Pipelined FFT/IFFT 64 points (Fast Fourier Transform) IP Core User Manual

Unicore Systems Ltd
60-A Saksaganskogo St
Office 1
Kiev 01033
Ukraine
Phone: +38-044-289-87-44
Fax: : +38-044-289-87-44
E-mail: o.uzenkov@unicore.co.ua
URL: www.unicore.co.ua

Overview

The USFFT64 User Manual contains description of the USFFT64 core architecture to explain its proper use. USFFT64 soft core is the unit to perform the Fast Fourier Transform (FFT). It performs one dimensional 64 – complex point FFT. The data and coefficient widths are adjustable in the range 8 to 16.

Features

- 64 -point radix-8 FFT.
- Forward and inverse FFT.
- Pipelined mode operation, each result is outputted in one clock cycle, the latent delay from input to output is equal to 163 clock cycles, simultaneous loading/downloading supported.
- Input data, output data, and coefficient widths are parametrizable in range 8 to 16.
- Two and three data buffers are selected.
- FFT for 10 bit data and coefficient width is calculated on Xilinx XC4SX25-12 FPGA at 250 MHz clock cycle, and on Xilinx XC5SX25-12 FPGA at 300 MHz clock cycle, respectively.
- FFT unit for 10 bit data and coefficients, and 2 data buffers occupies 1513 CLB slices, 4 DSP48 blocks, and 2,5 kbit of RAM in Xilinx XC4SX25 FPGA, and 700 CLB slices 4 DSP48E blocks, and 2,5 kbit of RAM in Xilinx XC5SX25 FPGA, data buffers are implemented on the distributed RAM.
- Overflow detectors of intermediate and resulting data are present.
- Two normalizing shifter stages provide the optimum data magnitude bandwidth.
- Structure can be configured in Xilinx, Altera, Actel, Lattice FPGA devices, and ASIC.
- Can be used in OFDM modems, software defined radio, multichannel coding.

Design Features

FFT is an algorithm for the effective Discrete Fourier Transform calculation

Discrete Fourier Transform (DFT) is a fundamental digital signal processing algorithm used in many applications, including frequency analysis and frequency domain processing. DFT is the decomposition of a sampled signal in terms of sinusoidal (complex exponential)

components. The symmetry and periodicity properties of the DFT are exploited to significantly lower its computational requirements. The resulting algorithms are known as Fast Fourier Transforms (FFTs). An 64-point DFT computes a sequence *x(n)* of 64 complex-valued numbers given another sequence of data *X(k)* of length 64 according to the formula

$$X(k) = \sum_{n=0}^{63} x(n) e^{-j2\pi nk/64} \quad ; \quad k = 0 \text{ to } 63.$$

To simplify the notation, the complex-valued phase factor $e^{-j2\pi nk/64}$ is usually defined as W_{64}^{n} where: $W_{64} = \cos(2\pi/64) - j \sin(2\pi/64)$. The FFT algorithms take advantage of the symmetry and periodicity properties of W_{64}^{n} to greatly reduce the number of calculations that the DFT requires. In an FFT implementation the real and imaginary components of W^{n}_{N} are called twiddle factors.

The basis of the FFT is that a DFT can be divided into smaller DFTs. In the processor USFFT64 a radix-8 FFT algorithm is used. It divides DFT into two smaller DFTs of the length 8, as it is shown in the formula:

$$X(k) = X(8r+s) = \sum_{m=0}^{7} W_8^{mr} W_{64}^{ms} \sum_{l=0}^{7} x(8l+m) W_8^{sl}, \; r = 0 \text{ to } 7, \; s = 0 \text{ to } 7,$$

which shows that 64-point DFT is divided into two smaller 8-point DFTs. The input complex data $x(n)$ are represented by the 2-dimensional array of data $x(8l+m)$. The columns of this array are computed by 8-point DFTs. The results of them are multiplied by the twiddle factors W_{64}^{ms}. And the resulting array of data $X(8r+s)$ is derived by 8-point DFTs of rows of the intermediate result array.

