

Eiropas Reģionālās attīstības fonds

Prioritāte: 2.1. Zinātne un inovācijas

Pasākums: 2.1.1.1. Zinātne, pētniecība un attīstība

Aktivitāte: 2.1.1.1. Atbalsts zinātnei un pētniecībai

Projekta nosaukums: „Plaukstas datu nolasīšanas un apstrādes sistēma” (PALMS)

Līguma noslēgšanas datums: 13.12.2013.g.

Projekta sākuma datums: 01.01.2014.g.

Projekta beigu datums: 30.06.2015.g.

Vienošānās Nr.2013/0035/2DP/2.1.1.1.0/13/APIA/VIAA/015

Eiropas Reģionālās attīstības fonda finansējuma saņēmējs: Elektronikas un datorzinātņu institūts (EDI)

ZINĀTNISKĀ PĒTĪJUMA PROGRESU APLIECINOŠĀ DOKUMENTĀCIJA

Pārskata numurs Nr.5. par periodu no 01.01.2015.g. līdz 31.03.2015.g.

Projekta zinātniskais vadītājs: Modris Greitāns, Dr.sc.comp., vad. pētnieks
Pētījuma projekta izpildītāju saraksts: Ričards Fuksis, pētnieks
Mihails Pudžs, pētnieks
Oļegs Ņikišins, pētnieks
Rinalds Ruskuls, asistents
Teodors Eglītis, elektronikas inženieris

Anotācija

PALMS ir Eiropas Reģionālās attīstības fonda līdzfinansēts projekts zinātnei un pētniecībai. Projekta vispārīgais mērķis ir inovatīva un konkurētspējīga produkta izveide. Projekta ietvaros plānots izstrādāt plaukstu datu nolasīšanas un apstrādes sistēmu, kas balstīta uz biometrisku parametru (plaukstu asinsvadu, nospiedumu un ģeometrijas datu) izmantošanu. Projekts tiek veikts prioritārā zinātnes virzienā „Inovatīvi materiāli un tehnoloģijas”.

Projekta aktivitātes ietver:

- rūpnieciskos pētījumus, kas ietver izpēti darbu, lai iegūtu zināšanas un prasmes izstrādājamās iekārtas radīšanai, kā arī atsevišķu komplektējošu daļu izstrādei un uzlabošanai;
- plaukstu datu nolasīšanas un apstrādes sistēmas prototipa eksperimentālo izstrādi;
- projekta vadību un publicitāti par projektu.

Projekts tiek īstenots sadarbībā starp Elektronikas un datorzinātņu institūtu un SIA „Infoserv-Rīga”. Izpildes ilgums – 18 mēneši.

Šajā dokumentā dots pārskats par projekta piektajā periodā (01.01.2015.-31.03.2015.) veiktajiem pētniecības darbiem un šobrīd sasniegtiem rezultātiem.

Ievads

Privāto objektu piekļuve šobrīd galvenokārt tiek kontrolēta izmantojot mehāniskās durvju slēdzenes, kuras iespējams atslēgt izmantojot atbilstošu atslēgu, vai elektromehāniskās slēdzenes, kuras galvenokārt atslēdz, izmantojot radio frekvenču identifikācijas (RFID) kartes, vai arī kodu slēdzenes. Taču šīm drošības metodēm ir vairāki trūkumi: atslēgas un RFID kartes mēdz nozaudēt vai nozagt; atslēgas mēdz viltot un slēdzenes uzlauzt; RFID karti var nodot kādai citai personai; koda slēdzenes var kādam pateikt, vai ievadīšanas brīdī nolasīt.

Lai risinātu šīs problēmas un samazinātu to rezultātā radušos zaudējumus, projekta ietvaros plānots izstrādāt personu atpazīšanas iekārtu, kas balstīta uz plaukstu datu izmantošanu. Ierīce balstīsies uz plaukstu asinsvadu, plaukstu nospiedumu un ģeometrijas datu nolasīšanu un atpazīšanu. Šāda nolasītāja priekšrocība ir tā, ka lietotājam personīgi ir jābūt klāt identifikācijas brīdī, lai piekļūtu objektam. Šādā gadījumā cilvēka plauksta ir objekta piekļuves atslēga. Tiek izslēgti tādi riski kā atslēgas nozaudēšana vai aizmiršana.

PALMS pētniecības grupas darbs ir saistīts ar augšminēto problēmu risinājumu. Šajā projekta pārskata posmā ir veikti darbi un sasniegti rezultāti sekojošos aktivitātes „Rūpnieciskie pētījumi” pētījumos:

- Plaukstu datu apstrādes algoritmu izpēti, lai apstrādātu dažādos reālos apstākļos iegūtus attēlus;
- Plaukstu datu iegūšanas un apstrādes algoritmu paralelizācija un implementēšana iegūtajās sistēmās;
- Plaukstu datu šifrēšanas paņēmieni izpēti, datu efektīvas glabāšanas paņēmieni izpēti un testēšana;
- Plaukstu datu nolasītāja datu drošības risinājumu, datu pārraides arhitektūras un nepieciešamā drošības līmeņa izpēti.

Trešajā pārskata periodā ir uzsākts darbs pie projekta aktivitātes „Eksperimentālā izstrāde” apakšaktivitātēm:

- Plaukstu datu nolasīšanas un apstrādes sistēmas prototipa elektroniski iespiedplašu izstrāde, montāža un testēšana;
- Plaukstu datu nolasīšanas un apstrādes sistēmas prototipa kopējā montāža un testēšana.

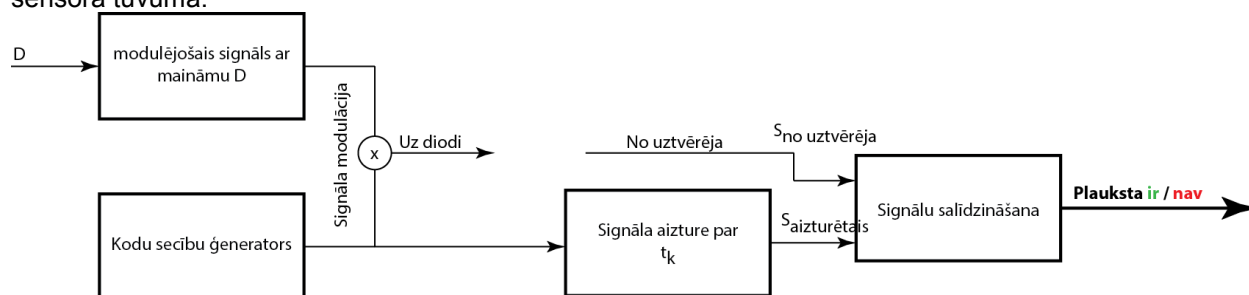
Rezultātu kopsavilkums

Projekta piektajā periodā (01.01.2015.-31.03.2015.) veikto pētniecības uzdevumu un rezultātu kopsavilkums ir sekojošs:

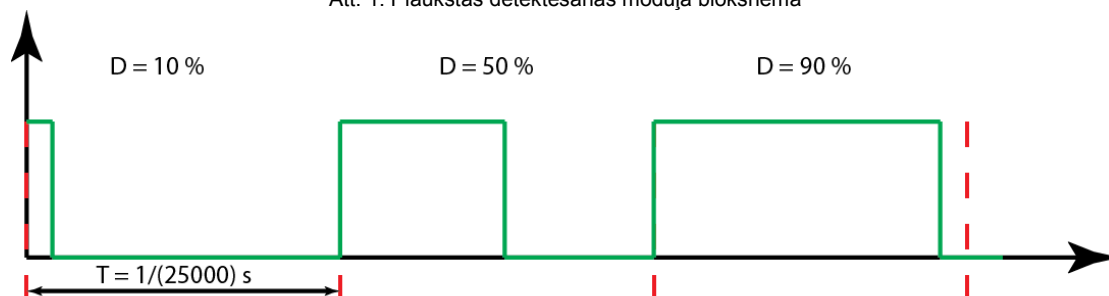
Plaukstas datu apstrādes algoritmu izpēte, lai apstrādātu dažādos reālos apstākļos iegūtus attēlus:

Plaukstas tuvuma detektēšana:

Lai detektētu, kad plauksta ir pietuvināta nolasošajai PALM's sistēmai un dotu signālu tālākai sistēmas darbības veikšanai: bilžu uzņemšanai, asinsvadu un rievu izdalīšanai, datu salīdzināšanai ar datu bāzi, tika izveidots modulis, kura pamatā tiek izmantota elektromagnētisko viļņu atstarošanās no kāda objekta (skatīt 1.att.). Šajā modulī ietilpst elektriskā shēma, kas raida un uztver izstaroto tuvās IR gaismas signālu, kā arī FPGA modulis, kur izmantojot VHDL programmēšanas valodu, tika izveidoti sekojoši bloki: 25 kHz modulējošais signāls ar maināmu darba cikla koeficientu (attēls 2), informācijas signāls (Barkera 7 kods), kā arī no uztvērēja pienākošā signāla apstrādes bloks, kas determinē vai ir detektēts kāds objekts sensora tuvumā.

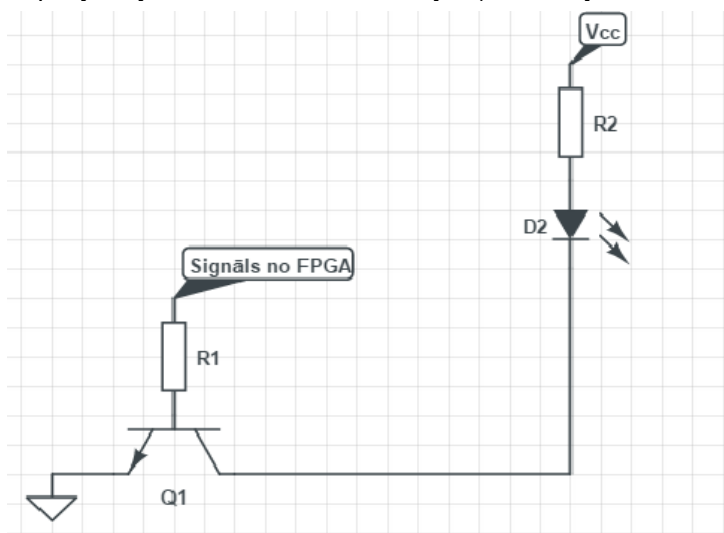


Att. 1. Plaukstas detektēšanas moduļa blokshēma



Att. 2. Modulējošais signāls, iespēja tam mainīt darbcikla koeficientu

Kā raidītājs tiek izmantots jau ierīcē esošās 850 nm diodes uz kurām tiek padots modulēts Barkera 7 kods, kas izmantots, lai novērstu iespējamus ārējus trokšņus (piemēram, apgaismojuma traucējumus, kas ielēgtu pārējo sistēmu, pat ja objekts nav tuvumā uztvērējam). Raidītāja elektriskā shēma parādīta 3.att.



