

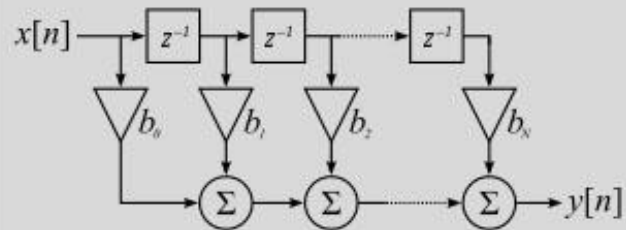
Прикарпатський національний університет
імені Василя Стефаника
Кафедра комп'ютерної інженерії та електроніки

Реалізація алгоритму швидкого перетворення Фур'є на ПЛІС

Андрій Павлишин, Володимир Грига, Володимир Мандзюк

Засоби обробки сигналів

Програмний фільтр



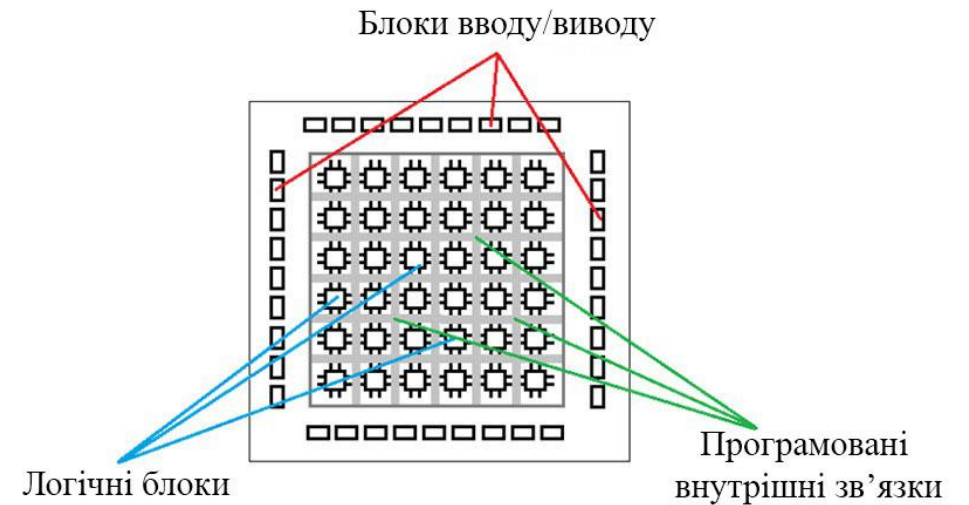
```
// set coefficients per tap
// tap=0 to (numCoeffs-1)
// numCoeffs = number_of_taps + 1
// offset[0],
// tapCoeffs[0],
// tapCoeffs[1],
// ...
// tapCoeffs[numCoeffs-1]

// set coefficients per tap
// tap=0 to (numCoeffs-1)
// numCoeffs = number_of_taps + 1
// offset[0],
// tapCoeffs[0],
// tapCoeffs[1],
// ...
// tapCoeffs[numCoeffs-1]

// set coefficients per tap
// tap=0 to (numCoeffs-1)
// numCoeffs = number_of_taps + 1
// offset[0],
// tapCoeffs[0],
// tapCoeffs[1],
// ...
// tapCoeffs[numCoeffs-1]

// set coefficients per tap
// tap=0 to (numCoeffs-1)
// numCoeffs = number_of_taps + 1
// offset[0],
// tapCoeffs[0],
// tapCoeffs[1],
// ...
// tapCoeffs[numCoeffs-1]
```

Структура FPGA-пристрою



Дискретне перетворення Фур'є

Визначення дискретного перетворення Фур'є:

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{2\pi i}{N}nm}$$

де

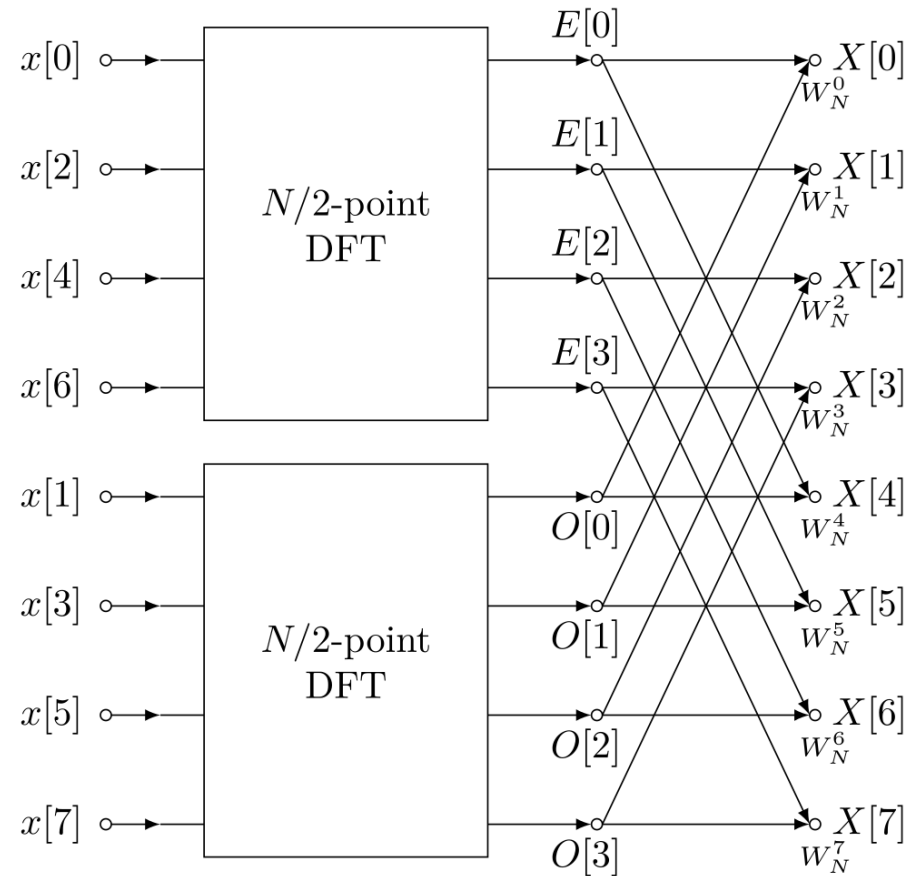
$\{x_n\} = x_0, x_1, \dots, x_{N-1}$ — дискретизовані вхідні дані

$\{X_k\} = X_0, X_1, \dots, X_{k-1}$ — результат перетворення, частотне представлення

