



## 16-Point Complex FFT/IFFT V1.0.3

December 17, 1999

Product Specification



Xilinx Inc.  
2100 Logic Drive  
San Jose, CA 95124  
Phone: +1 408-559-7778  
Fax: +1 408-559-7114  
URL: [www.xilinx.com/support/techsup/appinfo](http://www.xilinx.com/support/techsup/appinfo)  
[www.xilinx.com/ipcenter](http://www.xilinx.com/ipcenter)

### Features

- High-performance 16-point complex Fast Fourier Transform (FFT) and Inverse FFT (IFFT)
- 16-bit complex input and output data
- Two's complement arithmetic
- Scaling control
- Parallel architecture provides a new output sample on every clock edge
- Input data can be continuously streamed into the core
- Naturally ordered input and output data
- High performance and density guaranteed through Relational Placed Macro (RPM) mapping and placement technology
- Incorporates Xilinx Smart-IP technology for maximum performance
- To be used with version 2.1i or later of the Xilinx CORE Generator System

### General Description

The vFFT16 core uses a radix-4 Cooley-Tukey [1] algorithm to compute the Discrete Fourier Transform (DFT), or inverse DFT, of a complex data vector. The input data is a vector of 16 complex values represented as 16-bit 2's complement numbers: 16-bits for each of the real and imaginary component of an input data sample.

### Functional Description

This core accepts naturally ordered data on the input buses DI\_R and DI\_I and performs a complex FFT or IFFT. These buses are respectively the real and imaginary components of the input sequence. Data must be supplied in a continuous stream to the Core. An internal input data memory controller orders the data into blocks to be presented to the FFT processor. Each block is 16 samples in length. When a new block is assembled it is immediately passed on to the FFT engine. The calculation of a complete FFT requires 16 clock cycles. This means that transforms are performed in a block-continuous fashion. The low processing latency allows data to be continuously streamed into the core. The START signal is used to initiate the first transform. Once START has been asserted to begin the first transform there is no need to assert it again; the core will perform transforms continuously once it has been started. There is an initial 82 clock cycle latency from the time START is asserted to the availability of the first valid output sample. This is illustrated in Figure 2.

Transforms can be computed back-to-back. The MODE\_CE pin indicates when the operating mode signals FWD\_INV and SCALE\_MODE are sampled by the core. This signal is useful when alternating forward and inverse FFTs are performed.

The user can elect to assert START at any time to re-synchronize the processor to the input data. However, this causes the internal pipeline to be flushed. The result is that each time START is applied, the 82 clock cycle start-up latency is experienced.

Just as data is continuously streamed into the core, DFT samples are also continuously streamed out of the core on the XK\_R and XK\_I buses. These buses respectively provide the real and imaginary components of the complex output samples. The DONE signal identifies the start of a transform output vector. Figure 2 shows the timing relationship between DONE and the XK\_R and XK\_I buses. The DFT samples appear in natural order on the output buses starting with XK\_[R/I](0) as indicated in the figure.

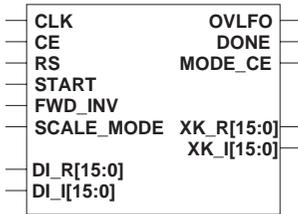


Figure 1: 16-Point FFT Schematic Symbol

## Theory of Operation

The discrete Fourier transform (DFT)  $X(k)$ ,  $k=0, \dots, N-1$  of a sequence  $x(n)$ ,  $n=0, \dots, N-1$  is defined as

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi nk/N} \quad k = 0, \dots, N-1 \quad (1)$$

where  $N$  is the transform size and  $j = \sqrt{-1}$ . The fast Fourier transform (FFT) is a computationally efficient algorithm for computing a DFT.

The Xilinx 16-point transform engine employs a Cooley-Tukey radix-4 decimation-in-frequency (DIF) FFT[1] to compute the DFT of a complex sequence. In general, this algorithm requires the calculation of columns, or ranks, of radix-4 butterflies. These radix-4 butterflies are sometimes referred to as *dragonflies*. Each processing rank consists of  $N/4$  dragonflies. For  $N=16$  there are two dragonfly ranks, with each rank comprising four dragonflies.