Att. 3. Raidītāja elektriskā shēma

Tālāk šis atstarotais signāls tiek uztverts ar uztvērēja palīdzību un padots uz FPGA, kur implementētā loģika apstrādā uztverto signālu, izmantojot uz skaitītājiem balstītu sistēmu, determinējot, vai plauksta ir prezentēta sistēmai. Gadījumā, ja plauksta tiek detektēta, IR diožu vadības signāls tiek padots pārējai sistēmai, lai uzņemtu attēlus vai attēlu sēriju līdz brīdim, kad sistēma ir pieņēmusi kādu no 3 iespējamajiem lēmumiem:

1. Plauksta attēlā vai attēlu sērijā nav atrasta;
2. Plauksta attēlā vai attēlu sērijā ir atrasta, bet nav atpazīta;
3. Plauksta attēlā vai attēlu sērijā ir atrasta un atpazīta.

Plaukstu datu iegūšanas un apstrādes algoritmu paralelizācija un implementēšana iegultās sistēmās:

Šajā periodā veikta izstrādāto Matlab algoritmu pārveidošana tā, lai tos varētu implementēt FPGA vidē šādām algoritma sastāvdaļām – plaukstu detektēšana attēlā, ROI lokalizācija un HOV vektora aprēķināšana.

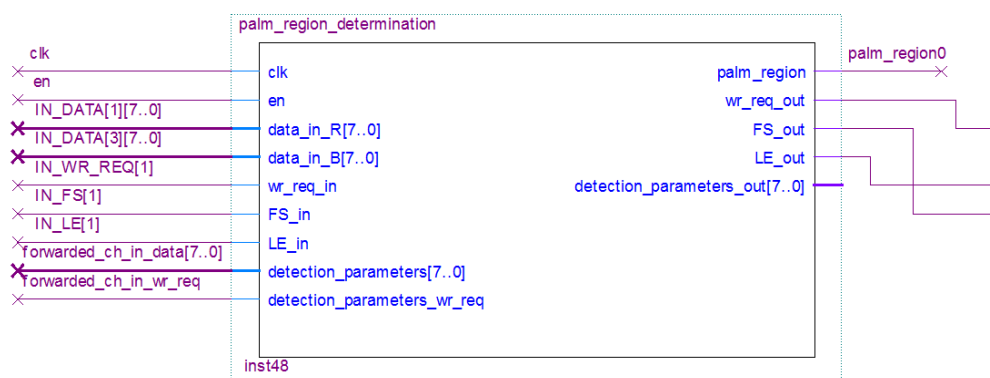
Plaukstu detektēšanas modulis.

Tā funkcijas:

- Plaukstu reģiona atlase pēc krāsas
- Plaukstu pirkstu starpu detektēšana
- Plaukstu pirkstu starpu klasificēšana, *izejā* nododot informāciju par pirkstu starpām $((x_1, y_1), (x_2, y_2))$, pēc kurām tiek pārprojicēts attēls pirms filtrēšanas.

Plaukstu detektēšanas modulis sastāv no vairākiem apakš blokiem, kuri aprakstīti turpmāk.

Plaukstu reģiona determinēšana



Att. 4. Plaukstu reģiona determinēšanas bloks Quartus vidē

Trīs taktu laikā tiek noteikti tādi pikseli, kuriem izpildās sakarības:

$$\begin{aligned}
 &0.85 \cdot b > r \\
 &\wedge \\
 &0.1 < r < 0.98 \\
 &\wedge \\
 &0.25 < b < 1
 \end{aligned}$$

Kur r un b – katra pikseļa sarkanās un zilās krāsas komponente. Koeficienti tiek katru reizi ieprogrammēti no Matlab vides, tādējādi paredzēta iespējamība tos vēlāk mainīt, veicot tālākus mērījumus un labojumus.

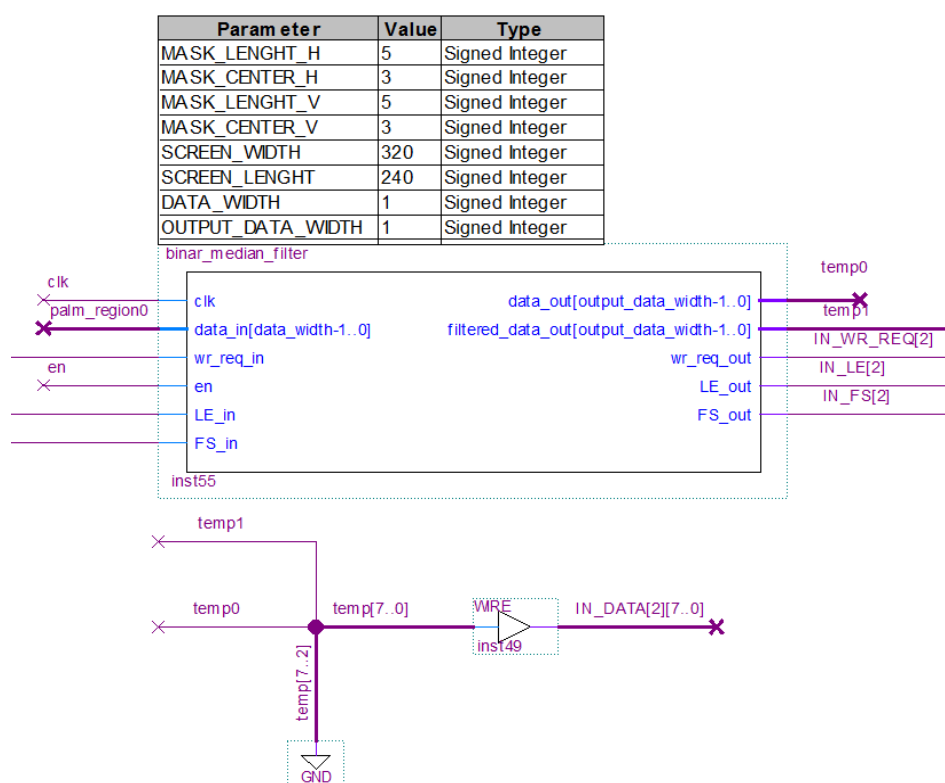
Universāls 2D filtrs

Vairāki tālākie apakš bloki veidoti kā 2D attēla filtri, par pamatu izmantojot **2D sparse** filtru. Tas pārveidots tā, lai tam varētu uzdot dažādus garumus x un y virzienos:

Parameter	Value	Type
MASK_LENGTH_H	5	Signed Integer
MASK_CENTER_H	3	Signed Integer
MASK_LENGTH_V	5	Signed Integer
MASK_CENTER_V	3	Signed Integer
SCREEN_WIDTH	320	Signed Integer
SCREEN_LENGTH	240	Signed Integer
DATA_WIDTH	1	Signed Integer
OUTPUT_DATA_WIDTH	1	Signed Integer

Att. 5. Universālā 2D filtra maināmie parametri

Binārs mediānas filtrs



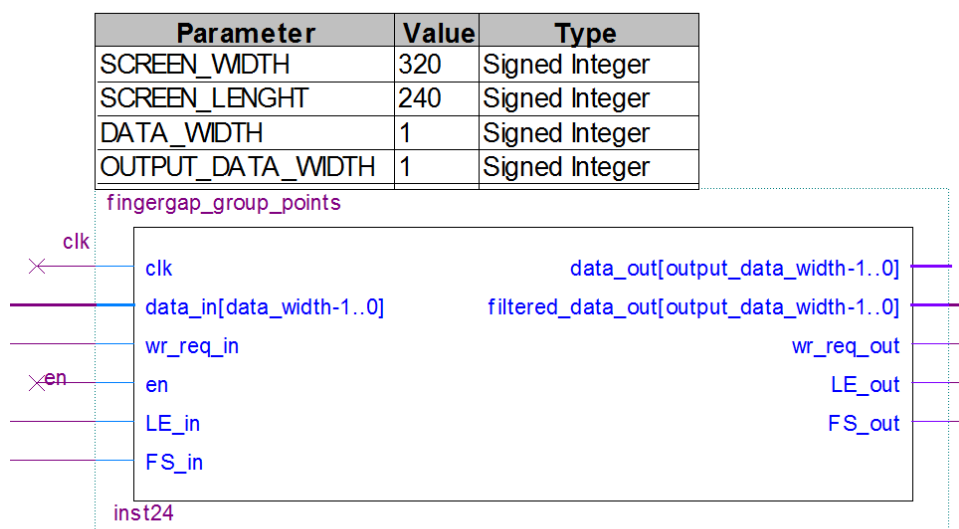
Att. 6. Binārā mediānas filtra bloks Quartus vidē

Īpašības – var uzdot brīvu izmēru (arī ne-kvadrātisku) ar nosacījumu, ka abu filtra malu garums ir nepāra skaitlis.

Lai determinētu izejas pikseļa vērtību, tiek aprēķināta aktīvās daļas pikseļu skaits, kuru vērtība ir “1”, izmantojot “tree adder”.

Izejas dati nosūtīšanai uz *Matlab* apvienoti shēmatēhniski, izmantojot Quartus rokas grāmatas piemēru (skatīt attēlu).

