

## Data sheet for FFT Core

### Functional Description

Fast Fourier Transform (FFT) is an efficient algorithm to compute the discrete Fourier Transform (DFT) and its inverse in computational effective way. It reduces the number of computations required to compute N point DFT from  $2 N^2$  to  $2N \log N$ . Therefore the ratio between a DFT computation and an FFT computation for the same N is proportional to  $N/\log(2) N$ . It takes time domain input data and converts into frequency spectral domain data, which would be used for spectral analysis and which contains most of the signal information for signal processing applications.

The FFT has many applications, all digital signal processing applications. In fact any field of that

uses sinusoidal signals will make use of FFT. Some of the areas are in Communications, Astronomy, Geology and Optics.

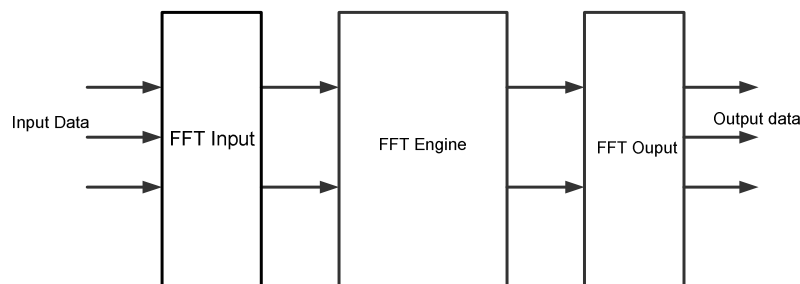
### Features:

- Configurable for N point data.
- Uses Radix-2 FFT for decimation in time.
- Optimal area implementation
- Can handle real and imaginary data.
- Uses Fixed point implementation.

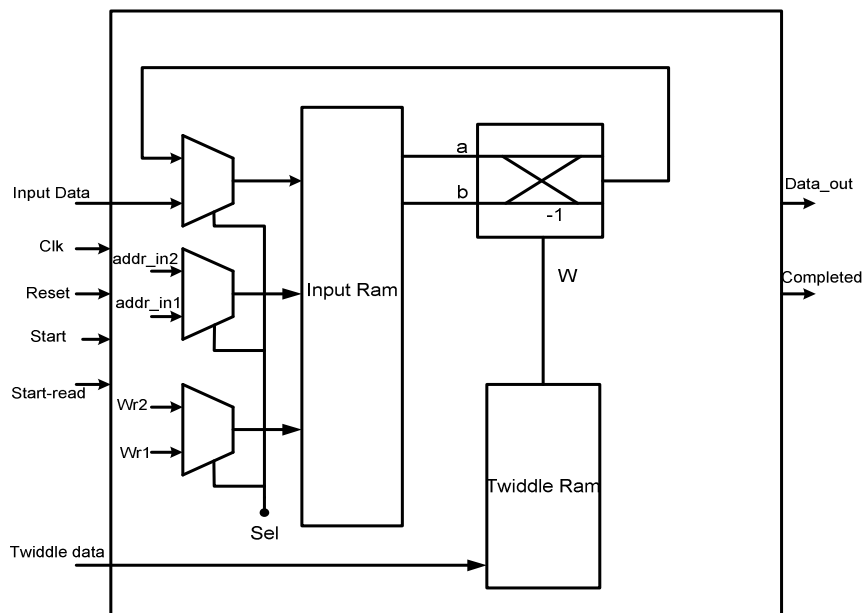
### Deliverables:

- Verilog RTL source code
- Test benches
- Synthesis and Simulation scripts
- Detailed user documentation, including RTL source code documentation

