DOCUMENTATION FOR FFT IMPLEMENTATION

TEAM

Group members:

- Navin Chandak
- Alok Yadav
- Rohan Prinja
- Mandagiri Sai Krishna
- Shirish Kumar Namdeo

FILES

Twiddle.vhd : The entity declaration and the architecture of **Twiddle**.

Butterfly.vhd : The entity declaration and the architecture of **Butterfly**.

layer.vhd : The entity declaration and the architecture of **Twiddle**.

fft_main.vhd : The entity declaration and the architecture of **fft_main**.

Main_fft_entity.vhd : The entity declaration and the architecture of **Main_fft_entity**.

INTERFACE

When the program is burned to the board only one LED light is glowing, indicating that the first input is to be given.

Overall, the user has to give 8 inputs corresponding to the eight points at which we are finding the Fast Fourier Transform. These inputs are taken to be real numbers with at most 3 radix places and whose integral part lies in the range -16 to 15. So, we use 4 bits for the integral part and 3 bits for the fractional part of each number. We also use 1 bit for the sign of the number. In total, 8 bits. The user adjust the 8 DIP switches to set the desired number and then presses the left push button.

This causes the second LED to start glowing and the first one to stop, indicating that we are ready to accept input 2. In this way, the LEDs glow to represent the binary number corresponding to the input which we are ready to accept. Each input is confirmed by pressing the left push button.

After the eight inputs are given, the output is displayed. The output is in fact a 32 bit number, 16 bits each for the real and complex parts, but we only display 16 bits (8 bits each for the real and complex parts) to avoid llediously displaying the fractional parts of the real and complex parts. First we display the real part of the first output, then the complex part of the first output, then the real part of the second output, then its complex part, and so on. There is a delay (intentionally introduced) between the acceptance of the last input and the display of the first output. To see the next output, press the push button and the LEDs will update to show the new output.

INPUT OUTPUT SPECIFICATIONS

Inputs : **16**-bit std_logic_vector representing a complex number **a + ib**

- Integral part of a: **4** bits.
- Decimal part of a: **3** bits.
- Sign bit : **1** bit
- Similarly for b. **5+3+5+3 = 16**

Intermediate values : 32-bit std_logic_vector

Outputs : **32**-bit std_logic_vector representing a complex number **c + id**

- Integral part of a: **8** bits.
- Decimal part of a: **8** bits.
- Similarly for b. **8+8+8+8 = 32**

Variables

- **c** : hardcoded variable storing the value of 1/sqrt(2)

Components

- **Twiddle** : given a and m, computes $a * e^{\frac{-\pi i m}{4}}$
- **Butterfly** : two complex numbers and m$(0 \leq m < 4)$ as input and two complex numbers as output
- **layer** : consists of four butterfly components. The output of the third layer component is our final result

HIGH LEVEL DESCRIPTION

We are calculating the FFT of the input in three stages as per the given diagram. Each stage consists of four butterfly components which take two inputs each and return two outputs according to some combinational logic. In each stage we obtain the output of its four butterfly components and pass them on to the next stage. The output of the last stage is our final output.

BRIEF IMPLEMENTATION DETAILS

Some details regarding the precise implementation of our project.

Debounce : We have implemented a debounce process to avoid taking input multiplte times due to the bouncing of the DIP switches and the push button.

Delay state: The FFT circuit is overall a combinational circuit with some nontrivial circuit delay caused by the many various gates inside it. Hence we cannot directly display the output after the last input is taken since at that point we do not know if the circuit has filled in the output or not. We need to wait for some time. Hence, after the last input is taken the circuit begins and we enter a delay state in which we wait while the combinational process of the circuit is going on. We have waited for roughly 200 clock cycles (any delay greater than or equal to the net delay of the FFT combinational circuit will do).

States: Our circuit consists of a number of states for taking input and displaying output. The actual computational part of our circuit is purely combinational and does not involved any state transition logic at all.