Hindawi Publishing Corporation
International Journal of Reconfigurable Computing
Volume 2011, Article ID 894530, 14 pages
doi:10.1155/2011/894530

Research Article

An FPGA-Based Adaptable 200 MHz Bandwidth Channel Sounder for Wireless Communication Channel Characterisation

David L. Ndzi, Kenneth Stuart, Somboon Toautachone, Yanyan Yang, and Victor Dunn

Microwave Telecommunication Systems Research Group, Department of Electronic and Computer Engineering, University of Portsmouth, Anglesea Building, Anglesea Road, Portsmouth PO1 3DJ, UK

Correspondence should be addressed to David L. Ndzi, david.ndzi@port.ac.uk

Received 21 September 2010; Accepted 10 January 2011

Academic Editor: Fernando Pardo

This paper describes the development of a fast adaptable FPGA-based wideband channel sounder with signal bandwidths of up to 200 MHz and channel sampling rates up to 5.4 kHz. The application of FPGA allows the user to vary the number of real-time channel response averages, channel sampling interval, and duration of measurement. The waveform, bandwidth, and frequency resolution of the sounder can be adapted for any channel under investigation. The design approach and technology used has led to a reduction in size and weight by more than 60%. This makes the sounder ideal for mobile time-variant wireless communication channels studies. Averaging allows processing gains of up to 30 dB to be achieved for measurement in weak signal conditions. The technique applied also improves reliability, reduces power consumption, and has shifted sounder design complexity from hardware to software. Test results show that the sounder can detect very small-scale variations in channels.

1. Introduction

Channel sounders are used to study wideband signal transmission channels [1]. Although simple to design, in theory, they are complex to build and hence, expensive to purchase. Most channel sounders are developed using bespoke electronic systems. Although the use of wideband wireless communication systems is widespread and is expected to continue to grow, research into the channels that support these high data rate transmissions is limited, in most part, to simulations [2, 3]. Many wideband channel studies involve the use of general purpose instruments such as a vector network analyzer that can only be used for short-range investigations and are not suitable for fast channel sampling and mobile measurements [4, 5].

Compared to narrow band systems, channel sounders have to measure all frequencies within the bandwidth in a very short time during which the channel is assumed to be stationary [6]. For any given bandwidth, the Nyquist sampling criterion must be met. Channel probing signals that are widely used include Pseudorandom Binary Sequence (PRBS) and chirp [7]. Since the frequency spectra of these signals are not perfectly rectangular, the sampling frequency (f_s) must be more than twice the signal bandwidth (B_w). This generates large volumes of data in a short time and transferring this data from a sampling system for storage presents a unique challenge.

To facilitate further studies into the causes and impact of high-speed channel variations on wireless communication, an adaptive and fast channel sampling sounder was required. The objective was "*to developed a very compact, low cost and highly adaptable sounder to measure the transfer function of a channel, over the widest possible bandwidth in the shortest possible time and as often as possible with the ability to improve the channel signal-to-noise ratio*". To meet the compact and adaptable objectives, the design required the use of reconfigurable hardware. There are two main technologies that could have been used: Digital Signal Processors (DSP) and Field Programmable Gate Array (FPGA). DSP devices are specialised microprocessors that are typically programmed in C and are well suited to mathematically intensive tasks and offer robust software programmability [8]. However, DSP architecture capable of supporting high-speed accumulators are complex and limited compared to FPGA [9]. In contrast,

an FPGA device is programmed by connecting logic gates to form digital blocks such as multipliers, registers, and adders [10, 11]. FPGA devices support high clock frequencies and, both concurrent and parallel processes. Their capabilities are only limited by their logic capacities. There are also a wide range of FPGA devices available with high numbers of gates which offer more flexibility and quicker reconfiguration, both of which are essential for an adaptive system. Although FPGA and microprocessors are integrated on some devices such as MicroBlaze [12], FPGA was chosen for use in the implementation of the sounder. This marked a shift from hardware to software specification in channel sounder development. The use of FPGA also allows the system to be easily upgraded to keep pace with future communication system development needs. Most importantly, the use of FPGA devices enables a combination of many discrete components in the system on a single device. This contributes to a reduction in size, power consumption, hardware complexity, cost, and improvement in reliability of the system. Part of the design objective was to achieve an optimum performance at low cost with minimum hardware and software complexity using an architecture that can be ported to new devices to take advantage of the higher speeds and processing power that they may offer.

This paper is organised as follows: Section 2 gives an overview of the design objectives of the sounder and Section 3 describes the development of the transmitter and the receiver. Details of the software development and the data acquisition configurations that are available to the user are given. Section 4 describes the tests conducted to ascertain the performance of the sounder and the results obtained. These are followed by Section 5.

2. Sounder Specification and Design Objectives

A channel sounder is made up of the transmitter and the receiver as shown in Figure 1. The transmitter generates a waveform that is up-converted using appropriate Radio Frequency (RF) components to a suitable carrier frequency and transmitted. For this particular sounder, the waveform that is transmitted is a Pseudorandom Gaussian Noise (PRGN) [6]. PRGN has a noise-like characteristic in the time domain and a rectangular spectrum in the frequency domain. An example of a time domain and frequency spectrum of a PRGN signal is shown in Figure 2. The receiver down converts the received signal to a suitable intermediary frequency, where it is digitised, preprocessed (averaged), and stored.

2.1. Transmitter. The main requirement of the transmitter was to be reconfigurable to transmit signals with different bandwidths and spectral resolution (number of spectral lines within the bandwidth). To achieve this, a programmable Arbitrary Waveform Synthesizer (AWS) was required. Because of the noise-like characteristics of the PRGN waveform, a high resolution Digital-to-Analog Converter (DAC) was needed to maintain the accuracy of the signal. All frequencies used within the transmitter must be phase-locked to an external reference (10 MHz) to eliminate

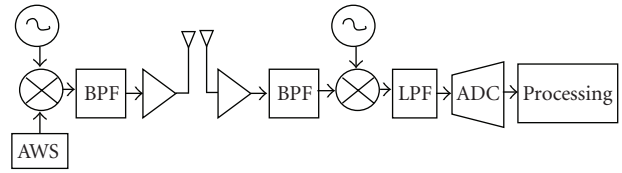

Figure 1: Simplified block diagram of the wideband channel sounder.

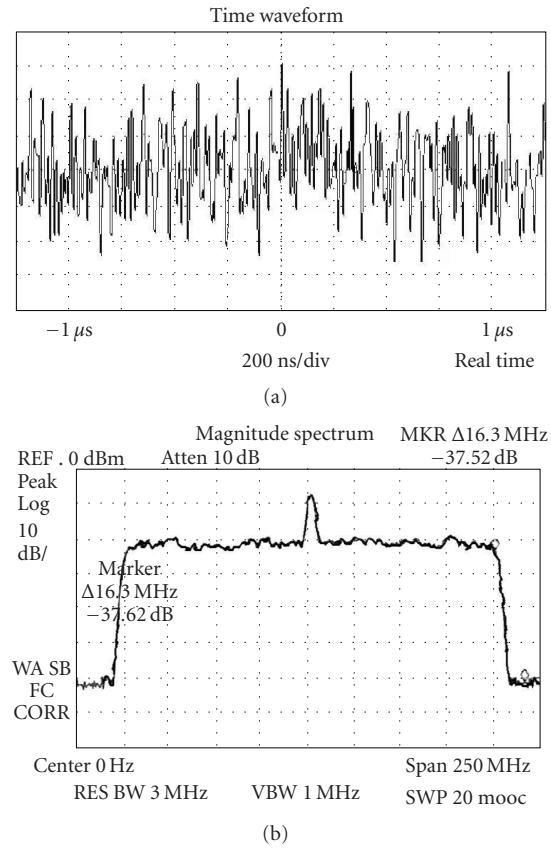

Figure 2: 200 MHz PRGN signal: (a) time domain signal and (b) magnitude spectrum.

transmitter to receiver range restriction. The generated waveform was to be up-converted to a carrier frequency of 2 GHz for transmission or as an Intermediary Frequency (IF) for further up-conversion.

