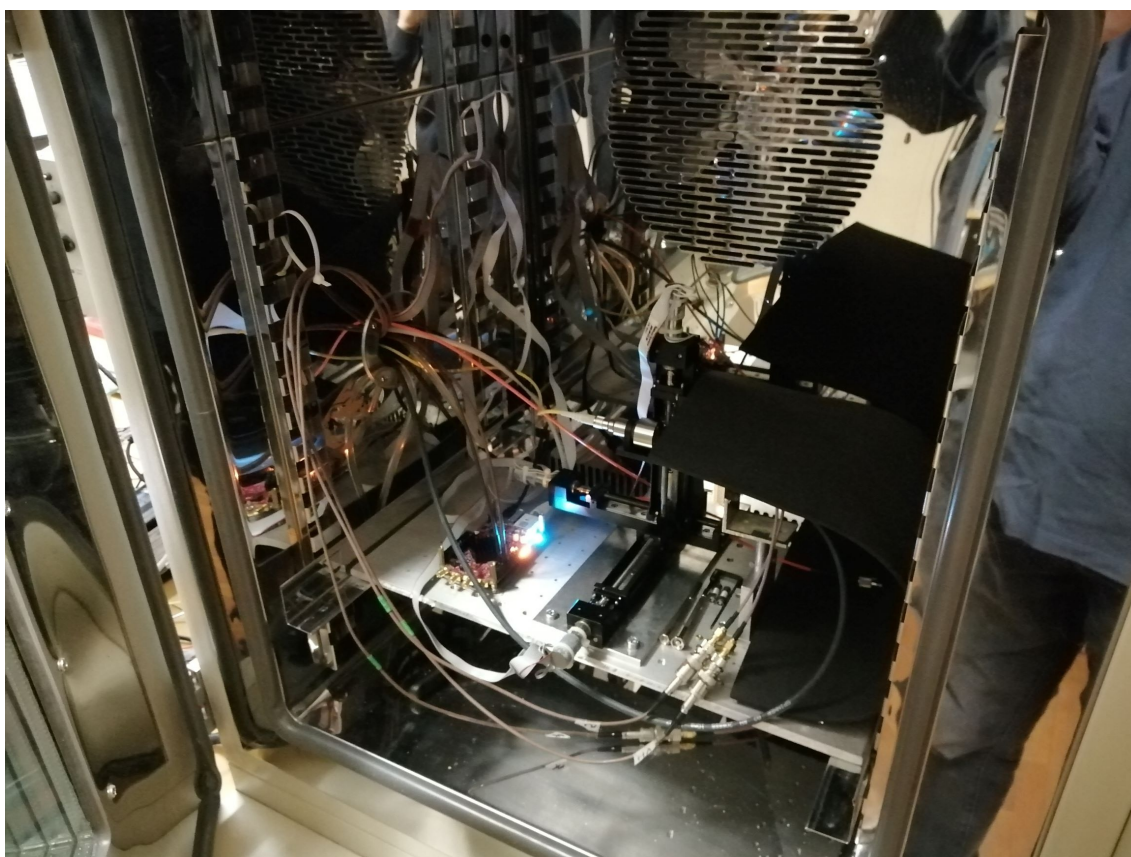
Meritve odvisnosti lastnosti TDCja od temperature sem opravljal na dva načina. Prvič sem kalibracijo zakasnilne linije opravil samo enkrat, in sicer pri sobni temperaturi (T čipa $\approx 63\text{ °C}$), potem pa vse meritve izvajal s to, **statično** kalibracijo. Drugič sem zakasnilno linijo kalibriral **dinamično**, torej pri vsaki spremembi temperature. Rezultate meritev časovnega intervala in RMS resolucije vidimo na slikah 4.27 in 4.28.



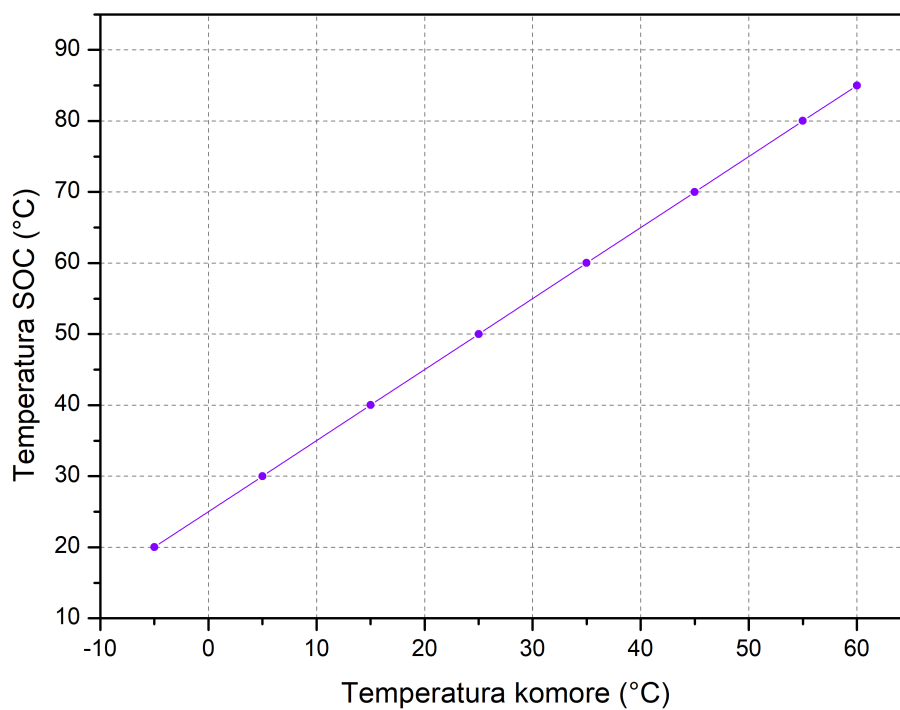
Slika 4.24: Za meritve v odvisnosti od temperature sem se prestavil k temperaturni komori Shanghai Jianheng Instruments.

Opazimo, da izmerjen interval s temperaturo pada, neodvisno od kalibracije, in sicer za cca. 10 ps v 60 °C intervalu. Verjetno zato, ker se s temperaturo spreminja zakasnitev poti, ki jo mora signal prepotovati do TDCja, vključno z vhodno/izhodnimi elementi na čipu FPGA. Drugače je pri odvisnosti RMS resolucije: če ne popravljamo kalibracije, ta počasi drsi, pri dinamični kalibraciji pa jo lahko držimo na konstantnih 15–16 ps skozi celotno temperaturno območje. Te ugotovitve se vsaj kvalitativno lepo ujemajo z rezultati v [12] in [25].

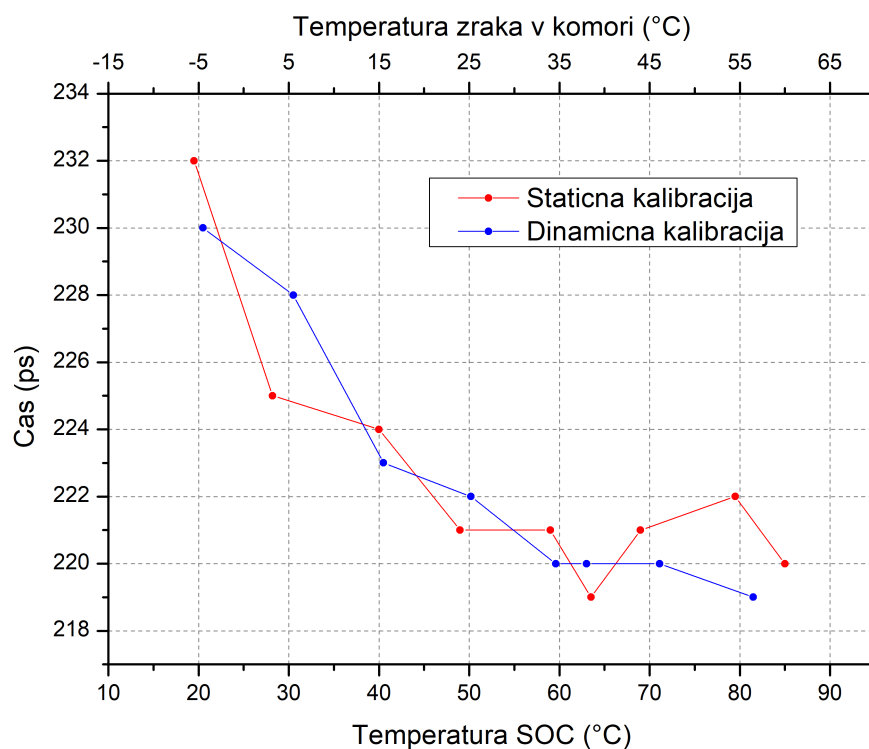
Če želimo torej ohranjati najvišjo ločljivost časovnih meritev, je priporočljivo, da se zakasnilna linija redno kalibrira ob spremembah temperature, ki jo lahko spremljamo na desni strani grafičnega vmesnika.



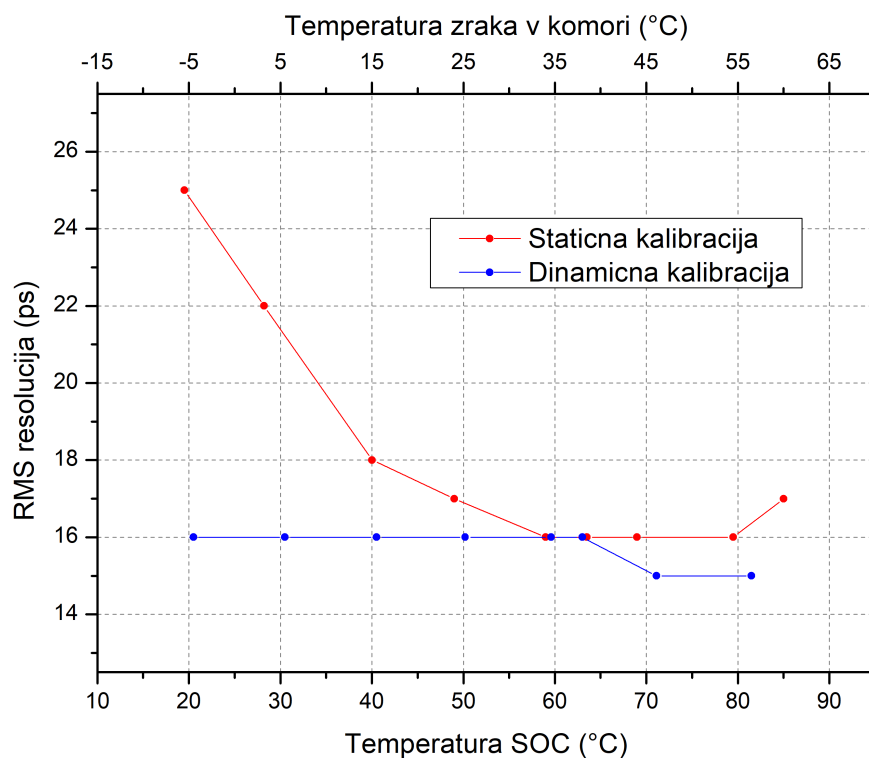
Slika 4.25: Red Pitaya v temperaturni komori.



Slika 4.26: Odvisnost temperature čipa od temperature zraka v komori. Med delovanjem je Zynq 25 °C toplejši od okolice.



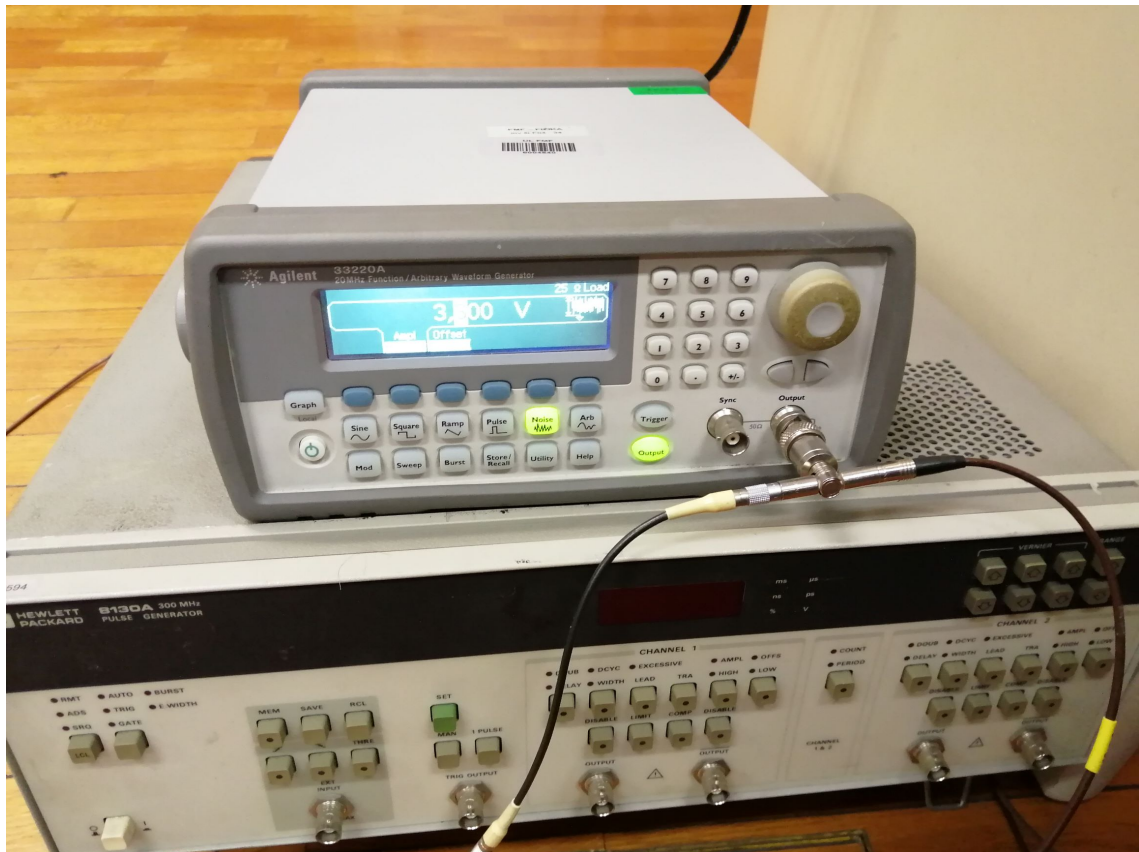
Slika 4.27: Odvisnost izmerjenega časovnega intervala od temperature TDCja pri statični in dinamični kalibraciji. Napaka posameznih meritev znaša približno 20 ps, kar je več od prikazane y skale, zato so negotovosti zaradi preglednosti izpuščene.



Slika 4.28: Odvisnost RMS časovne ločljivosti od temperature TDCja pri statični in dinamični kalibraciji.

4.5 Absolutna natančnost

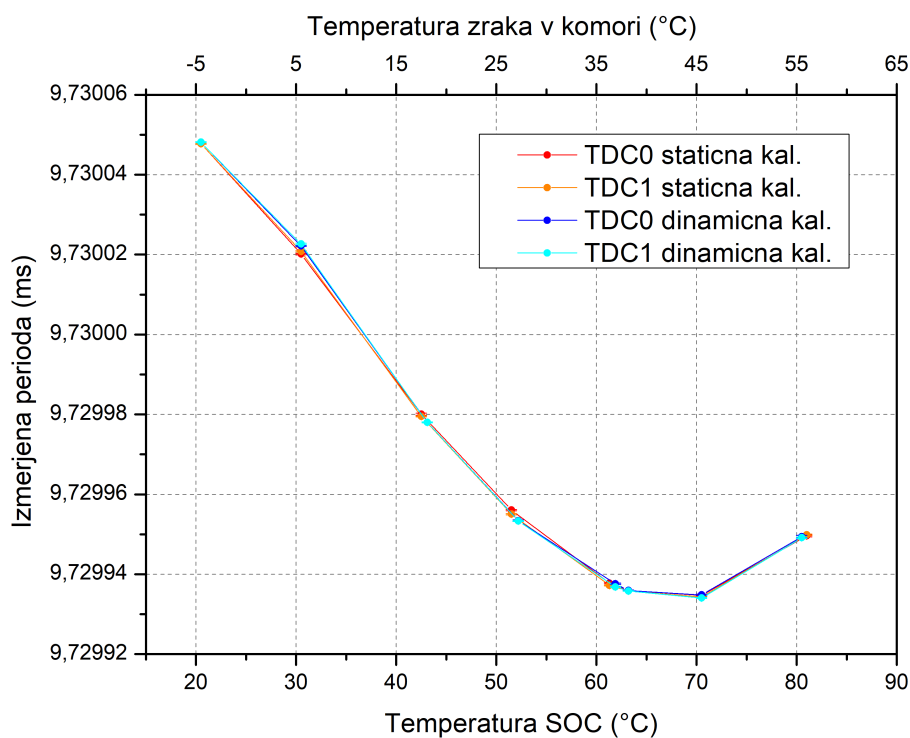
Vrnimo se nazaj na enokanalne TDC meritve, se pravi na merjenje periode vhodnega signala. Doslej še ni bilo govora o absolutni natančnosti časovnih meritev oziroma kako dobro TDC izmeri neko točno določeno periodo. Za ta namen sem si od profesorja Dušana Ponikvarja izposodil funkcijski generator Agilent 33220A – vidimo ga na sliki 4.29 – ki ga za razliko od prej uporabljenega HP8130A odlikuje izjemna natančnost pri generiranju periodičnih signalov. Kot navaja podatkovni list [27], lahko proizvede izbrano frekvenco z natančnostjo ± 20 ppm.



