DSPLab
Hardware Manual

Second Edition
Kyanoosh Shafaei

DSPgig.com

Registered at Library of Congress.

Second printing edition 2019.

kshafaei@dspgig.com

www.dspgig.com

IMPORTANT NOTICE

DSPgig, Inc. reserves the right to make changes to its products or to discontinue any product or service without notice. Customers are advised to obtain the latest version of relevant information to verify that the data being relied on is current before placing orders. DSPgig, Inc. warrants performance of its products and related software to current specifications in accordance with DSPgig's standard warranty. Testing and other quality control techniques are utilized to the extent deemed necessary to support this warranty. Please be aware that the products described herein are not intended for use in life-support appliances, devices, or systems. DSPgig does not warrant nor is DSPgig liable for the product described herein to be used in other than a development environment. DSPgig, Inc. assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does DSPgig warrant or represent any license, either express or implied, is granted under any patent right, copyright, or other intellectual property right of DSPgig, Inc. covering or relating to any combination, machine, or process in which such Digital Signal Processing development products or services might be or are used.

WARNING

This equipment is intended for use in a laboratory test environment only. It generates, uses, and can radiate radio frequency energy and has not been tested for compliance with the limits of computing devices pursuant to subpart J of part 15 of FCC rules, which are designed to provide reasonable protection against radio frequency interference. Operation of this equipment in other environments may cause interference with radio communications, in which case the user at his own expense will be required to take whatever measures necessary to correct this interference.

Contents

Preface

1) Introduction

DSPLab is a training package for TI-DSP processors. This book is one of five documents that comes with DSPLab. It focuses on hardware training for TI-DSP. Each chapter starts with hardware design and ends with the software required for one sample hardware.

The processor used in DSPLab hardware is from the 55xx family, and as a result, the code example in this book is for the specific processor used in DSPLab. But the hardware design is not specific to a single family and is more about all major families (such as 2000, 5000, and 6000). The goal is to teach the students how to read the schematics, which come from different DSP families, and design their first DSP hardware.

Each chapter is about one of the peripherals and how to design hardware and software. The software approach changes in every chapter in order to teach a new technique or tool. The book is all about giving students the experience they need to start a real system design. It is not about copying a schematic from another design.

2) Who should use this book?

The first step in DSP training is learning the software. Therefore, it is highly recommended that this book is used after review of the "DSPLab User's manual." Hardware design is not for everybody; some students have more passion for hardware, and will often have more experience. Although DSP hardware design is recommended for those who have designed at least one digital hardware before, even if this is your first hardware design, you are reading the right book.

3) Why is DSPLab hardware inside an aluminum box?

Both hardware and software for TI-DSP can be complicated. Showing complex hardware to the students who have just come to the lab to learn DSP does not help them to learn DSP at all. Instead, it just creates fear and pushback.

DSPLab hides the complex hardware from the students in their first step of DSP learning. After students get familiar with the software, then they can switch their attention to hardware.

In addition, DSPLab hardware is designed to be part of the DSP laboratory for years to come and so should be kept safe. The DSPLab is not a replacement for other DSP hardware. It is intended to be used in the DSP laboratories for the first 50 hours of the lab, and then students can switch to different hardware platforms for their final projects.

4) Why does DSPLab hardware use TMS320VC5509A as the processor?

Consider a laboratory for an intel core-i7 processor. If someone has no previous knowledge of intel processors, a good start is to first learn older generation processors such as 8086 or 80386 or Pentium. Going through each new generation and learning the upgrades to that family creates the foundation for learning more advanced systems. This is also true for beginners who want to learn the TI-DSP processor. A good starting point is a family with all the foundations which is not too complicated. The 55xx family does not have the complexities of the c6xxx family, while still being a powerful processor.

With a little patience, students will soon learn enough to switch to the best hardware for their target application.

5) Chapter introductions

‘Chapter 1: Hardware’ is about the basics of DSP hardware design. It covers a minimum system design for DSP hardware.

‘Chapter 2: JTAG’ introduces the JTAG connection and software setting for JTAG.

‘Chapter 3: PLL’ is for DSP core clock design.