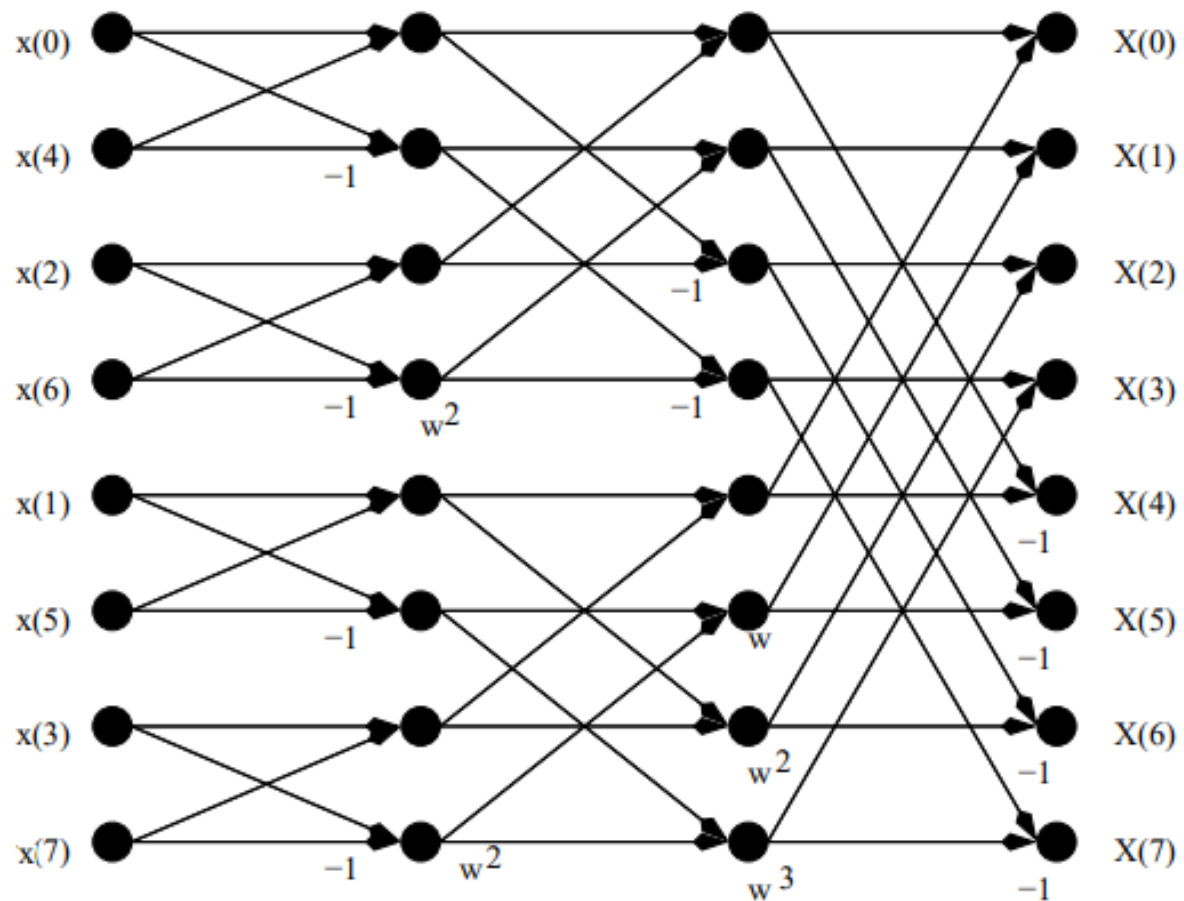
даних

$W_N^{nm} = e^{-\frac{2\pi i}{N}nm}$ — комплексний коефіцієнт.

