

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: „Multimodālas biometrijas tehnoloģija drošai un ērtai personu
autentifikācijai” (BiTe)

Līguma noslēgšanas datums: 10.12.2010.g.

Projekta sākuma datums: 01.01.2011.g.

Projekta beigu datums: 31.12.2013.g.

Vienošanās Nr.2010/0285/2DP/2.1.1.1.0/10/APIA/VIAA/098

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

ZINĀTNISKĀ PĒTĪJUMA PROGRESU APLIECINOŠĀ DOKUMENTĀCIJA

Pārskata numurs Nr.8. par periodu no 01.03.2013.g. līdz 31.05.2013.g.

Projekta zinātniskais vadītājs: Modris Greitāns, Dr.sc.comp., vad. pētnieks
Pētījuma projekta izpildītāju saraksts: Mihails Broitmans, Dr.sc.comp., vad. pētnieks
Rihards Fuksis, pētnieks
Mihails Pudžs, pētnieks
Rinalds Ruskuls, asistents
Oļegs Ņikišins, asistents
Zanda Seržāne, asistente
Artūrs Kadiķis, programmēšanas inženieris
Dāvis Barkāns, elektronikas inženieris
Teodors Eglītis, elektronikas inženieris
Mārtiņš Saulis, informātikas/elektronikas tehniķis

Satura rādītājs

Anotācija.....	3
Ievads	3
Rezultātu kopsavilkums	3
Aktivitāte: Lietišķie pētījumi	6
Biometrisku datu apstrādes algoritmu darbības izvērtēšana.....	6
Biometrisku datu iegūšanas algoritmu paralelizācija un implementēšana programmējamās loģiskās masīvos	9
Biometrisku datu kriptēšanas, glabāšanas un lietojuma apakšsistēmas izveide	13
Aktivitāte: Eksperimentālā izstrāde.....	15
Tehnoloģijas demonstratora eksperimentālā maketa montēšana.....	15
Algoritmu implementēšana eksperimentālajā maketā	20
Secinājumi	26

Anotācija

BiTe ir ERAF līdzfinansēts projekts zinātnei un pētniecībai. Projekta mērķis ir droša, ērta un plaši pielietojama personu identifikācijas risinājuma izveide. Projekts ietver sevī sekojošas aktivitātes: Lietišķos pētījumus biometrijas parametru ieguvē, apstrādes metožu implementēšanā ierobežotu resursu sistēmās, atsevišķu sistēmas komponentu un bloku izstrādē; Eksperimentālos pētījumus tehnoloģiju demonstratora izstrādē; Rūpniecisko tiesību aizsargāšana, patenta pieteikums; Projekta publicitāte. Šajā dokumentā dots pārskats par projekta astotajā periodā (01.03.2013.-31.05.2013) veiktajiem pētniecības darbiem un šobrīd sasniegtiem rezultātiem.

Projektu atbalsta Eiropas Reģionālās attīstības fonds, līguma Nr. 2010/0285/2DP/2.1.1.1.0/APIA/VIAA/098

Ievads

Identitātes nozagšana ir viens no modernajiem noziegumiem, kas „zaglim” dod iespēju radīt personām finansiālus zaudējumus. Katru gadu identitātes nozagšana pasaulē rada aptuveni 221 miljardu dolāru zaudējumus. Lielu interesi par biometrijas izmantošanu personu identifikācijā izrāda banku sektors. Bankas atzīst, ka šādas sistēmas ieviešana dotu iespēju aizstāt (vai kombinēt) uz parakstu vai PIN kodu ievadi balstītu norēķinu karšu autorizāciju ar kartes īpašnieka biometrisku atpazīšanu. Radot ērtu, lētu un drošu personu identifikācijas risinājumu, ir plašas iespējas to ieviest ikdienas dzīvē – bankās, tirdzniecības vietās, objektu piekļuves un lietošanas kontrolē (ieskaitot autotransportu) u.c.

BiTe pētniecības grupas darbs ir saistīts ar augstāk minēto problēmu risinājumu. Šajā projekta pārskata posmā ir veikti darbi un sasniegti rezultāti sekojošos pētījumos:

- Sejas un plaukstu attēlu iegūšanas metožu izpēte;
- Biometrisku datu apstrādes algoritmu darbības izvērtēšana;
- Biometrisku datu iegūšanas algoritmu paralelizācija un implementēšana programmējamajos loģiskos masīvos;
- Biometrisku datu kriptēšanas, glabāšanas un lietojama apakšsistēmas izveide;
- Algoritmu implementēšana eksperimentālajā maketā.

Rezultātu kopsavilkums

Projekta astotajā periodā (01.03.2013.-31.05.2013.) veikto pētniecības uzdevumu un eksperimentālo rezultātu kopsavilkums ir sekojošs:

- Sejas un plaukstu attēlu iegūšanas metožu izpēte – iepriekšējos pārskata periodos tika apskatītas plaukstu specifisko pazīmju izdalīšanas iespējas; izpētīti plaukstu ģeometrijas izmēri, kas minimāli atkarīgi no plaukstu stāvokļa; turpināts plaukstu ģeometrijas algoritmu apraksts MATLAB vidē; analizēta mērķa principa metodika; izpētīts sejas detektēšanas algoritms, darbam daudzkodolu procesorā; veikta

vienlaicīga oriģinālu un filtrētu bilžu iegūšana un noteikta lielākā iespējamā riņķa ievilkšana plaukstaru kontūrā, izmantojot Voronoi šūnas.

- Biometrisku datu apstrādes algoritmu darbības izvērtēšana – iepriekšējos periodos realizēts 2D sejas atpazīšanas algoritms, kas implementēts iegultā sistēmā; veidots automātisks sejas atpazīšanas algoritms TMS320C6416; pievienoti galvenie elementi kameras modulim; izveidotas programmas attēla vizualizācijai un kvalitātes novērtēšanai; analizētas sejas atpazīšanas algoritmu implementācijas iespējas; apskatīti iespējamie risinājumi sejas atpazīšanas izstrādei iegultās sistēmās un uzsākta iegultas sistēmas izveide uz Raspberry Pi. Astotajā pārskata periodā tika apskatīta LBP metode ar fiksētu izmēra sejas attēlu un attēlu izpludināšana pirms apstrādes.
- Biometrisku datu iegūšanas algoritmu paralelizācija un implementēšana programmējamajos loģiskajos masīvos – iepriekš tika izveidota uz FPGA bāzēta attēlu ieguves sistēma; izstrādāta iespiedplate sejas atpazīšanas algoritmu testēšanai; projektēts 2D filtrs; veikta attēla filtrēšana reālā laikā; izveidota jauna sistēmas arhitektūra; izpildīta jaunā Aptina MT9V024 attēlu sensora montāža; apskatīta datora saskarne ar prototipu; izveidota attēlu filtrācija FPGA platformā; apskatīti varianti kā samazināt dubulto leņķi; uzsākta LCD kontroliera izveide uz FPGA platformas. Šajā periodā tika izstrādāti plaukstaru ģeometrijas algoritmi un implementēti tie FPGA, t.s., plaukstaru zīmīgāko punktu detektēšanas algoritma izstrāde.
- Biometrisku datu kriptēšanas, glabāšanas un lietojuma apakšsistēmas izveide – iepriekšējos pārskata periodos tika izanalizēti algoritmi *biohash* funkcijas skaitļošanai; izveidots algoritms, kas iegūst vidēji tādu pašu EER kā salīdzinot nešifrētus datus; izstrādāta programmatūra, kas salīdzina nešifrētus un šifrētus datus uz Java viedkartes; izstrādāts protokols, kas ļauj izveidot drošu sakaru kanālu starp viedkarti un autentifikācijas iekārtu; sastādīts ROI noteikšanas algoritms; apskatīts Rīda–Solomona kļūdu koriģējošs algoritms; izveidota viedkartes programma, kurā ir iespējams ierakstīt cilvēka datus; uzlabots BioHash algoritms; apskatīta un pielietota vektoru rotācija, kas implementēta FPGA. Astotajā periodā tika apskatīts datu papildināšanas paņēmieni nekautāla FIR filtra gadījumā.
- Atsevišķu sistēmas komponentu un bloku izstrāde un izpēte – iepriekšējos periodos tika veidotas funkcionālo bloku elektriskās principiālās shēmas *Altium Designer* vidē; izveidota principiālā viedkaršu komunikācijas maketa shēma; uzsākta kameras moduļa iespiedplates montāža; tika programmēti Aptina MT9V024 attēlu sensora reģistri; projektēts biometrijas sistēmas ietvars; izveidota iespiedplate viedkaršu saskarnes nodrošināšanai ar FPGA izstrādes rīka; programmēts mikrokontroleris MSP430; realizēti LBP, LDP algoritmi MATLAB vidē un veikta fāzes korelācijas algoritma izveide plaukstaru biometrijas datu pārbaudei.
- Multimodālas biometrijas tehnoloģijas koncepta definēšana – iepriekšējos periodos tika aprakstīts prototipa funkcionālo bloku darbības principi un savstarpējā sadarbība.

- Tehnoloģijas demonstratora eksperimentālā maketa montēšana – piektajā pārskata periodā tika veikta biometrijas prototipa 1.versijas montāža un visi sistēmas bloki tika ievietoti organiskā stikla ietvarā, veidojot vienotu prototipēšanas sistēmu. Sestajā pārskata periodā tika uzņemta datubāze un apstrādāti attēli, izmantojot esošo prototipu. Šajā pārskata periodā tika izveidots sejas atpazīšanas modulis un izveidota durvju atvēršanas sistēma
- Algoritmu implementēšana eksperimentālajā maketā – iepriekšējos pārskata periodos veiktas CMF modifikācijas; nomainīts maskas izmērs no 9x9 uz 15 x15; izmainīta pati maska; gūts priekšstats par VHDL parametrizējamām ķēdēm; uzsākta Fast NH-CMF2 realizācija; veikta horizontālā, vertikālā un 2D parametrizējamā filtra implementācija FPGA. Astotajā pārskata periodā tika strādāts pie FPGA sistēmas darbības stabilitātes paaugstināšanas, RAM buferu darbības stabilizācijas un PC komunikācijas stabilizācijas

Turpmāk dokumentā ir detalizēti aprakstītas katras uzdevumu grupas pētniecības darbības un rezultāti.

