1300 Henley Court
Pullman, WA 99163
509.334.6306
www.digilentinc.com

Nexys4™ PDM Filter Project

Revised February 3, 2014
This manual applies to the Nexys4 rev. B

Introduction

This document aims to briefly explain the pulse-density modulation (PDM) [Ref. 2] interfacing design used to acquire data from the Nexys4's on-board ADMP421 MEMS microphone [Ref. 1]. This design is a hardware Xilinx Artix-7 FPGA implementation of the cascaded integrator-comb (CIC) decimation filter [Ref. 9] and finite impulse response (FIR) filter [Ref. 20], outputting the 16-bit filtered data at a 96 kHz rate.

This project was compiled with Xilinx ISE 14.6 and written in VHDL.

1 Functional Description and Implementation Details

In order to complete the sigma-delta conversion [Ref. 3] [Ref. 4], the filtering stage has been implemented using a multi-stage decimation strategy with one cascaded integrator-comb decimating filter, one half-band FIR filter, one low-pass FIR filter, and a first-order high-pass filter, as shown in Figure 1.

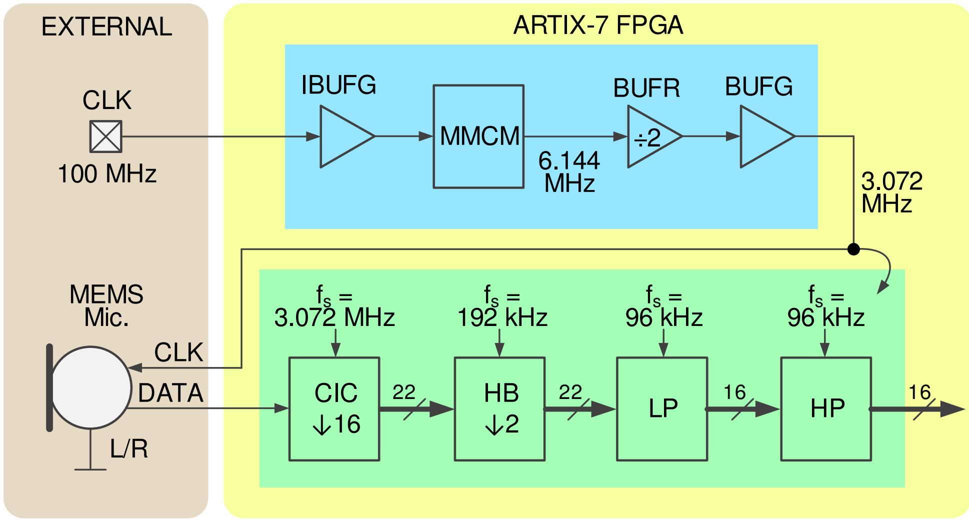

Figure 1. *PDM filter block diagram.*

The clock source to drive the PDM is generated by a mixed-mode clock manager proceeded by a regional primitive buffer that further divides the 6.144 MHz clock, making it a 3.072 MHz clock. The L/R select input of the microphone is tied to the ground, choosing the data to be sampled at the rising edge of Clk.

1.1 The CIC Stage

This is the first filtering stage of the PDM filter. It is implemented using Xilinx LogiCORE CIC Compiler [Ref. 11] [Ref. 12] [Ref. 13]. Table 1 shows the parameters used in the current filter configuration:

Parameter Name	Value	Observations
Filter Type	Decimation	
Number of Stages	5	Calculated with equation 3.5 of "*Design and VLSI Implementation of a Decimation filter for Hearing Aid Applications*" [Ref. 10], and knowing that the microphone [Ref. 1] has a fourth-order built-in sigma-delta modulator.
Differential Delay	1	Number of unit delays employed in each comb filter [Ref. 13].
Number of Channels	1	
Sample Rate Change	16	The actual decimation rate.
Hardware Oversampling Format	Frequency Specification	The input sampling frequency is used for specifying the hardware oversampling rate.
Input Sampling Frequency	3.072 MHz	The rate of the microphone output PDM data.
Clock Frequency	3.072 MHz	
Input Data Width	2	The minimum value that is achieved by converting the unipolar PDM data to bipolar (the LSB is always '1' and bit 1 is the PDM data).
Quantization	Full Precision	Outputs the full 22-bit data without any truncation or rounding.
Use XtremeDSP Slice	Yes	Use dedicated hardware DSP primitive.

Table 1. *CIC compiler settings*.

The CIC stage samples the input data at 3.072 MHz, with a resulting decimation rate of 16. The output sampling rate is 192 kHz.

1.2 The Half-Band Filter

Parameter	Value
FIR Order	14
F_s (Sampling frequency)	192 kHz
F_{pass} (Cut-off frequency)	21.8 kHz
A_{stop} (Stop-band attenuation)	80 dB
Coefficient width[1]	15

Table 2. *Half-band filter settings.*

The second filtering stage is the half-band decimation filter [Ref. 16] [Ref. 17] that was implemented with the Xilinx LogiCORE FIR Compiler [Ref. 21] [Ref. 22]. The filter coefficients were calculated using MATLAB, but they can also be calculated with any other software (e.g. Scilab, Octave, etc.) and stored in a *.coe* file according to the

[1] 15-bit wide coefficients were chosen to spread across the full 80 dBs of magnitude band.

specifications in Xilinx's "*LogiCORE IP FIR Compiler v6.3*" [Ref. 21]. The filter settings for MATLAB are displayed in Table 2, along with a visual representation of the filter's magnitude response (Figure 2):

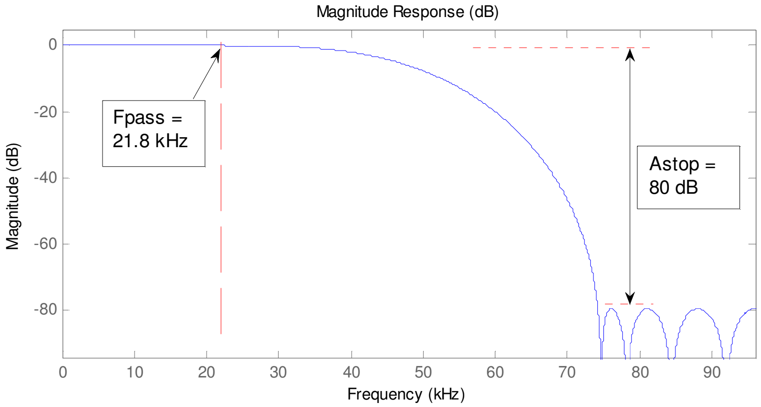

Figure 2. *Half-band filter settings*.

Having the coefficients calculated, the following settings were put in place for the FIR Compiler (Table 3):

Parameter Name	Value	Observations
Filter Type	Decimation	Decimation was selected at the Filter Type option.
Rate Change Type	Integer	Decimation rate value type
Decimation Rate Value	2	A decimation of 2 was used to achieve the final sampling rate of 96 kHz.
Hardware Oversampling Format	Frequency Specification	The input sampling frequency is used for specifying the hardware oversampling rate.
Input Sampling Frequency	0.192 MHz	The output data rate of the previous filtering stage (192 kHz).
Clock Frequency	3.072 MHz	
Quantization	Integer Coefficients	The integer coefficients quantization option is used to let the FIR Compiler analyze the coefficients and determine the minimum number of bits required for representation.
Coefficient Width	15	
Coefficient Structure	Half-band	The type of filter according to the coefficient structure [Ref. 22].
Input Data Type	Signed	The output data of the CIC stage is signed.
Input Data Width	22	
Output Rounding Mode	Truncated LSBs	Using Equations 3-4 from "*Digitally Removing a DC Offset: DSP Without Mathematics*" [Ref. 22] would give an output data width too large for this filter's purpose; hence, a truncation scheme for the output data has been used.
Output Width	22	Output signed data width truncated to fit a 22-bit bus.

Table 3. *FIR compiler settings for half-band filter*.

1.3 The Low-Pass Filter

This filtering stage was implemented using the Xilinx LogiCORE FIR Compiler [Ref. 21][Ref. 22] by having the coefficients calculated using the values in Table 4 (and Figure 3):

Parameter	Value
FIR Order	161
F_s (Sampling frequency)	96 kHz
F_{pass} (end of pass-band)	13 kHz
F_{stop} (beginning of stop-band)	15 kHz
W_{pass} (weight in pass-band)	1 dB
W_{stop} (weight in stop-band)	100 dB
Coefficient width[2]	20

Table 4. *Low-pass filter settings.*

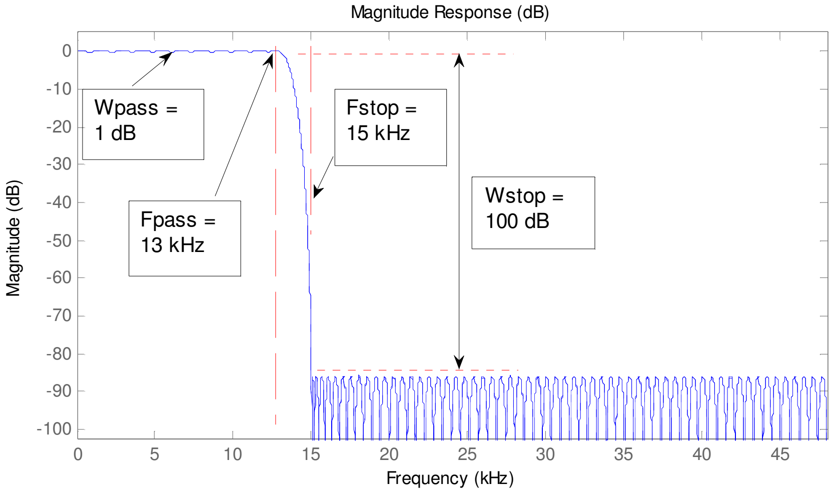

Figure 3. *Low-pass FIR filter settings.*

In order to properly configure the FIR Compiler, a description of its settings are shown in Table 5:

[2] 20-bit wide coefficients were chosen to spread across the full 100 dBs of magnitude band.

Parameter Name	Value	Observation
Filter Type	Single rate	Since the decimation ratio is 1, single rate filter type was selected.
Hardware Oversampling Format	Frequency Specification	The input sampling frequency is used to specify the hardware oversampling rate.
Input Sampling Frequency	0.096 MHz	The output data rate of the previous filtering stage (96 kHz).
Clock Frequency	3.072 MHz	
Coefficient Type	Signed	Signed coefficients with two's complement representation.
Coefficient Width	20	
Coefficient Structure	Symmetric	See Chapter 3 of "*LogiCORE IP FIR Compiler v7.0*" [Ref. 22].
Input Data Type	Signed	The output data of the CIC stage is signed.
Input Data Width	22	
Output Rounding Mode	Truncate LSBs	Using Equations 3-4 from "*LogiCORE IP FIR Compiler v7.0*" [Ref. 22] would give an output data width too large for this filter's purpose; thus, a truncation scheme for the output data has been used.
Output Width	22	Output signed data width truncated to fit a 22-bit bus.

Table 5. *FIR Compiler settings for low-pass filter.*

1.3 The High-Pass Filter

This represents the final filtering stage, used for removing any DC component induced by the sigma-delta modulator or the microphone itself (offset error). It is an implementation of digitally removing an offset [Ref. 24], but other methods can also be implemented, as described in "*DC Blocker Algorithms*" [Ref. 23] and in "*DSP Tricks: DC Removal*" [Ref. 25].

After sign extending the input data (16-bit to 17-bit), it is subtracted from the most significant result (17-bit) of the 1-tap integrator (the integral part of the integration [Ref. 24]). Using a multiplier value of 2^{-12} and one delay cell (z^{-1}) results in a cut-off frequency of 18.6 Hz with a 10 dB/decade slope.

Figure 4 shows the filter diagram.

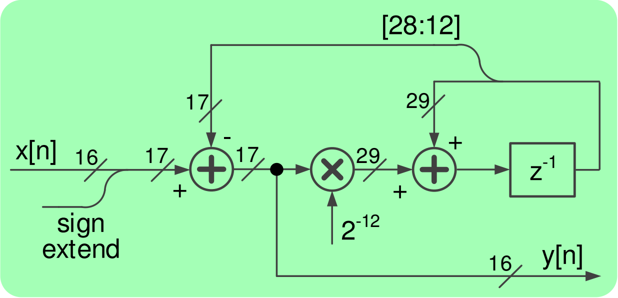

Figure 4. *High-pass filter schematic.*

2 Area and Performance

The actual, measured magnitude and phase responses of the resulting PDM filter are shown in Figure 5. Figure 6 shows the approximated[3] impulse response.

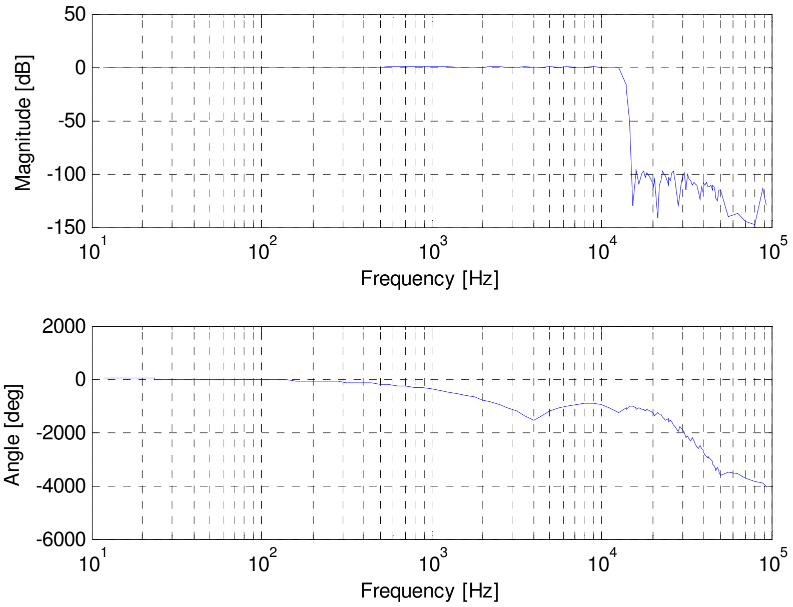

Figure 5. *Magnitude and phase response of the PDM filter.*

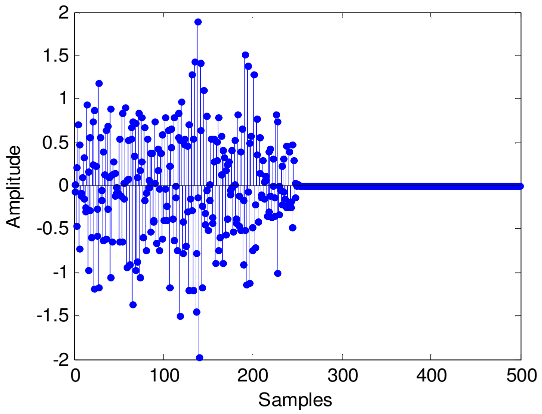

Figure 6. *Impulse response of the PDM filter.*

[3] In order to convert the acquired frequency response data to discrete-time filter numerator and denominator z polynomials (its transfer function), a Gauss-Newton iterative search of linear equations was used (function *invfreqz* in MATLAB) with an approximation error on the result as follows: magnitude response – 1%, phase response – 1.7%.

3 Design Usage Example: Schroeder Reverberator

The Schroeder reverberator [Ref. 28] was used as a demo to prove the PDM filter's utility. The following figure illustrates the block diagram of the reverberator and its components:

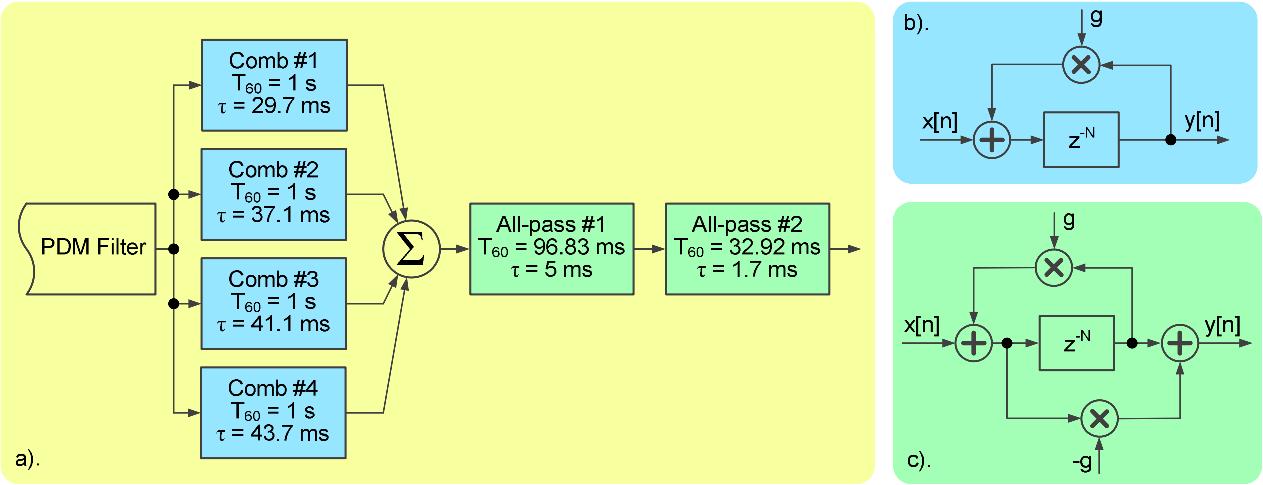

Figure 7. (*a) The block diagram of the Schroeder reverberator, (b) the structure of the comb filter, (c) and the structure of the all-pass filter.*

The output of the PDM filter is fed into four parallel comb filters, each with a different loop time to enhance the acoustic resonance. After this, the samples are fed into two consecutive all-pass filters to further enlarge the spectrum.

In all structures (comb and all-pass), the *g* multiplier constant and *N* number of delay stages are computed automatically in the VHDL module by using the following equations [Ref. 27]:

$$g = 10^{\frac{-3\tau}{T_{60}}}$$

$$N = \tau f_s$$

τ = loop time
T_{60} = reverb time
f_s = sampling frequency

References

[1] Analog Devices: "*Omnidirectional Microphone with Bottom Port and Digital Output*", ADMP421 Data Sheet, rev. D.

[2] Wikipedia: "*Pulse-density modulation*", http://en.wikipedia.org/wiki/Pulse-density_modulation, March 2013.

[3] U. Beis: *"An Introduction to Delta Sigma Converters"*, http://www.beis.de/Elektronik/DeltaSigma/DeltaSigma.html, 2008.

[4] J. D. Reiss: "*Understanding Sigma-Delta Modulation: The Solved and Unsolved Issues*", J. Audio Eng. Soc., Vol. 56, No. 1/2, January/February 2008.

[5] Z. Milivojević: *"Digital Filter Design"*, http://www.mikroe.com/products/view/268/digital-filter-design/.

[6] N. Hegde: "*Seamlessly Interfacing MEMS Microphones with Blackfin Processors*", EE-350 Engineer-to-Engineer Note, rev. 1, August 2010.

[7] Altera, Inc.: "*Understanding CIC Compensation Filters*", Application Note 455, ver. 1.0, April 2007.

[8] R. G. Lyons: "*Understanding cascaded integrator-comb filters*", http://www.embedded.com/design/configurable-systems/4006446/Understanding-cascaded-integrator-comb-filters, March 2005.

[9] M. P. Donadio: "*CIC Filter Introduction*", Free Publication by Iowegian, July 2000.

[10] S. Pandu: "*Design and VLSI Implementation of a Decimation filter for Hearing Aid Applications*", Master of Technology thesis, 2007.

[11] Xilinx, Inc.: "*LogiCORE CIC Compiler v3.0*", DS845 Product Specification, ver. 1.0, June 2011.

[12] Xilinx, Inc.: "*Cascaded Integrator-Comb (CIC) Filter v3.0*", Product Specification, March 2002.

[13] Xilinx, Inc.: "*LogiCORE CIC Compiler v4.0*", PG140 Product Guide for Vivado Design Suite, ver. 1.0, March 2013.

[14] G. J. Dolecek, J. D. Carmona: "*On Design of CIC Decimators*", Applications of MATLAB in Science and Engineering, ISBN: 978-953-307-708-6, pp. 225-246, 2011.

[15] A. C. Cherik, E. Farshidi: "*A new Configurable Decimation Filter using Pascal's Triangle Theorem*", World Academy of Science, Engineering and Technology 54, pp. 75-78, 2011.

[16] H. G. Göckler: "*Most Efficient Digital Filter Structures: The Potential of Halfband Filters in Digital Signal Processing*", Applications of Digital Signal Processing, ISBN: 978-953-307-406-1, pp. 237-278, 2011.

[17] T. Saramaki, T. Karema, T. Ritoniemi, H. Tenhunen: "*Multiplier-free Decimator Algorithms for Superresolution Oversampled Converters*", in Proc. 1990 IEEE International Symposium on Circuits and Systems (New Orleans, Louisiana), pp. 3275-3278, May 1990.

[18] P. Zahradnik, B. Simak, M. Kopp, M. Vlcek: "*Design of Half-Band FIR Filters for Signal Compression*", International Journal on Advances in Telecommunications, vol. 4, no. 3 & 4, 2011.

[19] P. P. Vaidyanathan, T. Q. Nguyen: “*A «TRICK» for the Design of FIR Half-Band Filters*”, IEEE Transactions on Circuits and Systems, vol. CAS-32, no. 3, March 1987.

[20] R. Thakur, K. Khare: “*High Speed FPGA Implementation of FIR Filter for DSP Applications*”, International Journal of Modeling and Optimization, Vol. 3, No. 1, February 2013.

[21] Xilinx, Inc.: “*LogiCORE IP FIR Compiler v6.3*”, DS795 Product Specification, ver. 1.3, October 2011.

[22] Xilinx, Inc.: “*LogiCORE IP FIR Compiler v7.0*”, PG149 Product Guide for Vivado Design Suite, ver. 1.0, March 2013.

[23] R. Yates, R. Lyons: “*DC Blocker Algorithms*”, IEEE Signal Processing Magazine, pp. 132-134, March 2008.

[24] K. Chapman: “*Digitally Removing a DC Offset: DSP Without Mathematics*”, WP279 Xilinx White Paper, ver. 1.0, July 2008.

[25] R. G. Lyons: “*DSP Tricks: DC Removal*”, http://www.embedded.com/design/configurable-systems/4007653/DSP-Tricks-DC-Removal, August 2008.

[26] E. Doering: “*Reverberation*”, http://cnx.org/content/m15471/latest/, March 2008.

[27] E. Doering: “*Schroeder Reverberator*”, http://cnx.org/content/m15491/1.2/, March 2008.

[28] J. Sun: “*Schroeder’s Reverberator: The Earliest Digital Solution of Sound Reverberation*”, EECS 195 Final Project, March 2005.

Appendix A: Half-Band Filter Coefficients

The following coefficients are used in the Xilinx FIR Compiler to implement the half-band filer:

```
radix = 10;

coefdata = -100, 0, 614, 0, -2295, 0, 9971, 16383, 9971, 0, -2295, 0, 614,
0,-100;
```

Appendix B: Low-Pass Filter Coefficients

The following coefficients are used in the Xilinx FIR Compiler to implement the low-pass filter:

```
radix = 16;

coefficient_width = 20;

coefdata = 00041, 00111, 002c8, 0056d, 0086d, 00a86, 00a33, 0067c, fffc5,
           ff81b, ff282, ff179, ff573, ffc43, 0020f, 00380, fffdd, ff9ae,
           ff54a, ff600, ffbb8, 002c4, 0063b, 0036e, ffc13, ff567, ff49a,
           ffaeb, 00457, 00a1b, 0078f, ffde6, ff3dd, ff12c, ff8b1, 005c2,
           00ef5, 00ce1, 00003, ff133, febb2, ff4be, 006e7, 01520, 0140d,
           002f3, fed69, fe3ad, fee8d, 007ce, 01d64, 01e2d, 0074f, fe801,
           fd7b5, fe4d3, 00880, 02984, 02dc2, 00e6b, fdfa3, fc42b, fd42d,
           00902, 03e33, 049ba, 01bf8, fcff2, f9d13, fb0cb, 00958, 06e10,
           08fee, 0416c, fa175, f18e6, f24d3, 00982, 19cd6, 34c97, 46294,
           46294, 34c97, 19cd6, 00982, f24d3, f18e6, fa175, 0416c, 08fee,
           06e10, 00958, fb0cb, f9d13, fcff2, 01bf8, 049ba, 03e33, 00902,
           fd42d, fc42b, fdfa3, 00e6b, 02dc2, 02984, 00880, fe4d3, fd7b5,
           fe801, 0074f, 01e2d, 01d64, 007ce, fee8d, fe3ad, fed69, 002f3,
           0140d, 01520, 006e7, ff4be, febb2, ff133, 00003, 00ce1, 00ef5,
           005c2, ff8b1, ff12c, ff3dd, ffde6, 0078f, 00a1b, 00457, ffaeb,
           ff49a, ff567, ffc13, 0036e, 0063b, 002c4, ffbb8, ff600, ff54a,
           ff9ae, fffdd, 00380, 0020f, ffc43, ff573, ff179, ff282, ff81b,
           fffc5, 0067c, 00a33, 00a86, 0086d, 0056d, 002c8, 00111, 00041;
```

Appendix C: Overall FPGA Usage

Slice Logic Utilization	Usage	Total	Percentage
Number of Slice Registers	951	126,800	1
- Number used as Flip Flops	951		
- Number used as Latches	0		
- Number used as Latch-thrus	0		
- Number used as AND/OR logics	0		
Number of Slice LUTs	693	63,400	1
- Number used as logic	325	63,400	1
- Number using O6 output only	235		
- Number using O5 output only	0		
- Number using O5 and O6	90		
- Number used as ROM	0		
- Number used as Memory	336	19,000	1
- Number used as Dual Port RAM	48		
- Number using O6 output only	8		
- Number using O5 output only	0		
- Number using O5 and O6	40		
- Number used as Single Port RAM	0		
- Number used as Shift Register	288		
- Number using O6 output only	247		
- Number using O5 output only	0		
- Number using O5 and O6	41		
- Number used exclusively as route-thrus	32		
- Number with same-slice register load	32		
- Number with same-slice carry load	0		
- Number with other load	0		
Number of occupied Slices	311	15,850	1
Number of LUT Flip Flop pairs used	895		
- Number with an unused Flip Flop	106	895	11
- Number with an unused LUT	202	895	22
- Number of fully used LUT-FF pairs	587	895	65
- Number of slice register sites lost to control set restrictions	0	126,800	0

I/O Utilization	Usage	Total	Percentage
Number of bonded IOBs	14	210	6
- Number of LOCed IOBs	14	14	100
- IOB Flip Flops	0		

Specific Feature Utilization	Usage	Total	Percentage
Number of RAMB36E1/FIFO36E1s	0	135	0
- Number using RAMB36E1 only	0		
- Number using FIFO36E1 only	0		
Number of RAMB18E1/FIFO18E1s	0	270	0
Number of BUFG/BUFGCTRLs	4	32	12

- Number used as BUFGs	4		
- Number used as BUFGCTRLs	0		
Number of IDELAYE2/IDELAYE2_FINEDELAYs	0	300	0
Number of ILOGICE2/ILOGICE3/ISERDESE2s	0	300	0
- Number used as ILOGICE2s	0		
Number used as ILOGICE3s	0		
- Number used as ISERDESE2s	0		
Number of ODELAYE2/ODELAYE2_FINEDELAYs	0		
Number of OLOGICE2/OLOGICE3/OSERDESE2s	0	300	0
- Number used as OLOGICE2s	0		
- Number used as OLOGICE3s	0		
- Number used as OSERDESE2s	0		
Number of PHASER_IN/PHASER_IN_PHYs	0	24	0
Number of PHASER_OUT/PHASER_OUT_PHYs	0	24	0
Number of BSCANs	0	4	0
Number of BUFHCEs	0	96	0
Number of BUFRs	1	24	4
Number of CAPTUREs	0	1	0
Number of DNA_PORTs	0	1	0
Number of DSP48E1s	16	240	6
Number of EFUSE_USRs	0	1	0
Number of FRAME_ECCs	0	1	0
Number of IBUFDS_GTE2s	0	4	0
Number of ICAPs	0	2	0
Number of IDELAYCTRLs	0	6	0
Number of IN_FIFOs	0	24	0
Number of MMCME2_ADVs	1	6	16
Number of OUT_FIFOs	0	24	0
Number of PCIE_2_1s	0	1	0
Number of PHASER_REFs	0	6	0
Number of PHY_CONTROLs	0	6	0
Number of PLLE2_ADVs	0	6	0
Number of STARTUPs	0	1	0
Number of XADCs	0	1	0