Instrument ShAred Artifact for Computing (ISAAC)

Newton QMFIR Spec

FPGA Specification for ISAAC Newton Quadrature Modulation with Polyphase Decimation Filter

ISAAC Document Number: D-xxxxxx

Revision: 1.0

FPGA Rev:

Date: **January 14, 2009**

FPGA Authors: Kayla Nguyen
Jason Zheng

Custodian: Kayla Nguyen

Summary: This document serves as the documentation for the FPGA design for ISAAC Newton Quadrature Modulation with Polyphase Decimation Filter. This document only has the information relating to the FPGA and its internal modules. This does not provide interface information between the FPGA and its surroundings.

.

Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, CA 91109-8099

Contents

List of Figures

List of Tables

Part I

INTRODUCTION

0.1 Purpose

This document serves as the ISAAC Newton Quadrature Modulation with Polyphase Decimation Filter (QMFIR) FPGA design specification. This has sufficient details for future ISAAC FPGA review and usage of the core.

0.2 ISAAC Background

ISAAC is a 3-year R&TD task (starting on fiscal year 2008), focusing on reusable modules used by different instruments. Currently there are four subtasks within the ISAAC framework: Asimov, Stern, Singer, and Newton. ISAAC Newton focuses on the radar application; specifically for fiscal year 2008, the on-board processing power which eventually supports the SMAP (Soil Moister Active-Passive) project. The FPGA module QMFIR is the on-board processing Quadrature Modulation and Polyphase Decimation Filter.

0.3 Overview of Requirements

Table 1: Digital Filter Design Parameters

Design Parameter	Requirement
Bandwidth	1 MHz
Input Sampling Rate	60 MHz
Output Sampling Rate	1.2 MHz
Passband Ripple	-40 dB
Stopband Ripple	0.1 dB
Integrate SideLobe Ratio (ISLR)	-20 dB
Peak SideLobe Ration (PSLR)	-30 dB
Pulse Broadening	1.2

Algorithm development for the digital filter was done by Charles Le (section 334).

Chapter 1

QMFIR FPGA Overall Architecture

1.1 One Stream (QM + FIRDecim)

The overall architecture of the ISAAC Newton FPGA consists of the Quadrature Modulation (QM) module along with three Polyphase Decimation Finite Impulse Response Filter (FIRDecim) modules. The data flow between the modules is as follows:

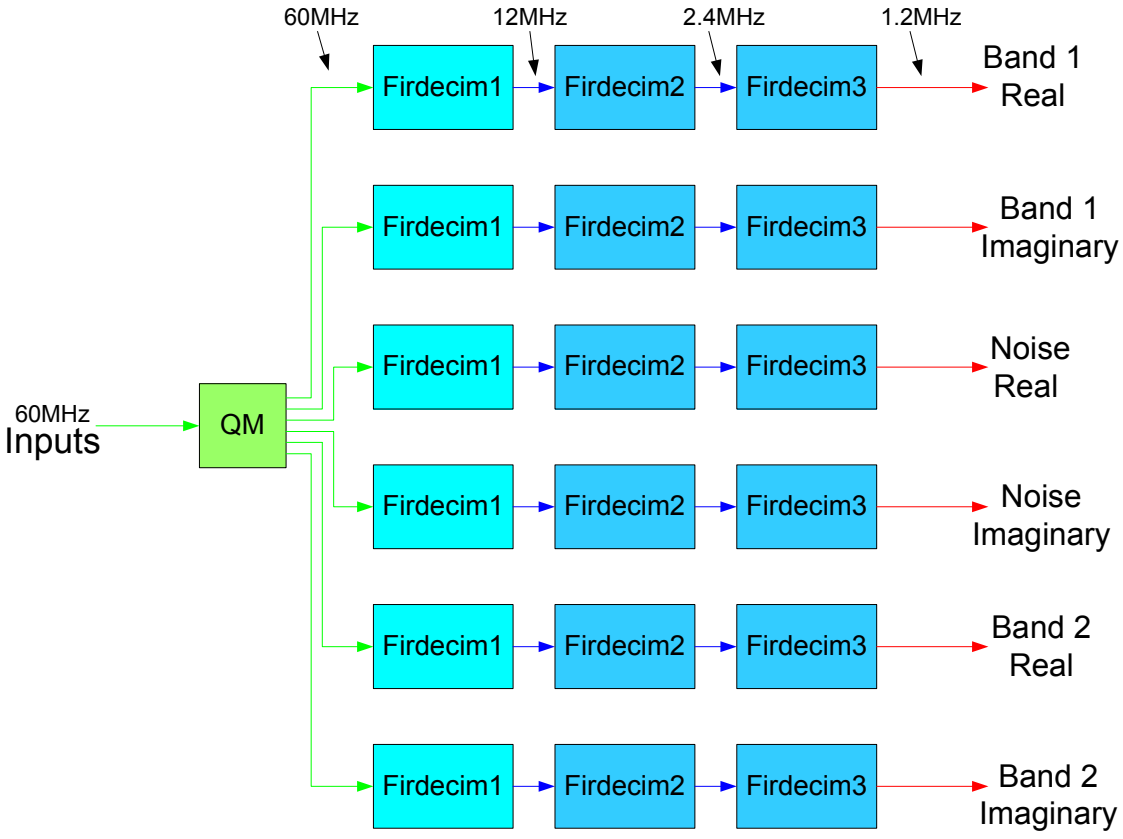

Figure 1.1: ISAAC Newton FPGA Architecture (QM + FIRDecim)

From figure (1.1), the input to the QM module is a 60MHz ADC signal. The goal is to bring the input

signal down to baseband and decimate it by 50 times such that the output is at 1.2MHz. The Quadrature Modulation module takes in one input and splits the signals into its respective Band 1, Band 2, and noise band (real and imaginary for all 3 bands). At the end, there are six chains of FIRDecim filters. Each chain of filters consists of three separate filter modules called FIRDecim1, FIRDecim2, and FIRDecim3. Each FIRDecim module is designed specifically to the input bandwidth, output bandwidth, and the decimation factor. The next few chapters describes in detail all of the blocks from the above figure.

Part II

QMFIR FPGA MODULES DESCRIPTION

Chapter 2

Quadrature Modulation

2.1 Theory of QM

Quadrature Modulation takes an incoming signal and demodulate the incoming signal from 60MHz down to baseband by performing a multiplication with sine or cosine:

$$\begin{aligned} \text{Real\_Output} &= \text{Input} * \cos(2\pi f) \\ \text{Imag\_Output} &= \text{Input} * \sin(2\pi f) \end{aligned}$$

where f is the frequency and is defined as

$$f = \frac{\text{center frequency}}{\text{sampling frequency}} \tag{2.1}$$

2.2 Interface

Figure (2.1) shows the interface block diagram of the QM module. It shows the inputs and outputs interface, as well as the internal modules that it uses.

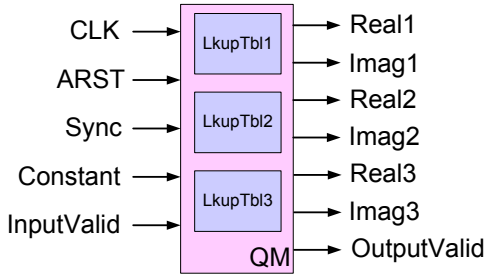

Figure 2.1: QM Interface Block

Table 2.1: Quadrature Modulation Module Signals

Signal	Direction	Size	Active Level	Description
CLK	Input	1-bit	Rising Edge	Clock input value.
ARST	Input	1-bit	High	Asynchronous reset. All module registers are cleared when this signal is high
Sync	Input	1-bit	High	Synchronization to beginning of the look-up tables. All counters are cleared when this signal is high.
Constant	Input	IWIDTH	N/A	Input data.
InputValid	Input	1-bit	High	Indicate that the value on the Constant signal is valid.
Real1	Output	OWIDTH	N/A	Band 1 real data output. Result of input by cosine of frequency 1.
Imag1	Output	OWIDTH	N/A	Band 1 imaginary data output. Result of input by sine of frequency 1.
Real2	Output	OWIDTH	N/A	Noise real data output. Result of input by cosine of noise frequency.
Imag2	Output	OWIDTH	N/A	Noise imaginary data output. Result of input by sine of noise frequency.
Real3	Output	OWIDTH	N/A	Band 2 real data output. Result of input by cosine of frequency 2.
Imag3	Output	OWIDTH	N/A	Band 2 imaginary data output. Result of input by sine of frequency 2.
OutputValid	Output	1-bit	High	Indicate that the output data are valid.

