BRUNEL UNIVERSITY

FPGA Embedded System for Ultrasonic Non-Destructive Testing

by

Lei Zhang

A thesis submitted for the degree of
Doctor of Philosophy

in the
School of Engineering and Design

February 2012

To my forever beloved parents.

Acknowledgements

I would first of all like to express my sincere thanks to my industrial supervisor, Professor Peter Mudge, my academic supervisor Professor Wamadeva Balachandran and Dr Abbes Amira for their advice and guidance during this research, as well as their encouragement and friendship. I would also like to thank Mr Robert Phillips at Phillips Consultant for sharing his knowledge for the hardware design.

I would like to acknowledge the TWI Ltd and Plant Integrity Ltd for the financial support for my research. My thanks go to my section manger Dr Paul Jackson who has been supportive for my study and my colleagues in the MK4 project design team for their cooperation in developing the Teletest®MK4 product. I also would like to thank everyone in the Long Range Ultrasonic section and the Plant Integrity Ltd for generously sharing their knowledge.

With deepest gratitude and love, I want to thank my dearest parents for their constant encouragement, love and support.

For there is too much thank to be assigned to each individual, I praise God for the fun and happiness that I enjoyed, as well as the struggle and frustration I experienced, and most importantly, the knowledge and confidence I achieved, which I am obliged to work ever harder to deserve.

Abstract

Long Range Ultrasonic Testing (LRUT) is an emerging ultrasound Non-destructive Testing (NDT) method. The LRUT is a variant of the conventional NDT approach. By using ultrasound guided waves (UGWs) , it is efficient in quick long range defect scanning, which is impossible with other traditional NDT techniques. Increasing numbers of requirements for quick long range testing have led to urgent need for the improvement of testing methods and the development of new testing equipment to help researchers in laboratory and help technicians in field inspection.

The market for multi-channel ultrasonic instruments is characterized by small volumes of sales and a high ratio of development costs to sale price. The consequence is that the investment required to develop an instrument is significant and upgrades and other changes to instrument specification or configuration are difficult for many manufacturers to justify. These factors are particularly relevant for long range guided wave technology, where the technology is still relatively new in the market place and the instrumentation has different characteristics from other type of ultrasonic systems. In order to support the design and manufacture of the Teletest®MK4 system, Plant Integrity Ltd has supported this study into the use of a field-programmable gates array (FPGA) based control circuits for the electronics, which allows the component count to be decreased and also permits the system to be modified via flexible reconfiguration of the FPGA. This has benefits of reduction of size, weight and power consumption of the electronics and provides a means of upgrading the product without expensive re-design of the hardware.

FPGA device has been the enabling technology and main thrust behind the evolution of many technologies, unleashing new opportunities for improving the performance of the LRUT equipment and providing more functions for industrial and academic applications. A novel system has been developed to provide multichannel waveform generation, data capture and signal processing for advanced lab-based research and faster-field testing. This system includes an FPGA to process multichannel data in parallel and to provide the flexibility of reconfiguration for various testing applications proposed by new research. The design was tested in pre-prototype hardware. Subsequent construction of the commercial prototype unit enabled successful operation of the design to be demonstrated. Implementation of the FPGA design reduced component count, PCB dimensions and power consumption. Though the Spartan$^{\text{TM}}$ 3A DSP FPGA and Xilinx ISE software have been used in this project, the concepts of the design

can be applied to other FPGA devices from other FPGA vendors using other design softwares.

Contents

List of Figures

List of Tables

Abbreviations

ADC : Analogue to Digital Converter

ASIC : Application Specific Integrated Circuit

BRAM : Block RAM (FPGA on-chp resource)

BSB : Base System Builder

CLB : Configurable Logic Block

CPLD : Complex Programmable Logic Device

CPU : Central Processing Unit

DAC : Digital to Analogue Converter

DAQ : Data Acquisition

DDR : Double Data Rate

DSP : Digital Signal Processing

DSP48 : A primitive optimized for DSP operation in some Xilinx FPGAs

DSPs : Digital Signal Processors

EDIF : Electronic Design Interchange Format

EDK : Embedded Design Kit

Ethernet MAC : Ethernet Media Access Controller

FBGA : Fine-pitch Ball Grid Arrays (a type of FPGA package)

FFs : Flip Flops

FFT : Fast Fourier Transform

FIFO : First In First Out (memory device)

FIR : Finite Impulse Response

FMC : Full Matrix Capture

FPGA : Field Programmable Gate Array

FSM : Finite State Machines

GPP : General Purpose Processors

HDL : Hardware Description Language

HT : High Tension

IBUF : Input Buffer (FPGA on-chp resource)

ICON : Integrated Controller

IIC : Inter-Integrated Circuit; generically referred to as 2-wire interface

ILA : Integrated Logic Analyzer

IO : Input/Output port

IOB : Input/Output Block (FPGA on-chip resource)

IP : Intellectual Property

ISE : Integrated Software Environment

ISP : In-System Programmable

LMB : Local Memory Bus

LRUT : Long Range Ultrasonic Testing

LSB : Least Significant Bit

LUT : Look-up table

MAC : Multiply and Accumulation

MPMC : Multi-Ports Memory Controller

MSB : Most Significant Bit

MSPS : Mega Samples Per Second

MUX : Multiplexer

NDT : Non-destructive Testing

NRE : Non Recurring Engineering

NV : Non-volatile

OBUF : Output Buffer (FPGA on-chp resource)

PAR : Place and Route

PCB : Print Circuit Board

PHY : Physical layer Transceiver

PIPs : Programmable Interconnect Points

PLB : Processor Local Bus

PLD : Programmable Logic Device

PSD : Programmable System Device

PSU : Power Supply Unit

RTL : Register Transfer Level

SDK : Software Development Kit

SDR : Software-Defined Radio

SDRAM : Synchronous Dynamic Radom Access Memory

SNR : Signal-to-Noise Ratio

SoC : System on Chip

SPI : Serial Peripheral Interface

SRAM : Static Radom Access Memory

SRL : Shift Register LUT

SRL16 : A 16-bit shift register in Xilinx FPGA

TOF : Time of Flight
UART : Universal Asynchronous Receiver/Transmitter
UGWs : Ultrasound Guided Waves
UT : Ultrasonic Testing
VHDL : Very high speed integrated circuits Hardware Description Language
XCL : Xilinx Cache Link
XST : Xilinx Synthesis Tool

Chapter 1

Introduction

1.1 Background to the project

This research project was carried out to support the design and development of the field-programmable gates array (FPGA) based embedded system for a new generation of Ultrasonic Guided Waves (UGWs) Non-destructive Testing (NDT) system (Teletest®MK4 system). It was required to use the latest electronic design technology to add receiver channels, and reduce cost, size, weight and power consumption compared with the previous version. The existing Teletest®MK3 system employs relatively low frequency UGWs for various Long Range Ultrasonic Testing (LRUT) applications such as pipeline inspection. There were several important reasons for upgrading the system. Firstly, the previous system was developed a decade ago and some of the electronic components have become obsolete, making it difficult to manufacture more systems. Secondly, although there is an FPGA resource in the previous system, it is very limited and no more functions can be added. Thirdly, growing market competition and customer expectations have created a demand for a new product with improved system parameters. The project can benefit the industry by providing a more portable and reliable test equipment for pipeline inspection; also providing a tool with more functionality and flexibility for scientists and researchers to carry out sophisticated experiments to investigate new LRUT methods.

The author has studied the state-of-the-art hardware design techniques and implemented improved hardware functions in order to meet the requirements of the new product. An FPGA based soft core processor is used to replace the previous stand-alone hardware microcontroller device to control the memory and peripherals in the system. A novel design has been developed for the multi-channel transmitters with reduced size and component count. A special Digital-to-Analogue Converter (DAC) device with configurable interpolation filter functions has been used to replace the parallel DAC device for smoothing transmitter waveforms without the necessity of increasing transmitter memory size, as well as reducing the FPGA pin count of the interface between FPGA and multiple DAC devices. An efficient and reliable multi-channel receiver system has also been designed, implemented in such a manner that fewer clock cycles are required for completing data acquisition, which makes it possible for meeting critical timing requirements for multi-channel data acquisition with real-time Digital Signal Processing (DSP) functions, such as accumulation and saturation detection. The system has been designed not only to meet the current hardware specifications, but also with the potential to increase the number of transmitter and receiver channels for a more complicated system.

In this chapter, section 1.2 compares conventional Ultrasonic Testing (UT) and LRUT technologies and introduces LRUT applications. Section 1.3 introduces the existing LRUT equipments. Section 1.4 provides fundamental concepts of FPGA embedded system, and briefly describes the system specification for implementing LRUT. Section 1.5 lists the main research aim and objectives. Section 1.6 explains the research methodology used in this project. The contributions to knowledge are given in section 1.7. The organization of the thesis is presented in section 1.8 to guide the reader through the document.

1.2 Introduction to LRUT

1.2.1 Conventional UT VS LRUT

NDT techniques are inspection methods whereby the condition of a construction or component can be investigated while left intact and undamaged. NDT is widely used in scientific and engineering applications, where inspections are carried out to find imperfections and for corrective action to be taken to prevent failures in service. Take pipe testing as an example; pipelines can be tested regularly in service for cracks and corrosion without having to be removed or cut.

The basic principle of UT is to use ultrasound to detect and locate discontinuities or geometric features in materials. Ultrasound is usually generated by a piezoelectric transducer, and the same transducer is used to receive the responses from the material under test.

In conventional UT, a piezoelectric crystal or ceramic is excited at its resonant frequency, generally ranging from 500 kHz up to 50 MHz, with the majority of industrial testing being performed between 1 MHz and 10 MHz. This resonant frequency is proportional to the thickness of the active element of which the transducer is made. By using resonance, high vibration amplitude may be produced for a given input voltage. This reduces the power requirement and circuit complexity of conventional UT instruments because they transmit a very short spike and therefore, the energy in the pulse is very small.

There are two, so-called bulk wave modes in conventional UT, the longitudinal mode and the transverse mode. Longitudinal waves propagate in the same direction of the vibration of the waves. Transverse waves, also referred to as 'shear waves', propagate in perpendicular to the direction of the vibration of the waves. The velocities of longitudinal wave and transverse wave can be derived from the material properties of the medium, assuming it is isotropic [1, 2, 3]. For instance, the typical longitudinal and transverse velocities in steel surrounded by vacuum are 5960 m/s and 3260 m/s respectively.

LRUT differs from conventional UT in several respects. Firstly, LRUT uses low frequency UGW, typically from 10 kHz to 100 kHz, which have low attenuation and can propagate tens of metres along the length of the component. To date, LRUT has successfully been applied using UGW to test up to 100 metres along pipelines from the location where the ultrasound transducers are mounted, while conventional UT can only test a few centimeters beneath the surface of the tested material due to high attenuation of the high frequency ultrasound. Secondly, LRUT uses an array of transducers to generate a tone burst wave with a certain waveform format rather than generating a pulse-wave at the resonant frequency of a single piezoelectric transducer as in conventional UT. This makes the system design for LRUT equipment more complicated than that for the conventional UT. Thirdly, many UGW wave modes exist in the tested specimen at the same frequency. The geometry and material of component and the frequency of UGW determine the wave modes generated. The UGW are dispersive during propagation. The velocity is a function of frequency. In LRUT, a frequency is selected according to the geometry and property of the tested specimen to generate a main wave mode and suppress the other modes in order to maximally avoid dispersion. An array of transducer segments are arranged into three rings with appropriately adjusted spacing between them to allow the excited tone burst to propagate in one direction. In pipe testing, these transducer rings encircle the whole circumference of the pipe to generate axial symmetric wave modes. A detailed description of UGW parameters that affect LRUT testing and define the requirements for the LRUT instrument is given in Chapter 2.

1.2.2 LRUT Applications

To date the most popular LRUT application is to inspect industrial pipelines to detect corrosion and defects. In LRUT, a PC-controlled instrument transmits low frequency (usually less than 100 kHz) pulses to the transducer array, which generates UGW along the specimen to be tested and captures signals reflected from discontinuities in pulse-echo, that is, the same transducers transmit the signals and also receive the reflected energy. The received signals are stored in the flaw detector unit and are then sent to the PC where a location vs. amplitude (A-scan) plot of the data is displayed. The axial location and approximate severity of the defect are determined by the time of flight (TOF) and the known geometric features such as welds and flanges.

The importance of LRUT for industrial pipeline inspection is due to its efficiency in long distance testing and capability of testing insulated and inaccessible lengths of pipes. The detection of corrosion of insulated pipe is problematic for conventional UT because many areas of the pipe are inaccessible; sometimes the costs of access can exceed the costs of inspection. LRUT provides rapid screening for in-service degradation of buried and insulated pipes, reduces the costs of gaining access and testing time by avoiding the removal and reinstatement of insulation or coating, except at location of transducer tool.

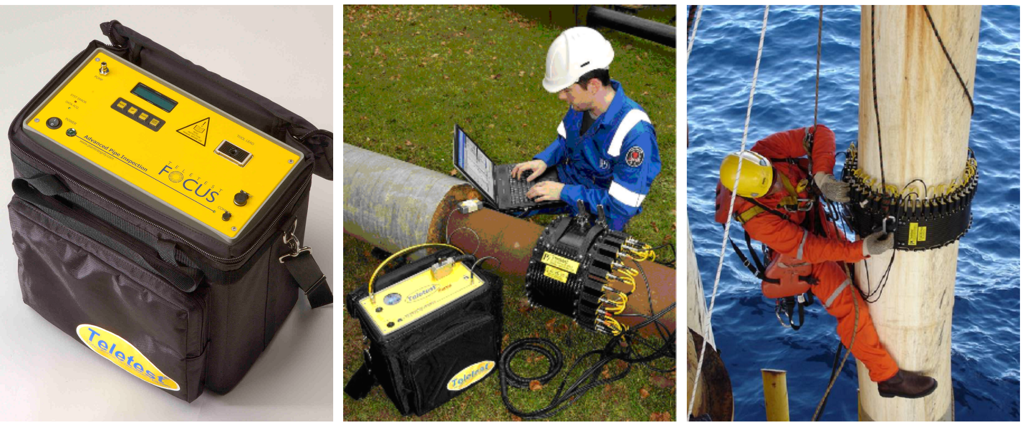
Research work is being carried out by others to investigate the potential of LRUT in many applications such as inspection and monitoring of railway rails, food and drinks pipelines, aging wiring in aircraft, offshore wind towers and blades and sub-sea risers. Due to the wide range applications of LRUT, commercial equipment is available from three manufacturers.

1.3 Introduction to LRUT Systems

Currently there are three commercial LRUT systems available for pipes and pipelines inspections: the Teletest®MK3 Focus System developed by TWI Ltd. and manufactured by Plant Integrity (Pi) Ltd.; the Wavemaker G3 system developed by Guided Ultrasonics Ltd.; and the MsSR 3030R developed by South West Research Institute (SWRI) in the USA and distributed in the UK by NDT Consultants Ltd. These systems have much in common as the wave propagation is based on the same physical principles.

One of the main features of these systems is the need of multiple channels for both transmit and receive functions and the corresponding control electronics. In order to transmit and receive multi-channel UGW signals in parallel, these systems require high speed parallel DAC, Analogue-to-Digital Converter (ADC) and memory interfaces, and an FPGA is an appropriate solution for providing both control logic and sufficient pins for interface connections.

Teletest®was the first commercial product on the market, first released in 1998 by Pi Ltd. It has been through a number of developments. In 2002 Rose and Mudge presented a method of using the multi-channel capability to enable focusing of the low frequency waves and thereby enhance the performance of the test[4]. The Teletest®MK3 equipment has 24 transmit channels and 8 receive channels. It is able to measure the severity of metal loss above 3% of the cross sectional area of a pipe. Figure 1.1(a), 1.1(b) and 1.1(c) show the Teletest®MK3 electronic unit and the system used in inspection of an insulated pipeline and off-shore riser respectively.



(a) Teletest®MK3 System Electronic Unit

(b) Teletest®MK3 System used in inspecting insulated pipeline

(c) Teletest®MK3 System used for testing off-shore riser

Figure 1.1: Teletest®MK3 FOCUS System

As a part of the process of continuous improvement of the Teletest®product and to take account of both feedbacks from customers and the requirement to implement more advanced LRUT test procedures, a project has been running for the past 3 years to produce a redesigned, enhanced and re-packaged electronics unit to replace the MK3. The design brief for the new electronics unit was for it to be smaller and lighter than the previous version, to consume less power but to have greater flexibility in order to be able to implement both developing and envisaged improved LRUT procedures for a variety of applications.

In the new system design, there was a conflict between the essential performance requirements of the device and the size, weight and complexity of the electronic unit. The number of independent channels required dictates the number of specific circuits which are required, for example the number of transmitters and receivers, but in addition to these there is an increasingly complex control requirement as the number of channels increases. This is needed, amongst other things, to turn each channel on and off, set gain, sequence the tests and allow test data to be saved. There was also the requirement for on-board memory to store test data. This added to the overall size, weight and cost of the instrument. Figure 1.2 shows the number of components required for control of a multi-channel system as the number of channels increases. It may be seen that the number of control and peripheral devices (i.e. ADCs, DACs and memory) increases dramatically as more channels are added to the system.

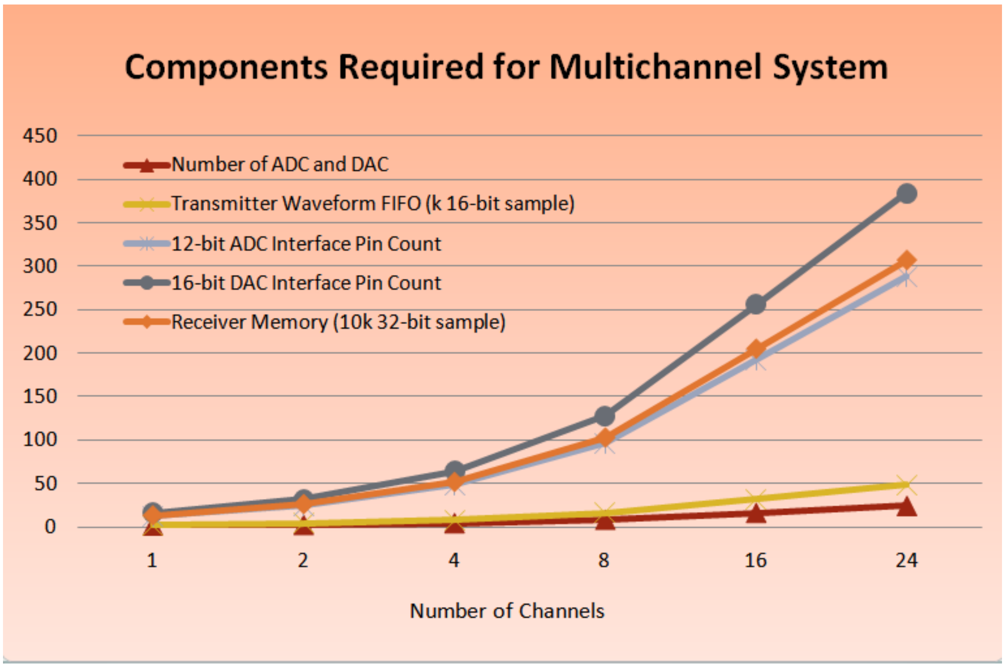


Figure 1.2: Increase in the number of components and pin-count for control of a multi-channel ultrasonic system as the number of channels increases

The author's role was to implement control and logic functions for the new unit. This entailed the use of innovative ways of reducing component count and power consumption, while increasing the capability of the system. At the same time, the development was subject to the usual commercial constraints of time and funding. Consideration of these requirements led, firstly to the decision to use an FPGA device to perform many of these functions and, secondly, to the novel aspects of implementation reported here.

1.4 FPGA Based Embedded System for LRUT

An FPGA is an electronic device consisting of many logic blocks and memory elements. The logic block can be programmed to perform add, multiply, and other complex logical functions. The decision of using FPGA in the developments reported in this thesis is based on the flexible functions and advantages of FPGA devices over the other solutions, which are reviewed in Chapter 3.

1.4.1 Embedded System Overview

An embedded system is a special-purpose computer system designed to perform one or a number of dedicated functions within an electronic system [5]. Embedded systems can be optimized for special tasks to reduce the size and cost of the system, as well as to increase the reliability and enhance the performance of the system. The architecture of an embedded system is defined by the requirements and user specifications. The embedded system can be designed using various control devices, such as microprocessor, microcontroller, Digital Signal Processors (DSPs), Application Specific Integrated Circuit (ASIC) and FPGA. Microprocessors can be reconfigured by software and are good for computation applications. Microcontrollers combine microprocessor and peripherals together; and can be used for system-on-chip (SoC) applications. DSPs are optimised for intensive DSP applications. ASICs are used for specific applications requiring high volume products. FPGAs have the flexibility to combine the strengths of processor, controller, DSPs and can be used to target a wide range of applications requiring low to medium volume products. These devices can also be combined in a single system to enhance system performance. For instance, the Teletest®MK3 system was designed combining a microcontroller for controlling peripherals and an FPGA for high speed multi-channel data transmitting and receiving.

Modern single-chip microcontrollers can be used to create an embedded system with fewer hardware components by integrating peripherals such as ADC, timer, and communication ports such as Universal Asynchronous Receiver/Transmitter (UART), Serial Peripheral Interface (SPI) and Inter-Integrated Circuit (IIC) on a single chip along with the microprocessor. The microcontroller-based system is adequate for most embedded system applications with simple functionality. However, for the Teletest®system, custom logic is needed for fast controlling of the transmitting and receiving of multi-channel data, and a number of additional peripherals are required.

In the last two decades, the density of 'slices' (logic cells) in FPGAs has increased from a few thousands gates to tens of millions. FPGA-based systems can now integrate digital logic design, processors and communication interface on one chip. Typical embedded systems based on FPGA are tuned for low cost, minimum power consumption, relatively high speed and minimum FPGA resource utilization. A major advantage of an FPGA is the combination of the functions of processor, controller and DSP. Stand-alone microcontrollers are a mature technology and are relatively more power and cost efficient than an FPGA, but an FPGA can offer more advantages such as flexible configurability with required peripherals, portability of code among various vendors, code reusable IP libraries and low cost programming software.

Therefore, FPGA is the most suitable solution for Teletest®system because it can achieve the necessary performance at a much lower design cost and provides the reconfiguration flexibility for the system prototype design. By using microcontroller implementation using the FPGA on-chip resource, there is no need to purchase an expensive off-the-shelf microcontroller that has a large number of peripherals, some of which are not required by this system. The FPGA allows the designer to tailor the design and only add the essential peripherals needed and also build any peripheral unavailable from an off-the-shelf microcontroller.

A typical embedded system design flow involves the following steps: defining the functional and non-functional requirements; producing system specifications for hardware components such as processor, peripherals, custom logics and user interfaces, as well as software programs for system controlling; system design for both hardware and software; system hardware and software integration for debugging and verification. In the design of Teletest®MK4 system, the system requirements are the tasks needed to be carried out by the system for the LRUT application, which is described in Chapter 2. The author had to work within system specifications and industrial constraints in terms of cost and predefined functions which already existed in Teletest®MK3 system. The author was given the task of producing the FPGA embedded system design which is integrated with the software programmed by software engineers in the project design team. The author was also involved in the debugging and verification of the designed prototype and the testing of the final product.

1.4.2 Teletest®MK3 Embedded System

The remote electronic unit of the existing Teletest®MK3 system is a typical embedded system composed of a microcontroller for over-all system control, 24 channel transmitters for generating and transmitting UGW, 8 channel receivers, on-board memories for data acquisition, communication link for uploading the received data to PC for signal analysis. Figure 1.3 is a block diagram for the Teletest®MK3 system. The details of the Teletest®MK3 hardware components and architecture is given in Appendix A.

The 8-bit 8051 microcontroller running at about 14 MHz provides overall control of the embedded system. It receives commands from the PC over the data communication link, and sets up and controls the various hardware elements of the unit to carry out the desired operation. After an acquisition is completed, the microcontroller needs to upload received data to the remote PC via an ArcNet communication interface. The overall testing speed is very much limited by the slow speed microcontroller. It is necessary to use a faster microcontroller to increase the system speed. However, a faster stand-alone microcontroller would increase the power consumption, board area and PCB routing complication. It has been found in literature [6] that an embedded system can be successfully implemented on a single FPGA chip with integrated microprocessor soft core or hard core on-chip, which overcomes the problems stated above and adds flexibility to the system. However, the low profile FPGA used in Teletest®MK3 system is an old generation FPGA device with limited functionality and does not have enough resource for implementing this function; neither can it be used for interfacing more receive channels in the system because the input/output (IO) pins of this FPGA are very limited. Moreover, it is a common problem in electronic design that old components become obsolete or are not supported by the new version design tool for latest functions, which puts risk into manufacturing and highly constrains the future upgrading of the commercial product. Therefore, it was necessary to replace the old FPGA with a high profile, up-to-date FPGA that

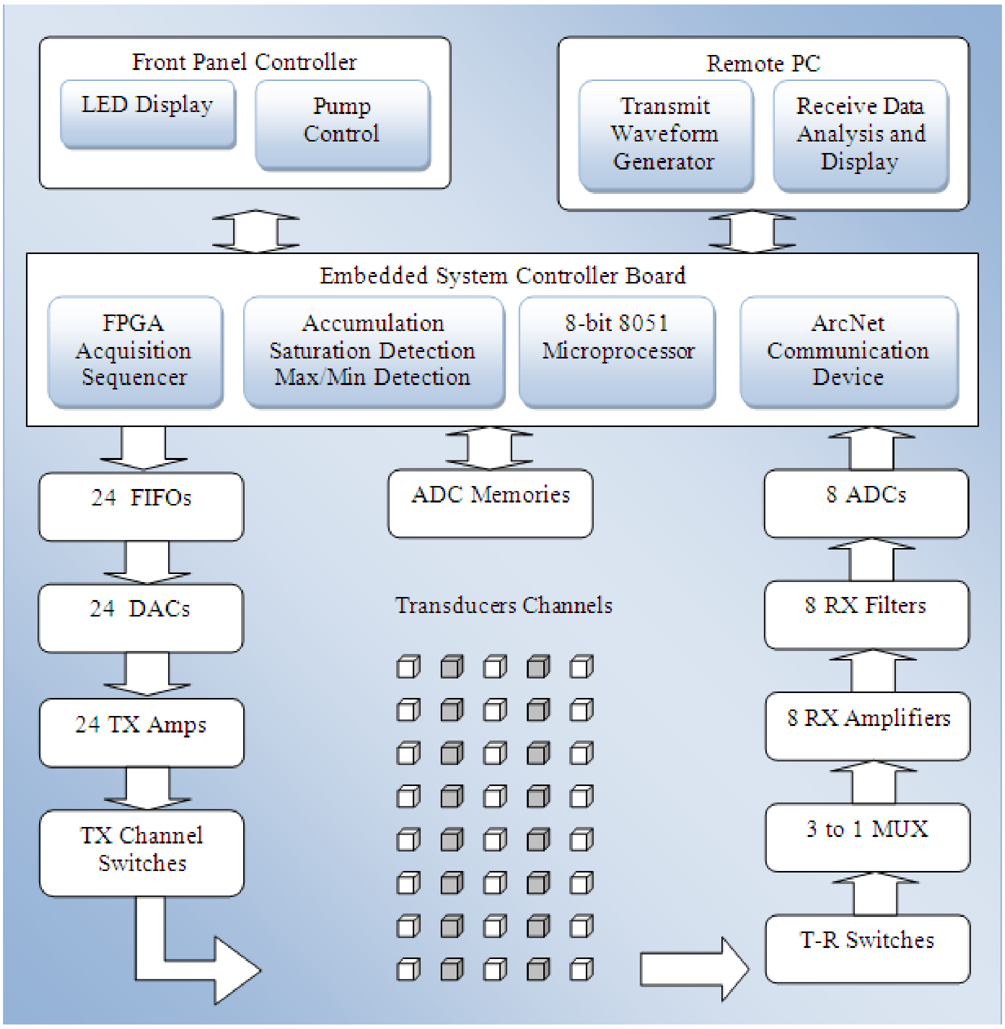

Front Panel Controller
LED Display
Pump Control
Remote PC
Transmit Waveform Generator
Receive Data Analysis and Display
Embedded System Controller Board
FPGA Acquisition Sequencer
Accumulation Saturation Detection Max/Min Detection
8-bit 8051 Microprocessor
ArcNet Communication Device
24 FIFOs
ADC Memories
8 ADCs
24 DACs
Transducers Channels
8 RX Filters
24 TX Amps
8 RX Amplifiers
TX Channel Switches
3 to 1 MUX
T-R Switches

FIGURE 1.3: Teletest®MK3 System Block Digaram

would be capable of implementing the required improvements of the system and would also be compatible to more advanced FPGA devices in the same product family to provide capability for future upgrading.

It should be noted that although many FPGA devices were potentially available for this project, the remit was to implement the necessary functionality within a certain budget. Therefore, a very low cost, low power consumption Spartan 3A DSP FPGA was chosen and a microprocessor soft core has been implemented with a total cost of about £80 per device, rather than a high profile FPGA such as a Virtex®device with integrated PowerPC hard core which could cost above £1,000 per device.

1.5 Research Aim and Objectives

The main aim of the research is to design and develop the FPGA embedded system to fit into the requirements of the Teletest®MK4 system. Meeting these requirements entailed the development of novel applications within the FPGA, so that the aim of this research is to ensure that the functionality achieved in the FPGA controller was sufficient for successful operation of the system as a whole. This research aim is reflected and expanded in each chapter throughout the whole thesis. Specific objectives of the research are given below:

- To review embedded system design using FPGA technology and to define the role of the FPGA device in the Teletest®MK4 system.
- To design and develop an FPGA based embedded system which would effectively implement the LRUT technology and enhance the over-all test performance, as well as reduce the size, weight, cost and power consumption and increase the speed, portability and reliability of the LRUT system.
- To evaluate the performance of the FPGA system against the stated specifications.
- To analyse the test results on the prototype system and provide suggestions for further improvements in system control.

1.6 Research Methodology

The research methodology adopted in the project is listed below.

Step 1 was to review the FPGA related functions and performance requirements of the Teletest®MK4 system. The system needs to control a large number of peripheral devices. Some of these devices such as Ethernet, Double Data Rate (DDR2) Synchronous Dynamic Radom Access Memory (SDRAM) and Flash memory are controlled by the MicroBlaze soft microprocessor in the FPGA. These devices are interfaced to the controller via a common peripheral bus and the controller can access only one device at a time. The other devices such as DACs, ADCs and Static Random Access Memory (SRAMs) are controlled synchronously to allow the system to transmit and receive multi-channel UGW signals in parallel. These devices have different interface specifications for the speed and timing constraints. Different controllers have to be implemented on the FPGA to control these devices. The hardware architecture of the MK4 system was generated based on the FPGA and the FPGA controlled peripheral devices.

Step 2 was to review the constraints imposed directly on the FPGA. The system specification requires reducing the size, weight, power consumption, component count and cost of the system and meanwhile increasing the speed, and the number of receiver channels. The project budget constrains the cost and resources of the FPGA that can be used for the design.

Step 3 was to investigate the latest FPGA and embedded system design methodology and to estimate the possibility for the design and development of a new FPGA embedded system to meet the specification of the Teletest®MK4 system. The previous three steps led to the selection of the low cost and low power Spartan 3A DSP 1800A FPGA device to make the required improvement and meet the budget.

Step 4 was to implement the individual functions required by the MK4 system specification on the off-the-shelf development boards for the selected FPGA, DAC, ADC, SRAM devices in order to evaluate the design concepts and the control strategy. The FPGA has limited on-chip resources. Therefore an efficient FPGA design including a central controller, peripheral interface controllers and other real-time functions has to be designed to meet the specifications.

Step 5 was to develop the Teletest®MK4 prototype boards and to test the performance of the integrated system, analyse the test results against the system requirements.

1.7 Contributions to Knowledge

Recent published literature on the LRUT technique has identified the necessity for advanced LRUT equipment[7, 8]. FPGA devices have been widely used for prototype development of new medical and consumer products because of their advantages over the other types of programmable logic devices (PLD). There have been published FPGA designs including transmitter and receiver control for various applications such as video optical transmitters[9] and software-defined radio (SDR)[10]. These FPGA designs are either beyond the project budget of the Teletest®MK4 system or are not designed for the frequency range of LRUT using UGW. To date there are no publications addressing the detailed design and development of multi-channel transmitters and receivers controlled by an FPGA embedded system which can be directly adopted for the LRUT equipment according to a literature review. Therefore a new system has to be designed to meet the specific requirements of the LRUT applications. In this work novel multi-channel LRUT system design and implementation have been achieved using an FPGA embedded system for control. This system incorporates reconfiguration flexibility to meet requirements of various LRUT applications, as well as to support future research on the potential of improvement of LRUT. The operation of the FPGA controller has been successfully tested in the prototype Teletest®MK4 system.

Specific contributions to knowledge are:

FPGA-based embedded system design concept using VHDL, permitting reconfigurable overall system control functions

A literature review on FPGA-based embedded systems has shown few publications on a complete prototype FPGA-based embedded system design of a high speed multi-channel commercialized product. A vast number of industrial products have been designed and developed, but the design information is not usually available. VHDL is short for VHSIC Hardware Description Language which describes digital circuits. VHSIC stands for Very High-speed Integrated Circuit. The FPGA-based embedded system design concept adds flexibility and reconfigurability to the system prototype and allows new functions to be added to the system to support research of future LRUT applications.

Multi-channel arbitrary waveform generator using interpolation DAC and efficient use of FPGA BRAM and control of multi-channel transmitters

1. FPGA Block RAM (BRAM) has been used to replace the external first in first out (FIFO) memory which stores transmitter waveforms. This reduced component count and PCB board layout complexity, as well as reduced system size and power consumption. Since the BRAM is located inside the FPGA, the access speed is very fast, up to 250 MHz clock frequency. However, the transmitter memory has to be reduced from 16 k samples to below 2 k samples for each channel because of the limit of the FPGA on-chip BRAM. FPGA with bigger BRAM generally has larger size and is more expensive. The effect of this is that the generated waveforms have fewer

samples per cycle and the voltage step between each two samples becomes larger, causing sudden current spike in the transmitter circuit. In order to produce smooth transmit waveforms from a limited number of samples, an interpolation function was introduced and a special DAC device with integrated interpolation function was used in the system.

2. The interpolation filter in the DAC device has four interpolation rates (x2, x4, x8, x16) and can interpolate one digital sample generated by FPGA up to 16 times. In this manner, 1k samples with x16 interpolation is equivalent to 16k samples without interpolation. This DAC device also has a serial data interface which needs only one FPGA pin per channel as digital data output port. For a 16-bit DAC with parallel data interface, 16 FPGA pins are needed for the data ports of each channel, and 24 channels will need 144 FPGA pins. Using a serial port device allows more transmitter channels to be connected to FPGA with reduced FPGA pin requirement. This combination makes the system more compact, reduces the size, complexity and power consumption of the circuit. This design can also be used in general applications requiring multi-channel signal generators.

Control of multi-channel receivers for data acquisition, including control of ADCs, SRAM and real-time accumulation of data

The Teletest®MK4 system needs to receive signals from 24 channels in parallel at up to 1 Mega sample per second (MSPS) for all channels. Data from multiple acquisition iterations are accumulated on all channels in order to eliminate environmental noise. In addition, ADC samples for all channels are compared to pre-set upper and lower threshold values to detect saturation of the ADC devices. Therefore, two real-time functions need to be implemented on all 24 channels. A comparison of various types of memory has shown that SRAM has fast access speed and low power consumption. SRAM was selected to store captured data during data acquisition. A three-bank SRAM memory structure was designed to meet the critical time requirement for 24-channel receiver system. Research revealed that the general asynchronous SRAM controller needs at least 3 clock cycles to complete one read or write access. In the accumulation function, one sample is read from one SRAM location, added with the sample captured by ADC, and written back to the SRAM, all of which takes at least 7 clock cycles. A custom SRAM controller was designed and implemented on the FPGA, in which a read-add-write sequence is completed within 2 clock cycles. It is more efficient than general controllers. It is possible to meet the time requirement using general

SRAM controller by increasing the clock frequency, with the cost of higher power consumption. Higher system clock frequency also causes strict time parameters and increased FPGA place and route complexity, which increases the synthesis and implementation time of the FPGA design tool.

1.8 Organization of the Thesis

Chapter 2 discusses the basics of LRUT technology, provides a survey of existing LRUT systems with comparison of their published features and detail specification of the Teletest®MK3 system; Chapter 3 introduces the concept of FPGA-based embedded system design and the benefits it can bring to the LRUT instrument; Chapter 4 proposes a new FPGA-based embedded system for LRUT, explains the system hardware requirements for implementing LRUT using an FPGA; Chapter 5 describes the design and implementation of an improved multi-channel arbitrary waveform generator; Chapter 6 describes the design and implementation of a multi-channel data acquisition system optimised for LRUT; Chapter 7 presents and analyses the implementation results of the FPGA design and compares the implementation results of different optimization strategies; Chapter 8 concludes the research in this thesis and suggests on future studies.

Chapter 2

Ultrasound Guided Waves Pipeline Inspection

In this chapter, section 2.1 provides a background review of using UGW for pipeline inspection; section 2.2 identifies the main system functional requirements and configuration; section 2.3 reviews the existing LRUT systems, compares their specifications; section 2.4 describes the structure and section 2.5 describes functions requirements of the Teletest® MK3 system; section 2.6 explains the current limitations of the Teletest® system and the planned improvements for future instrument that requires the development of an underpinning FPGA-based embedded system. As the main objective of this project is to support the design of an instrument to meet industrial and scientific research requirements for implementing LRUT technology, the principles of guided waves characteristics and the mathematics for topographical reconstruction are beyond the scope of this project. This chapter is necessary for understanding the fundamental knowledge of LRUT technology and the functional requirements of the testing instrument. Based on this knowledge, Chapter 3 provides a survey of FPGA embedded systems to gather information to support the design and development of a new Teletest® system, which is proposed in Chapter 4.

2.1 LRUT Background

UGW makes it possible for rapid screening of tens of metres of pipelines from a single test location. This is especially useful for situation where external access to the pipe is limited, such as insulated pipelines, buried road crossings, offshore risers submerged in water etc. LRUT technology is based on the potential that UGW can propagate axially long distances from the excitation location [11]. The first solution to wave propagation in hollow isotropic cylinders was published by Gazis in 1959 [12]. Axisymmetric longitudinal UGW was used by Mohr and Höller [13] to detect transverse defects, and torsional UGW was used to detect longitudinal defects. Finch studied both flexural and longitudinal UGW in hollow cylinders [14], and Kumar investigated UGW in hollow cylinders with liquid content [15, 16], which is especially useful for in-service pipeline inspections. The use of axisymmetric UGW is an attractive solution for pipeline inspection. Commercial products [17, 18, 19, 20] have been successfully developed to demonstrate this technology for both scientific and industrial purposes. Bessel functions have been used for fully calculating dispersion relationships of UGW [21]. Lowe [22] improved the calculation of dispersion curves by using the global matrix method to solve the wave propagation equation.

2.1.1 UGW Modes

A wave mode is the characteristic displacement pattern of the ultrasound as it propagates in an acoustic medium. The propagation of UGW is constrained by the boundaries of finite structures; therefore, the sound propagation is complex and a large number of wave modes exist. A wave mode in a pipe is generally defined as $X(n, m)$, where X refers to Longitudinal (L), Torsional (T) or Flexural (F). n is the circumferential order of the mode, which refers to the cycle of sinusoidal variation of the displacement pattern around the circumference. When n equals zero, there is no circumferential variation and the wave mode is axisymmetric. m is an index, indicating the number of wave modes in the acoustic medium. This index is the order that a mode comes into existence at a given circumferential order n. An increasing number of modes are generated with increased frequency. In pipeline inspections, cylindrical wave modes are mainly used. Longitudinal modes are axisymmetric and referred as $L(0, m)$. Torsional modes are also axisymmetric and referred as $T(0, m)$. All non-axisymmetric wave modes are classified as Flexural and referred as $F(n, m)$.

2.1.2 Dispersion Curves

The Phase velocity (Vph) of a wave is the velocity at which a point of phase propagates through a medium, i.e, The phase velocity is constant for every point in a continuous sinusoid at any one frequency. Group velocity (Vgr) is the apparent velocity with which a pulse travels through a medium. The phase velocity of a certain mode UGW depends on the selected frequency. Vph, Vgr and frequency (f) are related by equation (2.1).

$$V_{gr} = \frac{V_{ph}}{1 - \frac{f}{V_{ph}} \cdot \frac{dV_{ph}}{df}} \tag{2.1}$$

A continuous sinusoid in the time domain has a single frequency component. In LRUT, a short burst consisting of a number of cycles at a certain frequency is transmitted. This can be seen as a continuous sinusoid with a window function applied. The truncation of continuous signal in the time domain increases the bandwidth of the signal. Different frequency components in a guided wave pulse may propagate at different group velocities, causing the effect that the transmitted pulse tends to spread out over time. This is called dispersion, and the presence of dispersive effects is an important difference between LRUT and conventional UT. The frequency-velocity diagrams of various wave modes are referred as dispersion curves. In LRUT pipeline inspection, the group velocity depends on frequency, wave mode, pipe diameter, wall thickness and pipe contents. Figure 2.1 illustrates dispersion curves of four wave modes for a ferritic steel pipeline with 22 inches diameter, 28.58 mm wall thickness. It may be seen that Torsional $T(0, 1)$ is non-dispersive, with a constant propagation velocity at any frequency. Longitudinal $L(0, 2)$ only comes into existence above 20 kHz, and becomes nearly non-dispersive above 40 kHz. Both Longitudinal $L(0, 1)$ and Flexural $F(1, 3)$ are highly dispersive. Therefore, the former two modes are commonly used for pipe inspection.

2.2 LRUT System Function Requirements

The multiple modes and dispersive behaviour mean that the use of UGW in LRUT applications is more complicated than conventional UT. Therefore, the LRUT requires more sophisticated test apparatus that is capable of implementing LRUT functions. Table 2.1 lists the tasks that need to be performed by the system designed for LRUT.

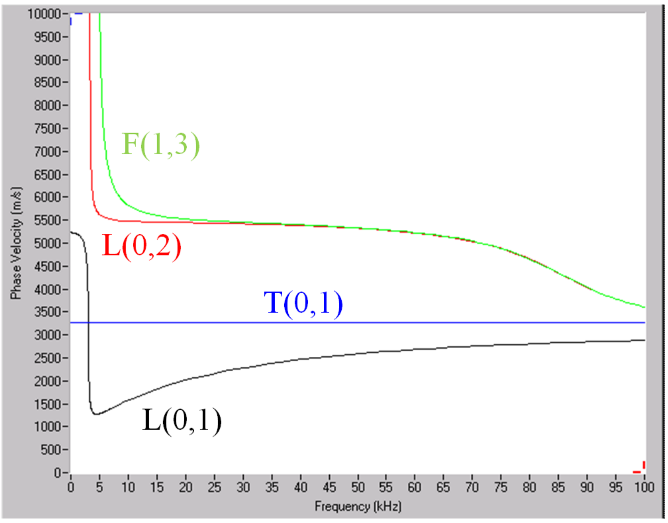


FIGURE 2.1: Dispersion Curves for a 22 inches diameter, 28.58 mm wall ferritic steel pipe

TABLE 2.1: Tasks of Teletest® System

Task	**Description**
Transducer Array	An array of transducers need to be housed in pneumatic mechanical structure and arranged in multiple rings round the tested specimen.
Waveform generation	Arbitrary waveforms generated from PC and loaded to system memories for selected transmit channels. Digital to analogue conversation of the stored waveforms. High voltage amplification of the analogue signals. Switching the wave between two sets of transducer arrays for longitudinal or torsional testings. Drive the corresponding piezoelectric transducers.

Task	**Description**
Data Acquisition	Capture analogue signal on selected receive channels. Filter the received analogue signal to remove noise. Amplify the received analogue signal before digitization to fill the ADC conversion range in order to increase the accuracy. Sample and digitize the analogue waveforms and store the data in the embedded system local memory. Accumulation and Averaging: A repetition number can be set up to allow multiple transmit and receive iterations in order to reduce noise. Transfer the captured data from the embedded system to remote PC via Ethernet communication link.
Transmit Receive Switching	The transducers are required to be switched in rapid succession from transmission to acquisition.
Data Analysis	Data analysis is carried out by PC software to extract the TOF measurement values and display the received data for multiple receiver channels.

2.3 Mainstream LRUT Systems

There are three commercial systems that employ UGW for LRUT inspections. The functionality of these is similar, as it is determined by the guided wave characteristics described above. The main specifications of these three systems are listed and compared in Table 2.2. In order to generate axisymmetric UGW to cover the pipe circumference for pipe inspections, Teletest® and GUL use an array of lead zirconate titanate (PZT) Transducers. The transducer array transmits a signal along the pipeline, where any changes in acoustic impedance caused by structure discontinuities such as welds, flanges, corrosion and defects cause incident wave to convert from one mode to various other modes, which is reflected back and captured at the transducer array. The location of discontinuities is calculated by the measurement of the Time-of-Flight (TOF). The PZT transducers are arranged in several transducer rings and pneumatically coupled to the pipe surface using a collar encircling the tested pipe. Non-symmetric flexural wave modes are unavoidably generated due to the difference of individual PZT transducers in the transducer array, which are discretely distributed. The bracelet tool configuration can reduce this effect. Flexural modes increases unwanted noise at the same frequency as the generated UGW and cannot be removed by traditional band-pass filters. MsSR uses a thin ferromagnetic strip. In this case, more axisymmetric UGW can be generated theoretically, but the non-evenly attachment of the ferromagnetic strip to the pipe surface can still cause flexural mode to be generated.

2.4 Teletest® MK3 System Structure

This section uses the system structure of Teletest® MK3 system as a reference to further explain the system requirements for LRUT. As illustrated in Figure 2.2, the Teletest® MK3 system is composed of a mechanical structure that houses the PZT transducer arrays, a remote electronic unit with an embedded system built in for ultrasound transmission and data acquisition, and a laptop PC installed with Teletest® control software for data analysis.

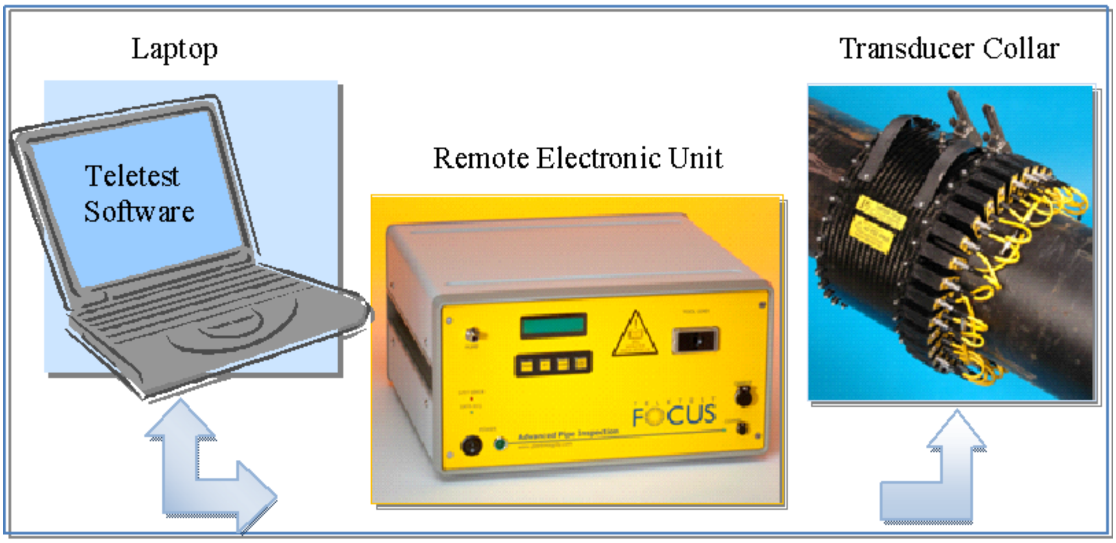


FIGURE 2.2: Teletest® MK3 System

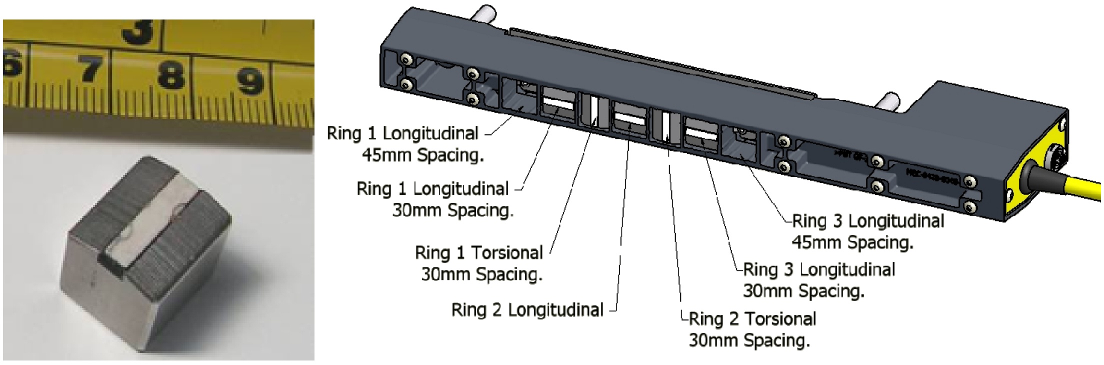
Table 2.2: Comparison of Existing LRUT Systems

Features	**Teletest® MK3 Focus®**[1]	**GUL Wavemaker G3®**[1]	**MsSR 3030®**[1]
Transducers	PZT	PZT	Magnetostrictive Coil
Number of transducer channels	40 (24 longitudinal & 16 torsional)	32	4
Number of transmitters	24	16	2
Number of receivers	8	16	2
Sampling Frequency	1 MHz	200 kHz	N/A
Waveforms	Arbitrary Waveforms	Arbitrary function Generator	Sine or square waves
Maximum Output Voltage	300 Vpp	400 Vpp	300 Vpp
Maximum Output Current	3 Amp	N/A	40 Amp
Output Attenuation	0 to 100% in 1% steps	N/A	0 to 100% in 20% steps
Wave mode	Longitudinal & Torsional	Torsional	Torsional
Frequency range	20 kHz to 100 kHz	15 kHz to 75 kHz	4 kHz to 250 kHz
Receiver Gain Range	20 to 100 dB in 1dB steps	10-120 dB	40 to 110dB in 2dB steps
Receiver Filters	Analogue low pass filters: switchable at 16, 32, 75, 150, 300, 600 kHz; Analogue high pass filter: switchable at 1, 5 kHz	Analogue high pass filters: Switchable at 5,10,20,40 kHz (minimum 4 pole);	Analogue low pass filter: 98 kHz (minimum 8 pole) 8 individual 4-pole active band pass filter modules
Maximum number of averages	4096	256	N/A
Inspection range/Maxium number of sample points	64k	128k (approx 950m)	150 meters
Weight	12 kg	8 kg	14.3 kg
Dimensions	41 x 21 x 31 cm	44 x 14 x 40 cm	46.3 x 47.1 x 14.7 cm
External power supply	16 VDC	19 VDC	N/A
Run time on one battery charge	12 hours typical use	10 hours typical use	N/A
Communication protocol	Ethernet	USB	N/A
Software	Teletest® WaveScan	WaveProG3	N/A

[1] Details obtained from commercial literatures.

2.4.1 Transducer Array and Mechanical Structure

Transducers convert electronic signals into mechanical waves and vice versa. Teletest® system uses specially manufactured transducers to generate and capture ultrasound waves with a wide frequency range from 10 kHz to 100 kHz for LRUT testing. Multiple transmit and receive channels (24-channel transmitters and 8-channel receivers are used for Teletest® MK3 system) are required to drive a large number of transducers in order to excite axisymmetric UGW that are uniform around the pipe circumference for even inspection in LURT. Because the capacitive load of piezoelectric elements from which the transducers are made varies, each individual transducer has a different efficiency in transmitting and receiving ultrasound. It is necessary to have normalized transmitter waveforms for transducers to generate axisymmetric UGW, otherwise unwanted Flexural modes would be generated which would make it difficult to diagnose the LRUT test results. Signal compensation has to be made on both transmitter and receiver channels in the electronic designs in order to normalise the performance of transducers and increase the efficiency.



(a) Teletest® Transducer

(b) Teletest® Transducer Module

FIGURE 2.3: Teletest® Transducer and Transducer Module

Figure 2.3(a) shows a piezoelectric transducer. Figure 2.3(b) shows a single module populated with five transducers, three for longitudinal waves and two for torsional. In pipe inspection, three rings of transducers are required for longitudinal mode and two rings of transducers are required for torsional mode. These arrangements are used to cancel out one direction transmission as well as to suppress unwanted UGW modes.

Figure 2.4 illustrates the transducer interface of the 24-channel transmitters. The Teletest® MK3 system has 24 transmitter channels, 16 out of which are switchable to drive two sets of transducers arranged for longitudinal and torsional mode, and

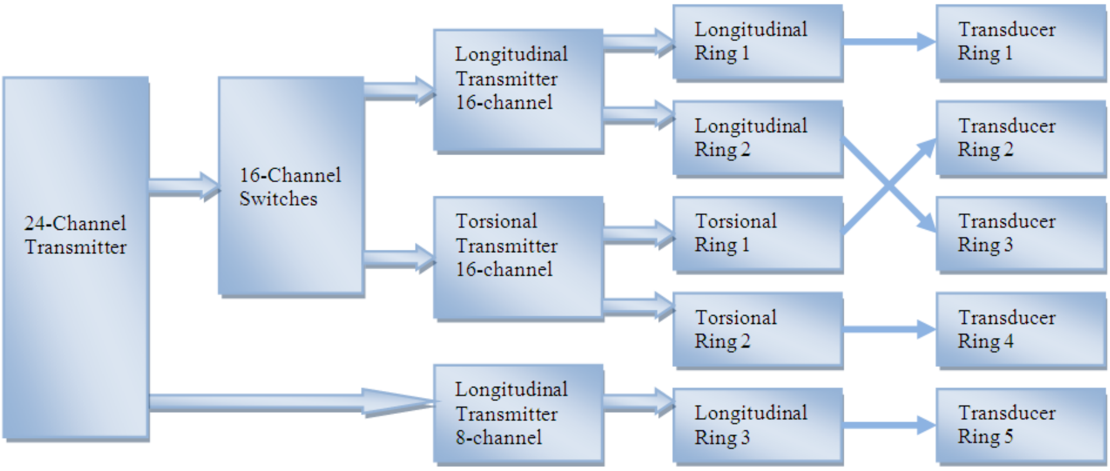


FIGURE 2.4: Transducer Interface for 24-Channel Transmitter

the rest 8 channels can only be used to drive longitudinal transducers. The unit has 8 receiver channels and has to complete 3 sets of testings to receive signals transmitted by 24 transmitters.

2.4.2 Remote Electronic Unit

The hardware components of the remote electronic unit (REU) are distributed onto several main PCB boards. The hardware size in terms of PCB board areas and component counts are described here and used in the latter chapters to compare with the Teletest® MK4 system to demonstrate the improvements made in the new design.

Power Supply Board

The power supply board sources from an external battery and converts it to various DC voltage levels required by the system via a series of switched-mode DC-DC convertors. The hardware functions supported by these converted power supply voltages are listed in Table 2.3. It can be seen that the +36V to +72V battery voltage has been stepped down to +12V and then stepped up to ±24V and ±150V. The power efficiency is reduced due to the step up and down process. It is proposed to overcome this inefficient conversion in the new Teletest® system design.

Controller Board

- Microcontroller: The micro-controller provides overall control of the unit hardware. It receives commands from the PC over the ArcNet

Table 2.3: Teletest® MK3 Power Supplies and Supported Hardware Functions

+36 to +72V In, +12V Out	This convertor takes raw, un-regulated battery DC power input and converts it into a steady +12V.
+12V In, +5V & +3.3V Out	This convertor provides the main power rails for most of the REU circuits.
+5V In, -5V Out	This converter provides the 5V rail required by the DAC and ADC pre-amplifiers.
+12V In, +24V & -24V Out	These convertors provide the standby power-down rails for the transmit amplifiers.
+12V In, +150V and 150V Out	These DC-DC convertors provide the high-voltage power rails for the transmit amplifiers.

communication link to set up and control the various hardware components to carry out the desired operation; and transmits the received data back to the remote PC. A Programmable System Device (PSD) is used as processor memory.

- FPGA: The FPGA is used for interfacing 24-channel DACs, 8-channel ADCs and 12 SRAMs.
- ARCNet communication Link to PC (COM20022): The ARCNet Controller with $2k \times 8$ on-chip RAM is configured at 5 MBit/s transmission speed. The reason of using the ARCNet rather the Ethernet was because it is a simple industrial point-to-point transmission protocol and easy to implement.

Transmitter Board

- 24 FIFOs: The FIFOs are used for storing the transmitter waveforms. The FIFO device is 16k (16384 points) long by 9-bit wide. Each FIFO memory is organized such that the data is read in the same sequential order that it was written.
- 24 DACs for converting the digital waveforms into their analogue forms.

Transmitter Amplifier Boards ×4

- The 24 transmitter amplifiers are distributed on four transmitter boards. Each board holds 6 channels high voltage amplifier circuits. These amplifiers are designed to drive PZT transducers with high capacitive load. Each amplifier can drive up to 16 transducers.

Receiver Board

- 8 sets anti-aliasing filters: The high pass filters can be set to either $1kHz$ or $5kHz$. The low pass filter can be set to one of six options: $16kHz$, $32kHz$, $75kHz$, $150kHz$, $300kHz$ and $600kHz$. The 8 channels can only be set up with the same filter configuration.

- 8 sets receiver amplifiers: Each receiver channel includes a set of configurable analogue amplifiers and filters. The gain values of the amplifiers can be set from 20 dB to 100 dB with 1dB step. These various gains are achieved by the combination of 4 attenuators (-1dB, -2dB, -4dB and -8dB) and 6 amplifiers (one 20dB/30dB pre-amplifier, three 20dB amplifiers and two 10dB amplifiers). The amplifier outputs are differential pairs providing about 5V peak-to-peak to the ADC inputs. The 8 channels can only be set up with the same gain configuration.

- 8 ADCs: This ADC device has a differential analogue input and parallel digital output. It has 12-bit resolution and operates at up to 1 MSPS sampling rate. All eight convertors operate simultaneously.

- 8 banks SRAM receiver memory: The 12 SRAMs are arranged to form 8 memory banks for 8 receiver channels. Each bank can store 64k (65536 points) 24-bit wide data. The ADC outputs are 12-bit wide. The extra 12-bit memory is used for accommodating accumulations of up to 4096 successive waveforms. The accumulation process can be described as ‘transverse’ in the sense that the data values accumulated are the sets of points from the equivalent point on waveforms of successive receiving iterations. This is distinct from successive points along the same waveform.

Channel Switching Board

- The channel switching board has 16 relays for switching between longitudinal and torsional transducer channels. It also multiplexes every 3 transducer channels into 1 receiver channel as there are totally 24 transducer channels, but only 8 receiver channels. It takes three transmitting and receiving sequences to receive data on all transducer channels.

Backplane Connection Board

- The backplane connection board functions as interfaces for the other function boards.

Front Panel Controller Board

- The front panel controller board controls the display, four buttons on the front panel for controlling the integral pump to maintain the pressure of the pneumatic transducer collar.

2.5 Teletest® MK3 System Specific Requirements

The typical requirements of the Teletest® MK3 system include

- Generate high voltage tone burst waveforms to drive 24-channel transducers arrays.
- Receive parallel data on 24-channel receivers and store the captured data in the system local memory.
- Programme the embedded system by a remote PC and upload received data to the PC.

2.5.1 Waveform Generator

High Power Output Transmitter

High power is required to drive the transducer array for the LRUT system due to the attenuation during UGW propagation. It is also required to use high voltage output to drive the PZT transducer array because the transducers have high capacitance load. The system requires 300 Vpp (Volts peak-to-peak) to drive up to 16 PZT transducers with frequency up to 100 kHz. It is possible to design and develop more complicated electronic unit with a higher power output. However, while higher voltage can increase the test distance it may also yield higher noise level, therefore the signal-to-noise ratio (SNR) is not necessarily increased. Besides, a higher voltage output level consumes more power and is undesirable for battery-powered portable LRUT instrument.

Transmit Waveform Frequency Range

Higher frequency ultrasound has a smaller wavelength and better test resolution. On the other hand, lower frequency ultrasound has a longer wavelength and lower attenuation. The relation between wavelength and frequency is $v = \lambda \times f$, where v is the ultrasound velocity in a particular material, λ is the ultrasound wavelength and f is the ultrasound frequency. The velocity of the UGW propagating in the tested specimen is constant during a test, therefore λ is the reciprocal of f and vice versa. Conventional UT uses small wavelength, in other words, high frequency (generally 1 MHz to 10 MHz) to detect defects within a few centimetres with high resolution. LRUT uses relatively low frequency UGW ranging between 20 kHz and 100 kHz to obtain much longer test distance. Besides, it is also necessary to use lower frequency range to avoid generating unwanted UGW modes as the higher the frequency, the more UGW wave modes come into existence.

Two most commonly used UGW modes in LRUT are $L(0,2)$ and $T(0,1)$. The longitudinal $L(0,2)$ mode is used at relatively non-dispersive frequency range: 50 kHz to 100 kHz [18]. Low frequency UGW has lower attenuation and longer propagation distance. Moreover, fewer longitudinal UGW modes are excited at low frequency range, while at higher frequency, more unwanted wave modes would be generated, complicating the inspection. The torsional $T(0,1)$ mode is uniquely non-dispersive and has been employed at lower frequency range: 10 kHz to 50 kHz [23]. This low frequency range has the same advantages as for the $L(0,2)$ mode. Consequently, the transmit waveform frequency range for LRUT is from 10 kHz to 100 kHz.

Transmitting Sample Rate

As the maximum transmit waveform frequency is 100 kHz, 200 kHz sampling frequency should be adequate according to Nyquist-Shannon theory[24]. Practically, 1 MHz sampling frequency is sufficient to present a smooth waveform with at least 10 samples per cycle. However, in order to drive the high capacitance load PZT transducer up to 300 Vpp within tens of microseconds, it is necessary to use higher sampling frequency and more transmit samples to reduce the voltage step between adjacent samples. Otherwise, the transmitter high power amplifiers will have over current. This will be explained in detail in Chapter 4.

2.5.2 Data Acquisition

Transmitter-Receiver Switch

There are two testing methods: pulse-echo and pitch-catch. In pulse-echo testing mode, the transducers are firstly set to transmit to excite a signal and then switched to receive mode. The signal will be reflected back to the transmission position and captured by the same transducers. LRUT pipeline inspection mainly uses pulse-echo method to test from one location; therefore the system has to switch between transmitting and receiving very rapidly. In the electronic unit, these switches are part of the receiver circuits. The receivers are only switched on after the transmitting is completed to avoid high voltage signals being fed into, and breaking, the receiver circuits. In pitch-catch testing mode, also called through transmission, the receive transducers are separated from the transmit transducers. The signal transmitted from one location is received at another location. This mode has mainly been used for scientific research applications.

Analogue Anti-aliasing filter

An anti-aliasing filter is used to restrict bandwidth of the received signal to satisfy the Nyquist-Shannon Theory. The cut-off frequency is set to be less than the Nyquist frequency (half of the sampling frequency) to suppress the mirror image and harmonics of the signals as well as broadband noise. The signal and its mirror image are symmetrical about the Nyquist frequency.

Receiver Gain Control

The variable gains adjust the receive signal to full scale of the ADC input in order to achieve a better digitization resolution. In Teletest® MK3, these gains are set up in common for all receiver channels, from 20 dB to 100 dB with 1 dB step.

Receiving Sampling Rate

The maximum receiving sampling rate is 1 MHz for the 20 kHz to 100 kHz UGW in the Teletest® MK3 system. With a certain amount of SRAM memory, a lower sampling frequency offers the capability of capturing data for a longer time and hence longer test distance. Therefore a selection of sampling rates: 1 MHz, 500 kHz, 250 kHz and 125 kHz are provided in the Teletest® MK3 system. The Nyquist frequency constraint has to be strictly met by any selected frequency to avoid aliasing in the sampled signal.

Receiving Memory

On-board memory is used for storing multi-channel received data. This memory needs to have fast access speed and a large memory size. The required memory size depends on the number of receive channels, sampling frequency and acquisition time. In order to meet the speed and size requirements, special care must be taken over the receiver memory selection and interface design.

Anti-aliasing Filter

There are six analogue low-pass filters in the Teletest® MK3 system. Their cut-off frequencies are at 600 kHz, 300 kHz, 150 kHz, 75 kHz, 32 kHz and 16 kHz respectively. An ideal filter would have a zero transient band in the frequency spectrum; while in practice, a simple first order RC anti-aliasing filter has a long transient band. The cut-off frequency is the frequency where the signal amplitude is $\sqrt{1/2}(or - 3dB)$ of the signal amplitude. It is possible to only maintain the 300 kHz low-pass filter as an anti-aliasing filter and replace the other analogue filters with digital filters in order to reduce the size and power consumption of analogue circuitry, especially when more receiving channels are integrated. Configurable digital filters implemented on FPGA can increase the flexibility of the system.

