Single-chip High-Density FPGA Implementation of the Synthetic Aperture Radar Azimuth Pre-filter for On-board Data Reduction

MiMi Aung Gudim, Frank Cheng, Soren Madsen, Robert A. Johnson, Charles Le, Mahta Moghaddam, Miguel Marina
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA

Abstract

Synthetic Aperture Radar (SAR) missions planned for the near future are pushing downlink data bandwidths to prohibitive levels. Successful reduction in data volume will provide many benefits for the future missions. In this paper, we describe a SAR on-board data-reduction scheme which takes advantage of data reduction in the azimuth domain while introducing no degradation in the range domain. This scheme, named the Azimuth Pre-filter (AzPF), has been implemented on a single chip with low-mass and low-power consumption by utilizing high-density Filed Programmable Gate Array (FPGA) technology.

I. Introduction

The AzPF has been successfully implemented on a single two million gate XILINX FPGA (XCV2000E-6) without any additional external chips. The filter first demodulates offset video signal in each input range line, and then performs the azimuth pre-filtering on the I/Q channels' range lines in the azimuth direction. The filter coefficients and the decimation factors are programmable. Thus the AzPF filter can be applicable to many missions which can afford resolution reduction in the azimuth domain. Data reduction factor is selectable (1, 2, 4, 8, 16, 32) to allow for different reduction levels required for different periods within a mission. The benefits of the AzPF-based on-board data reduction are as follows.

- The rapid access to data is especially required for emergency situations (floods, earthquakes, etc.), and is highly desired by operational agencies.
- Azimuth pre-filtering will reduce data volume by a selectable factor, enabling data downlink at lower data rates. This allows distributed ground receiving stations operated by groups of scientists or other user communities for fast access and analysis at lower-resolution from which they can quickly evaluate whether there is a need for the higher-resolution data already collected. If so, they will request such data. If not, the full-resolution data stored on-board or partially downlinked will be erased.
- This scenario is highly desired for large area mappings (e.g., Amazon basin, arctic circle, earthquake-prone areas), where lower-resolution data are routinely required on a large scale, and occasionally there is a need to study various sites at a high resolution to assess impact of dynamic events such as floods, fires, logging, earthquakes, and freeze/thaw events (arctic).

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

The AzPF filter design, performance evaluation, software modeling, FPGA design and challenges, and performance verification of the hardware implementation are described in this paper.

II. Functional requirements of the AzPF

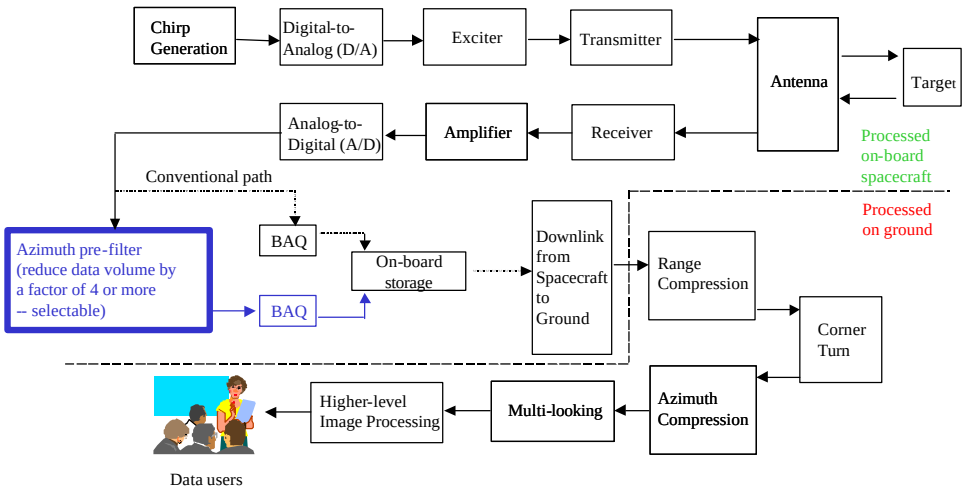
The major steps involved in SAR processing are shown in Figure 1. On board the spacecraft, the received radar echoes are digitized in the analog-to-digital convertor and the conventional SAR design is to quantize and transmit the digital samples to the ground. The key processing steps which usually occur on the ground after this 'raw' high-rate data downlink are: azimuth compression; corner turning; range compression; and multi-looking.

