VHDL VGA ОБРАБОТКА И ВИЗУАЛИЗАЦИЯ НА ДАННИ ОТ АЦП

Росен Спиров

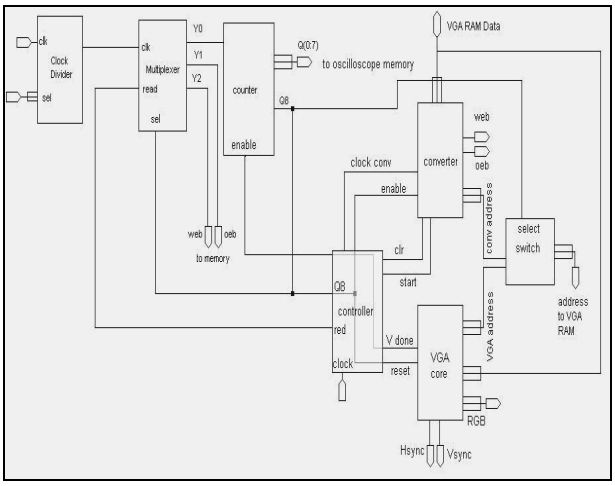
Abstract: The article presents VGA converter module for visualization and digital processing of analog signals through VHDL for FPGA .

Key words: VGA, Image processing, VHDL, FPGA

I. ВЪВЕДЕНИЕ

Статията представя VGA визуализиращ модул посредством VHDL [1] на цифрова осцилоскопна система за въвеждане и обработка на аналогови сигнали, реализирана на базата на Altera DE2 с FPGA CYCLONE II и развойна среда QUARTUS II [5].

Описанието на системния блок има вида от фиг.1.

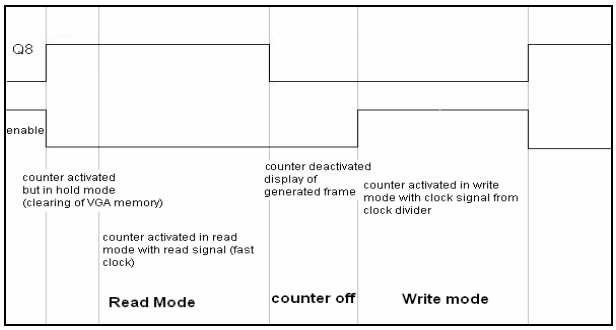


Фиг.1. Системен блок

Аналоговият сигнал се подава на аналого-цифрово преобразуване и така сигнала превръща своите амплитуди в поредица от битовете [2], които се съхраняват в осцилоскопна RAM памет. Адресите на тази осцилоскопна RAM се генерират от брояч, който осигурява режим „запис" и е основен за ADC. След отброяване на 256 адреса, осцилоскопната памет се запълва и може да се премине към режим „четете-не". Мултиплексорът превключва брояча към бързо тактуване (12 MHz). Следва преобразуване на амплитудите за дадения кадър и осъществяване на пикселна генерация на два етапа, като първо се изчиства VGA паметта и последващо построяване на маркиращи оси. Втория етап е четене на данните от осцилоскопната памет и запълване на VGA паметта. Следва процес на прочитане на VGA RAM до 256 –та клетка и идва ред VGA core управлението да визуализира генерирания кадър. След това целият цикъл се повтаря отново.

II. АНАЛИЗ И РЕЖИМИ НА РАБОТА НА СИСТЕМАТА

Работата на системата се характеризира с циклите, представени на фиг.2.