Aktivitāte: Lietiškie pētījumi

Biometrisku datu apstrādes algoritmu darbības izvērtēšana

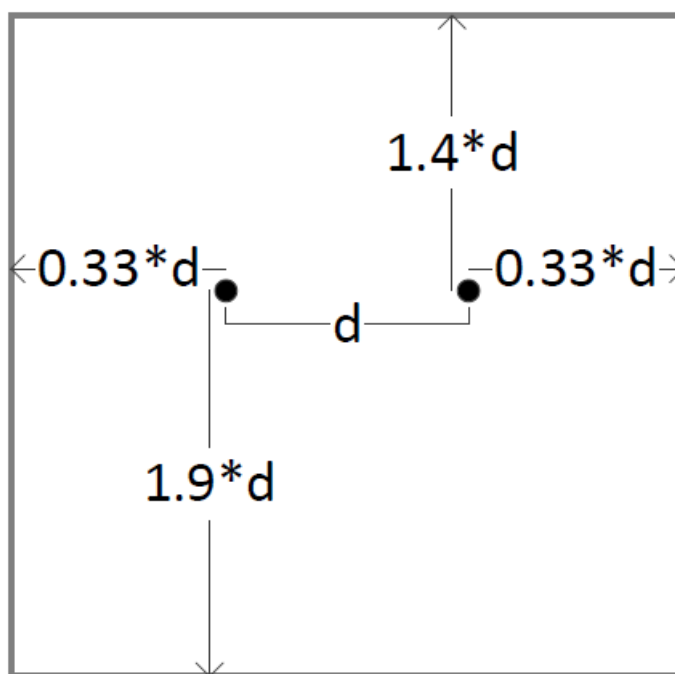
LBP ar fiksētu izmēra cilvēka seju

Tā kā ar LBP iegūtās histogrammas ir normalizētas, tad ir iespējams savā starpā salīdzināt dažādu izmēru sejas attēlus. Gan tādus, kur seja aizpilda visu no kameras iegūto attēlu, gan tādus, kur seja ir pēc iespējas tālāk no kameras. Šādai pieejai ir vairāki trūkumi. Pirmkārt, tā tiek samazināts atpazīšanas izpildes ātrums, jo sejas detektēšanas algoritmam ir jāpārbauda vairāki iespējamās sejas izmēri, līdz tiek atrasta seja. Otrkārt, tiek pieļauts, ka cilvēki var mēģināt autorizēties ar attēliem, kur sejas reģiona izmērs attēlā ir mazs un līdz ar to rezultātā iegūstamā informācija no neliela reģiona ir daudz mazāka. Pareizāka pieeja būtu ierobežot pieļaujamo sejas izmēru attēlā un pieņemt tikai tādus attēlus, kas atbilst noteiktajam intervālā.

Izvēloties atļauto izmēru intervālu, nedrīkst aizmirst par sistēmas lietojamību. Cilvēki ir dažāda garuma un līdz ar to, galva arī būs dažādos augstumos. Izvēlētajā intervāla mazākajai vērtībai ir jābūt pietiekami mazai, lai pieļautu cilvēka galvas pietiekamu vertikālu pārvietošanos attēlā, bet pietiekami lielai, lai netiktu samazināta atpazīšanas precizitāte.

No USB kameras iegūto attēlu izmērs ir 320x280. No kameras iegūtā sejas attēla sejas

reģiona attiecība (platums/garums) ir $\frac{0.9d + d + 0.9d}{1.4d + 1.9d} = \frac{2.8d}{3.3d} = \frac{2.8}{3.3}$. Kā sejas izmēra galvenais parametrs tiek izvēlēts sejas platums. Ar šādiem attēlu izmēriem un sejas attiecību, lielākais iespējamais sejas platums ir $280 / 3.3 * 2.8 = 237$.

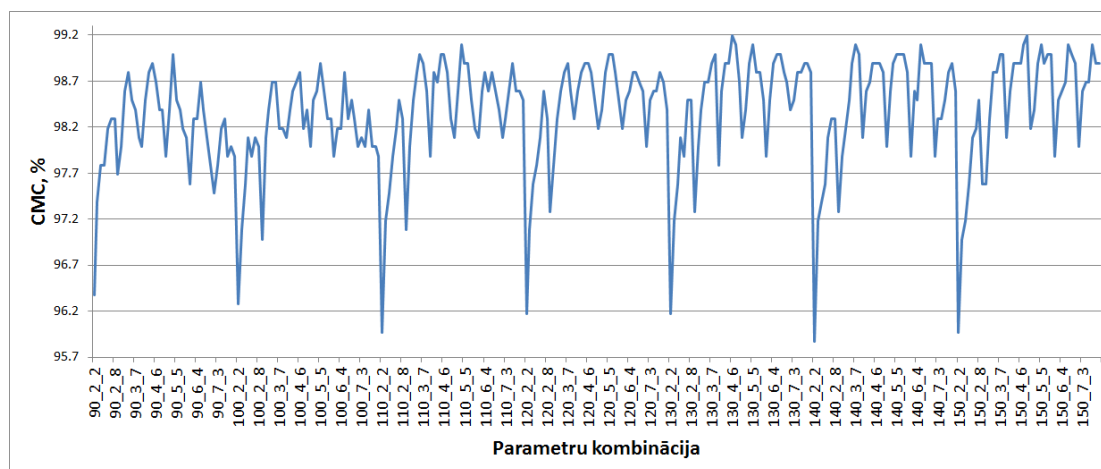


Izgrieztais sejas reģions izmantojot acis

Lai noskaidrotu optimālo sejas reģiona izmēru, tiek apskatīti dažādi sejas reģiona platumi, sākot no 50 pikseliem līdz 150 pikseliem. Vienlaicīgi paralēli tiks apskatītas dažādas LBP algoritma parametru (LBP rādiuss un reģionu daudzums) kombinācijas. LBP reģionu

daudzums tiek mainīts robežās [2;8] un LBP rādiuss tiek mainīts robežās [2;10]. Šādā veidā tiek apskatītas $16 * 7 * 9 = 1008$ dažādas parametru kombinācijas.

Parametru noskaidrošanai tiek izmantota FERET sejas attēlu datubāze. Tika izmantotas 992 personas, kur katrai ir divi attēli. Katram attēlam ir zināmas acu pozīcijas, līdz ar to tiek ignorēta sejas lokalizēšanas kļūda un ir iespējams koncentrēties tikai uz sejas atpazīšanas rezultātu. Tā kā datubāze satur augstas izšķirtspējas attēlus, tad nav iespējams tos izmantot bez priekšapstrādes. Pirms katra atpazīšanas mēģinājuma katru attēlu ir nepieciešams pagriezt un samazināt. Attēlu ir nepieciešams pagriezt, lai acis būtu vienā līmenī un sniegtu algoritmam ideālus apstākļus atpazīšanai. Zinot acu pozīcijas attēlā, ir iespējams noskaidrot cilvēka sejas platumu un līdz ar to ir arī zināms, cik ļoti ir nepieciešams samazināt attēlu, lai seja būtu nepieciešamajā platumā.



Rezultāti ir sakārtoti vispirms pēc sejas platumā, tad pēc reģionu daudzuma un visbeidzot pēc izmantotā rādiusa. Vislabākie rezultāti tek sasniegti divos punktos. Vienā, ja sejas platums ir 130 pikseļi un otrā, ja sejas platums ir 150 pikseļi. Tā kā rezultāti ir vienādi, tad ir loģiski izvēlēties parametru kombināciju, kurai ir mazāks sejas platums. Tādā veidā lietotājam būs vieglāk novietot savu seju pretī kamerai. Balstoties uz šī eksperimenta rezultātiem, turpmākajos testos sejas platums tiek noteikts 130 pikseļi, tiek izmantoti $K = 4$ reģioni un izmantotais rādiuss ir $R = 6$. Ar šādiem parametriem uz doto mirkli iegūtais rezultāts ir $CMC = 99.194\%$, $EER = 1.007\%$.

Attēlu izpludināšana pirms apstrādes

Uzņemot attēlus ir iespējams, ka attēlā kameras dēļ ir sastopami dažādi defekti, kas varētu pasliktināt atpazīšanas precizitāti. Lai to izlabotu, iespējamais risinājums ir attēlu izpludināt pirms LBP algoritma uzsākšanas.

Izpludināšana ir bieži izmantota attēlu apstrādes operācija, ko parasti izmanto, lai samazinātu trokšņus datos. Veids kā to izdarīt ir veikt konvolūciju starp ieejas attēlu un izpludināšanas kodolu, kur kodols apraksta cik daudz apkārtējos pikseļus ņem vērā un kādi ir to koeficienti.

Izpludināšanai tika izmantoti OpenCV pieejamie filtri, no kuriem tika izvēlēti sekojošie:

- Vidējās vērtības filtrs - vienkāršākais filtrs, kam kodolā visiem koeficientiem ir vienāda vērtība un līdz ar to aprēķina apkārtējo pikseļu vidējo vērtību,

- Gausa filtrs - kodolā koeficienti tiek piešķirti pēc divdimensionālas gausa funkcijas, lielākā vērtība ir attiecīgi kodola vidū un jo tālāk punkts ir no vidus, jo mazāka ir tā vērtība,
- Mediānas filtrs - iziet cauri visiem apkārtējiem pikseliem, kas tiek ņemti vērā un kā rezultātu atgriež mediānas vērtību no apskatītajām vērtībām,
- Bilaterālais filtrs - attēla izpludināšana strādā līdzīgi kā gausa filtram, bet tiek tā tiek veikta tā, lai netiktu izsmērētas attēlā redzamās malas.

Eksperimentu rezultātus ir iespējams redzēt tabulā. Novērtēšanai tiek izmantoti CMC un EER vērtēšanas kritēriji, kur CMC ir svarīgāks. Jo CMC ir lielāks, jo labāk, jo EER ir mazāks, jo labāk.

Izpludināšanas metode	Kodola izmērs	CMC, %	EER, %
Bez izpludināšanas	-	99.194	1.007
Vidējās vērtības filtrs	3	99.194	0.705
	5	98.993	0.705
	7	98.892	0.689
	9	98.792	0.811
	11	98.691	0.806
Gausa filtrs	3	99.194	0.705
	5	99.194	0.705
	7	98.993	0.705
	9	98.993	0.705
	11	98.691	0.704
Mediānas filtrs	3	99.194	0.705
	5	99.094	0.705
	7	98.892	0.663
	9	98.792	0.803
	11	98.792	0.705
Bilaterālais filtrs	3	99.194	0.987
	5	99.194	0.806
	7	99.194	0.758
	9	99.094	0.705
	11	98.993	0.805

CMC kolonā ir atzīmēti rezultāti, kas nav sliktāki par tiem, kas tiek iegūti bez izpludināšanas. Var redzēt, ka nav iegūti labāki rezultāti šajā kategorijā. Turpretī EER kolonā var redzēt, ka katrs rezultāts ir labāks nekā bez izpludināšanas. Vislabākie EER rezultāti atbilstoši vislabākajiem CMC rezultātiem ir atzīmēti. Lai arī ir četras vienlīdz labas dažādu filtru kombinācijas, gausa filtra rezultāti ir visstabilākie, jo atkārtoti spēj sasniegt labu EER vērtību. No abiem iespējamiem kodola izmēriem tiek izvēlēts "3". Tā kā mazāks kodols mazāk izpludina attēlu, tad šādā veidā tiktu samazināts risks attēlu pārāk izpludināt.

