|  |  |
| --- | --- |
| ЗМІСТ |  |
| ВСТУП………………………………………………………………………. | 6 |
| ПЕРШИЙ РОЗДІЛ. АНАЛІЗ ПИТАНЬ ТА ПОСТАНОВКА ЗАДАЧІ…. | 10 |
| 1.1 Основі обробки відео та зображень ……………………………….. | 10 |
| 1.2 Цифрові технології обробки сигналів ………...…………………... | 14 |
| 1.2.1 Прилад із зарядовим зв'язком (ПЗЗ / CCD)………………..… | 17 |
| 1.2.2 Технології CCD……………………………………………..… | 23 |
| 1.2.3 Комплементарна структура метал-оксид- |  |
| напівпровідника (КМОП / CMOS) …………………………………… | 30 |
| 1.3 Основи системи DSP …...…………………….………..…………… | 34 |
| 1.4 Чому ПЛІС? Впровадження та реалізація.………………………… | 36 |
| 1.5 Функціональність і продуктивність системи.………………..….... | 38 |
| 1.6 Системи реального часу.….…………………………………..…….. | 39 |
| 1.7 Послідовна і паралельна обробка зображень…...……………...…. | 41 |
| 1.8 Архітектура FPGA ……………………………………………….…. | 43 |
| 1.9 Методи зіставлення ПЛІС.……………...………………………….. | 46 |
| 1.10 Проблеми FPGA ……………………………...…………………… | 50 |
| 1.11 Висновки за розділом……………………………………………... | 52 |
| ДРУГИЙ РОЗДІЛ. ТЕОРЕТИЧНИЙ АНАЛІЗ ФІЛЬТРАЦІЇ  ЗОБРАЖЕНЬ…………………………………………………………………. | 55 |
| 2.1 Загальні положення……….……….……………………………… | 55 |
| 2.2 Методи цифрової обробки сигналів….…………………….…...... | 56 |
| 2.3 FIR фільтри …………………...…………………………..……….. | 61 |
| 2.4 Оптимізація FIR фільтрів………………...………………………... | 63 |
| 2.5 Каскадний інтегрально-гребінчастий фільтр (CIC) ……….…….. | 66 |
| 2.5.1 Багатоканальність фільтрів ……………………..…………... | 68 |
| 2.5.2 Аналіз ефективності СІС-фільтру..………………………….. | 71 |
| 2.6 Медіанний фільтр …………………………………………..…….. | 73 |
| 2.6.1 Медіанний алгоритм фільтрації ……………………………. | 74 |

|  |  |
| --- | --- |
| 2.6.2 Реалізація медіанного фільтра ……………………………... | 78 |
| 2.7 Висновки за розділом…………………………………………….. | 80 |
| ТРЕТІЙ РОЗДІЛ. РОЗРОБКА, МОДЕЛЮВАННЯ ТА ПОБУДОВА  СИСТЕМИ ОБРОБКИ ЗОБРАЖЕНЬ………………………………………. | 83 |
| 3.1 Реалізація систем обробки зображень із застосуванням фільтрів | 83 |
| 3.2 Розрахунок відносини сигнал/шум (ВСШ)………………….……. | 89 |
| 3.3 Реалізація медіанної фільтрації за допомогою MATLAB……….. | 92 |
| 3.4 Обробка тестового зображення MATLAB………………………... | 95 |
| 3.5 Висновки за розділом………………………………………………. | 98 |
| ВИСНОВКИ……………………………………………………………….... | 100 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ………………………………….. | 110 |

# ВСТУП

Актуальність теми. У наш час наука і техніка постійно розвиваються. Однією з самих важливих областей людської діяльності є електроніка і, в особливості, цифрові технології. Без них сучасне суспільство обійтися не може.

На даний час, багато уваги приділяється не тільки створенню нових пристроїв, поліпшенню їх можливостей і характеристик. На передній план зараз виходять такі важливі показники як вага, компактність, ціна, економічність в використанні тощо. І це добре видно, якщо подивитися, як змінили свої розміри і якісні показники телевізори, музичні програвачі, комп'ютери, мобільні телефони, і багато іншої побутової техніки. Зменшення розмірів різних електронних пристроїв відбувається завдяки тому, що з кожним роком елементи електроніки стають менше за розміром, і при цьому їх виконувані завдання збільшуються.

У наш час широкого розповсюдження набули методи аналізу швидкоплинних процесів за допомогою високошвидкісної обробки відеозображень.

У всіх галузях обробка зображень в режимі реального часу може істотно змінити нашу взаємодію з технологіями.

Все більше і більше електроніка на технологічних платформах починає покладатися на високошвидкісну обробку для захоплення і інтерпретації зображення, що дозволяє системі негайно реагувати відповідною дією в будь-яких сферах життя.

Технологія приладів із зарядним зв'язком (ПЗЗ) була домінуючим інструментом для електронних датчиків зображення протягом декількох десятиліть завдяки їх високій світлочутливості, низькому фіксованому шаблонного шуму, невеликого пікселя і великому розміру масиву.

Однак в останнє десятиліття датчики зображення привернули увагу багатьох дослідників завдяки їх низькому розсіюванню енергії, низької вартості, можливостям обробки схем.

Проте, необроблені вихідні зображення, отримані за допомогою датчиків, мають низьку якість відеозображення і вимагають подальшої обробки, в основному через шум, розмитість і погану контрастність. Щоб вирішити ці проблеми, схеми обробки зображення зазвичай пов'язують з датчиками зображення як частину всієї системи зору. Зазвичай дві області співіснують всередині однієї мікросхеми для вимірювання і попередньої обробки, які реалізовані в одній інтегральної мікросхемі.

Щоб протистояти високому потоку даних, викликаному алгоритмами комп'ютерного зору, альтернативний підхід полягає у виконанні певної обробки зображень в фокальній площині датчика. Інтеграція масиву пікселів і схем обробки зображень на одному монолітному кристалі робить систему більш компактною і дозволяє поліпшити поведінку і відгук датчика. Отже, для досягнення деяких простих завдань низькорівневої обробки зображень, інтелектуальний датчик об'єднує аналогові або цифрові схеми обробки.

Системи можуть бути побудовані з використанням контролера системи на кристалі (SoC), на архітектурі програмованої логічної інтегральної схеми (FPGA / ПЛІС). В цілому, системи, засновані на SoC, реалізують алгоритми бачення як програмне забезпечення і не мають можливості масштабування, коли мова йде про високопродуктивні прикладні програми з малою затримкою часу. Апаратні системи обробки зображень дуже швидкі і добре працюють після програмування. Однак найбільшою проблемою з апаратними системами є їх відносна негнучкість. Розробникам стає складно перепрограмувати систему в разі будь-яких змін в додатку.

FPGA об'єднують паралельну природу апаратного забезпечення і гнучкість програмного забезпечення так, що їх функціональність може бути перепрограмовано при необхідності. У перші часи FPGA мали дуже невелику обчислювальну потужність і для реалізації системи обробки

зображень потрібно більше одного процесора. З роками збільшувалася обчислювальна потужність ПЛІС і були розроблені гібридні системи, в яких вони використовувалися в якості прискорювачів зору. Сучасні системи мають можливість і ресурси для реалізації всієї програми на одній ПЛІС. Це робить їх гарним вибором для вбудованих систем відеоспостереження в реальному часі, особливо в високошвидкісних додатках.

Важливою перевагою побудови System-On-Chip для високошвидкісної обробки зображень - можливість об'єднання на одному кристалі аналогової, цифрової та обробної частин, що послужило основою для мініатюризації систем обробки сигналів і зниження їх вартості з огляду на відмову від додаткових процесорних мікросхем. Таким чином, завдання, які поставлені і вирішуються в роботі є актуальними.

# Мета і завдання роботи.

Розробка ефективних способів використання концепції System-On-Chip для високошвидкісної системи обробки зображень на базі FPGA - інтегрованих цифрових фільтрів. Для досягнення мети поставлені наступні завдання:

* дослідження методів та інструментальних засобів побудови систем високошвидкісної обробки відеозображень;
* дослідження шляхів підвищення ефективності функціонування систем високошвидкісної обробки інформації;
* визначення раціональної апаратно-програмної платформи для реалізації цифрових підсистем фільтрації в системах високошвидкісної обробки відео зображень;
* розробка практичних методів синтезу цифрових систем фільтрації для високошвидкісної обробки зображень з використанням концепції System-On- Chip.

**Об'єктом дослідження** є методи і засоби синтезу систем цифрової фільтрації.

**Предметом дослідження** є система паралельної обробки інформації.

**Наукова новизна** отриманих результатів полягає в наступному:

* пропонується новий метод реалізації паралельної обробки цифрових даних, заснований на використанні концепції System-On-Chip для високошвидкісної обробки зображень на базі програмованої логічної інтегральної схеми;
* пропонується новий спосіб попередньої обробки зображення і регулювання експозиції кадру за допомогою каскадного інтегрально- гребінчастого фільтра Хогенауера;
* пропонується новий спосіб практичної реалізації методу паралельної обробки цифрових даних за допомогою систем автоматизованого проектування MATLAB & SIMULINK і Xilinx ISE 13.3.

**Практичне значення** отриманих результатів полягає в тому, що запропоновані способи дозволяють:

* спростити процес синтезу систем цифрової фільтрації для високошвидкісної обробки відеозображень;
* підвищити ефективність роботи схем на базі FPGA за рахунок паралельної обробки цифрових даних;
* розширити область використання концепції System-On-Chip за рахунок практичної реалізації методу за допомогою систем автоматизованого проектування MATLAB & SIMULINK і Xilinx ISE 13.3.

Особистий внесок здобувача полягає в розробці практичних методів і способів побудови цифрових систем фільтрації для високошвидкісної обробки відеозображень.

# ПЕРШИЙ РОЗДІЛ

**АНАЛІЗ ПИТАННЯ ТА ПОСТАНОВКА ЗАВДАНЬ**

* 1. **Основи обробки зображень**

Електроніка зробила революцію в 20 столітті і продовжує впливати в

21 столітті. Народження і подальше зростання комп'ютерної індустрії, створення мобільної телефонії і загальна оцифровка телевізійних і радіослужб в значній мірі були відповідальні за недавнє зростання. У 1970-х і 1980-х роках електронні системи створювалися шляхом об'єднання стандартних компонентів, таких як мікропроцесори і мікросхеми пам'яті з цифровими логічними компонентами, наприклад, виділені інтегральні схеми (ІС) разом з виділеними компонентами введення/виведення (I/O) на друкованих платах (PCB). По мірі зростання рівнів інтеграції, виробництво працюючих друкованих плат ставало все більш складним, в основному через збільшення складності компонентів з точки зору збільшення числа транзисторів і виводів введення/виведення, а також розробки багатошарових плат з максимально можливою кількістю шарів. Таким чином, ймовірність неправильного з'єднання компонентів також зросла, особливо в зв'язку з неможливістю проектування і тестування діючої системи до того, як налагоджено було виробництво. Проблема ставала дедалі гострішою через складність еволюції описів системи в процесі розробки плат. Потреба в розробці систем, відповідних стандартам, які розвиваються, або які можуть змінитися після розроблення плати через зміни в системі або змін до специфікації проекту, означало, що при застосуванні концепції «повністю визначеної» технології, з точки зору побудови фізичної системи і розробки на процесорі, програмний код ставав все менш імовірним. Хоча використання програмованих процесорів, таких як мікроконтролери та мікропроцесори, давало конструкторові деяку свободу вносити зміни, щоб

виправити або модифікувати систему після виробництва. Даний факт був обмежений, оскільки зміни у взаємозв'язку компонентів на друкованій платі обмежувалися тільки можливістю підключення вводу/виводу самих процесорів. Таким чином, привабливість використання взаємозв'язку, програмованість відкривала значний потенціал. Наслідком цього стала підтримка концепції технології програмованої логічної інтегральної схеми (FPGA).

Обробка відео / зображення - це будь-яка форма обробки сигналу, для якої введенням є відеозображення, наприклад відеопотік або фотографія. Зображення повинно бути оброблено для кращого відезображення, зберігання та інших спеціальних цілей. Наприклад, вчені-медики покращують рентгенівські знімки шляхом пригнічення шумів, що супроводжують зображення, для лікарів, щоб поставити точний діагноз. Обробка зображень також створює міцну основу для комп'ютерного зору, стиснення відеозображень, машинного навчання і т. д.

Цифрове зображення може бути визначено як двовимірна функція f (*x; y*), де *x* і *y* є просторовими координатами, а амплітуда *f* (*x; y*) в будь-якому місці зображення називається інтенсивністю зображення в цій точці.

Кольорове зображення включає в себе інформацію про колір для кожного пікселя, тобто це можна розглядати як комбінацію окремих зображень різних колірних каналів. В основному використовується колірна система в комп'ютерних дисплеях - це простір RGB (червоний, зелений, синій).

Зображення в градаціях сірого відноситься до монохромного зображення. Єдиний колір кожного пікселя - це відтінок сірого. Фактично, сірий колір - це той, в якому червоний, зелений і синій компоненти мають однакову інтенсивність в просторі RGB. Отже, необхідно вказувати тільки одне значення інтенсивності для кожного пікселя, на відміну від уявлення кожного пікселя з трьома интенсивностями в повнокольорових зображеннях.

Для кожного колірного каналу зображення RGB і пікселя зображення в градаціях сірого інтенсивність знаходиться в заданому діапазоні між мінімальним і максимальним значеннями. Часто інтенсивність кожного пікселя зберігається з використанням 8- розрядного цілого числа, що дає 256 можливих різних ступенів від 0 до 255. Чорний колір дорівнює 0, а білий - 255 відповідно.

Двійкове зображення, або чорно-біле зображення, є свого роду цифровим зображенням, яке має тільки два можливих значення інтенсивності для кожного пікселя: 1 - для білого і 0 - для чорного. Об'єкт позначений кольором переднього плану, а інша частина зображення - кольором фону.

Відеопотік являє собою серію послідовних зображень. Кожен видеокадр складається з активних пікселів і порожнього зображення. В кінці кожного рядка є частина форми хвилі, яка називається горизонтальним інтервалом гасіння. Сигнал горизонтальної синхронізації «hsync» вказує початок наступного рядка. Починаючи зверху, всі активні лінії на область відображення скануються таким чином що як тільки система перевіряє всі активний видеокадр, з'являється інша частина форми сигналу, звана інтервалом вертикального гасіння. Сигнал вертикальної синхронізації

«vsync» вказує початок нового видеокадра.

Традиційні алгоритми обробки зображень мають послідовний характер. Коли ці алгоритми будуть реалізовані в системі реального часу, час відгуку буде високим. У вбудованій платформі такі алгоритми споживають більше енергії через більшу кількість тактів, необхідних для виконання алгоритму.З появою програмованих логічних інтегральних схем (FPGA / ПЛІС) можуть бути розроблені масивно паралельні архітектури для прискорення виконання декількох алгоритмів обробки зображень. Мови високого рівня, такі як Matlab, C ++, Open-CV і т. д., є загальними платформами для розробки і перевірки таких алгоритмів. Ці платформи найбільш підходять для прикладних програм, які не мають будь-яких часових обмежень. Іншими словами, де час відгуку не так важливий. У системах

реального часу платформи високого рівня не будуть хорошим вибором через брак часу і ресурсів.

З появою мікроконтролерів стало можливим проектувати вбудовані системи обробки зображень, які були б портативними, споживали менше енергії і часу. У процесорах з мікроконтролером виконання алгоритму є послідовним за своєю природою. Швидкість виконання значно підвищена в сучасних процесорах, в яких використовуються конвеєрні і суперскалярні архітектури. Вдосконалені процесори включають паралелізм на рівні команд, але загальне виконання алгоритму буде послідовним за своєю природою. Таким чином, система на основі мікроконтролера не може ефективно використовувати властивий їй паралелізм, який використовується в більшості алгоритмів обробки зображень. Це накладає обмеження на максимальну швидкість обробки. Таким чином, такі пристрої не підходять для прикладних програм, критичних до часу.

Програмовані логічні інтегральні схеми дозволяють створювати платформу для паралельного виконання. На основі ПЛІС, різні апаратні блоки виконують послідовності алгоритму паралельно, і таким чином, забезпечують швидку реакцію і високу частоту кадрів. Оскільки всі операції виконуються за меншу кількість тактів, енергоспоживання буде значно знижено в порівнянні з конструкціями на основі мікроконтролерів/ процесорів.

Вбудована система - це маленький комп'ютер, вбудований в інтегральну схему. Такі системи призначені для виконання конкретних завдань. Вбудовані системи, призначені для виконання операцій обробки зображень, відомі як вбудовані системи обробки зображень. Найбільш поширеним прикладом такої системи є цифрова камера. Тут попередня обробка і стиснення зображення виконується за допомогою мікроконтролерів. Як правило, вбудовані системи обробки зображень є критичними до часу. Такі системи відомі як системи реального часу. Іншими словами, системи реального часу вимагають відповіді протягом

певного часу, в іншому випадку система вважається несправною. Прикладом такої системи є система, яка використовується для визначення зони аварійної посадки безпілотного літального апарату (БПЛА) з використанням пасивних датчиків (камери). Під час аварійної посадки БПЛА повинен швидко знайти відповідне приміщення і повинен приймати рішення. Така прикладна програма вимагає дуже високої швидкості виконання алгоритмів обробки зображень. У таких прикладних програмах ПЛІС можуть ефективно замінити вбудовані контролери.

# Цифрові технології обробки сигналів

Цифрова обробка сигналів (DSP) використовується в самих різних областях: від телебачення високої чіткості, мобільної телефонії, цифрового аудіо, мультимедіа, цифрових камер, радарів, гідролокаторів, біомедичних зображень, глобального позиціонування, цифрового радіо, розпізнавання мови і т.д. Обробка цифрових сигналів обумовлена вимогами до прикладних програм, які вдалося реалізувати лише завдяки розвитку технології кремнієвих чіпів. Розробка програмованих мікросхем DSP і спеціалізованих рішень для системи на кристалі (SoC) для цих додатків була активною областю досліджень і розробок протягом останніх трьох десятиліть. Дійсно, клас виділених мікропроцесорів був розроблений спеціально для DSP, а саме мікропроцесорів DSP.

Зростаючі витрати на кремнієву технологію мали значний тиск на розробку спеціалізованих рішень SoC і означають, що технологія буде все більше використовуватися для великих обсягів або спеціалізованих ринків. Альтернативою є використання рішень у вигляді мікропроцесорів, таких як мікроконтролери, мікропроцесори і мікросхеми DSP, але в деяких випадках ці пропозиції не відповідають вимогам швидкості, площі і енергоспоживання багатьох додатків DSP. Пізніше, в якості апаратної технології для систем DSP, була запропонована програмована логічна інтегральна схема (FPGA),

оскільки вона дає можливість розробляти найбільш підходящу архітектуру схеми для обчислювальних вимог, вимог до пам'яті і потужності прикладних програм в аналогічному шляху до систем SoC. Це усунуло упередження про те, що ПЛІС використовуються тільки в якості «клейкою логіки», і більш реалістично показує, що ПЛІС представляють собою сукупність системних компонентів, за допомогою яких користувач може створити систему DSP. Незважаючи на те, що готовий аспект FPGA дозволяє уникнути багатьох глибоких проблем, що виникають при розробці реалізацій системи на кристалі (SoC), можливість створення ефективної реалізації на основі опису системи DSP залишається досить заплутаною проблемою.

Технології FPGA представляють явну перевагу з точки зору використання її програмованість для зниження ризику неправильного створення друкованих плат або розвитку виробленого продукту для наступних змін стандартів. Хоча це могло бути правдою в перші роки технології FPGA. Еволюція в кремнієвої технології перетворила FPGA з програмованої технології міжз’єднань в системний компонент. Якщо мікропроцесор або мікроконтролер розглядалися як програмований системний компонент, тоді поточні пристрої FPGA також повинні розглядатися в цьому ключі, що дає нам інший погляд на программируемость системи.

При проектуванні електронних систем головна привабливість мікропроцесорів/мікроконтролерів полягає в тому, що вони значно знижують ризик розвитку системи за рахунок зменшення складності проектування. Оскільки апаратне забезпечення виправлено, всі зусилля з проектування можуть бути сконцентровані на розробці коду, який змусить апарат працювати з необхідною специфікацією системи. Ця ситуація була доповнена розробкою ефективних програмних компіляторів, які в значній мірі усунули необхідність для проектувальника створювати мову асемблера; в деякій мірі це може звільнити розробника від докладного знання архітектури мікропроцесора (хоча багато практики стверджують, що це

важливо для створення гарного коду). Ця концепція набула популярності і мікропроцесори стали невід'ємною частиною будь-якого курсу по електротехніці, електроніці або комп'ютерної інженерії.

Більша частина цього процесу зводилася до здатності розробника програмного забезпечення використовувати базову архітектуру процесора, архітектуру фон Неймана. Однак ця перевага також є обмежуючим фактором в його застосуванні до цифровій обробці сигналів (DSP). В архітектурі фон Неймана операції обробляються послідовно, що дозволяє відносно просто інтерпретувати апаратні засоби для цілей програмування; однак це серйозно обмежує продуктивність в прикладних програмах DSP, які зазвичай демонструють високий рівень паралелізму і в яких операції не залежить від даних, що дозволяє застосовувати оптимізацію. Це вимагає паралельної реалізації, і хоча мікропроцесори йдуть деяким шляхом для вирішення цієї ситуації, забезпечуючи паралелізм в формі паралельної апаратної і програмної «конвейеризации», все ще існує концепція єдиної архітектури, придатною для вирішення всіх завдань DSP.

Це обмеження долається в FPGA, оскільки вони допускають те, що можна вважати другим рівнем програмованості, а саме програмування базової архітектури процесора. Створюючи архітектуру, яка найкращим чином відповідає алгоритмичним вимогам, можна досягти високих рівнів продуктивності з точки зору площі, швидкості і потужності. Ця концепція не нова, оскільки ідея створення архітектури системи, що відповідає алгоритмичним вимогам, є «наріжним каменем» в реалізації інтегральних схем для конкретних додатків або ASIC. Впровадження уУ великих обсягах ASIC привели до найекономічніших, швидких і енергозберігаючих рішень. Проте, зростаючі витрати на технології та їх вплив на реалізації системи, зробили FPGA набагато привабливішою альтернативою. У цьому сенсі FPGA фіксують аспекти продуктивності, пропоновані реалізацією ASIC, але з перевагою програмованісті, зазвичай асоційованої з програмованими процесорами. Таким чином, з'явилися рішення на основі ПЛІС, які в даний

час пропонують кілька сотень гігаоперацій в секунду (GOPS) на одній ПЛІС для деяких додатків DSP, що, по крайній мірі, на порядок вище продуктивності, ніж мікропроцесори.

# Прилад із зарядовим зв'язком (ПЗЗ / CCD)

Концепція пристрою з зарядовим зв'язком (ПЗС) виникла в результаті пошуку на основі кремнію електричного еквівалента магнітної бульбашкового пам'яті. У своїй простій реалізації структура CCD складається з серії близько розташованих електродів, відокремлених від нижче лежачої напівпровідникової підкладки тонким ізолюючим оксидним шаром (рис. 1.1). Коли напруга зсуву застосовується до електрода, область виснаження формується в напівпровіднику безпосередньо під це. Область виснаження фактично є потенційною ямою, яка може зберігати електричний зарядовий пакет. При пульсуючих електродах у відповідній послідовності потенціал значний, отже, його заряд може бути переданий через напівпровідник (рис. 1.2). Зсувний регістр може бути сформований шляхом додавання схем для вставки і виявлення заряджених пакетів.

Хоча ПЗС спочатку замислювався як пристрій для зберігання цифрової інформації, було очевидно, що оскільки потенціал може зберігати змінні величини заряду, він також може передавати аналогові сигнали. В результаті концепція CCD може застосовується до функцій обробки аналогових сигналів, включаючи прості лінії затримки і поперечні фільтри. Без сумніву, однак, найбільший вплив CCD був в застосуванні твердотільних датчиків зображення. Сьогодні ПЗС з'являється в споживчому, промисловому, медичному, охоронному та науковому секторах. Тільки відносно недавнє поява датчиків CMOS поставило під сумнів ПЗС на споживчих ринках мобільних телефонів і цифрових камер, де мініатюризація і зниження енергоспоживання є ключовими вимогами. Це напрошується на

питання про те, чому ПЗС стала кращим датчиком зображення майже для всіх систем візуалізації видимого світла і, зокрема, наукові застосування.

