Frequency Monitor

Ali Hakim Taşkiran[1]

[1]Department of Electrical and Electronics Engineering, Bilkent University, 06800, Ankara, Turkey

1.Objective

The objective of the project is to decompose signals into harmonic components as fast as roughly few nanoseconds. This provides a significant acceleration, compared to CPU.

2 Methodology

The project mainly based on hardware acceleration of DFT, complex amplitude computer and some additional peripheral components such analog audio circuitry, audio sensing, PWM generator and LED's. The modularized steps of the project is given Fig 2.1.

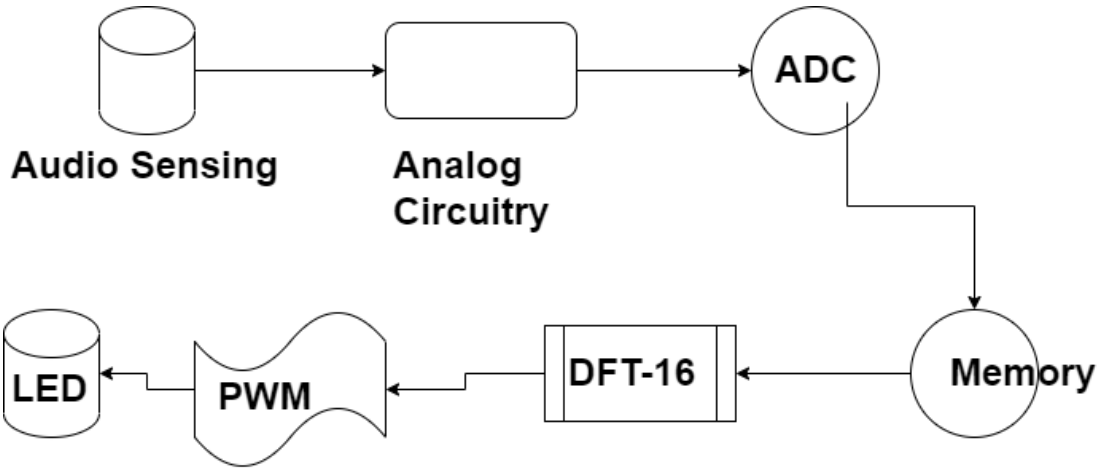

Fig 2.1: Chunks of project

2.1 Audio Sensing

Acoustic waves create a pressure gradient in the medium. The gradient is detected piezoelectric sensors.

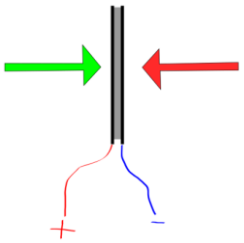

Fig 2.2: Piezoelectric sensor under forces

$$| F | = \iint \Delta P(x, y) dA$$

The voltage generated by the force is given by

$$V = Ftd/(\epsilon A)$$

where V is voltage generated by piezo, F is net force applied on piezo, t is thickness of piezo, d is piezoelectric constant of the material, ϵ is dielectric constant of the material and A is area of the

piezo [1]. The sensor generates a voltage between 0-14 mV. The experiment conducted on a loudy railway and created 14 mV peak voltage while the train pass on it [2].

Fig 2.3: Dual channel piezo sensors

2.2 Analog Audio Amplifier

Analog amplifier was created with an opamp. Incoming signal is amplified about approximately 101 times so as to amplify 0-14 mV piezo output[7] to 0-1 Volts range of Basys3 analog input.

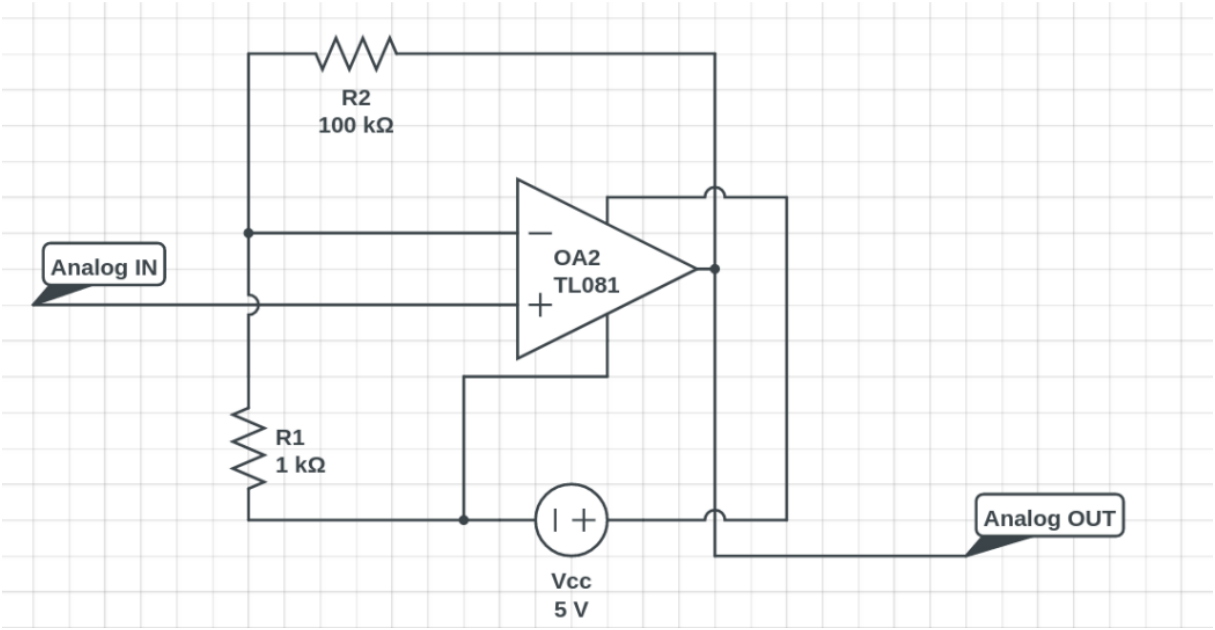

Fig 2.4: Circuit Scheme of Analog Amplifier

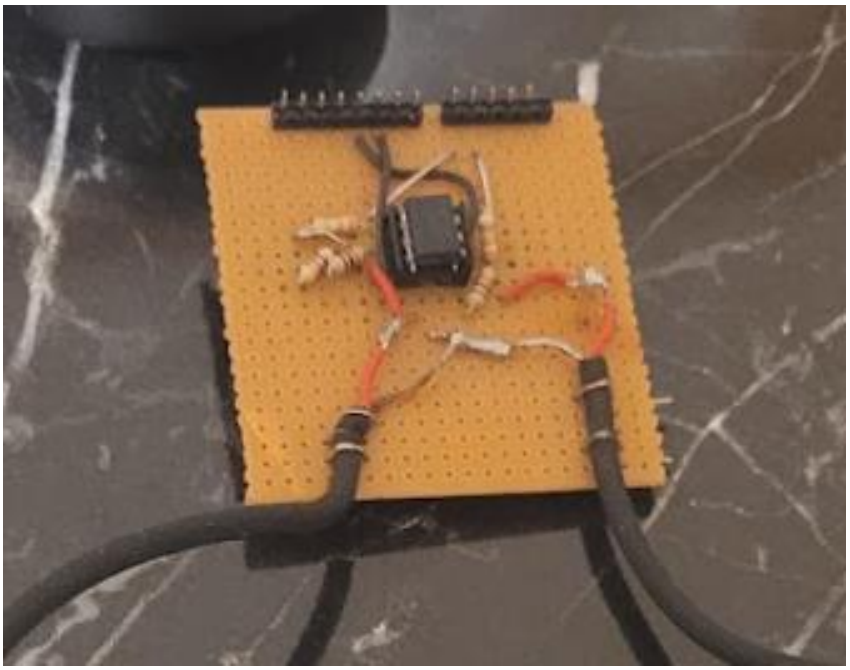

Fig 2.5: Analog audio amplifier

2.3 Fourier Transformer Hardware Accelerator

This part is designed to significantly accelerate Fourier transform. It has dedicated hardware engines to perform operations, unlike traditional CPU computing which use shared resources for different operations. The engine performs radix-16 butterfly graph computation for further scalable FFT computations.

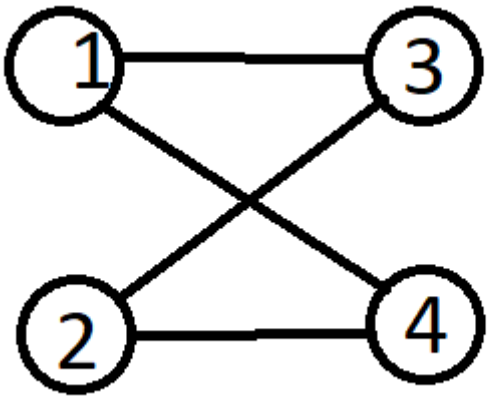

Fig 2.6: Radix-2 Butterfly

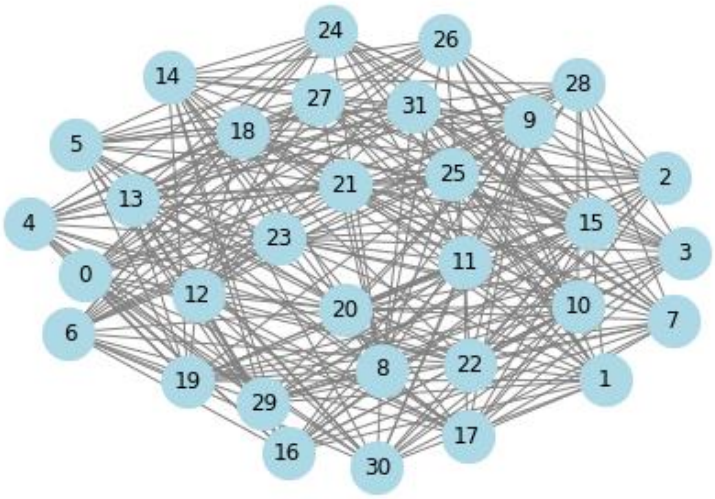

Fig 2.7: Radix-16 Butterfly

Butterfly implementation is key to an efficient Fast Fourier Transform (FFT) computation. The computation structure is given this name because the data flow diagram of this operation resembles a butterfly. In a radix-2 butterfly, two input points result in two output points. The operation comprises a complex addition, a complex subtraction, and a multiplication with a twiddle factor (a complex exponential function). The twiddle factor accounts for the rotation of the complex plane. A radix-2 FFT implementation breaks down a DFT (Discrete Fourier Transform) into smaller DFTs, recursively applying the radix-2 butterfly operation. However, higher radix butterflies like radix-16 are used to further optimize and reduce the computational complexity. The radix-16 butterfly takes 16 input points and gives 16 output points. The benefit is a reduction in the total number of computations required for the FFT. The Fourier Transform Hardware Accelerator is designed to perform these radix-16 butterfly operations efficiently. By implementing these computations in hardware, instead of relying on the CPU, the accelerator can significantly increase the speed of FFT calculations. This is particularly beneficial for applications where FFTs are used extensively, such as digital signal processing, image processing, and solving partial differential equations. These dedicated hardware engines can process these computations in parallel and at high speed, thus providing a significant improvement over traditional CPU computation, which must use shared resources and manage other processes concurrently. When designing and implementing a Fourier Transform Hardware Accelerator, several factors need to be considered such as the choice of radix, the memory architecture for storing intermediate results, how to handle the complex multiplications with the twiddle factors, and the overall system integration with other components. It should be noted that while radix-16 FFTs reduce computation, they also introduce more complexity and require more resources for implementation, particularly in terms of memory and multiplicative operations. However, with careful design and optimization, these challenges can be successfully addressed.

Implementation of radix-16 butterfly is done by hard coding twiddle factors and creating a matrix multiplication unit that handle complex number. This approach eliminates memory latency by storing non-volatile elements in LUT's (look up tables) and reduces the need for frequent data access from the main memory. By precomputing the twiddle factors and storing them in lookup tables, the computation process can be sped up considerably, since these complex exponential values are used repetitively in the Fourier Transform calculations. This method, however, requires careful planning and allocation of hardware resources to manage the larger look-up tables required for radix-16 butterflies as compared to radix-2.

The matrix multiplication unit is specifically designed to handle complex numbers, which are inherent in Fourier Transform computations. Complex multiplication, unlike real number multiplication, involves more computations due to the presence of real and imaginary parts. Designing a specific unit for this operation ensures that the complexities associated with these operations are handled efficiently, further enhancing the computational speed. The Fourier Transform Hardware Accelerator also leverages the benefits of parallel processing. In traditional CPU computing, operations are typically performed serially. However, the dedicated hardware engines in the accelerator can perform multiple operations simultaneously, taking full advantage of the inherent parallelism of butterfly operation. To minimize data traffic and ensure efficient utilization of resources, the memory architecture of the accelerator is meticulously designed. It needs to handle the storing and retrieving of intermediate results during the computation and manage the larger data requirements of the radix-16 butterfly implementation. Memory hierarchies are commonly used, with data caching strategies and data locality optimizations implemented to ensure minimal latency and maximized throughput.

However, despite its advantages, the implementation of a radix-16 butterfly FFT does bring challenges. The increased complexity and the larger resource requirement necessitate a trade-off between speed and system cost. Furthermore, integrating the hardware accelerator into an existing system requires careful consideration to ensure that it operates seamlessly with the other system components and doesn't bottleneck the overall system performance.

2.4 PWM and LED

LED are used to show magnitude of magnitude of each bin of frequency domain. In project proposal, I proposed to use 8 LED's and 8 PWM modules. However, due to infeasibilities of processing real world audio with limited resolution, 2.5 KHz/bin, I decided to omit the part associated to import audio into device and increase complexity of Fourier transform unit (from 8 bins to 16 bins) and increase number of LEDs not to reduce provided complexity of the project in proposal.

PWM signal generator implementation is done by incrementing a counter in each clock cycle until it reaches the pulse width then toggling state. Each LED is controlled by individual PWM and they don't interfere each other. LEDs are on-board 16-LEDs. This provides more robust implementation, as jumper connections prone to mechanical deformations.

3 Design Specifications

The design includes several hardware engines and modules for performing the operations.

3.1 Complex Multiplication Unit

Complex multiplier performs multiplication of two 64 bits complex numbers (real int32 + imaginary int32). This module instantiated 256 times in the other entities.

Port Component	Type	Description
Xre	32 bits Integer, IN	Real part of first multiplicand
Xim	32 bits Integer, IN	Imaginary part of first multiplicand
Yre	32 bits Integer, IN	Real part of second multiplicand
Yim	32 bits Integer, IN	Imaginary part of second multiplicand
Zre	32 bits Integer, OUT	Real part of product
Zim	32 bits Integer, OUT	Imaginary part of product

Table 3.1: Input-Output Terminals of Complex Multiplier Unit

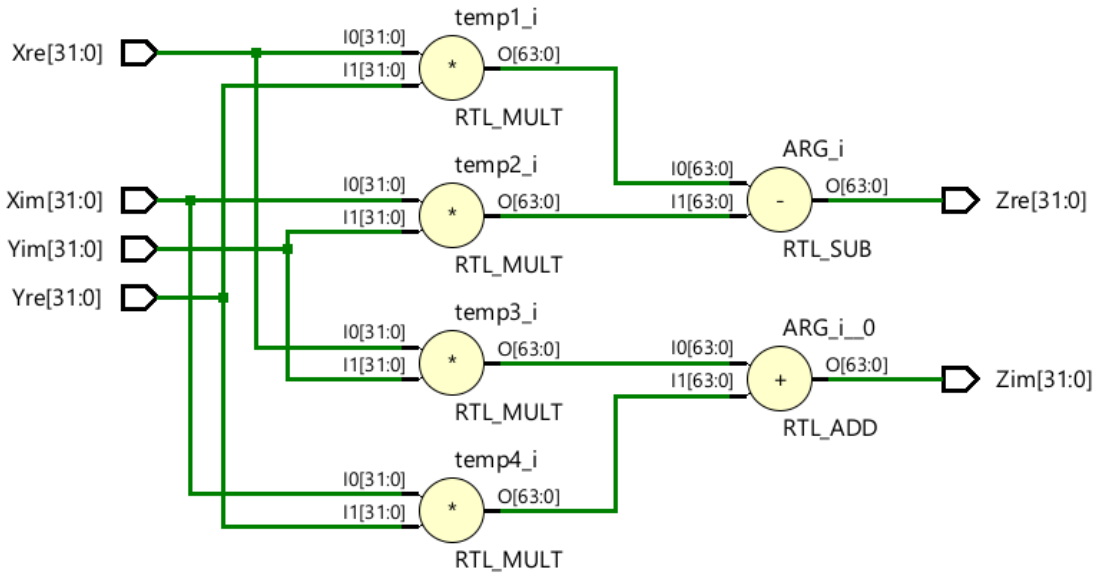

Fig 3.1: RTL schematics of Complex Multiplier

Overall, it performs Zre=Xre*Yre - Xim*Yim , Zim=Xre*Yim + Xim*Yre simultaneously.

3.2 Radix-16 Butterfly (DFT16):

This hardware engine computes butterfly graphs with highly parallelized architecture and eliminate memory bottleneck by storing precomputed twiddles in LUT. It performs 256 complex multiplications namely execute 256 adder, 256 subtractor and 1024 multiplier simultaneously.

Port Component	**Type**	**Description**
A_re0	Integer (32 bits), IN	Sequential component streamed in computation
A_im0	Integer (32 bits), IN	Sequential component streamed in computation
A_re1	Integer (32 bits), IN	Sequential component streamed in computation
A_im1	Integer (32 bits), IN	Sequential component streamed in computation
A_re2	Integer (32 bits), IN	Sequential component streamed in computation
A_im2	Integer (32 bits), IN	Sequential component streamed in computation
A_re3	Integer (32 bits), IN	Sequential component streamed in computation
A_im3	Integer (32 bits), IN	Sequential component streamed in computation
A_re4	Integer (32 bits), IN	Sequential component streamed in computation
A_im4	Integer (32 bits), IN	Sequential component streamed in computation
A_re5	Integer (32 bits), IN	Sequential component streamed in computation
A_im5	Integer (32 bits), IN	Sequential component streamed in computation

A_re6	Integer (32 bits), IN	Sequential component streamed in computation
A_im6	Integer (32 bits), IN	Sequential component streamed in computation
A_re7	Integer (32 bits), IN	Sequential component streamed in computation
A_im7	Integer (32 bits), IN	Sequential component streamed in computation
A_re8	Integer (32 bits), IN	Sequential component streamed in computation
A_im8	Integer (32 bits), IN	Sequential component streamed in computation
A_re9	Integer (32 bits), IN	Sequential component streamed in computation
A_im9	Integer (32 bits), IN	Sequential component streamed in computation
A_re10	Integer (32 bits), IN	Sequential component streamed in computation
A_im10	Integer (32 bits), IN	Sequential component streamed in computation
A_re11	Integer (32 bits), IN	Sequential component streamed in computation
A_im11	Integer (32 bits), IN	Sequential component streamed in computation
A_re12	Integer (32 bits), IN	Sequential component streamed in computation
A_im12	Integer (32 bits), IN	Sequential component streamed in computation
A_re13	Integer (32 bits), IN	Sequential component streamed in computation
A_im13	Integer (32 bits), IN	Sequential component streamed in computation
A_re14	Integer (32 bits), IN	Sequential component streamed in computation
A_im14	Integer (32 bits), IN	Sequential component streamed in computation
A_re15	Integer (32 bits), IN	Sequential component streamed in computation
A_im15	Integer (32 bits), IN	Sequential component streamed in computation
C_re0	Integer (32 bits), IN	Sequential component result of computation
C_im0	Integer (32 bits), IN	Sequential component result of computation
C_re1	Integer (32 bits), IN	Sequential component result of computation
C_im1	Integer (32 bits), IN	Sequential component result of computation
C_re2	Integer (32 bits), IN	Sequential component result of computation

C_im2	Integer (32 bits), IN	Sequential component result of computation
C_re3	Integer (32 bits), IN	Sequential component result of computation
C_im3	Integer (32 bits), IN	Sequential component result of computation
C_re4	Integer (32 bits), IN	Sequential component result of computation
C_im4	Integer (32 bits), IN	Sequential component result of computation
Cre5	Integer (32 bits), IN	Sequential component result of computation
C_im5	Integer (32 bits), IN	Sequential component result of computation
C_re6	Integer (32 bits), IN	Sequential component result of computation
C_im6	Integer (32 bits), IN	Sequential component result of computation
C_re7	Integer (32 bits), IN	Sequential component result of computation
C_im7	Integer (32 bits), IN	Sequential component result of computation
C_re8	Integer (32 bits), IN	Sequential component result of computation
C_im8	Integer (32 bits), IN	Sequential component result of computation
C_re9	Integer (32 bits), IN	Sequential component result of computation
C_im9	Integer (32 bits), IN	Sequential component result of computation
C_re10	Integer (32 bits), IN	Sequential component result of computation
C_im10	Integer (32 bits), IN	Sequential component result of computation
C_re11	Integer (32 bits), IN	Sequential component result of computation
C_im11	Integer (32 bits), IN	Sequential component result of computation
C_re12	Integer (32 bits), IN	Sequential component result of computation Sequential component result of computation
C_im12	Integer (32 bits), IN	Sequential component result of computation
C_re13	Integer (32 bits), IN	Sequential component result of computation
C_im13	Integer (32 bits), IN	Sequential component result of computation
C_re14	Integer (32 bits), IN	Sequential component result of computation

C_im14	Integer (32 bits), IN	Sequential component result of computation
C_re15	Integer (32 bits), IN	Sequential component result of computation
C_im15	Integer (32 bits), IN	Sequential component result of computation

Table 3.2: Input-Output Terminals of Radix-16 Butterfly

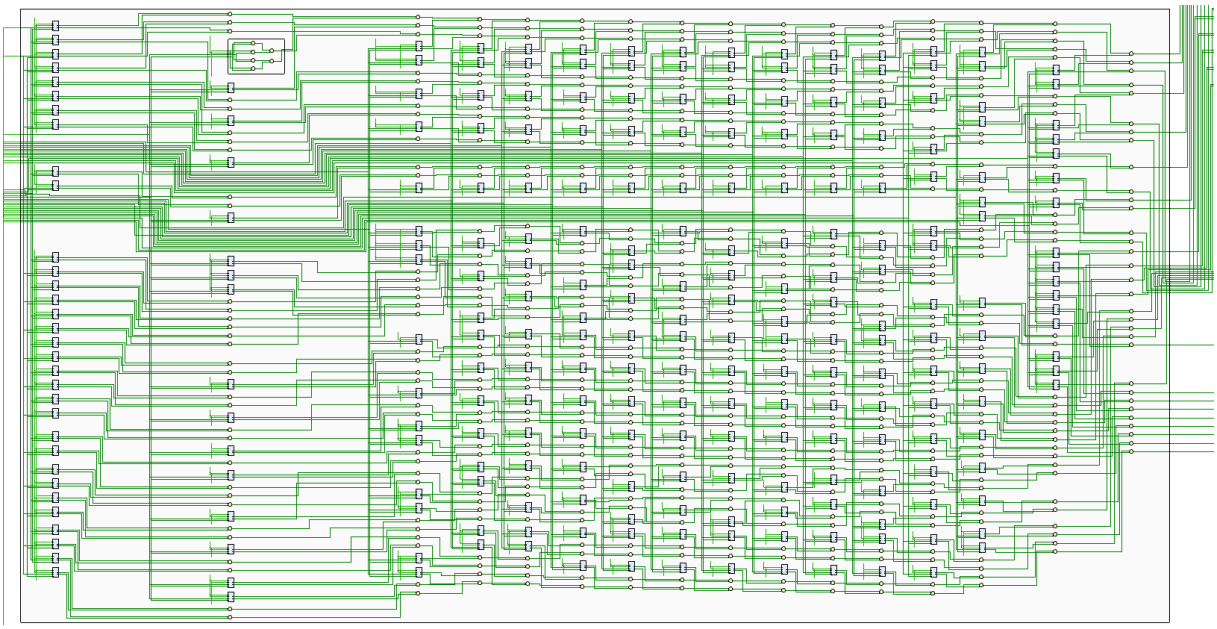

Fig 3.2: RTL schematics of Radix-16 Butterfly Graph Hardware Engine

3.3 Magnituder&Rescaler

This unit is not explicitly defined as a module, rather it is a primitive form. So, it doesn't have any port to describe. It computes squared magnitudes of result of DFT, then scales it by dividing by 52428 to ensure smooth functioning of PWM module. The end product of this hardware engine is pulse width of PWM.

