

УДК 004.921

## РОЗРОБКА КОДУ АЛГОРИТМУ ДЕКОДУВАННЯ ЗОБРАЖЕНЬ ЗАСОБАМИ АПАРАТНОЇ МОВИ VHDL

*Жиданова Ю.І., Самощенко О.В.*

*Донецький національний технічний університет*

*E-mail: nato\_net06@mail.ru*

*У даній статті розглядається методи генерації VHDL коду за допомогою спеціалізованого ПО окремих частин алгоритму декодування зображень на основі JPEG.*

### **Загальна постановка проблеми**

Алгоритм декодування на базі JPEG є не новиною. Але його виконання на апаратній частині визиває до себе інтерес. Сам алгоритм декодування є ніщо інше як зворотній алгоритм кодування JPEG. Але реалізація його полягає в декодуванні вихідного файлу з розширенням jpg. Ні для кого не секрет, що апаратне множення виконується швидше ніж програмне. А в даному випадку множення трапляється доволі часто. Тому було прийняте рішення про спробу генерування коду на апаратній мові VHDL за допомогою спеціалізованого ПО фірми Xilinx.

### **Постанова задач дослідження**

На даному етапі задача полягає в розробці алгоритму декодера зображень на базі JPEG за допомогою апаратної мови VHDL.

### **Рішення задач та результати досліджень**

1. Аналіз та вивчення необхідного для роботи ПО.

На початку роботи було встановлене необхідне ПО, AccelDSP, яке надає наступні можливості:

- читання та аналіз проекту MATLAB з плаваючою точкою;

- автоматичне створення еквівалентну проекту MATLAB з фіксованою точкою;
  - запуск моделювання MATLAB, щоб перевірити дизайн з фіксованою точкою;
  - створення синтезованих RTL HDL моделей та тестів для перевірки дизайну;
  - створення скриптів, які викликають інструменти, такі як HDL-симулятори, синтезатори RTL логіки та інструменти виконання Xilinx ISE.
2. Аналіз структури jpg-файлу та розробка моделі з плаваючою точкою у середовищі MATLAB.

Проаналізувавши структуру файлу формату jpg можна виявити проблемні та трудомісткі місця алгоритму декодування. Це множення на матриці дискретно-косинусного перетворення та квантування. Було прийнято рішення про реалізацію цих операцій на апаратурі – FPGA Xilinx Spartan 3e.

Розробка функції множення матриць представляє із себе основну частину синтезу апаратної моделі. За основу було взято два вектори, тому що вхідні дані повинні бути представлені у вигляді векторів. Наступний код показує м-функцію mul.m, розроблену в середовищі MATLAB.

```
function Y=mul(a,b)
m=length(b)/length(a);
n=length(a);
c=zeros(1,m);
for i=0:m-1
    c(i+1)=0;
    for j=0:n-1
        c(i+1)=c(i+1)+a(j+1)*b(i+j*m+1);
    end
end
Y=c;
End
```

Для прикладу було взято наступні вектори –  $(2\ 3)$  та  $(1111)$ . Відповідно результат дорівнює –  $(5,5)$ .

Згідно з вимогам програмного забезпечення AccelDSP було також розроблено код скрипту `mul_script.m`, який визиває всередині себе функцію вищого рівня `mul`:

```
a=load('vector.txt');
b=load('matr.txt');
for sample_num = 1:2
    % Call to hardware function
    X=mul(a,b);
end
% Display input data
figure(1)
plot(X);
```

Як бачимо спочатку зчитуються дані із текстових файлів, а потім передаються у головну функцію обчислення матриць.

У такому вигляді модель з плаваючою точкою, реалізована в MATLAB потрапляє до AccelDSP.

Далі у цьому середовищі потрібно виставити необхідні налаштування. Потрібно вибрати тип плати (Spartan 3e), для якої буде генеруватися апаратна модель, тип кристалу (fg320) та потік (ISE).

### 3. Отримання результатів синтезу VHDL-коду у середовищі Xilinx ISE

В результаті моделювання отримаємо готову RTL-модель у середовищі Xilinx ISE. В наступній таблиці представлено апаратні затрати щодо виконання даної апаратної моделі (Таблиця 1).

Перед тим як перелікувати використані ресурси, приведемо коротке їх описання.

Логічні ресурси організовані в «Слайс» (slices - шари), що складаються з таблиці перекодування LUT (lookup-table), мультиплексорів, блоку булевої логіки і блоку підсумовування / віднімання з формуванням ланцюга переносу. Чотири слайси

Таблица 1 – Апаратні затрати

Використання пристроїв взагалі (оціночні значення)			[-]
Логіка використання	Використаних	Доступних	Використання
Кількість слайсів	45	4656	0%
Кількість тригерів	65	9312	0%
Кількість 4 вхідних таблиць істиності	43	9312	0%
Кількість блоків входу\ виходу	34	232	14%
Кількість таймерів	1	24	4%

утворюють конфігуруємий логічний блок CLB (configurable logic block), який є базовим елементом для створення кінцевих автоматів, комбінаційної логіки, контролерів і послідовних схем.

Таким чином кількість використаних слайсів дорівнює 45, кількість тригерів – 65, таблиць перекодування – 43, блоків входу(виходу) – 34.

На рисунку 1 показано загальний вигляд блоку.