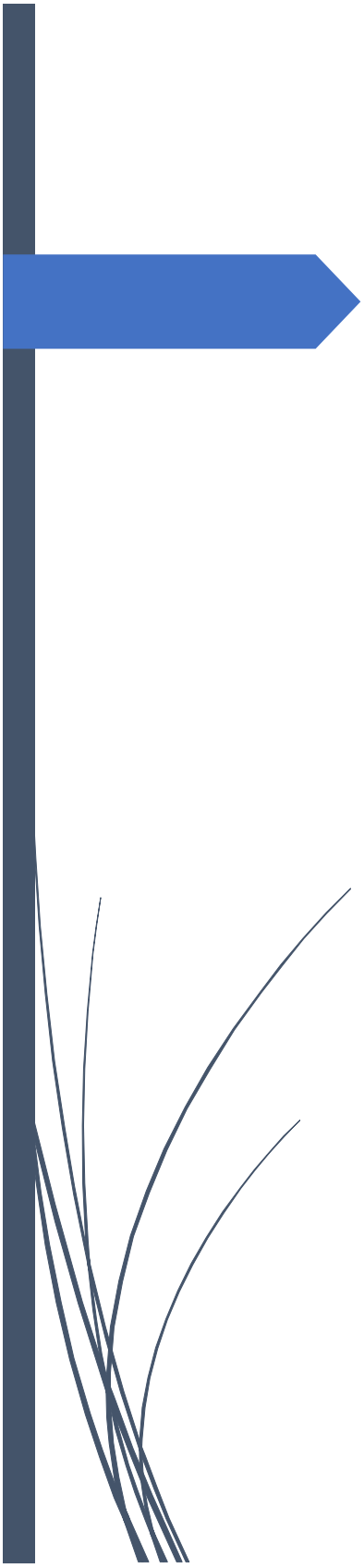
‘Chapter 4: Digital IO port’ shows how to design an external digital input or output, which is even easier to use than the internal GPIOs.

‘Chapter 5: DSP+FPGA’ explains how to connect an FPGA to a DSP. The FPGA can be used as a powerful coprocessor for a DSP. The chapter also covers some examples of mixed FPGA and DSP designs for ADC, DAC, and more.
‘Chapter 6: CODEC’ is mostly about an example of a CODEC design for a DSP and also covers what is inside a CODEC.
‘Chapter 7: MMC‘ shows the design of an MMC connection to a DSP and how to use TI packages to access the MMC.
‘Chapter 8: External Memory’ explains why most of the complexity of memory design is in PCB, not in the schematic. It also has code examples for memory initialization.
‘Chapter 9: USB’ is an example of another standard peripheral and how standard software can be used for that peripheral.

6) What is not covered?

The Signal Integrity(SI) and the Power Integrity(PI) analysis are needed for some very high-frequency signals. These are part of the PCB design and are not part of this document.

This book tries to cover as much as possible in less than 200 pages, but we recognize it is far from perfect. Please share your ideas with us, and we will do our best to cover more in future editions.



Chapter 1

Hardware Design

DSP Gig Co.
Kyanoosh Shafaei

1)Introduction

DSP hardware design is always a tough challenge. There are many resources available that can help with hardware design. TI and some other companies such as Spectrum Digital have many free schematics for different hardware. Use these schematics always is a good start for the beginners and even professionals. However, understanding the DSP hardware design is very important to be able to use the available resource properly.

In this chapter, we discuss a bare minimum system that has only one DSP processor. Many DSP processors have internal memory and a JTAG interface. A processor, memory, and a tool to load the code into memory are the three main minimum requirements for a minimum embedded system.