2.5.3 PC Software Control Mechanism

By transfer of messages over the serial data link, the system PC will be able to carry out at least the following functions:

- Download transmit waveform data to each of the 24 transmit FIFOs.
- Set pulse repetition frequency.
- Set mode to pulse-echo or pitch-catch.
- Set up receive parameters : filtering, gain, digitisation rate, number of receive samples, number of iteration for accumulation.
- Up-load received data.
- Enquire status of current operation such as system temperature, power supply voltage etc.
- Cancel current operation.

2.6 Existing System Limitations and Improvements

The study of the Teletest® MK3 embedded system has found out limitations of the existing system that can be improved in the following aspects:

1. FPGA: The logic resources and maximum operating speed of the current FPGA device is insufficient for advanced multi-channel DSP functions. Effort has been taken to find the optimal FPGA device for the system to support fast data transmission interfaces and multi-channel post-receive signal processing.

2. FPGA on-chip BRAM can be used as fast access memory to replace on-board memory devices and therefore reduce both the board size and system power consumption. The transmit waveform FIFOs are implemented on 24 external FIFOs and can be integrated into FPGA.

3. Soft Processor: A soft microprocessor implemented within the FPGA can be used to replace the standalone microprocessor and improve the system integrity and flexibility. A soft processor is a processor created out of the configurable logic in an FPGA. The existing 8-bit TEMEC 8051 Microcontroller in the Teletest® MK3 system works at 14 MHz. A soft processor can easily work at above 100 MHz depending on a certain FPGA device and a particular FPGA design.

4. More Receive Channels: Current system has 24 transmit channels but is limited to eight receive channels. Three transmit and receive sequences need to be taken per test. Fast testing speed can be achieved by increasing the number of receive channels to 24.

5. Serial Interface DAC and ADC: 10-bit DAC and 12-bit ADC with parallel data interfaces are used for the current system, that is, 10 I/O for each DAC and 12 I/O for each ADC. Therefore, a large number of FPGA I/O pins have to be assigned to interface these devices for multiple DACs and ADCs. The I/O pin number can be enormously reduced by using serial interface device to allow other peripherals to be interfaced to a single FPGA. The other benefits of serial data interface over parallel data interface include less PCB routing, smaller board area, reduced crosstalk issue among data lines, and less impedance matching circuit.

6. Digital Filters: Teletest® MK3 has a bank of analogue filters but no digital filter is applied. As the receive channels are increased from 8 to 24, the previous analogue filter circuitry would take more on board space. FPGA on-chip DSP48A elements can be used for implementing digital filters to replace the existing analogue filters to reduce PCB size and improve signal-to-noise ratio (SNR).

7. Normalization: The performance of piezoelectric transducers are various due to production. This can be compensated by adjusting the transmit waveform amplitude for each channel and make sure that the output surround the pipe is symmetrical. In the current normalization procedure, transducer are divided into three groups, i.e. three rings in the mechanical tool, the receiving signals of each group are added together, and those two groups with bigger outputs will be reduced to match the level of the group with the smallest output. This reduces the efficiency of the hardware. A new test method called Full Matrix Capture (FMC) has the potential to overcome the hardware inefficiency caused by normalization. In this method, one channel transmits at a time and all the rest channels receive, followed by the next transmit channel in sequence. The test result can be processed at a post-test stage. Both symmetrical and focus results can be formed using waveform superposition. In order to carry out FMC test, each transmit and receive channel needs to be individually controllable. The existing system does not have this flexibility.

8. The existing system uses ArcNet communication link to PC at a speed of 5 Mbit/s. 100/10 MBit/s Ethernet interface can be used to increase the test speed and reliability of the communication link to PC and provide potential for linking the embedded system directly to the Internet.

9. The system can be optimized to reduce power consumption in order to allow smaller battery and hence reduce the weight and size of the embedded system.

In summary, the hardware function of the current remote electronic unit in the Teletest® MK3 system was constrained by the limited capability of the old FPGA. This can be improved by using the latest FPGA embedded system design technology to include more transmit and receive channels, to increase both the data acquisition and data transfer speeds and to reduce the size, weight and power consumption of the system. The FPGA characteristics and its suitability for inclusion in the Teletest$^{®}$ MK4 system for all the improvements are discussed in Chapter 3. The selection of the FPGA device is limited by the project budget. The use of a high profile FPGA with large amount of resource was not feasible. This requires extra effort to be made for an efficient FPGA design to meet the system requirement using a low cost and low power FPGA device.

Chapter 3

FPGA Characteristics and Suitability for the Teletest® MK4 System

3.1 Introduction

This chapter reviews the characteristics of FPGA devices, the differences between the FPGA and ASIC devices and the benefits they bring to embedded system design. The previous system based on an old FPGA was no longer adequate and a new FPGA design and development had to be undertaken to make the system suitable for implementing the new functions required by the Teletest® MK4 system. Several FPGA devices have been compared and the Spartan 3A DSP 1800A FPGA device has been selected based on cost and the available resources on it. Section 3.2 compares FPGA and ASIC, and introduces the applications and functions of FPGA devices that can benefit the Teletest® system design. Section 3.3 lists important considerations for selecting the proper FPGA device for Teletest® MK4 system design. Section 3.4 explains the essential knowledge of FPGA architecture and FPGA resources which are important for improving the performance of Hardware Description Language (HDL) design. Section 3.5 explains the FPGA design flow and methods used in this research project; Section 3.6 provides an overview on up-to-date embedded systems and the benefits of FPGA-based embedded systems; as well as describing the FPGA-based processor and peripheral interfacing, which leads to the further detailed explanation of all the peripheral interfaces required by Teletest® MK4 system in Chapter 4.

3.2 FPGA Devices

3.2.1 FPGA VS ASIC

FPGAs are programmable logic devices (PLD) which consist of logic blocks that can be configured 'in the field' into digital circuitry with required functions. FPGA devices also contain on-chip memories and Input/Output (IO) drivers. They can be reconfigured an unlimited number of times by downloading a configuration file (bit stream). The flexibility of FPGA is gained by using more transistors than other IC chips such as ASIC to implement combinational logic functions. ASIC devices can only be configured once, permanently with the specific functions. ASIC devices are generally used for developing high volume products. The ASIC design procedure is much more complicated, the development time much longer and the cost much higher than FPGA design. A brief comparison between FPGA and ASIC is summarised in Table 3.1 and detailed comparison of area, power and performance of over 20 designs are given in [25]. Useful information on the advantages and disadvantages of both FPGA and ASIC is also provided in [26]. This project targets a prototype design for a commercial product with low volume production. Although ASIC devices have smaller chip size, lower power and higher speed, a carefully selected FPGA device with optimized design can meet the requirements of the Teletest® system with the essential advantages of using an FPGA over ASIC: shorter time to market, flexibility of reconfiguration and lower Non-Recurring Engineering (NRE) cost.

	FPGA	**ASIC**
NRE Cost	Low	High
Field Reconfiguration	Yes	No
Chip Size	Larger	Smaller
Power	Lower	Higher
Time to Market	Shorter	Longer
Speed	Slower	Higher
Suitable Application	Prototyping, small quantity	High quantity products

Table 3.1: FPGA VS ASIC

3.2.2 FPGA Advantages and Applications

The popularity of FPGAs is due to their many advantages such as reconfiguration, parallelism, fast speed and various on-chip resources. They can support applications with both processor and custom logic requirements. They are cost-effective for products with low volume production. Their in-system re-programmability shortens the design period from concept to silicon. They provide portability of code across various FPGA vendors and the availability of code libraries and low-cost programming tools make implementation relatively straightforward. Current high density FGPA devices contain not only thousands of Look-Up Tables (LUTs) and Flip-flops (FFs) for implementing complex digital logic, but also have a large variety of on-chip components such as memory, multipliers, DSP elements and other resources dedicated to various device families. FPGA may be used for traditional microcontroller applications and offer the flexibility to system designers to develop custom refined system architectures for specified applications.

FPGA are often used in products produced in small quantities. They can be used as bridge or glue logic to interface bus standards. They are frequently used for experimental instrumentation prototype design for scientific applications. New designs can be easily tested and verified with the flexibility to reconfigure the FPGA, whereby the function can be amended and upgraded to suit the research purpose. For industrial commercial products, FPGA can be reconfigured to adapt new functions, upgraded system specification and industry standards. The most common FPGA applications include data acquisition, industrial standard peripheral interface controllers such as IIC, SPI, UART, Ethernet MAC, memory controller such as SRAM, double data rate synchronous dynamic random access memory (DDR2 SDRAM). FPGA manufacture Intellectual Property (IP) cores and open source IP cores speed up the time to market of the FPGA design. Based on the growth of hard core and soft core microprocessor and peripheral controller IPs, FPGA embedded system technology is becoming mature.

The flexibility of FPGA comes with the undesirable fact that the FPGA on-chip resources are limited and have to be compromised with size, cost and power consumption. The larger devices are generally bigger, more costly and power hungry than smaller devices. The FPGA resource consumption defines the size of silicon area required and directly affects the cost of the FPGA device. The objectives of this project include enhancing the performance while reducing the size, cost and power of the system. Therefore, it is essential to take advantage of various FPGA functions and employ efficient FPGA design methods to achieve better system performance, as well as to optimize the synthesis and implementation to balance the area, speed and power consumption.

3.2.3 FPGA Functions

FPGA devices have many functions. One of the aims of this project is to take the advantage of these FPGA functions for the Teletest® system design. The main functions used in this project are listed as follow:

- Reconfiguration: FPGA can be reconfigured at any time, providing cost efficient fast prototyping and function updating for products. Teletest® is in a fast changing market and has to be capable of being updated to meet upgrade requirements. The reconfigure flexibility is required for the test and debugging process of the prototype development, as well as the system upgrade to support various research and industrial applications in the future.
- Parallelism: Parallel circuits can be designed on an FPGA for running parallel functions simultaneously. A hardware circuit design can be replicated many times on the FPGA given sufficient hardware resource. It is suitable for multichannel data transmission, receiving and processing for the Teletest® system. While most single core general purpose processors (GPP) are single threaded and can process one instruction at a time, multichannel system increases the central processing unit (CPU) utilization and limits the maximum speed achieved by each individual channel.
- High speed: A digital design implemented on FPGA hardware can run at fast clock rate. Combined with parallelism function of the FPGA, the speed and data throughput requirement for the multichannel transmitting and receiving system for Teletest® MK4 system can be easily met.
- An integrated soft microprocessor can be implemented in FPGA logic to replace an external microprocessor. The soft microprocessor can be parameterized for specified applications and can make the system more flexible, compact and power efficient. In Teletest® system, the microprocessor needs to include a variety of interface controllers to connect to external peripherals. It is very difficult to find one off-the-shelf microprocessor device exactly meet the requirements. Some high profile devices may meet the requirement but also include unwanted interfaces

and at relatively high cost. Some FPGA devices have integrated hard core microprocessor integrated in FPGA but also at much higher cost than those containing a soft microprocessor. A soft microprocessor can also be ported among different FPGA devices.

- On-chip memory: FPGA devices usually contain a certain amount of on-chip memory resource depending on the device family and the density of the device, which is relatively faster than external memory. This type of memory can be used as cache memory for the microprocessor implemented on FPGA. However, the FPGA local memory is limited and the Teletest® embedded system requires large memory to run the firmware. Therefore, an off-chip external DDR2 was added. In the Teletest® system, the on-chip memory is also used as dual port RAM for storing the waveforms of 24 transmitting channels. FPGA local memory is volatile and will lose stored data as well as FPGA configuration upon power-off.

- Flexible Pin Assignment (excluding the fixed Power and clock pins): Microprocessors have fixed pins, while FPGA pins can be assigned with different functions, this increases print circuit board (PCB) design flexibility and reduce PCB layout and routing difficulty.

- FPGA devices have abundant IO pins and support various IO voltages. The FPGA can be interfaced to multiple external devices with various IO standards and voltage levels. Same family devices are pin-compatible and can be upgraded if more logic resource is required.

- Digital clock management provides adjustable and reliable global clocks. The Teletest® system requires clocks at different frequency for transmitter, receiver and top-level sequencer respectively. The Digital Clock Manager (DCM) provides reliable global clocks with reduced jitter and phase alignment. It is useful for the synchronization of the multichannel transmitting and data acquisition system.

- DSP Element: These fast speed on-chip DSP elements add the function of in-system DSP and can be used to implement digital filters for the 24 receiver channels in the Teletest® MK4 system. Spartan 3A DSP devices are featured with a number of DSP48 elements[27].

- Pipelining: Pipelining is to divide a process into multiple tasks that can be carried out concurrently in order to increase data throughput and processing speed.

3.3 Considerations on FPGA Selection for Teletest® MK4

Xilinx® and Altera® are two mainstream FPGA vendors which together take up to 90% of the FPGA market worldwide. Other alternative vendors include Actel®, Lattice®, etc. Xilinx® first developed the FPGA concept and is the biggest FPGA manufacturer. All companies provide a wide range of products that support different level of applications. Both Xilinx® and Altera® FPGAs are SRAM-based, which is the most common FPGA architecture. There are also anti-fuse and Flash FPGAs from Actel® and Lattice®. SRAM-based FPGA are flexible, large scale devices that can be reprogrammed. Anti-fuse FPGAs are one time programmable and do not need to be configured after power on, but are less flexible and small scale devices [28, 29].

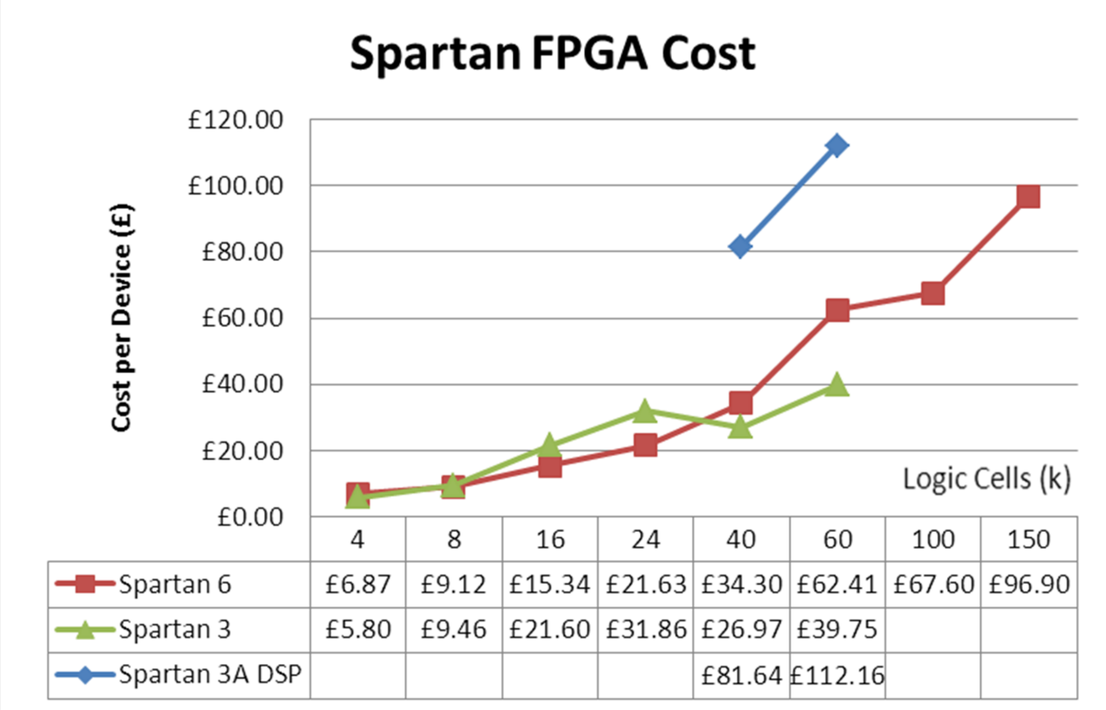
In Teletest® MK3 system, a Xilinx® SpartanXL Family FPGA XCS30XL is used. This device has limited speed and logic resources that cannot support more functionality to be added to the system. Therefore, it was necessary to use a new FPGA device in the new Teletest® system. The MicroBlaze soft processor IP core of the FPGA device can replace the hardware microcontroller chip device and the FPGA internal BRAMs can replace 24 external memory devices for storing 24-channel transmit waveforms. The total cost, component count and size of the system can be reduced and the system speed increased by using the state-of-the-art FPGA. There are several aspects to select an FPGA for an embedded system design. The FPGA selected must have sufficient resource and speed in order to implement the application. In this project, cost is also a highly important aspect which set the criteria of the design.

3.3.1 Estimation of Resource Utilization and Speed

The estimation of the required FPGA resource and speed is based on the resource usage of Teletest® MK3 and functionalities added to MK4. As a rule of thumb, it takes the FPGA design tool longer to automatically place and route the design when the resource usage is higher. If the resource usage is more than 80% of the total logic resource on the FPGA, the place and route may fail on timing. Some suggestions on how to select an FPGA based on the combination of the requirements of maximum frequency and routing resources are given in [30]. It recommends starting with the design taking up 50% of the FPGA logic to leave room for the inevitable future design changes.

3.3.2 Cost of FPGA Devices

The cost of the FPGA device was an important factor in this project. It is generally understood that FPGA is cheap in comparison to ASIC in applications where a relatively small number of quantities are required and frequent reconfigurations are involved. Figure 3.1 shows the up-to-date prices of three Xilinx® FPGA device families Spartan 6, Spartan 3 and Spartan 3A DSP taken from one of the main distributor Avnet®'s website on 10th April 2011. Similar price information for Altera® FPGA devices can be found in [31]. Each FPGA family includes a number of devices and each device may have different packagings and speed grades, causing great variances in individual prices.



	4	8	16	24	40	60	100	150
Spartan 6	£6.87	£9.12	£15.34	£21.63	£34.30	£62.41	£67.60	£96.90
Spartan 3	£5.80	£9.46	£21.60	£31.86	£26.97	£39.75		
Spartan 3A DSP					£81.64	£112.16		

FIGURE 3.1: Xilinx Spartan FPGA Devices Cost Vs Available Logic Cell Resource

Following the requirement to lower the developing budget for the commercial product, the lowest prices for each device is selected for the comparison. It can be spotted that low profile FPGA is relatively cheap, with the cost of below £10; on contrast, high profile FPGA devices can be extraordinarily expensive, up to £10,000. Within each family, the lowest cost per logic element falls on the lower-middle sized device. As this range of FPGAs become popular, the production quantity increases and drives the price per piece down the same way as in the ASIC industry. This leads to the result that this range of FPGA is chosen by more engineers in their design. The popularity of FPGA devices also means there is less risk for the device becoming obsolete in the near future, which reduces potential costs of upgrading the commercial product using the FPGA device. All the above have been taken into consideration in selecting the right FPGA for this project with the major aim to keep the manufacturing cost down. It is recommended to use mature products from one of the main FPGA vendors for a commercial product to guarantee FPGA device availability in product life time and device compatibility for upgrading unless special requirement can only be met by non-mainstream FPGA devices.

Other indirect costs need to be taken into consideration include FPGA recompile time, complexity of PCB layout and design software. Sometimes the greater power consumption may require extra thermal mechanism design to avoid system failure due to overheating.

3.3.3 FPGA Pin Compatibility

Another consideration for this project is that the selected FPGA device had to be compatible with high profile devices to allow system upgrading. FPGA IO pin out and power standards are generally compatible within one family from one manufacturer. Pin layout compatibility is useful for system upgrading without re-layout the PCB. In general FPGA design, PCB layout and logic design are separated tasks accomplished by different engineers. Because PCB layout and manufacture takes time, it is normally required to start either before, or concurrently with the FPGA logic design. As it is difficult to estimate the FPGA logic resource required accurately; the FPGA pin compatibility makes it possible to decide which FPGA chip to be mounted on the PCB after the logic design is completed. There are two pin compatible devices in the Spartan 3 DSP family FPGA: 1800A and 3400A. They have the same package but different logic and IO resources. This provides future-proofing for more complicated function logic designs, which require more logic resource, without the extra time and cost for redesign the PCB.

3.3.4 FPGA Design Tools

The FPGA design tool and development environment also need to be taken into consideration for chip selection. Mature development tools provided by FPGA manufacturers such as Xilinx® ISE, EDK and Altera® Quatus and Mentor Graphics® are very commonly used by FPGA design engineers. The efficiency and ease of use of the tools can reduce the development time and save design cost. Mainstream companies provide better technical support, logic design software maintenance and upgrade services, which reduce long term system development, upgrade and maintenance cost. The author has previous experience of using FPGA design tools from Xilinx®, Alteral® and Altium®. The selection of the Xilinx® FPGA is partially due to the familiarity and ease of use of the tool.

3.3.5 Spartan 3A DSP 1800A FPGA

The Spartan 3A DSP $XC3SD1800A$ FPGA speed grade -4 was selected for its low cost and low power consumption, in line with the requirements of the overall product development specification. It has $1800K$ system gates, 84 additional 18kbit on-chip block RAM (BRAM) and 84 additional XtremeDSP DSP48A slices. The different features of the Spartan-XL FPGA used in Teletest® MK3 system, the Spartan 3A DSP FPGA selected for MK4 system and its compatible device for upgrading are listed in Table 3.2.[32]

Two new generations of Xilinx® FPGA devices have been released to market since the Spartan 3A DSP FPGA was selected for the project. Due to high development and manufacture cost, it is difficult to update the industrial product to target the latest FPGA device. Nevertheless, the selected Spartan 3A DSP FPGA was the optimal solution at the time. The concepts and principles used for selecting the optimal FPGA device to meet the cost and performance requirement for an industrial product can be adopted for future development and other applications.

3.4 FPGA Architecture

The architectures of FPGA devices from various vendors commonly contain the following resources which are used in the Teletest® MK4 system design for implementing the required functions. The usage of different types of FPGA

	Teletest® MK3	**Teletest®** MK4	**Future Update**	
	Spartan-XL XCS30XL	*Spartan-3A DSP XC3SD1800A*	*Spartan-3A DSP XC3SD3400A*	*Spartan 6 LX XC6SLX150*
Supply Voltage	3.3V	1.2V core 1.8/3.3V IOs	1.2V core 1.8/3.3V IOs	1.2V, 1.0V
Total CLBs	576	4,160	5,968	5760
Total Slices (1 CLB = 4 Slices)	2,304	16,640	23,872	23,038
Number of Slice FFs	4,608	33,280	47,744	184,304
Number of LUTs	4,608	33,280 [1]	47,744	92,152 [2]
Number of 18kbit BRAM	24	84	126	268
DSP48A	24 (Multipliers)	84 (DSP48A)	126 (DSP48A)	180 (DSP48A1)
Maximum User I/O	192	519	496	576
Number of GCLKs	N/A	24	24	16
Price	£30	£81.64	£112.16	£266.25

Table 3.2: Comparison of Spartan XL and Spartan 3A DSP FPGAs

resources by each function such as the MicroBlaze microcontroller, the transmitter controller, the receiver controller and so on is discussed in section 7.2. The overall resource utilization for the Teletest® MK4 FPGA embedded system design is also analysed in section 7.2.

- General-purpose resources

  Logic resources (FFs and LUTs)

  IOB resources

  Clock resources (DCM and GCLK)

  Memory resources (BRAM, Distributed memory)

- Manufacturer and family specific resources

[1] Each Spartan-3 FPGA CLB contains four slices; each slice contains two 4-input LUTs and two FFs.

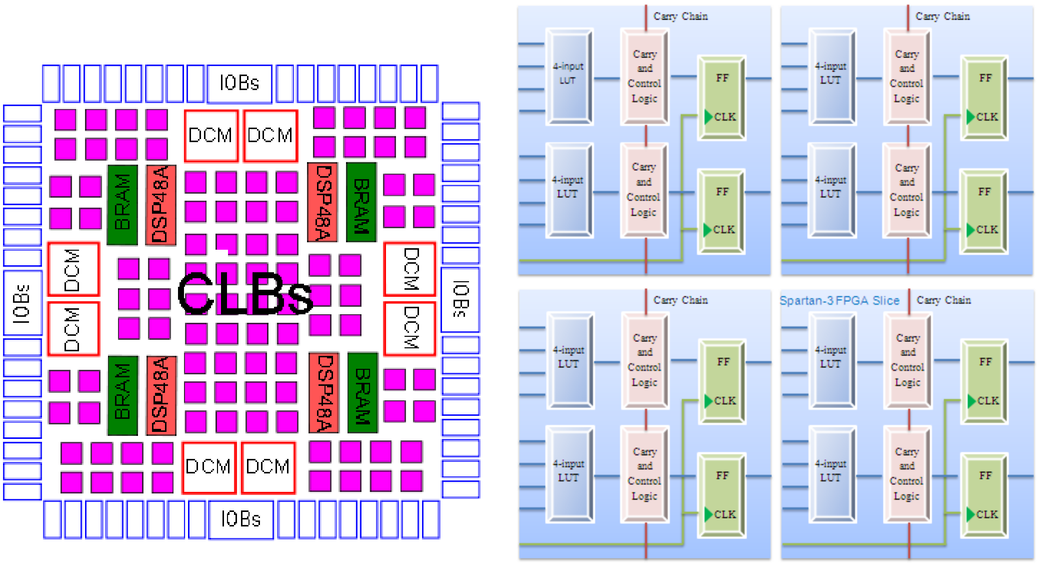
[2] Each Spartan-6 FPGA CLB contains two slices; each slice contains four 6-input LUTs and eight FFs.

Microprocessors

MUX

DSP48

The understanding of the FPGA hardware resources is important to improve the FPGA design efficiency in order to implement the Teletest® embedded system onto a low cost and low power FPGA device. The architecture of the Xilinx® Spartan 3A DSP FPGA is illustrated in Figure 3.2(a). The Spartan-3A DSP FPGAs consist of five fundamental functional elements: Configurable Logic Blocks (CLBs), Input/Output Blocks (IOBs), BRAMs, DSP48 elements and DCM [33]. The design concept used in this project can be expanded for FPGA devices from other family or vendors with similar architecture and resources.



(a) Spartan 3 FPGA Architecture[34]

(b) Spartan 3 FPGA CLB[34]

Figure 3.2: Spartan 3 FPGA Architecture and CLB

3.4.1 Logic Resources

CLBs contain LUTs used for implementing logic and storage elements used as FFs or latches. Every CLB contains four slices. A typical arrangement within CLB on Spartan device is illustrated in Figure 3.2(b). An overview of CLB on Spratan-3 FPGA device is given in [34]. The slices are the most essential and fundamental resource in an FPGA design. The number of CLBs offered by an FPGA device varies depending on FPGA family and range. A synthesis result of an FPGA design can provide an estimation of the required amount of logic resources and help to select the right FPGA for an application. The interconnect structure is built up by switch matrices between neighbouring CLBs. These switch matrices are called Programmable Interconnect Points (PIPs).

Each slice contains two LUTs and two registers. LUTs can be configured in three different ways: combinational logic, distributed RAM, and Shift Register Logic (SRL). A typical LUT has 4-input or 6-input and one output. When used as combinational logic, the capacity of a LUT is limited by the number of inputs rather than the complexity. So delay through the LUT is constant, regardless of what logic function is being performed inside.

LUT can be configured as distributed RAM to perform similar function as the dedicated BRAM resource, such as single or dual port, RAM or ROM functions. A 4-input or 6-input LUT can provide 16 bits or 64 bits memory respectively. LUT can also be used to build SRL that are primarily used as pipelined delay element for balancing pipelined applications. Pipelining is an effective way of designing to increase design performance since there are so many registers in FPGAs. It is necessary to delay some of the signals as pipelines can sometimes become unbalanced when too much logic must be generated. One of the best uses of the SRL is to add delay to balance pipelines. These SRL are simple serial-in serial-out shift register. If parallel load and reset are inferred in the HDL design, the synthesis tool will use extra regular registers and this can be a waste of FPGA resource. The design for the multichannel transmitter explained in Chapter 5 provides an example of reducing the FPGA logic resource usage for implementing the multichannel parallel to serial converters for 24 transmitter channels by cleverly configure the dual port BRAM with direct serial outputs.

There are also dedicated multiplexers (MUXs) used to improve the speed of signal multiplexing and save LUTs for other purposes. These dedicated multiplexers improve the speed of the design because they are faster than the equivalent logic built just with LUTs. They are also fast because there is dedicated routing between the multiplexers that is low fanout and fixed between the logic resources. In the HDL the dedicated multiplexer can be inferred by coding with CASE statement.

Slices also have carry logic resources which are dedicated hardware resources designed to implement fast arithmetic functions, such as addition, accumulation, subtraction, and compression. In common FPGA architecture, carry logic propagates the carry signal vertically upward in dedicated carry chains that run vertically in columns; the least significant bit is placed at the bottom of the carry chain.

The register resource can be used as storage element such as FF or latch. Each FF has three control ports, clock(CK), CE(clock enable) and synchronous set/reset(Set/Reset). CE and Set/Reset ports are both active high. The limitation of the control signals can limit the device resource utilization and maximum running speed of an FPGA design. It is important to group the associated control signals with a FF because only FFs which share the same control signals can be grouped into the same slice. Hence it is optimal to reduce the number of control signals in a design in order to get high device utilization.

Un-optimized HDL coding style can be used in an ASIC design and still results in a good speed performance, simply because ASIC are much faster than FPGA. However, it is essential to use optimized HDL coding style for a particular FPGA architecture in any FPGA design. Xilinx® Spartan 3 family FPGA [33] has 4-input LUT and the synthesis tool would use two LUTs in series for a combinational logic that has five inputs. This can increase the delay associated with this logic path and potentially reduce the maximum frequency achievable. In a good FPGA design using optimized HDL, all the delay paths in the design should be limited to as low a logic level as possible, in other words, less LUT cascaded in series, in order to get a faster maximum possible frequency. In the Teletest® MK4 FPGA design, there are pre-designed controllers for the commonly used peripherals such as Ethernet MAC, UART, SPI and so on. These controllers are ready-to-use codes which have been optimized by Xilinx® for the Spartan 3A DSP FPGA. However, the multichannel transmitter controller and receiver controller had to be coded manually in VHDL using efficient coding style to reduce FPGA resource usage, reduce the power consumption and increase the speed. The integrated design has been further optimized using the code optimization options provided by the Xilinx® synthesis and implementation tool ISE and the results are compared in section 7.4.

3.4.2 IOB Resources

IOBs are the interfaces between the FPGA and other devices. IOBs control the data flow between IO pins and FPGA internal logics. The IOBs contain registers to clock data into and out of the FPGA most efficiently. They support bidirectional and 3-state data flow. They also contain special interface logic designed to translate the internal voltage domain of the FPGA to other I/O standard required without the necessary of adding external interface logics. IOBs are grouped into IO banks located on the edges of the FPGA device. The IOBs on each bank share a common power supply for inputs ($Vref$) and outputs ($Vcco$). These voltages vary by IO standard. Most FPGA devices can support many different standards. Many of these standards can share the same power supplies and hence it is good practice to keep compatible IO standards in the same bank. Double Data-Rate (DDR) registers are included.

In the Teletest® system, the FPGA device is interfaced with a number of peripheral and memory devices on the PCB. These devices require different voltage standards for interfacing. Input signals fed to the FPGA external ports are connected to internal input buffers (IBUF). FPGA internal output buffers (OBUF) are used for output signals to external ports. The IOB resource utilization and the I/O standards and voltage levels of various interfaces in the Teletest® MK4 system are reported in section 7.2.3.

Serdes functionality is built into each IOB in Spartan 6 and Virtex 4 onwards family FPGAs. This includes both parallel to serial and serial to parallel conversion. Virtex 4 family FPGA have high power consumption and at high cost. Spartan 6 family was not yet available back in 2008 when the project started. The IOBs of the selected Spartan 3A DSP FPGA used for the Teletest® MK4 system does not have the Serdes function. Therefore all parallel-to-serial and serial-to-parallel conversions need to be built by using logic resources, as for the ADC and DAC controller interface designed and implemented for this project.

3.4.3 Memory Resources

On-chip memory: Xilinx® FPGAs offer two distinct memory options, BRAM and Distributed RAM. BRAM uses dedicated, on-chip, hardware resources and represents the most area-efficient RAM implementation. Each BRAM provides data storage in the form of 18-Kbit dual-port block. In the Teletest® MK4 FPGA design, a total number of 73 BRAMs out of the available 84 is used in the design for various functions such as the MicroBlaze controller local memory, the DDR2 SRAM controller, the transmitter waveform generator FIFOs and Ethernet. BRAMs offer high access speed for these peripheral controller functions but due to their fixed location on the chip, may incur slightly larger routing delays. The utilization of the used BRAMs on the FPGA for the Teletest® MK4 design is discussed in section 7.2.4.

The distributed RAM uses the LUTs in the FPGA slices to implement memory and in doing so will subtract from the slices available for logical operations. Distributed RAM can be located anywhere throughout the chip, therefore routing delays can be minimized and slightly higher performance can be achieved. Distributed RAM is an excellent option for small FIFOs. FPGA on-chip memory can be initialized in the design at configuration time, therefore the designed system can start from a known state. In the MK4 system, the distributed RAMs are used as configuration registers to setup the system with default parameters such as the number of accumulation (default 16), the transmitter interpolation rate (default x16) and so on.

3.4.4 3.4.4 Clock Resources

One important step in system design is to identify the number of different clocks used in the design and route these clocks efficiently. FPGA devices have dedicated global clock buffers connected to every FF on the FPGA. It is good design practice to take the advantage of these low-skew clock routing resources for the successful timing closure of an FPGA system design. The Teletest® system includes multiple clock domains for various functions. A 125 MHz external oscillator is used as the FPGA input clock. The system requires a 62.5 MHz clock for the soft microprocessor; a 20 MHz clock for the receiver sequence; a 16 MHz clock for the transmitter sequence; a 10 MHz clock for the top level sequence; and a pair of differential 125 MHz clocks for the DDR2 memory controller. These clocks are generated by two DCMs. DCMs are FPGA on-chip dedicated resources and can generate different clock frequencies from a single reference clock [35]. Custom logic can be used to divide an existing clock into lower frequency clocks. However, it is recommended by Xilinx® user guide to use DCM to generate slower clocks because the DCM has the advantages of achieving lower jitter for the generated clocks and specified phase relationship among generated clocks. Table 3.3 lists the different clock frequencies required by the Teletest® MK4 system. In the design the DDR2 SRAM memory controller uses the fastest clock at 125 MHz. The global clock constraint of the system is set by the input 125 MHz clock. However, there are functions in the system working at much lower clock rates. Figure 3.3 shows the FPGA logic design schematic for generating the required clock outputs from the 125 MHz system clock input using two DCMs. By putting constraints on the slower clocks to aid the FPGA implementation tool, the routing of the fastest clock can be improved and the overall implementation time can be reduced.

TABLE 3.3: Various Clocks for Teletest® MK4 Functions

Teletest® Function	**Clock Source**	**Speed**
DDR2 SDRAM	DCM0	125 MHz
MicroBlaze Microprocessor	DCM0	62.5 MHz
Transmitter Sequence: DAC Controller	DCM1	16 MHz
Receiver sequence: ADC, SRAMs	DCM1	20 MHz
Top Level Sequencer	DCM1	10 MHz
CPLDs	Top level sequence	1 MHz
Ethernet	25 MHz oscillator	100 MHz

3.4.5 Other Dedicated Resources

Different FPGA families have a wide range of device-specified dedicated resources which range from integrated microprocessor hard core to DSP elements. The dedicated resources have an intended purpose of saving LUTs and improving the speed of the design. These dedicated resources should be used as much as possible to free up the FPGA logic resource to achieve optimized resource utilization and place and route implementation results.

3.5 FPGA Design Methods

A complete FPGA design process includes the following steps: design (schematic or HDL), simulation, synthesis, implementation(mapping, place and route), timing analysis, power and speed analysis. For the Teletest® MK4 system, the multichannel transmitters and multichannel data acquisition are custom IP cores and were designed following the complete design cycle. The design is the first and most important step which mostly affects the final design result.

3.5.1 FPGA Design Flow

An FPGA can be designed using either HDL or schematic capture. HDL offers great flexibility over schematic capture and is portable for various FPGA devices.

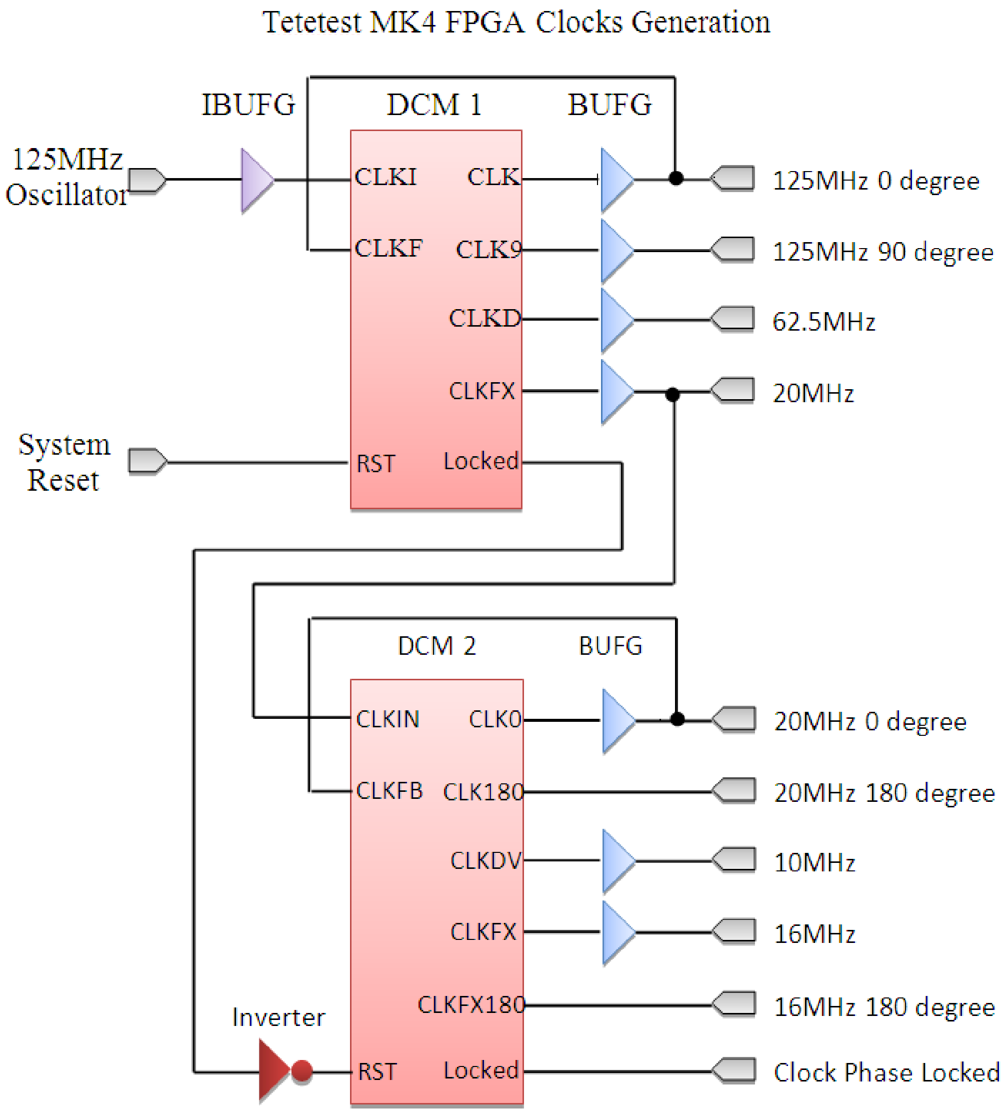


FIGURE 3.3: Teletest® MK4 System FPGA Clock Generation using Two DCMs

Strictly speaking, FPGA design is not just programming or coding, although a design language such as VHDL or Verilog is used, rather it is in essence a mean of describing a digital circuit which is implemented using the FPGA resources with specially configured interconnections to perform a certain function[36]. In the Teletest® MK4 system FPGA design, the design language VHDL is used for the FPGA logic design, the development tool Xilinx® ISE 11.4 is used for synthesis and implementation, and the simulation tool ModelSim XE III 6.1 is used for simulation. The design tools used can be configured to optimize the size or speed of the system and improve system performance, but the VHDL coding style and design method are the most important factors that affect the performance of the final system. The Xilinx® EDK 10.1 is used for designing the embedded system.

There are two methods for integrating the embedded system into an FPGA. The first is to import ISE logic function blocks into EDK project as custom peripheral IP cores. The second is to integrate the EDK project into ISE project as an IP core. The first method was used for the MK4 system FPGA design at the early development stage because the EDK project can export the hardware design directly to the firmware development tool SDK to generate the test programme to test the hardware design. The second method was used to wrap the final working EDK project into a top level ISE file because the ISE tool provides design optimization and power analysis.

Xilinx® Design Tools

- Integrated Software Environment (ISE): Design, synthesis, mapping, place and route;
- Embedded Design Kit (EDK): FPGA embedded processor design tool, supports custom peripherals;
- Software Development Kit (SDK): FPGA embedded processor software development tool, providing software code library for peripheral controller drivers, as well as memory and peripheral testing applications;
- ModelSim : HDL simulation tool, support both VHDL and Verilog;
- ChipScope Pro: On-Chip signal logic analyser.

Design Entry

The first step of the FPGA system design is to use HDL to describe the digital functions required. VHDL is used in designing the FPGA functions for the MK4 system because the author's preference based on previous design experience. HDL is target device independent language. Although the present day design tools can automatically optimize the HDL, the understanding of the architecture of the targeted FPGA device can help the designer to code for better synthesis and implementation results, as well as to generate more compact and faster design.

Simulation

HDL was originally used by digital circuit design engineers for the simulation and verification of logic circuits. In general, the test bench codes written in HDL for simulation and verification purpose are very important and usually take more time and effort to accomplish than the logic function codes. FPGA vendors provide various simulation tools to support their products. Xilinx® ISE has an integral simulator ISIM and also supports a third party tool ModelSim. There are three simulation stages: behaviour, post synthesis, post place and route. The VHDL design for the Teletest® MK4 transmitter and receiver controllers are simulated using ModelSim. The screenshot of the behaviour simulation result of the receiver controller is presented in section 6.5.

Synthesis

The synthesis tool (XST) is used to translate the Register Transistor Level (RTL) codes into FPGA architecture, in other words, to convert HDL logic into FPGA primitives such as LUTs, FFs, memories and other dedicated on-chip resources. It is recommended to use pre-defined HDL code from Xilinx® library so that the synthesis tool can infer to these resources efficiently. Various options can be selected for synthesis optimization, such as register balancing.

Implementation: Mapping, Place and Route

Mapping is to map the LUTs and FFs into FPGA slices. Traditionally, mapping is part of the implementation process. However, the latest Xilinx® synthesis tool can accomplish mapping within the synthesis process. The implementation tool decides where to place the mapped slices and memory resource and selects an appropriate routing that may include different kinds of routing resources. It tries to route the design to meet the timing requirement based on the user timing constraints. The placement and routing utility PAR of the ISE is used for the mapping and placement of logical resources. In the Teletest® MK4 design, certain timing critical components are manually placed using Planahead in order to meet the timing constraint. The mapping, place and route result of the FPGA design for the Teletest® MK4 system is presented in Chapter 7.

Debug with ChipScope Pro

A fair amount of design effort was related to hardware and hardware/software interfacing debug in the Teletest® MK4 FPGA design. The FPGA system for the Teletest® MK4 is a complex embedded system with a number of functional IPs. Since these IPs are encapsulated inside the FPGA and the internal signals are inaccessible for analysis and debug using traditional monitoring utilities such as digital analyser or oscilloscope, Xilinx® ChipScope Pro [37] was used to assist the debugging of the FPGA design. The following IP cores are used for capturing real-time digital signal in the FPGA.

- Integrated Controller (ICON)
- Integrated Logic Analyzer (ILA) [38]

3.5.2 Methods to Improve FPGA Performance

IP core

IP core is a 'black box' including circuits or codes of certain function. In recent years, the IP core design method has gradually become the mainstream design method [39]. Embedded system design technique is a fast growing technique. It can be impossible for one company or an individual engineer to develop a complicated embedded system; or it can be very time consuming even for a group of experienced design engineers [40]. Common function IP cores can reduce the work load and allow researchers and engineers to concentrate on design and development of new algorithm. The pre-designed IP cores used in the Teletest® MK4 system design include Ethernet MAC, DDR2 SDRAM memory controller, SRAM memory controller, UART, SPI, IIC peripheral controller. The details of these IP cores and the related external peripheral devices are discussed in Chapter 4.

Finite State Machine

Finite State Machines (FSM) can be used as sequence generators and detectors and are well suited for embedded system design [41]. An FSM consists of a sequence of states, which are implemented on an FPGA in a predefined order. An FSM design includes combinational logic and sequential circuits. The combinational logic circuits check the input values in the current state and determine the outputs and the next state. The sequential circuit updates the register values and carries out state transitions. It is more reliable to use synchronous design in which the state transitions take place at the rising or falling edge of the system clock. In asynchronous design, the combination logic outputs occur at different time due to propagation and logic delays, causing jitters and unpredictable results. The FSM is usually divided into three parts when described in HDL: sequential logic for synchronous registers and state value; combinational logic defining next state; combinational logic defining outputs. The RTL design methodology was presented in [42, 43]. There are three FSMs designed for the Teletest® MK4 system: the transmitter controller, receiver controller and top level sequence controller.

3.6 FPGA Embedded Processor and Peripherals

The leading FPGA vendors provide established hard and soft processor IP cores for using with their respective FPGAs for fast system prototyping. Based on the up-to-date literature research carried on for the past three years, there is not a dearth of literatures about FPGA-based embedded system designs. There have been many papers published to address certain aspect of FPGA design based on integrated processor on various applications, but details are generally excluded, making it uneasy to regenerate the results declared to compare with the Teletest® MK4 system design. A detail design example of an industrial robot controller using FPGA and Verilog HDL has been published in [44].

Soft core VS Hard core Processor

An FPGA based embedded system can have either a soft core or a hard core microprocessor. A hard core processor integrated on the FPGA device in general has better performance and is more expensive, but with fixed parameters and peripheral interfaces. A soft core processor implemented using FPGA logic resources is cheaper, more configurable and flexible, various peripheral interfaces can be added via standard peripheral controller IP cores, which are also implemented using FPGA resources. The performance of soft core processor depends on the FPGA device used. Teletest® MK4 system is designed using an FPGA with a soft core microprocessor to provide flexibility with limited system development budget. The required peripheral controllers are also implemented on the FPGA and equivalently form a customized microcontroller with specified peripheral interfaces. This helps to reduce cost and size of the system.

Xilinx® and Altera® both provide soft core and hard core embedded system solutions. Xilinx® has 8-bit PicoBlaze, 32-bit MicroBlaze soft core processor and Power PC hard core processor. Altera® has developed the NIOS II soft core processor and is bonded with ARM9 as its hard core processor.

A MicroBlaze is implemented in the Teletest® MK4 FPGA design. MicroBlaze is an embedded soft RISC processor with 32-bit separated data and address bus. It can run at speeds up to 200 MHz and can be extended with additional coprocessors. The processor is optimized for implementation in Xilinx® FPGAs, and supports on-chip BRAM and peripheral controller IP for interfacing custom devices. It is a relatively mature technique. In embedded system development, the hardware and the software development often take place at the same time. When problems inevitably arise, this dual development environment often makes it difficult to determine whether they are due to hardware or software errors. The Xilinx® embedded system development environment EDK and SDK are used for software and hardware co-development.

One of the advantages of soft processor core is to only implement the required functionality and not the functionality that is not required in order to save FPGA resources and help to improve the embedded system speed because there is less logic to be placed and routed. An embedded GPS receiver system using MicroBlaze presented in [45] and an FPGA based general purpose data acquisition controller presented in [46] provide two out of many successful examples for implementing an embedded system using MicroBlaze based on an FPGA.

Hardware-Software Co-design Using Xilinx® EDK

The hardware and software design tasks in the Teletest® MK4 system design are carried out independently. The firmware running on MicroBlaze and the software running on the remote control PC are developed by software team and are out of the scope of this thesis. In the Teletest® MK4 hardware design, the top level design is built in Xilinx® EDK. The synthesis and implementation procedures are completed by calling up ISE within EDK project. The hardware design flow takes the electronic design interchange format (EDIF) files of the MicroBlaze soft processor, merges them with the user written custom digital code and prepares a complete system netlist after synthesis. An FPGA netlist is a computer representation of a list of logic resources utilized in an FPGA design and the connections among them.

In the Teletest® MK4 System, the FPGA on-chip BRAMs are used to store instructions of MicroBlaze processor. A local memory bus (LMB) controller is configured to access this memory. However, the compiled firmware is too large to fit in the limited amount of FPGA on-chip memory. An external DDR2 SDRAM is added and a multi-ports memory controller (MPMC) is implemented in the FPGA to interface the memory to the MicroBlaze processor. The system architecture and peripherals of the Teletest® MK4 system are explained in Chapter 4.

3.7 Summary

This chapter has reviewed the up-to-date FPGA devices to provide information on the FPGA selection for the Teletest® MK4 system design based on the justification on performance, cost and capability of potential future update. The FPGA architecture and design methods which are essential for improving the performance for the design using VHDL are also covered. The benefit of using FPGA-based processor for the Teletest® MK4 design has been explained. The next chapter will examine how the hardware devices in the system are controlled by the FPGA-based processor and their interfaces to the FPGA.

Chapter 4

Teletest®MK4 System FPGA Peripherals and Interfaces

4.1 Introduction

The FPGA is used to fully control all functions in the new design for the Teletest®MK4 system. The design concept of using an FPGA-based soft microprocessor was explained in section 3.6. The system architecture of the MK4 system new design is entirely different from that of the previous MK3 system. New peripheral devices have been added to the system and these devices are directly controlled by the FPGA. This chapter explains various peripheral devices and how these devices are controlled by the FPGA in the Teletest®MK4 system. It demonstrates that the low cost and low power Spartan FPGA device can be used to successfully integrate various external peripherals of the MK4 embedded system and implement the real-time multichannel transmitter and receiver controllers. This provides flexibility to add new function for the Teletest®MK4 system to support the research of advanced LRUT applications. Section 4.2 provides a system block diagram to identify the system structure and FPGA hardware peripherals; section 4.3 explains the interfaces of these peripherals to the FPGA. The FPGA resource utilization and maximum operating frequency of these peripherals are reported and discussed in Chapter 7.

The Teletest®MK3 system uses an 8-bit 8051 Microcontroller TS80C32X2 [47] which is external to the FPGA. The microcontroller provides overall control of the hardware. It has 8 kBytes on-chip memory for storing operations and data. A Flash In-System Programmable (ISP) Peripherals PSD934F2-90J [48] is used as a configurable memory for the 8-bit microcontroller to provide off-chip memory. It includes 2 MBytes Flash memory and 32 kBytes SRAM. The PSD works with the microcontroller for all the controlling functions. The FPGA in the MK3 system is only used for interfacing the transmitter FIFOs, the receiver ADC and memory to the microcontroller, as well as simple data processing functions, such as accumulation, detection of the maximum and minimum values of the received data. A hardware system block diagram for the Teletest®MK3 system is provided in Appendix A.

In the Teletest®MK4 FPGA embedded system design, the FPGA device was specified as the central controller of the system. The soft microprocessor MicroBlaze is entirely implemented in the FPGA on-chip resources and the peripherals which are external to the FPGA need to be interfaced to the FPGA. FPGA on-chip BRAM resources are allocated to provide 4 kBytes local memory for the MicroBlaze. External memories and peripherals are added to meet the system requirements and improve the system performance. These external devices are interfaced to the FPGA and the interface controllers are implemented using the FPGA resources. The interface controller are usually referred to as FPGA IP cores which interface the MicroBlaze on the FPGA to the external peripheral devices such as DDR2 SDRAM memory, Flash memory, GPS device and Ethernet device, The FPGA interfaces for the commonly used devices are pre-designed and implemented as FPGA IP cores by the FPGA vendors. They can be added to the FPGA system directly. In Teletest®MK4 system, a number of such IP cores from Xilinx®are used, such as memory controller for DDR2 SDRAM, UARTs, Ethernet MAC, SPI, IIC. This reduced the development time. It needs to be emphasized that the MicroBlaze implemented in the FPGA provides the flexibility of adding various peripherals while in the MK3 system, the number and types of the peripherals can be added are constrained by the 8051 microprocessor which only has four 8-bit IO ports.

4.2 Teletest®MK4 FPGA Peripherals

FPGA peripherals are hardware devices that are connected to the FPGA. They are used to expand the capability of the FPGA device. The number of peripherals that can be added to an FPGA device depends on the interface of each peripheral and the total available number of the FPGA I/O pins. The Spartan-3A DSP 1800A FPGA device has four packages: Chip-scale Ball Grid Arrays (CS) CS484, CSG484

(Pb-free), and Fine-pitch Ball Grid Arrays (FBGA) FG676, FGG676 (Pb-free). The former two packages have 309 usable IOs and the latter two have 519 usable IOs. Teletest®MK4 system requires a number of peripherals and the FGG676 package was selected to provide sufficient IO pins and meet the environmentally friendly lead-free (Pb-free) standard.[49]

The peripherals required by Teletest®MK4 system are listed in Table 4.1.

Table 4.1: List of MicroBlaze peripherals for Teletest®MK4 System

Peripherals	**Description**
SRAM x 3	Memory for storing received data from 24 ADCs
DDR2 SDRAM	Memory for running firmware
DAC x 24	DACs for transmitter waveform generation
ADC x 24	ADCs for converting 24-channel received data
Ethernet PHY	Communication port to PC
SPI Flash Memory	Memory for storing FPGA configuration and compiled firmware
Temperature & Voltage Sensors	ADCs for PCB boards temperature and voltage measurements
CPLD x 4	CPLDs for individually controlling 24-channel receiver gains.
Non-volatile RAM	Memory for storing tooling information such as running time and pump cycles
Front Panel	Front panel 8051 controller for manually control and monitor system parameters
PC System Debug UART	Interface for PC debug
GPS	GPS device for tracking the location of the system globally
Watchdog Timer Reset	If the MicroBlaze system is not responding for a certain time; the watchdog timer will automatically reset and restart the system.

The block diagram of the Teletest®MK4 FPGA peripheral is given in Figure 4.1.

4.3 Teletest®MK4 FPGA Peripheral Interfaces

The peripheral controllers for the external devices shown in Figure 4.2 are implemented using the FPGA resources. Xilinx®EDK provides pre-designed IP cores (IP signifies that they are the intellectual property of Xilinx®) for

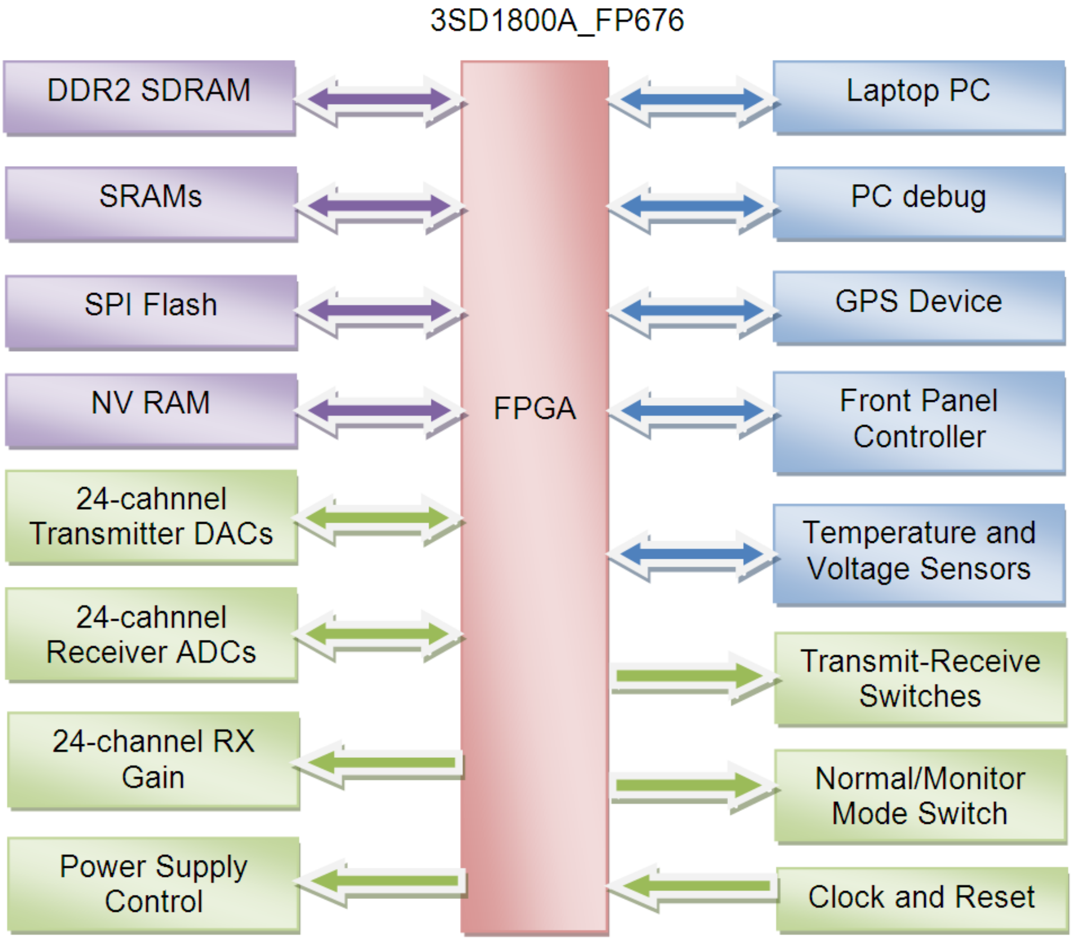


FIGURE 4.1: Teletest®MK4 System Block Diagram

interfacing commonly used hardware devices to the FPGA-based processors. In the Teletest®MK4 design, these interfaces share the MicroBlaze processor local bus (PLB). MicroBlaze is the master and all peripherals are slaves. Each peripheral is assigned a number of address locations. An address decoder is used which allows each slave device to be selected by the master processor. Some peripherals are application specific and custom IP cores have to be designed and implemented to control these hardware devices. The PLB bus is the system bus connecting the MicroBlaze processor to various peripherals. It is a high-performance 128-bit data, 32-bit address bus. It is of the IBM CoreConnect library. The Local Memory Bus (LMB) bus is a 32-bit bus specifically designed for MicroBlaze to connect the on-chip memory in a single master-slave mode. The system bus in the Teletest®MK3 system is only 8-bit, the microcontroller and all peripherals are connected to this 8-bit wide bus, while the PLB bus in the MK4 system is 32-bit and can provides four times faster data access speed for the MicroBlaze and its peripherals.

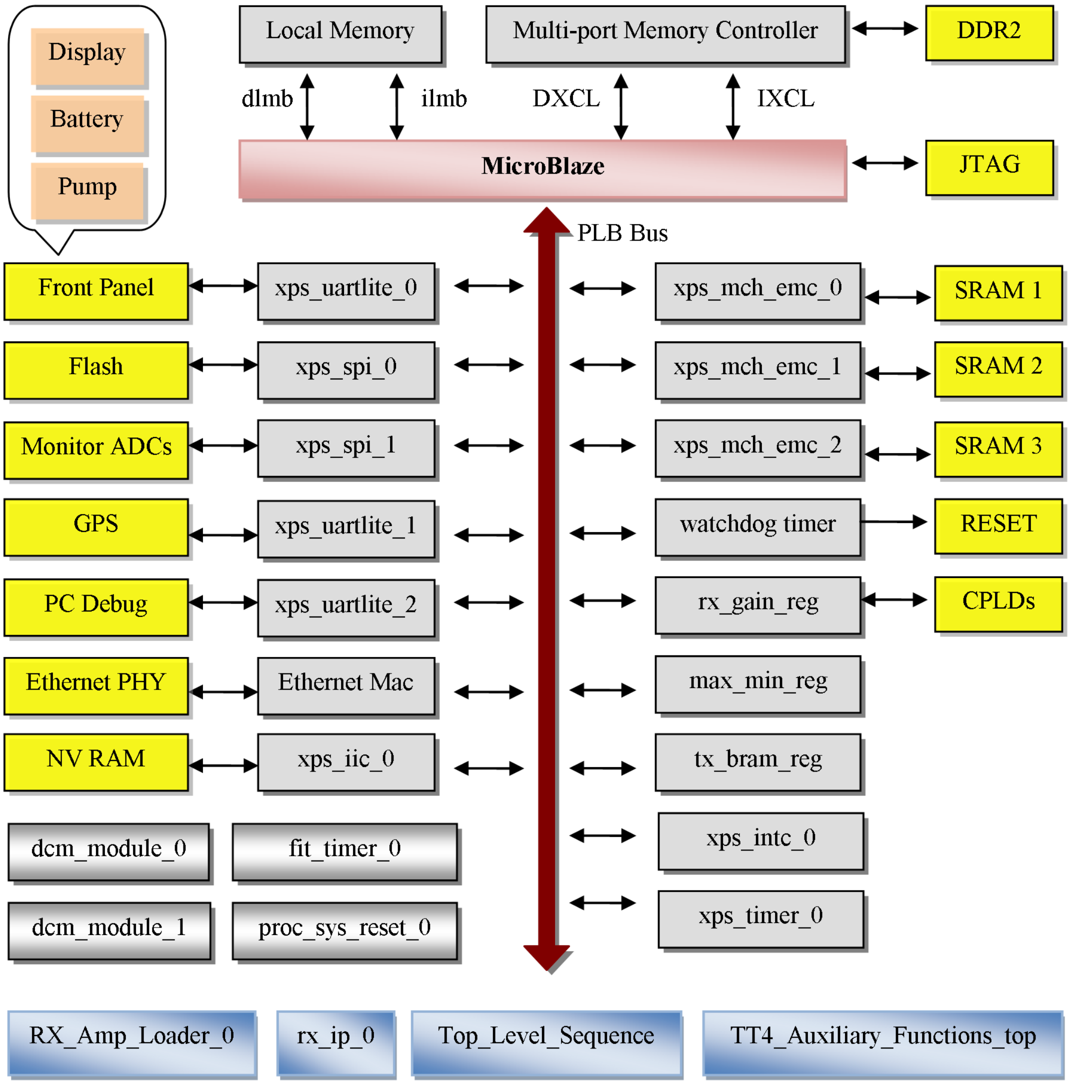


FIGURE 4.2: Teletest®MK4 FPGA IPs and Peripherals Block Diagram

The ADCs and DAC interfaces are not available as MicroBlaze peripherals and are not included in the Xilinx®EDK peripheral library. Therefore custom IPs were designed using VHDL for interfacing these parallel external devices to the FPGA. These devices are used for the parallel transmit and receive sequence. A peripheral designed for a single channel DAC or ADC device can be reused and duplicated for multichannel DACs and ADCs. This is more efficient than the single thread MicroBlaze processor controller due to the advantage of FPGA parallelism. The interfaces for the SRAM memories are custom IP created for real-time transmit and receive sequence. These SRAMs are also connected to the PLB bus for MicroBlaze to access the received data.

Figure 4.2 provides a block diagram with all FPGA peripheral controllers. The block diagram is provided to illustrate the relationship between the FPGA peripheral hardware devices and their respective interface controller implemented in the FPGA design.

The peripheral hardware devices are divided into two groups with the following perspectives:

- The MicroBlaze implemented on the FPGA controls the peripheral devices through standard peripheral controller IP cores which are also implemented on the FPGA. These devices are shown in grey and their corresponding interface controllers are shown in yellow in the block diagram.
- The multi-channel transmitters and receivers run in parallel during acquisition sequence and the related hardware devices are controlled directly by FPGA acquisition logic in order to meet critical time requirement. These devices are shown in blue in the block diagram.

The MicroBlaze embedded system is interfaced to the data acquisition system via a list of shared read/write registers.

4.4 MicroBlaze Controlled Peripherals

The MicroBlaze controlled hardware devices includes external memory and peripherals, the memory controllers and peripheral controllers are all implemented on a single FPGA device. There are four types of external memory devices in this system. The external memory devices are DDR2 SDRAM, Flash memory, Non-volatile RAM and three banks of SRAM. The hardware peripherals are temperature and voltage monitor ADCs, the Ethernet port linking to remote control PC, and UART ports linking to the GPS device and front panel 8051 controller. The front panel controller controls pumps, LED, and battery directly. In the Teletest$^{®}$MK4 MicroBlaze embedded system, the peripherals use the same PLB bus interface and are slaves on the bus. The MicroBlaze controller is the only master on the PLB bus. A dedicated address location is allocated to each peripheral in the system so that the MicroBlaze can ‘recognize’ each peripheral. Table 4.2 shows the memory address map of the Teletest$^{®}$MK4 MicroBlaze embedded system. Microprocessor control software is needed to control these peripherals. This map was passed onto the firmware design engineer to program the firmware for the MicroBlaze system.