Iegūtajiem sapludināšanas rezultātiem ir mazāks uzlabojums nekā tika gaidīts. To gan varētu izskaidrot ar to, ka samazinot attēlus no augstākas izšķirtspējas uz mazāku, samazināšanas rezultātā tiek veikta līdzīga operācija attēla izpludināšanai. Katra samazinātā attēla pikseļa vērtību ir ietekmējuši vairāki pikseli no sākotnējā attēla.

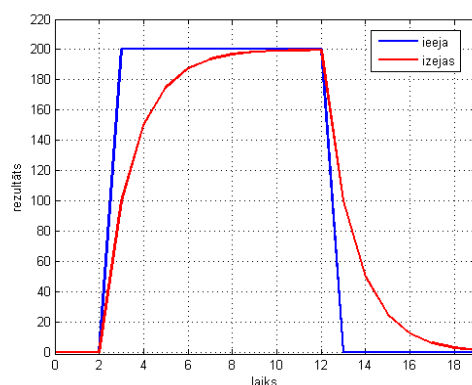
Kā redzams attēls sastāv no sešiem stāvokļiem {s0-s5}. Pirmais - {s0} ir gaidīšanas režīms, proti, kamēr lietotājs nav devis uzdevumu aprēķināt sliekšņa vērtību, netiek ievākta attēla histogramma un netiek veiktas citas operācijas. Ja lietotājs padod signālu {start = '1'}, notiek pāreja uz stāvokli {s1}, kurā tiek gaidīts kadra sākuma ({frame_start = '1'}) signāls. Šo signālu ģenerē attēlu sensors un brīdina, ka tiek uzsākta nākošā attēla iegūšana. Ir būtiski sagaidīt kadra sākuma signālu, jo tikai tad tiek ģenerēts signāls ({h_en}), kurš atļauj ievākt informāciju par attēlu - attēla histogramma. Sagaidot {frame_start = '1'} signālu sistēma pāriet stāvoklī {s2}, kurā tiek ievākta attēla histogramma. Tas notiek tik ilgi, kamēr netiek sagaidīts signāls {frame_end = '1'}, kurš liecina par kadra beigām un to, ka pikseli vairs nesešos. Sistēma pāriet stāvoklī {s3}, kurā tiek analizēta histogramma un meklēta maksimālā vērtība. Ar salīdzināšanas metodi tiek iziets cauri katrai vērtībai intervālā no 0 līdz {cnt_reg = 255} (intensitāšu līmeņu skaits). Atrastās maksimālās vērtības adrese tiek saglabāta un sistēma nonāk stāvoklī {s4}. Atkal tiek meklēta maksimālā vērtība, tikai ierobežotā intervālā, proti, adrešu vērtības, kuras atrodas ± 30 adrešu diapazonā ap stāvoklī {s3} atrasto maksimālo vērtību netiek ņemtas vērā. Tādējādi atrodot otru maksimālo vērtību, kad skaitītājs ir izgājis cauri visām vērtībām ({cnt_reg = 255}) sistēma nonāk stāvoklī {s5}, kurš ir noslēdzošais. Šajā stāvoklī tiek meklēta minimālā intensitāšu vērtība atmiņas apgabalā starp maksimālajām vērtībām no stāvokļiem {s3} un {s4}. Atrastās minimālās vērtības adrese ir arī sliekšņa vērtība - {threshold}, sistēma atgriežas gaidīšanas stāvoklī.

Iegūtie rezultāti sliekšņojot attēlu FPGA:



Divu dažādu personu plaukstu binarizētie attēli.

Binārizācijas procesam noritot reālajā laikā ir novērojama stipra attēla raustīšanās – binarizēts attēls nav vienmērīgs. Tas izskaidrojams ar straujo sliekšņa vērtību nomainīšanos. Lai novērstu raustīšanos un nevienmērīgu attēla binārizāciju, tika pielietots RC tipa ķēdes princips.

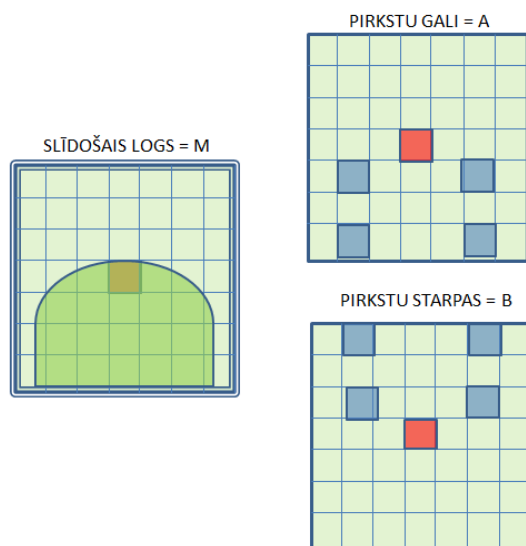


Šāda pieeja nozīmē, ka sliekšņa vērtība tiks pakāpeniski integrēta, nevis strauji nomainīta tajos momentos, kad tiek aprēķināta katra jaunā vērtība. Katra nākamā vērtība tiek pakāpeniski piemērota pēc vairāku kadru aiztures. Rezultātā tiek iegūta stabilizēts binārs attēls sistēmai darbojoties reālajā laikā.

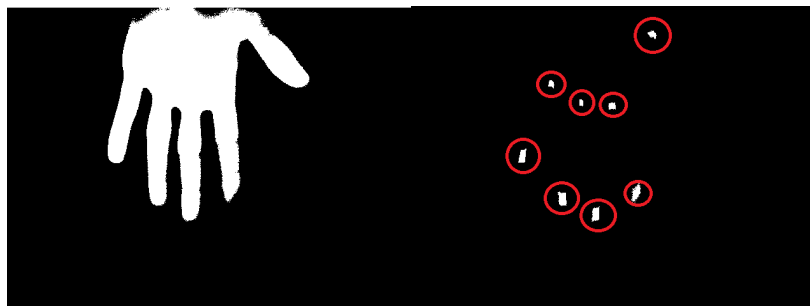
Plaukstu zīmīgo punktu detektēšana

Plaukstu zīmīgo punktu detektēšanai (pirkstu gali, pirkstu starpas) tika modelēts algoritms MATLAB, kurš tika arī realizēts FPGA.

Algoritma princips balstās uz morfoloģisko operāciju pielietošanas binarizētam attēlam. Veicot strukturējošo logu salīdzināšanu ar attēla fragmentu pielietojot loģiskās operācijas, izejā var iegūt pirkstu starpas un pirkstu galus. Attēla fragments ir slīdošais logs, respektīvi, attēlā katrs pikselis un tā blakus pikseļu vērtības tiek pārbaudītas ar strukturējošo logu. Intuitīvi pielietojamie strukturējošie logi pirkstu galu un starpu detektēšanai ir pakavveida formas kā attēlā parādīts.

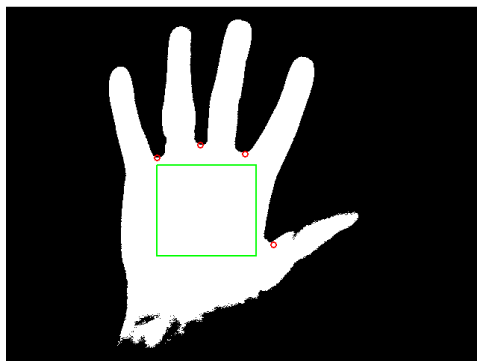


Lai iegūtu izejas attēlu nepieciešams veikt UN operāciju katram strukturējošajam logam A un B ar attēla fragmentu M. Izejas pikseļa vērtība = '1', ja $(M \cup A) \text{ XOR } (M \cup B)$, pretējā gadījumā '0'. Šāda algoritma realizācijai tiek izmantots binārais SPARSE filtrs, tikai atetos reizināšanas un tree addrer daļu. Šāda attēla filtrācija – zīmīgo punktu iegūšanas procedūra FPGA norit reālajā laikā. Attēlos var redzēt iegūtos rezultātus ar plaukstu attēlu:

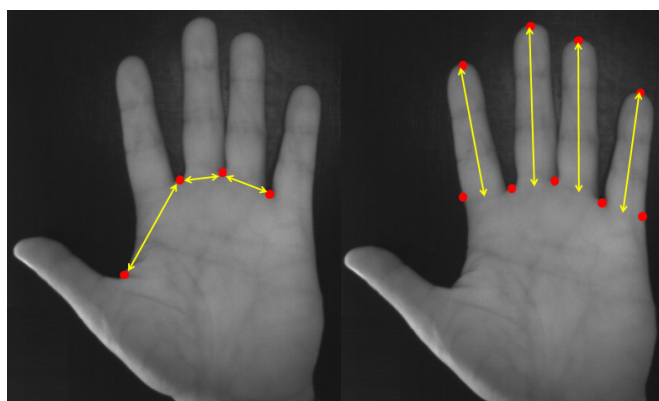


Simulācijā MATLAB algoritms ir izstrādāts līdz stadijai, kad tiek iegūti katras plaukstu zīmīgie punkti un apvienoti vienā vektorā, salīdzināšanai ar datu bāzi. Izmantojot pirkstu starpas ir

vienkārši ievilkt ROI plaukstā, ierobežojot reģionu citu algoritmu pielietošanai, kuri iegūst citus plaukstu parametrus kā asinsvadu tīklojumu.



No pirkstu starpu un galu detektēšanas algoritma, izmantojot morfoloģiskās operācijas, līdz šim tiek iegūti sekojošie plaukstu ģeometriskie parametri:



Izmantojot šos plaukstu parametrus un aprēķinot distanci starp tiem izmantojot Eiklīda distanci algoritms tika notestēts uz 6 personu datubāzi, kur katrai personai bija pa 2 attēli. Rezultātā tika secināts, ka šāds algoritms spēj atsijāt līdzīgākās plaukstu no datubāzes. Šajā gadījumā izvēloties sliekšņa vērtību $T = 0.97$ salīdzināšanas procesā starp 6 personām, tika panākts ka sistēma atsijā tikai salīdzināmās personas attēlu.