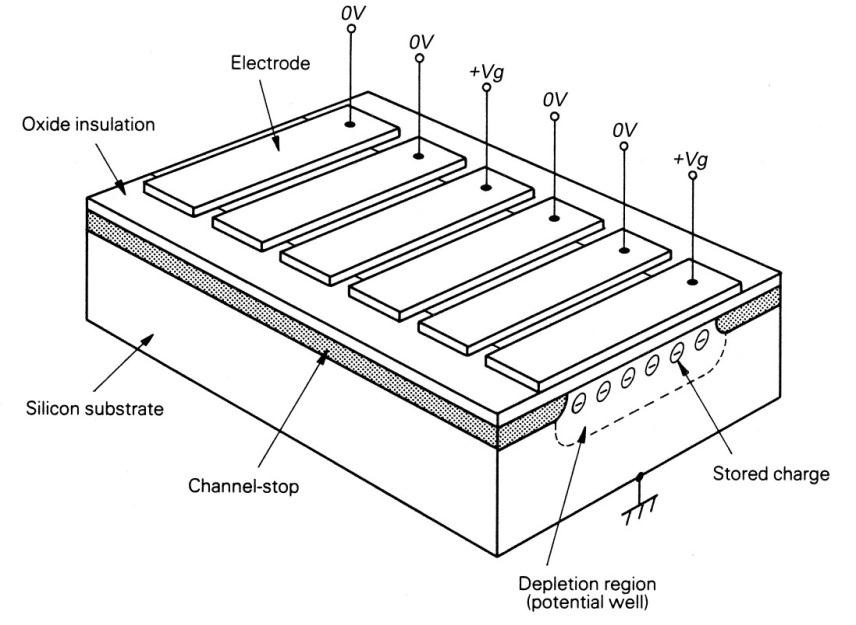


Рисунок 1.1 - Структура пристрою з зарядовим зв'язком

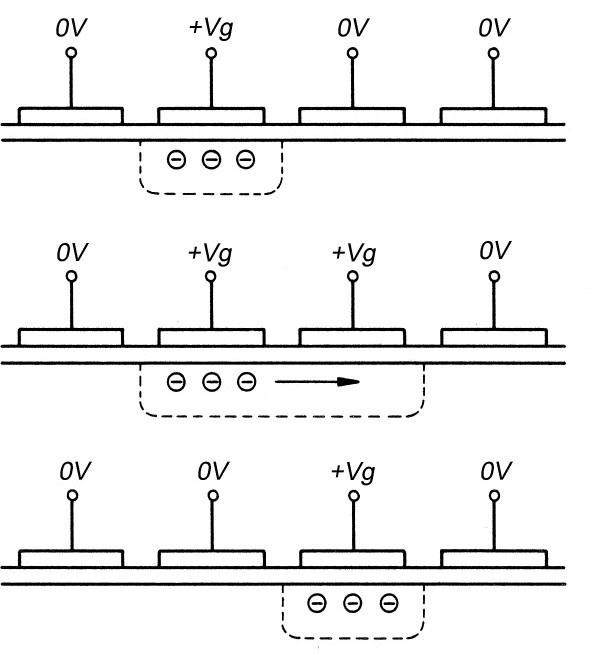


Рисунок 1.2 - Операція переносу заряду

До розробки ПЗС електронна візуалізація грунтувалася головним чином на використанні камерних трубок. Як напівпровідниковий пристрій, ПЗС пропонує безпосередні переваги компактності, міцності і роботи при

низькій напрузі. Важливо, що кремній дуже чутливий до видимого світла. На піку своєї чутливості ПЗС із заднім підсвічуванням здатна поглинати і сприймати майже всі падаючі фотони. При відповідному виготовленні і оптимізації кремній також може добре реагувати на екстремальне ультрафіолетове світло і м'яке рентгенівське випромінювання. У найпростішому вигляді базова структура датчика зображення на ПЗС- матриці сформована з масиву електродів, що йдуть ортогонально до серії ізольованих каналів передачі заряду (рис. 3).

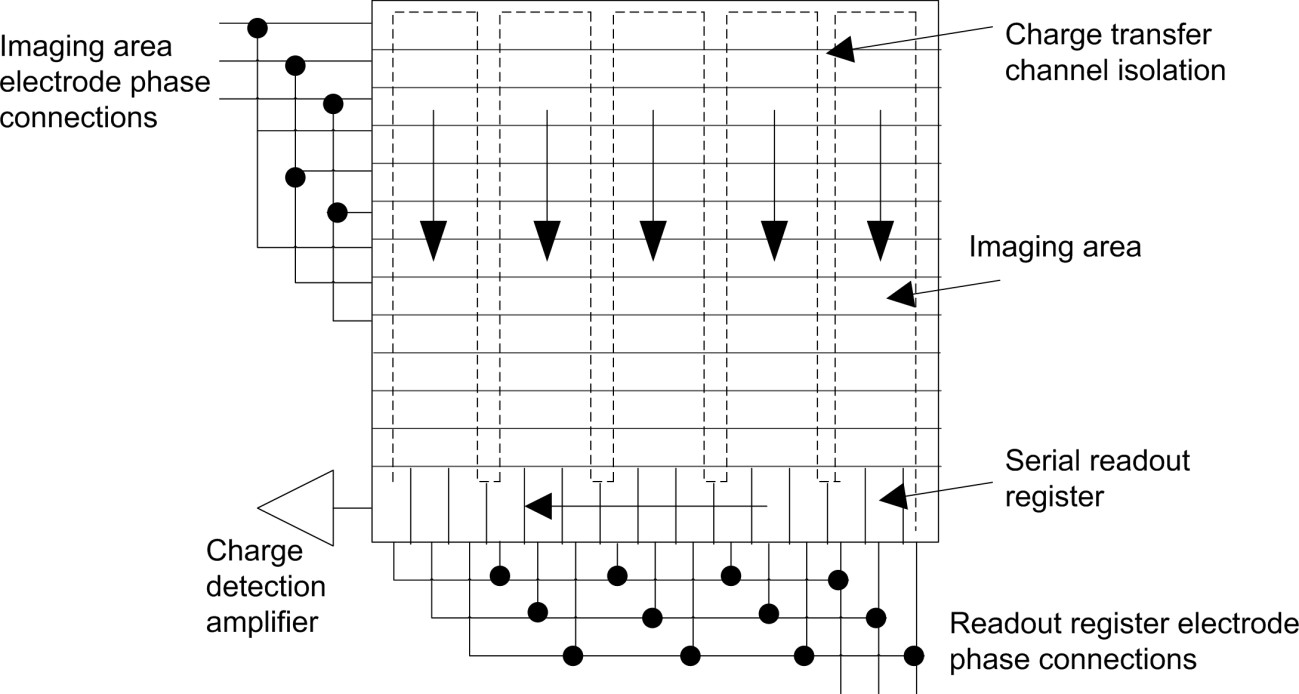
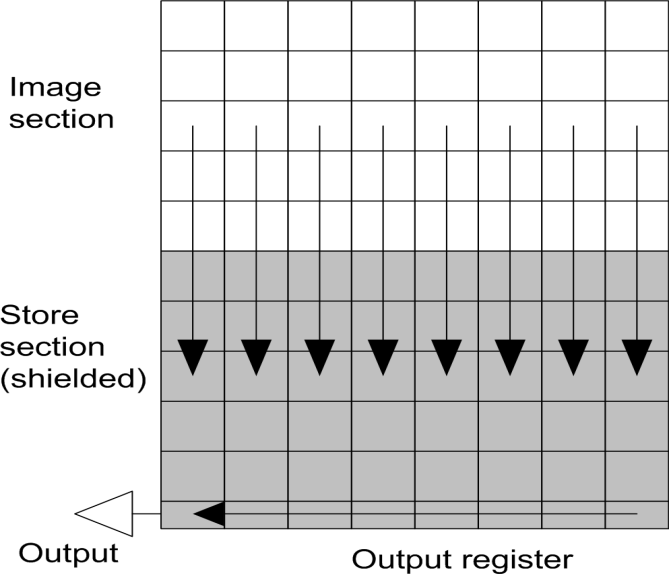


Рисунок 1.3 - Архітектура повнокадровому масиву CCD

Електроди зазвичай з'єднуються в групи з двох, трьох або чотирьох фаз. Зсув фаз електродів створює масив ізольованих потенційних ям або пікселів, які збирають і генеруються фотоном електрони. Після впливу області зображення електроди імпульсні або «синхронізовані», щоб передавати інтегровану діаграму заряду зображення по масиву по одному рядку за раз. Нижній рядок передається в регістр послідовного зчитування, який ортогонально проходить по каналах або стовпцями передачі області

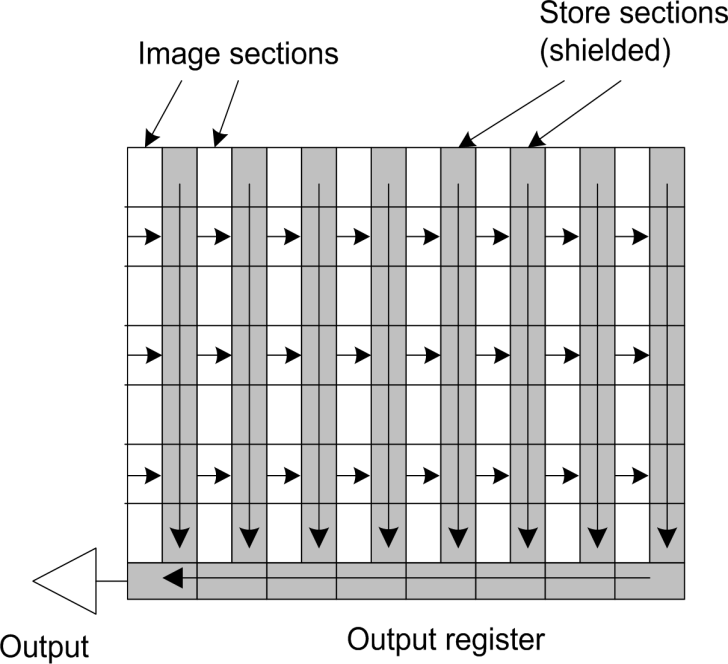
зображення. Цей регістр тактується окремо і з більш високою швидкістю, дозволяючи послідовно зчитувати кожен піксель через підсилювач виявлення заряду. Після зчитування всього рядка електроди області формування зображення можна знову синхронізувати, щоб наступний рядок зображення був переданий в регістр зчитування. Ця послідовність повторюється до тих пір, поки не будуть прочитані всі рядки зображення.

Описана вище архітектура ПЗС зазвичай називається полнокадровою ПЗС-матрицею, найбільш популярною для наукових прикладних програм візуалізації в космічній і наземної астрономії. Однак очевидним обмеженням є те, що зображення буде розмито, якщо ПЗС залишається експонованої під час зчитування, особливо якщо час зчитування кадру становить значну частку часу експонування. Поширеним рішенням є включення механічного затвора перед ПЗС-матрицею, щоб інтегроване зображення можна було зчитувати в темряві. Альтернативне рішення полягає в додаванні світлоекранованої матриці зберігання, в яку інтегрована діаграма зарядів може швидко передаватися в кінці періоду впливу. Масив зберігання може потім зчитуватися, поки нове зображення накопичується в масиві зображень. Існує два основні формати: CCD з передачею кадру (рис. 1.4) і CCD з передачею між рядками (рис. 1.5).



(a)

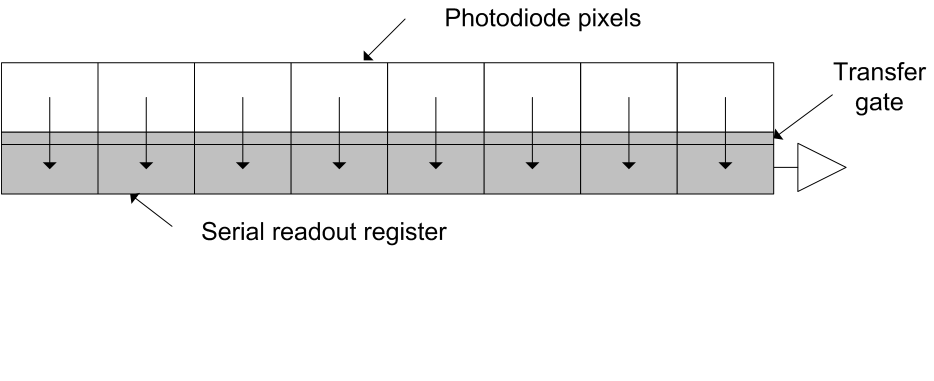
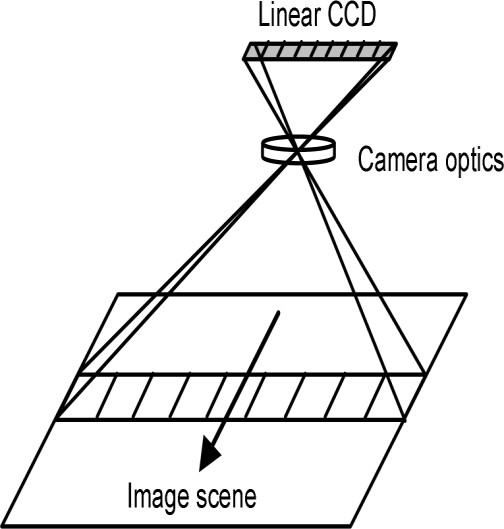
Рисунок 1.4 - Архітектура CCD з передачею кадрів



(b)

Рисунок 1.5 - ПЗС-архітектура з інтерлайн-перенесенням

У ПЗС з передачею кадрів масив зберігання додається під масив зображення. Це ефективно подвоює розмір кремнієвого чіпу і непривабливо в комерційних чутливих до вартості прикладних програмах. У ПЗС з міжрядковою передачею масив зберігання вбудований в світлоекрановані стовпці поруч зі стовпцями зображень. Недоліком цього підходу є те, що чутливість фактично зменшується вдвічі. Не дивно, що ця архітектура мало цікавила наукову спільноту, але швидко завоювала популярність на ринках цифрових камер і відеокамер. Ще дві CCD-архітектури, застосовні до дистанційного зондування Землі і планети, - це лінійна CCD (рис. 1.6) і CCD з інтеграцією з затримкою за часом (TDI) (рис. 1.7). Лінійна ПЗС складається з ряду фотодіодів, з яких генерується фотоном заряд передається в регістр послідовного зчитування. Отримання двовимірного зображення засноване на скануванні датчика за цільовою сцені. Лінійні ПЗС з 12 000 пікселів або більше легко доступні. Така техніка формування зображення використовується для отримання зображень з дуже високою роздільною здатністю (рис. 1.6).

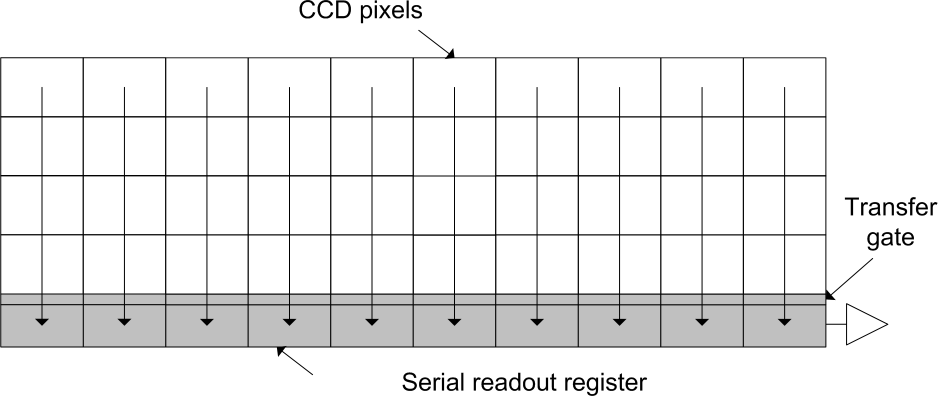


(a)

(b)

Рисунок 1.6 - Лінійна ПЗС (ліворуч), зображення з мітлою (праворуч)

TDI CCD є перевагою, коли недостатньо освітлення для отримання корисного сигналу від звичайної лінійної матриці. Його архітектура в основному така ж, як у повнокадровому CCD-масиві, за винятком того, що формат зображення *x-y* зазвичай набагато більший по *x*, ніж по *y* (рис. 1.7). Щоб уникнути появи плям, для формування зображення методом "pushbroom" потрібно, щоб інтегровані пакети заряду ПЗС були синхронізовані з масиву з тією ж швидкістю, що і проектується зображення (рис. 1.8). TDI забезпечує посилення, тому що цільова сцена тепер інтегрована за кількістю рядків у масиві. ПЗС-матриці TDI були успішно використані для отримання зображень з дуже високою роздільною здатністю.



(a)

Рисунок 1.7 - ПЗС-матриця TDI.



Рисунок 1.8 - TDI ПЗС-матриця pushbroom

# Технології CCD

Технології виготовлення ПЗС постійно удосконалюються для підвищення продуктивності. Розробка великоформатних ПЗС-матриць для наукових прикладних програм була в значній мірі обумовлена прагненнями наземного астрономічного співтовариства і вимогою використання твердотільного датчика зображення для заміни камери відеоспостереження в космічних застосуваннях. В кінці 1970-х років ряд астрономічних дослідницьких груп розробили системи камер, використовуючи перші комерційні доступні датчики. Найбільш ранні датчики мали всього 100 пікселів на 100 пікселів. Пізніше з'явилася задача розробки CCD зі 320 пікселями і 512 пікселями - мінімум, необхідний для відповідності вимогам стандарту телебачення США. Texas Instruments спільно з JPL за спонсорської підтримки НАСА розробили масиви 400 × 400 пікселів, а потім 800 × 800 пікселів - початок наукової CCD. Це прагнення створити твердотільний датчик зображення великої площі з високою чутливістю і низьким рівнем шуму привів до багатьох розробок, які тепер можна побачити в сучасному науковому або науковому каталозі ПЗС. Наукова ПЗС-матриця

зазвичай має кілька мільйонів пікселів, високу квантову ефективність в широкому спектральному діапазоні, низький рівень шуму.

**Ефективність переносу заряду.** Спочатку значного розвитку у виробництві CCD потребувала низька ефективність передачі заряду (CTE) простою архітектурою CCD з вбудованим каналом. Було виявлено, що інтерфейсні «пастки» на кордоні розділу діоксид-кремнію поглинають і вивільняють заряд з різними постійними часу, що призводить до розмивання пакетів зарядів, які передаються через них. CСD настільки бідний, що робить сенсори зображень великого формату непрактичними. Проблема була вирішена з введенням CCD з изольованим каналом, в якому додатковий шар кремнію n-типу формується у верхній частині підкладки p-типу і безпосередньо під ізолюючим шаром діоксиду кремнію. Кремній n-типу змушує потенціал формуватися глибше всередині підкладки і далеко від пасток на основі інтерфейсу діоксид-кремнію. Сигнальний заряд тепер може передаватися через підкладку, не вступаючи в контакт з пастками. Ефективність передачі тепер вище 99,999%, що дозволяє здійснювати тисячі передач без значної втрати сигналу або розмазування.

**Зчитування шуму.** Іншою важливою подією було зменшення шуму зчитування. Вихідною схемою ПЗС є підсилювач виявлення заряду, що складається з вихідний дифузії, яка має відповідну ємність паразитного вузла, транзистор скидання для перезарядки ємності вузла і вихідний транзистор, що працює в якості джерела-повторювача для визначення змін напруги на вузол (рис. 1.9). Для зчитування пікселів потрібно, щоб ємність вихідного вузла була попередньо заряджена до потенціалу «скидання» (Vrd) шляхом подачі імпульсів на транзистор скидання «вкл» і «викл» за допомогою тактового імпульсу скидання (r), як показано на рисунку 1.10. На практиці є невеликий розряд вихідного вузла, що виникає через подачу тактових імпульсів скидання, коли вони падають на низькому рівні. Сигнальний заряд послідовного зчитування (Vg) потім передається на вихідний вузол, що призводить до подальшої розрядки. Вихідний транзистор

видає вихідний відеосигнал (Vos). Коли час скидання зменшується шум Джонсона від опору канала транзистора заморожується на вихідний ємності

(C) і призводить до флуктуації напруги скидання. Цей шум зазвичай називають шумом скидання і він є домінуючим джерелом шуму зчитування CCD без методу придушення, відомого як корельована подвійна вибірка (CDS). CDS отримує дві вибірки вихідного сигналу CCD, одну до (S1) і одну після (S2) сигнальний заряд був перенесений на вихідний вузол (рис. 1.10). Віднімання двох вибірок усуває цей шум, оскільки він залишається

«корельованим» або незмінним між двома вибірками. Після скасування шумів скидання домінуючим джерелом шуму стає шум транзистора у вихідному транзисторі витокового повторювача. Результуючий шум зчитування буде залежати від ширини смуги вимірювання сигналу і, отже, від швидкості зчитування CCD. Більш високі швидкості зчитування вимагають більшої смуги пропускання зчитування і призводять до більш високого шуму зчитування. Кращі з сучасних ПЗС-матриць досягають шуму зчитування шести електронів RMS при зчитуванні на ≈ 1 МГц або всього лише два електрона RMS при зчитуванні на ≈ 100 кГц. Потрібно компроміс між низькою швидкістю зчитування, щоб мінімізувати шум зчитування і високою швидкістю зчитування, щоб мінімізувати загальний час зчитування кадру. Висока швидкість зчитування кадрів явно бажана для максимізації ефективності спостереження. Одним з рішень цієї дилеми є додавання додаткових підсилювачів зчитування в матрицю CCD так, щоб кілька підсекцій можна було читати паралельно. Низька швидкість зчитування для мінімального шуму тепер можлива без шкоди для ефективності спостереження. Додатковою перевагою є те, що додаткові підсилювачі зчитування забезпечують ступінь надмірності в разі відмови одного підсилювача зчитування. Недоліком, однак, є те, що потрібно додаткове калібрування даних для виправлення невідповідності посилення між декількома підсилювачами зчитування.