Table 4.2: Teletest®MK4 MicroBlaze Embedded System Peripheral Address Memory Map

Peripheral Interface	**Base Address**	**High Address**	**Size (Bytes)**
dlmb_cntlr	0x00000000	0x00003FFF	16k
ilmb_cntlr	0x00000000	0x00003FFF	16k
Ethernet_MAC	0x80000000	0x8000FFFF	64k
xps_uartlite_0	0x81000000	0x8100007F	128
xps_uartlite_1	0x81001000	0x8100107F	128
xps_uartlite_2	0x81002000	0x8100207F	128
xps_spi_0	0x81400000	0x8140007F	128
xps_spi_1	0x81401000	0x8140107F	128
xps_iic_0	0x81800000	0x818001FF	128
xps_timebase_wdt_0	0x81A00000	0x81A0000F	16
xps_gpio_0	0x81B00000	0x81B001FF	512
xps_gpio_1	0x81B01000	0x81B011FF	512
xps_intc_0	0x81C00000	0x81C0001F	32
xps_timer_0	0x83C00000	0x83C0FFFF	64k
mdm_0	0x84400000	0x8440FFFF	64k
tx_bram_reg20_0(Registers)	0xA0000000	0xA000FFFF	64k
rx_gain_reg24	0xA1000000	0xA100FFFF	64k
tx_bram_reg20_0(BRAM)	0xA2000000	0xA202FFFF	192k
SRAM_1	0xA3000000	0xA33FFFFF	4M
SRAM_2	0xA3400000	0xA37FFFFF	4M
SRAM_3	0xA3800000	0xA3BFFFFF	4M
DDR2_SDRAM	0xC0000000	0xC7FFFFFF	128M

4.4.1 DDR2 SDRAM memory

Two DDR2 SDRAM memory chips are used as 32-bit wide memory to provide 128$MBytes$ fast accessible memory for the 32-bit MicroBlaze processor. The memory is used for running firmware applications and its full memory range is cache memory for MicroBlaze. The Multi-Port Memory Controller (MPMC) IP implemented in the FPGA interconnects to MicroBlaze via Xilinx®Cache Link (XCL) bus and provides the memory device with low-level control signals. The memory device uses 125MHz differential clocks generated by Digital Clock Manager (DCM), which is also implemented on FPGA, and the maximum memory access speed is 250MHz. [50, 51, 52]

4.4.2 SPI Flash Memory

The flash memory is used to store the FPGA configuration bitstream and firmware applications. It has SPI interface working at $31.25MHz$ clock rate. The Atmel®SPI Flash memory provides $64Mbit$ and has low power dissipation.

4.4.3 Non-volatile RAM

This NV RAM is located in the remote tool lead (connector to the transducer arrays) to store the test information for the permanent mount application. A transistor buffered IIC interface is provided in the embedded system. An IIC controller IP is implemented on the FPGA and provides the $400kHz$ serial clock to the device.

4.4.4 SRAMs

The SRAM devices provide 12 MBytes of memory for storing received signals. There are three identical banks of SRAM. Each bank stores data for 8-channel receivers. Xilinx®Multi-Channel (MCH) PLBV46 external memory controller is implemented on the FPGA for MicroBlaze to access the SRAM via the PLB. Three controllers are used for three banks of SRAM. MicroBlaze can only access one bank at a time via PLB. In order to access all 24-channel receivers during the transmit and receive sequence, these three SRAM banks are also controlled directly by FPGA state machine sequencer logic, which is custom designed in VHDL.

4.4.5 Power Supply Voltage and Temperature Monitor ADCs

There are various power supplies in the system: $\pm 5V$, $+3.3V$, $\pm 150V$ and $\pm 24V$. These power supplies are monitored and the measured value is digitised using an 8-channel ADC with an SPI interface. This is an SPI controller IP implemented on the FPGA and it works at $1.953125MHz$ ($1:32$ of the $62.5MHz$ MicroBlaze system clock rate). Thermocouples are used for measuring temperatures of the Power Supply Unit (PSU) and the TX Amplifiers. A second 8-channel ADC with an SPI interface is used to digitise the temperature measurements. The same SPI controller is shared by both ADCs.

4.4.6 Ethernet PHY

The single chip Ethernet Physical layer transceiver (PHY) provides a 10/100 $Mbit/s$ (Mega bit per second) communication link between the embedded unit and the remote control PC. The related Ethernet MAC controller is implemented in the FPGA and is interfaced to MicroBlaze via PLB bus.

It is suggested that high speed 1 GMbit/s (Giga bit per second) Ethernet PHY and hardcore Ethernet MAC should be used for future development so that the overall communication speed between the embedded unit and the remote control PC can be increased. [53, 54]

4.4.7 Front Panel Controller

The 8051 front panel controller monitors the battery status, directly controls the front panel buttons and LED display, as well as the operation of the in-system inflation pump. It is interfaced to the FPGA via UART. Another UART is added to connect the FPGA directly to the GPS controller which is previously connected to the front panel in order to simplify the firmware control process.

4.5 Acquisition Logic Controlled Peripherals

The main reason to design the custom sequence controller rather than using the MicroBlaze controller is to meet the critical timing constraint of the 24-channel parallel transmitters and receivers in the Teletest®MK4 system. Although the MicroBlaze operates at 62.5MHz system clock, it has to firstly fetch an instruction from the memory and decode the instruction. This is a serial process which takes a few system clock cycles. The MicroBlaze controls the peripherals which do not have real-time requirement. These peripherals have already been explained in the last section. The custom coded sequencer has the advantage of being predicable since each task is scheduled in a certain order in the real-time operating sequence.

The acquisition logic controls the transmit and receive sequence to drive 24 channels with identical hardware components. Each channel is composed of: a DAC, a transmitter amplifier, a transmitter and receiver switch, a receiver amplifiers and filters, and an ADC. Four CPLD devices are used for 24-channel receiver gain and filter control. The functional parameters of these devices are set up by the embedded system. In a normal test routine, the remote PC sends messages with test request and test functional parameters to the embedded system, which decodes these messages, setups the acquisition parameters, and then requests the acquisition logic to run the transmit and receive sequence. The collected signal is stored in the embedded system local memory. On the completion of a test, the PC can request the MicroBlaze embedded system to send the received data to the PC for display and analysis.

A custom control sequencer is needed to control the multi-channel parallel transmit and receive procedure. It needs to switch on and off the power supply for various devices following a pre-defined sequence in order to reduce overall power consumption; and to control the DAC, High Tension (HT) transmitter amplifiers, ADCs, receiver filters and amplifiers, as well as the SRAM, which is used as data repository. The HT power supply and DAC power supply are only switched on during transmitting in order to reduce system power consumption. The ADC power supply is only switched on during transmitting and receiving in order to reduce system power consumption. Since accumulation over multiple acquisitions is required and further DSP function is implemented by FPGA, it is necessary to store the received data in the system local memory.

4.5.1 Transmit Waveform Generation DAC

The 24-channel DACs have 16-bit resolution and 24 serial data interfaces to the FPGA. They are driven in parallel by a custom DAC controller implemented on the FPGA. These DAC devices have integrated interpolation filters with adjustable interpolation rate of x2, x4, x8 and x16.

In LRUT applications, transmit waveforms are filtered by the capacitive load of piezoelectric transducers; therefore the DAC resolution is not critical for generating smooth signals.

4.5.2 HT amplifiers for Transmit Waveforms

There are 24-channel TX amplifiers corresponding to 24-channel DACs. Each TX amplifier is capable of driving a maximum of 16 piezoelectric transducers up to 300 V peak-to-peak with arbitrary waveforms within a bandwidth between $10kHz$ to $300kHz$. The equivalent load of a typical transducer is a 950 pF capacitor in serial with a 68 Ω resistor.

4.5.3 Transmit-receive Switches

During the transmitting time, the adjacent transducers cannot start receiving, as the cross-talk between the transducers will cause the receiving channels to pick up the very high amplitude signals driving the transmitter, which saturates the ADCs. This period of time is referred to as the receiving ‘dead zone’. However, this does not affect pitch-catch application, in which transmitter and receiver are separated by a certain distance.

The length of test dead zone depends on the length of transmitting signal and the transmitting sampling frequency, therefore, the number of transmitting signal samples has to be limited in order to avoid a long dead zone. The 24-channel transmit-receive switches control whether the transmitters or receivers are switched to the PZT transducers. They can be individually set up with two optional configuration modes.

Mode 1: The switch is switched to transmitter at the start of an acquisition. On completion of waveform transmitting, the switch is switched to receiver.

Mode 2: The corresponding channel does not transmit and is switched to receive at the start of an acquisition.

4.5.4 Receiver ADCs

These 12-bit ADCs have serial data interfaces and a fixed serial clock rate of 20 MHz. The wideband noise level in the receive signal makes it unnecessary to use ADC with higher resolution. Four converting rates options are available: 1 MSPS, 500 kSPS, 250 kSPS, 125 kSPS. This must be chosen based on the transmit waveform frequency. The RX converting rate must be at least double the transmit waveform frequency and it is suggested to be 10 times of the transmit waveform frequency in order to finely reconstruct the received signals. The merit of using a lower RX sampling frequency is to increase the test distance. As each RX channel has 128 k samples memory, which can capture data for 128 ms at 1 MSPS, or 1024 ms at 125 kSPS.

4.5.5 Receive Amplifiers and Filters

The 24-channel receive amplifiers can be configured individually with 1 dB step from 20 dB to 100 dB. The variable gain adjusts the received signal to full scale of the ADC input in order to achieve a better digitization resolution.

A series of first order RC filters with 300 kHz, 150 kHz, 75 kHz, 32 kHz and 16 kHz cut-off frequencies can be configured in common for all receiver channels. These filters are used to restrict bandwidth of the received signal to satisfy the Nyquist-Shannon Theory. The cut-off frequency is set to be less than the Nyquist frequency (half of the sampling frequency) to suppress the mirror image and harmonics of the signals as well as broadband noise. The signal and its mirror image are symmetrical about the Nyquist frequency. An ideal filter would have a zero transient band in the spectrum of the signal; however, in practice a simple first order RC anti-aliasing filter is used and it has a long transient band. The cut-off frequency, also called -3dB frequency, is the frequency where the signal amplitude is -3dB of the pass band amplitude.

It needs to be noted that various gain amplifiers and digital filters can be used to replace current discrete components to reduce the size and power consumption of analogue circuitry for future development. The advantage of the discrete component analogue amplifiers and filters is the low power consumption. The disadvantage is that a large number of components are used to build the amplifier and filter circuits. 11 control pins are needed for each receiver channel. Therefore, four CPLDs are used to extend IO ports for the FPGA to control RX amplifiers and filters. CPLD is conventionally used for low budget, simple function operations. Each RX channel can be configured individually with an 8-bit gain registers, while the filters for all 24-channel RX are setup in common.

4.6 Summary

This chapter described the hardware architecture, FPGA peripherals and interface controllers of the Teletest®MK4 system. The hardware architecture of the Teletest®MK4 system is entirely different from that of the previous MK3 system. The system performance has been improved in the MK4 system in terms of speed, size and power consumption. The new added external peripheral devices controlled directly by the FPGA include DDR2 SDRAM, Ethernet MAC, SPI Flash Memory, increased SRAM memory, serial interface ADCs and interpolation DACs. This design provides the flexibility required for adding new functions to the Teletest®MK4 system to implement advanced LRUT applications. The MicroBlaze controlled peripherals are interfaced to the FPGA using manufacture pre-designed IP cores. These general IP cores had to be configured for the Teletest®MK4 system. The multichannel transmitter and receiver controllers were designed by the author using VHDL and integrated in the system independent to the MicroBlaze processor. Chapter 5 will discuss the design for the multichannel transmitter controller. Chapter 6 will discuss the design for the multichannel receiver controller. Chapter 7 will provide the FPGA implementation result including resource utilization and maximum operating frequency for all interface controllers.

Chapter 5

Multichannel Transmitters Design

This chapter explains the design and implementation of the FPGA controller for the Teletest®MK4 multichannel transmitters and demonstrates the improvements made by the new design against the previous MK3 design. Firstly, the use of the interpolation DAC and the FPGA on-chip BRAM resource reduces the size, component count and power consumption. Secondly, in this design, the BRAM is configured as dual port memory with serial data output to replace parallel to serial converter which otherwise would consume FPGA logic resources. Section 5.1 introduces the transmitting waveform generation for the Teletest®MK4 system. Section 5.2 explains the hardware architecture of the transmitter system including FPGA Dual Port BRAM, DAC and high voltage amplifier. Section 5.3 explains the DAC controller and transmitter control sequence. Section 5.4 demonstrates the test result of the transmitter. Section 5.5 summarizes the multichannel transmitter design.

5.1 Waveform Generation

It was introduced in Chapter 2 that the Teletest®system uses a tone burst transmitting waveform, which is usually a number of sine wave cycles with a Hann of window for controlling the bandwidth. Figure 5.1 shows a 10-cycle, 200-points sine wave tone burst plotted in solid blue line, a Hann window plotted in dashed red line and the tone burst sine waveform constrained by the Hann window plotted in a solid purple line. This is the default waveform used by Teletest®MK3 system.

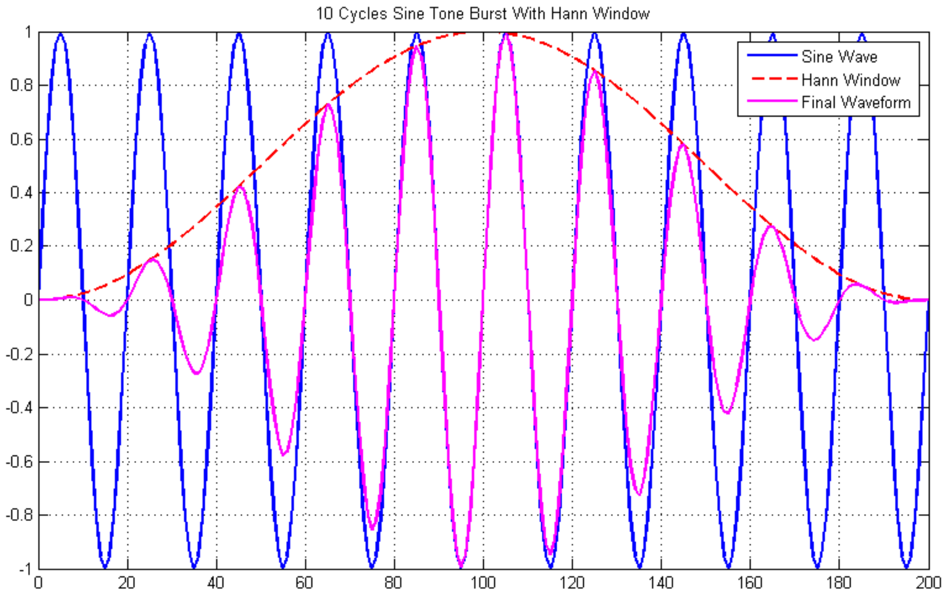


FIGURE 5.1: Transmitter Waveform Generation

This waveform is generated using Equation(5.1). The Hann function is named after the Austrian meteorologist Julius von Hann.

$$x(n) = \frac{1}{2} * sin(2 * pi * cycle * n/N) * (1 - cos(2 * pi * n/N)) \tag{5.1}$$

Where

- N is the number of samples of the digital waveform;
- n is the index running from 1 to N step by 1;
- cycle is the number of sine wave cycles constrained within the window.

Each cycle of the sine wave is constructed by N/cycle samples. According to Nyquist-Shannon sampling theory [55], the sample number per cycle needs to be equal or greater than two, practically, this number should be equal or greater than 10 to allow a smooth reconstruction of the waveform. In LRUT testing, the length and shape of the transmitting waveforms affect the energy and power consumption of the system; and the sampling frequency affects the signal condition of the transmitting waveform.

5.1.1 Transmitting Waveforms

The time taken for the pulse to be transmitted is defined by the wave period and the number of cycles. The wave period used for LRUT is in the range of 100 μs down to 10 μs, corresponding to the frequency range of 10 kHz to 100 kHz. The selection of UGW frequencies for a LRUT test is based on the calculation of dispersion curves as discussed in Chapter 2. This calculation is done by the specially designed software and is not within the scope of this thesis.

When a tone burst waveform is transmitted at a certain frequency without the smoothing window applied, the output energy is in proportion to the number of sine wave cycles. When the attenuation is high in the ultrasound propagation medium, the energy of the transmitted UGWs can be increased by increasing the number of cycles of the tone burst waveform to achieve a longer transmission distance. However, large number of cycles leads to a long transmitting waveform length, long transmitting time and more power consumption. A long transmitting waveform also has low test resolution. The commonly used and default configuration for the Teletest®system is 10 cycles sine wave tone burst with a Hann window applied. A lower cycle number can be used for a high resolution test, while pulses with more cycles can be used to minimise the frequency bandwidth [55]. The maximum likely pulse length is 1.6 ms(milliseconds) for time reversal [56] applications. This defines the required memory size for the Teletest®MK4 system. The FPGA on-chip BRAMs are used to store the 24-channel transmitter waveforms.

The window function is applied on the sine wave tone bust for two reasons. The first is to narrow the bandwidth and suppress the side lobes in the frequency domain. Figure 5.2 shows the spectra in the frequency domain for the tone burst waveforms without Hann window, the blue line; and with Hann window, the red line. The spectra are plotted using normalised frequency as the x-axis; linear magnitude in the left diagram and logarithmic magnitude (dB) in the right diagram as y-axes respectively. These spectra are generated based on the waveforms plotted in Figure 5.1, which are zero-padded to 1024 samples for the Fast Fourier Transform (FFT). The reason for using 1024 samples for the FFT is to get a better display resolution for the spectra without requiring too much calculation time. A short pulse has a wide bandwidth and can cause more UGW dispersion. It can be seen in the spectra that by applying Hann window to the waveform, the unwanted side lobes have been substantially suppressed. The second reason of using window function is to allow the transducers to be driven by gradually increasing peak amplitudes to protect the piezoelectric elements.

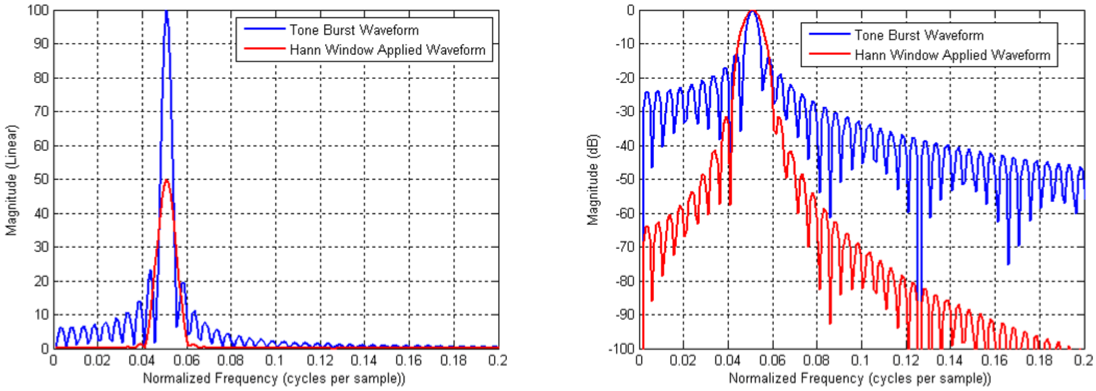
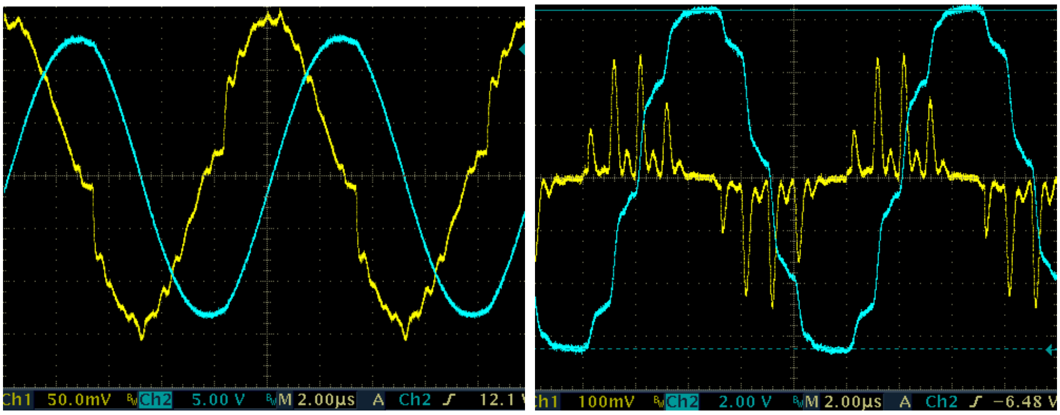


FIGURE 5.2: Spectra of Transmit Tone Burst Waveform and Hann Windowed Waveform

5.1.2 Sampling Frequency

The transmitters of Teletest®system are built with high power amplifiers that can drive the transducers with capacitive load up to 300 Vpp . When a low sampling frequency is used, a large voltage difference can be generated between adjacent samples. This can cause very high current across the transducers. The over current protection circuit will be triggered to force the system to shut down in order to protect the transmitter circuit. In the Teletest®MK3 system, the 24 parallel DACs are clocked at 10 MHz sampling frequency to avoid the surge current of the high power amplifiers. Experimental results, below, show that when 1 MHz sampling frequency is used to reconstruct a 100 kHz signal, the transmitting waveform is distorted over a 16 nF pseudo capacitive load. The 16 nF capacitive load is the maximum load of one transmitting channel for both MK3 and MK4 system. The following experimental result on the Teletest®MK3 system is presented to highlight the challenge in the MK4 FPGA design for the transmitter controller, which resolves the transmitter over-current problem by using interpolation filters rather than using high transmitter sampling frequency.

Figure 5.3(a) and Figure 5.3(b) show the amplifier outputs of the transmitter reconstructed at 10 MHz and 1 MHz clock rates respectively. These diagrams are captured by an oscilloscope. Figure 5.3(a) shows the output voltage in the blue trace with 50 V/div (Volts per division) and the load current in the yellow trace with 0.5 Amps/div at 10 MHz clock rate. The current waveform is fairly smooth with a peak value of 1.5 Amps for an output voltage of 260 Vpp. Figure 5.3(b) shows the output voltage in blue trace with 20 V/div and load current in the yellow trace with 1.0 Amps/div at 1 MHz clock rate. The peak current of the waveform reaches about 2.5 Amps when the output voltage is only 128 Vpp. The large output waveform voltage variance applied on the 16 nF capacitive load results in a surge of current and causes premature over-current in the transmitting circuit. The over-current protection is activated at 130 Vpp at 1 MHz transmit clock and cannot get to the required 300 Vpp. Therefore the 10 MHz clock rate is used in the Teletest®MK3 system.



(a) Transmit Waveform at 10MHz. TX waveform Reconstructed at10 MHz clock. TX output voltage at 50 V/div (Blue Trace). Load current at 0.5 Amps/div (Yellow Trace)

(b) Transmit Waveform at 1MHz. TX waveform reconstructed at 1 MHz clock. TX output voltage at 20 V/div (Blue Trace). Load current at 1.0 Amps/div (Yellow Trace)

Figure 5.3: Transmit Waveform Overcurrent Effect of Low Transmitter Sampling Frequency

The drawback of using the 10 MHz clock rate is that the number of samples needs to be stored for the same transmitting waveform is ten times larger than using 1 MHz clock rate. Therefore 24 FIFO devices are used to store the transmitter waveforms in the Teletest®MK3 system. Each FIFO can store 16k by 9-bit wide samples. This gives the system the ability to transmit continuously for a maximum period of 1.6 ms at 10 MHz clock rate. The footprints of the 24 channels FIFO (CY7C462A-15JC) and DAC (AD9760AR50) devices take at least 9185 mm^2 board area excluding PCB routing. This is reduced by 91% in the Teletest®MK4 desgin. These devices took up a whole PCB board in Teletest®MK3 system and the total operating power dissipation is 28.4 W. This is reduced by 70% in the MK4 desgin.

In the Teletest®MK4 system, the FPGA-based multichannel transmitter controller design has reduced size and power consumption in comparison with that of the Teletest®MK3 system. The FPGA on-chip BRAM replaces the external FIFO devices. A small footprint DAC8580 with integrated configurable interpolation filters is used. The Spartan 3A DSP 1800A FPGA has 84 BRAMs; each has 1k x 18bit memory. In the Teletest®MK4 system, two BRAMs are used to store up to 2k x 16-bit samples for one transmitter channel. The transmitter memory has to be reduced from 16k samples per channel in the MK3 system to 2k samples per channel to fit in the FPGA on-chip BRAM. Large BRAM memory is available in a number of high profile FPGA devices, which are also large in size and more expensive. The author is limited to the project budget and cannot use these FPGA devices because of the high cost and high power consumption of these devices. The author therefore has to use a novel design to implement the arbitrary waveform transmitter on a low cost FPGA to get the same performance. This increased the challenge of the design task. As mentioned previously, the reduced number of samples per cycle for the transmitting waveform results in a large voltage step between each two samples, causing sudden current surge in the transmitter circuit. The interpolation function of DAC8580 was used to produce smooth transmit waveforms from a limited number of samples.

5.2 Hardware Architecture

The hardware architecture of the Teletest®MK4 transmitter circuit is illustrated in Figure 5.4. The 24-channel transmitting waveforms are generated by the PC and transmitted to the FPGA via the Ethernet port. The waveforms are stored in the FPGA on-chip BRAMs. The DACs convert the digital waveforms into analogue signals. The high power amplifiers amplify the DAC output up to 300 Vpp, up to 3 App (Amp peak-to-peak) to drive the transducer arrays.

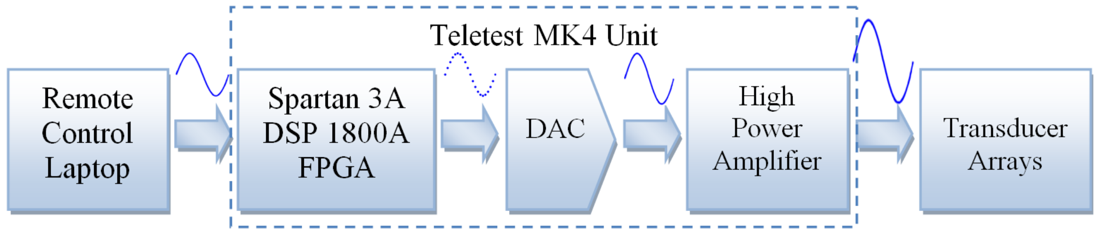


FIGURE 5.4: Teletest®MK4 Transmitter Hardware Architecture

5.2.1 FPGA Dual Port BRAM

In the Teletest®MK4 system the transmitter FIFO BRAMs need to be accessed by both the MicroBlaze processor and the transmitter controller finite state machine (FSM). The dual port BRAM control method is illustrated in Figure 5.5. The data samples of the transmitting waveforms are transmitted from the PC to the Ethernet MAC buffer implemented on the FPGA. These 16-bit samples are arranged from left to right (bit 15 down to bit 0), that is, the Most Significant Bit (MSB) is bit 15 on the far left, and the Least Significant Bit (LSB) is bit 0 on the far right. The MicroBlaze processor reads the Ethernet buffer and writes the samples into the transmitter FIFO BRAMs. Each transmitter channel is assigned with one transmitter FIFO BRAM which is connected to the PLB bus. The PLB bus is a 32-bit bus (arranged from LSB bit 0 to MSB bit 31) and BRAM data port A is connected to the upper 16 bits (bit 16 to bit 31) of the PLB bus. The clock of BRAM port A is connected to the 62.5 MHz system clock used by MicroBlaze processor and PLB bus. Each transmitter FIFO has 2048 memory addresses, therefore the address is 11-bit wide (bit 10 down to bit 0), which is connected to the address port of the PLB bus. The total memory for each transmitter FIFO is 32768 bits or 4096 bytes. Each FPGA BRAM has 18k bits memory. Therefor two BRAMs are required for each transmitter FIFO. The locations of the 24 transmitter FIFO blocks are adjacent in the MicroBlaze memory map, and the waveform for each channel is written into each FIFO sequentially.

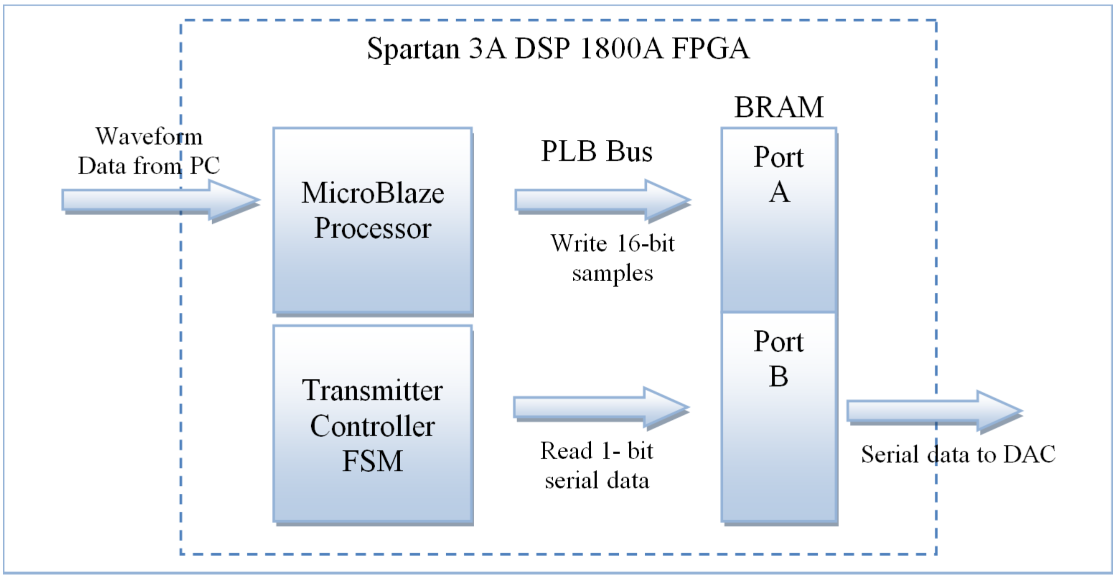


Figure 5.5: Teletest®MK4 FPGA Dual Port Block RAM

The transmitter controller FSM is designed for controlling the 24 transmitter channels to transmit waveforms in parallel. The DAC8580 requires serial data input and the MSB - bit 16 needs to be clocked in the DAC first. Therefore the waveform samples stored in the BRAM need to be serialized before transmitted to the DAC. The implementation of serialization of 16-bit samples for 24 channel transmitters requires at least 384 FFs. These FPGA resources were saved by configuring the port B of the BRAMs to have 1-bit wide output data port and corresponding 15-bit wide address port. The sampling frequency is 1 MSPS and each sample has 16 bits, hence the serial data clock for BRAM port B is 16 MHz. Port A and Port B of the BRAM are not accessed at the same time during system operation.

The Xilinx Block Memory Generator was used to generate the dual port BRAM. The use of IP core generator is recommended by Xilinx to improve the efficiency of the FPGA resource usage because the tool can automatically optimize the design according to the custom configuration. Table 5.1 lists the parameters setup for the dual port BRAM. The BRAM IP core is instantiated 24 times in the transmitter controller FSM. (FPGA manufacturers use the term 'instantiation' to describe an instance or a call up function of an implementation. To instantiate is to make a copy of an instance. In FPGA hardware design, it is common to use multiple copies of an implementation; each copy is an instance.)

Table 5.1: Transmitter FIFO Dual Port BRAM Generation

PORT A	**PORT B**
Write width: 16	Read width: 1
Write depth: 2048	Read depth: 32768
Address: 11-bit [10:0]	Address: 15-bit [14:0]
CLKA: 62.5 MHz (MicroBlaze Processor clock)	CLKB: 16 MHz (DAC controller Clock)

The external transmitter FIFO device in the MK3 system has a large footprint of 15.11 mm x 12.57 mm and 1.0 W power consumption using 5 V power supply. By using FPGA on-chip BRAMs to replace the external FIFOs, the new design has saved at least 362.64 mm x 301.68 mm PCB board area and 24 W @ 5 V power for the Teletest®MK4 system.

5.2.2 DAC8580

The DAC8580 is a high speed single channel DAC from Texas Instruments. It has 16-bit resolution, serial digital data input and voltage analogue output. The serial input interface can operate up to 30 MHz. This device integrates a programmable digital interpolation filter capable of oversampling the input sample rate by 2, 4, 8 or 16. The filter can be bypassed or permanently turned off. When tuned on, the filter smoothes the waveforms, increases component tolerance and reduces temperature drift. For output signals less than 200 kHz, no analogue anti-imaging filter is necessary. The filter function perfectly suits the design requirement of the LRUT embedded system. The transmit waveform is generated at 1MSPS to reduce the size of the transmitter FIFO implemented using the FPGA on-chip BRAM. The waveform then can be serialized and clocked into the DAC at 16 MSPS for 16-bit sample. The DAC has very small footprint compared to the DAC used in the Teletest®MK3 system as shown in Table 5.2. The DACs in the Teletest®MK4 design only take about 16.6% of the PCB board area of the DACs in the MK3 system. However, the power consumption for both devices is the same: 175 mW. In the MK4 system, the $\pm 5V$ double power rails are used to power the DAC8580 devices. In the MK3 system, only +5 V single power rail is used to power the AD9760 devices. This doubles the total power consumption of MK4 DACs. Custom control logic has been designed for switching off the power supply of the DACs to reduce the power consumption.

Table 5.2: Teletest®MK3 and MK4 Waveform Transmitter Design Hardware Devices

	Part No.	**Footprint**	**Power Consumption**
		Teletest®MK3	
FIFO	CY7C462A-15J [57]	15.11 mm x 12.57 mm	1.0 W @ 5V
DAC	AD9760AR50 [58]	18.10 mm x 10.65 mm	175 mW @ 5V
		Teletest®MK4	
DAC	DAC8580 [59]	6.6 mm x 5.1 mm	175 mW @ ±5V

The internal functions of the DAC8580 are illustrated in Figure 5.6. The DAC converts the serial data into a 16-bit sample, applies an interpolation filter on the waveform samples and generates analogue waveforms. The interpolation filter has four interpolation rates (x2, x4, x8 and x16) and can interpolate one digital sample generated by FPGA up to 16 times. In this manner, 1k samples with x16 interpolation are equivalent to 16 k samples without interpolation. This DAC device also has a serial data interface which needs only one FPGA pin per channel as digital data output port. For a 16-bit DAC with parallel data interface, 16 FPGA pins are needed for the data ports of each channel, and 24 channels will need 384 FPGA pins. The use of a serial port device allows more transmitter channels to be connected to FPGA with reduced FPGA pin requirement. This combination makes the system more compact, reduces the size, complexity and power consumption of the circuit.

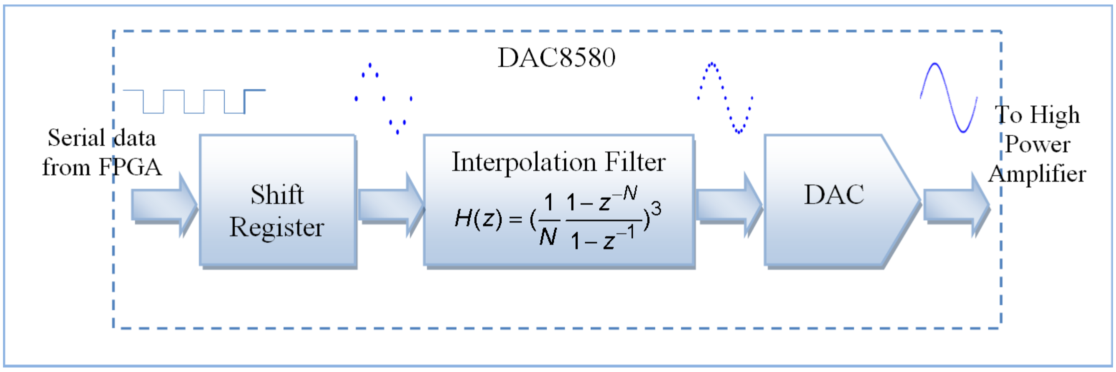


FIGURE 5.6: Teletest®MK4 DAC8580 Interpolation

The DAC8580 digital interpolation filter is a FIR filter with linear phase. It has a two-sample delay independent of the oversampling rate. The Z transform of the filter is shown in Equation(5.8), where N is the interperlation rate; and the x16 interpolation effect is illustrated in Figure 5.7.

$$H(z) = (\frac{1}{N} \cdot \frac{1 - z^{-N}}{1 - z^{-1}})^3 \tag{5.2}$$

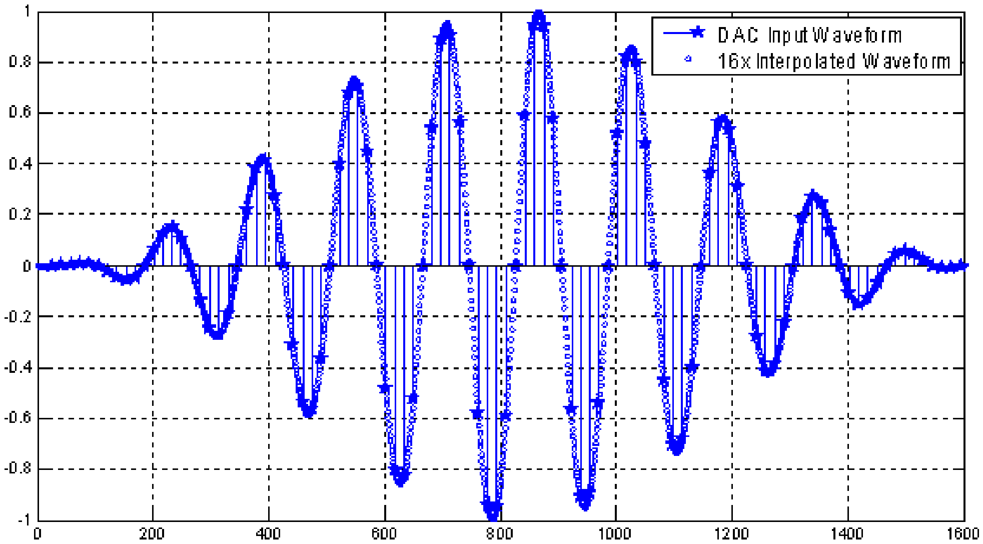


FIGURE 5.7: Test Waveform and 16x Interpolation

The solid stars marked the DAC input waveform using only 10 samples per cycle for the sine wave. The output of the filter will have 15 samples interpolated between each two original samples and the final waveform is much smoother. The filter removes the requirement for adding external analogue filters to smooth the DAC output waveform in the transmitter design. It also reduces the memory size required for transmitter FIFOs, making it possible to use the FPGA on-chip BRAM to replace the external FIFO devices, hence reduces the hardware component count, system size and power consumption. Since the number of waveform samples needed to be generated by the PC is reduced to a fraction of $\frac{1}{16}$ of the MK3 system, the transmitting time and overall testing time is accordingly reduced.

5.2.3 High Voltage Amplifiers

Transmitter Energy and Power over Capacitive Load

The transmitter requires a high voltage power supply to drive the piezoelectric transducers. The Teletest®system uses piezoelectric transducers, which have a capacitive load equal to 1 nF. The impedance of the each transmitter channel depends on the number of transducers driven in parallel by that channel and the frequency of the transmit waveform. In the Teletest®system, each transmit channel can drive up to sixteen transducers. The biggest transducer capacitive load of 16 nF is used in the calculation with the consideration of the worst case scenario. The equivalent impedance of the transducers can be calculated using Equation (5.3). The driving current of the transmitter feeding into the capacitive load is the first derivative of the applied voltage multiplied by the capacitance of the load as indicated in Equation (5.4). The current Equation of the sine wave toneburst and the Hann window applied waveforms are given by Equation (5.5)and Equation (5.6) respectively, which are substituted in Equation (5.7) and Equation (5.8) to calculate the energy and power dissipation.

$$Z_{impedance} = \frac{1}{2 \times \pi \times f \times C} \tag{5.3}$$

$$i(t) = C \times \frac{dV}{dt} \tag{5.4}$$

$$i_{sinewave}(t) = C \times \frac{dV_{sinewave}}{dt} = C \times A \times 2 \times \pi \times f \times cos(2 \times \pi \times f \times t) \tag{5.5}$$

$$\begin{aligned} i_{hann}(t) &= C\frac{dV_{hann}}{dt} \\ &= CA\pi f \left\{ \frac{1}{c} \sin(2\pi f t) \sin(\frac{2\pi f t}{c}) + \cos(2\pi f t) \left[1 - \cos(\frac{2\pi f t}{c}) \right] \right\} \end{aligned} \tag{5.6}$$

$$E = \int_0^T vi = \int_0^T i^2 Z = I_{RMS}^2 Z T \tag{5.7}$$

$$P_{average} = \frac{1}{T}\int_0^T vi = \frac{1}{T}\int_0^T i^2 Z = I_{RMS}^2 Z \tag{5.8}$$

Where Z is the impedance; f is the waveform frequency; C is the capacitance of transducer; A is the amplitude of the waveform; c is the number of sine wave cycles of the waveform; T is the length of the waveform in time; t is the time variable; i is the current; v is the voltage; P is the power; E is the energy.

Figure 5.8 shows how the energy and power of the 10-cycle tone burst sine waveform and the Hann windowed waveform change when frequency increases from 20 kHz to 100 kHz. It can be seen that the window function limits the output energy and power of the transmitter waveform. The energy consumed is independent to the frequency of the transmitter waveform in this case because the frequency of the transmit waveform is inversely proportional to the length of the waveform. The average power increases linearly when the frequency increases and the maximum average power occurs at 100 kHz because it is the highest frequency used by the Teletest®systems, 71.9 W for a sine wave toneburst and 27.2 W for a Hann windowed waveform. The maximum current for both waveforms is 1.508 A, which has to be considered for the power supply circuit design.

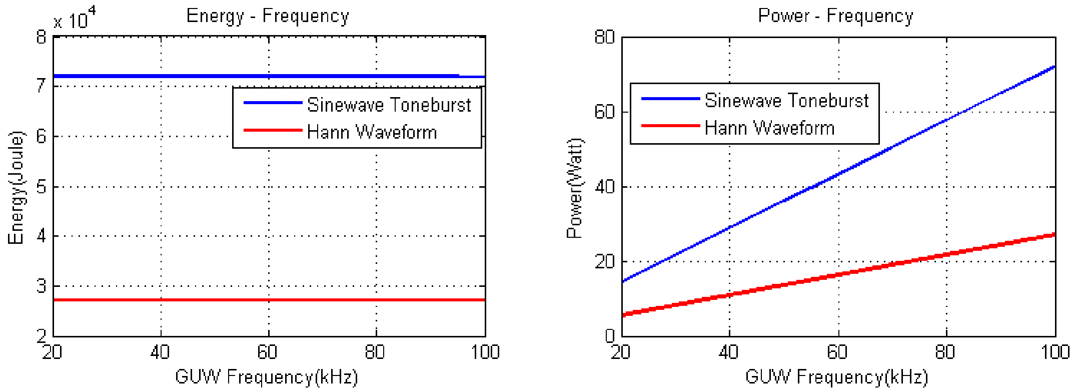


FIGURE 5.8: Transmitter Waveform Energy/Power Verses Frequency

Figure 5.9 shows how the transmitter waveform energy and power change when the number of cycles increases from 2 to 32, and the frequency is set at 50 kHz as a constant parameter for the illustration. The energy dissipation increases in proportional to the number of transmit sine wave cycles. The average power of the transmitter waveform is independent on the number of cycles.

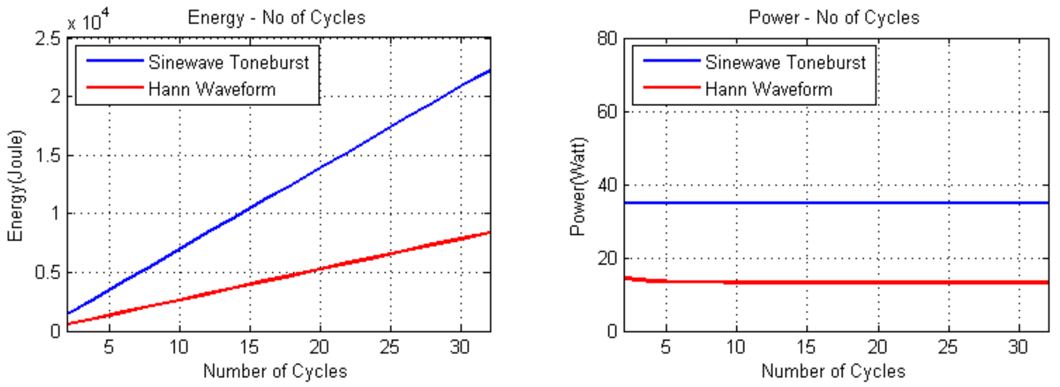


FIGURE 5.9: Transmitter Waveform Energy/Power Verses Number of Cycles

The energy and power dissipation of the 24 channel high power transmitters in the Teletest®MK4 system is a significant proportion of the overall system power consumption. The High Tension(HT) amplifier circuit generates a lot heat when switched on. It is necessary to use control logic to switch on the high voltage power supply only during the transmitting period to reduce the overall power consumption and ease the heating issue. Custom control logic had been designed to switch off the transmitter circuit including DACs and high power amplifiers to reduce the power consumption. The information of power requirement for the waveform transmitters was also used for the power supply and battery design for the Teletest®MK4 system, which was done by another design engineer in the project.

5.3 Transmitter Controller Design

The design of custom IP cores generally involves three steps. The first step is to define the inputs and outputs of the IP core. The second step is to identify the functions and data path. The third step is to develop the controlling sequence using FSM.

5.3.1 BRAM Configuration

The BRAM dual port configuration has 16-bit parallel data input and 1-bit serial data output. The BRAM memory can be either loaded with arbitrary waveform samples or written by the MicroBlaze with new waveforms if required. The serial data output can be used directly by the FPGA DAC controller with no need for the parallel-to-serial converters to be implemented using FPGA logic resource for the 24 transmitter channels. This saves at least 384 FFS and 24 LUTs.

5.3.2 DAC Controller Timing

The DAC control signals are generated by the FPGA. The DAC controller is designed to meet the timing requirements explained in the DAC8580 data sheet [59]. Figure 5.10 shows the DAC8580 controller timing diagram for one channel.

This timing diagram is generated by the author for designing the custom DAC controller to meet the time requirement for the DAC8580 stated in the datasheet [59]. The 1-bit data output of BRAM Port B appears one clock after the address. This is connected to the DAC serial data input port (sdo). The DAC

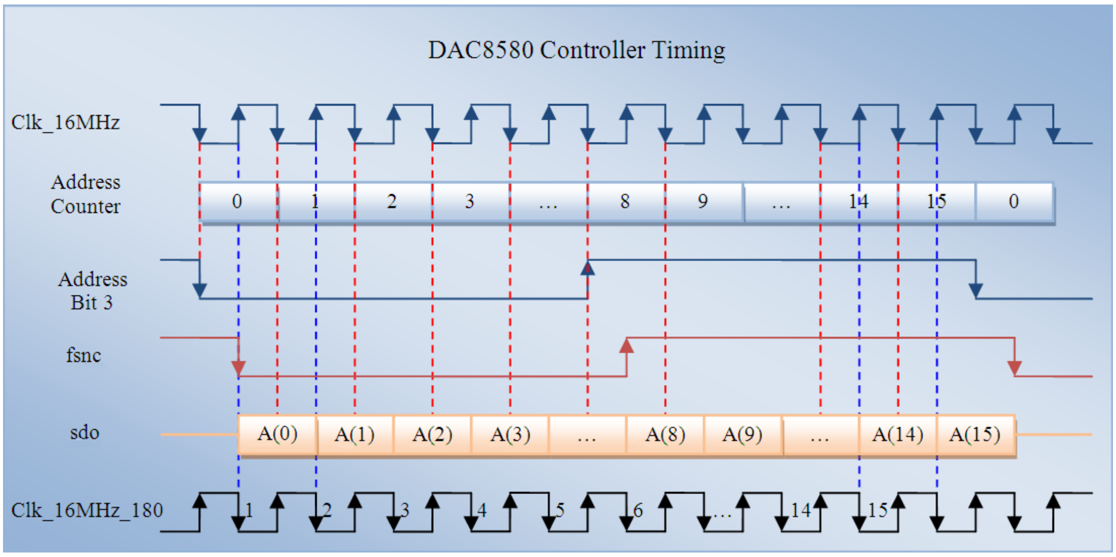


Figure 5.10: DAC8580 Controller Timing Diagram

requires a frame synchronous clock (fsnc) input signal to frame 16 bits into each sample. A new sample starts at the falling edge of the fsnc signal. The Clk_16MHz is the 16 MHz transmitter controller clock signal, used by FPGA for deriving the fsnc signal and counting the BRAM Port B address. Clk_16MHz_180 is an FPGA output clock signal which is 180 degree out of phase to the Clk_16MHz. This signal is connected to the DAC clock input (sclk). This design allows a 180 degree phase shift for address and data and therefore avoids glitches that may occur when address and data are clocked at the same clock edge.

5.3.3 Transmitter Control Sequence

The transmitter control sequence FSM includes 10 states. Table 5.3 lists these states and describes the control logic function of each state.

Figure 5.11 illustrates the block diagram of the transmitter FSM.

5.4 Design Verification

To verify the performance obtained, the arbitrary waveforms were loaded in the FPGA BRAM and the analogue outputs of the DACs were captured using an oscilloscope. Figure 5.12 shows the DAC outputs of a sine wave signals. Only one and half cycles are displayed to illustrate the effect of the interpolation filters. The DAC serial clock is 16 MHz and the sampling frequency is 1 MSPS.

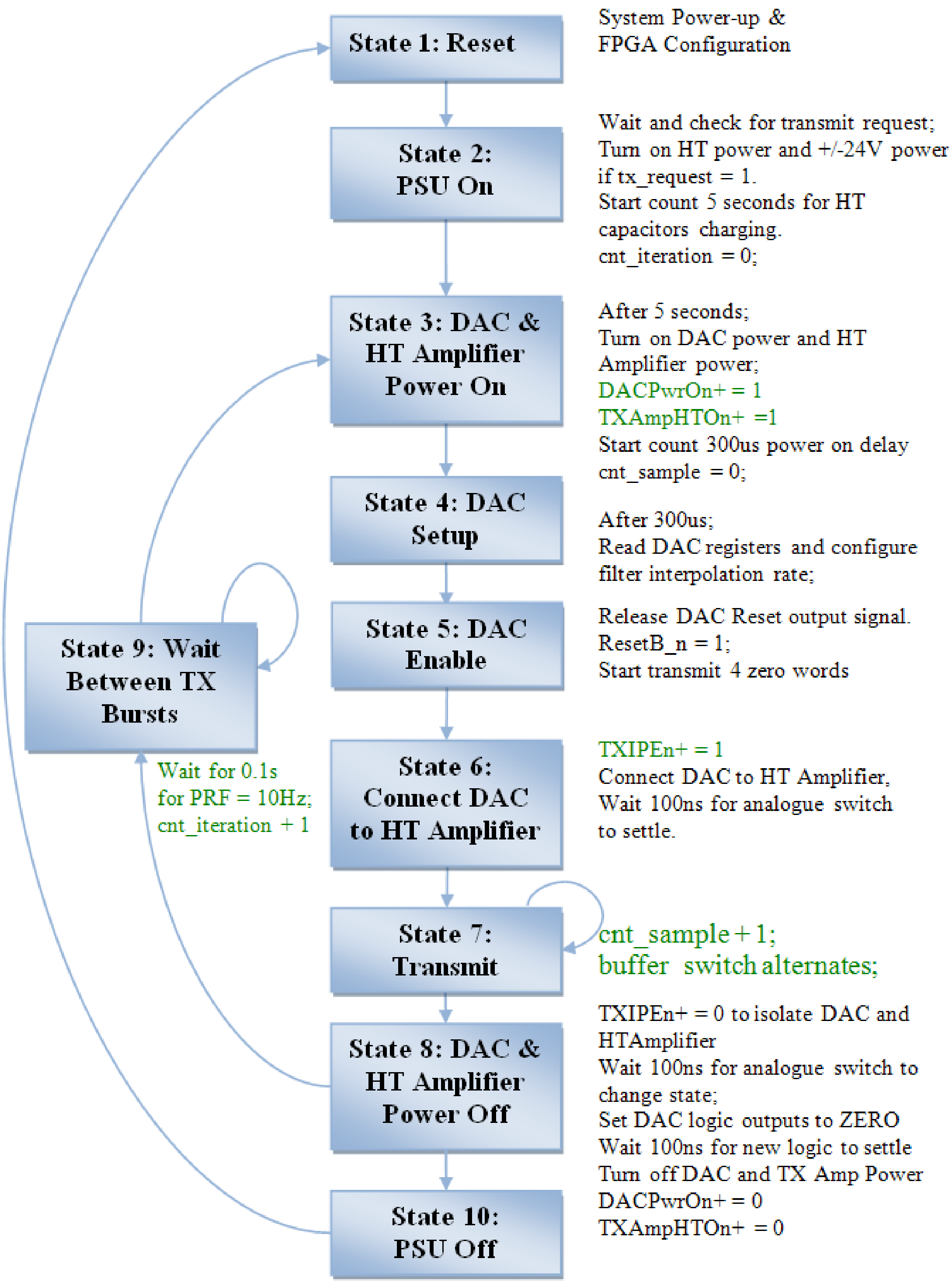
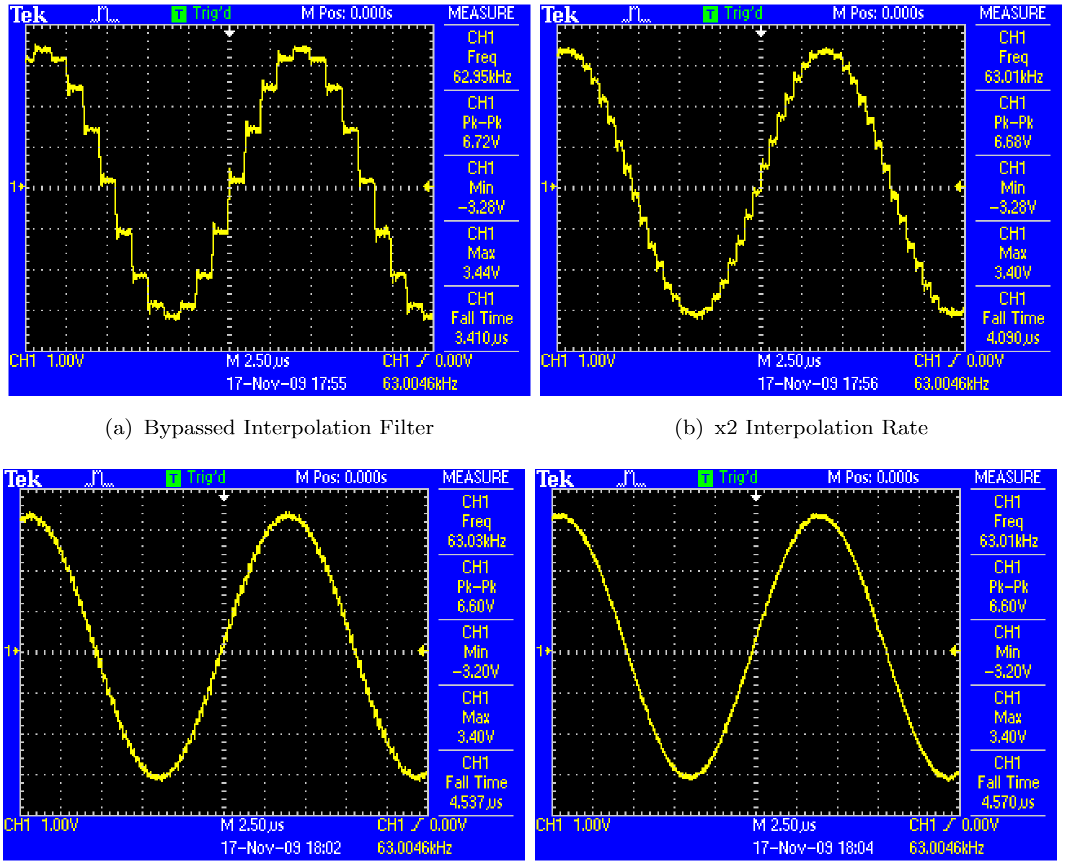


FIGURE 5.11: Teletest®MK4 Transmitter Finite State Machine Block Diagram

Table 5.3: Transmitter DAC FSM States Description

State	Descriptions
1	**Reset** When the FPGA is powered on, the FSM is in the Reset state. The reset state set the internal registers and output signals to default values. The transmitter DACs and high power amplifiers are powered off.
2	**PSU On** Wait and check for transmit request Turn on HT and ±24V power supplies when transmit request signal is detected.
3	**DAC & HT Amplifier Power On** Turn on DAC and TX amplifier power after waiting for 5 seconds
4	**DAC Setup** Read DAC registers and setup DAC after wait 300μs for TX amplifier and DAC to power up count_sample = 1024 interpolation filter rate = x16 count_iteration = 16
5	**DAC Enable** Bring DAC out from reset mode
6	**Connect DAC HT Amplifier** Connect DAC to HT amplifier Wait 100ns for analogue switch to settle
7	**Transmit** Connect DAC to HT amplifier Then start transmitting.
8	**DAC & HT Amplifier Power Off**
9	**Wait Between TX Bursts**
10	**PSU Off**

Each cycle sine wave is generated using 16 samples and the signal frequency is 62.5 kHz. Figure 5.12(a) shows the DAC output of the original waveform with the interpolation filter function turned off. There are 16 samples within one cycle and there are big voltage differences between adjacent samples. These differences can cause over current in the high power amplifier circuit when driving a high capacitive load. Figure 5.12(b) shows the DAC output with the interpolation filter set at x2 interpolation rate. There are 32 samples within one cycle and the voltage steps are reduced to half of those shown in Figure 5.12(a). Figure 5.12(c) shows that the DAC output is smoother at x4 interpolation rate than at x2 interpolation rate. Figure 5.12(d) shows further improvement of the DAC output which becomes very smooth at x16 interpolation rate. The x16 is used as the default configuration in the Teletest®MK4 design.



(a) Bypassed Interpolation Filter

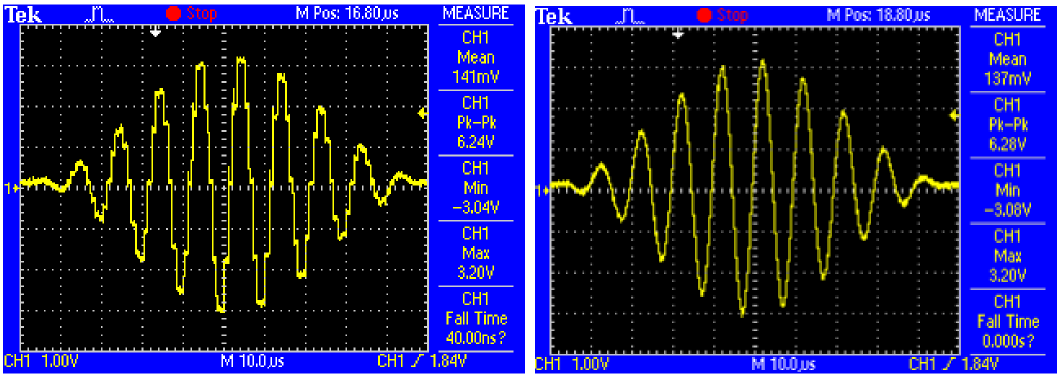
(b) x2 Interpolation Rate

(c) x4 Interpolation Rate

(d) x16 Interpolation Rate

FIGURE 5.12: Sine wave DAC Outputs

Figure 5.13(a) shows the non-interpolated DAC output of a Hann window applied ten cycles sine wave waveform. If this signal is used as input for the high power amplifier, overcurrent would occur and the output waveform of the high power amplifier would distort as described in section 5.1.2 and illustrated by Figure 5.3. Figure 5.13(b) shows the much smoother DAC output using the x16 interpolation filter integrated in the DAC8580. The resolution of the output waveform is significantly improved. The waveform stored within the FPGA BRAM eliminates the requirement for external transmitter FIFO devices. The transmit waveform is clocked out of the BRAM at 1 MSPS. The output of the DAC is equivalent to a signal generated using 16 MSPS sampling frequency. It needs to be noted that when a DAC device without the integrated interpolation function is used, the interpolation filters can also be implemented in the FPGA. In such manner, the FPGA BRAM can still be used as the transmit waveform FIFO to replace the external memory device in order to reduce the component count, PCB size and power consumption of the system.



(a) Bypassed Interpolation Filter

(b) X16 Interpolation Rate

Figure 5.13: Hann Window Waveform Interpolation

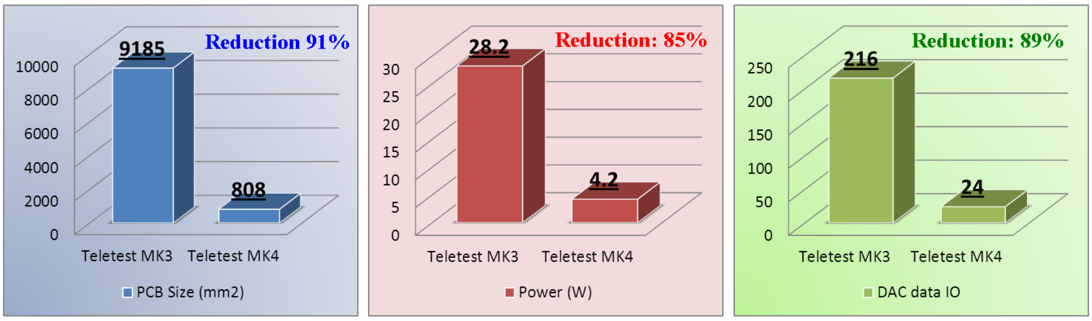
5.5 Summary

In summary, this chapter presented a novel FPGA design to control the multichannel transmitters of the Teletest®MK4 system. The FPGA on-chip BRAM resources are used to store arbitrary waveforms samples for the multichannel transmitters. The BRAMs are efficiently configured as dual port memory with parallel data input and serial data output to save the FPGA logic resource from implementing the parallel-to-serial converter for the serial data input DAC devices. The FPGA design also includes the interpolation DAC controller which resolved the transmitter over-current problems and reduced the requirement memory size for the multichannel transmitters. The FPGA design takes the advantages of the FPGA on-chip BRAM and the DAC8580 integrated interpolation filter and improves the performance of the system in several ways:

By using the FPGA internal BRAM and a small footprint serial data interface DAC to replace the 24 external FIFO devices and the large footprint parallel data interface DAC used in the MK3 system, firstly, the PCB size of the MK4 design was reduced by 91% from 9185 mm^2 in the MK3 system occupied by the 24-channel FIFO and DAC devices to 808 mm^2 in the MK4 system occupied only by the serial DAC devices. This also allows the complexity of the PCB routing to be significantly reduced. Secondly, the FPGA transmitter design allows the power consumption to be reduced by 70%. The power consumption of the 24-channel FIFO and DAC devices in the MK3 system is 28.2 W using 5 V power supply. The

24-channel DACs in the MK4 system only consume 4.2 W using ±5V power supply. Thirdly, the FPGA transmitter design reduces the overall components count by 24 by removing the FIFO devices and also reduces the FPGA IO requirement by using the serial data interface DAC. The DAC in the MK3 system has parallel data interface and each channel DAC needs 9 FPGA IO pins, and 216 FPGA IO pins would be required for 24 channels. The DAC in the MK4 system has serial interface and only requires 1 FPGA IO pin for the data interface per channel, 24 data pins for 24 channels. The FPGA IO requirement for the chip select (CS) and clock signals for each channel remains the same. This reduces the required number of the FPGA IO pins by 89%.

The improvements of the Teletest MK4 transmitter achieved by the FPGA design are illustrated by Figure 5.14.



(a) PCB Size (b) Power Consumption (c) ADC data IOBs

FIGURE 5.14: Improvements made by Teletest MK4 System Transmitter Design over Teletest MK3

Chapter 6

Multichannel Data Acquisition Design

6.1 Introduction

This chapter describes the FPGA design for controlling 24 channel data acquisition in the Teletest® MK4 system. The FPGA design improves the architecture of the Teletest® MK4 system with increased receiver channels, reduced PCB size, power consumption and FPGA IOB resource. Section 6.2 explains the hardware architecture of the receiver sub-system and the device selection for ADC and SRAM. Section 6.3 explains the main functions and data flow of the Teletest® MK4 data acquisition system. Section 6.4 explains the FPGA design for the ADC controller. Section 6.5 explains the FPGA design for the SRAM controller. Section 6.6 summarizes the multichannel data acquisition design, compares the MK4 system with the MK3 system and provides the evidence of system improvements.

The author was given the specification of the MK4 data acquisition sub-system to improve the hardware design. The MK3 system has 8-channel receivers and can store up to 64k samples for each channel. The MK4 system has 24-channel receivers and stores up to 128k samples for each channel. The total receiver number is increased 3-fold and the test distance is doubled due to the increased memory size per channel. Moreover, the amplifier gain of the 8-channel receivers in the MK3 system are controlled in common and all receiver channels have the same gain and filter settings. The 24-channel receivers in the MK4 system are required to be controlled individually. In order to achieve these system requirements, the author has searched the available electronic components at the time and selected the ADC and SRAM devices with small footprint, small power consumption for the MK4 system. In order to drive these devices for the 24 receiver channels in parallel, the author has designed and implemented the ADC controller and SRAM controller on the FPGA to control these devices synchronously. The FPGA used for MK4 design is a low cost device which is limited by the budget of the project. In order to control the 24 parallel ADCs, to drive the SRAM to implement the real-time accumulation function and to meet the timing requirement for 24-channel data acquisition, novel design has to be implemented on the FPGA in the Teletest® MK4 system to efficiently utilize the available FPGA resources.