Входи а та b – вхідні вектори, що перемножуються, ас\_

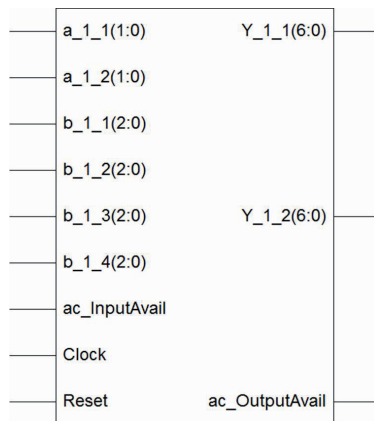


Рисунок 1 – Загальний вид апаратного модуля

InputAvail – вхід дозволу на операцію множення, Clock – такт, Reset – вхід скидання, as\_OutputAvail – дозвіл на вивід результату, Y - виходи результату.

На рисунку 2 показана більш детальна схема модуля.

Результати роботи апаратного модуля можна спостерігати за наступною часовою діаграмою на рисунку 3.

Як бачимо, результат є коректним. Алгоритм роботи послідовний і це можна віднести до його основного недоліку. Множення виконується за чотири такти – чотири операції множення. Ще одним недоліком є розрядна сітка. Даний конкретний приклад виконано для конкретних чисел. Подальша задача полягає в розробленні універсальної розрядної сітки для любых вхідних даних.

## Висновки

В цій статті розглянуто принцип формування моделі з плаваючою точкою у середовищі Matlab, яке перетворюється за допомогою AccelDSP у RLT-модель, яка в свою чергу синтезується у програмі Xilinx ISE, де і отримуються необхідні результати та часові діаграми. Подальша робота буде заключатися в реалізації повного алгоритму декодування зображень з використанням безпосередньо

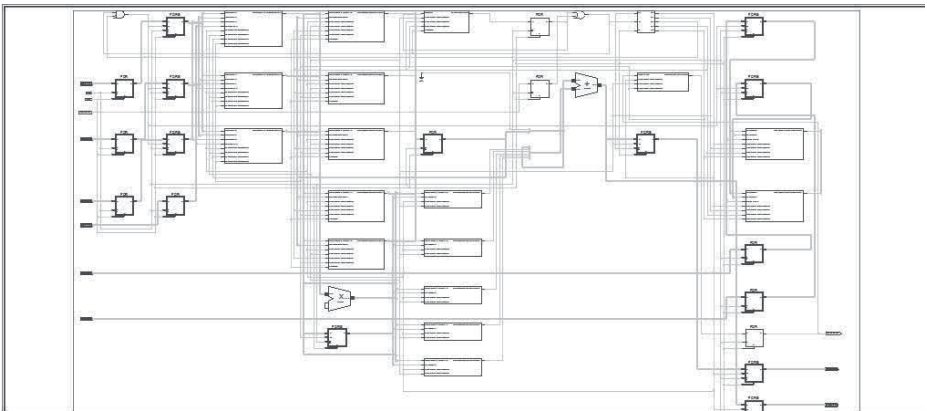


Рисунок 2 – Детальніша схема модуля

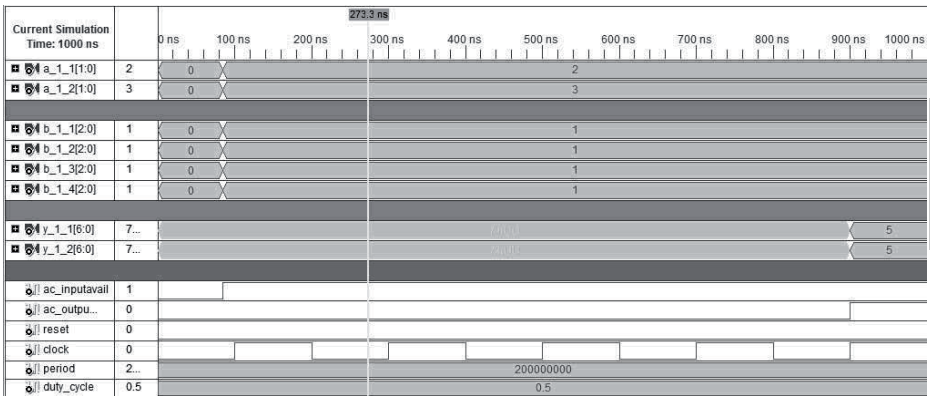


Рисунок 3 – Часова діаграма результатів множення векторів

відлагоджувальної плати.

## Література

- [1] AccelDSP Synthesis Tool. Офіційна документація по програмному забезпеченню - [електронний ресурс] - <http://www.xilinx.com/tools/acceldsp.htm>.
- [2] Декодування JPEG для чайників - [електронний ресурс] - <http://habrahabr.ru/blogs/algorithm/102521/>.
- [3] XtremeDSP Solutions Selection Guide - офіційне введення в організацію блоків DSP на FPGA
- [4] Бибило П.Н. Основы языка VHDL- М: Солон-Р, 2002 г. - 224 с.: ил.
- [5] Алгоритм JPEG - [електронний ресурс] - <http://www.univer.omsk.su/omsk/Edu/infpro/1/jpeg/algcz2.htm>
- [6] Языки проектирования на ПЛИС - [електронний ресурс] - <http://electronix.ru/forum/lofiversion/index.php/t62828.html>
- [7] Консультативний центр Matlab - [електронний ресурс] - <http://matlab.exponenta.ru/matlab/default.php>