The 8-point DFT, named as the base FFT operation, is implemented by the Winograd FFT algorithm, which provides the minimum additions and multiplications (only 2 complex multiplications to the factor W_8^{1}). As a result, the radix-8 FFT algorithm needs only 64 complex multiplications to the twiddle factors W_{64}^{ms} and 32 multiplications to the twiddle factor W_8^{1} except of 4096 complex multiplications in the origin DFT. Note that the well known radix-2 64-point FFT algorithm needs 192 complex multiplications.

Highly pipelined calculations

Each base FFT operation is computed by the computational unit, called FFT8, which is the data path for FFT calculations. FFT8 calculates the 8-point DFT in the high pipelined mode. Therefore in each clock cycle one complex number is read from the input data buffer RAM and the complex result is written in the output buffer RAM. The 8-point DFT algorithm is divided into several stages which are implemented in the stages of the FFT8 pipeline. This supports the increasing the clock frequency up to 250 MHz and higher. The latent delay of the FFT8 unit from input the first data to output the first result is equal to 14 clock cycles.

High precision computations

In the core the inner data bit width is higher to 4 digits than the input data bit width. The main error source is the result truncation after multiplication to the factors W_{64}^{ms}. Because the most of base FFT operation calculations are additions, they are calculated without errors. The FFT

results have the data bit width which is higher in 3 digits than the input data bit width, which provides the high data range of results when the input data is the sinusoidal signal. The maximum result error is less than the 1 least significant bit of the input data.
Besides, the normalizing shifters are attached to the outputs of FFT8 pipelines, which provide the proper bandwidth of the resulting data. The overflow detector outputs provide the opportunity to input the proper shift left bit number for these shifters.

Low hardware volume

The USFFT64 processor has the minimum multiplier number which is equal to 4. This fact makes this core attarctive to implement in ASIC. When configuring in Xilinx FPGA, these multipliers are implemented in 4 DSP48 units respectively. The user can select the input data, output data, and coefficient widths which provide application dynamic range needs. This can minimize both logic hardware and memory volume.

Interface:

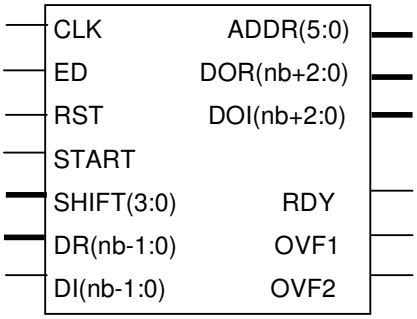

Figure 1. USFFT64 symbol.

Signal Description:

The descriptions of the core signals are represented in the table 1.

SIGNAL	TYPE	DESCRIPTION
CLK	input	Global clock
RST	input	Global reset
START	input	FFT start
ED	input	Input data and operation enable strobe
DR [nb-1:0]	input	Input data real sample
DI [nb-1:0]	input	Input data imaginary sample
SHIFT	input	Shift left code
RDY	output	Result ready strobe
WERES	output	Result write enable strobe
FFTRDY	output	Input data accepting ready strobe
ADDR [5:0]	output	Result number or address
DOR [nb+2:0]	output	Output data real sample
DOI [nb+2:0]	output	Output data imaginary sample
OVF1	output	Overflow flag
OVF2	output	Overflow flag

Table 1. IP core signals

Data representation

Input and output data are represented by *nb* and *nb*+3 bit twos complement complex integers, respectively. The twiddle coefficients are *nw* –bit wide numbers.

Typical Core Interconnection

The core interconnection depends on the application nature where it is used. The simple core interconnection considers the calculation of the unlimited data stream which are inputted in each clock cycle. This interconnection is shown on the figure 2.