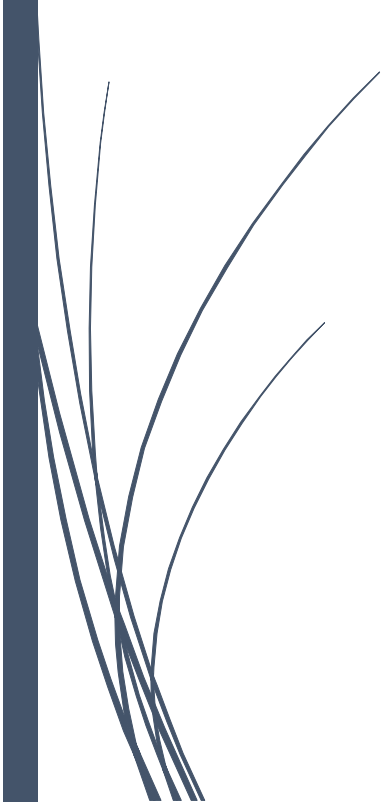
Put the hardware design knowledge in one place is not easy. In each step of hardware design, there are multiple options and tradeoffs. This chapter is intended for students who have never designed DSP hardware. This chapter can be used as a checklist for each step of the hardware design.

2) Step1: Choosing the DSP processor

In any DSP system, choosing the right processor can save designer time and effort. There is always a tendency to choose more advanced DSP, but whatever choosing, it is better to be from famous DSP processors. Especially for the beginners, selecting a DSP which is already used by many other designers provides them with many resources. That means going to the TI Web site and choosing from the list of the processors is ***not*** a recommended method for the beginners. Instead, it is better to look for any processor that already has a development board in TI[1] or Spectrum Digital[2] website. These processors are being used by other designers, which means there are many technical resources available for them.

[1] - www.ti.com
[2] - www.spectrumdigital.com

Chapter 2
JTAG Connection



DSP Gig Co.
Kyanoosh Shafaei

1)Introduction

JTAG is always needed when debugging the code for DSP processors. The JTAG for TI DSP processors is a board, which in most cases, utilizes a 14-pin connection[1] to communicate with the DSP processor. The communication standard is IEEE 1149.1[2] , which is employed by many other companies to build JTAGs. Be careful that the JTAG of no other company can be used to communicate with the TI DSP processors. For many years, TI has created professional JTAGs for debugging and communicating between the computer and processor, and CCS can only use these JTAGs to exchange data with TI processors. Figure 1 illustrates a 14-pin JTAG communication.

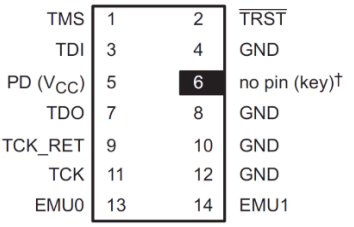


Figure 1: The JTAG 14-pin connector for the TI DSP board and the pins name

The JTAGs made for the DSPs are divided into three categories:

A-Series 560

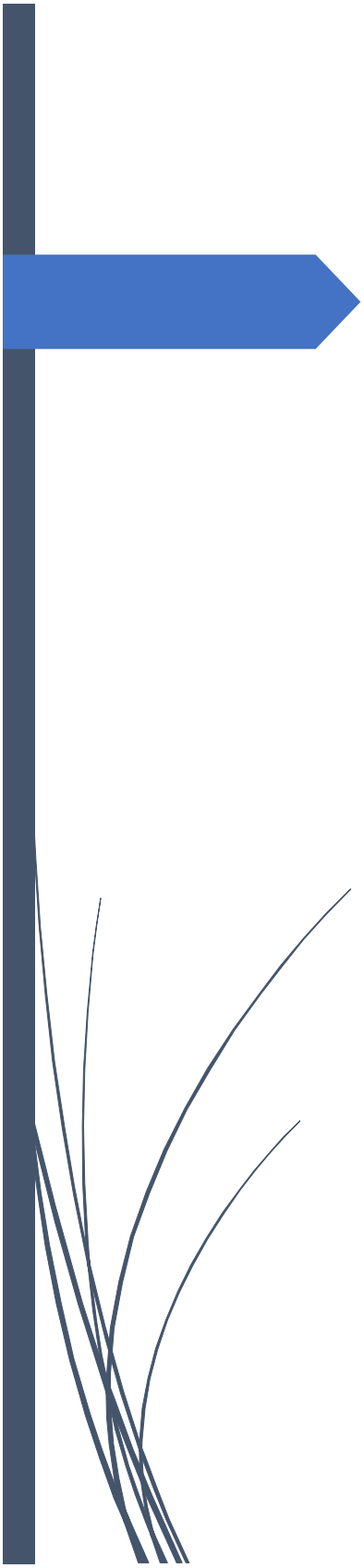
This is the most complete and advanced JTAG built for TI DSPs with the highest connection speed. With this JTAG, it is always possible to increase or decrease the JTAG communication speed[3] with the board. This feature is

[1] - For TI processors 10, 20, and 60 pins version of JTAG connection is also available. Each processor usually supports one of these standard connections. There are few processors that support multiple connections like 60 and 14 pins.