Pirkstu starpu detektors



Att. 7. pirkstu starpu detektora bloks Quartus vidē

Šis modulis atlasa tos punktus, kuru apkārtne līdzinās pirkstu starpai. Tā pat, kā iepriekšējais modulis – tas veidots kā 2D filtrs, taču tā kā tam ir stingri noteikta funkcionalitāte, tad filtra izmēri ir definēti tā iekšpusē, nevis kā *generic* parametri:

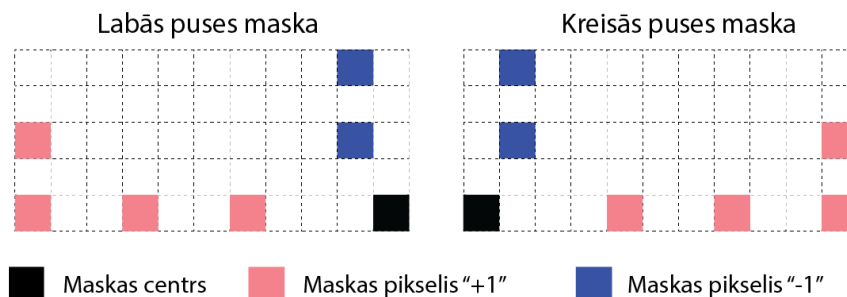
constant MASK LENGHT_H : integer:=21;

```

constant MASK_CENTER_H           : integer:=11;
constant MASK LENGHT_V           : integer:=5;
constant MASK_CENTER_V           : integer:=1;
constant CYCLES_HOLD_OUTPUT_DATA : integer := 2;

```

Tālāk filtrs veidots kā *sparse* filtrs:

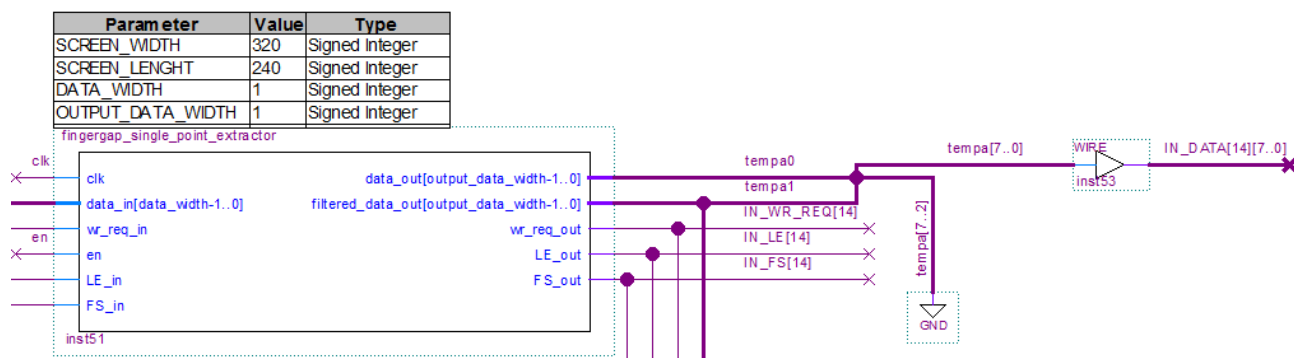


Att. 8. Pirkstu starpu detektora izmantotās Sparse tipa maskas

Tikai atšķirībā no sākotnējās ieceres, filtrs nevis dalīts divās daļās, bet gan veidots kā viens, ar aktīvās daļas izmēriem **21x5**.

Moduļa daļa, ar kuras palīdzību tiek izvēlēti aktīvās daļas pikseli, kuriem jāieņem vērtība +1 vai -1 izvēlēti, izmantojot *Matlab* skriptu - *fingergap_VHDL_code_generation.m*, kas nozīmē, ka kodu viegli atjaunot, veicot tālākus pētījumus.

Pirkstu starpu viena punkta atlase



Att. 9. Viena punkta atlases bloks Quartus vidē

Līdzīgi kā iepriekš, parametri definēti šī bloka *.vhd failā:

```

constant FINGERGAP_MASK_SIZE      : integer:=10;
constant MASK LENGHT_H            : integer:=FINGERGAP_MASK_SIZE*2+1;
constant MASK_CENTER_H            : integer:=FINGERGAP_MASK_SIZE+1;
constant MASK LENGHT_V            : integer:=2;
constant MASK_CENTER_V            : integer:=2;
constant CYCLES_HOLD_OUTPUT_DATA  : integer := 3;

```

Tā pat, kā *Matlab* vidē, aktīvās daļas rindas garumu nosaka parametrs **FINGERGAP_MASK_SIZE** (vienas maskas pusītes garums, bez centrālā pikseļa) bet rindu skaits ir 2, 2. rinda arī tiek uzskatīta par maskas centru, jo attēls filtrā ir apgriezts, un mēs meklējam tādu pikseli, zem kura nākošajā rindā nebūtu neviens pikselis.

Identiski, kā *Matlab* vidē, jāizpildās šādiem nosacījumiem:

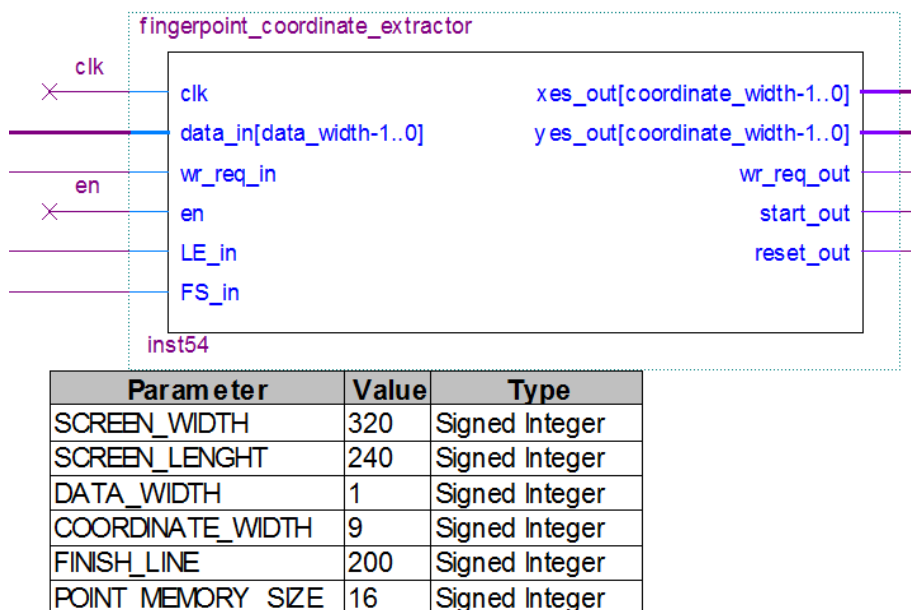
```

single_point_out_next <= "1" when (right_first_value_reg = left_first_value_reg OR ((1+ right_first_value_reg) =
left_first_value_reg))
AND
right_first_value_reg /= FINGERGAP_MASK_SIZE AND left_first_value_reg /=
FINGERGAP_MASK_SIZE
AND
first_stage_condition_reg_reg = '1'
AND
(possible_zeros_in_right_reg(right_first_value_reg) = '1' OR
possible_zeros_in_right_reg(right_first_value_reg+1) = '1')
else
"0";

```

lai tiktu atlasīts viens no grupas punktiem.

Pirkstu starpu koordinātu ekstraktors



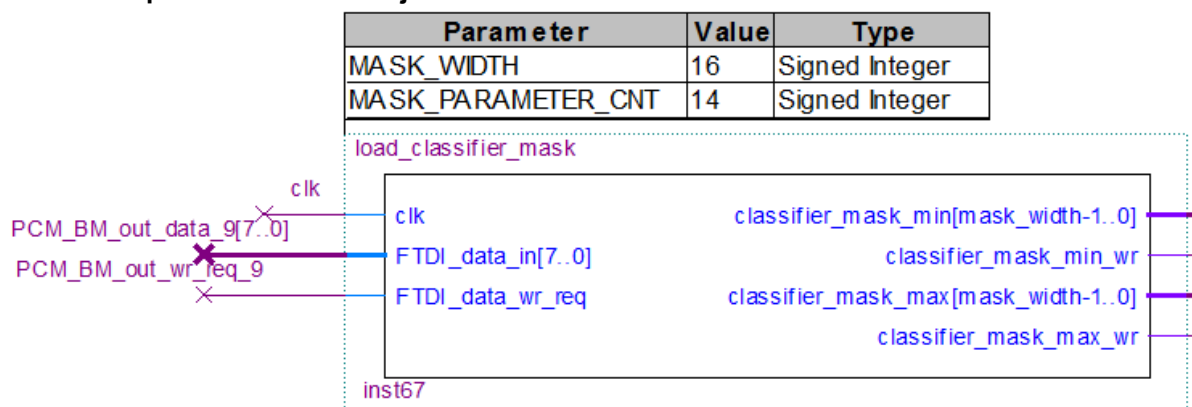
Att.10. Pirkstu starpu punktu koordinātu izdalīšanas bloks Quartus vidē

Moduļa uzdevums ir bināru attēlu pārvērst *x* un *y* koordinātu secībā. Tiek pārmeķlētas tikai rindas (1 ... *FINISH_LINE*) & moduļa iekšējā SDRAM atmiņā saglabāts *POINT_MEMORY_SIZE* skaits *x* un *y* koordinātu.

Ja ir savākts nepieciešamais koordinātu skaits vai apskatīta rinda ar numuru *FINISH_LINE*, tad ieejas attēla pikseļi vairs netiek apskatīti, bet tiek izvadītas koordinātas bloka izejā:

- Tiek saģenerēts *reset* signāls, ar kuru tālāk tiek atiestatīts klasifikators;
- Katrā *EN* taktā izvadīts viens koordinātu pāris (ar vienu *clk* takti garu *wr_req_out* signālu);
- Pēc pēdējā koordinātu pāra izvadīts *start*, ar kuru tiek palaists klasifikators.