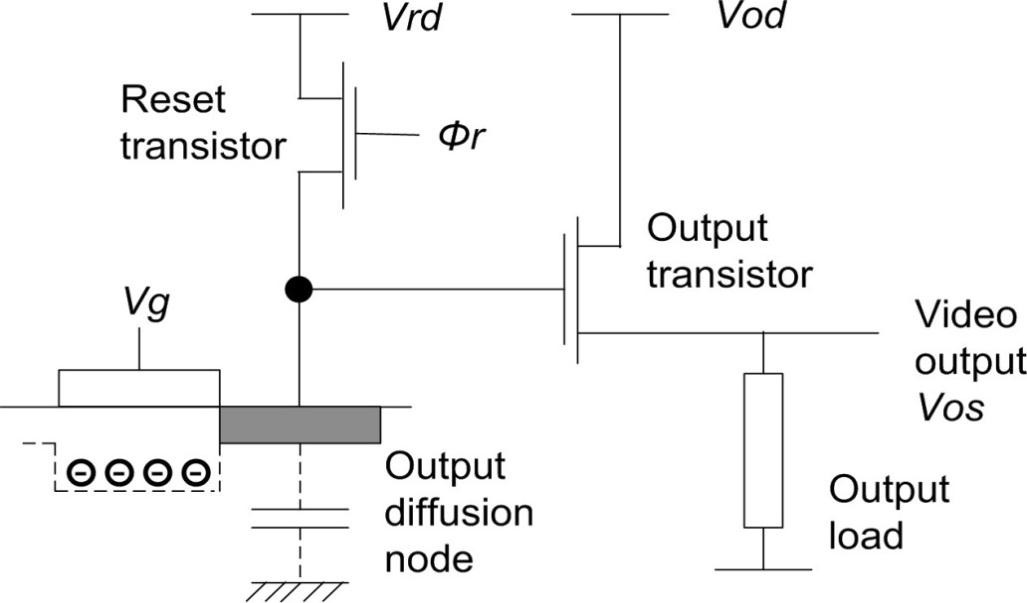


Рисунок 1.9 - Підсилювач виявлення заряду ПЗС

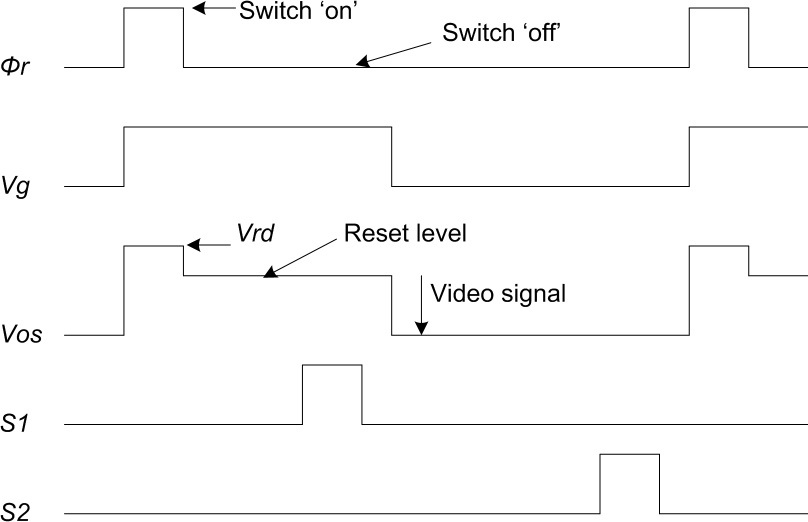


Рисунок 1.10 - Корельована подвійна вибірка

**Показання множення електронів.** Для того, щоб зменшити шум зчитування, два виробника ПЗС розробили електронні показники множення для забезпечення посилення на кристалі. Регістр множення, зазвичай складається з декількох сотень етапів, додається в кінець регістру

послідовного зчитування (рис. 1.11). Одна фаза електрода в регістрі множення синхронізується при значно більш високій напрузі (зазвичай до 40 В за амплітудою), що забезпечує можливість лавинного множення (рис. 1.12). Слово «ймовірність» є ключовим, оскільки чисте посилення одиночній ступені може становити лише 1,01, але після 500 каскадів чисте посилення становить 1,01500 або 145. Оскільки посилення застосовується до вихідного підсилювача виявлення заряду, ефективний шум зчитування ділиться. Це дозволяє підтримувати дуже низький шум зчитування при високих швидкостях зчитування телевізійних камер (10 МГц).

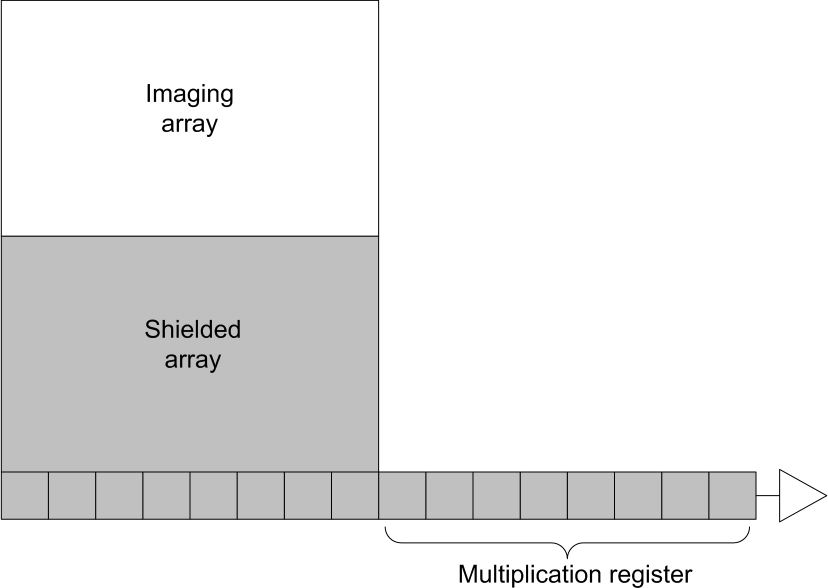


Рисунок 1.11 - ПЗС-матриця множення електронів

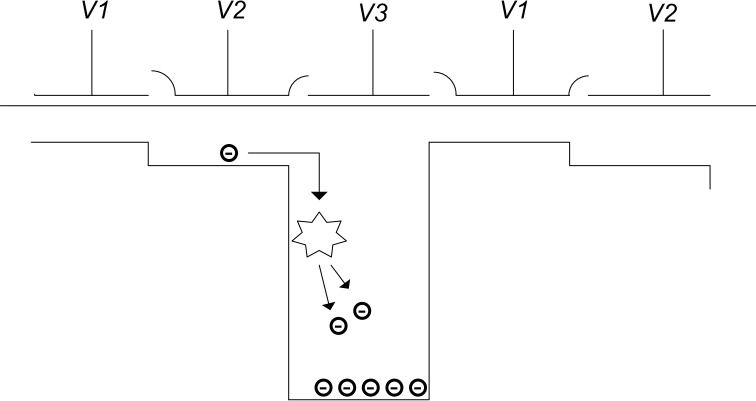


Рисунок 1.12 - Лавинний пробою посилення

**Спектральний діапазон.** Третім триваючим розвитком CCD була задача максимізувати його чутливість в самому широкому спектральному діапазоні. Перші ПЗС були «висвітлені спереду», що вимагало те, щоб падаючі фотони проникали через напівпрозорі електроди з полі кремнію для того, щоб досягти основної підкладки. На жаль, глибина поглинання кремнію така, що велика частина світла поглинається усередині електродів, особливо на більш коротких «синіх» довжинах хвиль. З іншого боку, існує велика ймовірність того, що довші «червоні» з ближньої інфрачервоної області спектра фотони будуть глибоко проникати в кремнієву підкладку нижче областей виснаження. Електрони від цих фотонів можуть вільно переміщатися в бічному напрямку в вільної від поля області, в результаті чого дозвіл зображення погіршується або розмивається. Рішення полягає в тому, щоб виготовити ПЗС на епітаксиальному кремнії, який складається з тонкого шару номінально легованого кремнію, зазвичай товщиною від 10 до

20 мкм, поверх об'ємної підкладки з дуже високим ступенем легування. Електрони, створені в об'ємної підкладці, ймовірно, рекомбинують, перш ніж вони дрейфують в області збіднення. Роздільна здатність зображення підтримується за рахунок чутливості до «червоним» довжинах хвиль. Щоб вирішити проблему відсутності чутливості до «блакитних» довжин хвиль, дослідники і виробники почали розробляти витончені ПЗС з «заднім підсвічуванням». Це засновано на тому, що велика частина кремнієвої підкладки віддаляється хімічним травленням, а ПЗС-підсвічування освітлювали не через полісиліконові електроди, а через задню поверхню. В принципі, чутливість, або квантова ефективність (QE), дуже сильно збільшена. Однак на необробленій поверхні кремнію буде висока щільність центрів рекомбінації або центрів захоплення. Будь-які фотоелектрони, які вступають в контакт з цією поверхнею, будуть повторно об'єднуватися з отворами і будуть втрачені. Отже, пасивація поверхні необхідна, щоб скористатися всіма перевагами проріджування, особливо для більш коротких

«синіх» довжин хвиль, які поглинаються близько до цієї поверхні. Метод, який званий зарядкою на задній поверхні, грунтувався на тому, що ПЗС-

матриця тримається в вакуумі і піддається впливу ультрафіолетового випромінювання. Це призвело до того, що природний оксид утримував негативний заряд, який відштовхував фотогенеровані електрони в напрямку областей виснаження. Експерименти показали дуже високий QE, але процес був нестабільним і вимагав періодичної перезарядки. Більш надійний і стабільний метод, званий легуванням на задній поверхні, спирається на тонкий шар легуючої домішки, зазвичай бору, який включається в задню поверхню для утворення шару p +. Зміна концентрації легування приводить до невеликого потенційному кроку, який відштовхує фотогенерованих електрони до областей збідніння.

**Покриття в фокальній площині.** Прикінцеві значні розробки ПЗС наукового рівня були спрямовані на вимогу більшого охоплення фокальній площині, ніж це було забезпечено першими ПЗС, які були обмежені максимальним розміром сітки виробничої фотолітографії до 20 мм 20 мм. Техніка «зшивання», як показано на рисунку 1.13, була розроблена для того, щоб дозволити побудувати ПЗС-матрицю великого формату за допомогою точної літографії з набору сіток, кожна з яких містить окрему частину загального дизайну ПЗС.

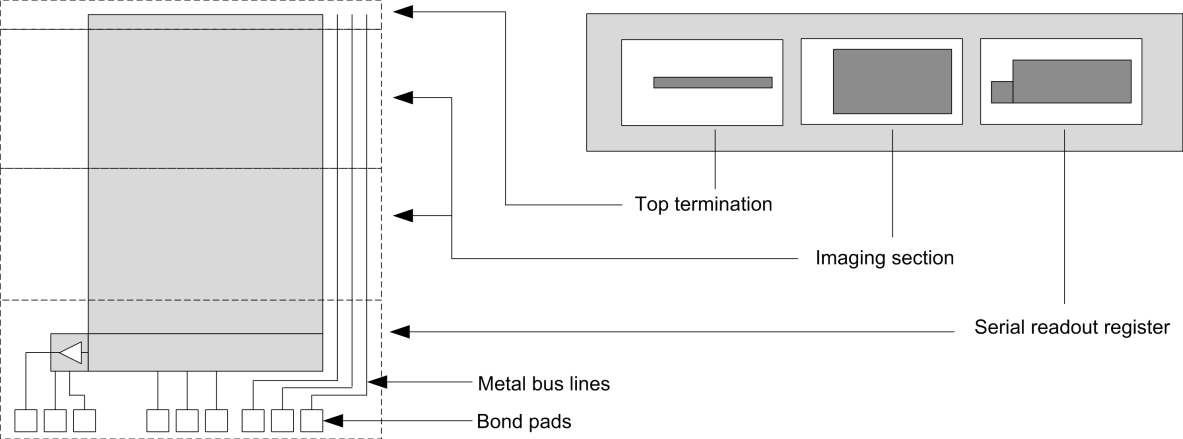


Рисунок 1.13 - Виготовлення ПЗС великого формату шляхом зшивання

**Динамічний діапазон.** Нарешті, дві з найбільш важливих метрик ПЗС наукового рівня - це повна ємність і шум зчитування. Разом вони визначають динамічний діапазон CCD приблизно 10 мільйонів або більше,

які можна отримати від кращих на сьогоднішній день пристроїв. З цими параметрами пов'язана відносна лінійність, яка зазвичай становить від 0,5% до 1% і в якій переважають характеристики підсилювача виявлення заряду. Повна ємність - це міра максимального сигнального заряду, який піксель може зберігати і передавати, і вона є функцією ізольованою області зберігання електронів, яка визначається як розміром пікселя, так і кількістю фаз електрода, використовуваних для визначення пікселя. Чотирьохфазний піксель утримує і переносить заряд на половині пікселя, тоді як трифазний піксель може працювати тільки з однією третиною площі пікселя. Двофазний піксель працює на площі менше половини площі пікселя, але щільність зберігання обмежена значно меншою глибиною потенційної ями. Наприклад, піксель 12х12 мкм зазвичай зберігає 60 000 електронів (2- фазний), 100 000 електронів (3-фазний) і 200 000 електронів (4-фазний). 4- фазний піксель розміром 24 мкм, 24 мкм може вмістити понад 1000000 електронів.

# Комплементарна структура метал-оксид-напівпровідника (КМОП / CMOS)

Поява технології CMOS-датчика зображення, мабуть, є найбільш важливим досягненням в області твердотільної візуалізації з моменту винаходу ПЗС. КМОП-датчики зображення, більшість з яких сьогодні можна назвати КМОП-активними пиксельними датчиками (APS), використовують ту ж технологію кремнієвих чіпів, яка використовується в мікропроцесорних системах. Залучення в основному пов'язано з природою технології CMOS в тому, що багато мільйонів транзисторів можуть бути інтегровані в одну кремнієву ланцюг. Це дає можливість інтегрувати великий масив пікселів, кожен зі своїм власним фотодіодом і транзисторами зчитування, поряд з усією допоміжною електронікою, необхідною для адресації масиву, буферизації аналогового відеосигналу і навіть оцифровки готового для обробки, зберігання або відображення.

КМОП-датчики, такі як КМОП-інтегральні схеми, працюють при значно більш низькій напрузі, ніж ПЗС-матриці, зазвичай від 1,8 В до 5 В, в залежності від форми процесу і геометрії. Масштабна функціональна інтеграція і низьковольтна робота явно дуже привабливі на споживчих ринках, які вимагають компактності, тривалого терміну служби батареї і низьких виробничих витрат. Як наслідок, багато цифрових камер і мобільних телефонів тепер використовують CMOS, а не датчики CCD.

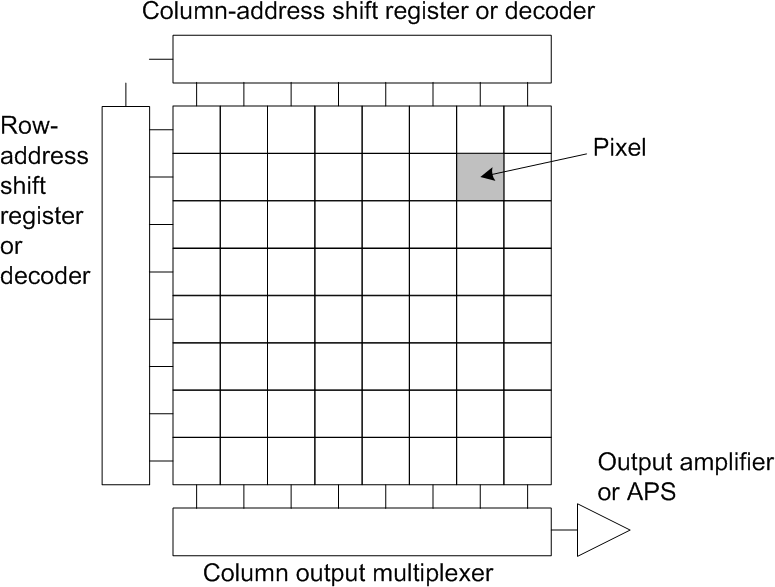
У найпростішому вигляді піксель CMOS складається з фотодіода і трьох транзисторів: один для попередньої зарядки фотодіода, один для вимірювання напруги сигналу на фотодіоді і один для вибору рядка. Масив пікселів зазвичай буде адресуватися за допомогою регістрів зсуву (рис. 1.14), хоча в деяких реалізаціях використовуються адресні декодери для забезпечення довільного доступу до пікселів. Доступ до масиву пікселів здійснюється по одному рядку за раз, завдяки включенню всіх транзисторів вибору рядка в одному рядку пікселів. У нижній частині масиву окремі пікселі в рядку вибираються і зчитуються стовпець за стовпцем. Відеосигнал може подаватися через мультиплексор на аналоговий вихідний підсилювач, послідовний аналого-цифровий перетворювач.

Рисунок 1.14 - Архітектура масиву

Недолік основного трехтранзисторного пікселя полягає в тому, що він схильний до дії шуму kBT C, який зазвичай є основним джерелом шуму зчитування. Виявлено, що CDS працює з різним ступенем успіху, зберігаючи кадр «скидання» поза чіпа до експонування, а потім віднімаючи його з кадру зчитування «сигналу». Для включення вбудованого CDS дослідники і виробники розробили чотирьохтранзисторні пікселі, в яких новий транзистор використовується в якості затвора між фотодіодом і сенсорним транзистором. Концепція полягає в тому, що вихідний вузол може бути обраний після скидання і до перемикання сигналу з фотодіода на вихідний вузол. Це схоже на схему вихідного підсилювача ПЗС, але вимагає, щоб фотодіод був «закріплений» до відповідного потенціалу. Це гарантує, що весь заряд сигналу передається на вихідний вузол, що дозволяє уникнути затримки зображення і що шум в kBT C залишається корельованим між двома вибірками CDS. Недолік закріпленого фотодіода полягає в тому, що закріплення значно знижує ємність накопичення заряду пікселя і, отже, також можливо застосовувати лінійний динамічний діапазон.

**Квантова ефективність.** Пошуки ефективного датчика CMOS сьогодні ще тільки зароджуються. Квантова ефективність CMOS-датчиків з фронтальним підсвічуванням знижується за рахунок електроніки в пікселях і алюмінієвих шинних доріжок, що знижує «коефіцієнт заповнення», міру частки площі пікселя, яка фактично чутлива до світла. Розташування типового пікселя CMOS показано на рисунку 1.15. У цьому приладі фотодиод займає тільки 19% площі пікселя (як визначено пунктирною лінією), але зазвичай дає коефіцієнт заповнення 30%. Втрати виникають в результаті відображення світла від алюмінієвих шин і індукованих фотонами електронів в підкладці, які поглинаються і губляться в електроніці транзистора в пікселях. Очевидне рішення проблеми полягає в тому, щоб стоншити датчик із заднім підсвічуванням, і кілька дослідницьких груп в даний час продемонстрували CMOS-датчики із заднім підсвічуванням, які досягають квантової ефективності відповідно до своїх аналогів CCD.

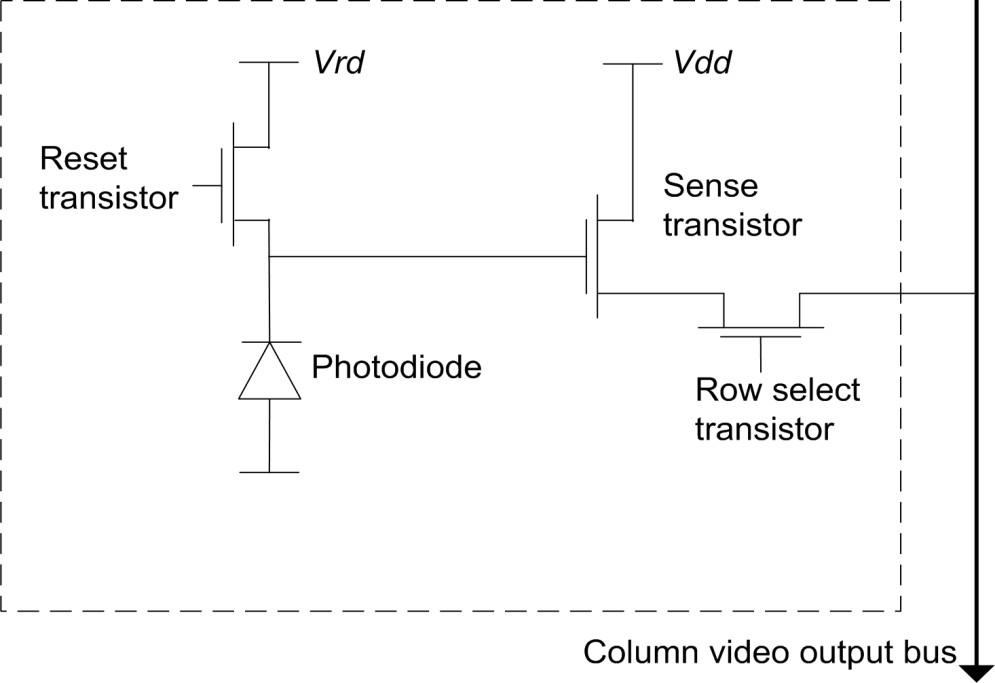


Рисунок 1.15 - Трьохтранзисторний CMOS активний піксельний

датчик

**Індикація шуму і динамічного діапазону.** Другою областю досліджень є мінімізація шуму зчитування і максимізація ємності зберігання та лінійності. Обмеженням є лінійне коливання напруги, яке може бути отримано за межами порогових областей транзистора в сучасних низьковольтних КМОП-процесах. Лінійний динамічний діапазон кращих на сьогоднішній день датчиків CMOS становить 5000, що значно менше, ніж у ПЗС. Досліджується кілька підходів до подолання проблеми, включаючи концепцію пікселів, що навмисно поводяться нелінійно, і датчики, які дозволяють окремим пікселям мати різні періоди експозиції. Більш складні п'яти- і шести-транзисторні пікселі також досліджуються з метою збільшення динамічного діапазону. Можна очікувати подальшого прогресу в майбутньому, оскільки дослідники адаптуються до використання переваг технології CMOS.

# Основи системи DSP

На зорі електроніки сигнали оброблялися і передавалися в їхньому природному стані. Зазвичай це був аналоговий сигнал, створений з вихідного сигналу, такого як мова, потім перетворений в електричні сигнали перед передачею через відповідне середовище передачі або широкосмугове з'єднання. Привабливість обробки сигналів у цифровій формі була визнана досить давно, по ряду причин. Цифрове обладнання, як правило, краще і надійніше аналогового, але може бути схильне до старіння і може давати невизначені характеристики при виробництві DSP, з іншого боку - дає гарантовану точність і практично ідеальну відтворюваність. Крім того, існує значний інтерес до об'єднання безлічі мереж, які передають ці сигнали, таких як мережі телефонної передачі, наземні телевізійні мережі і комп'ютерні мережі в одну або кілька цифрових середовищ передачі. Це забезпечує сильну мотивацію для перетворення широкого спектра інформаційних форматів в їх цифрові формати.

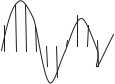
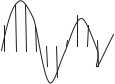
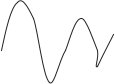
Мікропроцесори, DSP-мікросхеми і FPGA мають відповідну платформу для обробки таких цифрових сигналів, але важливо зрозуміти ряд основних питань, пов'язаних з реалізацією алгоритмів на DSP. Вибір алгоритму і арифметичних вимог може мати серйозні наслідки для якості кінцевої реалізації. Концепції DSP впливають на реалізацію апаратного забезпечення, такі як частота дискретизації, швидкість обчислень і затримка.

Існує зростаюча потреба в обробці, інтерпретації та розумінні інформації, включаючи численні промислові, військові та споживчі програми. Багато з них включають мову, музику, зображення або відео або можуть підтримувати системи зв'язку за допомогою виявлення та виправлення помилок і алгоритмів криптографії. Це включає в себе обробку в реальному часі значної кількості різних типів контенту з серією частот дискретизації в діапазоні від 1 Гц (як в біомедичних прикладних програмах) до десятків МГц (як в прикладних програмах для обробки зображень). У багатьох випадках мета полягає в тому, щоб обробити дані для посилення

частини сигналу, таких як при виявленні контурів при обробці зображень, або усуненні перешкод в радіолокаційних прикладних програмах, або видалення помилкового введення, як в разі луни або шумозаглушення в телефонії. Інші алгоритми DSP необхідні для захоплення, зберігання та передачі даних, аудіо і відео. Методи стиснення були успішно використані в цифровому мовленні та телекомунікації.

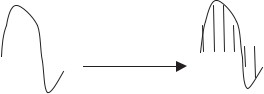
Що стосується стиснення зображень, то розвиток спільної групи експертів по фотографії (JPEG), а потім групи експертів по руху зображень (MPEG) привело до розробки стандартів JPEG і MPEG відповідно. Ці стандарти включають ряд основних алгоритмів DSP, зокрема DCT і оцінку і компенсацію руху.

Основне поширення DSP обумовлено доступністю все більш дешевого устаткування, що дозволяє легко зв'язати цю технологію з комп'ютерними технологіями і в багатьох випадках впроваджувати на тих же комп'ютерах. Потреба в багатьох прикладних програмах зумовила потребу у все більш складних системах DSP, що, в свою чергу, призвело до розширення галузі досліджень, пов'язаних з розробкою ефективної реалізації деяких алгоритмів DSP.

Базова реалізація систем DSP (рис. 1.16) показує, як сигнал оцифровується з використанням аналого-цифрового (A / D) перетворювача, обробляється в системі DSP перед перетворенням назад в аналоговий сигнал.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A/D конвертер |  | DSP система |  | D/A конвертер |

Квантування



Оцифровка

Рисунок 1.16 - Базова система DSP

Тут аналоговий сигнал набуває загальну якість зображення. Збільшуючи кількість даних, що видаляються, можна домогтися більшого зменшення розміру файлу за рахунок зниження якості зображення. Вейвлет- перетворення пропонують інформацію як в часовій, так і в частотній областях і грають роль не тільки в прикладних програмах для стиснення зображень але також для вилучення ключової інформації з сигналів і для придушення шумів. Одним з таких прикладів є отримання ключових ознак з медичних сигналів, таких як електроенцефалограма (ЕЕГ). Оцифровка медичних сигналів, таких як ЕЕГ, є результатом електричної активності в мозку, знятої з електродів, що контактують з поверхнею шкіри. За допомогою збору інформації основні сигнали можуть бути сильно

«забруднені» шумом. Частота дискретизації безпосередньо пов'язана зі швидкостями обчислень DSP і, отже, вимогами до продуктивності з точки зору необхідної пропускної спроможності. Якщо частота вибірки використовується в якості що вводить в оману порівняння, то тактова частота процесора також може розглядатися такою що вводить в оману метрика. Тактова частота зазвичай вказується постачальниками технологій як міра можливої продуктивності, оскільки пропускна здатність може бути розрахована шляхом ділення тактової частоти на кількість циклів, які необхідно виконати для вибірки. Найбільш помітно, що персональні комп'ютери (ПК) вказують тактові частоти в якості показника продуктивності, але найважливіше пропускна здатність.