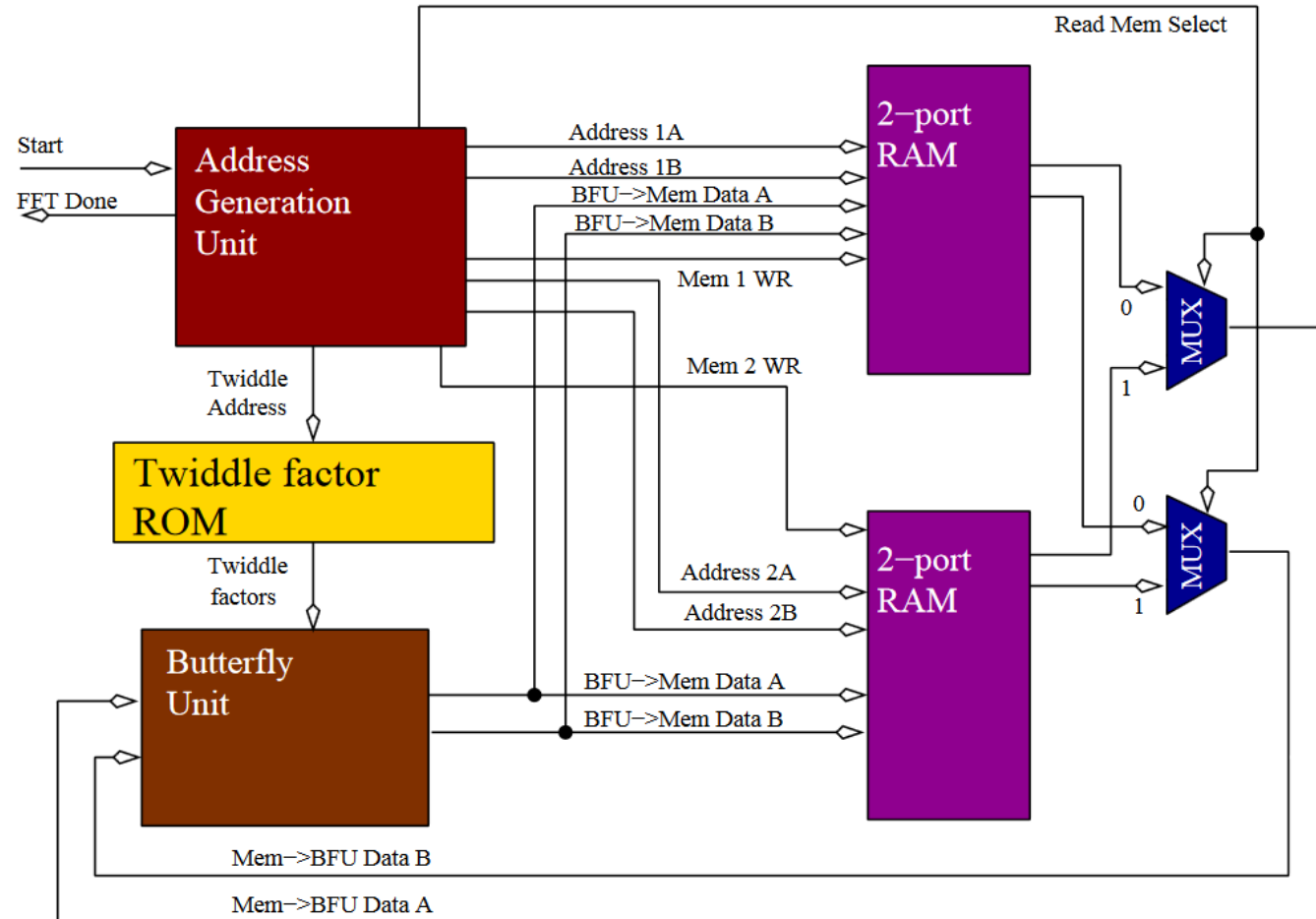
Розбиття дискретного перетворення Фур'є на два менших



Приклад перетворення ШПФ для 8-точок



Структура апаратного перетворення Фур'є.



Використаний формат чисел

При реалізації швидкого алгоритму обробки сигналу на апаратній частині, числа з плаваючою комою мають декілька недоліків:

- Числа з плаваючою комою мають велику розрядність і займають багато комірок пам'яті.
- Арифметичні операції з числами з плаваючою комою набагато складніші ніж з фіксованою комою або цілими числами. Потрібно багато логічних комірок.
- Швидкість та затримка погіршуються через додаткову складність логіки.
- Оскільки оцифровані сигнали мають фіксовану розрядність, числа з плаваючою комою не мають переваг в обробці.