2.2. Receiver. The design aims of the receiver are encompassed in the objectives of the system which require the receiver to be "*compact and highly adaptable to measure the transfer function of a channel, over the widest possible bandwidth in the shortest possible time and as often as possible with the ability to improve the channel signal-to-noise ratio*". To meet the compact objective, RF sampling was selected to minimise the number of components and reduce the complexity inherent in hardware in-phase and quadrature-phase demodulation [13]. The largest bandwidth was decided based upon the maximum clock frequency that a reasonably

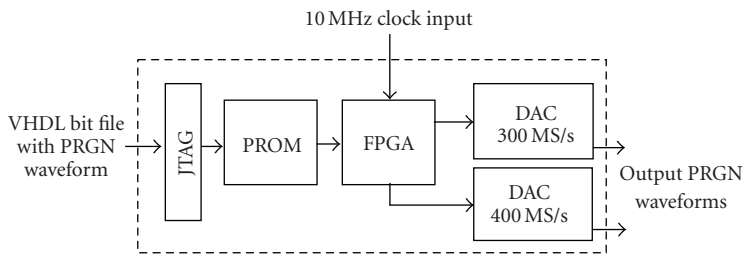

Figure 3: Simplified block diagram of the arbitrary waveform synthesizers.

priced FPGA device and a high-resolution Analog-to-Digital Converter (ADC) can support. Xilinx Virtex II FPGA devices were selected and they are specified to support clock frequencies up to 420 MHz. In order to meet the Nyquist sampling criterion, the design objective was geared to support a maximum bandwidth of 200 MHz to minimise uncertainties associated with operating close to the limits of devices.

The receiver must also be configurable to implement real-time averaging to improve the signal-to-noise ratio (SNR) in weak signal conditions. Other requirements include providing the ability for the user to adapt the receiver for different measurements by changing:

(i) the number of real-time averages to be carried out,

(ii) time gap between successive channel measurements,

(iii) time gap between a continual block of successive channel response measurements,

(iv) the duration of the measurement or the number of data files to capture,

(v) when configured in conjunction with the transmitter, the bandwidth and sequence length of the waveform.

3. Sounder Implementation

FPGA has been extensively used in the sounder to provide the flexibility required, minimize size, and improve reliability. Most importantly the developed architecture must be able to take advantage of current and future high-speed devices to increase the bandwidth, provide additional flexibility, and extend real-time processing capability of the system.

3.1. Transmitter. The main component in the transmitter is the arbitrary waveform generator. Although the RF unit is critical to the operation of the sounder, the main configuration that would be required is to change the carrier frequency. Therefore, more efforts were put into the design and construction of the AWS.

3.1.1. Arbitrary Waveform Synthesizer. The AWS was designed to be programmed through a Joint Test Action Group (JTAG) cable interface. A PROM is used to store the code that runs on the FPGA and the waveforms that are generated by the card. To ensure that the transmitter and receiver can be locked to one common frequency phase reference, a connection was provided for a 10 MHz input clock. The simplified block diagram of the implemented AWS card is shown in Figure 3.

A Xilinx Virtex II FPGA device was selected and used for its simplicity and low cost. Two 14-bit resolutions DACs with speeds of 400 MS/s and 300 MS/s were selected to provide flexibility and also allow robust testing to be carried out during development. The developed dual channel AWS can be used to generate two independent waveforms with bandwidths of up to 150 MHz and 200 MHz.

The operations of the AWS card and the functions of the VHDL software can be summarized as follows: at power up

(i) load the programme code and digital waveforms from the PROM into the FPGA;

(ii) check the status of the input 10 MHz clock;

(iii) if clock is available, generate the loaded waveform.

To generate the 150 MHz and 200 MHz bandwidth signals, 300 MHz and 400 MHz clocks are required. VHDL code was written to multiple the 10 MHz input reference to generate these high frequency clocks using the Digital Clock Management (DCM) modules. The FPGA device that has been used has 2 clock multiplier modules:

(i) low frequency module with output clock frequencies from 24 MHz to 240 MHz,

(ii) high frequency module with output clock frequencies from 24 MHz to 420 MHz.

The 300 and 400 MHz clocks could be generated using one low- and one high-frequency multiplier clock modules. However, for low-frequency module outputs of 130 MHz and above, the clock jitter is very high, more than 10% of the clock period. This is outside the jitter tolerance of the high-frequency module. Thus stable 75 MHz and 100 MHz clocks were first generated and two high frequency modules were then used to provide the final times 4 multiplication factor. Tests showed that changing the voltage reference of the FPGA device from 1.5 V to 1.65 V improved the stability of the generated clocks.

The width of the data bus of the DACs is 14 bits wide. Tests showed that clocking the data from a single register onto the data bus at 300 MHz and 400 MHz introduced high skew. This was overcome by using Double Data Registers (DDR). This enabled the clock frequency to be halved with data clocked from one register on the rising edge of the clock

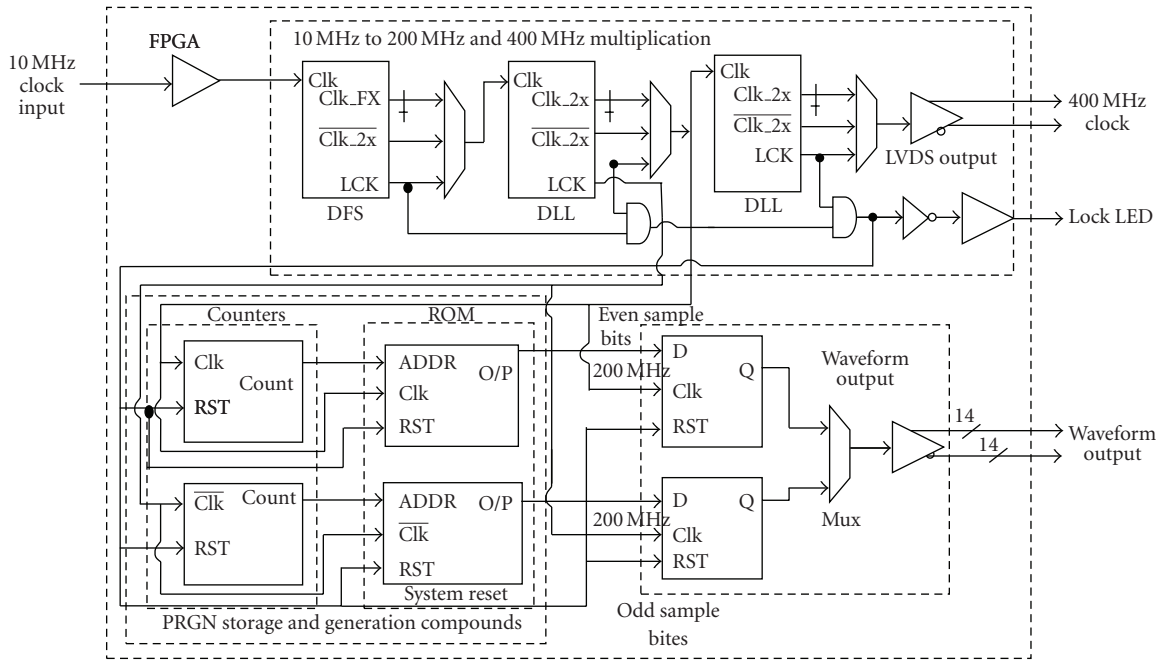

Figure 4: Block diagram of the 200 MHz bandwidth AWS VHDL design.

and from the second register on the falling edge, thus reducing jitter and data skew. The VHDL logic for the 200 MHz bandwidth signal generation is illustrated in Figure 4.

In the frequency domain, the PRGN waveform from the AWS has a "flat" amplitude spectrum. When clocked at 400 MHz, the waveform has a bandwidth of 200 MHz and has been designed to have 1024 discrete spectral lines in the range ±100 MHz, spaced by 195.313 kHz. Figure 2 shows the 200 MHz PRGN baseband time domain signal and the magnitude spectrum generated by the AWS, and Figure 5 is a photo of the built AWS card. The card has dimensions of 10 cm by 12 cm.

3.2. Receiver. The main goal of the receiver design was to develop a highly flexible data acquisition system that would enable the channel sounder to be adaptable for measurements in a wide range of scenarios. This required hardware that could be readily reconfigured. The receiver can be divided into; the RF unit, frequency reference unit, ADC, FPGA-based Peripheral Component interconnect Mezzanine Card (PMC) for real-time data processing, and a Linux workstation as shown in Figure 6.

3.2.1. RF Unit. The design philosophy of the receiver RF system was to optimise the sensitivity and adaptability of the receiver. Thus, a programmable digital attenuator was incorporated into the design for automatic gain control to maintain the signal within the linear range of the RF components and to protect the ADC from high input signals.

For a 200 MHz bandwidth signal transmission, the received signal is down converted to an IF of 100 MHz using a phase-locked oscillator. Frequency stability and phase synchronisation are achieved by using a stable 10 MHz reference phase-locked to the same source as the transmitter.

3.2.2. ADC and ADC Carrier Card. A 12-bit ADC with sampling speeds of up to 400 MS/s was selected to minimise quantization errors [14]. Due to the rectangular nature of the PRGN spectrum, the sampling frequency is set at exactly the Nyquist frequency. An ADC carrier card with an onboard FPGA device was designed to be used to generate the sampling clock required by the ADC [15]. The sampling clock is generated using the same technique developed for the AWS.