2.3 Parameters

The main parameters that are open to the user for the QM module are:

OWIDTH Output bit-width
IWIDTH Input bit-width
MAXCNT1 Counter up to repeating for channel 1 & 2 (Look-up table 1 & 3)
MAXCNT2 Counter up to repeating for channel 3 (Look-up table 2)
CNTWIDTH1 Counter bit-width for channel 1 & ch 2 (Look-up table 1 & 3)
CNTWIDTH2 Counter bit-width for channel 3 (or Look-up table 2)

The parameters MAXCNT1, MAXCNT2, CNTWIDTH1, CNTWIDTH2 can be found within the Look-up table Verilog files.

2.4 Look-up Table

In order to implement the sine and cosine function, look-up tables specialized in the specific frequencies are used.

2.4.1 Look-up Table Interface

The Look-up Table (LkupTbl) that is used by the QM module is the sin/cos look-up table. Figure (2.2) shows the interface of the look-up table module.

Figure 2.2: LUT Interface Block

Table 2.2: Quadrature Modulation Module Signals

Signal	Direction	Size	Active Level	Description
CLK	Input	1-bit	Rising Edge	Input clock value.
ARST	Input	1-bit	High	Asynchronous reset. All module registers are cleared when this signal is high
cntr	Input	CNTWIDTH1, CNTWIDTH2	N/A	Input counter to select memory space.
sine	Output	OWIDTH	N/A	Output sine data from memory.
cosine	Output	OWIDTH	N/A	Output cosine data from memory.

2.4.2 Look-up Table Generation

The look-up table used by the QM module is a memory mapping to each cntr address. The Verilog code for the look-up table is generated by a python script. This script is able to optimized the look-up table to a center frequency and to the sampling frequency. For the specific SMAP radar application, these are the requirements:

Sampling Frequency = 60MHz
Center Frequency 1 = 12.5MHz
Noise Frequency = 15MHz
Center Frequency 2 = 17.5MHz

The sine and cosine outputs are defined as follows:

$$\text{sine} = \sin\left(2\pi \cdot \text{cntr} \cdot \frac{\text{center frequency}}{\text{sampling frequency}}\right)$$
$$\text{cosine} = \cos\left(2\pi \cdot \text{cntr} \cdot \frac{\text{center frequency}}{\text{sampling frequency}}\right)$$

The outputs of the three look-up tables used by the QM module are different, depending on the specific frequencies.

2.4.3 Look-up Table Memory

As mentioned above, the look-up tables are optimized to their center frequencies and sampling frequencies. This means that the sine and cosine function repeats, minimizing the memory size of the look-up tables.

Table 2.3: Memory Size of the Three Look-up Tables in QM Module

	LkupTbl1	LkupTbl2	LkupTbl3
Memory	24	4	24

From the above table, the counter (cntr) into the module has fixed bit width, CNTWIDTH1/CNTWIDTH2, depending on the memory size. This parameter is also optimized during the python scripting step with no extra inputs from the user.

2.4.3.1 Look-up Table 1

Note that the values in the Look-up Tables are stored in two's complement binary number, with 2 bits of integer and 14 bits of decimal.

Center Frequency = 12.5MHz

Table 2.4: Memory Map of LkupTbl1 (Channel 1)

Memory (cntr)	Sine Value	Cosine Value	Equivalent Input in Radians
0	0000_0000_0000_0000	0100_0000_0000_0000	0
1	0011_1101_1101_0001	0001_0000_1001_0000	1.3090
2	0001_1111_1111_1111	1100_1000_1001_0100	2.6180
3	1101_0010_1011_1111	1101_0010_1011_1111	3.9270
4	1100_1000_1001_0100	0010_0000_0000_0000	5.2360
5	0001_0000_1001_0000	0011_1101_1101_0001	0.2618
6	0100_0000_0000_0000	0000_0000_0000_0000	1.5708
7	0001_0000_1001_0000	1100_0010_0010_1111	2.8798
8	1100_1000_1001_0100	1110_0000_0000_0001	4.1888
9	1101_0010_1011_1111	0010_1101_0100_0001	5.4978
10	0001_1111_1111_1111	0011_0111_0110_1100	0.5236
11	0011_1101_1101_0001	1110_1111_0111_0000	1.8326
12	0000_0000_0000_0000	1100_0000_0000_0000	3.1416
13	1100_0010_0010_1111	1110_1111_0111_0000	4.4506
14	1110_0000_0000_0000	0011_0111_0110_1100	5.7596
15	0010_1101_0100_0001	0010_1101_0100_0001	0.7854
16	0011_0111_0110_1100	1110_0000_0000_0000	2.0944
17	1110_1111_0111_0000	1100_0010_0010_1111	3.4034
18	1100_0000_0000_0000	0000_0000_0000_0000	4.7124
19	1110_1111_0111_0000	0011_1101_1101_0001	6.0214
20	0011_0111_0110_1100	0010_0000_0000_0000	1.0472
21	0010_1101_0100_0001	1101_0010_1011_1111	2.3562
22	1110_0000_0000_0001	1100_1000_1001_0100	3.6652
23	1100_0010_0010_1111	0001_0000_1001_0000	4.9742

2.4.3.2 Look-up Table 2

Center Frequency = 15MHz

Table 2.5: Memory Map of LkupTbl2 (Noise Channel)

Memory (cntr)	Sine Value	Cosine Value	Equivalent Input in Radians
0	0000_0000_0000_0000	0100_0000_0000_0000	0
1	0100_0000_0000_0000	0000_0000_0000_0000	1.5708
2	0000_0000_0000_0000	1100_0000_0000_0000	3.1416
3	1100_0000_0000_0000	0000_0000_0000_0000	4.7124

2.4.3.3 Look-up Table 3

Center Frequency = 17.5MHz

Table 2.6: Memory Map of LkupTbl3 (Channel 2)

Memory (cntr)	Sine Value	Cosine Value	Equivalent Input in Radians
0	0000_0000_0000_0000	0100_0000_0000_0000	0
1	0011_1101_1101_0001	1110_1111_0111_0000	1.8326
2	1110_0000_0000_0001	1100_1000_1001_0100	3.6652
3	1101_0010_1011_1111	0010_1101_0100_0001	5.4978
4	0011_0111_0110_1100	0010_0000_0000_0000	1.0472
5	0001_0000_1001_0000	1100_0010_0010_1111	2.8798
6	1100_0000_0000_0000	0000_0000_0000_0000	4.7125
7	0001_0000_1001_0000	0011_1101_1101_0001	0.2618
8	0011_0111_0110_1100	1110_0000_0000_0001	2.0944
9	1101_0010_1011_1111	1101_0010_1011_1111	3.9270
10	1110_0000_0000_0000	0011_0111_0110_1100	5.7596
11	0011_1101_1101_0001	0001_0000_1001_0000	1.3090
12	0000_0000_0000_0000	1100_0000_0000_0000	3.1416
13	1100_0010_0010_1111	0001_0000_1001_0000	4.9742
14	0001_1111_1111_1111	0011_0111_0110_1100	0.5236
15	0010_1101_0100_0001	1101_0010_1011_1111	2.3542
16	1100_1000_1001_0100	1110_0000_0000_0000	4.1888
17	1110_1111_0111_0000	0011_1101_1101_0001	6.0214
18	0100_0000_0000_0000	0000_0000_0000_0000	1.5708
19	1110_1111_0111_0000	1100_0010_0010_1111	3.4034
20	1100_1000_1001_0100	0001_1111_1111_1111	5.2360
21	0010_1101_0100_0001	0010_1101_0100_0001	0.7854
22	0010_0000_0000_0000	1100_1000_1001_0100	2.6180
23	1100_0010_0010_1111	1110_1111_0111_0000	4.4506

2.5 Quadrature Modulation Core

The Quadrature Modulation (QM) core consists of 6 multipliers along with the three look-up tables described above. The module takes in the measured data and demodulates it according to the center frequencies. The algorithm for the quadrature modulation is:

$$\begin{aligned} \text{Real} &= \text{Constant} * \text{Cosine\_n} \\ \text{Imag} &= \text{Constant} * \text{Sine\_n} \end{aligned}$$

where Cosine_n and Sine_n are the outputs of the Look-up Table for n=1:3.

