****

**FE**

**Block Guide**

Version: **1.1**

Revision date: **12 12, 2016**

**Revision Control**

Table 1 - Revision Control

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Description** | **Authors** | **Revised by** |
| 1.0 | 31/08/2016 | Preliminary | Tomazine, L | Krüger, C |
| 1.1 | 12/12/2016 | Revised | Tomazine, L |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Contents

[1 Introduction 4](#_Toc473097236)

[1.1 Acronyms and Abbreviations 4](#_Toc473097237)

[1.2 Nomenclatures, Bit and Sample Representation 4](#_Toc473097238)

[1.3 References 5](#_Toc473097239)

[2 High-level Functional Description 6](#_Toc473097240)

[2.1 Features 6](#_Toc473097241)

[3 Macro Architecture 7](#_Toc473097242)

[3.1 Description 7](#_Toc473097243)

[3.2 External Interface 7](#_Toc473097244)

[3.3 Register Map 8](#_Toc473097245)

[3.3.1 IP Memory Map 12](#_Toc473097246)

[3.3.2 IP Register Description 12](#_Toc473097247)

[3.4 Modes of Operation 12](#_Toc473097248)

[3.4.1 Functional mode 12](#_Toc473097249)

[3.4.2 Test mode 13](#_Toc473097250)

[3.5 Clock and reset 13](#_Toc473097251)

[4 Micro Architecture 13](#_Toc473097252)

[4.1 Power 4 Module 13](#_Toc473097253)

[4.1.1 Power 4 Interface 13](#_Toc473097254)

[4.1.2 Power 4 Complexity Estimation 14](#_Toc473097255)

[4.2 FFT 512 Module 14](#_Toc473097256)

[4.2.1 FFT 512 Interface 14](#_Toc473097257)

[4.2.2 FFT 512 Complexity Estimation 15](#_Toc473097258)

[4.3 Peak Module 15](#_Toc473097259)

[4.3.1 Peak Interface 15](#_Toc473097260)

[4.3.2 Peak Complexity Estimation 15](#_Toc473097261)

FE

# Introduction

This document describes the behavioral SystemC implementation of the frequency offset estimator based on the frequency domain Mth power algorithm. At the SystemC model the algorithm was named FE (Frequency Offset Estimator).

## Acronyms and Abbreviations

Table 2 contains sample acronyms and abbreviations used in a document.

Table 2 - Acronyms and Abbreviated Terms

|  |  |
| --- | --- |
| **Term** | **Meaning** |
| FE | Frequency Offset Estimator |
| DFT | Discrete Fourier Transform |
| FFT | Fast Fourier Transform |
| N | FFT size (Default 512 samples) |

## Nomenclatures, Bit and Sample Representation

Next it is present the representation of bits used on the document. Types of registers and wires.

Table 3 – Nomenclatures, Bit and Sample Representation

|  |  |  |  |
| --- | --- | --- | --- |
| **Sample Representation** | MSS | Most Significant Sample | (*MSS:LSS)* |
| LSS | Least Significant Sample |
| **Bit Representation** | MSB | Most Significant Bit | [*MSB:LSB*] |
| LSB | Least Significant Bit |
| **Fixed Point Representation** | WL | Total Number of Bits - Word length | <*WL,IWL[,QUANT][,OVFLW][,NBITS]* > bits |
| IWL | Number of Integer portion length |
| QUANT | Quantization by default: SC\_TRN |
| OVFLW | Overflow by default: SC\_WRAP |
| NBITS | Number of saturated bits, only used for overflow mode and specifies how many bits will be saturated |
| **Logic Values** | '1' | Bit asserted | |
| '0' | Bit de-asserted | |
| 'x' | Bit don't care | |

QUANT mode can be one of the following:

* SC\_TRN – Truncation
* SC\_RND – Round toward plus infinity

OVFLW mode can be one of the following:

* SC\_WRAP – Wrap-around
* SC\_SAT - Saturation

## References

The references must include full references, such as standards, others documents and so on.

* MORELLI, MENGALI, 1998
* NAKAGAWA2011, Wide-Range and Fast-Tracking Frequency Offset Estimator for Optical Coherent Receivers

# High-level Functional Description

In coherent reception systems, the signal received is mixed with the transmitted signal by and a local oscillator. For accurate detection of this signal, it is necessary that local oscillator laser has been locked in frequency and phase with the laser transmitter. This is a complicated task due to the laser phase noise and the high cost to maintaining the transmitter local oscillator operating exactly at same frequency. Fortunately, it is possible to estimate and compensate the differences between the phases and frequencies of the lasers in digital domain by a carrier recovery module. If this error phase is not compensated, the constellation of a modulation system with a format M-QAM or M-PSK will suffer rotations for each symbol, affecting the correct symbol estimation.