The carrier card was also designed to route control signals and data between the ADC and the PMC cards. In addition, the carrier card acts as an automatic gain controller. Figure 7 shows a simplified block diagram of the ADC carrier card.

3.2.3. PMC Card. To minimise cost, it was important to use standard computer components. To provide flexibility and ensure data integrity, an interface card was selected to be used between the ADC and the computer to process and buffer the captured data. To provide the data acquisition flexibility required, the interface card also had to

(i) have a data processing engine (DSP and/or FPGA),

(ii) have an external clock input to synchronise internal data processing and transfers with the rest of the system,

(iii) be programmable such that the data sequence, number of averages, and data transfer rates could be changed when required,

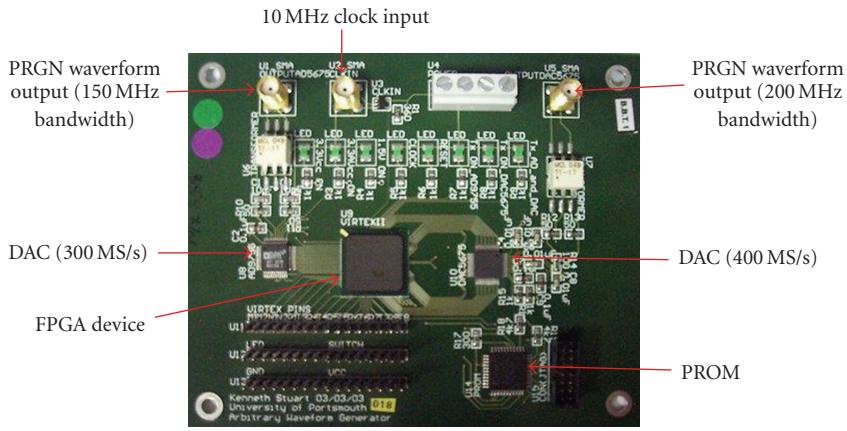

Figure 5: Photograph of the dual channel arbitrary waveform synthesizers.

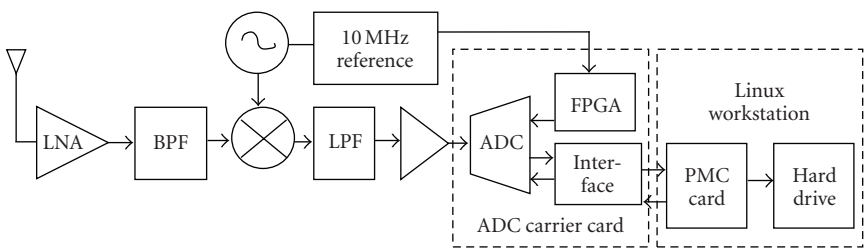

Figure 6: Simplified block diagram of the receiver system.

(iv) have a large memory to store a significant amount of data to provide fast channel sampling capability,

(v) support the computer data bus protocol.

After evaluating a number of programmable cards available on the market, the Alpha Data Ltd PMC-XRC card with an onboard FPGA device and memory banks was selected [16]. The PMC is mounted on a PMC-to-PCI bridged adapter card for connection to a computer motherboard [17].

3.2.4. Real-Time Averaging. One of the main design objectives was for the channel sounder to be effective in low SNR conditions. This could be achieved by performing periodic averaging of the received signal. Although averaging could be done offline, there was a need to reduce the quantity of data that has to be transferred and stored. With this in mind, processing of the data is done on the PMC card and has been implemented entirely in VHDL.

The ADC output has two channels which halve the speed of data output compared to the sampling rate. Therefore, the data averaging logic in the FPGA was doubled to process each channel as shown in Figure 8. The ADC produces Data Ready signals for each channel (DRA and DRB for channel A and B, resp.) that indicate when data samples are valid at the outputs. These are synchronised with the sampling clock and have frequencies that are half the ADC sampling frequency. These clocks are used to drive the data accumulator logic.

Table 1 shows the counter settings required for the different number of averages, the time it takes to capture one averaged snapshot (T_{as}), and the theoretical improvement in SNR (ΔSNR) [18]. The averaging counter values are calculated using (1), where N_s is the length of the sequence and N_a is the number of averages.

Table 1: Averaging counter value for possible number of averages, durations, and expected improvement in SNR.

N_a	Counter value (Hex)	T_{as} (ms)	Theoretical ΔSNR (dB)
0	FFF	0.01024	0
2	1FFF	0.02048	3
4	2FFF	0.04096	6
8	4FFF	0.08192	9
16	8FFF	0.16384	12
32	10FFF	0.32768	15
64	20FFF	0.65536	18
128	40FFF	1.31072	21
256	80FFF	2.62144	24
512	100FFF	5.24288	27
1024	200FFF	10.48576	30

$$\text{counter value} = \left(\frac{N_s N_a + 2N_s}{2}\right) - 1. \quad (1)$$

3.2.5. Data Transfer and Storage. The rate at which captured data can be transferred from the PMC to the host computer and written to the hard disk is key to the data acquisition strategy and, in particular, the channel sampling rate. If data

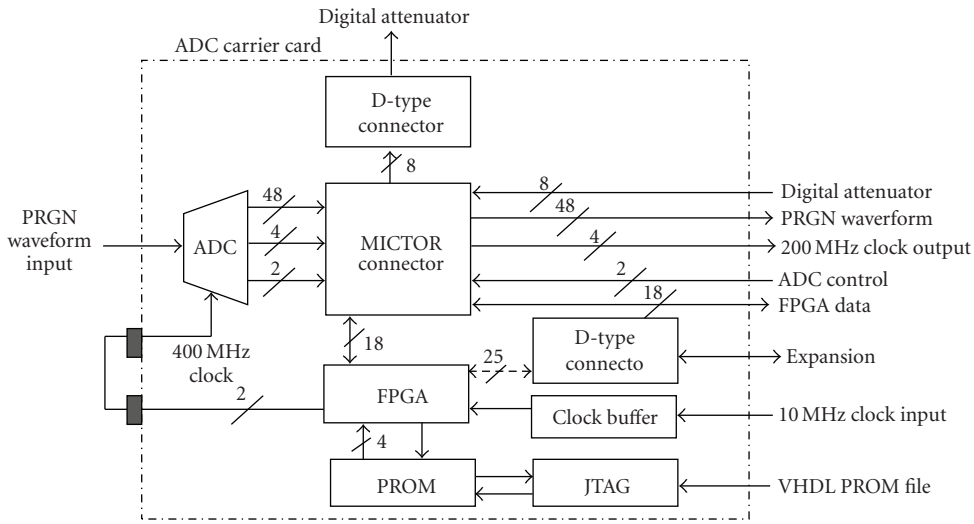

Figure 7: Simplified block diagram of the ADC carrier card.

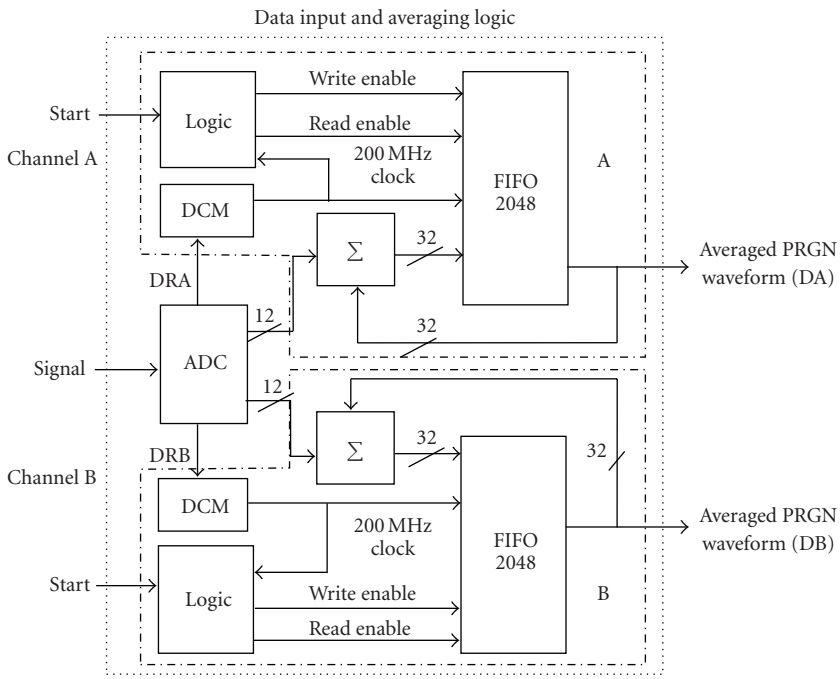

Figure 8: Block diagram of the averaging process.

is not read from the PMC card memory before the next set is captured, it will be overwritten. To prevent lost or corruption of data, two issues had to be resolved: the data transfer clock frequencies and the timing of when data should be transferred to the host computer.

