Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

As mobile technologies advance further connectivity becomes increasingly more important. Latest developments in smart phone industry created an environment where many devices communicate with each other. In this context Bluetooth technology makes it possible to connect together various devices with diverse sizes and complexities. On the other hand, FPGAs power many consumer electronics and industrial applications. [6] Internet of things has become a growing industry on its own with many practical applications and even FPGAs solely targeting this market are being developed. [5] Combining the technologies described above gives way to innovative applications and this is the main motivation behind this study.

In light of these developments this thesis explores a proof of concept application which establishes communication between a smart phone and a FPGA based system.

1.2 Objective

Primary objective of this work is to develop an application where an Android App can communicate with a FPGA based system using Bluetooth connection and manipulate data on the FPGA.

1.3 Scope

Requirements for this project are as follows:

- Android App should be able to transfer 16 8-bits numbers to FPGA over Bluetooth
- Android App should have an user friendly interface
- FPGA based system can receive data sent from Android App
- FPGA based system can save the data to a memory block which can be accessed by other FPGA modules
- Data on memory block can be observed from a display unit

Chapter 2

Preliminaries

This chapter presents an overview of relevant information on hardware, software and tools used in the implementation of this project.

2.1 Microblaze

The MicroBlaze embedded processor soft core is a reduced instruction set computer (RISC) optimized for implementation in XilinxÂő Field Programmable Gate Arrays (FPGAs). [3]

2.1.1 Architecture

The MicroBlaze soft core processor is highly configurable, allowing to select a specific set of features required by design. The fixed feature set of the processor includes:

- Thirty-two 32-bit general purpose registers
- 32-bit instruction word with three operands and two addressing modes
- Default 32-bit address bus, extensible to 64 bits on the data side
- Single issue pipeline

In addition to these fixed features, the MicroBlaze processor is parametrized to allow selective enabling of additional functionality.

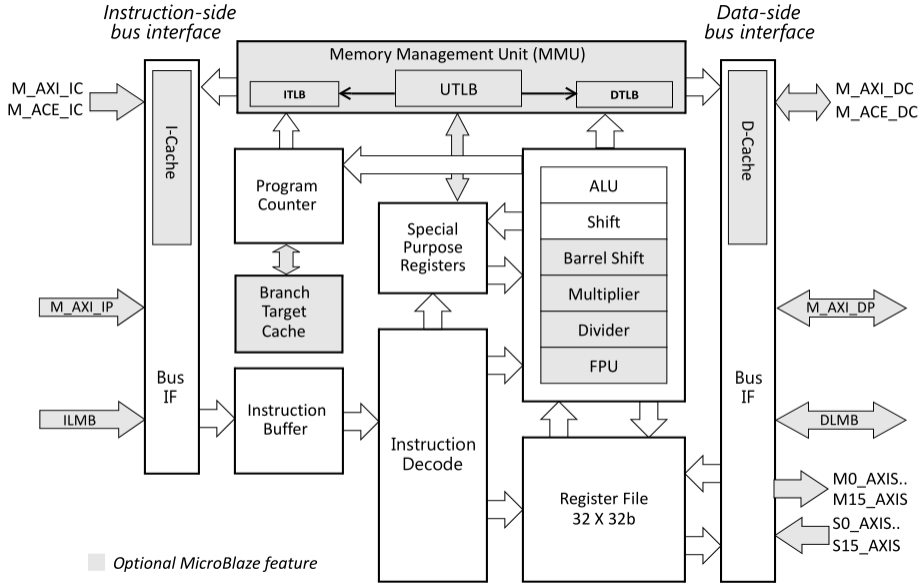

Figure 2-1: Microblaze block diagram

All MicroBlaze instructions are 32 bits and are defined as either Type A or Type B. Type A instructions have up to two source register operands and one destination register operand. Type B instructions have one source register and a 16-bit immediate operand (which can be extended to 32 bits by preceding the Type B instruction with an imm instruction). Type B instructions have a single destination register operand. Instructions are provided in the following functional categories: arithmetic, logical, branch, load/store, and special. Table 2-6 lists the MicroBlaze instruction set.