This block aims to estimate the frequency offset between transmitter and local oscillator lasers. The estimation employs the frequency domain Mth power method.

Assuming a signal modulation format M-PSK, when to it is applied an Mth power, it becomes with a non-zero average. Spectrally, this means that appears an impulse at zero frequency (DC level). In this case if a frequency offset exist, this impulse will be shifted. It is possible to implement a frequency offset estimator checking if the peak is shifted, using an approach in the frequency domain.

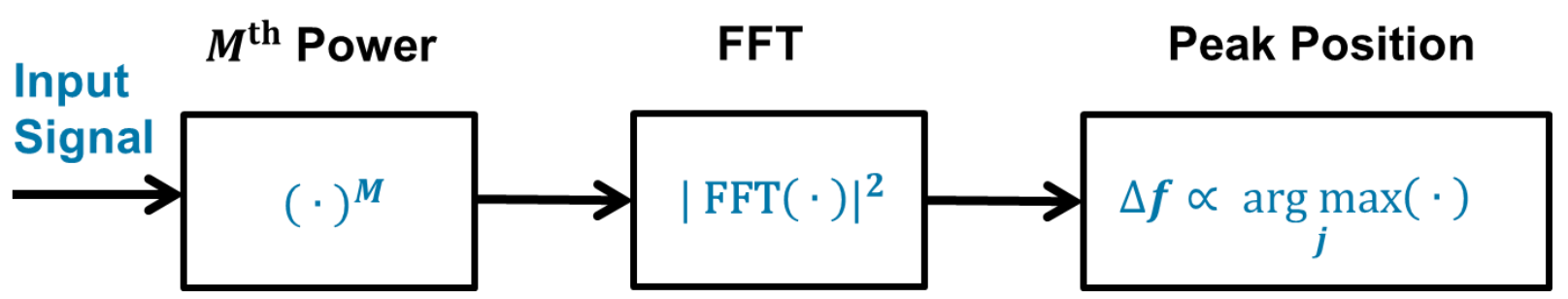


Figure 1 - FE Block Diagram.

## Features

The features of the block are listed below:

* Mth Power:
  + Each input sample is individually elevated to the Mth power
* 512 point FFT:
  + This block receives the Mth power signal and calculates its FFT
* Peak Position:
  + Finds the position of the maximum absolute value of the FFT result

# Macro Architecture

## Description

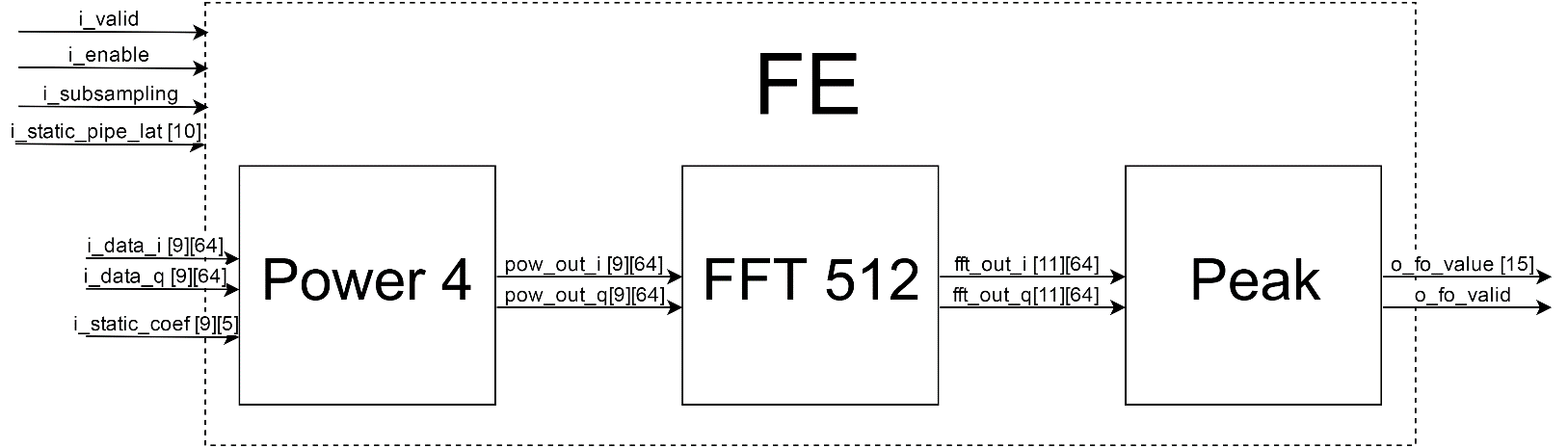
Description of the top module.