Klasifikatora parametru ierakstītājs



Att.11. Klasifikatora parametru ierakstītāja bloks Quartus vidē

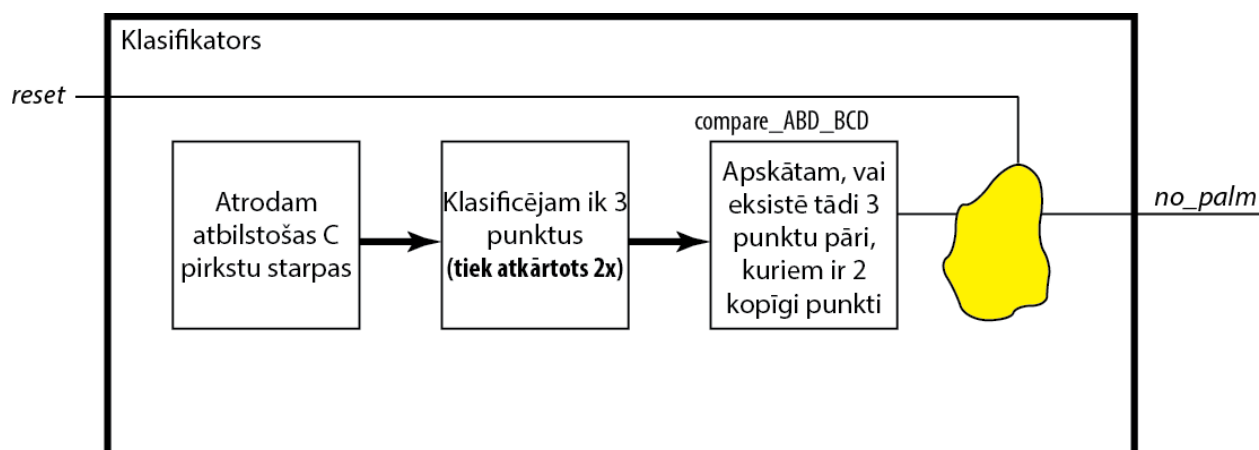
Veidots, jo menedžeris neatbalstīja vairāku baitu platus signālus.

Pirkstu starpu klasifikators



Att. 12. Pirkstu starpu klasifikatora bloks Quartus vidē

Izveidots pirkstu starpu klasifikators, kurš no atrasto objektu x un y koordinātām izdala tādas, kas ietilpst uzdotā diapazonā. Šis diapazons aprēķināts, izmantojot apmācības datubāzi. Klasifikators darbojas pēc attēlā redzamās blokshēmas.



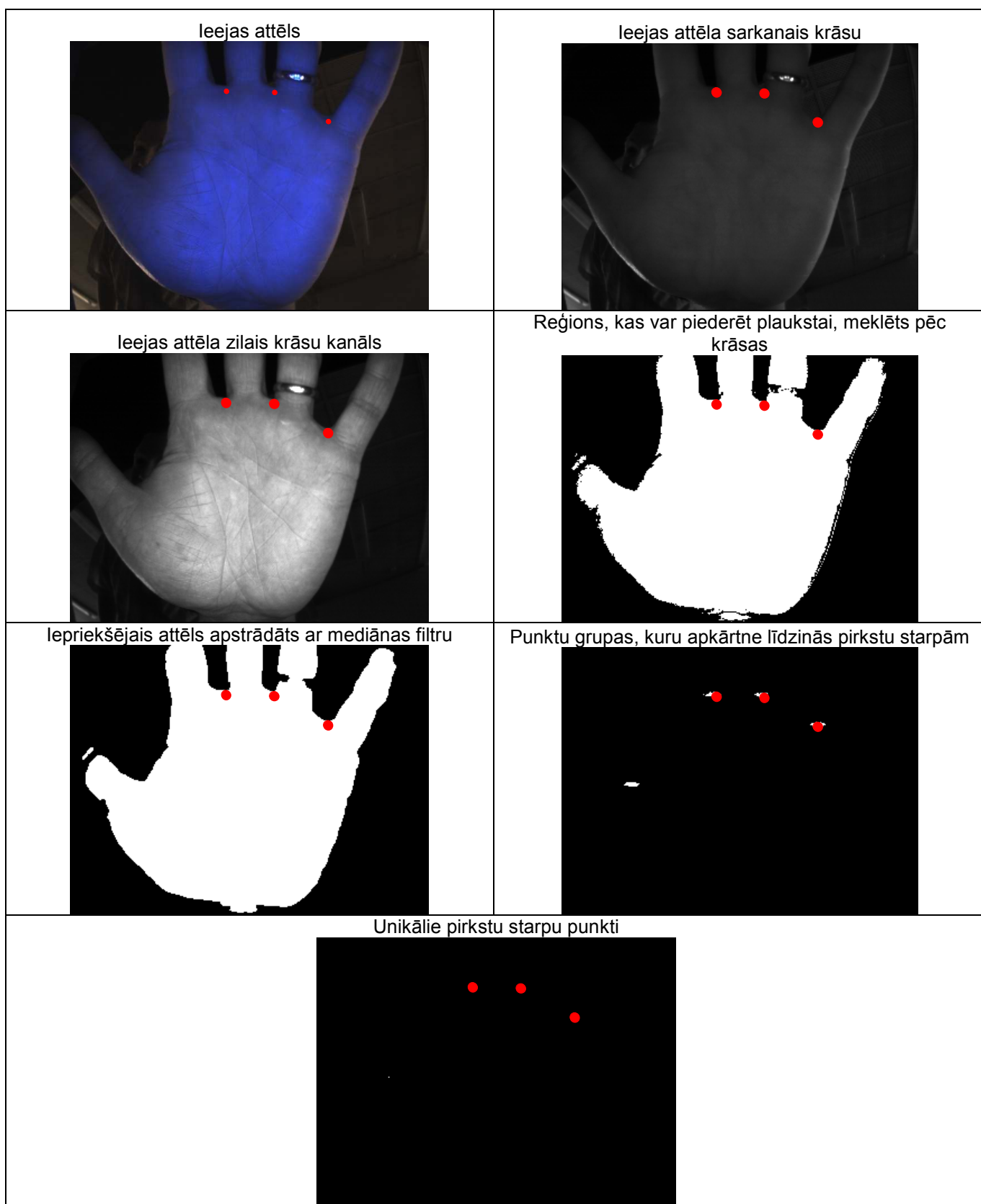
Att. 13. Klasifikatora blokshēma

Kur mākonītī esošais kods:

```
no_palm_next <= no_palm_compare_ABD_BCD when compare_ABD_BCD_finish_next = '1' else
                '1' when reset = '1' else
no_palm_reg;
```

Izveidota funkcija, ar kuru iespējams iegūt *Matlab* vidē dažādos plaukstas detektēšanas moduļa starprezultātus automatizēti, lai varētu veikt visu aprakstīto moduļu testēšanu un atklāšanu, veikt rezultātu salīdzināšanu.

Informācija, ko iespējams iegūt apskatāma zemāk. Uz visiem attēliem atzīmētas atrastās plaukstas pirkstu starpas.

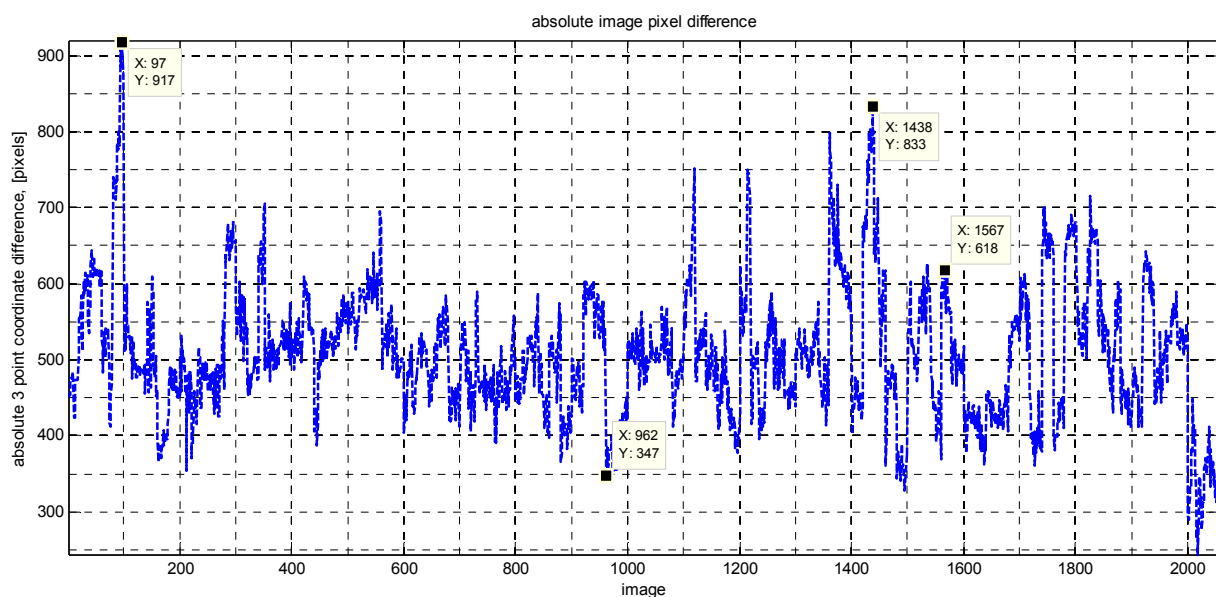


Att. 14. Dļespējamā testa attēlu iegūšana no dažādajiem plaukstas detektēšanas algoritma starpposmiem