The FFT processor input-data for the Core is a vector of 16 complex samples. The real and imaginary components of each sample are represented as 16-bit 2's complement numbers. The data input and output buffers are stored internally within the FPGA. The phase factors used in the FFT calculation are generated within the Core. Like the input-data, the phase factors are kept to a precision of 16 bits for each of the real and imaginary components.

## Finite Word Length Considerations

The radix-4 FFT algorithm processes an array of data by successive passes over the array. On each pass, the algorithm performs dragonflies, each dragonfly picking up four complex numbers and returning four complex numbers to the same addresses but in a different memory bank. The numbers returned to memory by the processor are larger than the numbers picked up from memory. A strategy must be employed to accommodate this dynamic range expansion. A full explanation of scaling strategies and their implications is beyond the scope of this document, the reader is

referred to several documents available in the open literature [2] [3] that discuss this topic.

The Xilinx 16-point FFT Core scales dragonfly results by a factor of 4 (or 2 bits) on each processing pass. The SCALE\_MODE pin can be used to force an additional scaling by one bit on the first processing pass only. The scaling results in the final output sequence being modified by the factor 1/16 when SCALE\_MODE=0 and 1/32 when SCALE\_MODE=1. Formally, the output sequence computed by the Core (when FWD\_INV=1) is defined by: The scaling results in the final output sequence being modified by the factor  $1/sN$  where  $N=256$  for the vfft256 core. Formally, the output sequence  $X'(k)$ ,  $k=0, 1, \dots, N-1$  computed by the core (when FWD\_INV=1) is defined in the equation below:

$$X'(k) = \frac{1}{sN} X(k) = \frac{1}{sN} \sum_{n=0}^{N-1} x(n)e^{-j2\pi nk/N} \quad k = 0, \dots, N-1 \quad (3)$$

where  $s=1$  when SCALE\_MODE=0 and  $s=2$  when SCALE\_MODE=1. The SCALE\_MODE pin can be used for both the forward and inverse FFT modes of operation.

The vfft256 core also computes the IFFT according to the following defining equation:

$$x(n) = \frac{1}{sN} \sum_{k=0}^{N-1} X(k)e^{j2\pi nk/N} \quad n = 0, \dots, N-1 \quad (2)$$

The built-in scaling in the core accounts for the  $1/N$  scale factor in front of the summation in Eq. (2). When SCALE\_MODE=1, an additional scaling by a factor of 1/2 will be scheduled in the core. The additional scaling by 1 bit is inserted during the memory write operation during the first of the 2 processing phases.

## Pinout

Signal names for the schematic symbol are shown in Figure 1 and described in Table 1.

## System Level Modeling Support

In addition to a VHDL behavioral model, a Matlab [4] compatible Dynamic Link Library (DLL) is available from Xilinx. The vfft DLL is a bit-true model of the vfft16 core that can be used in the Matlab environment for system level design and development.

## Behavioral Simulation

Release Version 1.0 of the vfft16 core has VHDL behavioral model, but does not include a verilog behavioral model.

## Implementation

The vfft16 core is supplied as a group of edif netlists. The top level netlist is called vfft16.edn. All of the netlists that are delivered with the core must be present in the user's project directory.

## Performance

The complete calculation of one 16-point FFT requires 16 clock cycles. The transform execution time is

$$T_{\text{FFT}} = \frac{16}{f_{\text{CLK}}} \text{ where } f_{\text{CLK}} \text{ is the system clock frequency.}$$

For example, for a system clock frequency of 100 MHz, the execution time is 160 ns. When the clock frequency is increased to 120 MHz the transform time is 133.33 ns.