Figure (2.3) shows the logic for the QM core.

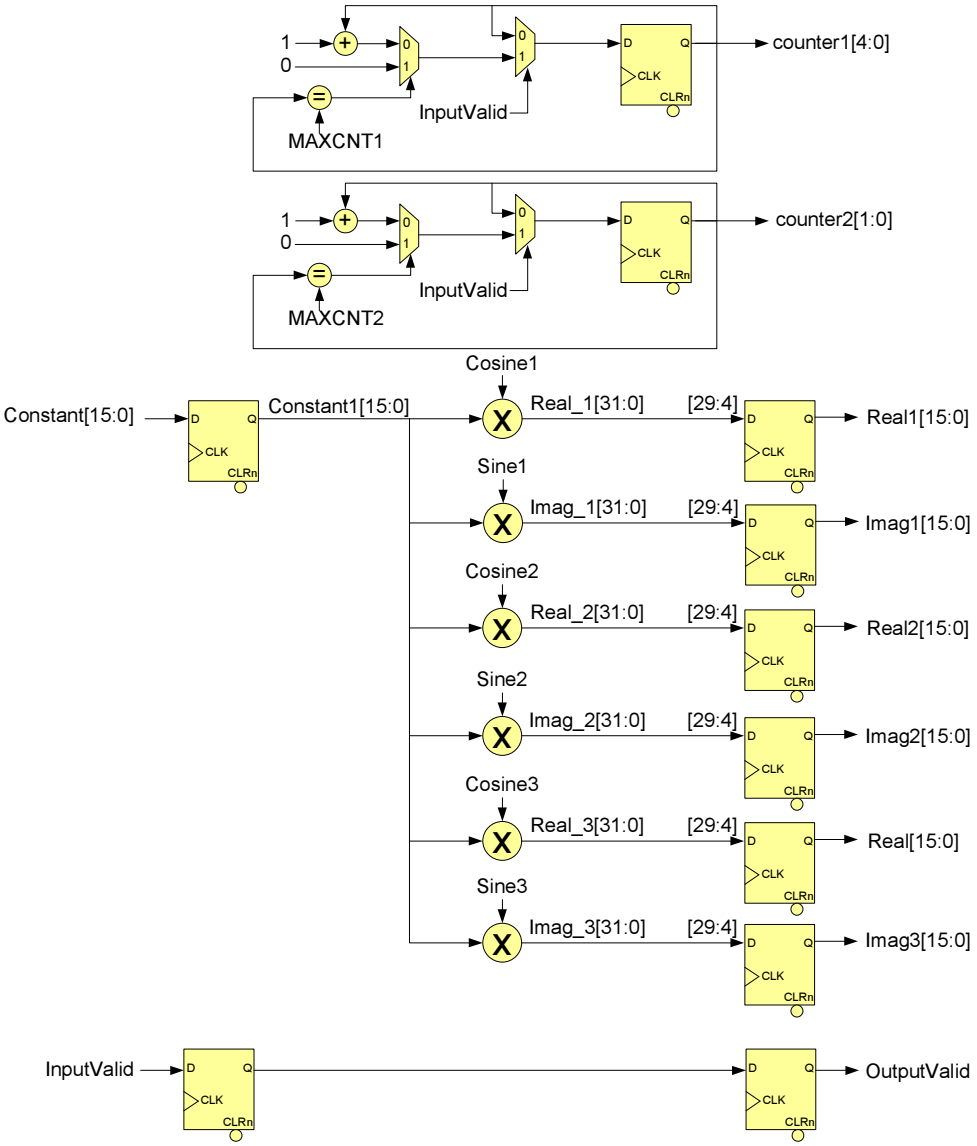

Figure 2.3: QM Logic

In figure (2.3), counter1 is the input counter to look-up table 1 and 3 since they have the same memory size, and counter2 is the input counter to look-up table 2.

2.6 Resource Usage

Table 2.6 summarizes the resource usage for different devices for the Quadrature Modulation module. Note that the input bit width (IWIDTH) and output bit width (OWIDTH) are 16-bit in this resource usage report.

Table 2.7: Resource Usage for Quadrature Modultion Module

Device	XC2V3000-BF957-4	XQR4VFX60-CF1144-10	XC5VFX130F-FF1738-1
Maximum Speed	115MHz	150MHz	175MHz
# of Multipliers	6	6	6
# of Flip Flops	177	177	177
# of LUTs	108	108	63
# of Slices	114	114	80
Power Usage			

2.7 Code Download

Chapter 3

Polyphase Decimation Filter 1

3.1 Theory of Polyphase Decimcation Filter

The purpose of the Polyphase Decimation Filter is to use a Finite Impulse Response Filter (FIR) to decimate the incoming signal by a factor of M.

3.2 Design of Firdecim1

The Polyphase Decimation Filter 1 (FIRDecim1) is a filter with 15 taps which filters the input signal and decimates the input by a factor of 5. The core of the FIRDecim1 design is the multiply-accumulate (MAC) unit. The basic idea behind the design for FIRDecim1 is given in figure (3.1).

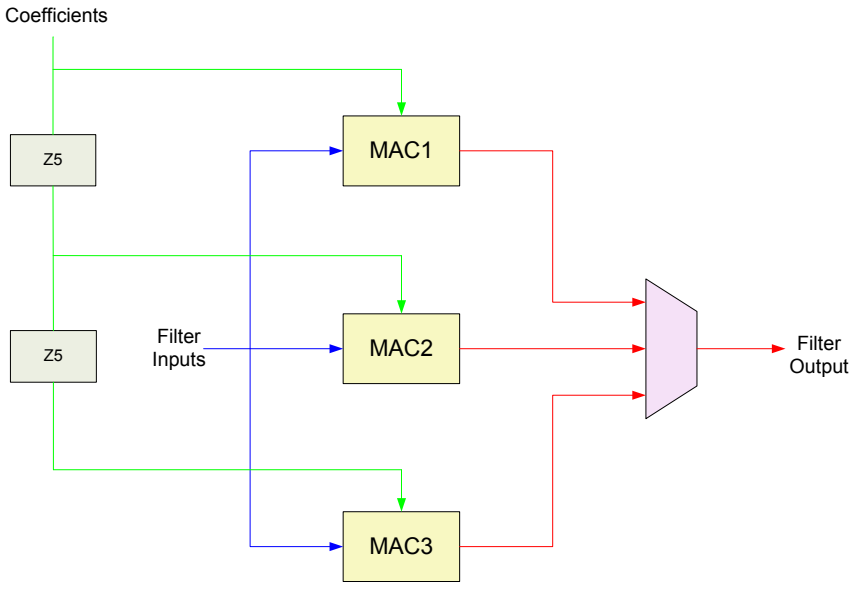

Figure 3.1: FIRDecim1 Design Block Diagram

Table 3.1: FIRDecim1 Design Parameters

Design Parameter	Requirement
Input clock rate	60 MHz
Input data rate	60 MHz
Output data rate	12 MHz
Number of taps	15
Decimation factor	5

3.3 Interface

Figure (3.2) shows the inputs and outputs of the FIRDecim1 module.