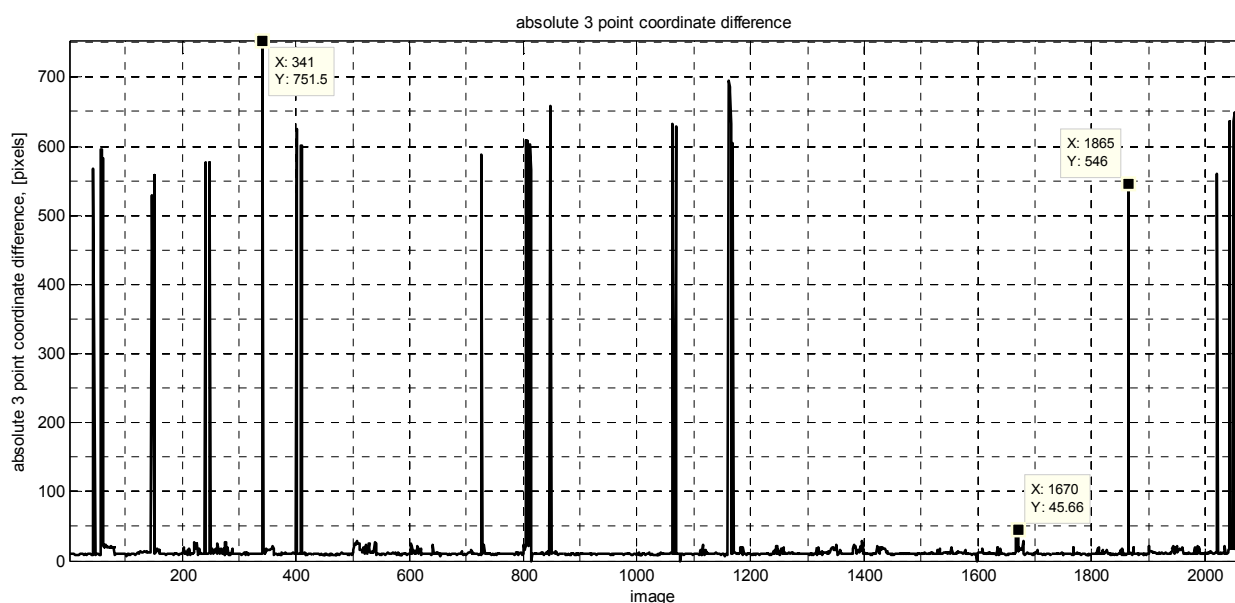
Lai spētu iegūt kādu novērtējumu FPGA implementācijai, tika salīdzināta Matlab un FPGA algoritmu performance. Tika izmantoti 103 personu datubāzes attēli (kopā $20 \times 103 = 2060$ attēli). Tie tika gan apstrādāti, izmantojot Matlab, gan FPGA. Iegūtas plaukstas binārās maskas (reģioni, kuri *varētu piederēt* plaukstai) kā arī pirkstu starpu koordinātas. Lai šos 2060 attēlus varētu kā apstrādāt, izveidoti grafiki –

attiecīgi, uz x ass – attēla numurs, bet uz y – vai nu summārais skaits tam, cik binārajās maskās pikseļi nesakrīt (pēc koordinātas), kā arī pirkstu starpu koordinātu, kuras iegūtas ar dažādajām metodēm, attālumu summa. Sprototams, ka ja tikai ar vienu no metodēm plauksta tikusi atrasta, tad attāluma summa būs kāds liels skaitlis.

Piezīme – attēlā kopā ir 320x240 pikseļi, kas kopā veido maksimālo iespējamo kļūdu 76'800.



Att. 15. Matlab un FPGA plauksta reģiona atlasē (binārā reģiona izdales) salīdzinājums

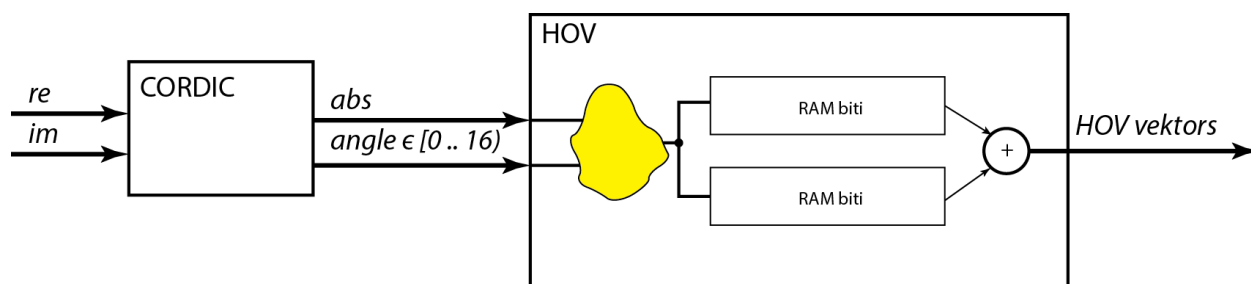


Att. 16. Matlab un FPGA atrasto pirksta starpu koordinātu salīdzinājums (attēlota to absolūtais attālums)

Redzams, ka reģiona atlasē un plauksta detektēšana strādā korekti. Nobīdes saistāmas ar to, ka tehnisku iemeslu dēļ FPGA vidē mediānas filtra izmērs ir 5x5, kamēr *Matlab* vidē – 4x4 pikseļi. Ar to arī izskaidrojams tas, ka dažreiz plauksta netiek detektēta / pirkstu starpas detektētas atšķirīgi kā *Matlab* vidē.

HOV algoritma implementācija

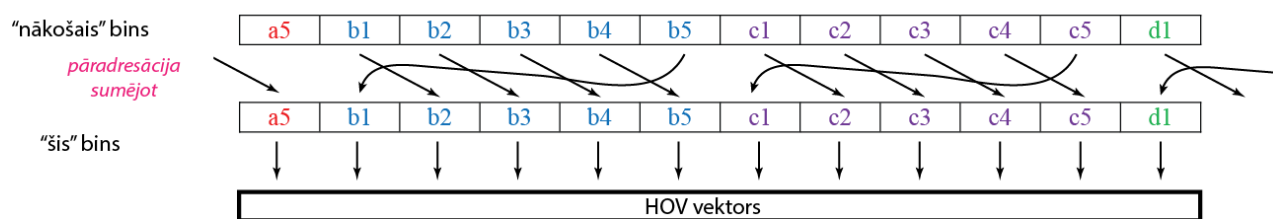
Šajā periodā arī tika implementēts HOV algoritms, iepriekš pārveidojot matemātisko darbību veikšanas veidu un secību, implementācijai FPGA. HOV algoritma FPGA implementācijas blokshēma parādīta attēlā.



Att. 17. HOV algoritma FPGA implementācijas blokshēma

HOV algoritmam ir nepieciešama informācija par izdalīto vektoru leņķi un to garumu (absolūto vērtību), taču filtra izejā iespējams iegūt tikai vektoru imaginārās un reālās daļas. Lai šāds pārveidojums būtu iespējams, pielietots CORDIC algoritms ar modifikāciju – tā izejā leņķis (0 .. 180 grādi) ir normēts kā [0, binu skaits) – t.i., ja tiek izmantoti 16 bini, tad leņķa vērtība normēta robežās 0 .. 15.99. Izmantojot šādu pieeju, tagad normētā leņķa veselā daļa liecina par binu, kurā tas ir jāieliek (sākot skaitīšanu no 0), taču daļskaitļa vērtība – sadalījumu starp šo binu un cikliski nākošo.

HOV modulī tiek noteikta katra attēla vektora piederība kādam no HOV vektora binu, izmantojot normētās leņķa vērtības veselo daļu, kā arī vektora pozīciju ieejas attēlā (blokshēmā mākonītis), taču absolūtā daļa tiek sadalīta daļā, kas jāpieskaita šim binam un cikliski nākošajam, izmantojot normētā leņķa decimālo daļu. Aprēķinātās vērtības tiek pieskaitītas attiecīgajām HOV vektora vērtībām, kas tiek glabātas abos atmiņas masīvos (RAM bitu blokos) – attiecīgi katrs atmiņas masīvs reprezentē šī un nākošā bina datus – abi masīvi ieejā tiek adresēti vienādi, lai gan satur informāciju par dažādiem binu – tā cikliski nobīdīta par binu skaitu, izejā nolasot šo informāciju – piemērs šādai nobīdei redzams attēlā. Šādi tiek izmantots minimāls skaits FPGA resursu, īpaši tos ietaupot pie aritmētisku darbību veikšanas kā arī atmiņas resursu adresēšanas, ļaujot HOV vektoru aprēķināt, neaizkavējot attēla plūsmu.



Att. 18. Datu nolasīšana no atmiņas, konstruējot HOV vektoru

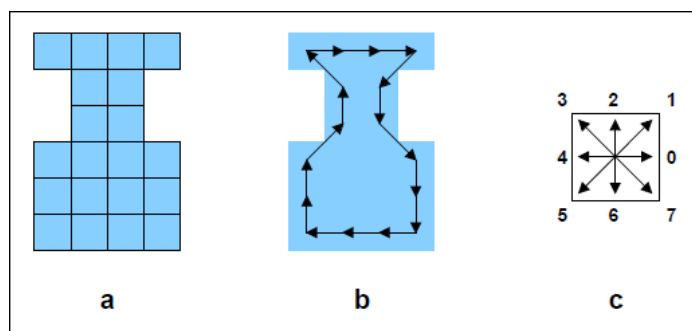
Izejā dati tiek normēti un padoti nākošajam kaskādes blokam.

Plaukstas datu šifrēšanas paņēmieni izpēte, datu efektīvas glabāšanas paņēmieni izpēte un testēšana:

Šajā ceturksnī, galvenokārt, tika pētītas, izstrādātas un testētas segmentācijas metodes, kas spētu uzlabot šifrēto datu stabilitāti. Panākot precīzu segmentu un apakš segmentu izdalīšanu, kas ir stabila no attēla uz attēlu (vienai personai) var panākt kopējās sistēmas augstāku precizitāti.

Tika izpētītas sekojošas segmentāciju metodes:

- Optimālā attēlu segmentācija, izmantojot minimālo kodēšanas garumu [1]
- Attēla robežu kodēšana izmantojot Frīmana ķēdes kodus [2]



Att. 19. Frīmana ķēdes koda piemērs

- Attēla robežu kodēšana izmantojot differences ķēžu kodus

Tika izpētīta, viena no superpikseļu iegūšanas metodēm [3], kā arī veikta datorprogrammas izstrāde MATLAB vidē.



Att. 20. Dažādas granularitātes superpikseļu piemēri

Tika izstrādāts paņēmiens, kā no superpikseļu reģioniem izdalīt HOV vektoru, taču veicot eksperimentus, atklājās, ka superpikseļu neregulāro kontūru dēļ, rezultāts nav stabils. Tika veikts superpikseļu centru aprēķins, kurus izmantojot tika veikts jaunu HOV vektoru aprēķins. Šāda pieeja ir jauna, taču ļoti atkarīga no izdalīto superpikseļu precizitātes un atkārtojamības. Lai attīstītu šo paņēmienu ir nepieciešami turpmāki pētījumi un eksperimenti. Tika izstrādāta arī datorprogramma, kas iegūtos HOV vektorus šifrē, izmantojot BioHash algoritmu un iegūst biokodu. Šī datorprogramma vēl ir izstrādes, testēšanas un atklūdošanas procesā.