The clock frequency options that are available when using a 32-bit PCI bus are 33.3 MHz and 66.6 MHz. The clocks used in the averaging process have a maximum frequency of 200 MHz. This means that, at least, three snapshots can be captured in the same time that one is transferred across the PCI bus. Furthermore, the two clock domains are separate and are not phase-locked. To overcome this asynchronous problem, First-In-First-Out (FIFO) buffers were used. The write clocks into the FIFOs are the DRA and DRB, and the read clock is the PCI bus clock. A block diagram of the data transfer logic is shown in Figure 9.

The read enable of the FIFOs is controlled by the value of the averaging counter such that on the last average, the data is transfered to the asynchronous FIFO. If the computer is the bus master, and the PMC card is the slave, the data contained in the asynchronous FIFOs will not be necessarily completely read before new data is available to be written to the FIFOs. This is because the computer uses a polled interrupt routine to service the PCI bus. Since it is critical

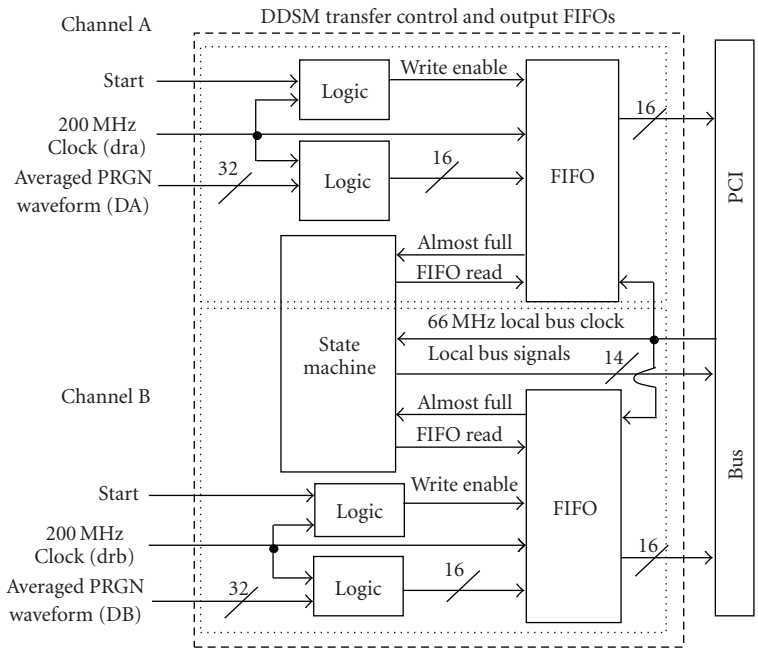

Figure 9: Block diagram of the output buffer VHDL.

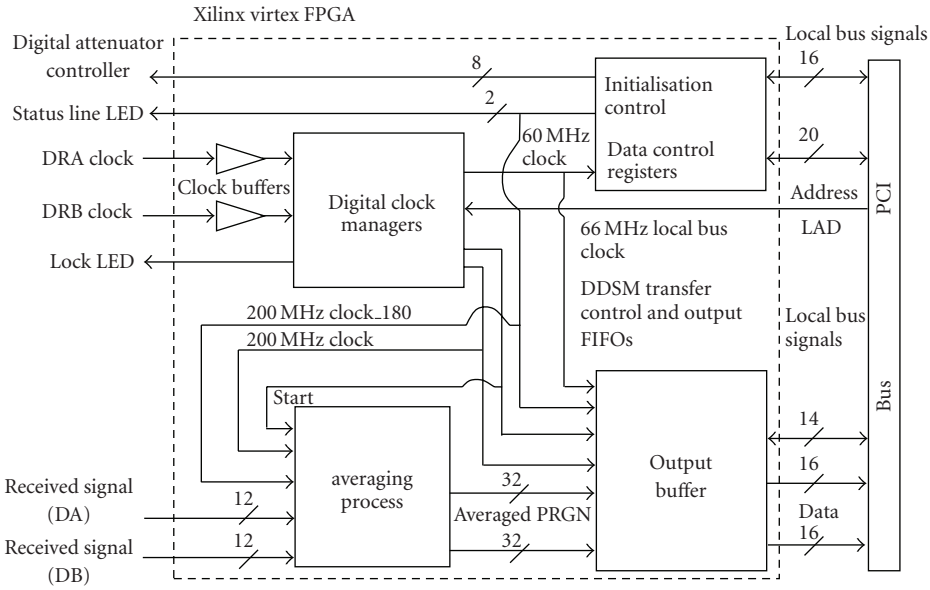

Figure 10: Block diagram of the PMC VHDL design.

that no sample is overwritten, the PMC card must be the PCI bus master. This was achieved by using Demand-mode Direct Memory Access (DDMA), where the PMC initializes a read transaction when one complete data sequence has been written to the asynchronous FIFOs. The computer then writes the data to disk and no samples are lost.

The data output of the ADC is 12-bit signed binary. To perform 1024 averages, a 22-bit adder is required. To complete the averaging process, the summed result must be divided by the number of averages. However, if this is performed in the FPGA, the averaged result will be truncated to a whole number. This is undesirable in low SNR conditions where an actual representation of the signal in comparison to the noise level is required [19]. Saving the data and carrying out floating point division in the computer provide the best solution. However, this would require 22 bits per sample to be transferred. Since the PCI bus used is 32 bits wide, only one sample would be transferred per clock period.

Since the ADC generates two samples per clock, the channel sampling rate would need to be halved for number of averages greater than 16. This was undesirable and hence the width of each sample was limited to 16 bits. For a number

of averages greater than 16, only the most significant 16 bits are transferred. Discarding some of the bits in this manner corresponds to predivision by a factor of 2^{nb}, where nb is the number of bits discarded. When 8 or less averages are performed, the number of data bits per sample is less than 16. Thus, to ensure data integrity the sign bit for each sample is duplicated to make the sample 16 bits wide.

Figure 10 shows a block diagram of the top level VHDL design for the PMC card illustrating how each part of the VHDL logic is linked together.

3.2.6. Data Acquisition. During measurements, a user must be able to implement a data acquisition strategy that is suitable for the channel under investigation. A combination of C and VHDL was used to implement the software. To achieve the design goals of the receiver as outline in Section 2.2, emphasis was placed upon transferring the measured data from the output of the ADC to the computer hard drive as fast as possible.

The PCI data bus rate and the hard drive read/write speed are the two most critical factors in the receiver. Tests were carried out to determine the sustainable data rate from the ADC to the hard drive. This was achieved by using a VHDL design that transfers data using DDMA from memory across the host machine's PCI bus. In addition, an operating system tool (hdparm) was used to test the read/write speed from/to the computer hard drive. The tests revealed that the average PCI bus data rate was 55.7 MB/s and that an average of 40 MB could be written to the hard disk per second. The data acquisition strategy was then designed around these results.

When sampling at 400 MS/s, the 12-bit ADC can generate over 600 MB of data per second. Since the hard disk writing speed is only 40 MB/s, a 15-time reduction in data rate was required. Due to the dual channel nature of the ADC, the first reduction in the data rate is achieved within the ADC from the interleaving configuration [14]. Further reduction could be achieved as a consequence of performing real-time averaging on the PMC card. The time taken to capture one averaged channel response (snapshot), T_{as}, is given by

$$T_{\text{as}} = \frac{N_s N_a}{2 f_c} + T_{\text{sg}}, \tag{2}$$

where N_s is the sequence length, N_a is the number of averages, T_{sg} is the time required to reset the counters and FIFO, and f_c is the input clock frequency in Hz. Due to timing constraints, T_{sg} was set to a minimum value of 20.48 ms. Considering a sequence length of 4096, each snapshot has a size of 8.192 kB. The quantity of data generated for each value of N_a is shown in Table 2. It can be seen that when the number of averages is set to 32 or greater, data can be captured continually. For number of averages less than 32, the quantity of data captured per second must be limited to 40 MB. This can be done by increasing the time gap (T_{sg}) between snapshots or block of continual channel response measurements such that the number of continual channel responses captured per second is less than 4882. During measurements, a trade-off must be made between the channel sampling rate and quantity of data to be captured.

Table 2: Time to capture one channel response, number of averages, and the volume of data generated per second.