Фиг. 2. Времедиаграми на циклите „четене" и „запис"

При цикъла „ЗАПИС" изходите на брояча първо се нулират ("000000000") тъй като Q8 = 0, а входа на брояча е свързан с изход Y0 на мултиплексора, като мултиплексора определя коя серия от контролирани сигнали да премине към изходния порт Y. Y0 ще бъде тактов сигнал, който постъпва от тактовия

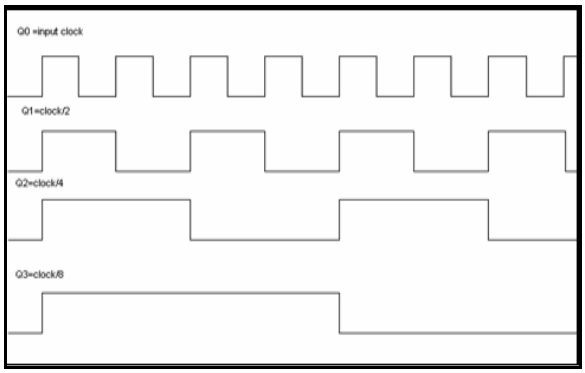
генератор „Clock Divider”. Тактовия генератор изпраща този тактов сигнал към ADC, съхранява се в брояча “counter” и синхронизира ADC. Y1 = "1" изключва извода „oeb” /enable-разрешение/ на осцилоскопната памет. Y2 = clock –„web” тактуване, което разрешава синхронен „запис” на изходните данни от ADC в паметта. Този цикъл ще се повтаря докато нивото на Q8 е високо, което означава , че всичките 256 места в паметта на осцилоскопа ще се запълнят. Това се променя като на вход “read” на мултиплексора постъпи "1". Този сигнал дава края на цикъла „запис”и началото на цикъла „четене”.

При цикъл „ЧЕТЕНЕ” , Q8 = "1" започва контролирано четене към изходния порт на мултиплексора. Y0 ще бъде еквивалентен на сигнал „четене’, който се генерира от следващия модул на следващата фаза, при която се четат съхраняваните данни (генериращ конвертор на кадри). Когато се активира Y1 = "0" се разрешава работата на осцилоскопа. Когато Y2 = "1" се отменя забраната за писане в паметта. Получения адрес за осцилоскопната памет от брояча се контролира от сигнала „четене”. Конверторния модул може да съхрани съдържанието на брояча в „hold” режим, който да поддържа сигнала „четене” постоянен, докато се изпълнява изчистването на VGA паметта, а след това се „чете" цялата осцилоскопна памет. Тези режими нагледно са представени в табл.1

Табл.1. Таблица на контролните сигнали по време на различните режими на работа

select	0	1
Y0 count	clk	Read
Y1 output	1	‘0’
Y2 write	clk	‘1’

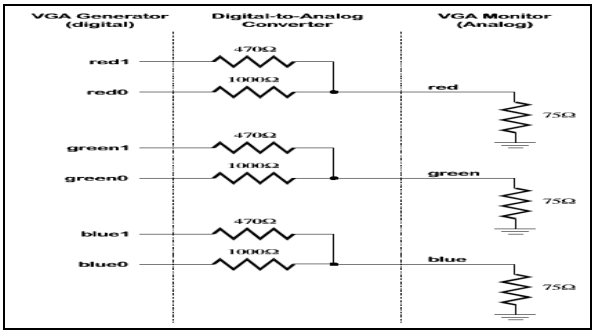
Деактивиране на брояча /стоп на броенето/. В края на цикъла „четене” Q8 ще "0", това ще стартира цикъл на „запис”, но не и докато останалата част на системата не е готова за работа. Сега постъпва разрешение за въвеждане за брояча. Този сигнал определя кога брояча е активен и кога не е. Броячът е активен и в двата цикъла на четене и запис, т.е. когато Q8 = "1" (цикъл”четене”) и по време на цикъл „запис” само ако останалата част на системата е готова да работи. Затова условието за активност е когато Q8 = "1" или активиране- enable=’1’ (т.е. системата е готова да продължи с друг цикъл на „запис”). Следователно това активира брояча по време на циклите „четене” и „запис”, всеки със собственото си тактуване, както и изключване когато паметта не се използва. Това означава, че броячът се спира само когато системата работи за визуализация на генериран кадър. Използването на ADC е само по време на „запис” (Q8 = "0"). Така Q8 се използва като сигнал за избор на чип за ADC. Q8 се използва и за “reset” на системата за контрол „controller”, а честотата на извадките се регулира с деление на основната честота от изходите Q0,Q1,Q2 и Q3 на брояча, както е показано на фиг.3.