Data acquisition (DAQ) is the process which involves the acquisition and digitising of the analogue waveforms; and then storing and processing the digital signals. The Teletest® MK4 system requires the acquired data to be captured and stored in the embedded system in real-time. The data acquisition must not be interrupted during the transmit and receive sequence unless acquisition is aborted due to hardware issues or data collection is halted by the operator. On-board fast speed memory is required to store the data. The data are then transmitted to a remote PC and analysed using Teletest® software. The FPGA needs to manage these 24 receiver channels in the MK4 system. The specification for the Teletest® MK4 system also requires the memory size for each receiver channel to be increased from 64 thousand samples to 128 thousand samples, allowing the system to collect more data for testing longer distance on the pipe. This creates more challenges for the hardware design with the requirement to reduce the size, weight and power consumption of the system. This chapter presents a new data acquisition system designed for the Teletest® MK4 with serial ADC and three parallel receiver memory banks to reduce the FPGA pin count, system size and power.

The Teletest® MK4 system needs to receive 24 channel signals in parallel up to 1 MSPS for all channels. Two real-time functions need to be implemented. Firstly, data from multiple acquisition iterations are accumulated in order to eliminate environmental noise. Secondly, ADC samples for all channels are compared to pre-set upper and lower threshold values to detect saturation of the ADC device. A comparison of various types of memory has shown that SRAM has fast access speed and low power consumption. Therefore SRAM is selected to store captured data during data acquisition. A three-bank SRAM memory structure is designed to meet the critical timing requirement for the 24-channel receiver system. Literature shows that the general asynchronous SRAM controller needs at least 3 clock cycles to complete one read or write access for a fast SRAM device [43, 42]. The number of clock cycles required for a write or read access indicates the number of the states in the FSM. One main FSM is designed to control the Teletest® MK4 data acquisition system including controlling the 24 ADC and 6 SRAM devices and the accumulation function, as well as the saturation detection function for 24 individual channels. In the accumulation function, one sample is read from one SRAM location, added with the sample captured by the ADC, and written back to the SRAM, all of which takes at least 7 clock cycles to complete using the general SRAM controller. This is not efficient enough to capture the 24-channel receiver data as explained in detail in this chapter. Therefore, a custom SRAM controller was designed and implemented on the FPGA, in which a read-add-write sequence can be completed within 2 clock cycles. This is more efficient than general controllers.

It needs to be noted that it is possible to meet the timing requirement using a general SRAM controller by increasing the clock frequency, with the cost of higher power consumption, longer FPGA synthesis and implementation time, and increased complexity in the PCB layout. The Teletest® MK4 system FPGA design can be divided into a number of functional blocks such as transmitter controller, receiver controller, MicroBlaze microcontroller, PLB bus and so on. These functional blocks can have different timing requirement depending on the functions. For example, the MicroBlaze and PLB bus needs to operate at 62.5 MHz, the transmitter serial data needs to be clocked at 16 MHz and the receiver serial data needs to be clocked at 20 MHz. When these functional blocks are implemented on the FPGA, the maximum operating frequency that can be achieved by each functional block is used to analyse the performance of the design to justify if the design can meet the timing requirement of a certain function. This maximum operating frequency also depends on the FPGA device selected for the design implementation. FPGA manufacturers have been focusing on new technology to achieve faster clock frequency for FPGA devices. The maximum clock frequency of the Spartan 3A DSP 1800A FPGA (speed grade -4) used for the Teletest® MK4 system is 300 MHz. The fastest clock speed of the complete FPGA design depends on the fastest speed achieved by each individual function. The timing closure is usually mostly affected by the function which has the fastest timing requirement in the system. However, higher clock frequency of an FPGA design leads to increased FPGA place and route complexity, which increases the synthesis and implementation time of the FPGA design tool. It also increases the power consumption of the FPGA. Furthermore, by lowering the maximum clock frequency of the data and address bus for the components, the PCB layout task can be simplified.

6.2 Hardware Architecture

The hardware architecture of the Teletest® MK4 receiver circuit is illustrated in Figure 6.1. The amplified and filtered analogue signal is converted to digital signal by the ADCs and stored in the on-board SRAM memory. The author was given the requisition to keep the receiver amplifiers and filters design the same as they are in the Teletest® MK3 system. The hardware architecture of the receiver amplifiers and filters for the Teletest® MK3 system was explained in section 2.4.2. In the MK4 system, the receiver amplifier and filter circuit is duplicated 24 times for the 24 parallel receiver channels. The amplifiers and filters have to be configured with the same value for all 8-channel receivers in the MK3 system, while in the MK4 system, each one of the total 24-channel receivers can be configured to have different gain of the amplifiers. Four CPLD devices are added to the system in order to provide the number of IO pins required for controlling the 24-channel various amplifiers and filters individually. This provides the flexibility required by scientists and researchers in order to employ the potential improvements in various LRUT applications.

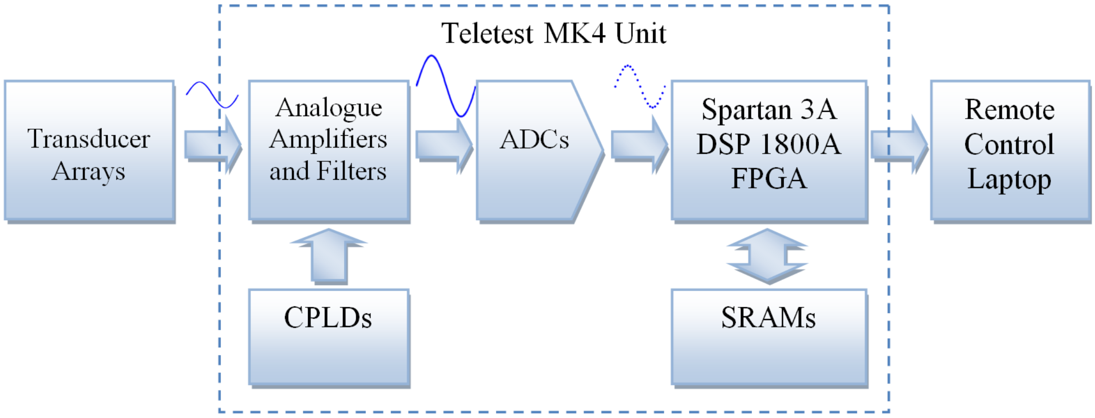


Figure 6.1: Teletest® MK4 Receiver Hardware Architecture

6.2.1 TI ADS7886 Analogue to Digital Converter Device

The ADS7886 is a single channel 12-bit resolution, 1MSPS, serial data output ADC device [60]. 24 ADS7886 devices are used to convert the analogue signals to digital signals for 24-channel receivers and these devices are controlled directly by the FPGA. This device is selected to reduce the footprint, power consumption and FPGA pin count in the Teletest® MK4 system design. In addition to meeting the requirement of the Teletest® MK4 system in terms of resolution and sampling frequency, the ADS7889 has three main benefits to the new design. First of all, the 6-pin SOT-23 package of the ADS7886 device has 3.05 mm by 3 mm footprint, which is 90% smaller than the 10.5 mm by 8.2 mm footprint of the ADS803E device used in the Teletest® MK3 system. It saves a total of 18.468 cm^2 PCB board area for the 24-channel receiver circuitry excluding PCB routing. Secondly, the ADS7886 only consumes 7.5 mW at 5V when operating at 1MSPS sampling rate, while the power consumption of the ADS803E parallel ADC used in the Teletest® MK3 is 115 mW at 5V operating at the same sampling rate. The ADS7886 reduces the power consumption per receiver channel by 93%. The reduction of the total power consumption for 24 channels is 2.58 W. Thirdly, the ADS7886 has serial interface and only needs one FPGA pin for 1-bit data per channel instead of 12 pins per channel. This reduces the FPGA pin count by a number of 264 for the 24-channel ADC interfaces and also reduces the complexity of PCB layout. Table 6.1 lists the footprint and power consumption for the ADC devices used for the Teletest® MK3 and MK4 system respectively.

TABLE 6.1: Teletest® MK3 and MK4 ADC Footprint and Power Consumption

	Part No.	**Footprint**	**Power Consumption**
Teletest® MK3	ADS803E	10.5 mm x 8.2 mm	115 mW @ 5 V
Teletest® MK4	ADS7886	3.05 mm x 3.0 mm	7.5 mW @ 5 V

The disadvantage of this device is that the power supply and the reference voltage shared the same pin on the chip; this potentially increases the noise level due to cross-talk between channels. The cross-talk signal might cause variance of the reference voltage and affect the ADC digitization precision. However, the electronic cross-talk is negligible compared to the mechanical cross talk between adjacent transducers in the sensor itself. The 12-bit data resolution provides -72dB SNR (signal to noise ratio) for 100 kHz signal at 5V. This is much lower than the ultrasound noise practically seen at -40dB owing to mode conversions and scattering of the UGWs and hence -72dB is adequate for the Teletest® MK4 application.

6.2.2 SRAM ISSI1024x16 Memory Device

The *ISSI-IS61WV102416* SRAM is a 1M x 16-bit high-speed asynchronous CMOS static RAM with 3.3V supply. The shortest read and write access time is 8 *ns* at 3.3V power supply. The advantage of the SRAM is small footprint, low power dissipation and fast write/read speed. The Teletest® MK4 system memory design is challenging at several aspects. The number of receiver channels is tripled and the memory length of each receiver channel is doubled in comparison with the Teletest® MK3 system. The Teletest® MK3 system uses 12 *IDT71V016SA12YI* SRAM devices to provide the memory required by 8 receiver channels; each channel has 24-bit wide by 64 k memory. As the MK3 system only has 8 analogue receiver channels, the system needs to be fired three times and the 8 receiver channels are switched among three transducer rings to collect signals for all 24 transmit channels. The ADC converts the analogue signal into 12-bit digital samples. The 24-bit wide sample arrangement in the Teletest® MK3 system allows the system to do up to 4096 accumulation with the extra 12 bits in the SRAM memory. The Teletest® MK4 SRAM design has increased the data sample bit-width from 24-bit to 32-bit for several reasons: The MicroBlaze processor has 32-bit wide data bus and can easily access the 32-bit data via the 32-bit PLB data bus. The accumulation number can be increased from 4096 to 1M by using 32-bit sample, with 12 bits ADC sample and 20 extra bits for the accumulation. The implementation of digital filters is proposed at the time of system design for the future LRUT applications. The 32-bit sample provides better precision for the multiplication result of the digital filters. When digital filters based on Multiply-accumulation (MAC) are implemented in the FPGA, the bit-width of the filtered signal depends on the resolution (bit-width) of the filter coefficients and the ADC output. 32-bit samples allow better resolution for the filter coefficient and the MAC results.

Table 6.2 lists the footprint and power consumption of the SRAMs used for the Teletest® MK3 and MK4 system respectively. The footprint of the IS61WV102416BLL SRAM (9 mm x 11 mm) is much smaller than the IDT71V016SA12YI SRAM (26.17 mm x 11.31 mm). The component count of the SRAM devices has been reduced from 12 in the MK3 system, to 6 in the MK4 system for 24 receiver channels. The total PCB board area for the 6 SRAMs in the Teletest® MK4 system is 594 mm^2, 83% less than the PCB board area for the 12 SRAMs in the MK3 system, that is 3551 mm^2. The power consumption of the SRAMs is reduced by 60% from 15 W in the MK3 system to 6 W in the MK4 system.

Table 6.3 compares the number of ADCs and SRAMs, the SRAM memory size, the total PCB area and the power consumption of the Teletest® MK3 and MK4 system respectively. In summary, the selected ADC and SRAM devices for the 24-channel receivers in the Teletest® MK4 has reduced the PCB board area by

Table 6.2: Teletest® MK3 and MK4 SRAM Footprint and Power Consumption

	Part No.	**Footprint**	**Power Consumption**
Teletest® MK3	IDT71V016SA12YI	26.1 mm x 11.3 mm	1.25 W @ 3.3 V
Teletest® MK4	IS61WV102416BLL	9 mm x 11 mm	1.0 W @ 3.3 V

81% on device footprints and the power consumption by 61% compared to the Teletest® MK3 8-channel receivers. This design concept can be used as a general reference to design system with tens or hundreds of receiver channels. For system required higher resolution ADC such as 24-bit, as long as the ADC device has serial data interface, only one FPGA pin is required for the data interface per channel. Otherwise, a parallel ADC with 24-bit resolution will require 24 FPGA pins for the parallel data interface. A significant number of FPGA IOB resources, which otherwise would be required to interface multi-channel parallel ADC devices, has been saved in the Teletest® MK4 data acquisition system design.

Table 6.3: Teletest® MK3 and MK4 Receivers Component Count, PCB Area and Power Consumption

	Teletest® MK3	**Teletest® MK4**
Number of ADCs	8	24
Number of SRAMs	12	6
SRAM Memory Size	1.5 MBytes	12 MBytes
Total PCB Area	4240.592 mm^2	813.6 mm^2
Total Power Consumption	15.92 W	6.18 W

6.3 Function Description and Data Flow

6.3.1 Function Description

The accumulation and averaging function is used to reduce environment noise, such as electromagnetic radiation from nearby high voltage equipment or cables, or mechanical interruption from the vibrations of high speed pumps.

The saturation detection function continuously checks the ADC converted samples for all 24 channels. If the amplitude of the ADC analogue input signal is above 5V or below 0V, the ADC converted value will saturate and the test result will be invalid. This function is designed to detect this situation and improve test efficiency.

The receiver clock divider is added to provide four selectable sampling frequencies: 1 MSPS, 500 kSPS, 250 kSPS and 125 kSPS. The high sampling frequency is used for achieving high resolution. The low sampling frequency is used for capturing data over long time period where the resolution can be reasonably traded off.

6.3.2 Data Flow

During acquisition, the FPGA reads the samples received in previous acquisitions from the SRAM sequentially, adds new samples received from the ADC at the current iteration and writes the new accumulation values back to the SRAM. To be more specific, when receiving the N^{th} sample, for each receiving channel, FPGA reads the previous stored value for this channel from the SRAM, add the new data received by the ADC for this channel, and write the addition result back to the SRAM.

The FPGA needs to control the accumulation-averaging to complete within 1μs for every received sample per channel because the ADC works at 1 MSPS. Every 8-channel ADCs share one SRAM memory bank. The memory address bus access time needs to be less than 1μs/8 = 125 ns and the data bus access time needs to be less than 62.5 ns. The SRAM device is capable of meeting this timing requirement according to the manufacturer's datasheet [61].

The data acquisition sub-system includes a Spartan 3A DSP FPGA, 24 ADS7886 ADCs and 6 ISSI SRAMs. The FPGA is used to control the ADC and SRAM devices. The ADCs and SRAMs are divided into three parallel groups with three FPGA interfaces. Each interface control 8 ADCs and 2 SRAMs devices. The ISSI SRAM is a 16-bit width by 1 M depth SRAM. The Teletest® MK4 system requires 128 M samples memory for each receiving channel. Each sample is 32-bit wide and therefore six 1M x 16-bit SRAMs are used to provide the required memory size. The challenge of the receiving sub-system design is to meet the time requirement. 24-channel 12-bit ADCs receive data at 1 MSPS sampling rate. Within 1 μs, the FPGA needs to read the precious samples of 24-channel from SRAMs and add the new received samples, then write the addition back to the SRAMs. The SRAM access time is 10 ns for both read and write. The ADCs and SRAMs are separated into 3 groups in order to increase the data access speed by using 3 separated data buses in parallel. FPGA parallelism is one of the keys to achieving increased operating speed over processor-based designs. It allows independent operations to be executed at the same time. Figure 6.2 illustrates the high-level block diagram and data flow for one group of 8-channel receivers for the Teletest® MK4 system. The data path (white arrow) represents the flow of data through the system. Once data acquisition is completed, the FPGA MicroBlaze processor transmits all received data to PC via Ethernet. The block diagram in Figure 6.2 also shows that ADC and SRAM signals are buffered by registers (FFS) in the FPGA to yield higher operating frequency for the data acquisition custom IP component.

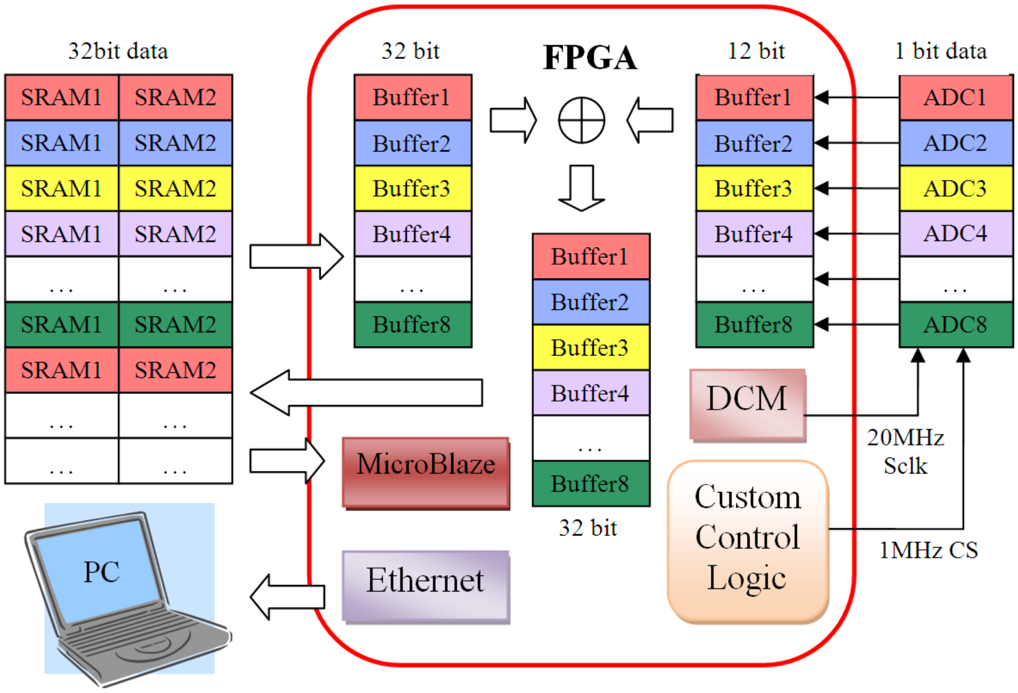


FIGURE 6.2: Teletest® MK4 Data Acquisition High Level Block Diagram

6.4 FPGA Design for ADC Controller

6.4.1 ADC Interface

Figure 6.3 shows the interface between one channel ADS7886 and the FPGA. In the Teletest® MK4 system, the digital data sent from ADC to FPGA is in the form of unsigned integer. The ADC7886 input voltage level is between 0V and +5V. The received data is firstly amplified by a bank of analogue amplifiers. The integer part can be represented by 3 bits and the fractional part can be represented by the rest 9 bits. This arrangement provides a fractional resolution of $\frac{1}{2^9}V$, which is approximately equivalent to 2 mV.

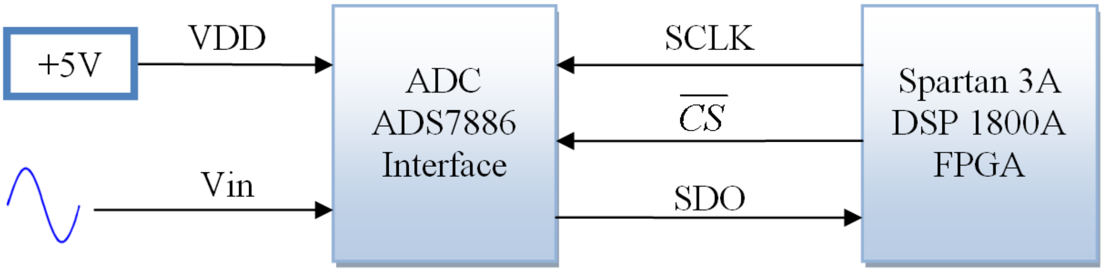


FIGURE 6.3: ADC ADS7886 Interface

6.4.2 ADC Controller Timing Diagram

Figure 6.4 illustrates the timing diagram of the ADS7886 controller. The serial clock (SCLK) of the ADS7886 device operates up to 20 MHz. The falling edge of the active low chip select signal $\overline{CS}$ initializes the conversion process and starts to sample one analogue signal. The ADS7886 output 16-bit data including 4-bit zeros and 12-bit non-zero data. One data word is composed of 4 leading zeros and 12 bits data in MSB first format. The falling edge of the $\overline{CS}$ clocks out the first zero and the following three zeros are clocked out on three falling edges of the serial clock signal (SCLK). The ADC interface design includes a 5-bit counter counting from 0 to 19. To acquire one ADC sample, the $\overline{CS}$ is activated at the first falling edged of SCLK, where the counter is 0 and 12-bit data is clocked in from the 4th falling edge to the 15th falling edge of SCLK. The $\overline{CS}$ is then inactivated at the 16th SCLK falling edge.

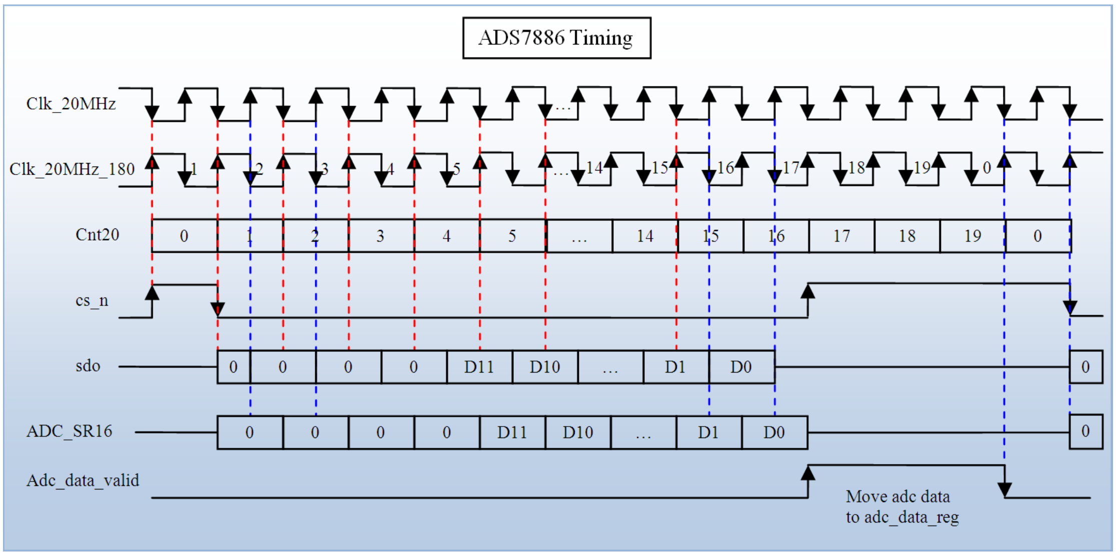


FIGURE 6.4: Teletest® MK4 System ADC Controller Timing Diagram

6.5 FPGA Design for SRAM Controller

The ISSI SRAM is an asynchronous device with no clock signal and level sensitive control signals for chip select (CE), write enable (WE) and output enable (OE). A memory controller is needed to connect the asynchronous SRAM to the synchronous FPGA system. The SRAM controller handles memory transactions request for reading or writing data from a specific address, generates the SRAM control signals, and adds registers to the control and data signals between the SRAM and the FPGA to generate a synchronous interface for the SRAM device. In the Teletest® MK4 FPGA design, the SRAM devices are accessed by two controllers. A custom SRAM controller is designed for the data acquisition sequence. Once the data acquisition is completed, the FSM which controls the transmit-receive sequence releases the control of the SRAMs. The MicroBlaze then takes over the control and sends the data stored in the SRAMs to the remote control PC via Ethernet. The XPS multi-channel external memory controller [62] is used to connect the SRAMs to the MicroBlaze via the PLB bus. This section describes the design of the custom SRAM controller.

6.5.1 SRAM Interfaces

Six SRAMs are used to provide a total of 32-bit wide by 3M deep memory for storing 24-channel receivers data. Each channel is allocated with 32-bit by 128k deep memory. Two SRAM devices are combined together to provide 1M 32-bit wide samples for one bank of 8-channel receivers. The FPGA interface for one SRAM bank combining two SRAM devices is illustrated in Figure 6.5.

Three SRAM banks are added to the design to increase the data throughput for the 24-channel data acquisition. The FPGA interfaces for three SRAM banks are illustrated in Figure 6.6. These three SRAM banks share 20-bit common address signals. Each individual SRAM bank has its dedicated 32-bit data and control signals. Each SRAM device has 16-bit wide data, equivalent to two bytes. The control signals for upper and lower bytes are active low and therefore are connected to ground for all six SRAM devices. The advantage of doing this is to save 12 FPGA I/O pins by trading off the byte accessible function which would be available otherwise.

Load Capacitance and Access Time Consideration

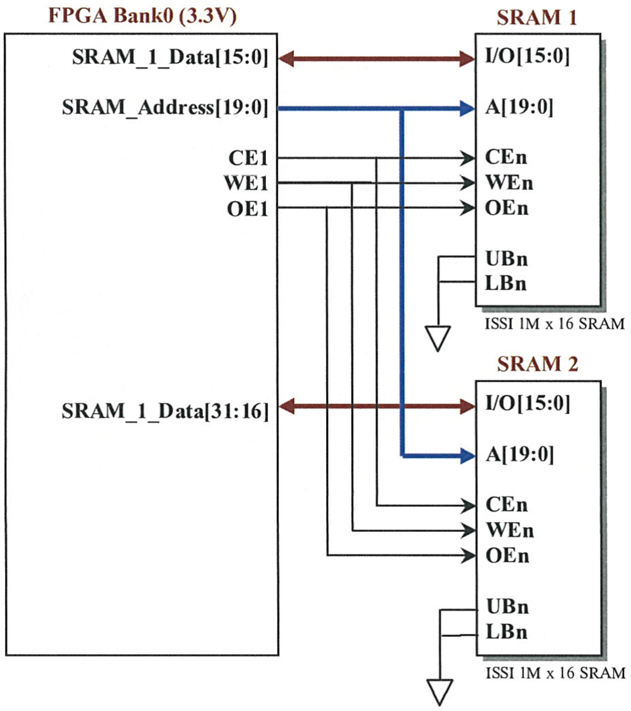


Figure 6.5: FPGA Interface for One SRAM Bank

The total capacitive load for an output pin of a device is defined as the sum of the input capacitance of all the other devices sharing the trace. The capacitance of the device driving the trace is not included. If the total load capacitance on a trace exceeds the output capacitance specification of the driving device then this total load capacitance is defined as 'excessive'. Typically a device input is specified with about 10 pF of capacitive load. For the $IS61WV102416$ SRAM, the input capacitance is 6 pF when Vin = 0V for address lines and the input/output capacitance is 8 pF when Vout = 0V for data lines.

To drive 6 SRAMs, the input capacitance to each FPGA address pin becomes 36 $pF(= 6\ pF \times 6\ lines)$, the input capacitance to each FPGA data pin becomes 24 $pF(= 8\ pF \times 3\ lines)$. Spartan-3A DSP FPGA supports Low-Voltage CMOS (LVCMOS) I/O standard on bank 1 and bank 3 with maximum 24 mA driving current strength. The LVCMOS standard is used for general-purpose applications at voltages form 1.2V to 3.3V. With input voltage swinging from 0V to 3.3V, 12 mA current drive, the transit time (dt) can be calculated using Equation(6.1). This is 9.9 ns for address lines and 6.6 ns for data lines.

$$I = C \times \frac{dV}{dt} \tag{6.1}$$

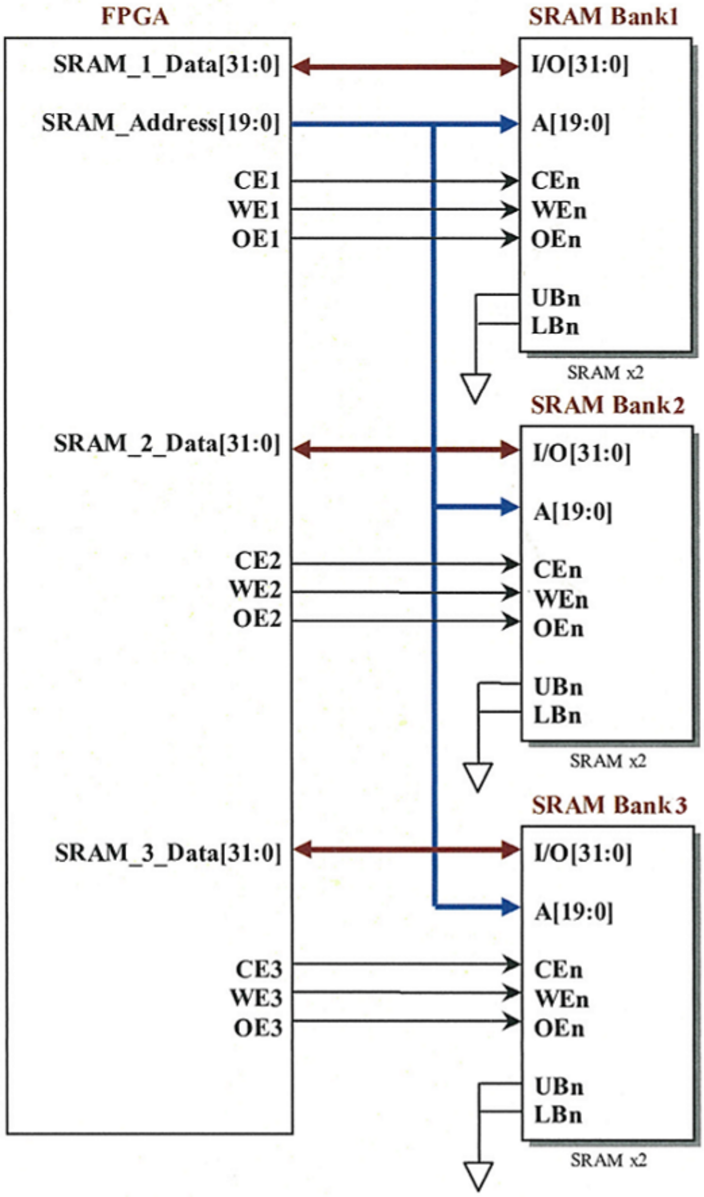


Figure 6.6: FPGA Interface for Three Parallel SRAM Banks

Speed and Power

The data acquisition function of the Teletest® MK4 system has a minimum speed requirement. There is usually no significant benefit in exceeding the target speed for most embedded system; however, failing in achieving the minimum speed requirement can degrade the system performance or even cause system failure. For many research subjects based on FPGA design, the maximum speed that can be achieved is the most important metric to measure the success of the design [36]. Nevertheless, in LRUT testing, the test speed is mainly affected by the time the UGWs transmitting over the tested pipelines. Exceeding the minimum system speed required by data acquisition for the UGWs does not result in a better end-user experience. Therefore it will not effectively improve the performance of the system.

The Teletest® MK4 system is a battery powered portable instrument. Therefore the embedded system is designed to increase the power efficiency. The FPGA dynamic power consumption is the power consumed by all non-zero frequency components as listed in Equation(6.2). High switching frequency in the FPGA design consumes more power; therefore exceeding the minimum speed required actually degrades the overall system performance.

$$Power_{dynamic} = kcV^2f \tag{6.2}$$

6.5.2 SRAM Controller Timing Diagram

Figure 6.7 illustrates the timing diagram of the SRAM controller. The falling edge of the 20 MHz receiver clock is used to clock a 21-bit wide address counter. This counter serves three purposes in the design. Firstly, the LSB (bit 0) of the counter is used as the write enable (we_n) signal of the SRAM. The data is written into the SRAM at the rising edge of the we_n signal which is marked with red up arrow in Figure 6.7. Secondly, the next 3 bits (bit 3 down to bit 1) count up 8 SRAM addresses for 8 channel receivers. Thirdly, the rest 17 bits count up the number of sample received with the maximum 128k samples per channel. These 8 samples are converted by 8 ADCs simultaneously and stored in the data buffers on the FPGA. Meanwhile the SRAM controller reads 8 samples which are captured from the previous transmit-receiver sequence from the SRAM, adds each sample with the new ADC samples in the data buffer, and writes the accumulation data back to the SRAM. The logic circuit for the 8-channel SRAM is replicated three times in the FPGA design for the 24-channel system. The multi-purpose counter is common for all 24 channels. It is used to simplify the SRAM controller design.

The period of the 20 MHz receiver clock is 50 ns. The SRAM address is assigned at the first falling edge of the clock. The maximum address access time of the SRAM is 20 ns[61]. After half of the clock period (25 ns), the SRAM data is ready to be read by the FPGA at the next rising edge of the clock. The addition of the ADC sample and the previous accumulation sample occurs at the next falling edge. And with the same address Cnt20(3:1), the data to be written to the SRAM (SRAM WR Data) is assigned at the next rising edge. Then after another

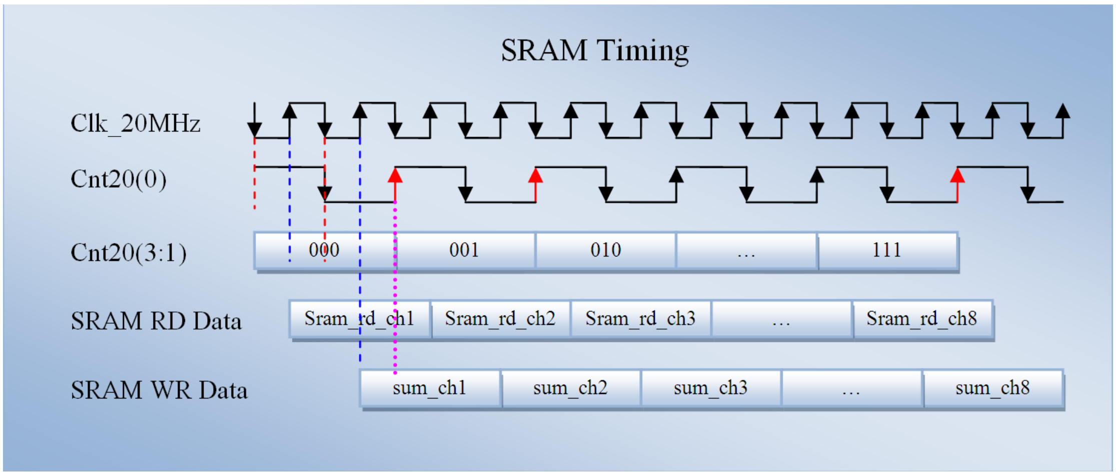
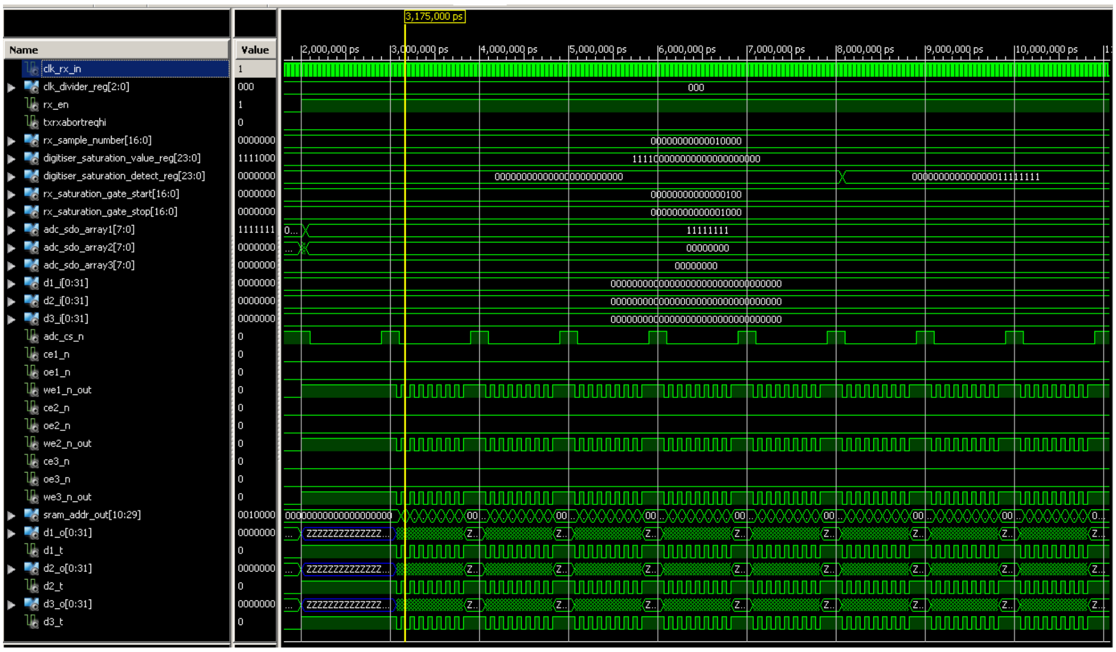


FIGURE 6.7: Teletest® MK4 System SRAM Controller Timing Diagram

25 ns, at the next falling clock edge, also the rising edge of the we_n signal, the sum value is written into the SRAM.

Figure 6.8 shows the functional behaviour simulation of the Teletest® MK4 FPGA design for the data acquisition. As mentioned early in Chapter 3, this is an important design step in verifying and debugging the design.



Name	Value
clk_rx_in	1
clk_divider_reg[2:0]	000
rx_en	1
txrxabortreqhi	0
rx_sample_number[16:0]	0000000
digitiser_saturation_value_reg[23:0]	1111000
digitiser_saturation_detect_reg[23:0]	0000000
rx_saturation_gate_start[16:0]	0000000
rx_saturation_gate_stop[16:0]	0000000
adc_sdo_array1[7:0]	1111111
adc_sdo_array2[7:0]	0000000
adc_sdo_array3[7:0]	0000000
d1_i[0:31]	0000000
d2_i[0:31]	0000000
d3_i[0:31]	0000000
adc_cs_n	0
ce1_n	0
oe1_n	0
we1_n_out	0
ce2_n	0
oe2_n	0
we2_n_out	0
ce3_n	0
oe3_n	0
we3_n_out	0
sram_addr_out[10:29]	0010000
d1_o[0:31]	0000000
d1_t	0
d2_o[0:31]	0000000
d2_t	0
d3_o[0:31]	0000000
d3_t	0

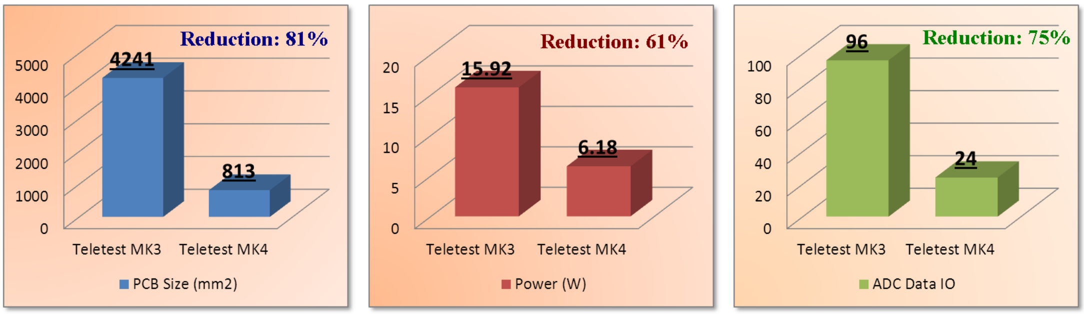
FIGURE 6.8: Teletest® MK4 Data Acquisition System FPGA Design Simulation

The memory controller together with accumulation, saturation detection and clock divider functions has been fully tested at the full SRAM memory range for the Teletest® MK4 system.

6.6 Summary

This chapter presented the FPGA design for controlling the multichannel receivers and improving the over-all performance for the Teletest® MK4 system. The FPGA controller design for the serial ADCs allows the system to capture the 24-channel received data in parallel. The serial-to-parallel converters are implemented suing FPGA logic resources. This design reduces the FPGA IOB resource usage. A novel SRAM controller was designed to enable the real-time accumulation function for all 24-channel receivers. The controller can read one sample from the SRAM, add it with the sample received by the ADC and write the data back to the SRAM all within two clock cycles. This reduces the FPGA clock frequency and hence reduces the FPGA power consumption.

The FPGA design has been successfully implemented to control the low power serial ADC and the low power fast SRAM devices for the Teletest® MK4 data acquisition. This allows the reduction of the PCB size, power consumption and FPGA IOB resource usage for the Teletest® MK4 system. These improvements are illustrated in Figure 6.9.



(a) PCB Size (b) Power Consumption (c) ADC data IOBs

FIGURE 6.9: Improvements made by Teletest® MK4 System Data Acquisition Design over Teletest® MK3

Chapter 7

FPGA Design Implementation and Optimization

7.1 Introduction

The various types of FPGA resources have been explained in section 3.4. The utilization of these resources in the Teletest® MK4 FPGA system design is discussed in this chapter. This chapter also presents and compares the implementation results of seven different optimization methods. Section 7.3 explains the implementation results for the Teletest® MK4 FPGA embedded system in detail. The ISE default optimization strategy is used for the synthesis and implementation. Section 7.4 compares the implementation results using various optimization strategies. Section 7.5 provides a summary for this chapter.

The Teletest® MK4 FPGA embedded system was built with the MicroBlaze based system, which is generated using the Base System Builder (BSB), part of the Xilinx EDK design tool. The custom IP cores were added without causing significant changes to the base system by using a common peripheral bus (PLB). As explained in Chapter 4, both the base system and custom IP cores in the MK4 system were designed using HDL and the bottom-up design approach, whereby a single top-level HDL file was created to instantiate all the low-level components used. Schematic capture software was used for the FPGA design of the previous MK3 system, in which the low-level components are connected by a large number of signals and nets in the top-level schematic file. This design approach is straightforward for a small system with a few simple FPGA functions such as in the Teletest® MK3 system because it can illustrate main components in the design and how they are connected. However, the schematic capture approach is an inefficient and error-prone task for implementing a complicated embedded system like the MK4 system, which has soft microprocessor, various peripheral controllers and custom functions to control 24-channel transmitters and receivers.

In the MK4 FPGA design, additional constraints were added for the FPGA input/output signals to interface the peripheral devices. These constraints contain the information which specifies the FPGA pin and IO standard to be used for each off-chip signal, as well as the timing constraints which indicates the synthesis tool to meet the timing requirements in the synthesis and implementation processes.

7.2 Teletest® MK4 System FPGA Design Resources Utilization

This section presents the FPGA implementation results for the FPGA resource utilization in the MK4 FPGA design. These resources include logic resource, clock resource, IOB resource, BRAM and DSP48s. The FPGA resource utilization summary is presented in Table 7.1.

Table 7.1: Teletest® MK4 System FPGA Resource Utilization Summary (ISE Implementation Result)

Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Flip Flops	10,897	33,280	32%
Number of 4 input LUTs	13,230	33,280	39%
Number of occupied Slices	11,189	16,640	67%
Total Number of 4 input LUTs	13,653	33,280	41%
Number of bonded IOBs	315	519	60%
Number of BUFGMUXs	12	24	50%
Number of DCMs	2	8	25%
Number of DSP48As	8	84	9%
Number of RAMB16BWERs	73	84	86%

7.2.1 Logic Resource Usage

The basic logic resource is the logic slice. Each slice includes FFs and LUTs. The system architecture and various FPGA functional blocks for the Teletest®

MK4 system have been discussed in Chapter 4. The FPGA logic resource usage for these various FPGA functional blocks implemented in the system is listed in Table 7.2. The functional blocks are sorted by the size of the slices used, starting from the MicroBlaze soft microcontroller, which is the functional block with the biggest number of slices. The MK4 system in its current form uses 67% of the slices in the Spartan 3A DSP 1800A FPGA.

Figure 7.1 illustrates the FPGA logic utilization distribution among the functional blocks, which uses most of the logic resource in the MK4 design. In the future if new function is required to be added in the system, the FPGA needs to be reconfigured and more logic resource may be required. The functional blocks using the most logic resource in the system can be looked at and optimized for area.

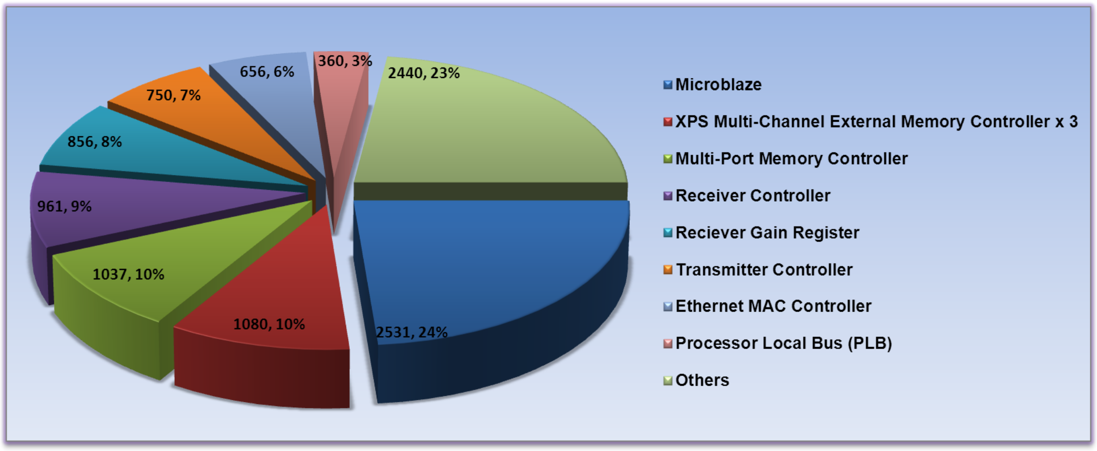


FIGURE 7.1: Teletest® MK4 FPGA Design Logic Resource Utilization Distribution for the Main Functional Blocks[1]

[1] Others includes all other peripheral controllers listed in Chapter 4. This does not include the unused logic resource of the FPGA

7.2.2 Clock Resource Usage

Section 3.4.4 has explained the FPGA clock resources and two DCMs are used to generate the required clocks for various functional blocks in the Teletest® MK4 system. The Spartan 3A DSP 1800A FPGA has totally 8 DCMs and 24 BUFGMUXs available [63]. Table 7.3 shows the clock report including the BUFGMUX location on the FPGA, the clock fanout, the net skew and the maximum delay of the clock net. It can be seen from the table that the 62.5 MHz clock net has the biggest fanout, the biggest net skew and the maximum delay.

Table 7.2: XPS Synthesis Logic Resource Usage Summary

Report	**Slices (16640)**	**Percent (%)**	**FFs (33280)**	**4-LUTs (33280)**
MicroBlaze	2531	15	2517	4019
XPS Multi-Channel External Memory Controller x 3	1080	6	1437	1059
Multi-Port Memory Controller	1037	6	1605	1006
Receiver Controller	961	6	846	1356
Receiver Gain Register	856	5	956	954
Transmitter Controller	750	5	772	1037
Ethernet MAC Controller	656	4	737	1101
Processor Local Bus (PLB)	360	2	166	597
XPS IIC Controller	322	2	379	479
XPS Timer	321	2	358	365
XPS SPI Controller 2	266	2	311	365
Top Level Sequencer	209	1	180	375
Receiver Amplifier Loader	199	1	88	355
XPS SPI Controller 1	197	1	242	285
XPS Interrupt Controller	153	1	219	183
XPS Watchdog Timer	122	1	166	134
XPS UART 1	114	1	152	142
XPS UART 3	114	1	152	142
XPS UART 2	108	1	144	132
MicroBlaze Debug Module (MDM)	88	1	119	147
XPS GPIO 1	86	1	115	99
XPS GPIO 2	71	0	111	64
Processor System Reset Module	41	0	67	52
Timer	10	0	11	24
Auxiliary Functions	7	0	7	13
DCM Module x 2	4	0	8	2
Data LMB Controller	3	0	2	6
Instruction LMB Controller	3	0	2	6
Data Local Memory Bus (LMB)	1	0	1	1
Instruction Local Memory Bus (LMB)	1	0	1	1
System	**10671**	**64**	**11871**	**14501**

This clock is used by the MicroBlaze microcontroller and the PLB bus. The peripherals and memories in the system are all connected to the MicroBlaze via the PLB bus. Therefore, this clock net is connected to a large number of FFs which are included in various peripheral controller blocks in the system. The large number of fanouts also results in big net skew and increased maximum delay of the clock net. This applies to other clock nets as well. The clock net with higher fanout generally has bigger net skew and larger maximum delay than the one with smaller fanout. All signals are 'completely routed' on the FPGA and all timing constraints have been met for the MK4 design. This shows that the timing requirements are met for the transmitter and receiver control sequences, the DAC, ADC and SRAM controller designs described in Chapter 5 and 6. The design goal has been particularly focused on using the lowest clock frequency to implement the required function on the FPGA to reduce the place and route effort and the FPGA power consumption. If a design fails to meet the timing requirement due to high fanout of the clock net, one suggested optimization without redesigning the function or rewriting the HDL codes is to duplicate the BUFGMUX used for the net to reduce the fanout of the clock net.

Table 7.3: Teletest® MK4 System FPGA Design Clock Net Skew and Delay Report for BUFGMUX

No.	**Clock Net**	**BUFGMUX Resource**	**Fanout**	**Net Skew (ns)**	**Max Delay (ns)**
1	clk_62_5000MHz	BUFGMUX_X1Y11	5658	0.301	1.766
2	clk_125_0000MHz90DCM0	BUFGMUX_X2Y10	252	0.237	1.765
3	clk_125_0000MHzDCM0	BUFGMUX_X1Y10	675	0.319	1.78
4	RX_Amp_Loader /sclk	BUFGMUX_X3Y5	61	0.208	1.009
5	clk100Hz	BUFGMUX_X0Y6	32	0.146	0.972
6	clk_10MHz	BUFGMUX_X2Y1	76	0.212	1.707
7	clk_1MHz	BUFGMUX_X3Y9	16	0.152	0.953
8	clk_20MHz	BUFGMUX_X1Y0	467	0.247	1.712
9	mdm_0/Dbg_Clk_1	BUFGMUX_X0Y2	109	0.205	0.995
10	clk_16MHz	BUFGMUX_X2Y0	82	0.169	1.647
11	clk_20MHz_270	BUFGMUX_X1Y1	1	0	1.615
12	clk_20MHz_int	BUFGMUX_X2Y11	4	0.007	1.612

7.2.3 BRAM Resource Usage

Table 7.4 lists the number of BRAM resources utilized by various functions in the Teletest® MK4 system FPGA design. The system uses 73 BRAMs out of the total of 84 available on the Spartan 3A DSP 1800A FPGA. 48 BRAMs are used for the 24-channel transmitter waveform FIFOs. Each BRAM is 18 kb (18,432 bits) in size, which includes 16 k data bits and 2 k parity bits. The BRAM is configured as 1 k x 16-bit without using the parity [63]. The required FIFO size for each transmitter channel is 2 k x 16-bit, which takes up 2 BRAM and hence 48 BRAMs are used. 4 BRAMs are used for the transmitter and receiver buffers of the Ethernet MAC IP core implemented on the FPGA. The Ethernet MAC Lite IP core uses 2 kByptes dual port memory for both transmitter (TX) and receiver (RX) to hold data for one complete frame as well as the interface control registers. The Teletest® MK4 system design also has an optional extra 2 kBytes dual port memory Ping-Pong double buffer for both TX and RX [53]. The BRAM is configured as 2 k x 8-bit without using the parity bits [63] and 4 BRAMs are required to provide the total 8 kBytes buffer size. 5 BRAMs are used as 32-bit read and write data path FIFOs in the MPMC for the DDR2 SDRAM memory [64]. 8 BRAMs are configured as 16 kBytes (4k x 32 bit) local memory for the MicroBlaze microprocessor on the LMB [65]. 8 BRAMs are used as instruction and data cache memory in the MicroBlaze IP core [66].By using the FPGA on-chip BRAM resources in the Teletest® MK4 system design to replace the external FIFO devices in the MK3 system, the PCB area, power consumption and FPGA pin count requirement have been reduced. This was explained in Chapter 5. The BRAMs are fast and flexible memory resources which provide flexibility and reliability to the MK4 system.

Table 7.4: Teletest® MK4 System FPGA Design BRAMs Resource Usage Summary

Functional Block	**Number of BRAMs (Available:84)**	**Percent (%)**
MicroBlaze	8	9%
Local Memory Bus BRAM	8	9%
Multi-Port Memory Controller	5	5%
Transmitter Controller	48	57%
Ethernet MAC	4	4%
Total Usage	73	86%

7.2.4 IOB Resource Usage

There are totally 519 IOB available on the Spartan 3A DSP 1800A FPGA. 316 are used in FPGA design for the Teletest® MK4 system. The FPGA IOB resources were previously explained in section 3.2. Table 7.5 lists the hardware devices as FPGA peripherals and their controller interfaces to the FPGA, as well as their I/O standards, voltage level and the number of required FPGA IO pins.

Table 7.5: Teletest® MK4 Interface IOB Resource Utilization, IO Standard and Voltage Level

FPGA External Devices	**Interfaces**	**I/O Standard**	**Voltage**	**Pins**
SRAM x 3	XPS MCH EMC	LVCMOS33	3.3V	125
DDR2 SDRAM	MPMC	SSTL18	1.8V	71
DAC x 24	Custom IP	LVCMOS33	3.3V	30
ADC x 24	Custom IP	LVCMOS33	3.3V	26
Ethernet PHY	Ethernet MAC	LVCMOS18	1.8V	18
SPI Flash Memory	SPI_0	LVTTL	3.3V	4
Temp and Volt Sensors	SPI_1	LVTTL	3.3V	5
CPLD Rx Gain Control	Custom IP	LVCMOS33	3.3V	5
Non-volatile RAM	IIC	LVTTL	3.3V	2
Front Panel Controller	UART_0 / Reset	LVTTL	3.3V	3
PC System Debug	UART_1	LVTTL	3.3V	2
GPS	UART_2	LVTTL	3.3V	2
Normal and Monitor Mode Switches	Custom IP	LVCMOS33	3.3V	2
Receiver Turn On at start or end of Transmit Switches	Custom IP	LVCMOS33	3.3V	4
Power Control	Custom IP	LVCMOS33	3.3V	7
System Clock Input	125 MHz Oscillator	LVCMOS33	3.3V	1
System Reset Input	System Reset	LVTTL	3.3V	1
Watchdog Timer Reset	Output	LVCMOS33	3.3V	1

Figure 7.2 shows the FPGA implementation results for the system on the device floorplan. The floorplan[67] was generated using Planahead [68]. The FPGA design flow including mapping, placing and routing were described in section 3.5. The full design is mapped onto the on-chip resource of the Spartan 3A DSP 1800A FPGA. The place and route tool then selects the appropriate interconnection (net) for the resource based on the timing constraints of various functions in the design.

The IOBs of Spartan 3 FPGA are divided into four banks which are highlighted at the four edges of the devices in the floorplan. Bank 0 is on the top edge of the device showing in purple. Bank 1 is on the right edge showing in green. Bank 2 is at the bottom showing in cyan. Bank 3 is at the left showing in pink. The FPGA resources such as DCM, BRAMs and DSP48 are also marked out in the floorplan. IO connectivity is displayed as green IO lines, showing the relative distance and connectivity of the nets.

7.3 FPGA Power Consumption

The SRAM-based FPGA devices are volatile and have to be powered up constantly in order to not lose the interconnect configuration. The FPGA power consumption includes static and dynamic power. Dynamic power can be reduced by HDL coding techniques. FPGA dynamic power consumption is dominated by the switching frequency of FFs. The functions running at various clock frequencies are assigned to different clock domains. It is good practice to run each function only at its required frequency rate and avoid using unnecessary high clock frequency for low frequency function to reduce FPGA power consumption, as Teletest® is a battery powered portable equipment. The reduction of the power consumption can increase the battery life and reduce the necessity for frequently recharging the system. The FPGA power consumption report for the Teletest® MK4 system is presented in Table 7.6.

TABLE 7.6: Teletest® MK4 System FPGA Power Consumption

Name	**Maximum Power Consumption**
Clocks	0.02494(W)
Logic	0.06718(W)
Signals	0.10200(W)
IOs	0.66932(W)
BRAMs	0.06033(W)
DCMs	0.04254(W)
DSPs	0.00006(W)
Total Quiescent Power	1.02888(W)
Total Dynamic Power	0.48301(W)
Total Power	1.51189(W)
Junction Temperature	49.0(degree C)

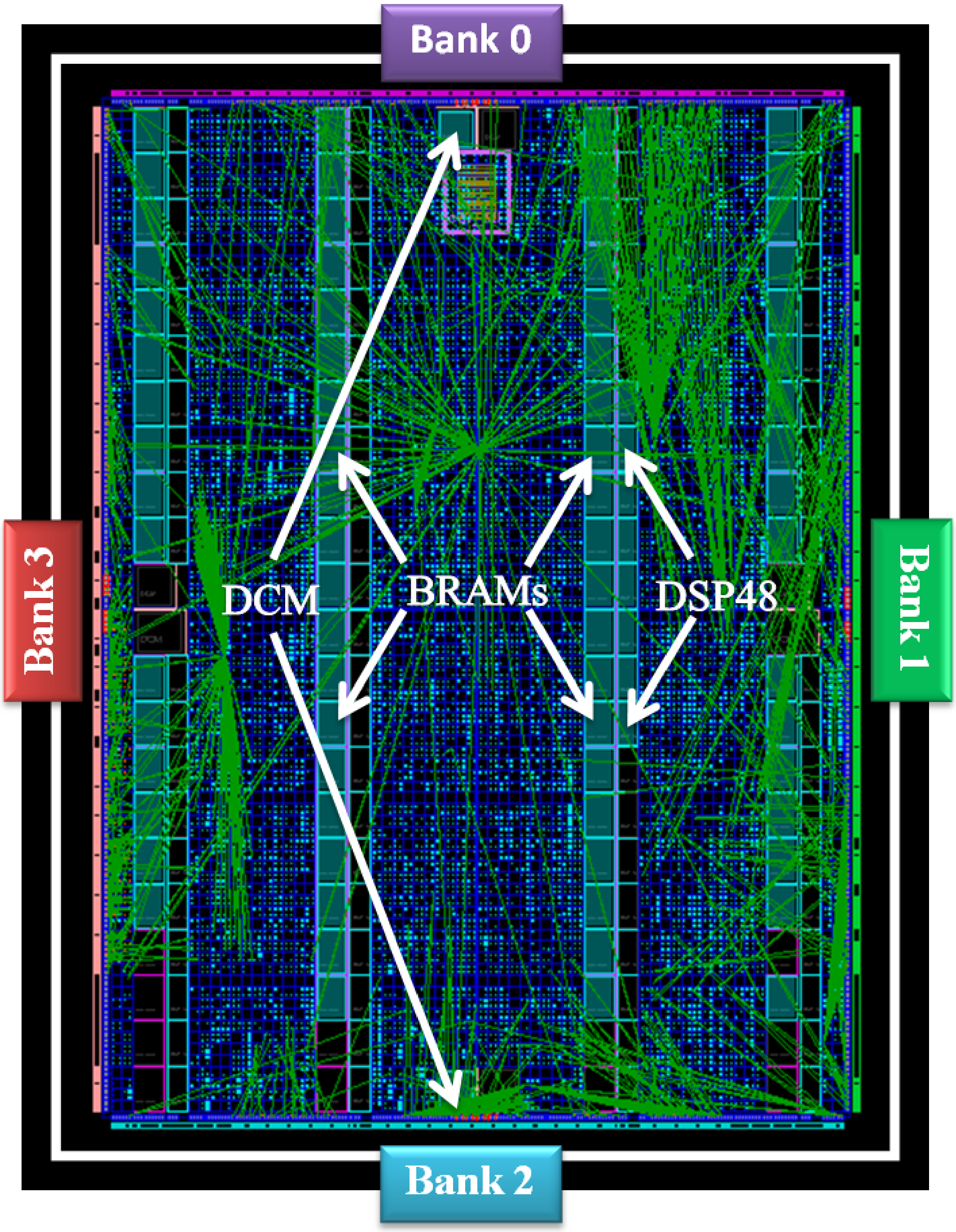


FIGURE 7.2: Teletest® MK4 FPGA Design Implementation Results showing the Placement of the FPGA On-chip Resource Usage and the IO Connectivity

7.4 Implementation Optimization Strategies

The design implementation results can be affected by various optimization strategies. The FPGA design tool ISE allows various options to be set for synthesis and implementation. The performance results of the Teletest® MK4 FPGA design using various optimization strategies are listed in Table 7.7. These results show the utilization of the FPGA logic resources including the number of FFs, LUTs used and the occupied slices, as well as the minimum period of the system, which is actually that of the slowest clock (10 MHz) used in the MK4 system.

Table 7.7: Teletest® MK4 FPGA Design Performance Results Chart using ISE Optimization Strategies

	Number of Slice Flip Flops[1]	**Number of 4 input LUTs**	**Number of occupied Slices**	**Total Number of 4 input LUTs**[2]	**Minimum Period (ns)**[3]
1	10,897 (32%)	13,230 (39%)	11,189 (67%)	13,653 (41%)	35.448ns
2	10,723 (32%)	13,226 (39%)	11,074 (66%)	13,642 (40%)	30.960ns
3	10,723 (32%)	13,226 (39%)	11,210 (67%)	13,642 (40%)	33.032ns
4	10,723 (32%)	13,226 (39%)	11,074 (66%)	13,642 (40%)	30.960ns
5	10,723 (32%)	13,226 (39%)	11,074 (66%)	13,642 (40%)	30.960ns
6	10,723 (32%)	13,226 (39%)	11,131 (66%)	13,645 (41%)	33.310ns
7	10,723 (32%)	13,189 (39%)	11,131 (66%)	13,612 (40%)	27.888ns

The main system clock for the Teletest® MK4 system is the 125 MHz clock. All the other clocks are derived from this main system clock using two DCMs on the FPGA. The minimum periods of these derived clocks achieved by different ISE optimization strategies are listed in Table 7.8.

The strategies of seven ISE implementation iterations are explained as follow:

1. ISE Default: This is the ISE 11 default strategy. The optimization strategy (cover mode) is set to 'area' with the highest priority of reducing the number of LUTs. All other optimization options are turned off. All signals are

[1] The percentages showing in brackets are the utilization percentages of the available resources on the Spartan 3A DSP 1800A FPGA device.

[2] The total number of 4 input LUTs equals the number of 4 input LUTs used for logic utilization, including logic, distributed RAM and shift register, plus the number of 4 input LUTs used as route-thru.

[3] The minimum period reported by the timing report is for the slowest clock in the MK4 system, which is the 10 MHz clock for the top level FSM.

Table 7.8: Teletest® MK4 FPGA Design Clock Minimum Periods using ISE Optimization Strategies

Functions	**System Clock**	**MicroBlaze**	**RX**[1]	**TX**	**Top FSM**
Clocks	125 MHz	62.5 MHz	20 MHz	16 MHz	10 MHz
Periods	8.000 ns	16.000 ns	50 ns	62.500 ns	100.00 ns
1	4.800 ns	15.787 ns	17.222 ns	20.385 ns	35.448 ns
2	4.800 ns	15.804 ns	18.108 ns	21.395 ns	30.960 ns
3	4.800 ns	15.833 ns	17.276 ns	19.945 ns	33.032 ns
4	4.800 ns	15.804 ns	18.108 ns	21.395 ns	30.960 ns
5	4.800 ns	15.804 ns	18.108 ns	21.395 ns	30.960 ns
6	4.800 ns	15.788 ns	24.052 ns	23.622 ns	33.310 ns
7	4.800 ns	15.895 ns	16.652 ns	24.695 ns	27.888 ns

completely routed and all timing requirement are met in the Teletest® MK4 FPGA design using the default strategy. This implementation result is analysed and discussed in detail in section 7.2.

2. Map Timing: The cover mode for is set to 'area'. This strategy packs internal FFs/latches into IOBs. FPGA contains registers in its IOBs. These registers can greatly reduce the clock-to-input (OFFSET IN) and the clock-to-output (OFFSET OUT) delay and therefore reduce the maximum delay and increase the speed of the design by 13%. This strategy also uses timing-driven MAP which gives priority to timing critical paths specified in the UCF during packing in the MAP process. The MAP effort level is set to high and extra effort level is set to normal. The PAR overall effort level is set to high and the extra effort level is set to normal. The implementation result shows that the minimum period is reduced from 35.45 ns in the ISE default iteration to 30.96 ns. However, the minimum periods of all the derived clocks are slightly increased in this iteration compared to the ISE default as the synthesis tool only concentrated on the paths with longest time delay, which are related to the lowest clock frequency (10 MHz). The synthesis tool also uses high effort level and normal extra effort level for MAP to optimize the design for area and has reduces the number of FFs, LUTs and total slice by 174, 11 and 115 respectively.

[1] This is the output of the second DCM which is used as the input clock of the receiver custom IP written in VHDL.

3. MapGlobalOptParHigh: In this strategy, the global optimization property is set for speed. The effort level for both MAP and PAR are set to high with none extra effort level. The implementation results show that the minimum period for the 10 MHz is better than the ISE default result but worse than the second iteration which used extra effort level in the MAP and PAR. The number of the occupied slices is the biggest among the seven iterations.