N_a	T_{as} (ms)	No. of snapshot per second	Data generated per second (MB)
0	0.03072	97656	266.666
2	0.04096	48828	200.000
4	0.06144	24414	133.333
8	0.10240	12207	80.0000
16	0.18432	6103	44.4444
32	0.34816	3051	23.5294
64	0.67584	1525	12.1212
128	1.33120	762	6.15385
256	2.64192	381	3.10076
512	5.26336	190	1.55642
1024	10.5062	95	0.77972

Although the continual channel sampling rate is limited to 4.882 KHz, the user can achieve channel sampling rates up to 5.42 KHz for subsecond continual channel measurements. For a bandwidth of 100 MHz and 2048 long sequence, channel sampling rates up to 15 KHz are achievable. This provides the flexibility to investigate very high speed events.

Figure 11 illustrates the data acquisition strategy that has been developed. The following terms, defined in the diagram, are used to describe the data sets:

(i) snapshot: a complete sequence (4096 samples long) of the PRGN waveform,

(ii) averaged snapshot: a complete sequence (4096 samples long), produced by the averaging process,

(iii) data block: 240 snapshots,

(iv) data file: 4 data block.

The user can change the number of averages (N_a), the time between snapshots (T_{sg}), and the time between data block (T_{dbg}) to suit the type of channel under investigation. The resulting time to capture one averaged snapshot (T_{as}), one data block (T_{db}), and one data file (T_{df}) can be calculated using (2), (3), and (4) respectively.

$$T_{\text{db}} = 240 T_{\text{as}} + 239 T_{\text{sg}}, \tag{3}$$

$$T_{\text{df}} = 4 T_{\text{db}} + 3 T_{\text{dbg}}, \tag{4}$$

where

$$T_{\text{sg}},\ T_{\text{dbg}} = N_g \left(\frac{2N_s}{f_c} \right), \tag{5}$$

and N_g is the number of multiples of the time taken to reset all counters and FIFOs used in the averaging process. T_{sg} and T_{dbg} are independent of each other. Table 3 provides the details of the timing parameters for each number of averages, N_a, and the possible channel sampling rates for 180 MHz bandwidth signal transmission.

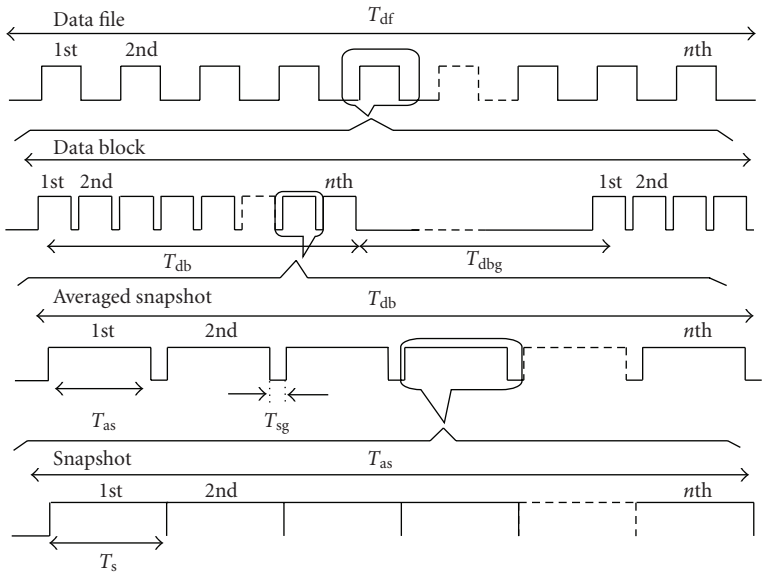

Figure 11: Data acquisition strategy for N_a averages.

Table 3: Channel sampling rates for different number of averages for 180 MHz bandwidth configuration.

N_a	Theoretical ΔSNR (dB)	T_{as} (ms)	Channel sampling rate (Hz)
0	0	0.12288	8138
2	3	0.12288	8138
4	6	0.12288	8138
8	9	0.20480	4882
16	12	0.36864	2712
32	15	0.69632	1436
64	18	1.35168	739
128	21	2.66240	375
256	24	5.38384	189
512	27	10.5267	94
1024	30	21.0125	47

4. Sounder Test and Results

Tests on the sounder highlighted a number of short comings in the software and hardware. The four areas of concern were the frequency stability of the sampling clock generated by the FPGA device on the ADC carrier card, the timing constraints of the software running on the PMC card, the data rate of the PCI bus, and the rate at which data is written to disk.

Frequency stability and the rate of data transfer to the storage device are critical to the implementation of a suitable data acquisition strategy, thus any associated problems had to be overcome. These included the addition of a new input clock for the ADC and revision of the VHDL design for the FPGA on the PMC card to improve the channel sampling rate. The following sections describe tests that were conducted, some of the issues that were identified, solutions implemented, and the performance of the whole system.

4.1. Frequency Stability of the Experimental System. High stability of all the frequency sources is critical to the real-time averaging of PRGN signals [18]. If the sources are not locked to a common phase reference, errors are introduced. Figure 12 shows measured channel transfer functions with and without the transmitter and receiver phase-locked to a common reference.

Using a frequency counter, all the FPGA derived clocks on the AWS and the ADC carrier card were measured and found to be locked to the frequency reference. However, the data sampled by the ADC was found to contain errors. These errors are not observed when the sampling clock is obtained from a dedicated source for example, signal generator. Tests revealed that the clock jitter was higher than the maximum recommended tolerance for the ADC input [20]. Figure 13 shows measurements taken with no signal applied to the input of the receiver. Errors with peak-to-peak values up to 3.25 V were obtained with the original FPGA generated clock. Since a low frequency clock from the FPGA had superior SNR and low jitter, a dedicated external clock multiplier could be used to generate the sampling clock for the ADC. Using the stable sampling clock, only the system noise with maximum peak-to-peak amplitudes of 0.005 V was present as shown in Figure 15(b). The tests were repeated with the PRGN signal connected to the input of the receiver. The results in Figure 14 show that the errors are eliminated when a stable clock is used.

4.2. Real-Time Averaging. In order to perform real-time averaging, consecutively captured snapshots must be added together and the result is divided by the number of snapshots. Since the PRGN waveform has noise-like characteristics, the tests carried out on the averaging logic were conducted using known test signals (counter). Included in the VHDL design and substituted for the ADC input to the adders, the Most Significant Bit (MSB) of the counter value was considered

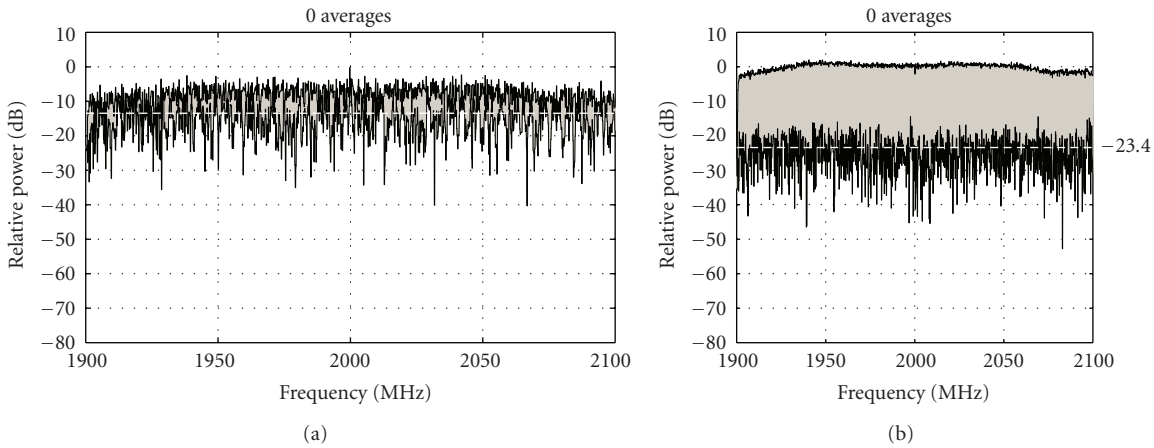

Figure 12: Measured transfer functions (a) without and (b) with the transmitter and receiver phase-locked to a common reference.