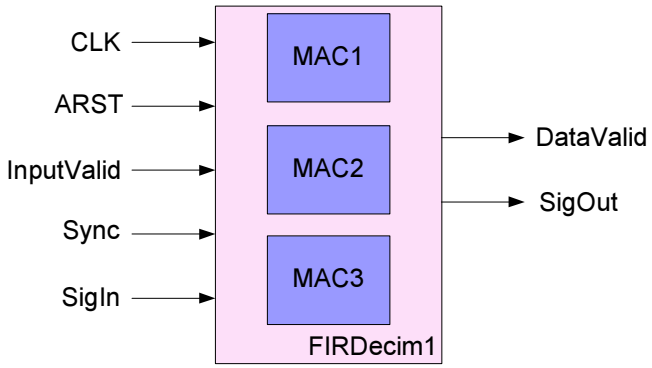

Figure 3.2: FIRDecim1 Inputs and Outputs

Table 3.2: Interface Signals for Firdecim1

Signal	**Direction**	**Size**	**Active Level**	**Description**
CLK	Input	1-bit	Rising Edge	Clock input value.
ARST	Input	1-bit	High	Asynchronous reset. All module registers are cleared when this signal is high
Sync	Input	1-bit	High	Synchronization to beginning of the look-up tables. All counters are cleared when this signal is high.
InputValid	Input	1-bit	High	Indicate that SigIn is valid.
SigIn	Input	IWIDTH	N/A	Input data.
DataValid	Output	1-bit	High	Indicate that SigOut is valid.
SigOut	Output	OWIDTH	N/A	Output data.

3.4 Parameters

The goal of ISAAC is to make cores which are parameterizable to different instrument needs. Therefore, the following is a list of the parameters that can be adjusted as needed according to design.

e1 - eNTAPS	Filter Coefficients (NTAPS of them)
OWIDTH	Output bit width
IWIDTH	Input bit width
NTAPS	Number of taps
CNTWIDTH	Counter bit width

Note that the current Verilog code is written for NTAPS = 15 taps.

3.5 Multiply-Accumulate (MAC)

3.5.1 MAC Interface

The inputs and output of the MAC module is given in figure (3.3).

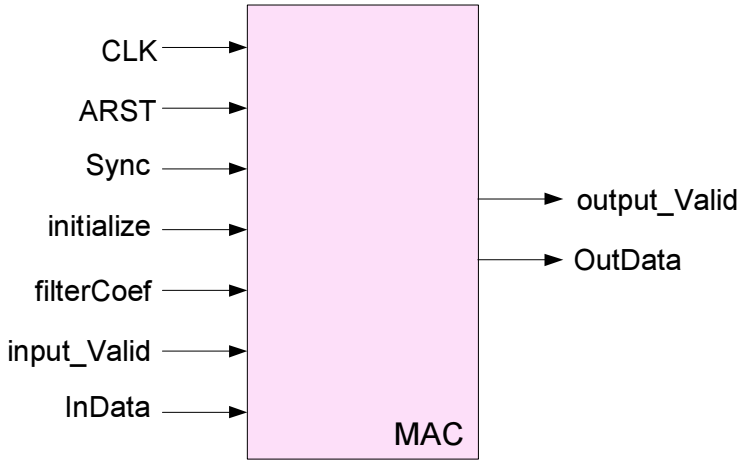

Figure 3.3: MAC Inputs and Outputs

Table 3.3: Interface Signals for MAC

Signal	Direction	Size	Active Level	Description
CLK	Input	1-bit	Rising Edge	Clock input value.
ARST	Input	1-bit	High	Asynchronous reset. All module registers are cleared when this signal is high
Sync	Input	1-bit	High	Synchronization to beginning of the look-up tables. All counters are cleared when this signal is high.
initialize	Input	1-bit	High	Initialize the accumulator and counter to 0
filterCoef	Input	IWIDTH	N/A	Filter coefficient input
input_Valid	Input	1-bit	High	Indicates that value in InData is valid
InData	Input	IWIDTH	N/A	Input data
output_Valid	Output	1-bit	High	Indicates that value in OutData is valid
OutData	Output	OWIDTH	N/A	Output Data from MAC

3.5.2 MAC Parameters

The Multiply-Accumulate (MAC) unit has been designed to be parameterizable. This MAC can be incorporated into any filter of any length with any numbers of taps. It also takes into account that not every input data is valid, and only accumulates the inputs that are valid. The parameters are:

IWIDTH Input bit width
OWIDTH Output bit width
AWIDTH Internal bit width (accumulator, multiplier)
NTAPS Total number of taps (including zeros)
NTAPSr Real number of taps in the filter
CNTWIDTH Counter bit width
NMULT Number of clocks it takes for multiplier to generate answer
This is to accommodate slow multipliers

3.5.3 MAC Logic

The logic for the MAC module is shown in figure (3.4). The design of this MAC allows the user to control when to reset the accumulator, counter, and when to assert the output_Valid signal. The reset of the accumulator and counter depends on the initialize signal. The assertion of output_Valid depends on the number of taps that is chosen as a parameter and the number of clocks it takes the multiplier to generate a valid answer.

For the current ISAAC design, the MAC resets and outputs a valid data every 15 clocks.

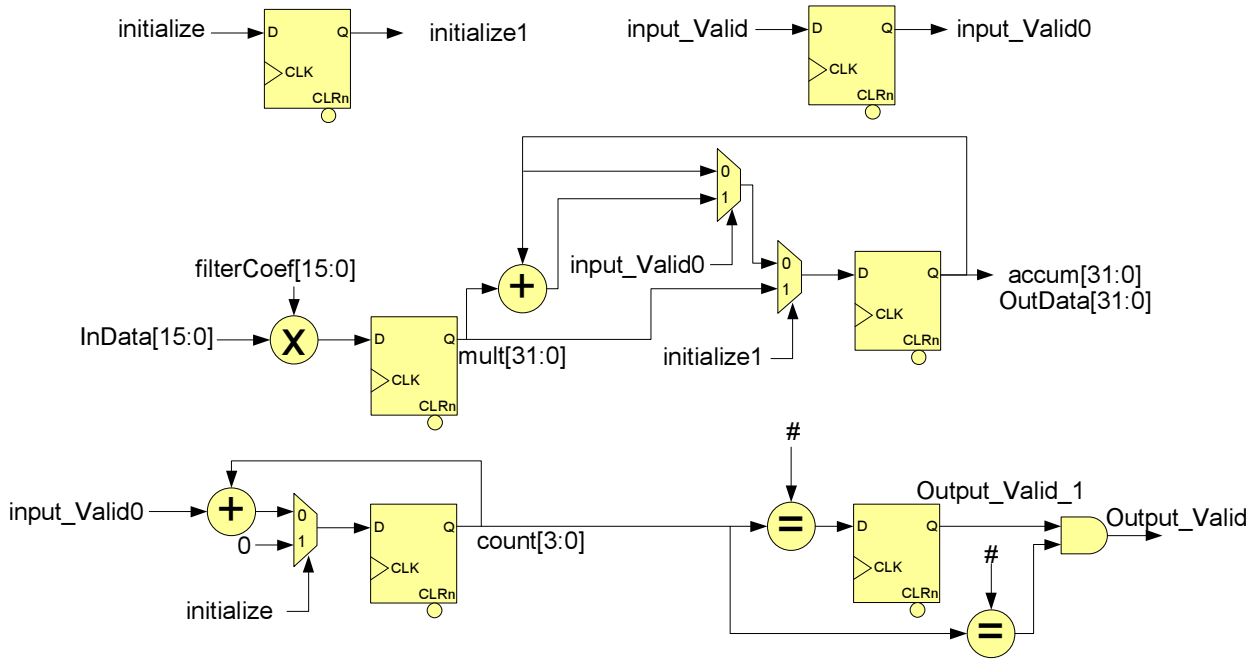

Figure 3.4: MAC Logic

3.6 FIRDecim1 Module

The FIRDecim1 block diagram was shown in figure (3.1). Figure (3.5) shows how the inputs to the MACs are generated inside the module, and how the outputs of the MACs are generated to be the outputs of the FIRDecim1 core.