Parameters for a typical spaceborne SAR are given in Table 1.

Table 1: Typical parameters for a spaceborne SAR

Parameter	Value
Azimuth Resolution	6 m
Range Resolution	25 m
Swath width	100 km
# of bits per downlinked sample	4
Downlink data rate	105 Mbps

The data rate required for the downlink to the ground is often ~105 Mbps. There are several NASA mission (EOS-8, EX-6, EX-7) planned for beyond 2000 which have SAR sensors that do not require the 6 m azimuth resolution in all data acquisition scenarios. For these missions that can afford a relaxed azimuth resolution, AzPF can significantly reduce required downlink bandwidth by a decimation factor of *M*. The cost of reducing the required downlink bandwidth by a factor *M* is simply the degradation of azimuth resolution by the same factor, *M*. Figure 2 shows the significant reduction in the required downlink data rates if the azimuth resolution can be relaxed from the best case 6 m to up to 1



km. Tables 2 and 3 show the parameters and the performance requirements of the AzPF proposed for these missions.

Fig. 1 : Simplified Radar Block Diagram

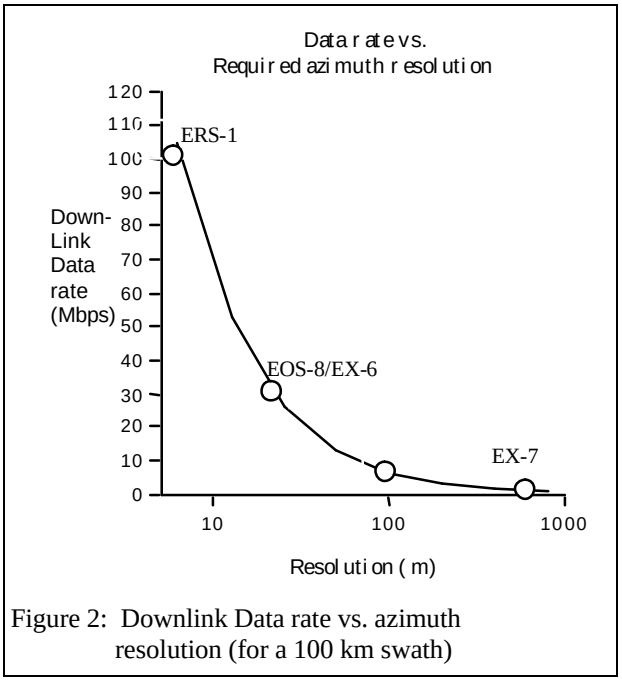


Figure 2: Downlink Data rate vs. azimuth resolution (for a 100 km swath)

Table 2: AzPF processor parameters

Parameter	Value
Input data type	off-set video
ADC sample rate	100 MHz
# bits/input sample	8
# samples/range line	8k
Pulse repetition rate	2000 Hz
AzPF data reduc. Factor	M (selectable)
# of bits/output sample	12

III. Functional description of the AzPF

The AzPF[1] is composed of:

- A digital I/Q demodulator, which will take the off-set video range echo to the complex baseband domain. This operation will ensure that the signal phase is preserved through azimuth pre-filtering.
- A poly-phase spatial domain pre-filter. An azimuth filter with 4 overlapping phases is applied.

A block diagram of the AzPF is shown in Figure 5.

Table 3: AzPF performance requirements

Parameter	Value
Range resolution	No degradation
Azimuth resolution	1/M of the single look resolution
Peak to Side-Lobe Ratio (PSLR)	-25 dB
Integrated Side-Lobe Ratio (ISLR)	-15 dB

Digital I/Q demodulator

The received analog off-set video signal can be represented as :

$g_{IF}(t) = a(t)\cos(2pf_{IF}t + f(t))$.

