2 слайд ----

СЛАЙД 3 – Чому саме FPGA? ----------------------------------------------------------

Багато проектів, що вимагають роботи в реальному часі, реалізуються на ПЛІС.

Серед них крипто процесори, моніторинг та захист трафіку, прискорювачі процесора. І ці системи під'єднуються до ПК через швидкісний інтерфейс PCI і забезпечують виконання певних операцій, що не можуть бути реалізовані на ЦП.

DSP (ЦОС) саме цей набір операцій дуже розповсюджений на ПЛІС. Переді мною стояла задача виконати одночасну генерацію сигналів і така сама одночасна обробка.

Проаналізувавши можливі платформи для реалізації, єдиним варіантом була саме ПЛІС.

ТЕКСТ СЛАЙДУ

Паралельні модулі існують і у графічних процесорах (CUDA) , але поєднані жорстко і направлені все ж на обробку масивів даних(графічної інформації)

Щодо спеціалізованих блоків, сучасні мікропроцесорні рішення, такі як STM32, мають поруч з основним ядром блоки FPU, що виконують операцію множення за 1 такт. Але такий UNIT там один, що не врятує від послідовних затримок у часі.

Загалом, спільними недоліками як CUDA так і ПЛІС є велика складність у проектуванні. Робота блоків CUDA задається спеціальною версією мови С, що направлена на опис паралельних процесів. Архітектура ПЛІС задається ; мовами опису апаратури, VHDL та Verilog. Ці мови розроблені дуже давно і не схожі ні на які сучасні мови програмування. Та й принципи роботи їх відрізняються. Тому дуже мало спеціалістів працює з ПЛІС.

У дипломі розглянуто варіант проектування даного продукту на мікропроцесорній системі Cortex M4. Це архітектура ARM, але попри звичайне ядро RISC має ще допоміжне для виконання дій над числами з плаваючою комою. Але все ж для забезпечення потрібного рівня реакції схеми, одноядерний варіант ARM не підійшов би для цього проекту.

СЛАЙД 4 ------------------------------------------------------------

Так виглядає приблизна схема цифрового синтезатора. Як видно вона складається з ЦП DSP та багатьох модулів пам’яті, що поєднанні величезними шинами. З огляду на роботу та можливості ця схема є чудовою, і саме так реалізовані усі синтезатори масового виробництва. Але як на мене головним недоліком є складність проектування такої схеми, розробка плати. Дуже велика кількість шин. І звісно надійність такої схеми менша за 1 кристальний продукт.

СЛАЙД 5 ------------------------------------------------------------------------

Це схема одного каналу синтезу, в мене це реалізовано модулем BLOCK.

Осцилятор з мультиплексором на виході, керується двома числами, що задають частоту та вибір виходу : меандр, пила, синус.

Вихід осцилятора подається на модуль обвідної, що приймає один з двох сигналів від 2-х інших модулів : LFO або ADSR. Модуль LFO генерує синус, ADSR - кусково-лінійну функцію.

Таймер керується натиском клавіші, та рахує час з моменту натискання і тим самим задає значення функції.

СЛАЙД 6 -----------------------------------------------------------------------------

З приводу принципу роботи схеми. Є варіант розроблення такого собі ядра, що мав би регістрову пам’ять, спілкувався з ПК, та надсилав би дані на окремі блоки і приймав результати назад і надсилав далі. По суті це б привело до варіанту мікроконтролера. А так як більшість модулів працюють як конвеєр, то я вирішив і робити модулі що з’єднані послідовно.

Принцип роботи конвеєра на ПЛІС полягає в простому розміщенні регістрів між певними стадіями процесу. І при кожному такті дані переходять на іншу стадію.

СЛАЙД 7 -----------------------------------------------------------------------------------

Це є діаграма модуля BLOCК, що був структурно наведений вище.

Всередині пристрою дані оброблюються в двох формах, як ціле число, так і у вигляді плаваючою коми.