There is an internal counter inside the FIRDecim1 block that controls when each MAC gets initialized. Each input signal to the MACs come straight from the inputs of the FIRDecim1 core. The coefficients to the MAC are generated this way:

MAC1	MAC filterCoef comes from coe0
	When the core gets Sync or ARST, the initialized value of coe0 is the parameter e0
MAC2	MAC filterCoef comes from coe10
	When the core gets Sync or ARST, the initialized value of coe10 is the parameter e10
MAC3	MAC filterCoef comes from coe5
	When the core gets Sync or ARST, the initialized value of coe5 is the parameter e5

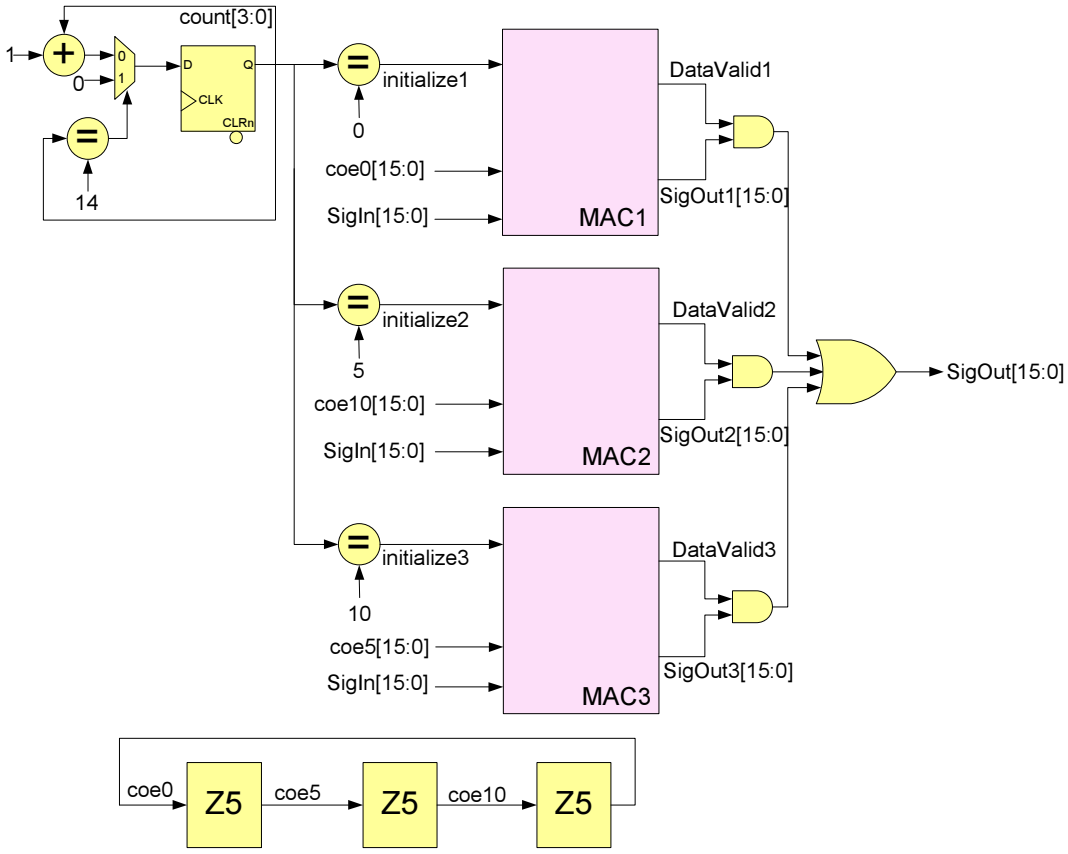

Figure 3.5: FIRDecim1 Logic

After initialization, the coefficients gets rotated every clock that there is a valid input data. This way, the inputs gets multiplied by the correct coefficient. To get a more detailed description of the data flow inside the MAC and the FIRDecim1 module, refer to appendix.

3.7 Resource Usage

Table 2.6 summarizes the resource usage for different devices for the FIRDecim1 module. Note that the input bit width (IWIDTH) and output bit width (OWIDTH) are 16-bit in this resource usage report.

Table 3.4: Resource Usage for FIRDecim1 Module

Device	XC2V3000-BF957-4	XQR4VFX60-CF1144-10	XC5VFX130F-FF1738-1
Maximum Speed	132MHz	159MHz	180MHz
# of Multipliers	3	3	3
# of Flip Flops	488	488	488
# of LUTs	634	634	494
# of Slices	369	369	175
Power Usage			

3.8 Code Download

Chapter 4

Polyphase Decimation Filter 2

4.1 Theory of Polyphase Decimcation Filter

The purpose of the Polyphase Decimation Filter is to use a Finite Impulse Response Filter (FIR) to decimate the incoming signal by a factor of M.

4.2 Design of Firdecim2

The Polyphase Decimation Filter 2 (FIRDecim2) is a filter with 20 taps which filters the incoming data and decimates the input by a factor of 5. The core of the FIRDecim2 design is the 4 accumulators. The basic idea behind the design for FIRDecim2 is given in figure (4.1).

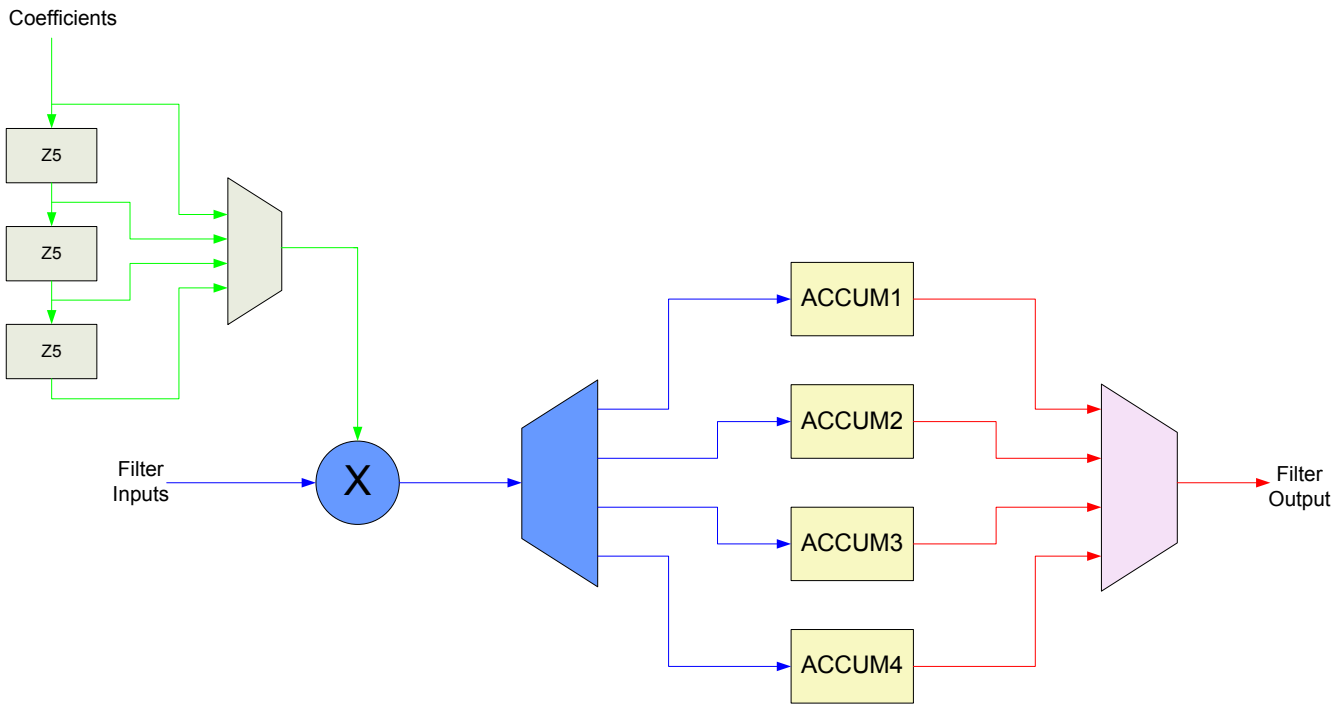

Figure 4.1: FIRDecim2 Design Block Diagram

Notice that the structure for FIRDecim2 is different than that of FIRDecim1. Table 4.2 is a list of the design parameters for FIRDecim2.