# Чому ПЛІС? Впровадження та реалізація

Технології на основі ПЛІС за своєю природою паралельні. Різні послідовності алгоритмів будуть відображатися на різні апаратні модулі в ПЛІС, які працюють одночасно. Наведемо основні причини вибору FPGA в якості вбудованої платформи обробки зображень:

* паралельна робота;
* швидкість виконання;
* гнучкість;
* виконання з низьким енергоспоживанням.

**Паралельна робота.** Вбудований алгоритм формування зображення може бути реалізований або з використанням мікроконтролера DSP- процесора, або з використанням FPGA. Мікроконтролер - це мінікомп'ютер, вбудований в мікросхему, який виконує певне завдання. Інструкції в шістнадцятковому форматі, які записуються в чіп, будуть виконуватися послідовно і декодовані в керуючі сигнали для виконання необхідного завдання.

Із зовнішньої точки зору FPGA представляє собою набір логічних елементів, які можуть бути електрично підключені. FPGA реалізує прикладні програми, розробляючи окремі апаратні засоби для кожної функції, і, отже, такі конструкції за своєю природою паралельні. Кожна функція, яку вводить програміст, буде відображатися в окремий апаратний компонент. Таким чином, така конструкція підходить для тих алгоритмів обробки зображень, в яких є значний паралелізм.

**Швидкість виконання.** Через паралельний характер FPGA швидкість виконання буде значно збільшена. У практичних прикладних програмах зображення буде розділене на різні субблоки, а потім кожен блок буде оброблятися паралельно. Це прискорить весь алгоритм. Загальна кількість процесорів буде дорівнювати кількості паралельних блоків.

**Гнучкість.** Система на основі FPGA забезпечує повну гнучкість програмування. Сучасні FPGA мають достатні логічні ресурси для реалізації навіть складних додатків в одному кристалі. Сучасні системи на основі FPGA будуть адаптивно перенастроюватися у відповідності з різними операційними середовищами. Отже, системи на основі FPGA за своєю природою є гнучкими.

**Виконання з низьким енергоспоживанням.** Схема на основі FPGA одночасно виконує кілька операцій за один такт. Це дозволяє значно знизити

тактову частоту. Фактично відбудеться зниження тактової частоти в порівнянні з послідовним процесором величиною 2 або більше. Зниження тактової частоти відповідає зниженню динамічного енергоспоживання системи. Таким чином, дизайн на основі FPGA полегшує проектування з низьким енергоспоживанням.

# Функціональність и продуктивність системи

Розроблена апаратна система повинна бути здатна захоплювати безперервний відеопотік з цифрової камери і повинна вміти обробляти його для виявлення країв у кожному кадрі. Граничне значення для виявлення країв має бути адаптивно обчислено для кожного кадру. Система повинна відображати оброблені кадри на пристрої відображення з виявленими краями, виділеними білим кольором. У симуляції середовища повинна бути розроблена модель для імітації функціональних можливостей цифрової камери (система збору даних). Дані зображення повинні бути перетворені в сигнали, які ідентичні сигналам від цифрової камери. Модуль обробки зображень на основі FPGA повинен бути змодельований в програмній платформі. Цей модуль повинен приймати вхідні дані від імітаційної моделі системи збору даних, формувати виявлений край кадру і відображати його в форматі зображення/відео в графічному інтерфейсі користувача (GUI). Графічний інтерфейс повинен дозволяти користувачам аналізувати форми вихідний хвилі і потік даних.

Продуктивність системи оцінюється за кількістю кадрів, які система може обробити в секунду. Цільова продуктивність складає 50 кадрів в секунду для 256 кадрів, що набагато вище частоти кадрів, яка може бути отримана в послідовному середовищі, такому як Matlab (в Matlab вона становить від 10 до 12 кадрів в секунду). Тактова частота системи повинна бути розрахована відповідно до цільової швидкості системи. Іншим важливим виміром продуктивності системи є затримка системи. Це різниця

між часом подачі пікселів в систему і часом отримання повністю оброблених пікселів на виході.

Алгоритм обробки зображень, реалізований у вбудованій платформі, відомий як обробка вбудованих зображень. Існує два типи систем обробки вбудованих зображень: апаратне і програмне. Вбудовані системи апаратного забезпечення щодо швидше і в основному розроблені з використанням FPGA. Оскільки ПЛІС можуть бути такими, що переконфігуровуються, то може бути досягнута така гнучкість, як і у вбудованої системи на основі програмного забезпечення, покращуючи швидкість, але тільки за рахунок збільшення вартості і рівня складності при проектуванні системи.

# Системи реального часу

Система реального часу - це система, в якій відповідь на подію повинна відбуватися протягом певного періоду часу, в іншому випадку система вважається збійною. З точки зору обробки зображень, система формування зображень в реальному часі - це система, яка отримує зображення, обробляє ці зображення для отримання деяких результатів, а потім використовує ці результати для подальшої обробки. Відповідь на подію має відбуватися за певний (вказане) час. Прикладами є системи бачення робота, в яких захоплені зображення будуть проаналізовані, щоб виявити перешкоди на його шляху.

Системи реального часу поділяються на два типи: жорсткі і м'які в реальному часі. У жорстких системах реального часу система вважається збійною, якщо відповідь не сталася в зазначений час. Прикладом є аварійна посадка транспортного засобу. Якщо місце посадки не визначено в зазначений час, система вважається збійною. З іншого боку, м'яка система реального часу - це система, в якій система не буде повністю відмовлена, навіть якщо відповіді будуть із запізненням. Наприклад, передача відео

через інтернет. Якщо кадр затримується або не декодується належним чином, це погіршить якість відео.

Простіше кажучи, затримка вбудованої системи формування зображення являє собою різницю між часом зчитування пікселя і часом, коли він відображається після здійснення внутрішніх операцій. Чим менше затримка, тим краще система. Один із способів зменшити затримку - збільшити системну тактову частоту. Але це призведе до збільшення енергоспоживання. Іншим і найбільш ефективним методом є реалізація багатоступінчастого конвеєра. Конвеєр являє собою набір апаратних блоків, які одночасно обробляють різні пікселі.

Пропускна здатність системи є синонімом ширини смуги системи. З точки зору вбудованої системи формування зображення, пропускна здатність системи - це кількість пікселів, оброблених за один тактовий імпульс. Операція з декількома пікселями потребує більше часу і меншої тактової частоти. Це знизить загальну швидкість системи, але пропускна здатність буде збільшена. Збільшення пропускної здатності системи може бути досягнуто за допомогою багатоступеневих конвеєрів. В таких проектах, в той час як поточні пікселі знаходяться на останній стадії обробки, попередні можуть бути на передостанній стадії. Таким чином, кількість пікселів, оброблених за один такт (цикл), будуть залежати від кількості етапів конвеєра. Для досягнення максимальної пропускної спроможності окремі ступені конвеєра повинні мати однаковий час. Ширина смуги системи визначається як спільне використання пам'яті системи за один такт. У вбудованих системах обробки зображень кадри будуть зберігатися в буфері. Пікселі будуть доступні з цієї основної пам'яті, будуть оброблені і записані назад в пам'ять. Така система буде мати часті звернення до пам'яті для отримання одного пікселя і запису обробленого пікселя. Частий доступ до пам'яті створить вузькі місця в пікселях і збільшить навантаження на систему. Велика ширина смуги системи негативно позначиться на ефективності системи. Ширина смуги системи може бути оптимізована з

використанням кеш-пам'яті. Пікселі можуть бути обрані у вигляді пакетів з чотирьох або восьми і можуть бути збережені в кеш-пам'яті. Доступ до кеш- пам'яті швидше, ніж до основної пам'яті, такий як блок пам'яті.

# Послідовна і паралельна обробка зображень

Платформи послідовної обробки зображень засновані на послідовній комп'ютерній архітектурі. Такий послідовний процесор працює шляхом послідовного вилучення інструкцій і декодування їх в арифметичні і логічні операції. Це завдання буде виконано ALU (арифметична логічна одиниця). Інша частина CPU (центральний процесор) постачає ALU необхідними даними. Компілятор скомпілює алгоритм в послідовність інструкцій а ці інструкції будуть декодувати ЦП (центральним процесором) і АЛУ протягом кожного тактового циклу. Отже, основна операція ЦП полягає в тому, щоб витягти функцію з пам'яті, декодувати, щоб в подальшому визначити, яку операцію і функцію виконати.

Алгоритм обробки зображень складається з послідовності операцій обробки зображень. Це форма часового паралелізму. Цю паралельну природу можна використовувати з багатопроцесорною архітектурою і конвеєрної технологією. Дані проходять через кожен процесор, поки алгоритм продовжується. Іншими словами, кожен процесор застосовує свої операції до даних і передає їх на наступний етап. Значне прискорення може бути досягнуто, якщо процесорам не потрібно чекати введення від будь-яких інших етапів алгоритму. Якщо алгоритм має значну кількість послідовних операцій, один процесор повинен буде чекати результату іншого. Це спричинить за собою додаткові витрати на зв'язок. Однак пропускна здатність системи може покращитися, оскільки перший процесор обробляє дані, в той час як частина даних обробляється у другому процесорі. Дані будуть відправлені на пристрій виведення до завершення всіх операцій, щоб зменшити затримку системи.

Алгоритм може бути реалізований в масивно паралельній архітектурі, тільки якщо він має значну кількість послідовностей, які можуть виконуватися паралельно. Послідовності алгоритмів, які мають взаємозалежності, створять вузькі місця і додаткові витрати зв'язку. Таким чином, важливо перетворити всі можливі послідовності послідовних алгоритмів в паралельні. Як тільки паралельні послідовності ідентифіковані, кожна послідовність буде відображена в апаратної архітектурі схеми. Цей процес буде повторюватися для всіх ідентифікованих паралельних процесів. На наступному кроці ці зіставлені підсхеми будуть інтегровані, і будуть досліджені різні витрати зв'язку між кожною підсхемою. Агрегація всіх окремих ланцюгів в сукупності дасть бажаний результат.

Після встановлення загальної архітектури для паралельних послідовностей алгоритму, наступним етапом буде розробка послідовної частини. Канал зв'язку і протокол між паралельними і послідовними процесорами будуть скасовані. Інтеграція між паралельними і послідовними процесорами має вирішальне значення. Будуть приховані взаємодії між двома модулями, що не були ідентифіковані під час початкових етапів. Ці недоліки повинні бути зменшені ітеративно. Ретельне інтеграційне тестування необхідно в кожній ітерації.

У вбудованому програмному забезпеченні обробки зображень велика ширина смуги системи призведе до збільшення затримки системи і, отже, буде потрібно більше часу для виконання завдання. Ширина смуги системи може бути мінімізована, якщо ресурси високошвидкісної пам'яті в FPGA досить високі. Великий об'єм кеш-пам'яті значно скоротить частий доступ до основної пам'яті і збереже ширину смуги системи. Дизайн з кеш-пам'яттю для зберігання даних і результатів дуже ефективний.

Ширина смуги системи і швидкість роботи обернено пропорційні один одному. Більш висока пропускна здатність призведе до збільшення системної затримки, а система буде працювати повільно. Системна швидкість може бути поліпшена шляхом збільшення тактової частоти. Але

тактова частота повинна бути синхронізована з вхідним потоком даних, в іншому випадку інші дані будуть втрачені. Крім того, тактова частота повинна бути достатньою для завершення операції пікселя на одному етапі багатоступінчастого конвеєрного проектування.

# Архітектура FPGA

Архітектура системи FPGA визначає загальну структуру системи, яка буде реалізована. Вибір цієї архітектури залежить від пропорції послідовностей в алгоритмі, які можуть виконуватися паралельно. На етапі вибору архітектури проекту повинні бути вказані доступні архітектури систем FPGA.

У цій архітектурі всі прикладні програми реалізовані з використанням паралельного процесора на основі FPGA. Кілька апаратних блоків будуть виконувати різні послідовності алгоритму одночасно. Максимальна вигода може бути залучена, якщо значна частина алгоритму може виконуватися паралельно. Така архітектура забезпечує максимальну швидкість роботи і більш високу ефективність. Загалом, такі архітектури відомі як потужно паралельна архітектура. Цей проект реалізований в автономній масивно паралельної архітектурі.

У складних алгоритмах обробки зображень будуть існувати послідовності, які повинні будуть чекати виконання попередніх або майбутніх послідовностей. Якщо такі послідовні операції реалізуються з використанням паралельного процесора, це погіршить ситуацію через комунікаційні витрати. У цих сценаріях може використовуватися архітектура співпроцесора. Основна частина алгоритму буде виконуватися паралельним процесором, а послідовна частина буде реалізована в послідовному. Ця архітектура аналогічна архітектурі співпроцесора, за винятком того факту, що завантаження алгоритму буде в рівній мірі розподілена між послідовним і

паралельним процесорами. Цю архітектуру можна використовувати в ситуаціях, коли тільки самі внутрішні цикли послідовного алгоритму можуть виконуватися паралельно. Належний протокол зв'язку повинен бути визначений між двома процесорами. Паралельний процесор буде процесором на основі ПЛІС, і, як правило, послідовний процесор буде заснований на мікроконтролері.

Цю архітектуру можна використовувати, коли більшість послідовностей алгоритмів мають взаємозалежності. В результаті більшість послідовностей має виконуватися послідовно. У конструкціях такого типу послідовний процесор буде виконувати основну частину алгоритму, а деякі операції, які можуть виконуватися паралельно, будуть передаватися паралельному процесору. Витрати на зв'язок між паралельними і послідовними процесорами повинні бути правильно враховані. Кращим прикладом такої архітектури є процесор з плаваючою комою, асоційований з високопродуктивними процесорами. Комплексні арифметичні операції з плаваючою комою будуть виконуватися виділеним процесором, а результат буде переданий в кеш-пам'ять, яка буде розподілена між обома процесорами (послідовним і паралельним).

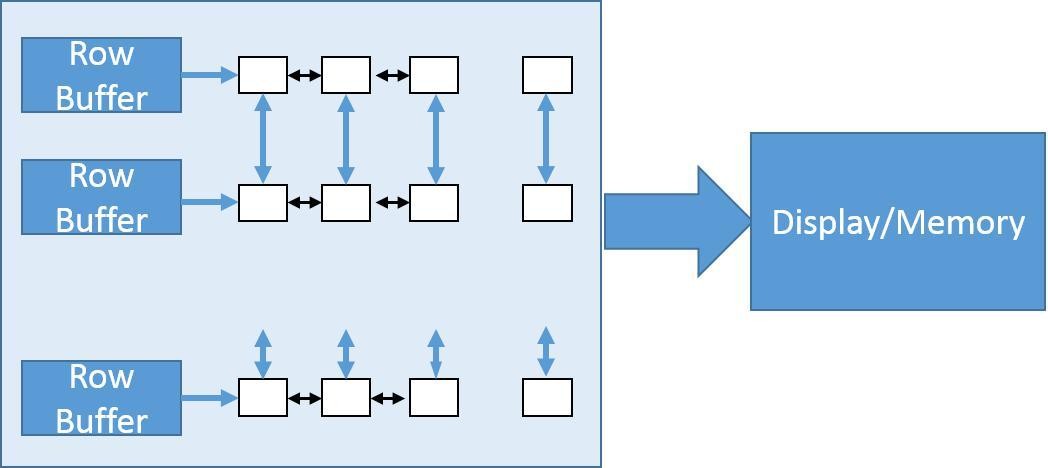
Обчислювальна архітектура визначає, як реалізуються обчислювальні аспекти алгоритму. Іншими словами, він описує, як кожна операція обробки зображення виконується всередині FPGA. Вибір обчислювальних архітектур залежить від програми.

У будь-якій прикладній програмі вбудованого зору дані записуються за допомогою цифрової камери. Одним із способів обробки вхідних кадрів є їх збереження у буфері кадрів, а потім обробка кожного кадру паралельно. Але в цьому випадку кількість звернень до пам'яті буде великим і, отже, час відгуку збільшиться. Якщо пікселі не мають залежності від попереднього, вони можуть бути оброблені «на льоту». Іншими словами, пікселі можуть бути оброблені під час їх читання з камери. Ця обробка називається

потокової обробкою. Щоб зменшити час відгуку, необхідно виконати максимальну обробку при потокової передачі зображення.

Одним з основних недоліків потокової обробки є фіксована тактова частота. Тактова частота обмежена вхідний частотою кадрів. Якщо тактова частота менше вхідної частоти кадрів, кілька кадрів будуть втрачені. Системна затримка буде мінімальною при обробці потоку.

Систолічний масив - це одновимірний або двомірний масив процесорів, в якому дані будуть оброблятися під час потокової передачі, і передаватися між сусідніми процесорами. Систолічний масив відрізняється від потокової обробки напрямком потоку даних. У більш пізньому випадку він є односпрямованим, тоді як в першому, дані можуть переміщатися в обох напрямках. Таким чином, якщо попередній піксель має яку-небудь залежність від поточного пікселя, він буде повернутий до попереднього етапу процесора. У порівнянні з обробкою потоку витрати на зв'язок більші через зворотній зв'язок. Операції будуть виконуватися в кожному тактовому циклі. Цей тип архітектури в основному використовується в прикладних програмах виявлення та відстеження об'єктів (рис. 1.17).



буфер рядка

буфер рядка

Дисплей / пам'ять

буфер рядка

Рисунок 1.17 - Обчислювальна архітектура з використанням систолічних масивів

У методі обробки з довільним доступом пікселі можуть бути доступні в будь-якому місці кадру. Для досягнення цього необхідно реалізувати кадровий буфер. Вступні кадри будуть накопичуватися в буфері, і будь-які пікселі з цього буфера можуть оброблятися випадковим чином. Цей тип архітектури корисний, коли алгоритм має значну частину послідовностей, які не можуть виконуватися паралельно. Цей вид обробки еквівалентний послідовній обробці даних. Одним з додаткових переваг є те, що не буде будь-яких жорстких обмежень, як у випадку потокової обробки. Але затримка системи буде більше, через велику кількість доступу до пам'яті.

Паралелізм даних може бути досягнутий шляхом поділу різних частин зображення на окремі процесори. Для цього необхідно кілька копій апаратного блоку відповідних кожній частині зображення. Кожна частина зображення буде знаходитися в локальному буфері відповідного апаратного блоку.

# Методи зіставлення ПЛІС

Методи зіставлення ПЛІС описують, наскільки ефективно кожна операція обробки зображення зіставляється з ресурсами ПЛІС. Іншими словами, це стандартні методи, прийняті для досягнення бажаної продуктивності системи з точки зору швидкості, пам'яті і ресурсів. На етапі вибору архітектури проекту буде наведено короткий список докладних переліків обмежень і методів підвищення ефективності системи. Необхідно враховувати три основних обмеження при відображенні послідовностей послідовних алгоритмів в FPGA, а саме:

* тимчасові обмеження;
* обмеження пропускної здатності;
* ресурсні обмеження.

Використання ресурсів залежить від ефективності проекту. Це необхідно враховувати на рівні логічного проектування.

У прикладних програмах реального часу швидкість вхідних даних є одним з основних обмежень. Обробка даних повинна виконуватися з частотою пікселів або швидше, щоб запобігти втраті даних. Низькорівнева можливість відразу пропускати цілий потік даних, не чекаючи завершення всього обчислення (pipelining), є одним з методів, використовуваних для подолання цих труднощів.

Pipelining розбиває операцію на більш дрібні етапи і завершує операцію в кілька етапів. В результаті затримка поширення на одному етапі або затримка в одному тактовому циклі буде зменшена. Оскільки за один тактовий цикл виконується тільки частина операції, загальна тактова частота може бути збільшена. Таким чином, для завершення однієї операції буде потрібно більше одного циклу. Якщо це обладнання дублюється, кілька пікселів можуть бути оброблені одночасно. Кількість пікселів, які можуть бути оброблені одночасно, залежить від загальної кількості етапів pipelining.

В якості прикладу розглянемо наведене нижче рівняння:

*y = ax2 + bx + с = ( ax + b) × · x + c*

Вищевказане рівняння може бути реалізовано з використанням одно, дво- і чотириступінчастих pipelining. Але оптимальна затримка може бути досягнута за допомогою двоступеневої обробки, оскільки це покращує пропускну здатність системи.

На рисунку 1.18 показано двоступеневий конвеєр як найбільш ефективний. Це пов'язано з тим, що операції додавання і множення рівномірно розподіляються між двома етапами. У разі чотириступінчастого конвеєра на першій і третій сходинках виконуються операції множення, а на другій і четвертій щаблях - операції додавання. Операція множення вимагає більше часу, ніж операція додавання. Іншими словами, розподіл термінів не є рівномірним. Це створить дисбаланс і збільшить затримку системи. Цей ефект може бути пом'якшений шляхом повторної синхронізації

послідовностей. У цьому методі частина операції множення з першого етапу буде завершена на другому етапі. Тому правильний вибір кількості ступенів pipelining є найбільш важливим в конструкціях ПЛІС.

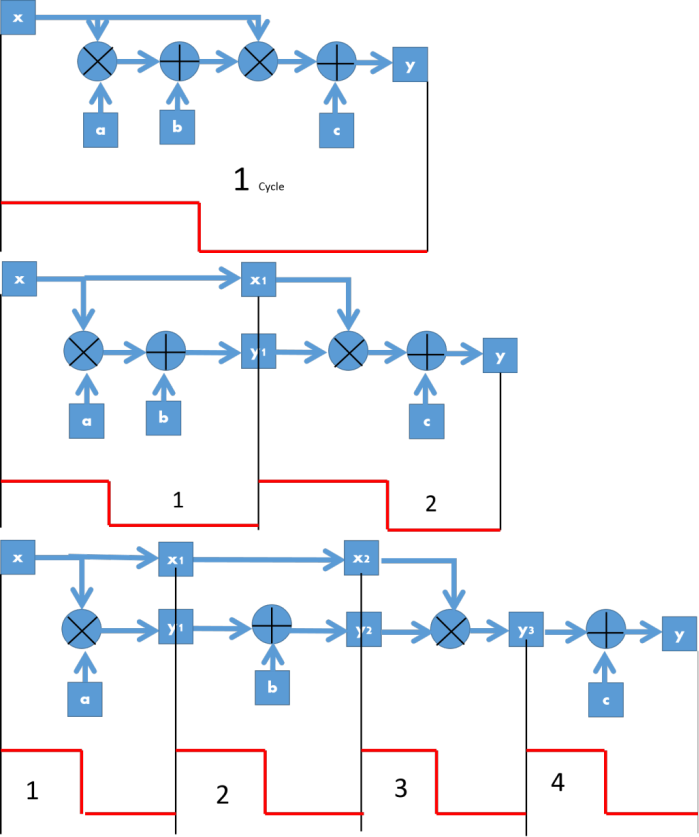


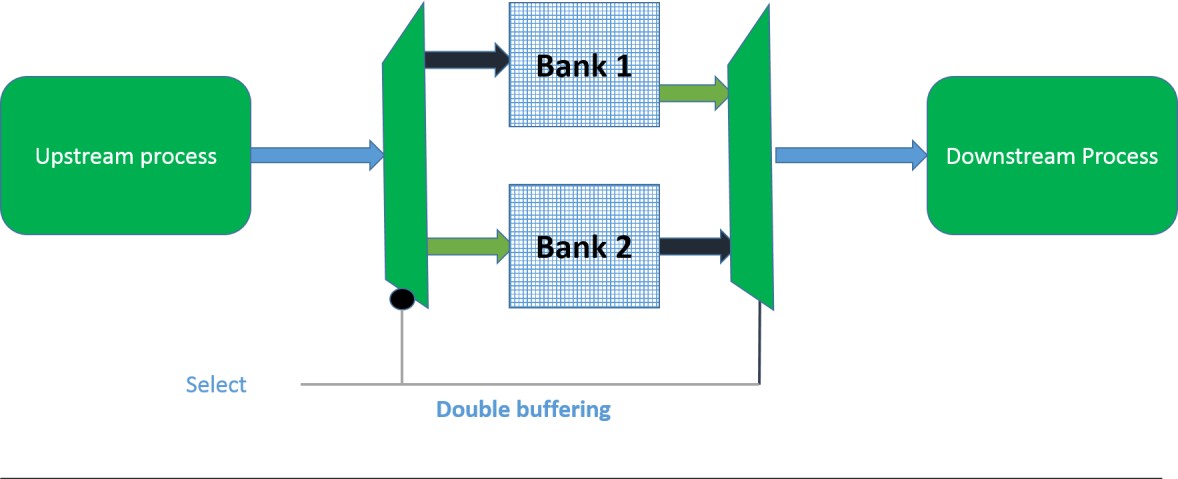
Рисунок 1.18 - Приклад pipeling

Якщо все зовнішні події, керуючі системою, слідують певним шаблоном і, якщо час обробки або затримки рівномірно розподілені між операціями, то можна використовувати глобальне планування. Це може бути досягнуто за допомогою глобального лічильника подій, який синхронізує і планує події. Потім плануються різні операції, визначаючи коли кожної операції потрібні свої дані і зіставляючи рахунок.