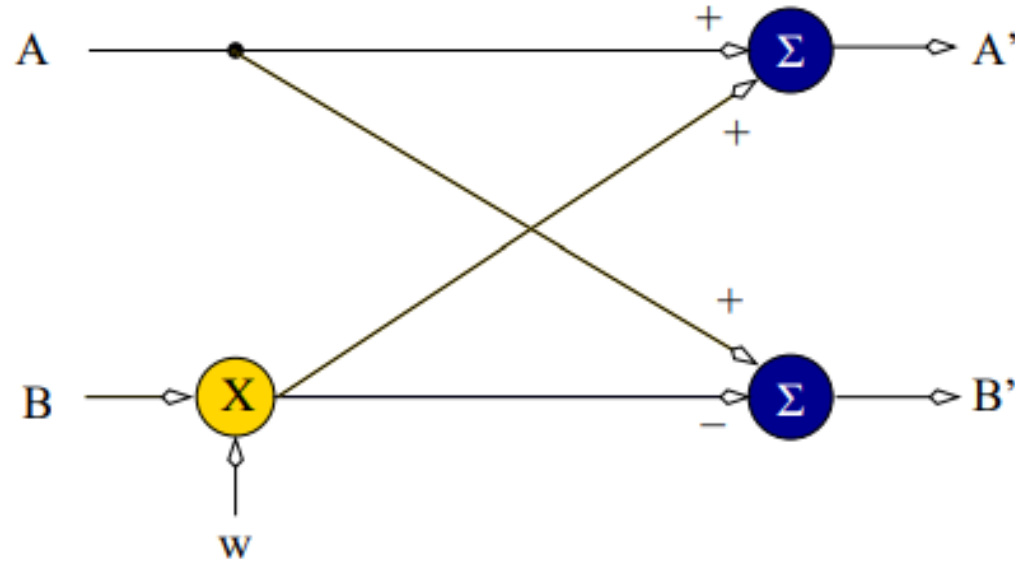
Тому тут використовуються 16-розрядні знакові числа з 12-бітними початковими даними, які мають діапазон значень від -1024 до 1024.

Привести уявну та комплексну компоненту експоненти до цілочисельного формату, описаного вище, можна наступним чином:

```
real_part = int(cos(2*PI*k/N)*1024)
```

```
img_part = int(-sin(2*PI*k/N)*1024)
```

Діаграма операції “метелика”