Turpmākie pētījumi:

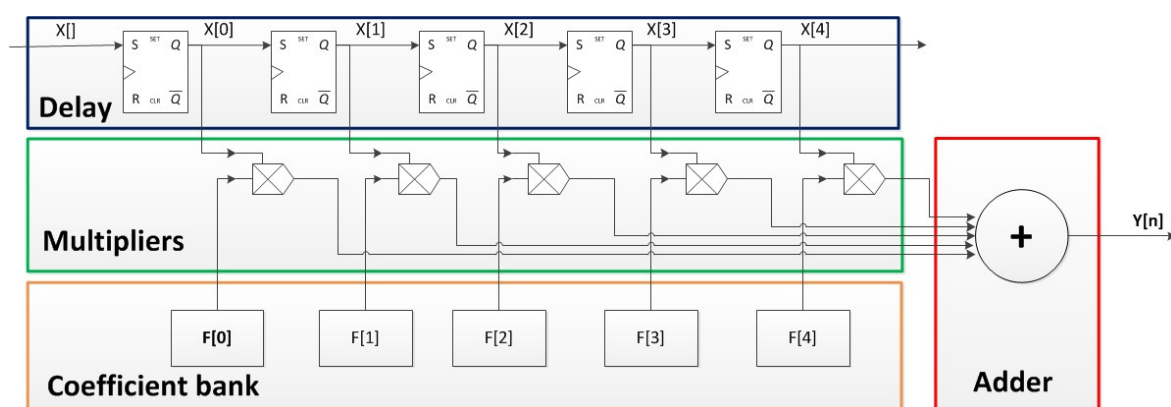
- Eksperimentālie pētījumi ar lielāku datubāzi;
- Algoritma uzlabošana zīmīgo punktu atrašanai un plaukstu ģeometrijas parametru iegūšanai.

Biometrisku datu kriptēšanas, glabāšanas un lietojama apakšsistēmas izveide

Datu papildināšanas paņēmiens nekauzāla FIR filtra gadījumā

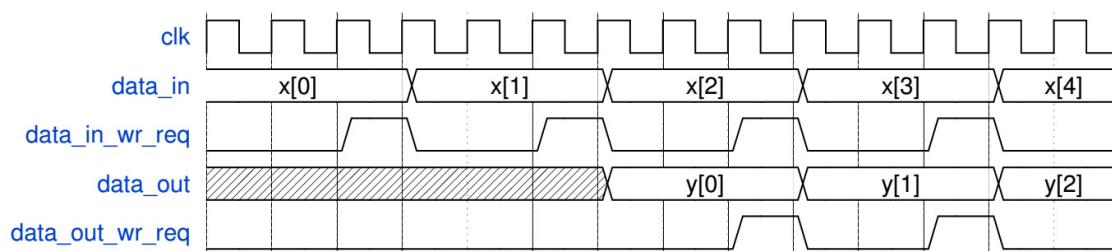
Nekauzāla FIR filtra gadījumā, neveicot ieejas attēla papildināšanu, uzsākot filtrāciju izejas dati (kas ir vienādi ar filtra impulsa reakcijas garuma pusi) ir nepareizi, tos standartgadījumos vienkārši ignorē. Ja ar šādiem filtriem ir jāapstrādā bilde, tad filtrētās bildes izmēri samazinās atkarībā no izmantotās filtra maskas garuma. Ir gadījumi, kad šādi filtri tiek slēgti kaskādē un tā rezultātā gala bildes izmēri ir pārāk mazi. Lai šo problēmu risinātu ir jāizveido attēla papildināšana, kas samazina kropļojumus attēla malās.

Ir vairākas iespējas kā izveidot "palīgshēmas", kas veic iepriekš izvēlētu signāla papildināšanas veidu. Pirms ieslīgt detaļās, apskatīsim izmantoto pieeju attēla filtrācijai uz FPGA bāzes.



1D FIR filtra realizācija, neizmantojot ieejas signāla papildināšanu

Attēlā tiek ilustratīvi parādīta 1D FIR filtra arhitektūra, kas ir realizējama FPGA. Ir redzamas četras sastāvdaļas – aiztures līnija (Delay), reizinātāji (Multipliers), koeficienti (Coefficient bank), kā arī vairāku ieeju saskaitītājs (Adder). Attēlā ir parādīts piektās kārtas 1D FIR filtrs. Attēlotā 1D FIR filtra arhitektūra ir pielietojama gan kauzālu, gan nekauzālu filtru realizācijai. Vienā vai otrā gadījumā ir atsevišķi papildinājumi. Tā, piemēram, nekauzāla 1D FIR filtra gadījumā izejas dati tiek padoti tikai pēc tā, kad filtrā ir iebīdījušies dati līdz pusei. Lai to panāktu tiek izmantots bīdes reģistrs datu pieprasījuma signāla ieejai (data_in_wr_req).

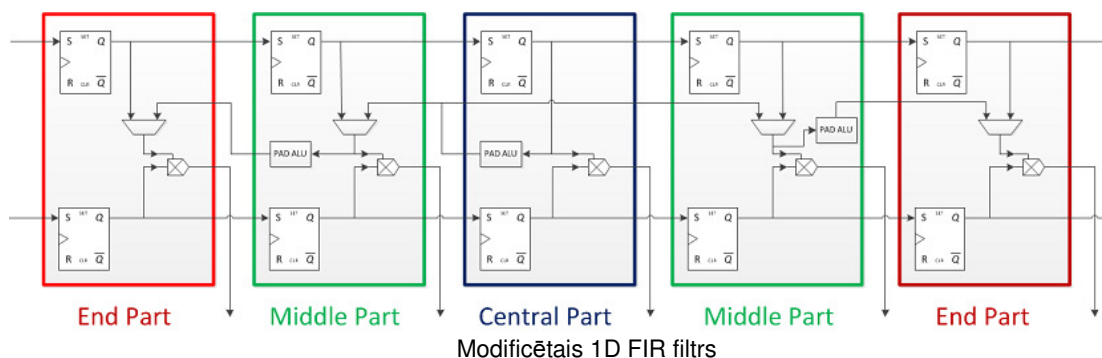


Datu pieprasījuma signāla nobīdes laika diagramma

Tāpat, papildinot šo shēmu, var filtrēt arī 2D signālus jeb attēlus. Vienkāršības labad "signālu papildināšana" tiks attēlota 1D FIR filtra gadījumā.

Esošā filtra realizācija tiek papildināta ar multipleksoriem, kas nodrošina datu izvēli starp iepriekšējās kaskādes papildināto signālu vai arī no datu reģistra, un aritmētiskiem-loģiskiem mezgliem "PAD ALU" (ALU – arithmetic logic unit), kas veic ieejas signāla modifikācijas

papildinātajā apgabalā pēc lietotāja uzstādījumiem.



Kā redzams no attēla, tad 1D FIR filtram ir trīs galvenās sastāvdaļas – centrālā daļa, vidusdaļa, beigu daļa. Kā jau tika minēts multipleksori un "PAD ALU" nodrošina gan ieejas signālu pievadīšanu, gan arī tā vērtības izrēķināšanu – atkarībā no uzstādījumiem. Papildus šim attēlam, kurā ir attēlota 1D FIR filtra arhitektūru ir nepieciešami papildus sastāvdaļas, kas kontrolē gan multipleksoru, gan "PAD ALU" darbību.

Šiem kontroles moduļiem ir jābūt aktīviem tikai divos gadījumos:

1. Filtrācijas sākumā – ieejas signāla dati ir nokļuvuši līdz filtra centrālajai daļai.
2. Filtrācijas beigās – 1D FIR filtram kādu noteiktu laika sprādi (atkarīgs no situācijas) ieejā nepienāk jauni signāla dati.

Šajos gadījumos ir jākontrolē gan multipleksori, gan "PAD ALU". Attiecīgi pirmajā gadījumā tiek kontrolēta 1D FIR filtra labā puse pēc centrālās daļas, savukārt otrajā – kreisā daļa līdz centrālajai daļai.

Nosacījumi, kad ieslēgt "labo" daļu:

- Ieejas datu ierakstīšanas pieprasījuma signāls (DATA_IN_WR_REQ) tiek aizturēts par pusi no filtra maskas garuma.
- Līdzko aizturētajā pieprasījuma signālā parādās loģiskais vieninieks, tas nozīmē, ka ir bijuši vismaz tik daudz ieejas datu pieprasījuma signālu, lai filtrs sāktu ģenerēt izejas datus. Šajā momentā tiek saslēgti multipleksori.
- Pēc katra nākamā ieejas datu ierakstīšanas pieprasījuma multipleksori tiek pārslēgti citā stāvoklī (ieejas datus ņem no datu reģistriem).
- Tas tiek darīts līdz tiek sasniegta pēdējā kaskāde.

Nosacījumi, kad ieslēgt "kreiso" daļu:

1. Ir izveidots skaitītājs, kas skaita impulsus, cik ilgi nav pienācis ieejas datu ierakstīšanas pieprasījums, ja šis intervāls pārsniedz iepriekš noteikto, tad tiek atļauta papildshēmas darbība;
2. Pēc šī momenta mākslīgi tiek uzģenerēti pieprasījuma signāli, kas izbīda filtrā palikušos datus, kā arī katrai šūnai, no kuras tika izbīdīti dati, tiek pārslēgts multipleksors, kas nodrošina datu papildināšanu;
3. Operācija tiek atkārtota līdz kamēr tiek izbīdīti dati līdz centrālajai daļai.

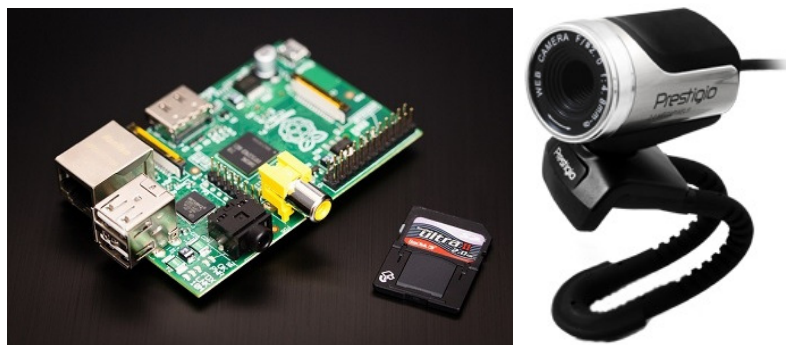
Līdzīgā veidā šo pieeju var attīstīt arī 2D FIR filtriem aizvietojo ar aktīvās rindas ar iepriekšminēto realizāciju.

Aktivitāte: Eksperimentālā izstrāde

Tehnoloģijas demonstratora eksperimentālā maketa montēšana

Sejas atpazīšanas modulis

Veidojot sejas atpazīšanas moduli, lai iegūtu cilvēka sejas attēlus, tika izvēlēta USB kamera Prestigio PWC2. Kā sejas attēla apstrādes algoritms ir izvēlēts LBP.



Šīs iekārtas programmatūra tiek rakstīta C++ valodā un tiek izmantota attēlu apstrādes bibliotēka OpenCV. Bibliotēka tiek izmantota, lai: iegūtu attēlus no USB kameras, izmantotu matricu klasi, izmantotu dažādas lineārās algebras operācijas un lietotu Hāra klasifikatoru sejas detektēšanai un lokalizēšanai.