4. MapLogicOptParHighExtra: This strategy packs internal FFs/latches into IOBs. The global optimization of MAP is set to speed. Effort levels are set to high and extra effort levels are set to normal for both MAP and PAR. Furthermore, the combinatorial logic optimization property is turned on to perform physical synthesis combinatorial logic optimizations during timing-driven packing with the aim to further reduce the resource usage. The register duplication is turned on to replicate the register in order to help control fanout.

5. MapGlobalOptLogicOptRetimingDupParHigh: This strategy has all the optimization properties set the same as the forth implementation, plus the retiming property is turned on, which moves the registers forward or backward in the logic to balance the delay in a timing path and hence increase the speed. Again, this implementation shares the same result as the second and the forth iterations. No obvious improvement was observed by using these optimization options for the Teletest® MK4 FPGA design.

6. MapTimingIgnoreKeepHierarchy: This strategy carries out the same optimization as the second iteration, the MAP Timing strategy, plus the Ignore User Timing Constraints property is turned on. This property allows the MAP process to perform optimization across hierarchical boundaries. When the hierarchy of the design is preserved for the purpose of simulation or partitions, better timing performance sometimes can be achieved by ignoring the hierarchy. However, the Teletest® MK4 FPGA design is a multiple time domains hierarchical design. The implementation result indicates that the minimum period for all the derived clocks used for various FPGA functional blocks are bigger than those of any other iteration. Therefore, it is not suitable to allow the synthesis tool to perform optimization across the hierarchical boundaries of the MK4 design.

7. MapCoverBalanced: The MAP optimization is balanced between area and speed. The PAR effort is high. The implementation result of this iteration

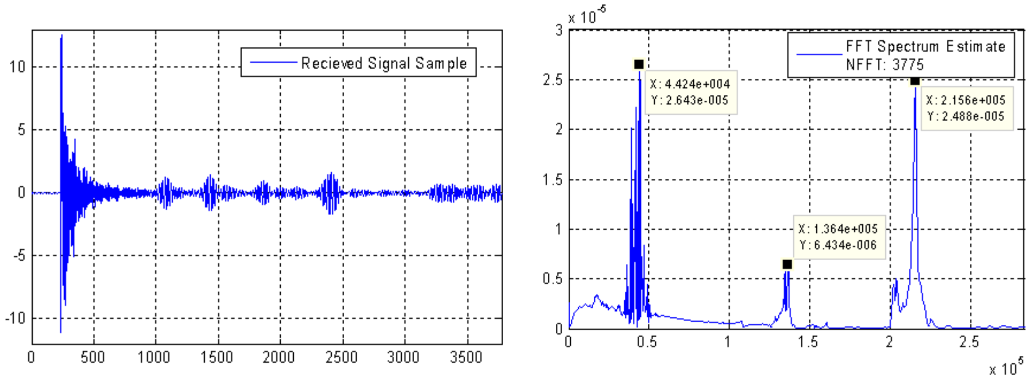
has the best minimum period 27.888 *ns* and uses the smallest number of LUTs.

In summary, the best minimum period is achieved by the 7th iteration. The minimum number of occupied slices is achieved equally by three iterations: 2nd, 4th and 5th. The FPGA design for the Teletest® MK4 system utilizes the resources to best fit the application into the Spartan 3A DSP 1800A FPGA.

7.5 Other FPGA Related Work - Digital Filters

The finite impulse response (FIR) filter was implemented in FPGA to process the post received data of the Teletest® MK4 system. The FIR filter has a linear phase characteristic with frequency, which is a desirable property for the LRUT application. The multiply-accumulator (MAC) architecture was used in the FIR filter design and was implemented using the dedicated FPGA DSP48 resource.

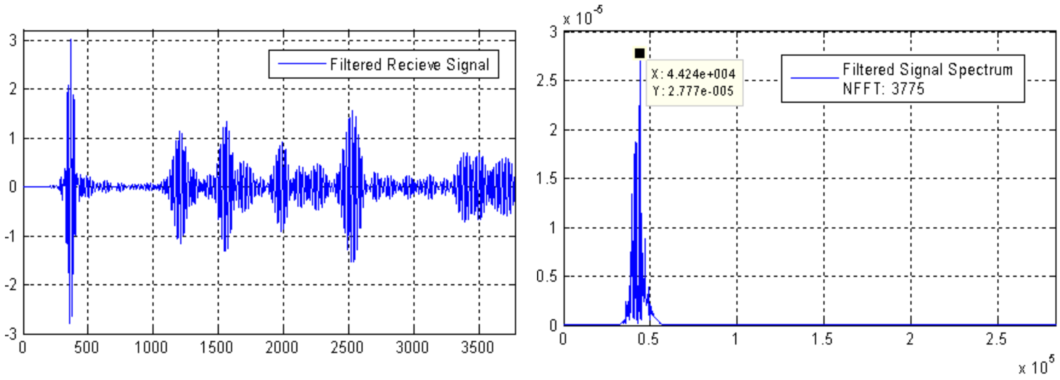
A received LRUT signal sample was used to estimate the filter performance. The signal and its FFT spectrum are plotted in Figure 7.3. The signal frequency was 44 kHz. The system is switched from transmitting to receiving after the transmitting. The spectrum shows the 3rd and 5th harmonics signals at 136 kHz and 215 kHz respectively. Hence a band pass FIR filter was designed for this signal. Figure 7.4 shows the filtered signal and its spectrum. It can be seen that the 3rd and 5th harmonics at high frequencies are suppressed by the designed filter.



(a) LRUT Received Signal Sample

(b) FFT Spectrum

FIGURE 7.3: LRUT Received Signal Sample and FFT Spectrum



(a) Filtered Signal

(b) Filtered Signal FFT Spectrum

FIGURE 7.4: LRUT Received Signal with FIR filter and its FFT Spectrum

7.6 Summary

This chapter presented the FPGA implementation results for the Teletest® MK4 system design. It also discussed the utilization of various FPGA resources: logic, clock, IOB and BRAM and the percentage of these resources used by the FPGA functions. The design was fully implemented on the FPGA. All signals are completely routed and the timing constraints are met for the design. Moreover, experiments have been carried out to optimize the design by using various synthesis and implementation optimizing options in the ISE tool in order to further improve the implementation results by reducing the resource usage and increasing the maximum clock frequency. The optimization results have shown improvement over different optimization options.

The contribution of this work is to demonstrate the implementation results of the working Teletest® MK4 prototype, which provides evidences that the low cost and low power Spartan 3A 1800A FPGA can be successfully used for the overall control of a complicated embedded system composed of MicroBlaze, various memory and peripheral controller, and multichannel transmitter and receivers. The system has been successfully implemented and tested with the functional requirements to implement the LRUT applications explained in Chapter 2. The author has written the Teletest® MK4 FOCUS+ System FPGA Design Specification which is provided in Appendix B as reference.

Chapter 8

Conclusions and Recommendations for Future Work

8.1 Conclusions

The specification had been drawn up for the Teletest® MK4 system requiring certain features to be added to the system which could not be implemented by the previous MK3 system. Further research work was required to establish how the additional requirements could be achieved while also achieving improvements in system size, weight and power consumption. Therefore, research has been carried out to implement these features on a new FPGA design which then enabled a number of significant improvements to be made for the working Teletest® MK4 system prototype.

This project resulted in the following contributions to knowledge:

FPGA-based embedded system design concept using VHDL, permitting reconfigurable overall system control functions

An FPGA-based embedded system has been designed using a state-of-the-art FPGA device. The design concept uses reconfigurable FPGA for overall system functions control. By using the reconfigurability of the FPGA and the flexible MicroBlaze soft IP core, the Teletest® MK4 system can be updated and new functions can be added for implementing future LRUT applications. By using the parallelism function of the FPGA, the multichannel transmitters and receivers can be controlled in parallel and the data throughput required by the multichannel system can be achieved. The FPGA-based embedded system design concept provides the reconfigurability for the overall system control and the flexibility required for extending system functions.

Multichannel arbitrary waveform generator using interpolation DAC and efficient use of FPGA BRAM and control of multi-channel transmitters

The FPGA controller design for the multichannel arbitrary waveform generator efficiently uses the FPGA on-chip BRAM resources and the interpolation filter of the DAC to reduce the sampling frequency and the complexity of the system, which consequently reduced the size, components count and power consumption of the system. The FPGA controller has been designed for the multichannel arbitrary waveform generator to control the multichannel interpolation DACs and the FPGA on-chip BRAMs, The efficient use of FPGA BRAM resource enable the reduced usage of FPGA logic resource. The FPGA controller has been successfully implemented and allows the improvements to be made for the Teletest® MK4 system against the MK3 system in several respects: The component count for the multichannel transmitter is reduced by 24 for the transmitter FIFO devices, all these devices now being contained within the FPGA itself; The PCB area size, the power consumption and the required FPGA IOB number of the waveform generator including 24 channel DACs and FIFOs are reduced by 81%, 70% and 89% respectively.

Control of multichannel receivers for data acquisition, including control of ADCs, SRAM and real-time accumulation of data

The FPGA controller design for the multichannel receivers allows the system to receive 24-channel signals in parallel and accumulate the received data in real-time. The FPGA design also controls the multichannel receivers for data acquisition, including control of ADCs, SRAM and real-time accumulation of data. A novel design for the SRAM controller allows the write-add-read of the SRAM samples for the accumulation function to be completed in only 2 clock cycles. This reduces the required FPGA operating speed from above 50MHz to 20MHz, and hence reduces the FPGA dynamic power consumption for the multichannel receiver controller by 60% referring to equation 6.2.

The new system designed was evaluated to assess various parameters such as speed, power consumption, size and weight. It has to be noted that because the Teletest® MK4 is a commercial product, many business aspects had to be taken into consideration in the design, including product delivery schedule, market demand and production cost. The cost of the Teletest® MK4 system was constrained by the project budget, which limits the options of using high profile FPGA devices. The design goal was to minimize the design cost while maximizing the system performance. Therefore, a low cost, low power FPGA was used and the extra effort had been made to efficiently use the available FPGA resource in order to implement the required functions and improve the performance of the system. Another design goal was to assemble systems that can be reused through a variety of designs and easily expanded to meet each new design need. The FPGA provides the flexibility of reconfiguration and the multichannel concurrency to meet the speed requirement. The FPGA-based embedded system was designed to enhance the performance of the Teletest® system hardware and to improve the capability of the system for implementing more complicated LRUT techniques such as the full matrix capture.

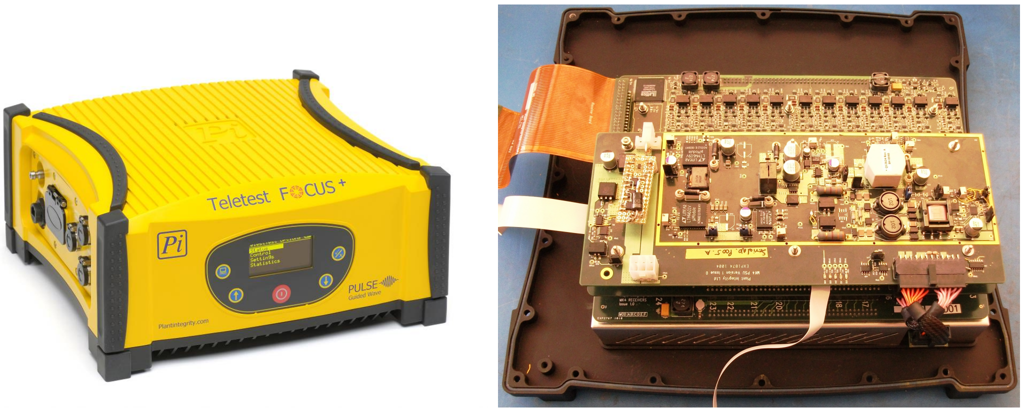
This thesis covers the details regarding the FPGA used as the central computational device of the embedded system for the Teletest® MK4 system design, the evaluation and motivation of using FPGA as an alternative to a microprocessor-based embedded system. The FPGA had also been used to control the multi-channel transmitters and receivers. The FPGA functionalities and advantages such as parallelism and reconfiguration and the FPGA resources such as BRAMs, soft microcontroller IP core had been used for the design and development of the Teletest® MK4 system.

The improvements made in the Teletest® MK4 system were compared with the previous MK3 system throughout the whole thesis. It is also explained how these improvements are implemented in new hardware design. It needs to be noted that an improvement can be made in many different ways in hardware design. The final solution was decided based on the compromise of many parameters such as component count, speed, size, weight, power consumption and cost. The system was designed based on the hardware components available at the time of the design. New FPGA devices have been developed by the manufactures during the development of the Teletest® MK4 system. These new FPGAs have advantages of lower power consumption, higher logic resource density, more IOBs and reduced cost. They can be considered for future updating of the system.

The concepts of FPGA based embedded system design used in this project are applicable over a range of FPGAs available from different vendors. The Teletest®

MK4 system design provides a solid and tangible example for FPGA based multi-channel transmitters and receivers system design and can be used as a reference for various applications.

Figure 8.1 shows the first product for the Teletest® MK4 system and its internal assembly of the first prototype unit. This product has been commercially successful globally since it was firstly released to the market in May 2011 for the improvements made by the new FPGA embedded system design over the previous Teletest® MK3 system.



(a) (b)

FIGURE 8.1: Teletest® MK4 Electronic Unit and Internal Assembly of the First Prototype System

8.2 Future Work

The PhD project involved the research and development to underpin the design and development of a commercial instrument. The system debug and fault correcting was one of the most time consuming part in the research. The scope of the project was limited by a number of factors such as the project budget, components lead time and PCB manufacture time.

The high profile FPGA devices have dedicated transceivers which can eliminate the necessarily of using either Ethernet MAC or PHY external device and integrate a complete network interface on-chip. Wireless communication can also be implemented on FPGA. These can further improve the system flexibility; reduce system component counts and power consumption and should be considered whenever the project budget allows.

This research work was a part of continuous development of the field. It raised questions for future research and provided seeds for further development. As it was anticipated that additional functions would be required in the near future to support further study and to carry out more complex LRUT testing, the Teletest® MK4 design does not only cover the minimum functional requirement for the current LRUT pipeline inspection applications, but also has the capability of implementing various custom functions by reconfiguring the FPGA. The project can be extended for further exploiting the capability of connecting multiple units to provide more transmit and receive channels for one test and implementing digital signal processing on the FPGA device utilizing the integral DSP48 elements. These subjects have been proposed for further extending the capability and flexibility of the Teletest® system to benefit the LRUT applications.

Daisy Chain

In order to be able to use the Teletest® MK4 system for LRUT applications which requires more than 24 transmitter and receiver channels without having to redesign the hardware system, which is a costly and time consuming process, it is proposed to connect multiple units in a daisy chain. In electrical and electronic engineering a daisy chain is a wiring scheme in which multiple devices are wired together in sequence or in a ring. The Teletest® daisy chain is proposed to connect multiple Teletest® MK4 system together to form more transmit and receive channels which can be used synchronously for certain applications. The challenge is to synchronize multiple Teletest® systems within a specified time delay margin.

Digital Filters

Future work can be done to compare the result of analogue filters and digital filters, and evaluate the possibility of allowing the filter function for all 24 receiver channels to be completely implemented within the FPGA to further reduce the analogue component count, power and size of the system.

References

[1] KF. Graff. *Wave Motion In Elastic Solids.* Clarendon Press, Oxford, 1975.

[2] BA. Auld. *Acoustic Fields and Waves in Solids*, volume 1. Krieger Publishing Company, Malabar, Florida, 1990.

[3] JL. Rose. *Ultrasonic Waves in Solid Media.* Cambridge University Press, Cambridge, UK, 1999.

[4] JL. Rose and PJ. Mudge. Flexural Mode Focusing in Pipe. In *The 8th European Conference on Non-Destructive Testing*, Barcelona, Spain, June 2002. European Federation for Non-Destructive Testing.

[5] M. Barr. Embedded Systems Glossary. http://www.netrino.com/Embedded-Systems/Glossary, 2011.

[6] P. Patel. Embedded systems design using FPGA. In *The 5th International Conference on Embedded Systems and Design*, Jan 2006.

[7] P. Catton, PJ. Mudge, D. D'Zurko, and J. Rose. Improved methodology for guided wave inspections of pipelines. *Pipeline & Gas Journal*, 1:36–44, 2008.

[8] PJ. Mudge and P. Catton. Quantification of Defect Size from Long Range Guided Wave Ultrasonic Tests on Pipes. In *AIP Conference Proceedings*, volume 975 of *Review of Progress in Quantitative Non-Destructive Evaluation*, pages 147–154, 2007.

[9] Changsen. Zhang and Dexin. Huang. Design of Multi-channel Digital Video Optical Transmitter Based on FPGA. In *Proceedings of the Third International Symposium on Electronic Commerce and Security Workshops(ISECS 2010)*, pages 091–093, Guangzhou, P. R. China, 29-31,July 2010.

[10] E.J. McDonald, N.W. Schlossberg, and E. Grayver. Hardware accelerated multichannel receiver. In *Aerospace conference, 2009 IEEE*, pages 1–7, 7-14 March 2009.

[11] JL. Rose. A baseline and vision of ultrasonic guided wave inspection potential. *Journal of Pressure Vessel Technology*, 124:273–282, 2002.

[12] DC. Gazis. Three Dimensional Investigation of the Propagation of Waves in Hollow Circular Cylinders. *J. Acoust. Soc. Am.*, 31:568–578, 1959.

[13] W. Mohr and P. Holler. On inspection of thin-walled tubes for torsional and longitudinal flaws by guided ultrasonic waves. *IEEE Trans. Sonics Ultrason*, SU-23:369–378, 1976.

[14] AH. Fitch. Observation of elastic-pulse propagation in axially symmetric and non-axially symmetric longitudinal modes of hollow cylinders. *Journal of the Acoustical Society of Americ*, 35(5):706–708, 1963.

[15] R. Kumar. Flexural vibrations of fluid-filled circular cylindrical shells. *Acoustic*, 24:137–146, 1971.

[16] R. Kumar. Dispersion of axially symmetric waves in empty and fluid-filled cylindrical shells. *Acoustic*, 27(6):317–329, 1972.

[17] PJ Mudge. Field application of the Teletest long-range ultrasonic testing. *Insight*, 43:74–77, 2001.

[18] DN. Alleyne, B. Pavlakovic, MJS. Lowe, and Cawley P. Rapid, Long Range Inspection of Chemical Plant Pipework using Guided Waves. *Insight*, 43:93–96,101, 2001.

[19] J. Barshinger, JL. Rose, and M.J. Avioli Jr. Guided wave resonance tuning for pipe inspection. *Journal of Pressure Vessel technology*, 124:303–310, 2002.

[20] Y.Y. Kim, C.I. Park, S.H. Cho, and S.W. Han. Torsional wave experiments with a new magnetostrictive transducer configuration. *J. Acoust. Soc. Am.*, 117(6):3459–3468, 2005.

[21] DE. Amos. A remark on algorithm 644: a portable package for bessel functions of a complex argument and non-negative order. *ACM transactions on mathematical Software*, 21(4):388–393, 1995.

[22] MJS. Lowe. Matrix techniques for modelling ultrasonic waves in multilayered media. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 42(4):525–542, 1995.

[23] A. Demma, P. Cawley, M. Lowe, and AG. Roosenbrand. The Reflection of Fundamental Torsional Mode from Cracks and Notches in Pipes. *J. Acoust. Soc. Am.*, 114(3):611–625, 2003.

[24] A.J. Jerri. The Shannon sampling theorem - Its various extensions and applications: A tutorial review. In *Proceedings of the IEEE*, volume 65, pages 1565 – 1596, Nov. 1977.

[25] I. Kuon and J. Rose. Measuring the gap between fpgas and asics. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 26:203 – 215, Jan 2007.

[26] J. Blyler. Navigating the Silicon Jungle: FPGA or ASIC? http://chipdesignmag.com/display.php?articleId=115&issueId=11, June/July 2005.

[27] Xilinx. XtremeDSP DSP48A for Spartan-3A DSP FPGAs User Guide,UG431 (v1.3), July 15 2008.

[28] Microsemi SoC Products Group(formerly Actel). http://www.actel.com/.

[29] Lattice Semiconductor Corporation. http://www.latticesemi.com/.

[30] Tim. Behne. FPGA Clock Schemes, Embedded Systems Programming. http://www.eetimes.com/story/OEG20030210S0053, March 13 2003.

[31] H.F.-W. Sadrozinski and J. Wu. *Applications of Field-Programmable Gate Arrays in Scientific Research*. Taylor and Francis Group, LLC, 2010. Figure 1.2.

[32] Xilinx. Spartan-3A DSP FPGA Family: Introduction and Ordering Information, DS610-1 (v2.2), March 11 2009.

[33] Xilinx. Spartan 3 Generation FPGA User Guide, Extended Spartan-3A, Spartan-3E, and Spartan-3 FPGA Families, Chapter 10, UG331 (v1.8), June 13 2011.

[34] Xilinx. Spartan-3 FPGA Family Data Sheet, DS099, December 4 2009.

[35] Xilinx. Using Digital Clock Managers (DCMs) in Spartan-3 FPGAs, XAPP462 (v1.1), January 5 2006.

[36] S. Kilts. *Advanced FPGA Design Architecture, Implementation, and Optimization*. John Wiley & Sons, Inc, 2007.

[37] Xilinx. ChipScope Pro 11.1 Software and Cores User Guide, UG029(v11.1), April 24, 2009.

[38] Xilinx. ChipScope ILA Tools Tutorial, UG044(v4.2.2), July 24 2003.

[39] E. Monmasson and M.N. Cirstea. FPGA Design Methodology for Industrial Control Systems - A Review. *Cergy-Pontoise Univ., Pontoise, IEEE Transactions on Industrial Electronics, Issue:4*, 54:1824 – 1842, Aug. 2007.

[40] Ronald. Sass and Andrew G. Schmidt. *Embedded Systems Design with Platform FPGAs: Principles and Practices: Principles, Practices and Economics*. Morgan Kaufmann, July 2010.

[41] Jack. Ganssle. *The Art of Designing Embedded Systems*. Elsevier, May, 2008.

[42] PP. Chu. *RTL Hardware Design Using VHDL: Coding for Efficiency, Portability, and Scalability*. Hohn Wiley & Sons, Inc, Hoboken, New Jersey., 2006.

[43] PP. Chu. *FPGA Prototyping by VHDL Examples: Xilinx Spartan-3 Version*. Hohn Wiley & Sons, Inc, Hoboken, New Jersey., 2008.

[44] Rahul. Dubey. *Introduction to Embedded System Design Using Field Programmable Gate Arrays*. Springer, 2011.

[45] Ershen. Wang, Shufang. Zhang, and Xiaowen. Hu Qing. Yi, Jiang. Sun. Implementation of an Embedded GPS Receiver Based on FPGA and MicroBlaze. In *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on*, pages 1–4, October 2008.

[46] C.C.W. Robson, A. Bousselham, and C. Bohm. An FPGA based general purpose data acquisition controller. *Real Time Conference, 14th IEEE-NPSS, 10-10 June 2005*, 53:4, 2005.

[47] Atmel. *TS80C32X2-MCB, 8-bit Microcontroller 8 Kbytes ROM/OTP, ROMless, Atmel*, Feb 2008.

[48] STMicroelectronics. Flash In-System Programmable (ISP) Peripherals For 8-bit MCUs, PSD913F2, January 2002.

[49] EU. Directive 2002/95/EC of the European Parliament and of the Council of 27 January 2003 on the restriction of the use of certain hazardous substances in electrical and electronic equipment, Official Journal L 037, 13/02/2003 P. 0019 - 0023. http://eur-lex.europa.eu/, 2002.

[50] Micron Technology, Inc. MT47H32M16 - 8 Meg x 16 x 4 banks DDR2 SRAM, 2004.

[51] Xilinx. Implementing DDR2 SDRAM Controllers, Chapter 8, Xilinx Memory Interface Generator (MIG) User Guide, Version 3.1, 24 June 2009.

[52] NuHorizons Electronics. *Interfacing Micron DDR2 Memories to Xilinx Spartan-3A/AN FPGAs:A Step-By-Step Guide, AN086 (Version 1.0)*, February 18, 2008.

[53] Xilinx. XPS Ethernet Lite Media Access Controller, Xilinx Product Specification, DS580, September 16 2009.

[54] Avnet. Embedded Networking with Xilinx MicroBlaze, Speedway Design Workshops, Version 1.2, Feb, 2008.

[55] P. Catton. *The Use of Ultrasonic Guided Waves for the In Situ Inspection of Industrial Pipelines for Corrosion Damage.* PhD thesis, School of Design and Engineering, Brunel University, Sep, 2008.

[56] J.-L. Robert, M. Burcher, C. Cohen-Bacrie, and M. Fink. Time reversal operator decomposition with focused transmission and robustness to speckle noise: Application to microcalcification detection. *J. Acoust. Soc. Am.*, 119:3848–3859, 2006.

[57] Cypress Semiconductor. Asynchronous, Cascadable 8K/16K/32K/64K x9 FIFOs. San Jose, USA: Cypress Semiconductor., October 4 1999.

[58] Analog Devices. *AD9760, 10-Bit, 125 MSPS, TxDAC D/A Converter*, 2000.

[59] TI. DAC8580, 16-BIT, HIGH-SPEED, LOW-NOISE, VOLTAGE OUTPUT, DIGITAL-TO-ANALOG CONVERTER, June 2005.

[60] TI. ADS7886 Datasheet, Texas Instruments, SLAS492, September 2005.

[61] ISSI. IS61WV102416ALL, 1M x 16 HIGH-SPEED ASYNCHRONOUS CMOS STATIC RAM WITH 3.3V SUPPLY, June 2009.

[62] Xilinx. XPS Multi-Channel External Memory Controller (XPS MCH EMC) (v3.01a),DS575, July 10 2009.

[63] Xilinx. Using Block RAM in Spartan-3 Generation FPGAs, Xilinx Application Note: Spartan-3 FPGA Family, XAPP463 (v2.0), March 1 2005.

[64] Xilinx. Multi-Port Memory Controller (MPMC) (v5.04.a), Xilinx Product Specification, DS643, December 1 2009.

[65] Xilinx. LMB BRAM Interface Controller (v2.10b), Xilinx Product Specification, DS452, December 2 2009.

[66] Xilinx. MicroBlaze Processor Reference Guide, Embedded Development Kit EDK 11.4, UG081 (v10.3), October 26 2009.

[67] Xilinx. Floorplanning Methodology Guide, UG633 (v 12.1), May 3 2010.

[68] Xilinx. PlanAhead User Guide. Included in the PlanAhead UG632 (v 11.4), 2009.

FPGA Embedded System for Ultrasonic Non-Destructive Testing

Lei Zhang

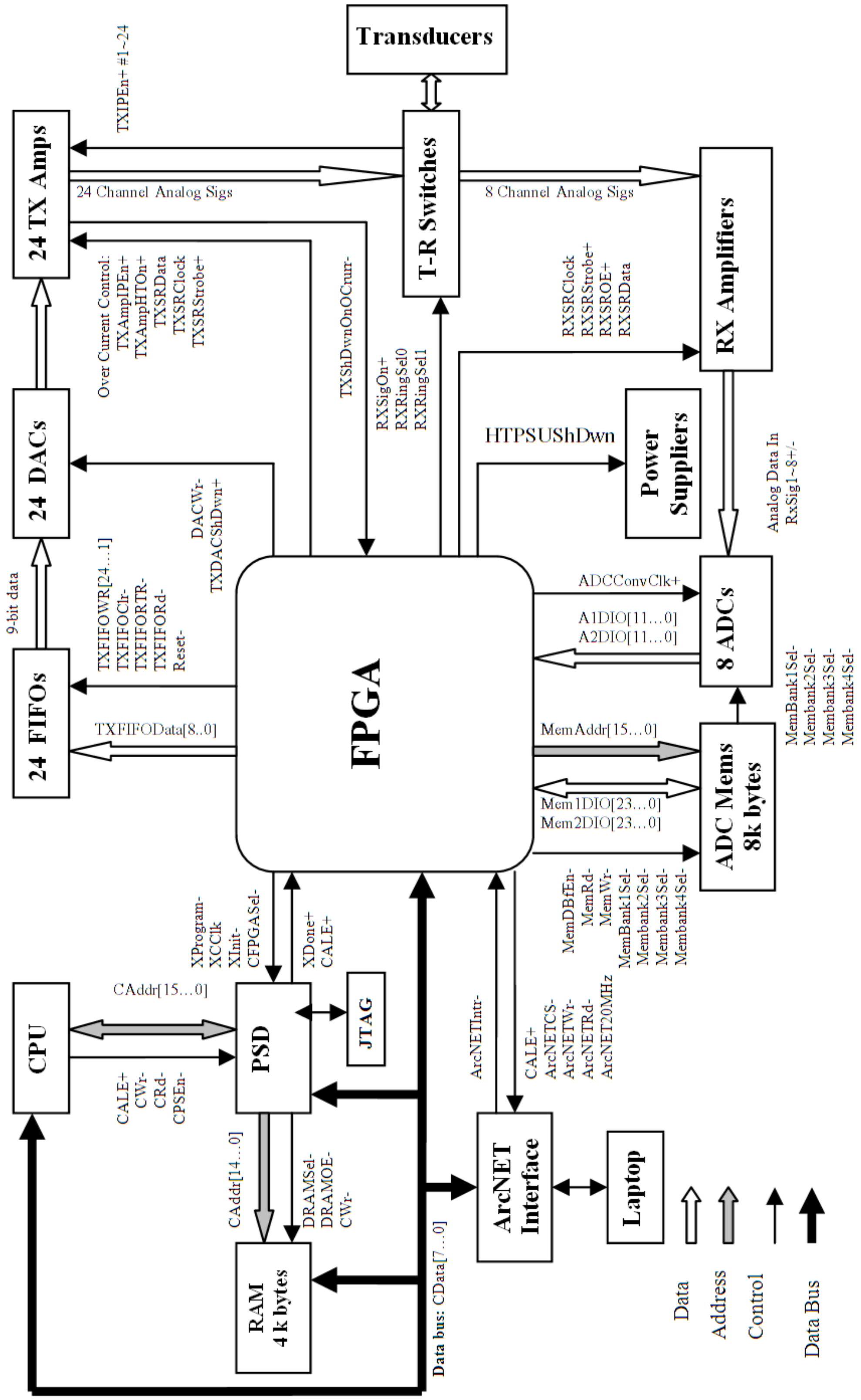
November 2011

Appendices

Notice

Appendix A

Teletest® MK3 Hardware Block Diagram


Transducers
T-R Switches
24 TX Amps
TXIPEn+ #1~24
24 Channel Analog Sigs
8 Channel Analog Sigs
RX Amplifiers
Over Current Control:
TXAmpIPEn+
TXAmpHTOn+
TXSRData
TXSRClock
TXSRStrobe+
TXShDwnOnOCurr-
RXSigOn+
RXRingSel0
RXRingSel1
RXSRClock
RXSRStrobe+
RXSROE+
RXSRData
24 DACs
DACWr-
TXDACShDwn+
HTPSUShDwn
Power Suppliers
Analog Data In
RxSig1~8+/-
9-bit data
24 FIFOs
TXFIFOWR[24…1]
TXFIFOClr-
TXFIFORTR-
TXFIFORd-
Reset-
TXFIFOData[8..0]
FPGA
ADCConvClk+
A1DIO[11…0]
A2DIO[11…0]
8 ADCs
MemAddr[15…0]
Mem1DIO[23…0]
Mem2DIO[23…0]
ADC Mems
8k bytes
MemBank1Sel-
Membank2Sel-
Membank3Sel-
Membank4Sel-
MemDBfEn-
MemRd-
MemWr-
MemBank1Sel-
Membank2Sel-
Membank3Sel-
Membank4Sel-
XProgram-
XCClk
XInit-
CFPGASel-
XDone+
CALE+
CPU
CAddr[15…0]
PSD
JTAG
CALE+
CWr-
CRd-
CPSEn-
CAddr[14…0]
DRAMSel-
DRAMOE-
CWr-
RAM
4 k bytes
Data bus: CData[7…0]
ArcNETIntr-
CALE+
ArcNETCS-
ArcNETWr-
ArcNETRd-
ArcNET20MHz
ArcNET
Interface
Laptop
Data
Address
Control
Data Bus

Appendix B

Teletest® MK4 FOCUS System FPGA Design Specification

Teletest MK4 FOCUS System

FPGA Design Specification

Plant Integrity Ltd

Project:	Teletest FOCUS MK4
Department:	Plant Integrity Ltd
Section Manager:	Paul Jackson
Technical Manager:	Peter Mudge
Production Manager:	Barry Elborn
Design Engineer:	Lei Zhang
First Related Date:	30-March-2011

Contents

Revision History

Revision	***Description***	***Data***
Rev 1.1	First Released	06-Jan-2011
Rev 1.2	Fit_timer IP interval time change from 10ms to 2ms to increase MicroBlaze operating system performance	27-Jan-2011
Rev 1.3	Added RX clock rate selection function	03-Feb-2011
Rev 1.4	Fixed problem of PRF when receiving time is longer than PRF interval	14-Feb-2011
Rev 1.4. e	(1) Top Level Sequencer VHDL changed for transmitter switched off immediately after over-current. (2) RX Amp loader VHDL changed for filter selection decoder. (3) SPI Interface for voltage and temperature measurement ADC has been modified: a. Added FIFO b. Data bit width set to 8-bit (4) Added 32 bit timer for HARDWARE watchdog.	18-Feb-2011
Rev 1.4.f	SPI IP set to 16-bit data wide for Debugging	24-Feb-2011
Rev 1.4.g	Added GPIO_1 port 2 as front panel controller reset	28-Feb-2011
Rev 1.4 h	MicroBlaze parameters have been modified to increase Ethernet performance a. Enabled Barrel Shifter b. Enabled Integer Divider c. Deselected optimization for area	1-Mar-2011
Rev 1.4 i	Released version for MK4 prototype	8-Mar-2011
Rev 1.4 k	Implemented TX Interpolation Rate Register	10-Mar-2011
Rev 1.4 l	(1) Add third UART for GPS (2) Add bit 9 for control status register to indicate HT PSU Good ON for MicroBlaze to test the PSU on temperature.	30-Mar-2011

Executive Summary

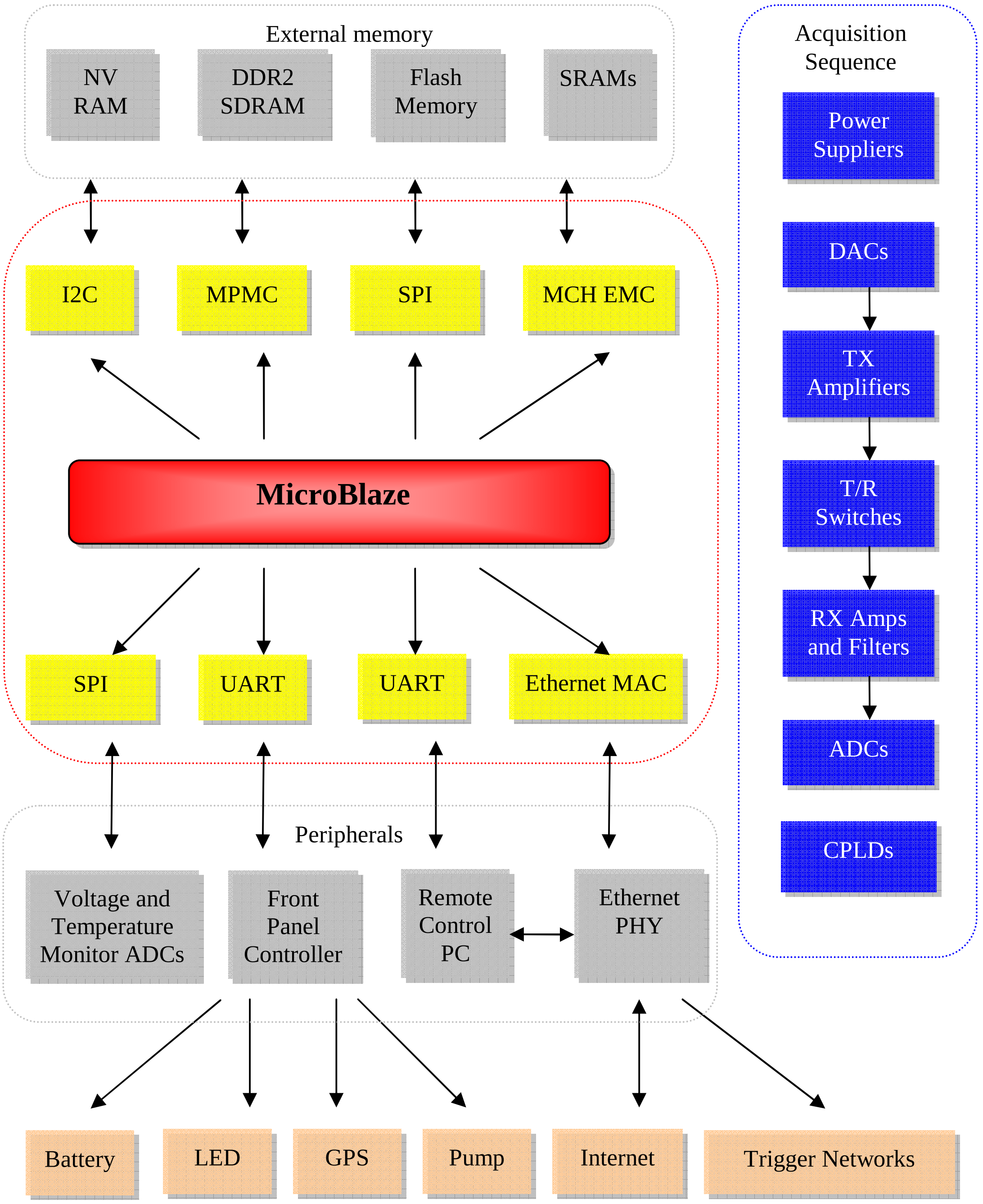
The outline of this document is summarised as below:

- Section 1 introduces the Teletest MK4 FOCUS system hardware and its block diagram.
- Section 2 provides a FPGA system block diagram and gives brief description of each function block in the FPGA design.
- Section 3 gives the technical description of each FPGA block for the acquisition logic.
- Section 4 gives the technical description of the MicroBlaze embedded system implemented in the FPGA.
- Section 5 describes the MicroBlaze and acquisition logic interface specification.
- Section 6 provides design summary.
- Section 7 discusses future functions.

1 Introduction of Hardware

This section briefly introduces the Teletest MK4 Focus system hardware and function requirement of the FPGA design.

1.1 Hardware Block Diagram



1.2 Hardware Structure and General Descriptions

The system hardware structure is illustrated in Figure 1. The peripheral hardware devices are divided into two groups with the following perspectives:

The MicroBlaze implemented on the FPGA controls the peripheral devices through standard peripheral controller IP cores which are also implemented on the FPGA. These devices are shown in grey and their corresponding interface controllers are shown in yellow in the block diagram.

The multi-channel transmitters and receivers run in parallel during acquisition sequence and the related hardware devices are therefore controlled directly by FPGA acquisition logic in order to meet critical time requirement. These devices are shown in blue in the block diagram.

The MicroBlaze embedded system is interfaced to the data acquisition system via a list of read/write registers.

1.3 FPGA

Part No: XC3SD1800A-4FGG676C

Manufacture: Xilinx

Quantity: 1

The FPGA device is the central controller of the system. The soft microprocessor MicroBlaze, external memory and peripheral controllers, as well as the transmitting and receiving acquisition sequence logics are all implemented using the FPGA resources.

1.4 MicroBlaze Controlled Hardware Devices

The MicroBlaze controlled hardware devices includes external memory and peripherals, the memory controllers and peripheral controllers are all implemented on a single FPGA device. The external memory devices are DDR2 SDRAM, Flash memory, Non volatile RAM and three banks SRAM. The hardware peripherals are temperature and voltage monitor ADCs, Ethernet port linking to remote control PC, and UART ports linking to GPS device and front panel 8051 controller. The front panel controller controls pumps, LED, and battery directly.

1.4.1 System External Memory Devices

There are **four** types of external memory devices in this system.

1.4.1.1 DDR2 SDRAM memory

Part No: MT47H32M16HR-25E:F

Manufacture: Micron

Quantity: 2

Two DDR2 SDRAM memory chips are used to for 32 M x 32 bit wide memory to provide 128 Bytes fast accessible memory for 32-bit MicroBlaze processor. The memory is used for running firmware applications and its full memory range is cashable for MicroBlaze. The Multi-Port Memory Controller (MPMC) IP implemented in the FPGA interconnects to MicroBlaze via Xilinx CashLink (XCL) bus and provides the memory device with low-level control signals. The memory device uses 125MHz differential clocks generated by Digital Clock Manager (DCM), which is also implemented on FPGA, and the maximum memory access speed is 250MHz.

1.4.1.2 SPI Flash Memory

Part No: AT45DB642D-CNU

Manufacture: Atmel

Quantity: 1

The Flash memory is used to store the FPGA configuration bit stream and firmware application. It has SPI interface working at 31.25 MHz clock rate. The Atmel SPI Flash memory provides 64 Megabit and has low power dissipation.

1.4.1.3 Non-volatile RAM

This NV RAM is located in the remote tool lead to store the test information for permanent mount application. A transistor buffered I2C interface is provided in the embedded system. An I2C controller IP is implemented on the FPGA and provide 400kHz serial clock to the device.

1.4.1.4 SRAM

Part No: IS61WV102416BLL

Manufacture: ISSI

Quantity: 6

The SRAM devices provide 12 Mega Bytes (9 Mega 32-bit samples) memory for storing received signals. There are three identical banks of SRAM. Each bank stores data for 8 channels receivers. Xilinx Multi-CHannel (MCH) PLBV46 external memory controller is implemented on the FPGA for MicroBlaze to access the SRAM via the Processor Local Bus (PLB). Three controllers are used for three banks SRAM. MicroBlaze can only access one bank at a time via PLB. These three SRAM banks are also controlled directly by FPGA state machine sequencer logic during receive sequence, whereby three banks SRAM can be accessed in parallel for all 24 channels receivers.

1.4.2 Peripherals

1.4.2.1 Power Supply Voltage and Temperature Monitor ADCs

Part No: ADC088S022

Manufacture: National Semiconductor

Quantity: 2

There are various power supplies in the system: +/-5V, +3.3V, +/-150V and +/-24V. These power supplies are monitored and the measured value is digitised using an 8-channel ADC with SPI interface. This is a SPI controller IP implemented on the FPGA and it works at 1.953125MHz. (1:32 of the 62.5MHz MicroBlaze system clock rate) Thermocouples are used for measuring temperatures of the PSU and TX Amplifiers. A second 8-channel ADC with SPI interface is used to digitise the temperature measurements. The same SPI controller is shared by both ADCs.

1.4.2.2 Ethernet PHY

Part No: LAN8710

Manufacture: SMSC

Quantity: 1

The single chip Ethernet Physical layer Transceiver (PHY) provides a 10/100Mbit/s communication link between the embedded unit and remote control PC. The related Ethernet MAC controller is implemented in the FPGA and is interfaced to MicroBlaze via PLB bus.

Note: It is suggested that high speed 1GMbit/s Ethernet PHY and hardcore Ethernet MAC should be used to increase the overall communication speed between embedded unit and remote control PC for future development.

1.4.2.3 Front Panel Controller

Part No: C8051F010-GQ

Manufacture: Silicon Labs

Quantity: 1

The 8051 front panel controller monitors the battery status, directly controls the front panel buttons and LED display, as well as starts and stops the in-system inflation pump. It is interfaced to the FPGA via UART. An extra UART is added to connect the FPGA directly to the GPS controller which is previously connected to the front panel in order to simply the firmware control process.

1.5 Acquisition Logic Controlled Hardware Devices

The acquisition logic controlled hardware devices form a transmitting and receiving system which includes 24 identical channels. Each channel is composed of DAC, transmitter amplifier, transmitter and receiver switch, receiver amplifiers and filters, ADC. 4 CPLD devices are used for 24-channel receiver gain and filter control. The functional parameters of these devices are setup by the embedded system. In a normal test routine, the remote PC sends messages with test request and test functional parameters to the embedded system, which decodes these messages, setups the acquisition parameters, and then requests the acquisition logic to run the transmitting and receiving sequence. The collected signal is stored in the embedded system local memory. On the completion of a test, the PC can request the MicroBlaze embedded system to send the received data to the PC for display and analyse.

1.5.1 Power Supplies

The HT power supply and DAC power supply are only switched on during transmitting in order to reduce system power consumption.

The ADC power supply is only switched on during transmitting and receiving in order to reduce system power consumption.

1.5.2 Transmit Waveform Generation DAC

Part No: DAC8580

Manufacture: Texas Instruments

Quantity: 24

The 24-channel DACs have 16-bit resolution and serial data interface to the FPGA. They are driven in parallel by a custom DAC controller implemented on the FPGA. These DACs devices have integrated interpolation filters with adjustable interpolation rate of x2, x4, x8 and x16.

Note: In LRUT applications, transmit waveforms are filtered by capacitance load of piezoelectric transducers; therefore the DAC resolution is not critical for generating smooth signals.

1.5.3 HT amplifiers for transmit waveforms

There are 24-channel TX amplifiers corresponding to 24-channel DACs. Each TX amplifier is capable of driving a maximum of 16 piezoelectric transducers (PZTs) up to 300V peak to peak with arbitrary waveforms within a bandwidth between 10 kHz to 100 kHz. The equivalent load of a typical PZT is a 950 pF capacitor in serial with a 68 Ohm resistor.

Note: In LRUT applications, high TX output voltage can increase the test distance but also cause high noise level, so practically it does not increase the signal-to-noise ratio (SNR). Besides, high voltage output level consumes more power and is undesirable for battery-powered portable instrument. Therefore it was proposed to reduce this output voltage down to 100V peak-to-peak in order to reduce size, weight and power consumption for the multi-channel system. However, experiments carried out on typical pipelines indicate that 100Vpp voltage level causes 20 inches loss in testing distance compared to 300Vpp due to high attenuation of the transmitting signals. 300Vpp output remains for the MK4 system. It is suggested that further test to be carried out on different pipelines using lower transmitting voltage combining with digital filtering functions to investigate the possibilities of further improving system characteristics such as size, weight and power consumption.

1.5.4 Transmit-receive switches

The 24-channel transmit-receive switches control whether the transmitters or receivers is switched to the PZT transducers. They can be individually set up with two optional configuration modes.

Mode 1: The switch is switched to transmitter at the start of an acquisition. On completion of waveform transmitting, the switch is switched to receiver.

Mode 2: The corresponding channel does not transmit and is switched to receive at the start of an acquisition.

1.5.5 Receiver ADCs

Part No: ADS7886

Manufacture: Texas Instruments

Quantity: 24

These 12-bit ADCs have serial data interface and a fixed serial clock rate of 20MHz. The wideband noise level in the receive signal makes it unnecessary to use ADC with higher resolution. Four converting rates options available: 1MSPS, 500kSPS, 250kSPS, 125kSPS. This must be chosen based on the transmit waveform frequency. The RX converting rate must be at least double the transmit waveform frequency and it is suggested to be 10 times of the transmit waveform frequency in order to finely reconstruct the received signals. The merit of using lower RX sampling frequency is to increase the test distance. As each RX channel has 128 k samples memory, which can capture data for 128 ms at 1MSPS, or 1024ms at 125kSPS.

1.5.6 Receive Amplifiers and filters

24-channel receive amplifiers can be configured individually with 1 dB step from 20 dB to 100 dB. The variable gain adjust the receive signal to full scale of the ADC input in order to achieve a better digitization resolution.

A series of first order RC filters with 300kHz, 150kHz, 75kHz, 32kHz and 15kHz cut-off frequencies can be configured in common for all receiver channels. These filters are used to restrict bandwidth of the received signal to satisfy the Nyquist-Shannon Theory. The cut-off frequency is set to be less than the Nyquist frequency (half of the sampling frequency) to suppress the mirror image and harmonics of the signals as well as broadband noise. The signal and its mirror image are symmetrical about the Nyquist frequency. An ideal filter would have a zero transient band in spectrum; while in practical, a simple first order RC anti-aliasing filter has a long transient band. The cut-off frequency, also called -3dB frequency, is the frequency where the signal amplitude is -3dB of the pass band amplitude.

Note: It is suggested to use various gain control device and digital filters in order to reduce the size and power consumption of analogue circuitry. Testing will be carried out to compare the results of analogue filters and digital filters.

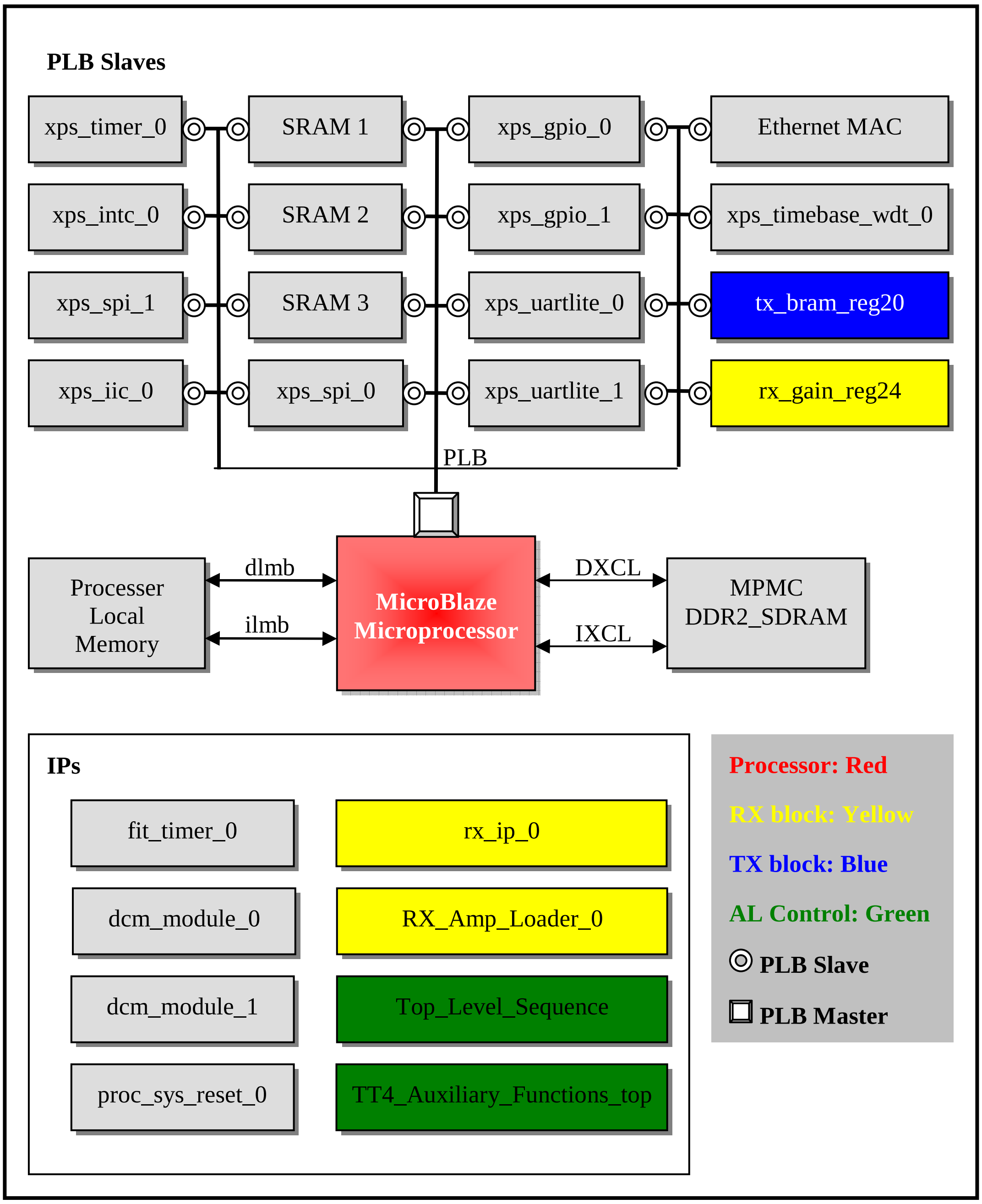
1.5.7 CPLDs for RX Amplifiers and filters control

Four CPLDs are used to add extra IOs for controlling RX amplifier and filters. Each RX channel can be configured individually with an 8-bit gain registers. The filters for all 24-channel RX are setup in common.

The detail information of the hardware electronics design is not covered in document. The hardware block diagram is provided to illustrate the relationship between the hardware peripherals and their respective peripheral controllers which are implemented in the FPGA design.

2 FPGA Design Block Diagram

2.1 FPGA Block Diagram



2.2 FPGA Design Overview

The function blocks in the FPGA block diagram are explained briefly to help the author for a fundamental understanding of the FPGA design structure. The technical specifications of these function blocks are described in section 3, 4 and 5. As shown in Figure 2, the FPGA design includes one soft microprocessor, one memory controller on processor local memory bus(LMB), one memory controller on Xilinx

Cashlink (XCL) bus, sixteen slaves IP cores on the processor local bus (PLB), and 8 other IP cores not connecting to MB.

2.2.1 FPGA Design Specifics

EDK Version	11.4
Device Family	spartan3adsp
Device	xc3sd1800afg676-4

2.2.2 XPS IP Cores Used

	IP Core	IP Version
1	MicroBlaze	7.20.d
1	Processor Local Bus (PLB) 4.6	1.04.a
2	Local Memory Bus (LMB) 1.0	1.00.a
1	Block RAM (BRAM) Block	1.00.a
2	LMB BRAM Controller	2.10.b
1	Multi-Port Memory Controller(DDR/DDR2/SDRAM)	5.04.a
3	XPS Multi-Channel External Memory Controller(SRAM/Flash)	3.01.a
1	XPS 10/100 Ethernet MAC Lite	3.01.a
2	Digital Clock Manager (DCM)	1.00.d
1	MicroBlaze Debug Module (MDM)	1.00.g
1	Processor System Reset Module	2.00.a
2	XPS UART (Lite)	1.01.a
1	XPS Watchdog Timer	1.01.a
1	Fixed Interval Timer	1.01.a
2	XPS General Purpose IO	2.00.a
1	XPS IIC Interface	2.02.a
2	XPS SPI Interface	2.01.b
1	XPS Interrupt Controller	2.00.a
1	XPS Timer/Counter	1.01.b

2.2.3 External Ports and Corresponding Signals

NAME	DIR	[LSB: MSB]	SIG
SHARED			
fpga_0_rst_1_sys_rst_pin	**I**	**1**	sys_rst_s
DDR2_SDRAM			
fpga_0_DDR2_SDRAM_DDR2_DQS_Div_I_pin	**I**	**1**	fpga_0_DDR2_SDRAM_DDR2_DQS_Div_I_pin
fpga_0_DDR2_SDRAM_DDR2_DQS_n_pin	**IO**	**3:0**	fpga_0_DDR2_SDRAM_DDR2_DQS_n_pin
fpga_0_DDR2_SDRAM_DDR2_DQS_pin	**IO**	**3:0**	fpga_0_DDR2_SDRAM_DDR2_DQS_pin
fpga_0_DDR2_SDRAM_DDRQ_pin	**IO**	**31:0**	fpga_0_DDR2_SDRAM_DDR2_DQ_pin
fpga_0_DDR2_SDRAM_DDR2_Addr_pin	**O**	**12:0**	fpga_0_DDR2_SDRAM_DDR2_Addr_pin
fpga_0_DDR2_SDRAM_DDR2_BankAddr_pin	**O**	**1:0**	fpga_0_DDR2_SDRAM_DDR2_BankAddr_pin
fpga_0_DDR2_SDRAM_DDR2_CAS_n_pin	**O**	**1**	fpga_0_DDR2_SDRAM_DDR2_CAS_n_pin
fpga_0_DDR2_SDRAM_DDR2_CE_pin	**O**	**1**	fpga_0_DDR2_SDRAM_DDR2_CE_pin
fpga_0_DDR2_SDRAM_DDR2_CS_n_pin	**O**	**1**	fpga_0_DDR2_SDRAM_DDR2_CS_n_pin
fpga_0_DDR2_SDRAM_DDR2_Clk_n_pin	**O**	**1:0**	fpga_0_DDR2_SDRAM_DDR2_Clk_n_pin
fpga_0_DDR2_SDRAM_DDR2_Clk_pin	**O**	**1:0**	fpga_0_DDR2_SDRAM_DDR2_Clk_pin
fpga_0_DDR2_SDRAM_DDR2_DM_pin	**O**	**3:0**	fpga_0_DDR2_SDRAM_DDR2_DM_pin
fpga_0_DDR2_SDRAM_DDR2_DQS_Div_O_pin	**O**	**1**	fpga_0_DDR2_SDRAM_DDR2_DQS_Div_O_pin
fpga_0_DDR2_SDRAM_DDR2_ODT_pin	**O**	**1**	fpga_0_DDR2_SDRAM_DDR2_ODT_pin
fpga_0_DDR2_SDRAM_DDR2_RAS_n_pin	**O**	**1**	fpga_0_DDR2_SDRAM_DDR2_RAS_n_pin

fpga_0_DDR2_SDRAM_DDR2_WE_n_pin	**O**	**1**	fpga_0_DDR2_SDRAM_DDR2_WE_n_pin
Ethernet_MAC			
fpga_0_Ethernet_MAC_PHY_col_pin	**I**	**1**	fpga_0_Ethernet_MAC_PHY_col_pin
fpga_0_Ethernet_MAC_PHY_crs_pin	**I**	**1**	fpga_0_Ethernet_MAC_PHY_crs_pin
fpga_0_Ethernet_MAC_PHY_dv_pin	**I**	**1**	fpga_0_Ethernet_MAC_PHY_dv_pin
fpga_0_Ethernet_MAC_PHY_rx_clk_pin	**I**	**1**	fpga_0_Ethernet_MAC_PHY_rx_clk_pin
fpga_0_Ethernet_MAC_PHY_rx_data_pin	**I**	**3:0**	fpga_0_Ethernet_MAC_PHY_rx_data_pin
fpga_0_Ethernet_MAC_PHY_rx_er_pin	**I**	**1**	fpga_0_Ethernet_MAC_PHY_rx_er_pin
fpga_0_Ethernet_MAC_PHY_tx_clk_pin	**I**	**1**	fpga_0_Ethernet_MAC_PHY_tx_clk_pin
fpga_0_Ethernet_MAC_PHY_MDIO_pin	**IO**	**1**	fpga_0_Ethernet_MAC_PHY_MDIO_pin
fpga_0_Ethernet_MAC_PHY_MDC_pin	**O**	**1**	fpga_0_Ethernet_MAC_PHY_MDC_pin
fpga_0_Ethernet_MAC_PHY_rst_n_pin	**O**	**1**	fpga_0_Ethernet_MAC_PHY_rst_n_pin
fpga_0_Ethernet_MAC_PHY_tx_data_pin	**O**	**3:0**	fpga_0_Ethernet_MAC_PHY_tx_data_pin
fpga_0_Ethernet_MAC_PHY_tx_en_pin	**O**	**1**	fpga_0_Ethernet_MAC_PHY_tx_en_pin
RX_Amp_Loader_top_0			
RX_Amp_Loader_top_0_SClockOut_pin	**O**	**1**	RX_Amp_Loader_top_0_SClockOut
RX_Amp_Loader_top_0_SData1Out_pin	**O**	**1**	RX_Amp_Loader_top_0_SData1Out
RX_Amp_Loader_top_0_SData2Out_pin	**O**	**1**	RX_Amp_Loader_top_0_SData2Out
RX_Amp_Loader_top_0_SData3Out_pin	**O**	**1**	RX_Amp_Loader_top_0_SData3Out
RX_Amp_Loader_top_0_SData4Out_pin	**O**	**1**	RX_Amp_Loader_top_0_SData4Out
TT4_Auxiliary_Functions_top_0			
ChannelSwitchAlternatelDrvLoOut_pin	**O**	**1**	ChannelSwitchAlternateDrvLo
ChannelSwitchNormalDrvLoOut_pin	**O**	**1**	ChannelSwitchNormalDrvLo
Front_Pannel_Reset_n_pin	**O**	**1**	Front_Pannel_Reset_n_Out
TT4_Top_Level_Sequencer_top_0			
HTPSUsOPGoodHiIn_pin	**I**	**1**	HTPSUsOPGoodHiIn
TXAmpsOvrCurrentLowIn_pin	**I**	**1**	TXAmpsOvrCurrentLowIn
HTPSUOnHiOut_pin	**O**	**1**	HTPSUOnHiOut
RXAmpPowerOnHiOut_pin	**O**	**1**	RXAmpPowerOnHiOut
TRSwitchFastOnAtTXEndOut_pin	**O**	**1**	TRSwitchFastOnAtTXEndOut
TRSwitchFastOnAtTXStartOut_pin	**O**	**1**	TRSwitchFastOnAtTXStartOut
TRSwitchHoldOnAtTXEndOut_pin	**O**	**1**	TRSwitchHoldOnAtTXEndOut
TRSwitchHoldOnAtTXStartOut_pin	**O**	**1**	TRSwitchHoldOnAtTXStartOut
TXAmpPowerOnHiOut_pin	**O**	**1**	TXAmpPowerOnHiOut
DACPwrOnHi_pin	**O**	**1**	TXAmpPowerOnHiOut
dcm_module_0			
fpga_0_clk_1_sys_clk_pin	**I**	**1**	dcm_clk_s
dcm_module_1			
clk_16MHz_180_pin	**O**	**1**	clk_16MHz_180
rx_ip_0_adc_clk	**O**	**1**	rx_clk_180_out
rx_ip_0			
rx_ip_0_adc_1_sdo_pin	**I**	**7:0**	rx_ip_0_adc_1_sdo
rx_ip_0_adc_2_sdo_pin	**I**	**7:0**	rx_ip_0_adc_2_sdo
rx_ip_0_adc_3_sdo_pin	**I**	**7:0**	rx_ip_0_adc_3_sdo
BANK_1_SRAM_D_pin	**IO**	**0:31**	BANK_1_SRAM_D
BANK_2_SRAM_D_pin	**IO**	**0:31**	BANK_2_SRAM_D
BANK_3_SRAM_D_pin	**IO**	**0:31**	BANK_3_SRAM_D
SRAM_BANK1_CE_pin	**O**	**1**	SRAM_BANK1_CE
SRAM_BANK1_OE_pin	**O**	**1**	SRAM_BANK1_OE
SRAM_BANK1_WE_pin	**O**	**1**	SRAM_BANK1_WE
SRAM_BANK2_CE_pin	**O**	**1**	SRAM_BANK2_CE
SRAM_BANK2_OE_pin	**O**	**1**	SRAM_BANK2_OE
SRAM_BANK2_WE_pin	**O**	**1**	SRAM_BANK2_WE
SRAM_BANK3_CE_pin	**O**	**1**	SRAM_BANK3_CE
SRAM_BANK3_OE_pin	**O**	**1**	SRAM_BANK3_OE
SRAM_BANK3_WE_pin	**O**	**1**	SRAM_BANK3_WE
fpga_0_SRAM_Mem_A_pin	**O**	**10:29**	SRAM_A_MUX_concat
rx_ip_0_adc_cs_n_pin	**O**	**1**	rx_ip_0_adc_cs_n
tx_bram_reg20_0			
TXAmpIPEnHi_pin	**O**	**1**	TXAmpIPEnHi
tx_ip_0_bsb_n_pin	**O**	**1**	tx_ip_0_bsb_n
tx_ip_0_fsnc_pin	**O**	**1**	tx_ip_0_fsnc
tx_ip_0_osr1_pin	**O**	**1**	tx_ip_0_osr1
tx_ip_0_osr2_pin	**O**	**1**	tx_ip_0_osr2
tx_ip_0_rstb_n_pin	**O**	**1**	tx_ip_0_rstb_n
tx_ip_0_sdo_pin	**O**	**0:23**	tx_ip_0_sdo
xps_gpio_0			
xps_gpio_0_CHAN_1_GPIO_I_pin	**I**	**0:0**	xps_gpio_0_GPIO_IO_I
xps_gpio_1			

xps_gpio_1_GPIO_IO_pin	**O**	**0:5**	xps_gpio_1_GPIO_IO_O
xps_iic_0			
xps_iic_0_Scl_pin	**IO**	**1**	xps_iic_0_Scl
xps_iic_0_Sda_pin	**IO**	**1**	xps_iic_0_Sda
xps_spi_0			
xps_spi_0_MISO_pin	**IO**	**1**	xps_spi_0_MISO
xps_spi_0_MOSI_pin	**IO**	**1**	xps_spi_0_MOSI
xps_spi_0_SCK_pin	**IO**	**1**	xps_spi_0_SCK
xps_spi_0_SS_pin	**IO**	**0:0**	xps_spi_0_SS
xps_spi_1			
xps_spi_1_MISO_pin	**IO**	**1**	xps_spi_1_MISO
xps_spi_1_MOSI_pin	**IO**	**1**	xps_spi_1_MOSI
xps_spi_1_SCK_pin	**IO**	**1**	xps_spi_1_SCK
xps_spi_1_SS_pin	**IO**	**0:1**	xps_spi_1_SS
xps_timebase_wdt_0			
xps_timebase_wdt_0_WDT_Reset_pin	**O**	**1**	xps_timebase_wdt_0_WDT_Reset
xps_uartlite_0			
xps_uartlite_0_RX_pin	**I**	**1**	xps_uartlite_0_RX
xps_uartlite_0_TX_pin	**O**	**1**	xps_uartlite_0_TX
xps_uartlite_1			
xps_uartlite_1_RX_pin	**I**	**1**	xps_uartlite_1_RX
xps_uartlite_1_TX_pin	**O**	**1**	xps_uartlite_1_TX

3 Microprocessor Control System Design

3.1 Processor

This is the processor of the system. It is implemented in FPGA and is tailored according to the requirement of the TT MK4 system.