Figure 2 – FE Diagram

## External Interface

External interface and parameters of the top module. List of I/O pins and communication with other blocks.

Table 3 – External Interface

| **Signal** | **Width (bits)** | **I/O** | **Description** |
| --- | --- | --- | --- |
| clk | 1 | I | Input clock (437.5 MHz with duty cycle of 50%) |
| rst\_async\_n | 1 | I | Input reset - Active Low |
| i\_valid | 1 | I | Valid input signal – Active High |
| i\_enable | 1 | I | Block enable signal – Active High |
| i\_subsampling | 1 | I | Sub sampling mode flag – Active High |
| i\_static\_pipe\_lat | 10 | I | Latency clock cycles between two operations |
| i\_static\_coef [5] | 9 | I | Fir filter coefficients for sub-sampling mode on |
| i\_data\_i [64] | 9 | I | In phase symbol component |
| i\_data\_q [64] | 9 | I | Quadrature symbol component |
| o\_fo\_valid | 1 | O | Valid output signal – Active High |
| o\_fo\_value | 15 | O | Frequency Offset estimation |

Table 4 – Parameters

| **Parameter name** | **Description** | **Value** |
| --- | --- | --- |
| NBW\_IN | Number of word bits of the input | 9 |
| NBI\_IN | Number of integer bits of the input | 2 |
| NBW\_FFT | Number of word bits of the FFT output | 11 |
| NBI\_FFT | Number of integer bits of the FFT output | 2 |
| FE\_NS\_IN | Number of input samples | 64 |
| FE\_NS\_FFT | Number of samples for the FFT | 512 |
| FE\_NB\_MAX | Number of bits for the max tree input | 8 |
| FE\_NS\_FIR | FIR filter order | 5 |

## Register Map

### IP Memory Map

|  |  |  |  |
| --- | --- | --- | --- |
| **IPxyz Control Register - Base Address = 0x1000** | | | |
| **ADDRESS OFFSET (*ADDR[11:1])*** | **REGISTER NAME** | **TYPE** | **DEFAULT** |
| 0x000 | i\_enable | R/W | 0x0000 |
| 0x001 | i\_static\_pipe\_lat | R/W | 0x0064 |
| 0x002 | i\_static\_coef\_0 | R/W | 0x0003 |
| 0x003 | i\_static\_coef\_1 | R/W | 0x001E |
| 0x004 | i\_static\_coef\_2 | R/W | 0x003C |
| 0x005 | i\_static\_coef\_3 | R/W | 0x001E |
| 0x006 | i\_static\_coef\_4 | R/W | 0x0003 |
| 0x007 | o\_fo\_value | R/W | 0x0000 |

### IP Register Description

##### **i\_enable Register**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bits** | **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Field** | reserved | | | | | | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bits** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Field** | reserved | | | | | | | enable |
| **Default** | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |

**Bit 15-1: reserved**

Register field *15 to 1* are reserved bits for future applications and should not be written.

**Bit 0: enable**

The *enable* bit is the block enable signal.

##### **i\_static\_pipe\_lat Register**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bits** | **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Field** | reserved | | | | | | pipe9 | pipe8 |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bits** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Field** | pipe7 | pipe6 | pipe5 | pipe4 | pipe3 | pipe2 | pipe1 | pipe0 |
| **Default** | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |

**Bit 15-10: reserved**

Register field *15 to 10* are reserved bits for future applications and should not be written.

**Bit 9-0: pipe**

The *pipe9-pipe0* bits determine the number of clock cycles between 2 operations.

**i\_static\_coef\_0 Register**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bits** | **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Field** | reserved | | | | | | | coef0\_8 |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bits** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Field** | coef0\_7 | coef0\_6 | coef0\_5 | coef0\_4 | coef0\_3 | coef0\_2 | coef0\_1 | coef0\_0 |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

**Bit 15-9: reserved**

Register field *15 to 9* are reserved bits for future applications and should not be written.