Table 4.1: FIRDecim2 Design Parameters

Design Parameter	Requirement
Input clock rate	60 MHz
Input data rate	12 MHz
Output data rate	2.4 MHz
Number of taps	15
Decimation factor	5

The main difference between FIRDecim1 and FIRDecim2 is the ratio of Input clock rate vs. Input data rate. In FIRDecim1, the Input clock rate vs. Input data rate ratio is 1:1. In FIRDecim2, the Input clock rate vs. Input data rate ratio is 5:1. This means that one input is valid for 5 internal clocks. By using this advantage, the design for FIRDecim2 utilized one single multiplier with 4 accumulators.

4.3 Interface

Figure (4.2) shows the inputs and outputs of the FIRDecim2 module.

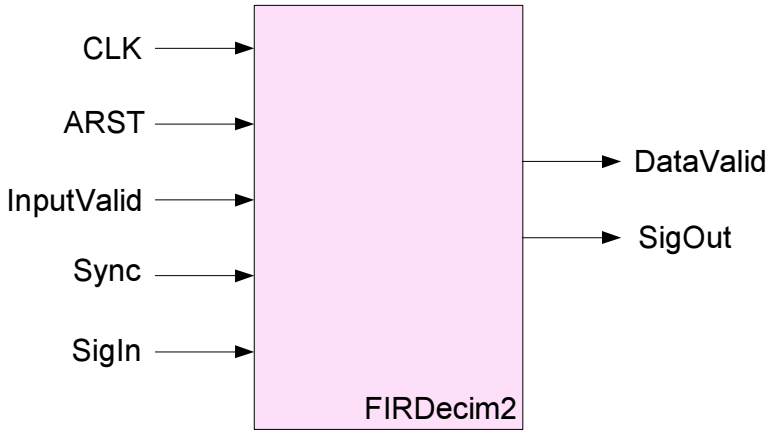

Figure 4.2: FIRDecim2 Inputs and Outputs

Table 4.2: Interface Signals for Firdecim2

Signal	Direction	Size	Active Level	Description
CLK	Input	1-bit	Rising Edge	Clock input value.
ARST	Input	1-bit	High	Asynchronous reset. All module registers are cleared when this signal is high
Sync	Input	1-bit	High	Synchronization to beginning of the look-up tables. All counters are cleared when this signal is high.
InputValid	Input	1-bit	High	Indicate that SigIn is valid.
SigIn	Input	IWIDTH	N/A	Input data.
DataValid	Output	1-bit	High	Indicate that SigOut is valid.
SigOut	Output	OWIDTH	N/A	Output data.

4.4 Parameters

The following is a list of the parameters that can be changed according to user.

e1 - eNTAPS Filter Coefficients (NTAPS number)
OWIDTH Output bit width
IWIDTH Input bit width
AWIDTH Internal bit width
NTAPS Number of taps
ACCUMCNTWIDTH Accumulator counter bit width
CNTWIDTH Counter bit width
NACCUM Number of accumulators

Note that the current Verilog code is written for 20 taps according to algorithm design.

4.5 Multiply-Accumulator

The Multiply-Accumulator in FIRDecim2 is different compared to the MAC in FIRDecim1. The function of the multiplier is to act as a filter with 20 taps. The function of the accumulator is to decimate the input down by 5. The multiplier in FIRDecim2 runs at the internal clock speed, but only gets a new input data every 5 clocks. The advantage of this is the design can use the same input data to multiply with different coefficients every clock. This allows the design to use one single multiplier.

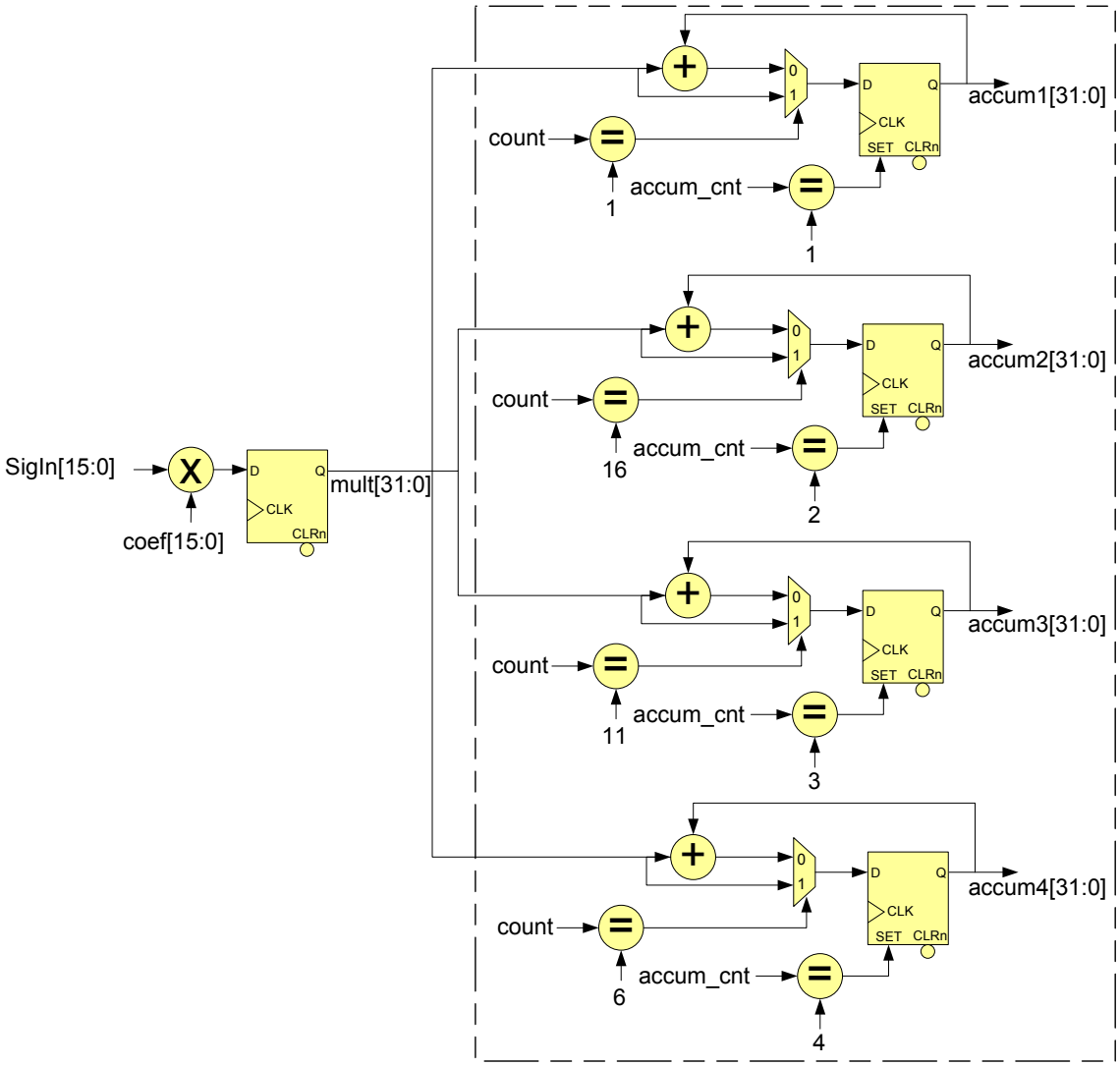

Figure 4.3: FIRDecim2 Multiply Accumulate

4.6 FIRDecim2 Module

The FIRDecim2 block diagram was shown in figure (4.1). Figure (4.4) shows how the output signals are generated with the inputs to the Multiply-Accumulator shown in the previous section.