MicroBlaze is an embedded soft RISC processor with 32-bit separated data and addresses bus. It can run at speeds up to 200MHz and can be extended with additional coprocessors. The processor is optimized for implementation in Xilinx FPGAs, and supports on-chip Block RAM and peripheral controller IP for interfacing custom devices. The Xilinx Embedded system development environment ISE and EDK are used for software and hardware co-development.

3.1.1 Port List

These are the ports listed in the MHS file. Please refer to the IP documentation for complete information about module ports.

Port List				
#	**NAME**	**DIR**	**[LSB:MSB]**	**SIGNAL**
0	MB_RESET	**I**	**1**	mb_reset
1	INTERRUPT	**I**	**1**	microblaze_0_INTERRUPT

3.1.2 Bus Interfaces

Bus Interfaces				
NAME	**TYPE**	**BUSSTD**	**BUS**	**Point 2 Point**
IXCL	**INITIATOR**	**XIL_MEMORY_CHANNEL**	microblaze_0_IXCL	DDR2_SDRAM
DXCL	**INITIATOR**	**XIL_MEMORY_CHANNEL**	microblaze_0_DXCL	DDR2_SDRAM
DPLB	**MASTER**	**PLBV46**	mb_plb	Ethernet_MAC
IPLB	**MASTER**	**PLBV46**	mb_plb	Ethernet_MAC
DLMB	**MASTER**	**LMB**	dlmb	dlmb_cntlr
ILMB	**MASTER**	**LMB**	ilmb	ilmb_cntlr
DEBUG	**TARGET**	**XIL_MBDEBUG2**	microblaze_0_mdm_bus	mdm_0

3.1.3 Parameters

These are the current parameter settings for this module.

Name	Value	Name	Value
C_FAMILY	spartan3adsp	C_ILL_OPCODE_EXCEPTION	1
C_INSTANCE	microblaze_0	C_INTERCONNECT	1
C_DCACHE_BASEADDR	0xC0000000	C_INTERRUPT_IS_EDGE	0
C_DCACHE_HIGHADDR	0xC3FFFFFF	C_IOPB_BUS_EXCEPTION	0
C_ICACHE_BASEADDR	0xC0000000	C_IPLB_BURST_EN	0
C_ICACHE_HIGHADDR	0xC3FFFFFF	C_IPLB_BUS_EXCEPTION	1
C_ADDR_TAG_BITS	17	C_IPLB_DWIDTH	32
C_ALLOW_DCACHE_WR	1	C_IPLB_NATIVE_DWIDTH	32
C_ALLOW_ICACHE_WR	1	C_IPLB_P2P	0
C_AREA_OPTIMIZED	0	C_I_LMB	1
C_CACHE_BYTE_SIZE	4096	C_I_OPB	1

C_DATA_SIZE	32	C_I_PLB	0
C_DCACHE_ADDR_TAG	17	C_MMU_DTLB_SIZE	4
C_DCACHE_ALWAYS_USED	1	C_MMU_ITLB_SIZE	2
C_DCACHE_BYTE_SIZE	4096	C_MMU_TLB_ACCESS	3
C_DCACHE_INTERFACE	0	C_MMU_ZONES	16
C_DCACHE_LINE_LEN	4	C_NUMBER_OF_PC_BRK	1
C_DCACHE_USE_FSL	1	C_NUMBER_OF_RD_ADDR_BRK	0
C_DCACHE_USE_WRITEBACK	1	C_NUMBER_OF_WR_ADDR_BRK	0
C_DEBUG_ENABLED	1	C_OPCODE_0x0_ILLEGAL	1
C_DIV_ZERO_EXCEPTION	0	C_PVR	2
C_DOPB_BUS_EXCEPTION	0	C_PVR_USER1	0x00
C_DPLB_BURST_EN	0	C_PVR_USER2	0x00000000
C_DPLB_BUS_EXCEPTION	1	C_RESET_MSR	0x00000000
C_DPLB_DWIDTH	32	C_SCO	0
C_DPLB_NATIVE_DWIDTH	32	C_UNALIGNED_EXCEPTIONS	1
C_DPLB_P2P	0	C_USE_BARREL	1
C_DYNAMIC_BUS_SIZING	1	C_USE_DCACHE	1
C_D_LMB	1	C_USE_DIV	1
C_D_OPB	1	C_USE_EXTENDED_FSL_INSTR	0
C_D_PLB	0	C_USE_EXT_BRK	0
C_EDGE_IS_POSITIVE	1	C_USE_EXT_NM_BRK	0
C_FPU_EXCEPTION	0	C_USE_FPU	1
C_FSL_DATA_SIZE	32	C_USE_HW_MUL	1
C_FSL_EXCEPTION	0	C_USE_ICACHE	1
C_FSL_LINKS	0	C_USE_INTERRUPT	0
C_ICACHE_ALWAYS_USED	1	C_USE_MMU	0
C_ICACHE_INTERFACE	0	C_USE_MSR_INSTR	1
C_ICACHE_LINE_LEN	4	C_USE_PCMP_INSTR	1
C_ICACHE_USE_FSL	1		

3.1.4 Memory Map

Memory Map				
D=DATA ADDRESSABLE I=INSTRUCTION ADDRESSABLE				
D	**I**	**BASE**	**HIGH**	**MODULE**
■		0x00000000	0x00003FFF	C_BASEADDR:C_HIGHADDR**dlmb_cntlr**
	■	0x00000000	0x00003FFF	C_BASEADDR:C_HIGHADDR**ilmb_cntlr**
■	■	0x80000000	0x8000FFFF	C_BASEADDR:C_HIGHADDR**Ethernet_MAC**
■	■	0x81000000	0x8100007F	C_BASEADDR:C_HIGHADDR**xps_uartlite_0**
■	■	0x81001000	0x8100107F	C_BASEADDR:C_HIGHADDR**xps_uartlite_1**
■	■	0x81400000	0x8140007F	C_BASEADDR:C_HIGHADDR**xps_spi_0**
■	■	0x81401000	0x8140107F	C_BASEADDR:C_HIGHADDR**xps_spi_1**
■	■	0x81800000	0x818001FF	C_BASEADDR:C_HIGHADDR**xps_iic_0**
■	■	0x81A00000	0x81A0000F	C_BASEADDR:C_HIGHADDR**xps_timebase_wdt_0**
■	■	0x81B00000	0x81B001FF	C_BASEADDR:C_HIGHADDR**xps_gpio_0**
■	■	0x81B01000	0x81B011FF	C_BASEADDR:C_HIGHADDR**xps_gpio_1**

■	■	0x81C00000	0x81C0001F	C_BASEADDR:C_HIGHADDR**xps_intc_0**
■	■	0x83C00000	0x83C0FFFF	C_BASEADDR:C_HIGHADDR**xps_timer_0**
■	■	0x84400000	0x8440FFFF	C_BASEADDR:C_HIGHADDR**mdm_0**
■	■	0xA0000000	0xA000FFFF	C_BASEADDR:C_HIGHADDR**tx_bram_reg20_0**
■	■	0xA1000000	0xA100FFFF	C_BASEADDR:C_HIGHADDR**rx_gain_reg24**
■	■	0xA2000000	0xA2001FFF	C_MEM0_BASEADDR:C_MEM0_HIGHADDR**tx_bram_reg20_0**
■	■	0xA2002000	0xA2003FFF	C_MEM1_BASEADDR:C_MEM1_HIGHADDR**tx_bram_reg20_0**
■	■	0xA2004000	0xA2005FFF	C_MEM2_BASEADDR:C_MEM2_HIGHADDR**tx_bram_reg20_0**
■	■	0xA2006000	0xA2007FFF	C_MEM3_BASEADDR:C_MEM3_HIGHADDR**tx_bram_reg20_0**
■	■	0xA2008000	0xA2009FFF	C_MEM4_BASEADDR:C_MEM4_HIGHADDR**tx_bram_reg20_0**
■	■	0xA200A000	0xA200BFFF	C_MEM5_BASEADDR:C_MEM5_HIGHADDR**tx_bram_reg20_0**
■	■	0xA200C000	0xA200DFFF	C_MEM6_BASEADDR:C_MEM6_HIGHADDR**tx_bram_reg20_0**
■	■	0xA200E000	0xA200FFFF	C_MEM7_BASEADDR:C_MEM7_HIGHADDR**tx_bram_reg20_0**
■	■	0xA2010000	0xA2011FFF	C_MEM8_BASEADDR:C_MEM8_HIGHADDR**tx_bram_reg20_0**
■	■	0xA2012000	0xA2013FFF	C_MEM9_BASEADDR:C_MEM9_HIGHADDR**tx_bram_reg20_0**
■	■	0xA2014000	0xA2015FFF	C_MEM10_BASEADDR:C_MEM10_HIGHADDR**tx_bram_reg20_0**
■	■	0xA2016000	0xA2017FFF	C_MEM11_BASEADDR:C_MEM11_HIGHADDR**tx_bram_reg20_0**
■	■	0xA2018000	0xA2019FFF	C_MEM12_BASEADDR:C_MEM12_HIGHADDR**tx_bram_reg20_0**
■	■	0xA201A000	0xA201BFFF	C_MEM13_BASEADDR:C_MEM13_HIGHADDR**tx_bram_reg20_0**
■	■	0xA201C000	0xA201DFFF	C_MEM14_BASEADDR:C_MEM14_HIGHADDR**tx_bram_reg20_0**
■	■	0xA201E000	0xA201FFFF	C_MEM15_BASEADDR:C_MEM15_HIGHADDR**tx_bram_reg20_0**
■	■	0xA2020000	0xA2021FFF	C_MEM16_BASEADDR:C_MEM16_HIGHADDR**tx_bram_reg20_0**
■	■	0xA2022000	0xA2023FFF	C_MEM17_BASEADDR:C_MEM17_HIGHADDR**tx_bram_reg20_0**
■	■	0xA2024000	0xA2025FFF	C_MEM18_BASEADDR:C_MEM18_HIGHADDR**tx_bram_reg20_0**
■	■	0xA2026000	0xA2027FFF	C_MEM19_BASEADDR:C_MEM19_HIGHADDR**tx_bram_reg20_0**
■	■	0xA2028000	0xA2029FFF	C_MEM20_BASEADDR:C_MEM20_HIGHADDR**tx_bram_reg20_0**
■	■	0xA202A000	0xA202BFFF	C_MEM21_BASEADDR:C_MEM21_HIGHADDR**tx_bram_reg20_0**
■	■	0xA202C000	0xA202DFFF	C_MEM22_BASEADDR:C_MEM22_HIGHADDR**tx_bram_reg20_0**
■	■	0xA202E000	0xA202FFFF	C_MEM23_BASEADDR:C_MEM23_HIGHADDR**tx_bram_reg20_0**
■	■	0xA3000000	0xA33FFFFF	C_MEM0_BASEADDR:C_MEM0_HIGHADDR**SRAM_1**
■	■	0xA3400000	0xA37FFFFF	C_MEM0_BASEADDR:C_MEM0_HIGHADDR**SRAM_2**
■	■	0xA3800000	0xA3BFFFFF	C_MEM0_BASEADDR:C_MEM0_HIGHADDR**SRAM_3**
■	■	0xC0000000	0xC7FFFFFF	C_MPMC_BASEADDR:C_MPMC_HIGHADDR**DDR2_SDRAM**

3.1.5 Post Synthesis Device Utilization

	Used	**Total**	**Percentage**
Number of Slices	2539	16640	15%
Number of Slice Flip Flops	2518	33280	7%
Number of 4 input LUTs	4042	33280	12%
Number used as logic	3612		
Number used as Shift registers	46		
Number used as RAMs	384		
Number of IOs	2296		
Number of bonded IOBs	0	519	0%
Number of BRAMs	6	84	7%
Number of DSP48s	8	84	9%

3.1.6 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: 12.812ns (Maximum Frequency: 78.052MHz)

Minimum input arrival time before clock: 8.950ns

Maximum output required time after clock: 8.947ns

Maximum combinational path delay: 3.640ns

3.2 Local Memory Data Bus

dlmb Local Memory Bus (LMB) 1.0
'The LMB is a fast, local bus for connecting MicroBlaze I and D ports to peripherals and BRAM'

3.2.1 Port List

These are the ports listed in the MHS file. Please refer to the IP documentation for complete information about module ports.

Port List				
#	**NAME**	**DIR**	**[LSB:MSB]**	**SIGNAL**
0	LMB_Clk	**I**	**1**	clk_62_5000MHz
1	SYS_Rst	**I**	**1**	sys_bus_reset

3.2.2 Bus Connections

Bus Connections		
INSTANCE	**INTERFACE TYPE**	**INTERFACE NAME**
microblaze_0	MASTER	DLMB
dlmb_cntlr	SLAVE	SLMB

3.2.3 Parameters

These are the current parameter settings for this module.

Name	Value
C_EXT_RESET_HIGH	1
C_LMB_AWIDTH	32
C_LMB_DWIDTH	32
C_LMB_NUM_SLAVES	1

3.2.4 Post Synthesis Device Utilization

	Used	Total	Percentage
Number of Slices	1	16640	0%
Number of Slice Flip Flops	1	33280	0%
Number of 4 input LUTs	1	33280	0%
Number of IOs	211		
Number of bonded IOBs	0	519	0%

3.2.5 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: 3.781ns (Maximum Frequency: 264.480MHz)

Minimum input arrival time before clock: 0.869ns

Maximum output required time after clock: 0.591ns

Maximum combinational path delay: 0.000ns

3.3 Local Memory Instruction Bus

dlmb   Local Memory Bus (LMB) 1.0
'The LMB is a fast, local bus for connecting MicroBlaze I and D ports to peripherals and BRAM'

3.3.1.1 Port List

These are the ports listed in the MHS file. Please refer to the IP documentation for complete information about module ports.

Port List				
#	**NAME**	**DIR**	**[LSB:MSB]**	**SIGNAL**
0	LMB_Clk	**I**	**1**	clk_62_5000MHz
1	SYS_Rst	**I**	**1**	sys_bus_reset

3.3.2 Bus Connections

Bus Connections		
INSTANCE	**INTERFACE TYPE**	**INTERFACE NAME**
microblaze_0	MASTER	ILMB
dlmb_cntlr	SLAVE	SLMB

3.3.3 Parameters

These are the current parameter settings for this module.

Name	Value
C_EXT_RESET_HIGH	1
C_LMB_AWIDTH	32
C_LMB_DWIDTH	32
C_LMB_NUM_SLAVES	1

3.3.4 Post Synthesis Device Utilization

	Used	Total	Percentage
Number of Slices	1	16640	0%

Number of Slice Flip Flops	1	33280	0%
Number of 4 input LUTs	1	33280	0%
Number of IOs	211		
Number of bonded IOBs	0	519	0%

3.3.5 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: 3.781ns (Maximum Frequency: 264.480MHz)

Minimum input arrival time before clock: 0.869ns

Maximum output required time after clock: 0.591ns

Maximum combinational path delay: 0.000ns

3.4 dlmb_cntlr LMB BRAM Controller

dlmb_cntlr LMB BRAM Controller
Local Memory Bus (LMB) Block RAM (BRAM) Interface Controller connects to an lmb bus.

3.4.1 Bus Interfaces

Bus Interfaces				
NAME	**TYPE**	**BUSSTD**	**BUS**	**Point 2 Point**
BRAM_PORT	INITIATOR	XIL_BRAM	dlmb_port	lmb_bram
SPLB	SLAVE	LMB	dlmb	microblaze_0

3.4.2 Parameters

These are the current parameter settings for this module.

Name	Value
C_BASEADDR ***LMB BRAM Base Address***	0x00000000
C_HIGHADDR ***LMB BRAM High Address***	0x00003FFF
C_LMB_AWIDTH ***LMB Address Bus Width***	32
C_LMB_DWIDTH ***LMB Data Bus Width***	32
C_MASK ***LMB Address Decode Mask***	0x00800000

3.4.3 Post Synthesis Device Utilization

	Used	Total	Percentage
Number of Slices	3	16640	0%

Number of Slice Flip Flops	2	33280	0%
Number of 4 input LUTs	6	33280	0%
Number of IOs	209		
Number of bonded IOBs	0	519	0%

3.4.4 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: No path found

Minimum input arrival time before clock: 1.320ns

Maximum output required time after clock: 1.802ns

Maximum combinational path delay: 0.791ns

3.5 ilmb_cntlr LMB BRAM Controller

ilmb_cntlr LMB BRAM Controller
Local Memory Bus (LMB) Block RAM (BRAM) Interface Controller connects to an lmb bus

3.5.1 Bus Interfaces

Bus Interfaces				
NAME	**TYPE**	**BUSSTD**	**BUS**	**Point 2 Point**
BRAM_PORT	INITIATOR	XIL_BRAM	ilmb_port	lmb_bram
SPLB	SLAVE	LMB	ilmb	microblaze_0

3.5.2 Parameters

These are the current parameter settings for this module.

Name	**Value**
C_BASEADDR ***LMB BRAM Base Address***	0x00000000
C_HIGHADDR ***LMB BRAM High Address***	0x00003FFF
C_LMB_AWIDTH ***LMB Address Bus Width***	32
C_LMB_DWIDTH ***LMB Data Bus Width***	32
C_MASK ***LMB Address Decode Mask***	0x00800000

3.5.3 Post Synthesis Device Utilization

	Used	**Total**	**Percentage**
Number of Slices	3	16640	0%

Number of Slice Flip Flops	2	33280	0%
Number of 4 input LUTs	6	33280	0%
Number of IOs	209		
Number of bonded IOBs	0	519	0%

3.5.4 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: No path found

Minimum input arrival time before clock: 1.320ns

Maximum output required time after clock: 1.802ns

Maximum combinational path delay: 0.791ns

3.6 Blcok RAM (BRAM) Block

Lmb_bram

The BRAM Block is a configurable memory module that attaches to a variety of BRAM Interface Controllers. This is the processor local memory. FPGA on-chip Block RAM memory resource is utilized. The BRAM is interfaced to processor using LMB BRAM Controller (dlmb_cntlr and ilmb_cntlr for data and instruction respectively). These two controllers are connected to processor via fast speed data/Instruction local memory bus (dlmb and ilmb).

3.6.1 Bus Interfaces

Bus Interfaces				
NAME	**TARGET**	**BUSSTD**	**BUS**	**Point 2 Point**
PORTA	**TARGET**	**XIL_BRAM**	ilmb_port	ilmb_cntlr
PORTB	**TARGET**	**XIL_BRAM**	dlmb_port	dlmb_cntlr

3.6.2 Parameters

These are the current parameter settings for this module.

Name	**Value**	**Name**	**Value**
C_FAMILY	spartan3adsp	C_PORT_AWIDTH	32
C_MEMSIZE	0x4000	C_PORT_DWIDTH	32
C_NUM_WE	4		

3.6.3 Post Synthesis Device Utilization

	Used	**Total**	**Percentage**
Number of Slices	0	16640	0%
Number of IOs	206		

Number of bonded IOBs	0	519	0%
Number of BRAMs	8	84	9%

3.6.4 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: No path found

Minimum input arrival time before clock: 0.750ns

Maximum output required time after clock: 2.800ns

Maximum combinational path delay: No path found

3.7 Processor Local Bus

mb_plb   Processor Local Bus (PLB) 4.6
'Xilinx 64-bit Processor Local Bus (PLB) consists of a bus control unit, a watchdog timer, and separate address, write, and read data path units with a a three-cycle only arbitration feature'.

3.7.1 Port List

These are the ports listed in the MHS file. Please refer to the IP documentation for complete information about module ports.

Port List				
#	**NAME**	**DIR**	**[LSB:MSB]**	**SIGNAL**
0	PLB_Clk	**I**	**1**	clk_62_5000MHz
1	SYS_Rst	**I**	**1**	sys_bus_reset
2	Bus_Error_Det	**O**	**1**	mb_plb_Bus_Error_Det

3.7.2 Bus Connections

Bus Connections		
INSTANCE	**INTERFACE TYPE**	**INTERFACE NAME**
microblaze_0	MASTER	DPLB
microblaze_0	MASTER	IPLB
Ethernet_MAC	SLAVE	SPLB
mdm_0	SLAVE	SPLB
xps_uartlite_0	SLAVE	SPLB
xps_uartlite_1	SLAVE	SPLB
xps_timebase_wdt_0	SLAVE	SPLB
xps_gpio_0	SLAVE	SPLB
xps_gpio_1	SLAVE	SPLB
SRAM_1	SLAVE	SPLB

SRAM_2	SLAVE	SPLB
SRAM_3	SLAVE	SPLB
rx_gain_reg24	SLAVE	SPLB
tx_bram_reg20_0	SLAVE	SPLB
xps_iic_0	SLAVE	SPLB
xps_spi_0	SLAVE	SPLB
xps_spi_1	SLAVE	SPLB
xps_intc_0	SLAVE	SPLB
xps_timer_0	SLAVE	SPLB

3.7.3 Parameters

These are the current parameter settings for this module.

Name	Value	Name	Value
C_FAMILY	spartan3adsp	C_IRQ_ACTIVE	1
C_BASEADDR	0b1111111111	C_NUM_CLK_PLB2OPB_REARB	5
C_HIGHADDR	0b0000000000	C_P2P	0
C_ADDR_PIPELINING_TYPE	1	C_PLBV46_AWIDTH	32
C_ARB_TYPE	0	C_PLBV46_DWIDTH	64
C_DCR_AWIDTH	10	C_PLBV46_MID_WIDTH	1
C_DCR_DWIDTH	32	C_PLBV46_NUM_MASTERS	2
C_DCR_INTFCE	0	C_PLBV46_NUM_SLAVES	17
C_EXT_RESET_HIGH	1		

3.7.4 Post Synthesis Device Utilization

	Used	Total	Percentage
Number of Slices	360	16640	2%
Number of Slice Flip Flops	165	33280	0%
Number of 4 input LUTs	597	33280	1%
Number of IOs	1630		
Number of bonded IOBs	0	519	0%

3.7.5 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: 6.514ns (Maximum Frequency: 153.515MHz)

Minimum input arrival time before clock: 6.474ns

Maximum output required time after clock: 4.638ns

Maximum combinational path delay: 3.627ns

3.8 Multi-Port Memory Controller (DDR/DDR2/SDRAM)

DDR2_SDRAM Multi-Port Memory Controller(DDR/DDR2/SDRAM) Multi-port memory controller.

The DDR2 SRAM is used as run-time memory for running firmware application. It is connected to the processor via Xilinx Cashlink bus (DXCL and IXCL). The DDR2 SRAM is configured to be cashable for the whole memory range.

3.8.1 Port List

Port List				
#	NAME	DIR	[LSB: MSB]	SIGNAL
0	MPMC_Clk0	I	1	clk_125_0000MHzDCM0
1	MPMC_Clk90	I	1	clk_125_0000MHz90DCM0
2	MPMC_Rst	I	1	sys_periph_reset
3	DDR2_Clk	O	1	fpga_0_DDR2_SDRAM_DDR2_Clk_pin
4	DDR2_Clk_n	O	1	fpga_0_DDR2_SDRAM_DDR2_Clk_n_pin
5	DDR2_CE	O	1	fpga_0_DDR2_SDRAM_DDR2_CE_pin
6	DDR2_CS_n	O	1	fpga_0_DDR2_SDRAM_DDR2_CS_n_pin
7	DDR2_ODT	O	1	fpga_0_DDR2_SDRAM_DDR2_ODT_pin
8	DDR2_RAS_n	O	1	fpga_0_DDR2_SDRAM_DDR2_RAS_n_pin
9	DDR2_CAS_n	O	1	fpga_0_DDR2_SDRAM_DDR2_CAS_n_pin
10	DDR2_WE_n	O	1	fpga_0_DDR2_SDRAM_DDR2_WE_n_pin
11	DDR2_BankAddr	O	1	fpga_0_DDR2_SDRAM_DDR2_BankAddr_pin
12	DDR2_Addr	O	1	fpga_0_DDR2_SDRAM_DDR2_Addr_pin
13	DDR2_DQ	IO	1	fpga_0_DDR2_SDRAM_DDR2_DQ_pin
14	DDR2_DM	O	1	fpga_0_DDR2_SDRAM_DDR2_DM_pin
15	DDR2_DQS	IO	1	fpga_0_DDR2_SDRAM_DDR2_DQS_pin
16	DDR2_DQS_n	IO	1	fpga_0_DDR2_SDRAM_DDR2_DQS_n_pin
17	DDR2_DQS_Div_O	O	1	fpga_0_DDR2_SDRAM_DDR2_DQS_Div_O_pin
18	DDR2_DQS_Div_I	I	1	fpga_0_DDR2_SDRAM_DDR2_DQS_Div

3.8.2 Bus Interfaces

Bus Interfaces				
NAME	TARGET	BUSSTD	BUS	Point 2 Point
XCL0	TARGET	XIL_MEMORY_CHANNEL	microblaze_0_IXCL	microblaze_0
XCL0_B	TARGET	XIL_MEMORY_CHANNEL	microblaze_0_DXCL	microblaze_0

3.8.3 Parameters

These are the current parameter settings for this module.

Name	Value
C_FAMILY	spartan3adsp
C_MPMC_BASEADDR ***Base Address***	0xC0000000
C_MPMC_CTRL_BASEADDR ***Control Base Address***	0xFFFFFFFF
C_MPMC_CTRL_HIGHADDR ***Control High Address***	0x00000000
C_MPMC_HIGHADDR ***High Address***	0xC7FFFFFF
C_MPMC_SW_BASEADDR ***MPMC PIMs Software Base Address***	0xFFFFFFFF
C_MPMC_SW_HIGHADDR	0x00000000

MPMC PIMs Software High Address	
C_PIM0_BASEADDR	0xFFFFFFFF
C_PIM0_HIGHADDR	0x00000000
C_PIM1_BASEADDR	0xFFFFFFFF
C_PIM1_HIGHADDR	0x00000000
C_PIM2_BASEADDR	0xFFFFFFFF
C_PIM2_HIGHADDR	0x00000000
C_PIM3_BASEADDR	0xFFFFFFFF
C_PIM3_HIGHADDR	0x00000000
C_PIM4_BASEADDR	0xFFFFFFFF
C_PIM4_HIGHADDR	0x00000000
C_PIM5_BASEADDR	0xFFFFFFFF
C_PIM5_HIGHADDR	0x00000000
C_PIM6_BASEADDR	0xFFFFFFFF
C_PIM6_HIGHADDR	0x00000000
C_PIM7_BASEADDR	0xFFFFFFFF
C_PIM7_HIGHADDR	0x00000000
C_SDMA_CTRL0_BASEADDR	0xFFFFFFFF
C_SDMA_CTRL0_HIGHADDR	0x00000000
C_SDMA_CTRL1_BASEADDR	0xFFFFFFFF
C_SDMA_CTRL1_HIGHADDR	0x00000000
C_SDMA_CTRL2_BASEADDR	0xFFFFFFFF
C_SDMA_CTRL2_HIGHADDR	0x00000000
C_SDMA_CTRL3_BASEADDR	0xFFFFFFFF
C_SDMA_CTRL3_HIGHADDR	0x00000000
C_SDMA_CTRL4_BASEADDR	0xFFFFFFFF
C_SDMA_CTRL4_HIGHADDR	0x00000000
C_SDMA_CTRL5_BASEADDR	0xFFFFFFFF
C_SDMA_CTRL5_HIGHADDR	0x00000000
C_SDMA_CTRL6_BASEADDR	0xFFFFFFFF
C_SDMA_CTRL6_HIGHADDR	0x00000000
C_SDMA_CTRL7_BASEADDR	0xFFFFFFFF
C_SDMA_CTRL7_HIGHADDR	0x00000000
C_SDMA_CTRL_BASEADDR ***SDMA Register Base Address***	0xFFFFFFFF
C_SDMA_CTRL_HIGHADDR ***SDMA Ctrl High Address***	0x00000000
C_ALL_PIMS_SHARE_ADDRESSES ***Use Common Base Address***	1
C_ARB0_ALGO ***<qt><b>Select Arbitration Algorithm</b></qt>***	ROUND_ROBIN
C_ARB0_NUM_SLOTS ***Number of Time Slots***	8
C_ARB0_SLOT0 ***Time Slot 0***	01234567
C_ARB0_SLOT1 ***Time Slot 1***	12345670
C_ARB0_SLOT10 ***Time Slot 10***	23456701
C_ARB0_SLOT11 ***Time Slot 11***	34567012
C_ARB0_SLOT12 ***Time Slot 12***	45670123
C_ARB0_SLOT13 ***Time Slot 13***	56701234
C_ARB0_SLOT14	67012345

Time Slot 14	
C_ARB0_SLOT15 ***Time Slot 15***	70123456
C_ARB0_SLOT2 ***Time Slot 2***	23456701
C_ARB0_SLOT3 ***Time Slot 3***	34567012
C_ARB0_SLOT4 ***Time Slot 4***	45670123
C_ARB0_SLOT5 ***Time Slot 5***	56701234
C_ARB0_SLOT6 ***Time Slot 6***	67012345
C_ARB0_SLOT7 ***Time Slot 7***	70123456
C_ARB0_SLOT8 ***Time Slot 8***	01234567
C_ARB0_SLOT9 ***Time Slot 9***	12345670
C_ARB_BRAM_INIT_00	0b00000000111111111111111111111111000000001111111111111111111111110000000011111111111111111111110000000011111111111111111100000000111111111110100010000110000000011111111111100100001101000000000111111111111000011010001000000001111111111110110100010000
C_ARB_BRAM_INIT_01	0b000000001111111111111111111111110000000011111111111111111111111100000000111111111111111111111100000000111111111111111111000000001111111111111111111111110000000011111111111111111111111100000000111111111111111111111111000000111111111111111111111111
C_ARB_BRAM_INIT_02	0b000000001111111111111111111111110000000011111111111111111111111100000000111111111111111111111100000000111111111111111111000000001111111111110110100010000000000011111111111101101000100000000000011111111111101101000100000000000111111111111011010001000
C_ARB_BRAM_INIT_03	0b000000001111111111111111111111110000000011111111111111111111111100000000111111111111111111111100000000111111111111111111000000001111111111111111111111110000000011111111111111111111111100000000111111111111111111111111000000111111111111111111111111
C_ARB_BRAM_INIT_04	0b00000000111111111111111111111111000000001111111111111111111111110000000011111111111111111111110000000011111111111111111100000000111111111110100010000110000000011111111111100100001101000000000111111111111000011010001000000001111111111110110100010000
C_ARB_BRAM_INIT_05	0b0000000011111111111111111111111100000000111111111111111111111111000000001111111111111111111111000000001111111111

	11111111111111110000000011111111111111 11111111110000000011111111111111111111 11110000000011111111111111111111111100 000000111111111111111111111111
C_ARB_BRAM_INIT_06	0b000000001111111111111111111111110000 00001111111111111111111111110000000011 11111111111111111111110000000011111111 11111111111111110000000011111111111101 10100010000000000011111111111101101000 10000000000011111111111101101000100000 000000111111111111011010001000
C_ARB_BRAM_INIT_07	0b000000001111111111111111111111110000 00001111111111111111111111110000000011 11111111111111111111110000000011111111 11111111111111110000000011111111111111 11111111110000000011111111111111111111 11110000000011111111111111111111111100 000000111111111111111111111111
C_ARB_PIPELINE ***Turn on Arbiter Pipeline***	1
C_ARB_USE_DEFAULT	0
C_B16_REPEAT_CNT	0
C_B32_REPEAT_CNT	0
C_B64_REPEAT_CNT	0
C_BASEADDR_CTRL0	0x000
C_BASEADDR_CTRL1	0x00E
C_BASEADDR_CTRL10	0x09E
C_BASEADDR_CTRL11	0x0A6
C_BASEADDR_CTRL12	0x0AE
C_BASEADDR_CTRL13	0x0B6
C_BASEADDR_CTRL14	0x0BE
C_BASEADDR_CTRL15	0x0D1
C_BASEADDR_CTRL2	0x018
C_BASEADDR_CTRL3	0x026
C_BASEADDR_CTRL4	0x030
C_BASEADDR_CTRL5	0x03E
C_BASEADDR_CTRL6	0x048
C_BASEADDR_CTRL7	0x05C
C_BASEADDR_CTRL8	0x06B
C_BASEADDR_CTRL9	0x087
C_CTRL_AP_COL_CNT_ENABLE_INDEX	0
C_CTRL_AP_COL_CNT_LOAD_INDEX	0
C_CTRL_AP_COL_DELAY	0
C_CTRL_AP_OTF_ADDR12_INDEX	0
C_CTRL_AP_PIPELINE1_CE_DELAY	0
C_CTRL_AP_PI_ADDR_CE_DELAY	0
C_CTRL_AP_PORT_SELECT_DELAY	0
C_CTRL_AP_PRECHARGE_ADDR10_INDEX	0
C_CTRL_AP_ROW_COL_SEL_INDEX	0
C_CTRL_ARB_RDMODWR_DELAY	0
C_CTRL_BRAM_INITP_00	0x111111111111000111111011111111111111 1001111110111111111111111001
C_CTRL_BRAM_INITP_01	0x111000000000000000001111111111111111 1111111111000000000111111011
C_CTRL_BRAM_INITP_02	0x111000000000000000000000000000000001 1111111111111111111111111111

C_CTRL_BRAM_INITP_03	0x0000000000000000000000000000000000000000000000000111111111111111
C_CTRL_BRAM_INITP_04	0x0000000000000000000000000000000000000000000000000000000000000000
C_CTRL_BRAM_INITP_05	0x0000000000000000000000000000000000000000000000000000000000000000
C_CTRL_BRAM_INITP_06	0x0000000000000000000000000000000000000000000000000000000000000000
C_CTRL_BRAM_INITP_07	0x0000000000000000000000000000000000000000000000000000000000000000
C_CTRL_BRAM_INIT_00	0x000002FC000002FC000002FC0000013C000019240000803C000082FC000082F8
C_CTRL_BRAM_INIT_01	0x000082FC000082F8000002FC000002FC000002FC000042E8000002FC000002FD
C_CTRL_BRAM_INIT_02	0x000002FC000002FC000002FC000042E8000002FC000002FD000016F4000082FC
C_CTRL_BRAM_INIT_03	0x000002FC000002FC000002FC0000013C000019240000803C000082FC000082F8
C_CTRL_BRAM_INIT_04	0x000082FC000082F8000002FC000002FC000002FC000042E8000002FC000002FD
C_CTRL_BRAM_INIT_05	0x000002FC000002FC000002FC000042E8000002FC000002FD000016F4000082FC
C_CTRL_BRAM_INIT_06	0x000002FC000002FC000002FC0000093C000019240000803C000082FC000082F8
C_CTRL_BRAM_INIT_07	0x000082FC000082F8000002FC000002FC000002FC000042E8000002FC000002FD
C_CTRL_BRAM_INIT_08	0x000002FC000002FC000002FC000042E8000002FC000006FD000016F4000082FC
C_CTRL_BRAM_INIT_09	0x000029240000093C000029240000093C000019240000803C000082FC000082F8
C_CTRL_BRAM_INIT_0A	0x000002FC000002FD000002FC000002FC000002FC0000093C000029240000093C
C_CTRL_BRAM_INIT_0B	0x000016F4000082FC000082FC000082F8000002FC000002FC000002FC000042E8
C_CTRL_BRAM_INIT_0C	0x000042E8000006FC000026F5000006FC000026F4000006FC000026F4000006FC
C_CTRL_BRAM_INIT_0D	0x0000093C000019240000803C000082FC000082F8000002FC000002FC000002FC
C_CTRL_BRAM_INIT_0E	0x0000093C000029240000093C000029240000093C000029240000093C00002924
C_CTRL_BRAM_INIT_0F	0x000002FC000002FC0000093C000029240000093C000029240000093C00002924
C_CTRL_BRAM_INIT_10	0x000082F8000002FC000002FC000002FC000042E8000002FC000002FD000002FC
C_CTRL_BRAM_INIT_11	0x000006FC000026F4000006FC000026F4000006FC000016F4000082FC000082FC
C_CTRL_BRAM_INIT_12	0x000006FC000026F4000006FC000026F4000006FC000026F4000006FC000026F4
C_CTRL_BRAM_INIT_13	0x000002FC000002FC000002FC000002FC000002FC000042E8000006FC000026F5
C_CTRL_BRAM_INIT_14	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_15	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_16	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_17	0x000002FC000042E8000002FC000002FC000

	002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_18	0x000002FC000002FC000002FC000002FC000002FC000002F0000002FC000002FC
C_CTRL_BRAM_INIT_19	0x000002FC000002FC000002FC000002FC000002FD000002FC000002FC000002FC
C_CTRL_BRAM_INIT_1A	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_1B	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_1C	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_1D	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_1E	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_1F	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_20	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_21	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_22	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_23	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_24	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_25	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_26	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_27	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_28	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_29	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_2A	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_2B	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_2C	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_2D	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_2E	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_2F	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_30	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_31	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_32	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_33	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC

C_CTRL_BRAM_INIT_34	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_35	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_36	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_37	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_38	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_39	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_3A	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_3B	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_3C	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_3D	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_3E	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_INIT_3F	0x000002FC000002FC000002FC000002FC000002FC000002FC000002FC000002FC
C_CTRL_BRAM_SRVAL	0x0000002FC
C_CTRL_COMPLETE_INDEX	0
C_CTRL_DFI_CAS_N_0_INDEX	0
C_CTRL_DFI_CAS_N_1_INDEX	0
C_CTRL_DFI_RAS_N_0_INDEX	0
C_CTRL_DFI_RAS_N_1_INDEX	0
C_CTRL_DFI_RDDATA_EN_INDEX	0
C_CTRL_DFI_WE_N_0_INDEX	0
C_CTRL_DFI_WE_N_1_INDEX	0
C_CTRL_DFI_WRDATA_EN_INDEX	0
C_CTRL_DP_LOAD_RDWDADDR_DELAY	0
C_CTRL_DP_RDFIFO_PUSH_INDEX	0
C_CTRL_DP_RDFIFO_WHICHPORT_DELAY	0
C_CTRL_DP_SIZE_DELAY	0
C_CTRL_DP_WRFIFO_POP_INDEX	0
C_CTRL_DP_WRFIFO_WHICHPORT_DELAY	0
C_CTRL_IS_WRITE_INDEX	0
C_CTRL_PHYIF_CAS_N_INDEX	0
C_CTRL_PHYIF_DQS_O_INDEX	0
C_CTRL_PHYIF_DUMMYREADSTART_DELAY	0
C_CTRL_PHYIF_FORCE_DM_INDEX	0
C_CTRL_PHYIF_RAS_N_INDEX	0
C_CTRL_PHYIF_WE_N_INDEX	0
C_CTRL_Q0_DELAY	0
C_CTRL_Q10_DELAY	0
C_CTRL_Q11_DELAY	0
C_CTRL_Q12_DELAY	0
C_CTRL_Q13_DELAY	0
C_CTRL_Q14_DELAY	0
C_CTRL_Q15_DELAY	0
C_CTRL_Q16_DELAY	0

C_CTRL_Q17_DELAY	0
C_CTRL_Q18_DELAY	0
C_CTRL_Q19_DELAY	0
C_CTRL_Q1_DELAY	0
C_CTRL_Q20_DELAY	0
C_CTRL_Q21_DELAY	0
C_CTRL_Q22_DELAY	0
C_CTRL_Q23_DELAY	0
C_CTRL_Q24_DELAY	0
C_CTRL_Q25_DELAY	0
C_CTRL_Q26_DELAY	0
C_CTRL_Q27_DELAY	0
C_CTRL_Q28_DELAY	0
C_CTRL_Q29_DELAY	0
C_CTRL_Q2_DELAY	0
C_CTRL_Q30_DELAY	0
C_CTRL_Q31_DELAY	0
C_CTRL_Q32_DELAY	0
C_CTRL_Q33_DELAY	0
C_CTRL_Q34_DELAY	0
C_CTRL_Q35_DELAY	0
C_CTRL_Q3_DELAY	0
C_CTRL_Q4_DELAY	0
C_CTRL_Q5_DELAY	0
C_CTRL_Q6_DELAY	0
C_CTRL_Q7_DELAY	0
C_CTRL_Q8_DELAY	0
C_CTRL_Q9_DELAY	0
C_CTRL_REPEAT4_INDEX	0
C_CTRL_RMW_INDEX	0
C_CTRL_SKIP_0_INDEX	0
C_CTRL_SKIP_1_INDEX	0
C_CTRL_SKIP_2_INDEX	0
C_DDR2_DQSN_ENABLE ***Enable DQSN in DDR2***	1
C_DEBUG_REG_ENABLE ***Enable Debug Registers***	0
C_DEVICE	3sd1800a
C_ECC_DATA_WIDTH ***Data Width***	0
C_ECC_DEC_THRESHOLD ***<qt>DEC Threshold</qt>***	1
C_ECC_DEFAULT_ON ***ECC Default is On***	1
C_ECC_DM_WIDTH	0
C_ECC_DQS_WIDTH	0
C_ECC_PEC_THRESHOLD ***<qt>PEC Threshold</qt>***	1
C_ECC_SEC_THRESHOLD ***SEC Threshold***	1
C_HIGHADDR_CTRL0	0x00D
C_HIGHADDR_CTRL1	0x017
C_HIGHADDR_CTRL10	0x0A5
C_HIGHADDR_CTRL11	0x0AD
C_HIGHADDR_CTRL12	0x0B5
C_HIGHADDR_CTRL13	0x0BD

C_HIGHADDR_CTRL14	0x0D0
C_HIGHADDR_CTRL15	0x0D8
C_HIGHADDR_CTRL2	0x025
C_HIGHADDR_CTRL3	0x02F
C_HIGHADDR_CTRL4	0x03D
C_HIGHADDR_CTRL5	0x047
C_HIGHADDR_CTRL6	0x05B
C_HIGHADDR_CTRL7	0x06A
C_HIGHADDR_CTRL8	0x086
C_HIGHADDR_CTRL9	0x09D
C_IDELAYCTRL_LOC ***IDELAYCTRL Constraint Locations (Hyphen separated)***	NOT_SET
C_IDELAY_CLK_FREQ	DEFAULT
C_INCLUDE_ECC_SUPPORT ***Enable ECC***	0
C_INCLUDE_ECC_TEST ***Include ECC Test***	0
C_IODELAY_GRP ***IODELAY Grouping***	DDR2_SDRAM
C_MAX_REQ_ALLOWED ***Number of Requests MPMC can Queue per Port***	1
C_MCB_DQ0_TAP_DELAY_VAL	0
C_MCB_DQ10_TAP_DELAY_VAL	0
C_MCB_DQ11_TAP_DELAY_VAL	0
C_MCB_DQ12_TAP_DELAY_VAL	0
C_MCB_DQ13_TAP_DELAY_VAL	0
C_MCB_DQ14_TAP_DELAY_VAL	0
C_MCB_DQ15_TAP_DELAY_VAL	0
C_MCB_DQ1_TAP_DELAY_VAL	0
C_MCB_DQ2_TAP_DELAY_VAL	0
C_MCB_DQ3_TAP_DELAY_VAL	0
C_MCB_DQ4_TAP_DELAY_VAL	0
C_MCB_DQ5_TAP_DELAY_VAL	0
C_MCB_DQ6_TAP_DELAY_VAL	0
C_MCB_DQ7_TAP_DELAY_VAL	0
C_MCB_DQ8_TAP_DELAY_VAL	0
C_MCB_DQ9_TAP_DELAY_VAL	0
C_MCB_LDQSN_TAP_DELAY_VAL	0
C_MCB_LDQSP_TAP_DELAY_VAL	0
C_MCB_LOC ***MCB Location***	NOT_SET
C_MCB_UDQSN_TAP_DELAY_VAL	0
C_MCB_UDQSP_TAP_DELAY_VAL	0
C_MCB_USE_EXTERNAL_BUFPLL	0
C_MEM_ADDR_ORDER	BANK_ROW_COLUMN
C_MEM_ADDR_WIDTH ***Memory Address Width***	13
C_MEM_AUTO_SR ***Auto Self Refresh***	ENABLED
C_MEM_BANKADDR_WIDTH ***<qt>Memory Bank Address Width</qt>***	2
C_MEM_BITS_DATA_PER_DQS	8
C_MEM_CALIBRATION_BYPASS	NO
C_MEM_CALIBRATION_DELAY	HALF
C_MEM_CALIBRATION_MODE	1

C_MEM_CALIBRATION_SOFT_IP	FALSE
C_MEM_CAL_WIDTH	DEFAULT
C_MEM_CAS_LATENCY	3
C_MEM_CAS_WR_LATENCY ***CAS Write Latency***	5
C_MEM_CE_WIDTH ***CE Width***	1
C_MEM_CHECK_MAX_INDELAY	0
C_MEM_CHECK_MAX_TAP_REG	0
C_MEM_CLK_WIDTH ***Clock Width***	2
C_MEM_CS_N_WIDTH ***CSn Width***	1
C_MEM_DATA_WIDTH ***Memory Data Width***	32
C_MEM_DM_WIDTH ***Memory DM Width***	4
C_MEM_DQS_IO_COL	0x000000000000000000
C_MEM_DQS_LOC_COL0 ***DQS groups in column # 0***	0x000000000000000000000000000000000000
C_MEM_DQS_LOC_COL1 ***DQS groups in column # 1***	0x000000000000000000000000000000000000
C_MEM_DQS_LOC_COL2 ***DQS groups in column # 2***	0x000000000000000000000000000000000000
C_MEM_DQS_LOC_COL3 ***DQS groups in column # 3***	0x000000000000000000000000000000000000
C_MEM_DQS_WIDTH ***Memory DQS Width***	4
C_MEM_DQ_IO_MS	0x000000000000000000
C_MEM_DYNAMIC_WRITE_ODT ***Dynamic Write ODT Setting***	OFF
C_MEM_HIGH_TEMP_SR ***High Temp Self Refresh***	NORMAL
C_MEM_IBUF_LPWR_MODE	DEFAULT
C_MEM_INCDEC_THRESHOLD	0x02
C_MEM_IODELAY_HP_MODE	DEFAULT
C_MEM_NDQS_COL0 ***Number of DQS groups in I/O column # 0***	0
C_MEM_NDQS_COL1 ***Number of DQS groups in I/O column # 1***	0
C_MEM_NDQS_COL2 ***Number of DQS groups in I/O column # 2***	0
C_MEM_NDQS_COL3 ***Number of DQS groups in I/O column # 3***	0
C_MEM_NUM_DIMMS ***Number of DIMMs***	1
C_MEM_NUM_RANKS ***No. of Ranks***	1
C_MEM_OCB_MONITOR	DEFAULT
C_MEM_ODT_TYPE ***ODT Setting***	0
C_MEM_ODT_WIDTH ***ODT Width***	1
C_MEM_PARTNO ***Part No.***	MT47H32M16-5E
C_MEM_PART_CAS_A ***<qt>CAS Lat. A</qt>***	3

C_MEM_PART_CAS_A_FMAX ***<qt>CAS Lat. A Fmax</qt>***	200
C_MEM_PART_CAS_B ***<qt>CAS Lat. B</qt>***	4
C_MEM_PART_CAS_B_FMAX ***<qt>CAS Lat. B Fmax</qt>***	200
C_MEM_PART_CAS_C ***<qt>CAS Lat. C</qt>***	0
C_MEM_PART_CAS_C_FMAX ***<qt>CAS Lat. C Fmax</qt>***	0
C_MEM_PART_CAS_D ***<qt>CAS Lat. D</qt>***	0
C_MEM_PART_CAS_D_FMAX ***<qt>CAS Lat. D Fmax</qt>***	0
C_MEM_PART_DATA_DEPTH ***Data Depth***	32
C_MEM_PART_DATA_WIDTH ***Data Width***	16
C_MEM_PART_NUM_BANK_BITS ***Bank Bits***	2
C_MEM_PART_NUM_COL_BITS ***Column Bits***	10
C_MEM_PART_NUM_ROW_BITS ***Row Bits***	13
C_MEM_PART_TAL	0
C_MEM_PART_TCCD ***tCCD***	2
C_MEM_PART_TDQSS	1
C_MEM_PART_TMRD ***tMRD (tCK)***	2
C_MEM_PART_TRAS ***tRAS (ps)***	40000
C_MEM_PART_TRASMAX ***tRASMAX (ps)***	70000000
C_MEM_PART_TRC ***tRC (ps)***	55000
C_MEM_PART_TRCD ***tRCD (ps)***	15000
C_MEM_PART_TREFI ***tREFI (ps)***	7800000
C_MEM_PART_TRFC ***tRFC (ps)***	105000
C_MEM_PART_TRP ***tRP (ps)***	15000
C_MEM_PART_TRRD ***tRRD (ps)***	10000
C_MEM_PART_TRTP	7500
C_MEM_PART_TWR ***tWR (ps)***	15000
C_MEM_PART_TWTR ***tWTR (ps)***	10000
C_MEM_PART_TZQCS ***tZQCS (tCK)***	64
C_MEM_PART_TZQINIT ***tZQINIT (tCK)***	512
C_MEM_PA_SR ***Partial Array Self Refresh***	0

C_MEM_PHASE_DETECT	DEFAULT
C_MEM_REDUCED_DRV ***<qt>Reduced Drive Output</qt>***	0
C_MEM_REG_DIMM ***Registered Memory***	0
C_MEM_SIM_CAL_OPTION	DEFAULT
C_MEM_SIM_INIT_OPTION	DEFAULT
C_MEM_SKIP_DYNAMIC_CAL	1
C_MEM_SKIP_DYN_IN_TERM	1
C_MEM_SKIP_IN_TERM_CAL	1
C_MEM_TYPE ***Type***	DDR2
C_MEM_TZQINIT_MAXCNT	512
C_MEM_WRLVL ***Enable Write Leveling***	1
C_MMCM_EXT_LOC ***MMCM_ADV Constraint Location (external)***	NOT_SET
C_MMCM_INT_LOC ***MMCM_ADV Constraint Location (internal)***	NOT_SET
C_MPMC_CLK0_PERIOD_PS ***<qt>Memory Clock Period (ps)</qt>***	8000
C_MPMC_CLK_MEM_2X_PERIOD_PS	1250
C_MPMC_CLK_MEM_PERIOD_PS	1
C_MPMC_CLK_WR_I0_PHASE	0
C_MPMC_CLK_WR_I1_PHASE	0
C_MPMC_CLK_WR_O0_PHASE	0
C_MPMC_CLK_WR_O1_PHASE	0
C_MPMC_CTRL_AWIDTH	32
C_MPMC_CTRL_DWIDTH	64
C_MPMC_CTRL_MID_WIDTH	1
C_MPMC_CTRL_NATIVE_DWIDTH	32
C_MPMC_CTRL_NUM_MASTERS	1
C_MPMC_CTRL_P2P	1
C_MPMC_CTRL_SMALLEST_MASTER	32
C_MPMC_CTRL_SUPPORT_BURSTS	0
C_NCK_PER_CLK	1
C_NUM_IDELAYCTRL ***Number of IDELAYCTRL Elements***	1
C_NUM_PORTS ***C_NUM_PORTS***	1
C_PACKAGE	fg676
C_PI0_ADDRACK_PIPELINE	1
C_PI0_PM_DC_CNTR	1
C_PI0_PM_USED	1
C_PI0_RD_FIFO_APP_PIPELINE	1
C_PI0_RD_FIFO_MEM_PIPELINE	1
C_PI0_RD_FIFO_TYPE	BRAM
C_PI0_WR_FIFO_APP_PIPELINE	1
C_PI0_WR_FIFO_MEM_PIPELINE	1
C_PI0_WR_FIFO_TYPE	BRAM
C_PI1_ADDRACK_PIPELINE	1
C_PI1_PM_DC_CNTR	1
C_PI1_PM_USED	1
C_PI1_RD_FIFO_APP_PIPELINE	1
C_PI1_RD_FIFO_MEM_PIPELINE	1
C_PI1_RD_FIFO_TYPE	BRAM

C_PI1_WR_FIFO_APP_PIPELINE	1
C_PI1_WR_FIFO_MEM_PIPELINE	1
C_PI1_WR_FIFO_TYPE	BRAM
C_PI2_ADDRACK_PIPELINE	1
C_PI2_PM_DC_CNTR	1
C_PI2_PM_USED	1
C_PI2_RD_FIFO_APP_PIPELINE	1
C_PI2_RD_FIFO_MEM_PIPELINE	1
C_PI2_RD_FIFO_TYPE	BRAM
C_PI2_WR_FIFO_APP_PIPELINE	1
C_PI2_WR_FIFO_MEM_PIPELINE	1
C_PI2_WR_FIFO_TYPE	BRAM
C_PI3_ADDRACK_PIPELINE	1
C_PI3_PM_DC_CNTR	1
C_PI3_PM_USED	1
C_PI3_RD_FIFO_APP_PIPELINE	1
C_PI3_RD_FIFO_MEM_PIPELINE	1
C_PI3_RD_FIFO_TYPE	BRAM
C_PI3_WR_FIFO_APP_PIPELINE	1
C_PI3_WR_FIFO_MEM_PIPELINE	1
C_PI3_WR_FIFO_TYPE	BRAM
C_PI4_ADDRACK_PIPELINE	1
C_PI4_PM_DC_CNTR	1
C_PI4_PM_USED	1
C_PI4_RD_FIFO_APP_PIPELINE	1
C_PI4_RD_FIFO_MEM_PIPELINE	1
C_PI4_RD_FIFO_TYPE	BRAM
C_PI4_WR_FIFO_APP_PIPELINE	1
C_PI4_WR_FIFO_MEM_PIPELINE	1
C_PI4_WR_FIFO_TYPE	BRAM
C_PI5_ADDRACK_PIPELINE	1
C_PI5_PM_DC_CNTR	1
C_PI5_PM_USED	1
C_PI5_RD_FIFO_APP_PIPELINE	1
C_PI5_RD_FIFO_MEM_PIPELINE	1
C_PI5_RD_FIFO_TYPE	BRAM
C_PI5_WR_FIFO_APP_PIPELINE	1
C_PI5_WR_FIFO_MEM_PIPELINE	1
C_PI5_WR_FIFO_TYPE	BRAM
C_PI6_ADDRACK_PIPELINE	1
C_PI6_PM_DC_CNTR	1
C_PI6_PM_USED	1
C_PI6_RD_FIFO_APP_PIPELINE	1
C_PI6_RD_FIFO_MEM_PIPELINE	1
C_PI6_RD_FIFO_TYPE	BRAM
C_PI6_WR_FIFO_APP_PIPELINE	1
C_PI6_WR_FIFO_MEM_PIPELINE	1
C_PI6_WR_FIFO_TYPE	BRAM
C_PI7_ADDRACK_PIPELINE	1
C_PI7_PM_DC_CNTR	1
C_PI7_PM_USED	1
C_PI7_RD_FIFO_APP_PIPELINE	1
C_PI7_RD_FIFO_MEM_PIPELINE	1
C_PI7_RD_FIFO_TYPE	BRAM
C_PI7_WR_FIFO_APP_PIPELINE	1
C_PI7_WR_FIFO_MEM_PIPELINE	1

C_PI7_WR_FIFO_TYPE	BRAM
C_PIM0_BASETYPE	1
C_PIM0_B_SUBTYPE	INACTIVE
C_PIM0_DATA_WIDTH ***NPI Width***	64
C_PIM0_OFFSET	0x00000000
C_PIM0_SUBTYPE	PLB
C_PIM1_BASETYPE	0
C_PIM1_B_SUBTYPE	INACTIVE
C_PIM1_DATA_WIDTH ***NPI Width***	64
C_PIM1_OFFSET	0x00000000
C_PIM1_SUBTYPE	INACTIVE
C_PIM2_BASETYPE	0
C_PIM2_B_SUBTYPE	INACTIVE
C_PIM2_DATA_WIDTH ***NPI Width***	64
C_PIM2_OFFSET	0x00000000
C_PIM2_SUBTYPE	INACTIVE
C_PIM3_BASETYPE	0
C_PIM3_B_SUBTYPE	INACTIVE
C_PIM3_DATA_WIDTH ***NPI Width***	64
C_PIM3_OFFSET	0x00000000
C_PIM3_SUBTYPE	INACTIVE
C_PIM4_BASETYPE	0
C_PIM4_B_SUBTYPE	INACTIVE
C_PIM4_DATA_WIDTH ***NPI Width***	64
C_PIM4_OFFSET	0x00000000
C_PIM4_SUBTYPE	INACTIVE
C_PIM5_BASETYPE	0
C_PIM5_B_SUBTYPE	INACTIVE
C_PIM5_DATA_WIDTH ***NPI Width***	64
C_PIM5_OFFSET	0x00000000
C_PIM5_SUBTYPE	INACTIVE
C_PIM6_BASETYPE	0
C_PIM6_B_SUBTYPE	INACTIVE
C_PIM6_DATA_WIDTH ***NPI Width***	64
C_PIM6_OFFSET	0x00000000
C_PIM6_SUBTYPE	INACTIVE
C_PIM7_BASETYPE	0
C_PIM7_B_SUBTYPE	INACTIVE
C_PIM7_DATA_WIDTH ***NPI Width***	64
C_PIM7_OFFSET	0x00000000
C_PIM7_SUBTYPE	INACTIVE
C_PM_DC_WIDTH ***<qt>Dead Cycle Counter Width</qt>***	48
C_PM_ENABLE ***Enable Performance Monitor***	0
C_PM_GC_CNTR ***<qt>Enable Global Cycle Counter</qt>***	1
C_PM_GC_WIDTH	48

<qt>Global Cycle Counter Width</qt>	
C_PM_SHIFT_CNT_BY ***<qt>Shift Value of Trans Counter</qt>***	1
C_PORT_CONFIG	1
C_PPC440MC0_BURST_LENGTH ***Burst Length***	4
C_PPC440MC0_PIPE_STAGES ***Pipe Stage***	1
C_PPC440MC1_BURST_LENGTH ***Burst Length***	4
C_PPC440MC1_PIPE_STAGES ***Pipe Stage***	1
C_PPC440MC2_BURST_LENGTH ***Burst Length***	4
C_PPC440MC2_PIPE_STAGES ***Pipe Stage***	1
C_PPC440MC3_BURST_LENGTH ***Burst Length***	4
C_PPC440MC3_PIPE_STAGES ***Pipe Stage***	1
C_PPC440MC4_BURST_LENGTH ***Burst Length***	4
C_PPC440MC4_PIPE_STAGES ***Pipe Stage***	1
C_PPC440MC5_BURST_LENGTH ***Burst Length***	4
C_PPC440MC5_PIPE_STAGES ***Pipe Stage***	1
C_PPC440MC6_BURST_LENGTH ***Burst Length***	4
C_PPC440MC6_PIPE_STAGES ***Pipe Stage***	1
C_PPC440MC7_BURST_LENGTH ***Burst Length***	4
C_PPC440MC7_PIPE_STAGES ***Pipe Stage***	1
C_RD_DATAPATH_TML_MAX_FANOUT ***Read Pipeline Max Fanout***	0
C_SDMA0_COMPLETED_ERR_RX ***Enable Completed Err on RX***	1
C_SDMA0_COMPLETED_ERR_TX ***Enable TX Completed Err***	1
C_SDMA0_PI2LL_CLK_RATIO ***MPMC to SDMA Clk Ratio***	1
C_SDMA0_PRESCALAR ***Clock Div. of Int. Timer Clk***	1023
C_SDMA1_COMPLETED_ERR_RX ***Enable Completed Err on RX***	1
C_SDMA1_COMPLETED_ERR_TX ***Enable Completed Err on TX***	1
C_SDMA1_PI2LL_CLK_RATIO ***MPMC to SDMA Clk Ratio***	1
C_SDMA1_PRESCALAR ***Clock Div. of Int. Timer Clk***	1023
C_SDMA2_COMPLETED_ERR_RX ***Enable Completed Err on RX***	1
C_SDMA2_COMPLETED_ERR_TX	1

Enable Completed Err on TX	
C_SDMA2_PI2LL_CLK_RATIO ***MPMC to SDMA Clk Ratio***	1
C_SDMA2_PRESCALAR ***Clock Div. of Int. Timer Clk***	1023
C_SDMA3_COMPLETED_ERR_RX ***Enable Completed Err on RX***	1
C_SDMA3_COMPLETED_ERR_TX ***Enable Completed Err on TX***	1
C_SDMA3_PI2LL_CLK_RATIO ***MPMC to SDMA Clk Ratio***	1
C_SDMA3_PRESCALAR ***Clock Div. of Int. Timer Clk***	1023
C_SDMA4_COMPLETED_ERR_RX ***Enable Completed Err on RX***	1
C_SDMA4_COMPLETED_ERR_TX ***Enable Completed Err on TX***	1
C_SDMA4_PI2LL_CLK_RATIO ***MPMC to SDMA Clk Ratio***	1
C_SDMA4_PRESCALAR ***Clock Div. of Int. Timer Clk***	1023
C_SDMA5_COMPLETED_ERR_RX ***Enable Completed Err on RX***	1
C_SDMA5_COMPLETED_ERR_TX ***Enable Completed Err on TX***	1
C_SDMA5_PI2LL_CLK_RATIO ***MPMC to SDMA Clk Ratio***	1
C_SDMA5_PRESCALAR ***Clock Div. of Int. Timer Clk***	1023
C_SDMA6_COMPLETED_ERR_RX ***Enable Completed Err on RX***	1
C_SDMA6_COMPLETED_ERR_TX ***Enable Completed Err on TX***	1
C_SDMA6_PI2LL_CLK_RATIO ***MPMC to SDMA Clk Ratio***	1
C_SDMA6_PRESCALAR ***Clock Div. of Int. Timer Clk***	1023
C_SDMA7_COMPLETED_ERR_RX ***Enable Completed Err on RX***	1
C_SDMA7_COMPLETED_ERR_TX ***Enable Completed Err on TX***	1
C_SDMA7_PI2LL_CLK_RATIO ***MPMC to SDMA Clk Ratio***	1
C_SDMA7_PRESCALAR ***Clock Div. of Int. Timer Clk***	1023
C_SDMA_CTRL0_AWIDTH	32
C_SDMA_CTRL0_DWIDTH	64
C_SDMA_CTRL0_MID_WIDTH	1
C_SDMA_CTRL0_NATIVE_DWIDTH	32
C_SDMA_CTRL0_NUM_MASTERS	1
C_SDMA_CTRL0_P2P	1
C_SDMA_CTRL0_SMALLEST_MASTER	32
C_SDMA_CTRL0_SUPPORT_BURSTS	0
C_SDMA_CTRL1_AWIDTH	32
C_SDMA_CTRL1_DWIDTH	64
C_SDMA_CTRL1_MID_WIDTH	1

C_SDMA_CTRL1_NATIVE_DWIDTH	32
C_SDMA_CTRL1_NUM_MASTERS	1
C_SDMA_CTRL1_P2P	1
C_SDMA_CTRL1_SMALLEST_MASTER	32
C_SDMA_CTRL1_SUPPORT_BURSTS	0
C_SDMA_CTRL2_AWIDTH	32
C_SDMA_CTRL2_DWIDTH	64
C_SDMA_CTRL2_MID_WIDTH	1
C_SDMA_CTRL2_NATIVE_DWIDTH	32
C_SDMA_CTRL2_NUM_MASTERS	1
C_SDMA_CTRL2_P2P	1
C_SDMA_CTRL2_SMALLEST_MASTER	32
C_SDMA_CTRL2_SUPPORT_BURSTS	0
C_SDMA_CTRL3_AWIDTH	32
C_SDMA_CTRL3_DWIDTH	64
C_SDMA_CTRL3_MID_WIDTH	1
C_SDMA_CTRL3_NATIVE_DWIDTH	32
C_SDMA_CTRL3_NUM_MASTERS	1
C_SDMA_CTRL3_P2P	1
C_SDMA_CTRL3_SMALLEST_MASTER	32
C_SDMA_CTRL3_SUPPORT_BURSTS	0
C_SDMA_CTRL4_AWIDTH	32
C_SDMA_CTRL4_DWIDTH	64
C_SDMA_CTRL4_MID_WIDTH	1
C_SDMA_CTRL4_NATIVE_DWIDTH	32
C_SDMA_CTRL4_NUM_MASTERS	1
C_SDMA_CTRL4_P2P	1
C_SDMA_CTRL4_SMALLEST_MASTER	32
C_SDMA_CTRL4_SUPPORT_BURSTS	0
C_SDMA_CTRL5_AWIDTH	32
C_SDMA_CTRL5_DWIDTH	64
C_SDMA_CTRL5_MID_WIDTH	1
C_SDMA_CTRL5_NATIVE_DWIDTH	32
C_SDMA_CTRL5_NUM_MASTERS	1
C_SDMA_CTRL5_P2P	1
C_SDMA_CTRL5_SMALLEST_MASTER	32
C_SDMA_CTRL5_SUPPORT_BURSTS	0
C_SDMA_CTRL6_AWIDTH	32
C_SDMA_CTRL6_DWIDTH	64
C_SDMA_CTRL6_MID_WIDTH	1
C_SDMA_CTRL6_NATIVE_DWIDTH	32
C_SDMA_CTRL6_NUM_MASTERS	1
C_SDMA_CTRL6_P2P	1
C_SDMA_CTRL6_SMALLEST_MASTER	32
C_SDMA_CTRL6_SUPPORT_BURSTS	0
C_SDMA_CTRL7_AWIDTH	32
C_SDMA_CTRL7_DWIDTH	64
C_SDMA_CTRL7_MID_WIDTH	1
C_SDMA_CTRL7_NATIVE_DWIDTH	32
C_SDMA_CTRL7_NUM_MASTERS	1
C_SDMA_CTRL7_P2P	1
C_SDMA_CTRL7_SMALLEST_MASTER	32
C_SDMA_CTRL7_SUPPORT_BURSTS	0
C_SKIP_1_VALUE	15
C_SKIP_2_VALUE	15
C_SKIP_3_VALUE	15