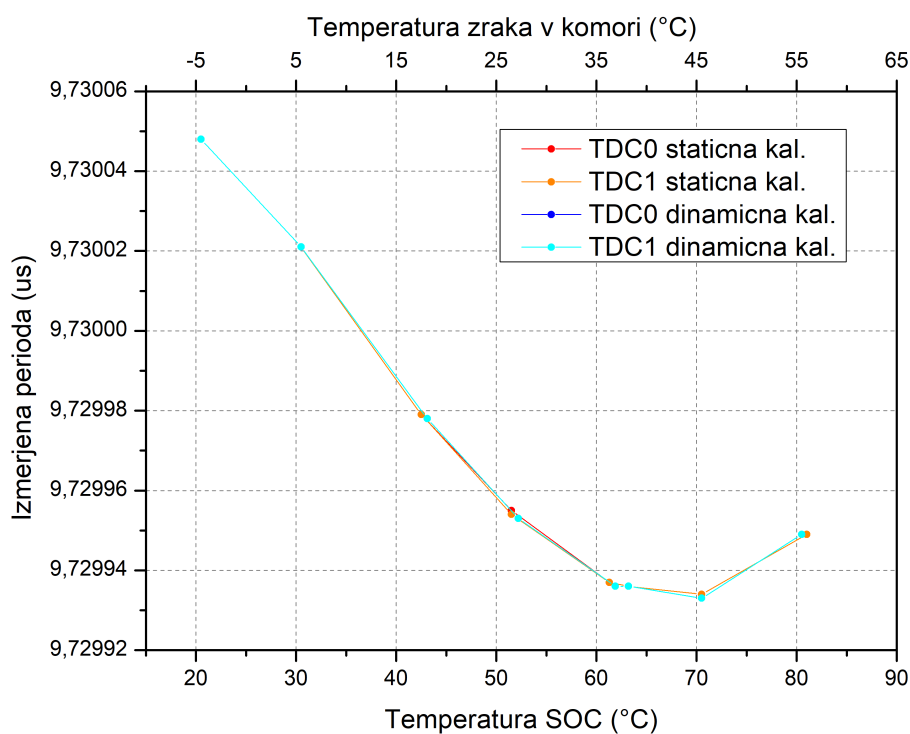
Slika 4.29: Funkcijski generator Agilent 33220A. Uporabljal sem ga za teste, ki so zahtevali zelo visoko frekvenčno stabilnost.

Spet sem opravljal meritve v temperaturni komori, in sicer za oba TDC kanala posebej v treh merilnih območjih; milisekundnem, mikrosekundnem in nanosekundnem. Zanimalo me je, kako dobro kanala merita nastavljeno periodo 9,730 ms, 9,730 μ s ter 272,27 ns v temperaturnem razponu 60 °C. Frekvence so nalašč izbrane tako nepravilno, da se značke ne bi po pomoti ujele s kakšno uro TDC kanalov, čeprav se je kasneje izkazalo, da je bil strah odveč. Zopet sem eno serijo izmeril s fiksno kalibracijo zakasnilne linije, drugič pa z dinamično. Rezultate meritev vidimo na slikah 4.30, 4.31 in 4.32.

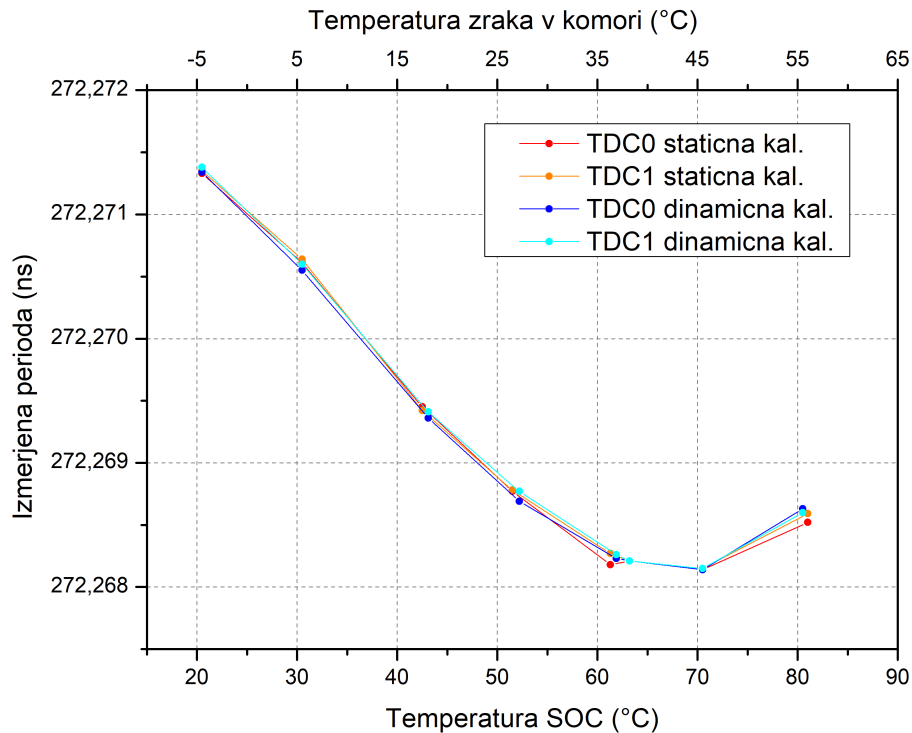
Takoj opazimo tri stvari. Prvič, sprotno kalibriranje linije nima nikakršnega vpliva na meritve. Drugič, krivulji obeh kanalov (TDC0 in TDC1) ležita praktično popolnoma ena na drugi, kar pomeni, da se oba kanala obnašata zelo podobno. Kar je vsekakor spodbudno. Daleč najbolj pomembna opazka pa je to, da vsi trije grafi



Slika 4.30: 9,73 ms v odvisnosti od temperature čipa za oba kanala. Napaka posameznih izmerkov znaša do ~ 200 ps, kar je na tej skali neopazno.



Slika 4.31: 9,73 μ s v odvisnosti od temperature čipa za oba kanala. Napaka posameznih izmerkov znaša do ~ 200 ps, kar je več od prikazane y skale, zato so negotovosti zaradi preglednosti izpuščene.



Slika 4.32: 272,27 ns v odvisnosti od temperature čipa za oba kanala. Napaka posameznih izmerkov znaša do ~ 200 ps, kar je več od prikazane y skale, zato so negotovosti zaradi preglednosti izpuščene.

izgledajo skoraj identično, samo da je enkrat časovna skala v ms, drugič v μs in tretjič v ns. To pomeni, da je relativna napaka oziroma lezenje TDCja povsod enako, ne glede na območje. Relativno odstopanje TDCja od nastavljene periode pri vseh treh meritvah skozi celotno temperaturno območje znaša manj kot 10^{-5} oziroma 10 ppm. Iz tega sledi, da imajo s stališča natančnosti frekvenčnega generatorja, ki znaša 20 ppm, izmerki praktično nemerljivo sistematsko napako. Z drugimi besedami, izdelani časovno-digitalni pretvornik je zelo točen.

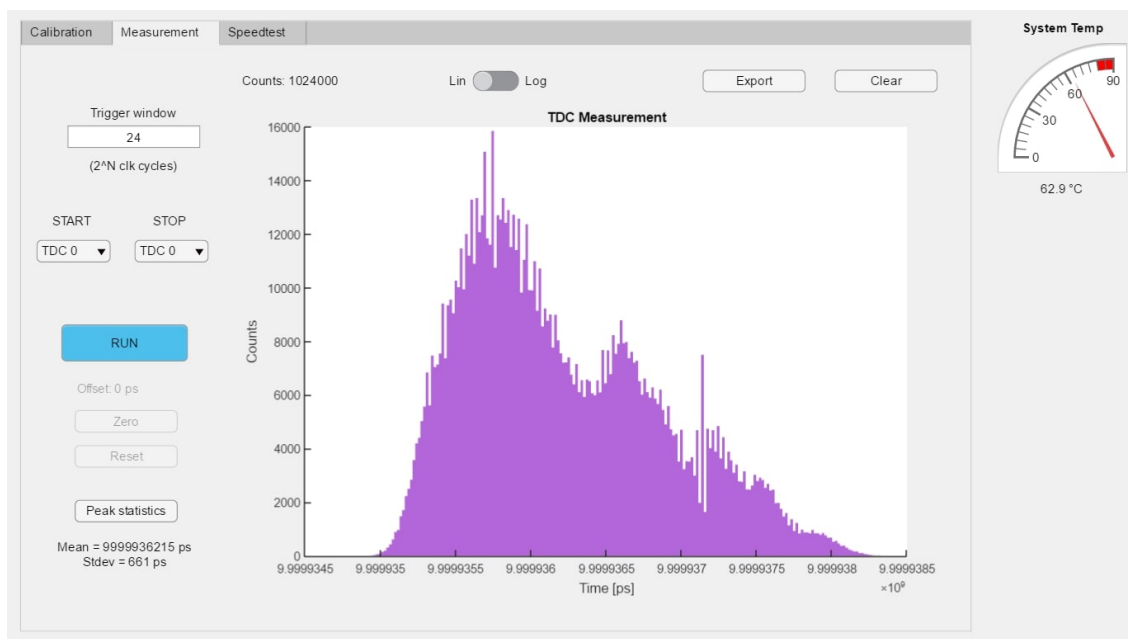
Sistematska napaka enokanalnih časovnih meritev je torej znotraj 10 ppm, ne glede na razliko med sosednjimi značkami. Ne vem, kaj je bolj impresivno; to, ali pa tista 11 ps ločljivost, ki smo jo izmerili prej. Vsekakor se mi je izdelava TDCja izjemno dobro posrečila. Je pa res, da sem v njegovo načrtovanje vložil tudi ogromno truda in predvsem časa.

4.6 Stabilnost

Zanimalo me je, kako stabilen je TDC pri zelo dolgih meritvah zunaj na 'sobni' temperaturi, ki ni regulirana. Vzel sem Agilentov generator, mu nastavil izhod na točno 100 Hz (perioda 10^7 ns) in pognal meritev periode s kanalom TDC0 do zapolnitve 1024000 značk, kar je trajalo približno 3 ure. Okno meritve vidimo na sliki 4.33.

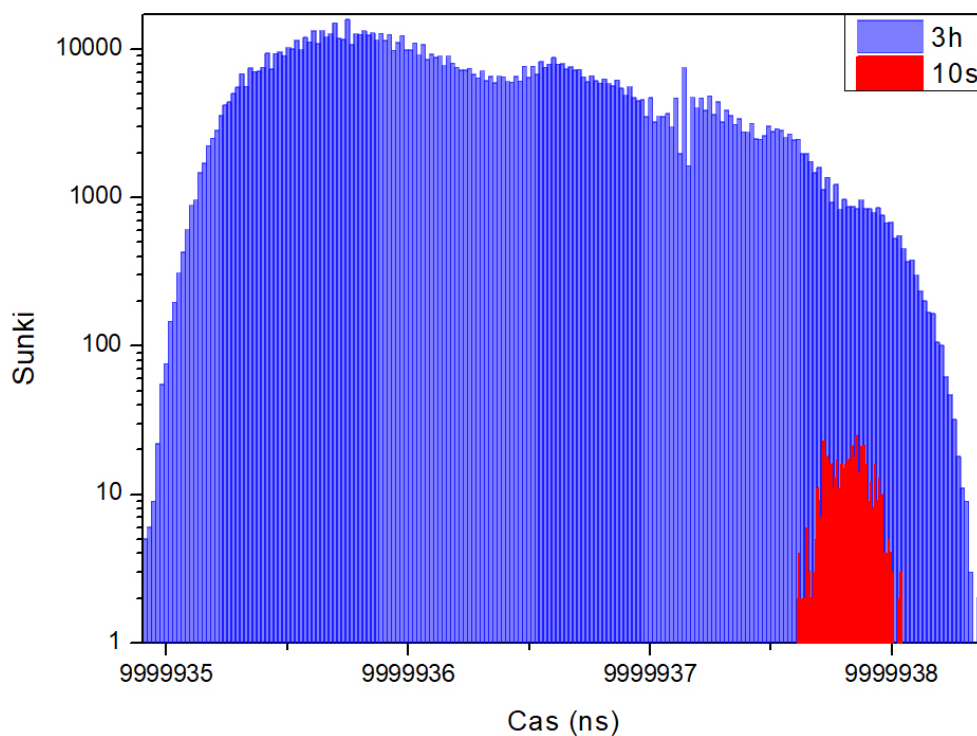
Za primerjavo sem naredil še kratko 10-sekundno meritev istega signala. Obe lahko vidimo na logaritemski skali na sliki 4.34. Opazimo, da je povprečje dolge meritve napram kratki v treh urah odtavalo za približno 2,5 ns, kar je $\sim 0,2$ ppm

Poglavje 4. Karakterizacija inštrumenta



Slika 4.33: Tri urna meritev periode 100 Hz signala iz generatorja Agilent 33220A pri sobnih pogojih.

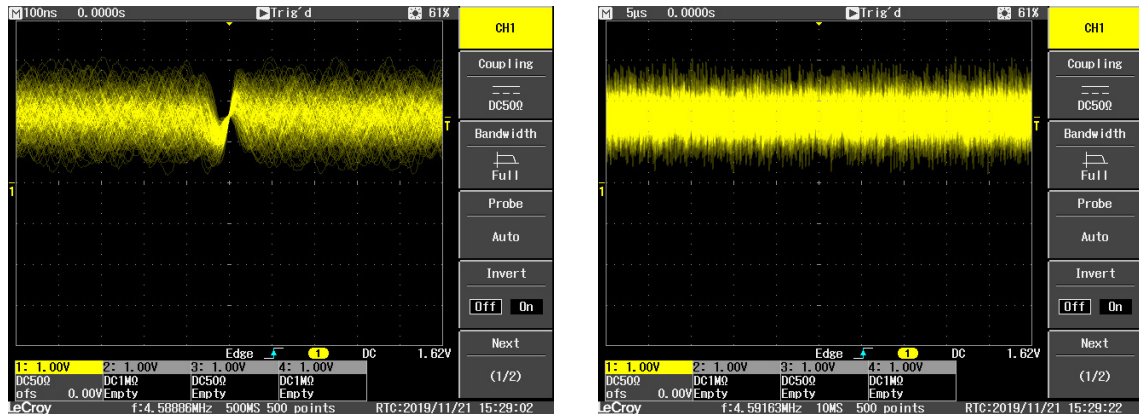
periode. V tem času se je temperatura čipa spremenila za približno eno stopinjo. To se popolnoma ujema s tem, kar smo izmerili v prejšnjem odseku. Predvidevamo, da ima nihanje temperature glavni vpliv na dolgoročno stabilnost TDCja.



Slika 4.34: 10-sekundna in 3-urna meritev periode 100 Hz signala.

4.7 Dodatek: Generator šuma

Za konec sem na generatorju Agilent 33220A odkril funkcijo generiranja šuma, ki na osciloskopu zglada tako, kot vidimo na sliki 4.35. Če takšen signal peljemo na vhod TDCja, bi teoretično morali z enokanalno meritvijo dobiti naključno razporejene časovne intervale med sosednjimi sunki.

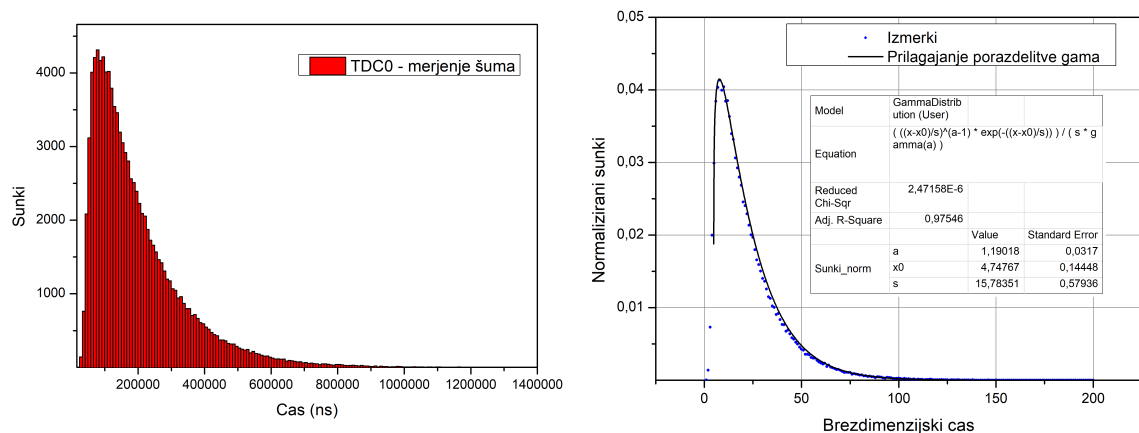


Slika 4.35: Agilentov generator 3,3 V šuma, posnet z digitalnim osciloskopom LeCroy waveJet 354. Levo je časovna skala 100 ns, desno 5 μ s.

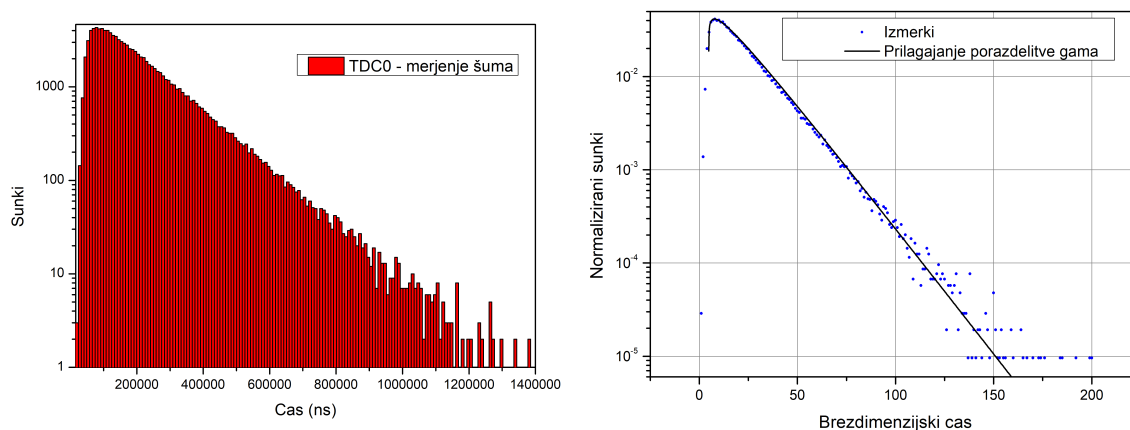
Rezultat meritve v linearni skali vidimo na sliki 4.36 levo, v logaritemski pa na sliki 4.37 levo. Predvidevamo, da gre za Poissonski (naključen) proces, pripadajoča porazdelitev pa mora odgovarjati na vprašanje ‘Kakšna je verjetnost, da bo čas med dvema zaporednima pulzoma enak Δt ?’ Kot me je z veseljem podučil kolega Žiga Krajnik, to ustreza porazdelitvi gama:

$$f(x, a, x_0, s) = \frac{\tilde{x}^{a-1} e^{-\tilde{x}}}{s \Gamma(a)}, \quad \tilde{x} = \frac{x - x_0}{s}, \quad (4.6)$$

kjer je $\Gamma(a)$ gama funkcija, a , x_0 in s pa so parametri porazdelitve.



Slika 4.36: Izmerjeni šum v linearni skali. Levo je originalna meritev, desno pa v obliki verjetnostne gostote (modre pikice), ki jo prilagajamo funkciji porazdelitve gama (črna krivulja). Prikazani so tudi rezultati prilagajanja.



Slika 4.37: Izmerjeni šum v logaritemski skali. Levo je originalna meritev, desno pa v obliki verjetnostne gostote (modre pikice), ki jo prilagajamo funkciji porazdelitve gama (črna krivulja).

Za prilagajanje modelske porazdelitve meritvam je treba najprej ustrezno pripraviti podatke. TDC aplikacija v MATLABu časovno os avtomatsko razdeli na 200 enakomernih časovnih intervalov, tako da se meritve izvozijo v obliki 200 točk 'čas-sunki'. Za namene prilagajanja statistične porazdelitve delamo kar v brezdimenzijskem času, torej od 1 do 200, pri čemer je širina posameznega intervala v tem konkretnem primeru 8350 ps. Prav tako število sunkov v posameznem izmerku delimo s skupnim številom vseh sunkov, da dobimo normalizirano verjetnostno gostoto. Rezultate prilagajanja funkcije porazdelitve gama tako pripravljenim izmerkom vidimo na slikah 4.36 in 4.37 desno. V logaritemski skali je opazen šum zaradi majhnega števila sunkov, a se izkaže, da to ne zmoti končnega rezultata prilagajanja. Dobljeni parametri porazdelitve gama znašajo $a = 1,19 \pm 0,03$, $x_0 = 4,7 \pm 0,2$ in $s = 15,8 \pm 0,6$. Ujemanje s podatki je dobro, iz česar zaključimo, da je generator relativno dober vir naključnega šuma. V oči pade le neničelna vrednost x_0 , ki predstavlja zamik porazdelitve od izhodišča. V fizikalnih enotah ta premik znaša ~ 50 ns, torej generator šuma ni zmožen generirati zaporednih sunkov, ki so bližje od tega časa.

4.8 Povzetek lastnosti TDCja

Tabela 4.1 prikazuje povzetek izmerjenih lastnosti posameznega kanala časovno-digitalnega pretvornika na čipu Zynq-7010.

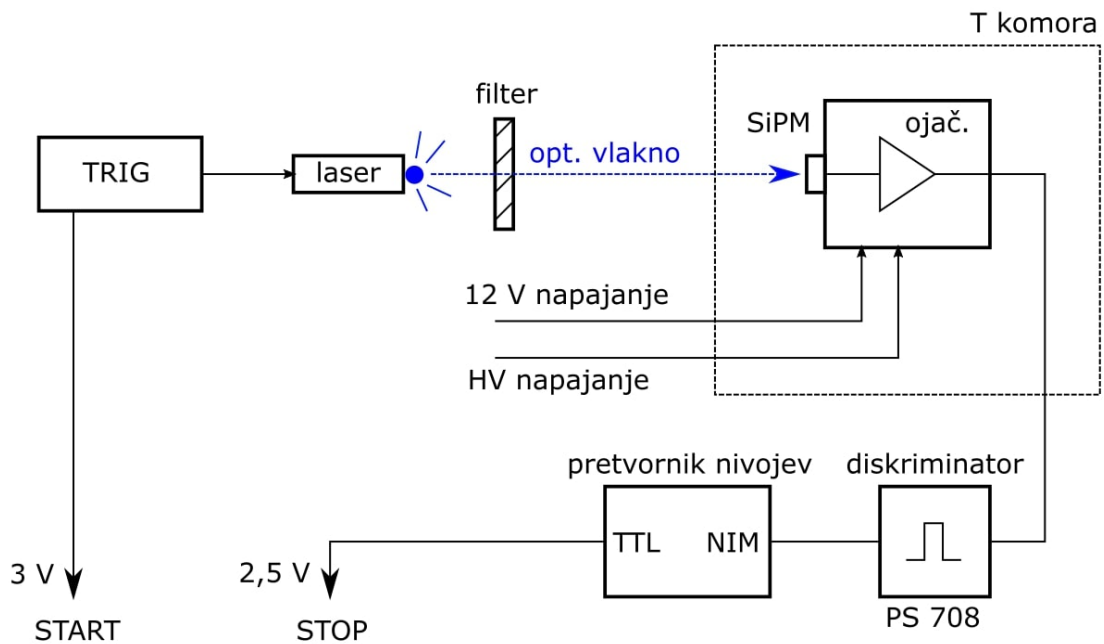
Tabela 4.1: Povzetek lastnosti kanala TDC

Frekvenca jedra	350 MHz
Št. elementov linije	192 (konfigurabilno)
Časovna ločljivost	≥ 11 ps
Natančnost	< 10 ppm
DNL	od -1 do $+4,5$ LSB
INL	od $+0,5$ do $+8,5$ LSB
Merilno območje	47,9 ms
Mrtvi čas	~ 14 ns
Max. hitrost	~ 70 MS/s

Poglavje 5

Fizikalni eksperiment s SiPM

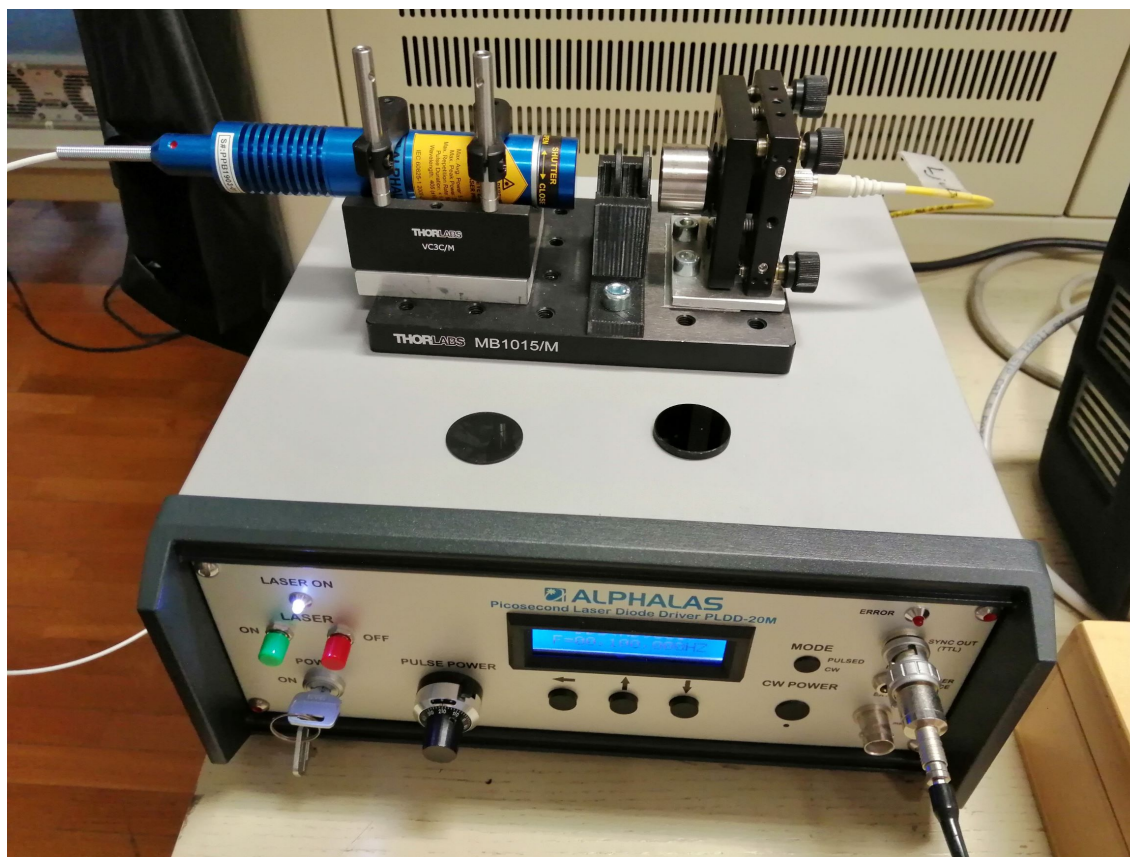
Zdaj, ko je novi merilni inštrument prestal vse teste, ga lahko začnemo uporabljati v resničnem svetu. V tipičnih fizikalnih eksperimentih imamo ponavadi nek prožilni signal, ki ga peljemo na START kanal TDCja, odziv sistema pa spremljamo na kanalu STOP. Zamislili smo si postavitve s pikosekundnim pulznim laserjem, čigar žarek zaznamo s silicijevo fotopomnoževalko, krajše SiPM. Gre za polprevodniški svetlobni detektor, ki je občutljiv na posamezne fotone, njegov odziv pa je sorazmeren številu zaznanih fotonov. Shemo celotnega eksperimenta prikazuje slika 5.1.



Slika 5.1: Eksperimentalna postavitve s pulznim laserjem in silicijevo fotopomnoževalko. Ob sprožitvi laserja se zabeleži časovna značka START. Svetloba potuje po enorodovnem optičnem vlaknu in posveti na polprevodniški detektor SiPM, ki se nahaja v popolni temi znotraj temperaturne komore. Na detektor je pritisnjena zaporna napetost -54 V , ojačan izhodni signal pa vodimo na diskriminator, ki analogno napetost spremeni v logični pulz po standardu NIM. Tega po potrebi pretvorimo v TTL, da ga lahko priklopimo na naš TDC, kjer se naposled zabeleži kot značka STOP.

Poglavje 5. Fizikalni eksperiment s SiPM

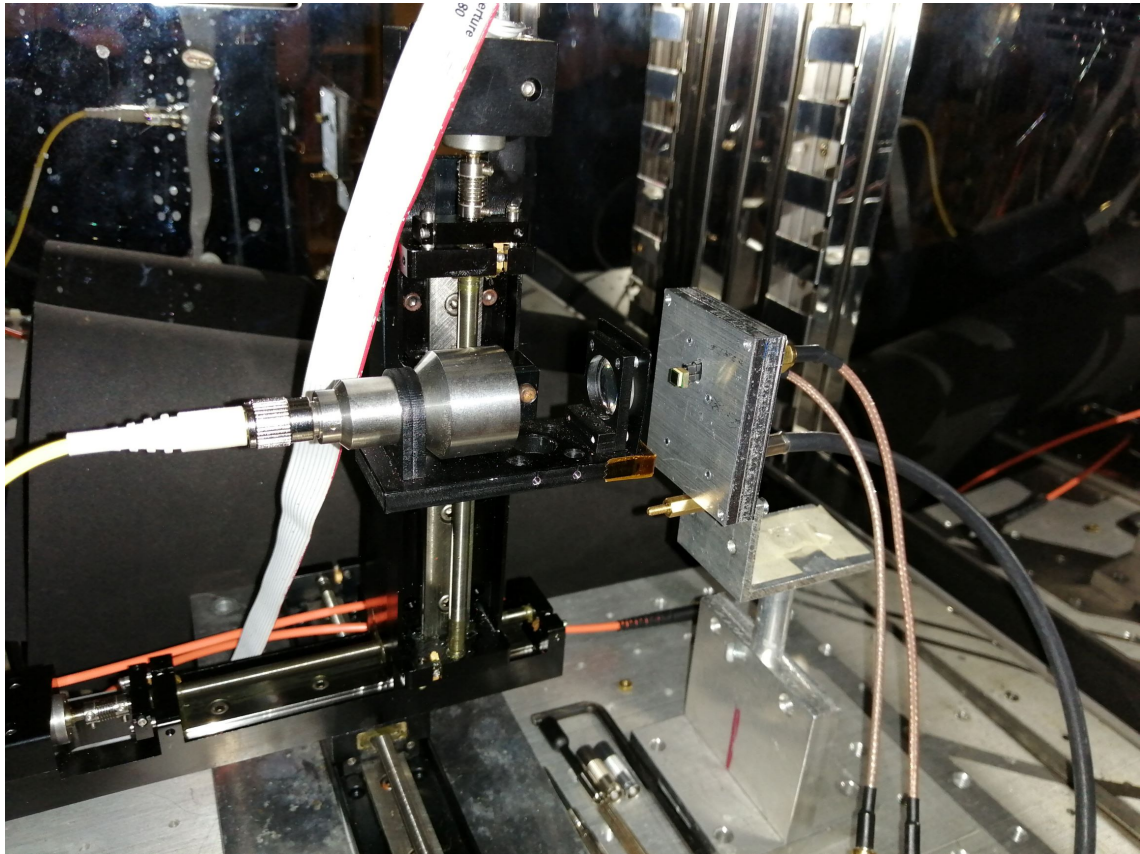
Pikosekundni pulzni laserski krmilnik proži lasersko diodo in začetni kanal TD-Cja. Za laser postavimo svetlobni filter, da zmanjšamo njegovo intenziteto, preostanek svetlobnega snopa pa vodimo po enorodovnem optičnem vlaknu do SiPM. Pred vhod v vlakno moramo postaviti posebno lečo, ki lasersko svetlobo usmeri v mikroskopsko jedro vlakna. Laser s krmilnikom, optičnim filtrom, lečo in optičnim vlaknom vidimo na sliki 5.2.