Фиг.3. Изходни сигнали на управляващия генератор

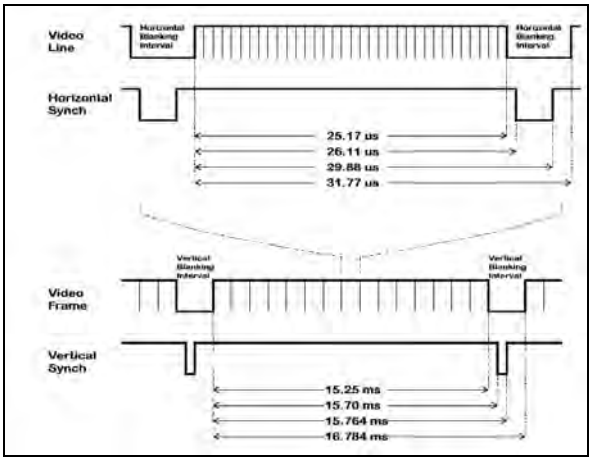
VGA интерфейс. VGA цветовите сигнали са три - червено, зелено и синьо - цветове, които изпращат информация към VGA монитора. Аналоговите нива са между 0 (напълно тъмно) и 0,7 V (максимална яркост) като тези контролни линии управляват монитора с различна наситеност за всеки от тези три основни цвята зада се съчетаят в определен цвят за даден пиксел върху екрана [3]. Всеки аналогов цветови вход може да се настрои към едно от четири нива с два цифрови изхода с помощта на обикновен двубитов цифрово-аналогов преобразувател фиг.4. Четирите възможни нива за всеки аналогов вход се комбинират в монитора, за да се създаде пиксел с един цвят от 4×4×4= 64 различни цвята. Така че шестте цифрови

изходни линии позволяват използване на палитра от 64 цвята.



Фиг.4. VGA изходни сигнали

Кадър с VGA видео формат обикновено има 640 реда с по 480 пиксела. За развивката се изисква два синхронизиращи сигнала за пускане и спиране на всяка от отклонителните Hs и Vs вериги в точния момент. Времедиаграмите на VGA сигналите за синхронизация са показани на фиг.5.



Фиг.5. VGA синхронизиращи сигнали

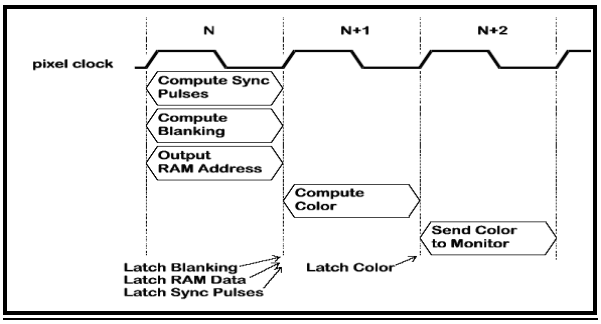
Отрицателните синхро-импулси за хоризонтална синхронизация маркират началото и края на линията и мониторът показва пиксела между левия и десния ръб в даден ред от видимата площ на екрана. Действителните пиксели се изпращат към монитора за визуализиране в рамките на 25.17 µs. Хоризонталният синхронизиращ импулс се подава след последния пиксел за 0.94 µs и остава в ниско ниво за 3.77 µs. Новата линия от пиксели може да започне да се изписва най-малко 1,89 µs след края на хоризонталния синхро- импулс. Така че един ред заема 25,17µs през интервал 31.77 µs. Другите 6.6 µs от всяка хоризонтална линия са за хоризонталния гасящ импулс, по време на който екрана е тъмен. По аналогичен начин отрицателните синхро-импулси за вертикална синхронизация маркират началото и края на кадъра, съставени от видео линии и се гарантира, че мониторът показва линиите между горната и долната част в рамката на видимия екран на монитора. Линиите се изпращат за наблюдение в рамките на един 15.25 ms прозорец. Вертикалният синхронизиращ сигнал изчаква най-малко 0,45 ms след последния ред и остава в ниско ниво за 64µs. Първия ред на следващия кадър може да започне най-малко 1,02 ms след края на вертикалния синхронизиращ импулс. Така всеки кадър заема 15,25 ms през 16,784 ms интервал. Останалите 1,534 ms от продължителността на кадъра е вертикалния импулс за нулиране, през който екрана е тъмен.