Generally, the intermediate frequency, f_{IF}, is chosen to be a quarter of the range sampling frequency, *fs*, and the Pulse Repetition Frequency(PRF) is chosen to be a multiple of f_{IF}. Each sample is indexed by a pulse line number, *m*, and a range sample number, *n*:

$g_{IF}(n,m) = a(n,m)\cos(np\ /2 + f(n,m))$.

An I/Q demodulator conceptually consists of a complex

down-conversion of this signal followed by low-pass filtering which eliminates the undesired sidebands. The complex down-conversion is given by:

$$
\begin{aligned}
g_{bb}(n,m) &= g_{IF}(n,m)\exp\left\{-j\frac{p}{2}n\right\} \\
&= \frac{a(n,m)}{2}\left(\exp\{jf(n,m)\}+(-1)^n\exp\{-f(n,m)\}\right) \\
&= \frac{a(n,m)}{2}\left(\left(1+(-1)^n\right)\cos\{jf(n,m)\}+j\left(1-(-1)^n\right)\sin\{jf(n,m)\}\right) \\
&= \begin{cases} a(n,m)\cos\{jf(n,m)\}; & \text{for } n \text{ even} \\ j\,a(n,m)\sin\{jf(n,m)\}; & \text{for } n \text{ uneven} \end{cases}
\end{aligned}
$$

A half-band filter is applied to both I and Q channels. Because all I-channel odd samples are zero and all even samples, except for sample zero, of the filter are zero, only a delay which will align the I-channel with the filter delay of the Q-channel is required for the low-pass filtering. In the Q-channel, the odd samples are different from zero. Thus, the filter needs to process all the odd samples.

Poly-phase pre-filtering

The AzPF architecture is based on the poly-phase filter approach. More simply the filter architecture is that of a number of overlapping and weighted integrate and dump filters. To avoid signal aliasing, the bandwidth of AzPF is 80% of PRF/*M* (*M*=decimation factor). Reference [2] has shown that a filter of length 4*M* can provide an acceptable transfer function characteristics. Thus a poly-phase filter with 4 phases is used to implement the AzPF with programmable decimation factors.

IV. AzPF algorithm study

Three approaches have been used for the filter study [2] :

- Filter design by using Hamming window.
- Filter design by using Remez multiple exchange algorithm to minimize the maximum error between the given desired and the approximating responses.
- Filter design by using Lagrange multiplier to maximally flatten the pass band.

The study for the filter coefficient quantization effect is shown in Figure 3 and the performance with respect to the decimation ratio is shown in Figure 4. It is clear from Figure 3 that the performance improvement after 8-bit quantization is insignificant. Depending on the trade off among resolution, PSR, and ISLR, the appropriate filter can be chosen based on Figure 4.

V. AzPF FPGA design

The block diagram in Figure 5 illustrates data paths of the AzPF. They perform the signal processing functions described in section III. The design is entirely synchronous, using clock enable ports on data path components such as registers, counters, and RAM, to enable a positive clock edge of the 100 MHz system clock at an appropriate instance. The design takes advantage of the 420 μsec interval between transmitter pulses to effectively reduce the clock rate, via clock enable ports, to 12.5 MHz. This allows more levels of combinatorial logic between registers.

The AzPF design consists of a translation of the block diagram of Figure 5 into a Verilog model consistent with the capabilities of the Xilinx Foundation ISE 3.3 tool suite and the target FPGA. The majority of the data path portion of the model amounted to an interconnection of Xilinx "coregen IP" components.

Design flow from Verilog modeling to hardware testing is illustrated in the Figure 6. The "tools" are PC based and consist of the Xilinx Foundation ISE 3.3 suite and Mentor Graphics ModelSim 5.5e. On-chip in-circuit debugging is facilitated using the Xilinx ChipScope Integrated Logic Analyzer via the MultiLinx cable.

The target FPGA was chosen to be the XCV2000 on the basis of on-chip memory requirements for the AzPF (>500 kbits). The two million gates in this high-density FPGA and sufficient on-chip memory allowed for the AzPF to be implemented on a single chip, resulting in a low-mass, low-power (<1 Watt) implementation shown in Figure 8. The prototype board shown is 6” by 10”, however the chip is only ~2.5” by 2.5”, i.e. size of the hardware implementation can be significantly reduced for flight.