[2] - IEEE 1149.1 is an industry-standard to test PCB and allow on-chip debugging. Many manufacturers have used this standard for the JTAG interface. Although many companies use this standard for hardware debugging, their JTAG is not compatible with each other, and each manufacturer usually has its own JTAG.

[3] - The JTAG has a dedicated input clock pin (TCK_RET) to synchronize data transfer to the board. Also, it has an output clock pin (TCK). By default, the output clock of JTAG must be shorted to the JTAG input clock inside the target board. This means the

Chapter 3
PLL



DSP Gig Co.
Kyanoosh Shafaei

1) What is PLL[1]?

PLL is an analog circuit that can convert a low-frequency clock to higher frequencies in DSPs. In many DSP, there is an internal PLL. Because DSPs usually work with high-speed clocks (higher than 100 MHz to several GHz), it is hard to make a circuit to generate these high frequencies outside the chip. For this reason and some other practical reasons, it is better to use a low-frequency crystal and then, by the help of internal PLL, multiply the crystal frequency to a higher (or even lower) frequency. Thus, changing the processor frequency can be done by the software. This is a handy feature. For example, in many low power applications, the software needs to dynamically change the CPU clock and run DSP with different frequencies.

This chapter is the only chapter that the internal registers are explained with more details. This may look like the datasheet of 5509A, but the registers are explained from a more practical point of view. The goal is to show how the clock circuit of a processor works and what the programmer needs to consider. This is also the first example of programming a peripheral. So both assembly and C (CSL) methods are explained.

Due to the project limited time, usually, there is a tendency to skip reading some documents. PLL is one of the easiest peripherals in the DSP, but here, we want to show, it is important to read the datasheet carefully, even for a simple peripheral.

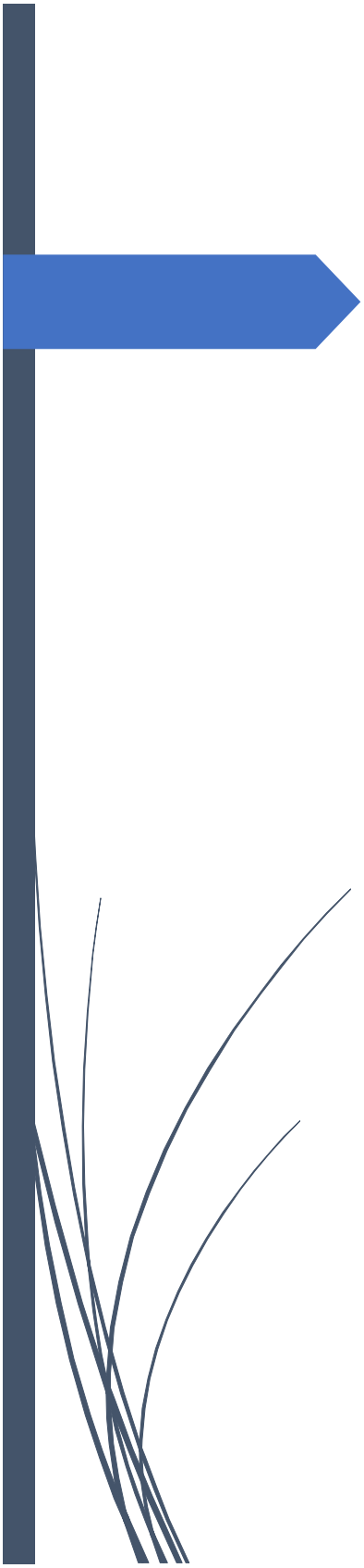
2) The frequency range of the board

There are two crystal oscillators[2] inside the mainboard. One of the crystal oscillators generates 12.5 MHz (XTL1), and the other one is not soldered (XTL2). The 12.5 MHz oscillator is directly connected to TMS320VC5509A (central processor), CODEC, and FPGA. That means always after reset, the operating frequency of the DSP is 12.5 MHz. The second crystal oscillator

[1]- Phase Lock Loop

[2]- Combination of a Crystal and an oscillator circuit is called crystal oscillator.