VGA алгоритъм за генериране на сигнали .

Псевдо кода има два външни цикли: един който показва L линии с видими пиксели, и друг който вмъква V празни редове и вертикален синхроимпулс. В рамките на първия цикъл имаме още два цикъла: един който изпраща P пиксела на всеки видео ред към монитора, а друг, които вмъква H празни пиксели и хоризонтален синхроимпулс. За края на визуализацията на всеки пиксел се сигнализира за получаване на следващия байт от паметта. Всеки байт съдържа четири дву-битови клетки. Малък цикъл итеративно оформя всеки пиксел да се изобрази от младшите два бита от полу-байта. Тогава байта се измества с два бита, така че следващия пиксел ще бъде в правилната позиция по време на следващата итерация на цикъла. Освен това в два бита за всеки пиксел се съхранява един от четирите цветове. Маркираната от дву-битовия пиксел стойност изисква от мониторната електроника да се извърши COLOR_MAP преобразуване.

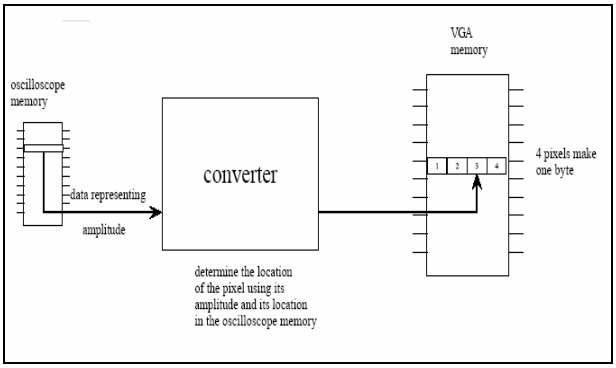
Генериране на VGA сигнали

Фигура 6 показва времедиаграма на формиране на цикли с отчитане времето за достъп до данните в RAM.



Фиг.6. Генериране на псевдо сигнален VGA код

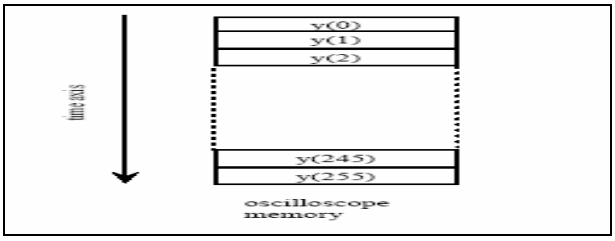
VGA конвертора представлява връзката между АЦП и VGA модула.



Фиг.7. Конверторен процес

III. ОСНОВНА ИДЕЯ И РЕАЛИЗАЦИЯ

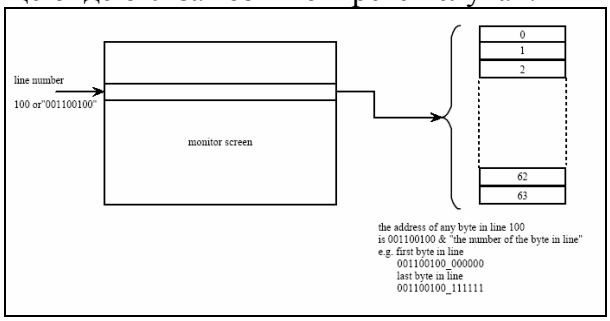
Данните, съхранявани в един байт на осцилоскопната памет се четат както е показано на фиг.8 и понататък се представят с две координати съответно вертикална ос за амплитудата, а хоризонтална е времевата ос.