[1] Hossein Mobahi, Shankar R. Rao, Allen Y. Yang, Shankar S. Sastry, Yi Ma, Segmentation of Natural Images by Texture and Boundary Compression, International Journal of Computer Vision, Vol. 95., October 2011., pp 86-98.

[2] Jongan Park; Chisty, K.M.M.; Jimin Lee; Youngeun An; Youngil Choi, "Image Retrieval Technique Using Rearranged Freeman Chain Code," Informatics and Computational Intelligence (ICI), 2011 First International Conference on , vol., no., pp.283,286, 12-14 Dec. 2011

[3] X. Ren and J. Malik. Learning a classification model for segmentation. In Proc. 9th Int. Conf. Computer Vision, volume 1, pages 10-17, 2003

Plaukstu datu nolasītāja datu drošības risinājumu, datu pārraides arhitektūras un nepieciešamā drošības līmeņa izpēte:

Šajā pārskata periodā tika analizēts, uzlabots un papildināts izstrādātais biometrijas datu drošības līmeņa noteikšanas algoritms. Tika apskatīta un analizēta biometrijas datu pārraides arhitektūra, ar mērķi salāgot to ar drošības risinājumiem. Uzsākta datu pārraides arhitektūras salāgošana.

Darbs veltīts arī plaukstu datu drošības uzlabošanai un pilnveidošanai, analizējot dažādus pieejamos risinājumus un novēršot problēmas.

Pēdējā Projekta pārskata periodā pastiprināts darbs tiks veikts pie plaukstu datu drošības risinājuma problēmu novēršanas un šī risinājuma realizācijas programmēšanas vidē, uzlabojot un pilnveidojot to. Tiks veikts arī izveidotā algoritma apraksts un izvērsta analīze.

Plaukstas datu nolasīšanas un apstrādes sistēmas prototipa elektroniski iespiedplašu izstrāde, montāža un testēšana, prototipa kopējā montāža un testēšana:

Šajā laika posmā tika uzsākts darbs pie PALMS prototipa izstrādes un veikti papildus sagatavošanās darbi, lai to sekmīgi izdarītu:

- Komponentu (bibliotēkas) izveide - automatizētās iespiedplašu projektēšanas vidē;
- arhitektūras koncepta izveide;
- principiālo elektrisko shēmu izveide

PALMS prototips tiek projektēts **Altium Designer** vidē, tas ir projektēšanas rīks, kurā ir daudz dažādu papildus apakšprogrammu, kas atvieglo kopējo projektēšanas gaitu.

Arhitektūras izveide

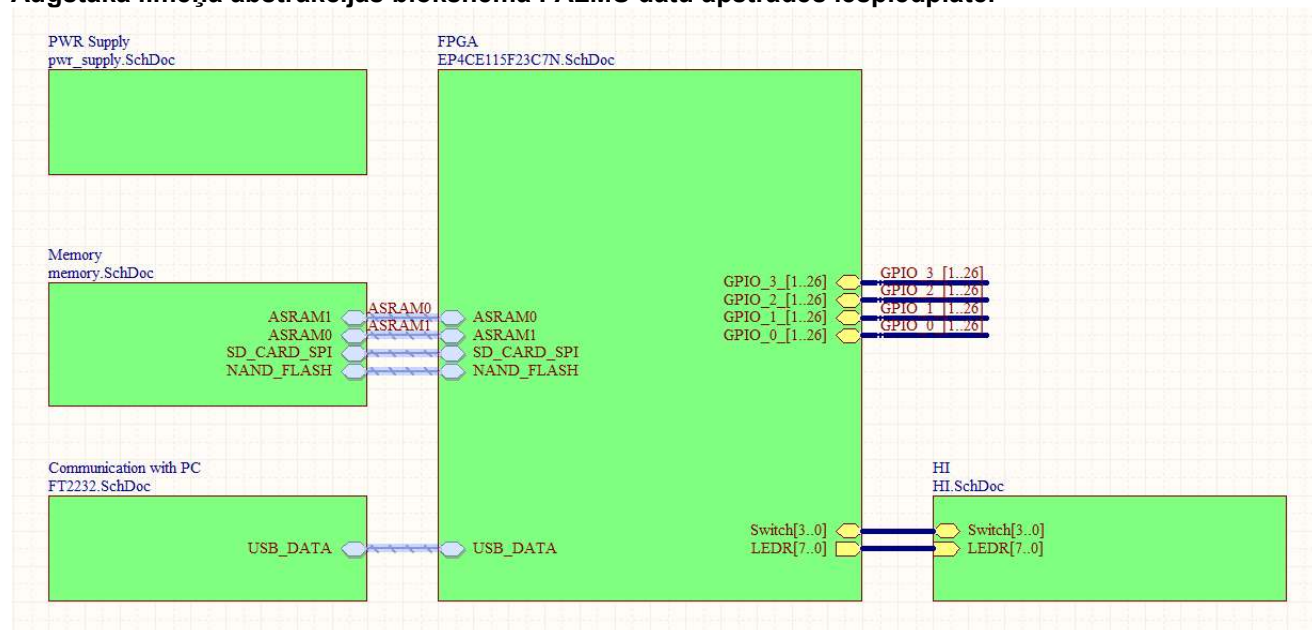
Pirms izstrādāt konkrētu tehnisko risinājumu tika domāts pie kopēja koncepta tā, lai iekārta būtu gana universāla un laika gaitā tās funkcionālās daļas varētu nomainīt pret citām. Pēc vairāku variantu izveides palikām pie modulāra risinājuma, tas ir, gala prototips sastāvēs no vairākām komplektējošām daļām:

- datu apstrādes plate;
- attēlu iegūšanas plate;
- gaismas diožu plate;
- utt.

“Datu apstrādes plate”

Šajā posmā tika uzsākts darbs pie “datu apstrādes plates” izstrādes un tās sastāvdaļu izvēles. Pamatojoties uz to, ka iepriekšējo testu veikšanai mēs izmantojām **Altera** izstrādes rīkus un, lai samazinātu iespējamās kļūdas, tad komponentus izvēlējāmies identiskus kādi ir atrodamī šajos izstrādes rīkos. Parējos komponentus izvēlējāmies no iepriekšējās pieredzes.

Augstākā līmeņa abstrakcijas blokshēma PALMS datu apstrādes iespiedplatei



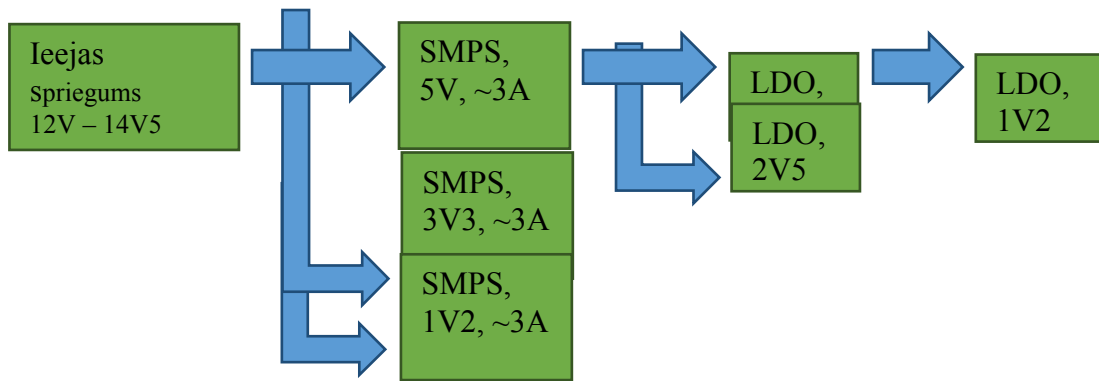
Att.21. Augsta līmeņa abstrakcijas blokshēma

Kā redzams 21.attēlā “datu apstrādes platei” ir 5 galvenās sastāvdaļas:

- barošanas sprieguma principiālā elektriskā shēma;
- atmiņas principiālā elektriskā shēma;
- komunikācijas principiālā elektriskā shēma;
- ievades, izvades principiālā elektriskā shēma;
- FPGA principiālā elektriskā shēma.

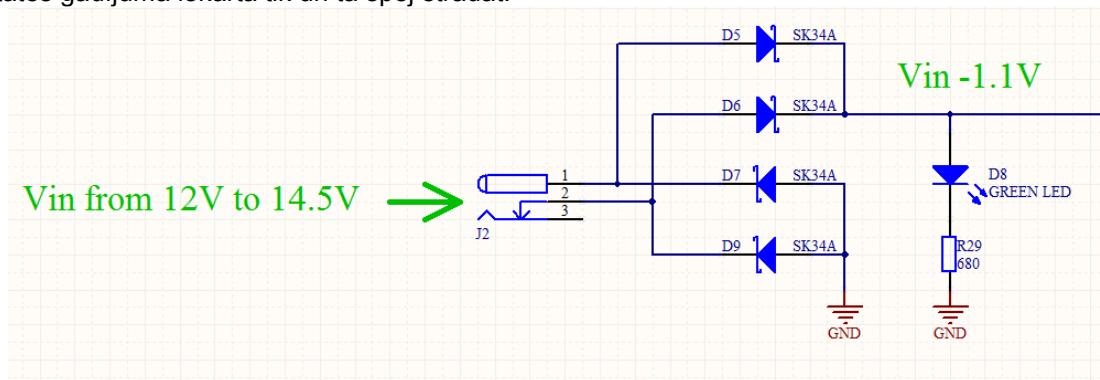
Barošanas sprieguma principiālā elektriskā shēma

Paredzēta lējas sprieguma pārveidošana uz dažādiem citiem spriegumiem (5V, 3V3, 2V5, 1V2), kas nepieciešami izmantoto komponentu darbināšanai.



SMPS – **S**witched-**M**ode **P**ower **S**upply (impulsu barošanas pārveidotājs). SMPS paredzēti galvenokārt nodrošināt barošanas spriegumu FPGA mikrosīmai. LDO sprieguma pārveidotāji ir izmantoti pārējiem komponentiem.