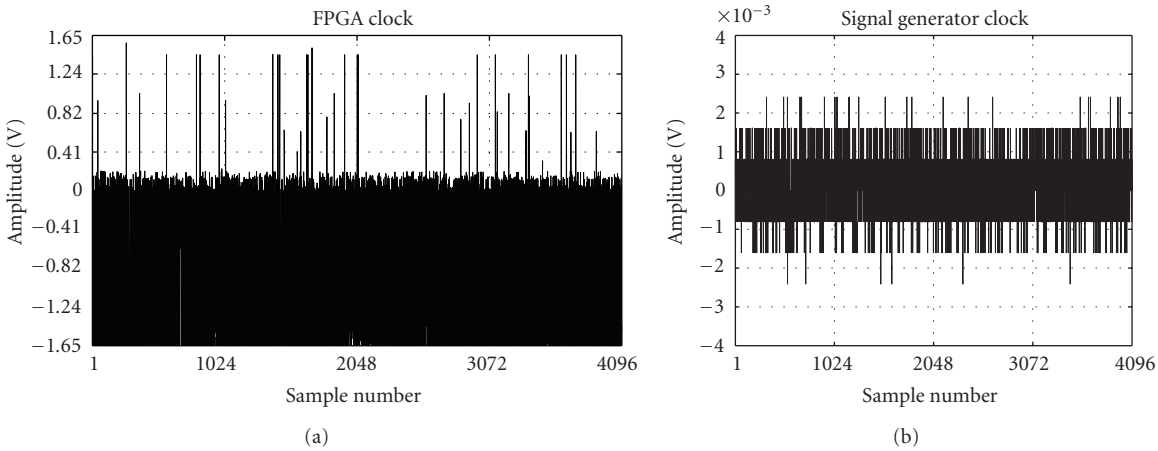

Figure 13: Data captured with no signal input using clocks from: (a) the FPGA and (b) the signal generator.

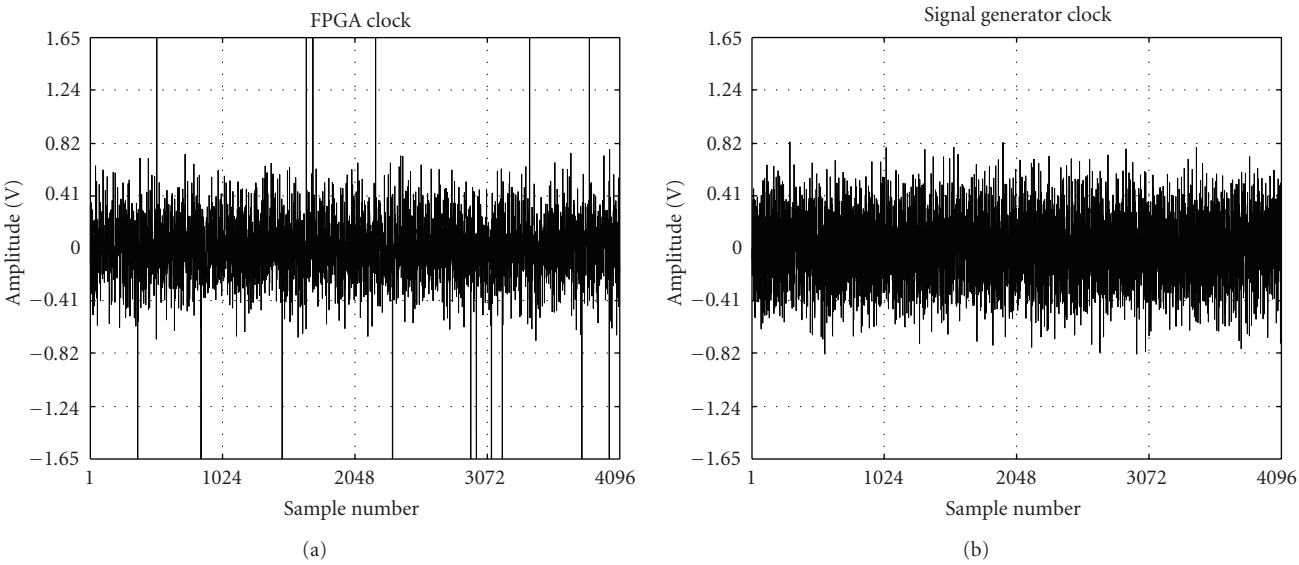

Figure 14: PRGN time domain signal captured with signal applied to ADC input with clocks from: (a) the FPGA and (b) signal generator.

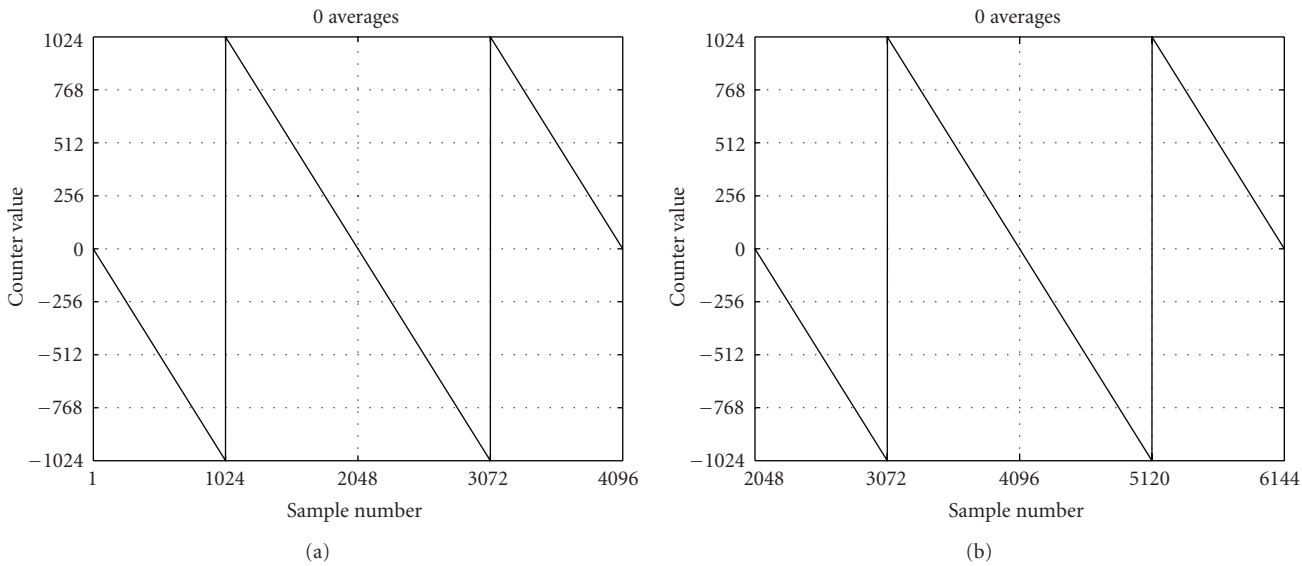

Figure 15: Averaging logic output (a) one sequence and (b) output showing transition between two sequences.

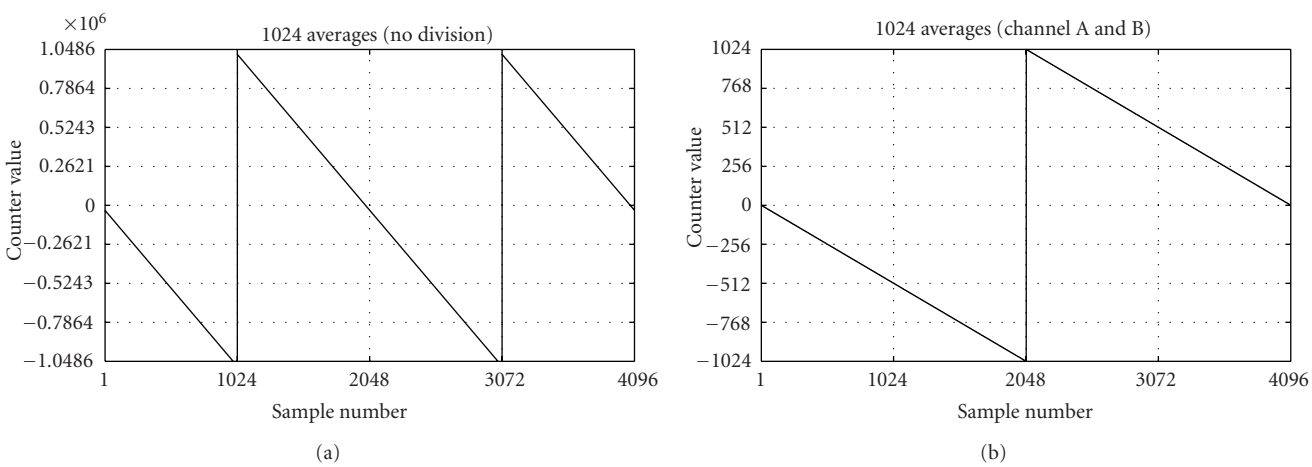

Figure 16: Averaging logic output for: (a) 1024 averages and (b) one sequnce from combined channels A and B.

as the sign bit of the data. The expected saw-tooth signal values lay between −1024 and +1023. Since the data is split into two channels, and the length of each sequence is 2048 samples long, each counter turns over twice to produce 4096 samples. This is equivalent, in length, to the PRGN sequence. Since the two channels use different clocks, they were tested separately. Figure 15(a) shows the output from the counter when no averaging was preformed. Figure 15(b) shows that accuracy is maintained across the sequence boundary which is critical for averaging.