Фиг.8. Четене на АЦП паметта

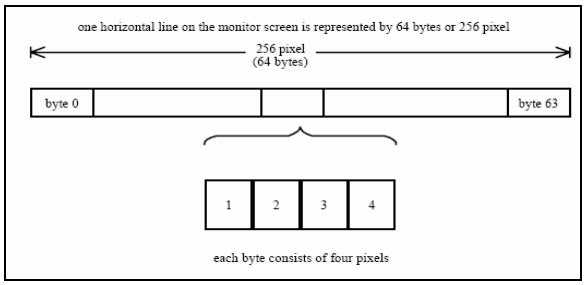
Имаме две основни операции, които се контролират от един сигнал за стартиране.

1. *Изчистване на VGA паметта*, което е много важно за да се изчисти стария кадър и да се подготви паметта за новия. Това е просто един нормален брояч свързан с адресната магистрала на VGA паметта, както е показано на фиг.9 и изпраща "11111111" за всеки байт в паметта последователно. Данните, изпратени при този процес представляват цвета на фона на екрана, които ще бъде бял за този конкретен случай.



Фиг.9. Адресиране

Изписването едновременно на вертикалните и хоризонталните линии на осите е другата функция на този процес, като те се означават с червени пиксели и се задават в паметта. Местоположението на тези пиксели може да се определи с адреса от брояча. Когато този процес завърши, един сигнал (CLR) се изпраща на контролера и се превключва другия режим.



Фиг. 10. Преобразуване

2. *Преобразуването е процес*, представен на фиг.10, при който се преобразуват амплитудите, съхранявани в осцилоскопната памет в пиксели за VGA паметта. Това е основното предназначение на конвертора, този процес се извършва за 256 тактови цикъла тъй като имаме работа само с 256 клетки в осцилоскопната памет. Всеки тактов цикъл е разделен на две части - в първата става превключване от появяването на тактов нарастващ фронт, като другият цикъл започва с превключване от падащия фронт. В първата част на тактовия цикъл данните се извикват от осцилоскопната памет, генерира се адреса на байта във VGA паметта и имаме извеждане и

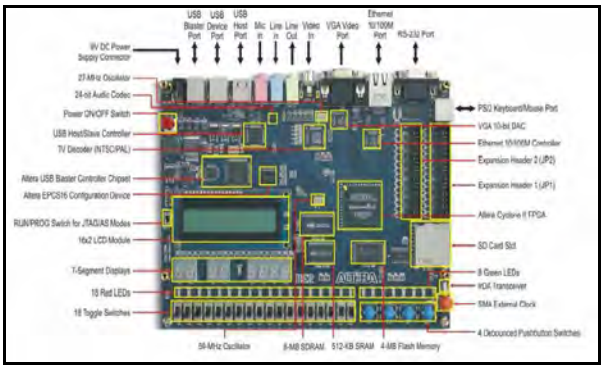
се четене. През втората половина на цикъла се фиксира местоположението на пиксела с байта, намиращ се във VGA паметта, който се определя, като се маскира същия с байт в съответствие с пикселното местоположение. След това байта се изпраща до VGA паметта и таймера нараства със една стъпка. Всеки цикъл се разделя на две, за да се даде на VGA паметта достатъчно време между двата цикъла да се чете и пише. Най-трудната част в този процес е откриване разположението на пикселите. За да се определи то от байта за пиксела, използваме само двата старши значещи бита, адресирвани от брояча.