Chapter 4
Digital IO Port



DSP Gig Co.
Kyanoosh Shafaei

1) Introduction

DSPLab has one 8-bit digital input and one 8-bit digital input. Access to these ports is easy: Both have an address in memory. For example, writing to digital output is done by writing to a specific address in memory. It is very similar to accessing a variable in memory. The only difference between the port address and the variable address is that there are multiple port addresses for each physical port. In other words, each variable has one address in memory, but each port has multiple addresses. This is due to the hardware design of the board and is a common practice to have less complicated hardware.

In this chapter, the hardware design is explained, and then the software code to access the port is shown. The goal of this chapter is to show how to add a simple IO port to a DSP hardware using the EMIF(external memory interface).

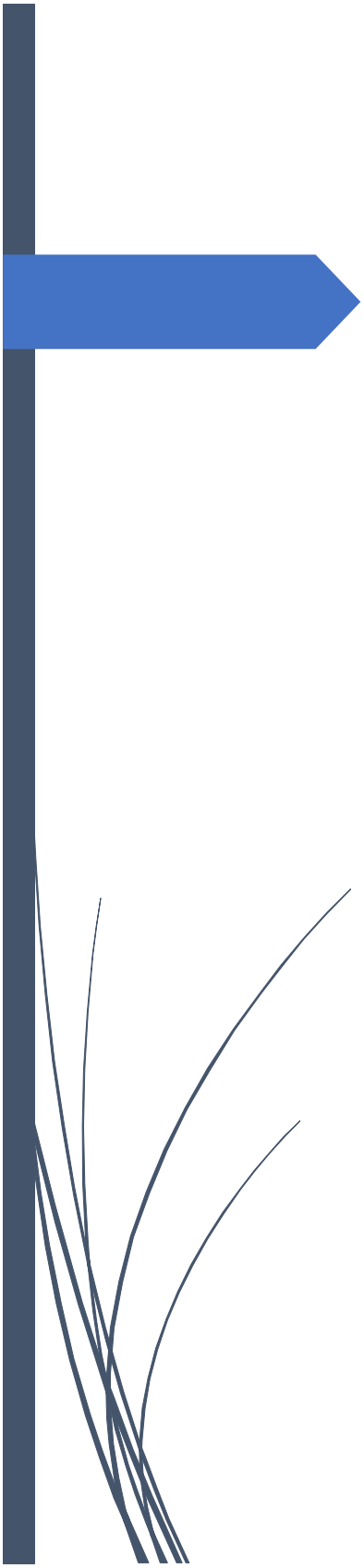
2) The input port

The input port in the DSPLab board is connected to 8 keys and 8 LEDs. The port is also connected to a connector called 'In' (which is not accessible by the user).

The reason the following details are explained here is to provide an example of how to design a Digital Port in the actual hardware. Using the 'In' connector will void the warranty of the DSPLab and is not recommended at all.

In the following figure, the place of LEDs and the connector of the Input port (In) are shown. The next board is not physically accessible by the user and is located inside the DSPLab aluminum box. Only the Keys and LEDs are accessible from the front left side of the box.

Chapter 5
DSP+FPGA



DSP Gig Co.
Kyanoosh Shafaei

1)Introduction

Inside DSPLab hardware, there are two DSPs and two FPGAs. The user only has access to one of the DSPs. The other DSP and two FPGAs are for creating essential peripherals such as ADC, DAC, UART, Image, PLL, and timers. On the top side of the DSPLab aluminum box, the schematic for most of the peripherals is shown.

Designing DSP and FPGA hardware is beyond the scope of this book, but in this chapter, some of the benefits of a mixed DSP & FPGA hardware are explained. Also, all the registers and their programming are covered.

2) DSP + FPGA: Why and where?

