Univerza v Ljubljani

Fakulteta za elektrotehniko

Domen Cvenkel

Prototip scintilacijskega detektorja za merjenje uhajanja pri izolirani perfuziji udov

Magistrsko delo

Magistrski študijski program druge stopnje Elektrotehnika

Mentor: izr. prof dr. Andrej Trost

Somentor: dr. Rok Pestotnik

Ljubljana, 2019

Zahvala

Zahvaljujem se mentorju, izr. prof. dr. Andreju Trostu, in somentorju dr. Roku Pestotniku za pomoč, usmerjanje in predloge pri izdelavi magistrske naloge. Zahvala gre tudi gospodu Eriku Marganu z inštituta Jožef Stefan, doc. dr. Marku Jankovcu s Fakultete za elektrotehniko v Ljubljani ter Petru Kržiču iz podjetja Hella Saturnus za pomoč pri izdelavi analognega dela spektrometra.

Zahvaljujem se svoji družini za vso podporo in potrpljnenje v času študija ter jim posvečam besede angleškega fizika Isaaca Newtona: "If I have seen further it is by standing on the shoulders of giants."

Povzetek

Pri rakavih bolnikih z melanomom ali sarkomom, omejenim na en ud, se kot alternativa amputaciji ali drugim operacijam izvaja kirurški poseg, imenovan izolirana perfuzija udov. Pacientovo okončino pri tem priklopijo na izoliran krvni obtok in vanj dodajo tekočino, ki med drugim vsebuje radioaktivno snov ter visoke odmerke zdravil. Med posegom je potrebno ves čas spremljati, da uhajanje tekočine iz izolirane okončine v preostali del telesa ne preseže dovoljene meje, saj bi v nasprotnem primeru to predstavljajo nevarnost za pacienta. V magistrski nalogi je predstavljen prototip prenosnega scintilacijskega detektorja za merjenje radioaktivnosti, ki se uporablja za zaznavanje uhajanja. Detektor je v grobem sestavljen iz scintilatorja, fotopomnožvalke, večkanalnega analizatorja in grafičnega uporabniškega vmesnika. Kot večkanalni analizator smo uporabili FPGA vezje na razvojni plošči Red Pitaya. Motiv za delo je sestaviti detektor, ki je majhen in lahko prenosen in bi v prihodnosti lahko zamenjal obstoječ detektor na Onkološki kliniki UKC Ljubljana.

Ključne besede: izolirana perfuzija udov, radioaktivnost, uhajanje, scintilator, fotopomnoževalka, FPGA, Red Pitaya

Abstract

Isolated limb perfusion is a surgical procedure used for treating cancer patients with melanoma or sarcoma isolated to a single limb. It is used as an alternative to limb amputation or other mutilating operations. Patient's limb is disconnected from systemic bloodstream and connected to its own isolated one in which radioactive tracer and high dosage of medication are injected. During procedure, constant monitoring of fluid leakage from isolated limb to systemic bloodstream is important for patient's safety. In case leakage exceeds safe threshold, the procedure is stopped. Thesis addresses a portable prototype scintillation detector for measuring radioactivity which indirectly measures leakage during isolated limb perfusion. Detector is based on scintillator, photomultiplier, multichannel analyzer and graphic user interface. Multichannel analyzer is made with FPGA circuit on Red Pitaya development board. The main motive for work was to build a small and portable detector which could in future replace the one that is currently used at the Institute of Oncology in Ljubljana.

Key words: isolated limb perfusion, radioactivity, leakage, scintillator, photomultiplier, Red Pitaya, FPGA

Vsebina

Seznam slik

Seznam tabel

Seznam uporabljenih simbolov

V magistrski nalogi so uporabljene sledeče veličine in simboli:

Veličina / oznaka		**Enota**	
Ime	Simbol	Ime	Simbol
čas	t	sekunda	s
napetost	V	Volt	V
tok	I	Amper	A
debelina	d	meter	m
aktivnost	A	Bekerel	Bq
energija	E	elektronvolt	eV
atenuacijski koeficient	μ	-	m^{-1}
atenuacija	I	-	eV/m^2t
razpadna hitrost	$rate$	-	razpad/s

Seznam uporabljenih kratic

UKC	Univerzitetni klinični center
UL	Univerza v Ljubljani
AD	analogno-digitalni (pretvornik)
RAM	bralno-pisalni pomnilnik (*ang.* Random Access Memory)
BRAM	blokovni bralno-pisalni pomnilnik (*ang.* Block RAM)
DP-BRAM	dvovhodni BRAM (ang. Dual Port Block RAM)
FPGA	programirljivo vezje (*ang.* Field Programmable Gate Array)
USB	univerzalno serijsko vodilov (ang. Universal Serial Bus)
AXI	komunikacijski vmesnik (*ang.* Advanced eXtensible Interface)
LUT	vpogledna tabela (*ang.* Look-Up Table)
FF	flip-flop
DSP	digitalno procesiranje signalov (*ang.* Digital Signal Processing)
PE	procesni element (*ang.* Processing Element)

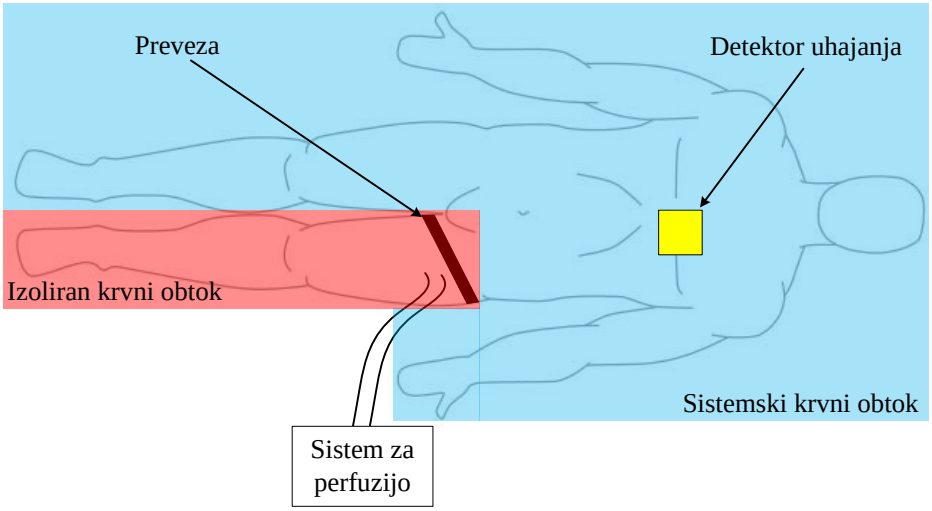
1 Uvod

1.1 Izolirana perfuzija udov

Izolirana perfuzija udov je postopek za zdravljenje rakavih obolenj, natančneje melanoma in sarkomov mehkih tkiv [1]. Pri tem mora biti rakavo obolenje izolirano na en ud. Uporablja se kot alternativa amputaciji okončine. Zaradi omejenosti posega na en ud, so lahko odmerki uporabljenih zdravil veliko višji. Pri posegu je potrebno ves čas spremljati, da zdravilo iz okončine ne uide v ostali del telesa.

Postopek zdravljenja obsega kirurški poseg, ki je prikazan na sliki 1.1 [2]. Rakasto okončino pacienta priklopijo na izoliran krvni obtok in pri tem prekinejo stik glavnih žil s sistemskim obtokom. Stik stranskih žil s sistemskim obtokom se prepreči s prevezo, ki stisne okončino. Zatem v sistemski krvni obtok vbrizgajo pacientovo kri, ki vsebuje majhno dozo radioaktivnega tehnecija Th-99, ki je sevalec gama. Z detektorjem sevanja, postavljenim nad srce, spremljajo razpadno hitrost tehnecija (poglavje 2.3.3), ki se ustali po nekaj minutah. Ustaljena razpadna hitrost predstavlja referenčno vrednost, glede na katero se med posegom meri uhajanje (*ang.* leakage).

V naslednjem koraku v izoliran krvni obtok vbrizgajo desetkratno dozo tehnecija v primerjavi z dozo, vbrizgano v sistemski krvni obtok. Za tem na detektorju opazujejo razpadno hitrost. Če pride do povečanja hitrosti, je to znak, da prihaja do uhajanja tehnecija v sistemski krvni obtok. V kolikor ni uhajanja, v izoliran krvni obtok vnesejo visoko koncentracijo zdravil. Uhajanje se z detektorjem spremlja med celotnim trajanjem posega. Če preseže dovoljeno mejo, se poseg prekine.



Slika 1.1: Izolirana perfuzija noge.

1.2 Namen in cilji naloge

Motiv za delo je izdelava majhnega in prenosnega prototipa scintilacijskega detektorja, ki bi sčasoma lahko nadomestil obstoječi detektor na Onkološkem inštitutu UKC v Ljubljani. Detektor, ki je trenutno v uporabi, je namreč okoren za uporabo. Cilji magistrske naloge zajemajo izdelavo prenosnega spektrometra iz scintilacijskega detektorja, napajalnika in večkanalnega analizatorja ter izdelavo grafičnega uporabniškega vmesnika, ki omogoča nastavitev spektrometra in časovno odvisno spremljanje uhajanja radioaktivne snovi med izolirano perfuzijo udov.

2 Teoretično ozadje

2.1 Interakcija žarkov gama s snovjo

Sevanje gama nastane, ko atomsko jedro določenega izotopa preide iz vzbujenega v osnovno energijsko stanje. Pri prehodu jedro izseva foton z visoko energijo, ki s snovjo interagira na tri različne načine. Načini interakcije so fotoefekt, Comptonovo sipanje in tvorba parov [3] [4] [5]. V našem primeru je snov, v katero vpadajo žarki gama, scintilacijski kristal, opisan v poglavju 2.2.1.

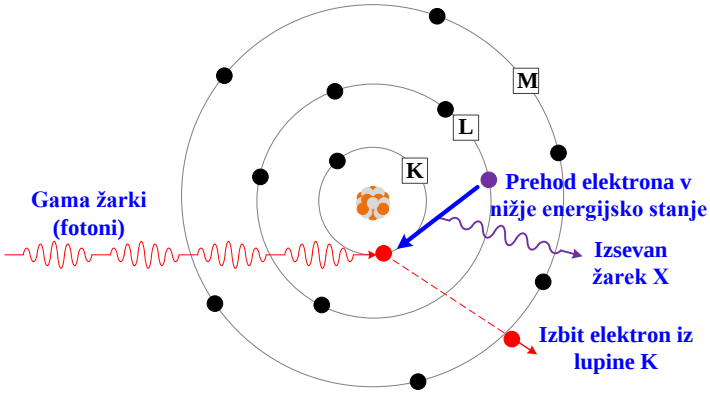
2.1.1 Fotoefekt

Pri fotoefektu, ki ga prikazuje slika 2.1, žarek gama interagira z vezanim elektronom v atomu snovi. Če je energija žarka večja od vezavne energije elektrona, žarek gama elektron izbije iz atoma. Izbit elektron ima kinetično energijo

$$E_{e^-} = E_\gamma - E_b \tag{2.1}$$

kjer sta E_γ energija žarka gama in E_b vezavna energija elektrona. Na mestu izbitega elektrona ostane vrzel, ki jo zapolni elektron iz višje lupine ter pri tem izseva karakterističen žarek X. Tudi žarek X lahko v snovi doživi fotoefekt na elektronih in posledično iz vpadnega žarka gama lahko nastaneta dva elektrona. Njuna skupna energija je približno enaka energiji žarka gama. Nastanek obeh elektronov se s strani detektorja zgodi istočasno, zato oba prispevata k zaznanemu sunku na detektorju. Pri meritvi odložene energije žarka gama v scintilatorju tako v energijskem spektru zaznamo vrh, imenovan fotovrh ali vrh popolne absorpcije. Njegova lega določa energijo žarkov gama. V primeru, da izsevani žarek X ne

doživi fotoefekta in pobegne iz snovi (scintilatorja), dobimo pri meritvi poleg fotovrha tudi energijski vrh pri energiji $E_\gamma - E_b$, ki se imenuje vrh pobega (ang. *escape peak*). Vrh pobega pri meritvi s scintilacijskim detektorjem opazimo samo pri nizkih energijah žarkov gama, saj se pri visokih fotovrh in vrh pobega zaradi ločljivosti spektrometra zlijeta skupaj.



Slika 2.1: Fotoefekt.

Verjetnost, da žarek gama v snovi interagira s fotoefektom, je podana z enačbo 2.2, kjer je Z vrstno število atoma in E_γ energija vpadnega žarka gama.

$$P \propto Z^5 E_\gamma^{-3,5} \tag{2.2}$$

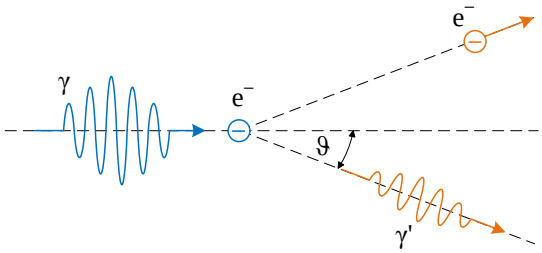
Iz enačbe je razvidno, da je verjetnost za fotoefekt močno odvisna od števila Z. Poleg tega je mogoče razbrati, da verjetnost za fotefekt pada z energijo žarkov gama.

2.1.2 Comptonovo sipanje

Comptonovo sipanje je pojav pri katerem se foton neelastično siplje na prostem elektronu. Pojav prikazuje slika 2.2. Vpadnemu fotonu se pri sipanju spremenita kinetična energija in smer gibanja. Energija fotona po sipanju znaša

$$E_{\gamma'} = \frac{E_\gamma}{1 + \frac{E_\gamma}{m_e c^2}(1 - \cos\vartheta)} \tag{2.3}$$

kjer je E_γ energija vpadnega fotona, ϑ kot, za katerega se fotonu spremeni smer gibanja in $m_e c^2$ mirovna energija elektrona.



Slika 2.2: Comptonovo sipanje.

Pri pojavu se ohranjata energija in gibalna količina sistema, zato razliko energij vpadnega in sipanega fotona odnese elektron, kar je razvidno iz enačbe 2.4.

$$E_{e^-} = E_\gamma - E_{\gamma'} \tag{2.4}$$

Energija elektrona je največja, kadar se foton siplje nazaj pod kotom 180°, elektron pa odleti v smeri vpadnega fotona. To prikazuje enačba 2.5.

$$E_{e^-\max} = \frac{2E_\gamma^2}{2E_\gamma + m_e c^2} \tag{2.5}$$

Maksimalna energija elektrona se imenuje tudi energija Comptonovega roba, ker pri meritvi v spektru nastane značilen energijski rob. Vpadni fotoni se sicer na elektronih sipajo pod različnimi koti, kar pomeni, da imajo elektroni različne kinetične energije. Energijski spekter sipanih elektronov je zato zvezen.

Sipani fotoni lahko znova interagirajo ali pobegnejo iz snovi. Če interagirajo s fotoefektom, se absorbirajo v scintilatorju. Na ta način žarek gama odda vso svojo energijo in v energijskem spektru ga zaznamo v fotovrhu. V primeru, da sipani foton ponovno interagira s Comptonovim sipanjem, ali pobegne iz scintilatorja, elektronom odda le del energije. Posledično ga zaznamo v zveznem Comptonovem spektru.

Vpadni fotoni se ne sipajo le v scintilatorju, temveč tudi v snovi, ki obdaja detektor. Ta snov je lahko ščit detektorja ali steklo, ki prekriva fotopomnoževako. Tudi ti sipani fotoni se lahko absorbirajo v scintilatorju, kar v spektru zaznamo v energijskem vrhu, ki se imenuje vrh povratnega sipanja.

Verjetnost za Comptonovo sipanje opisuje enačba 2.6.

$$P \propto Z E_\gamma^{-0,5} \tag{2.6}$$

Pri energijah, kjer fotoefekt izgubi dominantno vlogo interakcije žarkov s snovjo, to vlogo prevzame Comptonovo sipanje. Z večanjem energije žarkov gama postane tudi ta način interakcije zanemarljiv v primerjavi s tvorbo parov, ki prevzame glavno vlogo pri visokih energijah.

2.1.3 Tvorba parov

Tvorba parov je pojav, pri katerem se energija direktno pretvori v snov in obratno. Pojav je prikazan na sliki 2.3. Kadar ima vpadni žarek gama energijo, ki je vsaj dvakrat večja od mirovne energije elektrona $m_e c^2$, se lahko v bližini jedra atoma spremeni v par pozitron–elektron, pri čemer jedro prevzame ostanek gibalne količine. Za nastanek obeh delcev se porabi energija $2m_e c^2$, ostala energija vpadnega žarka gama pa predstavlja skupno kinetično energijo pozitrona in elektrona.

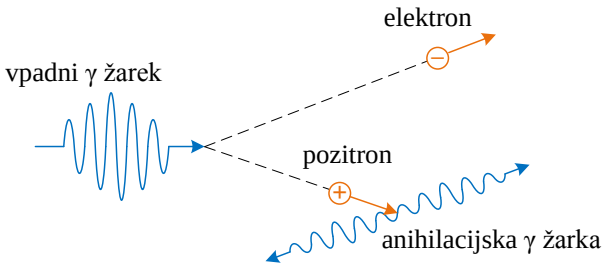
$$E_{e^+} + E_{e^-} = E_\gamma - 2m_e c^2 \tag{2.7}$$

Nastala delca se v scintilatorju upočasnita in mu oddata kinetično energijo. Ker je pozitron antidelec elektrona, ob zaustavitvi interagira z njim. Pri tem pride do uničenja (anihilacije) obeh delcev in nastanka dveh anihilacijskih žarkov gama z energijama $m_e c^2$, ki pod kotom 180° odletita vsak v svojo smer. Anihilacijska žarka gama se lahko v nadaljevanju absorbirata v scintilatorju, ali pobegneta iz njega. Če se absorbirata oba žarka, je celotna oddana energija v scintilatorju enaka energiji vpadnega žarka gama E_γ, zato v spektru to zaznamo v fotovrhu. V primeru, da en anihilacijski žarek pobegne iz scintilatorja, v spektru nastane vrh pri energiji $E_\gamma - m_e c^2$, ki se imenuje vrh enojnega pobega. Zadnja možnost je, da iz scintilatorja pobegneta oba anihilacijska žarka gama. V tem primeru v spektru nastane vrh pri energiji $E_\gamma - 2m_e c^2$, imenovan vrh dvojnega pobega.

Verjetnost za tvorbo parov je približno podana z enačbo 2.8. Verjetnost se z energijo veča, zato je tvorba parov primarni način interakcije žarkov gama s snovjo pri visokih energijah.

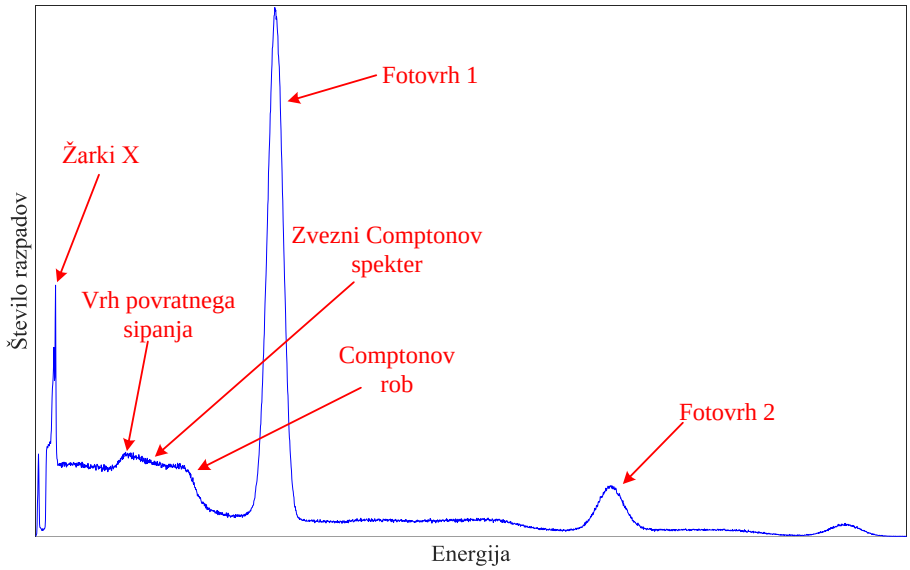
$$P \propto Z^2 E_\gamma \tag{2.8}$$

Slika 2.4 prikazuje energijski spekter radioaktivnega izotopa natrija Na-22, ki nastane kot posledica vseh treh interakcij žarkov gama s snovjo. Natrij seva žarke



Slika 2.3: Tvorba parov.

gama z dvema značilnima energijama, zato sta v spektru vidna dva fotovrha.



Slika 2.4: Energijski spekter Na-22.

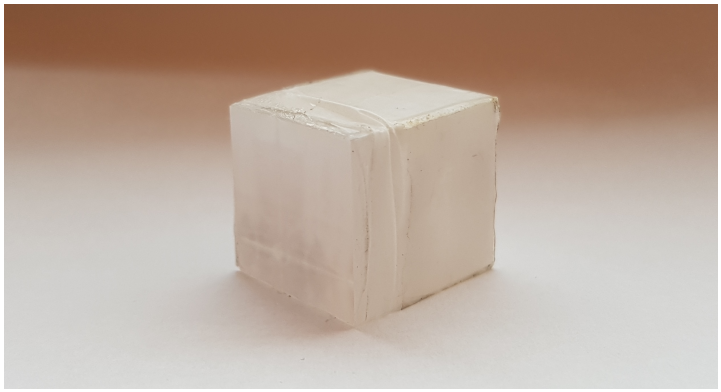
2.2 Detekcija žarkov gama

2.2.1 Zgradba detektorja

Scintilacijski detektor, uporabljen za zaznavanje žarkov gama, sestavljata scintilacijski kristal (scintilator) in fotopomnoževalka.

Scintilator je snov, ki izseva fotone v vidnem spektru svetlobe, ko je izpo-

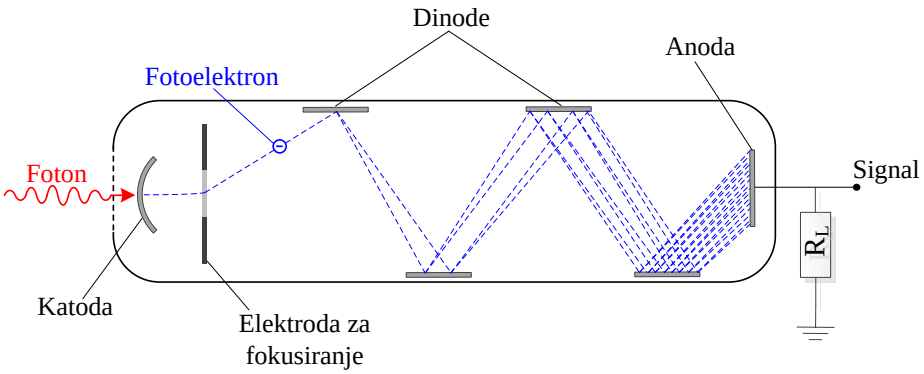
stavljena sevanju. V našem primeru smo uporabili scintilacijski kristal LYSO, prikazan na sliki 2.5. To je lutecij-itrijev oksi ortosilikat z dopiranim cerijem kot nečistočo in kemijsko formulo Ce:$(Lu,Y)_2SiO_5$. LYSO spada med anorganske scintilatorje. Izbrali smo ga zaradi kratkega razpadnega časa, po katerem se izseva foton (45 ns) in dobrega svetlobnega izkoristka (33 fotonov/keV) [6]. Poleg tega kristal izseva največ fotonov z valovno dolžino okoli 420 nm, kar ustreza območju maksimalnega odziva fotopomnoževalke. Kristal ima dimenzije 18×18×20 mm in lomni količnik 1,81.



Slika 2.5: Scintilacijski kristal LYSO.

Posledica interakcije žarkov gama s snovjo so nabiti delci, ki potujejo skozi kristal in ionizirajo snov. Pri anorganskih scintilatorjih nabiti delci svojo energijo oddajo elektronom v atomih kristala. V primeru, da je energija dovolj velika, elektron lahko preskoči iz valenčnega v prevodni pas in za sabo pusti prazno mesto, imenovano vrzel. Pari elektron-vrzel se širijo skozi kristalno rešetko, dokler ne naletijo na atom nečistoče. Pri tem vrzel vzame elektron atomu nečistoče, iz katerega nastane ion. Preostali elektron iz para elektron-vrzel se zatem rekombinira z ionom nečistoče in pri tem izseva foton. Valovna dolžina izsevanega fotona je odvisna od tipa nečistoče. Nečistoča je praviloma izbrana tako, da izsevana valovna dolžina svetlobe ustreza območju v katerem ima fotopomnoževalka največji odziv. Število fotonov je odvisno od energije vpadnega žarka gama.

Izsevane fotone zazna fotopomnoževalka, naprava, ki svetlobni signal pretvori v električnega. Shematsko jo prikazuje slika 2.6. Sestavljena je iz steklene vakuumske cevi, v kateri se nahajajo katoda, anoda in dinode. Foton skozi vhodno okence pade na katodo, kjer s fotoefektom iz nje izbije fotoelektron, imenovan



Slika 2.6: Shema fotopomnoževalke.

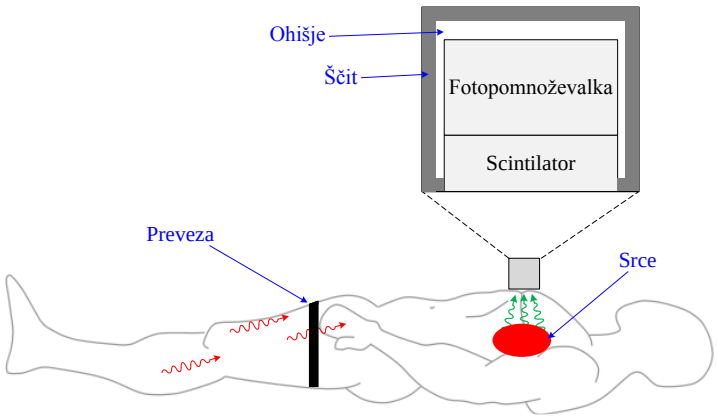
tudi primarni elektron. Fotoelektron se na elektrodi za fokusiranje usmeri na množilnik elektronov, ki ga sestavljajo dinode. Pri vpadu na prvo dinodo ima dovolj veliko kinetično energijo, da iz nje izbije več sekundarnih elektronov. Med dinodami je visoka razlika potencialov, pri tem pa je vsaka naslednja dinoda na višjem potencialu kot prejšnja. Posledično se sekundarni elektroni, izbiti iz prve dinode, v električnem polju pospešijo do druge dinode, kjer izbijejo nove elektrone. Proces se nadaljuje čez vse dinode, elektroni pa se na koncu zberejo na anodi, kjer, kot posledico vpadnega fotona, zaznamo tokovni sunek. Tokovni sunek v napetostni impulz pretvori upor R_L, vezan med anodo in ozemljitev. Ker je število vpadnih fotonov v fotopomnoževalko odvisno od energije žarka gama, je od energije odvisna tudi višina napetostnega impulza na anodi.

Na stiku scintilatorja in fotopomnoževalke je uporabljena optična mast, ki zmanjša izgube pri prehodu svetlobe iz kristala v vhodno okence fotopomnoževalke. Poleg tega kristal ovijemo z belo teflonsko folijo, ki fotone odbije proti fotopomnoževalki.

2.2.2 Ščitenje detektorja

Pri izolirani perfuziji udov detektor postavimo nad območje, kjer se nahaja srce pacienta, kar prikazuje slika 2.7 [2]. Zaznati želimo sevanje gama kot posledico sevalca in zdravila, ki sta iz izoliranega krvnega obtoka ušla v sistemski krvni obtok in dosegla srce. V ta namen je potrebno detektor zaščititi pred vsemi žarki gama, ki na scintilator ne vpadajo direktno, temveč iz izolirane okončine

prihajajo s strani. Na sliki so označeni z rdečo barvo, medtem ko so žarki gama, ki v detektor vpadajo direktno, označeni z zeleno barvo.



Slika 2.7: Ščitenje detektorja pred žarki gama s strani.