**Bit 8-0: coef0**

The *coef0\_8- coef0\_0* are the coefficient values of the FIR filter used in the sub-sampling mode.

**i\_static\_coef\_1 Register**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bits** | **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Field** | reserved | | | | | | | coef1\_8 |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bits** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Field** | coef1\_7 | coef1\_6 | coef1\_5 | coef1\_4 | coef1\_3 | coef1\_2 | coef1\_1 | coef1\_0 |
| **Default** | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |

**Bit 15-9: reserved**

Register field *15 to 9* are reserved bits for future applications and should not be written.

**Bit 8-0: coef1**

The *coef1\_8- coef1\_0* are the coefficient values of the FIR filter used in the sub-sampling mode.

**i\_static\_coef\_2 Register**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bits** | **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Field** | reserved | | | | | | | coef2\_8 |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bits** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Field** | coef2\_7 | coef2\_6 | coef2\_5 | coef2\_4 | coef2\_3 | coef2\_2 | coef2\_1 | coef2\_0 |
| **Default** | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |

**Bit 15-9: reserved**

Register field *15 to 9* are reserved bits for future applications and should not be written.

**Bit 8-0: coef2**

The *coef2\_8- coef2\_0* are the coefficient values of the FIR filter used in the sub-sampling mode.

**i\_static\_coef\_3 Register**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bits** | **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Field** | reserved | | | | | | | coef3\_8 |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bits** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Field** | coef3\_7 | coef3\_6 | coef3\_5 | coef3\_4 | coef3\_3 | coef3\_2 | coef3\_1 | coef3\_0 |
| **Default** | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |

**Bit 15-9: reserved**

Register field *15 to 9* are reserved bits for future applications and should not be written.

**Bit 8-0: coef3**

The *coef3\_8- coef3\_0* are the coefficient values of the FIR filter used in the sub-sampling mode.

**i\_static\_coef\_4 Register**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bits** | **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Field** | reserved | | | | | | | coef4\_8 |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bits** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Field** | coef4\_7 | coef4\_6 | coef4\_5 | coef4\_4 | coef4\_3 | coef4\_2 | coef4\_1 | coef4\_0 |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

**Bit 15-9: reserved**

Register field *15 to 9* are reserved bits for future applications and should not be written.

**Bit 8-0: coef4**

The *coef4\_8- coef4\_0* are the coefficient values of the FIR filter used in the sub-sampling mode.

**o\_fo\_value Register**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bits** | **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Field** | reserved | value14 | value13 | value12 | value11 | value10 | value9 | value8 |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bits** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Field** | value7 | value6 | value5 | value4 | value3 | value2 | value1 | value0 |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 15: reserved**

Register field *15* is reserved bits for future applications and should not be written.

**Bit 14-0: value**

The *value14 – value0* is the frequency offset estimation value.



## Modes of Operation

There are basically two modes of operation for the FE block: normal mode and sub-sampling mode. In normal mode the input samples are directly sent through the Mth power block and the result is sent to the FFT block, taking 8 clock cycles to complete the 512 samples needed to compute de 512 point FFT (8 cycles x 64 samples/cycle).

For the sub-sampling mode the block will consider only 1 sample in every 16 input samples. In that way, there will be only 4 valid samples in each clock cycle. To prevent aliasing, the samples are filtered using a 5 order FIR filter after the Mth power block. So, in the sub-sampling mode it is necessary 128 clock cycles (8\*16) to feed the FFT input.

In both modes there will be a configurable latency between a valid output and the start of a new operation, defined by the *i\_static\_pipe\_lat* input value. Also is important to mention that the output value will be the current value (calculated) plus the last output value and the sub-sampling flag will be registered only in an output valid (*o\_fo\_valid*) occurrence.

### Functional mode

For an operation in the functional mode the *i\_enable* port must be active, along with the *rst\_async\_n* signal.

### Test mode

## Clock and reset

The clock of the system, droved by the port *clk*, is 437.5 MHz with a 50% duty cycle. For the reset functionality, it is used the *rst\_async\_n* port, which have an asynchronal behavior and is active low. The reset port needs to be active for 2 cycles to a proper system restart.