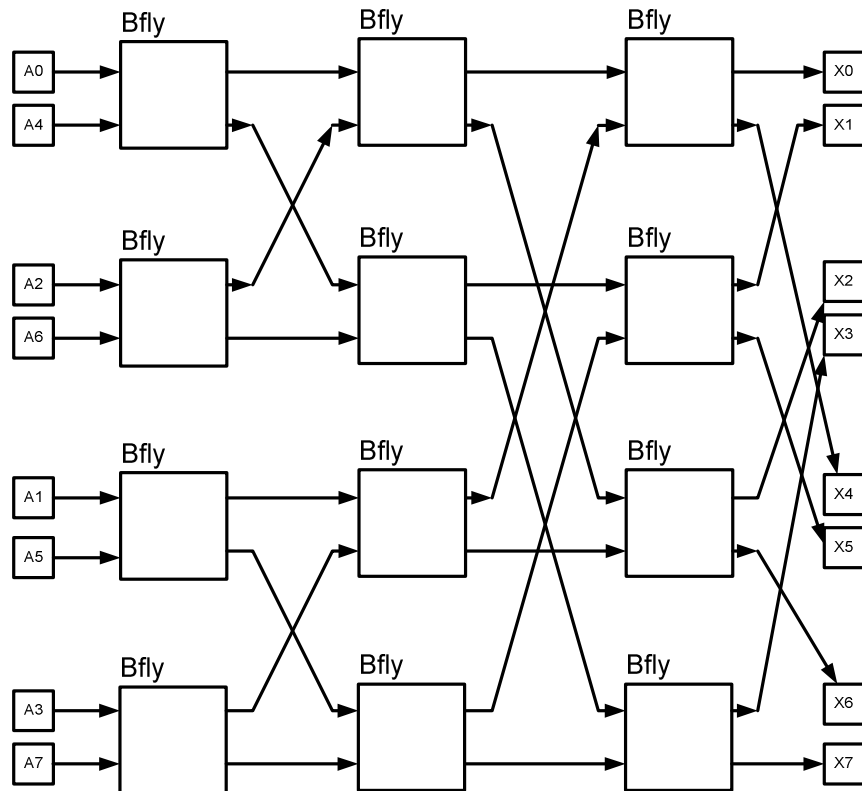
### Block Diagram:



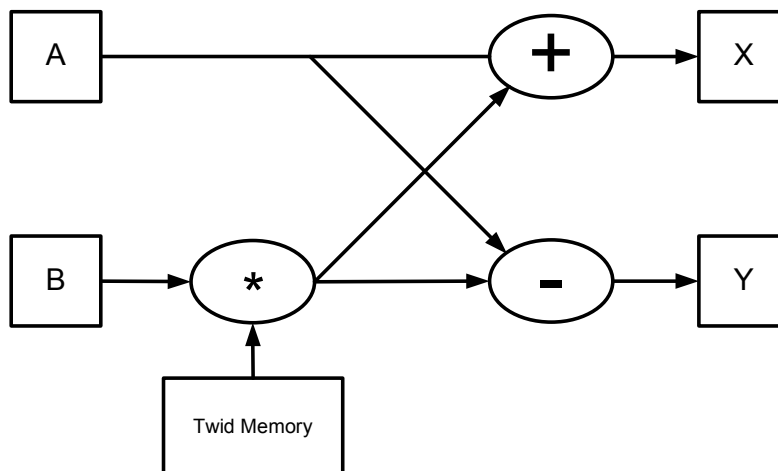
### Architectural Diagram:



## FFT Engine



## Radix-2 Butterfly:



## FFT Parameter Table

This table describes the general FFT parameters:

<i>Parameter</i>	<i>Type</i>	<i>Description</i>
POINT	Integer	Number of points, must be power of 2 for forward FFT.
DATA_WIDTH	Integer	This data width which is fed into the system and taken out after computations. Default value is 32 bit.

## Signals Defined

Signal definition table:

<i>Signal</i>	<i>Direction</i>	<i>Description</i>
<i>datain</i> <input_Re>	IN	Real inputs. <b>datain</b> is the port through which real parts of all inputs will be fed. For N point input, it will take N cycles to feed the entire data.
<i>datain</i> <input_Im>	IN	Through this port through which all imaginary parts of inputs will be fed.
<i>clk</i>	IN	This FFT system is single clock system and all I/Os and internal FFT processor is in synchronous with it.
<i>start</i>	IN	This signal enables the FFT process when it is enabled. This ensures the predictability of the output and synchronism.
<i>reset</i>	IN	This signal resets the system whenever it is enabled and all counters, registers are sets to starting point.
<i>twid_datain</i> <Re>	IN	This is the port through which real parts of all pre-calculated twiddle factors are fed into the twiddle ram. This port is been mapped to the input port of the twiddle ram. These are constant through the procedure and have to be stored once at the starting.
<i>twid_datain</i> <Im>	IN	Through this port all imaginary parts of twiddle factors will be stored in twiddle ram.
<i>data_out</i> <output_re>	OUT	This is the output port through which the real parts of the computed FFT output are taken out and it will take N cycles to read the output for N point FFT system.
<i>data_out</i> <output_Im>	OUT	All imaginary output data will be read from this port.
<i>completed</i>	OUT	This signal indicates the point of completion of FFT process and can be given as indicating signal to output system which would be integrated to FFT to be ready for reading.

## Schematic Symbol

The FFT schematic symbol:

