Dillon Engineering, Inc.

3925 West 50th Street, Suite 202, Edina, MN 55424
Tel.: 952-836-2413 Fax: 952-927-6514
www.dilloneng.com

Ultra High-Performance FFT/IFFT IP Core

1 Features

- ParaCore Architect™ IP Core
- Completely prove IP Core
- Radix-2 length FFT/IFFT
- Variable length selectable at run time
- Dual clocks, I/O and processor clocks
- Flow control on data I/O
- Fixed or floating point complex I/O
- Integrated Channing Window
- Optional Magnitude Output
- Continuous transforms (> 1 GSPS)
- Naturally ordered I/O streams
- Parallel I/O paths (if required)
- Parametric IP Core
- Massively parallel butterfly architecture
- Available in generic HDL or targeted EDIF formats
- Full test bench supplied
- Simple I/O connections
- Clock rates over 200MHz
- Any width data I/O
- Built in prefix generation for OFDM applications

2 General Description

The Dillon Engineering (DE) Fast Fourier Transform (FFT) Core is a ultra high performance forward FFT or inverse FFT (IFFT) processing engine. The FFT Core operates on any radix-2 length set of fixed or floating point data.

The processing speed of the core is adjusted by adding butterfly structures in parallel. For ultra high performance applications, parallel data paths into and out of the module can be added to allow the system to maintain a reasonable clock rate. The speed of operation is only limited by the amount of

available logic in the target device.

3 Theory of Operation

The DE FFT Core is a ParaCore Architect™ IP Core, making configuration option specification at compile time via parameters. A highly efficient Core is created as logic required for modes of operation not needed by this application aren't included in the Core. As an example, if the precision and dynamic range of floating point isn't required, the core created would contain no logic pertaining to floating point, even though the fixed point version is generated from the same source code.

3.1 Block Diagram

The block diagram shows the basic DE FFT structure.

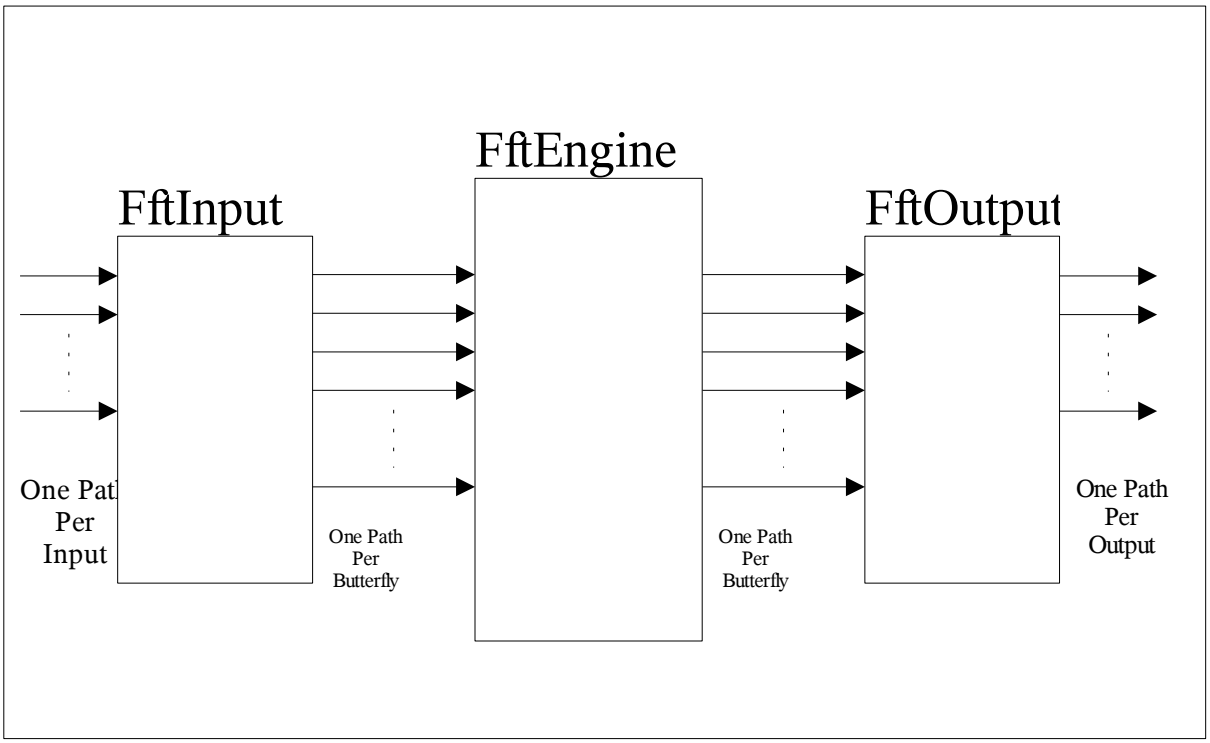


Figure 1- FFT Block Diagram

3.2 FftInput

The FftInput module converts naturally ordered data to the proper format feeding the FftEngine butterflies. It automatically arranges the data to accommodate the size of the FftEngine as the FftEngine has one set of inputs

for each butterfly that is instantiated.

The number of input paths is set via a parameter to par core Architect at compile time. The data rate to the DE FFT Core is set by adjusting the number of inputs to the FftInput module. The more inputs, the faster the throughput.

The data can be streamed in the FftInput module across one full processing time. The number of clock cycles required to input a full set of data is defined by the following formula:

$$inCycles = N / inputs$$

Where N is the number of points in the FFT and outputs is the number of parallel inputs selected by the *inputs* parameter.

3.3 FftEngine

The FftEngine is the main computational component of the DE FFT Core.

It computes the FFT/IFFT using standard Cooley-Tukey radix-2 decimation in time reduction principles.

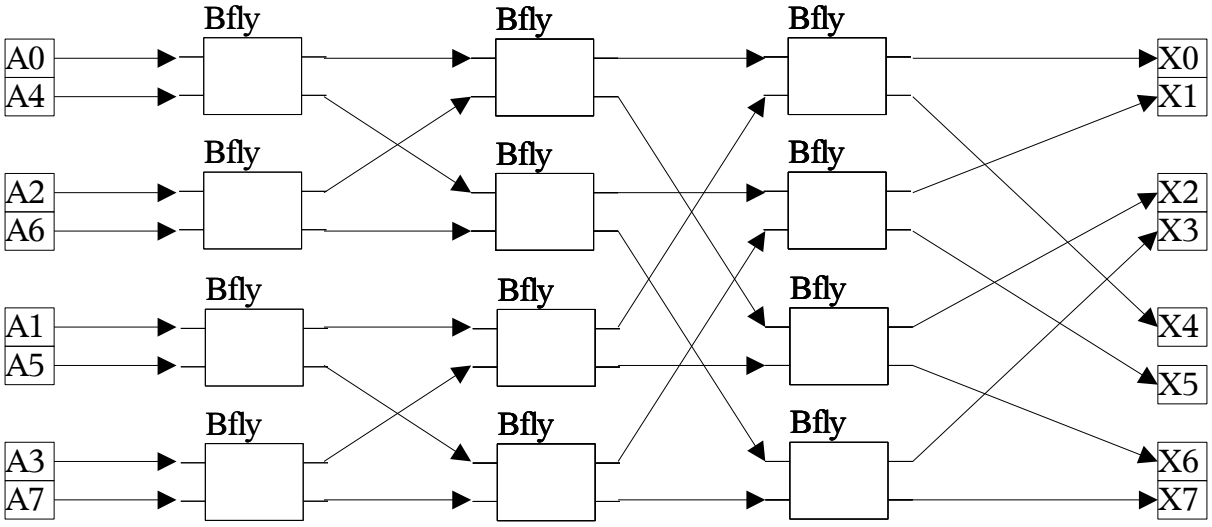


Figure 2: 8 Point DIT FFT

This diagram illustrates the data flow of a 8 point decimation in time FFT. Data is out of order on the input (taken care of by FftInput) and the data is properly ordered on the output.

The basic element is the FFT butterfly (Bfly in above diagram). The butterflies used can be either fixed or floating point with any width of data, all configured via parameters at compile time.

Each column of butterflies is considered a rank. When using fixed point math, the *scale* parameter determines if the data is scaled for each rank. Optional any individual or all ranks can be scaled by 1 bit to avoid overflows. The *scale* parameter is a list of length log2(N), with each element of the list determining if the rank is scaled.

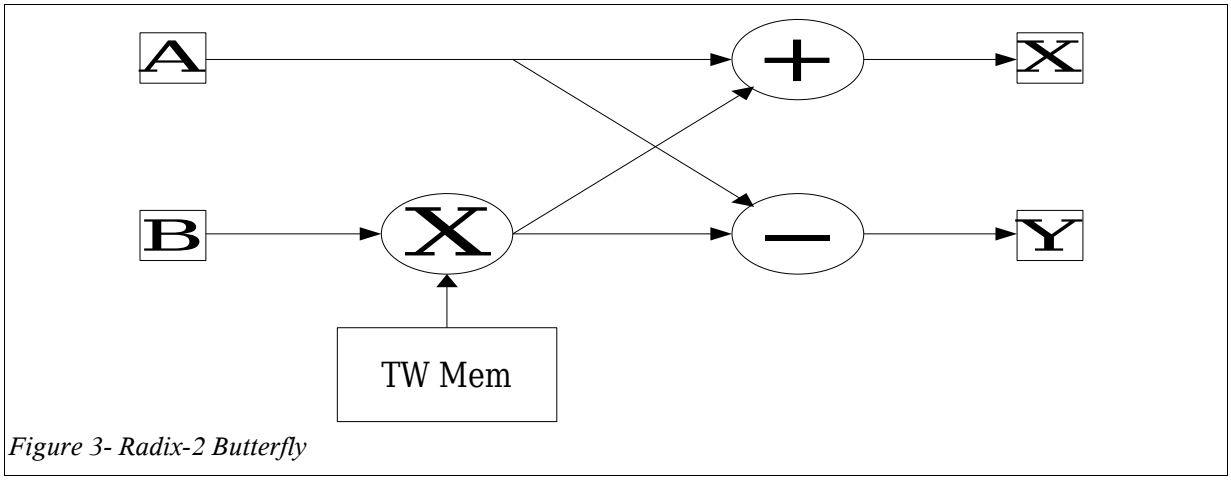


Figure 3- Radix-2 Butterfly

The radix-2 butterfly a complex multiply, complex addition and a complex subtractions as illustrated by the above diagram. The TW Mem is the twiddle factor memory, which is self contained within the DE FFT butterfly.

3.3.1 FftEngine Performance

Any number of butterflies can be instantiated, all in parallel. Each butterfly increases the performance. The following formula can be used to determine how fast the DE FFT Core can process data.

$$Cycles = N * log2(N) / (bfs * 2)$$

Where Cycles are the number of clock cycles to complete the FFT, N is the length of the FFT and bfs is the number of butterflies instantiated. For example a 1024 point FFT with 32 butterflies would be completed in:

With a 125 MHz clock, the transform would be completed in 1.28 uS. Another way to look at the speed of this FFT is 1.28 uS / 1024 is 1.25 nS per sample or 800 MSPS (mega samples per second).

$$Cycles = N * log2(N)/(bfs * 2)$$
$$Cycles = 1024 * log2(1024)/(32 * 2)$$
$$Cycles = 160$$

3.4 FftOutput

The FftOutput module queues the data from the FftEngine and funnels it onto the number of outputs required.

The rate the data leaves is set by the number of outputs at compile time. The following formula defines how many clock cycles it takes to get data out of FftOutput:

$$outCycles = N / outputs$$

Where N is the number of points in the FFT and outputs is the number of parallel outputs selected by the *output* parameter.

Data is available 1 clock cycle after SYNC_OUT is valid for outCycles count of clock cycles.

4 ParaCore Parameters

The DE FFT Core is very flexible since it was designed using ParaCore Architect. All configuration parameters are set at compile time via ParaCore Architect parameters.

4.1 DE FFT Parameter Table

This table describes the general DE FFT parameters.

DE FFT General Parameters		
Parameter	***Type***	***Description***
n	Integer	Number of points, must be power of 2.
forward	select	Forward, inverse, or both.
inputs	Integer	Number of data inputs. Must be a power of 2 and integer divisible into n. Inputs greater than 1 results in a parallel data input path.
outputs	Integer	Number of data outputs. Must be a power of 2 and integer divisible into n. Outputs greater than 1 results in a parallel data input path.
scale	List	Used for fixed point data to selectively scale the data by 1 bit after each rank is executed. Scaling removes errors created by the overflow caused by the complex addition in each butterfly.
type	nType	Selects the data type using nType structure defined below.
misc_width	Integer	Width of misc_in/misc_out ports. If misc_width == 0, no misc_in/misc_out ports exist. If misc_width > 1, then misc_in/misc_out are vectors.
mag_out	Boolean	Produce magnitude output instead of complex results.
hanning	Select	Pick hanning window formula.
butterflies	Integer	Number of butterflies, 0 == calculate and instantiate the required amount to keep up with incoming data.
real_input	Boolean	Input is real only, so use memory reduction in FftInput module.
data_en	Boolean	Build with data_en input, to control data flow into the Fft.
data_ack	Boolean	Build with data_ack input, to control data flow out of Fft.
prefix	Boolean	Build with prefix input that allows pre-pending a prefix on the output data.
sel_width	Integer	If != 0, adds input sel that selects the length of the current FFT, length is N/sel. Allows run time control of the FFT length.
null_opt	Boolean	Build with null_sig input used to generate output without FFT math perform. Used to troubleshoot logic around FFT in large design.