# Micro Architecture

## Power 4 Module

This block basically raises the input signal to the power of 4 aiming to undo the modulation. The relation between the input data and the output data is given by the equation:

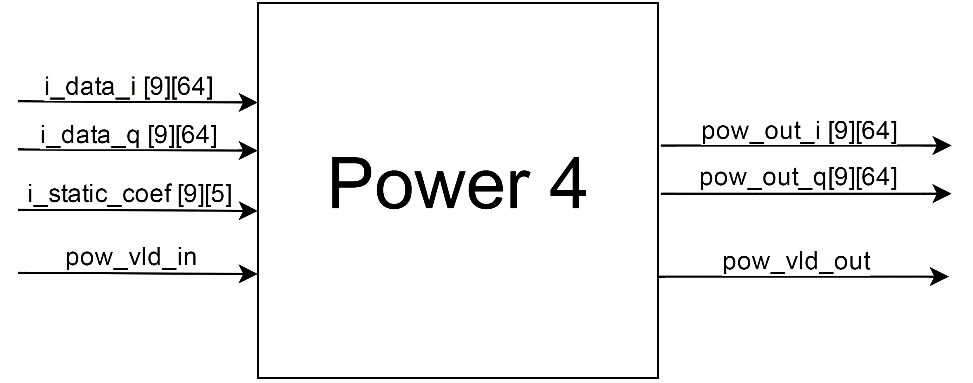


Figure 3 – Power 4 Diagram

### Power 4 Interface

Block I/O pins.

Table 6 – Power 4 Interface

| **Signal** | **I/O** | **Description** | **From/To** |
| --- | --- | --- | --- |
| clk | I | Input clock at 437.5Mhz with duty cycle of 50% | FE |
| rst\_async\_n | I | Input reset - Active Low | FE |
| pow\_vld\_in | I | Valid input signal – Active High | FE |
| i\_data\_i [64] | I | In phase symbol component | FE |
| i\_data\_q [64] | I | Quadrature symbol component | FE |
| i\_coef [5] | I | Fir filter coefficients for sub-sampling mode on | FE |
| pow\_out\_i [64] | O | Signal raised to power of 4 (in phase component) | FFT 512 |
| pow\_out\_q [64] | O | Signal raised to power of 4 (quadrature component) | FFT 512 |
| pow\_vld\_out | O | Valid output signal – Active High | FFT 512 |

### Power 4 Complexity Estimation

Complexity estimation of the sub-block using the number of arithmetic operators.

Table 9 – Power 4 Complexity

| **Operator** | **Number of instantiations** |
| --- | --- |
| Adder | 64 |
| Subtractor | 128 (2\*64) |
| Multiplier | 576 (9\*64) |

## FFT 512 Module

This module is aimed to perform the discrete Fourier transform (DFT) through the fast algorithm (FFT). The porpoise of pass the signal to the frequency domain is to estimate the frequency peak and therefore the frequency offset. The FFT is calculated using 64 instances of an 8 points serial FFT and 1 instance of a 64 points FFT. In this way the 512 points FFT is calculated in 8 steps of 64 points each one.