Kad veiksmīgi ir iegūts kameras attēls, ir nepieciešams to pārveidot melnbaltā formātā, jo pārsvara visi algoritmi izmanto tieši šo formātu. Pirmais solis, ko ir nepieciešams veikt ir lokalizēt seju, jeb noskaidrot, kur attēlā tā atrodas. Vislabākais veids kā to varētu izdarīt ir izmantojot acis. Pēc to savstarpējā attālumu d ir iespējams pielāgot sejas reģiona izmēru, kas aprakstītu reģionu attēlu, kur atrodas atpazīstamā sejas informācija. Gadījumā, ja cilvēks ir sagāzis attēlā galvu uz sāniem, tad izmantojot acu koordinātes ir iespējams pagriezt attēlu tā, lai acis būtu vienā līmenī. Diemžēl problēma ir tāda, ka izmantotais acu lokalizēšanas algoritms (OpenCV Hāra kaskādes, kas var atrast acis) ne vienmēr spēj atrast acis, un pat ja atrod, tad ar nelielu nobīdi.

Ar OpenCV kaskādēm ir arī iespējams izmantot sejas detektoru, kas norādītajā attēlā atrod cilvēka seju un atrod atrastās sejas reģionu kā kvadrātu. Līdz ar to, tā vietā, lai no sākuma sameklētu attēlā acis un tad no tām izteiktu sejas reģionu, uzreiz tiktu atrast sejas reģions. Šādai pieejai gan ir vēl viena problēma. Cilvēka galva varētu būt sagriezta uz sānu. Sejas detektors šādu lietu nepamana, tas vienmēr izgriež reģionu, kura malas ir paralēles attēla malām. Līdz ar to lietotājiem vajag piedomāt pie tā, lai galva tiktu turēta tā, lai acis būtu vienā līmenī. Šī uzdevuma veikšanai var tikt izmantoti palīg līdzekļi: spogulis aiz kameras, lai cilvēkiem būtu vieglāk novērtēt savas galvas stāvokli, vai gadījumā, ja tiek izmantots monitors, kur tiek attēlots kamerā redzamais attēls, pa virsu attēlam uzzīmē līnijas, kas palīdzētu novietot acis vienā līmenī.

Sejas detektora atgrieztais reģions ir kvadrāta formā. Šāda sejas reģiona forma šim projektam nav derīga. Tomēr ir iespējams to pielāgot vajadzīgajām proporcijām. Kvadrāta

formas reģionam ir iespējams izveidot saistību starp tā izmēru un sejas platumu. Izmantojot šo informāciju ir iespējams atgriezt reģionu pārveidot nepieciešamā formā.

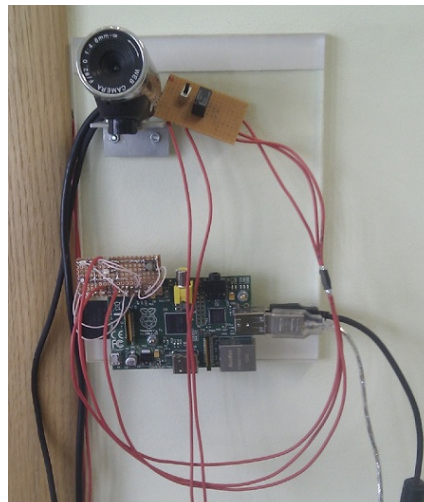
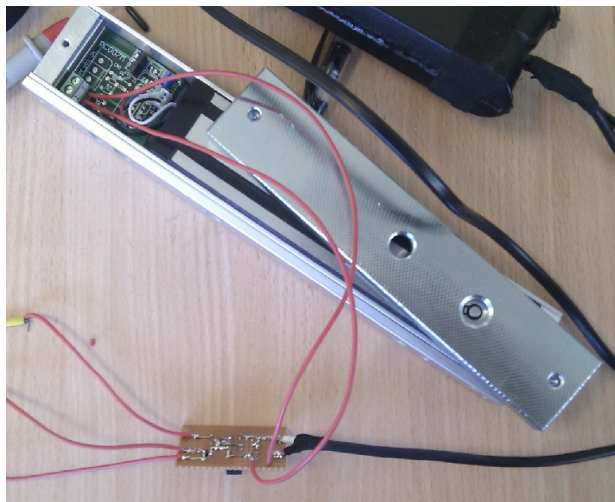
Kad ir beidzot iegūts sejas reģions, ir iespējams noskaidrot sejas platumu. Ja tas ir atļautajā intervālā, tad ir iespējams turpināt attēla apstrādi, ja nē, tad attēlu vajag uzņemt vēlreiz.

Izgrieztajam attēla reģionam tiek pielietota LBP funkcija. Rezultējošais attēls tiek sadalīts 25 vienāda izmēra taisnstūra veida reģionos. Sadalot attēlu reģionos tiek iegūtas katra apgabala histogrammas. Katra histogramma satur 256 kolonas, bet viss attēls satur 6400 kolonas. Šī apvienotā histogramma ir arī informācija, kas tiek izmantota divu cilvēku salīdzināšanā.

Programmai failu formā glabājas katra cilvēka vairāki etalona attēli. Attēli ir uzņemti dažādās pozīcijās, lai palielinātu iespēju, ka cilvēks tiks atpazīts pareizi. Ātrdarbības dēļ visi attēli ir jau iepriekš apstrādāti un no tiem iegūtas histogrammas. Tās tiek ielādētas startējot programmu. Kad persona cenšas tikt atpazīta sejas modulī un ir aprēķināta jaunā attēla histogramma, tad ir nepieciešams to salīdzināt ar lokālo attēlu datubāzi. Histogramma tiek salīdzināta ar visām datubāzes histogrammām un noskaidrota vislīdzīgākā. Ja šī lielākā līdzība ir lielāka par sliekšņa vērtību, tad tiek atzīts, ka paraugs ir pieņemts, ja ir mazāka, tad paraugs tiek noraidīts un cilvēks tiek aicināts mēģināt vēlreiz autorizēties.

Durvju atvēršanas sistēmas

Izmantojot izstrādāto sejas atpazīšanas moduli, ir izveidota automatizēta durvju atvēršanas sistēma. Durvju atvēršana un aizvēršana tiek veikta ar elektromagnētisko slēdzeni. Pie durvīm ir piestiprināts metāla gabals un pie durvju stenderes ir piestiprināts elektromagnēts. Kad elektromagnētam tiek padota strāva, elektromagnēts strādā un durvis nav iespējams atvērt.



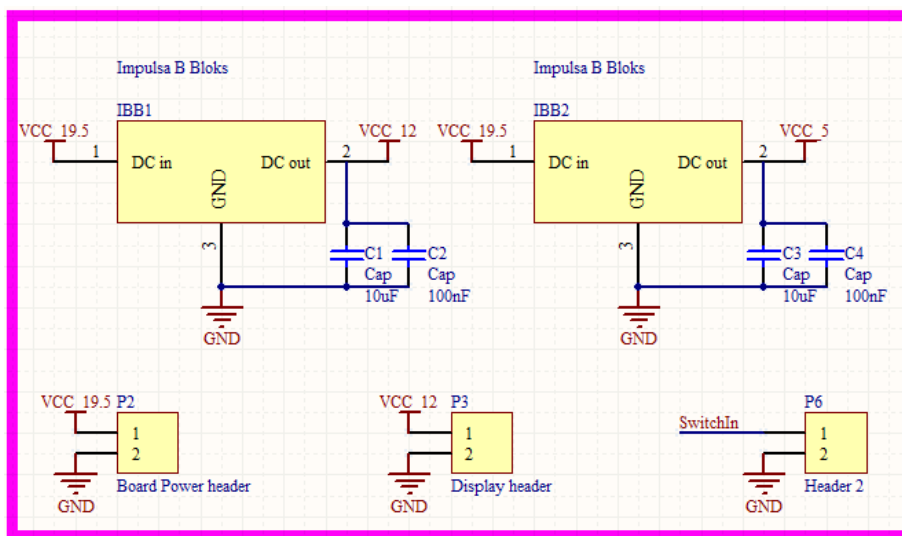
Elektromagnētiskā slēdzene

Sistēma tiek vadīta izmantojot Raspberry Pi GPIO izvadus. Lai varētu nosūtīt signālus pa attiecīgajiem izvadiem, tiek izmantota bibliotēka *WiringPi* [\[cite{wiringPi:Online}\]](http://wiringpi.com/).



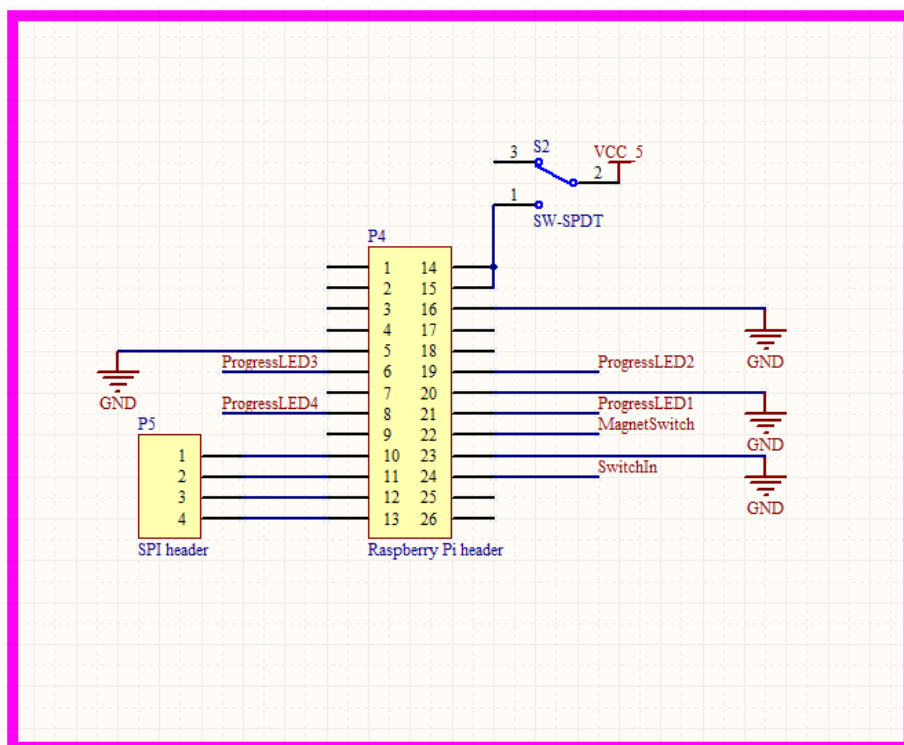
Raspberry Pi izvadi

Jaunizveidotās plates principiālā shēma, kura uzskatāmības dēļ ir sadalīta četrās daļās, un iespiedshēma ir veidota iekš Altium Designer. Rezultējošās iespiedplates platums ir 57.4 mm un garums ir 38.1 mm.