C_SKIP_4_VALUE	20
C_SKIP_5_VALUE	36
C_SKIP_6_VALUE	20
C_SKIP_7_VALUE	36
C_SKIP_SIM_INIT_DELAY ***Perform Shorter Simulation Initialization***	0
C_SPECIAL_BOARD ***Xilinx Special Physical Layer for Spartan3x Boards***	NONE
C_SPEEDGRADE	-4
C_SPEEDGRADE_INT	4
C_SPLB0_AWIDTH	32
C_SPLB0_DWIDTH	64
C_SPLB0_MID_WIDTH	1
C_SPLB0_NATIVE_DWIDTH ***Native Data Width of PLB***	64
C_SPLB0_NUM_MASTERS	1
C_SPLB0_P2P	1
C_SPLB0_SMALLEST_MASTER	32
C_SPLB0_SUPPORT_BURSTS	0
C_SPLB1_AWIDTH	32
C_SPLB1_DWIDTH	64
C_SPLB1_MID_WIDTH	1
C_SPLB1_NATIVE_DWIDTH ***Native Data Width of PLB***	64
C_SPLB1_NUM_MASTERS	1
C_SPLB1_P2P	1
C_SPLB1_SMALLEST_MASTER	32
C_SPLB1_SUPPORT_BURSTS	0
C_SPLB2_AWIDTH	32
C_SPLB2_DWIDTH	64
C_SPLB2_MID_WIDTH	1
C_SPLB2_NATIVE_DWIDTH ***Native Data Width of PLB***	64
C_SPLB2_NUM_MASTERS	1
C_SPLB2_P2P	1
C_SPLB2_SMALLEST_MASTER	32
C_SPLB2_SUPPORT_BURSTS	0
C_SPLB3_AWIDTH	32
C_SPLB3_DWIDTH	64
C_SPLB3_MID_WIDTH	1
C_SPLB3_NATIVE_DWIDTH ***Native Data Width of PLB***	64
C_SPLB3_NUM_MASTERS	1
C_SPLB3_P2P	1
C_SPLB3_SMALLEST_MASTER	32
C_SPLB3_SUPPORT_BURSTS	0
C_SPLB4_AWIDTH	32
C_SPLB4_DWIDTH	64
C_SPLB4_MID_WIDTH	1
C_SPLB4_NATIVE_DWIDTH ***Native Data Width of PLB***	64
C_SPLB4_NUM_MASTERS	1
C_SPLB4_P2P	1
C_SPLB4_SMALLEST_MASTER	32
C_SPLB4_SUPPORT_BURSTS	0

C_SPLB5_AWIDTH	32
C_SPLB5_DWIDTH	64
C_SPLB5_MID_WIDTH	1
C_SPLB5_NATIVE_DWIDTH ***Native Data Width of PLB***	64
C_SPLB5_NUM_MASTERS	1
C_SPLB5_P2P	1
C_SPLB5_SMALLEST_MASTER	32
C_SPLB5_SUPPORT_BURSTS	0
C_SPLB6_AWIDTH	32
C_SPLB6_DWIDTH	64
C_SPLB6_MID_WIDTH	1
C_SPLB6_NATIVE_DWIDTH ***Native Data Width of PLB***	64
C_SPLB6_NUM_MASTERS	1
C_SPLB6_P2P	1
C_SPLB6_SMALLEST_MASTER	32
C_SPLB6_SUPPORT_BURSTS	0
C_SPLB7_AWIDTH	32
C_SPLB7_DWIDTH	64
C_SPLB7_MID_WIDTH	1
C_SPLB7_NATIVE_DWIDTH ***Native Data Width of PLB***	64
C_SPLB7_NUM_MASTERS	1
C_SPLB7_P2P	1
C_SPLB7_SMALLEST_MASTER	32
C_SPLB7_SUPPORT_BURSTS	0
C_STATIC_PHY_RDDATA_CLK_SEL ***Power-on/reset Value of RDDATA_CLK_SEL Reg***	0
C_STATIC_PHY_RDDATA_SWAP_RISE ***Power-on/reset Value of RDDATA_SWAP_RISE Reg***	0
C_STATIC_PHY_RDEN_DELAY ***Power-on/reset Value of RDENDELAY Reg***	5
C_TBY4TAPVALUE	9999
C_TWR	0
C_USE_MIG_FLOW ***Use MIG Flow***	0
C_USE_STATIC_PHY ***Use Static PHY***	0
C_VFBC0_CMD_AFULL_COUNT ***VFBC Command FIFO Almost Full Count***	3
C_VFBC0_CMD_FIFO_DEPTH ***VFBC Command FIFO Depth***	32
C_VFBC0_RDWD_DATA_WIDTH ***VFBC Data FIFO Width***	32
C_VFBC0_RDWD_FIFO_DEPTH ***VFBC Data FIFO Depth***	1024
C_VFBC0_RD_AEMPTY_WD_AFULL_COUNT ***VFBC Data FIFO Almost Full/Empty Count***	3
C_VFBC1_CMD_AFULL_COUNT ***VFBC Command FIFO Almost Full Count***	3
C_VFBC1_CMD_FIFO_DEPTH ***VFBC Command FIFO Depth***	32
C_VFBC1_RDWD_DATA_WIDTH ***VFBC Data FIFO Width***	32

C_VFBC1_RDWD_FIFO_DEPTH ***VFBC Data FIFO Depth***	1024
C_VFBC1_RD_AEMPTY_WD_AFULL_COUNT ***VFBC Data FIFO Almost Full/Empty Count***	3
C_VFBC2_CMD_AFULL_COUNT ***VFBC Command FIFO Almost Full Count***	3
C_VFBC2_CMD_FIFO_DEPTH ***VFBC Command FIFO Depth***	32
C_VFBC2_RDWD_DATA_WIDTH ***VFBC Data FIFO Width***	32
C_VFBC2_RDWD_FIFO_DEPTH ***VFBC Data FIFO Depth***	1024
C_VFBC2_RD_AEMPTY_WD_AFULL_COUNT ***VFBC Data FIFO Almost Full/Empty Count***	3
C_VFBC3_CMD_AFULL_COUNT ***VFBC Command FIFO Almost Full Count***	3
C_VFBC3_CMD_FIFO_DEPTH ***VFBC Command FIFO Depth***	32
C_VFBC3_RDWD_DATA_WIDTH ***VFBC Data FIFO Width***	32
C_VFBC3_RDWD_FIFO_DEPTH ***VFBC Data FIFO Depth***	1024
C_VFBC3_RD_AEMPTY_WD_AFULL_COUNT ***VFBC Data FIFO Almost Full/Empty Count***	3
C_VFBC4_CMD_AFULL_COUNT ***VFBC Command FIFO Almost Full Count***	3
C_VFBC4_CMD_FIFO_DEPTH ***VFBC Command FIFO Depth***	32
C_VFBC4_RDWD_DATA_WIDTH ***VFBC Data FIFO Width***	32
C_VFBC4_RDWD_FIFO_DEPTH ***VFBC Data FIFO Depth***	1024
C_VFBC4_RD_AEMPTY_WD_AFULL_COUNT ***VFBC Data FIFO Almost Full/Empty Count***	3
C_VFBC5_CMD_AFULL_COUNT ***VFBC Command FIFO Almost Full Count***	3
C_VFBC5_CMD_FIFO_DEPTH ***VFBC Command FIFO Depth***	32
C_VFBC5_RDWD_DATA_WIDTH ***VFBC Data FIFO Width***	32
C_VFBC5_RDWD_FIFO_DEPTH ***VFBC Data FIFO Depth***	1024
C_VFBC5_RD_AEMPTY_WD_AFULL_COUNT ***VFBC Data FIFO Almost Full/Empty Count***	3
C_VFBC6_CMD_AFULL_COUNT ***VFBC Command FIFO Almost Full Count***	3
C_VFBC6_CMD_FIFO_DEPTH ***VFBC Command FIFO Depth***	32
C_VFBC6_RDWD_DATA_WIDTH ***VFBC Data FIFO Width***	32
C_VFBC6_RDWD_FIFO_DEPTH ***VFBC Data FIFO Depth***	1024
C_VFBC6_RD_AEMPTY_WD_AFULL_COUNT ***VFBC Data FIFO Almost Full/Empty Count***	3
C_VFBC7_CMD_AFULL_COUNT ***VFBC Command FIFO Almost Full Count***	3
C_VFBC7_CMD_FIFO_DEPTH	32

VFBC Command FIFO Depth	
C_VFBC7_RDWD_DATA_WIDTH ***VFBC Data FIFO Width***	32
C_VFBC7_RDWD_FIFO_DEPTH ***VFBC Data FIFO Depth***	1024
C_VFBC7_RD_AEMPTY_WD_AFULL_COUNT ***VFBC Data FIFO Almost Full/Empty Count***	3
C_WR_DATAPATH_TML_PIPELINE ***Write TML Pipeline***	1
C_WR_TRAINING_PORT ***Specifies Which Port's Write FIFO will be used for Memory Initialization***	0
C_XCL0_B_IN_USE ***Use Channel B***	1
C_XCL0_B_LINESIZE ***Channel B Line Size***	4
C_XCL0_B_WRITEXFER ***Channel B Write Transfer***	1
C_XCL0_LINESIZE ***Cache Line Size***	4
C_XCL0_PIPE_STAGES ***XCL Pipe Stage***	2
C_XCL0_WRITEXFER ***Write Transfer***	1
C_XCL1_B_IN_USE ***Use Channel B***	0
C_XCL1_B_LINESIZE ***Channel B Line Size***	4
C_XCL1_B_WRITEXFER ***Channel B Write Transfer***	1
C_XCL1_LINESIZE ***Cache Line Size***	4
C_XCL1_PIPE_STAGES ***XCL Pipe Stage***	2
C_XCL1_WRITEXFER ***Write Transfer***	1
C_XCL2_B_IN_USE ***Use Channel B***	0
C_XCL2_B_LINESIZE ***Channel B Line Size***	4
C_XCL2_B_WRITEXFER ***Channel B Write Transfer***	1
C_XCL2_LINESIZE ***Cache Line Size***	4
C_XCL2_PIPE_STAGES ***XCL Pipe Stage***	2
C_XCL2_WRITEXFER ***Write Transfer***	1
C_XCL3_B_IN_USE ***Use Channel B***	0
C_XCL3_B_LINESIZE ***Channel B Line Size***	4
C_XCL3_B_WRITEXFER ***Channel B Write Transfer***	1
C_XCL3_LINESIZE ***Cache Line Size***	4
C_XCL3_PIPE_STAGES	2

XCL Pipe Stage	
C_XCL3_WRITEXFER ***Write Transfer***	1
C_XCL4_B_IN_USE ***Use Channel B***	0
C_XCL4_B_LINESIZE ***Channel B Line Size***	4
C_XCL4_B_WRITEXFER ***Channel B Write Transfer***	1
C_XCL4_LINESIZE ***Cache Line Size***	4
C_XCL4_PIPE_STAGES ***XCL Pipe Stage***	2
C_XCL4_WRITEXFER ***Write Transfer***	1
C_XCL5_B_IN_USE ***Use Channel B***	0
C_XCL5_B_LINESIZE ***Channel B Line Size***	4
C_XCL5_B_WRITEXFER ***Channel B Write Transfer***	1
C_XCL5_LINESIZE ***Cache Line Size***	4
C_XCL5_PIPE_STAGES ***XCL Pipe Stage***	2
C_XCL5_WRITEXFER ***Write Transfer***	1
C_XCL6_B_IN_USE ***Use Channel B***	0
C_XCL6_B_LINESIZE ***Channel B Line Size***	4
C_XCL6_B_WRITEXFER ***Channel B Write Transfer***	1
C_XCL6_LINESIZE ***Cache Line Size***	4
C_XCL6_PIPE_STAGES ***XCL Pipe Stage***	2
C_XCL6_WRITEXFER ***Write Transfer***	1
C_XCL7_B_IN_USE ***Use Channel B***	0
C_XCL7_B_LINESIZE ***Channel B Line Size***	4
C_XCL7_B_WRITEXFER ***Channel B Write Transfer***	1
C_XCL7_LINESIZE ***Cache Line Size***	4
C_XCL7_PIPE_STAGES ***XCL Pipe Stage***	2
C_XCL7_WRITEXFER ***Write Transfer***	1

3.8.4 Post Synthesis Device Utilization

	Used	**Total**	**Percentage**
Number of Slices	1037	16640	6

Number of Slice Flip Flops	1605	33280	4
Number of 4-input LUTs	1006	33280	3
Number of IOs	13370	NA	NA
Number of bonded IOBs	71	519	13
Number of BRAMs	5	84	5

3.8.5 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: 7.575ns (Maximum Frequency: 132.013MHz)

Minimum input arrival time before clock: 36.153ns

Maximum output required time after clock: 5.541ns

Maximum combinational path delay: 4.520ns

3.9 SRAM 1

SRAM_1 XPS Multi-Channel External Memory Controller(SRAM/Flash)
Xilinx Multi-CHannel (MCH) PLBV46 external memory controller.Three SRAM controllers are used for three external SRAM memory banks, which are used to store received data for 24 channel receivers. These three external SRAM memory banks can be accessed by processor via these three memory controllers. During acquisition the SRAMs are controlled by acquisition logic and the accessing from processor is disabled in order to avoid confliction.

3.9.1 Port List

These are the ports listed in the MHS file. Please refer to the IP documentation for complete information about module ports.

#	NAME	DIR	[LSB: MSB]	SIGNAL
0	Mem_A	**O**	**0:31**	0b0000000000 & mb_sram1_addr_10_29 & 0b00
1	RdClk	**I**	**1**	clk_62_5000MHz
2	Mem_CEN	**O**	**1**	mb_sram1_cen
3	Mem_OEN	**O**	**1**	mb_sram1_oen
4	Mem_WEN	**O**	**1**	mb_sram1_wen
5	Mem_DQ_I	**I**	**0:31**	mb_dq1_i
6	Mem_DQ_O	**O**	**0:31**	mb_dq1_o

3.9.2 Bus Interfaces

Bus Interfaces				
NAME	**TYPE**	**BUSSTD**	**BUS**	**Point 2 Point**
SPLB	SLAVE	PLBV46	mb_plb	microblaze_0

3.9.3 Parameters

These are the current parameter settings for this module.

Name	Value
C_FAMILY ***Device Family***	spartan3adsp
C_MEM0_BASEADDR ***Base Address of Bank 0***	0xA3000000
C_MEM0_HIGHADDR ***High Address of Bank 0***	0xA33FFFFF
C_MEM1_BASEADDR ***Base Address of Bank 1***	0xFFFFFFFF
C_MEM1_HIGHADDR ***High Address of Bank 1***	0x00000000
C_MEM2_BASEADDR ***Base Address of Bank 2***	0xFFFFFFFF
C_MEM2_HIGHADDR ***High Address of Bank 2***	0x00000000
C_MEM3_BASEADDR ***Base Address of Bank 3***	0xFFFFFFFF
C_MEM3_HIGHADDR ***High Address of Bank 3***	0x00000000
C_INCLUDE_DATAWIDTH_MATCHING_0 ***Execute Multiple Memory Accesses To Match Bank 0 Data Bus Width To PLB Data Bus Width***	0
C_INCLUDE_DATAWIDTH_MATCHING_1 ***Execute Multiple Memory Accesses To Match Bank 1 Data Bus Width To PLB Data Bus Width***	0
C_INCLUDE_DATAWIDTH_MATCHING_2 ***Execute Multiple Memory Accesses To Match Bank 2 Data Bus Width To PLB Data Bus Width***	0
C_INCLUDE_DATAWIDTH_MATCHING_3 ***Execute Multiple Memory Accesses To Match Bank 3 Data Bus Width To PLB Data Bus Width***	0
C_INCLUDE_NEGEDGE_IOREGS ***Use Falling Edge IO Register in Interface Signals***	0
C_INCLUDE_PLB_IPIF ***Include PLB Slave Interface***	1
C_INCLUDE_WRBUF ***Include Write Buffer***	1
C_MAX_MEM_WIDTH ***Maximum Data Bus Width***	32
C_MCH0_ACCESSBUF_DEPTH ***Depth of Access Buffer of Ch 0***	16
C_MCH0_PROTOCOL ***Interface Protocol of Ch 0***	0
C_MCH0_RDDATABUF_DEPTH ***Depth of Read Data Buffer Depath of Ch 0***	16
C_MCH1_ACCESSBUF_DEPTH ***Depth of Access Buffer of Ch 1***	16
C_MCH1_PROTOCOL ***Interface Protocol of Ch 1***	0
C_MCH1_RDDATABUF_DEPTH ***Depth of Read Data Buffer of Ch 1***	16
C_MCH2_ACCESSBUF_DEPTH	16

Depth of Access Buffer of Ch 2	
C_MCH2_PROTOCOL ***Interface Protocol of Ch 2***	0
C_MCH2_RDDATABUF_DEPTH ***Depth of Read Data Buffer of Ch 2***	16
C_MCH3_ACCESSBUF_DEPTH ***Depth of Access Buffer of Ch 3***	16
C_MCH3_PROTOCOL ***Interface Protocol of Ch 3***	0
C_MCH3_RDDATABUF_DEPTH ***Depth of Read Data Buffer of Ch 3***	16
C_MCH_NATIVE_DWIDTH ***Data Bus Width of MCH***	32
C_MCH_SPLB_AWIDTH ***MCH and PLB Address Bus Width***	32
C_MCH_SPLB_CLK_PERIOD_PS ***MCH and PLB Clock Period***	16000
C_MEM0_WIDTH ***Data Bus Width of Bank 0***	32
C_MEM1_WIDTH ***Data Bus Width of Bank 1***	32
C_MEM2_WIDTH ***Data Bus Width of Bank 2***	32
C_MEM3_WIDTH ***Data Bus Width of Bank 3***	32
C_NUM_BANKS_MEM ***Number of Memory Banks***	1
C_NUM_CHANNELS ***Number of MCH Channels***	0
C_PAGEMODE_FLASH_0 ***Page mode flash enable of Bank 0***	0
C_PAGEMODE_FLASH_1 ***Page mode flash enable of Bank 1***	0
C_PAGEMODE_FLASH_2 ***Page mode flash enable of Bank 2***	0
C_PAGEMODE_FLASH_3 ***Page mode flash enable of Bank 3***	0
C_PRIORITY_MODE ***Arbitration Mode Between PLB and MCH Interface***	0
C_SPLB_DWIDTH ***PLB Data Bus Width***	32
C_SPLB_MID_WIDTH ***Master ID Bus Width of PLB***	1
C_SPLB_NUM_MASTERS ***Number of PLB Masters***	2
C_SPLB_P2P ***PLB Slave Uses P2P Topology***	0
C_SPLB_SMALLEST_MASTER ***Smallest Master Data Bus Width***	32
C_SYNCH_MEM_0 ***Bank 0 is Synchronous***	0
C_SYNCH_MEM_1 ***Bank 1 is Synchronous***	0
C_SYNCH_MEM_2 ***Bank 2 is Synchronous***	0
C_SYNCH_MEM_3 ***Bank 3 is Synchronous***	0

C_SYNCH_PIPEDELAY_0 ***Pipeline Latency of Bank 0***	2
C_SYNCH_PIPEDELAY_1 ***Pipeline Latency of Bank 1***	2
C_SYNCH_PIPEDELAY_2 ***Pipeline Latency of Bank 2***	2
C_SYNCH_PIPEDELAY_3 ***Pipeline Latency of Bank 3***	2
C_TAVDV_PS_MEM_0 ***TAVDV of Bank 0***	10000
C_TAVDV_PS_MEM_1 ***TAVDV of Bank 1***	15000
C_TAVDV_PS_MEM_2 ***TAVDV of Bank 2***	15000
C_TAVDV_PS_MEM_3 ***TAVDV of Bank 3***	15000
C_TCEDV_PS_MEM_0 ***TCEDV of Bank 0***	10000
C_TCEDV_PS_MEM_1 ***TCEDV of Bank 1***	15000
C_TCEDV_PS_MEM_2 ***TCEDV of Bank 2***	15000
C_TCEDV_PS_MEM_3 ***TCEDV of Bank 3***	15000
C_THZCE_PS_MEM_0 ***THZCE of Bank 0***	7000
C_THZCE_PS_MEM_1 ***THZCE of Bank 1***	7000
C_THZCE_PS_MEM_2 ***THZCE of Bank 2***	7000
C_THZCE_PS_MEM_3 ***THZCE of Bank 3***	7000
C_THZOE_PS_MEM_0 ***THZOE of Bank 0***	7000
C_THZOE_PS_MEM_1 ***THZOE of Bank 1***	7000
C_THZOE_PS_MEM_2 ***THZOE of Bank 2***	7000
C_THZOE_PS_MEM_3 ***THZOE of Bank 3***	7000
C_TLZWE_PS_MEM_0 ***TLZWE of Bank 0***	0
C_TLZWE_PS_MEM_1 ***TLZWE of Bank 1***	0
C_TLZWE_PS_MEM_2 ***TLZWE of Bank 2***	0
C_TLZWE_PS_MEM_3 ***TLZWE of Bank 3***	0
C_TPACC_PS_FLASH_0 ***TPACC of Bank 0***	25000
C_TPACC_PS_FLASH_1 ***TPACC of Bank 1***	25000
C_TPACC_PS_FLASH_2 ***TPACC of Bank 2***	25000
C_TPACC_PS_FLASH_3 ***TPACC of Bank 3***	25000
C_TWC_PS_MEM_0	8000

TWC of Bank 0	
C_TWC_PS_MEM_1 ***TWC of Bank 1***	15000
C_TWC_PS_MEM_2 ***TWC of Bank 2***	15000
C_TWC_PS_MEM_3 ***TWC of Bank 3***	15000
C_TWP_PS_MEM_0 ***TWP of Bank 0***	8000
C_TWP_PS_MEM_1 ***TWP of Bank 1***	12000
C_TWP_PS_MEM_2 ***TWP of Bank 2***	12000
C_TWP_PS_MEM_3 ***TWP of Bank 3***	12000
C_XCL0_LINESIZE ***Cacheline Size of Ch0***	4
C_XCL0_WRITEXFER ***Write Transfer Type of Ch0***	1
C_XCL1_LINESIZE ***Cacheline Size of Ch1***	4
C_XCL1_WRITEXFER ***Write Transfer Type of Ch1***	1
C_XCL2_LINESIZE ***Cacheline Size of Ch2***	4
C_XCL2_WRITEXFER ***Write Transfer Type of Ch2***	1
C_XCL3_LINESIZE ***Cacheline Size of Ch3***	4
C_XCL3_WRITEXFER ***Write Transfer Type of Ch3***	1

3.9.4 Post Synthesis Device Utilization

	Used	**Total**	**Percentage**
Number of Slices	360	16640	2%
Number of Slice Flip Flops	479	33280	1%
Number of 4 input LUTs	353	33280	1%
Number of IOs	627		
Number of bonded IOBs	0	519	0%

3.9.5 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: 10.474ns (Maximum Frequency: 95.474MHz)

Minimum input arrival time before clock: 2.931ns

Maximum output required time after clock: 1.829ns

Maximum combinational path delay: No path found

3.10 SRAM 2

SRAM_2 XPS Multi-Channel External Memory Controller(SRAM/Flash)
Xilinx Multi-CHannel (MCH) PLBV46 external memory controller

3.10.1 Port List

These are the ports listed in the MHS file. Please refer to the IP documentation for complete information about module ports.

#	NAME	DIR	[LSB: MSB]	SIGNAL
0	Mem_A	**O**	**0:31**	0b0000000000 & mb_sram1_addr_10_29 & 0b00
1	RdClk	**I**	**1**	clk_62_5000MHz
2	Mem_CEN	**O**	**1**	mb_sram1_cen
3	Mem_OEN	**O**	**1**	mb_sram1_oen
4	Mem_WEN	**O**	**1**	mb_sram1_wen
5	Mem_DQ_I	**I**	**0:31**	mb_dq1_i
6	Mem_DQ_O	**O**	**0:31**	mb_dq1_o

3.10.2 Bus Interfaces

Bus Interfaces				
NAME	**TYPE**	**BUSSTD**	**BUS**	**Point 2 Point**
SPLB	SLAVE	PLBV46	mb_plb	microblaze_0

3.10.3 Parameters

These are the current parameter settings for this module.

Name	Value
C_FAMILY ***Device Family***	spartan3adsp
C_MEM0_BASEADDR ***Base Address of Bank 0***	0xA3400000
C_MEM0_HIGHADDR ***High Address of Bank 0***	0xA37FFFFF
C_MEM1_BASEADDR ***Base Address of Bank 1***	0xFFFFFFFF
C_MEM1_HIGHADDR ***High Address of Bank 1***	0x00000000
C_MEM2_BASEADDR ***Base Address of Bank 2***	0xFFFFFFFF
C_MEM2_HIGHADDR ***High Address of Bank 2***	0x00000000
C_MEM3_BASEADDR ***Base Address of Bank 3***	0xFFFFFFFF
C_MEM3_HIGHADDR ***High Address of Bank 3***	0x00000000
C_INCLUDE_DATAWIDTH_MATCHING_0 ***Execute Multiple Memory Accesses To Match Bank 0 Data Bus Width To PLB Data Bus Width***	0
C_INCLUDE_DATAWIDTH_MATCHING_1 ***Execute Multiple Memory Accesses To Match Bank 1 Data***	0

Bus Width To PLB Data Bus Width	
C_INCLUDE_DATAWIDTH_MATCHING_2 ***Execute Multiple Memory Accesses To Match Bank 2 Data Bus Width To PLB Data Bus Width***	0
C_INCLUDE_DATAWIDTH_MATCHING_3 ***Execute Multiple Memory Accesses To Match Bank 3 Data Bus Width To PLB Data Bus Width***	0
C_INCLUDE_NEGEDGE_IOREGS ***Use Falling Edge IO Register in Interface Signals***	0
C_INCLUDE_PLB_IPIF ***Include PLB Slave Interface***	1
C_INCLUDE_WRBUF ***Include Write Buffer***	1
C_MAX_MEM_WIDTH ***Maximum Data Bus Width***	32
C_MCH0_ACCESSBUF_DEPTH ***Depth of Access Buffer of Ch 0***	16
C_MCH0_PROTOCOL ***Interface Protocol of Ch 0***	0
C_MCH0_RDDATABUF_DEPTH ***Depth of Read Data Buffer Depath of Ch 0***	16
C_MCH1_ACCESSBUF_DEPTH ***Depth of Access Buffer of Ch 1***	16
C_MCH1_PROTOCOL ***Interface Protocol of Ch 1***	0
C_MCH1_RDDATABUF_DEPTH ***Depth of Read Data Buffer of Ch 1***	16
C_MCH2_ACCESSBUF_DEPTH ***Depth of Access Buffer of Ch 2***	16
C_MCH2_PROTOCOL ***Interface Protocol of Ch 2***	0
C_MCH2_RDDATABUF_DEPTH ***Depth of Read Data Buffer of Ch 2***	16
C_MCH3_ACCESSBUF_DEPTH ***Depth of Access Buffer of Ch 3***	16
C_MCH3_PROTOCOL ***Interface Protocol of Ch 3***	0
C_MCH3_RDDATABUF_DEPTH ***Depth of Read Data Buffer of Ch 3***	16
C_MCH_NATIVE_DWIDTH ***Data Bus Width of MCH***	32
C_MCH_SPLB_AWIDTH ***MCH and PLB Address Bus Width***	32
C_MCH_SPLB_CLK_PERIOD_PS ***MCH and PLB Clock Period***	16000
C_MEM0_WIDTH ***Data Bus Width of Bank 0***	32
C_MEM1_WIDTH ***Data Bus Width of Bank 1***	32
C_MEM2_WIDTH ***Data Bus Width of Bank 2***	32
C_MEM3_WIDTH ***Data Bus Width of Bank 3***	32
C_NUM_BANKS_MEM ***Number of Memory Banks***	1
C_NUM_CHANNELS ***Number of MCH Channels***	0

C_PAGEMODE_FLASH_0 ***Page mode flash enable of Bank 0***	0
C_PAGEMODE_FLASH_1 ***Page mode flash enable of Bank 1***	0
C_PAGEMODE_FLASH_2 ***Page mode flash enable of Bank 2***	0
C_PAGEMODE_FLASH_3 ***Page mode flash enable of Bank 3***	0
C_PRIORITY_MODE ***Arbitration Mode Between PLB and MCH Interface***	0
C_SPLB_DWIDTH ***PLB Data Bus Width***	32
C_SPLB_MID_WIDTH ***Master ID Bus Width of PLB***	1
C_SPLB_NUM_MASTERS ***Number of PLB Masters***	2
C_SPLB_P2P ***PLB Slave Uses P2P Topology***	0
C_SPLB_SMALLEST_MASTER ***Smallest Master Data Bus Width***	32
C_SYNCH_MEM_0 ***Bank 0 is Synchronous***	0
C_SYNCH_MEM_1 ***Bank 1 is Synchronous***	0
C_SYNCH_MEM_2 ***Bank 2 is Synchronous***	0
C_SYNCH_MEM_3 ***Bank 3 is Synchronous***	0
C_SYNCH_PIPEDELAY_0 ***Pipeline Latency of Bank 0***	2
C_SYNCH_PIPEDELAY_1 ***Pipeline Latency of Bank 1***	2
C_SYNCH_PIPEDELAY_2 ***Pipeline Latency of Bank 2***	2
C_SYNCH_PIPEDELAY_3 ***Pipeline Latency of Bank 3***	2
C_TAVDV_PS_MEM_0 ***TAVDV of Bank 0***	10000
C_TAVDV_PS_MEM_1 ***TAVDV of Bank 1***	15000
C_TAVDV_PS_MEM_2 ***TAVDV of Bank 2***	15000
C_TAVDV_PS_MEM_3 ***TAVDV of Bank 3***	15000
C_TCEDV_PS_MEM_0 ***TCEDV of Bank 0***	10000
C_TCEDV_PS_MEM_1 ***TCEDV of Bank 1***	15000
C_TCEDV_PS_MEM_2 ***TCEDV of Bank 2***	15000
C_TCEDV_PS_MEM_3 ***TCEDV of Bank 3***	15000
C_THZCE_PS_MEM_0 ***THZCE of Bank 0***	7000
C_THZCE_PS_MEM_1 ***THZCE of Bank 1***	7000
C_THZCE_PS_MEM_2	7000

THZCE of Bank 2	
C_THZCE_PS_MEM_3 ***THZCE of Bank 3***	7000
C_THZOE_PS_MEM_0 ***THZOE of Bank 0***	7000
C_THZOE_PS_MEM_1 ***THZOE of Bank 1***	7000
C_THZOE_PS_MEM_2 ***THZOE of Bank 2***	7000
C_THZOE_PS_MEM_3 ***THZOE of Bank 3***	7000
C_TLZWE_PS_MEM_0 ***TLZWE of Bank 0***	0
C_TLZWE_PS_MEM_1 ***TLZWE of Bank 1***	0
C_TLZWE_PS_MEM_2 ***TLZWE of Bank 2***	0
C_TLZWE_PS_MEM_3 ***TLZWE of Bank 3***	0
C_TPACC_PS_FLASH_0 ***TPACC of Bank 0***	25000
C_TPACC_PS_FLASH_1 ***TPACC of Bank 1***	25000
C_TPACC_PS_FLASH_2 ***TPACC of Bank 2***	25000
C_TPACC_PS_FLASH_3 ***TPACC of Bank 3***	25000
C_TWC_PS_MEM_0 ***TWC of Bank 0***	8000
C_TWC_PS_MEM_1 ***TWC of Bank 1***	15000
C_TWC_PS_MEM_2 ***TWC of Bank 2***	15000
C_TWC_PS_MEM_3 ***TWC of Bank 3***	15000
C_TWP_PS_MEM_0 ***TWP of Bank 0***	8000
C_TWP_PS_MEM_1 ***TWP of Bank 1***	12000
C_TWP_PS_MEM_2 ***TWP of Bank 2***	12000
C_TWP_PS_MEM_3 ***TWP of Bank 3***	12000
C_XCL0_LINESIZE ***Cacheline Size of Ch0***	4
C_XCL0_WRITEXFER ***Write Transfer Type of Ch0***	1
C_XCL1_LINESIZE ***Cacheline Size of Ch1***	4
C_XCL1_WRITEXFER ***Write Transfer Type of Ch1***	1
C_XCL2_LINESIZE ***Cacheline Size of Ch2***	4
C_XCL2_WRITEXFER ***Write Transfer Type of Ch2***	1
C_XCL3_LINESIZE ***Cacheline Size of Ch3***	4

C_XCL3_WRITEXFER *Write Transfer Type of Ch3*	1

3.10.4 Post Synthesis Device Utilization

	Used	Total	Percentage
Number of Slices	360	16640	2%
Number of Slice Flip Flops	479	33280	1%
Number of 4 input LUTs	353	33280	1%
Number of IOs	627		
Number of bonded IOBs	0	519	0%

3.10.5 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: 10.474ns (Maximum Frequency: 95.474MHz)

Minimum input arrival time before clock: 2.931ns

Maximum output required time after clock: 1.829ns

Maximum combinational path delay: No path found

3.11 SRAM 3

SRAM_3 XPS Multi-Channel External Memory Controller(SRAM/Flash)
Xilinx Multi-CHannel (MCH) PLBV46 external memory controller

3.11.1 Port List

These are the ports listed in the MHS file. Please refer to the IP documentation for complete information about module ports.

#	NAME	DIR	[LSB: MSB]	SIGNAL
0	Mem_A	**O**	**0:31**	0b0000000000 & mb_sram1_addr_10_29 & 0b00
1	RdClk	**I**	**1**	clk_62_5000MHz
2	Mem_CEN	**O**	**1**	mb_sram1_cen
3	Mem_OEN	**O**	**1**	mb_sram1_oen
4	Mem_WEN	**O**	**1**	mb_sram1_wen
5	Mem_DQ_I	**I**	**0:31**	mb_dq1_i
6	Mem_DQ_O	**O**	**0:31**	mb_dq1_o

3.11.2 Bus Interfaces

Bus Interfaces				
NAME	**TYPE**	**BUSSTD**	**BUS**	**Point 2 Point**
SPLB	SLAVE	PLBV46	mb_plb	microblaze_0

3.11.3 Parameters

These are the current parameter settings for this module.

Name	Value
C_FAMILY ***Device Family***	spartan3adsp
C_MEM0_BASEADDR ***Base Address of Bank 0***	0xA3800000
C_MEM0_HIGHADDR ***High Address of Bank 0***	0xA3BFFFFF
C_MEM1_BASEADDR ***Base Address of Bank 1***	0xFFFFFFFF
C_MEM1_HIGHADDR ***High Address of Bank 1***	0x00000000
C_MEM2_BASEADDR ***Base Address of Bank 2***	0xFFFFFFFF
C_MEM2_HIGHADDR ***High Address of Bank 2***	0x00000000
C_MEM3_BASEADDR ***Base Address of Bank 3***	0xFFFFFFFF
C_MEM3_HIGHADDR ***High Address of Bank 3***	0x00000000
C_INCLUDE_DATAWIDTH_MATCHING_0 ***Execute Multiple Memory Accesses To Match Bank 0 Data Bus Width To PLB Data Bus Width***	0
C_INCLUDE_DATAWIDTH_MATCHING_1 ***Execute Multiple Memory Accesses To Match Bank 1 Data Bus Width To PLB Data Bus Width***	0
C_INCLUDE_DATAWIDTH_MATCHING_2 ***Execute Multiple Memory Accesses To Match Bank 2 Data Bus Width To PLB Data Bus Width***	0
C_INCLUDE_DATAWIDTH_MATCHING_3 ***Execute Multiple Memory Accesses To Match Bank 3 Data Bus Width To PLB Data Bus Width***	0
C_INCLUDE_NEGEDGE_IOREGS ***Use Falling Edge IO Register in Interface Signals***	0
C_INCLUDE_PLB_IPIF ***Include PLB Slave Interface***	1
C_INCLUDE_WRBUF ***Include Write Buffer***	1
C_MAX_MEM_WIDTH ***Maximum Data Bus Width***	32
C_MCH0_ACCESSBUF_DEPTH ***Depth of Access Buffer of Ch 0***	16
C_MCH0_PROTOCOL ***Interface Protocol of Ch 0***	0
C_MCH0_RDDATABUF_DEPTH ***Depth of Read Data Buffer Depath of Ch 0***	16
C_MCH1_ACCESSBUF_DEPTH ***Depth of Access Buffer of Ch 1***	16
C_MCH1_PROTOCOL ***Interface Protocol of Ch 1***	0
C_MCH1_RDDATABUF_DEPTH ***Depth of Read Data Buffer of Ch 1***	16
C_MCH2_ACCESSBUF_DEPTH	16

Depth of Access Buffer of Ch 2	
C_MCH2_PROTOCOL ***Interface Protocol of Ch 2***	0
C_MCH2_RDDATABUF_DEPTH ***Depth of Read Data Buffer of Ch 2***	16
C_MCH3_ACCESSBUF_DEPTH ***Depth of Access Buffer of Ch 3***	16
C_MCH3_PROTOCOL ***Interface Protocol of Ch 3***	0
C_MCH3_RDDATABUF_DEPTH ***Depth of Read Data Buffer of Ch 3***	16
C_MCH_NATIVE_DWIDTH ***Data Bus Width of MCH***	32
C_MCH_SPLB_AWIDTH ***MCH and PLB Address Bus Width***	32
C_MCH_SPLB_CLK_PERIOD_PS ***MCH and PLB Clock Period***	16000
C_MEM0_WIDTH ***Data Bus Width of Bank 0***	32
C_MEM1_WIDTH ***Data Bus Width of Bank 1***	32
C_MEM2_WIDTH ***Data Bus Width of Bank 2***	32
C_MEM3_WIDTH ***Data Bus Width of Bank 3***	32
C_NUM_BANKS_MEM ***Number of Memory Banks***	1
C_NUM_CHANNELS ***Number of MCH Channels***	0
C_PAGEMODE_FLASH_0 ***Page mode flash enable of Bank 0***	0
C_PAGEMODE_FLASH_1 ***Page mode flash enable of Bank 1***	0
C_PAGEMODE_FLASH_2 ***Page mode flash enable of Bank 2***	0
C_PAGEMODE_FLASH_3 ***Page mode flash enable of Bank 3***	0
C_PRIORITY_MODE ***Arbitration Mode Between PLB and MCH Interface***	0
C_SPLB_DWIDTH ***PLB Data Bus Width***	32
C_SPLB_MID_WIDTH ***Master ID Bus Width of PLB***	1
C_SPLB_NUM_MASTERS ***Number of PLB Masters***	2
C_SPLB_P2P ***PLB Slave Uses P2P Topology***	0
C_SPLB_SMALLEST_MASTER ***Smallest Master Data Bus Width***	32
C_SYNCH_MEM_0 ***Bank 0 is Synchronous***	0
C_SYNCH_MEM_1 ***Bank 1 is Synchronous***	0
C_SYNCH_MEM_2 ***Bank 2 is Synchronous***	0
C_SYNCH_MEM_3 ***Bank 3 is Synchronous***	0

C_SYNCH_PIPEDELAY_0 ***Pipeline Latency of Bank 0***	2
C_SYNCH_PIPEDELAY_1 ***Pipeline Latency of Bank 1***	2
C_SYNCH_PIPEDELAY_2 ***Pipeline Latency of Bank 2***	2
C_SYNCH_PIPEDELAY_3 ***Pipeline Latency of Bank 3***	2
C_TAVDV_PS_MEM_0 ***TAVDV of Bank 0***	10000
C_TAVDV_PS_MEM_1 ***TAVDV of Bank 1***	15000
C_TAVDV_PS_MEM_2 ***TAVDV of Bank 2***	15000
C_TAVDV_PS_MEM_3 ***TAVDV of Bank 3***	15000
C_TCEDV_PS_MEM_0 ***TCEDV of Bank 0***	10000
C_TCEDV_PS_MEM_1 ***TCEDV of Bank 1***	15000
C_TCEDV_PS_MEM_2 ***TCEDV of Bank 2***	15000
C_TCEDV_PS_MEM_3 ***TCEDV of Bank 3***	15000
C_THZCE_PS_MEM_0 ***THZCE of Bank 0***	7000
C_THZCE_PS_MEM_1 ***THZCE of Bank 1***	7000
C_THZCE_PS_MEM_2 ***THZCE of Bank 2***	7000
C_THZCE_PS_MEM_3 ***THZCE of Bank 3***	7000
C_THZOE_PS_MEM_0 ***THZOE of Bank 0***	7000
C_THZOE_PS_MEM_1 ***THZOE of Bank 1***	7000
C_THZOE_PS_MEM_2 ***THZOE of Bank 2***	7000
C_THZOE_PS_MEM_3 ***THZOE of Bank 3***	7000
C_TLZWE_PS_MEM_0 ***TLZWE of Bank 0***	0
C_TLZWE_PS_MEM_1 ***TLZWE of Bank 1***	0
C_TLZWE_PS_MEM_2 ***TLZWE of Bank 2***	0
C_TLZWE_PS_MEM_3 ***TLZWE of Bank 3***	0
C_TPACC_PS_FLASH_0 ***TPACC of Bank 0***	25000
C_TPACC_PS_FLASH_1 ***TPACC of Bank 1***	25000
C_TPACC_PS_FLASH_2 ***TPACC of Bank 2***	25000
C_TPACC_PS_FLASH_3 ***TPACC of Bank 3***	25000
C_TWC_PS_MEM_0	8000

TWC of Bank 0	
C_TWC_PS_MEM_1 ***TWC of Bank 1***	15000
C_TWC_PS_MEM_2 ***TWC of Bank 2***	15000
C_TWC_PS_MEM_3 ***TWC of Bank 3***	15000
C_TWP_PS_MEM_0 ***TWP of Bank 0***	8000
C_TWP_PS_MEM_1 ***TWP of Bank 1***	12000
C_TWP_PS_MEM_2 ***TWP of Bank 2***	12000
C_TWP_PS_MEM_3 ***TWP of Bank 3***	12000
C_XCL0_LINESIZE ***Cacheline Size of Ch0***	4
C_XCL0_WRITEXFER ***Write Transfer Type of Ch0***	1
C_XCL1_LINESIZE ***Cacheline Size of Ch1***	4
C_XCL1_WRITEXFER ***Write Transfer Type of Ch1***	1
C_XCL2_LINESIZE ***Cacheline Size of Ch2***	4
C_XCL2_WRITEXFER ***Write Transfer Type of Ch2***	1
C_XCL3_LINESIZE ***Cacheline Size of Ch3***	4
C_XCL3_WRITEXFER ***Write Transfer Type of Ch3***	1

3.11.4 Post Synthesis Device Utilization

	Used	**Total**	**Percentage**
Number of Slices	360	16640	2%
Number of Slice Flip Flops	479	33280	1%
Number of 4 input LUTs	353	33280	1%
Number of IOs	627		
Number of bonded IOBs	0	519	0%

3.11.5 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: 10.474ns (Maximum Frequency: 95.474MHz)

Minimum input arrival time before clock: 2.931ns

Maximum output required time after clock: 1.829ns

Maximum combinational path delay: No path found

3.12 dcm_module_0 Digital Clock Manager (DCM)

The digital clock manager module is a wrapper around the DCM primitive which allows it to be used in the EDK tool suite. Port List

These are the ports listed in the MHS file.

Port List				
#	NAME	DIR	[LSB:MSB]	SIGNAL
0	RST	**I**	**1**	sys_rst_s
1	CLKIN	**I**	**1**	dcm_clk_s
2	CLK90	**O**	**1**	clk_125_0000MHz90DCM0
3	CLK0	**O**	**1**	clk_125_0000MHzDCM0
4	CLKDV	**O**	**1**	clk_62_5000MHz
5	CLKFB	**I**	**1**	clk_125_0000MHzDCM0
6	LOCKED	**O**	**1**	Dcm_0_locked
7	CLKFX	**O**	**1**	clk_20MHz_int

3.12.1 Parameters

These are the current parameter settings for this module.

Name	Value	Name	Value
C_FAMILY ***Device Family***	spartan3adsp	C_CLKIN_BUF ***Insert a BUFG for CLKIN***	FALSE
C_CLK0_BUF ***Insert a BUFG for CLK0***	TRUE	C_CLKIN_DIVIDE_BY_2 ***CLKIN Divide By 2***	FALSE
C_CLK180_BUF ***Insert a BUFG for CLK180***	FALSE	C_CLKIN_PERIOD ***Input Clock Period***	8.00000
C_CLK270_BUF ***Insert a BUFG for CLK270***	FALSE	C_CLKOUT_PHASE_SHIFT ***Controls Use of Phase Shift***	NONE
C_CLK2X180_BUF ***Insert a BUFG for CLK2X180***	FALSE	C_CLK_FEEDBACK ***Clock Feedback Input***	1X
C_CLK2X_BUF ***Insert a BUFG for CLK2X***	FALSE	C_DESKEW_ADJUST ***Amount of Delay in the Feedback Path***	SYSTEM_SYNCHRONOUS
C_CLK90_BUF ***Insert a BUFG for CLK90***	TRUE	C_DFS_FREQUENCY_MODE ***Digital Frequency Synthesizer Clock Frequency Mode***	LOW
C_CLKDV_BUF ***Insert a BUFG for CLKDV***	TRUE	C_DLL_FREQUENCY_MODE ***Delay Locked Loop Frequency Mode***	LOW
C_CLKDV_DIVIDE ***CLKDV Divisor***	2.0	C_DSS_MODE ***DSS Mode***	NONE
C_CLKFB_BUF ***Insert a BUFG for CLKFB***	FALSE	C_DUTY_CYCLE_CORRECTION ***Duty Cycle Correction***	TRUE
C_CLKFX180_BUF ***Insert a BUFG for CLKFX180***	FALSE	C_EXT_RESET_HIGH ***Reset Polarity***	0
C_CLKFX_BUF ***Insert a BUFG for CLKFX***	TRUE	C_PHASE_SHIFT ***Phase Shift***	0
C_CLKFX_DIVIDE ***Divisor for the CLKFX***	25	C_STARTUP_WAIT ***Configuration Startup Wait***	FALSE

Output			
C_CLKFX_MULTIPLY ***Multiply Value of the CLKFX Output***	4		

3.12.2 Post Synthesis Device Utilization

	Used	**Total**	**Percentage**
Number of Slices	2	16640	0
Number of Slice Flip Flops	4	33280	0
Number of 4 input LUTs	1	33280	0
Number of IOs	26	NA	NA
Number of bonded IOBs	0	519	0
Number of GCLKs	4	24	16
DCMs	1	8	12

3.12.3 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: 1.263ns (Maximum Frequency: 791.766MHz)

Minimum input arrival time before clock: No path found

Maximum output required time after clock: No path found

Maximum combinational path delay: No path found

3.13 dcm_module_1 Digital Clock Manager (DCM)

The digital clock manager module is a wrapper around the DCM primitive which allows it to be used in the EDK tool suite.

3.13.1 Port List

These are the ports listed in the MHS file.

Port List				
#	NAME	DIR	[LSB:MSB]	SIGNAL
0	CLKIN	**I**	**1**	clk_20MHz_int
1	CLK0	**O**	**1**	clk_20MHz
2	CLKDV	**O**	**1**	clk_10MHz
3	CLKFB	**I**	**1**	clk_20MHz
4	LOCKED	**O**	**1**	Dcm_1_locked
5	RST	**I**	**1**	Dcm_0_locked
6	CLKFX	**O**	**1**	clk_16MHz
7	CLKFX180	**O**	**1**	clk_16MHz_180
8	CLK180	**O**	**1**	rx_clk_180_out
9	CLK270	**O**	**1**	clk_20MHz_270

3.13.2 Parameters

These are the current parameter settings for this module.

Name	Value	Name	Value
C_FAMILY ***Device Family***	spartan3adsp	C_CLKIN_BUF ***Insert a BUFG for CLKIN***	FALSE
C_CLK0_BUF ***Insert a BUFG for CLK0***	TRUE	C_CLKIN_DIVIDE_BY_2 ***CLKIN Divide By 2***	FALSE
C_CLK180_BUF ***Insert a BUFG for CLK180***	FALSE	C_CLKIN_PERIOD ***Input Clock Period***	50.00000
C_CLK270_BUF ***Insert a BUFG for CLK270***	TRUE	C_CLKOUT_PHASE_SHIFT ***Controls Use of Phase Shift***	NONE
C_CLK2X180_BUF ***Insert a BUFG for CLK2X180***	FALSE	C_CLK_FEEDBACK ***Clock Feedback Input***	1X
C_CLK2X_BUF ***Insert a BUFG for CLK2X***	FALSE	C_DESKEW_ADJUST ***Amount of Delay in the Feedback Path***	SYSTEM_SYNCHRONOUS
C_CLK90_BUF ***Insert a BUFG for CLK90***	FALSE	C_DFS_FREQUENCY_MODE ***Digital Frequency Synthesizer Clock Frequency Mode***	LOW
C_CLKDV_BUF ***Insert a BUFG for CLKDV***	TRUE	C_DLL_FREQUENCY_MODE ***Delay Locked Loop Frequency Mode***	LOW
C_CLKDV_DIVIDE ***CLKDV Divisor***	2.0	C_DSS_MODE ***DSS Mode***	NONE
C_CLKFB_BUF ***Insert a BUFG for CLKFB***	FALSE	C_DUTY_CYCLE_CORRECTION ***Duty Cycle Correction***	TRUE
C_CLKFX180_BUF ***Insert a BUFG for CLKFX180***	FALSE	C_EXT_RESET_HIGH ***Reset Polarity***	0
C_CLKFX_BUF ***Insert a BUFG for CLKFX***	TRUE	C_PHASE_SHIFT ***Phase Shift***	0
C_CLKFX_DIVIDE ***Divisor for the CLKFX Output***	5	C_STARTUP_WAIT ***Configuration Startup Wait***	FALSE
C_CLKFX_MULTIPLY ***Multiply Value of the CLKFX Output***	4		

3.13.3 Post Synthesis Device Utilization

	Used	Total	Percentage
Number of Slices	2	16640	0
Number of Slice Flip Flops	4	33280	0
Number of 4 input LUTs	1	33280	0
Number of IOs	26	NA	NA
Number of bonded IOBs	0	519	0
Number of GCLKs	4	24	16
DCMs	1	8	12

3.13.4 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: 1.263ns (Maximum Frequency: 791.766MHz)

Minimum input arrival time before clock: No path found

Maximum output required time after clock: No path found

Maximum combinational path delay: No path found

3.14 proc_sys_reset_0

proc_sys_reset_0 Processor System Reset Module

3.14.1 Port List

These are the ports listed in the MHS file. Please refer to the IP documentation for complete information about module ports.

Port List				
#	NAME	DIR	[LSB:MSB]	SIGNAL
0	Slowest_sync_clk	**I**	**1**	clk_62_5000MHz
1	Ext_Reset_In	**I**	**1**	sys_rst_s
2	MB_Debug_Sys_Rst	**I**	**1**	Debug_SYS_Rst
3	Dcm_locked	**I**	**1**	Dcm_1_locked
4	MB_Reset	**O**	**1**	mb_reset
5	Bus_Struct_Reset	**O**	**1**	sys_bus_reset
6	Peripheral_Reset	**O**	**1**	sys_periph_reset

3.14.2 Parameters

These are the current parameter settings for this module.

Name	Value	Name	Value
C_FAMILY ***Device Family***	spartan3adsp	C_EXT_RST_WIDTH ***Number of Clocks Before Input Change is Recognized On The External Reset Input***	4
C_AUX_RESET_HIGH ***Auxiliary Reset Active High***	1	C_NUM_BUS_RST ***Number of Bus Structure Reset Registered Outputs***	1
C_AUX_RST_WIDTH ***Number of Clocks Before Input Change is Recognized On The Auxiliary Reset Input***	4	C_NUM_PERP_RST ***Number of Peripheral Reset Registered Outputs***	1
C_EXT_RESET_HIGH ***External Reset Active High***	0		

3.14.3 Post Synthesis Device Utilization

	Used	Total	Percentage
Number of Slices	41	16640	0

Number of Slice Flip Flops	67	33280	0
Number of 4 input LUTs	52	33280	0
Number of IOs	20	NA	NA
Number of bonded IOBs	0	519	0

3.14.4 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: 4.818ns (Maximum Frequency: 207.555MHz)

Minimum input arrival time before clock: 1.320ns

Maximum output required time after clock: 0.591ns

Maximum combinational path delay: No path found

3.15 Interrupt Controller

xps_intc_0 XPS Interrupt Controller
intc core attached to the PLBV46

3.15.1 Port List

These are the ports listed in the MHS file. Please refer to the IP documentation for complete information about module ports.

Port List				
#	**NAME**	**DIR**	**[LSB:MSB]**	**SIGNAL**
0	Intr	**I**	**1**	mb_plb_Bus_Error_Det & fit_timer_0_Interrupt & xps_ethernetlite_0_IP2INTC_Irpt & xps_uartlite_0_Interrupt & xps_spi_0_IP2INTC_Irpt & xps_spi_1_IP2INTC_Irpt & xps_iic_0_IIC2INTC_Irpt & xps_gpio_0_IP2INTC_Irpt
1	Irq	**O**	**1**	microblaze_0_INTERRUPT

3.15.2 Bus Interfaces

Bus Interfaces				
NAME	TYPE	BUSSTD	BUS	Point 2 Point
SPLB	SLAVE	PLBV46	mb_plb	microblaze_0

3.15.3 Interrupt Priorities

PRIORITY	**SIGNAL**	**INSTANCE**
0	mb_plb_Bus_Error_Det	mb_plb
1	fit_timer_0_Interrupt	fit_timer_0

2	xps_ethernetlite_0_IP2INTC_Irpt	Ethernet_MAC
3	xps_uartlite_0_Interrupt	xps_uartlite_0
4	xps_spi_0_IP2INTC_Irpt	xps_spi_0
5	xps_spi_1_IP2INTC_Irpt	xps_spi_1
6	xps_iic_0_IIC2INTC_Irpt	xps_iic_0
7	xps_gpio_0_IP2INTC_Irpt	xps_gpio_0

3.15.4 Parameters

These are the current parameter settings for this module.

Name	Value	Name	Value
C_FAMILY	spartan3adsp	C_KIND_OF_INTR	0xFFFFFFFF
C_BASEADDR	0x81C00000	C_KIND_OF_LVL	0xFFFFFFFF
C_HIGHADDR	0x81C0001F	C_NUM_INTR_INPUTS	2
C_HAS_CIE	1	C_SPLB_AWIDTH	32
C_HAS_IPR	1	C_SPLB_DWIDTH	32
C_HAS_IVR	1	C_SPLB_MID_WIDTH	1
C_HAS_SIE	1	C_SPLB_NATIVE_DWIDTH	32
C_IRQ_ACTIVE	1	C_SPLB_NUM_MASTERS	1
C_IRQ_IS_LEVEL	1	C_SPLB_P2P	0
C_KIND_OF_EDGE	0xFFFFFFFF	C_SPLB_SUPPORT_BURSTS	0

3.15.5 Post Synthesis Device Utilization

	Used	Total	Percentage
Number of Slices	135	16640	0%
Number of Slice Flip Flops	196	33280	0%
Number of 4 input LUTs	150	33280	0%
Number of IOs	0	519	0%

3.15.6 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: 5.538ns (Maximum Frequency: 180.571MHz)

Minimum input arrival time before clock: 3.223ns

Maximum output required time after clock: 0.591ns

Maximum combinational path delay: No path found

3.16 xps_timebase_wdt_0

xps_timebase_wdt_0 XPS Watchdog Timer
Watchdog Timer with PLBV46 interface

3.16.1 Port List

These are the ports listed in the MHS file. Please refer to the IP documentation for complete information about module ports.

Port List				
#	NAME	DIR	[LSB:MSB]	SIGNAL
0	WDT_Reset	**O**	**1**	xps_timebase_wdt_0_WDT_Reset
1	Timebase_Interrupt	**O**	**1**	xps_timebase_wdt_0_Timebase_Interrupt
2	WDT_Interrupt	**O**	**1**	xps_timebase_wdt_0_WDT_Interrupt

3.16.2 Bus Interfaces

Bus Interfaces				
NAME	TYPE	BUSSTD	BUS	Point 2 Point
SPLB	SLAVE	PLBV46	mb_plb	microblaze_0

3.16.3 Parameters

These are the current parameter settings for this module.

Name	Value	Name	Value
C_FAMILY ***Device Family***	spartan3adsp	C_SPLB_NATIVE_DWIDTH ***Native Data Bus Width of PLB Slave***	32
C_BASEADDR ***Base Address***	0x81A00000	C_SPLB_NUM_MASTERS ***Number of PLB Masters***	2
C_HIGHADDR ***High Address***	0x81A0000F	C_SPLB_P2P ***PLB Slave Uses P2P Topology***	0
C_SPLB_AWIDTH ***PLB Address Bus Width***	32	C_SPLB_SUPPORT_BURSTS ***PLB Slave is Capable of Bursts***	0
C_SPLB_DWIDTH ***PLB Data Bus Width***	32	C_WDT_ENABLE_ONCE ***WDT Can Only Be Enabled Once***	1
C_SPLB_MID_WIDTH ***Master ID Bus Width of PLB***	1	C_WDT_INTERVAL ***The Exponent for Setting the Length of WDT Interval***	31

3.16.4 Post Synthesis Device Utilization

	Used	Total	Percentage
Number of Slices	112	16640	0
Number of Slice Flip Flops	166	33280	0
Number of 4 input LUTs	134	33280	0
Number of IOs	204	NA	NA
Number of bonded IOBs	0	519	0

3.16.5 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: 5.977ns (Maximum Frequency: 167.308MHz)

Minimum input arrival time before clock: 3.223ns

Maximum output required time after clock: 2.015ns

Maximum combinational path delay: No path found

3.17 xps_timer_0

xps_timer_0   XPS Timer/Counter
Timer counter with PLBV46 interface I P Specs

3.17.1 Port List

These are the ports listed in the MHS file. Please refer to the IP documentation for complete information about module ports.

Port List				
#	NAME	DIR	[LSB:MSB]	SIGNAL
0	Interrupt	**O**	**1**	xps_timer_0_Interrupt

3.17.2 Bus Interfaces

Bus Interfaces				
NAME	TYPE	BUSSTD	BUS	Point 2 Point
SPLB	SLAVE	PLBV46	mb_plb	microblaze_0

3.17.3 Parameters

These are the current parameter settings for this module.

Name	Value	Name	Value
C_FAMILY ***Device Family***	spartan3adsp	C_SPLB_DWIDTH ***PLB Data Bus Width***	32
C_BASEADDR ***Base Address***	0x83C00000	C_SPLB_MID_WIDTH ***Master ID Bus Width of PLB***	1
C_HIGHADDR ***High Address***	0x83C0FFFF	C_SPLB_NATIVE_DWIDTH ***Native Data Bus Width of PLB Slave***	32
C_COUNT_WIDTH ***The Width of Counter in Timer***	32	C_SPLB_NUM_MASTERS ***Number of PLB Masters***	2
C_GEN0_ASSERT ***GEN0 Active Level***	1	C_SPLB_P2P ***PLB Slave Uses P2P Topology***	0
C_GEN1_ASSERT ***GEN1 Active Level***	1	C_SPLB_SUPPORT_BURSTS ***PLB Slave is Capable of Bursts***	0
C_ONE_TIMER_ONLY ***Only One Timer is present***	0	C_TRIG0_ASSERT ***TRIG0 Active Level***	1
C_SPLB_AWIDTH ***PLB Address Bus Width***	32	C_TRIG1_ASSERT ***TRIG1 Active Level***	1

3.17.4 Post Synthesis Device Utilization

	Used	Total	Percentage

Number of Slices	321	16640	1
Number of Slice Flip Flops	358	33280	1
Number of 4 input LUTs	365	33280	1
Number of IOs	208	NA	NA
Number of bonded IOBs	0	519	0

3.17.5 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: 7.956ns (Maximum Frequency: 125.691MHz)

Minimum input arrival time before clock: 2.922ns

Maximum output required time after clock: 0.591ns

Maximum combinational path delay: No path found

3.18 fit_timer_0 Fixed Interval Timer

This timer generates an interrupt periodically.

3.18.1 Port List

These are the ports listed in the MHS file.

Port List				
#	NAME	DIR	[LSB:MSB]	SIGNAL
0	Interrupt	**O**	**1**	fit_timer_0_Interrupt
1	Rst	**I**	**1**	mb_reset
2	Clk	**I**	**1**	clk_62_5000MHz

3.18.2 Parameters

These are the current parameter settings for this module.

Name	**Value**
C_FAMILY ***Device Family***	spartan3adsp
C_EXT_RESET_HIGH ***External Reset Level***	1
C_INACCURACY ***Allowed Inaccuracy in The Number of Clock Cycles Between Strobes***	0
C_NO_CLOCKS ***Number of Clocks Between Strobes***	125000

3.18.3 Post Synthesis Device Utilization

	Used	**Total**	**Percentage**

Number of Slices	10	16640	0
Number of Slice Flip Flops	11	33280	0
Number of 4 input LUTs	24	33280	0
Number of IOs	3	NA	NA
Number of bonded IOBs	0	519	0

3.18.4 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: 5.913ns (Maximum Frequency: 169.119MHz)

Minimum input arrival time before clock: 1.799ns

Maximum output required time after clock: 0.591ns

Maximum combinational path delay: No path found

3.19 Ethernet_MAC

Ethernet_MAC XPS 10/100 Ethernet MAC Lite
'IEEE Std. 802.3 MII interface MAC with PLBV46 interface, lightweight implementation'.

Ethernet is used as the communication link between the embedded system and the remote control PC. The Ethernet MAC controller IP core is implemented in the FPGA to interface the Ethernet PHY hardware device.

3.19.1 Port List

These are the ports listed in the MHS file. Please refer to the IP documentation for complete information about module ports.

Port List				
#	NAME	DIR	[LSB:MSB]	SIGNAL
0	PHY_tx_clk	**I**	**1**	fpga_0_Ethernet_MAC_PHY_tx_clk_pin
1	PHY_rx_clk	**I**	**1**	fpga_0_Ethernet_MAC_PHY_rx_clk_pin
2	PHY_crs	**I**	**1**	fpga_0_Ethernet_MAC_PHY_crs_pin
3	PHY_dv	**I**	**1**	fpga_0_Ethernet_MAC_PHY_dv_pin
4	PHY_rx_data	**I**	**0:3**	fpga_0_Ethernet_MAC_PHY_rx_data_pin
5	PHY_col	**I**	**1**	fpga_0_Ethernet_MAC_PHY_col_pin
6	PHY_rx_er	**I**	**1**	fpga_0_Ethernet_MAC_PHY_rx_er_pin
7	PHY_rst_n	**O**	**1**	fpga_0_Ethernet_MAC_PHY_rst_n_pin
8	PHY_tx_en	**O**	**1**	fpga_0_Ethernet_MAC_PHY_tx_en_pin
9	PHY_tx_data	**O**	**0:3**	fpga_0_Ethernet_MAC_PHY_tx_data_pin
10	PHY_MDC	**O**	**1**	fpga_0_Ethernet_MAC_PHY_MDC_pin
11	IP2INTC_Irpt	**O**	**1**	xps_ethernetlite_0_IP2INTC_Irpt
12	PHY_MDIO	**IO**	**1**	fpga_0_Ethernet_MAC_PHY_MDIO_pin

3.19.2 Bus Interfaces

Bus Interfaces				
NAME	**TYPE**	**BUSSTD**	**BUS**	**Point 2 Point**
SPLB	SLAVE	PLBV46	mb_plb	microblaze_0

3.19.3 Parameters

These are the current parameter settings for this module.