To answer this question, we should compare the two languages use for DSP and FPGA. The C language is currently used in almost all embedded programming software. It has a rich standard library and lots of custom libraries. On the other side, the HDL languages, such as VHDL or Verilog, are for hardware design and can simulate what traditionally done by other hardware. Having programmable hardware (HDL) and software (C) in one platform can open new doors to new possibilities.

For example, assume a high-frequency bandlimited signal is sampled by a very high-frequency ADC (next picture).

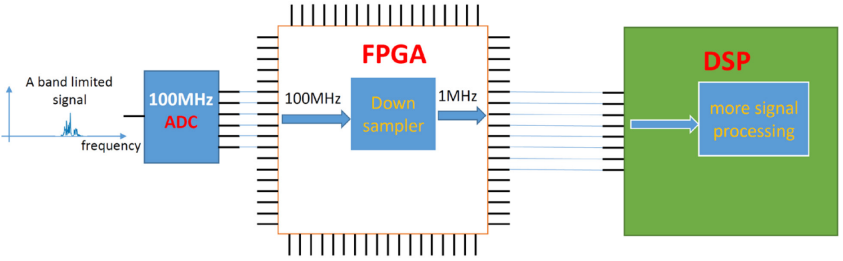
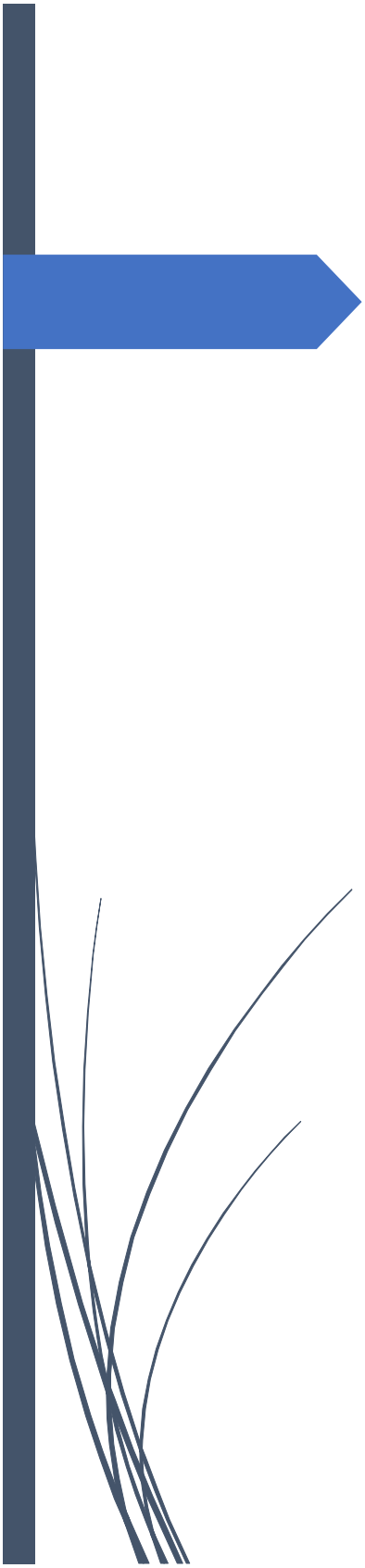


Figure 1: A simple combination of DSP and FPGA

Because the signal is bandlimited, it can be downsampled before being processed. Downsampling is just a simple filter, but because it works on a high

Chapter 6
CODEC



1) Introduction

One of the main applications for DSP Processors is audio signal processing. DSPs are idle for audio signals. They have many special instructions for basic and some advanced signal processing algorithms.

This chapter focuses on minimum audio processing hardware. Audio processing hardware needs at least a processor and an analog frontend to interface to analog voice signals.

The code generates in this chapter(an automated code generator) is a perfect example of how professionals write their codes to program peripherals. This code here is a perfect example of a well-structured code for firmware development.