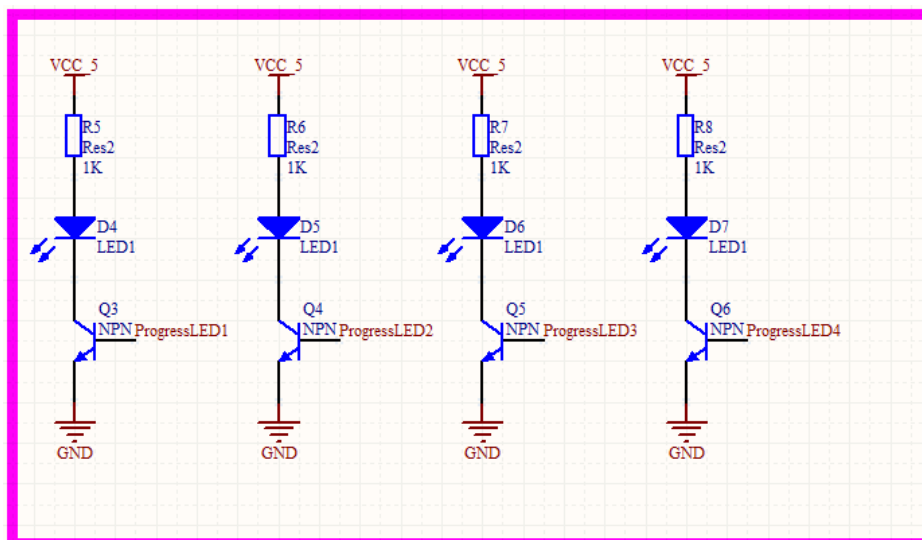
Plates principiālās shēmas pirmais izgriezums

Platei pienāk viens 19.5V barošanas vads, kas tiek sadalīts priekš Raspberry Pi (5V), monitora (12V) un durvju elektromagnēta (19.5V). Strāvas pārveidei tiek izmantoti divi impulsa barošanas bloki. Viens, kas pārveido no 19.5V uz 5V un otrs, kas pārveido no 19.5V uz 12V. Abiem barošanas blokiem pie strāvas izejas ir divi kondensatori, lai izlīdzinātu iegūto strāvu. Lai samazinātu nepieciešamo vadu daudzumu, Raspberry Pi barošana tiek nodrošināt nevis caur USB portu, bet gan caur GPIO izvadu. Shēmas izgriezumā ir redzama pienākošās barošanas pieslēgvietā, monitoram izejošās barošanas pieslēgvietā un pieslēgvietā slēdzim, kas pados signālu uz Raspberry Pi.



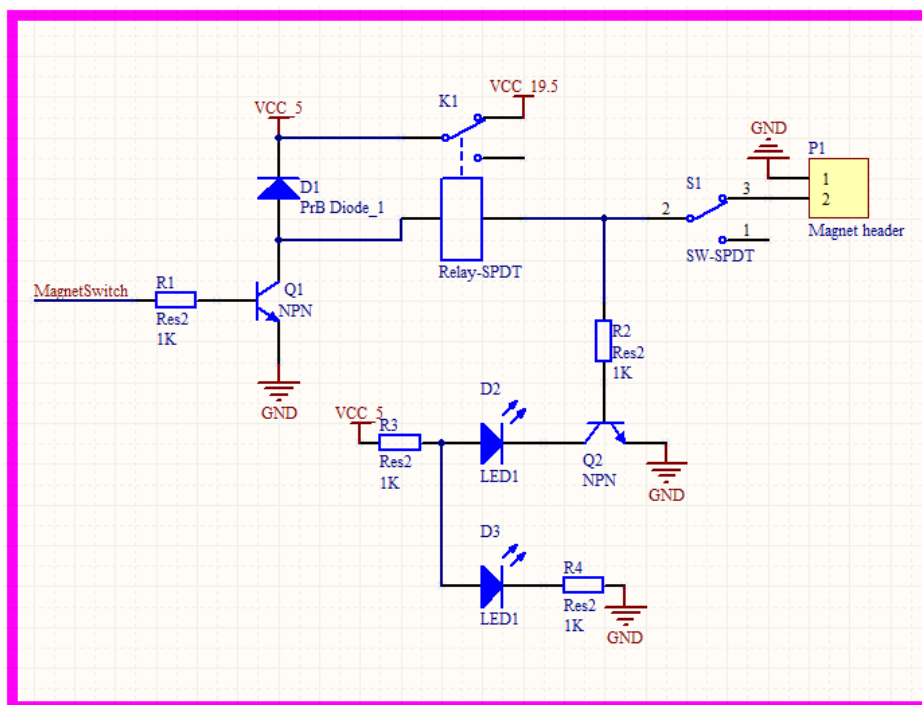
Plates principiālās shēmas otrais izgriezums

Shēmas otrajā izgriezumā ir redzami ir redzami izvadi no Raspberry Pi GPIO izvadiem. Caur slēdzi pienāk 5V barošana, lai būtu iespējams iekārtu izslēgt. Četri no GPIO izvadiem tiek izmantoti gaismas diodu spīdināšanai. 3.3V signāls tiek padots, ja vēlās ieslēgt attiecīgo diodi. Šeit ir iespējams arī redzēt izvadu, kas tiek izmantots slēdža signāla uztveršanai. Četri izvadi tiek izmantoti SPI komunikācijai un aizved attiecīgi uz SPI izvadiem uz plates. Un visbeidzot viens izvads tiek izmantots elektromagnētiskās slēdzenes izslēgšanai, ja vēlās izslēgt slēdzeni, tiek padots 3.3V signāls.



Plates principiālās shēmas trešais izgriezums

Shēmas trešajā izgriezumā ir iespējams redzēt gaismas diodes. Diodes ieslēgšanai tiek izmantot tranzistori, kas ieslēdzās pie padotā signāla.



Plates principiālās shēmas ceturtais izgriezums

Releju, kas tiek izmantots elektromagnētiskās slēdzenes strāvas izslēgšanai ir iespējams redzēt ceturtajā izgriezumā. No Raspberry Pi tiek padots signāls uz tranzistoru, kas noslēdz ķēdi un ar 5V signālu pārslēdz releju. Pie elektromagnēta barošanas izvada ir vēl viens slēdzis, lai būtu iespējams atslēgt durvis bez Raspberry Pi palīdzības. Pirms slēdža ir divas gaismas diodes, kas norāda vai durvis ir aizslēgtas vai atslēgtas. Ja elektromagnēta strāvas ķēde ir noslēgta, tad iedegās viena diode, bet ja nav noslēgta, tad iedegās otra diode.

Lai būtu iespējams lietotājam attēlot, kādu attēlu redz kamera, tiek izmantots 7 collu monitors, kas tiek piesprausts pie Raspberry Pi analogās video izejas. Ar monitora palīdzību lietotājs var pārliecināties, ka viņa seja ielien attēlā un galva nav sagriežta vai sagāzta uz vienu vai otru sānu.

Lai varētu atvērt durvis, lietotājam ir nepieciešams nostāties pretī kamerai un nospiest pogu uz plates. Nospiešanas rezultātā tiek uzņemts attēls. Uzņemto attēlu apstrādā un salīdzina pret lokālo datubāzi. Ja attēls sakrīt ar kādu no saglabātajiem attēliem, tad durvju elektromagnēts tiek atvienots uz piecām sekundēm.

Turpmākie plāni:

1. Plates salodēšana;
2. Sistēmas integrēšana pārējā modulī.

Algoritmu implementēšana eksperimentālajā maketā

FPGA sistēmas darbības stabilitātes paaugstināšana

Problēmas nostādne

Pēc Quartus II izstrādes vides datiem, veidojamās sistēmas apjoms un resursu patēriņš izmantojamā FPGA mikroshēmā (EP4CE115F29) aizņem jau tuvu pie 30% (att. M1).

Total logic elements	36,827 / 114,480 (32 %)
Total combinational functions	28,433 / 114,480 (25 %)
Dedicated logic registers	21,912 / 114,480 (19 %)
Total registers	21958
Total pins	238 / 529 (45 %)
Total virtual pins	0
Total memory bits	463,764 / 3,981,312 (12 %)
Embedded Multiplier 9-bit elements	163 / 532 (31 %)
Total PLLs	1 / 4 (25 %)

FPGA resursu izmantošana vienai no projekta pēdējām revīzijām

Pie šādiem nosacījumiem, izstrādes vidē esošajiem kompilācijas rīkiem, konkrēti ir runa par t.s. "Fitter", kura uzdevums ir fiziski izvietot projektā aprakstīto loģisko shēmu (ar visiem mezgliem un to starpsavienojumiem) dotajā FPGA ierīcē. Šis uzdevums nav triviāls, jo Fitter rīkam tiek uzdots vienlaicīgi optimizēt vairākus izveidotās loģiskās shēmas parametrus, t.i.:

- palielināt katra mezgla ātrdarbību,
- saīsināt mezglu starpsavienojošos vadus (lai signāliem nerastos liekas aiztures).

Izvēlētajā ierīces darba režīmā datu plūsmas nodrošināšanai (starp ierīces mezgliem) tiek izmantoti šādi signāli:

1. Vadības signāli:

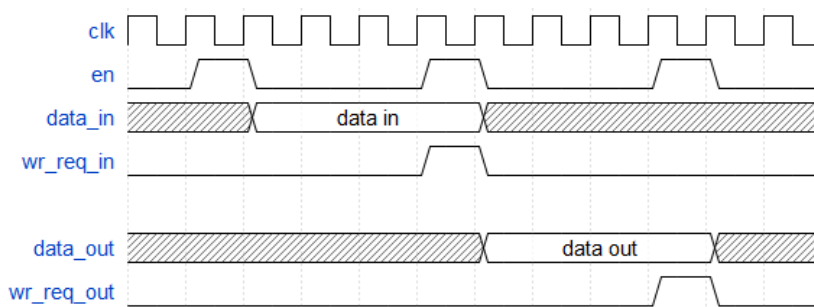
1.1. clk (takts signāls) – uzdod ierīces maksimālo darba frekvenci f_{clk} un laika apgabalu (clock domain);

1.2. en (rīcību atļaujošais signāls) – tiek izmantots, lai sasinchronizētu vairākus moduļus, kas strādā vienā laika apgabalā, bet ar zemāku frekvenci par f_{clk} (jo, atļaujot moduļiem rīkoties tikai noteiktajos laika momentos, to darbības frekvence tiks dalīta). Šis signāls ir aktīvs tikai tajos clk periodos, kad mezglam, kuram tas tiek padots, ir jāmaina savs stāvoklis;

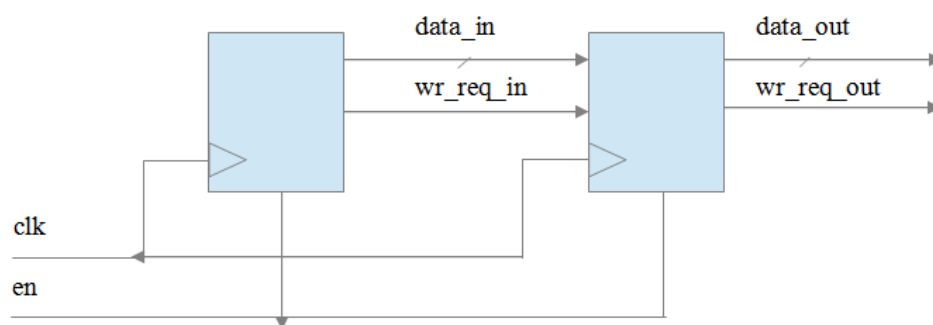
1.3. wr_req, rd_req (rakstīšanas, lasīšanas atļaujošie signāli) – pēc būtības ir en signāla īpašgadījums, kas atļauj ierīcei veikt papildus funkcijas, kuras vienkārši ar en netiek atļauts darīt. Šie signāli tiek ģenerēti tikai vienlaicīgi ar en signālu un bez tā tiek ignorēti.