MicroBlaze has an orthogonal instruction set architecture. It has thirty-two 32-bit general purpose registers and up to eighteen 32-bit special purpose registers, depending on configured options.

The thirty-two 32-bit General Purpose Registers are numbered R0 through R31. The register file is reset on bit stream download (reset value is 0x00000000). Figure 2-2 is a representation of a General Purpose Register and Table 2-7 provides a description of each register and the register reset value (if existing).

MicroBlaze supports reset, interrupt, user exception, break, and hardware exceptions.

The relative priority starting with the highest is:

1. Reset
2. Hardware Exception
3. Non-maskable Break
4. Break
5. Interrupt
6. User Vector (Exception)

2.1.2 Signal Interface

The MicroBlaze core is organized as a Harvard architecture with separate bus interface units for data and instruction accesses. The following two memory interfaces are supported: Local Memory Bus (LMB), and the AMBAÂő AXI4 interface (AXI4) and ACE interface (ACE). The LMB provides single-cycle access to on-chip dual-port block RAM. The AXI4 interfaces provide a connection to both on-chip and off-chip peripherals and memory. The ACE interfaces provide cache coherent connections to memory. MicroBlaze also supports up to 16 AXI4-Stream interface ports, each with one master and one slave interface.

MicroBlaze can be configured with the following bus interfaces:

- The AMBA AXI4 Interface for peripheral interfaces, and the AMBA AXI4 or AXI Coherency Extension (ACE) Interface for cache interfaces
- LMB provides a simple synchronous protocol for efficient block RAM transfers
- AXI4-Stream provides a fast non-arbitrated streaming communication mechanism
- Debug interface for use with the Microprocessor Debug Module (MDM) core
- Trace interface for performance analysis

2.1.3 Xilinx EDK

The Embedded Development Kit (EDK) is an integrated development environment for designing embedded processing systems. This pre-configured kit includes Xilinx Platform Studio and the Software Development kit, as well as all the documentation and IP that you require for designing Xilinx Platform FPGAs with embedded PowerPC hard processor cores and/or MicroBlaze soft processor cores. The Embedded Development Kit Provides:

Xilinx Platform Studio (XPS) Tool Suite - Including: Graphical IDE and command line support for developing hardware platforms for embedded applications. The Base System Builder wizard enables creation of a working embedded system within minutes. XPS also includes other intelligent design wizards to quickly configure the embedded system architecture, buses and peripherals.

Software Development Kit (SDK) for MicroBlaze and PowerPC - Including: GNU C/C++ compiler and debugger; Xilinx Microprocessor Debug (XMD) target server; Data2MEM utility for bitstream loading and updating. SDK is the recommended software-centric design environment based on the Eclipse IDE.

Real-Time Operating System and Embedded OS Support - Provides design support and board support package (BSP) generation for numerous third party suppliers in the Xilinx ecosystem, including vendors such as Wind River, Green Hills, Mentor, LynuxWorks and other embedded industry leaders.

Processing IP and MicroBlaze Soft Processor Core -Pre-verified IP catalog, including a wide variety of processing peripheral cores for customizing your embedded systems as well as the flexible MicroBlaze 32-bit soft processing core. The MicroBlaze processor offers memory management and FPU configuration options enabling commercial grade RTOS support, unique for a soft processor. [4]

2.2 Pmod CLS

The Digilent PmodCLS is a 16x2 character LCD module driven by an Atmel ATmega48 microcontroller.