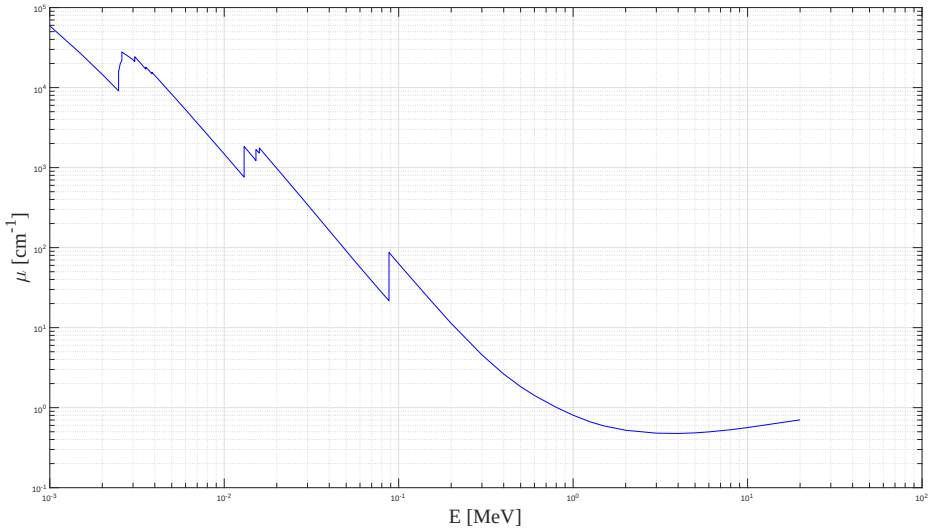
Pri ščitenju detektorja je ključna izbira materiala, ki zavira sevanje. Pomembne so tri lastnosti materiala: gostota, vrstno število in debelina [7]. Za zaustavitev žarkov gama z določeno energijo, višja gostota in vrstno število materiala pomenita manjšo debelino in obratno. V našem primeru smo za ščit izbrali svinec, ki ima visoko gostoto 11340 kg/m3 in visoko vrstno število 82.

2.2.2.1 Slabljenje sevanja gama

Slabljenje (ang. *attenuation*) sevanja gama v materialu opisuje enačba 2.9 [7]. Pri tem je I intenziteta sevanja po slabljenju, I_0 intenziteta pred slabljenjem, μ linearni atenuacijski koeficient in d debelina ščita. Intenziteta sevanja je definirana kot energija vpadnega sevanja na določeno površino v nekem času.

$$I(d) = I_0 e^{-\mu d} \tag{2.9}$$

Linearni atenuacijski koeficient se spreminja z energijo sevanja gama in je prikazan na sliki 2.8 [8].

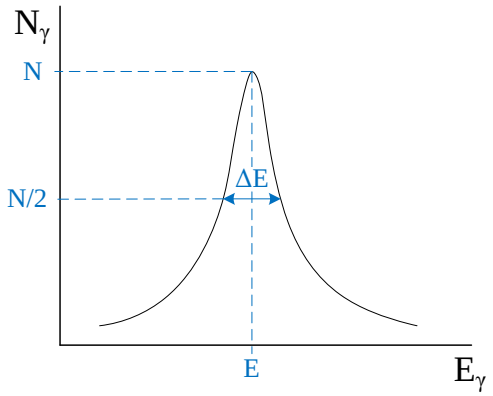


Slika 2.8: Odvisnost linearnega atenuacijskega koeficienta svinca od energije žarka gama.

Kot je opisano v poglavju 1.1, se pri izolirani perfuziji udov kot sevalec uporablja tehnecij Th-99. Pri radioaktivnem razpadu tehnecij seva žarke gama z energijama 140,5 keV in 142,6 keV. Za oceno debeline ščita lahko energijsko mejo z nekaj rezerve postavimo na 200 keV. V našem primeru zadostuje, da intenziteto žarka gama z mejno energijo oslabimo za faktor 50. Ker se površina ščita ne spreminja, je intenziteta glede na definicijo proporcionalna energiji. To pomeni, da bo imel žarek gama z mejno energijo 200 keV po prehodu skozi ščit energijo 4 keV, kar je pri območju med 0 keV in 200 keV pod šumno mejo našega detektorja in ga pri meritvi ne upoštevamo. Debelino ščita izračunamo tako, da iz 2.9 izpostavimo d, pri tem pa upoštevamo izbran faktor slabljenja in atenuacijski koeficient svinca pri mejni energiji 200 keV, ki znaša 11,32 cm-1. Potrebna debelina svinčenega ščita je tako 3,5 mm.

2.2.3 Ločljivost spektrometra

Ločljivost spektrometra je definirana kot količnik širine energijskega vrha na polovični višini in energije vrha, kot je prikazano na sliki 2.9. Podana je z enačbo 2.10.



Slika 2.9: Definicija ločljivosti.

$$R = \frac{\Delta E}{E} \tag{2.10}$$

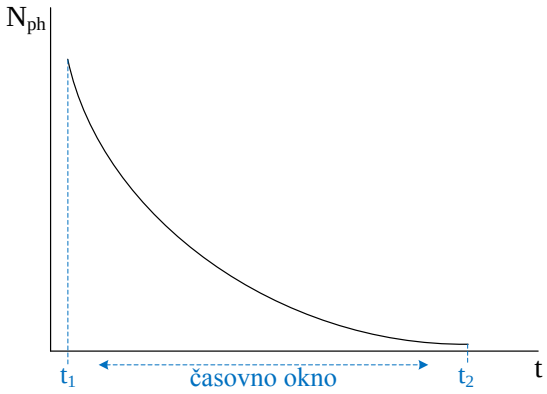
2.3 Obdelava signalov

Kot je opisano v poglavju 2.2.1, po prihodu žarka gama na anodi fotopomnoževalke zaznamo tokovne sunke, ki jih pretvorimo v napetostne impulze. Impulzi predstavljajo izsevane fotone iz scintilatorja, ki vpadejo na fotopomnoževalko. Posamezen vpadni žarek gama povzroči izsev večjega števila fotonov iz scintilatorja. Izsev vseh fotonov se ne zgodi hkrati, zato na izhodu fotopomnoževalke zaznamo več napetostnih impulzov. Časovni potek števila izsevanih fotonov $N_{\mathrm{ph}}(\mathrm{t})$ prikazuje slika 2.10. Pretečen čas od začetka odziva na vpad žarka gama (t_1) do časa, ko odziv izzveni (t_2), imenujemo časovno okno.

Skupen naboj, ki se na anodi fotopomnoževalke nabere v časovnem oknu, je sorazmeren energiji žarka gama. To pomeni, da so energiji žarka sorazmerni tudi tokovni sunki in posledično zaznani napetostni impulzi znotraj časovnega okna. Matematično to prikazujeta enačbi 2.11 in 2.12.

$$e_{\mathrm{FP}} \propto E_\gamma \tag{2.11}$$

$$e_{\mathrm{FP}} = \int_{t_1}^{t_2} I_{\mathrm{FP}} dt = \int_{t_1}^{t_2} \frac{V_{\mathrm{FP}}}{R_{\mathrm{L}}} dt \propto E_\gamma \tag{2.12}$$



Slika 2.10: Časovna odvisnost števila izsevanih fotonov.

e_{FP} – skupen naboj na anodi fotopomnoževalke znotraj časovnega okna
E_γ – energija žarka gama
I_{FP} – tokovni sunek
V_{FP} – napetostni impulz
R_{L} – bremenski upor

Upor R_{L} v enačbi 2.12 je konstanten, zato je energija žarka gama sorazmerna integralu napetostnih impulzov znotraj časovnega okna, kar lahko zapišemo kot:

$$\int_{t_1}^{t_2} V_{\mathrm{FP}} dt = V_\gamma \tag{2.13}$$

$$V_\gamma \propto E_\gamma \tag{2.14}$$

V_γ je analogna napetost na izhodu električnega vezja, ki integrira napetostne impulze iz fotopomnoževalke.

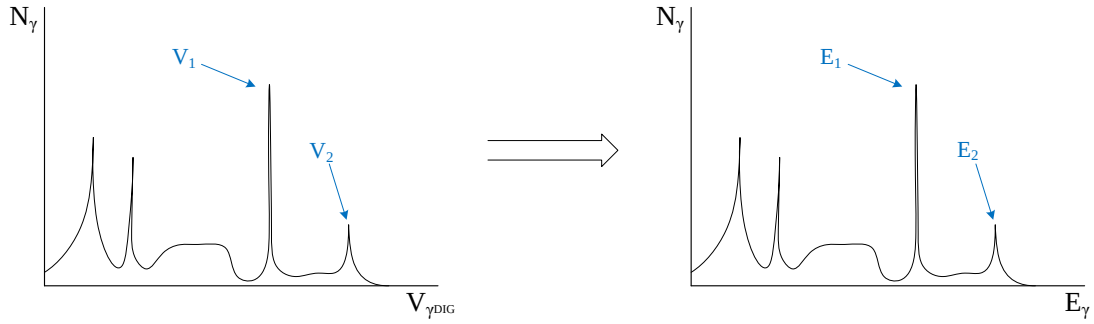
2.3.1 Energijski spekter

Z integracijo napetostnih impulzov iz fotopomnoževalke dobimo analogno napetost V_γ, ki je sorazmerna energiji vpadnega žarka gama E_γ. V našem primeru nas ne zanima le energija posameznega žarka, temveč porazdelitev števila zaznanih žarkov gama N_γ po energiji, oziroma energijski spekter sevanja gama $N_\gamma(E_\gamma)$, ki ga predstavimo s histogramom.

Energijski spekter dobimo v dveh korakih. V prvem koraku z meritvijo dobimo histogram $N_\gamma(V_{\gamma\mathrm{DIG}})$, kjer je $V_{\gamma\mathrm{DIG}}$ digitalni ekvivalent analogne napetosti V_γ po vzorčenju z AD pretvornikom ter predstavlja indeks stolpca histograma. V drugem koraku kalibriramo merilni sistem in tako iz histograma $N_\gamma(V_{\gamma\mathrm{DIG}})$ dobimo histogram $N_\gamma(E_\gamma)$, ki predstavlja energijski spekter sevanja.

2.3.2 Kalibracija spektrometra

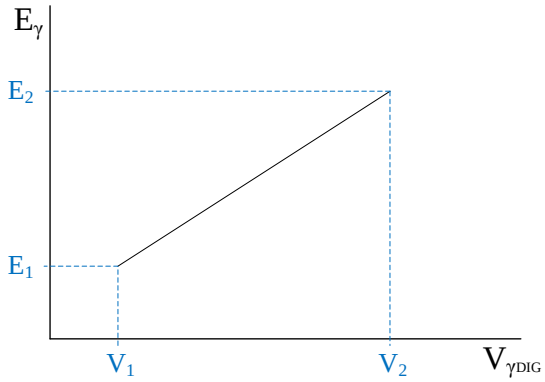
Kalibracijo spektrometra izvedemo s sevalci, ki imajo v energijskem spektru znane energije vrhov. V našem primeru smo za kalibracijo izbrali radioaktiven izotop natrija Na-22, ki ima v spektru značilna dva energijska vrhova. Postopek kalibracije prikazuje slika 2.11, na kateri je približno skiciran energijski spekter natrijevega izotopa.



Slika 2.11: Histogram, pridobljen z meritvijo (levo) in histogram, ki je rezultat kalibracije (desno).

Na levi strani slike se nahaja histogram $N_\gamma(V_{\gamma\mathrm{DIG}})$, ki je pridobljen v postopku meritve, na desni strani pa histogram $N_\gamma(E_\gamma)$, ki ga želimo. Umeriti je potrebno abscisno os. Za umeritev lahko izberemo znani energiji natrijevih fotovrhov z energijo E_1 in E_2. Ker sta poznana tako indeksa stolpcev obeh vrhov V_1 in V_2 kot tudi obe pripadajoči energiji E_1 in E_2, je mogoče narisati umeritveno krivuljo, ki jo prikazuje slika 2.12.

Iz slike razberemo enačbo premice, ki podaja relacijo med $V_{\gamma\mathrm{DIG}}$ in E_γ. Z



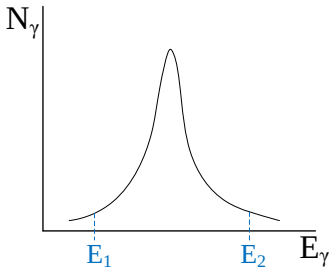
Slika 2.12: Umeritvena krivulja.

dobljeno enačbo 2.15 umerimo abscisno os.

$$E_\gamma = \frac{E_2 - E_1}{V_2 - V_1} \cdot V_{\gamma\mathrm{DIG}} + \left(E_1 - \frac{E_2 - E_1}{V_2 - V_1} \right) \cdot V_1 \tag{2.15}$$

2.3.3 Razpadna hitrost in uhajanje

Razpadna hitrost (*ang.* rate) je definirana kot število razpadov na časovno enoto. Določimo jo iz energijskega spektra sevalca $N_\gamma(E_\gamma)$. Razmerje med razpadno hitrostjo, izmerjeno v fotovrhu, in celotno aktivnostjo sevalca je za dano geometrijo pacient-kristal konstantno. Zaradi tega je meritev razpadne hitrosti v fotovrhu dobro merilo za celotno aktivnost. Fotovrh se v spektru nahaja med energijama E_1 in E_2, kar prikazuje slika 2.13.



Slika 2.13: Fotovrh.

Število vseh razpadov N_D, zaznanih v fotovrhu, podaja enačba 2.16.

$$N_{\mathrm{D}} = \int_{E_1}^{E_2} N_\gamma(E_\gamma) dE_\gamma \tag{2.16}$$

Meritev razpadne hitrosti poteka periodično, kjer pretečen čas med dvema meritvama znaša Δt. Hitrost razpadanja je torej število novih razpadov, zaznanih v času med dvema meritvama, kar opisuje enačba 2.17.

$$rate(t) = \frac{N_{\mathrm{D}}(t) - N_{\mathrm{D}}(t - \Delta t)}{\Delta t} \tag{2.17}$$

Kot je opisano v poglavju 1.1, se pri izolirani perfuziji udov ves čas spremlja uhajanje sevalca in zdravil iz izolirane okončine proti srcu. Uhajanje se izračuna z enačbo 2.18 [9]. Pri tem je *rate(t)* zaznana razpadna hitrost ob času t, $rate_{\mathrm{REF}}$ referenčna razpadna hitrost, izmerjena pred začetkom posega, A_{SKO} aktivnost sevalca, vnesenega v sistemski krvni obtok, in A_{IKO} aktivnost sevalca, vnesenega v izoliran krvni obtok.

$$L[\%] = \frac{rate(t)}{rate_{\mathrm{REF}}} \cdot \frac{A_{\mathrm{SKO}}}{A_{\mathrm{IKO}}} \tag{2.18}$$

Ker ima tehnecij Th-99, ki se uporablja kot sevalec, kratek razpolovni čas, je potrebno to upoštevati v enačbi 2.18. Razpolovni čas je čas, v katerem razpade polovica jeder sevalca. Za tehnecij je ta čas 6 ur, medtem ko poseg traja od 60 do 90 minut. To pomeni, da pri meritvi razpadne hitrosti enake količine tehnecija, ob dveh različnih časih, izmerimo različne razpadne hitrosti. Enačba 2.18 zato velja samo za sevalce z zelo dolgim razpolovnim časom, v primerjavi s časom trajanja posega, ki pa se za izolirano perfuzijo udov ne uporabljajo. Med posegom je zato potrebno izmerjeno razpadno hitrost *rate(t)* ekstrapolirati v razpadno hitrost $rate_0(t)$, ki bi jo izmerili ob času t, če se razpadna hitrost enake količine tehnecija ne bi spreminjala s časom. Enačba 2.19 opisuje število razpadov v odvisnosti od časa. Pri tem je N_0 začetno število nestabilnih jeder v vzorcu, t pretekli čas razpadanja, $t_{1/2}$ razpolovni čas in *N(t)* število preostalih jeder v vzorcu.

$$N(t) = N_0 \cdot 2^{-\frac{t}{t_{1/2}}} \tag{2.19}$$

Po istem zakonu kot razpadanje jeder, se spreminja tudi razpadna hitrost, kar

prikazuje enačba 2.20.

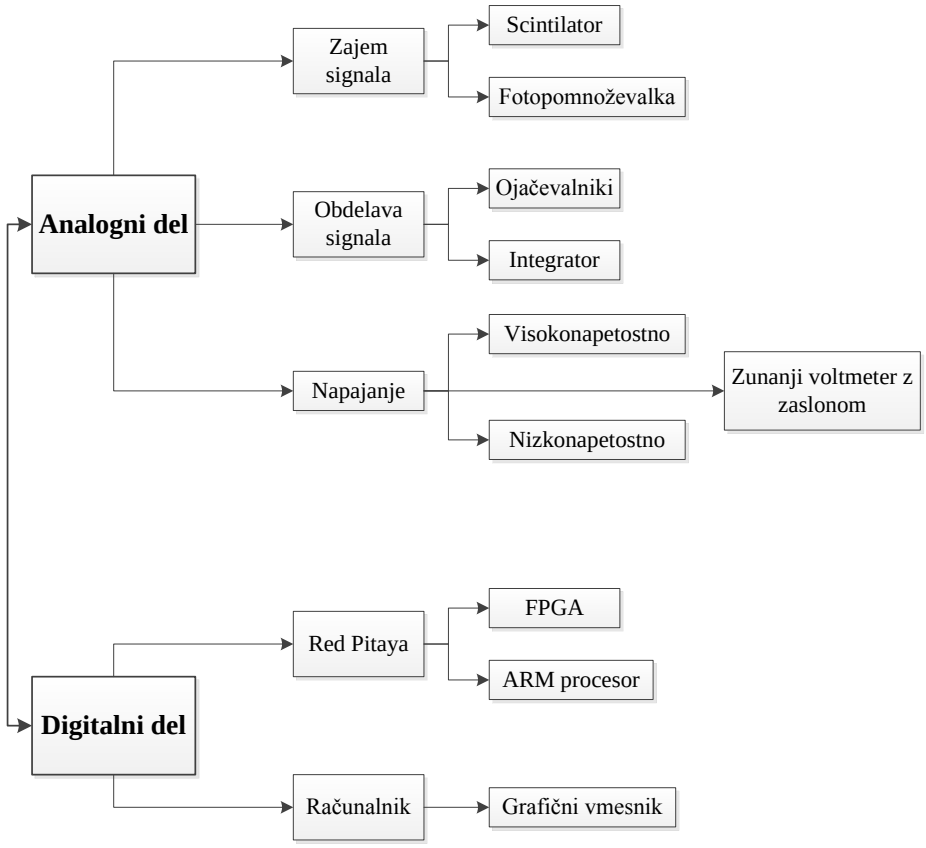
$$rate(t) = rate_0(t) \cdot 2^{-\frac{t}{t_{1/2}}} \tag{2.20}$$

Iz enačbe 2.20 izrazimo $rate_0(t)$ in ga vstavimo v enačbo 2.18. S tem dobimo enačbo za uhajanje, v kateri je upoštevan razpolovni čas tehnecija.

$$L[\%] = 2^{\frac{t}{t_{1/2}}} \cdot \frac{rate(t)}{rate_{\mathrm{REF}}} \cdot \frac{A_{\mathrm{SKO}}}{A_{\mathrm{IKO}}} \tag{2.21}$$

3 Sestavni deli spektrometra

Sestavni deli spektrometra za merjenje radioaktivne snovi, ki seva sevalce gama, so predstavljeni na sliki 3.1. V splošnem sistem razdelimo na analogni in digitalni del.



Slika 3.1: Zgradba spektrometra.

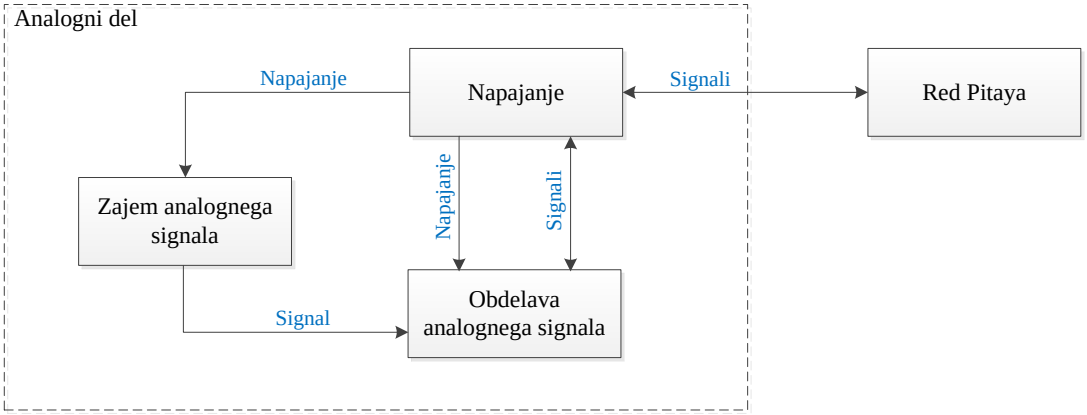
Analogni del sistema opravlja zajem in pretvorbo sevanja gama v električne

signale ter njihovo analogno obdelavo. Žarek gama v električni signal pretvorita scintilacijski kristal in fotopomnoževalka. Dobljeni signal v obliki impulzov se nato ojači in integrira, zatem pa ga prebere analogno-digitalni pretvornik na Red Pitayi. Napajanje analognega dela razdelimo na visokonapetostno, ki napaja fotopomnoževalko, in nizkonapetostno, ki napaja vse ostale elemente analognega dela. Za spremljanje in nastavitev napetosti na fotopomnoževalki sta dodana zunanji voltmeter z vgrajenim analogno-digitalnim pretvornikom in zaslonom ter rotacijski potenciometer.

Digitalni del sestavljata Red Pitaya in osebni računalnik. Red Pitaya je razvojna plošča, katere centralni del predstavlja sistem na čipu (ang. *System On Chip*) Zynq 7010. Zynq združuje FPGA vezje in vgrajeni dvojedrni ARM procesor, na katerem teče operacijski sistem Linux. Periferne enote na Red Pitayi, ki smo jih uporabili za realizacijo spektrometra, vključujejo dva 14-bitna analogno-digitalna pretvornika z nastavljivim merilnim območjem in sposobnostjo zajemanja podatkov s histostjo 125 Msps, večnamenske digitalne vhodno-izhodne priključke in Ethernet priključek za komunikacijo z računalnikom. Naloga Red Pitaye je pretvorba analognih signalov v digitalne, njihova digitalna obdelava ter posredovanje v računalnik. Obdelava signalov je izvedena na FPGA delu, komunikacija z računalnikom pa na procesorskem delu sistema Zynq. Na računalniku teče grafični vmesnik, kjer poteka prikaz meritev, hkrati pa uporabniku omogoča nastavitev spektrometra.

4 Analogni del

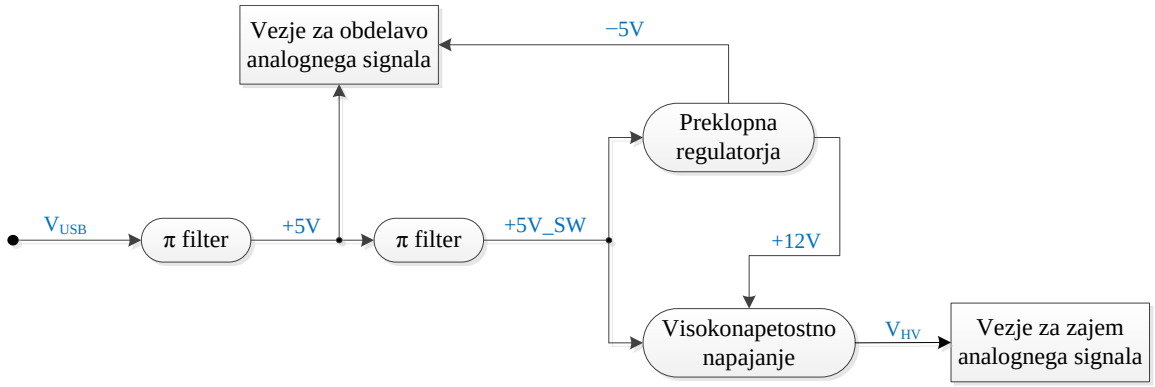
Analogni del je namenjen zajemu in obdelavi analognega signala. Prikazan je na sliki 4.1. Sestavljen je iz treh sklopov; vsak se nahaja na svojem tiskanem vezju. Signal, ki je posledica zaznanega žarka gama, tvorita scintilator in fotopomnoževalka na vezju za zajem analognega signala. Zatem signal potuje do vezja za obdelavo analognega signala, kjer se ojači in integrira, nato pa prek priključkov na napajalnem vezju do prvega AD pretvornika Red Pitaye. Hkrati do drugega AD pretvornika Red Pitaye vzporedno z integriranim signalom potuje tudi ojačen signal iz fotopomnoževalke, ki služi kot prožilec (ang. *trigger*). Prožilec FPGA del Red Pitaye obvešča o zaznavi novega žarka gama. Obratno je iz Red Pitaye prek napajalnega vezja na vezje za obdelavo analognega signala priključen signal za nadzor integratorja. Izdelave analognega dela sistema se nismo lotili od začetka, temveč smo vzeli nekatere že narejene dele in jih vključili v svoj izdelek.



Slika 4.1: Zgradba analognega dela spektrometra.

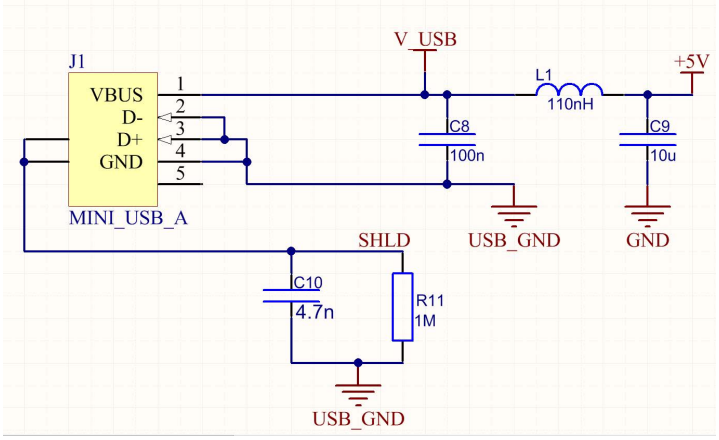
4.1 Napajanje

Različni sklopi analognega dela sistema za delovanje potrebujejo različne napajalne napetosti. Shemo napajanja prikazuje slika 4.2. Celoten analogni del je prek USB kabla napajan s petimi volti. Vezje za obdelavo analognega signala vsebuje ojačevalnike, ki potrebujejo simetrično napajanje ±5 V. Napetost +5 V dobimo iz USB vhoda, ki ga filtriramo, da preprečimo širjenje motenj. Napetost −5 V generira prvi od dveh preklopnih regulatorjev. Visokonapetostni (VN) del potrebuje napetosti +5 V in +12 V. Slednjo generira drugi preklopni regulator. Izhod iz VN dela je nastavljiva napetost V_{HV}, ki je napajalna napetost vezja za zajem analognega signala.



Slika 4.2: Shema napajanja.

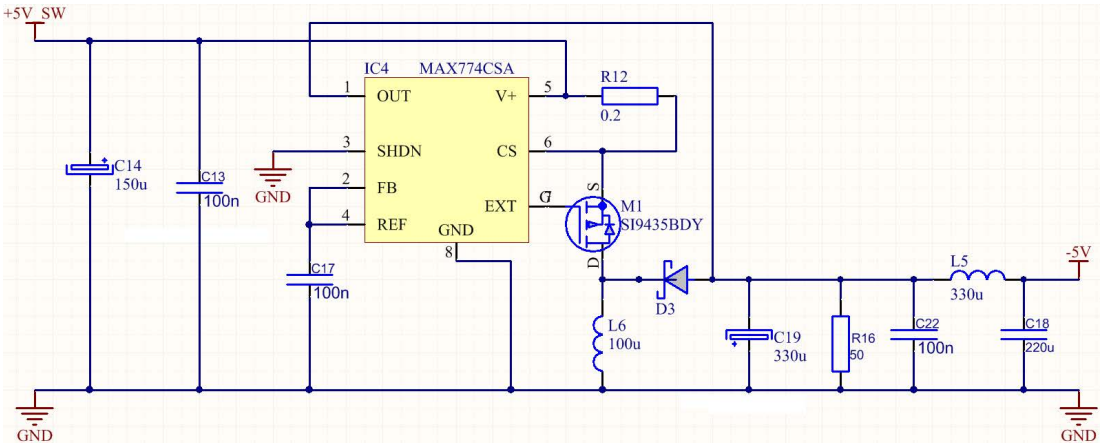
Kot je omenjeno v zgornjem odstavku, napajanje v analogni sistem dovedemo prek USB vhoda. Električna shema vezave je prikazana na sliki 4.3. Ker iz USB priključka potrebujemo le napajanje, smo podatkovni liniji D+ in D− ozemljili. Kondenzatorja C_8 in C_9 ter tuljava L_1 sestavljajo π filter, ki preprečuje širjenje motenj iz napajalnika na vezje in obratno. Kovinsko ohišje USB priključka je ozemljeno prek paralelne vezave kondenzatorja C_{10} in upora R_{11}, ki tvorita visoko prepustni filter.



Slika 4.3: Električna shema USB vhoda.