## Clock Enable

There are several issues involving the clock-enable *CE* pin that designers should be familiar with when developing systems with this core. *CE* is a high fan-out signal and should be presented to the Core via a low-skew clock buffer to achieve maximum operating frequency. Refer to the Xilinx device data book for more information on these features. The *CE* pin is a master clock enable for the entire Core. When in the inactive state ( $CE=0$ ), all core operations are stalled until *CE* is re-asserted ( $CE=1$ ).

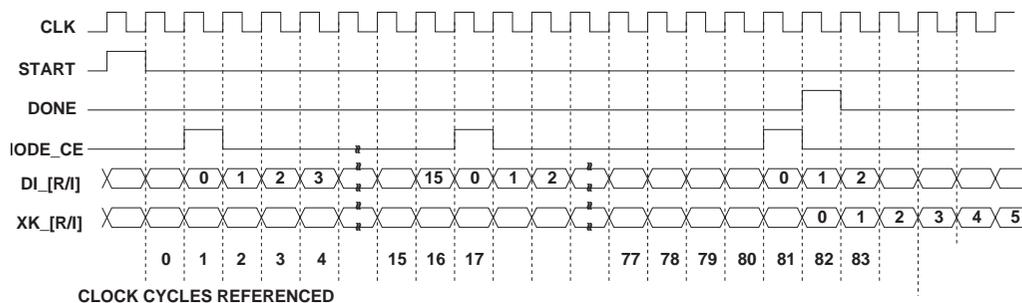


Figure 2: 16-Point FFT Core Timing

**Table 1: Core Signal Pinout**

Signal	Signal Direction	Description
CLK	Input	Clock input: Active rising edge
CE	Input	Clock enable: Active High
RS	Input	Master Reset: Active High
START	Input	FFT start control: Active High
FWD_INV	Input	Defines if a forward (FWD_INV=1) or inverse (FWD_INV=0) is performed
SCALE_MODE	Input	FFT scaling control. When SCALE_MODE=0 the FFT output vector is scaled by 1/16. When SCALE_MODE=1 the FFT output vector is scaled by 1/32.
DI_R[15:0]	Input	Input databus: real component
DI_I[15:0]	Input	Input databus: imaginary component
OVFLO	Output	Active High arithmetic overflow indicator. Even when employing a 2-bit scale factor for each FFT processing phase, certain input signals can cause arithmetic overflow. When additional scaling is employed by setting SCALE_MODE=1, there is no possibility of overflow occurring and this signal will not be active. OVFLO is removed when the core is reset by asserting RS, when START is asserted, or at the beginning of the next output result vector as indicated by DONE.
DONE	Output	Transform complete strobe Active High. This signal is present for one clock cycle at the beginning of a result vector (Active High)
MODE_CE	Output	Indicates when the FWD_INV and SCALE_MODE pins are sampled Active High
XK_R[15:0]	Output	DFT result: real component
XK_I[15:0]	Output	DFT result: imaginary component

## Core Resource Utilization

The 16-point FFT Core occupies 1386 logic slices. The geometry of the RPM requires it to be placed in a XCV300 or larger device.

## Ordering Information

This core is downloadable free of charge from the Xilinx IP Center ([www.xilinx.com/ipcenter](http://www.xilinx.com/ipcenter)), for use with version 2.1i or later of the Xilinx CORE Generator System. The CORE Generator System 2.1i tool is bundled with the Alliance 2.1i and Foundation 2.1i implementation tools.

To order Xilinx software contact your local Xilinx sales representative at [www.xilinx.com/company/sales.htm](http://www.xilinx.com/company/sales.htm).

## References

- [1] J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series", *Math. Compute.*, Vol. 10, pp. 297-301, April 1965.
- [2] W. R. Knight and R. Kaiser, "A Simple Fixed-Point Error Bound for the Fast Fourier Transform", *IEEE Trans. Acoustics, Speech and Signal Proc.*, Vol. 27, No. 6, pp. 615-620, Dec. 1979.
- [3] L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*, Prentice-Hall Inc., Englewood Cliffs, NJ, 1975.
- [4] The Mathworks Inc., *Matlab User's Guide*, Boston, MA, USA, 1999.
- [5] ModelSim, Model Technology Inc.