VI. AzPF Performance Evaluation and Hardware Verification

Performance evaluation and hardware verification were performed on the AzPF for decimation factor $M = 4$ and AzPF design coefficients selected using the Hamming window. The following software (S/W) models were developed for this purpose:

- A 32-bit floating point AzPF behavioral implemented in C language
- A fixed-point AzPF behavioral, corresponding to the FPGA implementation bit-width limitations, in C language
- A hardware behavior model implemented in the hardware description language, Verilog.

Performance Evaluation

To evaluate the AzPF algorithm performance, a 2-D simulated point-target SAR raw data set was first

generated. It contains 1K range lines and each range line has 225 samples. Chirp bandwidth is 45 MHz and chirp length is 5 μsec. This data set was processed by the 32-bit floating point AzPF S/W model. The azimuth pre-filtered data was then processed by a standard SAR processor to generate a 2-D point target. The resolution, PSR, and ISLR of the output were measured and compared versus those of the original point target, and to the AzPF requirements. Results of the comparison are shown in Table 4. Compared to the case of no azimuth pre-filtering, the resolution, PSR and ISLR are degraded as expected but still meet the AzPF requirements.

Next, to evaluate the quantization effect (due to finite bit-widths in the FPGA implementation) on the image quality, the above performance evaluation was repeated using the fixed-point AzPF S/W simulator. Results are shown in Figure 5, showing that the AzPF fixed bit-width implementation performance is slightly degraded compared to the cases of no pre-filtering and pre-filtering model with 32-bit as expected, but still meet the AzPF requirement.

After verifying that the fixed point performance meets the AzPF requirements, the finite bit-width AzPF was implemented in the hardware description language, Verilog. Verilog implementation was compared to the fixed-point simulationr for comparison to the original behavioral model. Due to the Verilog simulator processing time restriction, a smaller test data set with 30 range lines and 52 samples per range line is used for this comparison. After byte to-byte agreement was confirmed, the AzPF Verilog design is synthesized and placed and routed on the XILINX FPGA.

Hardware Evaluation

AzPF hardware implementation was verified by exercising the hardware real-time at 100 MHz sampling rate with an input data set, and comparing the output to the output the fixed point S/W simulation exercised with the same input data set.

The hardware setup for the real-time verification is shown in Figure 7. Hardware verification was performed with the following two data sets:

- 30 lines with 52 samples per range line used for the Verilog simulator.
- 2000 lines with 8 K samples. This data set is generated by padding zeros to the Verilog test data set.

For each of these test data sets, the output of the AzPF hardware implementation and the output of the fixed-point AzPF simulation matched exactly byte-to-byte. Since the fixed-point simulation has been shown to meet the AzPF requirements, byte-to-byte agreement with it implies that the AzPF hardware implementation, run real-time at 100 MHz, meets the requirements.

VII. Follow-on work

If future funding becomes available, follow-on work to be pursued are:

- Augment AzPF the capability to center the raw data to zero Doppler centroid;
- Verify the augmented AzPF with AIRSAR data for evaluation of the AzPF on a distributed target, beyond the single-point target that have been completed at this time.

VIII. Summary

A SAR data reduction algorithm, the AzPF, has been designed and successfully implemented on a single, high-density FPGA. Associated behavioral model simulations have also been developed for performance verification. A real-time data input/output capability at 100 MHz was also developed for hardware verification. Hardware implementation meets the performance requirements.

Acknowledgment

The authors wish to thank Telecom Signal Processing Engineering and AIR SAR groups for the integration and testing support

Reference

[1] S. Madsen,"On-board Azimuth Pre-filter Processor", Interoffice Memorandum,Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, October 2000.
[2] C. Le and S. Madsen, "Analysis and implementation of the presummer", Interoffice Memorandum,Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, August 2001; also to be published in the IEEE Trans. Geoscience Remote Sensing.