4.1.1 Preklopna regulatorja

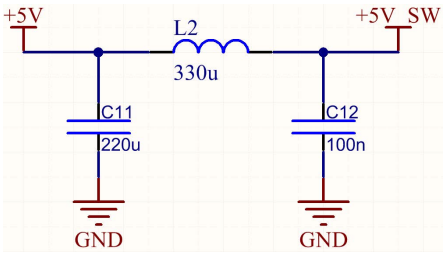
Za napajanje operacijskih ojačevalnikov na vezju za obdelavo analognega signala potrebujemo poleg +5 V tudi negativno napetost −5 V. V ta namen smo uporabili čip MAX774CSA proizvajalca Maxim Integrated, ki z dodatnimi komponentami omogoča realizacijo različnih preklopnih regulatorjev. V našem primeru smo uporabili konfiguracijo invertirajočega preklopnega regulatorja, ki je že podana v podatkovnem listu in prikazana na sliki 4.4.



Slika 4.4: Invertirajoči preklopni regulator.

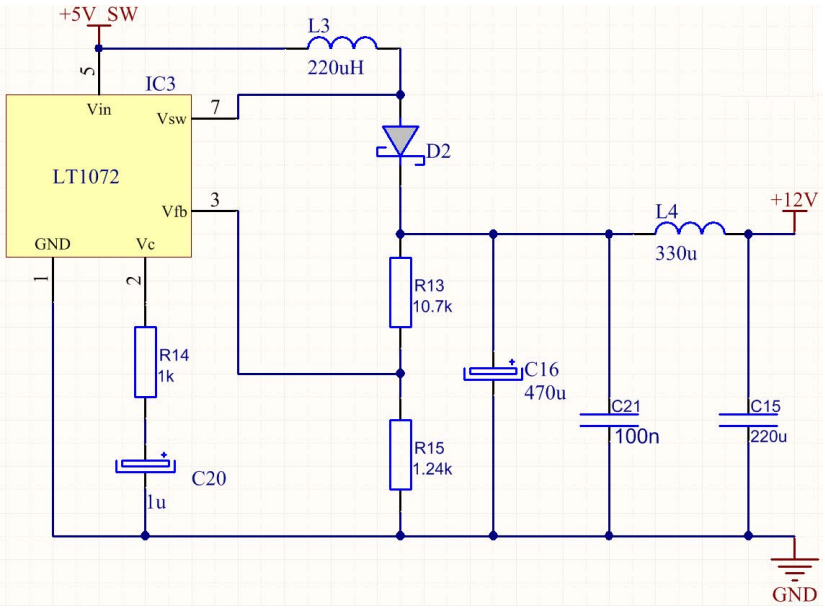
Zaradi majhne tokovne porabe operacijskih ojačevalnikov regulator deluje v

prekinjenem načinu. Posledično se ob vsakem zagonu regulatorja na vhodu in izhodu pojavijo napetostni sunki, ki predstavljajo motnjo. Sunke odpravimo tako, da dodamo upor R_{16}, ki služi kot dodatno breme. Na ta način povečamo tokovno porabo in regulator začne delovati v neprekinjenem načinu. Poleg bremenskega upora smo na izhod dodali LC filter (L_5 in C_{18}), ki zaduši preostali šum. Za odpravo motenj, ki jih regulator oddaja na vhod, smo dodali π filter, prikazan na sliki 4.5. S tem ločimo napajanje za oba regulatorja in visokonapetostni del, $+5V_{SW}$, od pozitivnega napajanja +5V ter preprečimo širjenje motenj iz regulatorjev na preostali del analognega sistema.



Slika 4.5: π filter.

Visokonapetostni del napajanja poleg napetosti +5 V potrebuje tudi napetost +12 V. V ta namen smo uporabili čip LT1072 podjetja Linear Technology, ki omogoča realizacijo preklopnega regulatorja navzgor. Električna shema je podana v podatkovnem listu in prikazana na sliki 4.6. Na izhod regulatorja je dodan še LC filter (L_4 in C_4) za preprečitev širjenja motenj.



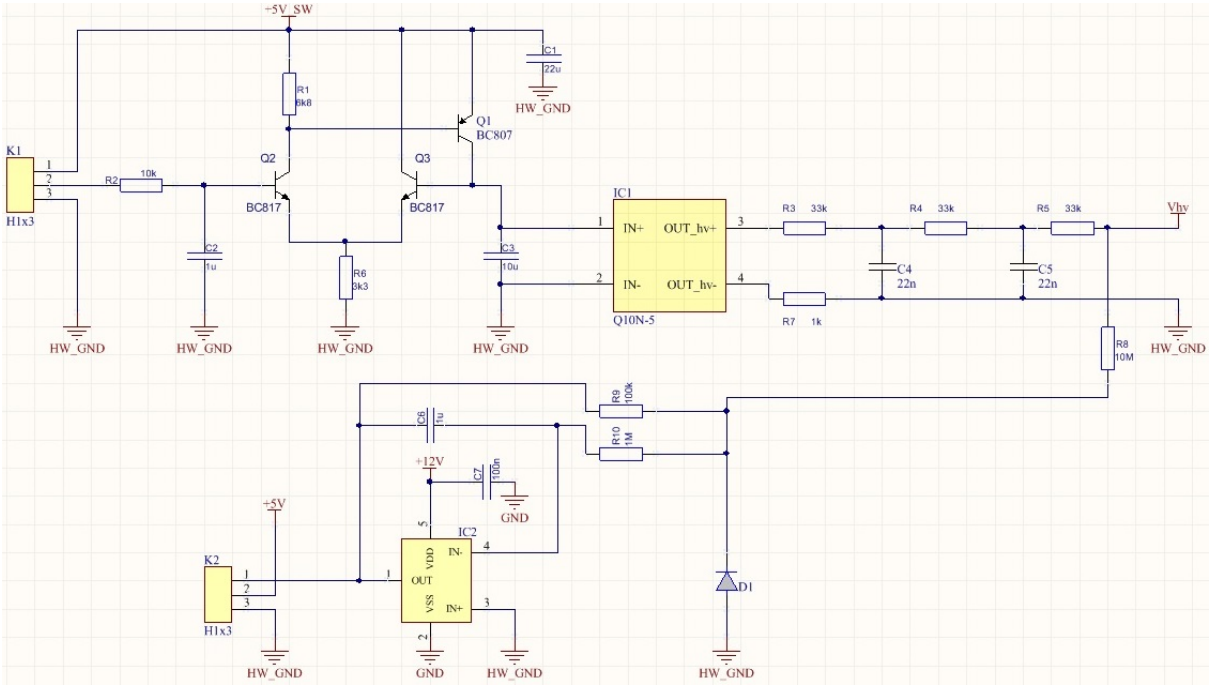
Slika 4.6: Preklopni regulator navzgor.

4.1.2 Visokonapetostno napajanje

Fotopomnoževalka, ki se nahaja na vezju za zajem analognih signalov, za delovanje potrebuje napetost -900 V. V ta namen smo uporabili že izdelano električno shemo [10], ki je prikazana na sliki 4.7. Glavni del predstavlja visokonapetostni enosmerni regulator, ki je na sliki označen z Q10N-5. Regulator vhodno napetost na območju med 0 V do 5 V pretvori v izhodno napetost na območju med 0 V in -1000 V.

Na levi strani slike se nahaja priključek K_1, kamor priklopimo potenciometer. Z nastavitvijo potenciometra spreminjamo napetost na vratih tranzistorja Q_6 in posledično napetost in tok na uporu R_6. Med bazo in emitorjem tranzistorja Q_6 je namreč glede na podatkovni list napetost 1,2 V. Ker sta tranzistorja Q_2 in Q_3 enaka, sta enaki tudi njuni napetosti na vratih. To pomeni, da s potenciometrom nastavimo vhodno napetost visokonapetostnega pretvornika in s tem posledično izhodno napetost V_{HV}. Kondenzator C_3 se na željeno napetost napolni prek tranzistorja Q_1 ali sprazni prek vrat tranzistorja Q_3 ter prek pretvornika Q10N-5. Upor R_2 omeji tok v vrata tranzistorja Q_2, napetost na uporu R_1 pa požene tok, ki krmili tranzistor Q_1. Na izhodni strani visokonapetostnega pretvornika se

nahaja visokoprepustni filter, ki odpravi motnje.

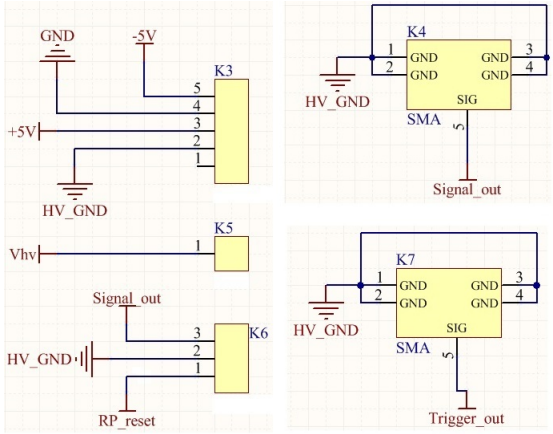


Slika 4.7: Visokonapetostno napajanje fotopomnoževalke.

Ker je visoka napetost za napajanje fotopomnoževalke uporabniško nastavljiva, želimo to napetost spremljati. V ta namen na priključek K_2 priklopimo voltmeter z zaslonom, iz katerega lahko preberemo izhodno napetost visokonapetostnega regulatorja. Ker voltmeter deluje le na območju napetosti med 0 V in 10 V, je potrebno visokonapetostno območje med 0 V in -1000 V pretvoriti v merilno območje voltmetra. Z razmerjem uporov R_8 in R_9 za faktor 100 zmanjšamo napetost V_{HV}. Tok teče iz izhoda operacijskega ojačevalnika prek upora R_9 in kondenzatorja C_6. Kondenzator se ob tem polni. Napetost na uporu R_9 in kondenzatorju C_6 je identična in pozitivna, hkrati pa po absolutni vrednosti stokrat manjša od napetosti V_{HV}. Napetost 9 V na voltmetru na primer pomeni napetost -900 V na izhodu regulatorja.

4.1.3 Priključki napajalnega vezja

Napajalno vezje je z ostalima sklopoma analognega dela sistema povezano z različnimi priključki, ki jih prikazuje slika 4.8. Priključki K_3, K_5 in K_6 so letvice, K_4 in K_7 pa sta SMA priključka za koaksialni kabel. K_3 povezuje napajalno vezje ter vezje za obdelavo analognega signala. Priključki K_4, K_6 in K_7 povezujejo Red Pitayo z vezjem za obdelavo signala, medtem ko K_5 ter HV_{GND} iz priključka K_3 povezujeta napajalno vezje in vezje za zajem analognega signala.



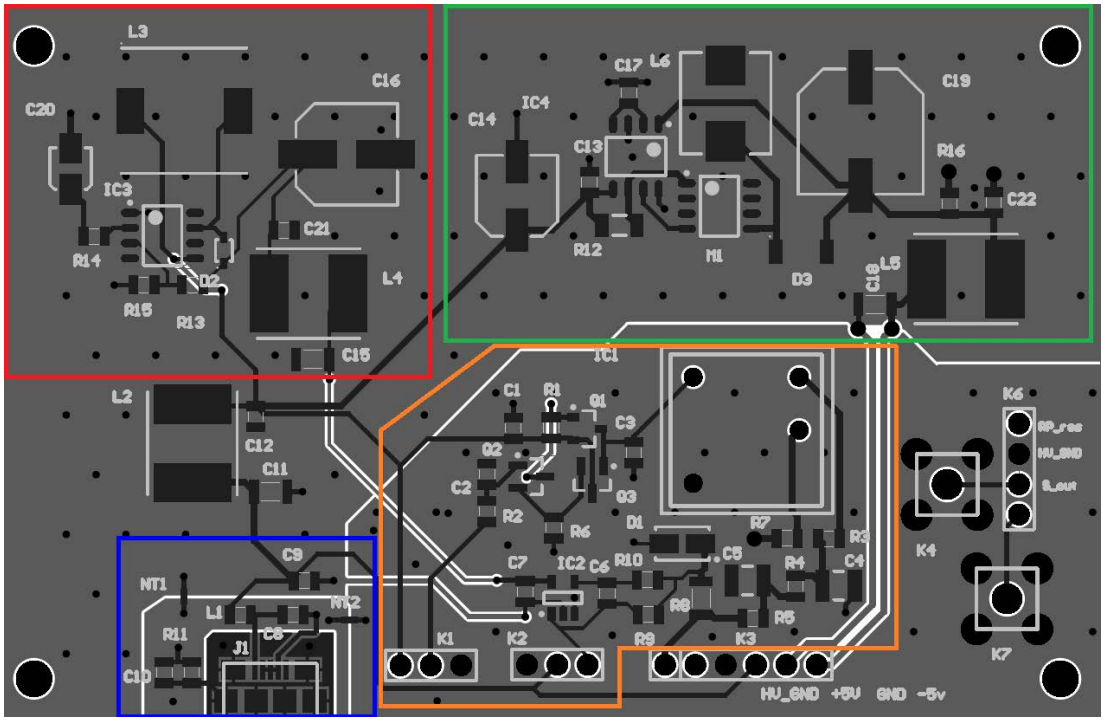
Slika 4.8: Priključki napajalnega vezja.

4.1.4 Tiskano vezje

Razporeditev elementov na tiskano vezje prikazuje slika 4.9. Z različnimi barvami so označeni posamezni sklopi napajalnega vezja. Modra barva označuje USB vhod, rdeča preklopni regulator navzgor iz +5 V na +12 V, zelena invertirajoči preklopni regulator iz +5 V na −5 V in oranžna visokonapetostno napajanje. Na sliki niso posebej označeni filter, s katerim ločimo napajanje preklopnih regulatorjev in visokonapetostnega dela od ostalega vezja (C_{11}, C_{12} in L_2 na levi strani slike), ter vhodno-izhodni priključki vezja, na sliki desno spodaj. Poleg tega ima vezje v kotih izvrtine za mehansko pritrditev.

Pri postavitvi in povezovanju elementov smo posebno pozornost namenili povezavam okoli obeh preklopnih regulatorjev. Povezave, po katerih teče velik tok,

morajo biti širše od tistih, po katerih teče majhen tok. Poleg tega morajo povezave oklepati čim manjšo površino, saj s tem zmanjšamo širjenje elektromagnetnih motenj.



Slika 4.9: Tiskano vezje.

Pri preklopnem regulatorju navzgor, ki je na sliki označen z rdečo barvo, mora biti povezava, ki se začne s čipom LT1072 (IC_3) ter nadaljuje čez tuljavo L_3, diodo D_2 ter kondenzator C_{16}, čim krajša. To je povezava med vhodom +5 V in izhodom +12 V. Zaradi velikosti elementov je ta zahteva le delno izpolnjena. Velikost tuljave L_3 je tudi razlog, da njena povezava s čipom LT1072 oklepa dokaj veliko površino. Širina povezav pri preklopnem regulatorju navzgor ni kritična, saj z njim napajamo le en operacijski ojačevalnik, ki ima tokovno porabo pod 1 mA.

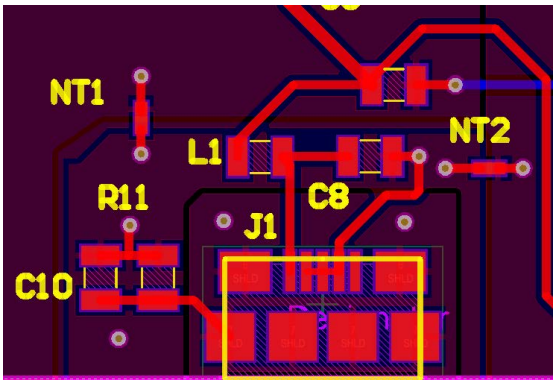
Pri invertirajočem preklopnem regulatorju, ki je na sliki označen z zeleno barvo, podatkovni list priporoča, da so ozemljitve vhodnega kondenzatorja C_{14}, izhodnega kondenzatorja C_{19}, tuljave L_6 in GND nogice čipa MAX774CSA čim bližje skupaj. Poleg tega je pomembno, da povezave med elementi, po katerih teče

velik tok, zajemajo čim manjšo površino. To je le delno izpolnjeno zaradi mehanskih velikosti elementov in prostora na vezju. Pomembna je tudi širina povezav, po katerih teče velik tok. V našem primeru je to povezava med kondenzatorjem C_{14}, uporom R_{12}, tranzistorjem M_1 in tuljavo L_6 na vhodni strani ter povezava med tuljavo L_6, Shotkeyevo diodo D_3, izhodnim kondenzatorjem C_{19} in dodanim bremenskim uporom R_{16} na izhodni strani regulatorja. Večjo širino povezav smo obdržali tudi pri povezovanju ostalih elementov na izhodu regulatorja.

Pri postavitvi in povezovanju elementov visokonapetostnega dela, ki je na sliki označen z oranžno barvo, je pomembno, da so povezave in elementi, med katerimi je visoka napetost, prostorsko dovolj ločeni. S tem preprečimo pojav napetostnega preboja. Prebojna trdnost zraka znaša 3 kV/mm, prebojna trdnost materiala FR4, iz katerega je narejeno vezje, pa 20 kV/mm. Prebojna trdnost zraka je manjša od materiala FR4, zato igra večjo vlogo. Posledično morajo biti povezave ali elementi pri maksimalni dosegljivi napetosti -1000 V med seboj oddaljeni vsaj 0,33 mm. Kritični elementi in povezave se na sliki nahajajo direktno pod visokonapetostnim enosmernim pretvornikom Q10-N. To so elementi R_7, R_3, R_4, C_4, C_5, R_5, R_8 in dioda D_1.

Poleg elementov in povezav je na vezju pomembna tudi postavitev priključkov. Dobro je, da se priključki nahajajo le na enem robu ali v enem kotu vezja. S tem dosežemo, da so vsi na enakem referenčnem potencialu, poleg tega pa s tem poskrbimo, da morebitne motnje, ki se širijo med priključki, ne potujejo čez celotno vezje [11]. V ta namen so vsi vhodno-izhodni priključki napajalnega vezja postavljeni na spodnji rob in na spodnjo stran desnega roba vezja.

Na koncu je pomembna še povezava različnih ozemljitev vezja. Analogni del ima tri različne ozemljitve, to so USB_{GND}, GND in HV_{GND}. Z ločenimi ozemljitvami preprečimo širjenje motenj iz enega dela vezja na drugega. USB_{GND} je ozemljitev za vhodno napetost, ki v analogni del pride prek USB vhoda, GND je ozemljitev za preklopne regulatorje in operacijske ojačevalnike na vezju za obdelavo analognih signalov ter HV_{GND} je ozemljitev za visokonapetostni del vezja in hkrati tudi referenčna ozemljitev za signale, ki jih merimo na ostalih dveh vezjih in AD pretvornikih Red Pitaye. Vse tri ozemljitve se prek povezav NT_1 in NT_2 združijo pri USB vhodu, kot je prikazano na sliki 4.10.



Slika 4.10: Povezava ozemljitev vezja.

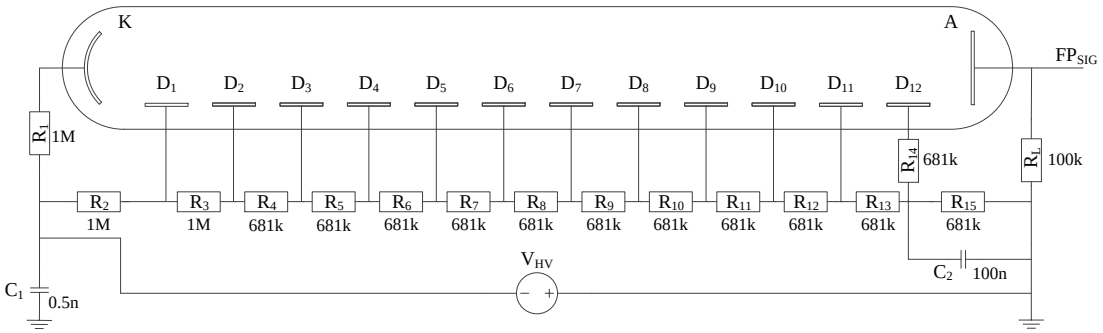
4.2 Zajem analognega signala

V poglavju 2.2.1 je opisano, da žarek gama vpade na scintilacijski kristal, ki izseva fotone. Fotoni zatem padejo na katodo fotopomnoževalke in na anodi dobimo električni signal, sorazmeren številu vpadnih fotonov in posredno energiji žarka gama. Vezja za zajem analognega signala nismo osnovali sami, vendar smo delujočega že prejeli pred začetkom dela [12]. Pri tem smo uporabili fotopomnoževalko Hamatsu R5900-m16, ki je stara, zato nismo uspeli najti njenega podatkovnega lista. Posledično smo se pri razumevanju delovanja že narejenega vezja zanašali na priročnik o osnovah fotopomnoževalk, ki ga je izdalo podjetje Hamatsu [13].

4.2.1 Priključitev fotopomnoževalke

Priključitev fotopomnoževalke prikazuje slika 4.11. Fotopomnoževalka za delovanje potrebuje visoko napetost med katodo in anodo, ki v našem primeru znaša -900 V. Med katodo in anodo se nahaja uporovni delilnik R_2 – R_{15}, ki s pravilno izbranimi upori skrbi za primeren gradient napetosti med dinodami D_1 – D_{12}, na katerih se pospešujejo elektroni. Izbira vrednosti uporov je pomembna, saj sta od tega odvisni linearnost odziva fotopomnoževalke ter maksimalna jakost vpadle svetlobe, preden fotopomnoževalka preide v nasičenje. Iz slike je razvidno, da imajo upori R_1, R_2 in R_3 večjo upornost od ostalih v uporovnem delilniku. Ta vezava omogoča večjo učinkovitost dinode D_1 pri zbiranju fotoelektronov iz

katode, kar pozitivno vpliva na razmerje signal-šum.



Slika 4.11: Shema priključitve fotopomnoževalke.

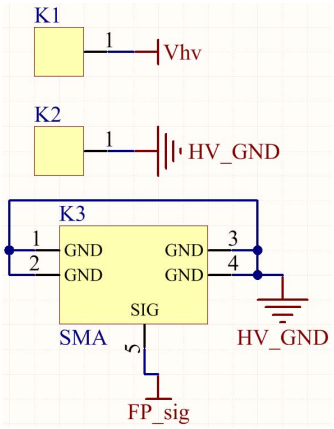
Pri zaznavanju žarkov gama fotopomnoževalka deluje v impulznem načinu. V tem načinu je za dober linearni odziv anodni tok omejen na majhen del toka, ki teče skozi uporovni delilnik. Omejitev odpravimo z vzporedno vezavo kondenzatorjev z zadnjimi nekaj upori v uporovnem delilniku. V našem primeru je uporabljen le kondenzator C_2, ki med nastankom impulza napaja zadnjo dinodo D_{12} z električnim nabojem in omeji padec napetosti med dinodo in anodo. To močno poveča območje linearnosti odziva fotopomnoževalke in izhodni anodni tok. Več toka na izhodu pomeni večji izhodni signal. Izhodni impulzi iz fotopomnoževalke so hitri, z dvižnim časom (*ang.* rise time) krajšim od 10 ns. Posledično po koncu impulza pride do prenihajev (*ang.* ringing), ki jih odpravimo z vezavo upora R_{14} med zadnjo dinodo in uporovni delilnik.

Izhodni signal iz fotopomnoževalke so tokovni sunki. Ker na izhodu želimo napetostni signal FP_{SIG}, je med izhod in ozemljitev povezan bremenski upor R_L. Izhodni napetostni impulzi imajo negativno polariteto zaradi načina vezave fotopomnoževalke.

4.2.2 Priključki vezja za zajem analognega signala

Vezje za zajem analognega signala je z ostalim deloma analognega sistema povezano prek priključkov, ki so prikazani na sliki 4.12. Priključka K_1 in K_2 sta letvici in skupaj tvorita priključek za dovod visoke napetosti za fotopomnoževalko iz

napajalnega vezja. Priključek K_3 je tipa SMA in je namenjen priključitvi koaksialnega kabla po katerem potuje signal iz fotopomnoževalke do vezja za obdelavo analognega signala.

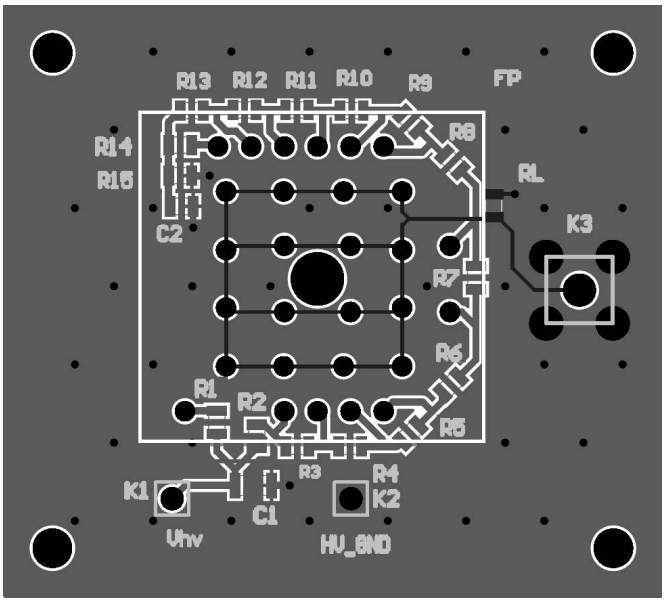


Slika 4.12: Priključki vezja za zajem analognega signala.

4.2.3 Tiskano vezje

Razporeditev elementov na tiskano vezje prikazuje slika 4.13. Elementi in povezave črne barve so postavljeni na zgornjo stran vezja, medtem ko se elementi in povezave sive barve nahajajo na spodnji strani.

Centralni del vezja zajema fotopomnoževalka. Njeno pozicijo označuje bel kvadrat, znotraj katerega se nahajajo priključki. Priključki ob notranjem robu kvadrata so katoda in dinode, medtem ko so centralni priključki, razporejeni v kvadrat, anode. Fotopomnoževalka Hamatsu R5900-m16 ima šestnajst anod, kar omogoča določanje pozicije vpadnih fotonov. V našem primeru nas zanimata le energija in število razpadov, zato smo vse anode povezali skupaj. Signal iz anod je nato speljan na upor R_L in na izhodni priključek K_3. Priključka za visoko napetost sta postavljena ob spodnji rob vezja in povezana z uporovnim delilnikom, ki je postavljen okoli fotopomnoževalke.



Slika 4.13: Tiskano vezje.

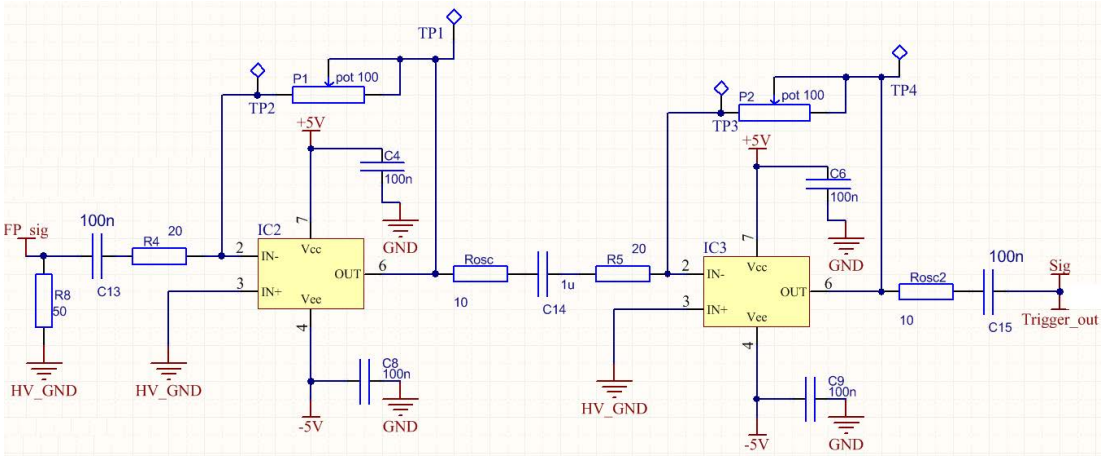
4.3 Obdelava analognega signala

Napetostni signal, ki pride s strani vezja za zajem analognega signala, je potrebno ojačati in integrirati, da dobimo napetostni signal V_γ, ki je sorazmeren energiji vpadnega žarka gama.

