Fast Fourier Transform MegaCore Function

Solution Brief 12 **January 1997, ver. 1**

Target Application:
Digital Signal Processing
Wireless Communications

Family:
FLEX 10K

Vendor:

2610 Orchard Parkway
San Jose, CA 95134-2020
Tel. (408) 894-7000
Fax (408) 944-0952
WWW http://www.altera.com

Features

- Optimized for the Altera® FLEX® 10K device architecture
- Complex data in and data out decimation-in-frequency (DIF) fast Fourier transform (FFT)
- Dual memory architecture
- Flexible memory interface—data and twiddle memory elements can be implemented in internal and/or external RAM
- Significantly faster than DSP processor solutions
- Parameterized data width, twiddle width, and number of points
- Block-floating point notation to provide optimal accuracy

General Description

The Altera `fft` MegaCore function implements a DIF algorithm, and contains all the logic functions necessary to implement an FFT algorithm. The memory configuration and I/O interface can be configured by the designer for optimum flexibility.

To optimize throughput, the `fft` MegaCore function uses a dual memory architecture that consists of the right and left memory banks. The dual memory architecture allows data to be read from one memory bank and written to another memory bank.

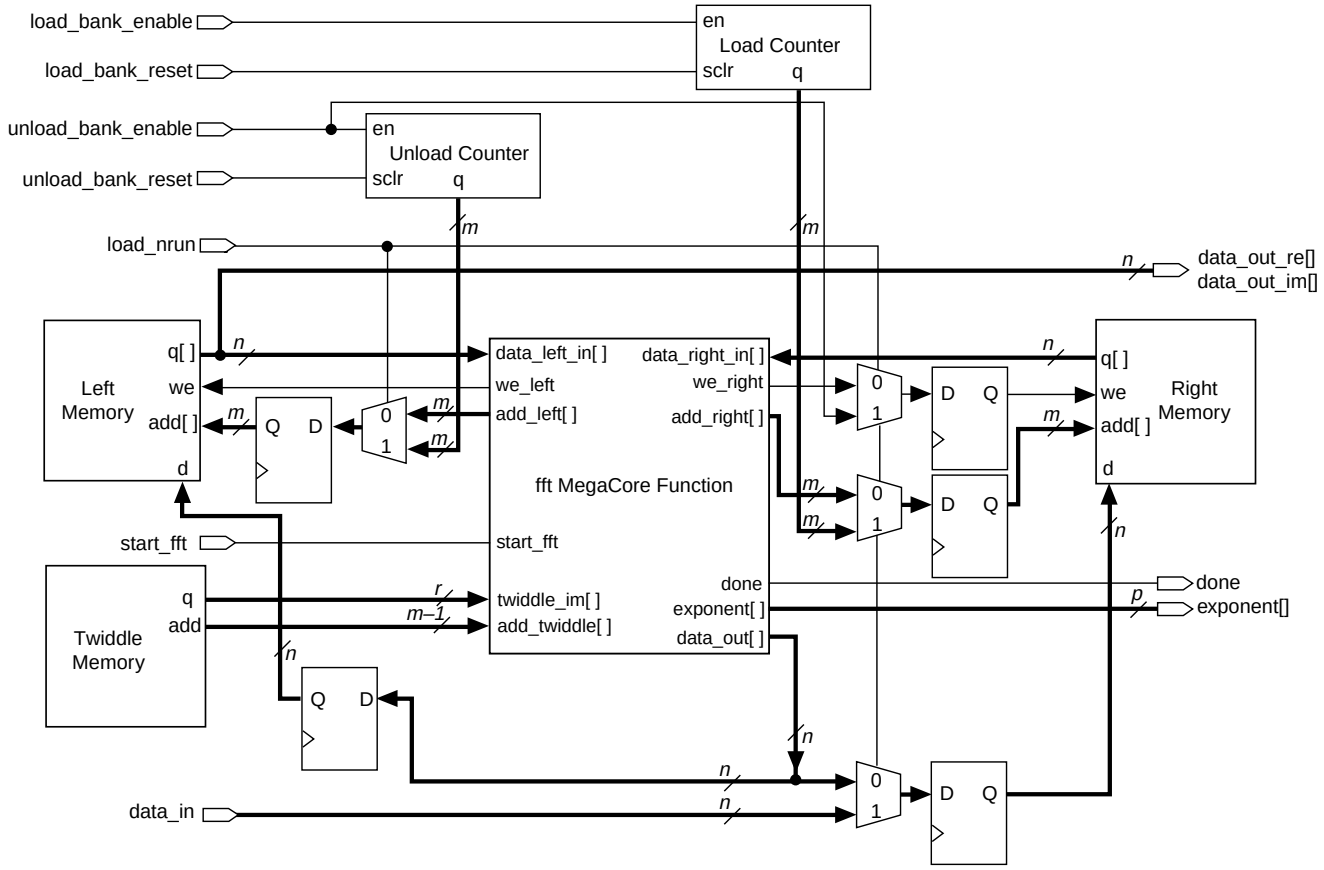
The `fft` MegaCore function also uses a third memory, known as the twiddle memory bank, that is kept separate from the right and left memory banks to maximize throughput.