In view of the fact that the division of the added snapshots is preformed after the set number of averages, the expected measured values that are written to file when the counters are used as the input to the adders (emulating the averaging process) are the original counter values multiplied by the number of averages. The expected output for 1024 averages is from −1048576 to +1048575. Note that all the sample bits were transferred to the computer for this test. The output from channel A after 1024 averages is shown in Figure 16(a). Test results using both data channels in an interleaving configuration, similar to when data is captured, is shown in Figure 16(b) after dividing by the number of averages. The results show that the averaging software is working as intended.

4.3. Signal to Noise Ratio. The transmitter and receiver RF units were designed to optimise the sensitivity of the receiver and minimise intermodulation products. In order to achieve this, measurement of input and output power and signal to intermodulation product ratios were evaluated at every stage in the RF chain.

All the components in a system have a noise figure and therefore generate noise that may not be band limited. The total noise power can be calculated by the integration of the

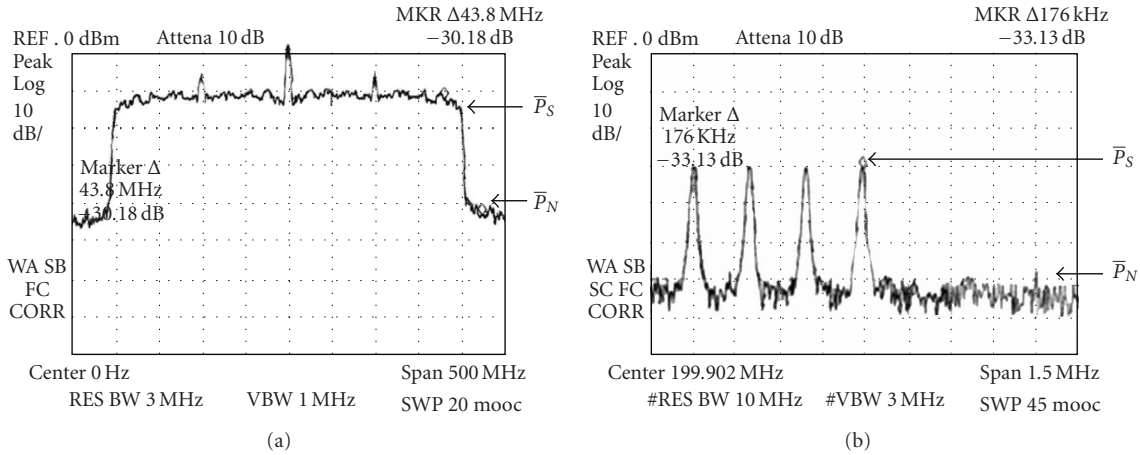

Figure 17: IF: (a) spectrum of PRGN signal with a 200 MHz bandwidth and (b) spectrum at the edge of the band.

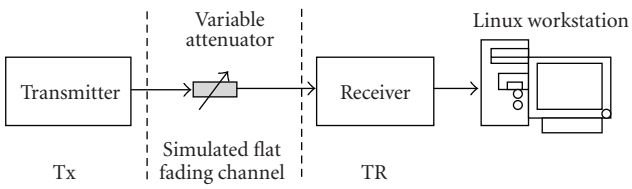

Figure 18: Block diagram of hardware for flat fading simulation.

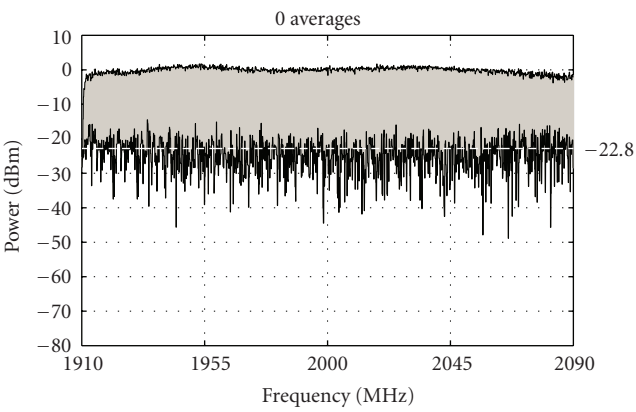

Figure 19: 180 MHz bandwidth PRGN spectrum with no averaging ($N_a = 0$).

continuous spectrum over a given bandwidth, B_W, which is taken to be the bandwidth of the last filter through which the signal plus noise is passed. The total noise power is given by

$$P_N = \int_0^{B_w} S_{N|\mathrm{RWB}=1\,\mathrm{Hz}} \partial f = S_{N|\mathrm{RWB}=1\,\mathrm{Hz}} \times B_w, \qquad (6)$$

where $S_{N|\mathrm{RWB}=1\,\mathrm{Hz}}$ is the noise power spectral density at 1 Hz Resolution Bandwidth (RBW). Therefore, the SNR can be evaluated by

$$\mathrm{SNR} = 10\log_{10}\left(\frac{P_S}{P_N}\right), \qquad (7)$$

where P_S is the total signal power. Figure 17 shows the PRGN spectrum at the receiver IF and the corresponding expansion of the spectrums at the edge of the band. The mean output power, $\overline{P_S}$, across the bandwidth at IF is −9 dB. The mean noise power level, $\overline{P_N}$, is −30 dB, giving an SNR of 21 dB.

Back-to-back tests were conducted to assess the performance of the data acquisition hardware and software under simulated fading conditions. Indoor channel measurements were also conducted to evaluate the sounder. Comparisons were made between the measured improved SNR and the theoretical value.

4.4. Simulation of Flat Fading Channel. To test the performance of the system under flat fading channel conditions, 180 MHz and 200 MHz bandwidth PRGN waveforms were used. Tests were carried out with different numbers of averages, N_a, at a carrier frequency of 2 GHz. The back-to-back test configuration is shown in Figure 18.

Figure 19 shows the transfer function of the waveform measured without averaging over a 180 MHz bandwidth. The system response introduces a slight amplitude variation across the band. These were not yet calibrated out. The magnitude of the components between the spectral lines represents noise plus any intermodulation products. Since it is not possible to separate any intermodulation products from the noise, the estimated mean level represents an equivalent Signal-to-Intermodulation and Noise Ratio (SINR). The estimated SINR is 22.8 dB.

The results of measurements with 2 and 8 averages are shown in Figure 20. The improvement in SINR can be seen as the number of averages is increased from 0 (Figure 19) to 8 (Figure 20). The results are comparable to theoretical values which show that improvement in SNR is given by $10\log N_a$. This validates the performance of the system and proves that improvement in SNR is close to the theoretical values with averaging.

To test the ability of the sounder to detect small changes in the channel, measurements were taken in indoor environment at an antenna separation of 5 m at 3 cm intervals

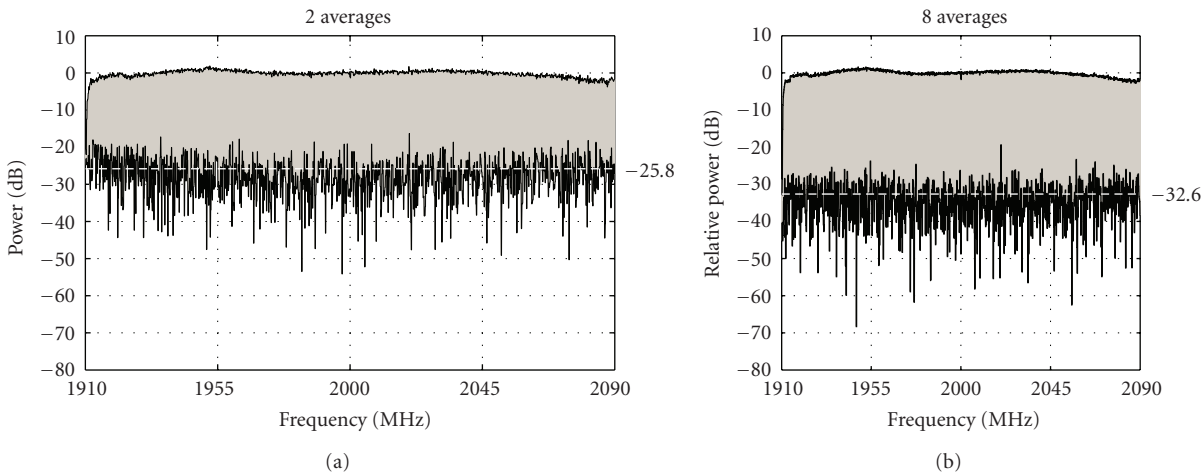

Figure 20: 180 MHz bandwidth spectrum with number of averages of (a) 2 and (b) 8 showing a 8.8 dB SNR improvement compared to 0 averages.