4.3.1 Ojačenje signala

Električna shema za ojačenje signala je prikazana na sliki 4.14. Signal FP_{SIG} iz fotopomnoževalke na vezje za obdelavo analognega signala pride prek koaksialnega kabla, ki je zaključen s 50-omskim uporom R_8. Pred ojačenjem signalu odstranimo enosmerno komponento s kondenzatorjem C_{13}. V nadaljevanju ojačenje poteka prek dveh invertirajočih operacijskih ojačevalnikov. Ojačenje za posamezen ojačevalnik je mogoče nastaviti s potenciometroma P_1 in P_2. Pomembno je, da sta ojačevalnika dovolj hitra, da lahko sledita signalu ter da ojačujeta na dovolj velikem frekvenčnem območju. V ta namen smo izbrali ojačevalnik MAX4304 proizvajalca Maxim Integrated, ki z dvižnim časom (*ang.* slew rate) 1400 V/μs in pasovno širino 740 MHz zadosti našim potrebam. Glede na po-

datkovni list ojačevalnik na izhodu potrebuje izolacijski upor (R_{OSC} in R_{OSC2}), ki prepreči oscilacije ojačevalnika, kadar z njim krmilimo kapacitivna bremena nad 10 pF. Kljub temu, da pri ojačenju signala ne krmilimo kapacitivnih bremen, smo upora dodali, saj jih je mogoče tekom testiranja odstraniti, če ne bosta potrebna. Ojačevalnika imata na izhodih enosmerno napetost različno od nič tudi, ko na vhodu ni signala. Odstranimo jo s kondenzatorjema C_{14} in C_{15}. Ker sta ojačevalnika zelo hitra in občutljiva, je pomembno, da za stabilizacijo napajanja uporabimo blokirne kondenzatorje, ki so na sliki označeni z C_4, C_6, C_8 in C_9. Za lažje testiranje vezja in nastavljanje potenciometrov P_1 in P_2 so dodane štiri testne točke $TP_{1...4}$. Končni ojačeni signal na izhodu drugega ojačevalnika se razdeli na dve liniji, označeni s *Sig* in $Trigger_{out}$. Linija *Sig* je povezana na vhod integratorja, linija $Trigger_{out}$ pa na izhodni priključek vezja in od tam prek priključkov na napajalnem vezju na enega od dveh AD pretvornikov Red Pitaye. $Trigger_{out}$ je prožilni signal za sistem, implementiran na FPGA vezju, kar je podrobneje opisano v poglavju 5.2.3.1.



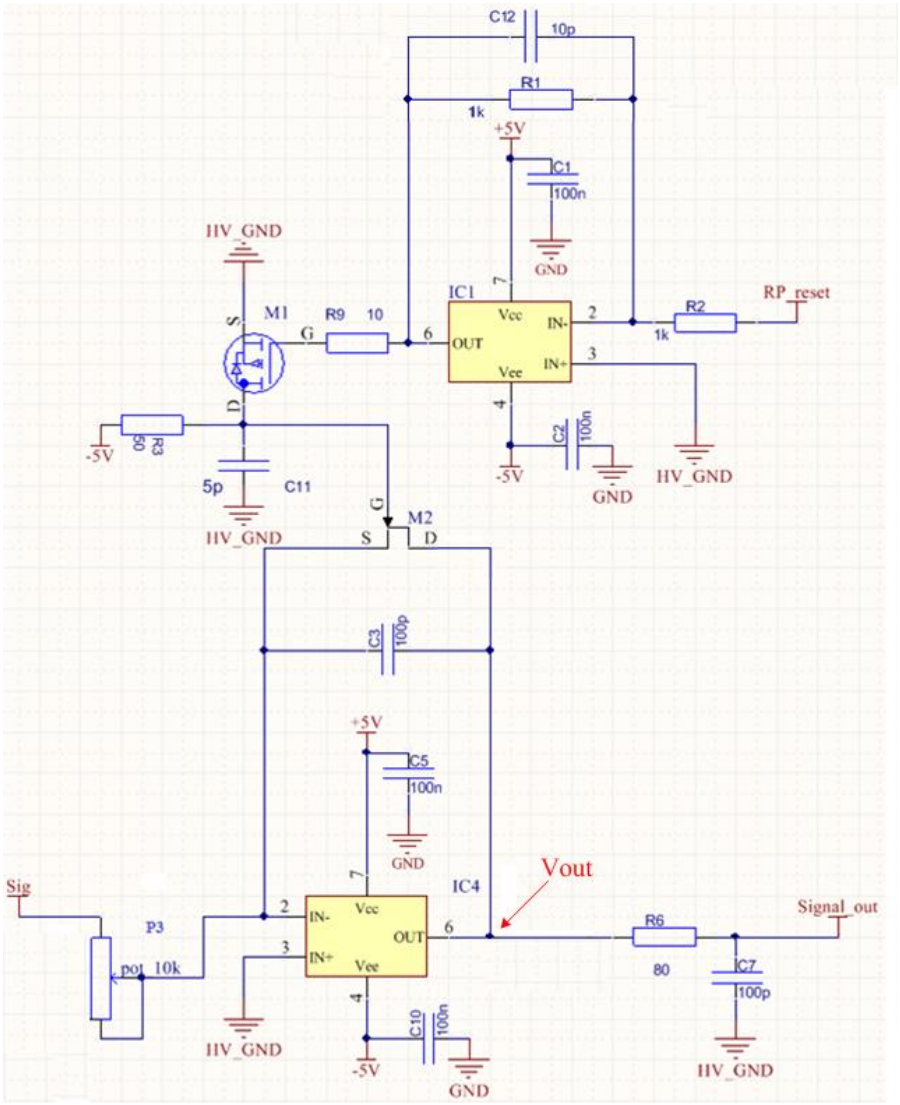
Slika 4.14: Električna shema ojačenja signala.

4.3.2 Integracija signala

Integrator z dodatnim vezjem za ponastavitev je prikazan na sliki 4.15. Namenjen je integraciji ojačenih napetostnih impulzov iz fotopomnoževalke. Izhodna napetost integratorja je označena na sliki in podana z enačbo 4.1.

$$V_{\text{out}}(t) = -\frac{1}{P_3 C_3} \int_{t_1}^{t_2} V_{\text{Sig}} dt \tag{4.1}$$

Hitrost integriranja oziroma časovno konstanto integratorja P_3C_3 je mogoče nastaviti s potenciometrom P_3. Na izhodni strani integratorja je dodan nizkopre-pustni filter (R_6 in C_7), ki zgladi nihaje znotraj linearnega naraščanja izhodne napetosti V_{out}. Signal $Signal_{out}$ na izhodu iz filtra je prek priključkov povezan na napajalno vezje in od tam prek koaksialnega kabla na AD pretvornik Red Pitaye.



Slika 4.15: Električna shema integratorja z vezjem za ponastavitev.

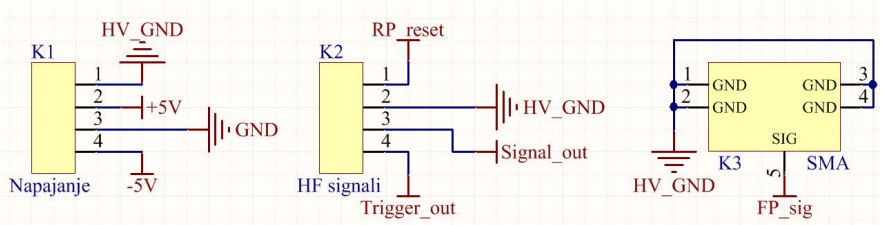
Ko se napetost na merilnem kondenzatorju C_3 veča, integrator sčasoma preide v nasičenje. Poleg tega želimo poljubno nastavljati časovno okno, v katerem poteka integracija. Da se izognemo nasičenju in omogočimo poljuben čas integriranja, je dodano vezje za ponastavitev integratorja. Integrator ponastavimo tako, da čez tranzistor M_2, ki je v vlogi stikala, spraznimo kondenzator C_3. Praznjenje čez tranzistor mora biti čim hitrejše, zato smo izbrali n-kanalni JFET. Tranzistor je odprt, ko je napetost V_{GS} med vrati in izvorom enaka 0 V, in zaprt, ko je napetost V_{GS} manjša od $-3{,}2$ V.

Zahteva za ponastavitev pride s strani Red Pitaye, prek priključkov na napajalnem vezju, na vhod v operacijski ojačevalnik IC_1. Signal za ponastavitev RP_{reset} je digitalen in zavzema logični stanji nič ali ena. Napetosti, ki ustrezata tema stanjema, sta 0 V in 3,3 V. V primeru, da je napetost na RP_{reset} enaka 0 V, je na vratih PMOS tranzistorja M_1 približno 0 V, tranzistor je zaprt in na vratih JFET tranzistorja M_2 je napetost -5 V. To pomeni, da je M_2 zaprt in poteka integracija signala. V nasprotnem primeru, ko je napetost na RP_{reset} enaka 3,3 V, je napetost na vratih tranzistorja M_1 enaka $-3{,}3$ V, tranzistor prevaja in na vratih tranzistorja M_2 je napetost 0 V. To pomeni, da M_2 prevaja in kondenzator C_3 se izprazni. Preklopi med integriranjem in ponastavitvijo ter obratno, so hitri. Posledično se takoj po preklopih na vratih tranzistorja M_2 pojavijo prenihaji, ki prek kapacitivnosti med vrati in ponorom tranzistorja polnijo merilni kondenzator C_3. To je problematično med preklopom iz stanja ponastavitve v stanje integriranja, ker vsako polnjenje kondenzatorja C_3, ki ni posledica signala *Sig*, predstavlja motnjo. Da prenihaje zmanjšamo, smo dodali kondenzator C_{11}.

Za pravilno delovanje integratorja je ključnega pomena, da ima ojačevalnik IC_4 majhen vhodni tok v invertirajoči vhod. V naprotnem primeru ojačevalnik preko izhoda sam polni merilni kondenzator C_3 tudi, ko na vhodu v integrator ni signala in integrator je neuporaben. V našem primeru smo uporabili operacijski ojačevalnik OPA656 z vhodnim tokom 2 pA. Ojačevalnik za ponastavitev integratorja je enak uporabljenim za ojačenje signala. Oba operacijska ojačevalnika imata na napajalnih linijah priključene blokirne kondenzatorje C_1, C_2, C_5 in C_{10}, ki stabilizirajo napajanje.

4.3.3 Priključki vezja za obdelavo analognega signala

Vezje za obdelavo analognega signala je z ostalima sklopoma analognega dela sistema povezana s priključki, ki jih prikazuje slika 4.16. Priključka K_1 in K_2 sta letvici. Prek priključka K_1 vezje dobi napajanje iz napajalnega sklopa. Priključek K_2 vsebuje povezave, ki prek napajalnega vezja povezujejo Red Pitayo in vezje za obdelavo analognega signala. K_3 je SMA priključek za koaksialni kabel in povezuje vezje za obdelavo analognega signala z vezjem za zajem signala.



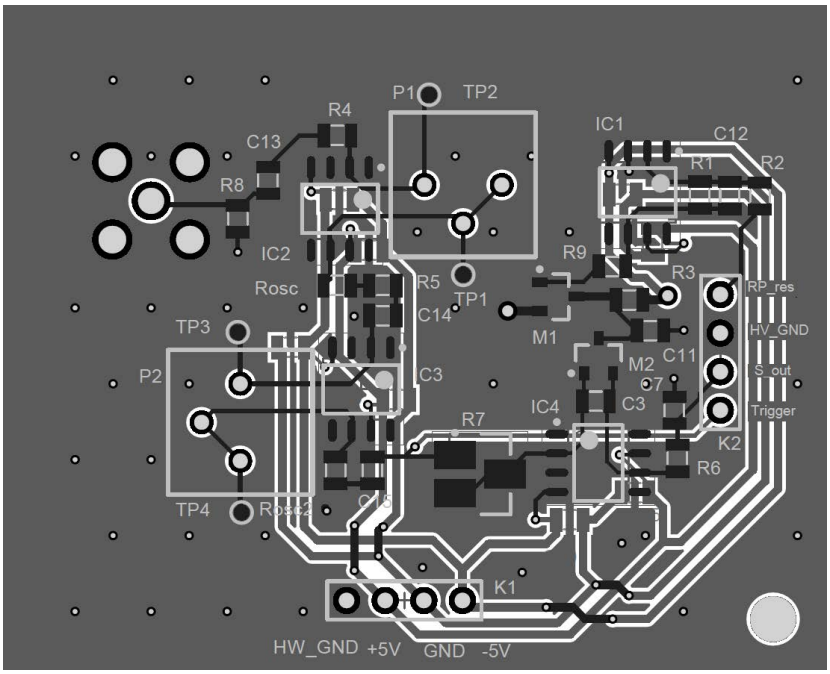
Slika 4.16: Priključki vezja za obdelavo analognega signala.

4.3.4 Tiskano vezje

Razporeditev elementov na tiskano vezje prikazuje slika 4.17. Na sliki je prikazan le spodnji desni del vezja, kjer so postavljeni elementi in povezave. Celotno tiskano vezje ima sicer enake dimenzije kot napajalno vezje. Priključka K_1 in K_2 ter mehanske izvrtine so postavljeni na enake pozicije kot na napajalnem vezju. Vezje za obdelavo analognega signala se zato direktno priklopi na napajalno vezje.

Na sliki so s črno barvo označeni elementi in povezave, ki ležijo na zgornji strani vezja, in s sivo elementi in povezave, ki ležijo na spodnji strani. Zgornja stran vezja je uporabljena za elemente in povezave, po katerih potuje signal, medtem ko je spodnja stran namenjena napajalnim linijam in blokirnim kondenzatorjem za operacijske ojačevalnike. Signal iz fotopomnoževalke je hiter; posamezen pulz traja okoli 10 ns. Pri povezovanju elementov smo zato glede na priporočilo podatkovnega lista operacijskih ojačevalnikov poskrbeli, da povezovalne linije ne oklepajo ostrih kotov, manjših od 90°. Poleg tega morajo biti linije čim krajše in direktne. Pomembna je tudi širina povezav, po katerih teče velik tok. Na zgornji strani vezja je to povezava tranzistorja M_1 in upora R_3 med napetostjo -5 V

in ozemljitvijo HV_{GND}. Na spodnji strani smo večjo širino povezav uporabili za napajalne linije.



Slika 4.17: Tiskano vezje.

Pri visokofrekvenčnih signalih je pomembno, da linije na spodnji strani vezja čim manj sekajo linije na zgornji strani. Pri visokih frekvencah se namreč tokovi na dvostranskih vezjih do izvora vračajo direktno pod signalno linijo in ne po najkrajši možni poti kot pri nizkih frekvencah [11]. Posledično smo napajalne linije na spodnji strani vezja, kjer je bilo mogoče, speljali tako, da čim manj sekajo povezave, po katerih potuje signal.

5 Digitalni del

Digitalni del je sestavljen iz Red Pitaye in osebnega računalnika. Red Pitaya vsebuje FPGA vezje, ki iz vhodnih podatkov računa histogram ter ARM procesor, ki skrbi za komunikacijo z računalnikom. Na AD pretvornika Red Pitaye sta povezana analogna signala $Signal_{out}$ in $Trigger_{out}$, torej izhodni signal integratorja in ojačen signal iz fotopomnoževalke. Signala sta vhoda v sistem, implementiran na FPGA vezju. Na računalniku teče grafični vmesnik za prikaz meritev in nastavitev spektrometra.

5.1 Histogram

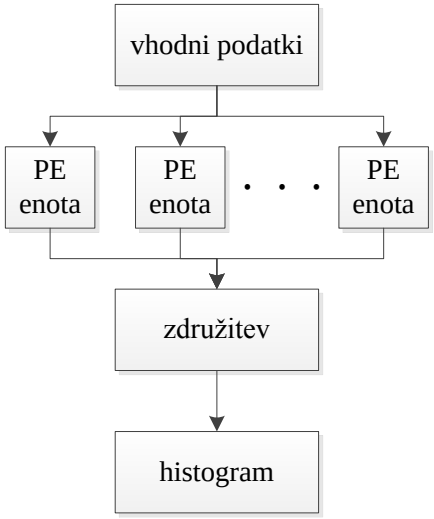
Histogram je predstavitev porazdelitve diskretnih podatkov na območju, ki ga podatki lahko zavzamejo. Območje razdelimo na poljubno število intervalov oziroma stolpcev (*ang.* bins) ter štejemo število podatkov, ki padejo v posamezen interval. Funkcija, ki v programskem jeziku C naredi histogram iz vhodnega seznama podatkov *data_in* velikosti *DATA_SIZE*, je sledeča [14]:

```
1  void histogram (int data_in[DATA_SIZE], int hist[])
2  {
3     int bin;
4
5     for (int i = 0; i < DATA_SIZE; i++)
6     {
7        bin = data_in[i];
8        hist[bin] = hist[bin] + 1;
9     }
10 }
```

Za vhodne sezname, ki vsebujejo veliko število podatkov, je zgornja implementacija počasna, saj je potrebno za vsak podatek iz seznama najprej iz pomnilnika prebrati staro vrednost hist[bin], jo povečati za 1 in nato novo vrednost zapisati nazaj v pomnilnik. V programskih jezikih, kjer se programi izvajajo zaporedno oz. serijsko (vrstica za vrstico), je zato zgornji algoritem slaba rešitev.

V našem primeru želimo razvrščanje podatkov v histogram implementirati na FPGA vezju. Pri tem uporabimo koncept računanja histograma, ki je podan v zgornji funkciji *histogram()*, ter hkrati izkoristimo prednosti, ki jih omogočajo FPGA vezja pred klasičnimi mikrokrmilniki.

FPGA vezja vsebujejo nastavljive bralno-pisalne pomnilniške enote, imenovane blok RAM (BRAM), ki podpirajo dvovratni dostop (DP-BRAM). Vsaka vrata (*ang.* port) pomnilnika delujejo neodvisno, zato pomnilnik omogoča hkratno branje in pisanje podatkov, kar poveča hitrost računanja histograma. Nadaljnje lahko izkoristimo še eno lastnost FPGA vezij, to je paralelno izvajanje programskih inštrukcij. Algoritem za računanje histograma, ki upošteva paralelizem, prikazuje slika 5.1. V prvi fazi razdelimo vhodne podatke na več manjših delov in vsakega od njih vodimo v svojo enoto za izračun histograma, označeno s PE. Vse PE enote hkrati in neodvisno izračunajo histogram za svoj del podatkov. V drugi fazi sledi združitev delnih histogramov v končni histogram, ki zajema vse vhodne podatke.



Slika 5.1: Algoritem za računanje histograma na FPGA vezju.

5.2 Implementacija na FPGA vezju

Pri delu z FPGA delom Red Pitaye smo uporabili prosto dostopno programsko ogrodje, narejeno s strani razvijalcev te razvojne plošče. Ogrodje je dostopno na GitHubu, vendar se vseskozi posodablja. V našem primeru smo uporabili verzijo, uporabljeno na vajah pri predmetu Digitalna integrirana vezja in sistemi [15], ki se izvaja na Fakulteti za elektrotehniko UL. Ogrodje vsebuje vrhovno komponento (*ang.* top level) *red_pitaya_top*, kamor smo vključili svojo komponento, ki računa histograme.

Pri implementaciji sistema za računanje histogramov smo kot osnovno vodilo upoštevali algoritem iz slike 5.1. Sčasoma smo ugotovili, da ima tak sistem nekaj odvečnih delov, hkrati pa ne zadostuje vsem potrebam, ki jih zahteva naša naloga. Posledično smo implementirali generičen sistem, ki ima več načinov delovanja (*ang.* modes). Sistem je realiziran znotraj vrhovne komponente *hist_system_top*, ki je prikazana v prilogi A. Različne načine in obnašanje sistema znotraj posameznega načina delovanja nastavimo v VHDL kodi z generičnimi spremenljivkami, ki so:

- ***mode***: Nastavi način delovanja sistema. Možni so trije načini delovanja MODE_1, MODE_2 in MODE_3.
- ***hist_bit_num***: Število bitov številke, ki predstavlja indeks stolpca v histogramu.
- ***bin_num***: Število intervalov oziroma stolpcev v histogramu. Vsi stolpci imajo širino enako ena. Vrednost te spremenljivke mora biti vedno enaka $2^{hist\_bit\_num}$.
- ***buffer_size***: Bitna velikost pomnilniške komponente *BRAM_buffer*, ki hrani zajete podatke. Poleg tega je 2^{buffer_size} maksimalno število podatkov v histogramu.
- ***num_of_units***: Število komponent *BRAM_buffer* in *PE_unit*.

Vrhovna komponenta *hist_system_top* vsebuje komponente *ADC_unit*, *BRAM_buffer* in *PE_unit*. Prav tako se v njej izvajajo trije procesi. Proces *HIST_RECONSTRUCITON* združi delne histograme iz komponent *PE_unit*, tako da sešteje posamezne stolpce z enakim indeksom in to naredi na vseh *bin_num*

intervalih histograma. Ostala dva procesa *REC_FROM_μC* in *SEND_TO_μC* skrbita za komunikacijo s procesorskim delom sistema Zynq. Medsebojna povezava komponent v *hist_system_top* je odvisna od načina delovanja. Komponente *ADC_unit*, *BRAM_buffer* in *PE_unit* se glede na izbran način povežejo preko multiplekserjev in demultiplekserjev, ki jih krmili generična spremenljivka *mode*. Poleg tega je delovanje vsake od komponent prirejeno za posamezen način. Kljub temu, da ima sistem tri načine delovanja, samo način MODE_3 ustreza zahtevam naše naloge. Posledično sta načina MODE_1 in MODE_2 opisana konceptualno, medtem ko je način MODE_3 opisan podrobno.

5.2.1 Način delovanja MODE_1

Način delovanja MODE_1 je implementacija algoritma za računanje histogramov iz slike 5.1. Najprej potrebujemo vhodne podatke. Za to skrbi komponenta *ADC_unit*, ki podatke zajema iz AD pretvornika Red Pitaye. V naslednjem koraku je potrebno podatke med zajemanjem razdeliti na enako velike dele in jih shraniti za kasnejšo obdelavo v komponentah *PE_unit*. Temu služijo pomnilniške komponente *BRAM_buffer*. Njihovo število in število komponent *PE_unit* nastavimo z generično spremenljivko *num_of_units*. Komponenta *ADC_unit* zajete podatke pošilja v komponente *BRAM_buffer*. Začne s prvo in, ko se ta napolni, preklopi na drugo ter tako nadaljuje do zadnje. Ko so polne vse komponente, začnejo hkrati pošiljati podatke v komponente *PE_unit*. Zatem se v komponentah *PE_unit* hkrati izračunajo delni histogrami, ki se v procesu *HIST_RECONSTRUCTION* združijo v končni histogram. Nazadnje končni histogram prebere procesor in postopek se ponovi.

Način MODE_1 omogoča hitro obdelavo vhodnih podatkov, ker upošteva paralelizem, opisan v poglavju 5.1. Hitrost algoritma torej pride do izraza potem, ko so vhodni podatki že zbrani. V našem primeru podatke šele beremo iz AD pretvornika, zato v sistem prihajajo serijsko oziroma "eden za drugim". Posledično shranjevanje podatkov v komponente *BRAM_buffer* za kasnejšo paralelno obdelavo ne prinese časovne prednosti, saj lahko posamezen podatek obdelamo in shranimo v histogram takoj, ko ga preberemo iz AD pretvornika. Pomnilniške komponente *BRAM_buffer* so torej odveč. Poleg tega potrebujemo samo eno komponento *PE_unit*, ki sproti računa histogram iz podatkov, ki jih pošilja kom-

ponenta *ADC_unit*. Vse to naredi način delovanja MODE_1 neprimeren za uporabo v naši nalogi.

5.2.2 Način delovanja MODE_2

Način delovanja MODE_2 se od načina MODE_2 razlikuje v tem, da ne uporablja pomnilniških komponent *BRAM_buffer* ter da je generična spremenljivka *num_of_units* nastavljena na vrednost 1. To pomeni, da sta v komponenti *hist_system_top* poleg treh procesov le komponenta *ADC_unit* in ena komponenta *PE_unit*. Vloga *ADC_unit* je enaka kot v načinu MODE_1, razlika je le, da podatkov ne pošilja v več komponent *BRAM_buffer*, temveč v eno komponento *PE_unit*, ki sproti računa histogram. Ko je računanje histograma končano, procesor prebere končni histogram iz komponente *PE_unit*. Postopek se nato ponovi. Proces *HIST_RECONSTRUCTION*, ki združi delne histograme, v tem načinu delovanja nima vloge. Delovanje sistema v načinu MODE_2 ni primerno za našo nalogo, ker komponenta *ADC_unit* komponenti *PE_unit* pošlje vsak nov zajeti podatek. V tem načinu namreč še niso implementirane nekatere možnosti za upravljanje s sistemom, kot so na primer časovno okno zajemanja podatkov, pragovna vrednost, pod katero podatkov ne obdelujemo, prožilni signal, ki sistem obvešča o prihodu žarka gama, nadzor integratorja na vezju za obdelavo analognega signala in možnost globalne ponastavitve sistema. Poleg tega lahko procesor histogram prebere šele, ko je računanje končano, kar ne ustreza zahtevi, da mora biti sistem čim bolj realnočasen.

5.2.3 Način delovanja MODE_3

Način delovanja MODE_3 je posodobljena različica načina MODE_2. Vrhovna komponenta *hist_system_top* še vedno vsebuje tri procese, komponento *ADC_unit* in eno komponento *PE_unit*, vendar so v načinu MODE_3 odpravljene vse pomanjkljivosti, navedene v poglavju 5.2.2, zaradi katerih način MODE_2 ni primeren za našo nalogo.

5.2.3.1 Komponenta *ADC_unit*

Komponenta *ADC_unit* bere in uredi podatke iz obeh AD pretvornikov. Enota ima osem vhodov in štiri izhode, ki so opisani v tabeli 5.1.

	Funkcija
Vhod	
clk	Signal ure.
START	Uporabniško nastavljiv signal za zagon ali zaustavitev sistema.
threshold	Uporabniško nastavljiva pragovna vrednost, pod katero sistem ne shranjuje vhodnih podatkov.
time_window	Uporabniško nastavljivo časovno okno integriranja.
data_in	Vhodno vodilo, povezano z izhodom prvega AD pretvornika. Vodilo ima digitalno vrednost analognega signala $Signal_{out}$.
HW_trigger_in	Vhodno vodilo, povezano z izhodom drugega AD pretvornika. Vodilo ima digitalno vrednost analognega signala $Trigger_{out}$.
SW_trigger_in	Uporabniško nastavljiva pragovna vrednost prožilnega signala.
ADC_unit_feedback	Kontrolni signal, ki omogoči ali onemogoči komponento. Signal je aktiven v nizkem stanju.
Izhod	
ADC_wr_en	Signal, ki omogoči vpis izhodnega podatke iz *ADC_unit* v *BRAM_buffer* (način $MODE_1$) ali v *PE_unit* (način $MODE_2$ in $MODE_3$).
ADC_addr	Naslovno vodilo za komponentno *BRAM_buffer*. Ni v uporabi v načinu $MODE_3$.
INT_res	Kontrolni signal za integrator na vezju za obdelavo analognega signala.
data_out	Izhodno vodilo. Njegova vrednost predstavlja inteks intervala (stolpca) v histogramu.

Tabela 5.1: Vhodi in izhodi komponente *ADC_unit*.

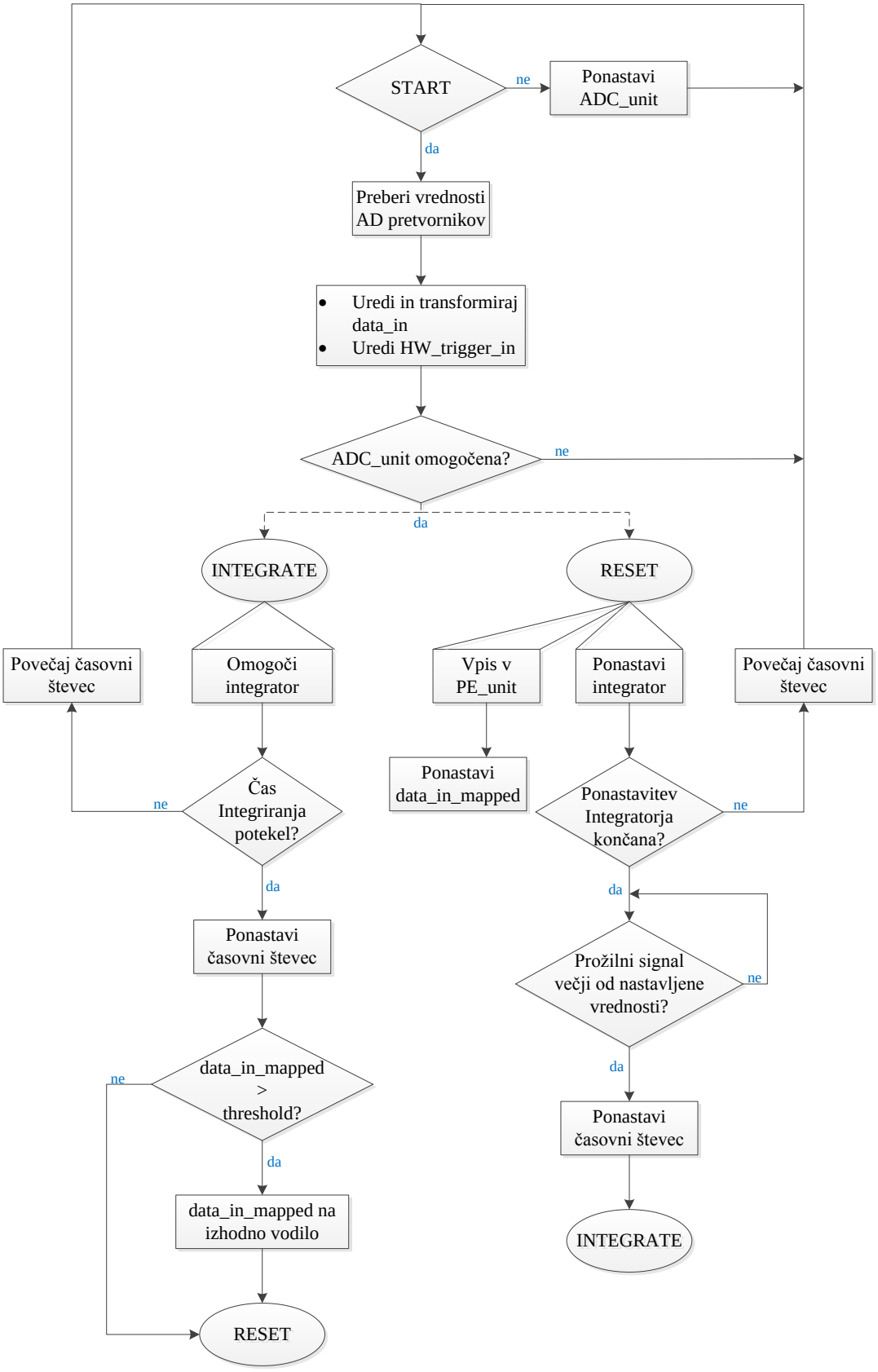
Delovanje komponente *ADC_unit* prikazuje bločni diagram na sliki 5.2. Komponenta ob vsakem pozitivnem prehodu ure preveri, ali je s strani uporabnika prišla zahteva za zagon sistema. Dokler je sistem zaustavljen, se izvaja ponastavitev, kar pomeni, da so vsi notranji signali enote postavljeni na začetne, privzete vrednosti.