4.2 DE Number Type Definition (nType)

The following parameters pertain to DE number types in the above para-

meters.

DE Number Type Definition (ntype)		
Para-meter	***Type***	***Description***
type	Select	'int' or 'float' selects data type.
width	Integer	For type == 'int', selects the width. For type == 'float', automatically set to e_width + m_width + 1
e_width	Integer	Only used when type == 'float'. Defines the exponent width, for single precision should be set to 8, for double 11.
m_width	Integer	Only used when type == 'float'. Defines the exponent width, for single precision should be set to 23, for double 52.

5 Interface

All interface to the FFT IP module is synchronous CLK input and defined in the following sections.

5.1 Schematic Symbol

The FFT schematic symbol:

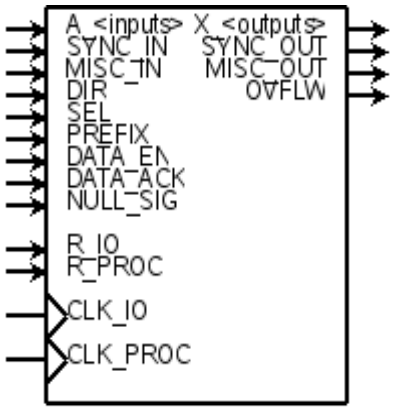


Figure 4: FFT Symbol

A_<inputs> is actually a series of ports, based upon the *inputs* parameter. If *inputs* == 1, then A_0 is the only input port, otherwise A_0, A_1, ... A_N are the input ports, where N == *inputs* -1. Real data is on the most significant half of the port and imaginary data in on the least significant half of the port.

X_<inputs> is actually a series of ports, based upon the *outputs* parameter. If outputs == 1, then X_0 is the only input port, otherwise X_0, X_1, ... X_N are the input ports, where N == outputs -1. Real data is on the most significant half of the port and imaginary data in on the least significant half of the port.

5.2 Signals Defined

Signal definition table:

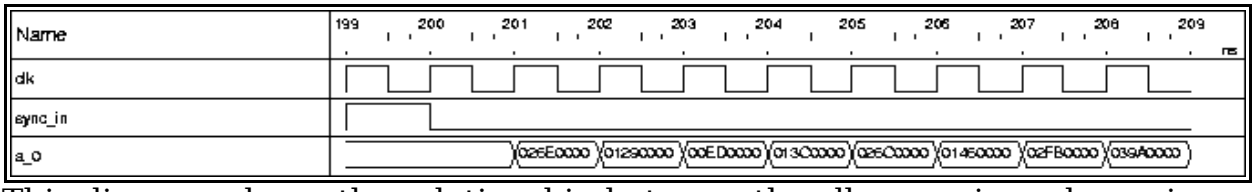
Signal	***Direc-tion***	***Description***
A_<inputs>	IN	Real and complex inputs. Repeated as A_0, A_1 and so on for each input set by parameter *inputs*. Real data in on A_0[width-1:width/2] and imaginary data is on A_0[width/2-1:0].
SYNC_IN	IN	Synchronize incoming data to module. Active for one clock to indicate new set of data to process.
R_IO	IN	Reset input, synchronize to CLK_IO.
R_PROC	IN	Reset input, synchronize to CLK_PROC. Should be asserted after power up and prior to the first SYNC_IN.
CLK_IO	IN	I/O clock, all I/O from FFT IP is synchronous to CLK_IO.
CLK_PROC	IN	Internal FFT processor clock. If a single clock system, then tie to same clock as CLK_IO.
MISC_IN	IN	Available if misc_width parameter is greater than 0. Sampled on the SYNC_IN cycle to tag the current data set. If misc_width is greater than 1, than this port is a vector of that length.
DIR	IN	Available if forward parameter is both, select FFT or IFFT mode.
SEL	IN	Available if sub_length parameter != 0, selects the sub length of the FFT length that is computed for the current set.
PREFIX	IN	Available if prefix parameter == true, select prefix starting point for the current set.
DATA_EN	IN	Available if data_en parameter == true, used to control data flow into the FFT block.
DATA_ACK	IN	Available if data_ack parameter == true, used to control data flow out of the FFT block.
NULL_SIG	IN	Available if null_opt == true. NULL_SIG == 1, not math is performed by FFT block, input data flows through logic and is passed to output with timing identical to FFT operation. Used for testing purposes.
X_<outputs>	OUT	Real and complex outputs. Repeated as X_0, X_1 and so on for each output set by parameter *outputs*. Real data in on X_0[width-1:width/2] and imaginary data is on X_0[width/2-1:0].
SYNC_OUT	OUT	Indicates result is ready and will begin streaming out