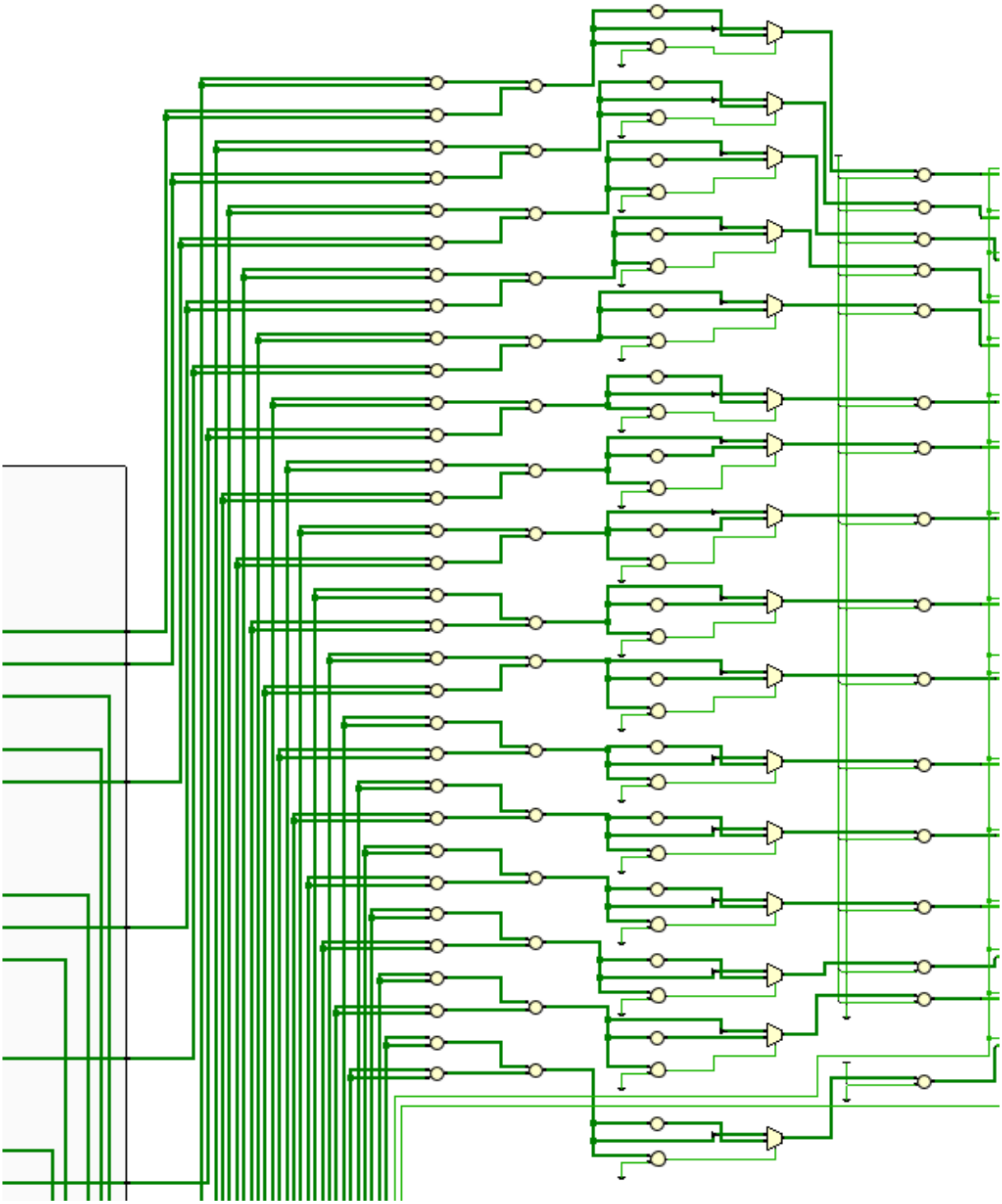

Fig 3.3: RTL Schematics of Magnituder&Rescaler

3.4 PWM Signal Generator

This module generates PWM signal upon the fed pulse width. PWM resolution of 1000000. It was carefully fine-tuned to display best results for human eye.

Port Component	Type	Description
clk	std_logic, IN	Clock signal
duty_cycle	Integer range 0 to 1000000, IN	Specify pulse width of square wave. Pulse_width=duty_cycle/1000000
reset	std_logic, IN	Reset the counter
pwm_out	std_logic, OUT	output

Table 3.3: Input-Output Terminals of PWM module

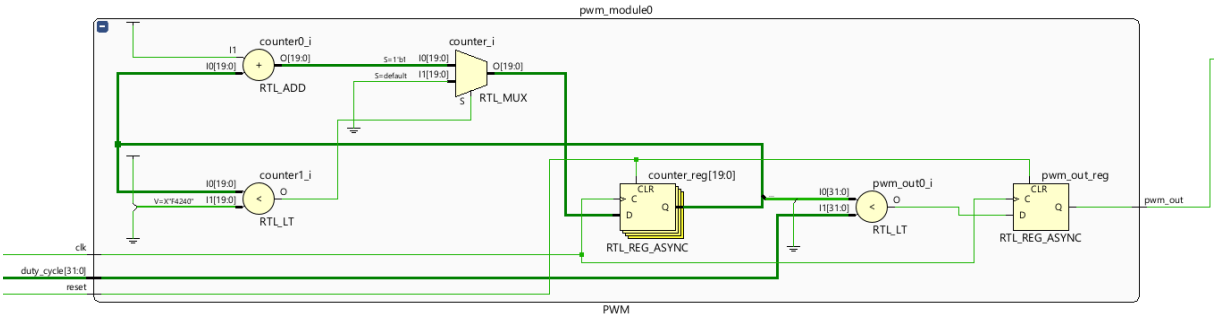

Figure 3.4: RTL schematics of PWM module

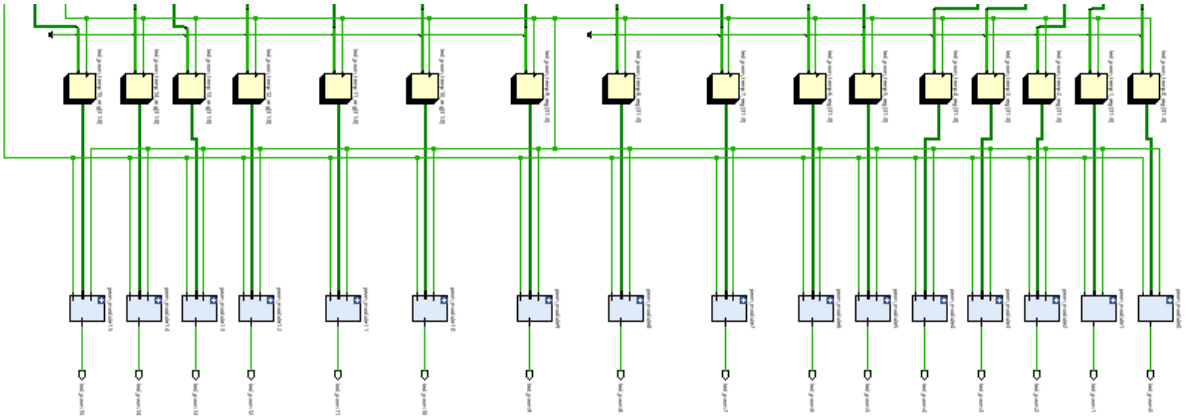

Figure 3.4: RTL schematics of LED drivers

3.5 Main (Top level entity)

Finally, all necessary components are ready. This entity includes all former modules and provide a frame for them to work. It has clk and reset inputs and it also control 16-LEDs. As mentioned in section 2.4, due to practical limitations, low-resolution, external world interactions are limited and the Fourier transformed signal is hardcoded in this module in LUTs.

Port Component	Type	Description
clk	std_logic, IN	Clock
reset	std_logic, IN	Resets the module
led_pwm0	Std_logic, OUT	Control the LED 0
led_pwm1	Std_logic, OUT	Control the LED 1
led_pwm2	Std_logic, OUT	Control the LED 2
led_pwm3	Std_logic, OUT	Control the LED 3
led_pwm4	Std_logic, OUT	Control the LED 4
led_pwm5	Std_logic, OUT	Control the LED 5
led_pwm6	Std_logic, OUT	Control the LED 6

led_pwm7	Std_logic, OUT	Control the LED 7
led_pwm8	Std_logic, OUT	Control the LED 8
led_pwm9	Std_logic, OUT	Control the LED 9
led_pwm10	Std_logic, OUT	Control the LED 10
led_pwm11	Std_logic, OUT	Control the LED 11
led_pwm12	Std_logic, OUT	Control the LED 12
led_pwm13	Std_logic, OUT	Control the LED 13
led_pwm14	Std_logic, OUT	Control the LED 14
led_pwm15	Std_logic, OUT	Control the LED 15

Table 3.4: Input-Output Terminals of main

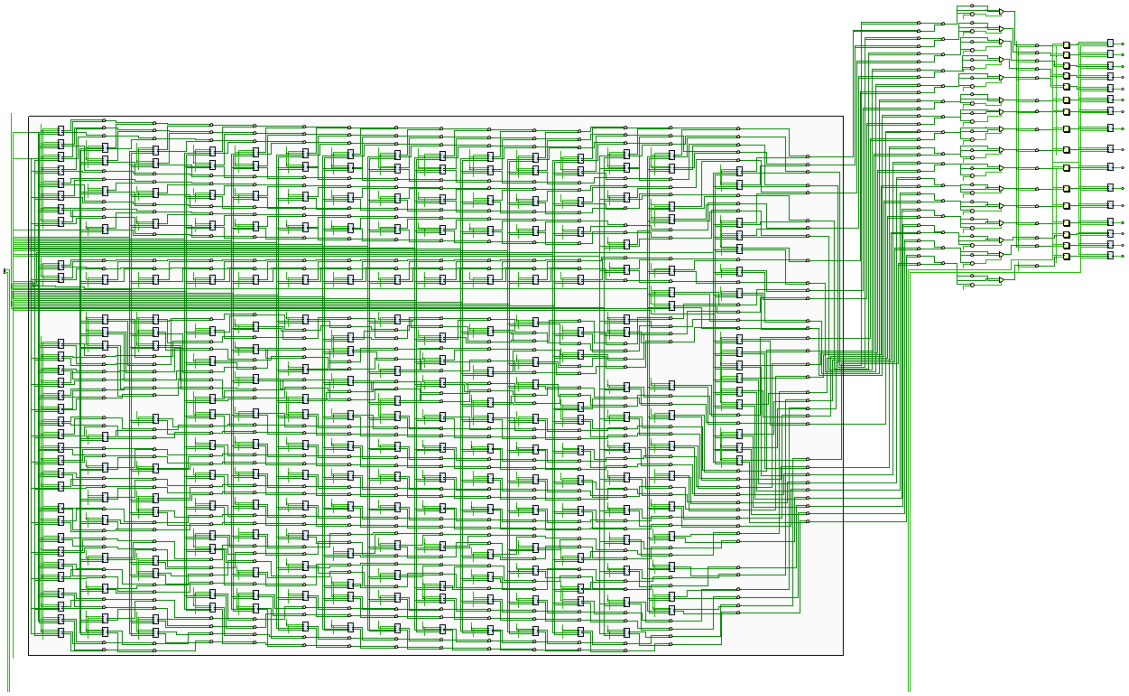

Figure 3.5: RTL schematics of main module

3.6 Physical Design

Physical configuration of abstract entities depicted Fig 3.6. Starting from LED 0, until LED 15; PWM encoded scaled dot products are displayed on LED array.

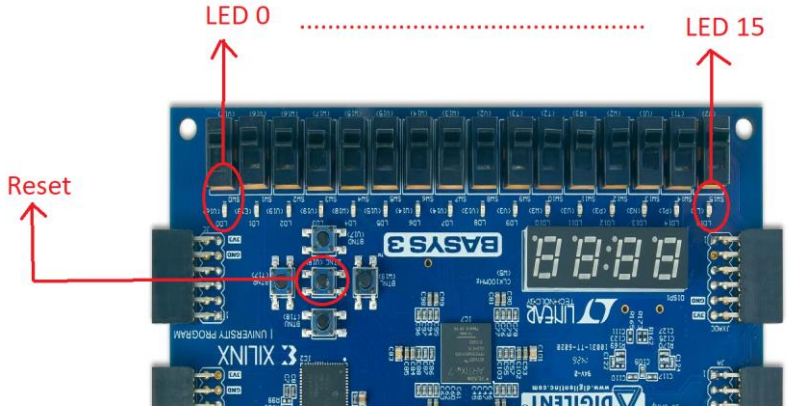

Fig 3.6: Physical configuration

4. Results

The design successfully verified in real hardware and test bench.

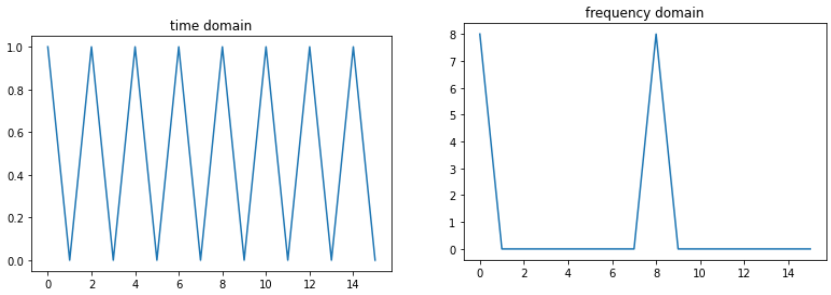

Fig 4.1: Test signal

Fig 4.2: Real Hardware Implemention

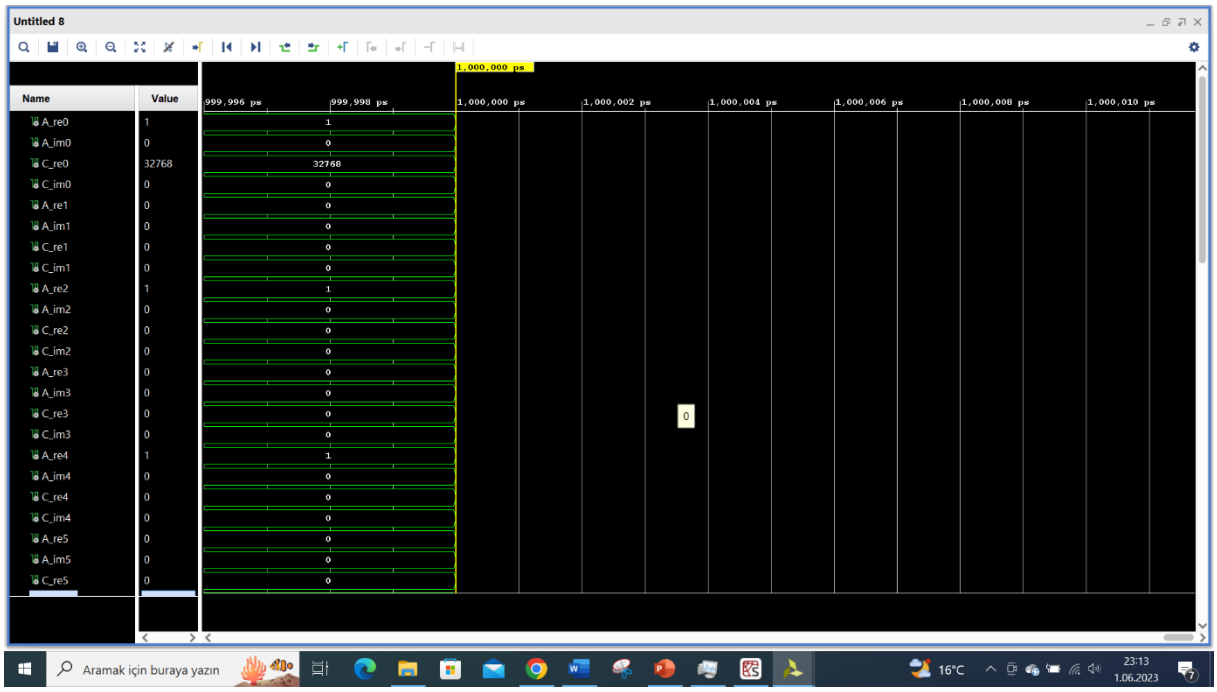

Fig 4.3: TestBench of signal given in Fig 4.1

Fig 4.4: TestBench of signal given in Fig 4.1 (continued-1)

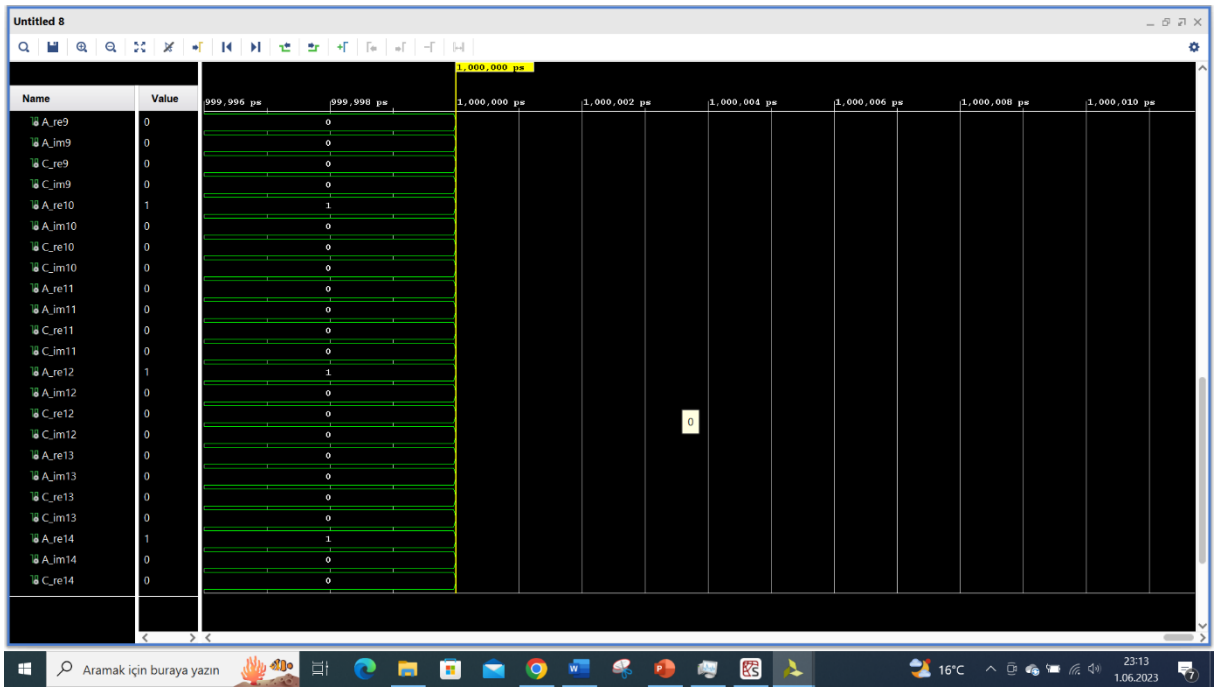

Fig 4.5: TestBench of signal given in Fig 4.1 (continued-2)

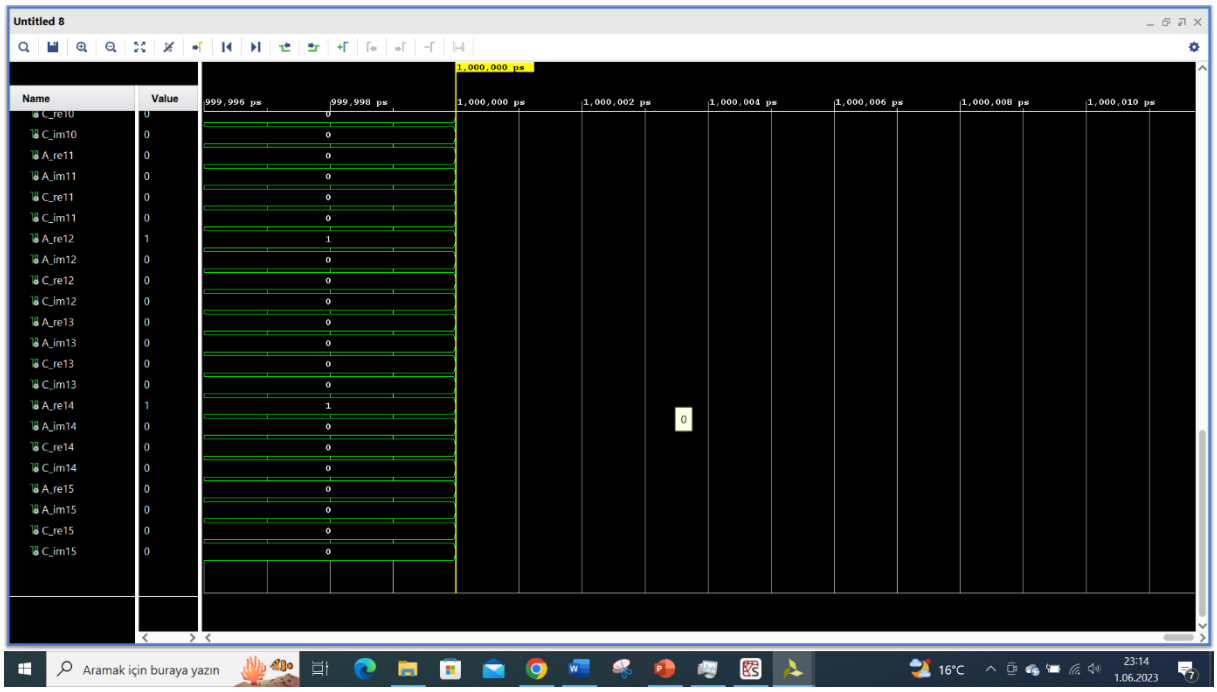

Fig 4.6: TestBench of signal given in Fig 4.1 (continued-3)

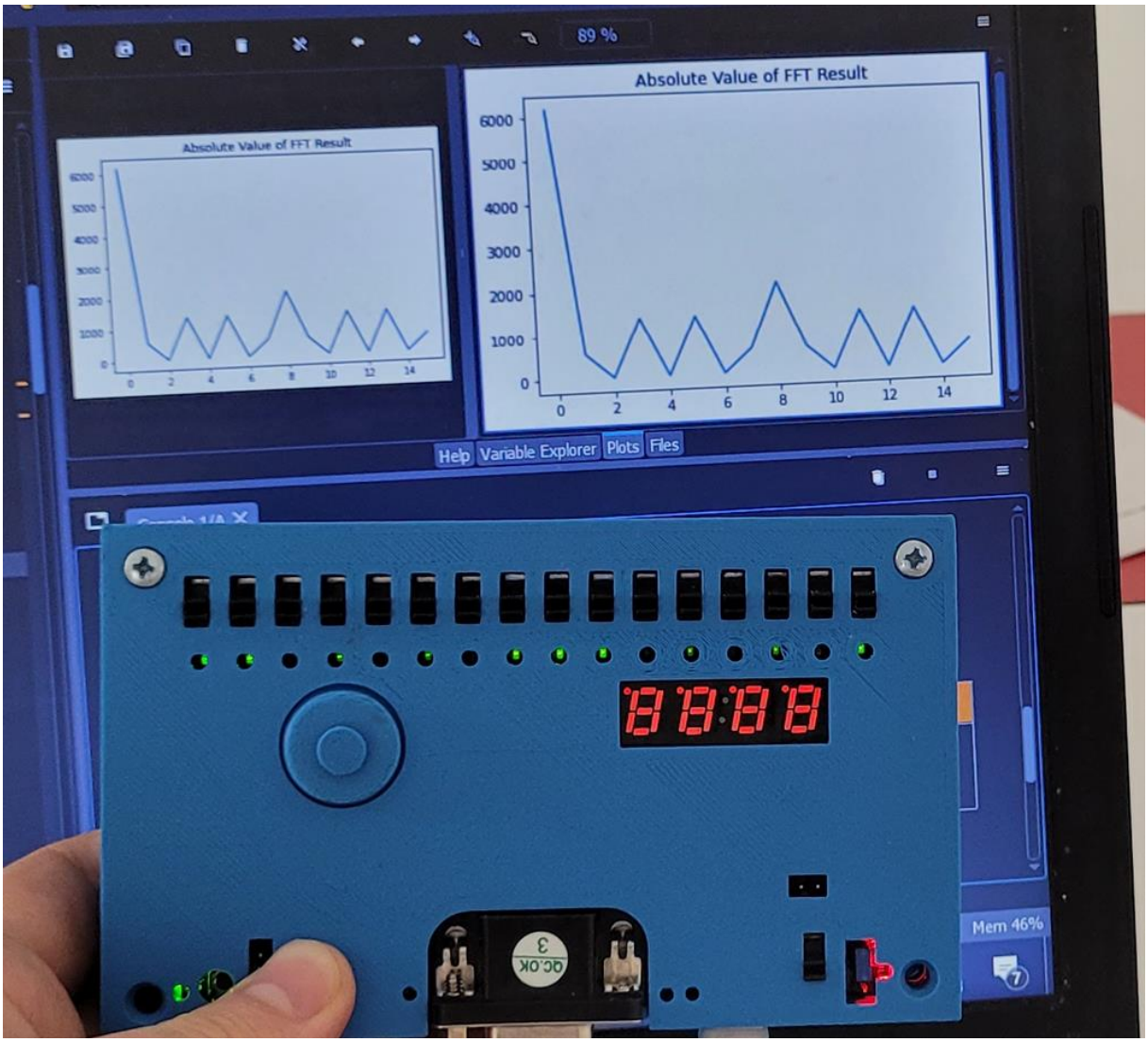

Fig 4.7: Demonstration of real hardware with another signal

Limitations:

- The system is prone overflow: All numbers, included magnituder, operate in integer-32 bits; so, maximum 46340 can be fed into magnituder to ensure that there is no overflow (sqrt(2^31)).
- The signal is hardcoded. Therefore for fully utilizing the hardware engine by external components such as PC express, it needs to further scaled to meet demanded bandwidth communication, DMA (direct memory access) and driver software.