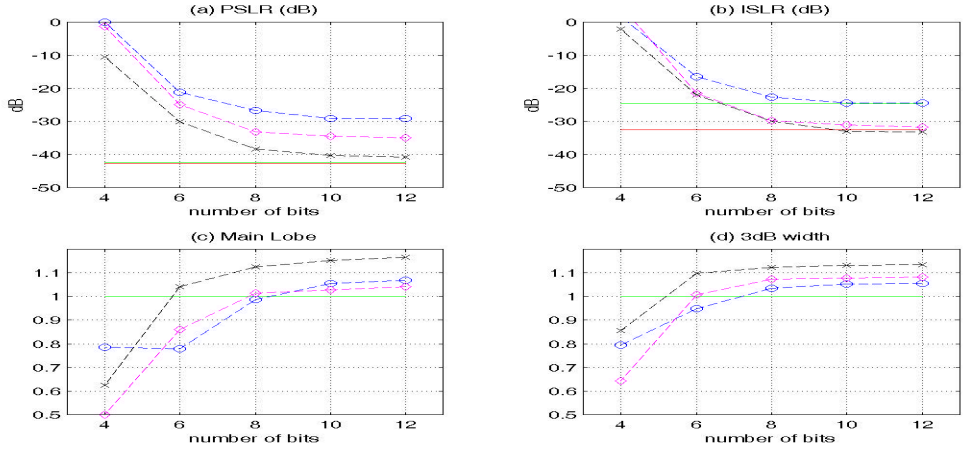


Fig. 3: (a) PSLR, (b) ISLR, (c) MLW, and (d) 3dBW in terms of the number of quantization bits for M = 4, N = 4M; solid red: simulated fullband signal; solid green: simulated subband signal; dotted blue with circles: using Hamming window; dotted black with crosses: using the Remez exchange algorithm; and dotted magenta with diamonds: using the Lagrange multiplier technique.

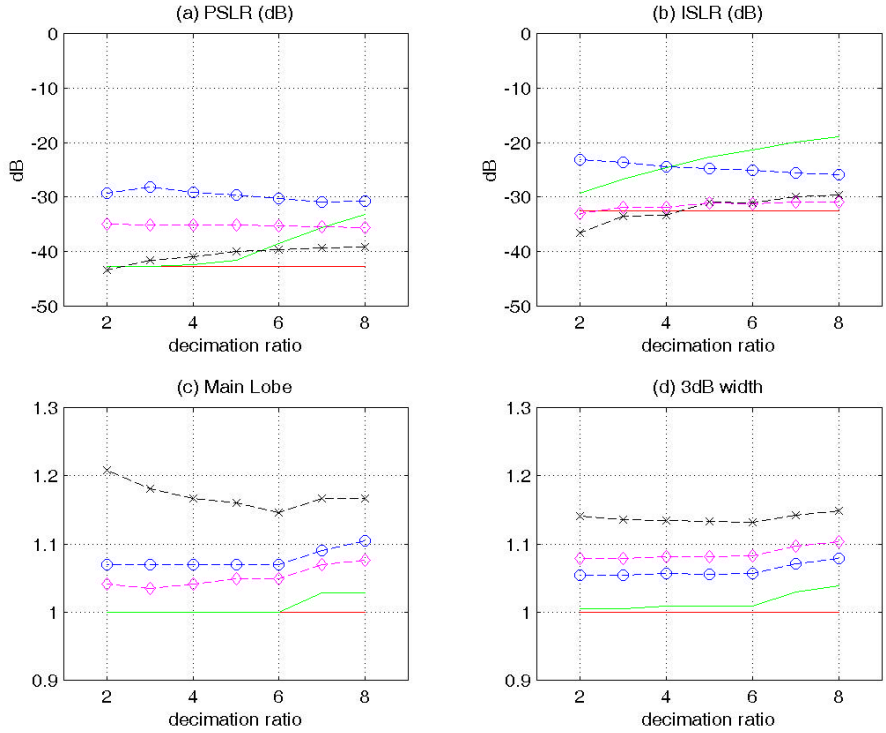


Fig. 4: (a) PSLR, (b) ISLR, (c) MLW, and (d) 3dBW in terms of the decimation ratio M with N = 4M; solid red: simulated fullband signal; solid green: simulated subband signal; dotted blue with circles: using Hamming window; dotted black with crosses: using the Remez exchange algorithm; and dotted magenta with diamonds: using the Lagrange multiplier technique.