Slika 5.2: Uporabljen pikosekundni pulzni laserski sistem APLHALAS PICOPOWER-LD-405-20 z optičnimi filtri (tanke črne ploščice), zbiralno lečo in enorodovnim optičnim vlaknom. TTL izhod SYNC OUT proži začetni kanal časovno-digitalnega pretvornika.

SiPM detektor smo zaprli v že poznano temperaturno komoro in ga še dodatno pokrili s črnim pregrinjalom, da je bil med eksperimentom v popolni temi, saj želimo zaznavati izključno fotone, ki izvirajo iz laserja. Polprevodniški senzor ima zraven pritrjeno še pripadajočo napajalno in ojačevalno elektroniko, ki proizvede merljiv signal. Oboje lahko vidimo na sliki 5.3. Ojačevalnik potrebuje 12 V napajanje, na SiPM pa moramo pripeljati visoko napetost (HV) v zaporni smeri, ki je v našem primeru znašala -54 V.

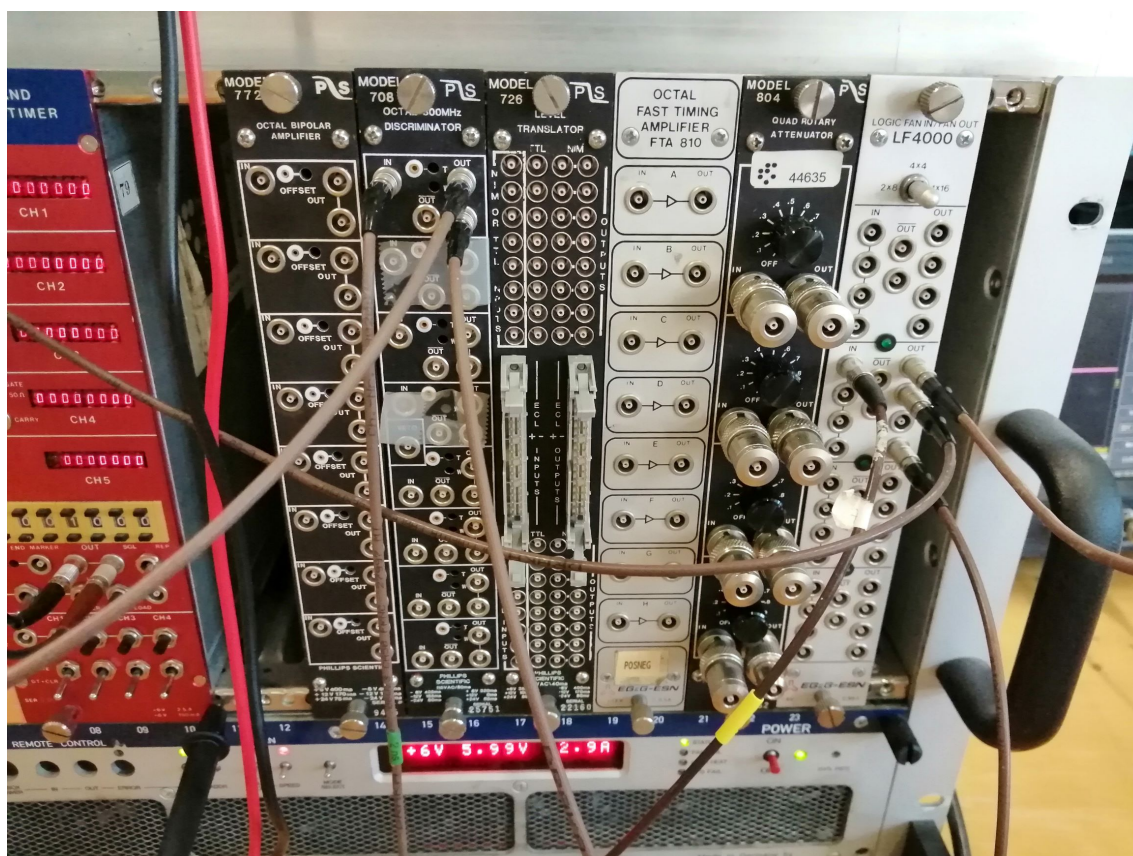
Izhod iz ojačevalnika peljemo ven na diskriminator PS 708, ki vhodne sunke pretvarja v digitalen NIM izhod. Standard NIM temelji na negativni tokovni logiki, kjer logično '1' predstavlja tok -16 mA, kar na 50Ω zaključitvenem uporju pomeni $-0,8$ V. Po drugi strani LVCMOS33 vhodi FPGA pričakujejo pozitivne napetostne nivoje do $+3,5$ V, zato NIM izhod diskriminatorja s pretvornikom logičnih nivojev spremenimo v $+2,5$ V TTL signal, ki ga lahko priklopimo na STOP kanal TDCja.



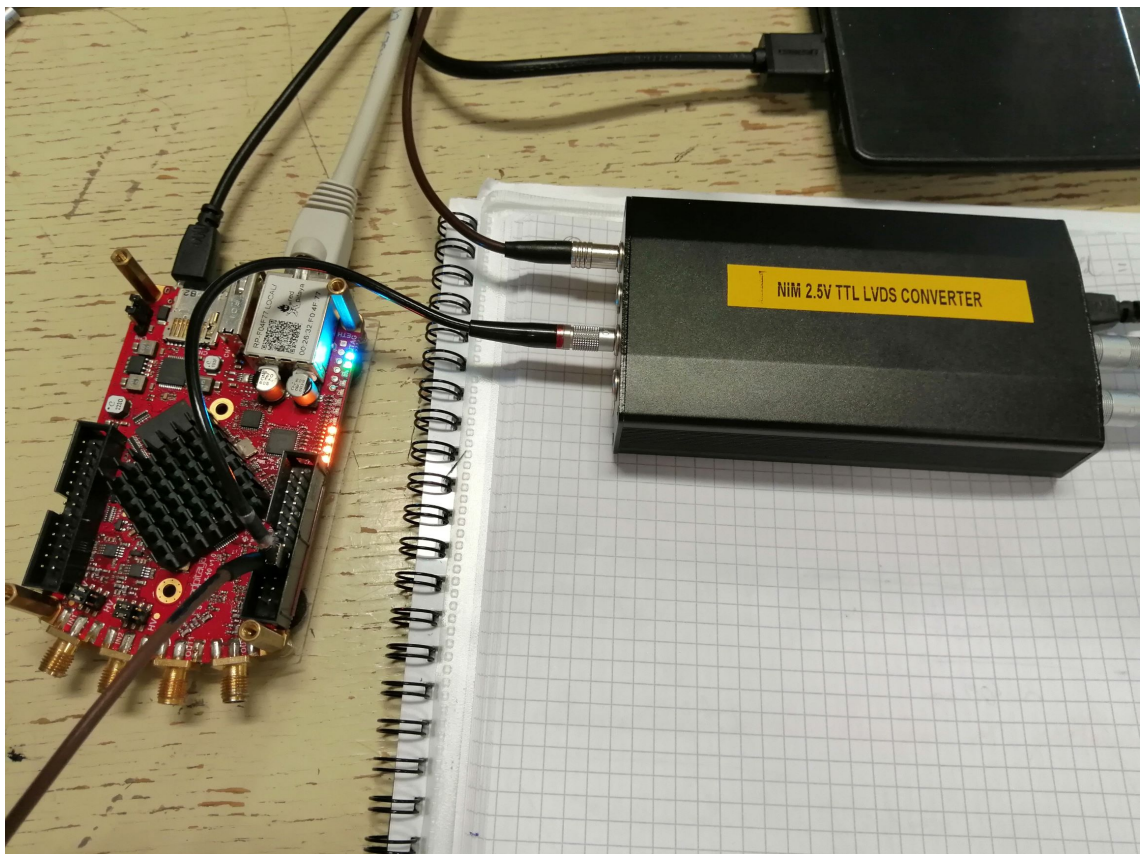
Slika 5.3: Ojačevalnik s silicijevo fotopomnoževalko, na katero svetimo z optičnim vlaknom. Ta del postavitve je zaprt v temperaturno komoro zaradi popolne teme.

Na sliki 5.4 lahko vidimo vrsto modulov NIM, od katerih uporabljamo diskriminator PS 708. Uporabljeni pretvornik nivojev z Red Pitayo prikazuje slika 5.5.

Da bi dokazali, da lahko popolnoma zaupamo izdelanemu časovno-digitalnemu pretvorniku, smo vse meritve opravili še z referenčnim 8-kanalnim TDCjem KC3781A japonskega proizvajalca Kaizuworcs, ki ga vidimo na sliki 5.6. Ponaša se z maksimalno ločljivostjo 25 ps, vsi vhodi pa uporabljajo standard NIM, zato ga lahko brez pretvorbe priklopimo direktno na izhod diskriminatorja. Kanal START prožimo s krmilnikom laserja, ki poleg uporabljenega TTL ponuja tudi sinhronizacijski NIM izhod na hrbtni strani.



Slika 5.4: Moduli NIM z diskriminatorjem PS 708 proizvajalca Phillips Scientific. Za pretvorbo signala iz standarda NIM v TTL smo sprva uporabljali pretvornik PS 726 (na sliki tik ob diskriminatorju), a se je kasneje izkazalo, da je pretvornik na sliki 5.5 mnogo hitrejši in primernejši za prenos časovno občutljivih signalov.



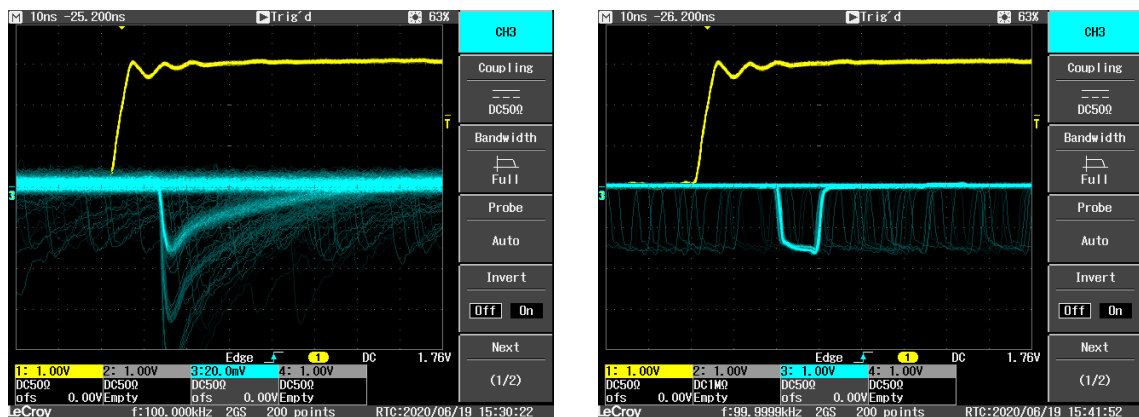
Slika 5.5: Na inštitutu izdelani dvokanalni pretvornik nivojev iz standarda NIM v 2,5 V TTL. Izhod ni sposoben napajati 50Ω bremena, zato kabel ni impedančno zaključen. Posledično mora biti povezava med pretvornikom in Red Pitayo čim krajša, da se izognemo odbojem in popačitvi signala.



Slika 5.6: Referenčni 8-kanalni NIM TDC Kaizuworks KC3781A s časovno ločljivostjo 25 ps. Na kanal START priklopimo sinhronizacijski izhod laserja, na STOP pa izhod iz diskriminatorja.

5.1 Režim posameznih fotonov

Poskus začnemo pri nizki moči laserja (nastavitev potenciometra = 392), ki ga prožimo s frekvenco 100 kHz. Intenziteta posameznih svetlobnih sunkov je tako majhna, da SiPM zaznava posamezne fotone. Ojačen signal, ki ga ob tem proizvede, lahko opazujemo z osciloskopom na sliki 5.7 levo. Gledamo modro sled, pozitivna fronta rumenega signala pa označuje trenutek proženja laserskega pulza. Na izhodu



Slika 5.7: Levo: signal silicijeve fotopomnoževalke v režimu posameznih fotonov (modra sled) pri napetostni skali 20 mV in časovni 10 ns. Lepo se vidi enofotonski vrh, kdaj pa zaznamo tudi dva ali tri fotone hkrati. Desno: NIM izhod diskriminatorja pri nastavljenem pragu -20 mV; napetostna skala je 500 mV, časovna zopet 10 ns. Rumena sled na obeh slikah predstavlja sinhronizacijski signal laserskega krmilnika, svetlobni pulz pa se sproži ob njegovi pozitivni fronti. Tudi v vseh nadaljnjih posnetkih zaslona osciloskopa bo rumeni signal vedno označeval proženje laserja.

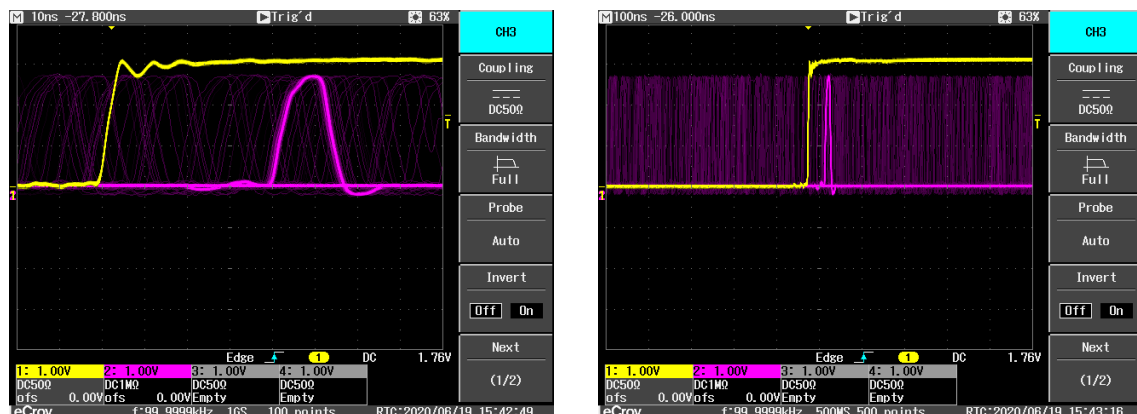
SiPM opazimo sunke, ki so posledica laserskega pulza in za njim zaostajajo dobrih 10 ns. Prisotni so enofotonski vrhovi pri amplitudi -30 mV, dvofotonski pri -60 mV in še nekaj višjih trofotonskih. Hkrati opazimo še ogromno enofotonskih vrhov, ki se pojavljajo povsod in niso korelirani z laserskim pulzom – gre za tako imenovane temne sunke zaradi termičnih ekscitacij v polprevodniku. Temnih sunkov je v tem režimu praviloma veliko več od zaznanih laserskih fotonov.

Analogni signal SiPM pretvorimo v logičnega z diskriminatorjem, ki mu nastavimo prag na -20 mV, širino izhodnih sunkov pa na približno 10 ns. Rezultat vidimo na sliki 5.7 desno. Občutljivi smo torej na vse proizvedene sunke, vključno s temnimi. NIM izhod diskriminatorja peljemo na pretvornik nivojev, njegov 2,5 V TTL izhod pa priklopimo na TDC. Slika 5.8 prikazuje obliko TTL signala, ki ga merimo s TDCjem.