Signal	***Direc-tion***	***Description***
		on the next clock cycle.
OVFLW	OUT	Overflow indication, if OVFLW == 1, then date for the current output frame is invalid. OVFLW can occur anywhere during the data frame output.
MISC_OUT	OUT	Available if misc_width parameter is greater than 0. Sampled with SYNC_OUT, show frame set tag associated with the output data set. If misc_width is greater than 1, than this port is a vector of that length.

5.3 Signal Diagrams

The following diagram show the clock timing relationships between the I/O signals.

5.3.1 Input Signals



This diagram shows the relationship between the clk, sync_in and a_x signals.

In the diagram above, after the rising edge of the clock samples sync_in active, the next rising edge on clk samples the first input point. The following N-1/paths rising edges of the clk samples the rest of the input data.

If there are multiple paths, then the input data is split between each a_x input.

Data is input in normal order.

5.3.2 Output Signals

This diagram shows the relationship between the clk, sync_out and x_x signals.

The output data follows the sync_out signal. Each N/paths cycle after the cycle containing sync_out active will contain the result in normal order.

6 Device Fit

This table provides a small sample of FFT configuration and resulting device logic usage. The bit counts used in this section refer to the real or imaginary component, so the total width is twice this count.

The rate refers to the mega samples per second (MSPS) that the FFT can input/output. The DE FFT operates on a continuous data stream based upon the number of I/O ports.

Faster and slower versions are also available, this is meant to be a sample of the FFT speeds and sizes.

6.1 Virtex II - Real Inputs - 18 Bits

All data is 18 bits with real inputs and complex outputs. All calculations are 18 bit fixed point.

Xilinx Virtex II 125MHz – Real Inputs – 18 bit data					
Length	***I/O Ports***	***Slices***	***Mul-t18x18***	***Block RAMs***	***Rate (MSPS)***
256	1	4,090	32	16	125
256	2	10,724	64	32	250
256	4	16,546	128	64	500
1024	1	4,024	32	48	125
1024	2	10,430	64	64	250
1024	4	20,826	128	64	500
2048	1	5,050	32	48	125
2048	2	8,376	64	96	250
2048	4	14,989	128	128	500
8192	1	4,043	32	80	125
8192	2	8,215	64	128	250

6.2 Virtex II - Complex Inputs - 18 Bits

All data is 18 bits with complex inputs and outputs. All calculations are 18 bits fixed point.

Xilinx Virtex II 125MHz – Complex Inputs – 18 bit data					
Length	***I/O Ports***	***Slices***	***Mul-t18x18***	***Block RAMs***	***Rate (MSPS)***
256	1	4,364	32	16	125
256	2	10,724	64	32	250
256	4	23,707	128	64	500
1024	1	4,408	32	48	125
1024	2	10,430	64	64	250
1024	4	25,970	128	64	500
2048	1	4,482	32	48	125
2048	2	4,482	32	48	250
2048	4	9,410	64	128	500

6.3 Virtex II - Real Inputs - 24 Bits

All data is 24 bits with real inputs and outputs. All calculations are 24 bits fixed point.

Xilinx Virtex II 125MHz – Real Inputs – 24 bit data					
Length	***I/O Ports***	***Slices***	***Mul-t18x18***	***Block RAMs***	***Rate (MSPS)***
128	1	7825	128	24	125
256	1	9297	128	24	125
1024	1	8424	128	64	125
2048	1	8565	128	64	125
8192	1	8417	128	128	125

6.4 Virtex II - Complex Inputs - 18 Bits Single Butterfly

All data is 18 bits with complex inputs and outputs. All calculations are 18 bits fixed point.

This is the slowest available version of the FFT IP Core, it uses a single but-

terfly.

Xilinx Virtex II 125MHz – Real Inputs – 18 bit data – Single Butterfly					
Length	***I/O Ports***	***Slices***	***Mul-t18x18***	***Block RAMs***	***Rate (MSPS)***
1024	1	658	4	10	24
2048	1	694	4	20	22
8192	1	673	4	82	19