5. Conclusion

The Frequency Monitor project described here demonstrates the potential of hardware acceleration for signal processing applications, with a particular focus on decomposing signals into harmonic components with nanosecond-level speed. The project successfully implemented key components such as audio sensing with piezoelectric sensors, an analog audio amplifier, a Fourier Transformer Hardware Accelerator, and a PWM generator for LED controls.

The piezoelectric sensors reliably detected pressure gradients created by acoustic waves. The analog audio amplifier was designed to amplify the sensor output, facilitating the processing of signals within the acceptable input range of the Basys3. The heart of the project, the Fourier Transformer Hardware Accelerator, showcased the power of hardware acceleration with an efficient implementation of the radix-16 butterfly computation, a key operation in Fast Fourier Transform computations.

The resultant magnitude of each frequency bin was successfully displayed through on-board LEDs, controlled by individual PWM signals. This hardware-based approach significantly increases the speed of signal decomposition compared to conventional CPU processing. However, as with any technical implementation, there are certain limitations and challenges. The system is prone to overflow due to the use of 32-bit integers in the magnitude computation, and the signal is currently hardcoded. This, combined with the increased complexity of the radix-16 butterfly implementation, presents scalability issues for real-world applications. In the future, these challenges can be addressed through the use of DMA and driver software for improved bandwidth communication, as well as optimized resource allocation and memory management strategies for the hardware engine.

The Frequency Monitor project is an excellent demonstration of the power and potential of hardware acceleration in signal processing. It opens up possibilities for further exploration into more complex applications, with the potential to significantly enhance the speed and efficiency of signal decomposition and analysis. The insights gained from this project can be leveraged to drive advancements in areas such as digital signal processing, image processing, and the solution of partial differential equations, where Fast Fourier Transforms are extensively used.

References

[1] Shekhani. H. (2020). COMSOL Simulation of Direct Piezoelectric Effect. https://www.ultrasonicadvisors.com/comsol-simulation-of-the-direct-piezoelectric-effect

[2] Taşkiran, A. H. (2019). Tren Raylarındaki Titreşimler ile Tren Takibi (Application No. 1689B012018060). TUBITAK 2204-A Proje Yarışması.

Appendix

Appendix 1: main.vhd

```
library ieee;

use ieee.std_logic_1164.all;

use ieee.numeric_std.all;


entity complex_mul is--Z<=X*Y

  Port (

    Xre : in integer;

    Xim : in integer;

    Yre : in integer;
```

```
        Yim : in integer;
        Zre : out integer;
        Zim : out integer
    );
end entity complex_mul;

architecture Behavioral of complex_mul is
    -- Intermediate signal declarations
    signal temp1, temp2, temp3, temp4 : integer;
begin
    -- Perform the multiplications
    temp1 <= Xre * Yre;           -- Product of real parts
    temp2 <= Xim * Yim;           -- Product of imaginary parts
    temp3 <= Xre * Yim;           -- Cross-product term
    temp4 <= Xim * Yre;           -- Cross-product term

    -- Calculate the real and imaginary parts of the result
    Zre <= temp1 - temp2;
    Zim <= temp3 + temp4;
end architecture Behavioral;
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity DFT16 is
    port (
    A_re0 : in integer;
    A_im0 : in integer;
    C_re0 : out integer;
    C_im0 : out integer;
    A_re1 : in integer;
```

```
A_im1 : in integer;
C_re1 : out integer;
C_im1 : out integer;
A_re2 : in integer;
A_im2 : in integer;
C_re2 : out integer;
C_im2 : out integer;
A_re3 : in integer;
A_im3 : in integer;
C_re3 : out integer;
C_im3 : out integer;
A_re4 : in integer;
A_im4 : in integer;
C_re4 : out integer;
C_im4 : out integer;
A_re5 : in integer;
A_im5 : in integer;
C_re5 : out integer;
C_im5 : out integer;
A_re6 : in integer;
A_im6 : in integer;
C_re6 : out integer;
C_im6 : out integer;
A_re7 : in integer;
A_im7 : in integer;
C_re7 : out integer;
C_im7 : out integer;
A_re8 : in integer;
A_im8 : in integer;
C_re8 : out integer;
C_im8 : out integer;
```

```
    A_re9 : in integer;
    A_im9 : in integer;
    C_re9 : out integer;
    C_im9 : out integer;
    A_re10 : in integer;
    A_im10 : in integer;
    C_re10 : out integer;
    C_im10 : out integer;
    A_re11 : in integer;
    A_im11 : in integer;
    C_re11 : out integer;
    C_im11 : out integer;
    A_re12 : in integer;
    A_im12 : in integer;
    C_re12 : out integer;
    C_im12 : out integer;
    A_re13 : in integer;
    A_im13 : in integer;
    C_re13 : out integer;
    C_im13 : out integer;
    A_re14 : in integer;
    A_im14 : in integer;
    C_re14 : out integer;
    C_im14 : out integer;
    A_re15 : in integer;
    A_im15 : in integer;
    C_re15 : out integer;
    C_im15 : out integer
    );
end entity DFT16;
architecture Behavioral of DFT16 is
```

```
signal W0_0_re, W0_0_im : integer;
signal tmp0_0_re, tmp0_0_im : integer;
signal W0_1_re, W0_1_im : integer;
signal tmp0_1_re, tmp0_1_im : integer;
signal W0_2_re, W0_2_im : integer;
signal tmp0_2_re, tmp0_2_im : integer;
signal W0_3_re, W0_3_im : integer;
signal tmp0_3_re, tmp0_3_im : integer;
signal W0_4_re, W0_4_im : integer;
signal tmp0_4_re, tmp0_4_im : integer;
signal W0_5_re, W0_5_im : integer;
signal tmp0_5_re, tmp0_5_im : integer;
signal W0_6_re, W0_6_im : integer;
signal tmp0_6_re, tmp0_6_im : integer;
signal W0_7_re, W0_7_im : integer;
signal tmp0_7_re, tmp0_7_im : integer;
signal W0_8_re, W0_8_im : integer;
signal tmp0_8_re, tmp0_8_im : integer;
signal W0_9_re, W0_9_im : integer;
signal tmp0_9_re, tmp0_9_im : integer;
signal W0_10_re, W0_10_im : integer;
signal tmp0_10_re, tmp0_10_im : integer;
signal W0_11_re, W0_11_im : integer;
signal tmp0_11_re, tmp0_11_im : integer;
signal W0_12_re, W0_12_im : integer;
signal tmp0_12_re, tmp0_12_im : integer;
signal W0_13_re, W0_13_im : integer;
signal tmp0_13_re, tmp0_13_im : integer;
signal W0_14_re, W0_14_im : integer;
signal tmp0_14_re, tmp0_14_im : integer;
signal W0_15_re, W0_15_im : integer;
```

```
signal tmp0_15_re, tmp0_15_im : integer;
signal W1_0_re, W1_0_im : integer;
signal tmp1_0_re, tmp1_0_im : integer;
signal W1_1_re, W1_1_im : integer;
signal tmp1_1_re, tmp1_1_im : integer;
signal W1_2_re, W1_2_im : integer;
signal tmp1_2_re, tmp1_2_im : integer;
signal W1_3_re, W1_3_im : integer;
signal tmp1_3_re, tmp1_3_im : integer;
signal W1_4_re, W1_4_im : integer;
signal tmp1_4_re, tmp1_4_im : integer;
signal W1_5_re, W1_5_im : integer;
signal tmp1_5_re, tmp1_5_im : integer;
signal W1_6_re, W1_6_im : integer;
signal tmp1_6_re, tmp1_6_im : integer;
signal W1_7_re, W1_7_im : integer;
signal tmp1_7_re, tmp1_7_im : integer;
signal W1_8_re, W1_8_im : integer;
signal tmp1_8_re, tmp1_8_im : integer;
signal W1_9_re, W1_9_im : integer;
signal tmp1_9_re, tmp1_9_im : integer;
signal W1_10_re, W1_10_im : integer;
signal tmp1_10_re, tmp1_10_im : integer;
signal W1_11_re, W1_11_im : integer;
signal tmp1_11_re, tmp1_11_im : integer;
signal W1_12_re, W1_12_im : integer;
signal tmp1_12_re, tmp1_12_im : integer;
signal W1_13_re, W1_13_im : integer;
signal tmp1_13_re, tmp1_13_im : integer;
signal W1_14_re, W1_14_im : integer;
signal tmp1_14_re, tmp1_14_im : integer;
```

```
signal W1_15_re, W1_15_im : integer;
signal tmp1_15_re, tmp1_15_im : integer;
signal W2_0_re, W2_0_im : integer;
signal tmp2_0_re, tmp2_0_im : integer;
signal W2_1_re, W2_1_im : integer;
signal tmp2_1_re, tmp2_1_im : integer;
signal W2_2_re, W2_2_im : integer;
signal tmp2_2_re, tmp2_2_im : integer;
signal W2_3_re, W2_3_im : integer;
signal tmp2_3_re, tmp2_3_im : integer;
signal W2_4_re, W2_4_im : integer;
signal tmp2_4_re, tmp2_4_im : integer;
signal W2_5_re, W2_5_im : integer;
signal tmp2_5_re, tmp2_5_im : integer;
signal W2_6_re, W2_6_im : integer;
signal tmp2_6_re, tmp2_6_im : integer;
signal W2_7_re, W2_7_im : integer;
signal tmp2_7_re, tmp2_7_im : integer;
signal W2_8_re, W2_8_im : integer;
signal tmp2_8_re, tmp2_8_im : integer;
signal W2_9_re, W2_9_im : integer;
signal tmp2_9_re, tmp2_9_im : integer;
signal W2_10_re, W2_10_im : integer;
signal tmp2_10_re, tmp2_10_im : integer;
signal W2_11_re, W2_11_im : integer;
signal tmp2_11_re, tmp2_11_im : integer;
signal W2_12_re, W2_12_im : integer;
signal tmp2_12_re, tmp2_12_im : integer;
signal W2_13_re, W2_13_im : integer;
signal tmp2_13_re, tmp2_13_im : integer;
signal W2_14_re, W2_14_im : integer;
```

```
signal tmp2_14_re, tmp2_14_im : integer;
signal W2_15_re, W2_15_im : integer;
signal tmp2_15_re, tmp2_15_im : integer;
signal W3_0_re, W3_0_im : integer;
signal tmp3_0_re, tmp3_0_im : integer;
signal W3_1_re, W3_1_im : integer;
signal tmp3_1_re, tmp3_1_im : integer;
signal W3_2_re, W3_2_im : integer;
signal tmp3_2_re, tmp3_2_im : integer;
signal W3_3_re, W3_3_im : integer;
signal tmp3_3_re, tmp3_3_im : integer;
signal W3_4_re, W3_4_im : integer;
signal tmp3_4_re, tmp3_4_im : integer;
signal W3_5_re, W3_5_im : integer;
signal tmp3_5_re, tmp3_5_im : integer;
signal W3_6_re, W3_6_im : integer;
signal tmp3_6_re, tmp3_6_im : integer;
signal W3_7_re, W3_7_im : integer;
signal tmp3_7_re, tmp3_7_im : integer;
signal W3_8_re, W3_8_im : integer;
signal tmp3_8_re, tmp3_8_im : integer;
signal W3_9_re, W3_9_im : integer;
signal tmp3_9_re, tmp3_9_im : integer;
signal W3_10_re, W3_10_im : integer;
signal tmp3_10_re, tmp3_10_im : integer;
signal W3_11_re, W3_11_im : integer;
signal tmp3_11_re, tmp3_11_im : integer;
signal W3_12_re, W3_12_im : integer;
signal tmp3_12_re, tmp3_12_im : integer;
signal W3_13_re, W3_13_im : integer;
signal tmp3_13_re, tmp3_13_im : integer;
```

```
signal W3_14_re, W3_14_im : integer;
signal tmp3_14_re, tmp3_14_im : integer;
signal W3_15_re, W3_15_im : integer;
signal tmp3_15_re, tmp3_15_im : integer;
signal W4_0_re, W4_0_im : integer;
signal tmp4_0_re, tmp4_0_im : integer;
signal W4_1_re, W4_1_im : integer;
signal tmp4_1_re, tmp4_1_im : integer;
signal W4_2_re, W4_2_im : integer;
signal tmp4_2_re, tmp4_2_im : integer;
signal W4_3_re, W4_3_im : integer;
signal tmp4_3_re, tmp4_3_im : integer;
signal W4_4_re, W4_4_im : integer;
signal tmp4_4_re, tmp4_4_im : integer;
signal W4_5_re, W4_5_im : integer;
signal tmp4_5_re, tmp4_5_im : integer;
signal W4_6_re, W4_6_im : integer;
signal tmp4_6_re, tmp4_6_im : integer;
signal W4_7_re, W4_7_im : integer;
signal tmp4_7_re, tmp4_7_im : integer;
signal W4_8_re, W4_8_im : integer;
signal tmp4_8_re, tmp4_8_im : integer;
signal W4_9_re, W4_9_im : integer;
signal tmp4_9_re, tmp4_9_im : integer;
signal W4_10_re, W4_10_im : integer;
signal tmp4_10_re, tmp4_10_im : integer;
signal W4_11_re, W4_11_im : integer;
signal tmp4_11_re, tmp4_11_im : integer;
signal W4_12_re, W4_12_im : integer;
signal tmp4_12_re, tmp4_12_im : integer;
signal W4_13_re, W4_13_im : integer;
```

```
signal tmp4_13_re, tmp4_13_im : integer;
signal W4_14_re, W4_14_im : integer;
signal tmp4_14_re, tmp4_14_im : integer;
signal W4_15_re, W4_15_im : integer;
signal tmp4_15_re, tmp4_15_im : integer;
signal W5_0_re, W5_0_im : integer;
signal tmp5_0_re, tmp5_0_im : integer;
signal W5_1_re, W5_1_im : integer;
signal tmp5_1_re, tmp5_1_im : integer;
signal W5_2_re, W5_2_im : integer;
signal tmp5_2_re, tmp5_2_im : integer;
signal W5_3_re, W5_3_im : integer;
signal tmp5_3_re, tmp5_3_im : integer;
signal W5_4_re, W5_4_im : integer;
signal tmp5_4_re, tmp5_4_im : integer;
signal W5_5_re, W5_5_im : integer;
signal tmp5_5_re, tmp5_5_im : integer;
signal W5_6_re, W5_6_im : integer;
signal tmp5_6_re, tmp5_6_im : integer;
signal W5_7_re, W5_7_im : integer;
signal tmp5_7_re, tmp5_7_im : integer;
signal W5_8_re, W5_8_im : integer;
signal tmp5_8_re, tmp5_8_im : integer;
signal W5_9_re, W5_9_im : integer;
signal tmp5_9_re, tmp5_9_im : integer;
signal W5_10_re, W5_10_im : integer;
signal tmp5_10_re, tmp5_10_im : integer;
signal W5_11_re, W5_11_im : integer;
signal tmp5_11_re, tmp5_11_im : integer;
signal W5_12_re, W5_12_im : integer;
signal tmp5_12_re, tmp5_12_im : integer;
```

```
signal W5_13_re, W5_13_im : integer;
signal tmp5_13_re, tmp5_13_im : integer;
signal W5_14_re, W5_14_im : integer;
signal tmp5_14_re, tmp5_14_im : integer;
signal W5_15_re, W5_15_im : integer;
signal tmp5_15_re, tmp5_15_im : integer;
signal W6_0_re, W6_0_im : integer;
signal tmp6_0_re, tmp6_0_im : integer;
signal W6_1_re, W6_1_im : integer;
signal tmp6_1_re, tmp6_1_im : integer;
signal W6_2_re, W6_2_im : integer;
signal tmp6_2_re, tmp6_2_im : integer;
signal W6_3_re, W6_3_im : integer;
signal tmp6_3_re, tmp6_3_im : integer;
signal W6_4_re, W6_4_im : integer;
signal tmp6_4_re, tmp6_4_im : integer;
signal W6_5_re, W6_5_im : integer;
signal tmp6_5_re, tmp6_5_im : integer;
signal W6_6_re, W6_6_im : integer;
signal tmp6_6_re, tmp6_6_im : integer;
signal W6_7_re, W6_7_im : integer;
signal tmp6_7_re, tmp6_7_im : integer;
signal W6_8_re, W6_8_im : integer;
signal tmp6_8_re, tmp6_8_im : integer;
signal W6_9_re, W6_9_im : integer;
signal tmp6_9_re, tmp6_9_im : integer;
signal W6_10_re, W6_10_im : integer;
signal tmp6_10_re, tmp6_10_im : integer;
signal W6_11_re, W6_11_im : integer;
signal tmp6_11_re, tmp6_11_im : integer;
signal W6_12_re, W6_12_im : integer;
```

```
signal tmp6_12_re, tmp6_12_im : integer;
signal W6_13_re, W6_13_im : integer;
signal tmp6_13_re, tmp6_13_im : integer;
signal W6_14_re, W6_14_im : integer;
signal tmp6_14_re, tmp6_14_im : integer;
signal W6_15_re, W6_15_im : integer;
signal tmp6_15_re, tmp6_15_im : integer;
signal W7_0_re, W7_0_im : integer;
signal tmp7_0_re, tmp7_0_im : integer;
signal W7_1_re, W7_1_im : integer;
signal tmp7_1_re, tmp7_1_im : integer;
signal W7_2_re, W7_2_im : integer;
signal tmp7_2_re, tmp7_2_im : integer;
signal W7_3_re, W7_3_im : integer;
signal tmp7_3_re, tmp7_3_im : integer;
signal W7_4_re, W7_4_im : integer;
signal tmp7_4_re, tmp7_4_im : integer;
signal W7_5_re, W7_5_im : integer;
signal tmp7_5_re, tmp7_5_im : integer;
signal W7_6_re, W7_6_im : integer;
signal tmp7_6_re, tmp7_6_im : integer;
signal W7_7_re, W7_7_im : integer;
signal tmp7_7_re, tmp7_7_im : integer;
signal W7_8_re, W7_8_im : integer;
signal tmp7_8_re, tmp7_8_im : integer;
signal W7_9_re, W7_9_im : integer;
signal tmp7_9_re, tmp7_9_im : integer;
signal W7_10_re, W7_10_im : integer;
signal tmp7_10_re, tmp7_10_im : integer;
signal W7_11_re, W7_11_im : integer;
signal tmp7_11_re, tmp7_11_im : integer;
```

```
signal W7_12_re, W7_12_im : integer;
signal tmp7_12_re, tmp7_12_im : integer;
signal W7_13_re, W7_13_im : integer;
signal tmp7_13_re, tmp7_13_im : integer;
signal W7_14_re, W7_14_im : integer;
signal tmp7_14_re, tmp7_14_im : integer;
signal W7_15_re, W7_15_im : integer;
signal tmp7_15_re, tmp7_15_im : integer;
signal W8_0_re, W8_0_im : integer;
signal tmp8_0_re, tmp8_0_im : integer;
signal W8_1_re, W8_1_im : integer;
signal tmp8_1_re, tmp8_1_im : integer;
signal W8_2_re, W8_2_im : integer;
signal tmp8_2_re, tmp8_2_im : integer;
signal W8_3_re, W8_3_im : integer;
signal tmp8_3_re, tmp8_3_im : integer;
signal W8_4_re, W8_4_im : integer;
signal tmp8_4_re, tmp8_4_im : integer;
signal W8_5_re, W8_5_im : integer;
signal tmp8_5_re, tmp8_5_im : integer;
signal W8_6_re, W8_6_im : integer;
signal tmp8_6_re, tmp8_6_im : integer;
signal W8_7_re, W8_7_im : integer;
signal tmp8_7_re, tmp8_7_im : integer;
signal W8_8_re, W8_8_im : integer;
signal tmp8_8_re, tmp8_8_im : integer;
signal W8_9_re, W8_9_im : integer;
signal tmp8_9_re, tmp8_9_im : integer;
signal W8_10_re, W8_10_im : integer;
signal tmp8_10_re, tmp8_10_im : integer;
signal W8_11_re, W8_11_im : integer;
```

```
signal tmp8_11_re, tmp8_11_im : integer;
signal W8_12_re, W8_12_im : integer;
signal tmp8_12_re, tmp8_12_im : integer;
signal W8_13_re, W8_13_im : integer;
signal tmp8_13_re, tmp8_13_im : integer;
signal W8_14_re, W8_14_im : integer;
signal tmp8_14_re, tmp8_14_im : integer;
signal W8_15_re, W8_15_im : integer;
signal tmp8_15_re, tmp8_15_im : integer;
signal W9_0_re, W9_0_im : integer;
signal tmp9_0_re, tmp9_0_im : integer;
signal W9_1_re, W9_1_im : integer;
signal tmp9_1_re, tmp9_1_im : integer;
signal W9_2_re, W9_2_im : integer;
signal tmp9_2_re, tmp9_2_im : integer;
signal W9_3_re, W9_3_im : integer;
signal tmp9_3_re, tmp9_3_im : integer;
signal W9_4_re, W9_4_im : integer;
signal tmp9_4_re, tmp9_4_im : integer;
signal W9_5_re, W9_5_im : integer;
signal tmp9_5_re, tmp9_5_im : integer;
signal W9_6_re, W9_6_im : integer;
signal tmp9_6_re, tmp9_6_im : integer;
signal W9_7_re, W9_7_im : integer;
signal tmp9_7_re, tmp9_7_im : integer;
signal W9_8_re, W9_8_im : integer;
signal tmp9_8_re, tmp9_8_im : integer;
signal W9_9_re, W9_9_im : integer;
signal tmp9_9_re, tmp9_9_im : integer;
signal W9_10_re, W9_10_im : integer;
signal tmp9_10_re, tmp9_10_im : integer;
```

```
signal W9_11_re, W9_11_im : integer;
signal tmp9_11_re, tmp9_11_im : integer;
signal W9_12_re, W9_12_im : integer;
signal tmp9_12_re, tmp9_12_im : integer;
signal W9_13_re, W9_13_im : integer;
signal tmp9_13_re, tmp9_13_im : integer;
signal W9_14_re, W9_14_im : integer;
signal tmp9_14_re, tmp9_14_im : integer;
signal W9_15_re, W9_15_im : integer;
signal tmp9_15_re, tmp9_15_im : integer;
signal W10_0_re, W10_0_im : integer;
signal tmp10_0_re, tmp10_0_im : integer;
signal W10_1_re, W10_1_im : integer;
signal tmp10_1_re, tmp10_1_im : integer;
signal W10_2_re, W10_2_im : integer;
signal tmp10_2_re, tmp10_2_im : integer;
signal W10_3_re, W10_3_im : integer;
signal tmp10_3_re, tmp10_3_im : integer;
signal W10_4_re, W10_4_im : integer;
signal tmp10_4_re, tmp10_4_im : integer;
signal W10_5_re, W10_5_im : integer;
signal tmp10_5_re, tmp10_5_im : integer;
signal W10_6_re, W10_6_im : integer;
signal tmp10_6_re, tmp10_6_im : integer;
signal W10_7_re, W10_7_im : integer;
signal tmp10_7_re, tmp10_7_im : integer;
signal W10_8_re, W10_8_im : integer;
signal tmp10_8_re, tmp10_8_im : integer;
signal W10_9_re, W10_9_im : integer;
signal tmp10_9_re, tmp10_9_im : integer;
signal W10_10_re, W10_10_im : integer;
```

```
signal tmp10_10_re, tmp10_10_im : integer;
signal W10_11_re, W10_11_im : integer;
signal tmp10_11_re, tmp10_11_im : integer;
signal W10_12_re, W10_12_im : integer;
signal tmp10_12_re, tmp10_12_im : integer;
signal W10_13_re, W10_13_im : integer;
signal tmp10_13_re, tmp10_13_im : integer;
signal W10_14_re, W10_14_im : integer;
signal tmp10_14_re, tmp10_14_im : integer;
signal W10_15_re, W10_15_im : integer;
signal tmp10_15_re, tmp10_15_im : integer;
signal W11_0_re, W11_0_im : integer;
signal tmp11_0_re, tmp11_0_im : integer;
signal W11_1_re, W11_1_im : integer;
signal tmp11_1_re, tmp11_1_im : integer;
signal W11_2_re, W11_2_im : integer;
signal tmp11_2_re, tmp11_2_im : integer;
signal W11_3_re, W11_3_im : integer;
signal tmp11_3_re, tmp11_3_im : integer;
signal W11_4_re, W11_4_im : integer;
signal tmp11_4_re, tmp11_4_im : integer;
signal W11_5_re, W11_5_im : integer;
signal tmp11_5_re, tmp11_5_im : integer;
signal W11_6_re, W11_6_im : integer;
signal tmp11_6_re, tmp11_6_im : integer;
signal W11_7_re, W11_7_im : integer;
signal tmp11_7_re, tmp11_7_im : integer;
signal W11_8_re, W11_8_im : integer;
signal tmp11_8_re, tmp11_8_im : integer;
signal W11_9_re, W11_9_im : integer;
signal tmp11_9_re, tmp11_9_im : integer;
```

```
signal W11_10_re, W11_10_im : integer;
signal tmp11_10_re, tmp11_10_im : integer;
signal W11_11_re, W11_11_im : integer;
signal tmp11_11_re, tmp11_11_im : integer;
signal W11_12_re, W11_12_im : integer;
signal tmp11_12_re, tmp11_12_im : integer;
signal W11_13_re, W11_13_im : integer;
signal tmp11_13_re, tmp11_13_im : integer;
signal W11_14_re, W11_14_im : integer;
signal tmp11_14_re, tmp11_14_im : integer;
signal W11_15_re, W11_15_im : integer;
signal tmp11_15_re, tmp11_15_im : integer;
signal W12_0_re, W12_0_im : integer;
signal tmp12_0_re, tmp12_0_im : integer;
signal W12_1_re, W12_1_im : integer;
signal tmp12_1_re, tmp12_1_im : integer;
signal W12_2_re, W12_2_im : integer;
signal tmp12_2_re, tmp12_2_im : integer;
signal W12_3_re, W12_3_im : integer;
signal tmp12_3_re, tmp12_3_im : integer;
signal W12_4_re, W12_4_im : integer;
signal tmp12_4_re, tmp12_4_im : integer;
signal W12_5_re, W12_5_im : integer;
signal tmp12_5_re, tmp12_5_im : integer;
signal W12_6_re, W12_6_im : integer;
signal tmp12_6_re, tmp12_6_im : integer;
signal W12_7_re, W12_7_im : integer;
signal tmp12_7_re, tmp12_7_im : integer;
signal W12_8_re, W12_8_im : integer;
signal tmp12_8_re, tmp12_8_im : integer;
signal W12_9_re, W12_9_im : integer;
```

```
signal tmp12_9_re, tmp12_9_im : integer;
signal W12_10_re, W12_10_im : integer;
signal tmp12_10_re, tmp12_10_im : integer;
signal W12_11_re, W12_11_im : integer;
signal tmp12_11_re, tmp12_11_im : integer;
signal W12_12_re, W12_12_im : integer;
signal tmp12_12_re, tmp12_12_im : integer;
signal W12_13_re, W12_13_im : integer;
signal tmp12_13_re, tmp12_13_im : integer;
signal W12_14_re, W12_14_im : integer;
signal tmp12_14_re, tmp12_14_im : integer;
signal W12_15_re, W12_15_im : integer;
signal tmp12_15_re, tmp12_15_im : integer;
signal W13_0_re, W13_0_im : integer;
signal tmp13_0_re, tmp13_0_im : integer;
signal W13_1_re, W13_1_im : integer;
signal tmp13_1_re, tmp13_1_im : integer;
signal W13_2_re, W13_2_im : integer;
signal tmp13_2_re, tmp13_2_im : integer;
signal W13_3_re, W13_3_im : integer;
signal tmp13_3_re, tmp13_3_im : integer;
signal W13_4_re, W13_4_im : integer;
signal tmp13_4_re, tmp13_4_im : integer;
signal W13_5_re, W13_5_im : integer;
signal tmp13_5_re, tmp13_5_im : integer;
signal W13_6_re, W13_6_im : integer;
signal tmp13_6_re, tmp13_6_im : integer;
signal W13_7_re, W13_7_im : integer;
signal tmp13_7_re, tmp13_7_im : integer;
signal W13_8_re, W13_8_im : integer;
signal tmp13_8_re, tmp13_8_im : integer;
```

```
signal W13_9_re, W13_9_im : integer;
signal tmp13_9_re, tmp13_9_im : integer;
signal W13_10_re, W13_10_im : integer;
signal tmp13_10_re, tmp13_10_im : integer;
signal W13_11_re, W13_11_im : integer;
signal tmp13_11_re, tmp13_11_im : integer;
signal W13_12_re, W13_12_im : integer;
signal tmp13_12_re, tmp13_12_im : integer;
signal W13_13_re, W13_13_im : integer;
signal tmp13_13_re, tmp13_13_im : integer;
signal W13_14_re, W13_14_im : integer;
signal tmp13_14_re, tmp13_14_im : integer;
signal W13_15_re, W13_15_im : integer;
signal tmp13_15_re, tmp13_15_im : integer;
signal W14_0_re, W14_0_im : integer;
signal tmp14_0_re, tmp14_0_im : integer;
signal W14_1_re, W14_1_im : integer;
signal tmp14_1_re, tmp14_1_im : integer;
signal W14_2_re, W14_2_im : integer;
signal tmp14_2_re, tmp14_2_im : integer;
signal W14_3_re, W14_3_im : integer;
signal tmp14_3_re, tmp14_3_im : integer;
signal W14_4_re, W14_4_im : integer;
signal tmp14_4_re, tmp14_4_im : integer;
signal W14_5_re, W14_5_im : integer;
signal tmp14_5_re, tmp14_5_im : integer;
signal W14_6_re, W14_6_im : integer;
signal tmp14_6_re, tmp14_6_im : integer;
signal W14_7_re, W14_7_im : integer;
signal tmp14_7_re, tmp14_7_im : integer;
signal W14_8_re, W14_8_im : integer;
```

```
signal tmp14_8_re, tmp14_8_im : integer;
signal W14_9_re, W14_9_im : integer;
signal tmp14_9_re, tmp14_9_im : integer;
signal W14_10_re, W14_10_im : integer;
signal tmp14_10_re, tmp14_10_im : integer;
signal W14_11_re, W14_11_im : integer;
signal tmp14_11_re, tmp14_11_im : integer;
signal W14_12_re, W14_12_im : integer;
signal tmp14_12_re, tmp14_12_im : integer;
signal W14_13_re, W14_13_im : integer;
signal tmp14_13_re, tmp14_13_im : integer;
signal W14_14_re, W14_14_im : integer;
signal tmp14_14_re, tmp14_14_im : integer;
signal W14_15_re, W14_15_im : integer;
signal tmp14_15_re, tmp14_15_im : integer;
signal W15_0_re, W15_0_im : integer;
signal tmp15_0_re, tmp15_0_im : integer;
signal W15_1_re, W15_1_im : integer;
signal tmp15_1_re, tmp15_1_im : integer;
signal W15_2_re, W15_2_im : integer;
signal tmp15_2_re, tmp15_2_im : integer;
signal W15_3_re, W15_3_im : integer;
signal tmp15_3_re, tmp15_3_im : integer;
signal W15_4_re, W15_4_im : integer;
signal tmp15_4_re, tmp15_4_im : integer;
signal W15_5_re, W15_5_im : integer;
signal tmp15_5_re, tmp15_5_im : integer;
signal W15_6_re, W15_6_im : integer;
signal tmp15_6_re, tmp15_6_im : integer;
signal W15_7_re, W15_7_im : integer;
signal tmp15_7_re, tmp15_7_im : integer;
```

```
signal W15_8_re, W15_8_im : integer;
signal tmp15_8_re, tmp15_8_im : integer;
signal W15_9_re, W15_9_im : integer;
signal tmp15_9_re, tmp15_9_im : integer;
signal W15_10_re, W15_10_im : integer;
signal tmp15_10_re, tmp15_10_im : integer;
signal W15_11_re, W15_11_im : integer;
signal tmp15_11_re, tmp15_11_im : integer;
signal W15_12_re, W15_12_im : integer;
signal tmp15_12_re, tmp15_12_im : integer;
signal W15_13_re, W15_13_im : integer;
signal tmp15_13_re, tmp15_13_im : integer;
signal W15_14_re, W15_14_im : integer;
signal tmp15_14_re, tmp15_14_im : integer;
signal W15_15_re, W15_15_im : integer;
signal tmp15_15_re, tmp15_15_im : integer;
begin
initial_process: process
begin
   W0_0_re <= 4096;
   W0_0_im <= 0;
   W0_1_re <= 4096;
   W0_1_im <= 0;
   W0_2_re <= 4096;
   W0_2_im <= 0;
   W0_3_re <= 4096;
   W0_3_im <= 0;
   W0_4_re <= 4096;
   W0_4_im <= 0;
   W0_5_re <= 4096;
   W0_5_im <= 0;
```

```
W0_6_re <= 4096;
W0_6_im <= 0;
W0_7_re <= 4096;
W0_7_im <= 0;
W0_8_re <= 4096;
W0_8_im <= 0;
W0_9_re <= 4096;
W0_9_im <= 0;
W0_10_re <= 4096;
W0_10_im <= 0;
W0_11_re <= 4096;
W0_11_im <= 0;
W0_12_re <= 4096;
W0_12_im <= 0;
W0_13_re <= 4096;
W0_13_im <= 0;
W0_14_re <= 4096;
W0_14_im <= 0;
W0_15_re <= 4096;
W0_15_im <= 0;
W1_0_re <= 4096;
W1_0_im <= 0;
W1_1_re <= 3784;
W1_1_im <= -1567;
W1_2_re <= 2896;
W1_2_im <= -2896;
W1_3_re <= 1567;
W1_3_im <= -3784;
W1_4_re <= 0;
W1_4_im <= -4096;
W1_5_re <= -1567;
```

```
W1_5_im <= -3784;
W1_6_re <= -2896;
W1_6_im <= -2896;
W1_7_re <= -3784;
W1_7_im <= -1567;
W1_8_re <= -4096;
W1_8_im <= 0;
W1_9_re <= -3784;
W1_9_im <= 1567;
W1_10_re <= -2896;
W1_10_im <= 2896;
W1_11_re <= -1567;
W1_11_im <= 3784;
W1_12_re <= 0;
W1_12_im <= 4096;
W1_13_re <= 1567;
W1_13_im <= 3784;
W1_14_re <= 2896;
W1_14_im <= 2896;
W1_15_re <= 3784;
W1_15_im <= 1567;
W2_0_re <= 4096;
W2_0_im <= 0;
W2_1_re <= 2896;
W2_1_im <= -2896;
W2_2_re <= 0;
W2_2_im <= -4096;
W2_3_re <= -2896;
W2_3_im <= -2896;
W2_4_re <= -4096;
W2_4_im <= 0;
```

```
W2_5_re <= -2896;
W2_5_im <= 2896;
W2_6_re <= 0;
W2_6_im <= 4096;
W2_7_re <= 2896;
W2_7_im <= 2896;
W2_8_re <= 4096;
W2_8_im <= 0;
W2_9_re <= 2896;
W2_9_im <= -2896;
W2_10_re <= 0;
W2_10_im <= -4096;
W2_11_re <= -2896;
W2_11_im <= -2896;
W2_12_re <= -4096;
W2_12_im <= 0;
W2_13_re <= -2896;
W2_13_im <= 2896;
W2_14_re <= 0;
W2_14_im <= 4096;
W2_15_re <= 2896;
W2_15_im <= 2896;
W3_0_re <= 4096;
W3_0_im <= 0;
W3_1_re <= 1567;
W3_1_im <= -3784;
W3_2_re <= -2896;
W3_2_im <= -2896;
W3_3_re <= -3784;
W3_3_im <= 1567;
W3_4_re <= 0;
```

```
W3_4_im <= 4096;
W3_5_re <= 3784;
W3_5_im <= 1567;
W3_6_re <= 2896;
W3_6_im <= -2896;
W3_7_re <= -1567;
W3_7_im <= -3784;
W3_8_re <= -4096;
W3_8_im <= 0;
W3_9_re <= -1567;
W3_9_im <= 3784;
W3_10_re <= 2896;
W3_10_im <= 2896;
W3_11_re <= 3784;
W3_11_im <= -1567;
W3_12_re <= 0;
W3_12_im <= -4096;
W3_13_re <= -3784;
W3_13_im <= -1567;
W3_14_re <= -2896;
W3_14_im <= 2896;
W3_15_re <= 1567;
W3_15_im <= 3784;
W4_0_re <= 4096;
W4_0_im <= 0;
W4_1_re <= 0;
W4_1_im <= -4096;
W4_2_re <= -4096;
W4_2_im <= 0;
W4_3_re <= 0;
W4_3_im <= 4096;
```

```
W4_4_re <= 4096;
W4_4_im <= 0;
W4_5_re <= 0;
W4_5_im <= -4096;
W4_6_re <= -4096;
W4_6_im <= 0;
W4_7_re <= 0;
W4_7_im <= 4096;
W4_8_re <= 4096;
W4_8_im <= 0;
W4_9_re <= 0;
W4_9_im <= -4096;
W4_10_re <= -4096;
W4_10_im <= 0;
W4_11_re <= 0;
W4_11_im <= 4096;
W4_12_re <= 4096;
W4_12_im <= 0;
W4_13_re <= 0;
W4_13_im <= -4096;
W4_14_re <= -4096;
W4_14_im <= 0;
W4_15_re <= 0;
W4_15_im <= 4096;
W5_0_re <= 4096;
W5_0_im <= 0;
W5_1_re <= -1567;
W5_1_im <= -3784;
W5_2_re <= -2896;
W5_2_im <= 2896;
W5_3_re <= 3784;
```

```
W5_3_im <= 1567;
W5_4_re <= 0;
W5_4_im <= -4096;
W5_5_re <= -3784;
W5_5_im <= 1567;
W5_6_re <= 2896;
W5_6_im <= 2896;
W5_7_re <= 1567;
W5_7_im <= -3784;
W5_8_re <= -4096;
W5_8_im <= 0;
W5_9_re <= 1567;
W5_9_im <= 3784;
W5_10_re <= 2896;
W5_10_im <= -2896;
W5_11_re <= -3784;
W5_11_im <= -1567;
W5_12_re <= 0;
W5_12_im <= 4096;
W5_13_re <= 3784;
W5_13_im <= -1567;
W5_14_re <= -2896;
W5_14_im <= -2896;
W5_15_re <= -1567;
W5_15_im <= 3784;
W6_0_re <= 4096;
W6_0_im <= 0;
W6_1_re <= -2896;
W6_1_im <= -2896;
W6_2_re <= 0;
W6_2_im <= 4096;
```

```
W6_3_re <= 2896;
W6_3_im <= -2896;
W6_4_re <= -4096;
W6_4_im <= 0;
W6_5_re <= 2896;
W6_5_im <= 2896;
W6_6_re <= 0;
W6_6_im <= -4096;
W6_7_re <= -2896;
W6_7_im <= 2896;
W6_8_re <= 4096;
W6_8_im <= 0;
W6_9_re <= -2896;
W6_9_im <= -2896;
W6_10_re <= 0;
W6_10_im <= 4096;
W6_11_re <= 2896;
W6_11_im <= -2896;
W6_12_re <= -4096;
W6_12_im <= 0;
W6_13_re <= 2896;
W6_13_im <= 2896;
W6_14_re <= 0;
W6_14_im <= -4096;
W6_15_re <= -2896;
W6_15_im <= 2896;
W7_0_re <= 4096;
W7_0_im <= 0;
W7_1_re <= -3784;
W7_1_im <= -1567;
W7_2_re <= 2896;
```

```
W7_2_im <= 2896;
W7_3_re <= -1567;
W7_3_im <= -3784;
W7_4_re <= 0;
W7_4_im <= 4096;
W7_5_re <= 1567;
W7_5_im <= -3784;
W7_6_re <= -2896;
W7_6_im <= 2896;
W7_7_re <= 3784;
W7_7_im <= -1567;
W7_8_re <= -4096;
W7_8_im <= 0;
W7_9_re <= 3784;
W7_9_im <= 1567;
W7_10_re <= -2896;
W7_10_im <= -2896;
W7_11_re <= 1567;
W7_11_im <= 3784;
W7_12_re <= 0;
W7_12_im <= -4096;
W7_13_re <= -1567;
W7_13_im <= 3784;
W7_14_re <= 2896;
W7_14_im <= -2896;
W7_15_re <= -3784;
W7_15_im <= 1567;
W8_0_re <= 4096;
W8_0_im <= 0;
W8_1_re <= -4096;
W8_1_im <= 0;
```

```
W8_2_re <= 4096;
W8_2_im <= 0;
W8_3_re <= -4096;
W8_3_im <= 0;
W8_4_re <= 4096;
W8_4_im <= 0;
W8_5_re <= -4096;
W8_5_im <= 0;
W8_6_re <= 4096;
W8_6_im <= 0;
W8_7_re <= -4096;
W8_7_im <= 0;
W8_8_re <= 4096;
W8_8_im <= 0;
W8_9_re <= -4096;
W8_9_im <= 0;
W8_10_re <= 4096;
W8_10_im <= 0;
W8_11_re <= -4096;
W8_11_im <= 0;
W8_12_re <= 4096;
W8_12_im <= 0;
W8_13_re <= -4096;
W8_13_im <= 0;
W8_14_re <= 4096;
W8_14_im <= 0;
W8_15_re <= -4096;
W8_15_im <= 0;
W9_0_re <= 4096;
W9_0_im <= 0;
W9_1_re <= -3784;
```

```
W9_1_im <= 1567;
W9_2_re <= 2896;
W9_2_im <= -2896;
W9_3_re <= -1567;
W9_3_im <= 3784;
W9_4_re <= 0;
W9_4_im <= -4096;
W9_5_re <= 1567;
W9_5_im <= 3784;
W9_6_re <= -2896;
W9_6_im <= -2896;
W9_7_re <= 3784;
W9_7_im <= 1567;
W9_8_re <= -4096;
W9_8_im <= 0;
W9_9_re <= 3784;
W9_9_im <= -1567;
W9_10_re <= -2896;
W9_10_im <= 2896;
W9_11_re <= 1567;
W9_11_im <= -3784;
W9_12_re <= 0;
W9_12_im <= 4096;
W9_13_re <= -1567;
W9_13_im <= -3784;
W9_14_re <= 2896;
W9_14_im <= 2896;
W9_15_re <= -3784;
W9_15_im <= -1567;
W10_0_re <= 4096;
W10_0_im <= 0;
```

```
W10_1_re <= -2896;
W10_1_im <= 2896;
W10_2_re <= 0;
W10_2_im <= -4096;
W10_3_re <= 2896;
W10_3_im <= 2896;
W10_4_re <= -4096;
W10_4_im <= 0;
W10_5_re <= 2896;
W10_5_im <= -2896;
W10_6_re <= 0;
W10_6_im <= 4096;
W10_7_re <= -2896;
W10_7_im <= -2896;
W10_8_re <= 4096;
W10_8_im <= 0;
W10_9_re <= -2896;
W10_9_im <= 2896;
W10_10_re <= 0;
W10_10_im <= -4096;
W10_11_re <= 2896;
W10_11_im <= 2896;
W10_12_re <= -4096;
W10_12_im <= 0;
W10_13_re <= 2896;
W10_13_im <= -2896;
W10_14_re <= 0;
W10_14_im <= 4096;
W10_15_re <= -2896;
W10_15_im <= -2896;
W11_0_re <= 4096;
```

```
W11_0_im <= 0;
W11_1_re <= -1567;
W11_1_im <= 3784;
W11_2_re <= -2896;
W11_2_im <= -2896;
W11_3_re <= 3784;
W11_3_im <= -1567;
W11_4_re <= 0;
W11_4_im <= 4096;
W11_5_re <= -3784;
W11_5_im <= -1567;
W11_6_re <= 2896;
W11_6_im <= -2896;
W11_7_re <= 1567;
W11_7_im <= 3784;
W11_8_re <= -4096;
W11_8_im <= 0;
W11_9_re <= 1567;
W11_9_im <= -3784;
W11_10_re <= 2896;
W11_10_im <= 2896;
W11_11_re <= -3784;
W11_11_im <= 1567;
W11_12_re <= 0;
W11_12_im <= -4096;
W11_13_re <= 3784;
W11_13_im <= 1567;
W11_14_re <= -2896;
W11_14_im <= 2896;
W11_15_re <= -1567;
W11_15_im <= -3784;
```

```
W12_0_re <= 4096;
W12_0_im <= 0;
W12_1_re <= 0;
W12_1_im <= 4096;
W12_2_re <= -4096;
W12_2_im <= 0;
W12_3_re <= 0;
W12_3_im <= -4096;
W12_4_re <= 4096;
W12_4_im <= 0;
W12_5_re <= 0;
W12_5_im <= 4096;
W12_6_re <= -4096;
W12_6_im <= 0;
W12_7_re <= 0;
W12_7_im <= -4096;
W12_8_re <= 4096;
W12_8_im <= 0;
W12_9_re <= 0;
W12_9_im <= 4096;
W12_10_re <= -4096;
W12_10_im <= 0;
W12_11_re <= 0;
W12_11_im <= -4096;
W12_12_re <= 4096;
W12_12_im <= 0;
W12_13_re <= 0;
W12_13_im <= 4096;
W12_14_re <= -4096;
W12_14_im <= 0;
W12_15_re <= 0;
```

```
W12_15_im <= -4096;
W13_0_re <= 4096;
W13_0_im <= 0;
W13_1_re <= 1567;
W13_1_im <= 3784;
W13_2_re <= -2896;
W13_2_im <= 2896;
W13_3_re <= -3784;
W13_3_im <= -1567;
W13_4_re <= 0;
W13_4_im <= -4096;
W13_5_re <= 3784;
W13_5_im <= -1567;
W13_6_re <= 2896;
W13_6_im <= 2896;
W13_7_re <= -1567;
W13_7_im <= 3784;
W13_8_re <= -4096;
W13_8_im <= 0;
W13_9_re <= -1567;
W13_9_im <= -3784;
W13_10_re <= 2896;
W13_10_im <= -2896;
W13_11_re <= 3784;
W13_11_im <= 1567;
W13_12_re <= 0;
W13_12_im <= 4096;
W13_13_re <= -3784;
W13_13_im <= 1567;
W13_14_re <= -2896;
W13_14_im <= -2896;
```

```
W13_15_re <= 1567;
W13_15_im <= -3784;
W14_0_re <= 4096;
W14_0_im <= 0;
W14_1_re <= 2896;
W14_1_im <= 2896;
W14_2_re <= 0;
W14_2_im <= 4096;
W14_3_re <= -2896;
W14_3_im <= 2896;
W14_4_re <= -4096;
W14_4_im <= 0;
W14_5_re <= -2896;
W14_5_im <= -2896;
W14_6_re <= 0;
W14_6_im <= -4096;
W14_7_re <= 2896;
W14_7_im <= -2896;
W14_8_re <= 4096;
W14_8_im <= 0;
W14_9_re <= 2896;
W14_9_im <= 2896;
W14_10_re <= 0;
W14_10_im <= 4096;
W14_11_re <= -2896;
W14_11_im <= 2896;
W14_12_re <= -4096;
W14_12_im <= 0;
W14_13_re <= -2896;
W14_13_im <= -2896;
W14_14_re <= 0;
```

W14_14_im <= -4096;

W14_15_re <= 2896;

W14_15_im <= -2896;

W15_0_re <= 4096;

W15_0_im <= 0;

W15_1_re <= 3784;

W15_1_im <= 1567;

W15_2_re <= 2896;

W15_2_im <= 2896;

W15_3_re <= 1567;

W15_3_im <= 3784;

W15_4_re <= 0;

W15_4_im <= 4096;

W15_5_re <= -1567;

W15_5_im <= 3784;

W15_6_re <= -2896;

W15_6_im <= 2896;

W15_7_re <= -3784;

W15_7_im <= 1567;

W15_8_re <= -4096;

W15_8_im <= 0;

W15_9_re <= -3784;

W15_9_im <= -1567;

W15_10_re <= -2896;

W15_10_im <= -2896;

W15_11_re <= -1567;

W15_11_im <= -3784;

W15_12_re <= 0;

W15_12_im <= -4096;

W15_13_re <= 1567;

W15_13_im <= -3784;

```
    W15_14_re <= 2896;
    W15_14_im <= -2896;
    W15_15_re <= 3784;
    W15_15_im <= -1567;
wait;
end process;
mul0_0 : entity work.complex_mul
    port map (
        Xre => W0_0_re,
        Xim => W0_0_im,
        Yre => A_re0,
        Yim => A_im0,
        Zre => tmp0_0_re,
        Zim => tmp0_0_im
    );
mul0_1 : entity work.complex_mul
    port map (
        Xre => W0_1_re,
        Xim => W0_1_im,
        Yre => A_re1,
        Yim => A_im1,
        Zre => tmp0_1_re,
        Zim => tmp0_1_im
    );
mul0_2 : entity work.complex_mul
    port map (
        Xre => W0_2_re,
        Xim => W0_2_im,
        Yre => A_re2,
        Yim => A_im2,
        Zre => tmp0_2_re,
```

```
        Zim => tmp0_2_im
    );
mul0_3 : entity work.complex_mul
    port map (
        Xre => W0_3_re,
        Xim => W0_3_im,
        Yre => A_re3,
        Yim => A_im3,
        Zre => tmp0_3_re,
        Zim => tmp0_3_im
    );
mul0_4 : entity work.complex_mul
    port map (
        Xre => W0_4_re,
        Xim => W0_4_im,
        Yre => A_re4,
        Yim => A_im4,
        Zre => tmp0_4_re,
        Zim => tmp0_4_im
    );
mul0_5 : entity work.complex_mul
    port map (
        Xre => W0_5_re,
        Xim => W0_5_im,
        Yre => A_re5,
        Yim => A_im5,
        Zre => tmp0_5_re,
        Zim => tmp0_5_im
    );
mul0_6 : entity work.complex_mul
    port map (
```

```
        Xre => W0_6_re,
        Xim => W0_6_im,
        Yre => A_re6,
        Yim => A_im6,
        Zre => tmp0_6_re,
        Zim => tmp0_6_im
    );
mul0_7 : entity work.complex_mul
    port map (
        Xre => W0_7_re,
        Xim => W0_7_im,
        Yre => A_re7,
        Yim => A_im7,
        Zre => tmp0_7_re,
        Zim => tmp0_7_im
    );
mul0_8 : entity work.complex_mul
    port map (
        Xre => W0_8_re,
        Xim => W0_8_im,
        Yre => A_re8,
        Yim => A_im8,
        Zre => tmp0_8_re,
        Zim => tmp0_8_im
    );
mul0_9 : entity work.complex_mul
    port map (
        Xre => W0_9_re,
        Xim => W0_9_im,
        Yre => A_re9,
        Yim => A_im9,
```

```
      Zre => tmp0_9_re,
      Zim => tmp0_9_im
   );
mul0_10 : entity work.complex_mul
   port map (
      Xre => W0_10_re,
      Xim => W0_10_im,
      Yre => A_re10,
      Yim => A_im10,
      Zre => tmp0_10_re,
      Zim => tmp0_10_im
   );
mul0_11 : entity work.complex_mul
   port map (
      Xre => W0_11_re,
      Xim => W0_11_im,
      Yre => A_re11,
      Yim => A_im11,
      Zre => tmp0_11_re,
      Zim => tmp0_11_im
   );
mul0_12 : entity work.complex_mul
   port map (
      Xre => W0_12_re,
      Xim => W0_12_im,
      Yre => A_re12,
      Yim => A_im12,
      Zre => tmp0_12_re,
      Zim => tmp0_12_im
   );
mul0_13 : entity work.complex_mul
```

```
    port map (
        Xre => W0_13_re,
        Xim => W0_13_im,
        Yre => A_re13,
        Yim => A_im13,
        Zre => tmp0_13_re,
        Zim => tmp0_13_im
    );
mul0_14 : entity work.complex_mul
    port map (
        Xre => W0_14_re,
        Xim => W0_14_im,
        Yre => A_re14,
        Yim => A_im14,
        Zre => tmp0_14_re,
        Zim => tmp0_14_im
    );
mul0_15 : entity work.complex_mul
    port map (
        Xre => W0_15_re,
        Xim => W0_15_im,
        Yre => A_re15,
        Yim => A_im15,
        Zre => tmp0_15_re,
        Zim => tmp0_15_im
    );
mul1_0 : entity work.complex_mul
    port map (
        Xre => W1_0_re,
        Xim => W1_0_im,
        Yre => A_re0,
```

```
        Yim => A_im0,
        Zre => tmp1_0_re,
        Zim => tmp1_0_im
    );
mul1_1 : entity work.complex_mul
    port map (
        Xre => W1_1_re,
        Xim => W1_1_im,
        Yre => A_re1,
        Yim => A_im1,
        Zre => tmp1_1_re,
        Zim => tmp1_1_im
    );
mul1_2 : entity work.complex_mul
    port map (
        Xre => W1_2_re,
        Xim => W1_2_im,
        Yre => A_re2,
        Yim => A_im2,
        Zre => tmp1_2_re,
        Zim => tmp1_2_im
    );
mul1_3 : entity work.complex_mul
    port map (
        Xre => W1_3_re,
        Xim => W1_3_im,
        Yre => A_re3,
        Yim => A_im3,
        Zre => tmp1_3_re,
        Zim => tmp1_3_im
    );
```

```
mul1_4 : entity work.complex_mul
   port map (
      Xre => W1_4_re,
      Xim => W1_4_im,
      Yre => A_re4,
      Yim => A_im4,
      Zre => tmp1_4_re,
      Zim => tmp1_4_im
   );
mul1_5 : entity work.complex_mul
   port map (
      Xre => W1_5_re,
      Xim => W1_5_im,
      Yre => A_re5,
      Yim => A_im5,
      Zre => tmp1_5_re,
      Zim => tmp1_5_im
   );
mul1_6 : entity work.complex_mul
   port map (
      Xre => W1_6_re,
      Xim => W1_6_im,
      Yre => A_re6,
      Yim => A_im6,
      Zre => tmp1_6_re,
      Zim => tmp1_6_im
   );
mul1_7 : entity work.complex_mul
   port map (
      Xre => W1_7_re,
      Xim => W1_7_im,
```

```
        Yre => A_re7,
        Yim => A_im7,
        Zre => tmp1_7_re,
        Zim => tmp1_7_im
    );
mul1_8 : entity work.complex_mul
    port map (
        Xre => W1_8_re,
        Xim => W1_8_im,
        Yre => A_re8,
        Yim => A_im8,
        Zre => tmp1_8_re,
        Zim => tmp1_8_im
    );
mul1_9 : entity work.complex_mul
    port map (
        Xre => W1_9_re,
        Xim => W1_9_im,
        Yre => A_re9,
        Yim => A_im9,
        Zre => tmp1_9_re,
        Zim => tmp1_9_im
    );
mul1_10 : entity work.complex_mul
    port map (
        Xre => W1_10_re,
        Xim => W1_10_im,
        Yre => A_re10,
        Yim => A_im10,
        Zre => tmp1_10_re,
        Zim => tmp1_10_im
```

```
    );
mul1_11 : entity work.complex_mul
    port map (
        Xre => W1_11_re,
        Xim => W1_11_im,
        Yre => A_re11,
        Yim => A_im11,
        Zre => tmp1_11_re,
        Zim => tmp1_11_im
    );
mul1_12 : entity work.complex_mul
    port map (
        Xre => W1_12_re,
        Xim => W1_12_im,
        Yre => A_re12,
        Yim => A_im12,
        Zre => tmp1_12_re,
        Zim => tmp1_12_im
    );
mul1_13 : entity work.complex_mul
    port map (
        Xre => W1_13_re,
        Xim => W1_13_im,
        Yre => A_re13,
        Yim => A_im13,
        Zre => tmp1_13_re,
        Zim => tmp1_13_im
    );
mul1_14 : entity work.complex_mul
    port map (
        Xre => W1_14_re,
```

```
      Xim => W1_14_im,
      Yre => A_re14,
      Yim => A_im14,
      Zre => tmp1_14_re,
      Zim => tmp1_14_im
   );
mul1_15 : entity work.complex_mul
   port map (
      Xre => W1_15_re,
      Xim => W1_15_im,
      Yre => A_re15,
      Yim => A_im15,
      Zre => tmp1_15_re,
      Zim => tmp1_15_im
   );
mul2_0 : entity work.complex_mul
   port map (
      Xre => W2_0_re,
      Xim => W2_0_im,
      Yre => A_re0,
      Yim => A_im0,
      Zre => tmp2_0_re,
      Zim => tmp2_0_im
   );
mul2_1 : entity work.complex_mul
   port map (
      Xre => W2_1_re,
      Xim => W2_1_im,
      Yre => A_re1,
      Yim => A_im1,
      Zre => tmp2_1_re,
```

```
      Zim => tmp2_1_im
   );
mul2_2 : entity work.complex_mul
   port map (
      Xre => W2_2_re,
      Xim => W2_2_im,
      Yre => A_re2,
      Yim => A_im2,
      Zre => tmp2_2_re,
      Zim => tmp2_2_im
   );
mul2_3 : entity work.complex_mul
   port map (
      Xre => W2_3_re,
      Xim => W2_3_im,
      Yre => A_re3,
      Yim => A_im3,
      Zre => tmp2_3_re,
      Zim => tmp2_3_im
   );
mul2_4 : entity work.complex_mul
   port map (
      Xre => W2_4_re,
      Xim => W2_4_im,
      Yre => A_re4,
      Yim => A_im4,
      Zre => tmp2_4_re,
      Zim => tmp2_4_im
   );
mul2_5 : entity work.complex_mul
   port map (
```

```
        Xre => W2_5_re,
        Xim => W2_5_im,
        Yre => A_re5,
        Yim => A_im5,
        Zre => tmp2_5_re,
        Zim => tmp2_5_im
    );
mul2_6 : entity work.complex_mul
    port map (
        Xre => W2_6_re,
        Xim => W2_6_im,
        Yre => A_re6,
        Yim => A_im6,
        Zre => tmp2_6_re,
        Zim => tmp2_6_im
    );
mul2_7 : entity work.complex_mul
    port map (
        Xre => W2_7_re,
        Xim => W2_7_im,
        Yre => A_re7,
        Yim => A_im7,
        Zre => tmp2_7_re,
        Zim => tmp2_7_im
    );
mul2_8 : entity work.complex_mul
    port map (
        Xre => W2_8_re,
        Xim => W2_8_im,
        Yre => A_re8,
        Yim => A_im8,
```

```
      Zre => tmp2_8_re,
      Zim => tmp2_8_im
   );
mul2_9 : entity work.complex_mul
   port map (
      Xre => W2_9_re,
      Xim => W2_9_im,
      Yre => A_re9,
      Yim => A_im9,
      Zre => tmp2_9_re,
      Zim => tmp2_9_im
   );
mul2_10 : entity work.complex_mul
   port map (
      Xre => W2_10_re,
      Xim => W2_10_im,
      Yre => A_re10,
      Yim => A_im10,
      Zre => tmp2_10_re,
      Zim => tmp2_10_im
   );
mul2_11 : entity work.complex_mul
   port map (
      Xre => W2_11_re,
      Xim => W2_11_im,
      Yre => A_re11,
      Yim => A_im11,
      Zre => tmp2_11_re,
      Zim => tmp2_11_im
   );
mul2_12 : entity work.complex_mul
```

```
    port map (
        Xre => W2_12_re,
        Xim => W2_12_im,
        Yre => A_re12,
        Yim => A_im12,
        Zre => tmp2_12_re,
        Zim => tmp2_12_im
    );
mul2_13 : entity work.complex_mul
    port map (
        Xre => W2_13_re,
        Xim => W2_13_im,
        Yre => A_re13,
        Yim => A_im13,
        Zre => tmp2_13_re,
        Zim => tmp2_13_im
    );
mul2_14 : entity work.complex_mul
    port map (
        Xre => W2_14_re,
        Xim => W2_14_im,
        Yre => A_re14,
        Yim => A_im14,
        Zre => tmp2_14_re,
        Zim => tmp2_14_im
    );
mul2_15 : entity work.complex_mul
    port map (
        Xre => W2_15_re,
        Xim => W2_15_im,
        Yre => A_re15,
```

```
      Yim => A_im15,
      Zre => tmp2_15_re,
      Zim => tmp2_15_im
   );
mul3_0 : entity work.complex_mul
   port map (
      Xre => W3_0_re,
      Xim => W3_0_im,
      Yre => A_re0,
      Yim => A_im0,
      Zre => tmp3_0_re,
      Zim => tmp3_0_im
   );
mul3_1 : entity work.complex_mul
   port map (
      Xre => W3_1_re,
      Xim => W3_1_im,
      Yre => A_re1,
      Yim => A_im1,
      Zre => tmp3_1_re,
      Zim => tmp3_1_im
   );
mul3_2 : entity work.complex_mul
   port map (
      Xre => W3_2_re,
      Xim => W3_2_im,
      Yre => A_re2,
      Yim => A_im2,
      Zre => tmp3_2_re,
      Zim => tmp3_2_im
   );
```

```
mul3_3 : entity work.complex_mul
   port map (
      Xre => W3_3_re,
      Xim => W3_3_im,
      Yre => A_re3,
      Yim => A_im3,
      Zre => tmp3_3_re,
      Zim => tmp3_3_im
   );
mul3_4 : entity work.complex_mul
   port map (
      Xre => W3_4_re,
      Xim => W3_4_im,
      Yre => A_re4,
      Yim => A_im4,
      Zre => tmp3_4_re,
      Zim => tmp3_4_im
   );
mul3_5 : entity work.complex_mul
   port map (
      Xre => W3_5_re,
      Xim => W3_5_im,
      Yre => A_re5,
      Yim => A_im5,
      Zre => tmp3_5_re,
      Zim => tmp3_5_im
   );
mul3_6 : entity work.complex_mul
   port map (
      Xre => W3_6_re,
      Xim => W3_6_im,
```

```
        Yre => A_re6,
        Yim => A_im6,
        Zre => tmp3_6_re,
        Zim => tmp3_6_im
    );
mul3_7 : entity work.complex_mul
    port map (
        Xre => W3_7_re,
        Xim => W3_7_im,
        Yre => A_re7,
        Yim => A_im7,
        Zre => tmp3_7_re,
        Zim => tmp3_7_im
    );
mul3_8 : entity work.complex_mul
    port map (
        Xre => W3_8_re,
        Xim => W3_8_im,
        Yre => A_re8,
        Yim => A_im8,
        Zre => tmp3_8_re,
        Zim => tmp3_8_im
    );
mul3_9 : entity work.complex_mul
    port map (
        Xre => W3_9_re,
        Xim => W3_9_im,
        Yre => A_re9,
        Yim => A_im9,
        Zre => tmp3_9_re,
        Zim => tmp3_9_im
```

```
    );
mul3_10 : entity work.complex_mul
    port map (
        Xre => W3_10_re,
        Xim => W3_10_im,
        Yre => A_re10,
        Yim => A_im10,
        Zre => tmp3_10_re,
        Zim => tmp3_10_im
    );
mul3_11 : entity work.complex_mul
    port map (
        Xre => W3_11_re,
        Xim => W3_11_im,
        Yre => A_re11,
        Yim => A_im11,
        Zre => tmp3_11_re,
        Zim => tmp3_11_im
    );
mul3_12 : entity work.complex_mul
    port map (
        Xre => W3_12_re,
        Xim => W3_12_im,
        Yre => A_re12,
        Yim => A_im12,
        Zre => tmp3_12_re,
        Zim => tmp3_12_im
    );
mul3_13 : entity work.complex_mul
    port map (
        Xre => W3_13_re,
```

```
        Xim => W3_13_im,
        Yre => A_re13,
        Yim => A_im13,
        Zre => tmp3_13_re,
        Zim => tmp3_13_im
    );
mul3_14 : entity work.complex_mul
    port map (
        Xre => W3_14_re,
        Xim => W3_14_im,
        Yre => A_re14,
        Yim => A_im14,
        Zre => tmp3_14_re,
        Zim => tmp3_14_im
    );
mul3_15 : entity work.complex_mul
    port map (
        Xre => W3_15_re,
        Xim => W3_15_im,
        Yre => A_re15,
        Yim => A_im15,
        Zre => tmp3_15_re,
        Zim => tmp3_15_im
    );
mul4_0 : entity work.complex_mul
    port map (
        Xre => W4_0_re,
        Xim => W4_0_im,
        Yre => A_re0,
        Yim => A_im0,
        Zre => tmp4_0_re,
```

```
        Zim => tmp4_0_im
    );
mul4_1 : entity work.complex_mul
    port map (
        Xre => W4_1_re,
        Xim => W4_1_im,
        Yre => A_re1,
        Yim => A_im1,
        Zre => tmp4_1_re,
        Zim => tmp4_1_im
    );
mul4_2 : entity work.complex_mul
    port map (
        Xre => W4_2_re,
        Xim => W4_2_im,
        Yre => A_re2,
        Yim => A_im2,
        Zre => tmp4_2_re,
        Zim => tmp4_2_im
    );
mul4_3 : entity work.complex_mul
    port map (
        Xre => W4_3_re,
        Xim => W4_3_im,
        Yre => A_re3,
        Yim => A_im3,
        Zre => tmp4_3_re,
        Zim => tmp4_3_im
    );
mul4_4 : entity work.complex_mul
    port map (
```

```
        Xre => W4_4_re,
        Xim => W4_4_im,
        Yre => A_re4,
        Yim => A_im4,
        Zre => tmp4_4_re,
        Zim => tmp4_4_im
    );
mul4_5 : entity work.complex_mul
    port map (
        Xre => W4_5_re,
        Xim => W4_5_im,
        Yre => A_re5,
        Yim => A_im5,
        Zre => tmp4_5_re,
        Zim => tmp4_5_im
    );
mul4_6 : entity work.complex_mul
    port map (
        Xre => W4_6_re,
        Xim => W4_6_im,
        Yre => A_re6,
        Yim => A_im6,
        Zre => tmp4_6_re,
        Zim => tmp4_6_im
    );
mul4_7 : entity work.complex_mul
    port map (
        Xre => W4_7_re,
        Xim => W4_7_im,
        Yre => A_re7,
        Yim => A_im7,
```

```
      Zre => tmp4_7_re,
      Zim => tmp4_7_im
   );
mul4_8 : entity work.complex_mul
   port map (
      Xre => W4_8_re,
      Xim => W4_8_im,
      Yre => A_re8,
      Yim => A_im8,
      Zre => tmp4_8_re,
      Zim => tmp4_8_im
   );
mul4_9 : entity work.complex_mul
   port map (
      Xre => W4_9_re,
      Xim => W4_9_im,
      Yre => A_re9,
      Yim => A_im9,
      Zre => tmp4_9_re,
      Zim => tmp4_9_im
   );
mul4_10 : entity work.complex_mul
   port map (
      Xre => W4_10_re,
      Xim => W4_10_im,
      Yre => A_re10,
      Yim => A_im10,
      Zre => tmp4_10_re,
      Zim => tmp4_10_im
   );
mul4_11 : entity work.complex_mul
```

```
    port map (
        Xre => W4_11_re,
        Xim => W4_11_im,
        Yre => A_re11,
        Yim => A_im11,
        Zre => tmp4_11_re,
        Zim => tmp4_11_im
    );
mul4_12 : entity work.complex_mul
    port map (
        Xre => W4_12_re,
        Xim => W4_12_im,
        Yre => A_re12,
        Yim => A_im12,
        Zre => tmp4_12_re,
        Zim => tmp4_12_im
    );
mul4_13 : entity work.complex_mul
    port map (
        Xre => W4_13_re,
        Xim => W4_13_im,
        Yre => A_re13,
        Yim => A_im13,
        Zre => tmp4_13_re,
        Zim => tmp4_13_im
    );
mul4_14 : entity work.complex_mul
    port map (
        Xre => W4_14_re,
        Xim => W4_14_im,
        Yre => A_re14,
```

```
      Yim => A_im14,
      Zre => tmp4_14_re,
      Zim => tmp4_14_im
   );
mul4_15 : entity work.complex_mul
   port map (
      Xre => W4_15_re,
      Xim => W4_15_im,
      Yre => A_re15,
      Yim => A_im15,
      Zre => tmp4_15_re,
      Zim => tmp4_15_im
   );
mul5_0 : entity work.complex_mul
   port map (
      Xre => W5_0_re,
      Xim => W5_0_im,
      Yre => A_re0,
      Yim => A_im0,
      Zre => tmp5_0_re,
      Zim => tmp5_0_im
   );
mul5_1 : entity work.complex_mul
   port map (
      Xre => W5_1_re,
      Xim => W5_1_im,
      Yre => A_re1,
      Yim => A_im1,
      Zre => tmp5_1_re,
      Zim => tmp5_1_im
   );
```

```
mul5_2 : entity work.complex_mul
   port map (
      Xre => W5_2_re,
      Xim => W5_2_im,
      Yre => A_re2,
      Yim => A_im2,
      Zre => tmp5_2_re,
      Zim => tmp5_2_im
   );
mul5_3 : entity work.complex_mul
   port map (
      Xre => W5_3_re,
      Xim => W5_3_im,
      Yre => A_re3,
      Yim => A_im3,
      Zre => tmp5_3_re,
      Zim => tmp5_3_im
   );
mul5_4 : entity work.complex_mul
   port map (
      Xre => W5_4_re,
      Xim => W5_4_im,
      Yre => A_re4,
      Yim => A_im4,
      Zre => tmp5_4_re,
      Zim => tmp5_4_im
   );
mul5_5 : entity work.complex_mul
   port map (
      Xre => W5_5_re,
      Xim => W5_5_im,
```

```
        Yre => A_re5,
        Yim => A_im5,
        Zre => tmp5_5_re,
        Zim => tmp5_5_im
    );
mul5_6 : entity work.complex_mul
    port map (
        Xre => W5_6_re,
        Xim => W5_6_im,
        Yre => A_re6,
        Yim => A_im6,
        Zre => tmp5_6_re,
        Zim => tmp5_6_im
    );
mul5_7 : entity work.complex_mul
    port map (
        Xre => W5_7_re,
        Xim => W5_7_im,
        Yre => A_re7,
        Yim => A_im7,
        Zre => tmp5_7_re,
        Zim => tmp5_7_im
    );
mul5_8 : entity work.complex_mul
    port map (
        Xre => W5_8_re,
        Xim => W5_8_im,
        Yre => A_re8,
        Yim => A_im8,
        Zre => tmp5_8_re,
        Zim => tmp5_8_im
```

```
    );
mul5_9 : entity work.complex_mul
    port map (
        Xre => W5_9_re,
        Xim => W5_9_im,
        Yre => A_re9,
        Yim => A_im9,
        Zre => tmp5_9_re,
        Zim => tmp5_9_im
    );
mul5_10 : entity work.complex_mul
    port map (
        Xre => W5_10_re,
        Xim => W5_10_im,
        Yre => A_re10,
        Yim => A_im10,
        Zre => tmp5_10_re,
        Zim => tmp5_10_im
    );
mul5_11 : entity work.complex_mul
    port map (
        Xre => W5_11_re,
        Xim => W5_11_im,
        Yre => A_re11,
        Yim => A_im11,
        Zre => tmp5_11_re,
        Zim => tmp5_11_im
    );
mul5_12 : entity work.complex_mul
    port map (
        Xre => W5_12_re,
```

```
        Xim => W5_12_im,
        Yre => A_re12,
        Yim => A_im12,
        Zre => tmp5_12_re,
        Zim => tmp5_12_im
    );
mul5_13 : entity work.complex_mul
    port map (
        Xre => W5_13_re,
        Xim => W5_13_im,
        Yre => A_re13,
        Yim => A_im13,
        Zre => tmp5_13_re,
        Zim => tmp5_13_im
    );
mul5_14 : entity work.complex_mul
    port map (
        Xre => W5_14_re,
        Xim => W5_14_im,
        Yre => A_re14,
        Yim => A_im14,
        Zre => tmp5_14_re,
        Zim => tmp5_14_im
    );
mul5_15 : entity work.complex_mul
    port map (
        Xre => W5_15_re,
        Xim => W5_15_im,
        Yre => A_re15,
        Yim => A_im15,
        Zre => tmp5_15_re,
```

```
      Zim => tmp5_15_im
   );
mul6_0 : entity work.complex_mul
   port map (
      Xre => W6_0_re,
      Xim => W6_0_im,
      Yre => A_re0,
      Yim => A_im0,
      Zre => tmp6_0_re,
      Zim => tmp6_0_im
   );
mul6_1 : entity work.complex_mul
   port map (
      Xre => W6_1_re,
      Xim => W6_1_im,
      Yre => A_re1,
      Yim => A_im1,
      Zre => tmp6_1_re,
      Zim => tmp6_1_im
   );
mul6_2 : entity work.complex_mul
   port map (
      Xre => W6_2_re,
      Xim => W6_2_im,
      Yre => A_re2,
      Yim => A_im2,
      Zre => tmp6_2_re,
      Zim => tmp6_2_im
   );
mul6_3 : entity work.complex_mul
   port map (
```

```
        Xre => W6_3_re,
        Xim => W6_3_im,
        Yre => A_re3,
        Yim => A_im3,
        Zre => tmp6_3_re,
        Zim => tmp6_3_im
    );
mul6_4 : entity work.complex_mul
    port map (
        Xre => W6_4_re,
        Xim => W6_4_im,
        Yre => A_re4,
        Yim => A_im4,
        Zre => tmp6_4_re,
        Zim => tmp6_4_im
    );
mul6_5 : entity work.complex_mul
    port map (
        Xre => W6_5_re,
        Xim => W6_5_im,
        Yre => A_re5,
        Yim => A_im5,
        Zre => tmp6_5_re,
        Zim => tmp6_5_im
    );
mul6_6 : entity work.complex_mul
    port map (
        Xre => W6_6_re,
        Xim => W6_6_im,
        Yre => A_re6,
        Yim => A_im6,
```

```
      Zre => tmp6_6_re,
      Zim => tmp6_6_im
   );
mul6_7 : entity work.complex_mul
   port map (
      Xre => W6_7_re,
      Xim => W6_7_im,
      Yre => A_re7,
      Yim => A_im7,
      Zre => tmp6_7_re,
      Zim => tmp6_7_im
   );
mul6_8 : entity work.complex_mul
   port map (
      Xre => W6_8_re,
      Xim => W6_8_im,
      Yre => A_re8,
      Yim => A_im8,
      Zre => tmp6_8_re,
      Zim => tmp6_8_im
   );
mul6_9 : entity work.complex_mul
   port map (
      Xre => W6_9_re,
      Xim => W6_9_im,
      Yre => A_re9,
      Yim => A_im9,
      Zre => tmp6_9_re,
      Zim => tmp6_9_im
   );
mul6_10 : entity work.complex_mul
```

```
    port map (
        Xre => W6_10_re,
        Xim => W6_10_im,
        Yre => A_re10,
        Yim => A_im10,
        Zre => tmp6_10_re,
        Zim => tmp6_10_im
    );
mul6_11 : entity work.complex_mul
    port map (
        Xre => W6_11_re,
        Xim => W6_11_im,
        Yre => A_re11,
        Yim => A_im11,
        Zre => tmp6_11_re,
        Zim => tmp6_11_im
    );
mul6_12 : entity work.complex_mul
    port map (
        Xre => W6_12_re,
        Xim => W6_12_im,
        Yre => A_re12,
        Yim => A_im12,
        Zre => tmp6_12_re,
        Zim => tmp6_12_im
    );
mul6_13 : entity work.complex_mul
    port map (
        Xre => W6_13_re,
        Xim => W6_13_im,
        Yre => A_re13,
```

```
      Yim => A_im13,
      Zre => tmp6_13_re,
      Zim => tmp6_13_im
   );
mul6_14 : entity work.complex_mul
   port map (
      Xre => W6_14_re,
      Xim => W6_14_im,
      Yre => A_re14,
      Yim => A_im14,
      Zre => tmp6_14_re,
      Zim => tmp6_14_im
   );
mul6_15 : entity work.complex_mul
   port map (
      Xre => W6_15_re,
      Xim => W6_15_im,
      Yre => A_re15,
      Yim => A_im15,
      Zre => tmp6_15_re,
      Zim => tmp6_15_im
   );
mul7_0 : entity work.complex_mul
   port map (
      Xre => W7_0_re,
      Xim => W7_0_im,
      Yre => A_re0,
      Yim => A_im0,
      Zre => tmp7_0_re,
      Zim => tmp7_0_im
   );
```

```
mul7_1 : entity work.complex_mul
   port map (
      Xre => W7_1_re,
      Xim => W7_1_im,
      Yre => A_re1,
      Yim => A_im1,
      Zre => tmp7_1_re,
      Zim => tmp7_1_im
   );
mul7_2 : entity work.complex_mul
   port map (
      Xre => W7_2_re,
      Xim => W7_2_im,
      Yre => A_re2,
      Yim => A_im2,
      Zre => tmp7_2_re,
      Zim => tmp7_2_im
   );
mul7_3 : entity work.complex_mul
   port map (
      Xre => W7_3_re,
      Xim => W7_3_im,
      Yre => A_re3,
      Yim => A_im3,
      Zre => tmp7_3_re,
      Zim => tmp7_3_im
   );
mul7_4 : entity work.complex_mul
   port map (
      Xre => W7_4_re,
      Xim => W7_4_im,
```

```
        Yre => A_re4,
        Yim => A_im4,
        Zre => tmp7_4_re,
        Zim => tmp7_4_im
    );
mul7_5 : entity work.complex_mul
    port map (
        Xre => W7_5_re,
        Xim => W7_5_im,
        Yre => A_re5,
        Yim => A_im5,
        Zre => tmp7_5_re,
        Zim => tmp7_5_im
    );
mul7_6 : entity work.complex_mul
    port map (
        Xre => W7_6_re,
        Xim => W7_6_im,
        Yre => A_re6,
        Yim => A_im6,
        Zre => tmp7_6_re,
        Zim => tmp7_6_im
    );
mul7_7 : entity work.complex_mul
    port map (
        Xre => W7_7_re,
        Xim => W7_7_im,
        Yre => A_re7,
        Yim => A_im7,
        Zre => tmp7_7_re,
        Zim => tmp7_7_im
```

```
    );
mul7_8 : entity work.complex_mul
    port map (
        Xre => W7_8_re,
        Xim => W7_8_im,
        Yre => A_re8,
        Yim => A_im8,
        Zre => tmp7_8_re,
        Zim => tmp7_8_im
    );
mul7_9 : entity work.complex_mul
    port map (
        Xre => W7_9_re,
        Xim => W7_9_im,
        Yre => A_re9,
        Yim => A_im9,
        Zre => tmp7_9_re,
        Zim => tmp7_9_im
    );
mul7_10 : entity work.complex_mul
    port map (
        Xre => W7_10_re,
        Xim => W7_10_im,
        Yre => A_re10,
        Yim => A_im10,
        Zre => tmp7_10_re,
        Zim => tmp7_10_im
    );
mul7_11 : entity work.complex_mul
    port map (
        Xre => W7_11_re,
```

```
        Xim => W7_11_im,
        Yre => A_re11,
        Yim => A_im11,
        Zre => tmp7_11_re,
        Zim => tmp7_11_im
    );
mul7_12 : entity work.complex_mul
    port map (
        Xre => W7_12_re,
        Xim => W7_12_im,
        Yre => A_re12,
        Yim => A_im12,
        Zre => tmp7_12_re,
        Zim => tmp7_12_im
    );
mul7_13 : entity work.complex_mul
    port map (
        Xre => W7_13_re,
        Xim => W7_13_im,
        Yre => A_re13,
        Yim => A_im13,
        Zre => tmp7_13_re,
        Zim => tmp7_13_im
    );
mul7_14 : entity work.complex_mul
    port map (
        Xre => W7_14_re,
        Xim => W7_14_im,
        Yre => A_re14,
        Yim => A_im14,
        Zre => tmp7_14_re,
```

```
      Zim => tmp7_14_im
   );
mul7_15 : entity work.complex_mul
   port map (
      Xre => W7_15_re,
      Xim => W7_15_im,
      Yre => A_re15,
      Yim => A_im15,
      Zre => tmp7_15_re,
      Zim => tmp7_15_im
   );
mul8_0 : entity work.complex_mul
   port map (
      Xre => W8_0_re,
      Xim => W8_0_im,
      Yre => A_re0,
      Yim => A_im0,
      Zre => tmp8_0_re,
      Zim => tmp8_0_im
   );
mul8_1 : entity work.complex_mul
   port map (
      Xre => W8_1_re,
      Xim => W8_1_im,
      Yre => A_re1,
      Yim => A_im1,
      Zre => tmp8_1_re,
      Zim => tmp8_1_im
   );
mul8_2 : entity work.complex_mul
   port map (
```

```
        Xre => W8_2_re,
        Xim => W8_2_im,
        Yre => A_re2,
        Yim => A_im2,
        Zre => tmp8_2_re,
        Zim => tmp8_2_im
    );
mul8_3 : entity work.complex_mul
    port map (
        Xre => W8_3_re,
        Xim => W8_3_im,
        Yre => A_re3,
        Yim => A_im3,
        Zre => tmp8_3_re,
        Zim => tmp8_3_im
    );
mul8_4 : entity work.complex_mul
    port map (
        Xre => W8_4_re,
        Xim => W8_4_im,
        Yre => A_re4,
        Yim => A_im4,
        Zre => tmp8_4_re,
        Zim => tmp8_4_im
    );
mul8_5 : entity work.complex_mul
    port map (
        Xre => W8_5_re,
        Xim => W8_5_im,
        Yre => A_re5,
        Yim => A_im5,
```

```
      Zre => tmp8_5_re,
      Zim => tmp8_5_im
   );
mul8_6 : entity work.complex_mul
   port map (
      Xre => W8_6_re,
      Xim => W8_6_im,
      Yre => A_re6,
      Yim => A_im6,
      Zre => tmp8_6_re,
      Zim => tmp8_6_im
   );
mul8_7 : entity work.complex_mul
   port map (
      Xre => W8_7_re,
      Xim => W8_7_im,
      Yre => A_re7,
      Yim => A_im7,
      Zre => tmp8_7_re,
      Zim => tmp8_7_im
   );
mul8_8 : entity work.complex_mul
   port map (
      Xre => W8_8_re,
      Xim => W8_8_im,
      Yre => A_re8,
      Yim => A_im8,
      Zre => tmp8_8_re,
      Zim => tmp8_8_im
   );
mul8_9 : entity work.complex_mul
```

```
    port map (
        Xre => W8_9_re,
        Xim => W8_9_im,
        Yre => A_re9,
        Yim => A_im9,
        Zre => tmp8_9_re,
        Zim => tmp8_9_im
    );
mul8_10 : entity work.complex_mul
    port map (
        Xre => W8_10_re,
        Xim => W8_10_im,
        Yre => A_re10,
        Yim => A_im10,
        Zre => tmp8_10_re,
        Zim => tmp8_10_im
    );
mul8_11 : entity work.complex_mul
    port map (
        Xre => W8_11_re,
        Xim => W8_11_im,
        Yre => A_re11,
        Yim => A_im11,
        Zre => tmp8_11_re,
        Zim => tmp8_11_im
    );
mul8_12 : entity work.complex_mul
    port map (
        Xre => W8_12_re,
        Xim => W8_12_im,
        Yre => A_re12,
```

```
      Yim => A_im12,
      Zre => tmp8_12_re,
      Zim => tmp8_12_im
   );
mul8_13 : entity work.complex_mul
   port map (
      Xre => W8_13_re,
      Xim => W8_13_im,
      Yre => A_re13,
      Yim => A_im13,
      Zre => tmp8_13_re,
      Zim => tmp8_13_im
   );
mul8_14 : entity work.complex_mul
   port map (
      Xre => W8_14_re,
      Xim => W8_14_im,
      Yre => A_re14,
      Yim => A_im14,
      Zre => tmp8_14_re,
      Zim => tmp8_14_im
   );
mul8_15 : entity work.complex_mul
   port map (
      Xre => W8_15_re,
      Xim => W8_15_im,
      Yre => A_re15,
      Yim => A_im15,
      Zre => tmp8_15_re,
      Zim => tmp8_15_im
   );
```

```
mul9_0 : entity work.complex_mul
   port map (
      Xre => W9_0_re,
      Xim => W9_0_im,
      Yre => A_re0,
      Yim => A_im0,
      Zre => tmp9_0_re,
      Zim => tmp9_0_im
   );
mul9_1 : entity work.complex_mul
   port map (
      Xre => W9_1_re,
      Xim => W9_1_im,
      Yre => A_re1,
      Yim => A_im1,
      Zre => tmp9_1_re,
      Zim => tmp9_1_im
   );
mul9_2 : entity work.complex_mul
   port map (
      Xre => W9_2_re,
      Xim => W9_2_im,
      Yre => A_re2,
      Yim => A_im2,
      Zre => tmp9_2_re,
      Zim => tmp9_2_im
   );
mul9_3 : entity work.complex_mul
   port map (
      Xre => W9_3_re,
      Xim => W9_3_im,
```

```
        Yre => A_re3,
        Yim => A_im3,
        Zre => tmp9_3_re,
        Zim => tmp9_3_im
    );
mul9_4 : entity work.complex_mul
    port map (
        Xre => W9_4_re,
        Xim => W9_4_im,
        Yre => A_re4,
        Yim => A_im4,
        Zre => tmp9_4_re,
        Zim => tmp9_4_im
    );
mul9_5 : entity work.complex_mul
    port map (
        Xre => W9_5_re,
        Xim => W9_5_im,
        Yre => A_re5,
        Yim => A_im5,
        Zre => tmp9_5_re,
        Zim => tmp9_5_im
    );
mul9_6 : entity work.complex_mul
    port map (
        Xre => W9_6_re,
        Xim => W9_6_im,
        Yre => A_re6,
        Yim => A_im6,
        Zre => tmp9_6_re,
        Zim => tmp9_6_im
```

```
    );
mul9_7 : entity work.complex_mul
    port map (
        Xre => W9_7_re,
        Xim => W9_7_im,
        Yre => A_re7,
        Yim => A_im7,
        Zre => tmp9_7_re,
        Zim => tmp9_7_im
    );
mul9_8 : entity work.complex_mul
    port map (
        Xre => W9_8_re,
        Xim => W9_8_im,
        Yre => A_re8,
        Yim => A_im8,
        Zre => tmp9_8_re,
        Zim => tmp9_8_im
    );
mul9_9 : entity work.complex_mul
    port map (
        Xre => W9_9_re,
        Xim => W9_9_im,
        Yre => A_re9,
        Yim => A_im9,
        Zre => tmp9_9_re,
        Zim => tmp9_9_im
    );
mul9_10 : entity work.complex_mul
    port map (
        Xre => W9_10_re,
```

```
        Xim => W9_10_im,
        Yre => A_re10,
        Yim => A_im10,
        Zre => tmp9_10_re,
        Zim => tmp9_10_im
    );
mul9_11 : entity work.complex_mul
    port map (
        Xre => W9_11_re,
        Xim => W9_11_im,
        Yre => A_re11,
        Yim => A_im11,
        Zre => tmp9_11_re,
        Zim => tmp9_11_im
    );
mul9_12 : entity work.complex_mul
    port map (
        Xre => W9_12_re,
        Xim => W9_12_im,
        Yre => A_re12,
        Yim => A_im12,
        Zre => tmp9_12_re,
        Zim => tmp9_12_im
    );
mul9_13 : entity work.complex_mul
    port map (
        Xre => W9_13_re,
        Xim => W9_13_im,
        Yre => A_re13,
        Yim => A_im13,
        Zre => tmp9_13_re,
```

```
      Zim => tmp9_13_im
   );
mul9_14 : entity work.complex_mul
   port map (
      Xre => W9_14_re,
      Xim => W9_14_im,
      Yre => A_re14,
      Yim => A_im14,
      Zre => tmp9_14_re,
      Zim => tmp9_14_im
   );
mul9_15 : entity work.complex_mul
   port map (
      Xre => W9_15_re,
      Xim => W9_15_im,
      Yre => A_re15,
      Yim => A_im15,
      Zre => tmp9_15_re,
      Zim => tmp9_15_im
   );
mul10_0 : entity work.complex_mul
   port map (
      Xre => W10_0_re,
      Xim => W10_0_im,
      Yre => A_re0,
      Yim => A_im0,
      Zre => tmp10_0_re,
      Zim => tmp10_0_im
   );
mul10_1 : entity work.complex_mul
   port map (
```

```
        Xre => W10_1_re,
        Xim => W10_1_im,
        Yre => A_re1,
        Yim => A_im1,
        Zre => tmp10_1_re,
        Zim => tmp10_1_im
    );
mul10_2 : entity work.complex_mul
    port map (
        Xre => W10_2_re,
        Xim => W10_2_im,
        Yre => A_re2,
        Yim => A_im2,
        Zre => tmp10_2_re,
        Zim => tmp10_2_im
    );
mul10_3 : entity work.complex_mul
    port map (
        Xre => W10_3_re,
        Xim => W10_3_im,
        Yre => A_re3,
        Yim => A_im3,
        Zre => tmp10_3_re,
        Zim => tmp10_3_im
    );
mul10_4 : entity work.complex_mul
    port map (
        Xre => W10_4_re,
        Xim => W10_4_im,
        Yre => A_re4,
        Yim => A_im4,
```

```
      Zre => tmp10_4_re,
      Zim => tmp10_4_im
   );
mul10_5 : entity work.complex_mul
   port map (
      Xre => W10_5_re,
      Xim => W10_5_im,
      Yre => A_re5,
      Yim => A_im5,
      Zre => tmp10_5_re,
      Zim => tmp10_5_im
   );
mul10_6 : entity work.complex_mul
   port map (
      Xre => W10_6_re,
      Xim => W10_6_im,
      Yre => A_re6,
      Yim => A_im6,
      Zre => tmp10_6_re,
      Zim => tmp10_6_im
   );
mul10_7 : entity work.complex_mul
   port map (
      Xre => W10_7_re,
      Xim => W10_7_im,
      Yre => A_re7,
      Yim => A_im7,
      Zre => tmp10_7_re,
      Zim => tmp10_7_im
   );
mul10_8 : entity work.complex_mul
```

```
    port map (
        Xre => W10_8_re,
        Xim => W10_8_im,
        Yre => A_re8,
        Yim => A_im8,
        Zre => tmp10_8_re,
        Zim => tmp10_8_im
    );
mul10_9 : entity work.complex_mul
    port map (
        Xre => W10_9_re,
        Xim => W10_9_im,
        Yre => A_re9,
        Yim => A_im9,
        Zre => tmp10_9_re,
        Zim => tmp10_9_im
    );
mul10_10 : entity work.complex_mul
    port map (
        Xre => W10_10_re,
        Xim => W10_10_im,
        Yre => A_re10,
        Yim => A_im10,
        Zre => tmp10_10_re,
        Zim => tmp10_10_im
    );
mul10_11 : entity work.complex_mul
    port map (
        Xre => W10_11_re,
        Xim => W10_11_im,
        Yre => A_re11,
```

```
      Yim => A_im11,
      Zre => tmp10_11_re,
      Zim => tmp10_11_im
   );
mul10_12 : entity work.complex_mul
   port map (
      Xre => W10_12_re,
      Xim => W10_12_im,
      Yre => A_re12,
      Yim => A_im12,
      Zre => tmp10_12_re,
      Zim => tmp10_12_im
   );
mul10_13 : entity work.complex_mul
   port map (
      Xre => W10_13_re,
      Xim => W10_13_im,
      Yre => A_re13,
      Yim => A_im13,
      Zre => tmp10_13_re,
      Zim => tmp10_13_im
   );
mul10_14 : entity work.complex_mul
   port map (
      Xre => W10_14_re,
      Xim => W10_14_im,
      Yre => A_re14,
      Yim => A_im14,
      Zre => tmp10_14_re,
      Zim => tmp10_14_im
   );
```

```
mul10_15 : entity work.complex_mul
   port map (
      Xre => W10_15_re,
      Xim => W10_15_im,
      Yre => A_re15,
      Yim => A_im15,
      Zre => tmp10_15_re,
      Zim => tmp10_15_im
   );
mul11_0 : entity work.complex_mul
   port map (
      Xre => W11_0_re,
      Xim => W11_0_im,
      Yre => A_re0,
      Yim => A_im0,
      Zre => tmp11_0_re,
      Zim => tmp11_0_im
   );
mul11_1 : entity work.complex_mul
   port map (
      Xre => W11_1_re,
      Xim => W11_1_im,
      Yre => A_re1,
      Yim => A_im1,
      Zre => tmp11_1_re,
      Zim => tmp11_1_im
   );
mul11_2 : entity work.complex_mul
   port map (
      Xre => W11_2_re,
      Xim => W11_2_im,
```

```
        Yre => A_re2,
        Yim => A_im2,
        Zre => tmp11_2_re,
        Zim => tmp11_2_im
    );
mul11_3 : entity work.complex_mul
    port map (
        Xre => W11_3_re,
        Xim => W11_3_im,
        Yre => A_re3,
        Yim => A_im3,
        Zre => tmp11_3_re,
        Zim => tmp11_3_im
    );
mul11_4 : entity work.complex_mul
    port map (
        Xre => W11_4_re,
        Xim => W11_4_im,
        Yre => A_re4,
        Yim => A_im4,
        Zre => tmp11_4_re,
        Zim => tmp11_4_im
    );
mul11_5 : entity work.complex_mul
    port map (
        Xre => W11_5_re,
        Xim => W11_5_im,
        Yre => A_re5,
        Yim => A_im5,
        Zre => tmp11_5_re,
        Zim => tmp11_5_im
```

```
    );
mul11_6 : entity work.complex_mul
    port map (
        Xre => W11_6_re,
        Xim => W11_6_im,
        Yre => A_re6,
        Yim => A_im6,
        Zre => tmp11_6_re,
        Zim => tmp11_6_im
    );
mul11_7 : entity work.complex_mul
    port map (
        Xre => W11_7_re,
        Xim => W11_7_im,
        Yre => A_re7,
        Yim => A_im7,
        Zre => tmp11_7_re,
        Zim => tmp11_7_im
    );
mul11_8 : entity work.complex_mul
    port map (
        Xre => W11_8_re,
        Xim => W11_8_im,
        Yre => A_re8,
        Yim => A_im8,
        Zre => tmp11_8_re,
        Zim => tmp11_8_im
    );
mul11_9 : entity work.complex_mul
    port map (
        Xre => W11_9_re,
```

```
        Xim => W11_9_im,
        Yre => A_re9,
        Yim => A_im9,
        Zre => tmp11_9_re,
        Zim => tmp11_9_im
    );
mul11_10 : entity work.complex_mul
    port map (
        Xre => W11_10_re,
        Xim => W11_10_im,
        Yre => A_re10,
        Yim => A_im10,
        Zre => tmp11_10_re,
        Zim => tmp11_10_im
    );
mul11_11 : entity work.complex_mul
    port map (
        Xre => W11_11_re,
        Xim => W11_11_im,
        Yre => A_re11,
        Yim => A_im11,
        Zre => tmp11_11_re,
        Zim => tmp11_11_im
    );
mul11_12 : entity work.complex_mul
    port map (
        Xre => W11_12_re,
        Xim => W11_12_im,
        Yre => A_re12,
        Yim => A_im12,
        Zre => tmp11_12_re,
```

```
      Zim => tmp11_12_im
   );
mul11_13 : entity work.complex_mul
   port map (
      Xre => W11_13_re,
      Xim => W11_13_im,
      Yre => A_re13,
      Yim => A_im13,
      Zre => tmp11_13_re,
      Zim => tmp11_13_im
   );
mul11_14 : entity work.complex_mul
   port map (
      Xre => W11_14_re,
      Xim => W11_14_im,
      Yre => A_re14,
      Yim => A_im14,
      Zre => tmp11_14_re,
      Zim => tmp11_14_im
   );
mul11_15 : entity work.complex_mul
   port map (
      Xre => W11_15_re,
      Xim => W11_15_im,
      Yre => A_re15,
      Yim => A_im15,
      Zre => tmp11_15_re,
      Zim => tmp11_15_im
   );
mul12_0 : entity work.complex_mul
   port map (
```

```
        Xre => W12_0_re,
        Xim => W12_0_im,
        Yre => A_re0,
        Yim => A_im0,
        Zre => tmp12_0_re,
        Zim => tmp12_0_im
    );
mul12_1 : entity work.complex_mul
    port map (
        Xre => W12_1_re,
        Xim => W12_1_im,
        Yre => A_re1,
        Yim => A_im1,
        Zre => tmp12_1_re,
        Zim => tmp12_1_im
    );
mul12_2 : entity work.complex_mul
    port map (
        Xre => W12_2_re,
        Xim => W12_2_im,
        Yre => A_re2,
        Yim => A_im2,
        Zre => tmp12_2_re,
        Zim => tmp12_2_im
    );
mul12_3 : entity work.complex_mul
    port map (
        Xre => W12_3_re,
        Xim => W12_3_im,
        Yre => A_re3,
        Yim => A_im3,
```

```
      Zre => tmp12_3_re,
      Zim => tmp12_3_im
   );
mul12_4 : entity work.complex_mul
   port map (
      Xre => W12_4_re,
      Xim => W12_4_im,
      Yre => A_re4,
      Yim => A_im4,
      Zre => tmp12_4_re,
      Zim => tmp12_4_im
   );
mul12_5 : entity work.complex_mul
   port map (
      Xre => W12_5_re,
      Xim => W12_5_im,
      Yre => A_re5,
      Yim => A_im5,
      Zre => tmp12_5_re,
      Zim => tmp12_5_im
   );
mul12_6 : entity work.complex_mul
   port map (
      Xre => W12_6_re,
      Xim => W12_6_im,
      Yre => A_re6,
      Yim => A_im6,
      Zre => tmp12_6_re,
      Zim => tmp12_6_im
   );
mul12_7 : entity work.complex_mul
```

```
    port map (
        Xre => W12_7_re,
        Xim => W12_7_im,
        Yre => A_re7,
        Yim => A_im7,
        Zre => tmp12_7_re,
        Zim => tmp12_7_im
    );
mul12_8 : entity work.complex_mul
    port map (
        Xre => W12_8_re,
        Xim => W12_8_im,
        Yre => A_re8,
        Yim => A_im8,
        Zre => tmp12_8_re,
        Zim => tmp12_8_im
    );
mul12_9 : entity work.complex_mul
    port map (
        Xre => W12_9_re,
        Xim => W12_9_im,
        Yre => A_re9,
        Yim => A_im9,
        Zre => tmp12_9_re,
        Zim => tmp12_9_im
    );
mul12_10 : entity work.complex_mul
    port map (
        Xre => W12_10_re,
        Xim => W12_10_im,
        Yre => A_re10,
```

```
        Yim => A_im10,
        Zre => tmp12_10_re,
        Zim => tmp12_10_im
    );
mul12_11 : entity work.complex_mul
    port map (
        Xre => W12_11_re,
        Xim => W12_11_im,
        Yre => A_re11,
        Yim => A_im11,
        Zre => tmp12_11_re,
        Zim => tmp12_11_im
    );
mul12_12 : entity work.complex_mul
    port map (
        Xre => W12_12_re,
        Xim => W12_12_im,
        Yre => A_re12,
        Yim => A_im12,
        Zre => tmp12_12_re,
        Zim => tmp12_12_im
    );
mul12_13 : entity work.complex_mul
    port map (
        Xre => W12_13_re,
        Xim => W12_13_im,
        Yre => A_re13,
        Yim => A_im13,
        Zre => tmp12_13_re,
        Zim => tmp12_13_im
    );
```

```
mul12_14 : entity work.complex_mul
   port map (
      Xre => W12_14_re,
      Xim => W12_14_im,
      Yre => A_re14,
      Yim => A_im14,
      Zre => tmp12_14_re,
      Zim => tmp12_14_im
   );
mul12_15 : entity work.complex_mul
   port map (
      Xre => W12_15_re,
      Xim => W12_15_im,
      Yre => A_re15,
      Yim => A_im15,
      Zre => tmp12_15_re,
      Zim => tmp12_15_im
   );
mul13_0 : entity work.complex_mul
   port map (
      Xre => W13_0_re,
      Xim => W13_0_im,
      Yre => A_re0,
      Yim => A_im0,
      Zre => tmp13_0_re,
      Zim => tmp13_0_im
   );
mul13_1 : entity work.complex_mul
   port map (
      Xre => W13_1_re,
      Xim => W13_1_im,
```

```
        Yre => A_re1,
        Yim => A_im1,
        Zre => tmp13_1_re,
        Zim => tmp13_1_im
    );
mul13_2 : entity work.complex_mul
    port map (
        Xre => W13_2_re,
        Xim => W13_2_im,
        Yre => A_re2,
        Yim => A_im2,
        Zre => tmp13_2_re,
        Zim => tmp13_2_im
    );
mul13_3 : entity work.complex_mul
    port map (
        Xre => W13_3_re,
        Xim => W13_3_im,
        Yre => A_re3,
        Yim => A_im3,
        Zre => tmp13_3_re,
        Zim => tmp13_3_im
    );
mul13_4 : entity work.complex_mul
    port map (
        Xre => W13_4_re,
        Xim => W13_4_im,
        Yre => A_re4,
        Yim => A_im4,
        Zre => tmp13_4_re,
        Zim => tmp13_4_im
```

```
    );
mul13_5 : entity work.complex_mul
    port map (
        Xre => W13_5_re,
        Xim => W13_5_im,
        Yre => A_re5,
        Yim => A_im5,
        Zre => tmp13_5_re,
        Zim => tmp13_5_im
    );
mul13_6 : entity work.complex_mul
    port map (
        Xre => W13_6_re,
        Xim => W13_6_im,
        Yre => A_re6,
        Yim => A_im6,
        Zre => tmp13_6_re,
        Zim => tmp13_6_im
    );
mul13_7 : entity work.complex_mul
    port map (
        Xre => W13_7_re,
        Xim => W13_7_im,
        Yre => A_re7,
        Yim => A_im7,
        Zre => tmp13_7_re,
        Zim => tmp13_7_im
    );
mul13_8 : entity work.complex_mul
    port map (
        Xre => W13_8_re,
```

```
        Xim => W13_8_im,
        Yre => A_re8,
        Yim => A_im8,
        Zre => tmp13_8_re,
        Zim => tmp13_8_im
    );
mul13_9 : entity work.complex_mul
    port map (
        Xre => W13_9_re,
        Xim => W13_9_im,
        Yre => A_re9,
        Yim => A_im9,
        Zre => tmp13_9_re,
        Zim => tmp13_9_im
    );
mul13_10 : entity work.complex_mul
    port map (
        Xre => W13_10_re,
        Xim => W13_10_im,
        Yre => A_re10,
        Yim => A_im10,
        Zre => tmp13_10_re,
        Zim => tmp13_10_im
    );
mul13_11 : entity work.complex_mul
    port map (
        Xre => W13_11_re,
        Xim => W13_11_im,
        Yre => A_re11,
        Yim => A_im11,
        Zre => tmp13_11_re,
```

```
      Zim => tmp13_11_im
   );
mul13_12 : entity work.complex_mul
   port map (
      Xre => W13_12_re,
      Xim => W13_12_im,
      Yre => A_re12,
      Yim => A_im12,
      Zre => tmp13_12_re,
      Zim => tmp13_12_im
   );
mul13_13 : entity work.complex_mul
   port map (
      Xre => W13_13_re,
      Xim => W13_13_im,
      Yre => A_re13,
      Yim => A_im13,
      Zre => tmp13_13_re,
      Zim => tmp13_13_im
   );
mul13_14 : entity work.complex_mul
   port map (
      Xre => W13_14_re,
      Xim => W13_14_im,
      Yre => A_re14,
      Yim => A_im14,
      Zre => tmp13_14_re,
      Zim => tmp13_14_im
   );
mul13_15 : entity work.complex_mul
   port map (
```

```
        Xre => W13_15_re,
        Xim => W13_15_im,
        Yre => A_re15,
        Yim => A_im15,
        Zre => tmp13_15_re,
        Zim => tmp13_15_im
    );
mul14_0 : entity work.complex_mul
    port map (
        Xre => W14_0_re,
        Xim => W14_0_im,
        Yre => A_re0,
        Yim => A_im0,
        Zre => tmp14_0_re,
        Zim => tmp14_0_im
    );
mul14_1 : entity work.complex_mul
    port map (
        Xre => W14_1_re,
        Xim => W14_1_im,
        Yre => A_re1,
        Yim => A_im1,
        Zre => tmp14_1_re,
        Zim => tmp14_1_im
    );
mul14_2 : entity work.complex_mul
    port map (
        Xre => W14_2_re,
        Xim => W14_2_im,
        Yre => A_re2,
        Yim => A_im2,
```

```
      Zre => tmp14_2_re,
      Zim => tmp14_2_im
   );
mul14_3 : entity work.complex_mul
   port map (
      Xre => W14_3_re,
      Xim => W14_3_im,
      Yre => A_re3,
      Yim => A_im3,
      Zre => tmp14_3_re,
      Zim => tmp14_3_im
   );
mul14_4 : entity work.complex_mul
   port map (
      Xre => W14_4_re,
      Xim => W14_4_im,
      Yre => A_re4,
      Yim => A_im4,
      Zre => tmp14_4_re,
      Zim => tmp14_4_im
   );
mul14_5 : entity work.complex_mul
   port map (
      Xre => W14_5_re,
      Xim => W14_5_im,
      Yre => A_re5,
      Yim => A_im5,
      Zre => tmp14_5_re,
      Zim => tmp14_5_im
   );
mul14_6 : entity work.complex_mul
```

```
    port map (
        Xre => W14_6_re,
        Xim => W14_6_im,
        Yre => A_re6,
        Yim => A_im6,
        Zre => tmp14_6_re,
        Zim => tmp14_6_im
    );
mul14_7 : entity work.complex_mul
    port map (
        Xre => W14_7_re,
        Xim => W14_7_im,
        Yre => A_re7,
        Yim => A_im7,
        Zre => tmp14_7_re,
        Zim => tmp14_7_im
    );
mul14_8 : entity work.complex_mul
    port map (
        Xre => W14_8_re,
        Xim => W14_8_im,
        Yre => A_re8,
        Yim => A_im8,
        Zre => tmp14_8_re,
        Zim => tmp14_8_im
    );
mul14_9 : entity work.complex_mul
    port map (
        Xre => W14_9_re,
        Xim => W14_9_im,
        Yre => A_re9,
```

```
      Yim => A_im9,
      Zre => tmp14_9_re,
      Zim => tmp14_9_im
   );
mul14_10 : entity work.complex_mul
   port map (
      Xre => W14_10_re,
      Xim => W14_10_im,
      Yre => A_re10,
      Yim => A_im10,
      Zre => tmp14_10_re,
      Zim => tmp14_10_im
   );
mul14_11 : entity work.complex_mul
   port map (
      Xre => W14_11_re,
      Xim => W14_11_im,
      Yre => A_re11,
      Yim => A_im11,
      Zre => tmp14_11_re,
      Zim => tmp14_11_im
   );
mul14_12 : entity work.complex_mul
   port map (
      Xre => W14_12_re,
      Xim => W14_12_im,
      Yre => A_re12,
      Yim => A_im12,
      Zre => tmp14_12_re,
      Zim => tmp14_12_im
   );
```

```
mul14_13 : entity work.complex_mul
   port map (
      Xre => W14_13_re,
      Xim => W14_13_im,
      Yre => A_re13,
      Yim => A_im13,
      Zre => tmp14_13_re,
      Zim => tmp14_13_im
   );
mul14_14 : entity work.complex_mul
   port map (
      Xre => W14_14_re,
      Xim => W14_14_im,
      Yre => A_re14,
      Yim => A_im14,
      Zre => tmp14_14_re,
      Zim => tmp14_14_im
   );
mul14_15 : entity work.complex_mul
   port map (
      Xre => W14_15_re,
      Xim => W14_15_im,
      Yre => A_re15,
      Yim => A_im15,
      Zre => tmp14_15_re,
      Zim => tmp14_15_im
   );
mul15_0 : entity work.complex_mul
   port map (
      Xre => W15_0_re,
      Xim => W15_0_im,
```

```
        Yre => A_re0,
        Yim => A_im0,
        Zre => tmp15_0_re,
        Zim => tmp15_0_im
    );
mul15_1 : entity work.complex_mul
    port map (
        Xre => W15_1_re,
        Xim => W15_1_im,
        Yre => A_re1,
        Yim => A_im1,
        Zre => tmp15_1_re,
        Zim => tmp15_1_im
    );
mul15_2 : entity work.complex_mul
    port map (
        Xre => W15_2_re,
        Xim => W15_2_im,
        Yre => A_re2,
        Yim => A_im2,
        Zre => tmp15_2_re,
        Zim => tmp15_2_im
    );
mul15_3 : entity work.complex_mul
    port map (
        Xre => W15_3_re,
        Xim => W15_3_im,
        Yre => A_re3,
        Yim => A_im3,
        Zre => tmp15_3_re,
        Zim => tmp15_3_im
```

```
    );
mul15_4 : entity work.complex_mul
    port map (
        Xre => W15_4_re,
        Xim => W15_4_im,
        Yre => A_re4,
        Yim => A_im4,
        Zre => tmp15_4_re,
        Zim => tmp15_4_im
    );
mul15_5 : entity work.complex_mul
    port map (
        Xre => W15_5_re,
        Xim => W15_5_im,
        Yre => A_re5,
        Yim => A_im5,
        Zre => tmp15_5_re,
        Zim => tmp15_5_im
    );
mul15_6 : entity work.complex_mul
    port map (
        Xre => W15_6_re,
        Xim => W15_6_im,
        Yre => A_re6,
        Yim => A_im6,
        Zre => tmp15_6_re,
        Zim => tmp15_6_im
    );
mul15_7 : entity work.complex_mul
    port map (
        Xre => W15_7_re,
```

```
        Xim => W15_7_im,
        Yre => A_re7,
        Yim => A_im7,
        Zre => tmp15_7_re,
        Zim => tmp15_7_im
    );
mul15_8 : entity work.complex_mul
    port map (
        Xre => W15_8_re,
        Xim => W15_8_im,
        Yre => A_re8,
        Yim => A_im8,
        Zre => tmp15_8_re,
        Zim => tmp15_8_im
    );
mul15_9 : entity work.complex_mul
    port map (
        Xre => W15_9_re,
        Xim => W15_9_im,
        Yre => A_re9,
        Yim => A_im9,
        Zre => tmp15_9_re,
        Zim => tmp15_9_im
    );
mul15_10 : entity work.complex_mul
    port map (
        Xre => W15_10_re,
        Xim => W15_10_im,
        Yre => A_re10,
        Yim => A_im10,
        Zre => tmp15_10_re,
```

```
      Zim => tmp15_10_im
   );
mul15_11 : entity work.complex_mul
   port map (
      Xre => W15_11_re,
      Xim => W15_11_im,
      Yre => A_re11,
      Yim => A_im11,
      Zre => tmp15_11_re,
      Zim => tmp15_11_im
   );
mul15_12 : entity work.complex_mul
   port map (
      Xre => W15_12_re,
      Xim => W15_12_im,
      Yre => A_re12,
      Yim => A_im12,
      Zre => tmp15_12_re,
      Zim => tmp15_12_im
   );
mul15_13 : entity work.complex_mul
   port map (
      Xre => W15_13_re,
      Xim => W15_13_im,
      Yre => A_re13,
      Yim => A_im13,
      Zre => tmp15_13_re,
      Zim => tmp15_13_im
   );
mul15_14 : entity work.complex_mul
   port map (
```

```
      Xre => W15_14_re,
      Xim => W15_14_im,
      Yre => A_re14,
      Yim => A_im14,
      Zre => tmp15_14_re,
      Zim => tmp15_14_im
   );
mul15_15 : entity work.complex_mul
   port map (
      Xre => W15_15_re,
      Xim => W15_15_im,
      Yre => A_re15,
      Yim => A_im15,
      Zre => tmp15_15_re,
      Zim => tmp15_15_im
   );
process
begin
C_re0 <= tmp0_0_re + tmp0_1_re + tmp0_2_re + tmp0_3_re + tmp0_4_re + tmp0_5_re + tmp0_6_re
+ tmp0_7_re + tmp0_8_re + tmp0_9_re + tmp0_10_re + tmp0_11_re + tmp0_12_re + tmp0_13_re +
tmp0_14_re + tmp0_15_re;
C_im0 <= tmp0_0_im + tmp0_1_im + tmp0_2_im + tmp0_3_im + tmp0_4_im + tmp0_5_im +
tmp0_6_im + tmp0_7_im + tmp0_8_im + tmp0_9_im + tmp0_10_im + tmp0_11_im + tmp0_12_im +
tmp0_13_im + tmp0_14_im + tmp0_15_im;
wait on tmp0_0_re'event;
end process;
process
begin
C_re1 <= tmp1_0_re + tmp1_1_re + tmp1_2_re + tmp1_3_re + tmp1_4_re + tmp1_5_re + tmp1_6_re
+ tmp1_7_re + tmp1_8_re + tmp1_9_re + tmp1_10_re + tmp1_11_re + tmp1_12_re + tmp1_13_re +
tmp1_14_re + tmp1_15_re;
C_im1 <= tmp1_0_im + tmp1_1_im + tmp1_2_im + tmp1_3_im + tmp1_4_im + tmp1_5_im +
tmp1_6_im + tmp1_7_im + tmp1_8_im + tmp1_9_im + tmp1_10_im + tmp1_11_im + tmp1_12_im +
tmp1_13_im + tmp1_14_im + tmp1_15_im;
```

```
wait on tmp1_0_re'event;

end process;

process

begin

C_re2 <= tmp2_0_re + tmp2_1_re + tmp2_2_re + tmp2_3_re + tmp2_4_re + tmp2_5_re + tmp2_6_re
+ tmp2_7_re + tmp2_8_re + tmp2_9_re + tmp2_10_re + tmp2_11_re + tmp2_12_re + tmp2_13_re +
tmp2_14_re + tmp2_15_re;

C_im2 <= tmp2_0_im + tmp2_1_im + tmp2_2_im + tmp2_3_im + tmp2_4_im + tmp2_5_im +
tmp2_6_im + tmp2_7_im + tmp2_8_im + tmp2_9_im + tmp2_10_im + tmp2_11_im + tmp2_12_im +
tmp2_13_im + tmp2_14_im + tmp2_15_im;

wait on tmp2_0_re'event;

end process;

process

begin

C_re3 <= tmp3_0_re + tmp3_1_re + tmp3_2_re + tmp3_3_re + tmp3_4_re + tmp3_5_re + tmp3_6_re
+ tmp3_7_re + tmp3_8_re + tmp3_9_re + tmp3_10_re + tmp3_11_re + tmp3_12_re + tmp3_13_re +
tmp3_14_re + tmp3_15_re;

C_im3 <= tmp3_0_im + tmp3_1_im + tmp3_2_im + tmp3_3_im + tmp3_4_im + tmp3_5_im +
tmp3_6_im + tmp3_7_im + tmp3_8_im + tmp3_9_im + tmp3_10_im + tmp3_11_im + tmp3_12_im +
tmp3_13_im + tmp3_14_im + tmp3_15_im;

wait on tmp3_0_re'event;

end process;

process

begin

C_re4 <= tmp4_0_re + tmp4_1_re + tmp4_2_re + tmp4_3_re + tmp4_4_re + tmp4_5_re + tmp4_6_re
+ tmp4_7_re + tmp4_8_re + tmp4_9_re + tmp4_10_re + tmp4_11_re + tmp4_12_re + tmp4_13_re +
tmp4_14_re + tmp4_15_re;

C_im4 <= tmp4_0_im + tmp4_1_im + tmp4_2_im + tmp4_3_im + tmp4_4_im + tmp4_5_im +
tmp4_6_im + tmp4_7_im + tmp4_8_im + tmp4_9_im + tmp4_10_im + tmp4_11_im + tmp4_12_im +
tmp4_13_im + tmp4_14_im + tmp4_15_im;

wait on tmp4_0_re'event;

end process;

process

begin
```

```
C_re5 <= tmp5_0_re + tmp5_1_re + tmp5_2_re + tmp5_3_re + tmp5_4_re + tmp5_5_re + tmp5_6_re
+ tmp5_7_re + tmp5_8_re + tmp5_9_re + tmp5_10_re + tmp5_11_re + tmp5_12_re + tmp5_13_re +
tmp5_14_re + tmp5_15_re;

C_im5 <= tmp5_0_im + tmp5_1_im + tmp5_2_im + tmp5_3_im + tmp5_4_im + tmp5_5_im +
tmp5_6_im + tmp5_7_im + tmp5_8_im + tmp5_9_im + tmp5_10_im + tmp5_11_im + tmp5_12_im +
tmp5_13_im + tmp5_14_im + tmp5_15_im;

wait on tmp5_0_re'event;

end process;

process

begin

C_re6 <= tmp6_0_re + tmp6_1_re + tmp6_2_re + tmp6_3_re + tmp6_4_re + tmp6_5_re + tmp6_6_re
+ tmp6_7_re + tmp6_8_re + tmp6_9_re + tmp6_10_re + tmp6_11_re + tmp6_12_re + tmp6_13_re +
tmp6_14_re + tmp6_15_re;

C_im6 <= tmp6_0_im + tmp6_1_im + tmp6_2_im + tmp6_3_im + tmp6_4_im + tmp6_5_im +
tmp6_6_im + tmp6_7_im + tmp6_8_im + tmp6_9_im + tmp6_10_im + tmp6_11_im + tmp6_12_im +
tmp6_13_im + tmp6_14_im + tmp6_15_im;

wait on tmp6_0_re'event;

end process;

process

begin

C_re7 <= tmp7_0_re + tmp7_1_re + tmp7_2_re + tmp7_3_re + tmp7_4_re + tmp7_5_re + tmp7_6_re
+ tmp7_7_re + tmp7_8_re + tmp7_9_re + tmp7_10_re + tmp7_11_re + tmp7_12_re + tmp7_13_re +
tmp7_14_re + tmp7_15_re;

C_im7 <= tmp7_0_im + tmp7_1_im + tmp7_2_im + tmp7_3_im + tmp7_4_im + tmp7_5_im +
tmp7_6_im + tmp7_7_im + tmp7_8_im + tmp7_9_im + tmp7_10_im + tmp7_11_im + tmp7_12_im +
tmp7_13_im + tmp7_14_im + tmp7_15_im;

wait on tmp7_0_re'event;

end process;

process

begin

C_re8 <= tmp8_0_re + tmp8_1_re + tmp8_2_re + tmp8_3_re + tmp8_4_re + tmp8_5_re + tmp8_6_re
+ tmp8_7_re + tmp8_8_re + tmp8_9_re + tmp8_10_re + tmp8_11_re + tmp8_12_re + tmp8_13_re +
tmp8_14_re + tmp8_15_re;

C_im8 <= tmp8_0_im + tmp8_1_im + tmp8_2_im + tmp8_3_im + tmp8_4_im + tmp8_5_im +
tmp8_6_im + tmp8_7_im + tmp8_8_im + tmp8_9_im + tmp8_10_im + tmp8_11_im + tmp8_12_im +
tmp8_13_im + tmp8_14_im + tmp8_15_im;

wait on tmp8_0_re'event;
```

```
end process;

process

begin

C_re9 <= tmp9_0_re + tmp9_1_re + tmp9_2_re + tmp9_3_re + tmp9_4_re + tmp9_5_re + tmp9_6_re
+ tmp9_7_re + tmp9_8_re + tmp9_9_re + tmp9_10_re + tmp9_11_re + tmp9_12_re + tmp9_13_re +
tmp9_14_re + tmp9_15_re;

C_im9 <= tmp9_0_im + tmp9_1_im + tmp9_2_im + tmp9_3_im + tmp9_4_im + tmp9_5_im +
tmp9_6_im + tmp9_7_im + tmp9_8_im + tmp9_9_im + tmp9_10_im + tmp9_11_im + tmp9_12_im +
tmp9_13_im + tmp9_14_im + tmp9_15_im;

wait on tmp9_0_re'event;

end process;

process

begin

C_re10 <= tmp10_0_re + tmp10_1_re + tmp10_2_re + tmp10_3_re + tmp10_4_re + tmp10_5_re +
tmp10_6_re + tmp10_7_re + tmp10_8_re + tmp10_9_re + tmp10_10_re + tmp10_11_re +
tmp10_12_re + tmp10_13_re + tmp10_14_re + tmp10_15_re;

C_im10 <= tmp10_0_im + tmp10_1_im + tmp10_2_im + tmp10_3_im + tmp10_4_im + tmp10_5_im +
tmp10_6_im + tmp10_7_im + tmp10_8_im + tmp10_9_im + tmp10_10_im + tmp10_11_im +
tmp10_12_im + tmp10_13_im + tmp10_14_im + tmp10_15_im;

wait on tmp10_0_re'event;

end process;

process

begin

C_re11 <= tmp11_0_re + tmp11_1_re + tmp11_2_re + tmp11_3_re + tmp11_4_re + tmp11_5_re +
tmp11_6_re + tmp11_7_re + tmp11_8_re + tmp11_9_re + tmp11_10_re + tmp11_11_re +
tmp11_12_re + tmp11_13_re + tmp11_14_re + tmp11_15_re;

C_im11 <= tmp11_0_im + tmp11_1_im + tmp11_2_im + tmp11_3_im + tmp11_4_im + tmp11_5_im +
tmp11_6_im + tmp11_7_im + tmp11_8_im + tmp11_9_im + tmp11_10_im + tmp11_11_im +
tmp11_12_im + tmp11_13_im + tmp11_14_im + tmp11_15_im;

wait on tmp11_0_re'event;

end process;

process

begin

C_re12 <= tmp12_0_re + tmp12_1_re + tmp12_2_re + tmp12_3_re + tmp12_4_re + tmp12_5_re +
tmp12_6_re + tmp12_7_re + tmp12_8_re + tmp12_9_re + tmp12_10_re + tmp12_11_re +
tmp12_12_re + tmp12_13_re + tmp12_14_re + tmp12_15_re;
```

```
C_im12 <= tmp12_0_im + tmp12_1_im + tmp12_2_im + tmp12_3_im + tmp12_4_im + tmp12_5_im +
tmp12_6_im + tmp12_7_im + tmp12_8_im + tmp12_9_im + tmp12_10_im + tmp12_11_im +
tmp12_12_im + tmp12_13_im + tmp12_14_im + tmp12_15_im;

wait on tmp12_0_re'event;

end process;

process

begin

C_re13 <= tmp13_0_re + tmp13_1_re + tmp13_2_re + tmp13_3_re + tmp13_4_re + tmp13_5_re +
tmp13_6_re + tmp13_7_re + tmp13_8_re + tmp13_9_re + tmp13_10_re + tmp13_11_re +
tmp13_12_re + tmp13_13_re + tmp13_14_re + tmp13_15_re;

C_im13 <= tmp13_0_im + tmp13_1_im + tmp13_2_im + tmp13_3_im + tmp13_4_im + tmp13_5_im +
tmp13_6_im + tmp13_7_im + tmp13_8_im + tmp13_9_im + tmp13_10_im + tmp13_11_im +
tmp13_12_im + tmp13_13_im + tmp13_14_im + tmp13_15_im;

wait on tmp13_0_re'event;

end process;

process

begin

C_re14 <= tmp14_0_re + tmp14_1_re + tmp14_2_re + tmp14_3_re + tmp14_4_re + tmp14_5_re +
tmp14_6_re + tmp14_7_re + tmp14_8_re + tmp14_9_re + tmp14_10_re + tmp14_11_re +
tmp14_12_re + tmp14_13_re + tmp14_14_re + tmp14_15_re;

C_im14 <= tmp14_0_im + tmp14_1_im + tmp14_2_im + tmp14_3_im + tmp14_4_im + tmp14_5_im +
tmp14_6_im + tmp14_7_im + tmp14_8_im + tmp14_9_im + tmp14_10_im + tmp14_11_im +
tmp14_12_im + tmp14_13_im + tmp14_14_im + tmp14_15_im;

wait on tmp14_0_re'event;

end process;

process

begin

C_re15 <= tmp15_0_re + tmp15_1_re + tmp15_2_re + tmp15_3_re + tmp15_4_re + tmp15_5_re +
tmp15_6_re + tmp15_7_re + tmp15_8_re + tmp15_9_re + tmp15_10_re + tmp15_11_re +
tmp15_12_re + tmp15_13_re + tmp15_14_re + tmp15_15_re;

C_im15 <= tmp15_0_im + tmp15_1_im + tmp15_2_im + tmp15_3_im + tmp15_4_im + tmp15_5_im +
tmp15_6_im + tmp15_7_im + tmp15_8_im + tmp15_9_im + tmp15_10_im + tmp15_11_im +
tmp15_12_im + tmp15_13_im + tmp15_14_im + tmp15_15_im;

wait on tmp15_0_re'event;

end process;

end architecture Behavioral;
```