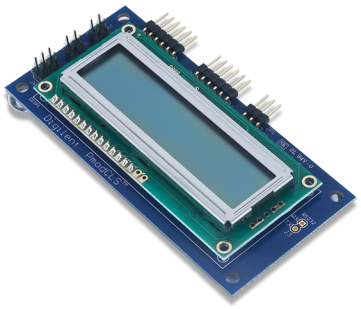

Figure 2-2: Pmod CLS LCD Module

The PmodCLS module can be used to display important information during program development or as a user interface after the project has been completed. The module is ideally suited for microcontroller boards but can also be used in projects using a FPGA board. The module is capable of executing a variety of instructions such as erasing specific characters, setting different display modes, scrolling, and displaying user-defined characters. These instructions are specified using escape sequences to send commands to the boardâĂŹs embedded Atmel ATmega48 microcontroller. The display on the module is driven by this AVR and controls all of the features of the board.

The PmodCLS can communicate with the host board through the SPI, I 2 C, and UART ports.Through these protocol standards, users are able to set the cursor location and send other instructions by sending escape sequences. And escape sequence is specified by first sending the escape character (0x1B) followed by a left square bracket '[' (0x5B), and then one or more numeric parameters followed by the command character for the specific command.

2.3 HC-06 Bluetooth Module

HC-06 module is a low-cost Bluetooth module which can be used by any UART implementation. This module works slave mode only and supports AT commands. By default the baud rate of device is set to 9600, however it can be set to a different rate by using AT commands.

Figure 2-3: HC-06 Bluetooth Module

The module has four external pins; VCC : Power Supply GND : Ground TXD : Transmitter RXD : Receiver

The pin which TXD module connects to must be set to PULLUP mode to ensure reliable data transfer.

2.4 Digilent Nexys 3 FPGA Board

The Nexys 3 is a complete, ready-to-use digital circuit development platform based on the Xilinx Spartan-6 LX16 FPGA. The Spartan-6 is optimized for high performance logic, and offers more than 50% higher capacity, higher performance, and more resources as compared to the Nexys 2's Spartan-3 500E FPGA.

Features include:

- Xilinx Spartan-6 LX16 FPGA in a 324-pin BGA package
- 16Mbyte Cellular RAM (x16)
- 16Mbytes SPI (quad mode) PCM non-volatile memory
- 16Mbytes parallel PCM non-volatile memory