Вхідні кадри повинні бути частково або повністю збережені в більшості операцій обробки зображень. Сучасні системи FPGA мають великі ресурси пам'яті поза чіпа. Але неефективне використання цих ресурсів призведе до поганої системи прикладних програм. Якщо системна зовнішня

пам'ять використовується часто, це вплине на час відгуку системи, а також на затримку системи.

Подвійна буферизація використовується в відображенні ПЛІС для зменшення кількості операцій з пам'яттю. Вона використовується між двома послідовними операціями обробки зображень, щоб уникнути уповільнення системи при використанні загальної пам'яті. Цей метод в основному використовується з pipeling або з обробкою довільного доступу. Дані будуть оброблятися «на льоту» і завантажуватися в буфер для обробки довільного доступу. Ця методика точно вписується між першою обробкою і обробкою довільного доступу. Вона використовує два підключених банку пам'яті, як показано на рисунку 1.19. Процес висхідного потоку захоплює дані з введення і записує в один з банків пам'яті. Подальший процес зчитує дані паралельно з іншого банку і відображає їх. Коли кадр буде завершено, роль двох банків буде змінена на протилежну так, що дані, тільки що завантажені вихідним потоком, тепер будуть доступні для подальшого процесу. Отже, один період кадру буде додано до затримки системи.



вибір

подвійна буферизація

Банк даних 2

процес накопичення

процес виділення

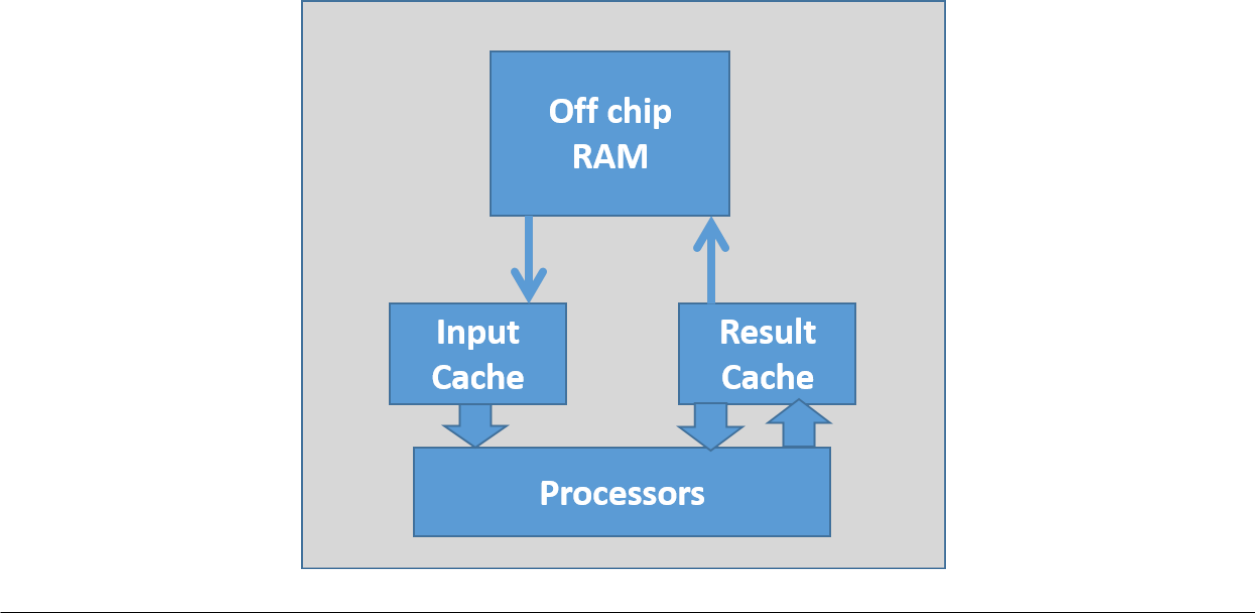
Банк даних 1

Рисунок 1.19 - Подвійна буферизація

Кеш-пам'ять - це високошвидкісна пам'ять, розташована поруч з процесором, в якій зберігаються найбільш часто використовувані дані або результати попередніх операцій. Вона діє як високошвидкісний буфер.

Перевага такої пам'яті в тому, що доступ до даних буде значно швидший в порівнянні з пам'яттю поза чіпа, і, отже, час відгуку буде значно покращений. На FPGA кеш можна використовувати для збільшення затримки системи. У більшості проектів кожен елемент обробки буде супроводжуватися кеш-пам'яттю. Отже, ширина смуги системи буде значно зменшена. У деяких конфігураціях буде присутнім кеш даних і кеш результатів для зберігання звичайних даних і обчислених результатів. У деяких прикладних програмах кеш розміщується поруч з процесором і

основною пам'яттю, з доступом до кешу, а не до пам'яті (рис. 1.20).



процесор

результатів кеш

вхідний кеш

зовнішня оперативна пам'ять

Рисунок 1.20 - Кешування як інтерфейс до пам'яті

# Проблеми FPGA

У перші дні ПЛІС розглядалися як логічні чіпи, що склеюють, , які використовуються для з'єднання компонентів в складні системи. ПЛІС все більше стали розглядатися як цілісні системи самі по собі. На додаток до розвитку технологій, ряд інших міркувань прискорили це. Наприклад, поява FPGA як платформи DSP прискорилася завдяки застосуванню методів розподіленої арифметики (DA). DA дозволила виконувати ефективні

реалізації FPGA з використанням конструкцій LUT суматорів блоків FPGA і дозволив отримати значний виграш в продуктивності для деяких перетворень DSP, таких як фільтрація з фіксованим коефіцієнтом і функції перетворення, такі як швидке перетворення Фур'є (FFT). Хоча ці методи продемонстрували, що ПЛІС можуть створювати високоефективні рішення для прикладних програм DSP, концепція витіснення останнього аспекту продуктивності з апаратного забезпечення ПЛІС і, що більш важливо, витрачаючи кілька людино-місяців на створення таких інноваційних конструкцій, стала неприйнятною. Збільшення складності через еволюції технологій означало, що в області застосування сучасної технології FPGA зростає розрив і здатність проектувальника ефективно розробляти рішення з використанням доступних в даний час інструментів. Проблема не настільки серйозна в реалізації FPGA, так як проектувальнику не доводиться мати справу з проблемами проектування в субмікрометровій області. Проте, існує ряд ключових питань, які включають в себе розуміння того, як відобразити функціональність DSP в FPGA. Деякі з аспектів є відносно основними в таких областях як множення, додавання і затримки, які відображаються на бортових множниках, суматорах і регістрах і компонентах RAM відповідно. Однак функції з плаваючою комою і фіксованою комою, оптимізація довжини слова, функція вартості алгоритмічного перетворення для ПЛІС і вплив затримки маршрутизації є проблемами, які необхідно враховувати на системному рівні, і на цьому рівні може бути набагато складніше працювати.

**Мови програмування.** В даний час мови написання апаратного забезпечення, такі як VHDL і Verilog, і їх відповідні потоки синтезу, добре відомі. Проте, користувачі тепер дивляться на FPGA з недавнім збільшенням складності, що призводить до інтеграції як фіксованих, так і програмованих ядер мікропроцесорів в єдину систему, і шукають конструктивні уявлення, які більш чітко уявляють опис системи. Зараз приділяється підвищена увага використанню C ++ в якості мови проектування, але існують і інші уявлення, такі як методи, засновані на моделях обчислень як синхронний потік даних.

У зв'язку з відсутністю швидких і надійних рішень для мови проектування і завдань синтезу, ринок IP в реалізації SoC з'явився, щоб заповнити пробіл і дозволити швидке прототипування обладнання. М'які ядра особливо привабливі, тому що функціональні можливості дизайну можуть бути отримані з використанням HDL і ефективно перетворені в обрану технологію FPGA за допомогою традиційних інструментів синтезу. Крім того, були розроблені процесорні ядра, які дозволяють додавати спеціальні функції. Залучення цих підходів полягають в тому, що вони дозволяють швидко створювати специфічні функціональні можливості програми, оскільки платформа в значній мірі виправлено.

**Розрахунковий потік.** Більшість можливостей процесу проектування заснована на розробці функціональних можливостей ПЛІС, в основному для складних функцій. В даний час реальність така, що технологія FPGA розвивається з такою швидкістю, що системи, які містять FPGA і процесори, починають перетворюватися в платформу SoC, оскільки вони мають вбудовані процесори, високошвидкісний зв'язок і програмований ресурс, і це можна розглядати як цілісну систему. Традиційно програмні потоки були більш просунутими для процесорів і навіть для кількох процесорів, оскільки архітектура фіксована. Хоча інструменти були розроблені для апаратних платформ, таких як FPGA, існує певна потреба в програмному забезпеченні для потоків для гетерогенних платформ, тобто тих, які включають як процесори, так і FPGA.

# Висновки

Вибір методу відображення FPGA залежить від програми. Безліч доступних методів має бути вибрано і використано відповідним чином у всіх проектах. Наявність у виробі FPGA означає феноменальну гнучкість логіки цього виробу (в більшості випадків це заводська можливість корекції логіки роботи пристрою без проведення будь-яких електромонтажних операцій, що

дозволяє врахувати потреби споживачів). Ресурси сучасних FPGA (навіть молодших в обраному сімействі) вже дозволяють реалізовувати складні алгоритми, в тому числі, алгоритми цифрової обробки сигналів.

Можливість створювати на основі FPGA багатоканальні системи паралельної обробки даних з гарантованою логічною незалежністю процесів обробки даних (в поєднанні з високою надійністю самих FPGA) створює серйозну альтернативу сигнальним процесорам (DSP) і ARM в високонадійних завданнях управління і контролю. Разом з тим, в світі явно відчувається і зворотна тенденція - реалізація засобами FPGA процесорів і контролерів з масштабованої архітектурою.

DSP є клас апаратних пристроїв, які знаходяться десь між ASIC і ПК з точки зору продуктивності і складності конструкції. Вони можуть бути запрограмовані або за допомогою асемблерного коду, або на мові програмування, що є одним з явних переваг платформи. Однак алгоритми, розроблені для DSP, не можуть бути високопаралельними без використання декількох DSP. Продуктивність алгоритму, безумовно, вище, ніж на ПК, але в деяких випадках системи ASIC або FPGA є єдиним вибором для проектування. Недавні досягнення в технології DSP привели до дуже швидким реалізацій алгоритму. Хоча переваги ASIC і FPGA все ще застосовні, це нове покоління DSP змусило деяких інженерів переглянути розробку FPGA. Проте, у міру появи нових DSP на ринку з'являються і нові FPGA, і очікується, що дві архітектури матимуть однаково збільшується продуктивність для кожного нового покоління процесорів.

На підставі мікросхем цифрового сигнального процесора (DSP) був розроблений універсальний ПЗС-контролер (який може працювати з ПЗС- матрицею), для реалізації таких важливих операцій, як зчитування CCD, взаємодія з хостом і корельована подвійна вибірка для усунення скидання шуму. Після вивчення функціональних можливостей плати DSP, плати зміщення і тактового генератора, інтерфейсу хосту і збору даних з коду DSP виводиться алгоритм алгоритму кінцевого автомата.

Прилад із зарядним зв'язком (ПЗС) був розроблений в основному як компактний датчик зображення для споживчого та промислового ринків, але в даний час також є видатним видимим і ультрафіолетовим датчиком довжини хвилі зображення у багатьох областях наукових досліджень включаючи космічну науку і дистанційне зондування Землі і планети. ПЗС- матриця буде прагнути захоплювати максимальну кількість пікселів в фокальній площині, ефективно реєструвати фотони в самому широкому спектральному діапазоні, і динамічний сигнал в діапазоні при збереженні мінімально можливого шуму зчитування. Щодо недавня поява технології сенсорних зображень на основі оксиду металу і напівпровідників (CMOS), можливо, є найбільш важливим досягненням в області твердотільної візуалізації з моменту винаходу ПЗС. Технологія CMOS дозволяє інтегрувати одиночну кремнієву мікросхему з великою кількістю пікселів фотодіодів разом з усіма допоміжними елементами електроніки, необхідних для адресації масиву і оцифровки результуючого аналогового відео сигналу. У порівнянні з CCD, CMOS обіцяє більш компактну, меншу масу, меншу потужність і потенційно більш радіаційно-стійке камера.

# РОЗДІЛ ДРУГИЙ

**ТЕОРЕТИЧНИЙ АНАЛІЗ ФІЛЬТРАЦІЇ ЗОБРАЖЕННЯ**

* 1. **Загальні положення**

Спочатку FPGA розглядалися як конкурент CPLD (технологія програмованих логічних пристроїв зі складністю), які спочатку використовувалися в якості сполучною логіки для складних цифрових схем. ПЛІС використовувалися для особливих застосувань, де обсяг виробництва невеликий і використання ASIC (інтегральна схема, чиєю спеціалізацією є пошук рішення для виконання одного завдання) було б занадто дорогим. Коли їх розмір, швидкість і характеристики збільшилися, ПЛІС перемістилися в багато додаткові області, включаючи ті, які зазвичай покриваються мікроконтролерами або дискретними ASIC. Постачальники почали використовувати функції, які ідеально підходять для додатків DSP, з вбудованою пам'яттю і спеціальними множниками. У порівнянні з процесорами DSP, FPGA часто містять відносно велику кількість ресурсів DSP. В цьому випадку FPGA можуть запропонувати значно більшу ширину смуги множника, оскільки множення можуть виконуватися паралельно.

Алгоритми обробки сигналів, раніше реалізовані на DSP, тепер можуть бути легко реалізовані на FPGA. Через паралельний характер FPGA (і ASIC) їх використання для виконання алгоритмів обробки сигналів може бути перевагою, якщо все зроблено правильно. Замість традиційного універсального DSP, FPGA може використовуватися для розробки індивідуального рішення конкретної проблеми, а інженер-конструктор може уникнути оплати за невикористовувані функції. У той же час цифрову систему можна модернізувати або поліпшити, не використовуючи нічого, крім оновлення програмного забезпечення.

В останні десятиліття цифрова обробка сигналів набула великої популярності в цифровому світі завдяки її здатності ефективно перетворювати цифрові і/або аналогові сигнали за рахунок зниження витрат, пов'язаних з розробкою і реалізацією. Тому вона стала конкурентом аналогової обробки сигналів, замінивши велику кількість прикладних програм. Цифрова обробка сигналів може використовуватися в різних прикладних програмах, деякі з яких - обробка швидкості, зображень та аудіо. Хоча DSP могли бути реалізовані в процесорах цифрових сигналів, віхою в реалізації цифрової обробки сигналів стала поява в 1980-х роках програмованих логічних інтегральних схем (FPGA/ПЛІС). ПЛІС - це інтегральна схема, яку можна запрограмувати після виготовлення, щоб вона функціонувала як будь-яка цифрова схема, яка визначається розробником. Основна відмінність між ПЛІС і спеціалізованої інтегральної схемою (ASIC) полягає в тому, що ASIC може використовуватися тільки для певної програми. Основні будівельні блоки в сучасних FPGA складаються з конфігурованих логічних блоків (CLB), з'єднань, блоків введення / виводу (IOB) і вбудованих блоків, таких як блоки DSP.

# Цифрова обробка сигналів

Фільтри з кінцевою імпульсною характеристикою (FIR) є одним з найбільш часто використовуваних алгоритмів цифрової обробки сигналів, використовуваних в даний час, коли FPGA - це пристрій, що використовується для його реалізації. Безперервний розвиток пристроїв ПЛІС, шляхом додавання виділених блоків, викликало необхідність вивчення переваг, пропонованих різними родинами ПЛІС. В даній дипломній роботі, вивчаються особливості, пропоновані FPGA для FIR-фільтрів, і представлена модель використання ресурсів. Використовуваний метод складається з декількох етапів, включаючи читання, класифікацію ознак і генерацію коефіцієнтів. Результати показують, що FPGA мають загальні

характеристики, але також мають певні відмінності у функціях, а також у використанні ресурсів. Було показано, що при роботі з ПЛІС існує невірне уявлення про КІХ-фільтри в порівнянні з ASIC.

У 1985 році Xilinx випустила перший комерційний FPGA-пристрій XC2064. Xilinx - відома компанія в індустрії FPGA. Пристрій XC2064 має дві таблиці пошуку з трьома входами (LUT) і мінімальну кількість логічних елементів. Таблиця перетворення (LUT) - це інформаційна таблиця, за допомогою якої можна обчислити «різницю між вихідним матеріалом (відео) і бажаним результатом».

Згодом був доданий ряд функцій для виконання різних завдань. Постійне зростання вимог клієнтів привів до розробки FPGA, в якій були додані виділені блоки, такі як блоки DSP, для виконання певних функцій, включаючи FIR-фільтр.

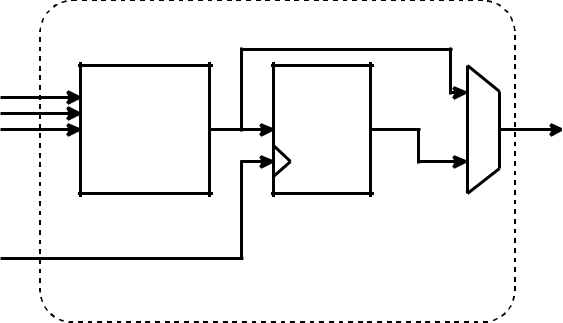
В даний час сучасні ПЛІС мають мільйони логічних елементів і стали набагато складнішими, ніж його перший запуск в 1985 році. Триваючий розвиток пристрої ПЛІС шляхом додавання виділених блоків викликало необхідність вивчення переваг, пропонованих різними ПЛІС сімействами, а потім визначити наслідки використання певної ПЛІС замість іншої. Тому метою даної дипломної роботи є поглиблене вивчення функцій, пропонованих різними родинами ПЛІС, які спеціально розроблені для реалізації FIR-фільтрів і, таким чином, представляють модель витрат у формі використання ресурсів, щоб допомогти проектувальнику вибрати відповідний проект при реалізації FIR-фільтра після врахування різних можливих сценаріїв.

Існує кілька популярних методів реалізації, які використовуються в сучасній промисловості для відображення FIR-фільтра на FPGA. Існують наступні методи реалізації:

* традиційний метод множення і складання;
* розподілена арифметика з відносним помножувачем;
* складання і зрушення, тобто перетворення помножувачів в суматори.

Дослідження і аналіз різних структур КІХ-фільтрів проводяться для підвищення ефективності фільтрів на ПЛІС. Була реалізована і проаналізована архітектура швидкої FIR, конвеєрної і паралельної обробки. Конвеєрні структури у вигляді вставки елементів затримки після суматорів не потребують збільшення площі. Паралельна обробка призводить до активнішого використання ресурсів завдяки збільшеній частоті дискретизації. Можна зробити висновок про те, що конвеєрна структура є найкращою альтернативою, коли мова йде про скорочення ресурсів.

Логічний блок (CLB), що конфігурується, використовується для реалізації логічних функцій і математичних операцій. Усередині кожного CLB знаходяться так звані настроювані логічні елементи (CLE), які, в свою чергу, складаються з таблиці перетворення (LUT), D-тригерів і мультиплексорів 2 в 1 (рис. 2.1). Метою LUT є виконання математичних операцій, таких як додавання.



LUT

суматор

а/b/с

Рисунок 2.1 - Спрощена внутрішня структура типового CLE

Блок LUT і його три входи показані на рис. 2.2. Три входу LUT керують вибором вхідного біта 1/0 мультиплексорів.

a/b/c



1/0

1/0

1/0

1/0

1/0

1/0

1/0

1/0

Рисунок 2.2 - Спрощена внутрішня структура LUT

Ядро FPGA складається з тисяч копій CLE, пов'язаних один з одним. LUT складається з серії каскадних мультиплексорів, де входи LUT використовуються в якості ліній вибору, а входи в мультиплексори - це 1- бітна пам'ять SRAM, яка встановлена в 0 або 1.

Як показано на рис. 2.1, велика частина архітектури FPGA використовується вертикально і горизонтально розподіленими з'єднаннями. Ці з'єднання складаються з розподільних коробок і сполучних коробок, щоб отримати бажане з'єднання. Кожен CLB оточений групою сполучних блоків, які, в свою чергу, з'єднані один з одним через групу розподільних блоків (рис. 2.3). SB відноситься до розподільних коробок, а CB відноситься до з'єднувальних коробок. Ці сполучені блоки використовуються для з'єднання різних CLB один з одним і, крім того, забезпечують з'єднання як з блоками вводу/виводу, так і з вбудованими блоками.

**SB**

**CB**

**SB**

**CB**

**CB**

**CLB**

**SB**

**CB**

**SB**

Рисунок 2.3 - Спрощена мережа з'єднань

Хоча логічний блок, що конфігурується, здатний виконувати арифметичні операції і зберігати дані, він буде занадто повільним і використовує величезну кількість ресурсів, оскільки CLB призначений для того, щоб бути відносно загальним для виконання різних типів функцій. Оскільки попит на ПЛІС зростає, сучасні ПЛІС пропонують вбудовані блоки для виконання спеціальних функцій, деякі з яких - блоки пам'яті та блоки DSP. Впровадження цих вбудованих блоків явно зменшує використання площі, енергоспоживання і, як наслідок, призводить до підвищення продуктивності. Хоча внутрішня структура блоків DSP і пов'язані з ними деталі різняться в залежності від виробника і версії пристрою, загальна архітектура більшості блоків DSP виглядає однаково.

Блок DSP складається з суматора і множника, за яким слідує акумулятор. Метою впровадження блоків DSP є підвищення продуктивності цих арифметичних операцій. Крім того, в кожному блоці DSP є певні з’єднання, які можна використовувати для підключення декількох блоків DSP один до одного, що, в свою чергу, може бути використано для реалізації ефективного FIR-фільтра.

# FIR-фільтри

Цифровий фільтр використовується для зміни характеристик сигналу у часовій і/або частотний метод. Цифровий сигнал отримують шляхом дискретизації безперервного сигналу в різних часових інтервалах, після чого сигнал представляється у вигляді послідовності дискретних значень, на відміну від аналогового сигналу, який є безперервним і представлений як функція часу.

Алгоритми цифрового КІХ-фільтра можуть бути математично описані за допомогою графіка потоку сигналів, які представлені у вигляді блок-схем (рис. 2.4):

х [n] y [n]



**ak**

**ak**

х [n-k ]

Рисунок 2.4 - Приклад графіка потоку

Основними компонентами, використовуваними в цифрових фільтрах (рис. 2.4) є множники, суматори і елементи затримки. Хоча різницеві рівняння використовуються для опису взаємозв'язку між вхідний і вихідний послідовностями і, отже, є зовнішньою поведінкою фільтра, графіки потоку сигналів є більш докладними і використовуються для точного обчислення вихідних значень y [n] і внутрішніх значень в фільтр.