2. Informatīvie signāli – nes sevī jebkura veida citu informāciju, nekā vadības signāli.

Tipiskā laika diagramma vienai datu transakcijai veidotajā FPGA sistēmā izskatās sekojoši:

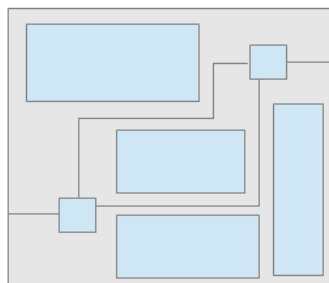


Tipiskā datu transakcija vienam funkcionālam mezglam ar viena en perioda izpildīšanas laiku



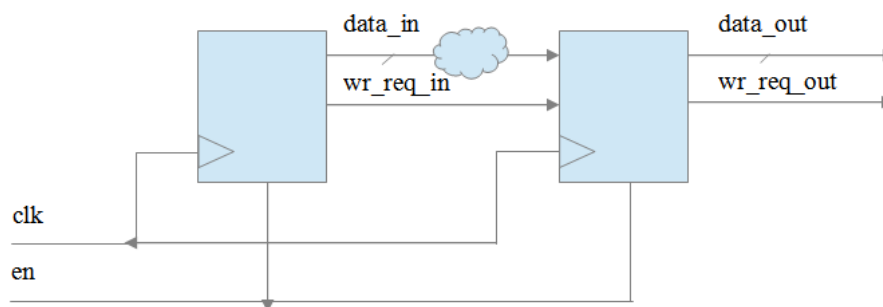
Datu transakcijai, kas norit starp diviem mezgliem, izmantojamie signāli

Jāsaprot, ka pie iepriekš minētas ierīces sarežģītības reālais (uz FPGA) mezglu izvietojums nebūs vienmēr tik izdevīgs, kā gribētos; kā arī paši funkcionālie mezgli varētu tikt realizēti pa daļām un “izmētāti” pa FPGA iekšpusi, nevis koncentrēti.



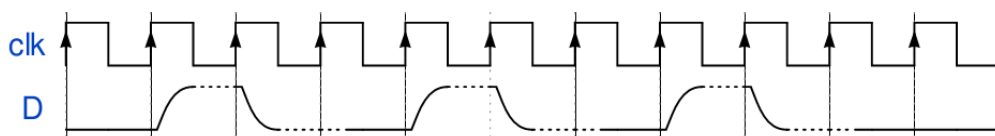
Piemērs FPGA izkārtojumam čipā ar gariem mezglu starpsavienojumiem

Situāciju vēl vairāk sarežģī tas, ka tiek izmantoti asinhronie datu apstrādes mezgli:



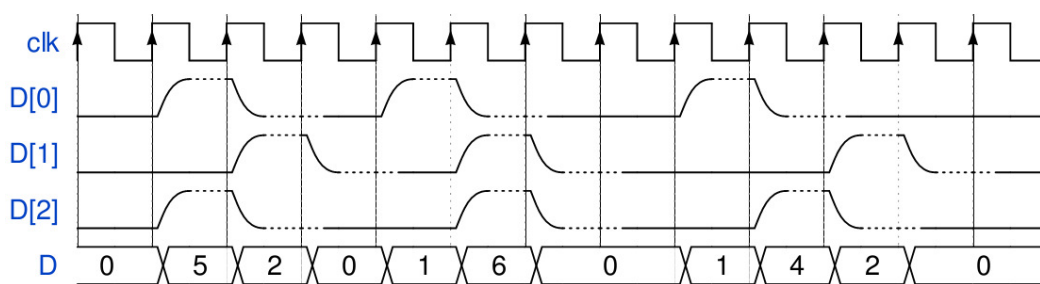
Divi datu apmaiņas mezgli ar asinhronu datu apstrādi data_in līnijā

Starpsavienojumu garumi, asinhronie datu apstrādes mezgli, kā arī sinhrono mezglu kombinatoriskās loģikas daļas ievieš aiztures pārraidāmajos signālos. Šo aizturu dēļ, pārraidītie virknes signāli tiek aizturēti (nobīdīti) laikā attiecībā pret to vadības signāliem (clk, en, wr_req, u.tml.).



Virknes signālu kropļojumi

Paralēli pārvadājamiem signāliem notiek arī laika nobīdes starp dažādu bitu pārraidošiem signāliem.



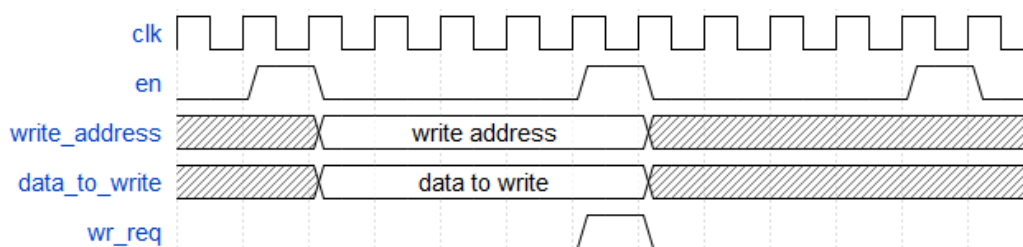
Paralēlo signālu kropļojumi

Veidojamajā sistēmā, virknē kodētie signāli netiek pārvadīti, tāpēc aktuāla ir paralēli kodēto signālu kropļojumu problēma. Jūtīgākie mezgli ir tie, kas strādā ar maksimālo frekvenci (125MHz) vai veic sarežģītu darbu, izmantojot īsākus vadības signālus, piemēram, tie, kam ir jāieraksta dati wr_req laikā (125MHz takts frekvencei tās ir 8ns) vai jānolasa dati tikpat īsā intervālā.

Šie mezgli ir: RAM bitu izmantojošie aiztures buferi un PC komunikācijas moduļa bloka atmiņas kanāli.

RAM buferu darbības stabilizācija

Tipiskā signālu diagramma rakstīšanai RAM buferos ir parādīta attēlā:



RAM bitiem pienākošie rakstīšanas signāli

Tik sarežģītai sistēmai (izmantojot ap 30% FPGA resursu) wr_req impulsa laikā (8ns) dati var nepaspēt ierakstīties vajadzīgajā RAM bitu šūnā pilnībā, bet gan tikai daļēji. Par šādu parādību liecina daļēji kropļotie dati, kas ir dabūjami attēla filtra (kas satur sevī vairākas RAM bitu aiztures līnijas) izejā, kad sistēmai pieslēdz Signal Tap Logic Analyzer, kas aizņem vēl

vairāk FPGA resursu un līdz ar to rada vēl vairāk aiztures pārvadāmiem signāliem. Sekojošajā attēlā parādīts piemērs šādiem attēla datu kropļojumiem – attēls tika sūtīts no PC uz FPGA, tur filtrēts, un tad saņemts uz PC atpakaļ. Jāatzīmē, ka izvērse signāli, kas netiek pārvadīti caur RAM bitu aiztures līnijām, bet gan caur reģistriem, netiek kropļoti.

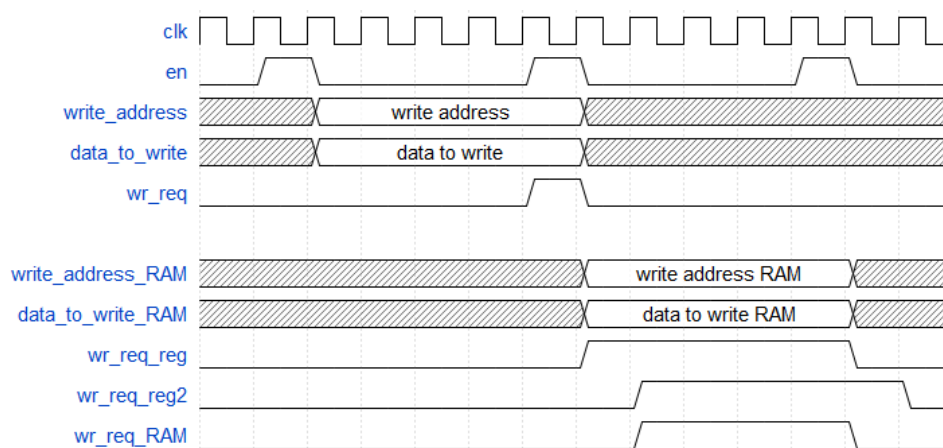


a) tāds, kādam vajadzēja būt

b) tāds, kāds sanāk, neveicot optimizācijas

Ar Gausa 2D Blur Filtru apstrādātais attēls ar tekstu

Lai cīnīties ar šo problēmu, ierakstīšanas signālu `wr_req` pāildzina, kā arī nostabilizē šī signāla laikā `write_address` un `data_to_write` vērtības:



Datu ierakstīšanas procedūras optimizācija RAM bitiem

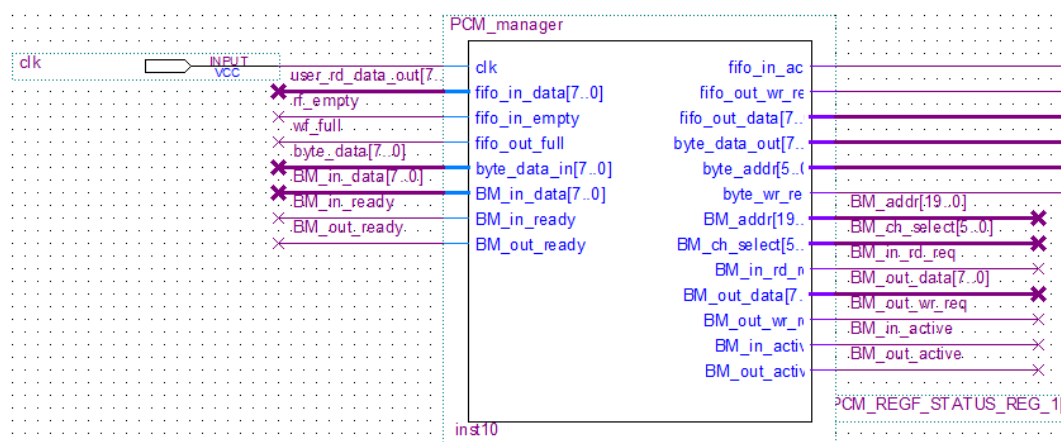
Tagad uz RAM bitiem nāk signāli `write_address_RAM`, `data_to_write_RAM` un `wr_req_RAM` (kas tiek formēts kā `wr_req_reg AND wr_req_reg2`), rezultātā ierakstīšanas laiks ir jau 32ns, nevis 8ns.