Functional Description

Figure 1 shows a block diagram of an example system implementation that uses the `fft` MegaCore function and additional logic in a FLEX 10K device. The designer can choose from a variety of memory and I/O interface options. The implementation shown in Figure 1 has on-chip RAM, an odd number of passes (i.e., an odd number of address bits), and no data buffering. The right, left, and twiddle memory elements are all implemented in FLEX 10K embedded array blocks (EABs). After new data is loaded into the right memory bank, the `fft` function can sequentially process data. The `fft` function cannot process data while new data is being loaded into the right memory bank. When the `fft` function is not processing data, new data can be loaded and unloaded simultaneously from both the right and left memory banks.

For more detailed information on this `fft` function implementation, as well as other implementation schemes, refer to the *fft Fast Fourier Transform Data Sheet*.

Figure 1. Block Diagram of a System Implementation with an fft MegaCore Function



where: $n = 2 \times$ WIDTH_DATA
$m =$ WIDTH_ADD
$p =$ WIDTH_EXPONENT
$r =$ WIDTH_TWIDDLE

Ports

Table 1 describes the ports of the fft MegaCore function.

Table 1. fft MegaCore Function Ports

Name	Type	Description
clock	Input	Clock signal.
start_fft	Input	Starts the computation after data is loaded.
data_left_in_re[]	Input	Real data input from the left memory bank.
data_left_in_im[]	Input	Imaginary data input from the left memory bank.
data_right_in_re[]	Input	Real data input from the right memory bank.
data_right_in_im[]	Input	Imaginary data input from the right memory bank.
twiddle_re[]	Input	Real data input from the twiddle memory bank.
twiddle_im[]	Input	Imaginary data input from the twiddle memory bank.
done	Output	Goes high when the function has completed the calculation.
data_direction	Output	When high, data is read from the left memory bank and written to the right memory bank. When low, the inverse is performed.
we_left	Output	Write enable for the left memory bank.
we_right	Output	Write enable for the right memory bank.
add_left[]	Output	Address bus for the left memory bank.
add_right[]	Output	Address bus for the right memory bank.
add_twiddle[]	Output	Address bus for the twiddle memory bank.
data_out_re[]	Output	Real data output to both the left and right memory banks.
data_out_im[]	Output	Imaginary data output to both the left and right memory banks.
exponent[]	Output	Exponent of the resultant data. Valid after done goes high. Exponent of the block floating notation. All data should be scaled by $2^{exponent[\ ]}$.

Performance

The fft MegaCore function operates at a maximum speed of 41 MHz. Table 2 provides parameter ranges and performance details on an 1024-point fft MegaCore function.

Table 2. Typical Utilization Results of the Fast Fourier Transform MegaCore Function

Parameter	Typical Values	Sample Implementation
fft function length	Integer power of 2	1024 points
WIDTH_DATA	Integer	16 bits
WIDTH_TWIDDLE	Integer	16 bits
WIDTH_ADD	Integer	10 bits
WIDTH_EXPONENT	Integer	6 bits
Memory latency	Integer	2
Logic elements (LEs) used	–	2993
Performance (in EPF10K100-3)	–	250 microseconds

Implementing the fft MegaCore function

To help designers implement the `fft` MegaCore function into their projects, Altera provides:

- OpenCore system which allows the designer to preview the `fft` MegaCore function before purchase. The designer can also instantiate and simulate the `fft` MegaCore function in MAX+PLUS® II.
- **twiddle** utility program—distributed with MAX+PLUS II—allows designers to generate a Memory Initialization File (**.mif**) that contains all the twiddle data. This MIF can be translated to other formats for use in off-chip twiddle memory.
- `fft_on_chip` reference design which is a complete sample design for an FFT system that uses on-chip twiddle and data RAM in FLEX 10K devices. It is available at no additional charge.

For additional information on the `fft` MegaCore function, refer to the *fft Fast Fourier Transform Data Sheet*.

2610 Orchard Parkway
San Jose, CA 95134-2020
(408) 894-7000
http://www.altera.com