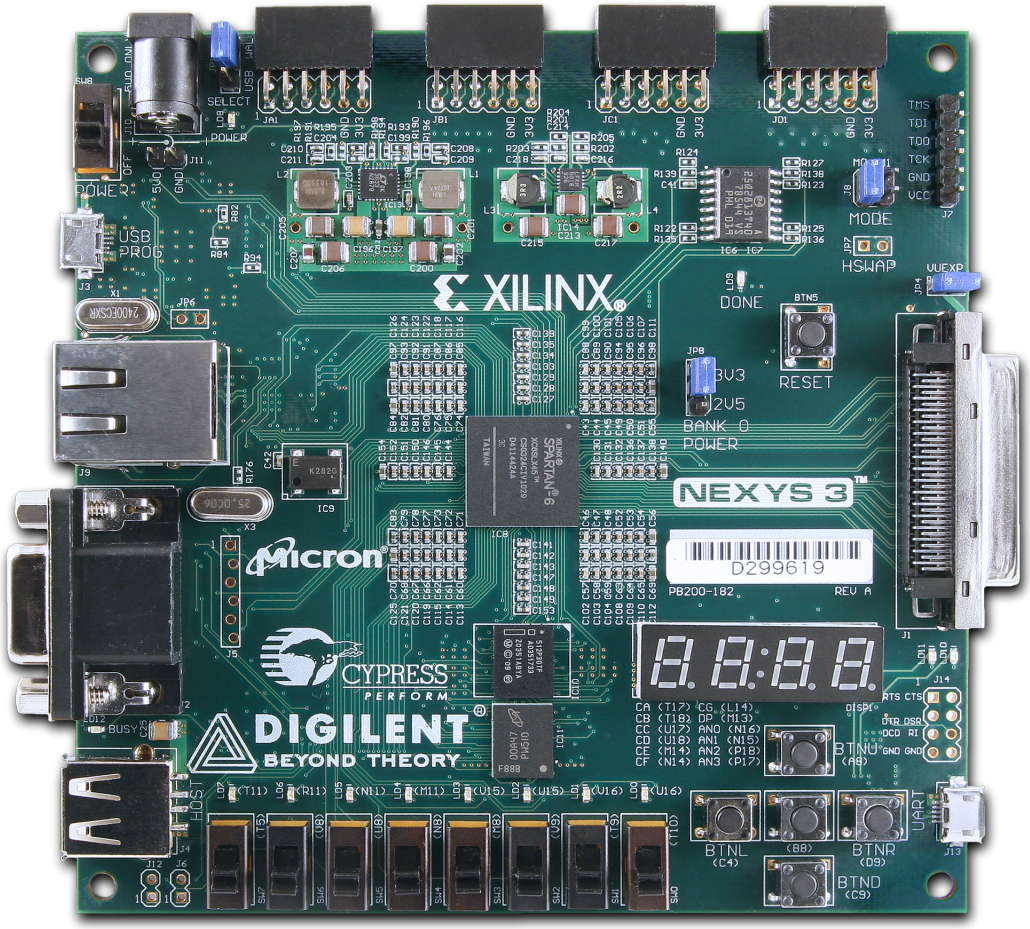

Figure 2-4: Nexys 3 Spartan 6 Development Board

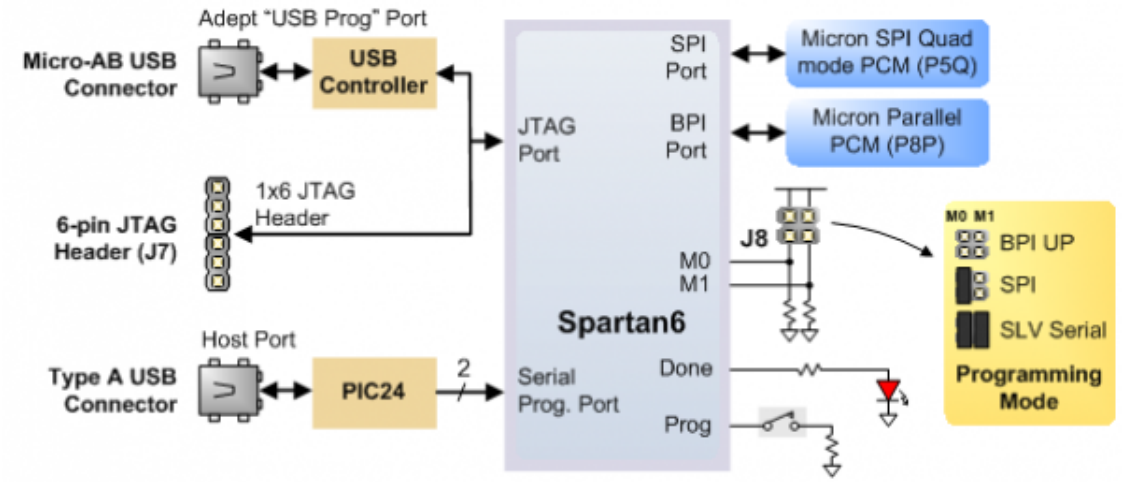

Figure 2-5: Nexys 3 Configuration

- 10/100 Ethernet PHY
- USB-UART and USB-HID port (for mouse/keyboard)
- 8-bit VGA port
- 100MHz CMOS oscillator
- 72 I/Os routed to expansion connectors
- GPIO includes 8 LEDs, 5 buttons,8 slide switches and 4-digit seven-segment display

2.5 Software

Two different software developed for the complete project. One is written in C and compiled using Microblaze Toolchain provided by Xilinx EDK [4]. The other written in Java and compiled with Android SDK provided by Android Studio [2].

2.6 Android

Android is an open source, Linux-based software stack created for a wide array of devices and form factors.