Kods, kas realizē minētas operācijas:

```
process(clk)
begin
  if rising_edge(clk) then
    if (en and wr_req = '1') then
      write_address_RAM    <= write_address;
      data_to_write_RAM    <= data_to_write;
    else
      write_address_RAM    <= write_address_RAM;
      data_to_write_RAM    <= data_to_write_RAM;
    end if;
    if (en = '1') then
      wr_req_reg           <= wr_req;
    else
      wr_req_reg           <= wr_req_reg;
    end if;
    wr_req_reg2            <= wr_req_reg;
  end if;
end process;
wr_req_RAM                <= wr_req_reg and wr_req_reg2;
```

PC komunikācijas stabilizācija

Veidojamajai sistēmai par komunikāciju ar PC atbild galvenokārt PC communication module (PCM) Manager mezgls:

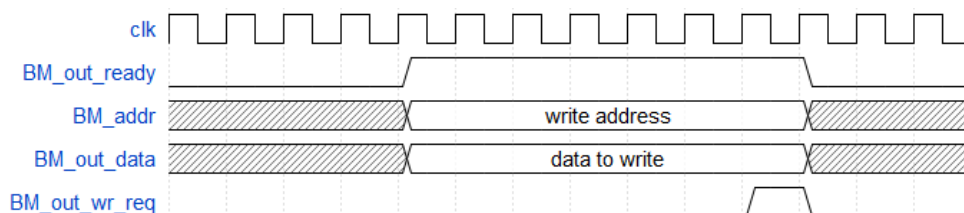


Komunikācijas moduļa signāli

Par bloku atmiņas operācijām atbild signāli ar „BM_”. Rakstīšanas kanālam („BM_out_”) tiek izmantoti šādi signāli:

4. BM_addr (uzdod rakstīšanas adresi, 20 biti),
5. BM_out_data (rakstāmie dati, no PC uz FPGA),
6. BM_out_ready (signalizē FPGA gatavību saņemt datus),
7. BM_out_wr_req (rakstīšanas pieprasījums no PC puses).

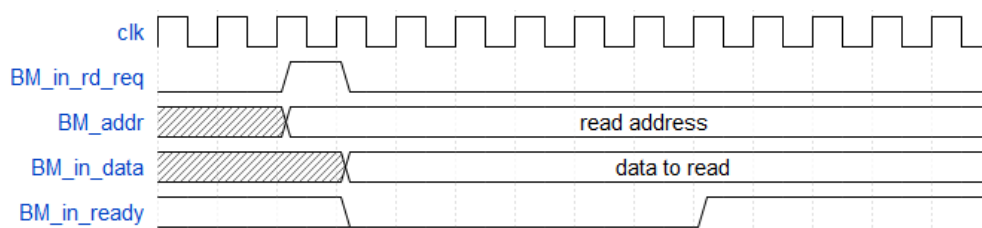
Rakstīšanas transakcija notiek pēc sekojoša protokola:



Rakstīšanas transakcija – viena baita nosūtīšanas secība

1. PCM_Manager gaida gatavības signālu no FPGA – BM_out_ready,
2. PCM_Manager saģenerē datus, ko nosūtīt un adresi,
3. PCM_Manager padod rakstīšanas pieprasījumu BM_out_wr_req un FPGA apstrādā nosūtīto baitu; kamēr notiek apstrāde gatavības signālam ir jābūt zemā līmenī.

Lasīšanas transakcija notiek pēc sekojoša protokola:

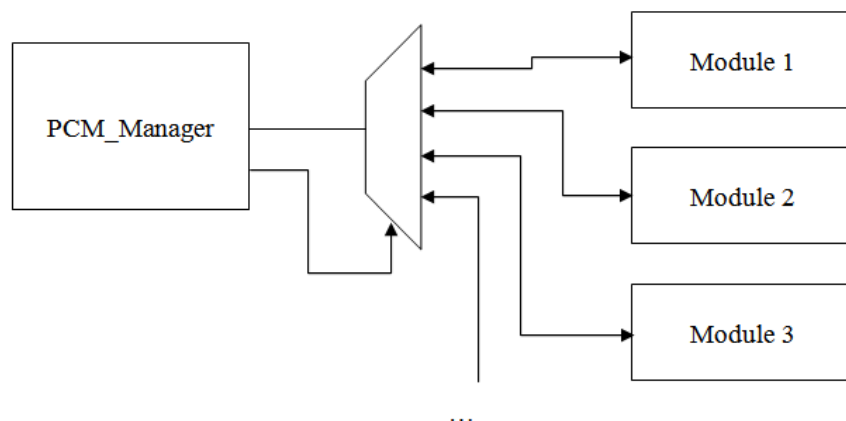


Lasīšanas transakcija – viena baita saņemšanas secība

- PCM_Manager padod lasīšanas adresi un nosūta lasīšanas pieprasījumu – BM_in_rd_req,

- FPGA sagatavo datus, ko nosūtīt un saģenerē gatavības signālu BM_in_ready,
- PCM_Manager apstrādā saņemtos datus.

Neskatoties uz to, ka PCM_Manager spēj strādāt līdz pat ar 64 divvirzienu atmiņas kanāliem, pašam modulim datu signāli katram atmiņas kanālam netiek dublēti – kā bija redzams attēlā, kas attēlo komunikācijas moduļa signālus. Vairāku kanālu datu pārraidei tiek izmantots BM_ch_select[5..0] kanāla izvelošais signāls, bet pārējie dati tiek multipleksēti.



Vairākanālu datu pārraides realizācija

Visas augstāk minētas transakcijas notiek ar maksimālo FPGA darba frekvenci 125MHz, līdz ar to datu un vadības signālu stabilizācija kļūst aktuāla. Tāpēc, lai stabilizētu šo mezglu darbību, attēlā parādītajās multipleksorā ieejās un izejās labajā jeb moduļu pusē, veic signālu buferēšanu.

Rezultāti

Veicot iepriekš minētas optimizācijas izdevās palielināt sistēmas darbības stabilitāti arī tad, ja shēmā tiek implementēts Quartus II atklūdošanas rīks Signal Tap Logic Analyzer. Turpmākie plāni ietver sevī arī citas sistēmas darbības un resursu patēriņu optimizācijas.

Secinājumi

Projekta „BiTe” pētnieciskā darbība „Lietišķo pētījumu” aktivitātē un eksperimentālā darbība „Eksperimentālās izstrādes” aktivitātē sekmīgi turpinās arī astotajā periodā. Noris izstrādājamās tehnoloģijas atsevišķu moduļu, algoritmu un elektrisko principiālo shēmu izpēte un izveide.

Sejas un plaukstu attēlu iegūšanas metožu izpētē turpmāk tiks veikti papildus pētījumi, lai novērtētu algoritmu stabilitāti un iespēju izmantot to kā interesējošā reģiona izdalīšanas algoritmu.

Sejas atpazīšanas sistēmas izveidei tiek plānota funkcionālo bloku nepieciešamības novērtēšana automatiskajā sejas atpazīšanas procesā. Plānots arī papildināt izveidoto programmu, novēršot radušās kļūdas.

Biometrisku datu iegūšanas algoritmu paralelizācijas un implementēšanas programmējamajos loģiskos masīvos aktivitātes ietvaros nākotnē paredzēts uzlabot esošo algoritmu darbību.

Biometrisku datu kriptēšanas, glabāšanas un lietojama apakšsistēmas izveides aktivitātē turpmāk tiek plānots veikt eksperimentus ar lielāku datubāzi un uzlabot algoritmu, kas paredzēts zīmīgāko punktu atrašanai.

Tehnoloģijas demonstratora eksperimentālā maketa montēšanas apakšaktivitātē maketa detalizēts izveides apraksts ir sniegts iepriekšējos un šajā pārskata periodā. Turpmāk veiktie iekārtas papildinājumi saistīti ar plates izstrādi un sistēmas integrēšanu modulī.

Projekta pētnieciskie rezultāti tiek apspriesti izpildītāju iknedēļas sanāksmēs. Šīs sanāksmes tiek veidotas, lai projekta izpildītāji kopīgi izdiskutētu par pētījumiem un problēmām, kas jārisina, lai aktivitātes veiksmīgi attīstītos.

Iepriekšējos periodos sagatavots zinātniskais raksts „Biohashing and Fusion of Palmprint and Palm Blood Vein Biometric Data”, kas prezentēts starptautiskā IEEE organizētā konferencē ICHB2011, Honkongā. Projekta izpildītāji piedalījās Starptautiskajā izgudrojumu izstādē „MINOX 2012”, kurā ieguva „Dienas Biznesa” speciālbalvu. Šajā pārskata periodā tika sagatavots zinātniskais raksts „Palmprint Image Processing With Non-Halo Complex Matched Filters For Forensic Data Analysis”, kas tika prezentēts International Workshop on Biometrics and Forensics (IWBF) konferencē Lisabonā, Portugālē.

Turpinās darbs pie aktivitātes „Intelektuālā īpašuma tiesību aizsargāšana”, iesniegts starptautiska patenta pieteikums Nr.PCT/LV2012/000005 „Biometric authentication apparatus and biometric authentication method”, ar kuru ir aizsargātas projekta pētījumu oriģinālo rezultātu intelektuālās īpašuma tiesības. Saņemts PCT (Patent Cooperation Treaty) pozitīvs atzinums un uzsākts tā tulkošanas process.

Ir noteikti nākamā perioda pētniecisko darbu uzdevumi un sasniedzamie rezultāti. Par projekta pētnieciskiem sasniegumiem tiek plānots informēt arī piedaloties ar referātiem un publikācijās konferencēs un semināros.