Контрол Първо Q8 се инициализира с висок потенциал, конвертора е включен, а VGA core е деактивиран (нулиран). Това се прави чрез изпращане enable='1' към контролера. Контролерът след това получава сигнал „clr" или signal='0' от конвертора, като по този начин се изпраща start='1' на конвертора и read='0' към мултиплексора, стробиращ рутиращия брояч. Това установява брояча в състяние „hold". Когато конвертора изпрати clr ='1' контролера изпраща start='0', и read= clk към брояча мултиплексора. Тогава броячния изход застава в състояние „hold". Така когато конвертора завърши своята работа, Q8 става ниско ниво. Това дава възможност на VGA, да започне визуализация върху дисплея на запаметените данни във VGA паметта и конвертора да е деактивиран. В същото време брояча е забранен като V_done = "0", защото VGA е била рестартирана по-рано, когато Q8 е бил с високо ниво, този сигнал е изпратен от VGA core /VGA контролера/ към контролера за стартиране рутирането на брояча с неговия 'enable' сигнал. Когато 200 кадъра са визуализирани от VGA, сигнала „V_done" става с високо ниво, с което отново да започва нов цикъл на броене за „запис". В това време VGA е все още активна и продължава да визуализира на дисплея предварително запаметените данни от паметта, докато Q8 премине отново във високо ниво и така цикъла се повтаря.

Цялостно процеса на визуализиране има вида:

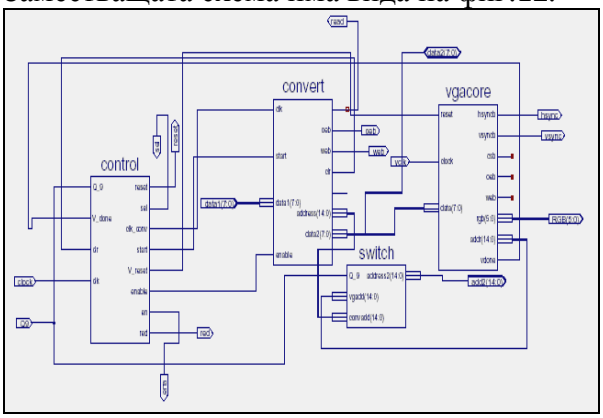
```
/* Изпрати L линии от видеото към монитора
* /
  for line_cnt=1 to L
  /* Изпрати P пиксела за всяка линия */
  for pixel_cnt=1 to P
  /* вземи данни за пиксела от RAM */
  data = RAM (address)
  address = address + 1
  /* RAM даннов байт съдържа 4 пиксела */
  for d=1 to 4
  /* маскирай off пикселите в долните два бита
  */
  pixel = data & 00000011
  /* Следваща смяна на пикселите в долните
  два бита */
  data = data>>2
  /* вземи цвят за двата бита пиксели */
  color = COLOR_MAP (pixel)
  send color to monitor
  d = d + 1
  /* Увеличаване с четири броя на пикселите */
  pixel_cnt = pixel_cnt + 4
  /* Черен монитор за H пиксели */
  for horiz_blank_cnt=1 to H
  color = BLANK
  send color to monitor
  /* Импулс за хоризонтална синхронизация в
  точния момент */
  if horiz_blank_cnt>HB0 and
  horiz_blank_cnt<HB1
  hsync = 0
  else
  hsync = 1
  horiz_blank_cnt = horiz_blank_cnt + 1
  line_cnt = line_cnt + 1
  /* Черен монитор за V линии и въвеждане на
  вертикална синхронизация */
  for vert_blank_cnt=1 to V
  color = BLANK
  send color to monitor
  /* Импулс за вертикална синхронизация в
  точния момент*/
  if vert_blank_cnt>VB0 and vert_blank_cnt<VB1
  vsync = 0
  else
  vsync = 1
  vert_blank_cnt = vert_blank_cnt + 1
  /* Връщане в началото на картината в RAM
  */
address = 0
```

За екпериментална основа служи развойната среда Quartus II на хардуерната FPGA система Altera DE2 , показана на фиг. 11.