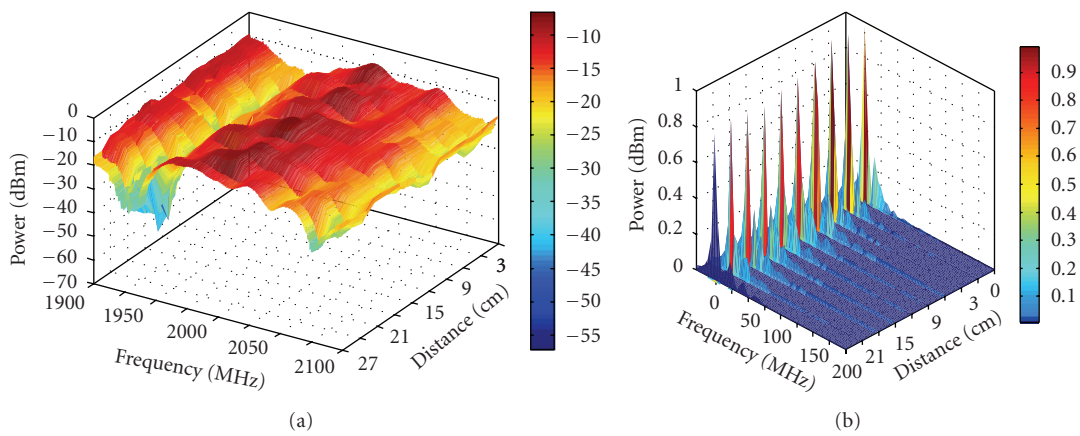

Figure 21: Small-scale variation measurement in an indoor channel; (a) channel transfer function and (b) impulse response.

over a distance of 27 cm at 2 GHz. The transfer functions and the corresponding channel impulse responses are given in Figure 21. It can be seen that the system can clearly detect the evolution of the channel response over small distances. A good understanding of small-scale indoor channel variation is key to the effective delivery of wireless high data rate services [21, 22].

5. Conclusions

The development of a compact, adaptable and low cost channel sounder with transmission bandwidths of up to 200 MHz has been presented. The sounder has been developed using FPGA technology which allows future upgrades to be achieved simply by replacing the FPGA devices with newer and faster devices. The extensive use of FPGA at both the transmitter and receiver marked a shift in sounder development from hardware specification to software. Because of the high cost of channel sounders, efforts were put into reducing the complexity of the system from hardware to software. In addition, to meet the low-cost design objective of the system and maintain low hardware complexity, simpler FPGA devices were used.

The transmitter unit is built around a dual channel arbitrary waveform synthesizer card that can generate two independent waveforms with bandwidths up to 150 MHz and 200 MHz simultaneously. This would facilitate colocation of more than one transmitter. The main challenges faced in the development of the arbitrary waveform synthesizer included difficulties with generating stable high frequency clocks to drive the digital-to-analog converters. In the receiver, the signal is sampled at an intermediary frequency for example, 100 MHz for a 200 MHz bandwidth signal. The signal is digitised using a 12-bit analog-to-digital converter with sampling speeds up to 400 MS/s. The data is transferred to an FPGA-based data processing card for real-time averaging. It is buffered before it is transferred across the PCI bus for storage on the computer hard disk. All the control signals on the sampling cards were derived from a 10 MHz input frequency reference. A robust data acquisition system has

been developed to ensure data integrity. Test results show that the real-time averaging process improves the signal-to-noise ratio inline with expected value defined by $\Delta\text{SNR} = 10\log N_a$, where N_a is the number of averages. Channel measurements over small spatial intervals of 3 cm in indoor environment have shown that the sounder performance is very high and it is capable of detecting small changes in the channel.

This sounder will be invaluable in the study of wideband channels that support current and future broadband wireless systems. The design approach has resulted in more than 60% reduction in the weight of the sounder compared to similar systems used elsewhere [23]. Channel sampling rates of 5.4 kHz over 200 MHz bandwidth have been achieved with the capability of up to 15 kHz with 100 MHz bandwidth signal transmissions. The sounder is ideally suited to the study of mobile wireless channels because of its light weight and fast channel sampling capability. The use of a faster device such as Xilinx Virtex 6 or 7 Series FPGA will boost not only the channel sampling rates but allow for more extensive real-time processing to be carried out.

References

[1] P. Kyösti, J. Meinilä, L. Hentilä et al., "WINNER II Channel Models," D1.1.2 V1.1, Information Society Technologies, November 2007.

[2] C. A. Middleton and A. Bryne, "An exploration of user-generated wireless broadband infrastructures in digital cities," *Telematics and Informatics*, vol. 28, no. 3, pp. 163–175, 2011.

[3] N. Savage, D. Ndzi, E. Vilar, and J. Austin, "Comparison of wideband indoor channel characteristics along two corridors with different wall surfaces," in *UK National USRI Symposium*, Mavern, UK, December 2001.

[4] A. G. Siamarou and M. Al-Nuaimi, "A wideband frequency-domain channel-sounding system and delay-spread measurements at the license-free 57- to 64-GHz band," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 3, pp. 519–526, 2010.

[5] I. Parez, S. Loredo, L. Valle, and R. P. Torres, "Experimental estimation of wideband radio channel parameters with the use of a spectrum analyzer and the Hilbert transform," *Microwave and Optical Technology Letters*, vol. 34, no. 5, pp. 393–397, 2002.

[6] J. Austin, W. P. A. Ditmar, W. K. Lam, E. Vilar, and K. W. Wan, "A spread spectrum communications channel sounder," *IEEE Transactions on Communications*, vol. 45, no. 7, pp. 840–847, 1997.

[7] A. Hewitt and E. Vilar, "Sective fading on LOS microware links: classical and spread_spectrum measurements techniques," *IEEE Transactions on Communications*, vol. 36, no. 7, pp. 789–796, 1988.

[8] Hunt Engineering, "Choosing FPGA or DSP for Your Application," 2010, http://www.hunteng.co.uk/info/fpga-or-dsp.htm.

[9] Mango DSP, September 2010, http://www.mangodsp.com/default.asp?id=11&item=24.

[10] M. Parker, "FPGA versus DSP design reliability and maintenance," November 2010, http://www.dsp-fpga.com/articles/id/?2207.

[11] D. C. M. Bilsby, R. L. Walke, and R. W. M. Smith, "Comparison of a programmable DSP and a FPGA for real-time multiscale convolution," *IEE Colloquium*, no. 197, pp. 4/1–4/5, 1998.

[12] P. Mu, M. Raulet, J.-F. Nezan, and J.-G. Cousin, "Automatic code generation for multi-microblaze system with syndex," in *Proceedings of the 15th European Signal Processing Conference (EUSIPCO '07)*, Poznañ, Poland, September 2007.

[13] P. Naraine, "Direct modulation radio hardware architectures for 3G communications systems," RF Design (Defense Electronics), February 2005, http://rfdesign.com/ar/502rfdf2.pdf.

[14] Analog Devices, "AD12401 Data Sheet," 2010, http://www.analog.com/en/analog-to-digital-converters/ad-converters/ad12401/products/product.html.

[15] "Advanced Filter Bank Technical Description," V CORP Technologies, Inc., http://www.v-corp.com/afb.htm.

[16] Alpha Data Ltd, ADM-XRC-II Xilinx Virtex-II PMC Data Sheet, Alpha Data, 2010, http://www.alpha-data.com/adm-xrc-ii.html.

[17] Technobox Inc., 64-Bit PMC-to-PCI Adapter Card Data Sheet, 2010, http://www.technobox.com/cat3673.pdf.

[18] K. Godfrey, *Perturbation Signals for System Identification*, Prentice-Hall, London, UK, 1993.

[19] Y. Lee, Y. Choi, M. Lee, and S. Ko, "Performance analysis of bit-width reduced floating-point arithmetic units in FPGAs: a case study of neural network-based face detector," *EURASIP Journal on Embedded Systems*, vol. 2009, Article ID 258921, 11 pages, 2009.

[20] Xilinx Inc., "Virtex-II CLKFX Jitter Calculator," 2010.

[21] D. Ndzi, J. Austin, and E. Vilar, "Hyper-resolution indoor channel impulse responses: multipath components and k-factors," *Electronics Letters*, vol. 35, no. 9, pp. 698–699, 1999.

[22] D. L. Ndzi, N. Savage, and B. Gremont, "Spatial and temporal variation of wideband indoor channels," *International Journal of Antennas and Propagation*, vol. 2010, Article ID 735434, 11 pages, 2010.

[23] H. Farhat, G. Grunfelder, A. Carcelen, and G. El Zein, "MIMO channel sounder at 3.5 GHz: application to WiMAX system," *Journal of Communications*, vol. 3, no. 5, pp. 23–30, 2008.