Лінійно-інваріантний фільтр (LTI) є одним з найбільш часто використовуваних цифрових фільтрів. LTI-фільтр можна описати його імпульсної характеристикою h [n]. Імпульсна характеристика h [n] - це вихід,

отриманий шляхом застосування імпульсу в якості вхідної послідовності. Вихідні дані y [n] отримують за допомогою процесу згортки, який згортає імпульсну характеристику і вхідну послідовність в тимчасовій області і може бути математично представлений як y [n] = x [n] \* h [n].

Фільтри LTI зазвичай діляться на два типи: з кінцевою імпульсною характеристикою (FIR / КІХ) і з нескінченною імпульсною характеристикою (IIR/БІХ). Кінцева імпульсна характеристика означає, що характеристика стає нульовою, тобто має обмеженість за часом. Такі фільтри називають нерекурсівними, тому що вони не мають зворотного зв'язку. Навпаки, нескінченна імпульсна характеристика має нескінченну довжину в тимчасовій області. Такі фільтри називають рекурсивні і вони зі зворотним зв'язком.

Існують різні способи побудови КІХ-фільтра. Два загальних з них класифікуються як:

- структура прямої форми;

* транспонована структура прямої форми.

Структура прямої форми - це класична реалізація рівняння FIR- фільтра. Вхідні відліки затримуються за допомогою елементів затримки в формі D-тригерів. Структура прямої форми залежить від так званого ланцюгового суматора, в якому вхідні дані частково обчислюються і виводяться через ланцюжок невеликих суматорів. Метод ланцюгового суматора призводить до значного скорочення використання площі.

Критичний шлях КІХ-фільтра - найдовший шлях логіки між двома регістрами. Затримка - це кількість вставлених конвеєрних регістрів між входом і виходом. На рис. 2.5 показана структура прямого виду КІХ-фільтра. Транспонована пряма форма (рис. 2.6) виходить з прямої форми після застосування теореми про транспозиції. Транспонування досягається шляхом заміни входів на виходи і заміни виходів на входи, де входи безпосередньо пов'язані з множниками. Критичний шлях транспонованою структури значно

зменшений в порівнянні з прямою формою.



T

Рисунок 2.5 - Пряма структура форми FIR-фільтра.



**h[0]**



**h[N ­ 1]**

**h[N ­ 2]**

**h[N ­ 3]**

T

T

T





Рисунок 2.6 - Транспонована структура прямої форми КІХ-фільтра

# Оптимізація FIR фільтра

Існує кілька методів, які практично використовуються для КІХ- фільтра, щоб поліпшити реалізацію фільтра за рахунок зменшення використання площі і часу.

КІХ-фільтр має здатність отримувати лінійну фазу, що, в свою чергу, означає, що імпульсна характеристика h (n) являє симетрію, де h (n) = h (N n), де N - порядок фільтра, а x [n] - вхідні вибірки. На рис. 2.7 показана імпульсна характеристика КІХ-фільтра з 9 коефіцієнтами, на рис. 2.8 та 2.9 показано представлення КІХ-фільтра з непарним числом на графіку потоку сигналів в прямій і транспонованою прямій формі відповідно.

Математичне обгрунтування скорочення ресурсів при використанні симетрії відповідає розподільної властивості множення, як показано в наступному прикладі: (xy)+(xz)=x\*(y+z), в той час як ліва частина рівняння виконує два множення і одне додавання, права частина виконує тільки одне множення і одне додавання.

h[4]



Рисунок 2.7 - Імпульсна характеристика 9-відвідного лінійного фазового FIR-фільтра

Реалізація симетрії імпульсного відгуку на рис. 2.8 може бути показана на рис. 2.9, де вхідний ланцюжок затримки передається вперед і назад для економії ресурсів. Два симетричних коефіцієнта тепер представлені одним множником. Кожен множник представляє два коефіцієнта і тому загальне використання множників зменшується до половини.





h[0]



х[n]

y[n]

Рисунок 2.8 - Структура симетричного КІХ-фільтра в прямій формі

Симетрія імпульсної характеристики на рис. 2.9 реалізована в транспонованій прямій формі, де загальне використання множників зменшено до половини. Кожен множник представляє два коефіцієнта, і тому загальне використання множників зменшується до половини. Пунктирна риса позначає одну таблицю пошуку.



x[n]

**h[N ­ 1]**

**h[N ­ 2]**

T

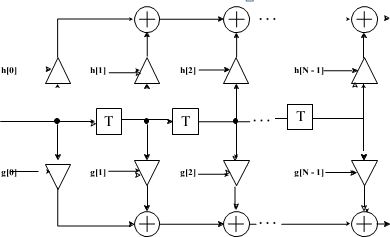
T

T

y[n]

Рисунок 2.9 - Структура симетричного КІХ-фільтра в транспонованій прямій формі

Спільне використання декількох FIR-фільтрів необхідно при роботі з великою кількістю фільтрів через їх здатність зберігати область даних. Спільне використання вхідних даних фільтра FIR прямої форми може бути показано на рис. 2.10, де елемент затримки розподіляється між двома фільтрами.



h[0]

х[n]

g[0]

y[n]

y[n]

исунок 2.10 - Спільне використання вхідних даних двох FIR-фільтрів h і

Рисунок 2.10 - Спільне використання вхідних даних двох FIR-фільтрів h і g

Р в прямій формі

# 2.6 Каскадний інтегрально-гребінчастий фільтр

Каскадні інтегральні фільтри (CIC, також відомий як фільтр Хогенауера) знаходять безліч застосувань у багатьох електронних пристроях, де потрібна ефективна обробка великого коефіцієнта частоти дискретизації. Ці фільтри зазвичай розташовані після сигма-дельта модулятора і мають регулярну структуру. Ці типи фільтрів не вимагають помножувачів і зберігання коефіцієнтів на відміну від звичайних цифрових фільтрів FIR і IIR, оскільки всі коефіцієнти фільтра дорівнюють одиниці. Отже, він може бути ефективно реалізований для роботи на високій швидкості.

Призначення фільтра CIC двояке; по-перше, щоб видалити шум фільтрації, який міг би бути накладено на сигнали основної смуги частот, і, по-друге, перетворити потік даних з високою частотою дискретизації на виході сигма-дельта-модулятора в потік даних з більш низькою частотою дискретизації. Цей процес також відомий як проріджування, що по суті виконує функції усереднення і зниження швидкості одночасно. Каскадні фільтри-інтегратори використовуються для проріджування та інтерполяції по цілочисельному співвідношенню, забезпечуючи ефективну фільтрацію зображень. Фільтр децимації CIC складається з каскадів цифрового інтегратора, що працюють з високою швидкістю вхідних даних, за якими слідують каскади гребінчастих або диференціюють каскадів, які працюють з низькою частотою дискретизації.

Фільтр CIC може використовуватися для децимації або інтерполяції. Фільтр CIC проріджування містить каскад інтеграторів (звані секцією інтеграторів), за якими слід блок понижувальної вибірки (деціматор) і каскад диференціатора (так званий роздільник або секція гребінки). Аналогічно, інтерполяційний фільтр CIC містить каскад блок дискретизації, що диференціює (інтерполятор) і каскад інтеграторів.

У фільтрі CIC секції інтегратора і гребінки мають однакову кількість інтеграторів і диференціаторів. Кожне з'єднання інтегратора і

диференціатора називається ступенем. Кількість ступенів (N) безпосередньо впливає на частотну характеристику фільтра CIC. Відгук фільтра визначається шляхом настройки кількості ступенів N, коефіцієнта зміни швидкості R і кількості затримок в диференціаторі (званих диференціальна затримка).

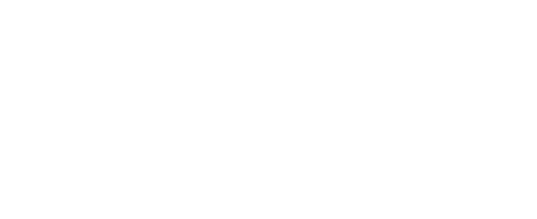
На рис. 2.11 і 2.12 показані інтегратор і диференціатор.



**+**

**z-1**

Рисунок 2.11 - Інтегратор



**z- M**

**-**

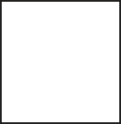
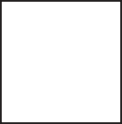
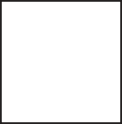
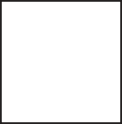
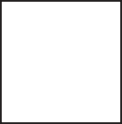
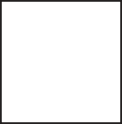
**+**

Рисунок 2.12 - Диференціатор

На рис. 2.13 показані три інтегратора і три диференціатора, які об'єднані з використанням деціматора для створення триступеневого фільтра CIC проріджування.

**High Sampling Frequency**

**Low Sampling Frequency**



висока частота вибірки

низька частота вибірки

**І**

**І**

**І**

**Д**

**Д**

**Д**

секція інтегратора

секція диференціювання

**R**

Рисунок 2.13 - Триступеневий фільтр децимації

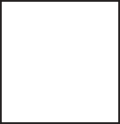
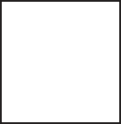
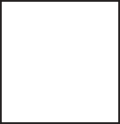
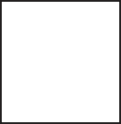
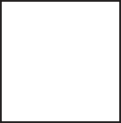
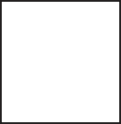
Integrator Section

Differentiator Section

На рис 2.14 показані три диференціатор а і три інтегратора, які об'єднані з використанням інтерполятора для створення триступеневого інтерполяційного фільтра CIC.

**Low Sampling Frequency**

**High Sampling Frequency**



низька частота вибірки

висока частота вибірки

**Д**

**Д**

**Д**

**І**

**І**

**І**

секція диференціювання

секція інтегратора

**R**

Рисунок 2.14 - Триступеневий фільтр інтерполяції

Differentiator Section

Integrator Section

Коли коефіцієнт проріджування є нецілим числом, фільтри, призначені для цілочисельного або простого дрібного проріджування, не можуть використовуватися безпосередньо. Одним з рішень є використання інтерполяційних фільтрів на основі поліномів. Серед них фільтр лінійної інтерполяції має просту структуру реалізації, потрібно тільки одне множення.

# Багатоканальність фільтрів

В системі цифрової обробки сигналів (DSP) часто є багато каналів даних, в яких потрібна фільтрація фільтрами CIC з тієї ж конфігурацією. Вони можуть бути об'єднані в один фільтр, який розділяє суматори, і на кожному етапі зменшує загальне споживання ресурсів. Наприклад, двоканальний паралельний фільтр вимагає двох тактів для обчислення двох виходів даних. Отримане обладнання повинно було б працювати з подвоєною швидкістю передачі даних, ніж індивідуальний фільтр. Це особливо корисно для більш високих темпів змін, коли суматори працюють швидко.

Для мінімізації кількість логічних елементів створили архітектуру з безліччю входів і одним виходом (MISO) для фільтрів проріджування і один вхід з декількома виходами (SIMO) для інтерполяційних фільтрів.

У багатьох практичних конструкціях сигнали каналів надходять від різних вхідних інтерфейсів. На кожен інтерфейс входу потрапляють параметри, включаючи коефіцієнти зміни швидкості, і їх застосовують до даних каналу, які збирається обробити фільтр CIC. Функції CIC дозволяють гнучко використовувати поділ часу секцій диференціатора низької швидкості за рахунок надання декількох вхідних інтерфейсів і ланцюжків обробки інформації з високою швидкістю, потім об'єднують всю обробку, пов'язану з більш низькою швидкістю в один ланцюжок. Ця стратегія може привести до повного використання ресурсів і являє собою найбільш ефективну реалізацію обладнання.

На рис. 2.15 показаний приклад архітектури MISO для фільтра CIC, який обробляє всього чотири канали. У цьому прикладі символи A, B, C, D мультиплексуються в один вихід A, B, C, D.

(A, C)

(B, D)

І І І 3

І І І 3



Д Д Д

(A, B, C, D)

Рисунок 2.15 - Архітектура з декількома входами і одним виходом

Частота дискретизації вхідних даних така, що можливо тільки мультиплексувати по два канали на шину, тому фільтр CIC повинен бути налаштований на два вхідних інтерфейсу. Оскільки потрібні два інтерфейси, коефіцієнт зміни швидкості також повинен бути не менше двох, щоб використовувати цю архітектуру. Таким способом можна підтримувати до 1024 каналів за допомогою декількох вхідних інтерфейсів.

Один вхід і кілька виходів (SIMO) - це функція, пов'язана з інтерполяційними фільтрами CIC. У цій архітектурі всі сигнали каналу, які представлені для фільтрації, надходять з одного вхідного інтерфейсу. Як і в разі MISO, можна використовувати розділ диференціатора з низькою частотою дискретизації серед більшої кількості каналів, так і в секціях інтегратора з більш високою частотою дискретизації.

Отже, ця архітектура має один екземпляр секції диференціатора і кілька паралельних примірників секцій інтегратора. Після обробки секцією диференціатора сигнали каналу поділяються на кілька паралельних секцій для обробки на високій частоті дискретизації секціями інтегратора.

На рис. 2.16 показаний приклад архітектури SIMO для фільтра CIC, який обробляє всього вісім каналів. У цьому прикладі символи A, B, C, D, E, F, G, H демультіплексуються на чотири виходи А, Е; B, F; С, G; D, H.

(A, E)



(A, B, C, D, E, F, G, H)

**Д**

**Д**

**Д**



|  |  |  |  |
| --- | --- | --- | --- |
| **8** | **І** | **І** | **І** |
| **8** | **І** | **І** | **І** |
| **8** | **І** | **І** | **І** |
| **8** | **І** | **І** | **І** |

(B, F)



(C, G)

(D, H)

Рисунок 2.16 - Архітектура з одним входом і декількома виходами

Необхідна частота дискретизації вихідних даних така, що можливо тільки мультиплексувати за часом два канали на шину даних. Тому фільтр CIC повинен бути налаштований на чотири вихідних інтерфейсу. Оскільки потрібно чотири інтерфейси, коефіцієнт зміни швидкості також повинен бути не менше чотирьох, щоб використовувати цю архітектуру.

Архітектура SIMO застосовується, коли обраний тип фільтра інтерполяції, а кількість інтерфейсів, обраних більше одного. Загальна кількість вхідних каналів має бути кратною кількості інтерфейсів. Щоб задовольнити цю вимогу, може знадобитися або вставити фіктивні канали, або використовувати більше однієї функції CIC.

Фільтри CIC мають характеристику фільтра нижніх частот. Існує тільки три параметри (коефіцієнт R зміни швидкості, кількість ступенів N і диференціальна затримка M), які можна змінити, щоб змінити характеристики смуги пропускання і відхилення псевдоніма формування зображення.

Однак через їх коефіцієнти підсилення в смузі пропускання і широких перехідних областей одні тільки фільтри CIC не можуть забезпечити продуктивність фільтра з плоскою смугою пропускання і вузькою перехідною області, яка зазвичай потрібно в додатках з фільтрацією децимації або інтерполяції.

# Аналіз ефективності CIC-фільтра

Ля чого необхідно змінити частоту дискретизації в фільтрі? Є дві причини. Перше - це продуктивність. Друге - це вартість. Багатошвидкісні системи зазвичай виконують завдання обробки з поліпшеними характеристиками продуктивності, одночасно пропонуючи цю продуктивність при значно нижчих витратах, ніж традиційні підходи. Багатошвидкісні фільтри - це цифрові фільтри, що працюють з ще однією зміною частоти дискретизації, і які вбудовані в архітектуру обробки сигналів. Іноді використання зміна частоти дискретизації при фільтрації є природним процесом в ланцюжку обробки сигналів. В інших випадках зміна частоти дискретизації застосовується для отримання доступу до переваг вартості, пов'язаних з многошвидкісною обробкою.

CIC-фільтр не вимагає зберігання коефіцієнтів фільтра і множників, оскільки всі коефіцієнти дорівнюють одиниці. Крім того, його реалізація на Soc кристалі ефективна завдяки своїй регулярній структурі, що складається з трьох основних будівельних блоків, мінімального зовнішнього управління і при цьому потрібно менш складна локальна синхронізація а її коефіцієнти зміни можна перенастроювати з додаванням схеми масштабування і мінімальними змінами синхронізації фільтра.

Призначення фільтра CIC двояке; по-перше, щоб видалити шум фільтрації, який міг би бути накладено назад на сигнали основної смуги частот, по-друге, перетворити потік даних з високою частотою дискретизації на виході модулятора в потік даних з більш низькою частотою дискретизації. Цей процес також відомий як проріджування, який по суті виконує функції усереднення і зниження швидкості одночасно.

Техніка передискретизації обмежує можливість використання накладення спектрів в основній смузі. Таким чином, вхідний сигнал може бути дискретизований з більш високою швидкістю, щоб запобігти накладення. Пізніше, дискретизований сигнал проріджується і фільтрується з використанням фільтра нижніх частот цифрового проріджування. Комбінація фільтрації і проріджування перед виконанням іншої операції фільтрації даних називається багатоступеневою цифровою фільтрацією проріджування. Він використовує переваги, пов'язані з обробкою сигналів з надлишковою дискретизацією, і продовжує знижувати вартість і енергоспоживання за рахунок децимації. Завдяки багаторазовому проріджування складність всього фільтра знижується, і наступні фільтри працюють з більш низькою частотою дискретизації, що додатково знижує енергоспоживання. Беручи до уваги, що одноступінчатий фільтр проріджування використовує велику кількість коефіцієнтів фільтра, що не підходить для енергоефективної реалізації. Багатоступінчастий цифровий фільтр децимації також забезпечує економію обчислень за рахунок перетворення частоти дискретизації і фільтрації в кілька етапів при

використанні ефективної структури реалізації, такої як багатофазні фільтри. Перевага багатоступінчастого підходу полягає в тому, що економічні фільтри нижчого порядку працюють з більш високою частотою дискретизації а фільтри більш високого порядку працюють з більш низькою частотою дискретизації. Архітектура фільтра розроблена для мінімізації арифметичної і апаратної складності. Завдяки застосуванню фільтра у вигляді каскаду з декількох лінійно-фазових каскадів, відношення швидкості введення до ширини смуги переходу різко знижується для кожного окремого каскаду. Цей багатоступінчастий фільтр може досягати як значно зменшених довжин фільтра, так і обчислювальних швидкостей у порівнянні зі стандартною конструкцією одношвидкісного фільтра і, таким чином, забезпечує практичне вирішення іншої складної проблеми.

Коли коефіцієнт проріджування і порядок фільтра високі, фільтр CIC використовує високе енергоспоживання, оскільки каскад інтегратора працює з найвищою частотою передискретизації з великим внутрішнім обсягом даних. Швидкість ланцюга також буде обмежена рекурсивним контуром каскаду інтегратора. Таким чином, нерекурсивний гребінчастий фільтр введений для заміни, коли коефіцієнт проріджування і порядок фільтра високі. Цей фільтр є фільтром проріджування з регулярною структурою. Ця властивість робить його придатним для реалізації створення надвеликих інтегральних схем. CIC-фільтр досліджений і розроблений завдяки його перевагам, таким як відсутність рекурсивного циклу і використання низького енергоспоживання через обчислення, які виконуються при більш низькій частоті дискретизації. Нерекурсивний гребінчастий фільтр має широкі можливості зміни швидкості.

# Медіанний фільтр

Медіанні фільтри досить часто застосовуються на практиці як засіб попередньої обробки цифрових даних. Специфічною особливістю фільтрів є

слабка реакція на відліки, що різко виділяються на тлі сусідніх. Ця властивість дозволяє застосовувати медіанну фільтрацію для усунення аномальних значень в масивах даних та зменшення імпульсних перешкод. Характерною особливістю медіанного фільтра є його нелінійність. У багатьох випадках застосування медіанного фільтра виявляється більш ефективним в порівнянні з лінійними фільтрами, оскільки процедури лінійної обробки є оптимальними при рівномірному або гаусовому розподілі перешкод, що в реальних сигналах може бути далеко не так.

Лінійні фільтри мають тенденцію розмивати края і інші деталі зображення. Лінійний фільтр погано працює та заважає впливом в каналі передачі інформації. Для подолання цих недоліків використовуються нелінійні фільтри.

Нелінійні фільтри можуть зберігати края і дуже ефективно усувають імпульсні перешкоди. Медіанний фільтр сортує значення пікселів в околиці, а потім замінює центральний піксель медіанним значенням в відсортованої групі, і він відомий як фільтр статистики вищого порядку.

Медіанний фільтр грає важливу роль в прикладних програмах попередньої обробки зображень, а також у збереженні країв зображення. Таким чином, щоб зберегти баланс між усуненням шуму і збереженням країв, розмір вікна повинен бути ретельно обраний.

# Медіанний алгоритм фільтрації

Медіанний фільтр - це метод цифрової фільтрації нелінійних упорядкованих статистичних даних, який зазвичай використовується для значного зниження шуму на зображенні. Це один з кращих віконних операторів з багатьох віконних операторів, таких як середній фільтр, мінімальний і максимальний фільтри і фільтр режимів. Проста ідея полягає в тому, щоб перевірити значення вибірки вхідного сигналу і вирішити, чи є він представницьким для сигналу. У зв'язку з цим медіанний фільтр часто

виконує кращу роботу, ніж методика фільтрації щодо збереження корисних деталей на зображенні.

Медіанний фільтр фільтрує кожен піксель в зображенні по черзі, і його сусідні використовуються, щоб вирішити, чи є він представницьким для його оточення. Зазвичай замість того, щоб замінити значення пікселя середнім значенням сусідніх значень, медіанний фільтр замінює його медіаною цих значень. Таким чином, значення з навколишньої околиці спочатку сортуються в числовому порядку, а потім значення розглянутого пікселя замінюється середнім (медіанного) значенням пікселя. Околиця називається вікном. Вікно може мати різні форми з центром в цільовому пікселі. Квадрат є типовою формою, обраної для вікон, визначених для 2D- зображень. Слід зазначити, що в нормальних умовах медіанний фільтр виконується з використанням вікна, що містить непарну кількість пікселів. Якщо розглянута околиця складається з парного числа пікселів, медіанне значення, вибране в якості вихідного, є середнім з двох середніх значень пікселів.

Активність медіанного фільтра пов'язана з його здатністю зберігати края зображень і багато корисних детале при одночасному видаленні або, по крайній мере, усунення шумового ефекту при певних умовах. Цей фільтр є локальним фільтром, який створює середній елемент з відсортованих значень кількості пікселів, взятих із зазначеного вікна, як показано на рис. 2.17. Відфільтрований піксель залежить від своїх найближчих сусідів, щоб вирішити, чи є він хорошим представником свого оточення, тому фільтр замінює значення пікселя медіаной зазначеного вікна.

Медіанна обробка фільтра може бути підсумована за допомогою наступних кроків; по-перше: вибирається вказане вікно в залежності від довжини фільтра, потім використовується алгоритм, який сортує пікселі в порядку зростання або зменшення. Другий етап обробки цього фільтра: видається медіанне значення відсортованих елементів. Кількість пікселів у вікні маскування переважно має бути непарним числом для вибору

середнього значення в якості медіани. Але коли це число парне, фільтр приймає середнє значення двох середніх пікселів. Останній крок обробки фільтра: медіана поміщається в центр обраного вікна. Ці процеси виконуються для всіх пікселів зображення шляхом переміщення покажчика пікселів по горизонталі і вертикалі.

|  |  |  |
| --- | --- | --- |
| **9** | **5** | **6** |
| **3** | **2** | **4** |
| **7** | **1** | **8** |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |

Рисунок 2.17 - Процедура сортування для традиційної медіанної конструкції фільтра.

Одна з переваг медіанного фільтра над фільтрами іншого порядку рангу, особливо середнього фільтра, полягає в тому, що медіанне значення є більш стійким середнім, ніж середнє значення; середнє значення істотно не порушено одним дуже непредставницьким пикселем в околиці. Медіанне значення оточуючих пікселів, швидше за все, буде значенням одного з пікселів в околиці вікна. Таким чином, медіанний фільтр з найменшою вірогідністю створює нові нереалістичні значення пікселів, особливо коли фільтр працює в перехідних зонах. З цієї причини метод медіанної фільтрації набагато краще, ніж метод середньої фільтрації з точки зору збереження гострих країв.