Ko uporabnik zažene sistem, *ADC_unit* vsak urin cikel prebere dve 14-bitni vrednosti iz obeh AD pretvornikov, ki sta zapisani na vhodih *data_in* ter *HW_trigger_in*. AD pretvornika na Red Pitayi delujeta tako v pozitivnem kot tudi negativnem območju vhodnih analognih napetosti, kar pomeni, da je njuno 14-bitno merilno območje simetrično razdeljeno na dva dela. Podatek o tem, ali prebrani digitalni podatek predstavlja pozitivno ali negativno vhodno napetost, je zapisan v MSB (*ang.* most significat bit) bitu pretvorbe. Digitalna vrednost vhoda *data_in* je vedno pozitivna, zato s spodnjim izsekom kode zagotovimo, da upoštevamo le pozitivno merilno območje AD pretvornika.

```
1 data_in_reduced <= data_in(12 downto 0) when data_in(13) = '0'
2                    else (others => '0') when data_in(13) = '1';
```

Nasprotno je digitalna vrednost vhoda *HW_trigger_in* vedno negativna, zaradi negativnih napetostnih impulzov na izhodu fotopomnoževalke. Zaradi bolj priročne obravnave signala jo pretvorimo v pozitivno digitalno vrednost in odrežemo MSB bit tako kot pri *data_in_reduced*.

```
1 HW_trigger_in_inverted <= (not(HW_trigger_in) + 1)
2                           when HW_trigger_in(13) = '1'
3                           else (others => '0');
4
5 HW_trigger_in_reduced <= HW_trigger_in_inverted(12 downto 0);
```



Slika 5.2: Algoritem komponente *ADC_unit*.

V naslednjem koraku 13-bitno digitalno vrednost vodila *data_in_reduced* linearno transformiramo v vrednost, ki leži znotraj nastavljenega območja histograma in jo priredimo vodilu *data_in_mapped*. Kot je opisano v poglavju 5.2, je območje histograma določeno z generično spremenljivko *hist_bit_num* in obsega intervale širine ena od 0 do $2^{hist\_bit\_num}$–1, medtem ko je območje vodila *data_in_reduced* med 0 in 2^{13}–1. Vrednost signala *data_in_mapped* izračunamo po sledeči enačbi:

$$data\_in\_mapped = \frac{2^{hist\_bit\_num}-1}{2^{13}-1} \cdot data\_in\_reduced \tag{5.1}$$

Ko je vhodni podatek urejen, preverimo, ali je komponenta omogočena. V načinu $MODE_3$ je za omogočanje *ADC_unit* odgovorna komponenta *PE_unit* (glej prilogo A). *PE_unit* preko vhoda *ADC_unit_feedback* omogoči komponento *ADC_unit* ter ji s tem sporoči, da je pripravljena na sprejem podatkov. Ob zagonu sistema je *ADC_unit* vedno omogočena. Nadaljnja obdelava se nadaljuje v sinhronem avtomatu stanj. Avtomat ima dve stanji, med katerima prehaja; *INTEGRATE* in *RESET*.

V stanju *INTEGRATE*, ki je privzeto stanje avtomata, se preko izhoda *INT_res* omogoči integrator na vezju za obdelavo analognega signala. Uporabnik pred zagonom sistema nastavi časovno okno integriranja, zato se v stanju *INTEGRATE* najprej preveri, ali je čas potekel. Če je pretekli čas integriranja še znotraj časovnega okna, se povečuje časovni števec, ki šteje preteklo število urinih ciklov od začetka integriranja. Ker poznamo periodo cikla, iz števca razberemo trenutni čas integriranja, tako da vrednost števca pomnožimo s periodo. Ko časovno okno integriranja poteče, se časovni števec ponastavi. Poleg tega se preveri, ali je zadnji zajet podatek *data_in_mapped* večji od uporabniško nastavljene pragovne vrednosti, pod katero se podatki ne shranjujejo. V kolikor je pogoj izpolnjen, se podatek priredi na izhodno vodilo *data_out* in avtomat preide v stanje *RESET*.

V stanju *RESET* komponenta *ADC_unit* iz AD pretvornika še vedno vsak urin cikel prejema nove podatke, vendar jih ne upošteva. Ko avtomat preide v to stanje, se prek izhoda *INT_res* omogoči praznjenje integratorja. Poleg tega se na izhodu *ADC_wr_en* pojavi pulz dolžine enega urinega cikla, ki v načinu $MODE_3$ omogoči vpis vrednosti na izhodu *data_out* v komponento *PE_unit*. Za-

tem se vrednost vodila *data_in_mapped* ponastavi na privzeto vrednost. Ker se kondenzator v integratorju ne more izprazniti hipno, komponenti *ADC_unit* v postopku programiranja nastavimo čas praznjenja kondenzatorja, ki je s simulacijo integratorja ter kasneje eksperimentalno določen na 300 ns. V stanju *RESET* se tako vsak urin cikel preveri, ali je nastavljen čas potekel. V kolikor je pretekli čas še znotraj nastavljenega časa, povečujemo števec urinih ciklov. Tako kot v stanju *INTEGRATE* tudi v stanju *RESET* prek števca izračunamo pretekli čas praznjenja kondenzatorja. Ko je praznjenje končano, avtomat čaka v stanju *RESET*, dokler detektor ne zazna novega žarka gama. Nov žarek gama sproži ponastavitev časovnega števca in prehod avtomata v stanje *INTEGRATE*. Stanji avtomata se izmenjujeta, dokler komponenta *PE_unit* ne onemogoči komponente *ADC_unit* ali uporabnik ne zaustavi sistema.

5.2.3.2 Komponenta *PE_unit*

PE_unit je komponenta, ki iz prejetih podatkov računa histogram. Hkrati je iz nje mogoče branje histograma s strani procesorja na sistemu Zynq. Vhode in izhode komponente opisuje tabela 5.2.

	Funkcija
Vhod	
clk	Signal ure.
PE_START	Uporabniško nastavljiv signal za zagon ali zaustavitev sistema.
write_A_M3	Signal, ki v načinu MODE_3 omogoči vpis podatka v *PE_unit*.
PE_RESET	Uporabniško nastavljiv signal za ponastavitev sistema.
data_in	Vhodno vodilo. Njegova vrednost predstavlja indeks stolpca histograma.
μC_wr_en	Signal s katerim procesor poda zahtevo za branje novega stolpca histograma.
Izhod	
data_out	Izhodno vodilo.
μC_rd_en_current	Signal, ki procesorju omogoči branje sprotnega histograma.
μC_rd_en_final	Signal, ki procesorju omogoči branje končnega histograma.

BUFF_addr	Naslovno vodilo za *BRAM_buffer*. Ni v uporabi v načinu $MODE_3$.
ADC_unit_RESET	Signal, ki v načinu $MODE_3$ onemogoči in ponastavi *ADC_unit*.
PE_wr_rd_en	Signal, ki omogoči branje iz komponente *BRAM_buffer* ali omgoči pošiljanje iz komponente *PE_unit*. Ni v uporabi v načinu $MODE_3$.

Tabela 5.2: Vhodi in izhodi komponente *PE_unit*.

Sestava komponente je razdeljena na dva dela, na pomnilniški in kontrolni. Pomnilniški del zajema dve DP-BRAM enoti, kamor se shranjuje histogram. Kontrolni del sestavlja sinhroni avtomat stanj, ki skrbi za pravilno delovanje pomnilniškega dela ter upravlja s kontrolnimi signali, ki so uporabljeni pri komunikaciji s procesorskim delom.

5.2.3.2.1 Pomnilniški del

5.2.3.2.1.1 Blok RAM z dvovratnim dostopom

DP-BRAM je pomnilniška enota, ki omogoča hkratno branje in pisanje podatkov. Enota ima dvoje vrat, označenih z A in B, ki so med seboj neodvisna. Vsaka vrata imajo svoj signal ure (*clk*), podatkovni vodili (*data_in* in *data_out*), naslovno vodilo (*addr*) ter kontrolna signala za omogočanje branja in vpisovanja podatkov (*bram_en* in *wr_en*). Globina in širina enote DP-BRAM sta v našem primeru določeni z generičnima spremenljivkama *bin_num* in *buffer_size*. Spremenljivka *bin_num* vsebuje informacijo o številu podatkov, ki jih enota lahko shrani, medtem ko *buffer_size* določa širino posameznega podatka v bitih. S spremenljivkama definiramo polje *bram* za hranjenje podatkov, tako kot je prikazano s sledečim izsekom iz kode [16]:

```
type BRAM_ARRAY is array (0 to (bin_num - 1))
                of unsigned (buffer_size downto 0);

shared variable bram: BRAM_ARRAY := (others => (others => '0'));
```

Vpisovanje v polje *bram* in branje iz njega (prek A vrat DP-BRAM enote) je prikazano s spodnjo programsko kodo. Do polja ima dostop oboje vrat.

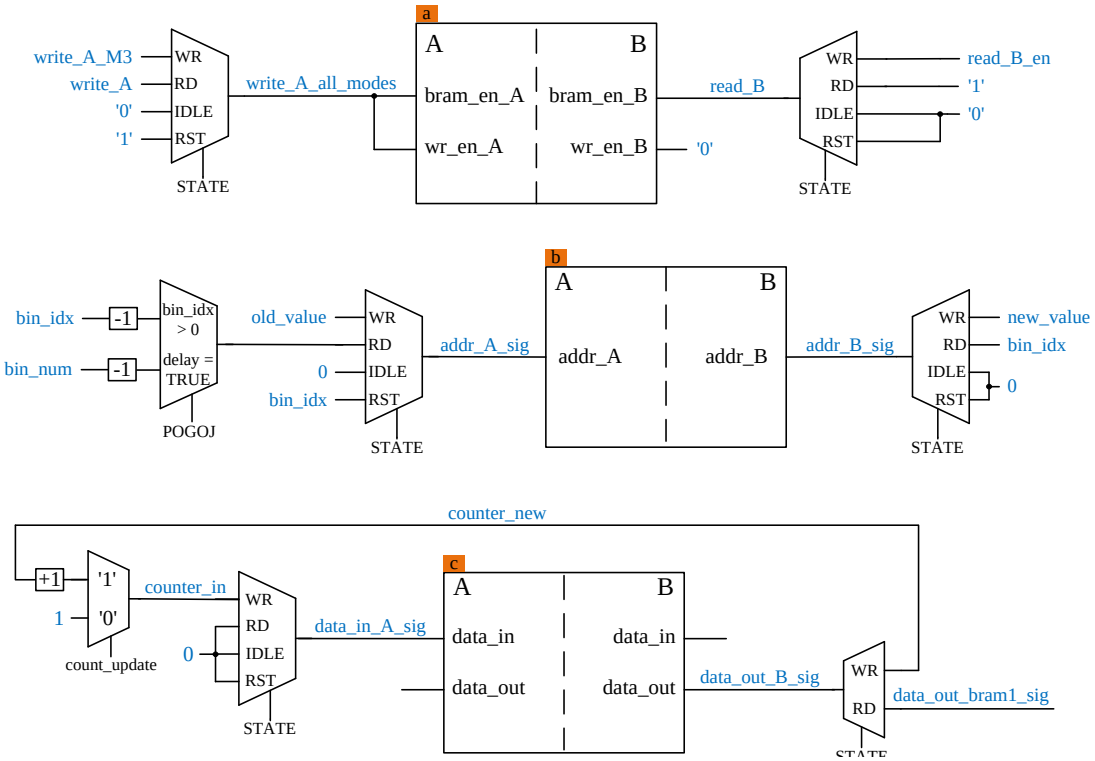
```
A_vrata: process (clk_A)
 begin
  if (rising_edge(clk_A)) then
   if (bram_en_A = '1') then  -- vrata A so omogocena
    if (wr_en_A = '1') then  -- vpis omogocen
     bram(to_integer(addr_A)) := data_in_A; --vpis podatka
    end if;
    data_out_A <= bram(to_integer(addr_A)); --branje podatka
   end if;
  end if;
 end process;
```

5.2.3.2.1.2 DP-BRAM in komponenta *PE_unit*

Komponenta *PE_unit* vsebuje dve DP-BRAM enoti za potrebe računanja, hranjenja in pošiljanja histograma v procesor . Obe sta uporabljeni tako, da prek A vrat poteka samo vpisovanje podatkov v enoto, medtem ko prek B vrat poteka samo branje podatkov iz enote. Poleg tega imata strani A in B obeh enot isti urin signal, kar pomeni, da sta vpisovanje in branje podatkov sinhrona.

DP-BRAM$_1$ je glavna enota, zadolžena za računanje in hranjenje histograma. Na sliki 5.3a so prikazane povezave kontrolnih signalov za omogočanje branja in pisanja podatkov, na sliki 5.3b je prikazana povezava naslovnih vodil ter na sliki 5.3c povezava podatkovnih vodil. Signali in vodila so označeni z modro barvo. Z njimi prek multiplekserjev upravlja kontrolni del komponente *PE_unit*, natančneje sinhroni avtomat stanj, podrobneje opisan v poglavju 5.2.3.2.2.

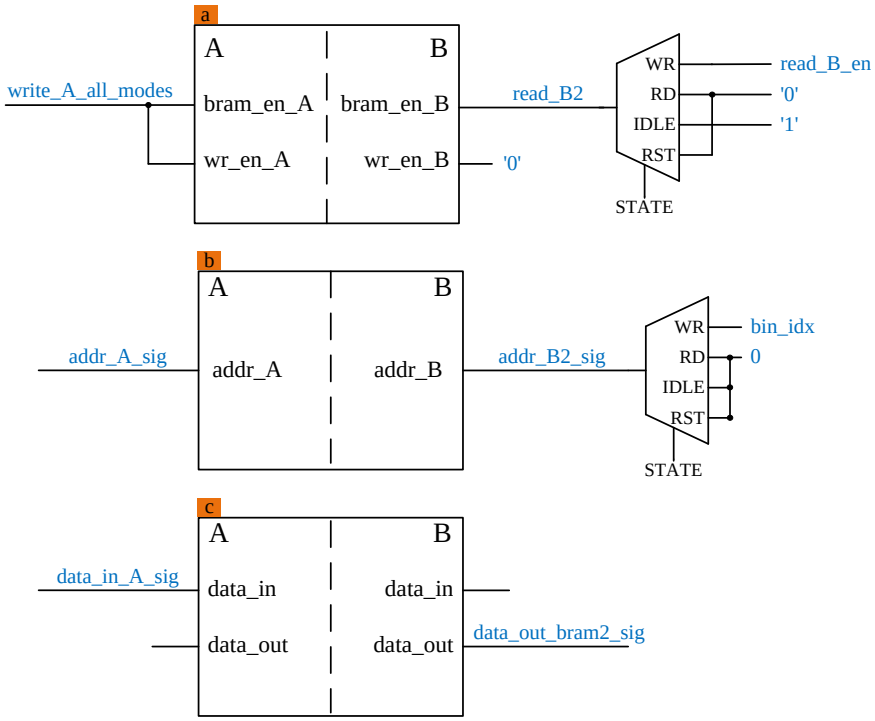
Med zajemanjem podatkov in računanjem histograma so zasedena oboja vrata enote *DP-BRAM*$_1$. Posledično v tem času procesor nima dostopa do podatkov v



Slika 5.3: Pomnilniška enota *DP-BRAM$_1$*.

enoti, torej ne more prebrati trenutnega histograma, ki je shranjen v njej. Ker mora biti sistem realnočasen, problem rešimo tako, da v komponento *PE_unit* dodamo drugo DP-BRAM enoto - *DP_BRAM$_2$*.

DP-BRAM$_2$ je dodatna enota, prikazana na sliki 5.4. Tako kot v enoti *DP-BRAM$_1$*, se tudi v *DP-BRAM$_2$* med zajemanjem podatkov shranjuje histogram, vendar enota ni vključena v samo računanje. To pomeni, da so B vrata enote prosta. Prek njih je procesorju omogočeno sprotno branje histograma, med tem, ko zajemanje in obdelava novih podatkov še potekata. Z modro označeni signali in vodila so tako kot pri *DP-BRAM$_1$* pod nadzorom sinhronega avtomata stanj, zato so opisani v poglavju 5.2.3.2.2.



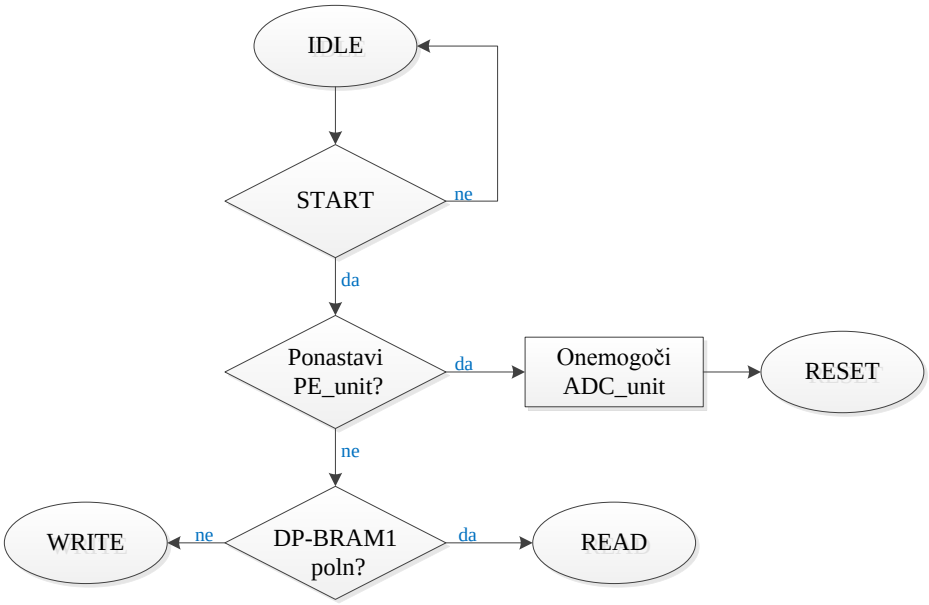
Slika 5.4: Pomnilniška enota *DP-BRAM₂*.

5.2.3.2.2 Kontrolni del

Kontrolni del skrbi za pravilno delovanje pomnilniškega dela ter za signale, ki so uporabljeni pri komunikaciji s procesorjem. Sestavlja ga sinhroni avtomat stanj, ki ima štiri stanja, med katerimi prehaja. To so *IDLE*, *WRITE*, *READ* in *RESET*.

5.2.3.2.2.1 Stanje *IDLE*

Stanje *IDLE*, prikazano na sliki 5.5, je privzeto stanje avtomata, kar pomeni, da se ob zagonu sistema komponenta *PE_unit* vedno nahaja v njem. Ob vsakem pozitivnem prehodu ure se najprej preveri, ali je uporabnik zagnal sistem. V kolikor je sistem zaustavljen, *PE_unit* ostaja v stanju *IDLE*. To pomeni, da imajo notranji signali komponente začetno privzeto vrednost, obe pomnilniški enoti pa sta prazni. Ko je sistem zagnan, se preveri, ali je s strani uporabnika prišla zahteva za ponastavitev. Ta pogoj v praksi ne pride v poštev takoj po



Slika 5.5: Algoritem stanja *IDLE*.

zagonu sistema, temveč kasneje, ko se avtomat vrača v stanje *IDLE* iz ostalih stanj. V primeru, da je pogoj izpolnjen, komponenta *PE_unit* v načinu $MODE_3$ preko izhoda *ADC_unit_RESET* onemogoči in ponastavi komponento *ADC_unit* ter preide v stanje *RESET*. V kolikor pogoj za ponastavitev sistema ni izpolnjen, se preveri, ali je pomnilniška enota $DP\text{-}BRAM_1$ polna. Glede na to, avtomat preide v enega izmed stanj *WRITE* ali *READ*.

Dogajanje v pomnilniškem delu komponente v stanju *IDLE* je razvidno iz slik 5.3 in 5.4. Konfiguracija enot $DP\text{-}BRAM_1$ in $DP\text{-}BRAM_2$ je sledeča:

$DP\text{-}BRAM_1$:

- Signala *write_A_all_modes* in *read_B* sta postavljena na logično '0', zato so oboja vrata enote onemogočena.
- Naslovni vodili *addr_A_sig* in *addr_B_sig* sta postavljeni na vrednost 0.
- Vhodno podatkovno vodilo *data_in_A_sig* je postavljeno na vrednost 0, vsa izhodna podatkovna vodila (*data_out_B_sig* ter *data_out_bram1_sig*) in števci (*counter_new* ter *counter_in*) imajo privzete vrednosti 0.

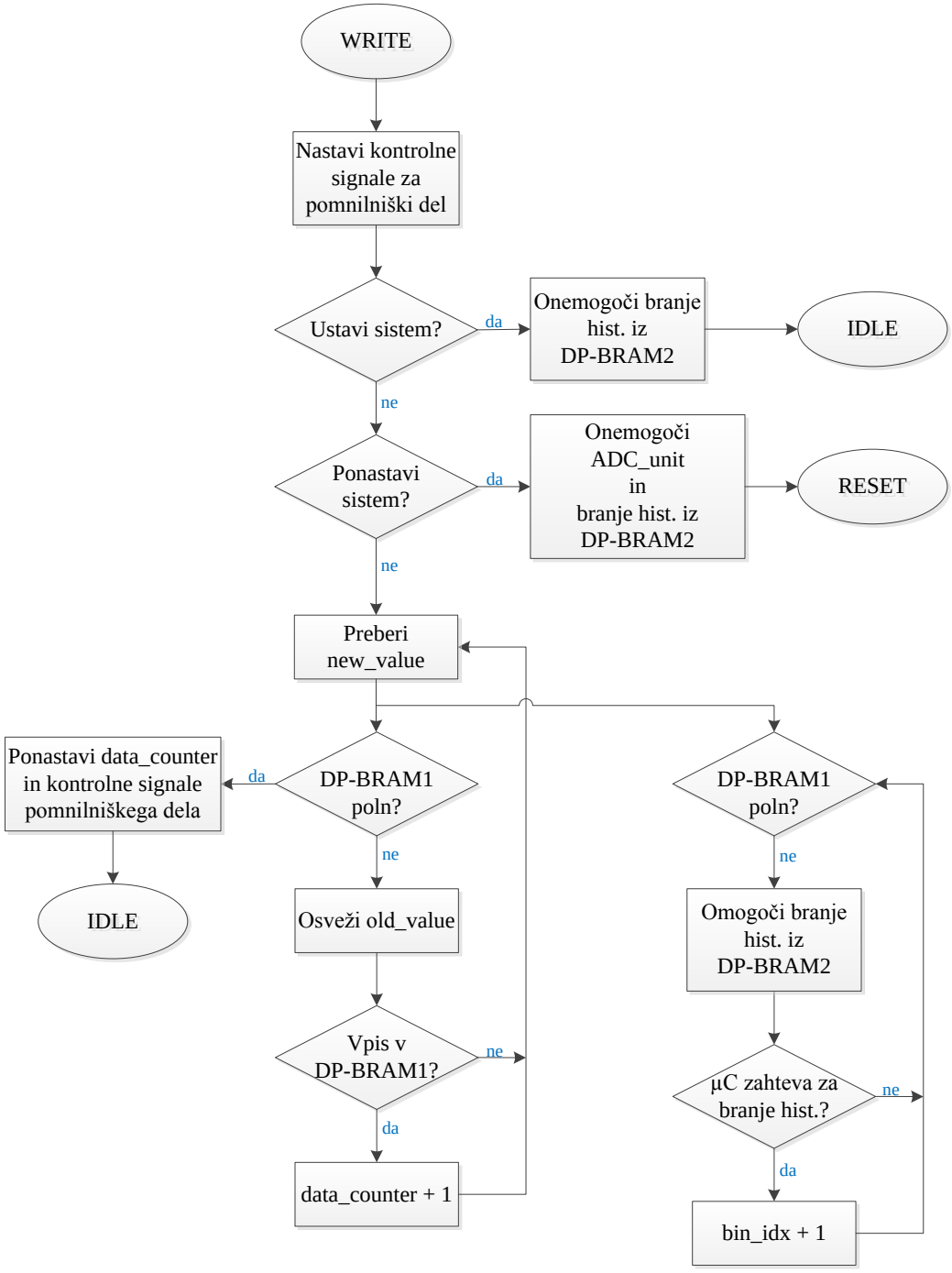
$DP\text{-}BRAM_2$:

- Vrata A so onemogočena, saj si enota kontrolni signal *write_A_all_modes* deli z *DP-BRAM$_1$*. Branje iz vrat B je omogočeno (signal *read_B2* ima logično vrednost '1'), kar nastavi vodilo *data_out_bram2_sig* na privzeto vrednost.
- Naslovni vodili sta postavljeni na vrednost 0.
- Vhodno podatkovno vodilo si enota deli z *DP-BRAM$_1$*, izhodno podatkovno vodilo *data_out_bram2_sig* ima vrednost 0.

5.2.3.2.2.2 Stanje *WRITE*

Stanje WRITE, prikazano na sliki 5.6, je stanje avtomata, v katerem v glavni pomnilniški enoti *DP-BRAM$_1$* poteka računanje in shranjevanje histograma. Dodatna enota *DP-BRAM$_2$* ima med tem vlogo, da sprotni histogram še med zajemanjem novih podatkov pošilja v procesor. Podatke v komponento *PE_unit* pošilja komponenta *ADC_unit*.

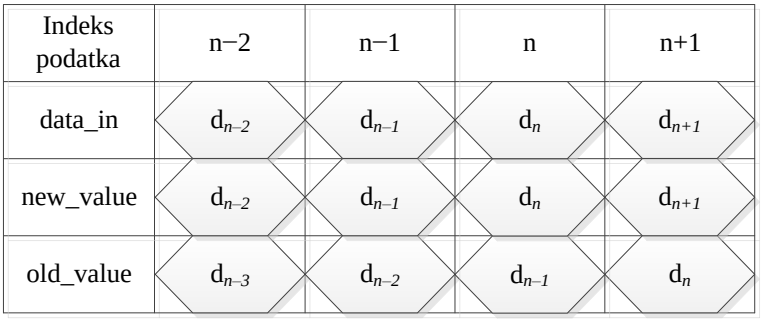
V stanju *WRITE* se najprej nastavijo kontrolni signali za obe pomnilniški enoti. To pomeni, da signal *count_update* iz slike 5.3c zavzame logično vrednost '1'. Multiplekser, ki ga krmili *count_update*, takoj po prehodu v stanje *WRITE* zagotovi veljavno vrednost števca *counter_in* in s tem signala *data_in_A_sig*. Števec *counter_new* je namreč nedefiniran eno periodo urinega signala po prehodu avtomata, zatem se mu priredi vrednost signala *data_out_B_sig*. Poleg signala *count_update* se nastavi signal *read_B_en*, ki omogoči branje podatkov iz B vrat obeh pomnilniških enot. Zatem se preveri, ali je s strani uporabnika prišla zahteva za zaustavitev ali ponastavitev sistema. V kolikor je sistem zaustavljen, se procesorju onemogoči sprotno branje histograma ter avtomat preide v stanje *IDLE*. Če uporabnik zahteva ponastavitev sistema, komponenta *PE_unit* onemogoči komponento *ADC_unit* in branje sprotnega histograma s strani procesorja. Za tem avtomat preide v stanje *RESET*. V nadaljevanju se algoritem stanja *WRITE* razdeli na dva dela. Vsak od delov skrbi za svojo pomnilniško enoto.