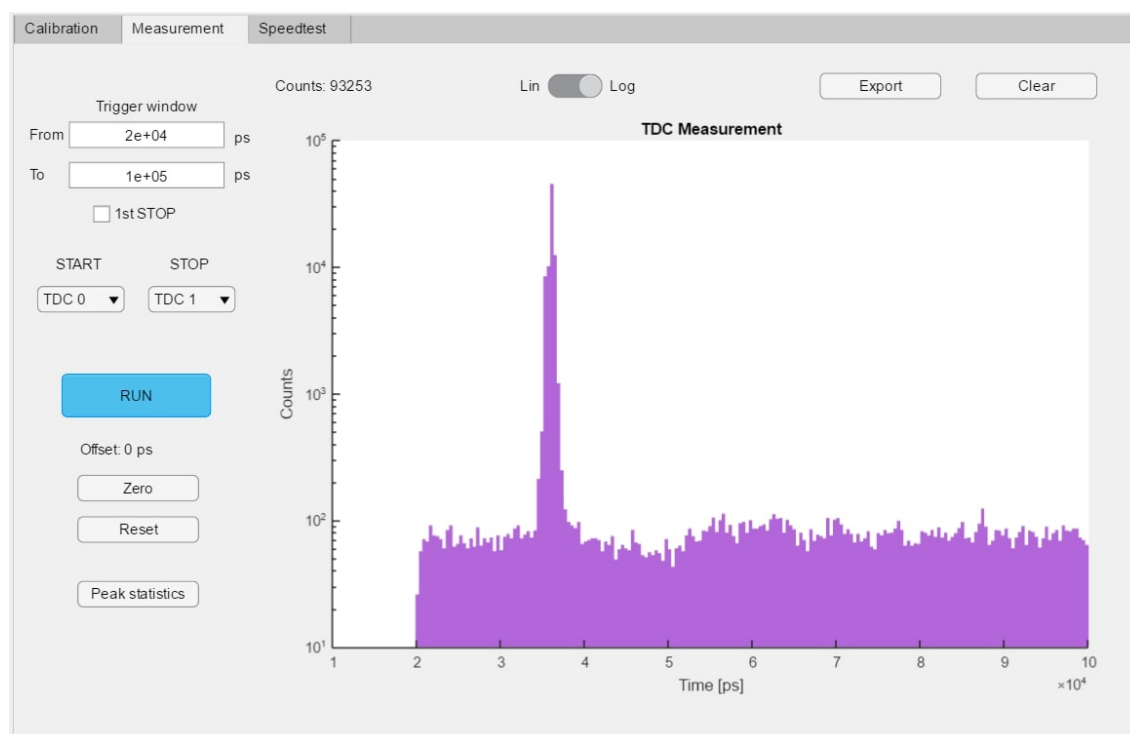
Rezultat časovne meritve z našim TDCjem vidimo na sliki 5.9, kjer je prikazano okno širine 80 ns s sunki v logaritemski skali. Opazimo konstantno ozadje temnih sunkov in zelo očiten laserski vrh. Tik za vrhom je rahla vdolbina v ozadju, široka dobrih 10–20 ns, kar ustreza času do ponastavitve diskriminatorja ob večfotonskih sunkih; glej sliko 5.7 levo.

Najbolj zanimiva je seveda meritev laserskega vrha, ki ga lahko z izdelanim časovno-digitalnim pretvornikom podrobno preučimo in primerjamo s referenčnim TDCjem. Vrh, ki ga izmerimo z Red Pitayo, prikazuje slika 5.10, tistega z NIM

Poglavje 5. Fizikalni eksperiment s SiPM



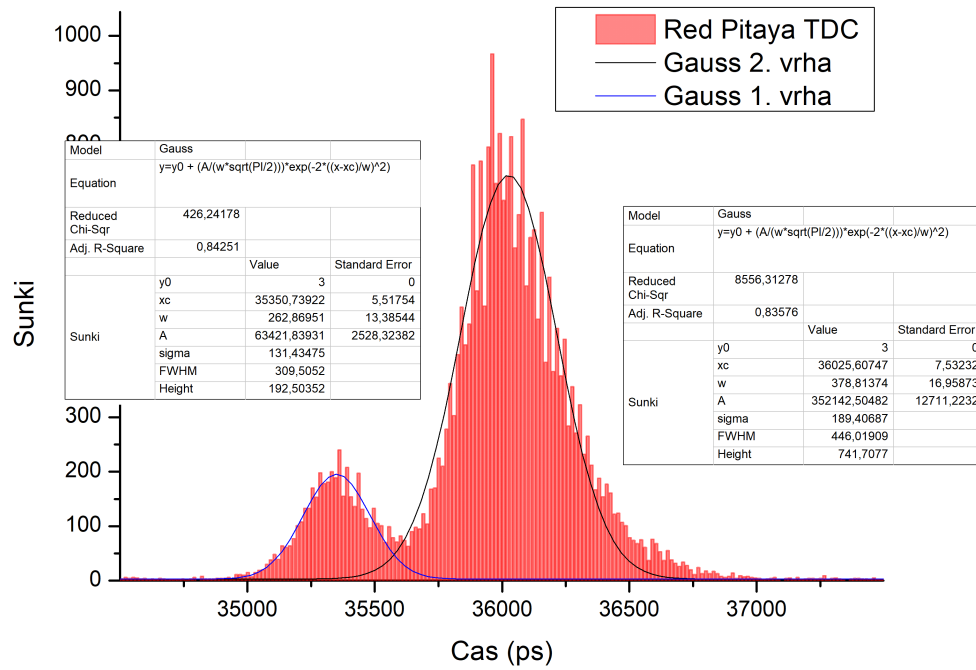
Slika 5.8: TTL signal pretvornika nivojev (rdeča sled), ki ga peljemo na STOP kanal TDCja. Synchronizacijski rumeni signal priklopimo na START. Časovna skala levo je 10 ns, desno 100 ns, napetostna pa povsod 1 V. Lepo se vidijo sunki, ki so posledica laserskega pulza, in temni sunki, ki so prisotni vsepovsod.



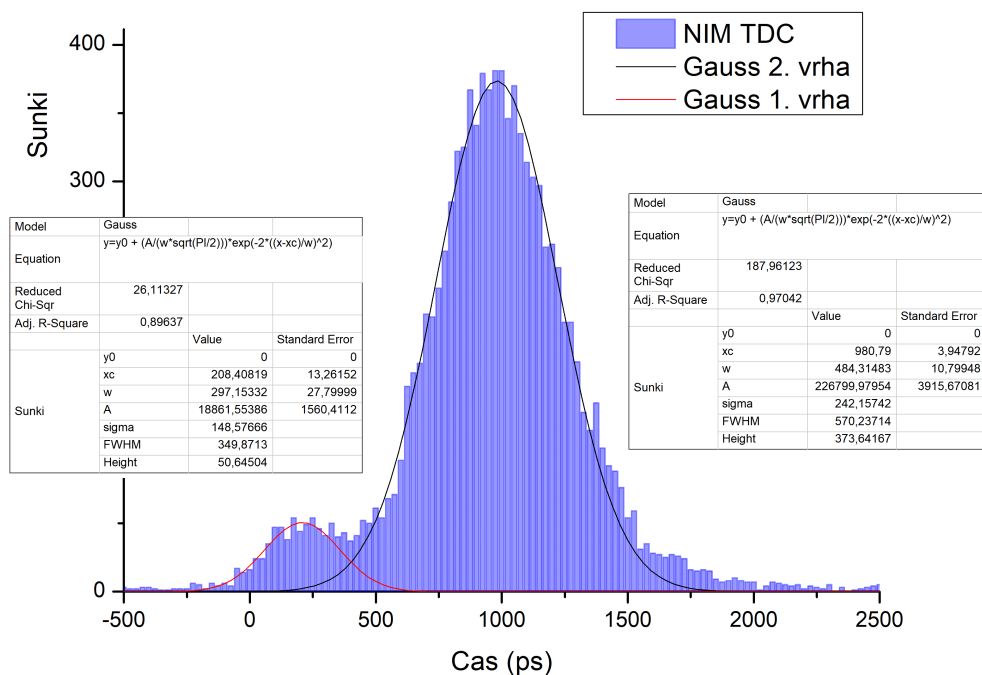
Slika 5.9: Okno TDC meritve SiPM v režimu posameznih fotonov. Širina okna je približno 80 ns, y skala pa je logaritemska. Čas 0 je laserski pulz.

TDCjem pa slika 5.11. Vidimo, da je vrh pravzaprav dvojen, kar je najverjetneje posledica dejstva, da sta strmini naraščanja enofotonskega in dvofotonskega signala različni. Če pogledamo sliko 5.7 levo, večfotonski signal narašča za odtenek hitreje od enofotonskega, zato prag diskriminatorja doseže nekoliko prej. Čas prihoda signala je torej rahlo odvisen od njegove amplitude. Razlika je minimalna, a vseeno dobro merljiva pri tako visoki časovni ločljivosti. Obema vrhovoma sem prilagodil Gaussovo funkcijo, da jima lahko določimo širino in medsebojno oddaljenost. Kot

vidimo na slikah 5.10 in 5.11, se meritvi obeh TDCjev relativno dobro ujemata, tako po obliki vrha kot izračunanih parametrih, kar je odlično. Popolnega ujemanja sicer ne gre pričakovati, ker sta postavitvi pri meritvah z enim in drugim TDCjem različni, zato so majhne razlike pričakovane. Prav tako ni primerljiv izmerjeni čas od izhodišča. Gledamo le relativne čase med vrhovoma in njuni širini.



Slika 5.10: Meritev glavnega vrha SiPM v režimu posameznih fotonov s TDCjem na Red Pitayi. Širina prvega vrha je $\sigma_1 \approx 130$ ps, drugega $\sigma_2 \approx 190$ ps, razmik med njima pa znaša približno 670 ps. Raztros RMS celotne porazdelitve v prikazanem oknu je 348 ps.



Slika 5.11: Meritev glavnega vrha SiPM v režimu posameznih fotonov s TDCjem KC3781A. Širina prvega vrha je $\sigma_1 \approx 150$ ps, drugega $\sigma_2 \approx 240$ ps, razmik med njima pa znaša približno 770 ps. Raztros celotne porazdelitve je 358 ps RMS.

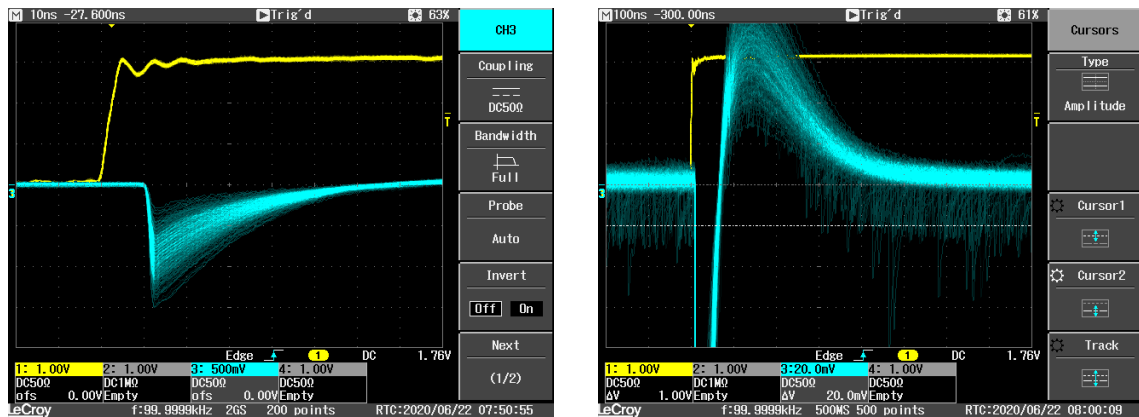
5.2 Režim visoke intenzitete

5.2.1 Nizek prag diskriminatorja

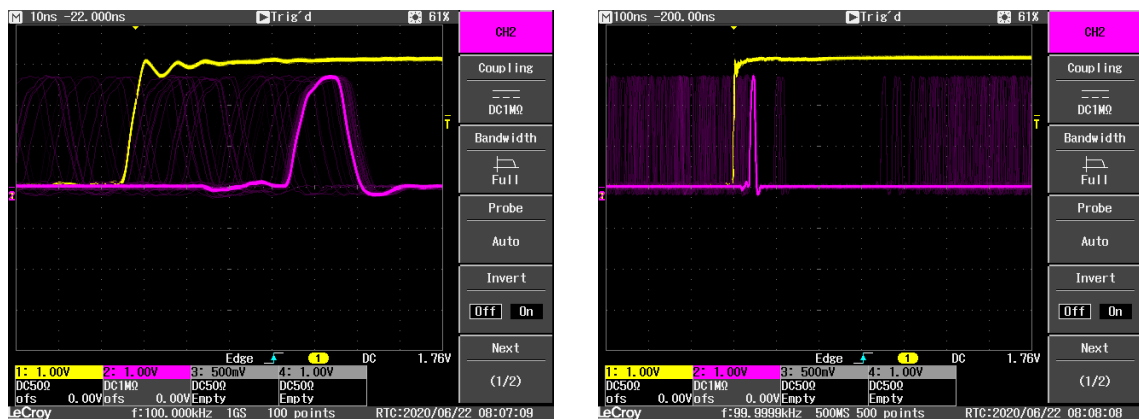
V nadaljevanju povečamo moč laserja na nastavitev potenciometra 620, prag diskriminatorja pa pustimo na -20 mV. SiPM zdaj zaznava veliko več fotonov, signal pa zraste na skoraj $-1,5$ V, kar vidimo na sliki 5.12. Diskriminator je še vedno občutljiv na posamezne fotone in s tem temne sunke, njegov izhod po pretvorbi v TTL pa prikazuje slika 5.13.

Na sliki 5.14 je 3000 ns široko okno pripadajoče časovne meritve z našim TDCjem v logaritemski skali. Pričakovano opazimo laserski vrh in ozadje, v oči pa bodeta še en manjši vrh in odsotnost ozadja še kakšnih 300–400 ns za glavnim vrhom. Krivka za to luknjo, ki smo jo opazili že z osciloskopom na sliki 5.13, najdemo v preničaju signala SiPM, ki se lepo vidi na sliki 5.12 desno in traja ravno cca. 400 ns. V tem času temni sunki zaradi njihaja enostavno ne dosežejo praga diskriminatorja, zato ozadje na videz izgine. Gre torej za artefakt meritve, temni sunki pa so v resnici vedno prisotni.

Precej bolj skrivnosten je majhen sekundarni vrh za glavnim, ki ga bolje vidimo na sliki 5.15, kjer je okno meritve povečano na 150 ns. Vrh se za primarnim pojavi po približno 50 ns in s časom eksponentno upada (na logaritemski skali to zgleda kot raven naklon). Razlago za to zopet iščemo na sliki 5.12, a tokrat levo. Kot vidimo, se signal SiPM ob zaznavi svetlobe strmo vzhva (v negativno smer) in sproži diskriminator, potem pa potrebuje približno 50 ns, da zopet upade na prag diskriminatorja, ki je na skali 500 mV praktično ob ničli. Ker signal zaradi temnih sunkov stalno nekoliko "šumi", to ponovno sproži diskriminator in ustvari navidezne



Slika 5.12: Signal SiPM (modra sled) v režimu visoke intenzitete svetlobe, kjer zaznavamo več deset fotonov hkrati. Napetostna skala levo je tokrat 500 mV, časovna pa 10 ns. Na desni vidimo močno povečan signal ob napetostni skali 20 mV, časovno pa razširimo na 100 ns. Na ta način opazimo, da ima signal ob povratku rahel prenihaj, ki traja približno 400 ns od začetka pulza. Za orientacijo je z belo črtkano črto označen prag diskriminatorja pri -20 mV.

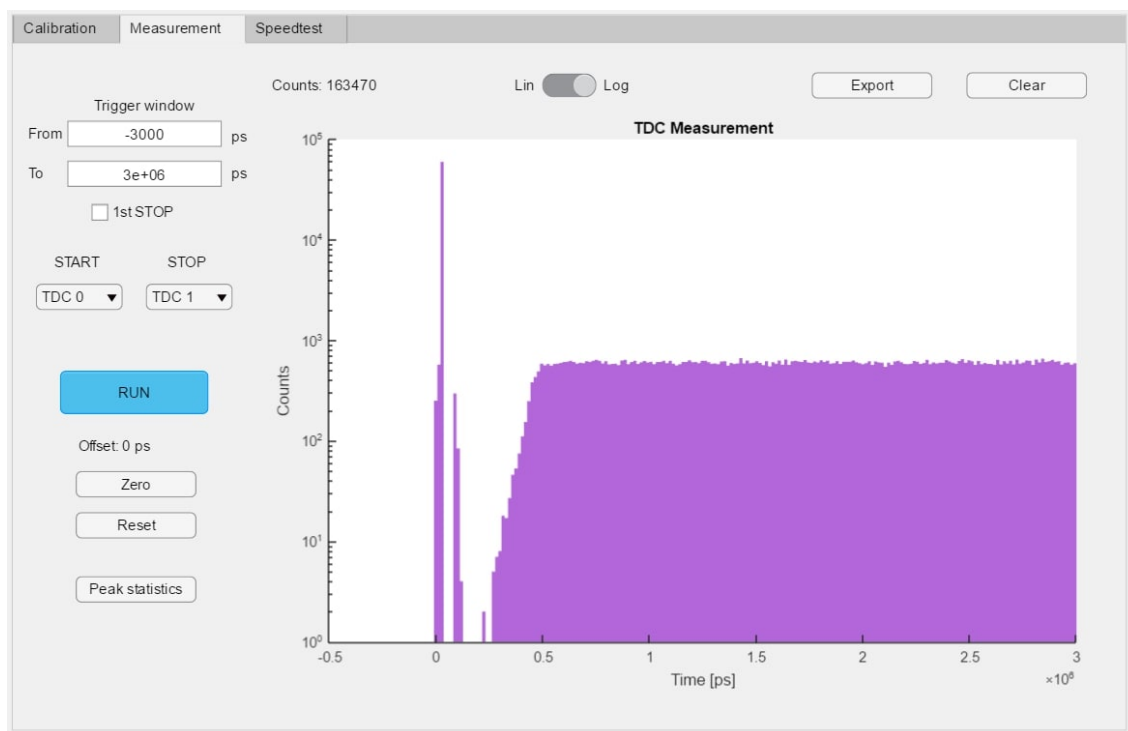


Slika 5.13: TTL signal (rdeča sled), ki ga peljemo na STOP kanal TDCja v režimu visoke intenzitete svetlobe pri pragu diskriminatorja -20 mV. Zaradi nizkega praga še vedno zaznavamo temne sunke, opazimo pa, da je ~ 300 ns za laserskim pulzom nekakšna praznina. Časovna skala levo je 10 ns, desno 100 ns, napetostna pa povsod 1 V.