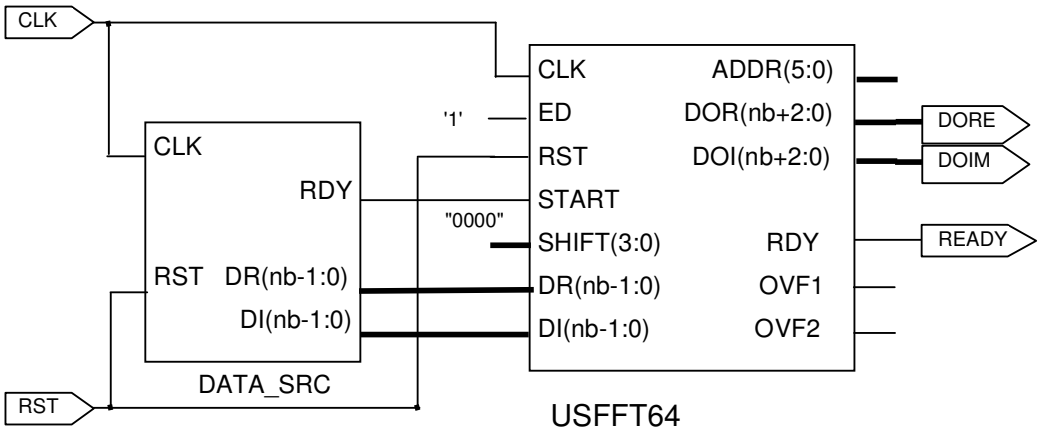

Figure 2. Simple core interconnection

Here DATA_SRC is the data source, for example, the analog-to-digital converter, USFFT64 is the core which is customized as one with 3 inner data buffers. The FFT algorithm starts with the impulse START. The respective results are outputted after the READY impulse and followed by the address code ADDR. The signal START is needed for the global synchronization, and can be generated once before the system operation.
The input data have the natural order, and can be numbered from 0 to 63. When 3 inner data buffers are configured then the output data have the natural order. When 2 inner data buffers are configured then the output data have the 8-th inverse order, i.e. the order is 0,8,16,...56,1,9,17,... .

Block Diagram

The basic block diagram of the USFFT64 core with two data buffers is shown in the fig.3.

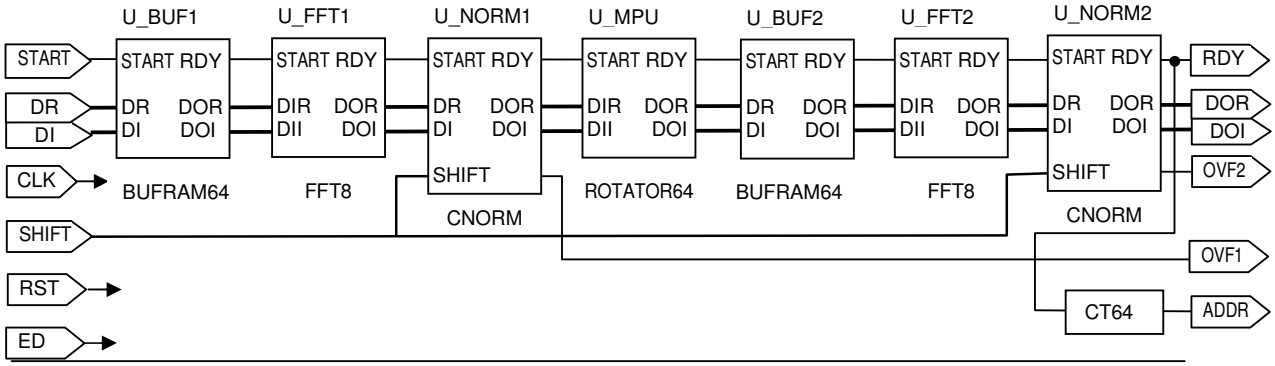

Fig.3. Block diagram of the USFFT64 core with two data buffers

Components:

BUFRAM64 – data buffer with row writing and column reading;
FFT8 – datapath, which calculates the 8-point DFT;
CNORM – shifter to 0,1,2,3 bit left shift;
ROTATOR64 – complex multiplier with twiddle factor ROM;
CT64 – counter modulo 64.

Implementation Data

The following table 2 illustrates the performance of the USFFT64 core with two data buffers based on BlockRAMs in Xilinx Virtex™ device when implementing 64-point FFT for 10 and 16-bit data and coefficients. Note that 4 DSP48 units and 8 RAMB16 units are used.

Target device and its data bit width	XC 4VSX25-12		XC 5VLX30-3	
	10	16	10	16
Area, Slices	2082 (20%),	2261 (22%),	785 (16%),	1071(22%)
Maximum system clock	254 MHz	228 MHz	315 MHz	300 MHz

Table 2. Implementation Data – Xilinx Virtex FPGA