# Реалізація медіанного фільтра

Реалізація медіанного фільтра включає в себе кілька етапів обробки. Одним з цих етапів є попередня обробка зображення, при якій зображення має бути сегментованим і кожен сегмент повинен бути перетвореним в

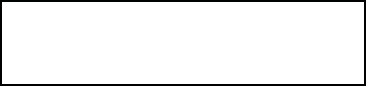
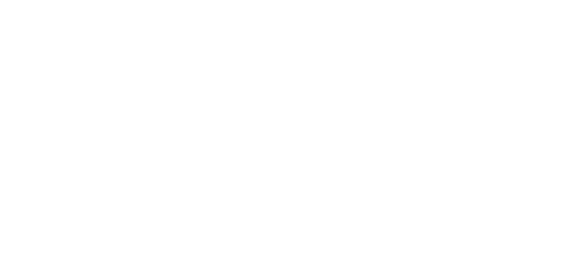
одновимірну форму. Вікно обробки k×k потрібно для сканування зображень, тому використовується генератор вікон ak×k, який імітує процес сегментації, генеруючи (k×k) пікселів за один раз при скануванні всього зображення. Ці пікселі подаються паралельно. k визначається відповідно до використовуваної середньою довжини фільтра.

Другий і найважливіший етап фокусується на оптимальному виборі підходящої техніки апаратної реалізації медіанного фільтра на основі відповідної H/W-платформи. У даній роботі розроблений конструкція і апаратна реалізація цього фільтр. який залежить головним чином від набору блоків XILINX System Generator (XSG). Останній крок в H/W реалізації медіанного фільтра - постоброка зображення. Оскільки вихідні дані системи будуть з плаваючою комою, їх необхідно перетворити в ціле число без знака. Вихідний піксель буде окремим пікселем, який необхідно перетворити назад в 2D- дані, використовуючи блок зміни форми. Зображення тепер готове для відображення.

Ці обробки можуть бути коротко викладені таким чином:

* читання вхідного зображення сірої шкали;
* розрахунок розміру вхідного зображення;
* ініціалізація покажчика пікселя, який вказує на перший піксель;
* фільтрація;
* одна додаткова ітерація застосовується для усунення непотрібного шуму на краях;
* відображення відфільтрованого зображення.

Коли моделювання завершено, блоки System Generator (SG) використовуються з обраним типом набору FPGA для створення нового блоку, який представляє конструкцію блоку Xilinx на останньому кроці моделювання для формування реального набору. На рис. 2.18 показаний розрахунковий потік реалізації з використанням системного генератора.



Simulink алгоритм

Системний генератор технологій

VHDL код

Рисунок 2.18 - Потік проектування з використанням XSG

Як уже згадувалося раніше, ядром медіанного дизайну фільтра є процес сортування, який вимагає безлічі операцій порівняння або порівняльних елементів. Цей процес може бути виконаний з використанням S/W (програмного) або H/W (апаратного) методів. Для впровадження програмного забезпечення; кількість необхідних операцій порівняння прямо пов'язане з кількістю відсортованих вибірок, як показано в рівнянні.

Процес сортування N зразків вимагає операції порівняння Z наступним чином:

Z = (N-1) + (N-2) + (N-3) ….. +1

Перша вибірка потребує (N-1) операціях порівняння, щоб бути найменшим або найбільшим елементом серед відсортованих вибірок, оскільки вона повинна порівнюватися з усіма своїми послідовниками, яким вони рівні (N-1). Таким чином, другі вибірки потребують (N-2) операцій порівняння, щоб бути в порядку сортування елементів, так як вони повинні порівнюватися з усіма своїми послідовниками, які враховуються. Цей процес триває до досягнення останніх зразків, які не потребують ні в якій операції порівняння.

Апаратна реалізація медіанного фільтра може бути виконана різними методами, включаючи оптимізовані конвеєрні методи. Медіанний фільтр являє собою найбільш важливу частину в багатьох системах шумозаглушення зображень, тому особлива увага повинна бути приділена його розробці і реалізації.

Основним недоліком архітектури медіанного фільтра в архітектурі систолічного масиву є вироблення на більш складному обладнанні в порівнянні з традиційними технологіями, оскільки одночасно повинно виконуватися кілька модулів виконання.

Проблема вартості апаратного систолічного масиву з'являється тільки з першої спроби. Потім ця проблема може бути в значній мірі усунена для поліпшених і оптимізованих систолічних конструкцій. Складність апаратного забезпечення може бути значно зменшена для медіанного фільтра в залежності від кількості стадій проектування. Таким чином, внутрішня структура кожного етапу може бути ефективно спрощена.

Максимально пам'ять, що використовується, покращує швидкодію, а розміри будуть менше з оптимізованої конструкцією систолічного масиву, оскільки число проміжних результатів, які повинно бути збереженим і буде зменшуватися зі зменшенням кількості порівнюваних вузлів.

Час виконання є дуже важливим фактором для середньої конструкції фільтра, тому його значення повинно бути якомога менше при збереженні апаратної простоти. Фактично спрощення апаратної складності конструкцій архітектури систолічного масиву призводить до ефективного поліпшення часу виконання, яке залежить головним чином від ступеня паралелізму розробленої системи для досягнення відмінних загальних параметрів продуктивності реалізується фільтра.

Час виконання медіанного фільтра значно вище при традиційному методі проектування, але при архітектурі систолічного масиву швидкість обробки значно поліпшується, а час виконання мінімізується до приблизно 8% від його значення традиційним методом. Час виконання фільтра

збільшується експоненціально зі збільшенням довжини фільтра, а також ця експоненціальна форма має місце при переході від традиційного до систолічного масиву. Швидкість обробки безпосередньо пов'язана з довжиною фільтра, а також на цю довжину сильно впливають необхідні апаратні компоненти, які потрібно швидко збільшити, щоб подолати межі доступних компонентів ПЛІС.

# Висновки за розділом

Традиційно структури FIR-фільтра розглядаються як суматори, помножувачі і елементи затримки, які з'єднуються послідовно, коли вони відображаються апаратно. На відміну від цього, підрахунок FPGA здійснюється шляхом визначення кількості блоків DSP і пошукових таблиць LUT. Є різниця в розрахунку ресурсів суматорів, помножувачів і елементів затримки при обчисленні блоків DSP. Причина в тому, що технології на основі ASIC забезпечує більшу гнучкість в досягненні необхідного сигналу і, отже, в проектуванні бажаного результату.

Генератор структур FIR-фільтрів має деякі обмеження, такі як відсутність можливості використання симетрії в транспонованій прямій формі, а також спільне використання вхідних даних декількох структур FIR- фільтрів.

З винаходом CIC-фільтра, з'явився ефективний спосіб виконання проріджування і інтерполяції. Це гнучкий фільтр без помножувачів, що підходить для апаратної реалізації, який також може обробляти довільні і великі зміни швидкості. Тому що їх структура складається з секції інтегратора, що працює з високою частотою дискретизації, і секції гребінки, що працює з низькою частотою дискретизації. Фільтр CIC еквівалентний N FIR-фільтрам, кожен з яких має прямокутну імпульсну характеристику. Оскільки всі коефіцієнти цих FIR-фільтрів рівні одиниці і, отже, є

симетричними, фільтр CIC має лінійну фазову характеристику та постійну групову затримку. Різкість фільтра може використовуватися для поліпшення відгуку фільтра CIC. У більшості додатків потрібно мати смугу пропускання, інакше вихідний сигнал може бути зруйнований. На жаль, один фільтр CIC страждає зі зменшенням смуги пропускання, що в багатьох випадках не може бути прийнято. Для поліпшення характеристик смуги пропускання і поліпшення характеристик CIC-фільтра можливо використовувати безліч методів, таких як компенсаційний фільтр, каскадний з CIC-фільтром, метод підвищення різкості, FIR-фільтр з багатофазним проріджуванням для досягнення широкосмугового компенсації CIC-фільтра.

Цифрові зображення часто спотворюються імпульсним шумом через помилки, що генеруються шумовим датчиком помилок, які виникають в процесі перетворення сигналів з аналого-цифрових, а також помилок, що генеруються в каналах зв'язку. Щоб видалити імпульсний шум і поліпшити якість зображення, необхідні методи основані на покращеному алгоритмі медіанної фільтрації.

Медіанні фільтри досить часто застосовуються на практиці як засіб попередньої обробки цифрових даних. Специфічною особливістю фільтрів є слабка реакція на відліки, що різко виділяються на тлі сусідніх. Ця властивість дозволяє застосовувати медіанну фільтрацію для усунення аномальних значень в масивах даних, зменшення імпульсних перешкод. Характерною особливістю медіанного фільтра є його нелінійність. У багатьох випадках застосування медіанного фільтра виявляється більш ефективним в порівнянні з лінійними фільтрами, оскільки процедури лінійної обробки є оптимальними при рівномірному або гаусовому розподілі перешкод, що в реальних сигналах може бути далеко не так. У випадках, коли перепади значень сигналів великі в порівнянні з дисперсією адитивного білого шуму, медіанний фільтр дає менше значення середньоквадратичної помилки в порівнянні з оптимальними лінійними фільтрами. Особливо ефективним медіанний фільтр виявляється при очищенні сигналів від

імпульсних шумів при обробці зображень, акустичних сигналів, передачі кодових сигналів і т.п. Однак детальні дослідження властивостей медіанний фільтрів як засобу фільтрації сигналів різного типу є досить рідкісними.

Основним недоліком архітектури медіанного фільтра - це необхідність в більш складному обладнанні в порівнянні з традиційними технологіями. Ця проблема вирішується з використанням оптимізованої архітектури, яка повинна скасувати будь-які неважливі апаратні компоненти, а також компоненти з повторними діями. Апаратна складність може бути поліпшена для конструкції фільтра в залежності від технологічного підходу. Час виконання є дуже важливим фактором для технології медіанного фільтра, тому його вартість повинна бути зменшена якомога менша зберігаючи апаратну простоту. При цьому спрощення апаратної складності проектів архітектури призводить до ефективного поліпшення часу виконання, що залежить в основному від ступеня паралелізму системи, щоб зробити відмінну загальну продуктивність параметрів реалізованого фільтра.

З урахуванням всіх переваг, багатофункціональності та перспектив у розвитку цифрових фільтрів, пропонується до розробки модель медіанного фільтра, яка в підсумку призведе до скорочення енергоспоживання і часу швидкості обробки зображень.

# ТРЕТІЙ РОЗДІЛ

**РОЗРОБКА, МОДЕЛЮВАННЯ ТА ПОБУДОВА СИСТЕМИ ОБРОБКИ ЗОБРАЖЕНЬ**

* 1. **Реалізація систем обробки зображень із застосуванням фільтрів**

Всі лінійні алгоритми фільтрації призводять до згладжування різких перепадів яскравості зображень, що пройшли обробку. Цей недолік, особливо істотний, якщо споживачем інформації є людина, принципово не може бути виключений в рамках лінійної обробки. Справа в тому, що лінійні процедури є оптимальними при гаусовському розподілі сигналів, перешкод і спостережуваних даних. Реальні зображення, строго кажучи, не підкоряються цим розподілом ймовірностей. Причому, одна з основних причин цього полягає в наявності у зображень різноманітних кордонів, перепадів яскравості, переходів від однієї текстури до іншої і т. п. Саме це і служить причиною поганої передачі кордонів при лінійної фільтрації.

Друга особливість лінійної фільтрації - її оптимальність. Зазвичай цій умові відповідають шумові перешкоди на зображеннях, тому при їх придушенні лінійні алгоритми мають високі показники. Однак, часто доводиться мати справу з зображеннями, перекрученими перешкодами інших типів. Однією з них є імпульсна перешкода. При її впливі на зображенні спостерігаються білі або/і чорні точки, хаотично розкидані по кадру. Застосування лінійної фільтрації в цьому випадку неефективно - кожен з вхідних імпульсів дає відгук у вигляді імпульсної характеристики фільтра, а їх сукупність сприяє поширенню перешкоди на всю площу кадру.

Вдалим рішенням перерахованих проблем є застосування медіанної фільтрації, запропонованої Дж. Тьюки в 1971 р. Відзначимо, що медіанна фільтрація являє собою евристичний метод обробки, її алгоритм не є математичним рішенням строго сформульованої задачі. Тому дослідниками

приділяється велика увага аналізу ефективності обробки зображень на її основі і порівнянні з іншими методами.

При застосуванні медіанного фільтра (МФ) відбувається послідовна обробка кожної точки кадру, в результаті чого утворюється послідовність оцінок. В ідейному відношенні обробка в різних точках незалежна, але в цілях її прискорення доцільно алгоритмічно на кожному кроці використовувати раніше виконані обчислення.

При медіанної фільтрації використовується двовимірне вікно (апертура фільтру), зазвичай має центральну симетрію, при цьому його центр розташовується в поточній точці фільтрації. На рис. 3.1 показані два приклади найбільш часто вживаних варіантів вікон у вигляді хреста і у вигляді квадрата. Розміри апертури належать до числа параметрів, що оптимізуються в процесі аналізу ефективності алгоритму. Відлік зображення, що опинилися в межах вікна, утворюють робочу вибірку поточного кроку.

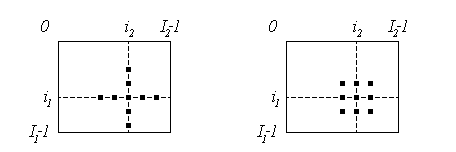


Рис. 3.1 – Приклади вікон при медіанній фільтрації

Двовимірний характер вікна дозволяє виконувати, по суті, двовимірну фільтрацію, оскільки для утворення оцінки залучаються дані як з поточних рядки і стовпці, так і з сусідніх. Позначимо робочу вибірку у вигляді одновимірного масиву; число його елементів дорівнює розміру вікна, а їх розташування довільно. Зазвичай застосовують вікна з непарним числом точок n (це автоматично забезпечується при центральній симетрії апертури і при входженні самої центральної точки в її склад). Якщо впорядкувати послідовність вибірки по зростанню, то її медіаною буде той

елемент, який займає центральне положення в цій впорядкованої послідовності. Отримане таким чином число і є продуктом фільтрації для поточної точки кадру. Зрозуміло, що результат такої обробки справді не залежить від того, в якій послідовності представлені елементи зображення в робочій вибірці Y. Введемо формальне позначення описаної процедури у вигляді:

x = med\*(y1, y2,…, yn) (3.1)

З (3.1) випливає, що дія медіанного фільтра полягає в "ігноруванні" екстремальних значень вхідної вибірки - як позитивних, так і негативних викидів. Такий принцип придушення перешкоди може бути застосований і для ослаблення шуму на зображенні. Однак дослідження придушення шуму за допомогою медіанної фільтрації показує, що її ефективність при вирішенні цього завдання нижче, ніж у лінійної фільтрації.

Результати експериментів, що ілюструють роботу медіанного фільтра, наведені на рис. 3.2. В експериментах застосовувався медіанний фільтр, що має квадратну апертуру зі стороною рівною 3. У лівому ряду представлені зображення, які спотворені перешкодою, в правому - результати їх медіанної фільтрації. На рис. 3.2*а* і рис. 3.2*в* показано вихідне зображення, спотворене імпульсною завадою. При її накладення використовувався датчик випадкових чисел з рівномірним законом розподілу на інтервалі 0-1, що виробляє у всіх точках кадру незалежні випадкові числа. Інтенсивність перешкоди задавалася ймовірністю її виникнення в кожній точці. Результати обробки говорять про практично повне придушення перешкоди в першому випадку і про її значне послаблення в другому.

На рис.3.2*д* показано зображення, спотворене незалежним гаусовським шумом, а на рис. 3.2*е* - результат його фільтрації медіанним фільтром.

*а*) *б*)

 *в*)  *г*)

 *е*)  *ж*)

Рис. 3.2. – Зображення до і після медіанної фільтрації

Разом з тим, як демонструє рис. 3.2*е*, медіанна фільтрація в меншій мірі згладжує кордон зображення, ніж будь-яка лінійна фільтрація. Механізм цього явища дуже простий і полягає в наступному. Припустимо, що апертура фільтру знаходиться поблизу кордону, що розділяє світлий і темний ділянки зображення, при цьому її центр розташовується в області темної ділянки. Тоді, найімовірніше, робоча вибірка буде містити більшу кількість елементів з малими значеннями яскравості, і, отже, медіана перебуватиме серед тих елементів робочої вибірки, які відповідають цій області зображення. Ситуація змінюється на протилежну, якщо центр апертури зміщений в область більш високої яскравості. Але це і означає наявність чутливості у медіанному фільтрі до перепадів яскравості. Існує величезна безліч інтерпретацій методів роботи медіанного фільтра, розглянемо ще один, на прикладі використання його при обробці зображення клітин крові - гранулоцитів. Перед вимірюванням розміру гранулоцита його зображення було піддано згладжування медіанним фільтром з метою усунення гранул, які можуть впливати на результат вимірювання. Зазвичай в процесі медіанної фільтрації значення сигналу в деякій околиці точки, в якій обчислюється відгук фільтра, за допомогою сортування за зростанням або зменшенням шикуються в варіаційний ряд. Відгук фільтра визначається як медіана - значення сигналу середини (центру) варіаційного ряду. Надалі цю околицю будемо називати вікном фільтра. Крім того, для спрощення будемо розглядати фільтр з квадратним вікном розміром n×n.

Отже, при обчисленні медіани в вікні фільтра число операцій з даними, наприклад, число операцій сортування, так само n2. При обробці зображення розміром M×N точок (пікселів) число операцій з даними буде велике і складе M×N×n2. Різні операції вимагають різних витрат часу виконання. При послідовному скануванні зображення кількість найбільш трудомістких операцій сортування можна скоротити. Так, при переході від точки О1 з вікном w1 до точки О2 з вікном w2 на рис. 3.3 можна з варіаційного ряду вікна w1 виключити точки стовпчика 1, впорядкувати

точки стовпчика 6 і об'єднати два отриманих варіаційних ряду в один. Такий алгоритм працює швидше в порівнянні з незалежним сортуванням в кожному вікні, проте загальне число маніпуляцій з даними (нехай і менш трудомістких), наприклад, хоча б перебір даних, залишається тим же самим, тобто досить великим. Тому при медіанної фільтрації зображень зазвичай обмежуються вікнами 3×3 або 5×5 і рідко більше, що цілком достатньо, наприклад, для усунення імпульсних перешкод.

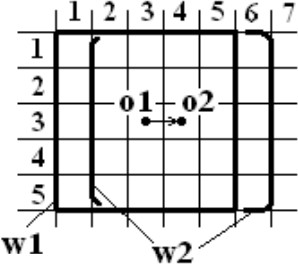


Рис. 3.3 - Сканування зображення вікном медіанного фільтра

Такі ж обмеження вимушено приймаються і для різних нелінійних операцій морфологічної обробки, що виконується в геометричному просторі зображення і, які на відміну від лінійних операцій, неможливо виконувати в просторі Фур'є. Разом з тим існує ряд завдань обробки зображень, які можна було б ефективно вирішити за допомогою медіанного фільтра, але вони вимагають вікна великого розміру. Одна з таких завдань буде розглянута нижче. Тому можливе підвищення швидкості медіанної фільтрації обіцяє великі перспективи в задачах обробки зображень.

Методи медіанної фільтрації достатню різноманітні. Їх можна вдосконалити. Один з таких апгрейдів називається - адаптивна медіанна фільтрація.

Медіанна фільтрація має і свої недоліки. Зокрема, експериментально встановлено, що у даного методу відносно слабка ефективність при фільтрації так званого флуктуаційного шуму. Крім того, при збільшенні розміру маски відбувається розмиття контурів зображення і, як наслідок, зниження чіткості зображення. Зазначені недоліки методу можна зменшити до мінімуму, якщо скористатися медіанної фільтрацією з динамічним розміром маски (адаптивною медіанною фільтрацією). Принцип обчислення центрального відліку при локальній обробці зображення ковзаючим вікном залишається все той же. Ця медіана з набору впорядкованих відліків, що потрапили у вікно (маску), а розмір ковзаючого вікна (маски) динамічний і залежить від яскравості сусідніх пікселів.

# Розрахунок відносини сигнал/шум (ВСШ)

Для того щоб розрахувати ВСШ скористаємося програмним пакетом MatLab. Як об'єкт дослідження візьмемо рис. 3.4 накладемо на нього шум

«Salt and Peper» і опрацюємо зображення медіанного фільтром з радіусом в 4 пікселя.

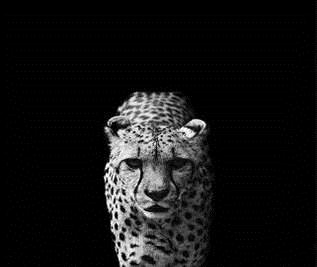


Рис. 3.4 - Зображення без шуму

В результаті отримаємо зображення на рис. 3.5

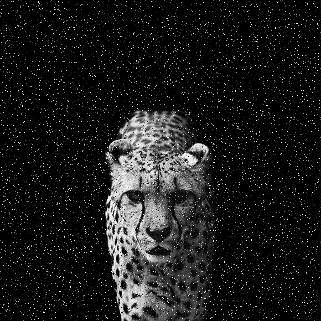


Рис. 3.5 - Зображення з шумом

Після обробки зображення медіанним фільтром з радіусом 4 пікселя отримаємо зображення на рис. 3.6.

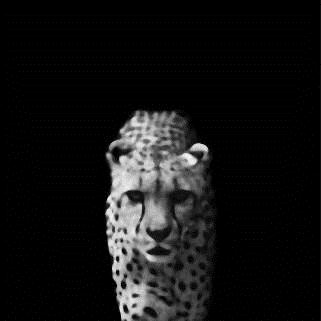


Рис. 3.6 - Зображення оброблене медіанного фільтром з радіусом 4

пікселя

Лістинг медіанної фільтрації:

SNR1 (black) = 101,4 - ВСШ для зображення на рис. 3.5 SNR2 (black) = 50 - ВСШ для зображення на рис. 3.6 SNR1 (white) = 1,8 - ВСШ для зображення на рис. 3.5 SNR2 (white) = 0,6 - ВСШ для зображення на рис. 3.6

З наведених значень видно, що після обробки зображення медіанним фільтром величина шумовиї складової по чорному кольору знизилася на 22 одиниці, а по білому на 1 одиницю. Але варто нагадати, що фільтрація відбувалася з радіусом в 4 пікселя, тобто якість зображення помітно погіршився.

Пікове відношення сигналу до шуму (англ. Peak signal-to-noise ratio) позначається абревіатурою PSNR і є інженерним терміном, що означає співвідношення між максимумом можливого значення сигналу і потужністю шуму, яке спотворює значення сигналу. Оскільки багато сигналів мають широкий динамічний діапазон, PSNR зазвичай вимірюється в логарифмічною шкалою в децибелах.

PSNR найбільш часто використовується для вимірювання рівня спотворень при стисненні зображень. Найпростіше його визначити через середньоквадратичну помилку (СКП), яка для двох монохромних зображень I і K розміру m× n, одне з яких вважається зашумленим наближенням іншого, обчислюється так:

 (3.2)

У нашому випадку СКП = 1,6×1000. У випадку, якби медіанний фільтр спрацював ідеально відкинувши всі шуми, СКП було б дорівнювати 0.

PSNR визначається так:



де MAXІ - це максимальне значення, яке приймається пікселем зображення. Коли пікселі мають 8 біт, MAXІ = 255.

Термін «Пікове відношення сигналу до шуму» є часто вживаним, але не зовсім вірним дослівним перекладом англійського терміну «peak signal-to- noise ratio». Знову ж правильним перекладом буде «ставлення пікового рівня сигналу до шуму». Тут враховується той факт, що при обчисленні PSNR обчислюється саме ставлення максимально можливого («пікового») сигналу по відношенню до рівня шуму, а не шукається максимальне ( «пікове») ставлення обчисленого значення сигнал/шум. У нашому випадку PSNR = 16,1. Тобто, ставлення пікового рівня сигналу до шуму 16,1. Чим ближче це значення до нуля, тим більше шумів в досліджуваному зображенні.