Ieejas barošanas ķēdei ir aizsardzības daļa, kas nodrošina, ka sajaukta ieejas barošanas sprieguma polaritātes gadījumā iekārta tik un tā spēj strādāt.

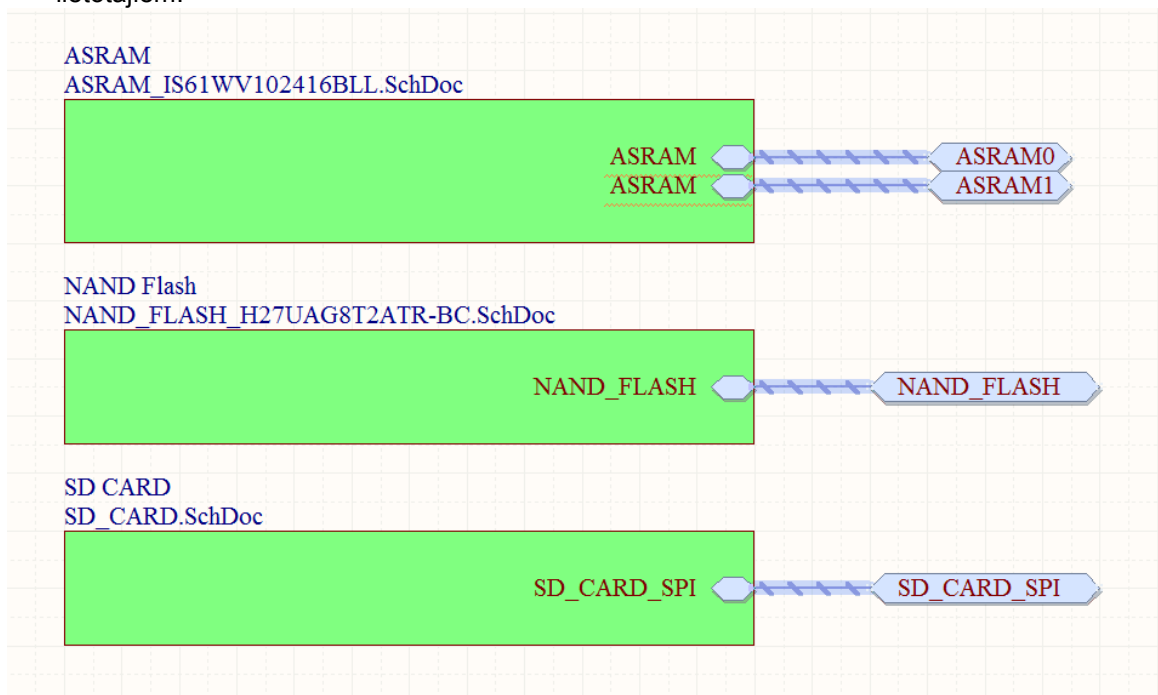


Šī shēmas daļa nodrošina barošanas sprieguma filtrēšanu, kā arī samazina EMI.

Atmiņas principiālā elektriskā shēma

Sastāv no trīs apakšgrupām:

- **ASRAM** - asinhronās brīvpiekļuves atmiņas;
- **SDcard** – patstāvīgā atmiņa, kas paredzēta liela apjoma datu glabāšanai;
- **NAND FLASH** – pastāvīga atmiņa, kas paredzēta liela apjoma datu glabāšanai, nav pieejama lietotājiem.



Tiek izmantotas divas **ASRAM** integrālās mikroshēmas, lai īslaicīgi glabātu aprēķinātos starprezultātus, piemēram, datus, kuri ir aprēķināti filtrācijas gaitā. Katrai ASRAM ir 16Mbitu liels datu apjoms un rakstīšanas/lasīšanas ātrums ir ~200MB/s, šie parametri ir īpaši svarīgi, lai apstrādi veiktu reālā laikā. *Apstrāde reālā laikā nozīmē to, ka ieejas dati tiek apstrādāti ar tādu pat ātrumu kā tie pienāk.

SDkartes atmiņā ir paredzēts, ka var tikt glabāti sistēmas konfigurācijas parametri:

- filtru – gausa, NHCMF;
- klasifikatora;
- attēla sensora;
- gaismas diožu intensitāte.

NAND FLASH atmiņā ir paredzēts, ka tiks glabāti biometrijas datubāze, kas ir šifrētā veidā, lai tos nevarētu tik vienkārši uzlauzt.

Komunikācijas principiālā elektriskā shēma

Paredzēta, lai nodrošinātu prototipa komunikāciju ar datoru. Galvenokārt tiek lietots iekārtu izstrādājot un atklūdojot kā arī nosūtot iekārtai konfigurācijas parametrus – filtram, gaismas intensitātei, attēla sensoram.

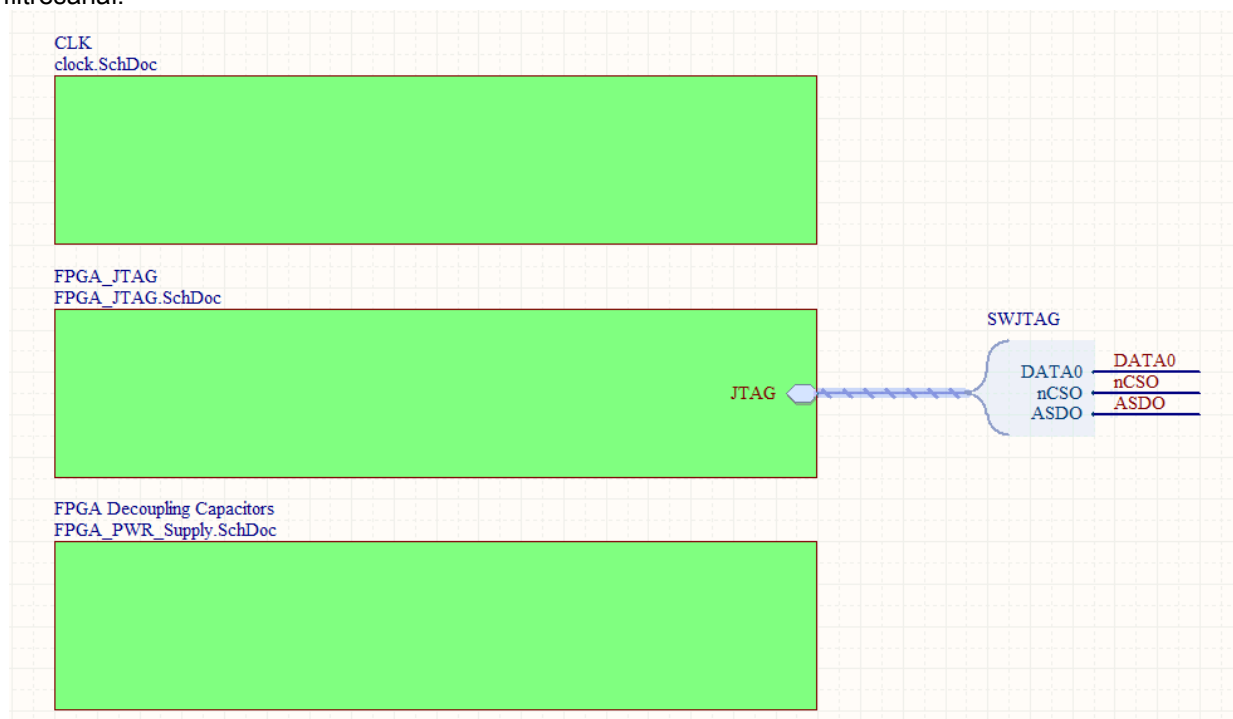
Tika izvēlēta FT2232HL mikroshēma, kas nodrošina līdz pat 15MB/s lielu datu pārraides ātrumu.

Ievades, izvades principiālā elektriskā shēma

Paredzēts, lai varētu ievadīt, izvadīt informāciju, tas ir, pogas kā ievades komponenti, gaismas diodes kā izvades komponenti.

FPGA principiālā elektriskā shēma

FPGA integrālās mikroshēmas shēma, kas savieno savstarpēji iepriekš aprakstītās shēmas sastāvdaļas. Papildus tam ir shēmas daļas, kas nepieciešamas FPGA ieprogrammēšanai, barošanas sprieguma filtrēšanai.



Takts frekvences principiālā elektriskā shēma

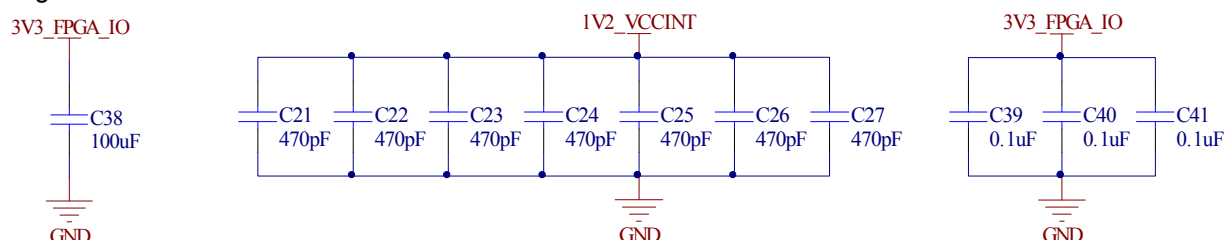
FPGA var tik izmantots asinhronā vai sinhronā vai kombinētā veidā. Sinhronais režīms nozīmē to, ka ir kāds globāls references (atbalsta) signāls, pēc kura tad notiek datu apstrāde. Šī shēmas daļa nodrošina šādu signālu ģenerēšanu - viens ir 50MHz otrs 27MHz. Izmantojot iekšējos **FPGA** resursu kā **PLL** ieejas signāla frekvence var tik mainīta gan uz paaugstināta, gan uz samazināta.

Sprieguma barošanas filtrēšana

leejas barošanas spriegumu ir nepieciešams filtrēt, lai tas saglabātu konstantu vērtību noteiktā frekvenču diapazonā. FPGA integrālai mikroshēmai tiks padota 100MHz liela takts frekvence, respektīvi līdz šādam signālam mums ir jāspēj efektīvi filtrēt pienākošie barošanas spriegumi.