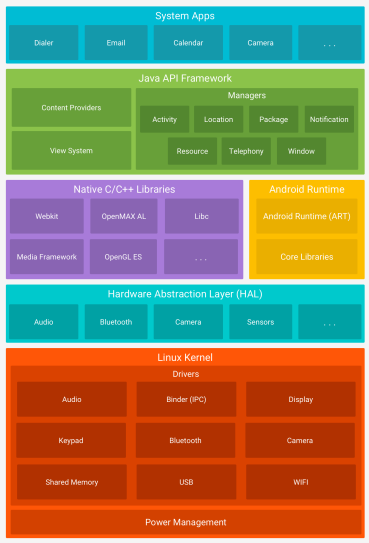

Figure 2-6: The Android software stack

Android provides a rich application framework that allows to build innovative apps and games for mobile devices in a Java language environment. Android apps are built as a combination of distinct components that can be invoked individually. For instance, an individual activity provides a single screen for a user interface, and a service independently performs work in the background.

Android apps are written in the Java programming language. The Android SDK tools compile code along with any data and resource files into an APK, an Android package, which is an archive file with an .apk suffix. One APK file contains all the contents of an Android app and is the file that Android-powered devices use to install the app.

App components are the essential building blocks of an Android app. Each component is an entry point through which the system or a user can enter your app. Some components depend on others.

There are four different types of app components:

- Activities
- Services
- Content providers

- Broadcast receivers

Each type serves a distinct purpose and has a distinct lifecycle that defines how the component is created and destroyed. Three of the four component typesâĂŤactivities, services, and broadcast receiversâĂŤare activated by an asynchronous message called an intent. Intents bind individual components to each other at runtime.

An Android app is composed of more than just code, it requires resources that are separate from the source code, such as images, audio files, and anything relating to the visual presentation of the app. For example; animations, menus, styles, colors, and the layout of activity user interfaces can be defined with XML files. Using app resources makes it easy to update various characteristics of an app without modifying code. Providing sets of alternative resources enables to optimize an app for a variety of device configurations, such as different languages and screen sizes. [1]

Chapter 3

Hardware Design

3.1 Objective

Design goal is to build a system that is capable of receiving data via Bluetooth and save the data to an external memory. As a separate functionality, reading data from external memory and displaying it with an LCD module is required.

3.2 Overview

The input interface of system is a Bluetooth module, with this module the system can communicate with the outside world. The Bluetooth module sends and receives serial data and is connected to processor via an UART module. UART module is controlled by the processor. The processor saves the data on an external BRAM block using a Bram Controller. In order to provide a user feedback a LCD is used. This LCD is connected to the processor via another UART module. In a typical use case scenario, data is received via Bluetooth, processed by the processor, saved and then displayed on LCD.