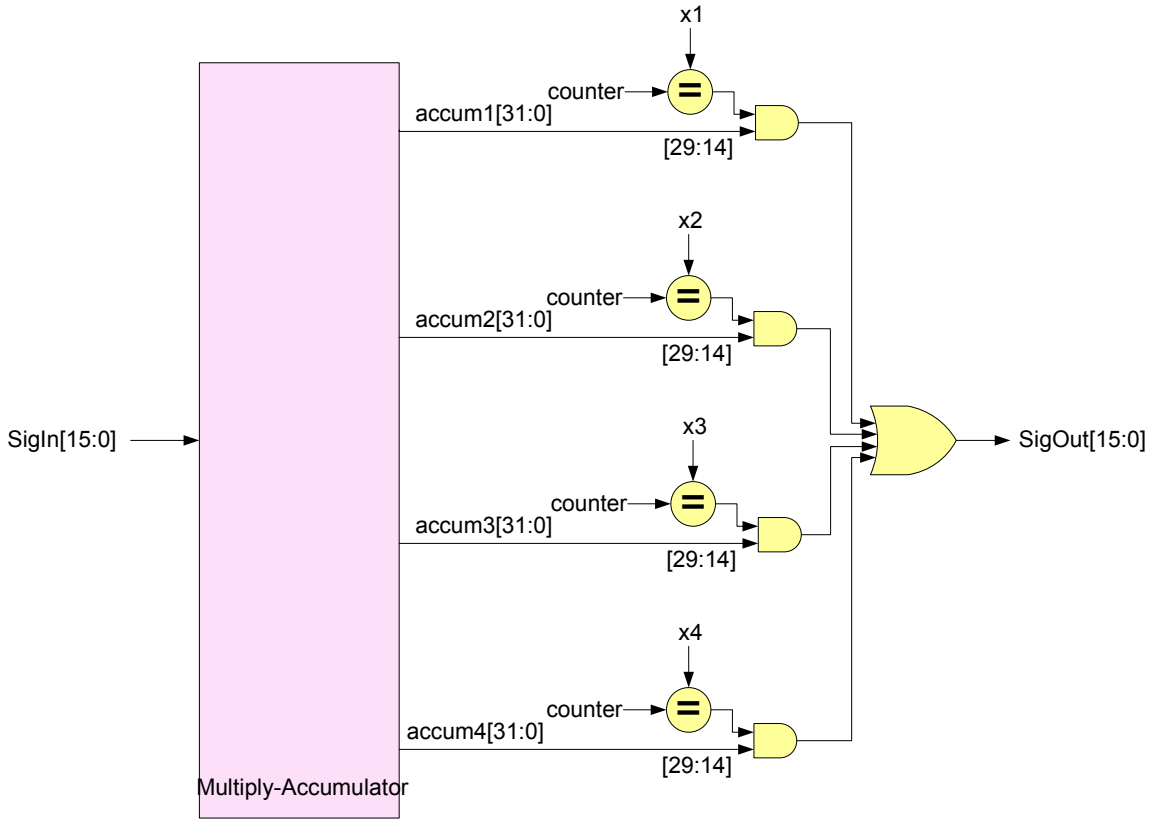

Figure 4.4: FIRDecim2 Logic

4.7 Resource Usage

Table 2.6 summarizes the resource usage for different devices for the FIRDecim2 module. Note that the input bit width (IWIDTH) and output bit width (OWIDTH) are 16-bit in this resource usage report.

Table 4.3: Resource Usage for FIRDecim2 Module

Device	XC2V3000-BF957-4	XQR4VFX60-CF1144-10	XC5VFX130F-FF1738-1
Maximum Speed	100MHz	128MHz	145MHz
# of Multipliers	1	1	1
# of Flip Flops	540	534	538
# of LUTs	749	749	695
# of Slices	402	406	237
Power Usage			

4.8 Code Download

Chapter 5

Polyphase Decimation Filter 3

5.1 Theory of Polyphase Decimcation Filter

The purpose of the Polyphase Decimation Filter is to use a Finite Impulse Response Filter (FIR) to decimate the incoming signal by a factor of M.

5.2 Design of Firdecim3

The Polyphase Decimation Filter 3 (FIRDecim3) is a filter with 50 taps which filters the incoming data and decimates the input by a factor of 2. The core of the FIRDecim3 design is the 25 accumulators. The basic idea behind the design for FIRDecim3 is given in figure (5.1).

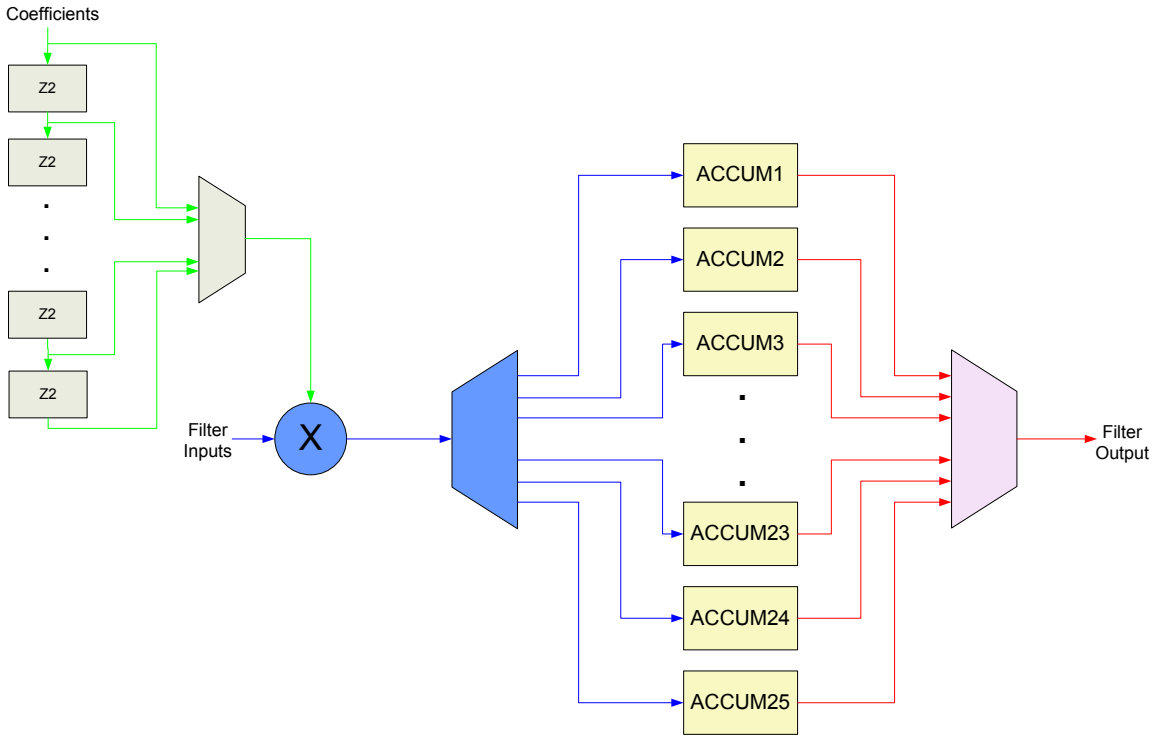

Figure 5.1: FIRDecim3 Design Block Diagram

Notice that the structure for FIRDecim3 is similar to that of FIRDecim2. Table 5.2 is a list of the design parameters for FIRDecim3.