Slika 5.6: Algoritem stanja *WRITE*.

Kot je omenjeno v začetku poglavja, enota $DP\text{-}BRAM_1$ računa in shranjuje histogram. Zamislimo si, da komponenta *ADC_unit* v *PE_unit* pošlje podatek d_n.

Ta podatek se pojavi na vhodu *data_in* komponente *PE_unit* in v istem trenutku priredi vodilu *new_value*. Zatem se preveri, ali je enota *DP-BRAM1* polna in če pogoj ni izpolnjen, osvežimo vodilo *old_value*. Vodilo *old_value* vsebuje podatek d_{n-1}, to je podatek, poslan v enoto *PE_unit* tik pred d_n. Relacije med *data_in*, *new_value* in *old_value* prikazuje slika 5.7.



Slika 5.7: Relacije med vodili *data_in*, *new,_value* in *old_value*.

Sledi vpis v *DP-BRAM1*, ki ga prek vhoda *write_A_M3* s pulzom omogoči komponenta *ADC_unit*. Vhodni podatek je številka intervala oziroma indeks stolpca histograma, ki ga je potrebno povečati za 1 ter novo vrednost shraniti v pomnilnik. Sedaj poglejmo sliko 5.3 in se osredotočimo na vrata A. Razvidno je, da podatkovni signal *data_in_A_sig* tik pred prihodom podatka d_n v komponento *PE_unit* vsebuje informacijo o višini stolpca podatka d_{n-1}. Naslovnemu vodilu *addr_A_sig* je namreč prirejeno vodilo *old_value*. To pomeni, da se ob prihodu podatka d_n v *DP-BRAM1* na naslov *old_value* vpiše posodobljen stolpec vhodnega podatka d_{n-1}.

Poglejmo še dogajanje na vratih B. Ko v *PE_unit* pride podatek d_n, iz *DP-BRAM1* preberemo višino stolpca na naslovu *new_value* in jo priredimo vodilu *data_out_B_sig* ter naprej števcu *counter_new*. Višina stolpca se posodobi (poveča za 1) in priredi vodilu *counter_in* ter naprej vodilu *data_in_A_sig*. To pomeni, da je posodobljen stolpec podatka d_n pripravljen na vpis v *DP-BRAM1*, ko v komponento *PE_unit* pride podatek d_{n+1}.

Če povzamemo zgornjo razlago, histogram v enoti *DP-BRAM1* nastaja z enim podatkom zamika, glede na obdelavo podatkov v komponenti *ADC_unit*. Po končanem računanju histograma za podatek d_n, se poveča števec *data_counter*, ki šteje število podatkov, zapisanih v enoti *DP-BRAM1*. Ko je enota *DP-BRAM1*

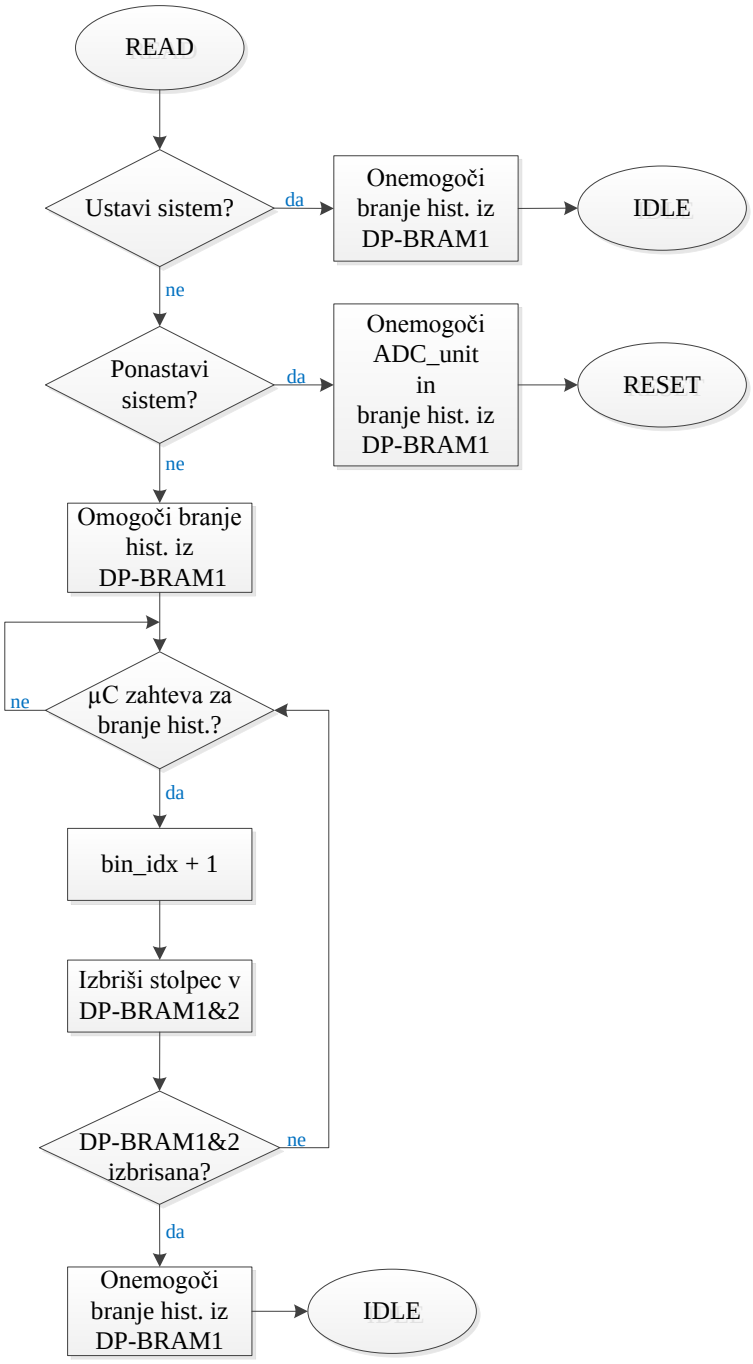
polna, kar pomeni, da vsebuje končni histogram 2^{buffer_size} podatkov, se števec *data_counter* ponastavi. Prav tako se ponastavijo kontrolni signali za obe pomnilniški enoti. Zatem avtomat preide v stanje *IDLE*.

Med tem, ko v *DP-BRAM₁* poteka računanje histograma, enota *DP-BRAM₂* skrbi, da je trenutni histogram dostopen procesorju. Iz slike 5.4 je razvidno, da ima enota na vratih A iste kontrolne signale ter naslovna in podatkovna vodila kot *DP-BRAM₁*, zato je histogram shranjen v obeh pomnilniških enotah. Prek izhoda *μC_rd_en_current* se procesorju omogoči sprotno branje histograma, poleg tega se izhodu iz komponente *PE_unit*, *data_out*, priredi izhodno podatkovno vodilo enote *DP-BRAM₂*, *data_out_bram2_sig*. Procesor za vsak stolpec histograma poda zahtevo za branje prek vhoda *μC_wr_en*. Ko je podatek prebran, se poveča indeks stolpca, zapisan v števcu *bin_idx*, ki je prirejen naslovnemu vodilu B vrat enote *DP-BRAM₂*, *addr_B2_sig* (slika 5.4b). Ob naslednjem pozitivnem prehodu urinega signala se tako na izhodu *data_out* pojavi nov stolpec histograma. Branje se začne s prvim stolpcem in konča z zadnjim. Število stolpcev je določeno z generično spremenljivko *bin_num*. Ko procesor prebere zadnji stolpec, se indeks ponastavi in branje novega sprotnega histograma se lahko začne. Branje se zaključi, ko se napolni enota *DP-BRAM₁* in avtomat preide v stanje *IDLE*.

5.2.3.2.2.3 Stanje *READ*

Stanje READ prikazuje slika 5.8. V njem procesor prebere končni histogram iz pomnilniške enote *DP-BRAM₁*. Hkrati se ponastavita obe enoti *DP-BRAM₁* in *DP-BRAM₂*, kar pomeni, da se pobrišejo vsi podatki znotraj pomnilniškega dela.

Na začetku se, tako kot v stanju *WRITE*, preveri, ali je uporabnik podal zahtevo za zaustavitev ali ponastavitev sistema. V primeru, da je sistem zaustavljen, se procesorju onemogoči branje končnega histograma ter avtomat preide v stanje *IDLE*. Če je prišla zahteva za ponastavitev sistema, komponenta *PE_unit* onemogoči komponento *ADC_unit* in zaustavi branje končnega histograma. Zatem avtomat preide v stanje *RESET*.



Slika 5.8: Algoritem stanja *READ*.

V primeru, da s strani uporabnika ni zahtev za zaustavitev ali ponastavitev sistema, se izvajanje algoritma stanja *READ* nadaljuje. Branje končnega histo-

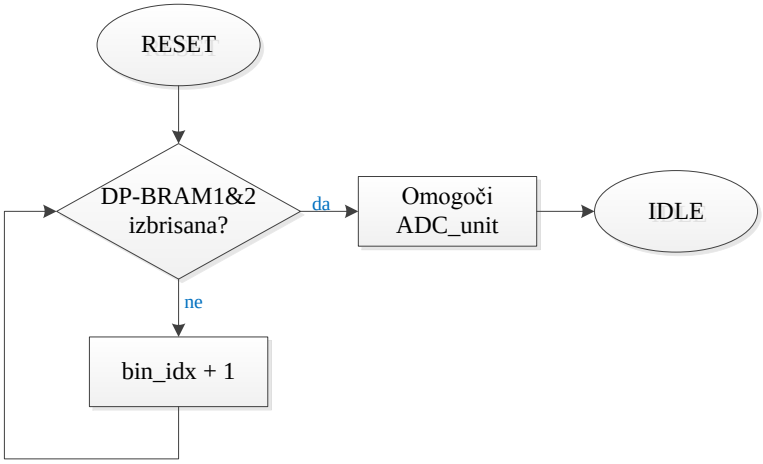
grama poteka podobno kot branje trenutnega histograma v stanju *WRITE*, z nekaj razlikami. Izhod komponente *PE_unit*, ki procesorju omogoči branje končnega histograma, je *μC_rd_en_final*, za razliko od branja sprotnega histograma, ki ga omogoči izhod *μC_rd_en_current*. Poleg tega procesor končni histogram bere iz B vrat pomnilniške enote *DP-BRAM₁*, medtem ko branje trenutnega histograma poteka iz B vrat enote *DP-BRAM₂*. V ta namen so v stanju *READ* B vrata enote *DP-BRAM₁* omogočena (slika 5.3a). Poleg tega je izhod komponente *PE_unit*, *data_out*, prirejen vodilu *data_out_bram1_sig* (slika 5.3c). Nasprotno so B vrata enote *DP-BRAM₂* onemogočena (slika 5.4a). Kot v stanju *WRITE*, tudi v stanju *READ* procesor za vsak stolpec histograma poda zahtevo za branje prek vhoda *μC_wr_en*. Ko je stolpec prebran, se poveča indeks, zapisan v števcu *bin_idx*, ki je prirejen naslovnemu vodilu B vrat enote *DP-BRAM₁*, *addr_B_sig*.

Med tem, ko procesor bere histogram z B vrat enote *DP-BRAM₁*, na A vratih obeh pomnilniških enot poteka brisanje shranjenega histograma, kar je enako vpisu ničel na vsa mesta *DP-BRAM₁* in *DP-BRAM₂*. Brisanje pomnilniških enot se omogoči s signalom *write_A*, ko procesor prebere prvi stolpec histograma (slika 5.3a). Iz slik 5.3b in 5.3c je razvidno, da se hkrati zgodita branje stolpca na naslovu *bin_idx* (vrata B) in brisanje oziroma vpis ničle v stolpec z naslovom *bin_idx−1* ali *bin_num−1* (vrata A). Naslov za vpis ničle na vratih A je odvisen od dveh pogojev. Dokler je *bin_idx* večji od 0, je naslov za vpis *bin_idx−1*. Ko procesor prebere zadnji stolpec z vrat A, se števec *bin_idx* ponastavi na vrednost 0. To pomeni, da zadnji stolpec histograma ostane nepobrisan. Da se pobriše še zadnji stolpec, se postavi zastavica (*ang.* flag) *delay*, ki za en urin cikel zamakne prehod avtomata iz stanja *READ* v stanje *IDLE*. V tem ciklu se na naslov *bin_num−1*, kjer se nahaja zadnji stolpec histograma, vpiše ničla, hkrati pa se procesorju onemogoči branje. Za tem avtomat preide v stanje *IDLE*.

5.2.3.2.2.4 Stanje *RESET*

Avtomat preide v stanje *RESET*, ko uporabnik prek vhoda *PE_RESET* poda zahtevo za ponastavitev sistema. Namen stanja je izbris podatkov iz pomnilniškega dela komponente *PE_unit*. Stanje *RESET* prikazuje slika 5.9. Z vsakim urinim ciklom se preveri, ali so podatki v enotah *DP-BRAM₁* in *DP-BRAM₂* izbrisani. Dokler pogoj ni izpolnjen, se povečuje števec *bin_idx*, ki, tako kot v stanjih *WRITE*

in *READ*, predstavlja indeks stolpca histograma. Ko sta obe pomnilniški enoti prazni, se omogoči komponento *ADC_unit* in avtomat preide v stanje IDLE.



Slika 5.9: Algoritem stanja *RESET*.

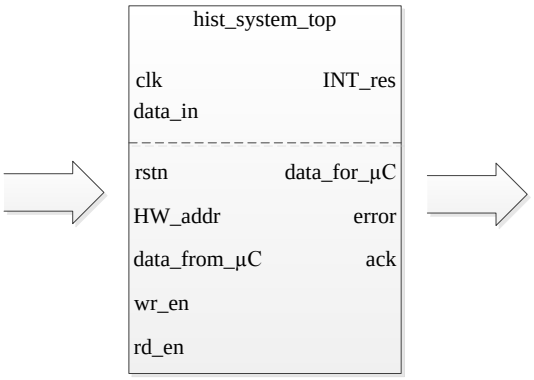
Iz slike 5.3 je podrobno razviden postopek brisanja pomnilniške enote *DP-BRAM₁*. Vrata B so v stanju *RESET* onemogočena, medtem ko so vrata A omogočena. Prek vrat A se na naslov *bin_idx* vsak urin cikel vpiše ničla in izbriše stolpec histograma. Števec *bin_idx* se povečuje in ko se na vseh *bin_num* stolpcih vpišejo ničle, je enota *DP-BRAM₁* prazna. Enako velja za enoto *DP-BRAM₂*.

5.2.3.3 Komunikacija s procesorjem

Procesor v sistemu Zynq s perifernimi enotami komunicira preko vodila AXI4. Njegova realizacija je dokaj kompleksna, zato obstaja vmesnik, ki AXI4 prevede na poenostavljeno vodilo in ga uporabimo za povezavo vrhovne komponente *hist_system_top* s procesorjem [17]. Povezavo prikazuje slika 5.10.

Slika 5.10: Povezava komponente *hist_system_top* s procesorjem.

Pri delu smo vmesnik obravnavali kot črno škatlo, kamor smo priklopili komponento *hist_system_top*, ki jo prikazuje slika 5.11. Puščici označujeta vhode in izhode komponente. Vhod *data_in* in izhod *INT_res* sta v notranjo-



Slika 5.11: Vhodi in izhodi komponente *hist_system_top*.

sti *hist_system_top* povezana na komponento *ADC_unit*, *clk* pa je signal ure, isti za cel sistem in komunikacijsko vodilo. Ostali vhodi in izhodi komponente, ki se nahajajo pod prekinjeno črto, so uporabljeni za komunikacijo s procesorjem. Opisuje jih tabela 5.3.

	Funkcija
Vhod	
clk	Signal ure.
rstn	Signal za ponastavitev vodila. Je aktiven v nizkem stanju.
HW_addr	32-bitno naslovno vodilo.
data_from_μC	32-bitno vodilo za vhodne podatke v komponento.
wr_en	Signal za omogočanje vpisa v komponento.
rd_en	Signal za omogočanje branja iz komponente.
Izhod	
data_for_μC	32-bitno vodilo za branje iz komponente.
error	Signal za javljanje napake v prenosu podatkov.
ack	Signal za potrditev prenosa podatkov.

Tabela 5.3: Vhodi in izhodi poenostavljenega komunikacijskega vodila.

Na začetku poglavja 5.2 je omenjeno, da vrhovna komponenta *hist_system_top* vsebuje dva procesa, ki realizirata logiko za komunikacijo s procesorjem. To sta procesa *REC_FROM_μC* in *SEND_TO_μC* (glej prilogo A).

Sinhroni proces *REC_FROM_μC* skrbi za branje podatkov iz vodila. Ob vsakem pozitivnem prehodu ure se preverita dve možnosti. Prva možnost je, da je vodilo ponastavljeno. To pomeni, da ima vhod *rstn* vrednost '0', ni sprejema podatkov in sistem znotraj komponente *hist_system_top* je zaustavljen. Druga možnost je, da želi procesor poslati podatke v komponento. V tem primeru je vrednost vhoda *wr_en* enaka '1' ter vrednost naslovnega vodila H*W_addr* enaka x"00000". Vrednost x"00000" je naslov komponente *hist_system_top*, ki ga označimo z *HIST_SYS_ADDR*. Posledično se podatki z vhodnega vodila *data_from_μC* vpišejo v komponento. Struktura podatkov znotraj *data_from_μC*, je podrobno opisana v poglavju 5.3.1.1.

Proces *SEND_TO_μC* izvaja pisanje podatkov na vodilo. Je asinhron, kar pomeni, da ni odvisen od signala ure. V primeru, da ima vhodno naslovno vodilo *HW_addr* vrednost *HIST_SYS_ADDR*, se podatki iz komponente *hist_system_top* vpišejo na izhodno podatkovno vodilo *data_for_μC*. V nasprotnem primeru, je vrednost na *data_for_μC* enaka nič. Struktura podatkov znotraj vrednosti *data_for_μC*, je opisana v poglavju 5.3.1.1. Proces skrbi tudi za izhod *ack*, ki javi potrditev prenosa. V primeru, da je eden od signalov, ki omogočita branje ali pisanje (*wr_en* ali *rd_en*) enak '1', zavzame *ack* vrednost '1'.

Vhod *rd_en* ima še dodatno vlogo. Predstavlja namreč zahtevo procesorja za branje podatkov iz komponente *hist_system_top*, med katerimi so glavni podatki o histogramu. V poglavju 5.2.3.2.2, kjer sta opisani stanji *WRITE* in *READ* sinhronega avtomata znotraj komponente *PE_unit*, zahtevo procesorja za branje predstavlja vhod *μC_wr_en*. V vrhovni komponenti *hist_system_top* zato vhod *rd_en* priredimo vhodu *μC_wr_en*.

5.2.3.4 Zasedenost FPGA vezja

Zasedenost FPGA vezja zaradi komponente *hist_system_top* je odvisna od načina delovanja sistema in nastavitve generičnih spremenljivk. Ker je za našo nalogo

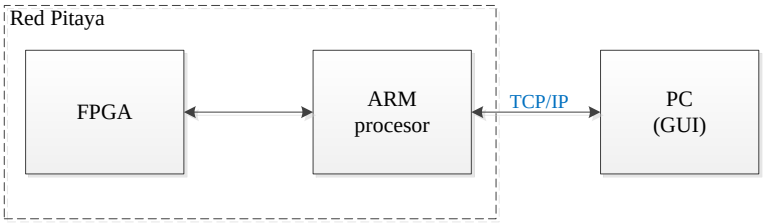
primeren samo način delovanja $MODE_3$, je v tabeli 5.4 podana ocena zasedenosti le za ta način.

Enota	**Število porabljenih enot**	**Delež porabljenih enot [%]**
LUT	600	3,41
FF	173	0,49
BRAM	2	3,33
DSP	1	1,25

Tabela 5.4: Zasedenost FPGA vezja zaradi komponente *hist_system_top*.

5.3 Procesorski del

Procesor na Red Pitayi ima vlogo posrednika med FPGA delom in osebnim računalnikom, kjer poteka prikaz rezultatov meritev. Shemo komunikacije prikazuje slika 5.12. Procesor bere histograme iz FPGA dela in jih pošilja v grafični uporabniški vmesnik (*ang.* GUI–Graphic User Interface), ki se nahaja na računalniku. Obratno uporabnik prek grafičnega vmesnika na računalniku pošilja ukaze v procesor, ki jih nato posreduje v FPGA.



Slika 5.12: Shema komunikacije med Red Pitayo in osebnim računalnikom.

Komunikacija med FPGA delom in vgrajenim ARM procesorjem je s strani FPGA opisana v poglavju 5.2.3.3, s strani procesorja pa pomeni branje in vpisovanje vrednosti v registre na določenem naslovu. Komunikacija Red Pitaya–PC je izvedena z internetnim povezovalnim protokolom TCP/IP (*ang.* TCP–Transmission Control Protocol, IP–Internet Protocol). Program, ki teče v procesorju, na Red Pitayi ustvari strežnik (*ang.* server), kamor se prek ethernet kabla poveže računalnik, ki ima vlogo odjemalca (*ang.* client).

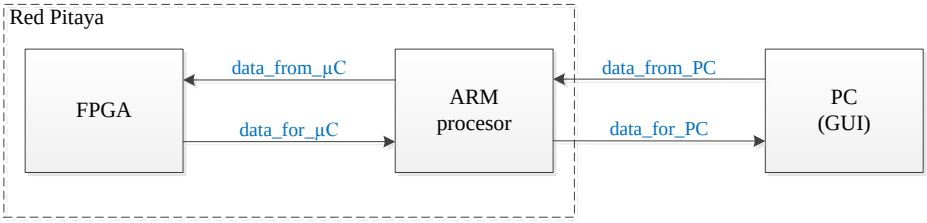
5.3.1 Zgradba programa

5.3.1.1 Podatkovna struktura in prenos podatkov

Procesor, kot posrednik med FPGA delom in računalnikom, upravlja s podatki, ki prihajajo in odhajajo iz dveh strani. V ta namen pri programiranju v jeziku C definiramo strukturo, ki vsebuje vse podatke o stanju sistema in prek katere poteka izmenjava informacij med FPGA in osebnim računalnikom. Definicija strukture je prikazana s spodnjo programsko kodo.

```
1  typedef struct
2  {
3    /* ****** PC --> procesor --> FPGA ****** */
4    uint32_t start; // zastavica za zagon sistema
5    uint32_t reset; // zastavica za ponastavitev sistema
6    uint32_t HW_config_flag; // zahteva za nastavitev parametrov
       FPGA vezja
7    uint32_t send_data_flag; // zahteva za posiljanje podatkov v PC
8    uint32_t disconnect_flag; // zahteva za prekinitev povezave
       procesor - PC
9    uint32_t threshold; // pragovna vrednost za komponento ADC_unit
10   uint32_t acq_window; // casovno okno zajemanja za komponento
       ADC_unit
11   uint32_t sw_trigger; // uporabnisko nastavljiva pragovna
       vrednost prozilca komponente ADC_unit
12
13   /* ****** FPGA --> procesor --> PC ****** */
14   uint32_t hist_data; // trenutni stolpec histograma
15   uint32_t UC_rd_en_current; // zastavica za sprotni histogram
16   uint32_t UC_rd_en_final; // zastavica za koncni histogram
17
18 } HW_system;
```

Med delovanjem sistema do strukture dostopajo funkcije, ki skrbijo za komunikacijo in vpisujejo ali berejo podatke. Podatki se na relacijah FPGA – procesor in procesor – PC prenašajo v obliki štirih 32-bitnih vrednosti *data_from_PC*, *data_for_PC*, *data_from_µC* in *data_for_µC*. Njihovo razporeditev prikazuje slika 5.13.



Slika 5.13: Prenos podatkov na relacija FPGA – procesor in procesor – PC.

Spodnje tri slike prikazujejo, kako so podatki razvrščeni znotraj 32-bitnih vrednosti.

- *data_from_PC:*

sw_trigger [31:27]	acq_window[26:19]	threshold [18:8]	neuporabljeno [7:5]	disconnect_flag [4:4]
send_data_flag [3:3]	HW_config_flag [2:2]	RESET [1:1]	START [0:0]	

Slika 5.14: *data_from_PC.*

- *data_for_PC* in *data_for_µC:*

HISTOGRAM [31:8]	neuporabljeno [7:2]	µC_rd_en_current [1:1]	µC_rd_en_final [0:0]

Slika 5.15: *data_for_PC* in *data_for_µC* .

- *data_from_µC:*

sw_trigger [31:27]	acq_window[26:19]	threshold [18:8]	neuporabljeno [7:2]	RESET [1:1]
START [0:0]				

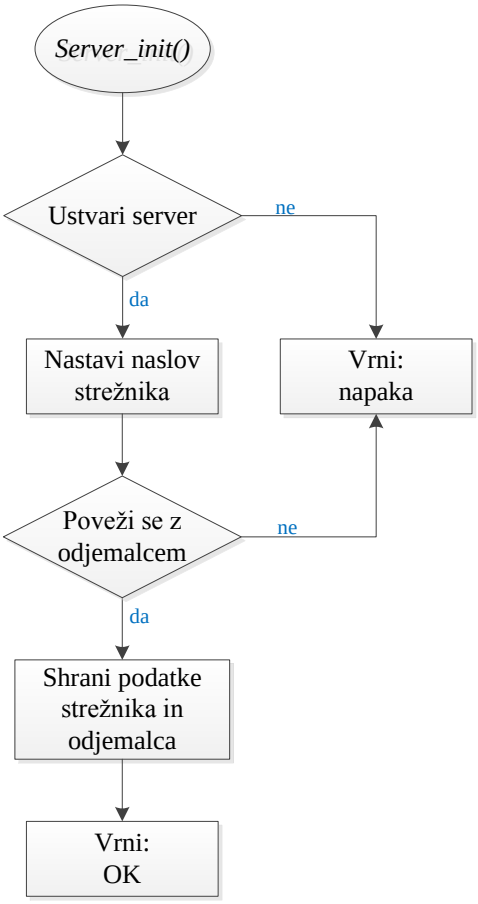
Slika 5.16: *data_from_µC* .

5.3.1.2 Komunikacija procesor – PC

Komunikacija med Red Pitayo in računalnikom poteka prek internetnega protokola TCP/IP; to je protokol, ki teče med strežnikom in odjemalcem. Na Red Pitayi teče operacijski sistem Linux, ki vsebuje programsko knjižnico *socket.h*, v kateri so funkcije za postavitev strežnika. Pri programiranju smo si pomagali z delujočim primerom, ki je prosto dostopen na spletu [18] [19].

5.3.1.2.1 Funkcija *Server_init()*

Funkcija *Server_init()* na Red Pitayi ustvari strežnik in se poveže z odjemalcem. Algoritem funkcije prikazuje slika 5.17. Strežnik ustvari funkcija *socket()* in zatem nastavi funkcija *setsockopt()*. Če pri tem pride do napake, se *Server_init()* konča in vrne napako. V naslednjem koraku se nastavi naslov strežnika. Zatem se strežnik poveže z odjemalcem, za kar skrbita funkciji *listen()* in *accept()*. V kolikor je povezava uspešna, se shranijo podatki o strežniku in odjemalcu ter *Server_init()* vrne status OK. Nasprotno funkcija vrne napako.

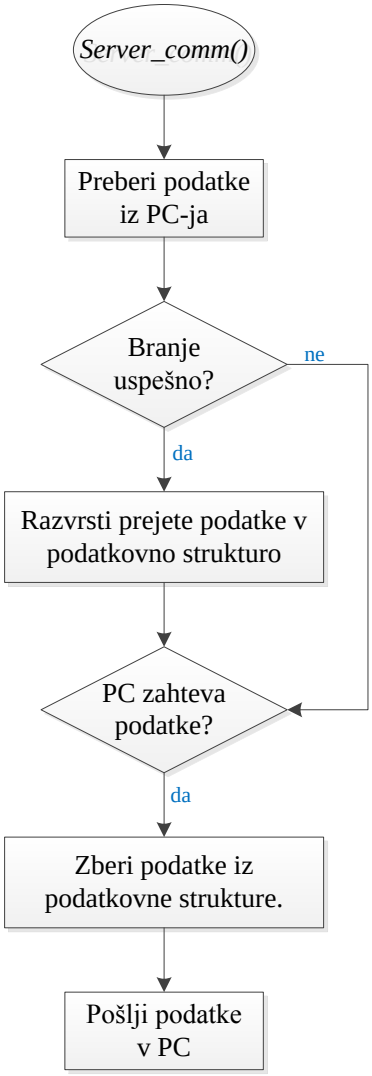


Slika 5.17: Algoritem funkcije *Server_init()*.