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity PWM is
    port (
        clk : in std_logic;
        reset : in std_logic;
        duty_cycle : in integer;
        pwm_out : out std_logic
    );
end entity PWM;

architecture behavior of PWM is
    signal counter: integer range 0 to 1000000;
begin
    process(clk, reset)
    begin
        if reset = '1' then
            counter <= 0;
            pwm_out <= '0';
        elsif rising_edge(clk) then
            if counter < 1000000 then
                counter <= counter + 1;
            else
                counter <= 0;
            end if;

            if counter < duty_cycle then
```

```
            pwm_out <= '1';
        else
            pwm_out <= '0';
        end if;
    end if;
  end process;
end behavior;
-- ================================
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity main is
  port (
    clk : in std_logic;
    reset : in std_logic;
    led_pwm0 : out std_logic;
    led_pwm1 : out std_logic;
    led_pwm2 : out std_logic;
    led_pwm3 : out std_logic;
    led_pwm4 : out std_logic;
    led_pwm5 : out std_logic;
    led_pwm6 : out std_logic;
    led_pwm7 : out std_logic;
    led_pwm8 : out std_logic;
    led_pwm9 : out std_logic;
    led_pwm10 : out std_logic;
    led_pwm11 : out std_logic;
    led_pwm12 : out std_logic;
    led_pwm13 : out std_logic;
    led_pwm14 : out std_logic;
```

```
        led_pwm15 : out std_logic
    );
end entity main;

architecture behavior of main is
    component DFT16
        port (
            A_re0, A_im0 :in integer;
            C_re0, C_im0 :out integer;
            A_re1, A_im1 :in integer;
            C_re1, C_im1 :out integer;
            A_re2, A_im2 :in integer;
            C_re2, C_im2 :out integer;
            A_re3, A_im3 :in integer;
            C_re3, C_im3 :out integer;
            A_re4, A_im4 :in integer;
            C_re4, C_im4 :out integer;
            A_re5, A_im5 :in integer;
            C_re5, C_im5 :out integer;
            A_re6, A_im6 :in integer;
            C_re6, C_im6 :out integer;
            A_re7, A_im7 :in integer;
            C_re7, C_im7 :out integer;
            A_re8, A_im8 :in integer;
            C_re8, C_im8 :out integer;
            A_re9, A_im9 :in integer;
            C_re9, C_im9 :out integer;
            A_re10, A_im10 :in integer;
            C_re10, C_im10 :out integer;
            A_re11, A_im11 :in integer;
            C_re11, C_im11 :out integer;
```

```
        A_re12, A_im12 :in integer;
        C_re12, C_im12 :out integer;
        A_re13, A_im13 :in integer;
        C_re13, C_im13 :out integer;
        A_re14, A_im14 :in integer;
        C_re14, C_im14 :out integer;
        A_re15, A_im15 :in integer;
        C_re15, C_im15 :out integer
    );
end component;

component PWM
    port (
        clk : in std_logic;
        reset : in std_logic;
        duty_cycle : in integer;
        pwm_out : out std_logic
    );
end component;

signal real0, imag0, led_pwm_temp0 : integer;
signal input_re0, input_im0 : integer;
signal real1, imag1, led_pwm_temp1 : integer;
signal input_re1, input_im1 : integer;
signal real2, imag2, led_pwm_temp2 : integer;
signal input_re2, input_im2 : integer;
signal real3, imag3, led_pwm_temp3 : integer;
signal input_re3, input_im3 : integer;
signal real4, imag4, led_pwm_temp4 : integer;
signal input_re4, input_im4 : integer;
signal real5, imag5, led_pwm_temp5 : integer;
```

```
    signal input_re5, input_im5 : integer;
    signal real6, imag6, led_pwm_temp6 : integer;
    signal input_re6, input_im6 : integer;
    signal real7, imag7, led_pwm_temp7 : integer;
    signal input_re7, input_im7 : integer;
    signal real8, imag8, led_pwm_temp8 : integer;
    signal input_re8, input_im8 : integer;
    signal real9, imag9, led_pwm_temp9 : integer;
    signal input_re9, input_im9 : integer;
    signal real10, imag10, led_pwm_temp10 : integer;
    signal input_re10, input_im10 : integer;
    signal real11, imag11, led_pwm_temp11 : integer;
    signal input_re11, input_im11 : integer;
    signal real12, imag12, led_pwm_temp12 : integer;
    signal input_re12, input_im12 : integer;
    signal real13, imag13, led_pwm_temp13 : integer;
    signal input_re13, input_im13 : integer;
    signal real14, imag14, led_pwm_temp14 : integer;
    signal input_re14, input_im14 : integer;
    signal real15, imag15, led_pwm_temp15 : integer;
    signal input_re15, input_im15 : integer;
begin
    process
    begin
        input_re0 <= 24;
        input_im0 <= 0;
        input_re1 <= 24;
        input_im1 <= 0;
        input_re2 <= 24;
        input_im2 <= 0;
        input_re3 <= 0;
```

```
    input_im3 <= 0;
    input_re4 <= 24;
    input_im4 <= 0;
    input_re5 <= 24;
    input_im5 <= 0;
    input_re6 <= 24;
    input_im6 <= 0;
    input_re7 <= 24;
    input_im7 <= 0;
    input_re8 <= 24;
    input_im8 <= 0;
    input_re9 <= 0;
    input_im9 <= 0;
    input_re10 <= 24;
    input_im10 <= 0;
    input_re11 <= 24;
    input_im11 <= 0;
    input_re12 <= 24;
    input_im12 <= 0;
    input_re13 <= 0;
    input_im13 <= 0;
    input_re14 <= 24;
    input_im14 <= 0;
    input_re15 <= 0;
    input_im15 <= 0;
    wait;
end process;

dft_module: DFT16 port map (
    A_re0 => input_re0,
    A_im0 => input_im0,
```

```
C_re0 => real0,
C_im0 => imag0,
A_re1 => input_re1,
A_im1 => input_im1,
C_re1 => real1,
C_im1 => imag1,
A_re2 => input_re2,
A_im2 => input_im2,
C_re2 => real2,
C_im2 => imag2,
A_re3 => input_re3,
A_im3 => input_im3,
C_re3 => real3,
C_im3 => imag3,
A_re4 => input_re4,
A_im4 => input_im4,
C_re4 => real4,
C_im4 => imag4,
A_re5 => input_re5,
A_im5 => input_im5,
C_re5 => real5,
C_im5 => imag5,
A_re6 => input_re6,
A_im6 => input_im6,
C_re6 => real6,
C_im6 => imag6,
A_re7 => input_re7,
A_im7 => input_im7,
C_re7 => real7,
C_im7 => imag7,
A_re8 => input_re8,
```

A_im8 => input_im8,

C_re8 => real8,

C_im8 => imag8,

A_re9 => input_re9,

A_im9 => input_im9,

C_re9 => real9,

C_im9 => imag9,

A_re10 => input_re10,

A_im10 => input_im10,

C_re10 => real10,

C_im10 => imag10,

A_re11 => input_re11,

A_im11 => input_im11,

C_re11 => real11,

C_im11 => imag11,

A_re12 => input_re12,

A_im12 => input_im12,

C_re12 => real12,

C_im12 => imag12,

A_re13 => input_re13,

A_im13 => input_im13,

C_re13 => real13,

C_im13 => imag13,

A_re14 => input_re14,

A_im14 => input_im14,

C_re14 => real14,

C_im14 => imag14,

A_re15 => input_re15,

A_im15 => input_im15,

C_re15 => real15,

C_im15 => imag15

```
);

process(clk)
begin
   if rising_edge(clk) then
      led_pwm_temp0 <= abs(real0*real0 + imag0*imag0)/52428;
      led_pwm_temp1 <= abs(real1*real1 + imag1*imag1)/52428;
      led_pwm_temp2 <= abs(real2*real2 + imag2*imag2)/52428;
      led_pwm_temp3 <= abs(real3*real3 + imag3*imag3)/52428;
      led_pwm_temp4 <= abs(real4*real4 + imag4*imag4)/52428;
      led_pwm_temp5 <= abs(real5*real5 + imag5*imag5)/52428;
      led_pwm_temp6 <= abs(real6*real6 + imag6*imag6)/52428;
      led_pwm_temp7 <= abs(real7*real7 + imag7*imag7)/52428;
      led_pwm_temp8 <= abs(real8*real8 + imag8*imag8)/52428;
      led_pwm_temp9 <= abs(real9*real9 + imag9*imag9)/52428;
      led_pwm_temp10 <= abs(real10*real10 + imag10*imag10)/52428;
      led_pwm_temp11 <= abs(real11*real11 + imag11*imag11)/52428;
      led_pwm_temp12 <= abs(real12*real12 + imag12*imag12)/52428;
      led_pwm_temp13 <= abs(real13*real13 + imag13*imag13)/52428;
      led_pwm_temp14 <= abs(real14*real14 + imag14*imag14)/52428;
      led_pwm_temp15 <= abs(real15*real15 + imag15*imag15)/52428;
   end if;
end process;

pwm_module0: PWM port map (
   clk => clk,
   reset => reset,
   duty_cycle => led_pwm_temp0,
   pwm_out => led_pwm0
);
```

```
pwm_module1: PWM port map (
    clk => clk,
    reset => reset,
    duty_cycle => led_pwm_temp1,
    pwm_out => led_pwm1
);

pwm_module2: PWM port map (
    clk => clk,
    reset => reset,
    duty_cycle => led_pwm_temp2,
    pwm_out => led_pwm2
);

pwm_module3: PWM port map (
    clk => clk,
    reset => reset,
    duty_cycle => led_pwm_temp3,
    pwm_out => led_pwm3
);

pwm_module4: PWM port map (
    clk => clk,
    reset => reset,
    duty_cycle => led_pwm_temp4,
    pwm_out => led_pwm4
);

pwm_module5: PWM port map (
    clk => clk,
    reset => reset,
```

```
    duty_cycle => led_pwm_temp5,
    pwm_out => led_pwm5
);

pwm_module6: PWM port map (
    clk => clk,
    reset => reset,
    duty_cycle => led_pwm_temp6,
    pwm_out => led_pwm6
);

pwm_module7: PWM port map (
    clk => clk,
    reset => reset,
    duty_cycle => led_pwm_temp7,
    pwm_out => led_pwm7
);

pwm_module8: PWM port map (
    clk => clk,
    reset => reset,
    duty_cycle => led_pwm_temp8,
    pwm_out => led_pwm8
);

pwm_module9: PWM port map (
    clk => clk,
    reset => reset,
    duty_cycle => led_pwm_temp9,
    pwm_out => led_pwm9
);
```

```
pwm_module10: PWM port map (
    clk => clk,
    reset => reset,
    duty_cycle => led_pwm_temp10,
    pwm_out => led_pwm10
);

pwm_module11: PWM port map (
    clk => clk,
    reset => reset,
    duty_cycle => led_pwm_temp11,
    pwm_out => led_pwm11
);

pwm_module12: PWM port map (
    clk => clk,
    reset => reset,
    duty_cycle => led_pwm_temp12,
    pwm_out => led_pwm12
);

pwm_module13: PWM port map (
    clk => clk,
    reset => reset,
    duty_cycle => led_pwm_temp13,
    pwm_out => led_pwm13
);

pwm_module14: PWM port map (
    clk => clk,
```

```
        reset => reset,
        duty_cycle => led_pwm_temp14,
        pwm_out => led_pwm14
    );

    pwm_module15: PWM port map (
        clk => clk,
        reset => reset,
        duty_cycle => led_pwm_temp15,
        pwm_out => led_pwm15
    );

end behavior;

--==============================

--        _____            _____            _____
--       /\    \          /\    \          /\    \
--      /::\    \        /::\____\        /::\    \
--     /::::\    \      /::::|   |        /::::\    \
--    /::::::\    \    /::::|   |       /::::::\    \
--   /:::/\:::\    \  /::::::|   |      /:::/\:::\    \
--  /:::/__\:::\    \/:::/|::|   |     /:::/  \:::\    \
--  /::::\   \:::\    \::/ |::|   |    /:::/    \:::\    \
-- /::::::\   \:::\    \/  /:::/   |::|   | _____    /:::/    / \:::\    \
-- /:::/\:::\   \:::\    \ /:::/    |::|   |/\    \  /:::/    /   \:::\ ___\
--/:::/__\:::\   \:::\____\/:: /    |::|   /::\____\/:::/____/     \:::|    |
--\:::\   \:::\   \::/    /\::/    /|::|  /:::/    /\:::\    \     /:::|____|
-- \:::\   \:::\   \/____/  \/____/ |::| /:::/    /  \:::\    \   /:::/    /
--  \:::\   \:::\    \              |::|/:::/    /    \:::\    \ /:::/    /
--   \:::\   \:::\____\             |:::::/    /      \:::\    /:::/    /
--    \:::\   \::/    /             |:::::/    /        \:::\  /:::/    /
--     \:::\   \/____/              |::::/    /          \:::\/:::/    /
--      \:::\    \                  /:::/    /            \::::::/    /
--       \:::\____\                /:::/    /              \::::/    /
--        \::/    /                \::/    /                \::/____/
--         \/____/                  \/____/                  ~~
```

Appendix 2: TestBench.vhd

This testbench was designed to test DFT16 module

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity DFT16_tb is
end entity DFT16_tb;

architecture Behavioral of DFT16_tb is
   -- Component Declaration for the Unit Under Test (UUT)
   component DFT16
      port(
   A_re0 : in integer;
   A_im0 : in integer;
   C_re0 : out integer;
   C_im0 : out integer;
   A_re1 : in integer;
   A_im1 : in integer;
   C_re1 : out integer;
   C_im1 : out integer;
   A_re2 : in integer;
   A_im2 : in integer;
   C_re2 : out integer;
   C_im2 : out integer;
   A_re3 : in integer;
   A_im3 : in integer;
   C_re3 : out integer;
   C_im3 : out integer;
   A_re4 : in integer;
```

A_im4 : in integer;

C_re4 : out integer;

C_im4 : out integer;

A_re5 : in integer;

A_im5 : in integer;

C_re5 : out integer;

C_im5 : out integer;

A_re6 : in integer;

A_im6 : in integer;

C_re6 : out integer;

C_im6 : out integer;

A_re7 : in integer;

A_im7 : in integer;

C_re7 : out integer;

C_im7 : out integer;

A_re8 : in integer;

A_im8 : in integer;

C_re8 : out integer;

C_im8 : out integer;

A_re9 : in integer;

A_im9 : in integer;

C_re9 : out integer;

C_im9 : out integer;

A_re10 : in integer;

A_im10 : in integer;

C_re10 : out integer;

C_im10 : out integer;

A_re11 : in integer;

A_im11 : in integer;

C_re11 : out integer;

C_im11 : out integer;

```
    A_re12 : in integer;
    A_im12 : in integer;
    C_re12 : out integer;
    C_im12 : out integer;
    A_re13 : in integer;
    A_im13 : in integer;
    C_re13 : out integer;
    C_im13 : out integer;
    A_re14 : in integer;
    A_im14 : in integer;
    C_re14 : out integer;
    C_im14 : out integer;
    A_re15 : in integer;
    A_im15 : in integer;
    C_re15 : out integer;
    C_im15 : out integer
    );
end component DFT16;
signal A_re0, A_im0, C_re0, C_im0 : integer;
signal A_re1, A_im1, C_re1, C_im1 : integer;
signal A_re2, A_im2, C_re2, C_im2 : integer;
signal A_re3, A_im3, C_re3, C_im3 : integer;
signal A_re4, A_im4, C_re4, C_im4 : integer;
signal A_re5, A_im5, C_re5, C_im5 : integer;
signal A_re6, A_im6, C_re6, C_im6 : integer;
signal A_re7, A_im7, C_re7, C_im7 : integer;
signal A_re8, A_im8, C_re8, C_im8 : integer;
signal A_re9, A_im9, C_re9, C_im9 : integer;
signal A_re10, A_im10, C_re10, C_im10 : integer;
signal A_re11, A_im11, C_re11, C_im11 : integer;
signal A_re12, A_im12, C_re12, C_im12 : integer;
```

```
signal A_re13, A_im13, C_re13, C_im13 : integer;
signal A_re14, A_im14, C_re14, C_im14 : integer;
signal A_re15, A_im15, C_re15, C_im15 : integer;

begin
   -- Instantiate the Unit Under Test (UUT)
   uut: DFT16 port map (
   A_re0 => A_re0,
   A_im0 => A_im0,
   C_re0 => C_re0,
   C_im0 => C_im0,
   A_re1 => A_re1,
   A_im1 => A_im1,
   C_re1 => C_re1,
   C_im1 => C_im1,
   A_re2 => A_re2,
   A_im2 => A_im2,
   C_re2 => C_re2,
   C_im2 => C_im2,
   A_re3 => A_re3,
   A_im3 => A_im3,
   C_re3 => C_re3,
   C_im3 => C_im3,
   A_re4 => A_re4,
   A_im4 => A_im4,
   C_re4 => C_re4,
   C_im4 => C_im4,
   A_re5 => A_re5,
   A_im5 => A_im5,
   C_re5 => C_re5,
   C_im5 => C_im5,
```

A_re6 => A_re6,

A_im6 => A_im6,

C_re6 => C_re6,

C_im6 => C_im6,

A_re7 => A_re7,

A_im7 => A_im7,

C_re7 => C_re7,

C_im7 => C_im7,

A_re8 => A_re8,

A_im8 => A_im8,

C_re8 => C_re8,

C_im8 => C_im8,

A_re9 => A_re9,

A_im9 => A_im9,

C_re9 => C_re9,

C_im9 => C_im9,

A_re10 => A_re10,

A_im10 => A_im10,

C_re10 => C_re10,

C_im10 => C_im10,

A_re11 => A_re11,

A_im11 => A_im11,

C_re11 => C_re11,

C_im11 => C_im11,

A_re12 => A_re12,

A_im12 => A_im12,

C_re12 => C_re12,

C_im12 => C_im12,

A_re13 => A_re13,

A_im13 => A_im13,

C_re13 => C_re13,

```
    C_im13 => C_im13,
    A_re14 => A_re14,
    A_im14 => A_im14,
    C_re14 => C_re14,
    C_im14 => C_im14,
    A_re15 => A_re15,
    A_im15 => A_im15,
    C_re15 => C_re15,
    C_im15 => C_im15
    );

stim_proc: process
begin
    A_re0 <= 1;
    A_im0 <= 0;
    A_re1 <= 0;
    A_im1 <= 0;
    A_re2 <= 1;
    A_im2 <= 0;
    A_re3 <= 0;
    A_im3 <= 0;
    A_re4 <= 1;
    A_im4 <= 0;
    A_re5 <= 0;
    A_im5 <= 0;
    A_re6 <= 1;
    A_im6 <= 0;
    A_re7 <= 0;
    A_im7 <= 0;
    A_re8 <= 1;
    A_im8 <= 0;
```

```
    A_re9 <= 0;
    A_im9 <= 0;
    A_re10 <= 1;
    A_im10 <= 0;
    A_re11 <= 0;
    A_im11 <= 0;
    A_re12 <= 1;
    A_im12 <= 0;
    A_re13 <= 0;
    A_im13 <= 0;
    A_re14 <= 1;
    A_im14 <= 0;
    A_re15 <= 0;
    A_im15 <= 0;

    wait;
end process stim_proc;

end architecture Behavioral;
```

Appendix 3: YouTube Video

Youtube youtube

https://www.youtube.com/watch?v=w6kL3huQ6k4