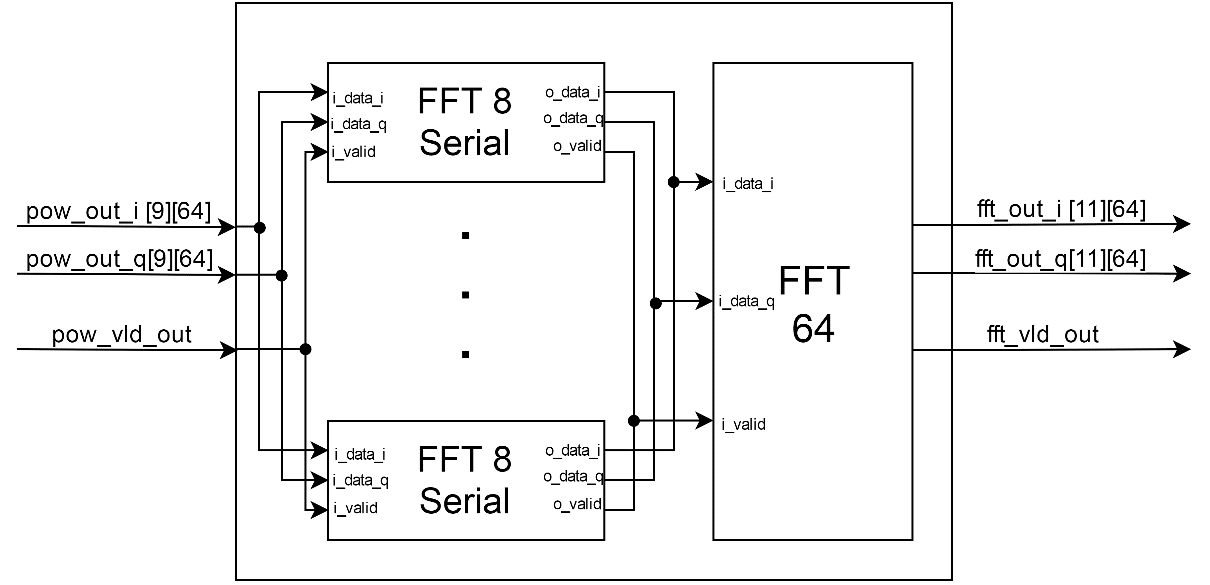


Figure 3 –FFT 512 Diagram

### FFT 512 Interface

Block I/O pins.

Table 6 – FFT 512 Interface

| **Signal** | **I/O** | **Description** | **From/To** |
| --- | --- | --- | --- |
| clk | I | Input clock at 437.5Mhz with duty cycle of 50% | FE |
| rst\_async\_n | I | Input reset - Active Low | FE |
| pow\_out\_i [64] | I | Signal raised to power of 4 (in phase component) | POWER 4 |
| pow\_out\_q [64] | I | Signal raised to power of 4 (quadrature component) | POWER 4 |
| pow\_vld\_out | I | Valid input signal – Active High | POWER 4 |
| fft\_out\_i [64] | O | FFT result data (in phase component) | PEAK |
| fft\_out\_q [64] | O | FFT result data (quadrature component) | PEAK |
| fft\_vld\_out | O | Valid output signal – Active High | PEAK |

### 

### FFT 512 Complexity Estimation

Resources: the DIT FFT radix-2 has 5N(log2 N) operations. So, for an N equal to 64: 1920 operations and for an N equal to 8: 120 operations. Then for this block the estimated resources are 1920+64\*120 = 9600.

## Peak Module

This module is aimed to obtain the absolute value of the complex FFT output signal and then find the peak position of the these values. With this information, the block also calculate the frequency offset of the estimator input data.

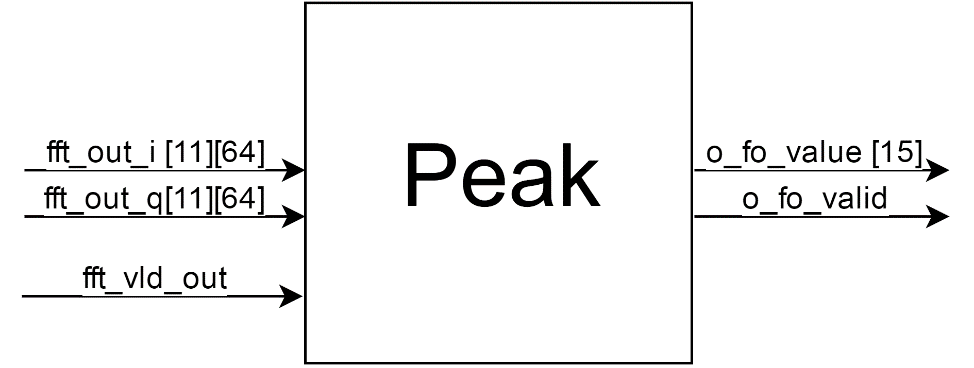


Figure 3 – Peak Diagram

### Peak Interface

Block I/O pins.

Table 6 – Peak Interface

| **Signal** | **I/O** | **Description** | **From/To** |
| --- | --- | --- | --- |
| clk | I | Input clock at 437.5Mhz with duty cycle of 50% | FE |
| rst\_async\_n | I | Input reset - Active Low | FE |
| fft\_out\_i [64] | I | FFT result data (in phase component) | FFT 512 |
| fft\_out\_q [64] | I | FFT result data (quadrature component) | FFT 512 |
| fft\_vld\_out | I | Valid input signal – Active High | FFT 512 |
| o\_fo\_value | O | Valid output signal – Active High | FE |
| o\_fo\_valid | O | Frequency Offset estimation | FE |

### Peak Complexity Estimation

Complexity estimation of the sub-block using the number of arithmetic operators.

Table 9 – Peak Complexity

| **Operator** | **Number of instantiations** |
| --- | --- |
| Adder | 64 |
| Multiplier | 128 (64\*2) |
| Comparators | 63 (32+16+8+4+2+1) |