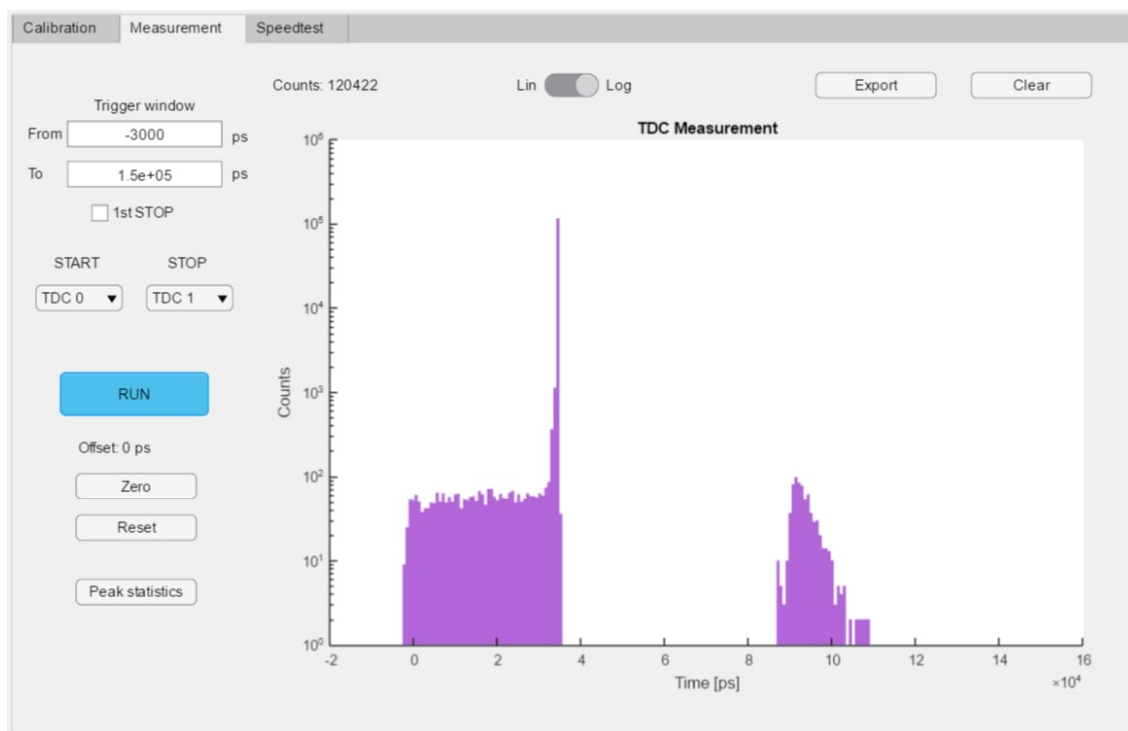
sekundarne pulze, ki ob eksponentnem upadanju signala v prenihaj za naslednjih 300 ns spet izginejo. Sumimo torej, da gre tudi tokrat za artefakt. Druga možnost pa bi lahko bila, da se za glavnim vrhom pojavljajo tako imenovani zapozneli sunki (angl. ‘afterpulsing’). Ti so posledica nosilcev naboja v SiPM, ki se za hip ujamejo na defektih silicijeve kristalne rešetke, ob sprostitvi pa povzročajo nove plazove [28].

Tako kot prej se sedaj osredotočimo na laserski vrh, ki ga podrobno izmerimo z obema TDCjema. Slika 5.16 prikazuje meritev z Red Pitayo, 5.17 pa z referenčnim inštrumentom KC3781A. Tokrat je vrh samo en, njegovi širini pa se pri obeh meritvah odlično ujemata. Vrh je znatno ožji kot v režimu posameznih fotonov, kar pomeni, da so svetlejši laserski pulzi bolj primerni za natančne časovne meritve s silicijevo fotopomnoževalko.

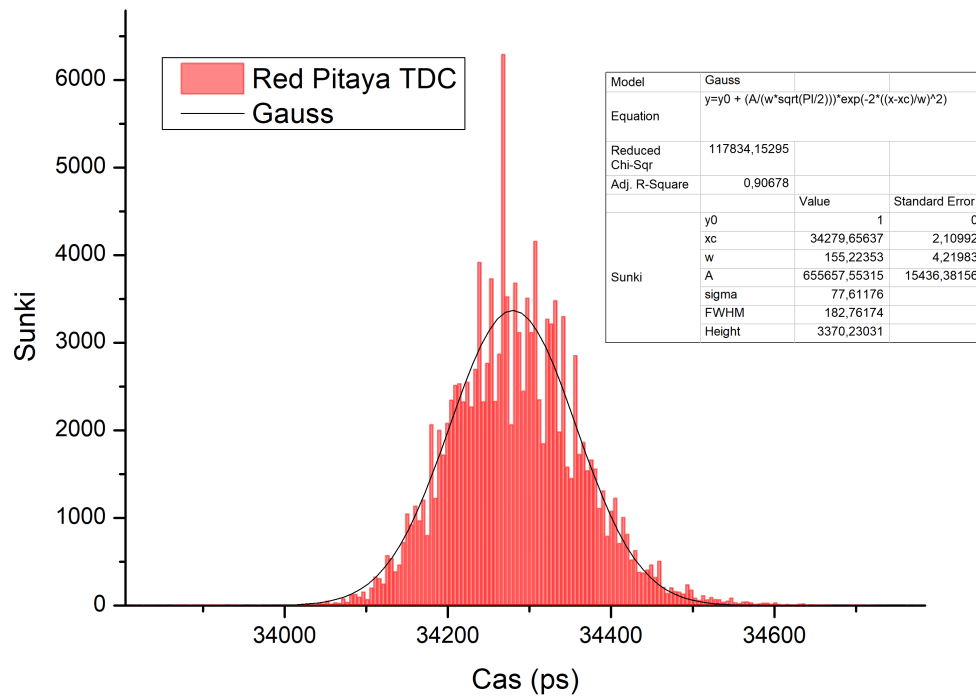
Poglavje 5. Fizikalni eksperiment s SiPM



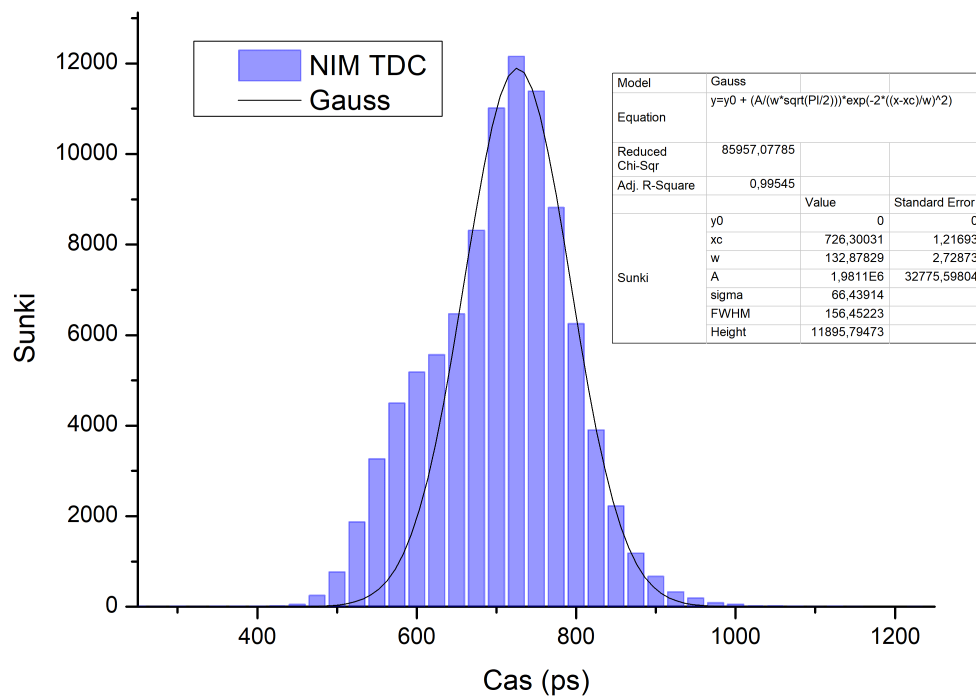
Slika 5.14: Približno 3000 ns široko okno TDC meritve SiPM v režimu visoke intenzitete svetlobe pri pragu diskriminatorja -20 mV. Skala je logaritemska.



Slika 5.15: Približno 150 ns široko okno TDC meritve SiPM v režimu visoke intenzitete svetlobe pri pragu diskriminatorja -20 mV. Sredi praznine za laserskim vrhom se pojavi še en mali vrh, ki s časom eksponentno upada. Skala je logaritemska.



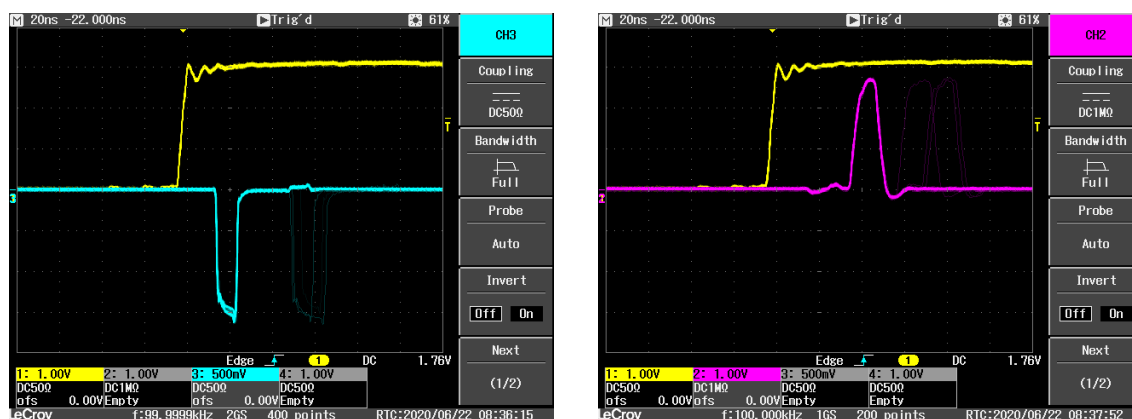
Slika 5.16: Meritev glavnega vrha SiPM v režimu visoke intenzitete s pragom -20 mV s TDCjem na Red Pitayi. Izmerjena širina vrha je $\sigma \approx 78$ ps oziroma 81 ps RMS.



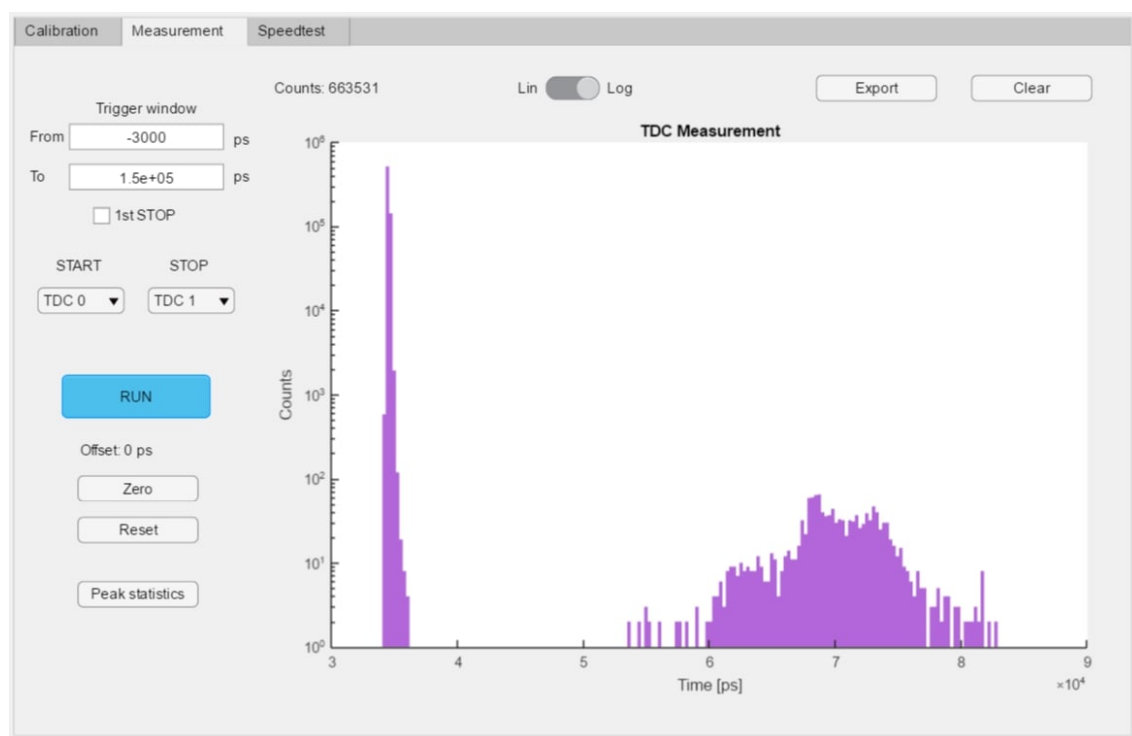
Slika 5.17: Meritev glavnega vrha SiPM v režimu visoke intenzitete s pragom -20 mV s TDCjem KC3781A. Vrh je rahlo čudne oblike in na levi znatno odstopa od Gaussove krivulje. Širina ožjega dela je $\sigma \approx 66$ ps, celotnega vrha pa 88 ps RMS.

5.2.2 Visok prag diskriminatorja

Ta del eksperimenta je enak kot prej, z edino spremembo, da prag diskriminatorja dvignemo na -150 mV, nad nivo temnih sunkov, kar pomeni, da ozadje izgine. Izhod iz diskriminatorja in pripadajoči signal po pretvorbi v TTL vidimo na sliki 5.18, okno celotne TDC meritve pa na sliki 5.19. Približno 30–40 ns za laserskim vrhom zopet opazimo nek sekundarni signal, ki ima isti vzrok kot prej.

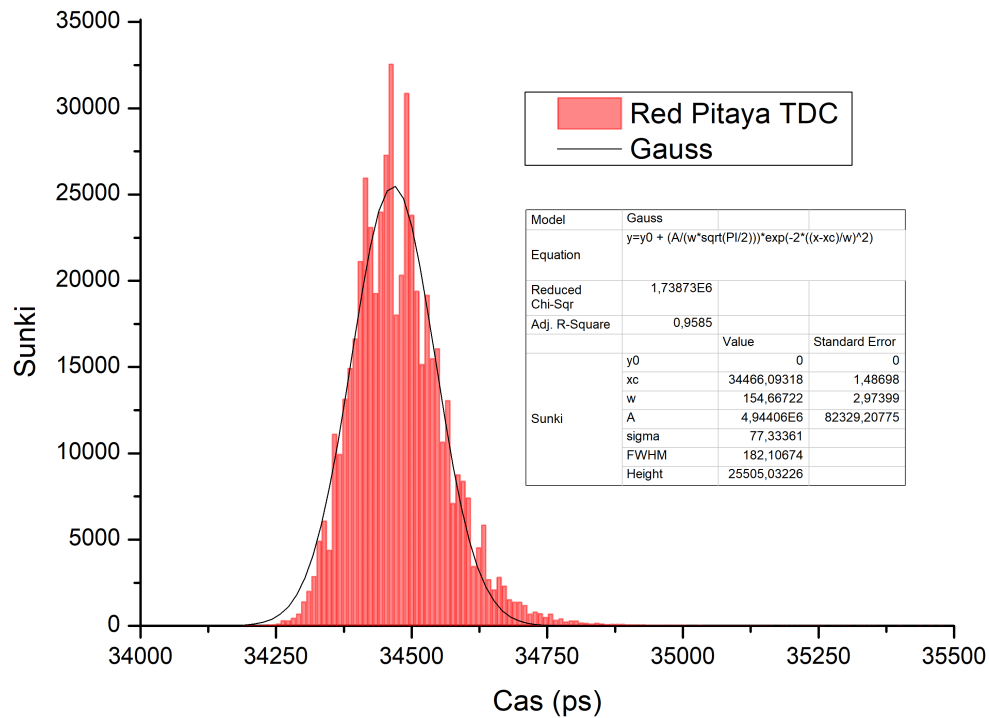


Slika 5.18: NIM izhod iz diskriminatorja (modra sled) in TTL signal (rdeča sled), ki ga peljemo na STOP kanal TDCja v režimu visoke intenzitete svetlobe pri pragu diskriminatorja -150 mV. Napetostna skala levo je 500 mV, desno 1 V, časovna pa povsod 20 ns. Signal je čist, brez temnih sunkov.