5.3.1.2.2 Funkcija *Server_comm()*

Funkcija *Server_comm()*, prikazana na sliki 5.18, skrbi za izmenjavo podatkov med Red Pitayo in računalnikom. Ob vsakem klicu se najprej izvede branje podatkov, ki jih je poslal računalnik. Računalnik pošlje 32-bitno vrednost *data_from_PC*. Branje podatkov izvede funkcija *recv()* in vrne število prejetih bajtov. Če so bili prejeti štirje bajti, pomeni, da je bilo branje uspešno in prejeti podatki se razvrstijo v podatkovno strukturo. V primeru, da podatkov ni ali so nepopolni, se razvrščanje v podatkovno strukturo ne izvede. Zatem se preveri, ali je s strani računalnika prišla zahteva za branje histogramov iz FPGA dela, kar pomeni, da ima zastavica *send_data_flag* v podatkovni strukturi vrednost ena. V kolikor je računalnik podal zahtevo, se ustrezni podatki iz podatkovne struk-

ture zberejo v spremenljivki *data_for_PC*. Na koncu se kliče funkcija *send()*, ki vrednost *data_for_PC* pošlje v računalnik.



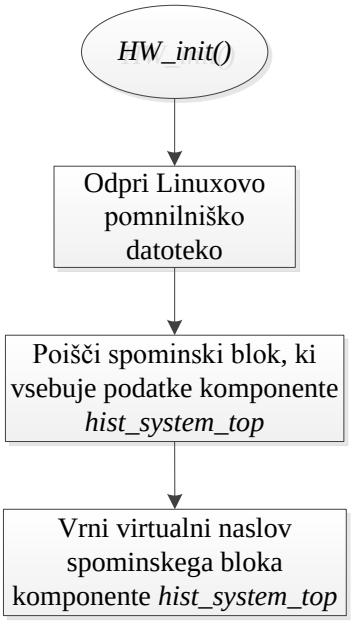
Slika 5.18: Algoritem funkcije *Server_comm()*.

5.3.1.3 Komunikacija procesor – FPGA

Algoritem za komunikacijo med procesorjem in FPGA delom temelji na programu monitor [20]. Monitor je program, napisan s strani razvijalcev Red Pitaye, ki ga kličemo v terminalu kadar želimo dostopati do poljubne lokacije v pomnilniku.

5.3.1.3.1 Funkcija *HW_init()*

Funkcija *HW_init()* locira in vrne naslov v pomnilniku Red Pitaye, kjer so zapisani podatki o FPGA komponenti za izračun histograma *hist_system_top*. Njen potek prikazuje slika 5.19. V operacijskem sistemu Linux, ki teče na Red Pitayi, se v mapi *root/dev* nahaja datoteka *mem*, ki vsebuje fizične naslove pomnilnika. Funkcija *HW_init()* zato na začetku s klicom funkcije *open()* odpre datoteko. V datoteki *mem* se nahaja naslov *HW_SYS_ADDR*, ki je fizični naslov komponente *hist_system_top*. V naslednjem koraku se kliče funkcija *mmap()*, ki preslika fizični naslov komponente v virtualnega in ga vrne kot rezultat. Virtualni naslov uporablja procesor za dostopanje do komponente in je hkrati tudi vrednost, ki jo vrne funkcija *HW_init()*.

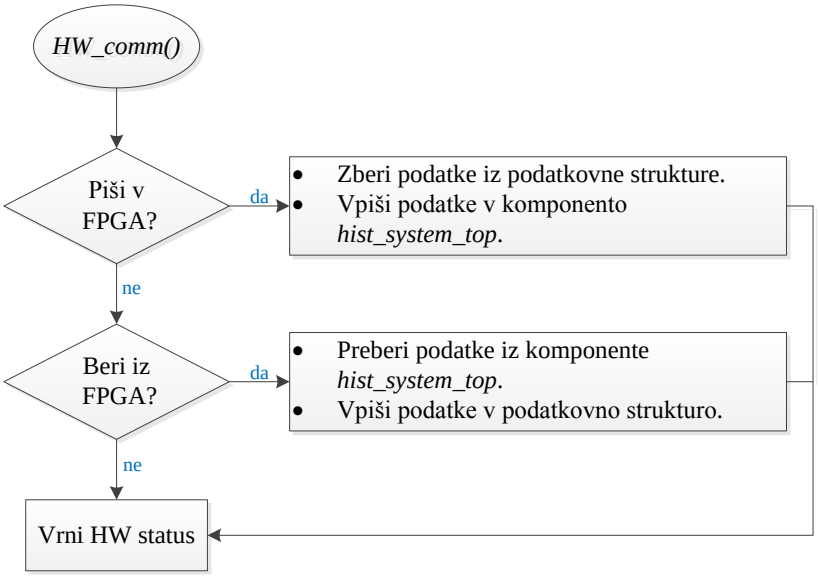


Slika 5.19: Algoritem funkcije *HW_init()*.

5.3.1.3.2 Funkcija *HW_comm()*

Funkcija *HW_comm()*, prikazana na sliki 5.20, skrbi za izmenjavo podatkov med procesorjem in FPGA komponento *hist_system_top*. Kot vhodni argument prejme virtualni naslov komponente, ki ga vrne funkcija *HW_init()*. Najprej se pre-

veri, ali je s strani računalnika prišla zahteva za nastavitev parametrov FPGA komponente, kar pomeni, da ima zastavica *HW_config_flag* v podatkovni strukturi vrednost ena. V kolikor je računalnik podal zahtevo, se ustrezni podatki iz strukture zberejo v spremenljivki *data_from_μC* in vpišejo na naslov komponente *hist_system_top*. V naslednjem koraku se preveri, ali je računalnik podal zahtevo za branje histogramov iz FPGA komponente. To pomeni, da ima zastavica *send_data_flag* v podatkovni strukturi vrednost ena. V tem primeru se iz naslova komponente *hist_system_top* prebere 32-bitno vrednost in shrani v spremenljivko *data_for_μC*. Na koncu se podatki iz *data_for_μC* razvrstijo v podatkovno strukturo. Funkcija *HW_comm()* vrne status, ki pove, ali je v danem klicu prišlo do vpisa ali branja podatkov.

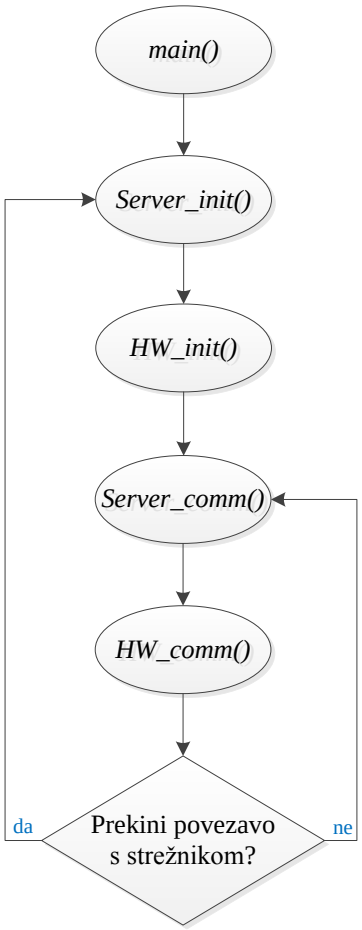


Slika 5.20: Algoritem funkcije *HW_comm()*.

5.3.1.4 Funkcija *main()*

Funkcija *main()* je glavna funkcija programa. Prikazana je na sliki 5.21. Ob zagonu programa se na Red Pitayi s klicem funkcije *Server_init()* vzpostavi strežnik. Program čaka znotraj funkcije, dokler odjemalec ne vzpostavi povezave. V naslednjem koraku se s funkcijo *HW_init()* locira naslov FPGA komponente *hist_system_top*. Zatem se neprestano kličeta funkciji *Server_comm()* in

HW_comm(), ki skrbita za komunikacijo z računalnikom in FPGA delom, dokler s strani računalnika ne pride zahteva za prekinitev povezave s strežnikom. V tem primeru zavzame zastavica *disconnect_flag* v podatkovni strukturi vrednost ena. Posledično se povezava strežnik – odjemalec prekine in podatkovna struktura se ponastavi. Nato se ponovno kliče funkcija *Server_init()*, ki spet čaka, da se odjemalec poveže s strežnikom.



Slika 5.21: Algoritem funkcije *main()*.

5.4 Zagon sistema na Red Pitayi

Postopek zagona sistema na Red Pitayi je sledeč:

1. Opis vezja za FPGA del Red Pitaye naredimo v okolju Vivado. Po sintezi in implementaciji generiramo nastavitveno datoteko (*ang.* bitstream) *ime_datoteke.bit*.
2. Program za procesorski del Red Pitaye sprogramiramo v programskem okolju SDK (*ang.* Software Development Kit), ki je del Vivada. Po postopku, ki program pretvori v strojno kodo (*ang.* build), se v lokalno mapo projekta shrani datoteka *ime_projekta.elf*.
3. Obe datoteki prek Ethernet povezave prenesemo v Red Pitayo s programom WinSCP [21].
4. Zaženemo terminalski program Putty [22], vpišemo uporabniško ime *"root"* in geslo *"root"* za Red Pitayo ter se postavimo v mapo, ki vsebuje prej preneseni datoteki *ime_datoteke.bit* in *ime_projekta.elf*.
5. FPGA sprogramiramo z ukazom *cat ime_datoteke.bit > /dev/xdevcfg*
6. Procesor sprogramiramo v dveh korakih:
   (a) Datoteko *ime_projekta.elf* naredimo izvršilno z ukazom *chmod +x ime_projekta.elf*
   (b) Program zaženemo z ukazom *./ime_projekta.elf*

5.5 Grafični uporabniški vmesnik

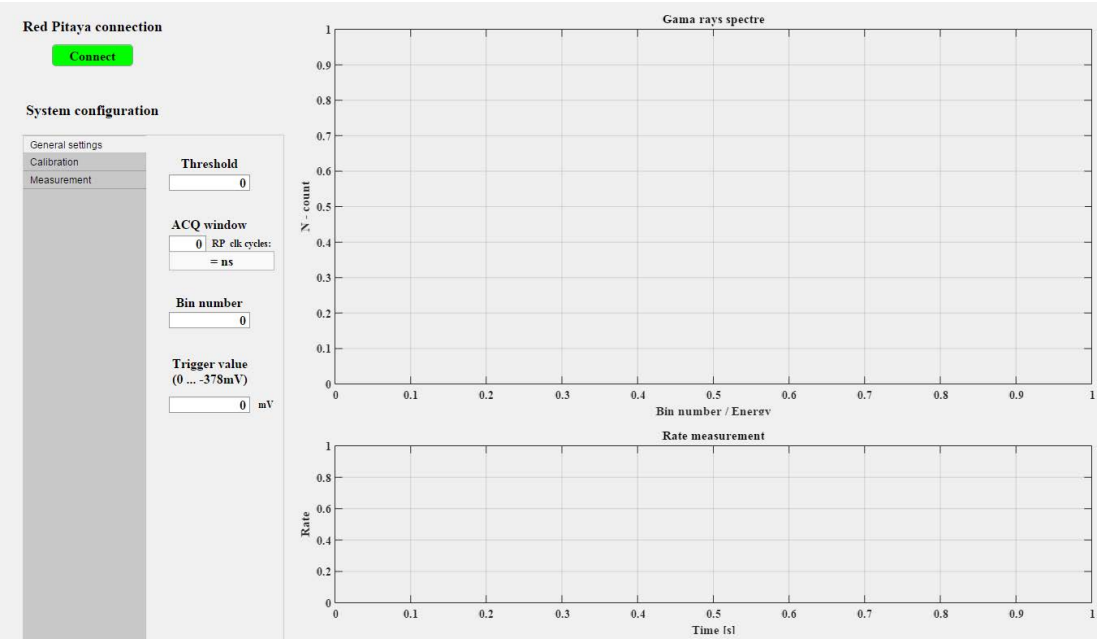
Grafični uporabniški vmesnik je program, ki teče na osebnem računalniku in omogoča interakcijo uporabnika s sistemom na Red Pitayi. Vmesnik ima dve funkciji. Prva je nastavitev parametrov delovanja sistema, druga pa prikaz rezultatov meritev in njihova nadaljnja obdelava. Za programiranje vmesnika smo uporabili programsko okolje App Designer, ki je vključeno v program Matlab

R2016a. Okolje ima dva načina, med katerima preklapljamo v postopku izdelave aplikacije. Prvi način se imenuje "Design View". V tem načinu izdelamo grafični del aplikacije, tako da komponente (tipke, meniji, grafi, oznake, itd.) iz vgrajene knjižnice z miško postavimo na želeno mesto na ekranu. Drugi način se imenuje "Code View". V tem načinu je s programsko kodo opisano delovanje komponent aplikacije. Vsaka komponenta ima svojo funkcijo, ki se izvede, ko pride do določenega dogodka (*ang.* event). Dogodek je na primer pritisk gumba ali vnos številske vrednosti v podatkovno okno.

5.5.1 Zgradba in delovanje vmesnika

5.5.1.1 Povezava z Red Pitayo

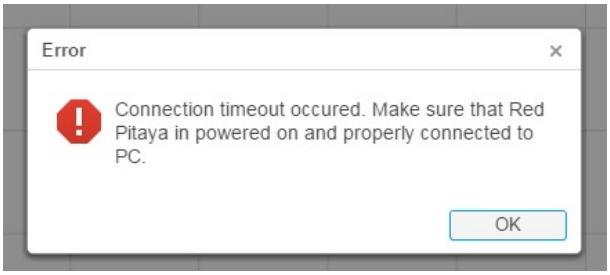
Grafični vmesnik je prikazan na sliki 5.22.



Slika 5.22: Grafični uporabniški vmesnik.

V levem zgornjem kotu se pod napisom "Red Pitaya connection"nahaja zelen gumb z napisom "Connect" za povezavo računalnika s strežnikom na Red Pitayi.

Če je po kliku na gumb povezava uspešna, se gumb obarva rdeče in napis se spremeni v "Disconnect". Ob ponovnem kliku se povezava z Red Pitayo prekine ter gumb zavzame svoj prvoten izgled. V kolikor je povezava neuspešna, se odpre pojavno okno, prikazano na sliki 5.23, ki javi napako.



Slika 5.23: Pojavno okno, ki se prikaže ob neuspešni povezavi.

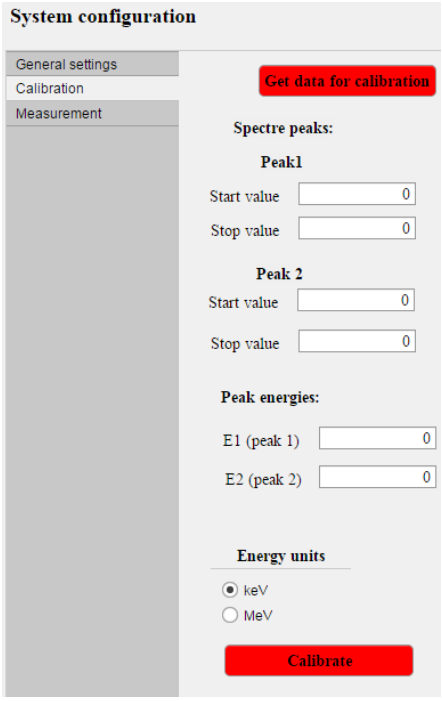
5.5.1.2 Nastavitev in upravljanje spektrometra

Na levi strani slike 5.22 se pod napisom "System configuration" nahaja okno z menijem, kjer izbiramo med tremi zavihki "General settings", "Calibration" in "Measurement". S klikom na zavihek se na desni strani okna odprejo različne možnosti za nastavitev sistema

Zavihek "General settings" je namenjen splošnim nastavitvam sistema. Omogoča nastavitev pragovne vrednosti ("Threshold"), s katero nastavimo mejni indeks stolpca v histogramu, ki ga še želimo shraniti in posledično mejno analogno vrednost napetosti na vhodu v AD pretvornik, pod katero sistem ne shranjuje meritev. Ostale nastavitve zajemajo nastavitev časovnega okna integriranja ("ACQ window"), števila stolpcev histograma ("Bin number") ter pragovne vrednosti prožilca ("Trigger value"), ki označuje prihod novega žarka gama. Pragovna vrednost, časovno okno integriranja in vrednost prožilca se ob zagonu sistema razvrstijo v 32-bitno vrednost *data_from_PC*, ki jo računalnik pošlje v Red Pitayo, medtem ko se mora vnesena vrednost "Bin number"ujemati z vrednostjo generične spremenljivke *bin_num*, uporabljene pri implementaciji vezja za računanje histogramov na FPGA.

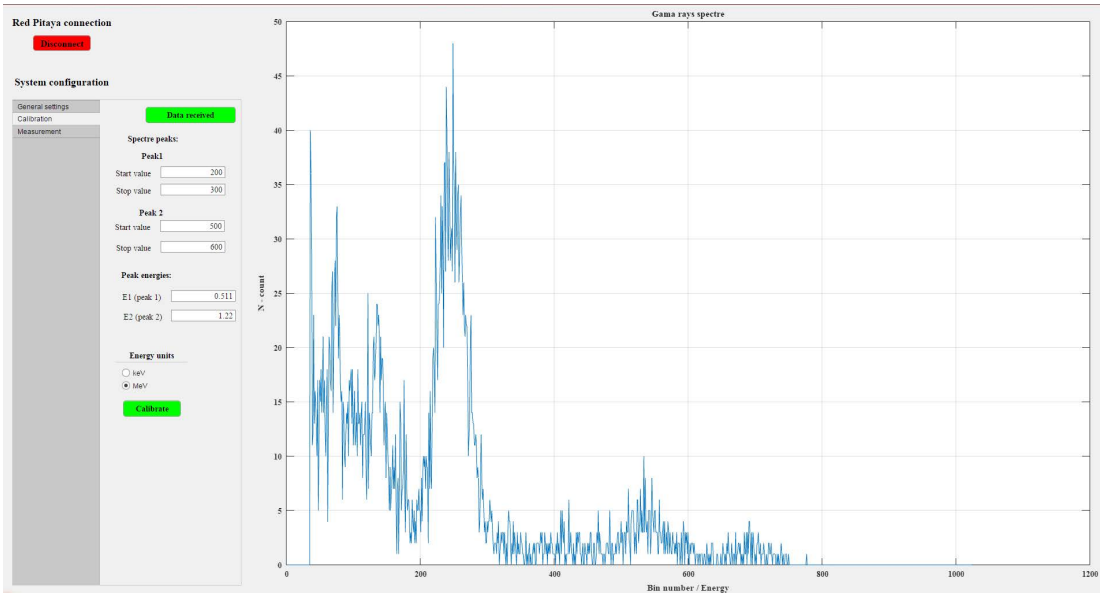
Naslednji zavihek je "Calibration", namenjen umeritvi spektrometra. Prikazan je na sliki 5.24. S klikom na gumb "Get data for calibration" za deset

sekund zaženemo sistem. Zatem računalnik prebere histogram iz Red Pitaye. Dobljeni histogram je spekter sevalca in prestavlja število razpadov v odvisnosti od indeksa stolpca. Ker je indeks stolpca digitalni ekvivalent filtrirane analogne napetosti V_γ na integratorju, je dobljeni spekter enak $N_\gamma(V_{\gamma_{DIG}})$ (poglavje 2.3.1). Spekter $N_\gamma(E_\gamma)$, ki predstavlja število razpadov v odvisnosti od energije, dobimo s kalibracijo spektrometra. Pod gumbom "Get data for calibration" se nahaja



Slika 5.24: Zavihek za kalibracijo.

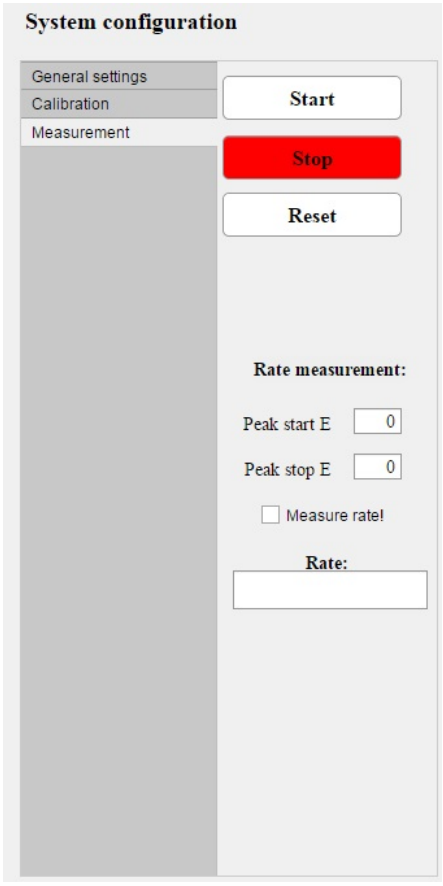
šest oken za vnos številskih vrednosti. Njihov pomen je razviden iz slike 5.25. Z začetno (*ang.* Start value) in končno vrednostjo (*ang.* End value) izberemo območji, znotraj katerih se na spektru nahajata značilna energijska vrhova sevalca. V našem primeru sta to dva fotovrha sevalca Na-22. Zatem v zadnji dve okni pod napisom "Peak energies" vnesemo znani energiji vrhov, izberemo enoto za energijo in s klikom na gumb "Calibrate" izvedemo kalibracijo. V postopku kalibracije se znotraj izbranih območij poiščeta indeksa stolpcev V_1 in V_2, kjer se nahajata maksimalni vrednosti razpadov. Iz znanih energij vrhov in vrednosti V_1 ter V_2 se nato po enačbi 2.15 izračuna umeritvena krivulja in posledično energija, ki jo predstavlja posamezen indeks stolpca v spektru, kar je opisano v poglavju 2.3.2.



Slika 5.25: Primer kalibracije spektrometra.

Zadnji zavihek je "Measurement", namenjen izvajanju meritev. Prikazan je na sliki 5.26. Na vrhu se nahajajo kontrolni gumbi za zagon (START), zaustavitev (STOP) in ponastavitev sistema (RESET). Na tem mestu je spektrometer kalibriran in izmerjeni spekter je $N_\gamma(E_\gamma)$. Med delovanjem sistema se spekter sevalca posodablja na približno vsake 2,1 sekunde. Ta čas je omejen s hitrostjo pošiljanja, branja in obdelave podatkov od detektorja do uporabniškega vmesnika in obratno. Sistem je mogoče zaustaviti ali ponastaviti ob poljubnem času. Ob ponastavitvi se trenutni spekter izbriše.

Pod kontrolnimi gumbi se nahaja del za nadzor merjenja razpadne hitrosti (*ang.* Rate measurement). Ker želimo razpadno hitrost meriti samo na določenem energijskem območju, v okni za vnos številskih vrednosti vpišemo začetno in končno energijsko mejo. Meritev vključimo tako, da odkljukamo okence zraven napisa "Measure rate!". Razpadna hitrost se iz spektra izračuna po enačbi 2.17 in rezultat se prikaže v oknu pod napisom "Rate:". Spreminjanje razpadne hitrosti v odvisnosti od časa spremljamo na spodnjem grafu iz slike 5.22.



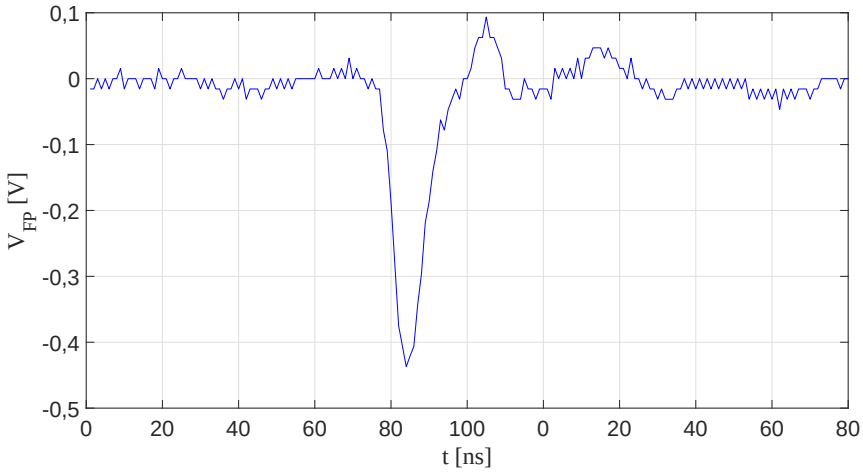
Slika 5.26: Zavihek za izvajanje meritev.

6 Rezultati

6.1 Pomembnejši analogni signali

6.1.1 Fotopomnoževalka

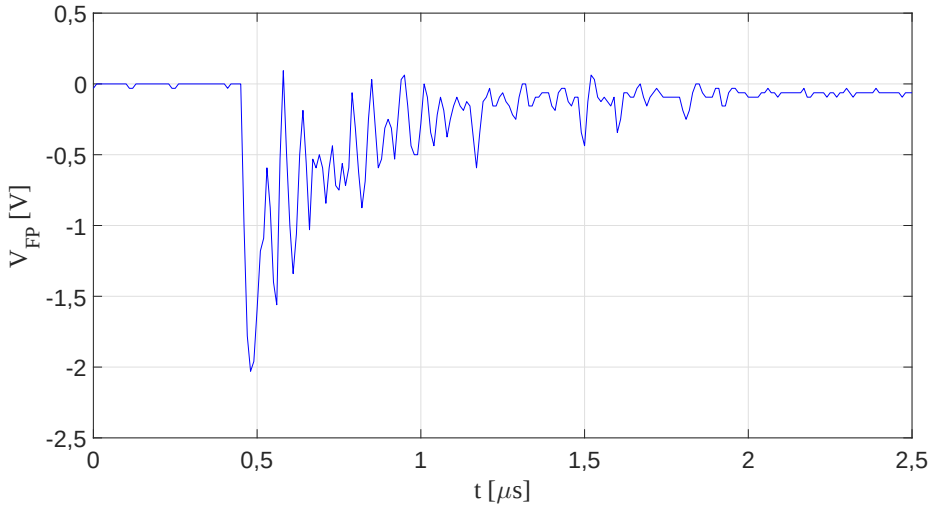
Slika 6.1 prikazuje ojačen izhodni signal iz fotopomnoževalke. Signal se generira na vezju za zajem analognega signala in po koaksialnem kablu potuje do vezja za obdelavo analognega signala, kjer smo ga izmerili. Prenihaji, ki sledijo impulzu in trajajo približno 30 ns, so posledica hitrega časa padanja impulza in niso povsem odpravljeni pri vezavi fotopomnoževalke. Poleg tega se, kljub 50-ohmski zaključitvi koaksialnega kabla, še vedno pojavijo majhni odboji signala.



Slika 6.1: Izhodni signal iz fotopomnoževalke.

Z večanjem časovne in napetostne skale je mogoče opazovati, kako se po vpadu žarka gama v scintilator, število izsevanih fotonov spreminja s časom. Glede na

teoretično ozadje, ki narekuje eksponentno padanje števila izsevanih fotonov po vpadu žarka gama, prikazano na sliki 2.10, pričakujemo, da višina napetostnih impulzov prav tako pada eksponentno. Rezultat meritve, prikazan na sliki 6.2, to potrjuje. Poleg tega iz časa trajanja odziva na zaznani žarek gama ocenimo čas integriranja signala.



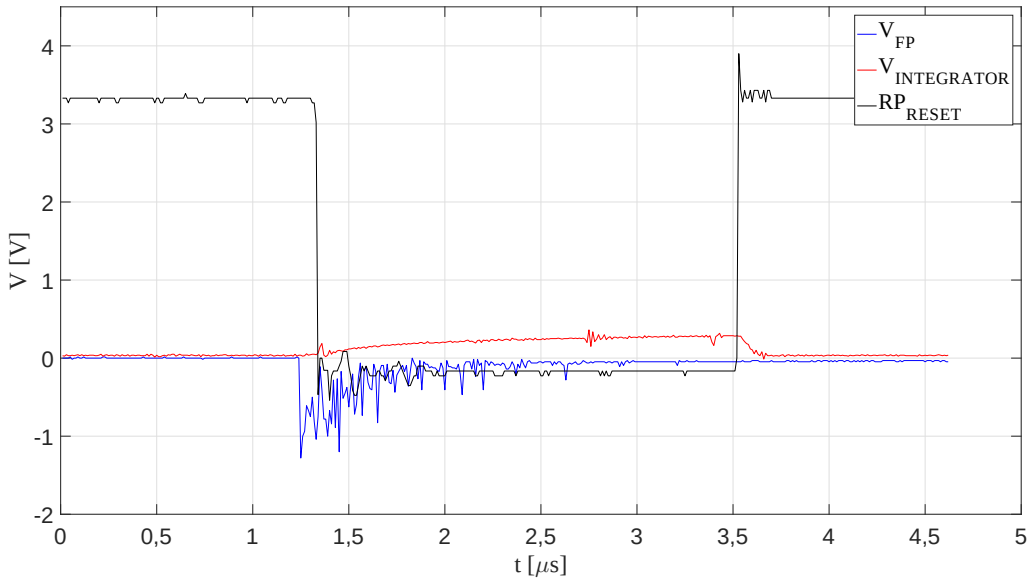
Slika 6.2: Časovna odvisnost števila izsevanih fotonov iz scintilatorja po vpadu žarka gama.