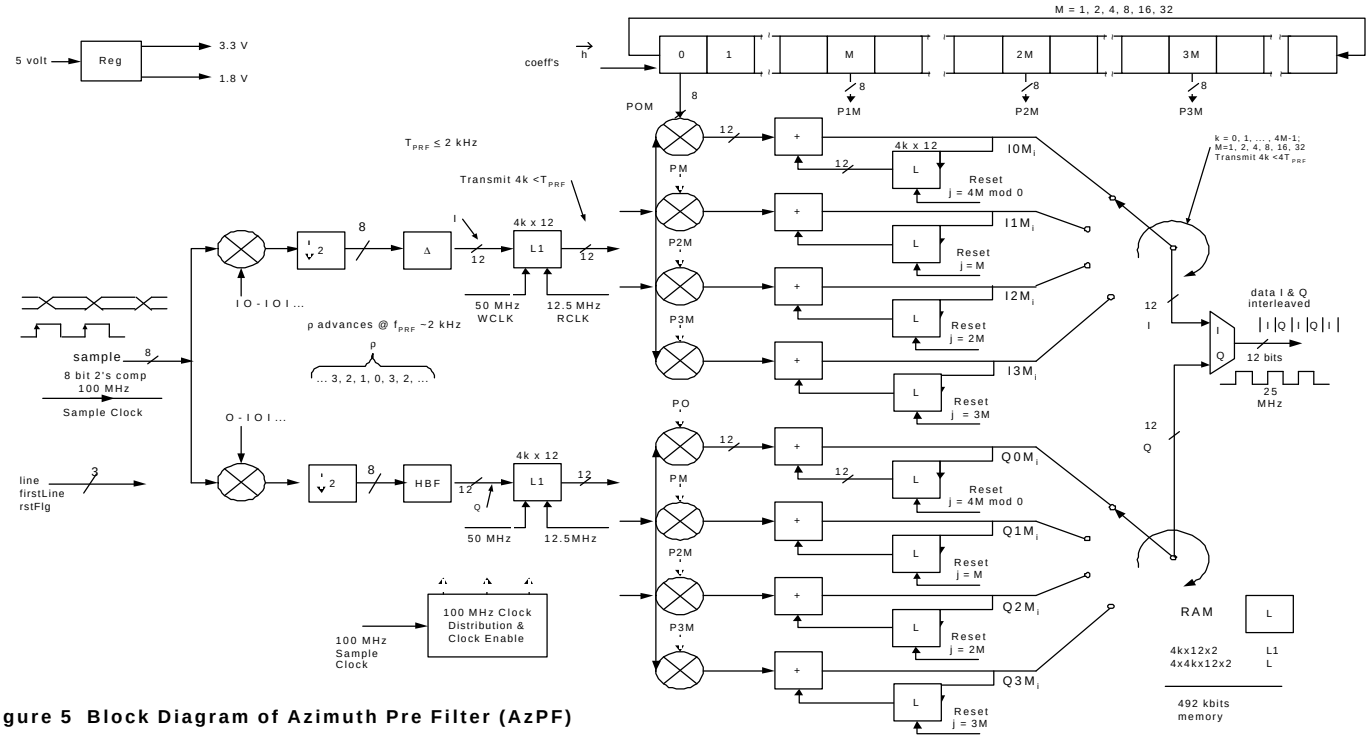


Figure 5 Block Diagram of Azimuth Pre Filter (AzPF)