# Реалізація медіанної фільтрації за допомогою MATLAB

В MATLAB медіанна фільтрація схожа на використання усередненого фільтра, де значення інтенсивності кожного пікселя є усереднене значення інтенсивностей пікселів відповідної околиці. У багатьох випадках використання медіанного фільтра при вирішенні задач усунення шуму є більш ефективним, ніж звичайне усереднення. Застосування медіанної фільтрації призводить до менших спотворень кордонів, в порівнянні з операцією усереднення. Функція medfilt2 виконує медіанну фільтрацію.

Нижче представлений лістинг медіанної фільтрації: clc, clear all, close all

input = imread ( 'E: \ input \ evariste-011.jpg'); figure, imshow (input);

L = medfilt2 (input, [2 + 2]); figure, imshow (L)

В даному випадку зображення на рис. 3.5 було оброблено фільтром маскою з апертурою [2; 2]. На виході отримаємо зображення, яке показано на рис. 3.7.

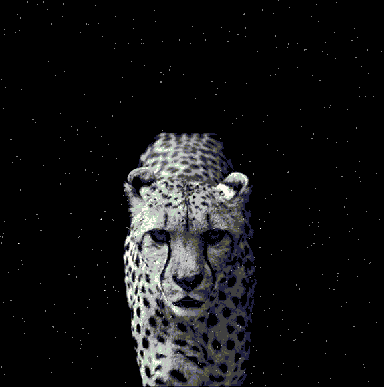


Рис. 3.7 - Результат медіанної фільтрації, апертура [2; 2]

Збільшимо апертуру до [4; 4], отримаємо зображення, яке показано на рис. 3.8.

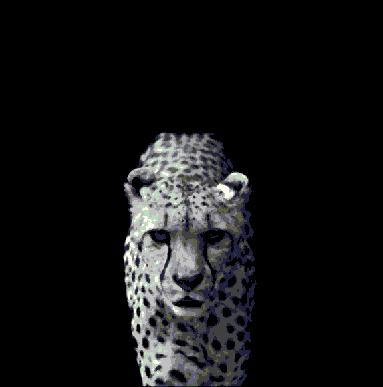


Рис. 3.8 - Результат медіанної фільтрації, апертура [4; 4]

Збільшимо апертуру до [7; 7], отримаємо зображення, яке показано на рис. 3.9.

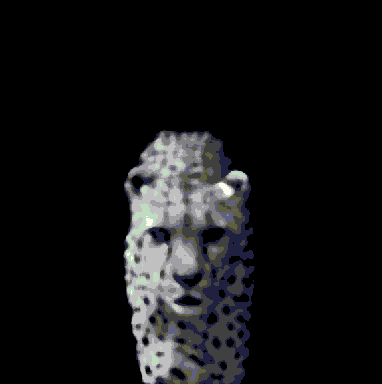


Рис. 3.9 - Результат медіанної фільтрації, апертура [7; 7]

Останній рис. 3.9 показує, що медіанна фільтрація має межу, не долаючи яку можна отримати досить якісно оброблене зображення. Збільшуючи апертуру маски більш ніж [7; 7] зображення стає розмитим. Усереднення на кордонах чорного і білого розтягує інформативну частину зображення і в результаті цього пропадають чіткі контури.

За допомогою MatLab розрахуємо шумові складові і порівняємо якість роботи медіанного фільтра.

Нижче представлена зведена таблиця 3.1, в якій порівнювалися вихідне зображення (рис. 3.5), з обробленими медіанного фільтром:

Таблиця 3.1 – Результати розрахунків

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № рисунку | PSNR | РСНР по СКП  (білий) | РСНР по СКП  (чорний) | РСНР по СЛП  (білий) | РСНР  по СЛП (чорний) |
| Рис. 3.7 | 20,4 | 0,98 | 80,1 | 0,98 | 80,1 |
| Рис. 3.8 | 25,3 | 0,7 | 66,3 | 0,70 | 66,3 |
| Рис. 3.9 | 27,1 | 0,61 | 52,4 | 0,61 | 52,4 |

# Обробка тестового зображення MATLAB

За допомогою програми MATLAB досліджуємо роботу медіанного фільтра. В якості об'єкту дослідження виберемо тестове зображення

«Kadr.bmp».

Лістинг програми виконує медіанну фільтрацію: clc, clear all, close all

input = imread ( 'E: \ input \ Kadr.bmp'); figure, imshow (input);

L = medfilt2 (input, [2 + 2]); figure, imshow (L)

Нанесемо на «Kadr.bmp» шуми «Salt and Pepper» (рис.3.10) і опрацюємо медіанним фільтром, збільшуючи апертуру. На рис. 3.11-3.14 показано зображення, які оброблені з різною апертурою. Останній рисунок буде оброблений з апертурою [10; 10] з метою показати який результат обробки при надмірно великому радіусі робочого вікна фільтра.



Рис. 3.10 - Зображення з шумом

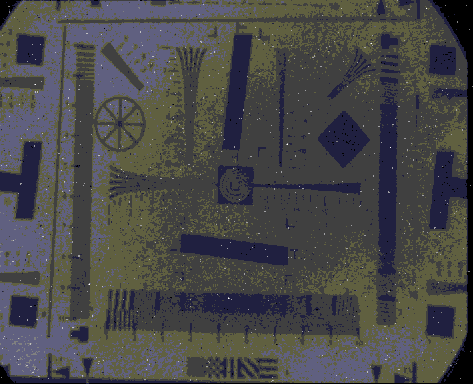


Рис. 3.11 - Результат медіанної фільтрації, апертура [2; 2]



Рис. 3.12 - Результат медіанної фільтрації, апертура [4; 4]



Рис. 3.13 - Результат медіанної фільтрації, апертура [5; 5]

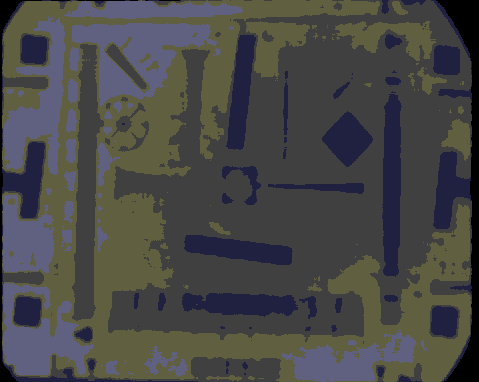


Рис. 3.14 - Результат медіанної фільтрації, апертура [10; 10]

З рисунку 3.14 видно, що чіткі межі зникли в результаті некоректної фільтрації. Нижче в таблиці 3.2 наведено дані про розрахунки шумів і ДВК. Розрахунок був виконаний за допомогою програми MATLAB.

Таблиця 3.2 - Результати розрахунків

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № рисунку | PSNR | РСНР по СКП  (білий) | РСНР по СКП  (чорний) | РСНР по СЛП  (білий) | РСНР по СЛП  (чорний) | ДВК |
| Рис. 5.2 | 18,4 | 1,5 | 10,72 | 1,3 | 2,34 | 0,001 |
| Рис. 5.3 | 33,6 | 10,7 | 10,00 | 12,1 | 2,3,0 | 0,290 |
| Рис. 5.4 | 33,2 | 10,3 | 9,40 | 11,3 | 2,20 | 0,150 |
| Рис. 5.5 | 32,6 | 18,1 | 10,10 | 20,9 | 2,00 | 0,120 |
| Рис. 5.6 | 30,1 | 46,2 | 11,80 | 61,3 | 3,10 | 0,003 |

З табл.. 3.2 видно, що при настільки сильному зашумленні, старндартний медіанний фільтр успішно впорався з відновленням зображення, прибравши шум і лише щодо несильно спотворивши контури об'єктів.

# Висновки за розділом

Медіанна фільтрація - достатньо часто застосовуваний метод попередньої обробки сигналів. Специфічною особливістю медіанний фільтрів є вибірковість по відношенню до елементів масиву, які представляють собою немонотонну складову послідовності чисел в межах вікна (апертури) фільтра, і різко виділяються на тлі сусідніх відліків. У той же час на монотонну складову послідовності медіанний фільтр не діє, залишаючи її без змін. Завдяки цій особливості, медіанний фільтри при оптимально обраної апертурі можуть, наприклад, зберігати без спотворень різкі межі об'єктів, ефективно пригнічуючи некорельовані або слабо корельовані перешкоди і малорозмірні деталі. Ця властивість дозволяє застосовувати медіанну фільтрацію для усунення аномальних значень в масивах даних, зменшення викидів і імпульсних перешкод. Характерною особливістю медіанного фільтра є його нелінійність. У багатьох випадках застосування медіанного фільтра виявляється більш ефективним в порівнянні з лінійними фільтрами, оскільки процедури лінійної обробки є оптимальними при рівномірному або гаусовому розподілі перешкод, що в реальних сигналах може бути далеко не так. У випадках, коли перепади значень сигналів великі в порівнянні з дисперсією адитивного білого шуму, медіанний фільтр дає менше значення середньоквадратичної помилки в порівнянні з оптимальними лінійними фільтрами. Особливо ефективним медіанний фільтр виявляється при очищенні сигналів від імпульсних шумів при обробці зображень, акустичних сигналів, передачі кодових сигналів і т.п. Розроблена модифікація медіанного фільтра має високі характеристики і може застосовуватися в різних пристроях цифрової обробки зображень, що працюють в різній складної сигнально-перешкодній обстановці. Описаний алгоритм в більшості випадків повністю прибирає імпульсний шум типу «Salt and Pepper» і добре зберігає кордони об'єктів зображення. З цієї причини він не може бути застосований на етапі

попередньої обробки в пристроях розпізнання та аналізу зображень, де вимоги до якості зображень часто буваю досить високими.

Запропонований алгоритм медіанної фільтрації за попередніми розрахунками допускає ефективну апаратну реалізацію на сучасних сигнальних процесорах і мікроконтролерах.

# ВИСНОВКИ

Вибір методу відображення FPGA залежить від програми. Безліч доступних методів має бути вибрано і використано відповідним чином у всіх проектах. Наявність у виробі FPGA означає феноменальну гнучкість логіки цього виробу (в більшості випадків це заводська можливість корекції логіки роботи пристрою без проведення будь-яких електромонтажних операцій, що дозволяє врахувати потреби споживачів). Ресурси сучасних FPGA (навіть молодших в обраному сімействі) вже дозволяють реалізовувати складні алгоритми, в тому числі, алгоритми цифрової обробки сигналів.

Можливість створювати на основі FPGA багатоканальні системи паралельної обробки даних з гарантованою логічною незалежністю процесів обробки даних (в поєднанні з високою надійністю самих FPGA) створює серйозну альтернативу сигнальним процесорам (DSP) і ARM в високонадійних завданнях управління і контролю. Разом з тим, в світі явно відчувається і зворотна тенденція - реалізація засобами FPGA процесорів і контролерів з масштабованої архітектурою.

DSP є клас апаратних пристроїв, які знаходяться десь між ASIC і ПК з точки зору продуктивності і складності конструкції. Вони можуть бути запрограмовані або за допомогою асемблерного коду, або на мові програмування, що є одним з явних переваг платформи. Однак алгоритми, розроблені для DSP, не можуть бути високопаралельними без використання декількох DSP. Продуктивність алгоритму, безумовно, вище, ніж на персональних комп’ютерах, але в деяких випадках системи ASIC або FPGA є єдиним вибором для проектування. Недавні досягнення в технології DSP привели до дуже швидким реалізаціям алгоритму. Хоча переваги ASIC і FPGA все ще застосовні, це нове покоління DSP змусило деяких інженерів переглянути розробку FPGA. Проте, у міру появи нових DSP на ринку з'являються і нові FPGA, очікується, що дві архітектури матимуть однаково збільшувати продуктивність для кожного нового покоління процесорів.

На підставі мікросхем цифрового сигнального процесора (DSP) був розроблений універсальний ПЗС-контролер (який може працювати з ПЗС- матрицею), для реалізації таких важливих операцій, як зчитування CCD, взаємодія з хостом і корельована подвійна вибірка для усунення скидання шуму. Після вивчення функціональних можливостей плати DSP, плати зміщення і тактового генератора, інтерфейсу хоста і збору даних з коду DSP виводено алгоритм алгоритму кінцевого автомата.

Прилад із зарядним зв'язком був розроблений в основному як компактний датчик зображення для споживчого та промислового ринків, але в даний час також є видатним видимим і ультрафіолетовим датчиком довжини хвилі зображення у багатьох областях наукових досліджень включаючи космічну науку і дистанційне зондування Землі і планети. ПЗС- матриця буде прагнути захоплювати максимальну кількість пікселів в фокальній площині, ефективно реєструвати фотони в самому широкому спектральному діапазоні, і динамічний сигнал в діапазоні при збереженні мінімально можливого шуму зчитування. Щодо недавня поява технології сенсорних зображень на основі оксиду металу і напівпровідників (CMOS), можливо, є найбільш важливим досягненням в області твердотільної візуалізації з моменту винаходу ПЗС. Технологія CMOS дозволяє інтегрувати одиночну кремнієву мікросхему з великою кількістю пікселів фотодіодів разом з усіма допоміжними елементами електроніки, необхідних для адресації масиву і оцифровки результуючого аналогового відео сигнал. У порівнянні з CCD, CMOS обіцяє більш компактну, меншу масу, меншу потужність і потенційно більш радіаційно-стійке камера.

На ринок цифрової техніки успішно проникли CMOS-датчики зображення з високими характеристиками, які мають вбудовані засоби регулювання глибини різкості, і дозволяє отримувати зображення відмінної якості при малому фокусній відстані. Ці датчик містять вбудовані цифрові фільтри, які використовуються для балансування нерівномірності освітлення або миттєвої корекції дефектів в момент зйомки. Фільтри обробки зображень

застосовуються для накладення різних ефектів, підвищення якості зображень шляхом усунення помилкових даних або поліпшення характеристик. Тип використовуваного фільтра залежить від завдання. При виборі необхідно орієнтуватися не тільки на вигляд зашумленості, якщо завдання полягає у відновленні зображення, а й на саме зображення. У ряді випадків можливе істотно зменшити перешкоди, що вносяться камерою, оптичною системою.

Традиційно структури FIR-фільтра розглядаються як суматори, помножувачі і елементи затримки, які з'єднуються послідовно, коли вони відображаються апаратно. На відміну від цього, підрахунок FPGA здійснюється шляхом визначення кількості блоків DSP і пошукових таблиць LUT. Є різниця в розрахунку ресурсів суматорів, помножувачів і елементів затримки, при обчисленні блоків DSP. Причина в тому, що технології на основі ASIC забезпечують більшу гнучкість в досягненні необхідного сигналу і, отже, в проектуванні бажаного результату.

Генератор структур FIR-фільтрів має деякі обмеження, такі як відсутність можливості використання симетрії в транспонованою прямій формі, а також спільне використання вхідних даних декількох структур FIR- фільтрів.

З винаходом CIC-фільтра, з'явився ефективний спосіб виконання проріджування і інтерполяції. Це гнучкий фільтр без помножувачів, що підходить для апаратної реалізації, який також може обробляти довільні і великі зміни швидкості. Їх структура складається з секції інтегратора, що працює з високою частотою дискретизації, і секції гребінки, яка працює з низькою частотою дискретизації. Фільтр CIC еквівалентний N FIR-фільтрам, кожен з яких має прямокутну імпульсну характеристику. Оскільки всі коефіцієнти цих FIR-фільтрів рівні одиниці і, отже, є симетричними, фільтр CIC має лінійну фазову характеристику та постійну групову затримку. Різкість фільтра може використовуватися для поліпшення відгуку фільтра CIC. У більшості прикладних програмах потрібно мати смугу пропускання, інакше вихідний сигнал може бути зруйнований. На жаль, один фільтр CIC

страждає з падінням смуги пропускання, що в багатьох випадках не може бути прийнято. Для поліпшення характеристик смуги пропускання і поліпшення характеристик CIC-фільтра ми можемо використовувати безліч методів, таких як компенсаційний фільтр, каскадний з CIC-фільтром, метод підвищення різкості, FIR-фільтр з багатофазним проріджуванням для досягнення широкосмугового компенсації CIC-фільтра.

Цифрові зображення часто спотворюються імпульсним шумом через помилки, що генеруються шумовим датчиком, помилок, що виникають в процесі перетворення сигналів з аналого-цифрових, а також помилок, що генеруються в каналах зв'язку. Щоб видалити імпульсний шум і поліпшити якість зображення, на яке ми впливаємо, необхідні методи, що засновані на покращеному алгоритмі медіанної фільтрації.

Медіанні фільтри досить часто застосовуються на практиці як засіб попередньої обробки цифрових даних. Специфічною особливістю фільтрів є слабка реакція на відліки, що різко виділяються на тлі сусідніх. Ця властивість дозволяє застосовувати медіанну фільтрацію для усунення аномальних значень в масивах даних, зменшення імпульсних перешкод. Характерною особливістю медіанного фільтра є його нелінійність. У багатьох випадках застосування медіанного фільтра виявляється більш ефективним в порівнянні з лінійними фільтрами, оскільки процедури лінійної обробки є оптимальними при рівномірному або гаусовому розподілі перешкод, що в реальних сигналах може бути далеко не так. У випадках, коли перепади значень сигналів великі в порівнянні з дисперсією адитивного білого шуму, медіанний фільтр дає менше значення середньоквадратичної помилки в порівнянні з оптимальними лінійними фільтрами. Особливо ефективним медіанний фільтр виявляється при очищенні сигналів від імпульсних шумів при обробці зображень, акустичних сигналів, передачі кодових сигналів і т.п. Однак детальні дослідження властивостей медіанних фільтрів як засобу фільтрації сигналів різного типу є досить рідкісними.

Основним недоліком архітектури медіанного фільтра - це необхідність в більш складному обладнанні в порівнянні з традиційними технологіями. Ця проблема вирішується з використанням оптимізованої архітектури, яка повинна скасувати будь-які неважливі апаратні компоненти, а також компоненти з повторними діями. Апаратна складність може бути поліпшена для конструкції фільтра, в залежності від технологічного підходу. Час виконання є дуже важливим фактором для технології медіанного фільтра, тому його вартість повинна бути зменшена якомога менше зберігаючи апаратну простоту. При цьому спрощення апаратної складності проектів архітектури призводить до ефективного поліпшення часу виконання, що залежить в основному від ступеня паралелізму системи, щоб зробити відмінну загальну продуктивність параметрів реалізованого фільтра.

З урахуванням всіх переваг, багатофункціональності та перспектив у розвитку цифрових фільтрів, пропонується до розробки модель медіанного фільтра, яка в підсумку призведе до скорочення енергоспоживання, часу швидкості обробки зображень.

Медіанна фільтрація - достатньо часто застосовуваний метод попередньої обробки сигналів. Специфічною особливістю медіанний фільтрів є вибірковість по відношенню до елементів масиву, які представляють собою немонотонну складову послідовності чисел в межах вікна (апертури) фільтра, і різко виділяються на тлі сусідніх відліків. У той же час на монотонну складову послідовності медіанний фільтр не діє, залишаючи її без змін. Завдяки цій особливості, медіанні фільтри при оптимально обраної апертурі можуть, наприклад, зберігати без спотворень різкі межі об'єктів, ефективно пригнічуючи некорельовані або слабо корельовані перешкоди і малорозмірні деталі. Ця властивість дозволяє застосовувати медіанну фільтрацію для усунення аномальних значень в масивах даних, зменшення викидів і імпульсних перешкод. Характерною особливістю медіанного фільтра є його нелінійність. У багатьох випадках застосування медіанного фільтра виявляється більш ефективним в порівнянні

з лінійними фільтрами, оскільки процедури лінійної обробки є оптимальними при рівномірному або гаусовому розподілі перешкод, що в реальних сигналах може бути далеко не так. У випадках, коли перепади значень сигналів великі в порівнянні з дисперсією адитивного білого шуму, медіанний фільтр дає менше значення середньоквадратичної помилки в порівнянні з оптимальними лінійними фільтрами. Особливо ефективним медіанний фільтр виявляється при очищенні сигналів від імпульсних шумів при обробці зображень, акустичних сигналів, передачі кодових сигналів і т.п. Розроблена модифікація медіанного фільтра має високі характеристики і може застосовуватися в різних пристроях цифрової обробки зображень, що працюють в різній складної сигнально-перешкодній обстановці. Описаний алгоритм в більшості випадків повністю прибирає імпульсний шум типу «Salt and Pepper» і добре зберігає кордони об'єктів зображення. З цієї причини він не може бути застосований на етапі попередньої обробки в пристроях розпізнання та аналізу зображень, де

вимоги до якості зображень часто буваю досить високими.

Запропонований алгоритм медіанної фільтрації за попередніми розрахунками допускає ефективну апаратну реалізацію на сучасних сигнальних процесорах і мікроконтролерах.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Антонью А. Цифрові фільтри: аналіз і проектування. - М .: Радіо та зв’язок, 1983. - 320 с.
2. Баскаков С.І. Радіотехнічні ланцюги і сигнали: Підручник для вузів.

- М .: Вища школа, 1988.- 448 с.

1. Бат М. Спектральний аналіз в геофізики. - М .: Недра, 1980.- 535 с.
2. Бендат Дж., Пірсол А. Прикладний аналіз випадкових даних. - М .: Світ, 1989. - 540 с.
3. Блейхут Р. Швидкі алгоритми цифрової обробки сигналів. - М .: Світ, 1989. - 448 с.
4. Обчислювальна математика і техніка в розвідувальної геофізики: Довідник геофізика. - М .: Недра, 1990. - 498 с.
5. Гольденберг Л.М. та ін. Цифрова обробка сигналів: Довідник. - М .: Радіо та зв’язок, 1985.- 312 с.
6. Гольденберг Л.М. та ін. Цифрова обробка сигналів: Навчальний посібник для вузів. - М .: Радио и связь, 1990.- 256 с.
7. Гутников В.С. Фільтрація вимірювальних сигналів. - Л .: Вища школа, 1990. - 192 с.
8. Даджіон Д., Мерсер Р. Цифрова обробка багатовимірних сигналів. - М .: Світ, 1988. - 488 с.
9. Дмитрієв В.І. Прикладна теорія інформації: Підручник для студентів вузів. - М .: Вища школа, 1989.- 325 с.
10. Канасевіч Е.Р. Аналіз тимчасових послідовностей в геофізики. - М.: Недра, 1985.- 300 с.
11. Клаербоут Д.Ф. Теоретичні основи обробки геофізичної інформації з додатком до розвідки нафти. - М .: Недра, 1981. - 304 с.
12. Кулханек О. Введення в цифрову фільтрацію в геофізики. - М .: Недра, 1981. - 198 с.
13. Купер Дж., МакГілл А. Імовірнісні методи аналізу сигналів і систем. - М .: Світ, 1989. - 376 с.
14. Макс Ж. Методи і техніка обробки сигналів при фізичних вимірюваннях: У 2-х томах. - М .: Мир, 1983.
15. Марпл-мл. С.Л. Цифровий спектральний аналіз та його додатки. - М .: Світ, 1990. - 584 с.
16. Нікітін А.А. Теоретичні основи обробки геофізичної інформації: Підручник для вузів. - М .: Недра, 1986.- 342 с.
17. Оппенгейм А.В., Шафер Р.В. Цифрова обробка сигналів. - М .: Связь, 1979. - 416 с.
18. Відніс Р., Еноксон Л. Прикладний аналіз часових рядів. - М .: Світ, 1982. - 428 с.
19. Рабинер Л., Гоулд Б. Теорія і застосування цифрової обробки сигналів. - М .: Світ, 1978. - 848 с.
20. Рапопорт М.Б. Обчислювальна техніка в польовий геофізики: Підручник для вузів. - М .: Недра, 1993.- 350 с.
21. Сиберт У.М. Ланцюги, сигнали, системи. - М .: Світ, 1988. - 336 с.
22. Хемминг Р.В. Цифрові фільтри. - М .: Недра, 1987. - 221 с.
23. Лосєв А.К. Лінійні радіотехнічні ланцюга: Підручник для вузів. - М .: Вища школа, 1971. - 560 с.
24. Васильєв Д.В. Радіотехнічні ланцюги і сигнали: Навчальний посібник для вузів. - М .: Радио и связь, 1982. - 528 с.
25. Зінов'єв А.Л., Філіппов Л.І. Введення в теорію сигналів і ланцюгів: Навчальний посібник для вузів. - М .: Вища школа, 1975. - 264 с.
26. Севостьянов Б.А. Курс теорії ймовірностей і математичної статистики. - М .: Наука, 1982. - 256 с.