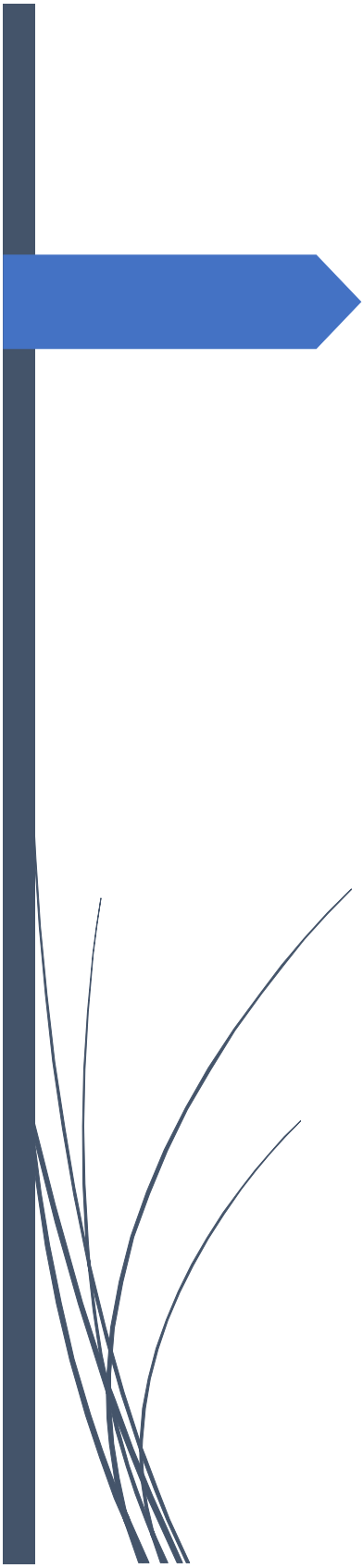
2) What is CODEC?

CODEC[1] is a two-way analog to digital converter. A CODEC has ADC[2] for incoming and DAC[3] for outgoing analog voice. Many audio systems have two-way audio communication. In a typical audio system, there is a need to generate an audio signal and simultaneously sampling and processing another audio signal. CODEC is a complete circuit for audio frontend. A CODEC has an input filter to remove high frequencies, an analog to digital converter, a digital to analog converter, and an analog output filter.

[1] - Compression / DECompression (usually for audio signal)
[2] - Analog to Digital Converter
[3] - Digital to Analog Converter

Chapter 7

MMC



1)What is MMC?

MultiMediaCard (MMC) is a flash memory card. Typically, an MMC is used as storage media for a portable device, in a form that can easily be removed for access by a PC.

Secure Digital (SD) is a flash (non-volatile) memory card format suitable for large storage. MMC and SD cards differ in their physical size, capacity, and usage. While MMCs are compatible with a standard SD card slot, but SD card cannot be used in an MMC slot. That means hardware capable of accessing the SD card can also read MMC. TMS320VC5509A can read and write both MMC and SD cards.

2) Communication Protocols of MMC

two protocols are available to access a MultiMediaCard (MMC) by DSP:

1- Native Protocol
2- SPI Protocol

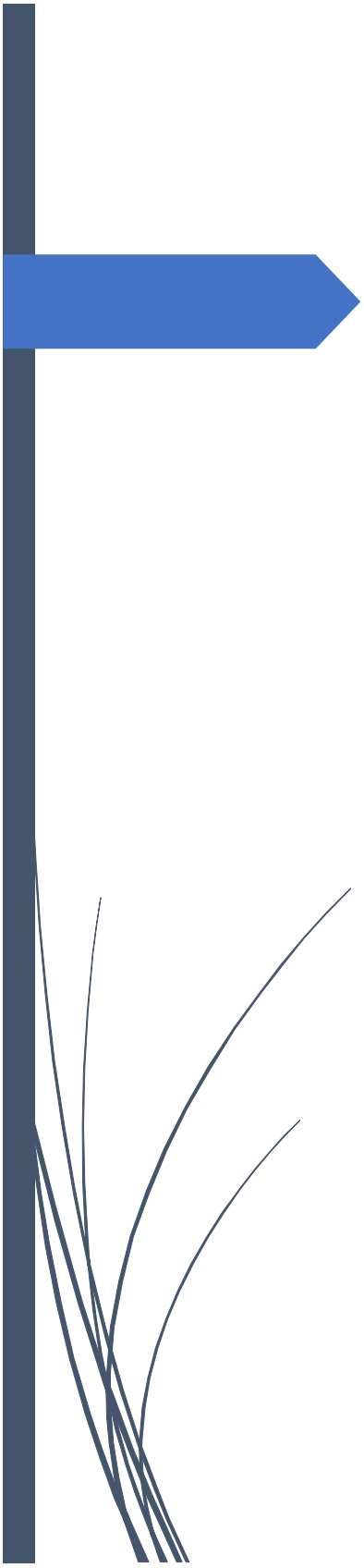
2.1) Native Protocol

This protocol can be utilized for both MMC and SD memories. In MMC, data are transmitted and received through one data line while SD card benefits from 4 parallel data lines, which improves data speed.