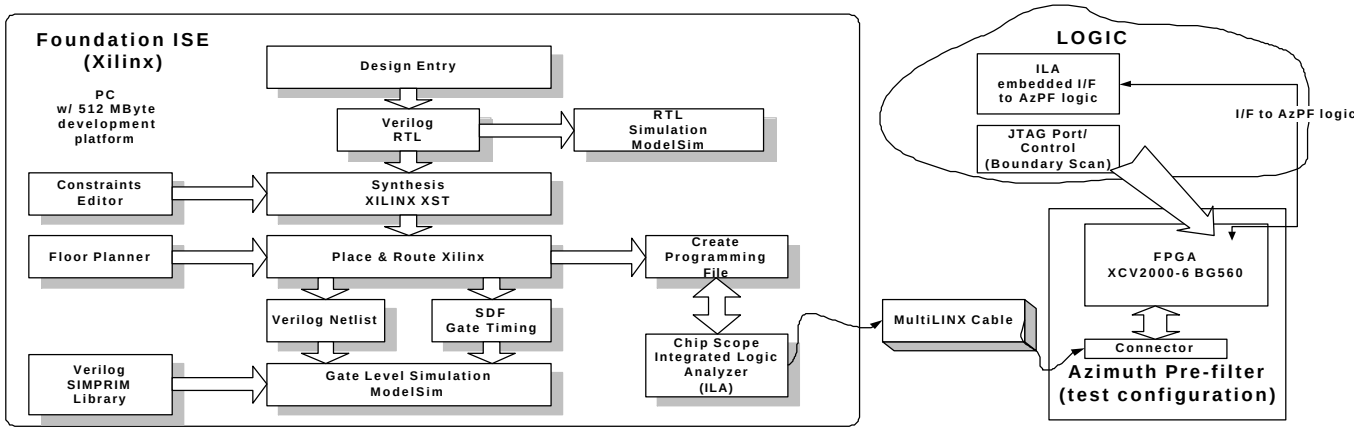


Figure 6 Design Flow

Table 4 : Performance Comparison (AzPF design with Hamming window, decimation factor 4)

	Requirement	Fixed Point AzPF	32bit AzPF	no filtering
ISLR	–15 dB	-25.1 dB	-25.2 dB	-31 dB
PSR	-25 dB	-30 dB	-30.6 dB	-43.8 dB
Res.	¼ of original Res.	4% broadening	4% broadening	0% broadening

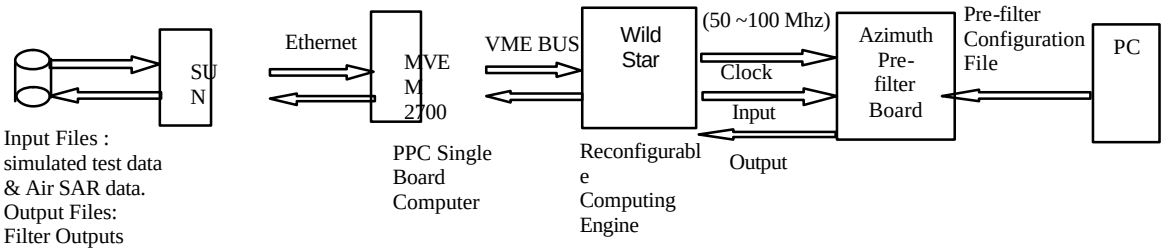


Figure 7. AzPF hardware test setup

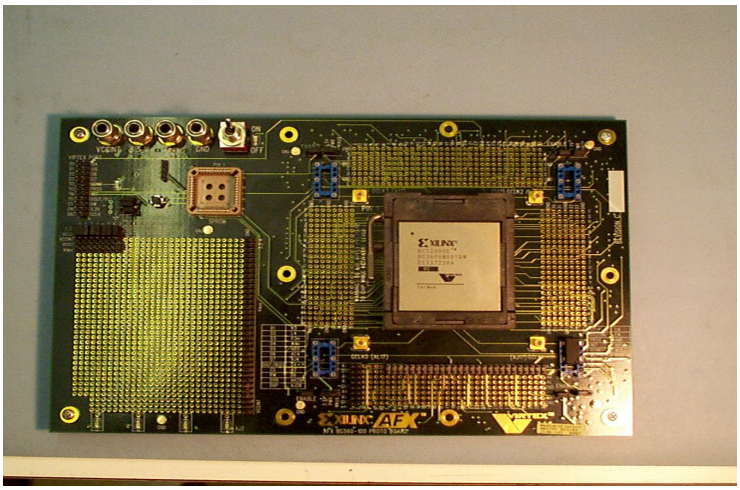


Figure 8. AzPF single-chip Hardware implementation (this prototype board is 6”X10”, but the single-chip containing the AzPF is only ~2.5”X2.5”)