6.1.2 Ojačenje in integracija signala

Ojačenje signala iz fotopomnoževalke je opisano v poglavju 4.3.1. Signal se ojači prek dveh invertirajočih ojačevalnikov, kar prikazuje slika 4.14. Pri testiranju vezja je prišlo do močnega sinusnega nihanja obeh ojačevalnikov, ki ga nismo uspeli odpraviti. Z vzporedno vezavo kondenzatorjev s kapacitivnostjo od 5 pF do 20 pF z uporoma v povratni vezavi in serijsko vezavo uporov na izhodu obeh ojačevalnikov, nam je uspelo zmanjšati amplitudo nihanja. Kljub temu sta nestabilna ojačevalnika neuporabna, zato smo signal iz fotopomnoževalke ojačili z zunanjimi ojačevalniki v laboratoriju. Problema z nestabilnostjo ojačevalnikov nismo odpravili zaradi časovnega okvira za dokončanje magistrske naloge.

Ojačen signal iz fotopomnoževalke je nadaljnje povezan na integrator. Slika 6.3 prikazuje potek in rezultat integracije. Z modro barvo je označen odziv fotopomnoževalke na vpadni žarek gama, z rdečo filtriran izhod integratorja, povezan

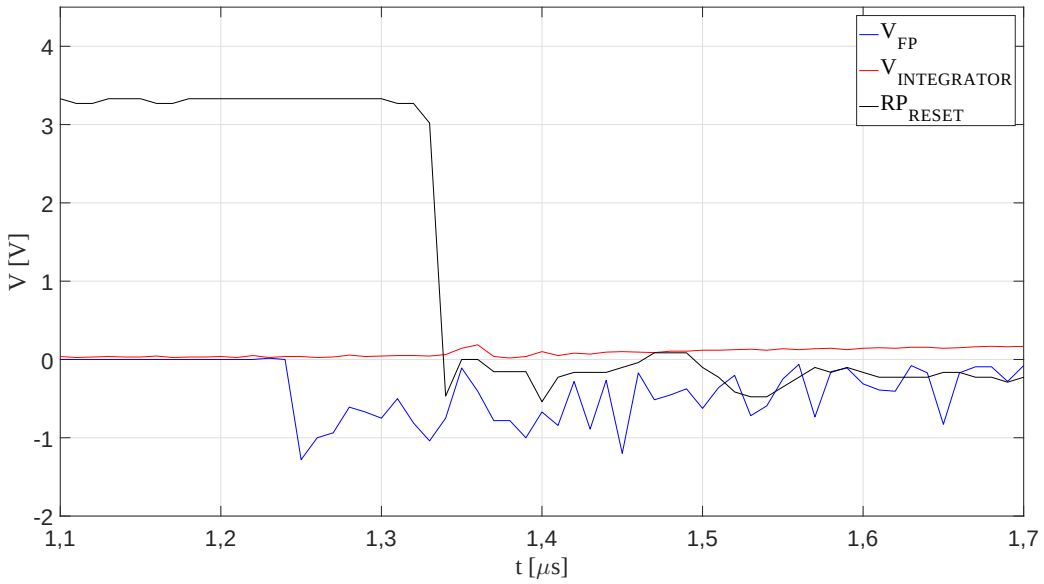
na analogno-digitalni pretvornik Red Pitaye in s črno kontrolni signal Red Pitaye za ponastavitev integratorja. Časovno okno integriranja je nastavljeno na 2,04 μs, prožilec za začetek integriranja pa na vrednost -300 mV.



Slika 6.3: Integracija signala iz fotopomnoževalke.

Podrobnejši pogled na začetek integriranja razkrije dogajanje, prikazano na sliki 6.4. Iz slike je razvidno, da se integriranje odziva na žarek gama prične s približno 100 nanosekundnim zamikom. To je posledica čakanja sistema, da signal iz fotopomnoževalke preseže uporabniško nastavljeno vrednost prožilca in prične z integriranjem. Zamik sam po sebi ne predstavlja problema, dokler je signal, ki ga integriramo, daljši od njega. Večji problem predstavlja dejstvo, da zamik ni konstanten. AD pretvornik, ki spremlja amplitudo napetostnih impulzov, signal vzorči s periodo 8 ns. Iz slike 6.1 je razvidno, da posamezen napetostni impulz traja približno 10 ns. To pomeni, da njegova amplituda lahko preseže nastavljeno vrednost prožilca integriranja, vendar sistem tega ne zazna, ker hitrost vzorčenja impulzov ni dovolj visoka. Posledično lahko za dva žarka gama z enako energijo pride do razlike napetosti na integratorju po koncu integriranja. Če je napetostna razlika dovolj velika, v spektru to zaznamo kot prihod dveh žarkov gama z različnima energijama. Nekonstanten zamik torej kvari ločljivost spektrometra. Problem bi lahko odpravili s hitrejšimi AD pretvorniki, vendar bi za to potrebovali drugo razvojno ploščo. AD pretvorniki na Red Pitayi namreč ne omogočajo

hitrejše periode vzorčenja od 8 ns.



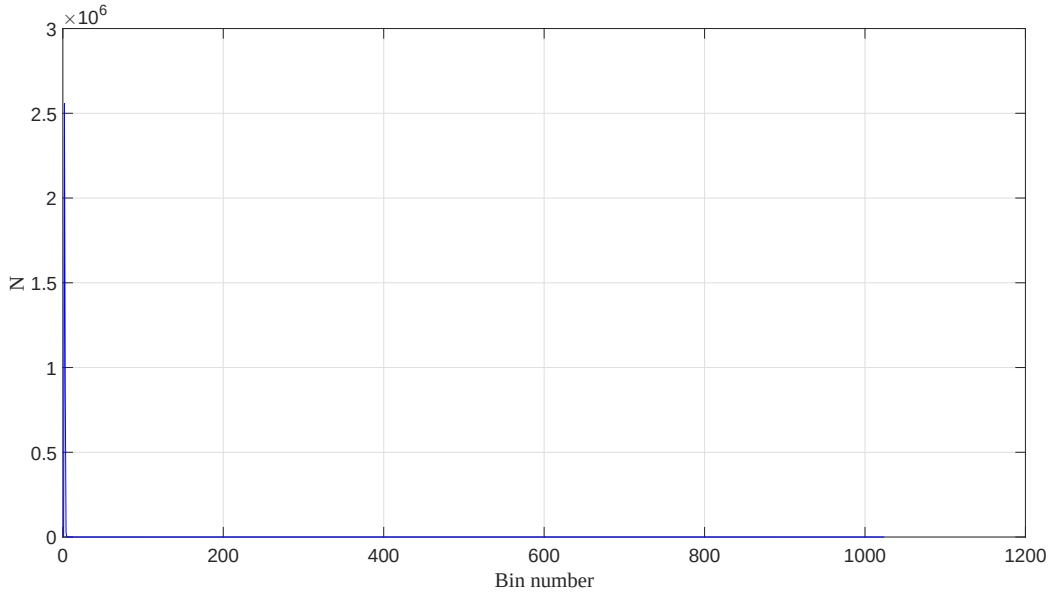
Slika 6.4: Zamik integriranja glede na odziv na vpadni žarek gama.

6.2 Meritev energijskega spektra in razpadne hitrosti

Slika 6.5 prikazuje rezultat meritve spektra izotopa natrija Na-22 pred umeritvijo spektrometra. Nastavitve sistema preko grafičnega vmesnika so sledeče:

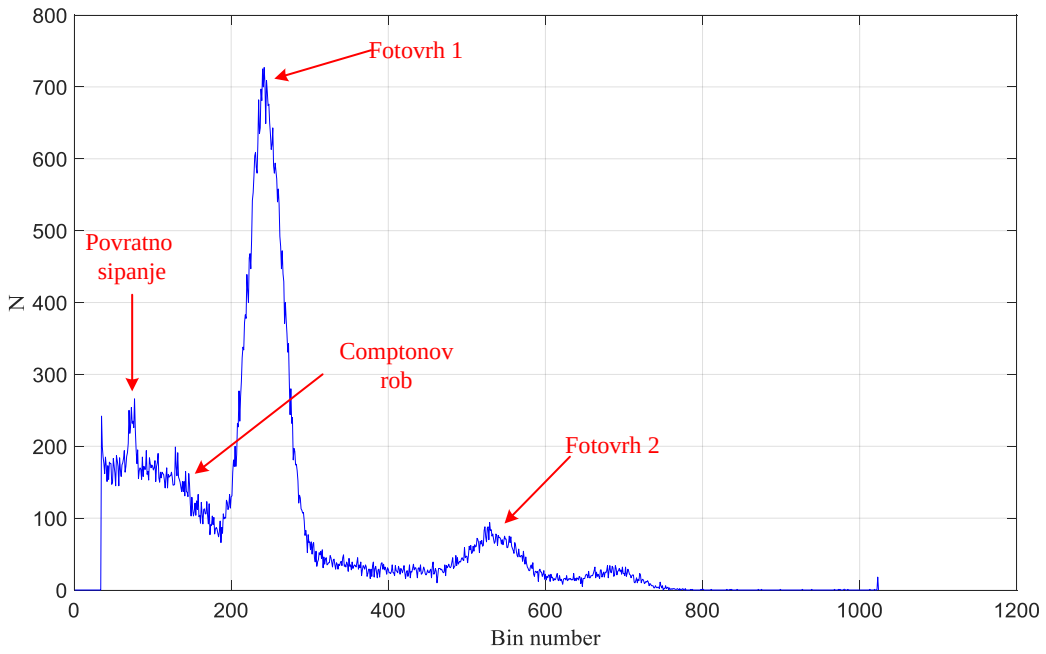
- Pragovna vrednost (Threshold): 1
- Časovno okno (ACQ window): 2,04 μs
- Število stolpcev histograma (Bin number): 1024
- Vrednost prožilca integriranja (Trigger value): -300 mV

Iz slike ni mogoče razbrati spektra, viden je le en vrh, ki se na abscisni osi nahaja med vrednostmi 1 in 34. Vrh ni posledica razpada gama, temveč predstavlja šum AD pretvornika, ki ni popolnoma umerjen, zato tudi v primeru, da ni vhodnega signala, na izhod postavi majhno digitalno vrednost. Prikaz šumnega vrha v spektru odpravimo z nastavitvijo pragovne vrednosti, pod katero sistem ne shranjuje meritev.



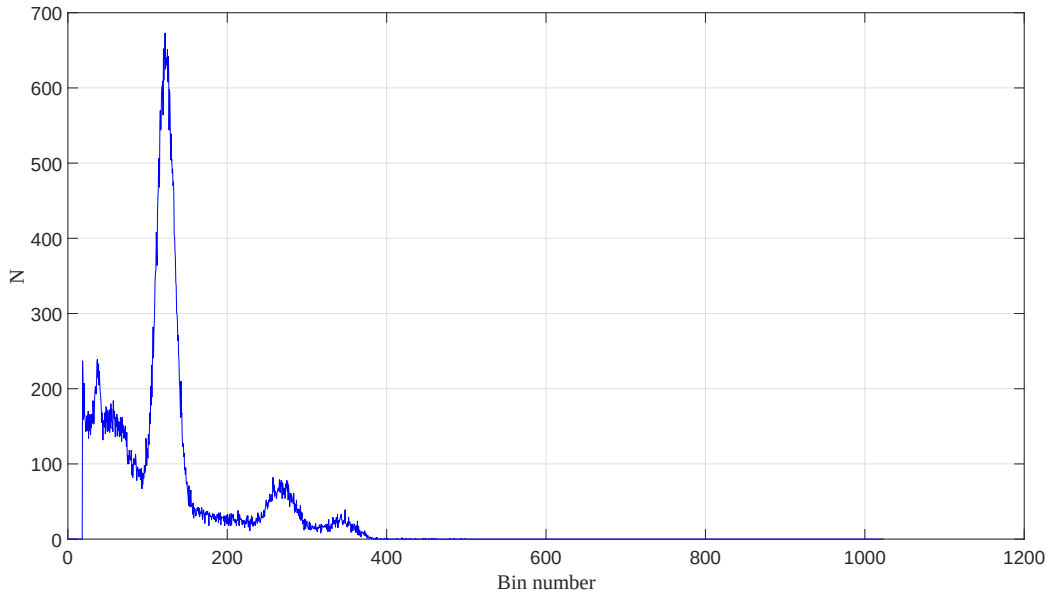
Slika 6.5: Spekter Na-22 pred umeritvijo z nastavljeno pragovno vrednostjo 1.

Spekter Na-22 s spremenjeno pragovno vrednostjo iz 1 na 35, prikazuje slika 6.6. Energijski vrhovi so vidni in označeni.



Slika 6.6: Spekter Na-22 pred umeritvijo z nastavljeno pragovno vrednostjo 35.

Spekter je mogoče s primerno nastavitvijo detektorja raztegniti ali skrčiti po abscisni osi in s tem spremeniti resolucijo detektorja ter energijsko območje, ki ga detektor lahko zazna. Na širino spektra vplivajo ojačenje signala iz fotopomnoževalke ter hitrost in časovno okno integriranja. Ojačenje signala je zaenkrat mogoče spreminjati samo z zunanjimi ojačevalniki, saj invertirajoča ojačevalnika na vezju za obdelavo analognega signala ne delujeta. Hitrost integriranja je mogoče nastaviti s spreminjanjem časovne konstante integratorja, kot je opisano v poglavju 4.3.2, medtem ko je časovno okno integriranja uporabniško nastavljivo v grafičnem vmesniku. Nastavljivo energijsko območje je v našem primeru pomembno, saj imajo sevalci, ki se uporabljajo pri izolirani perfuziji udov, približno petkrat manjšo energijo kot izotop natrija Na-22, s katerim smo testirali detektor. Kot primer je na sliki 6.7 prikazan spekter Na-22, kjer je hitrost integriranja zmanjšana na polovico v primerjavi s spektrom na sliki 6.6, vse ostale nastavitve pa so enake.

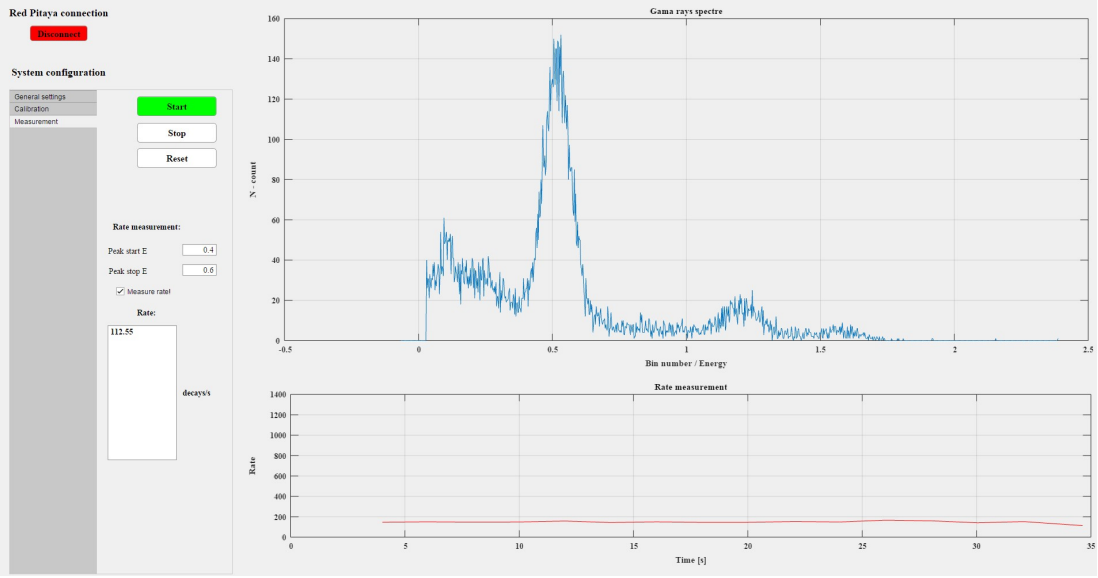


Slika 6.7: "Skrčeni" spekter.

Izračunali smo resoluciji spektrometra za oba spektra, ki znašata $R_1 = 27$ % in $R_2 = 26$ %, pri čemer je R_1 resolucija spektra iz slike 6.6 in R_2 resolucija "skrčenega" spektra iz slike 6.7. Rezultata sta si zelo podobna in znotraj napake zaradi zaokroževanja. Predpostavimo lahko, da hitrost integriranja ne vpliva močno na resolucijo, vendar bi bilo za zanesljive podatke potrebno izvesti več

meritev.

Spektrometer pred meritvijo razpadne hitrosti v realnem času kalibriramo. Ko je kalibracija končana, poženemo meritev, nastavimo energijsko območje, v katerem želimo meriti razpadno hitrost, in v grafičnem vmesniku označimo možnost "Measure rate!". Slika 6.8 prikazuje rezultat realnočasne meritve umerjenega spektra (zgornji graf) in razpadne hitrosti izotopa natrija (spodnji graf) z energijo žarkov gama med 0,4 MeV in 0,6 MeV, kar zaznamo v prvem fotovrhu.



Slika 6.8: Realnočasna meritev spektra in razpadne hitrosti Na-22.

7 Zaključek

V magistrski nalogi smo prototip spektrometra pripeljali do točke, kjer je primeren za prva resnejša testiranja. Iz fotopomnoževalke in scintilatorja smo sestavili scintilacijski detektor za zaznavanje sevanja gama. Analogne signale smo ustrezno obdelali in jih poslali v Red Pitayo, ki ima vlogo večkanalnega analizatorja. Večkanalni analizator je povezan na osebni računalnik, kjer smo pripravili grafični uporabniški vmesnik, ki omogoča nastavitev spektrometra in časovno odvisno spremljanje razpadne hitrosti sevalca pri izolirani perfuziji udov. Dodatna lastnost spektrometra je nastavljivo energijsko območje merjenja. Celoten sistem je majhen, kompakten in lahko prenosen.

Izdelava končnega prototipa presega okvir magistrske naloge, zato smo v nadaljevanju na kratko predstavili smernice za nadaljnje delo in nadgradnjo.

7.1 Nadaljnje delo in nadgradnja

Nadaljnje delo zajema:

- odpravo nihanja ojačevalnikov na vezju za obdelavo analognega signala, ki onemogoča ojačenje signala iz fotopomnoževalke,
- izdelavo ohišja detektorja in ščita pred žarki gama, ki prihajajo s strani,
- posodobitev uporabniškega vmesnika, da poleg prikaza razpadne hitrosti omogoča tudi prikaz uhajanja zdravila v sistemski krvni obtok,
- izdelavo skripte za samodejni zagon programa na Red Pitayi,
- različne teste delovanja spektrometra. Sem spadajo:

- test ščitenja detektorja pred žarki gama, ki na detektor vpadajo s strani,
- test linearnosti odziva spektrometra na sevanje gama,
- meritev odvisnosti ločljivosti in energijskega območja spektrometra od ojačenja signala ter hitrosti in časovnega okna integriranja,
- test delovanja s sevalci, ki se uporabljajo pri izolirani perfuziji udov ter določitev kritične razpadne hitrosti, ki sproži alarm za prekinitev posega in
- primerjava delovanja našega spektrometra s spektrometrom, ki se trenutno uporablja na Onkološkem inštitutu UKC Ljubljana.

Možna nadgradnja spektrometra vključuje priklop manjšega zaslona na dotik na Red Pitayo za prikaz grafičnega uporabniškega vmesnika in interakcijo uporabnika s spektrometrom. S tem dosežemo, da sistem ni več odvisen od računalnika in s tem še bolj prenosen in kompakten, vendar bi bilo v tem primeru potrebno izdelati nov uporabniški vmesnik.

Literatura

[1] S. Kuhar in M. Hočevar, "Izolirana ekstremitetna perfuzija," *Zdravniški vestnik*, vol. 85, no. 1, str. 33–40, 2016.

[2] Nanoxyde, "Human body." Dosegljivo: https://commons.wikimedia.org/wiki/File:Human_body_front_and_side.svg. Tip licence Creative Commons: priznanje avtorstva + deljenje pod istimi pogoji [Dostopano: 24.8.2019].

[3] Nuclear Power, "Interaction of gamma radiation with matter [Online]." Dosegljivo: https://www.nuclear-power.net/nuclear-power/reactor-physics/interaction-radiation-matter/interaction-gamma-radiation-matter/. [Dostopano: 15.7.2019].

[4] R. Pestotnik, "Spektrometrija žarkov γ s scintilacijskim spektrometrom [Online]." Dosegljivo: https://www.fmf.uni-lj.si/~jazbinsek/Praktikum5/Spektrometrija_zarkov_gama.pdf. [Dostopano: 15.7.2019].

[5] M. Urbanč, "Spektrometrija gama," seminar pri predmetu moderna fizika, Fakulteta za matematiko in fiziko, jan. 2012. Dosegljivo: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=2ahUKEwjmps2JyqfkAhUi_SoKHW6lBHgQFjAAegQIAxAC&url=https%3A%2F%2Ffiz.fmf.uni-lj.si%2F~zgonik%2FModernaFizika%2FSEMINARJI%2FSeminar%2520Moderna%2520fizika%2520SPEKTROMETRIJA%2520GAMA.docx&usg=AOvVaw0VWjE2qyKNT6yBQmIAe0HJ [Dostopano: 15.7.2019].

[6] Saint-Gobain Crystals, "Lyso scintillation material [Online]." Dosegljivo: https://www.crystals.saint-gobain.com/sites/imdf.crystals.com/files/documents/lyso-material-data-sheet.pdf. [Dostopano: 1.9.2019].

[7] Nuclear Power, "Shielding of gamma radiation [Online]." Dosegljivo: `https://www.nuclear-power.net/nuclear-power/reactor-physics/atomic-nuclear-physics/radiation/shielding-of-ionizing-radiation/shielding-gamma-radiation/`. [Dostopano: 20.7.2019].

[8] National Institute of Standards and Technology, "X-ray mass attenuation coefficients – lead." Dosegljivo: `https://physics.nist.gov/PhysRefData/XrayMassCoef/ElemTab/z82.html`. [Dostopano: 7.8.2019].

[9] M.T. Sapienza, G.C. Campos-Neto, M.A.C. Oliveira, E. Akaishi, T. Watanabe, C.R. Ono, P.L.A. Costa, C.J. Rodrigues, A.J. Rodrigues Jr. in C.A Buchpiguel , "Leakage monitoring during isolated limb perfusion: Comparison of two radioisotopic techniques in a patient with malignant melanoma," *World Journal of Nuclear Medicine*, vol. 4, no. 3, str. 108–110, 2005.

[10] Erik Margan, Inštitut Jožef Stefan, "Električna shema za visokonapetostno napajanje fotopomnoževalke." privatna komunikacija 2019.

[11] Marko Jankovec, Fakulteta za elektrotehniko UL, "Construction of electronic systems, pcb design for emc." Skripta pri predmetu Konstruiranje elektronskih naprav.

[12] Samo Korpar, Inštitut Jožef Stefan, "Električna shema za priključitev fotopomnoževalke hamatsu-r5900." privatna komunikacija 2019.

[13] Hamatsu Photonics K.K, *Photomultiplier Tubes – Basics and Applications*. Hamatsu, 3 izd., August 2007.

[14] R. Kastner, J. Matai in S. Neuendorffer, *Parallel Programming for FPGAs*, ch. 8. 2018. Dosegljivo: `https://arxiv.org/pdf/1805.03648.pdf`.

[15] Red Pitaya in Andrej Trost, "Redpitaya2017." Dosegljivo: `lniv.fe.uni-lj.si/courses/div/RedPitaya2017.zip`. [Dostopano: 25.8.2018].

[16] M. Možek, "Načrtovanje digitalnih vezij." Prosojnice predavanj pri predmetu Načrtovanje digitalnih vezij, dosegljivo na naslovu: `http://ndv.fe.uni-lj.si/predmet/predavanja/NDV_Prosojnice.pdf`.

[17] A. Trost, "Skaliranje podatkov z nasičenjem." Navodila za vaje pri predmetu Digitalna integrirana vezja in sistema, dosegljivo na naslovu: `https://lniv.fe.uni-lj.si/courses/div/divs17_lab7.pdf`.

[18] A. Potočnik, "Red pitaya fpga project 5 – high-bandwidth averager [Online]." Dosegljivo: `http://antonpotocnik.com/?p=514765`. [Dostopano: 22.12.2018].

[19] A. Potočnik, "Red pitaya fpga project 5 – high-bandwidth averager [Online]." Dosegljivo: `https://github.com/apotocnik/redpitaya_guide/blob/master/projects/5_averager/server/server.c`. Programska koda na GitHub-u, [Dostopano: 22.12.2018].

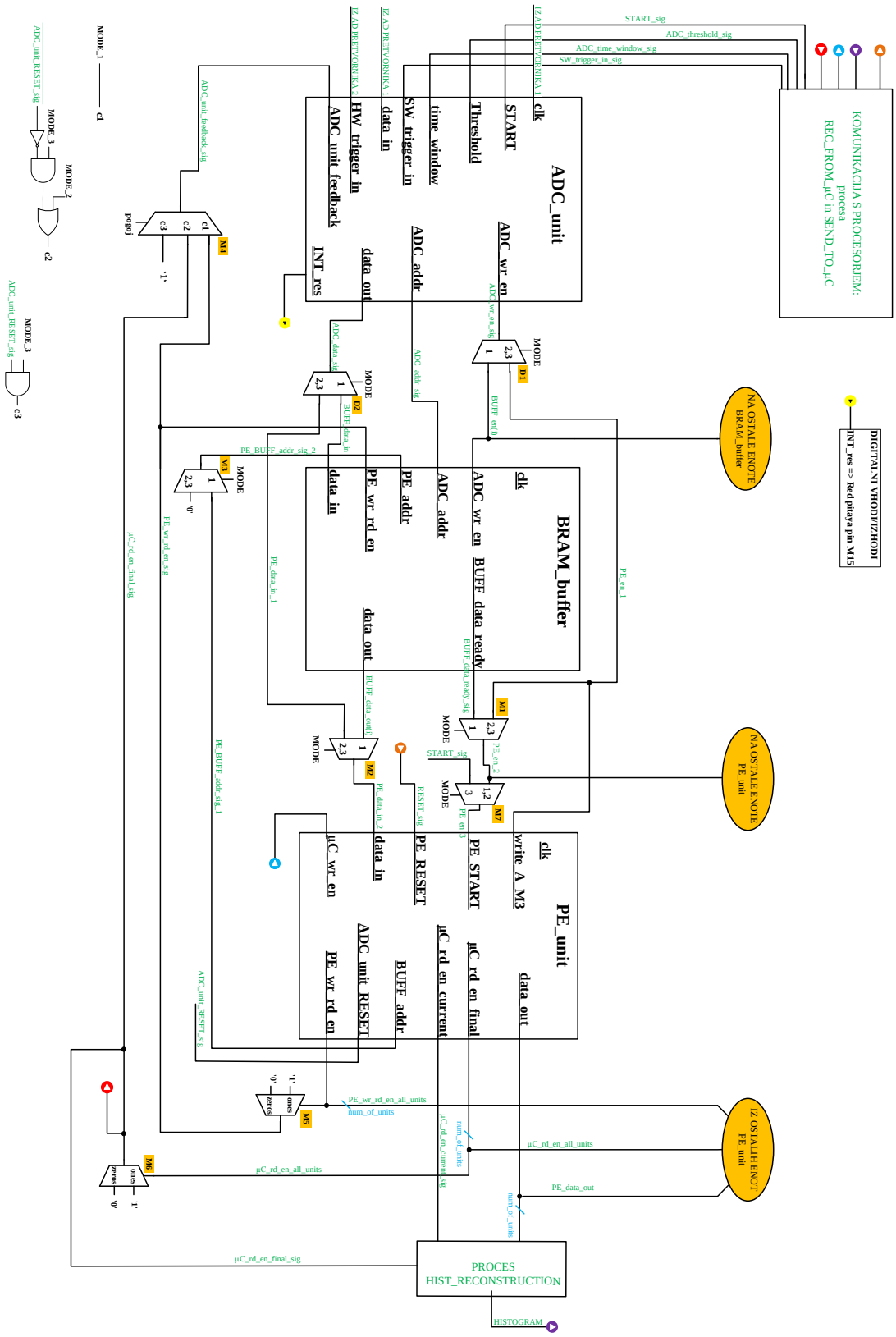
[20] Črt Valentinčič, "monitor.c [Online]." Dosegljivo: `https://github.com/RedPitaya/RedPitaya/blob/master/Test/monitor/monitor.c`. [Dostopano: 22.12.2018].

[21] Win SCP, "Win scp [Online]." Dosegljivo: `https://winscp.net/eng/index.php`. [Dostopano: 15.11.2018].

[22] S. Tatham, "Putty [Online]." Dosegljivo: `https://www.putty.org/`. [Dostopano: 15.11.2018].

Dodatek

A Komponenta *hist_system_top* za računanje histogramov



Slika A.1: Arhitektura komponente *hist_system_top*.