Name	Value	Name	Value
C_FAMILY ***Device Family***	spartan3adsp	C_SPLB_DWIDTH ***PLB Data Bus Width***	32
C_BASEADDR ***Base Address***	0x80000000	C_SPLB_MID_WIDTH ***Master ID Bus Width of PLB***	1
C_HIGHADDR ***High Address***	0x8000FFFF	C_SPLB_NATIVE_DWIDTH ***Native Data Bus Width of PLB Slave***	32
C_DUPLEX ***Duplex Mode***	1	C_SPLB_NUM_MASTERS ***Number of PLB Masters***	2
C_INCLUDE_INTERNAL_LOOPBACK ***Include Internal Loopback***	0	C_SPLB_P2P ***PLB Slave Uses P2P Topology***	0
C_INCLUDE_MDIO ***Include MII Management Module***	1	C_SPLB_SMALLEST_MASTER ***Smallest Master Data Bus Width***	32
C_RX_PING_PONG ***Include Second Receiver Buffer***	1	C_SPLB_SUPPORT_BURSTS ***PLB Slave is Capable of Bursts***	1
C_SPLB_AWIDTH ***PLB Address Bus Width***	32	C_TX_PING_PONG ***Include Second Transmitter Buffer***	1
C_SPLB_CLK_PERIOD_PS ***Clock Period of PLB Slave***	16000		

3.19.4 Post Synthesis Device Utilization

	Used	Total	Percentage
Number of Slices	656	16640	3
Number of Slice Flip Flops	737	33280	2
Number of 4 input LUTs	1101	33280	3
Number of IOs	222	NA	NA
Number of bonded IOBs	0	519	0
BRAMs	4	84	4

3.19.5 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: 9.326ns (Maximum Frequency: 107.227MHz)

Minimum input arrival time before clock: 4.294ns

Maximum output required time after clock: 4.294ns

Maximum combinational path delay: 0.648ns

3.20 xps_gpio_0

XPS General Purpose IO
General Purpose Input/Output (GPIO) core for the PLBV46 bus.

3.20.1 Port List

These are the ports listed in the MHS file. Please refer to the IP documentation for complete information about module ports.

Port List				
#	NAME	DIR	[LSB:MSB]	SIGNAL
0	GPIO_IO_I	**I**	**1**	xps_gpio_0_GPIO_IO_I
1	IP2INTC_Irpt	**O**	**1**	xps_gpio_0_IP2INTC_Irpt
2	GPIO2_IO_O	**O**	**1**	xps_gpio_0_GPIO2_IO_O

3.20.2 Parameters

These are the current parameter settings for this module.

Name	Value	Name	Value
C_FAMILY ***Device Family***	spartan3adsp	C_IS_DUAL ***Enable Channel 2***	1
C_BASEADDR ***Base Address***	0x81B00000	C_SPLB_AWIDTH ***PLB Address Bus Width***	32
C_HIGHADDR ***High Address***	0x81B001FF	C_SPLB_DWIDTH ***PLB Data Bus Width***	32
C_ALL_INPUTS ***Channel 1 is Input Only***	1	C_SPLB_MID_WIDTH ***Master ID Bus Width of PLB***	1
C_ALL_INPUTS_2 ***Channel 2 is Input Only***	0	C_SPLB_NATIVE_DWIDTH ***Native Data Bus Width of PLB Slave***	32
C_DOUT_DEFAULT ***Channel 1 Data Out Default Value***	0x00000000	C_SPLB_NUM_MASTERS ***Number of PLB Masters***	2
C_DOUT_DEFAULT_2 ***Channel 2 Data Out Default Value***	0x00000000	C_SPLB_P2P ***PLB Slave Uses P2P Topology***	0
C_GPIO2_WIDTH ***GPIO2 Data Channel Width***	1	C_SPLB_SUPPORT_BURSTS ***PLB Slave is Capable of Bursts***	0
C_GPIO_WIDTH ***GPIO Data Channel Width***	1	C_TRI_DEFAULT ***Channel 1 Tri-state Default Value***	0xFFFFFFFF
C_INTERRUPT_PRESENT ***GPIO Supports Interrupts***	1	C_TRI_DEFAULT_2 ***Channel 2 Tri-state Default Value***	0xFFFFFFFF

3.20.3 Post Synthesis Device Utilization

	Used	Total	Percentage
Number of Slices	86	16640	0
Number of Slice Flip Flops	115	33280	0
Number of 4 input LUTs	99	33280	0
Number of IOs	208	NA	NA

Number of bonded IOBs	0	519	0

3.20.4 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: 4.974ns (Maximum Frequency: 201.045MHz)

Minimum input arrival time before clock: 2.971ns

Maximum output required time after clock: 2.189ns

Maximum combinational path delay: No path found

3.21 xps_gpio_1

XPS General Purpose IO
General Purpose Input/Output (GPIO) core for the PLBV46 bus.

3.21.1 Port List

These are the ports listed in the MHS file. Please refer to the IP documentation for complete information about module ports.

Port List				
#	NAME	DIR	[LSB:MSB]	SIGNAL
0	GPIO_IO_O	**I**	**0:5**	xps_gpio_0_GPIO_IO_O

3.21.2 Bus Interfaces

Bus Interfaces				
NAME	TYPE	BUSSTD	BUS	Point 2 Point
SPLB	SLAVE	PLBV46	mb_plb	microblaze_0

3.21.3 Parameters

These are the current parameter settings for this module.

Name	**Value**	**Name**	**Value**
C_FAMILY ***Device Family***	spartan3adsp	C_IS_DUAL ***Enable Channel 2***	0
C_BASEADDR ***Base Address***	0x81B01000	C_SPLB_AWIDTH ***PLB Address Bus Width***	32
C_HIGHADDR ***High Address***	0x81B011FF	C_SPLB_DWIDTH ***PLB Data Bus Width***	32
C_ALL_INPUTS ***Channel 1 is Input Only***	0	C_SPLB_MID_WIDTH ***Master ID Bus Width of PLB***	1
C_ALL_INPUTS_2 ***Channel 2 is Input Only***	0	C_SPLB_NATIVE_DWIDTH ***Native Data Bus Width of PLB***	32

		Slave	
C_DOUT_DEFAULT ***Channel 1 Data Out Default Value***	0x00000000	C_SPLB_NUM_MASTERS ***Number of PLB Masters***	2
C_DOUT_DEFAULT_2 ***Channel 2 Data Out Default Value***	0x00000000	C_SPLB_P2P ***PLB Slave Uses P2P Topology***	0
C_GPIO2_WIDTH ***GPIO2 Data Channel Width***	32	C_SPLB_SUPPORT_BURSTS ***PLB Slave is Capable of Bursts***	0
C_GPIO_WIDTH ***GPIO Data Channel Width***	6	C_TRI_DEFAULT ***Channel 1 Tri-state Default Value***	0xFFFFFFFF
C_INTERRUPT_PRESENT ***GPIO Supports Interrupts***	1	C_TRI_DEFAULT_2 ***Channel 2 Tri-state Default Value***	0xFFFFFFFF

3.21.4 Post Synthesis Device Utilization

	Used	**Total**	**Percentage**
Number of Slices	71	16640	0
Number of Slice Flip Flops	111	33280	0
Number of 4 input LUTs	64	33280	0
Number of IOs	316	NA	NA
Number of bonded IOBs	0	519	0

3.21.5 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: 4.815ns (Maximum Frequency: 207.684MHz)

Minimum input arrival time before clock: 2.329ns

Maximum output required time after clock: 0.591ns

Maximum combinational path delay: No path found

3.22 xps_iic_0

xps_iic_0 XPS IIC Interface
PLBV46 interface to Philips I2C bus v2.1.

These are the ports listed in the MHS file. Please refer to the IP documentation for complete information about module ports.

Port List				
#	NAME	DIR	[LSB:MSB]	SIGNAL
0	Scl	**IO**	**1**	xps_iic_0_Scl
1	Sda	**IO**	**1**	xps_iic_0_Sda
2	IIC2INTC_Irpt	**O**	**1**	xps_iic_0_IIC2INTC_Irpt

3.22.1 Bus Interfaces

Bus Interfaces				
NAME	TYPE	BUSSTD	BUS	Point 2 Point
SPLB	SLAVE	PLBV46	mb_plb	microblaze_0

3.22.2 Parameters

These are the current parameter settings for this module.

Name	Value	Name	Value
C_FAMILY ***Device Family***	spartan3adsp	C_SDA_INERTIAL_DELAY ***Width of glitches removed on SDA input***	0
C_BASEADDR ***Base Address***	0x81800000	C_SPLB_AWIDTH ***PLB Address Bus Width***	32
C_HIGHADDR ***High Address***	0x818001FF	C_SPLB_DWIDTH ***PLB Data Bus Width***	32
C_CLK_FREQ ***PLBv46 Bus Clock Frequency***	62500000	C_SPLB_MID_WIDTH ***Master ID Bus Width of PLB***	1
C_GPO_WIDTH ***Width of GPIO***	1	C_SPLB_NATIVE_DWIDTH ***Native Data Bus Width of PLB Slave***	32
C_IIC_FREQ ***Output Frequency of SCL Signal***	400000	C_SPLB_NUM_MASTERS ***Number of PLB Masters***	2
C_SCL_INERTIAL_DELAY ***Width of glitches removed on SCL input***	0	C_TEN_BIT_ADR ***Use 10-bit Address***	0

3.22.3 Post Synthesis Device Utilization

	Used	Total	Percentage
Number of Slices	322	16640	1
Number of Slice Flip Flops	379	33280	1
Number of 4 input LUTs	479	33280	1
Number of IOs	209	NA	NA
Number of bonded IOBs	0	519	0

3.22.4 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: 7.879ns (Maximum Frequency: 126.926MHz)

Minimum input arrival time before clock: 5.571ns

Maximum output required time after clock: 3.332ns

Maximum combinational path delay: No path found

3.23 xps_spi_0

xps_spi_0 XPS SPI Interface
PLBV46 to Motorola Serial Peripheral Interface (SPI) adapter.

3.23.1 Port List

These are the ports listed in the MHS file. Please refer to the IP documentation for complete information about module ports.

Port List				
#	NAME	DIR	[LSB:MSB]	SIGNAL
0	SCK	**IO**	**1**	xps_spi_0_SCK
1	MISO	**IO**	**1**	xps_spi_0_MISO
2	MOSI	**IO**	**1**	xps_spi_0_MOSI
3	SS	**IO**	**1**	xps_spi_0_SS
4	IP2INTC_Irpt	**O**	**1**	xps_spi_0_IP2INTC_Irpt

3.23.2 Bus Interfaces

Bus Interfaces				
NAME	TYPE	BUSSTD	BUS	Point 2 Point
SPLB	SLAVE	PLBV46	mb_plb	microblaze_0

3.23.3 Parameters

These are the current parameter settings for this module.

Name	Value	Name	Value
C_FAMILY ***Device Family***	spartan3adsp	C_SPLB_AWIDTH ***PLB Address Bus Width***	32
C_BASEADDR ***Base Address***	0x81400000	C_SPLB_DWIDTH ***PLB Data Bus Width***	32
C_HIGHADDR ***High Address***	0x8140007F	C_SPLB_MID_WIDTH ***Master ID Bus Width of PLB***	1
C_FIFO_EXIST ***Include both Receiver and Transmitter FIFOs***	1	C_SPLB_NATIVE_DWIDTH ***Native Data Bus Width of PLB Slave***	32
C_NUM_SS_BITS ***Total Number of Slave Select Bits in SS Vector***	1	C_SPLB_NUM_MASTERS ***Number of PLB Masters***	2
C_NUM_TRANSFER_BITS ***Number of SPI transfer bits***	8	C_SPLB_P2P ***PLB Slave Uses P2P Topology***	0
C_SCK_RATIO ***Ratio of PLB Clock Frequency To SCK Frequency***	2	C_SPLB_SUPPORT_BURSTS ***PLB Slave is Capable of Bursts***	0

3.23.4 Post Synthesis Device Utilization

	Used	**Total**	**Percentage**
Number of Slices	197	16640	1
Number of Slice Flip Flops	242	33280	0
Number of 4 input LUTs	285	33280	0

Number of IOs	215	NA	NA
Number of bonded IOBs	0	519	0

3.23.5 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: 8.425ns (Maximum Frequency: 118.697MHz)

Minimum input arrival time before clock: 6.902ns

Maximum output required time after clock: 3.264ns

Maximum combinational path delay: No path found

3.24 xps_spi_1

xps_spi_1 XPS SPI Interface
PLBV46 to Motorola Serial Peripheral Interface (SPI) adapter.

3.24.1 Port List

These are the ports listed in the MHS file. Please refer to the IP documentation for complete information about module ports.

Port List				
#	NAME	DIR	[LSB:MSB]	SIGNAL
0	SCK	**IO**	**1**	xps_spi_1_SCK
1	MISO	**IO**	**1**	xps_spi_1_MISO
2	MOSI	**IO**	**1**	xps_spi_1_MOSI
3	SS	**IO**	**0:1**	xps_spi_1_SS
4	IP2INTC_Irpt	**O**	**1**	xps_spi_1_IP2INTC_Irpt

3.24.2 Bus Interfaces

Bus Interfaces				
NAME	TYPE	BUSSTD	BUS	Point 2 Point
SPLB	SLAVE	PLBV46	mb_plb	microblaze_0

3.24.3 Parameters

These are the current parameter settings for this module.

Name	Value	Name	Value
C_FAMILY ***Device Family***	spartan3adsp	C_SPLB_AWIDTH ***PLB Address Bus Width***	32
C_BASEADDR ***Base Address***	0x81401000	C_SPLB_DWIDTH ***PLB Data Bus Width***	32
C_HIGHADDR	0x8140107F	C_SPLB_MID_WIDTH	1

High Address		***Master ID Bus Width of PLB***	
C_FIFO_EXIST ***Include both Receiver and Transmitter FIFOs***	1	C_SPLB_NATIVE_DWIDTH ***Native Data Bus Width of PLB Slave***	32
C_NUM_SS_BITS ***Total Number of Slave Select Bits in SS Vector***	2	C_SPLB_NUM_MASTERS ***Number of PLB Masters***	2
C_NUM_TRANSFER_BITS ***Number of SPI transfer bits***	16	C_SPLB_P2P ***PLB Slave Uses P2P Topology***	0
C_SCK_RATIO ***Ratio of PLB Clock Frequency To SCK Frequency***	32	C_SPLB_SUPPORT_BURSTS ***PLB Slave is Capable of Bursts***	0

3.24.4 Post Synthesis Device Utilization

	Used	**Total**	**Percentage**
Number of Slices	266	16640	1
Number of Slice Flip Flops	311	33280	0
Number of 4 input LUTs	365	33280	1
Number of IOs	217	NA	NA
Number of bonded IOBs	0	519	0

3.24.5 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: 7.536ns (Maximum Frequency: 132.696MHz)

Minimum input arrival time before clock: 5.906ns

Maximum output required time after clock: 3.264ns

Maximum combinational path delay: No path found

3.25 xps_uartlite_0

xps_uartlite_0 XPS UART (Lite)
Generic UART (Universal Asynchronous Receiver/Transmitter) for PLBV46 bus.

3.25.1 Port List

These are the ports listed in the MHS file. Please refer to the IP documentation for complete information about module ports.

Port List				
#	NAME	DIR	[LSB:MSB]	SIGNAL
0	RX	**I**	**1**	xps_uartlite_0_RX
1	TX	**O**	**1**	xps_uartlite_0_TX
2	Interrupt	**O**	**1**	xps_timer_0_Interrupt

3.25.2 Bus Interfaces

Bus Interfaces				
NAME	TYPE	BUSSTD	BUS	Point 2 Point
SPLB	SLAVE	PLBV46	mb_plb	microblaze_0

3.25.3 Parameters

These are the current parameter settings for this module.

Name	Value	Name	Value
C_FAMILY ***Device Family***	spartan3adsp	C_SPLB_DWIDTH ***PLB Data Bus Width***	32
C_BASEADDR ***Base Address***	0x81000000	C_SPLB_MID_WIDTH ***Master ID Bus Width of PLB***	1
C_HIGHADDR ***High Address***	0x8100007F	C_SPLB_NATIVE_DWIDTH ***Native Data Bus Width of PLB Slave***	32
C_BAUDRATE ***UART Lite Baud Rate***	9600	C_SPLB_NUM_MASTERS ***Number of PLB Masters***	2
C_DATA_BITS ***Number of Data Bits in a Serial Frame***	8	C_SPLB_P2P ***PLB Slave Uses P2P Topology***	0
C_ODD_PARITY ***Parity Type***	1	C_SPLB_SUPPORT_BURSTS ***PLB Slave is Capable of Bursts***	0
C_SPLB_AWIDTH ***PLB Address Bus Width***	32	C_USE_PARITY ***Use Parity***	1
C_SPLB_CLK_FREQ_HZ ***Clock Frequency of PLB Slave***	62500000		

3.25.4 Post Synthesis Device Utilization

	Used	Total	Percentage
Number of Slices	114	16640	0
Number of Slice Flip Flops	152	33280	0
Number of 4 input LUTs	142	33280	0
Number of IOs	204	NA	NA
Number of bonded IOBs	0	519	0

3.25.5 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: 6.371ns (Maximum Frequency: 156.961MHz)

Minimum input arrival time before clock: 3.223ns

Maximum output required time after clock: 0.591ns

Maximum combinational path delay: No path found

3.26 xps_uartlite_1

xps_uartlite_1 XPS UART (Lite)
Generic UART (Universal Asynchronous Receiver/Transmitter) for PLBV46 bus

3.26.1 Port List

These are the ports listed in the MHS file. Please refer to the IP documentation for complete information about module ports.

Port List				
#	NAME	DIR	[LSB:MSB]	SIGNAL
0	RX	**I**	**1**	xps_uartlite_1_RX
1	TX	**O**	**1**	xps_uartlite_1_TX

3.26.2 Bus Interfaces

Bus Interfaces				
NAME	TYPE	BUSSTD	BUS	Point 2 Point
SPLB	SLAVE	PLBV46	mb_plb	microblaze_0

3.26.3 Parameters

These are the current parameter settings for this module.

Name	Value	Name	Value
C_FAMILY ***Device Family***	spartan3adsp	C_SPLB_DWIDTH ***PLB Data Bus Width***	32
C_BASEADDR ***Base Address***	0x81001000	C_SPLB_MID_WIDTH ***Master ID Bus Width of PLB***	1
C_HIGHADDR ***High Address***	0x8100107F	C_SPLB_NATIVE_DWIDTH ***Native Data Bus Width of PLB Slave***	32
C_BAUDRATE ***UART Lite Baud Rate***	9600	C_SPLB_NUM_MASTERS ***Number of PLB Masters***	2
C_DATA_BITS ***Number of Data Bits in a Serial Frame***	8	C_SPLB_P2P ***PLB Slave Uses P2P Topology***	0
C_ODD_PARITY ***Parity Type***	1	C_SPLB_SUPPORT_BURSTS ***PLB Slave is Capable of Bursts***	0
C_SPLB_AWIDTH ***PLB Address Bus Width***	32	C_USE_PARITY ***Use Parity***	0
C_SPLB_CLK_FREQ_HZ ***Clock Frequency of PLB Slave***	62500000		

3.26.4 Post Synthesis Device Utilization

	Used	Total	Percentage
Number of Slices	108	16640	0
Number of Slice Flip Flops	144	33280	0

Number of 4 input LUTs	132	33280	0
Number of IOs	204	NA	NA
Number of bonded IOBs	0	519	0

3.26.5 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: 6.371ns (Maximum Frequency: 156.961MHz)

Minimum input arrival time before clock: 3.223ns

Maximum output required time after clock: 0.591ns

Maximum combinational path delay: No path found

3.27 Debugger

mdm_0 MicroBlaze Debug Module (MDM)

Debug module for MicroBlaze Soft Processor.

3.27.1 Port List

These are the ports listed in the MHS file. Please refer to the IP documentation for complete information about module ports.

Port List				
#	**NAME**	**DIR**	**[LSB:MSB]**	**SIGNAL**
0	Debug_SYS_Rst	**O**	**1**	Debug_SYS_Rst

3.27.2 Bus Interfaces

Bus Interfaces				
NAME	TYPE	BUSSTD	BUS	Point 2 Point
MBDEBUG_0	INITIATOR	XIL_MBDEBUG2	microblaze_0_mdm_bus	microblaze_0
SPLB	SLAVE	PLBV46	mb_plb	microblaze_0

3.27.3 Parameters

These are the current parameter settings for this module.

Name	Value	Name	Value
C_FAMILY	spartan3adsp	C_SPLB_DWIDTH	32
C_BASEADDR	0x84400000	C_SPLB_MID_WIDTH	3
C_HIGHADDR	0x8440FFFF	C_SPLB_NATIVE_DWIDTH	32
C_INTERCONNECT	1	C_SPLB_NUM_MASTERS	8
C_JTAG_CHAIN	2	C_SPLB_P2P	0
C_MB_DBG_PORTS	1	C_SPLB_SUPPORT_BURSTS	0

C_OPB_AWIDTH	32	C_UART_WIDTH	8
C_OPB_DWIDTH	32	C_USE_UART	1
C_SPLB_AWIDTH	32	C_WRITE_FSL_PORTS	0

3.27.4 Post Synthesis Device Utilization

	Used	**Total**	**Percentage**
Number of Slices	88	16640	0%
Number of Slice Flip Flops	119	33280	0%
Number of 4 input LUTs	147	33280	0%
Number used as logic	124		
Number used as Shift registers	23		
Number of IOs	498		
Number of bonded IOBs	0	519	0%
Number of GCLKs	2	24	8%

3.27.5 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: 8.848ns (Maximum Frequency: 113.020MHz)

Minimum input arrival time before clock: 5.419ns

Maximum output required time after clock: 7.574ns

Maximum combinational path delay: 2.669ns

4 Technical description of Acquisition Logic Blocks

4.1 TT4_Top_Level_Sequencer_top_0

4.1.1 Port List

These are the ports listed in the MHS file.

Port List				
#	NAME	DIR	[LSB:MSB]	SIGNAL
0	MResetHiIn	**I**	**1**	AcquisitionLogicResetHi
1	Clk10MHzIn	**I**	**1**	clk_10MHz
2	CollectionControlReg3In	**I**	**0:2**	CollectionControlReg3In
3	PRFPeriod10mSecReg10In	**I**	**0:9**	PRFPeriod10mSecReg10In
4	RequiredTXRXCyclesCount13RegIn	**I**	**0:12**	RequiredTXRXCyclesCount13RegIn
5	RequiredTRSwitchOnDelayCount8RegIn	**I**	**0:7**	RequiredTRSwitchOnDelayCount8RegIn
6	HTPSUsOPGoodHiIn	**I**	**1**	HTPSUsOPGoodHiIn
7	TXRunningHiIn	**I**	**1**	TXRunningHiIn
8	HTPSUOnHiOut	**O**	**1**	HTPSUOnHiOut
9	RXAmpPowerOnHiOut	**O**	**1**	RXAmpPowerOnHiOut
10	TXAmpPowerOnHiOut	**O**	**1**	TXAmpPowerOnHiOut
11	TXRXRunReqHiOut	**O**	**1**	TXRXRunReqHiOut
12	TXRXAbortReqHiOut	**O**	**1**	TXRXAbortReqHiOut
13	CollectionStatusReg8Out	**O**	**0:7**	CollectionStatusReg8Out
14	NumberOfTXRXCyclesCompletedCount13Out	**O**	**0:12**	NumberOfTXRXCyclesCompletedCount13Out
15	TRSwitchFastOnAtTXStartOut	**O**	**1**	TRSwitchFastOnAtTXStartOut
16	TRSwitchHoldOnAtTXStartOut	**O**	**1**	TRSwitchHoldOnAtTXStartOut
17	TRSwitchFastOnAtTXEndOut	**O**	**1**	TRSwitchFastOnAtTXEndOut
18	TRSwitchHoldOnAtTXEndOut	**O**	**1**	TRSwitchHoldOnAtTXEndOut
19	uBInterruptReqHiOut	**O**	**1**	uBInterruptReqHiOut
20	RXAmpLoadCPLDReqHiOut	**O**	**1**	LoadReqHiIn
21	Clk1MHzStrbOut	**O**	**1**	clk_1MHz
22	TXAmpsOvrCurrentLoIn	**I**	**1**	TXAmpsOvrCurrentLowIn
23	Clk100HzOut	**O**	**1**	Clk100Hz
24	AcqRunningHiOut	**O**	**1**	AcqRunningHi
25	RX_one_iteration_done	**I**	**1**	rx_ip_0_rx_one_iteration_done

4.1.2 Post Synthesis Device Utilization

	Used	**Total**	**Percentage**
Number of Slices	209	16640	1
Number of Slice Flip Flops	180	33280	0
Number of 4 input LUTs	375	33280	1
Number of IOs	76	NA	NA
Number of bonded IOBs	0	519	0
Number of GCLKs	2	24	8

4.1.3 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: 11.110ns (Maximum Frequency: 90.009MHz)

Minimum input arrival time before clock: 8.046ns

Maximum output required time after clock: 1.232ns

Maximum combinational path delay: No path found

4.2 tx_bram_reg20_0

Custom IP core.

TX Block RAM Format

24 user logic memory spaces are implemented in the tx_bram_reg20 IP core using BRAM resources. These memory spaces are allocated for storing 24-channel transmitter waveforms. Each memory space can store maximum 2048 transmit samples, each TX sample is 16-bit wide. All 24 memory spaces are contiguously mapped as the MicroBlaze memory address. Channel 1 is located at the base address, followed by channel 2, 3… up to channel 24. In each memory space, the first sample is located at the lowest address. MicroBlaze writes to these memory spaces sequentially, channel by channel.

Within each block, data will be read out to create the TX waveforms by starting from the lowest address and working upwards. The base address of each block in the PLB is the address of the location assigned to the lowest absolute address. The block occupying the lowest addresses will be fed to TX#1, the next block up to TX#2 etc.

Once Collection starts, PLB access to this memory will be blocked until it completes.

Data coding

Each location holds a 16 bit data word with its' least significant bit placed in bit 0 of the location. Data coding is linear with 0x0000 corresponding to a maximum negative TX output and 0xFFFF a maximum positive output.

Data Padding

It is the responsibility of the uB to ensure that all memory locations in all 24 buffer blocks contain valid data in all locations from offset address 0 up to the location holding the highest data word to be read out. If necessary additional mid-scale values of 0x8000 should be written into the TX blocks to achieve this situation.

The highest location to be read out is determined by the number loaded in the TX Word Count register.

4.2.1 Port List

These are the ports listed in the MHS file. Please refer to the IP documentation for complete information about module ports.

Port List

#	NAME	DIR	[LSB: MSB]	SIGNAL
0	CollectionControlReg	**O**	**0:2**	CollectionControlReg3In
1	PRFPeriodReg	**O**	**0:9**	PRFPeriod10mSecReg10In
2	RequiredTXRXCyclesCount13Reg	**O**	**0:12**	RequiredTXRXCyclesCount13RegIn
3	RequiredTRSwitchOnDelayCount8Reg	**O**	**0:7**	RequiredTRSwitchOnDelayCount8RegIn
4	NumberOfTXRXCyclesCompletedCount13Out	**I**	**0:12**	NumberOfTXRXCyclesCompletedCount13Out
5	CollectionStatusReg8Out	**I**	**0:7**	CollectionStatusReg8Out
6	TX_word_count_Reg	**O**	**0:10**	TX_word_count_Reg
7	RX_bandwidth_n_mode_reg	**O**	**0:5**	RXAmpFiltModeRegIn6
8	TX_Enables_Reg	**O**	**0:23**	TX_Enables_Reg
9	Misc_functions_reg	**O**	**0:7**	MiscellaniousFunctionsReg
10	RX_number_of_samples_reg	**O**	**0:16**	RX_number_of_samples_reg
11	clk_16MHz	**I**	**1**	clk_16MHz
12	reset	**I**	**1**	AcquisitionLogicResetHi
13	TX_SAMPLE_NUM	**I**	**0:10**	TX_word_count_Reg
14	tx_en	**I**	**1**	TXRXRunReqHiOut
15	sdo	**O**	**0:23**	tx_ip_0_sdo
16	bsb_n	**O**	**1**	tx_ip_0_bsb_n
17	osr2	**O**	**1**	tx_ip_0_osr2
18	osr1	**O**	**1**	tx_ip_0_osr1
19	rstb_n	**O**	**1**	tx_ip_0_rstb_n
20	fsnc	**O**	**1**	tx_ip_0_fsnc
21	TXRunningHi	**O**	**1**	TXRunningHiIn
22	TXRXAbortReqHi	**I**	**1**	TXRXAbortReqHiOut
23	muteb_n	**O**	**1**	TXAmpIPEnHi
24	RX_Digitiser_Sampling_Rate_reg	**O**	**0:2**	RX_Digitiser_Sampling_Rate_reg
25	FPGA_Version_reg	**I**	**0:31**	0b00000000000000000000000000000001
26	FPGA_Varient_reg	**I**	**0:31**	0b00000000000000000000000000000100

4.2.2 Bus Interfaces

Bus Interfaces				
NAME	**TYPE**	**BUSSTD**	**BUS**	**Point 2 Point**
SPLB	SLAVE	PLBV46	mb_plb	microblaze_0

4.2.3 Parameters

These are the current parameter settings for this module.

Name	Value	Name	Value
C_FAMILY	spartan3adsp	C_MEM22_BASEADDR	0xA202C000
C_BASEADDR	0xA0000000	C_MEM22_HIGHADDR	0xA202DFFF
C_HIGHADDR	0xA000FFFF	C_MEM23_BASEADDR	0xA202E000
C_MEM0_BASEADDR	0xA2000000	C_MEM23_HIGHADDR	0xA202FFFF
C_MEM0_HIGHADDR	0xA2001FFF	C_MEM2_BASEADDR	0xA2004000
C_MEM10_BASEADDR	0xA2014000	C_MEM2_HIGHADDR	0xA2005FFF
C_MEM10_HIGHADDR	0xA2015FFF	C_MEM3_BASEADDR	0xA2006000
C_MEM11_BASEADDR	0xA2016000	C_MEM3_HIGHADDR	0xA2007FFF
C_MEM11_HIGHADDR	0xA2017FFF	C_MEM4_BASEADDR	0xA2008000
C_MEM12_BASEADDR	0xA2018000	C_MEM4_HIGHADDR	0xA2009FFF
C_MEM12_HIGHADDR	0xA2019FFF	C_MEM5_BASEADDR	0xA200A000
C_MEM13_BASEADDR	0xA201A000	C_MEM5_HIGHADDR	0xA200BFFF
C_MEM13_HIGHADDR	0xA201BFFF	C_MEM6_BASEADDR	0xA200C000
C_MEM14_BASEADDR	0xA201C000	C_MEM6_HIGHADDR	0xA200DFFF
C_MEM14_HIGHADDR	0xA201DFFF	C_MEM7_BASEADDR	0xA200E000
C_MEM15_BASEADDR	0xA201E000	C_MEM7_HIGHADDR	0xA200FFFF
C_MEM15_HIGHADDR	0xA201FFFF	C_MEM8_BASEADDR	0xA2010000
C_MEM16_BASEADDR	0xA2020000	C_MEM8_HIGHADDR	0xA2011FFF
C_MEM16_HIGHADDR	0xA2021FFF	C_MEM9_BASEADDR	0xA2012000
C_MEM17_BASEADDR	0xA2022000	C_MEM9_HIGHADDR	0xA2013FFF
C_MEM17_HIGHADDR	0xA2023FFF	C_INCLUDE_DPHASE_TIMER	1
C_MEM18_BASEADDR	0xA2024000	C_SPLB_AWIDTH	32
C_MEM18_HIGHADDR	0xA2025FFF	C_SPLB_CLK_PERIOD_PS	16000
C_MEM19_BASEADDR	0xA2026000	C_SPLB_DWIDTH	32
C_MEM19_HIGHADDR	0xA2027FFF	C_SPLB_MID_WIDTH	1
C_MEM1_BASEADDR	0xA2002000	C_SPLB_NATIVE_DWIDTH	32
C_MEM1_HIGHADDR	0xA2003FFF	C_SPLB_NUM_MASTERS	2
C_MEM20_BASEADDR	0xA2028000	C_SPLB_P2P	0
C_MEM20_HIGHADDR	0xA2029FFF	C_SPLB_SMALLEST_MASTER	32
C_MEM21_BASEADDR	0xA202A000	C_SPLB_SUPPORT_BURSTS	0
C_MEM21_HIGHADDR	0xA202BFFF		

4.2.4 Post Synthesis Device Utilization

	Used	**Total**	**Percentage**
Number of Slices	801	16640	4
Number of Slice Flip Flops	782	33280	2
Number of 4 input LUTs	1103	33280	3
Number of IOs	472	NA	NA
Number of bonded IOBs	0	519	0
BRAMs	48	84	57

4.2.5 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: 9.053ns (Maximum Frequency: 110.461MHz)

Minimum input arrival time before clock: 5.055ns

Maximum output required time after clock: 4.814ns

Maximum combinational path delay: 0.791ns

4.3 rx_gain_reg24

4.3.1 Port List

These are the ports listed in the MHS file. Please refer to the IP documentation for complete information about module ports.

Port List				
#	NAME	DIR	[LSB:MSB]	SIGNAL
0	slv_reg0_out	**O**	**0:31**	reg24_0_slv_reg0_out
1	slv_reg1_out	**O**	**0:31**	reg24_0_slv_reg1_out
2	slv_reg2_out	**O**	**0:31**	reg24_0_slv_reg2_out
3	slv_reg3_out	**O**	**0:31**	reg24_0_slv_reg3_out
4	slv_reg4_out	**O**	**0:31**	reg24_0_slv_reg4_out
5	slv_reg5_out	**O**	**0:31**	reg24_0_slv_reg5_out
6	slv_reg6_out	**O**	**0:31**	reg24_0_slv_reg6_out
7	slv_reg7_out	**O**	**0:31**	reg24_0_slv_reg7_out
8	slv_reg8_out	**O**	**0:31**	reg24_0_slv_reg8_out
9	slv_reg9_out	**O**	**0:31**	reg24_0_slv_reg9_out
10	slv_reg10_out	**O**	**0:31**	reg24_0_slv_reg10_out
11	slv_reg11_out	**O**	**0:31**	reg24_0_slv_reg11_out
12	slv_reg12_out	**O**	**0:31**	reg24_0_slv_reg12_out
13	slv_reg13_out	**O**	**0:31**	reg24_0_slv_reg13_out
14	slv_reg14_out	**O**	**0:31**	reg24_0_slv_reg14_out
15	slv_reg15_out	**O**	**0:31**	reg24_0_slv_reg15_out
16	slv_reg16_out	**O**	**0:31**	reg24_0_slv_reg16_out
17	slv_reg17_out	**O**	**0:31**	reg24_0_slv_reg17_out
18	slv_reg18_out	**O**	**0:31**	reg24_0_slv_reg18_out
19	slv_reg19_out	**O**	**0:31**	reg24_0_slv_reg19_out
20	slv_reg20_out	**O**	**0:31**	reg24_0_slv_reg20_out
21	slv_reg21_out	**O**	**0:31**	reg24_0_slv_reg21_out
22	slv_reg22_out	**O**	**0:31**	reg24_0_slv_reg22_out
23	slv_reg23_out	**O**	**0:31**	reg24_0_slv_reg23_out

4.3.2 Parameters

These are the current parameter settings for this module.

Name	**Value**	**Name**	**Value**

C_FAMILY	spartan3adsp	C_SPLB_MID_WIDTH	1
C_BASEADDR	0xA1000000	C_SPLB_NATIVE_DWIDTH	32
C_HIGHADDR	0xA100FFFF	C_SPLB_NUM_MASTERS	2
C_INCLUDE_DPHASE_TIMER	1	C_SPLB_P2P	0
C_SPLB_AWIDTH	32	C_SPLB_SMALLEST_MASTER	32
C_SPLB_CLK_PERIOD_PS	16000	C_SPLB_SUPPORT_BURSTS	0
C_SPLB_DWIDTH	32		

4.3.3 Post Synthesis Device Utilization

	Used	**Total**	**Percentage**
Number of Slices	856	16640	5
Number of Slice Flip Flops	956	33280	2
Number of 4 input LUTs	954	33280	2
Number of IOs	969	NA	NA
Number of bonded IOBs	0	519	0

4.3.4 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: 9.710ns (Maximum Frequency: 102.987MHz)

Minimum input arrival time before clock: 2.923ns

Maximum output required time after clock: 0.591ns

Maximum combinational path delay: No path found

4.4 RX_Amp_Loader_top_0

4.4.1 Port List

These are the ports listed in the MHS file.

Port List				
#	NAME	DIR	[LSB: MSB]	SIGNAL
0	MResetHiIn	**I**	**1**	AcquisitionLogicResetHi
1	RXAmpGainRegCh1In8	**I**	**0:7**	reg24_0_slv_reg0_out[24:31]
2	RXAmpGainRegCh2In8	**I**	**0:7**	reg24_0_slv_reg1_out[24:31]
3	RXAmpGainRegCh3In8	**I**	**0:7**	reg24_0_slv_reg2_out[24:31]
4	RXAmpGainRegCh4In8	**I**	**0:7**	reg24_0_slv_reg3_out[24:31]
5	RXAmpGainRegCh5In8	**I**	**0:7**	reg24_0_slv_reg4_out[24:31]
6	RXAmpGainRegCh6In8	**I**	**0:7**	reg24_0_slv_reg5_out[24:31]
7	RXAmpGainRegCh7In8	**I**	**0:7**	reg24_0_slv_reg6_out[24:31]
8	RXAmpGainRegCh8In8	**I**	**0:7**	reg24_0_slv_reg7_out[24:31]
9	RXAmpGainRegCh9In8	**I**	**0:7**	reg24_0_slv_reg8_out[24:31]
10	RXAmpGainRegCh10In8	**I**	**0:7**	reg24_0_slv_reg9_out[24:31]
11	RXAmpGainRegCh11In8	**I**	**0:7**	reg24_0_slv_reg10_out[24:31]
12	RXAmpGainRegCh12In8	**I**	**0:7**	reg24_0_slv_reg11_out[24:31]
13	RXAmpGainRegCh13In8	**I**	**0:7**	reg24_0_slv_reg12_out[24:31]

14	RXAmpGainRegCh14In8	**I**	**0:7**	reg24_0_slv_reg13_out[24:31]
15	RXAmpGainRegCh15In8	**I**	**0:7**	reg24_0_slv_reg14_out[24:31]
16	RXAmpGainRegCh16In8	**I**	**0:7**	reg24_0_slv_reg15_out[24:31]
17	RXAmpGainRegCh17In8	**I**	**0:7**	reg24_0_slv_reg16_out[24:31]
18	RXAmpGainRegCh18In8	**I**	**0:7**	reg24_0_slv_reg17_out[24:31]
19	RXAmpGainRegCh19In8	**I**	**0:7**	reg24_0_slv_reg18_out[24:31]
20	RXAmpGainRegCh20In8	**I**	**0:7**	reg24_0_slv_reg19_out[24:31]
21	RXAmpGainRegCh21In8	**I**	**0:7**	reg24_0_slv_reg20_out[24:31]
22	RXAmpGainRegCh22In8	**I**	**0:7**	reg24_0_slv_reg21_out[24:31]
23	RXAmpGainRegCh23In8	**I**	**0:7**	reg24_0_slv_reg22_out[24:31]
24	RXAmpGainRegCh24In8	**I**	**0:7**	reg24_0_slv_reg23_out[24:31]
25	RXAmpFiltModeRegIn6	**I**	**0:5**	RXAmpFiltModeRegIn6
26	RXAmpTRSwitchOnTimeRegIn24	**I**	**0:23**	TX_Enables_Reg
27	LoadReqHiIn	**I**	**1**	LoadReqHiIn
28	SData1Out	**O**	**1**	RX_Amp_Loader_top_0_SData1Out
29	SData2Out	**O**	**1**	RX_Amp_Loader_top_0_SData2Out
30	SData3Out	**O**	**1**	RX_Amp_Loader_top_0_SData3Out
31	SData4Out	**O**	**1**	RX_Amp_Loader_top_0_SData4Out
32	SClockOut	**O**	**1**	RX_Amp_Loader_top_0_SClockOut
33	Clk1MHzStrbIn	**I**	**1**	clk_1MHz

4.4.2 Post Synthesis Device Utilization

	Used	**Total**	**Percentage**
Number of Slices	217	16640	1
Number of Slice Flip Flops	89	33280	0
Number of 4 input LUTs	399	33280	1
Number of IOs	230	NA	NA
Number of bonded IOBs	0	519	0
Number of GCLKs	1	24	4

4.4.3 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: 5.001ns (Maximum Frequency: 199.960MHz)

Minimum input arrival time before clock: No path found

Maximum output required time after clock: 2.650ns

Maximum combinational path delay: No path found

4.5 rx_ip_0

4.5.1 Port List

These are the ports listed in the MHS file.

Port List				
#	NAME	DIR	[LSB:MSB]	SIGNAL
0	rx_en	**I**	**1**	TXRXRunReqHiOut

1	mb_sram1_addr	**I**	**0:19**	mb_sram1_addr_10_29
2	mb_sram2_addr	**I**	**0:19**	mb_sram2_addr_10_29
3	mb_sram3_addr	**I**	**0:19**	mb_sram3_addr_10_29
4	mb2sram_d1	**I**	**0:31**	mb_dq1_o
5	sram2mb_d1	**O**	**0:31**	mb_dq1_i
6	mb2sram_d2	**I**	**0:31**	mb_dq2_o
7	sram2mb_d2	**O**	**0:31**	mb_dq2_i
8	mb2sram_d3	**I**	**0:31**	mb_dq3_o
9	sram2mb_d3	**O**	**0:31**	mb_dq3_i
10	mb_sram1_cen	**I**	**1**	mb_sram1_cen
11	mb_sram1_oen	**I**	**1**	mb_sram1_oen
12	mb_sram1_wen	**I**	**1**	mb_sram1_wen
13	mb_sram2_cen	**I**	**1**	mb_sram2_cen
14	mb_sram2_oen	**I**	**1**	mb_sram2_oen
15	mb_sram2_wen	**I**	**1**	mb_sram2_wen
16	mb_sram3_cen	**I**	**1**	mb_sram3_cen
17	mb_sram3_oen	**I**	**1**	mb_sram3_oen
18	mb_sram3_wen	**I**	**1**	mb_sram3_wen
19	ce1_n	**O**	**1**	SRAM_BANK1_CE
20	oe1_n	**O**	**1**	SRAM_BANK1_OE
21	we1_n_out	**O**	**1**	SRAM_BANK1_WE
22	ce2_n	**O**	**1**	SRAM_BANK2_CE
23	oe2_n	**O**	**1**	SRAM_BANK2_OE
24	we2_n_out	**O**	**1**	SRAM_BANK2_WE
25	ce3_n	**O**	**1**	SRAM_BANK3_CE
26	oe3_n	**O**	**1**	SRAM_BANK3_OE
27	we3_n_out	**O**	**1**	SRAM_BANK3_WE
28	sram_addr_out	**O**	**0:19**	SRAM_A_MUX_concat
29	d1	**IO**	**0:31**	BANK_1_SRAM_D
30	d2	**IO**	**0:31**	BANK_2_SRAM_D
31	d3	**IO**	**0:31**	BANK_3_SRAM_D
32	adc_cs_n	**O**	**1**	rx_ip_0_adc_cs_n
33	adc_sdo_array1	**I**	**0:7**	rx_ip_0_adc_1_sdo
34	adc_sdo_array2	**I**	**0:7**	rx_ip_0_adc_2_sdo
35	adc_sdo_array3	**I**	**0:7**	rx_ip_0_adc_3_sdo
36	TXRXAbortReqHi	**I**	**1**	TXRXAbortReqHiOut
37	clk_mb	**I**	**1**	clk_62_5000MHz
38	rx_sample_number	**I**	**0:16**	RX_number_of_samples_reg
39	reset	**I**	**1**	AcquisitionLogicResetHi
40	sram_clr	**I**	**1**	SRAM_HD_CLEAR
41	clk_rx_in	**I**	**1**	clk_20MHz
42	clk_divider_reg	**I**	**0:2**	RX_Digitiser_Sampling_Rate_reg
43	clk_rx_270_in	**I**	**1**	clk_20MHz_270
44	rx_one_iteration_done	**O**	**1**	rx_ip_0_rx_one_iteration_done

4.5.2 Post Synthesis Device Utilization

	Used	**Total**	**Percentage**
Number of Slices	630	16640	3
Number of Slice Flip Flops	822	33280	2
Number of 4 input LUTs	625	33280	1
Number of IOs	538	NA	NA
Number of bonded IOBs	0	519	0
TBUFs	96	0	NA

4.5.3 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: 12.966ns (Maximum Frequency: 77.125MHz)

Minimum input arrival time before clock: 6.960ns

Maximum output required time after clock: 10.474ns

Maximum combinational path delay: 2.113ns

4.6 TT4_Auxiliary_Functions_top_0

IP Specs		
Core	**Version**	**Documentation**
TT4_Auxiliary_Functions_top_0	1.00.a	IP [1]

4.6.1 Port List

These are the ports listed in the MHS file.

Port List				
#	NAME	DIR	[LSB:MSB]	SIGNAL
0	Clk100HzIn	**I**	**1**	Clk100Hz
1	ExternalResetHiIn	**I**	**1**	sys_rst_s
2	AcqRunningHiIn	**I**	**1**	AcqRunningHi
3	MiscellaniousFunctionsReg	**I**	**0:7**	MiscellaniousFunctionsReg
4	ClockManagerStableHiIn	**I**	**1**	Dcm_1_locked
5	InternalResetHiIn	**I**	**1**	sys_periph_reset
6	AcquisitionLogicResetHiOut	**O**	**1**	AcquisitionLogicResetHi
7	ChannelSwitchNormalDrvLoOut	**O**	**1**	ChannelSwitchNormalDrvLo
8	ChannelSwitchAlternateDrvLoOut	**O**	**1**	ChannelSwitchAlternateDrvLo
9	SRAM_Hardware_Clear_Out	**O**	**1**	SRAM_HD_CLEAR
10	Front_Pannel_Reset_n_Out	**O**	**1**	Front_Pannel_Reset_n_Out

4.6.2 Post Synthesis Device Utilization

	Used	**Total**	**Percentage**
Number of Slices	7	16640	0
Number of Slice Flip Flops	7	33280	0
Number of 4 input LUTs	13	33280	0
Number of IOs	18	NA	NA
Number of bonded IOBs	0	519	0

4.6.3 Timing Summary

Estimated based on synthesis

Speed Grade: -4

Minimum period: 2.742ns (Maximum Frequency: 364.697MHz)

Minimum input arrival time before clock: 2.845ns

Maximum output required time after clock: 0.591ns

Maximum combinational path delay: 0.791ns

5 MicroBlaze and Acquisition Logic Interface

5.1 General Description

MicroBlaze must set up all the configuration registers, load TX waveforms to BRAM, and clear RX SRAM before initiate a collection. Once AL is started and running, MicroBlaze can not write to the WO registers. Writing to these registers when AL is running will have no effect on the current Collection. The only exemption is the Collection_control register which may be written to at any time in order to allow the MicroBlaze to abandon the current collection.

The MicroBlaze must not load invalid or inconsistent data at any time. If it does, results will be un-predictable at best and will probably result in serious errors in acquired data and in extreme cases might even cause permanent damage to the Teletest hardware. Where practical the AL will check for any such data errors but it is not practicable to make it completely foolproof so the main responsibility must rest with the MicroBlaze firmware writer(s).

MicroBlaze decodes messages sent from the remote PC via Ethernet link and set the related registers to carry out the following main functions:

- Download transmit waveform data to each of the 24 transmit BRAM
- Set PRF
- Set TX enable / RX mode- Receive-after-Transmit or Receive-while-Transmit
- Set up receive parameters - gain, filtering, digitisation rate, number of samples, number of accumulation.
- Up-load received data.
- Enquire status of current operation - number of RX records obtained so far.
- Abandon current operation.

5.2 Register Map

	Register Name	Type	[MSB: LSB]	Memory Address
	tx_bram_reg20_0			
1	CollectionControlReg	w	[2:0]	0xA0000000
2	PRFPeriodReg	w	[9:0]	0xA0000004
3	RequiredTXRXCyclesCount13Reg	w	[12:0]	0xA0000008
4	RequiredTRSwitchOnDelayCount8Reg	w	[7:0]	0xA000000C
5	TX_word_count_Reg	w	[10:0]	0xA0000010
6	TX_Waveform_Interpolation_Rate_Reg	w	[2:0]	0xA0000014
7	TX_Enables_Reg	w	[23:0]	0xA0000018
8	RX_bandwidth_n_mode_reg	w	[5:0]	0xA000001C
9	RX_blanking_time_reg	w	[7:0]	0xA0000020
10	RX_Digitiser_Sampling_Rate_reg	w	[1:0]	0xA0000024
11	RX_number_of_samples_reg	w	[16:0]	0xA0000028
12	Misc_functions_reg	w	[7:0]	0xA000002C
13	CollectionStatusReg8Out	r	[7:0]	0xA0000030
14	NumberOfTXRXCyclesCompletedCount13Out	r	[12:0]	0xA0000034
15	Digitiser_saturation_detect_flags_reg	r	[23:0]	0xA0000038
16	FPGA_Version_reg	r	[31:0]	0xA000003C

17	FPGA_Varient_reg	r	[31:0]	0xA0000040
	rx_gain_reg24			
18	reg24_0_slv_reg0	r	[7:0]	0xA1000000
19	reg24_0_slv_reg1	r	[7:0]	0xA1000004
20	reg24_0_slv_reg2	r	[7:0]	0xA1000008
21	reg24_0_slv_reg3	r	[7:0]	0xA100000C
22	reg24_0_slv_reg4	r	[7:0]	0xA1000010
23	reg24_0_slv_reg5	r	[7:0]	0xA1000014
24	reg24_0_slv_reg6	r	[7:0]	0xA1000018
25	reg24_0_slv_reg7	r	[7:0]	0xA100001C
26	reg24_0_slv_reg8	r	[7:0]	0xA1000020
27	reg24_0_slv_reg9	r	[7:0]	0xA1000024
28	reg24_0_slv_reg10	r	[7:0]	0xA1000028
29	reg24_0_slv_reg11	r	[7:0]	0xA100002C
30	reg24_0_slv_reg12	r	[7:0]	0xA1000030
31	reg24_0_slv_reg13	r	[7:0]	0xA1000034
32	reg24_0_slv_reg14	r	[7:0]	0xA1000038
33	reg24_0_slv_reg15	r	[7:0]	0xA100003C
34	reg24_0_slv_reg16	r	[7:0]	0xA1000040
35	reg24_0_slv_reg17	r	[7:0]	0xA1000044
36	reg24_0_slv_reg18	r	[7:0]	0xA1000048
37	reg24_0_slv_reg19	r	[7:0]	0xA100004C
38	reg24_0_slv_reg20	r	[7:0]	0xA1000050
39	reg24_0_slv_reg21	r	[7:0]	0xA1000054
40	reg24_0_slv_reg22	r	[7:0]	0xA1000058
41	reg24_0_slv_reg23	r	[7:0]	0xA100005C

5.3 Register Description

5.3.1 Collection Control Register

CollectionControlReg – 3 bits, WO, default value: 000b

This register controls start, abort and warm restart of the acquisition logic.

Bit 0	Collection_start_request bit This bit should be set by MicroBlaze when a collection is requested. This will cause the AL logic to start an acquisition using the configuration values previously loaded in registers. This bit should be cleared when a collection terminates and all required data and status information are collected and sent to the remote PC. By clearing this bit, all configuration registers are reset to their default values in preparation for the next collection. Note that clearing this flag is not an alternative way to initiate collection abort – this must be done by setting the Collection_abandon_request flag. If a collection abort occurs whether as a result of a request from the MicroBlaze, or a hardware event, the same remarks as above apply to the Collection_start_request flag which must be kept set until the MicroBlaze has read back any status flags etc that it wishes to access.
Bit1	Collection_abandon_request bit

	If the MicroBlaze desires to halt an Collection before it has completed normally then it should set this bit. The AL will respond by shutting down the Collection process and once this is complete will clear the Collection_running status flag. The MicroBlaze must clear this bit at some stage after the Collection has terminated (as indicated by the status flag Collection_running going to clear) and starting the next Collection.
Bit2	Perform Warm Restart_request bit If this bit is set by the MicroBlaze the effect will be to cause the AL to shut down any collection that may be in progress and then activate a hardware function to cause the FPGA to be reloaded with configuration data.

5.3.2 PRFPeriod Register

PRF register – 10 bits, WO, default value: 10d

This register must be loaded with a binary-coded value between 10d to 1000d corresponding to a range of pulse repetition frequency period time from 0.1 second (10Hz) to 10 seconds (0.1Hz).

5.3.3 Required Tx-Rx Cycles Count Register

RequiredTXRXCyclesCount13Reg, 13 bits, WO, default value: 1d

This register must be loaded with a binary-coded value corresponding to the number of TX-RX cycles to be carried out for accumulation. For any value greater than 1, the stored value for each sample of RX data will be the arithmetic sum of all the corresponding samples from each successive TX-RX cycle. The permitted range is 1d to 4096d. A setting of 1 will give a single, non-accumulative TX/RX cycle.

5.4 Required Tx-Rx Switch On Delay Count Register

RequiredTRSwitchOnDelayCount8Reg, 8 bits, WO, default value: 0d

RX T/R switch on-time is selected by the state of the corresponding TX Amp enable

- If disabled, T/R switch must be closed before/by the TX Burst Start time.

- If enabled, T/R switch starts to close at TX Burst End time + Blanking Time as defined by
contents of RX Blanking register.

5.4.1 TX_word_count_Reg

TX_word_count register – 12 bits, WO, default value: 0d

This register must be loaded with a binary coded number in the range 10d to 3072d corresponding to the number of words to read out of each TX Waveform Buffer.

5.4.2 TX_Waveform_Interpolation_Rate_Reg

TX_Waveform_Interpolation_Rate register – 3 bits, WO, default value: 7d

This register must be loaded with a binary-coded number corresponding to one of the valid numbers specified in the table below:

Code	bit [2:0]	Interpolation	Interpolation Sample Rate
7d	111	x 16	16 MSPS
5d	101	x 8	8 MSPS
3d	011	x 4	4 MSPS
1d	001	x 2	2 MSPS
0d	000	No Interpolation	1 MSPS

5.4.3 TX_Enables_Reg

TX_Enables register – 24 bits, WO, default value: 0d

Each TX Amplifier has an individual enable/disable control which selects the transmit value between the data stored in the waveform BRAM and a fixed value representing the mid level analogue output. These bits are used to disable individual transmitters without having to change the stored waveform data. The register bits 0 to 23 correspond to TX channel number 1 to 24 respectively.

In addition to their primary effect as describe above, each TX Enable bit will also affect the time at which the associated channel switches from TX to RX. If a TX Enable bit is set to ‘Disabled’ then the associated channel is switched to RX before the next TX/RX cycle commences, in which case the RX channel is able to receive signals while the enabled TX channels are transmitting. This is used for pitch-catch mode. If the TX Enable bit is set to ‘Enabled’, the associated channel will only be switched from TX to RX after the TX burst has completed.

Bit 0 to 23	0- Transmitter Disabled; TX-RX switch to RX at the start of transmit. 1- Transmitter Enabled; TX-RX switch to RX at the end of transmit.

5.4.4 RX_bandwidth_n_mode_reg

RX_bandwidth_&_mode register – 6 bits, WO, default value: 001001b

The RX bandwidth and mode register controls Low-pass frequency, High-pass frequency and Mode (normal or TX Monitor). The settings held in this register are applied in common to all 24 RX Amps.

Note: RX Mode

Normal - the receivers operate normally where the inputs are connected directly to the tooling transducers signals.

TX Monitor - the inputs are transferred to signals obtained from voltage dividers on transmitter outputs.

Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
Low-pass filter						
x	X	x	0	0	1	600 KHz (default)
x	X	x	0	1	0	300 KHz
x	X	x	0	1	1	150 KHz
x	X	x	1	0	0	75 KHz
x	X	x	1	0	1	32 KHz
x	X	x	1	1	0	16 KHz
High-pass filter						
x	0	1	x	x	x	1 KHz (default)
x	1	0	x	x	x	5 KHz
RX Mode						
0	X	x	x	x	x	Normal (default)
1	X	x	x	x	x	Transmit Monitor

5.4.5 RX_blanking_time_reg

RX Blanking Time Register – 8 bits, W0, default value: 0d

If it is desired to delay the time at which the Transmit/Receive switches change over to Receive mode then this register should be loaded with the appropriate value. Coding is 8 bit unsigned binary, scaling 1 bit = 10 uSec.

Note that this delay will only affect those channels where there is an enabled transmitter. For channels where the transmitter is disabled, the respective T/R switches will always be in their ‘receive’ states at the start of each new TX-RX cycle regardless of the value in this register.

5.4.6 RX_Digitiser_Sampling_Rate_reg

RX_Digitiser_Sampling_Rate register – 3 bits, WO, default value: 01d

This is a single register which sets the sample rate for all 24 RX channels to a common value selected from the table below:

Code	bit [2:0]	RX Sample Rate

01d	001	1 MSPS
02d	010	500 KSPS
03d	011	250 KSPS
04d	100	125 KSPS

5.4.7 RX_number_of_samples_reg

RX_number_of_samples register – 18 bits, WO, default value: 1000d

This register should be loaded with the number of RX digitiser samples to be taken in each TX-RX cycle. The settings held in this register are common to all 24 RX channels. The permissible range is 1 to 20000h (128k).

5.4.8 Misc_functions_reg

Misc_functions registers – 8 bit WO, default value:0d

Bit 0	Acquisition Logic Asynchronous Reset Clear : Acquisition Logic is held in Reset Set : Acquisition logic may be operating normally provided no other resets are active **Warning** This function is principally provide to allow the MB to start up and initialise itself after power up and/or FPGA configuration while all the acquisition logic is held in a safe reset condition. It also allows the MB to force a reset of the acquisition logic at any time, but this should only ever be done under emergency recovery conditions and never as a routine procedure as it may cause temporary abnormal conditions to arise in the acquisition logic which in theory could cause conditions to arise where a full power cycle was necessary to restore normal operation.
Bit1	Transducer Connection Configuration Clear : ‘Normal’ Set : ‘Alternate’ This bit effectively controls the setting of the relays on the tooling switch board which route 16 of the TX-RX connections to an alternative set of tooling elements.
Bit2	SRAM Hardware Clear Set: By setting this bit, an FPGA hardware function is called to clear the full range SRAM memory. This takes about 128 ms. Clear: This bit must be cleared before starting a collection.
Bit3 to 7	Reserved

5.4.9 CollectionStatusReg8Out

Collection_status register – 8 bits, RO, reset value: 0d

This register contains a series of bits indicating the current state of Collection.

Bit 0	Collection_running bit When set this bit indicates that an active Collection is underway, when clear indicates that it is inactive/has terminated. The MicroBlaze may poll this bit to determine when a collection has terminated and then inspect the status bits to determine the reason
Bit 1	Collection_terminated_without _errors_or_abandon_request bit If a collection terminates normally then this bit will be set, otherwise will be clear
Bit 2	Collection_terminated_on_abandon_request bit If a Collection terminates as a result of a abandon request from the MicroBlaze then this bit will be set. It will remain set until the next Collection starts.
Bit 3	Collection_terminated_on_TX_amp_overcurrent bit If a Collection terminates as a result of a TX amplifier overcurrent event then this bit will be set. It will remain set until the next Collection starts.
Bit 4	Collection_terminated_on_HT_PSU_Overcurrent_event bit If an Collection terminates as a result of the HT PSU going into over-current then this bit will be set. It will remain set until the next Collection starts.
Bit 5	Collection terminated on HT PSU on timeout If the HT PSU failed to assert its' 'OP Good' signal within the required time from being switched on then the collection will be aborted and this bit will be set.
Bit 6	Master_mode bit Set if unit is operating as the Master in a multi-unit system, cleared otherwise.
Bit 7	Slave_mode bit Set if unit is operating as a Slave in a multi-unit system, cleared otherwise.

5.4.10 NumberOfTXRXCyclesCompletedCount13Out

Number_of_TX/RX_cycles_count register – 13 bits, RO, reset value:0d

This register may be read by the MicroBlaze at any time and will return the number of completed TX-RX cycles carried out so far in the current collection. The value will be returned in unsigned binary format in the range 0 to 4096.

5.4.11 Digitiser_saturation_detect_flags_reg

Digitiser_saturation_detect_flags registers – 24 bits, RO, reset value:0d

This register contains the results so far of the digitiser saturation detection. The bits in this register are allocated in the order bit 0 to RX Channel#1, bit 1 to RX Channel#2 etc. A bit set indicates that one or more digitiser saturation event/s has/have been detected in that/those channel/s since the start of the current TX-RX cycle.

This register may be read by the MicroBlaze at any time.

The AL will reset these flags at the start of each new TX-RX cycle which means that the MicroBlaze may need to read this register on a regular basis, at least once per TX-RX cycle.

5.4.12 FPGA_Version_reg

FGPA_Version register – 32 bits RO

This read-only register will contain data to identify the version register of the FPGA logic. The current content is an integer number starting from 1.

5.4.13 FPGA_Varient_reg

FGPA_Varient register – 32 bits RO

This read-only register will contain data to identify any variant of the FPGA logic – it is a quite common occurrence that a variant is produced for some special situation. It is therefore useful to have a register which can be read back to determine if this is the case. The current content is an integer starting from 1.

5.4.14 RX Gain and filter registers

RX_Gain registers – 8 bits, WO x 24, default values: 40d

There are 24 of these registers, one for each RX channel and are mapped into the PLB to occupy contiguous addresses. The register with the lowest address in the PLB address map will control the gain of RX Channel 1 etc up to RX Channel 24.

Register Bits 0..7: binary-coded gain in units of 0.5 dB

Permitted values : 40d to 200d in steps of 2d corresponding to gain in 1dB steps over range +20dB to +100dB e.g. to set the gain to +47dB, a value of 94d would be loaded into these bits.

6 FPGA Design Summary

Report	Flip Flops Used	LUTs Used	BRAMS Used
system	11683	13728	71
xps_timer_0	358	365	
xps_intc_0	196	150	
xps_spi_1	311	365	
xps_spi_0	242	285	
xps_iic_0	379	479	
tx_bram_reg20_0	782	1103	48
tt4_auxiliary_functions_top_0	7	13	
rx_amp_loader_top_0	89	399	
rx_ip_0	822	625	
rx_gain_reg24	956	954	
sram_3	479	353	
sram_2	479	353	
sram_1	479	353	
tt4_top_level_sequencer_top_0	180	375	
xps_gpio_1	111	64	
xps_gpio_0	115	99	
fit_timer_0	11	24	
xps_timebase_wdt_0	166	134	
xps_uartlite_1	144	132	
xps_uartlite_0	152	142	
proc_sys_reset_0	67	52	
mdm_0	119	147	
dcm_module_1	4	1	
dcm_module_0	4	1	
ethernet_mac	737	1101	4
ddr2_sdram	1605	1006	5
lmb_bram			8
ilmb_cntlr	2	6	
dlmb_cntlr	2	6	
dlmb	1	1	
ilmb	1	1	
mb_plb	165	597	
microblaze_0	2518	4042	6

7 REFERENCED DOCUMENTS

[1] Xilinx, “Spartan-3A DSP FPGA Family: Introduction and Ordering Information,” Prodect Specification, DS610-1 (v2.2) March 11, 2009.

[2] Xilinx, “Spartan 3 Generation FPGA User Guide,” Chapter 10, UG331 (v1.5) March 11, 2009.

[3] Xilinx, “XtremeDSP DSP48A for Spartan-3A DSP FPGAs User Guide,” UG431 (v1.3) July 15, 2008.

[4] DAC8580 datasheet, SLAS458B - June 2005 – Revised August 2005, Burr-Brown from Texas Instrument.

[5] XPS Ethernet Lite Media Access Controller, Prodect Specification, Xilinx, DS580, June 24, 2009

[6] Embedded Networking with Xilinx MicroBlaze, Speedway Design Workshops, Avnet, Version 1.2, Feb, 2008

[7] W5100 Datasheet Version 1.1.8.

[8] ADS7886 Datasheet, Texas Instruments, SLAS492-September 2005

[9] MT47H32M16 – 8 Meg x 16 x 4 banks DDR2 SRAM datasheet, Micron.

[10] Implementing DDR2 SDRAM Controllers , Chapter 8, Xilinx Memory Interface Generator (MIG) User Guide, Version 3.1, Xilinx, 24 June 2009

[11] “Interfacing Micron DDR2 Memories to Xilinx Spartan-3A/AN FPGAs:A Step-By-Step Guide”, Application Note, AN086 February 18, 2008 (Version 1.0), NuHorizons Electronics.

[12] Ershen Wang; Shufang Zhang; Qing Hu; Jiang Yi; Xiaowen Sun, “Implementation of an Embedded GPS Receiver Based on FPGA and MicroBlaze,” Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on 12-14 Oct. 2008, Page(s):1 - 4

[13] Robson, C.C.W.; Bousselham, A.; Bohm, C., “An FPGA based general purpose data acquisition controller,” Real Time Conference, 2005. 14th IEEE-NPSS, 10-10 June 2005 Page(s):4 pp.

[14] A. Peled, Liu. Bede, “A new hardware realization of digital filters,” *Acoustics, Speech and Signal Processing, IEEE,* Volume 22, Issue 6, Dec 1974 pp.456 – 462

[15] U. Meyer-Baese, “Digital signal processing with Field Programmable Gate Arrays,” Third Eddition, Springer-Verlag, 2007, pp.115-120.

[16] Xilinx, “FIR Compiler v4.0”, Product Specification, DS534 June 27, 2008.