Table 5.1: FIRDecim3 Design Parameters

Design Parameter	Requirement
Input clock rate	60 MHz
Input data rate	2.4 MHz
Output data rate	1.2 MHz
Number of taps	50
Decimation factor	2

Similar to FIRDecim2, the Input Clock rate is faster than the Input Data rate (25:1) for FIRDecim3. This means that one input is valid for 25 internal clocks. By using this advantage, the design for FIRDecim3 utilized one single multiplier with 25 accumulators.

5.3 Interface

Figure (5.2) shows the inputs and outputs of the FIRDecim3 module.

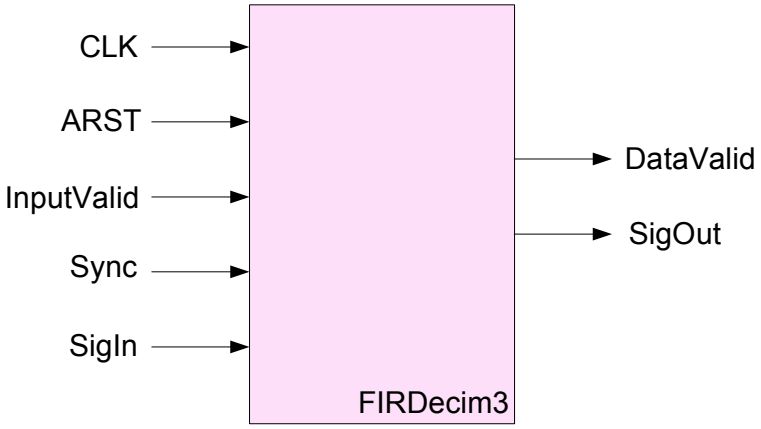

Figure 5.2: FIRDecim3 Inputs and Outputs

Table 5.2: Interface Signals for Firdecim3

Signal	Direction	Size	Active Level	Description
CLK	Input	1-bit	Rising Edge	Clock input value.
ARST	Input	1-bit	High	Asynchronous reset. All module registers are cleared when this signal is high
Sync	Input	1-bit	High	Synchronization to beginning of the look-up tables. All counters are cleared when this signal is high.
InputValid	Input	1-bit	High	Indicate that SigIn is valid.
SigIn	Input	IWIDTH	N/A	Input data.
DataValid	Output	1-bit	High	Indicate that SigOut is valid.
SigOut	Output	OWIDTH	N/A	Output data.

5.4 Parameters

The following is a list of the parameters that can be changed according to user.

e1 - eNTAPS Filter Coefficients (NTAPS number)
OWIDTH Output bit width
IWIDTH Input bit width
AWIDTH Internal bit width
NTAPS Number of taps
ACCUMCNTWIDTH Accumulator counter bit width
CNTWIDTH Counter bit width
NACCUM Number of accumulators

Note that the current Verilog code is written for 50 taps according to algorithm design.

5.5 Multiply-Accumulator

The Multiply-Accumulator for FIRDecim3 is the same as that of FIRDecim2 with the difference of 25 accumulators instead of 4 accumulators. Please refer to section 4.5 for the Multiply-Accumulator details.

5.6 FIRDecim3 Module

FIRDecim3 module is the same as that of FIRDecim2 module with the difference of 25 accumulator results instead of 4 accumulator results. Please refer to section 4.6 for the FIRDecim module details.

5.7 Resource Usage

Table 2.6 summarizes the resource usage for different devices for the FIRDecim3 module. Note that the input bit width (IWIDTH) and output bit width (OWIDTH) are 16-bit in this resource usage report.

Table 5.3: Resource Usage for FIRDecim3 Module

Device	XC2V3000-BF957-4	XQR4VFX60-CF1144-10	XC5VFX130F-FF1738-1
Maximum Speed	101MHz	123MHz	134MHz
# of Multipliers	1	1	1
# of Flip Flops	1,787	1,792	1,785
# of LUTs	3,207	3,197	3,037
# of Slices	1,653	1,674	864
Power Usage			

5.8 Code Download

Part III

APPENDIX

Table 5.4: Stream Representation of FIRDecim1

Input	i1	i2	i3	i4	i5	i6	i7	i8	i9	i10	i11	i12	i13	i14	i15	i16	i17	i18	i19	i20	i21	i22	i23	i24	i25	i26	i27	i28	i29	i30	i31	i32	i33	i34	i35
MAC1	e0	e1	e2	e3	e4	e5	e6	e7	e8	e9	e10	e11	e12	e13	**e14**	*e0*	*e1*	*e2*	*e3*	*e4*	*e5*	*e6*	*e7*	*e8*	*e9*	*e10*	*e11*	*e12*	*e13*	**e14**	e0	e1	e2	e4	e5
MAC2	*e10*	*e11*	*e12*	*e13*	*e14*	e0	e1	e2	e3	e4	e5	e6	e7	e8	e9	e10	e11	e12	e13	**e14**	*e0*	*e1*	*e2*	*e3*	*e4*	*e5*	*e6*	*e7*	*e8*	*e9*	*e10*	*e11*	*e12*	*e13*	**e14**
MAC3	*e5*	*e6*	*e7*	*e8*	*e9*	*e10*	*e11*	*e12*	*e13*	*e14*	e0	e1	e2	e3	e4	e5	e6	e7	e8	e9	e10	e11	e12	e13	**e14**	*e0*	*e1*	*e2*	*e3*	*e4*	*e5*	*e6*	*e7*	*e8*	*e9*
Output					ox1					ox2					o1a					o2a					o3a					o1b					o2b

From table III, i's are inputs, e's are coefficients, o's are valid outputs. Each MAC accumulates for 15 clocks then gets an initialize signal to reset itself. Before it gets reset, it outputs the last valid accumulated value to the FIRDecim1 core. Each MAC stream starts to accumulate 5 clocks apart, and outputs a valid data 5 clocks apart. The following is a brief description of the MAC process. t1 means 1st clock, t2 means 2nd clock, etc.

MAC1:

t1.) accum = e0·i1
t2.) accum = accum + e1·i2
t3.) accum = accum + e2·i3
t4.) accum = accum + e3·i4
⋮ ⋮
t12.) accum = accum + e11·i12
t13.) accum = accum + e12·i13
t14.) accum = accum + e13·i14
t15.) accum = accum + e14·i15, RESET
t16.) accum = (accum = 0) + e0·i16
t17.) accum = accum + e1·i17
⋮ ⋮
t29.) accum = accum + e13·i29
t30.) accum = accum + e14·i30, RESET
⋮ ⋮

MAC2:

t1.) accum = x·i1
t2.) accum = accum + x·i2
⋮ ⋮
t4.) accum = accum + x·i4
t5.) accum = accum + e14·i5, RESET
t6.) accum = (accum = 0) + e0·i6
t7.) accum = accum + e1·i7
⋮ ⋮
t19.) accum = accum + e13·i19
t20.) accum = accum + e14·i20, RESET
t21.) accum = (accum = 0) + e0·i21
t22.) accum = accum + e1·i22
⋮ ⋮
t34.) accum = accum + e13·i34
t35.) accum = accum + e14·i35, RESET
⋮ ⋮

MAC3:

t1.) accum = x·i1
t2.) accum = accum + x·i2
t3.) accum = accum + x·i3
⋮ ⋮
t8.) accum = accum + x·i8
t9.) accum = accum + x·i9
t10.) accum = accum + e14·i10, RESET
t11.) accum = (accum = 0) + e0·i11
t12.) accum = accum + e1·i12
⋮ ⋮
t24.) accum = accum + e13·i24
t25.) accum = accum + e14·i25, RESET
⋮ ⋮

From the above description, a valid output (in red) happens every 5 clocks on time (t5, t10, t15, t20, t25, t30, etc...). The outputs from time t5 and t10 are invalid outputs and should not be considered. Also after the MAC outputs a valid data, its accumulator gets reset to 0 and starts accumulating from the next clock with initial value 0.