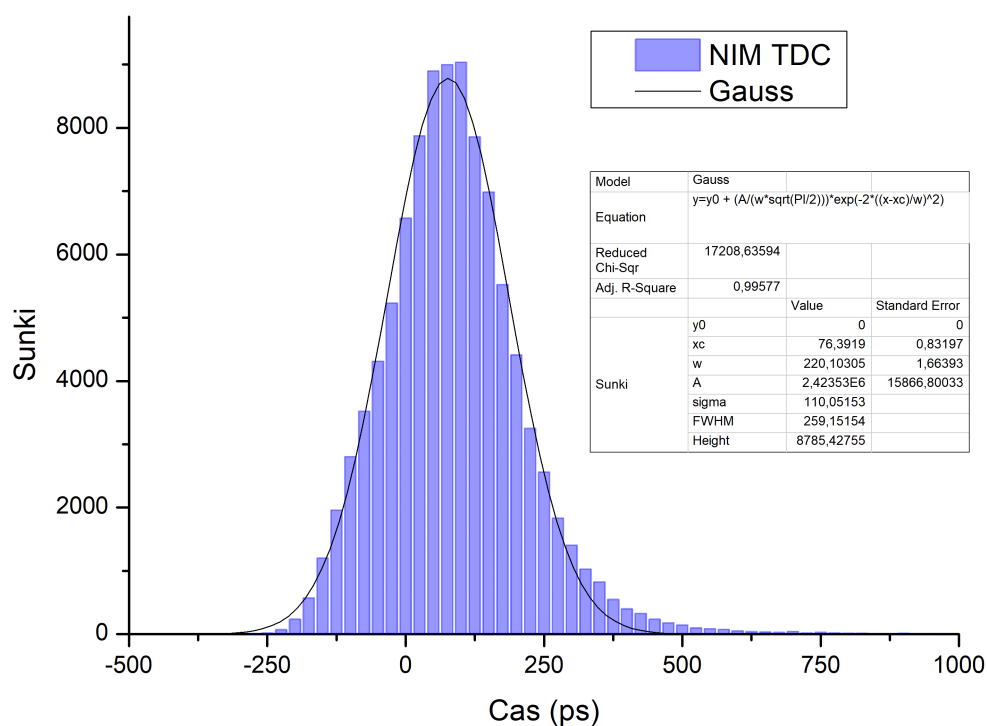


Slika 5.19: Celotno, približno 60 ns široko okno TDC meritve SiPM v režimu visoke intenzitete svetlobe pri pragu diskriminatorja -150 mV. Skala je logaritemska.

Če želimo podrobno preučiti le prvi vrh, brez ročnega popravljanja okna meritve, pride prav na novo dodana opcija “1st STOP” v meniju na levi strani grafičnega vmesnika, ki prikazuje le značke, ki prispejo prve za signalom START. Na ta način se avtomatsko osredotočimo le na direktno posledico laserskega pulza brez kasnejših motenj. Referenčni TDC KC3781A tako deluje že po zasnovi, zato lahko z njim vedno merimo le glavni vrh. Obe meritvi vidimo na slikah 5.20 in 5.21. Širina vrha je enaka kot prej pri pragu -20 mV, se je pa nekoliko spremenila njegova oblika; na desni strani se pojavi nekakšen eksponentni rep. Iz neznanega razloga se TDC KC3781A tokrat odreže nekoliko slabše.



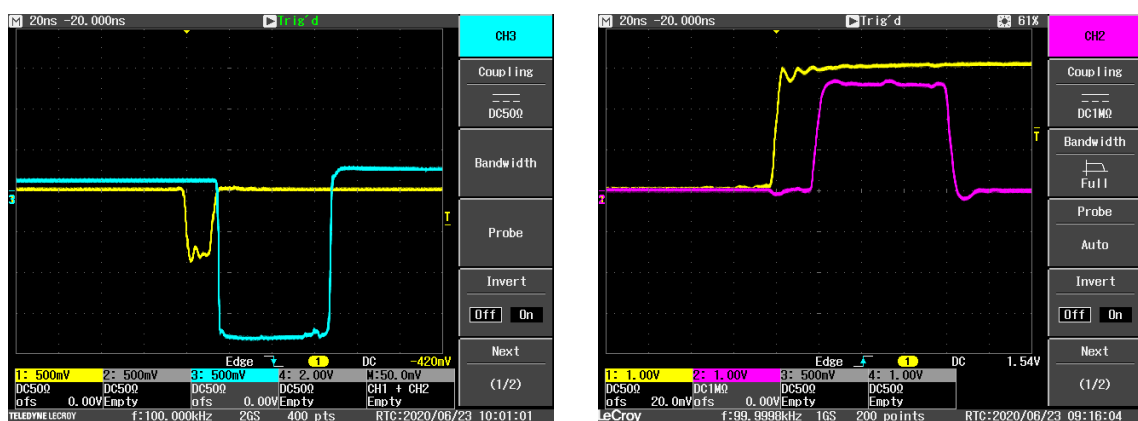
Slika 5.20: Meritev glavnega vrha SiPM v režimu visoke intenzitete s pragom -150 mV s TDCjem na Red Pitayi. Izmerjena širina vrha je $\sigma \approx 77$ ps oziroma 86 ps RMS. Na desni strani opazimo majhen eksponenten rep, zato prilagajanje Gaussove funkcije ni povsem idealno.



Slika 5.21: Meritev glavnega vrha SiPM v režimu visoke intenzitete s pragom -150 mV s TDCjem KC3781A. Izmerjena širina vrha je $\sigma \approx 110$ ps oziroma 127 ps RMS, kar je nekoliko slabše od pričakovanega. Na desni strani opazimo majhen eksponenten rep, zato prilagajanje Gaussove funkcije ni povsem idealno.

5.3 Zasičenje

Ker nas zanima, kakšna je največja možna časovna ločljivost uporabljenega laserja in silicijeve fotopomnoževalke, gremo za konec v skrajnost in izpred laserja odstranimo optični filter. Ob tem popolnoma zaslepimo SiPM, analogen izhod detektorja pa v bistvu postane digitalen, ker je predojačevalnik v totalnem zasičenju. Iz sistema zato odstranimo diskriminator in kot NIM logični signal interpretiramo kar direktno izhod SiPM, ki ga vidimo na sliki 5.22 levo. Signal detektorja sega do približno $-1,8$ V, kar je še vedno primerno za vhode NIM. Na desni vidimo isti signal po pretvorbi v nivo TTL, ki ga potrebujemo za priključitev na Red Pitayo.

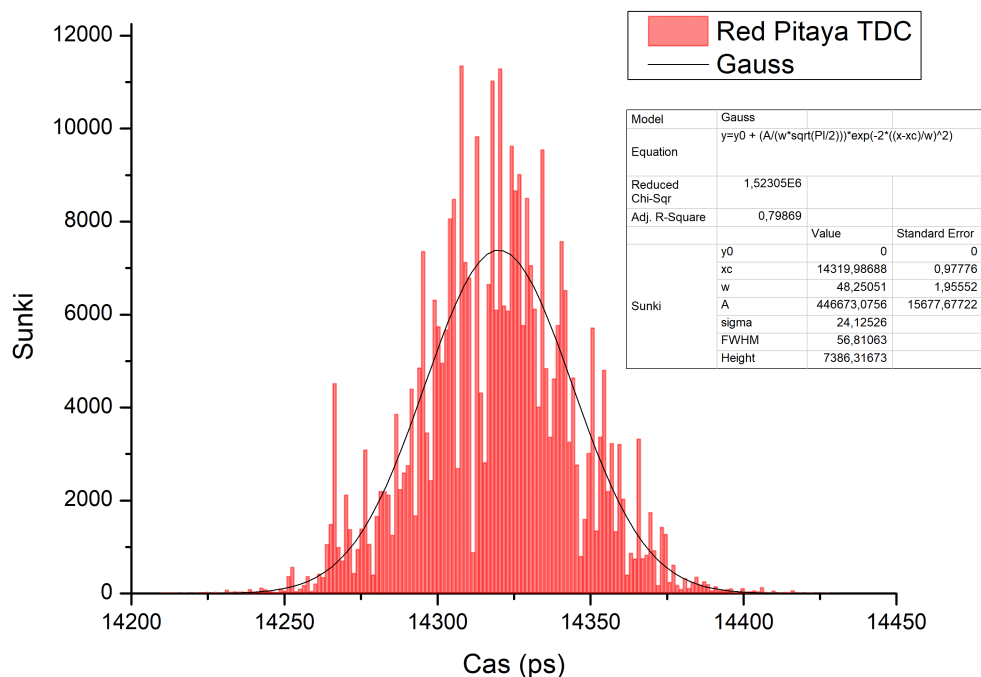


Slika 5.22: Levo: izhod SiPM v zasičenju (modra sled), ki ga lahko interpretiramo kot logični signal NIM, skupaj z NIM sinhronizacijskim signalom laserskega krmilnika (rumena sled). Oboje priključimo direktno na TDC KC3781A. Napetostna skala je 500 mV, časovna 20 ns. Desno: izhod SiPM po pretvorbi v TTL (rdeča sled) s TTL sinhronizacijskim signalom laserskega krmilnika, ki ju priključimo na Red Pitayo. Napetostna skala je 1 V, časovna ponovno 20 ns.

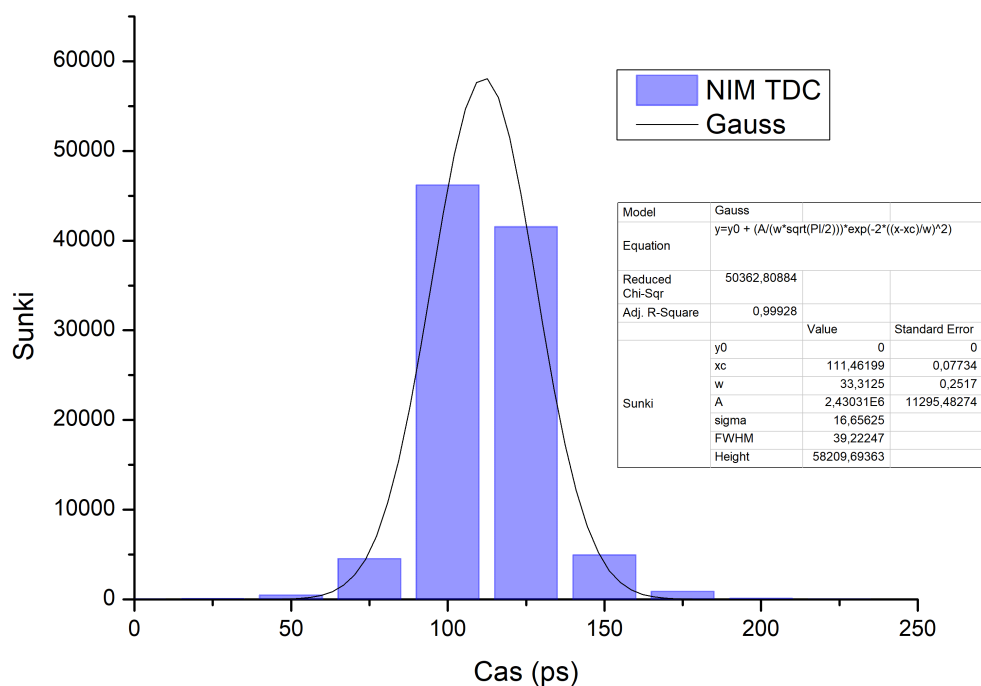
Časovna ločljivost, ki jo dosežemo na ta način, je bistveno boljše kot prej in znaša neverjetnih 24 ps, kar je na sami meji obeh TDCjev. Meritev prikazujeta sliki 5.23 in 5.24. Kot vidimo, referenčni TDC KC3781A meritve sortira v fiksne časovne intervale širine 25 ps, zato dobimo zelo zrnat histogram.

V tej konfiguraciji torej SiPM deluje v režimu najvišje časovne ločljivosti, ki je primerljiva z resolucijo izdelanega časovno-digitalnega pretvornika. To lahko s pridom izkoristimo; spomnimo se odseka 4.4.1 iz prejšnjega poglavja, kjer smo preučevali odvisnost ločljivosti TDCja od medkanalne zakasnitve. Pri električnih vodnikih smo nad 100 ns naleteli na težave zaradi prevelikega dušenja signala v dolgem kablu, pri postavitvi z laserjem in SiPM pa tega problema nimamo, ker so izgube v optičnem vlaknu bistveno manjše kot v bakru. Z optičnim kablom, ki ga uporabljamo pri tem eksperimentu, lahko dosežemo zakasnitev do 260 ns, kar vidimo na grafu na sliki 5.25. Graf je identičen tistemu na sliki 4.21, le da smo mu dodali še tri točke iz postavitve laser-optika-SiPM. Še višje bi prišli z uporabo daljšega optičnega vlakna.

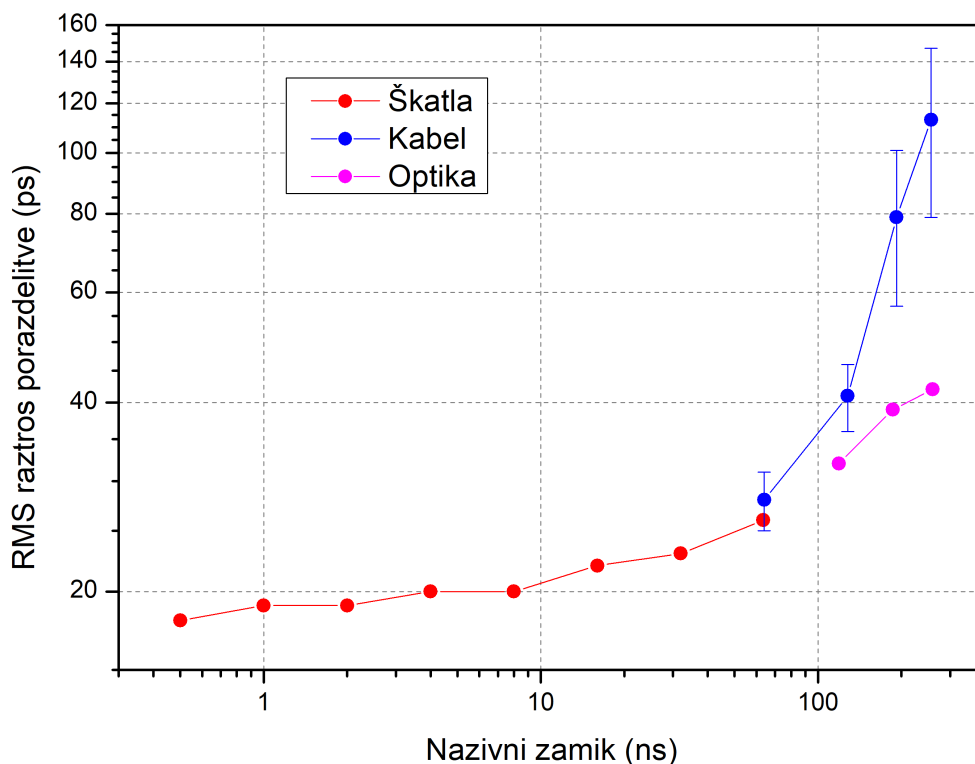
Poglavje 5. Fizikalni eksperiment s SiPM



Slika 5.23: Meritev časovnega odziva SiPM na laserski pulz v zasičenju s TDCjem na Red Pitayi. Izmerjena širina vrha je $\sigma \approx 24$ ps oziroma 25 ps RMS, kar je na robu zmogljivosti TDCja.



Slika 5.24: Meritev časovnega odziva SiPM na laserski pulz v zasičenju s TDCjem KC3781A. Izmerki so široki 25 ps, zato ima histogram le nekaj stolpcev. Za širino vrha sicer dobimo $\sigma \approx 17$ ps, kar pa zaradi majhnega števila točk ni povsem merodajen rezultat. RMS znaša 19 ps.



Slika 5.25: Izmerjena odvisnost RMS ločljivosti medkanalnih meritev v odvisnosti od časovnega intervala. Dodane so tri meritve iz eksperimenta s hitro SiPM v režimu zasičenja, kjer je časovna resolucija najvišja. Zakasnitev optične postavitve spreminjamo z dodanim generatorjem zamika DL6001 na liniji STOP.