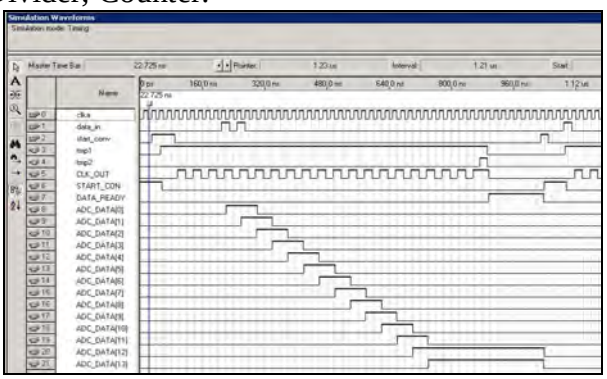
Фиг.11. Altera DE2 board

Заместващата схема има вида на фиг.12.

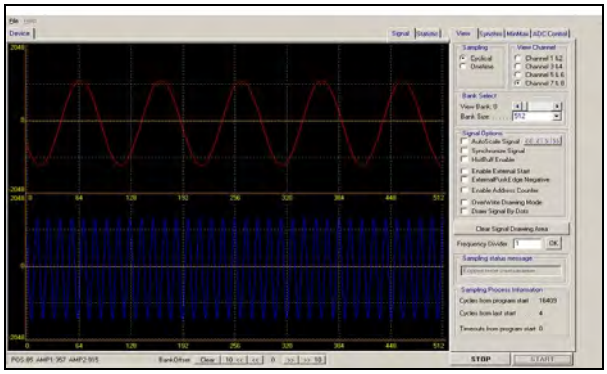


Фиг.12. Заместващата схема

Заместващата схема на VGA визуализиращата система има вида от фиг.12, като всеки блок притежава своето VHDL описание и връзки [4]. Визуално сигналите от ADC имат вида от фиг.13, а на фиг. 14 се вижда визуализиране на аналогови сигнали 50Hz и 1кHz с амплитуда 5V във визуализиращия прозорец на QuartusII, с последвапа имплементация по JTAG към Altera FPGA CycloneII [5]. Разработката цялостно е обезпечена с разработен пакет програми на VHDL за всеки интегриран отделен блок: VGA Driver, PIXEL Generator, VGA Controller, Converter, Switch, Divider, Counter.



Фиг.13. Експериментални данни за АЦП



Фиг.14. VGA визуализация

IV. ИЗВОДИ

Бързодействие и точност на работа на реализираната система, гъвкавост при синтеза към корекции. Визуализиращия модул предоставя възможност за цифрови обработки на входните аналогови сигнали , филтрации, корекции и други вторични обработки на основа VHDL за целите на разпознаване на аудио и видео обекти, а това ги прави изключително удобни за приложения за наблюдение, контрол, идентификация и др. Разработката потвърждава актуалността и приложимостта на VHDL разработките от този тип като перспективни и надеждни за реализация, пренастройване и изграждане на уникални системи.

Литература:

[1]. Иванов Н. И. Алгебра на препрограмируемите прибори -2част „Атика”,София,2008 ISBN 978-954-729-253-6

[2]. Путятин Е.П., Аверин С.И. Обработка изображений в робототехнике. — М: Машиностроение, 1990. — 320 с.

[3]. Yu Z., Bajaj Ch. Image Segmentation Using Gradient Vector Diffusion and Region Merging. 16th International Conference on Pattern Recognition (ICPR'02). — 2002. Vol 2. — P. 20941

[4]. Wolf W. FPGA- Based System Design, Upper Saddle River, NJ07458 ISBN 0-13-142461-0 http://www.phptr.com

[5]. http://www.altera.com

За контакти:
инж. Росен Спиров, докторант в Катедра”Електротехника и Микроелектроника” при ФЕ на ТУ-Варна , ул. Студентска № 1, 502Е
e-mail: rosexel@abv.bg