Для економії місця доцільніше зберігати дані у цілочисельному форматі, так само як і сумувати їх. Але забезпечення обвідної функції потрібно було використовувати число від 0 до 1 з плаваючою комою, для керування амплітуди сигналу. Через це в проекті використовується 2 блоки перетворення у формат IEEE754 та назад. А в середині модуля ADSR використовується суматор для плаваючої коми.

СЛАЙД 8 ---------------------------------------------------------------------------------

Це головний рівень. Для тестування я не міг розмістити більше 2 модулів синтезу, тому ця схема для такого варіанту. Але взагалі ідеєю будо мати як мінімум 5 а краще всі 12 модулів для кожної клавіші.

Надання кожному блоку власної пам’яті не є економічно вигідним та розумним.

Так як максимальна частота звуку генерованого близька до 16 кГц, а кожен період складається з 4096 точок, то загальна частота дискретизації виходить 60МГц, звісно це є величезна частота і настільки велике значення не потрібне.

Через це можливе використання 1 модуля тоді частоту роботи пам’яті потрібно збільшити у 12 разів від потрібної частоти дискретизації.

Отже схема складається з цього модуля пам’яті, 2-х модулів синтезу, суматора, дельта-сигма ЦАП, UART ресивера, обробника клавіш, та PLL.

СЛАЙД 9 -----------------------------------------------------------------------------------

На слайді показана необхідна аналогова периферія для повної роботи пристрою. ЦАП бажано саме мікросхему. ФНЧ складається з RC ланцюга. Операційний підсилювач з двохполярним живленням. І це схема отримання цього живлення.

СЛАЙД 10 ----------------------------------------------------------------------------

Робота таймера полягає у встановленні стану в якому має знаходитися ADSR.

Також на нього покладена функція повідомляти що даний BLOCK зайнятий, цей стан не можна прив’язати до клавіші, так як BLOCK працює навіть після відпускання клавіші.

Значення які має досягти таймер в кожному стані приходять через порти.

СЛАЙД 11 ----------------------------------------------------------------------------

Даний модуль реагує на зміну порту BUTTON, та в залежності від поточного значення октави, має встановити потрібний доданок для отримання потрібної частоти.

Для економії пам’яті було вирішено зберігати лише значення доданків для найнижчої октави, тобто 12 32-бітних чисел. Завдяки тому, що частоти тієї самої ноти в наступній октаві буде у 2 рази вища, то отримання потрібного доданку відбувається зсувом на значення поточної октави.

Також модуль займається арбітражем BLOCKів, приймаючи сигнал BUSY, помічає BLOCK як зайнятий.

СЛАЙД 12 ----------------------------------------------------------------------------

Передача інформації починається коли RX перейде з 1 до 0. Для того щоб не зберігати значення RX як раз на зміні фронту, спочатку автомат перейде до стану START. Тоді дані будуть зберігатися посередині імпульсу. Так як UART передає по 8 бітів у модулі UARTPROC реалізовано збереження слова по сигналу READY. І коли буде зібрано 5 слів (адреса і 32 бітне число), відбуваються зміни налаштувань.

СЛАЙД 13 ----------------------------------------------------------------------------

Модуль складається з блока, що виконує операції додавання та віднімання, в залежності від керуючого сигналу. Перший вхід суматора приєднано з його виходом через регістр. Інший вхід з’єднаний з мультплексором.

Через те, що суматор конвеєрний, перед кожною зміною стану відбувається очистка. Сигнал приходить від ТАЙМЕРА. НА час очистки та заповнення новими значеннями регістр першого доданку блокується, так само як і вихід модуля, для забезпечення постійного сигналу.

СЛАЙД 14 ----------------------------------------------------------------------------

Принцип роботи цього ЦАП полягає у переводі вхідного бітового вектору у однобітний сигнал. Вхідне значення керує шириною вихідного імпульсу. Подаючи цей сигнал на ФНЧ, імпульс вирівнюється у певну напругу. На малих частотах даний ЦАП видає досить велику похибку. Для забезпечення найкращої якості необхідно використовувати спеціалізовану паралельну мікросхему.