Kot vidimo, je časovna ločljivost TDCja pri optični postavitvi bistveno boljša, kot če uporabimo električne kable, in je tudi pri 260 ns še vedno krepko pod 50 ps. Ravno kar videno je dokaz, da za doseganje dobrih rezultatov pri preciznih fizikalnih meritvah ni dovolj, da imamo natančen merilni inštrument, ampak moramo dobro poznati in optimizirati celotno eksperimentalno postavitvev.

Poglavje 6

Zaključek

Časovno-digitalni pretvorniki pikosekundnih časovnih ločljivosti so specialni merilni inštrumenti, ki tipično nikakor niso poceni. V sklopu te magistrske naloge smo zato želeli raziskati, če lahko z uporabo moderne tehnologije FPGA na cenovno dostopni plošči domačega porekla Red Pitaya realiziramo TDC s primerljivimi lastnostmi, ki bo uporaben za delo na fakulteti in inštitutu. Naloga nam je več kot odlično uspela, saj smo demonstrirali TDC z najvišjo izmerjeno ločljivostjo 11 ps, kar je krepko bolje od zadanih 50 ps. Inštrument lahko vzorči 70 milijonov značk na sekundo, merilno območje v trenutni konfiguraciji pa znaša 48 ms. TDC pride z grafičnim uporabniškim vmesnikom (aplikacijo), ki omogoča kalibriranje inštrumenta, izvajanje časovnih meritev in spremljanje temperature FPGA. Lahko izvajamo medkanalne meritve, v primeru, da sta signala START in STOP ločena, ali pa navadne enokanalne, kjer merimo razmik med sosednjimi pulzi. Tipična uporaba je merjenje frekvence. Relativna natančnost meritev je znotraj 10 ppm v 60 °C širokem temperaturnem intervalu.

Srce časovno-digitalnega pretvornika je zakasnilna linija iz elementov prenosne logike na čipu FPGA, ki omogoča natančno merjenje časa prihoda signala med sosednjimi urinimi cikli. Zakasnitev med posameznimi elementi mora biti čimbolj enakomerna, zato je pomembno, da se uporabljene rezine držijo ena druge v ravni črti, brez vmesnih prekinitev. Na srečo je razvojno orodje Vivado 2018 že samodejno ustvarilo lepo verigo, zato ročni popravki niso bili potrebni. "Na srečo" zato, ker se moderna načrtovalska orodja močno zanašajo na avtomatiko, možnosti ročnega poseganja v implementacijo pa so posledično močno okrnjene. Izkazalo se je tudi, da zakasnilna linija ni tako hudo občutljiva na spremembe temperature, kar smo lahko preverili v odsekih 4.1 in 4.4.3. Tudi ob spremembi zunanje temperature za 60 °C brez popravljanja kalibracije je ločljivost finih medkanalnih meritev ostala pod 25 ps, kar sploh ni slabo.

Izdelani TDC ni povsem fiksni, ampak se ga da tudi nekoliko prilagajati. Privzeta dolžina zakasnilne linije našega časovno-digitalnega pretvornika je 192 elementov, kar se da enostavno spreminjati po korakih po 12 ob vključevanju komponente AXI TDC v celoten sistem. V primeru implementacije TDC na Red Pitayi je bilo aktivnih le dobrih 160 elementov, zato bi linijo lahko skrajšali na 180 ali celo 168 elementov, s čimer bi ustrezno zmanjšali mrtvi čas in porabo gradnikov polja FPGA. Po potrebi se lahko podaljša tudi merilno območje TDCja, in sicer s povečanjem bitne širine grobega števca, ki trenutno znaša 24 bitov. To sicer ni mogoče na nivoju jedra IP, tako kot z zakasnitvenimi elementi, ampak je treba minimalno popraviti

VHDL kodo.

Moram priznati, da sem z izdelanim časovno-digitalnim pretvornikom izjemno zadovoljen, kar pa še zdaleč ne pomeni, da sistem ni brez pomanjkljivosti. Nekatere bi se dalo odpraviti relativno hitro, druge precej težje. Med možnimi izboljšavami, ki so na dosegu roke, je recimo optimizacija grafičnega vmesnika, ki obdeluje in prikazuje časovne značke. Nad hitrostjo delovanja trenutne aplikacije v MATLABu sem precej razočaran, saj je število obdelanih značk na sekundo veliko nižje, kot sem si sprva obetal. Zagotovo bi se dalo izboljšati algoritem osveževanja prikazanih histogramov in optimizirati kodo, da bi tekla hitreje. Mogoče ne bi bila slaba ideja, da se vse skupaj prepíše v Python, ki je verjetno hitrejši, za povrh vsega pa še brezplačen, kar niti slučajno ne velja za MATLAB.

Še ena stvar, ki bi jo bilo dobro spremeniti, se tiče same strojne opreme, natančneje vhodov časovno-digitalnega pretvornika. Trenutno na čipu FPGA uporabljamo navadne CMOS zunanje priključke z visoko vhodno impedanco, kar pa nekako po naravi ne gre skupaj s prenosom hitrih signalov, ki na koncu linije nujno potrebujejo $50\ \Omega$ zaključitvene upore, da ne pride do popačitve signala in odbojev. To je precej nerodno, ker moramo na konec kabla sami prispajkati upor, poleg tega pa to povzroči dodatne nevšečnosti, saj standarda TTL in CMOS tega upora originalno ne predvidevata. Posledično s $50\ \Omega$ zaključitvijo pogosto preobremenimo napetostni vir signala, ob konstantni logični enici pa lahko upor zaradi velike toplotne obremenitve celo uničimo, saj te male SMD komponente ne prenesejo večjih moči. Rešitev so diferencialni vhodi po standardu LVDS, ki je že v štartu zasnovan za to. Omogoča zanesljiv prenos hitrih signalov po paru žic, ki so zaključene s $100\ \Omega$ impedanco. Ta standard podpira tudi čip Zynq in ga je na Red Pitayi enostavno implementirati. Razlog, zakaj se za logiko LVDS nisem odločil že od začetka, je v tem, da večina funkcijskih generatorjev nima diferencialnih izhodov, zato je bilo TDC lažje testirati z navadnimi priključki CMOS in ročno prispajkanimi $50\ \Omega$ upori.

Nekatere druge omejitve TDCja niso tako lahko odpravljive, ker izvirajo iz same zasnove časovno-digitalnega pretvornika. Že med načrtovanjem vezja so se začela pojavljati vprašanja o skalabilnosti TDCja na večje, mnogokanalne sisteme, ki jih potrebujemo v napravah ToF-PET ali pa detektorjih delcev velikih pospeševalnikov. Trenutna zasnova je namreč taka, da posamezen kanal TDC shranjuje časovne značke v svoj pomnilnik BRAM, ki je del naslovnega prostora procesorskega sistema Zynq. Enota TDC je torej na sistemsko vodilo AXI4 priklopljena kot suženj in čaka, da procesor pride prebrat izmerjene značke. Tak pristop komunikacije med časovno-digitalnim pretvornikom in procesorjem je sicer najbolj enostaven, a verjetno ne najbolj primeren za večkanalne sisteme. Že pri implementaciji dveh kanalov smo videli, da je celotna poraba gradnikov polja FPGA nenavadno velika, zato bi za mnogo kanalov potrebovali precej večje in dražje vezje. Slutimo tudi težave s hitrostjo vezja; matrika povezljivosti AXI ni zmogla niti privzete frekvence $125\ \text{MHz}$, zato jo je bilo treba znižati na $100\ \text{MHz}$. Pri sistemu z mnogimi kanali bi povezljivost za AXI zasedala večino čipa, kar bi skoraj zagotovo pomenilo še počasnejše vezje. A težava ni samo v tem. Glavni problem je pravzaprav v sami ideji, da mora procesor zaporedno brati značke iz posameznih kanalov, enega za drugim. Kot smo videli v poglavju 4.3, to že samo pri enem kanalu pomeni konkretno ozko grlo. Če pa bi hoteli brati recimo 10, ali celo 100 kanalov hkrati, bi hitrost prenosa značk iz posameznega TDC padla na povsem neuporabno vrednost.

Ravnokar omenjeni pristop je odličen za demonstracijo delovanja izdelanega

časovno-digitalnega pretvornika ter za manj zahtevno znanstvenoraziskovalno rabo, študentske vaje itd. Kar je bil navsezadnje namen tega magisterija. V primeru, da bi hoteli izdelati profesionalen mnogokanalni sistem za zahtevnejše eksperimente, na primer v fiziki osnovnih delcev, pa bi se bilo treba zadeve lotiti drugače. Zasnovo jedra TDC lahko obdržimo, a namesto da čakamo na procesor, lahko značke s pretokom podatkov samodejno pošiljamo naprej. Za to bi uporabili protokol AXI4-Stream, s katerim bi se izmerki preko direktnega dostopa do pomnilnika (DMA) pretakali v sistemski pomnilnik, kjer so potem na voljo za obdelavo. Še boljša ideja pa je, da pretok preusmerimo na posebne zelo hitre serijske oddajnike, ki so na voljo na zmogljivejših čipih FPGA in omogočajo direktno pošiljanje podatkov v omrežje, kjer jih lahko potem shranjujemo na računalniške gruče. Seveda bi bilo treba zamenjati tudi Red Pitayo s kakšno bolj konkretno ploščo z večjim vezjem FPGA in mnogo vhodno/izhodnimi priključki. Na čipu bi tekel kvečjemu kakšen kontrolni program, brez operacijskega sistema.

V tem magistrskem delu predstavljeni TDC je odlična pridobitev za našo fakulteto in širšo fizikalno skupnost, primeren pa je tako za izobraževalne kot raziskovalne namene. Prav je, da izdelek ne postane sam sebi namen, ampak se začne temu primerno uporabljati. Prvi projekt prihaja že to poletje, in sicer inštalacija s TDCjem opremljene Red Pitaye pri vaji Kotna korelacija anihilacijskih žarkov gama na Praktikum, kjer med drugim merimo koincidence γ fotonov. Gre tudi za izjemno priložnost, da se naši študentje fizike že na prvostopenjskem študiju spoznajo s čedalje bolj pomembno tehnologijo programirljivih vezij FPGA.

Literatura

- [1] J. Kalisz, "Review of methods for time interval measurements with pico-second resolution," *Metrologia*, vol. 41, no. 1, str. 17–32, februar 2004. DOI:10.1088/0026-1394/41/1/004.
- [2] C. Ugur, E. Bayer, N. Kurz in M. Traxler, "A 16 channel high resolution (<11 ps RMS) Time-to-Digital Converter in a Field Programmable Gate Array," *Journal of Instrumentation*, vol. 7, C02004, februar 2012. DOI:10.1088/1748-0221/7/02/C02004.
- [3] Texas Instruments, "TDC7200 Time-to-Digital Converter for Time-of-Flight Applications in LIDAR, Magnetostrictive and Flow Meters," *podatkovni list za TDC7200*, februar 2015 [posodobljeno marca 2016]. Dostopno na <http://www.ti.com/lit/ds/symlink/tdc7200.pdf?ts=1591283881691>.
- [4] R. E. Schmitz, A. M. Alessio in P. E. Kinahan, "The Physics of PET/CT scanners," University of Washington. Dostopno na <http://depts.washington.edu/imreslab/education/Physics%20of%20PET.pdf>.
- [5] M. Adamič, "Digital IC design using FPGA technology," seminar v drugem letniku magistrskega študija, Fakulteta za matematiko in fiziko, marec 2019. Dostopno na <https://www.ijs.si/ijsw/F9/dipl>.
- [6] Xilinx. <https://www.xilinx.com/products/silicon-devices/soc/zynq-7000.html>.
- [7] ZedBoard. <http://zedboard.org/>.
- [8] Red Pitaya. <https://www.redpitaya.com/>.
- [9] J. Serrano, "Digital Signal Processing Using Field Programmable Gate Arrays," *Proc. 13th Beam Instrumentation Workshop (BIW'08)*, Tahoe City, CA, maj 2008, članek THTTT01, str. 346–355.
- [10] J. Song, Q. An in S. Liu, "A High-Resolution Time-to-Digital Converter Implemented in Field-Programmable-Gate-Arrays," *IEEE Transactions on Nuclear Science*, vol. 53, no. 1, str. 236–241, februar 2006. DOI:10.1109/TNS.2006.869820.
- [11] Xilinx, "7 Series FPGAs Configurable Logic Block," *User Guide*, UG474 (v1.8), 27. september 2016.

- [12] C. Liu in Y. Wang, "A 128-Channel, 710 M Samples/Second, and Less Than 10 ps RMS Resolution Time-to-Digital Converter Implemented in a Kintex-7 FPGA," *IEEE Transactions on Nuclear Science*, vol. 62, no. 3, str. 773–783, junij 2015. DOI:10.1109/TNS.2015.2421319.
- [13] Y. Wang, J. Kuang, C. Liu in Q. Cao, "A 3.9-ps RMS Precision Time-to-Digital Converter Using Ones-Counter Encoding Scheme in a Kintex-7 FPGA," *IEEE Transactions on Nuclear Science*, vol. 64, no. 10, str. 2713–2718, oktober 2017. DOI:10.1109/TNS.2017.2746626.
- [14] C. Favi in E. Charbon, "A 17ps Time-to-Digital Converter Implemented in 65nm FPGA Technology," *FPGA'09 Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, februar 2009, str. 113–120. DOI:10.1145/1508128.1508145.
- [15] Xilinx, "Vivado Design Suite 7 Series FPGA and Zynq-7000 SoC Libraries Guide," UG953 (v2018.3), 5. december 2018.
- [16] Xilinx, "7 Series FPGAs Memory Resources," *User Guide*, UG473 (v1.13), 5. februar 2019.
- [17] L. H. Crockett, R. A. Elliot, M. A. Enderwitz in R. W. Stewart, *The Zynq Book: Embedded Processing with the ARM Cortex-A9 on the Xilinx Zynq-7000 All Programmable SoC*, First Edition, Strathclyde Academic Media, 2014. Na voljo na <http://www.zynqbook.com/>.
- [18] Xilinx, "AXI Block RAM (BRAM) Controller v4.0," *LogiCORE IP Product Guide*, PG078, 5. oktober 2016.
- [19] Xilinx, "AXI GPIO v2.0," *LogiCORE IP Product Guide*, PG144, 5. oktober 2016.
- [20] Xilinx, "7 Series FPGAs Clocking Resources," *User Guide*, UG472 (v1.14), 30. julij 2018.
- [21] Anton Potočnik, osebna spletna stran. <http://antonpotocnik.com/>.
- [22] M. Adamič in A. Trost, "A Fast High-Resolution Time-to-Digital Converter Implemented in a Zynq 7010 SoC," *proceedings of the 2019 Austrochip Workshop on Microelectronics, IEEE*, oktober 2019, str. 29–34. DOI:10.1109/Austrochip.2019.00017.
- [23] Xilinx, "Zynq-7000 SoC: DC and AC Switching Characteristics," *Product Specification*, DS187 (v1.20.1), 2. julij 2018.
- [24] J. Wu in Z. Shi, "The 10-ps wave union TDC: Improving FPGA TDC resolution beyond its cell delay," *2008 IEEE Nuclear Science Symposium Conference Record*, oktober 2008, str. 3440–3446. DOI:10.1109/NSSMIC.2008.4775079.
- [25] W. Pan, G. Gong in J. Li, "A 20-ps Time-to-Digital Converter (TDC) Implemented in Field-Programmable Gate Array (FPGA) with Automatic Temperature Correction," *IEEE Transactions on Nuclear Science*, vol. 61, no. 3, str. 1468–1473, junij 2014. DOI:10.1109/TNS.2014.2320325.

- [26] Q. Shen et al., “A 1.7 ps Equivalent Bin Size and 4.2 ps RMS FPGA TDC Based on Multichain Measurements Averaging Method,” *IEEE Transactions on Nuclear Science*, vol. 62, no. 3, str. 947–954, junij 2015. DOI:10.1109/TNS.2015.2426214.
- [27] Keysight, “Keysight 33220A 20 MHz Function/Arbitrary Waveform Generator,” *Data Sheet*, 21. maj 2015. Dostopno na <https://www.keysight.com/us/en/assets/7018-01144/data-sheets/5988-8544.pdf>.
- [28] P. Eckert, H.-C. Schultz-Coulon, W. Shen, R. Stamen in A. Tadday, “Characterisation Studies of Silicon Photomultipliers,” *Nucl. Instrum. Methods Phys. Res. A*, 620(2-3), str. 217–226, avgust 2010. DOI:10.1016/j.nima.2010.03.169.