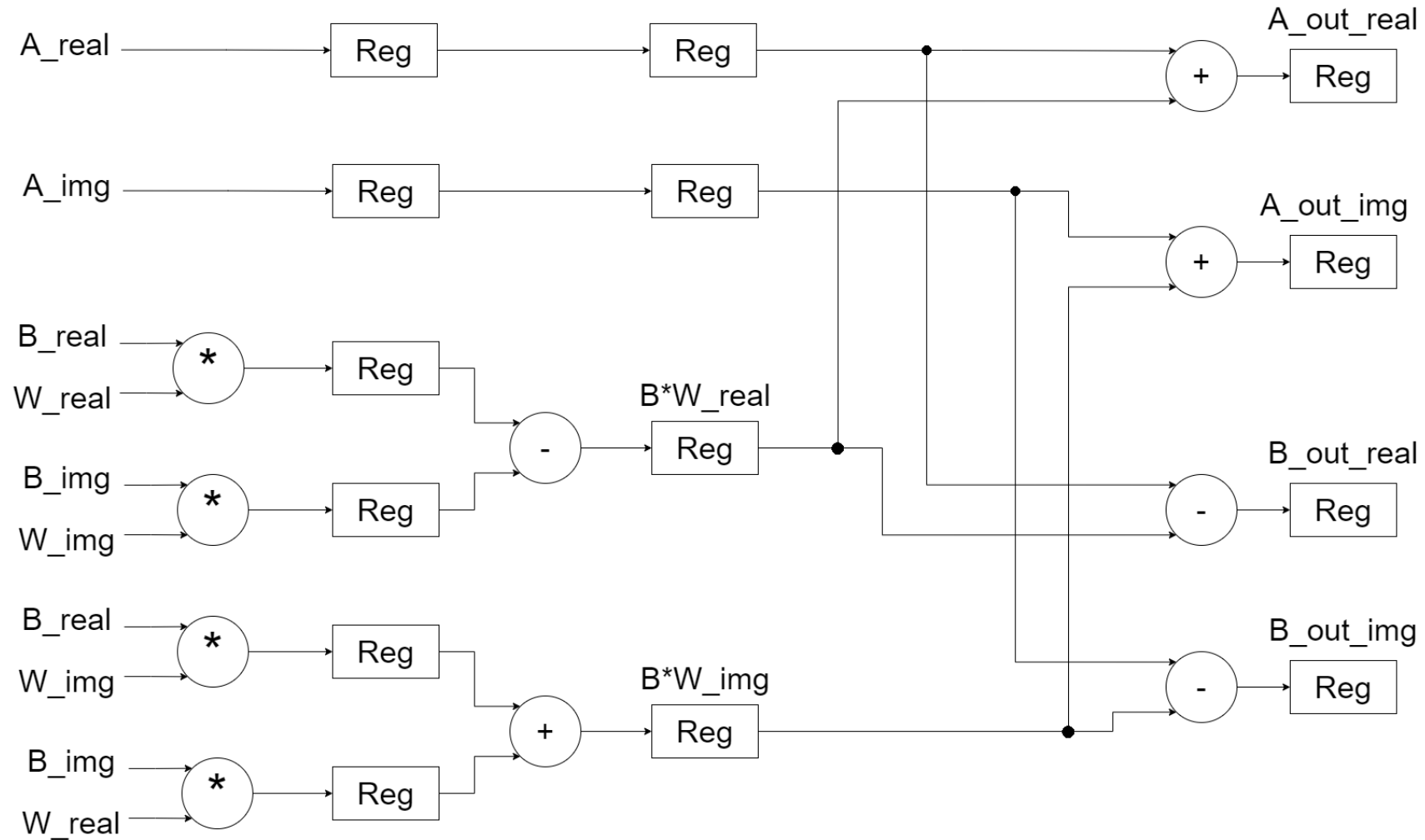


$$A' = A + B \cdot W$$

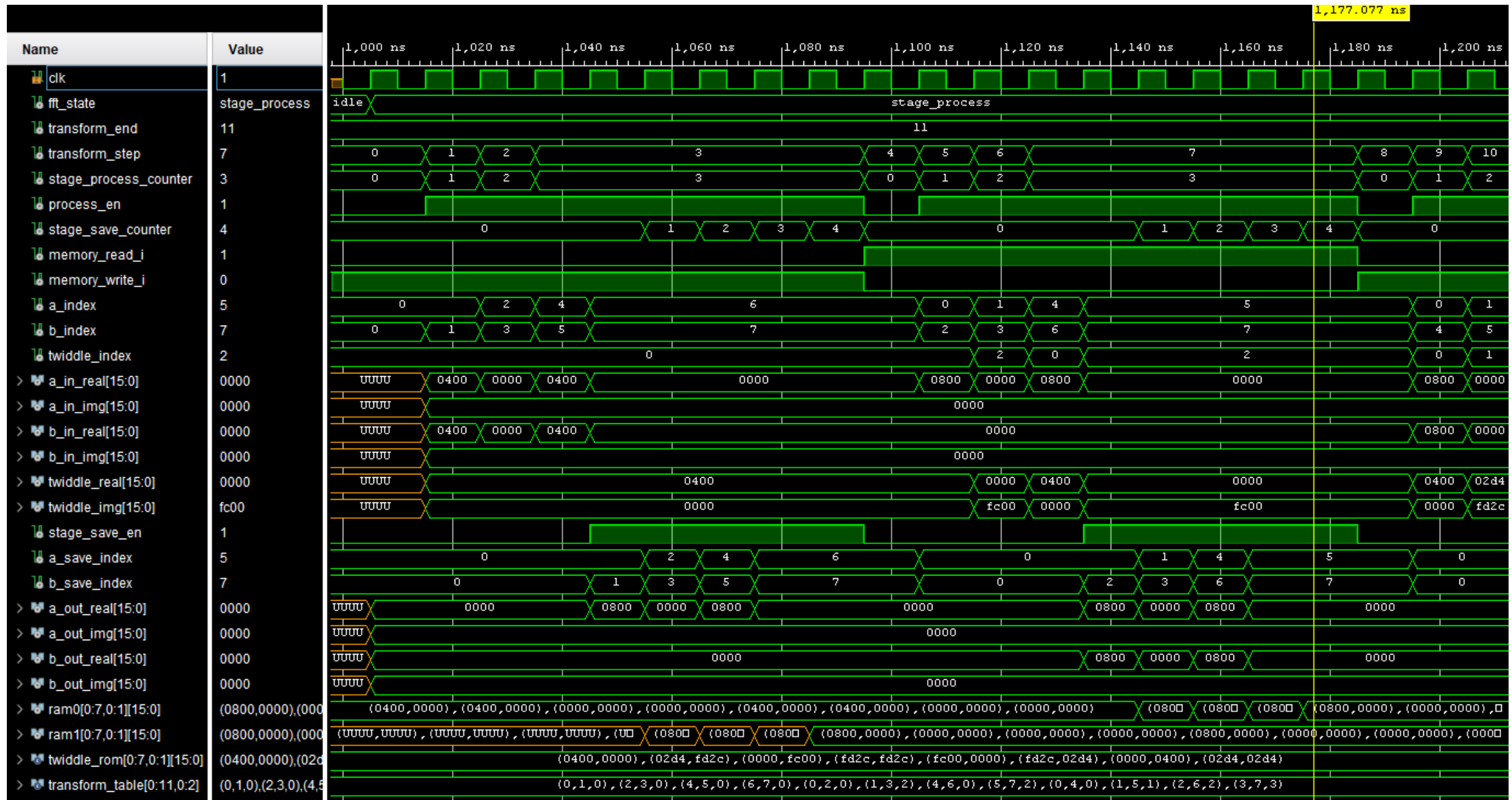
$$B' = A - B \cdot W$$

$$\begin{aligned} B * W &= (B_{real} + B_{img}) \cdot (W_{real} + W_{img}) = \\ &= (B_{real} \cdot W_{real} - B_{img} \cdot W_{img}) + (B_{real} \cdot W_{img} + B_{img} \cdot W_{real}) \end{aligned}$$

Блок-діаграма реалізованого модуля операції “метелик”



Процес роботи модуля ШПФ



Часові характеристики

Розмір перетворення	Час перетворення
8	270нс
16	520нс
32	1мкс
64	2.2мкс
128	4.8мкс
256	10.6мкс
512	23мкс
1024	51мкс

Для прикладу, популярний мікроконтролер ESP32, зі стандартною тактовою частотою 240МГц, виконує швидке перетворення Фур'є розміром 1024 точок за 1.5мс.

Перспективи подальших досліджень

- Заміна модуля ROM пам'яті, який зберігає експонентні коефіцієнти та збільшується із розміром перетворення, на модуль, який їх обчислює в реальному часі та має фіксований розмір.
- Використання схеми з лише одним блоком RAM-пам'яті, що призведе до зниження швидкодії перетворення, але і зменшить потреби оперативної пам'яті вдвічі.

Висновки

- Виконано апаратну реалізацію алгоритму швидкого перетворення Фур'є на мові VHDL та проведено синтез на базі ПЛІС сімейства Artix-7 (Xilinx) з використанням САПР Vivado IDE.
- Встановлено, що завдяки конвеєризації та паралельному виконанню операцій реалізація апаратного швидкого перетворення Фур'є відбувається в 30 разів швидше, ніж його виконання на мікроконтролері чи мікропроцесорі загального призначення.
- Запропоновано варіанти для вдосконалення структурної схеми пристрою з метою зменшення ресурсів, необхідних для реалізації перетворення.

Дякую за увагу!