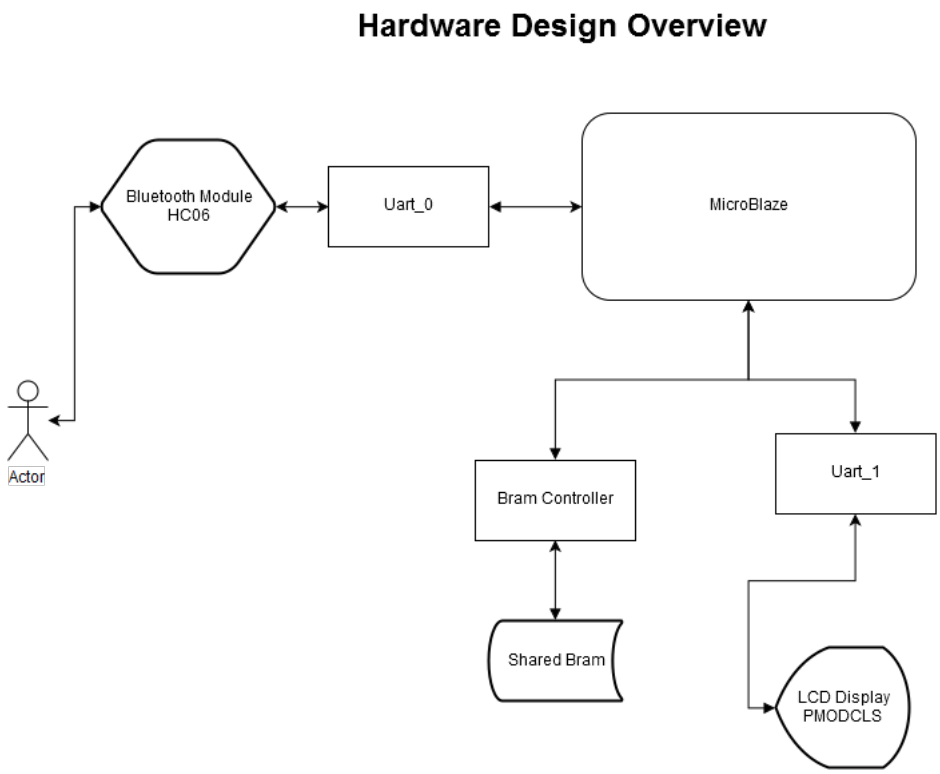

Figure 3-1: Component Diagram

3.3 MicroBlaze

As processing unit a MicroBlaze soft core processor is implemented on the FPGA in order to utilise other IP blocks and manage data transfer via Bluetooth. By using Xilinx Platform Studio tool MicroBlaze processor can be customized to fit design requirements. Any number of IP blocks can also be added as long as capacity of FPGA permits. Moreover interconnections between these components via memory bus can be specified. Since MicroBlaze only has one interrupt input, an interrupt controller needs to be included into design in order to support multiple inputs. These interrupts and their priorities are configured using the visual interface.

3.3.1 Processor Design

During the design process two UART IP modules added to the design in order to utilize the Bluetooth module and LCD. A Bram block and a Bram Controller is

added to design for the data storage. The MicroBlaze processor and the modules mentioned above are connected over the AXI bus.

3.4 Bluetooth Connection

As a Bluetooth module, HC-06 is used. It has its own buffer which it can use to save and receive data. The TX and RX pins are connected to the UART module. Any received data over the Bluetooth comes to the FIFO receive register of the UART module. Similarly, any data to be send first written to FIFO send register of the UART module and then sent via the Bluetooth module. UART Controller is programmed and utilized by the software written on the processor. Interrupt mode of the Bluetooth controller is used for receiving data, on the other hand for sending data Polling mode is used.

3.5 Data Storage

Storing data in a memory unit outside the memory map of the processor is one of the key aspects of the project. This way the system can share data with other systems without corrupting the memory of the processor. In order to do this a Bram block and a Bram controller are used.

The system accepts 8-bit data from user however Bram block has 32-bit registers. In order to utilize the memory more efficiently an algorithm is developed to write four 8-bit unsigned integers on a single register and keep the track of the memory addresses. This algorithm also provides the necessary information to read a specific data from a specific register.

3.5.1 Functional Description

At runtime processor receives data via its Bluetooth interface and keeps it on its own memory. After initial processing of the data it is saved on the Bram block by utilizing Bram Controller. Bram Controller has a vendor library which provides a vast number

operations on the Bram block. With this library writing data on Bram becomes a simple task of writing data on a memory address provided by the Bram Controller.

3.6 Display

In order to provide a useful feedback to user a LCD module is used. It is connected to a the processor via UART interface. It allows user to see what data is saved on BRAM block. The module chosen for this project is PmodCLS. It supports the UART protocol and some vendor provided basic commands.

3.7 Software

All of the software is written in C language using the tools provided by the vendor. Software is divided in subsystems corresponding to the hardware modules. One of the biggest challenge was to make sure all the components work as intended in cooperation with each other. Therefore software needed to be designed with extra care and detail.

Bibliography

[1] Android api guide.

[2] Android studio.

[3] Microblaze processor reference guide.

[4] Xilinx design tools.

[5] Max Maxfield. Lattice introduces ice40 ultraplus high-performance low-power fpgas.

[6] Xilinx Yvonne Lin. Using fpgas to solve challenges in industrial applications.