Galvenokārt šim nolūkam tiek izmantotas atbilstošas spriegumu pārveidotāju mikroshēmas, tās tika aprakstītas iepriekš un papildus tam kondensatori un spoles, droseles.

Barošanas sprieguma filtrācija FPGA tika veikta pēc sekojoša principa. Sākuma tiek izmantots lielākas kapacitātes kondensators, piemēram, 100uF, tad zemākas (no 10uF – 0.47uF), tad vēl zemākas kapacitātes – 0.1uF līdz 22nF. Zemākās kapacitātes kondensatoru ir vairāk un tos maksimāli jānovieto integrālo mikroshēmu izvadu tuvumā.

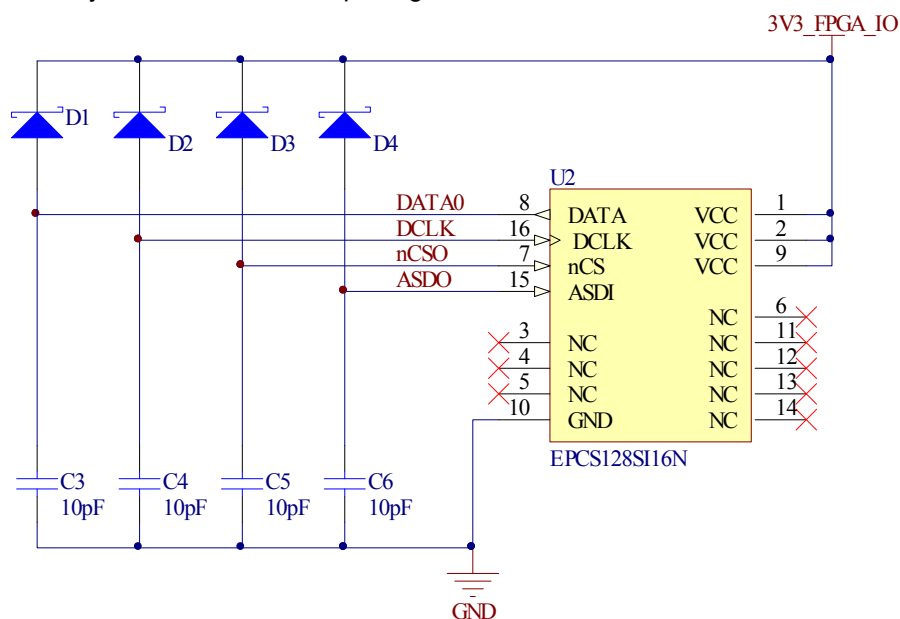


FPGA programmēšana

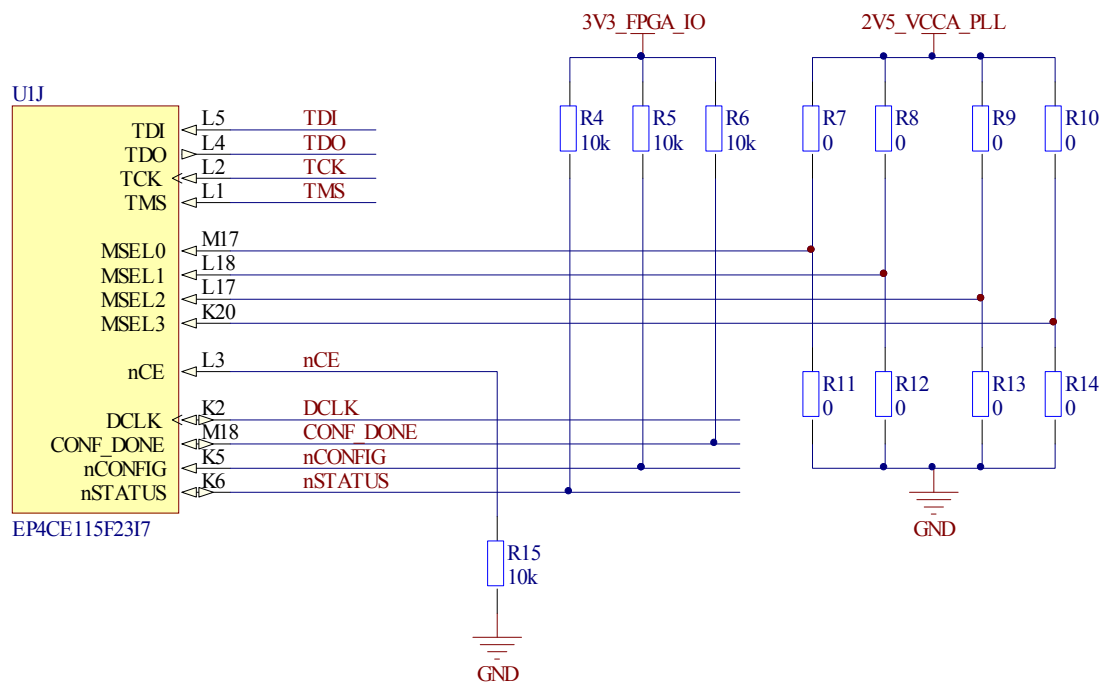
FPGA atbalsta divu veidu ieprogrāmēšanu – **JTAG**, **AS**.

JTAG lielākoties izmanto projektējot un atklūdojot iekārtu, dizains ir aktīvs kamēr iekārtai ir nodrošināts barošanas spriegums.

AS ir paredzēts ieprogrāmēt ārēju **EEPROM** integrālo mikroshēmu, katru reizi ieslēdzot barošanas spriegumu **FPGA** nolasa saglabāto dizainu no šīs **EEPROM** integrālās mikroshēmas un uzsāk darbu. Šādu pieeju izmanto, ja iekārtas izstrāde ir pabeigta.



Att.22. AS principiālā elektriskā shēma



Att.23. JTAG principiālā elektriskā shēma

Secinājumi

Projekta PALMs pētnieciskā darbība „Rūpniecisko pētījumu” aktivitātē tiek sekmīgi turpināta. Noris plaukstu datu apstrādes algoritmu izpēte, plaukstu datu iegūšanas un apstrādes algoritmu paralelizācija, implementēšana iegultās sistēmās un plaukstu datu šifrēšanas paņēmieni izpēte un plaukstu datu nolasītāja datu drošības risinājumu, datu arhitektūras un nepieciešamā līmeņa izpēte.

Plaukstu datu apstrādes algoritmu izpēte, lai apstrādātu dažādos reālos apstākļos iegūtus attēlus. Uzdevums bija plaukstu tuvuma detektēšana, kas dotu signālu tālākai sistēmas darbības veikšanai. Uzdevuma ietvaros tika izveidots modulis, kura pamatā izmantota elektromagnētisko viļņu atstarošanās no objekta. Sistēmas uzdevums ir pieņemt vienu no trim lēmumiem – plauksta attēlā nav atrasta, plauksta attēlā ir atrasta, bet nav atpazīta vai plauksta attēlā ir atrasta un ir atpazīta. Pēdējā pārskata periodā turpināsies darbs pie izveidoto algoritmu testēšanas un uzlabošanas.

Plaukstu datu iegūšanas un apstrādes algoritmu paralelizācija un implementēšana iegultās sistēmās. Darbs pie plaukstu detektēšanas, plaukstu reģiona determinēšanas, universāla 2D filtra izveides, bināra mediānas filtra izveides, pirkstu starpu detektora izveides, pirkstu starpu viena punkta atlasē, pirkstu starpu koordinātu ekstraktora izveides, klasifikatora parametru ierakstītāja izveides, pirkstu starpu klasifikatora izveides un HOV algoritma implementācijas.

Plaukstu datu šifrēšanas paņēmieni izpēte, datu efektīvas glabāšanas paņēmieni izpēte un testēšana. Šajā pārskata periodā galvenokārt tika pētītas, izstrādātas un testētas segmentācijas metodes, kas spētu uzlabot šifrēto datu stabilitāti. Panākot precīzu segmentu un apakšsegmentu izdalīšanu, kas ir stabila no attēla uz attēlu var panākt kopējās sistēmas augstāku precizitāti. Lai attīstītu šo paņēmieni ir nepieciešami turpmāki pētījumi un eksperimenti.

Plaukstu datu nolasītāja datu drošības risinājumu, datu pārraides arhitektūras un nepieciešamā drošības līmeņa izpēte. Tika analizēts, uzlabots un papildināts izstrādātais biometrijas datu drošības līmeņa noteikšanas algoritms. Tika apskatīta un analizēta biometrijas datu pārraides arhitektūra, ar mērķi salāgot to ar drošības risinājumiem. Uzsākta datu pārraides arhitektūras salāgošana. Pēdējā Projekta pārskata periodā pastiprināts darbs tiks veikts pie plaukstu datu drošības risinājuma problēmu novēršanas un šī risinājuma realizācijas programmēšanas vidē.

Šajā periodā turpinās arī projekta darbība „Eksperimentālās izstrādes” aktivitātē. Noris plaukstu datu nolasīšanas un apstrādes sistēmas prototipa elektroniska iespaidplašu izstrāde, montāža un testēšana un plaukstu datu nolasīšanas un apstrādes sistēmas prototipa kopējā montāža un testēšana. Tika uzsākts darbs pie PALMS prototipa izstrādes – arhitektūras izveide, datu apstrādes plates izstrāde, barošanas sprieguma, atmiņas, komunikācijas, ievades/izvades, FPGA un takts frekvences principiālo elektrisko shēmu izveide, sprieguma barošanas filtrēšana un FPGA programmēšana. Pēdējā pārskata periodā turpināsies darbs pie Projekta prototipa izveides, testēšanas un atklādošanas.

Publicitātes aktivitātes ietvaros ir izveidota un tiek uzturēta projekta mājas lapa EDI mājas lapas ietvaros, informācija par projektu ievietota gan www.bite.edi.lv, gan biometrijas grupas twitter kontā, par projekta uzsākšanu ir ievietota preses relīze nacionālā ziņu aģentūrā LETA un projekta norises vietā ir izvietota plāksne. Sekmīgi ir nodrošināta projekta vadība, regulāri notiek vadības grupas sēdes. Uzraudzības padomes sēdēs veikta projekta gaitas un iespējamo risku analīze.