MMC uses only 3 pins in Native protocol:

1- CLK: is an output pin that provides the MMC clock.
2- CMD: is for command exchange and is Input-Output-High impedance (I/O/Z) pin.
3- DAT0: is considered as the data transmission line and is Input-Output-High impedance (I/O/Z) pin.

These pins are illustrated in Figure 1.



Chapter 8

External Memory

1) Introduction:

One of the main parts of DSP hardware is its external memory. Usually, signal processing algorithms require a lot of memory, but the internal memory of the DSP processors rarely exceeds one megabyte. For example, the 5509 has 256 KB of internal memory, which is among the largest in the 5000 series. However, in many applications, this memory is not enough. In many projects, due to the high capabilities of the DSPws processor, a large part of the applications is executed on the DSP processor. A small team of two or three engineers can easily consume this memory in less than a year.

For this reason, in many cases, it is necessary to add a high-speed auxiliary memory to the board. The speed of this memory should be close to the DSP processor speed, else the memory access would be slowed down. Two types of memory can be connected to the External Memory InterFace (EMIF) of TMS320VC5509A: SRAM and DRAM. The DRAM is recommended due to its higher speed.

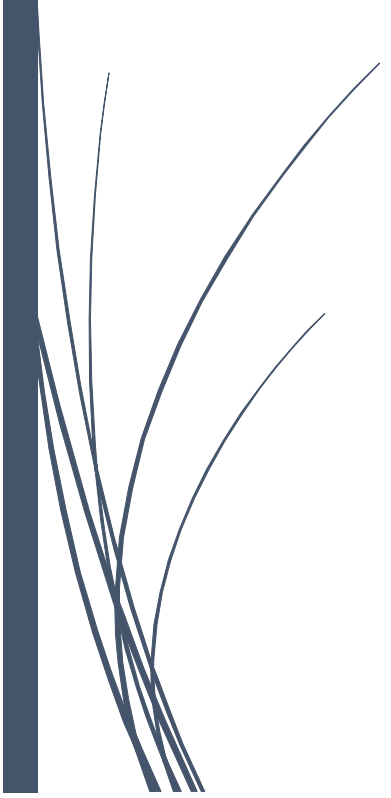
In this chapter, an external memory hardware design is reviewed. Then the required software to initialize the memory interface is explained. The purpose of this chapter is to show how to begin designing an external memory for a DSP board.

2) What is DRAM[1]?

The first semiconductor memory came to market was SRAM[2]. SRAM memory is one of the most popular memories used in various systems. This memory can hold written information until the memory power is not unplugged. After SRAM, DRAM[3] came to market. DRAM needs refreshing to keep

[1] - Dynamic RAM
[2] - NASA first used SRAM in the ILLIAC IV supercomputer in 1973. They use an 8 bytes SRAM chip from Fairchild and build 16k bytes memory card.
[3]- The first DRAM came to commercial computers in 1976 by Trade Ministry(a Japanese company). It was so successful and threaten American memory suppliers that caused some political dispute between US and Japan.

Chapter 9
USB



1) Introduction

There is a USB connector inside the board which is connected to the USB port of the TMS320VC5509A processor. The port is a convenient interface for PC connection. To set up a USB port, in addition to the code running on TMS320VC5509A, an application must be written in the Windows environment, which talks to the USB port.

The easiest way to load the code into DSP memory is JTAG. However, USB can also load the code. The only drawback is the lack of debugging. In this chapter, first using USB for loading the code into DSP memory is shown. Then more details on how to transfer data between PC and DSP board using USB port are explained.

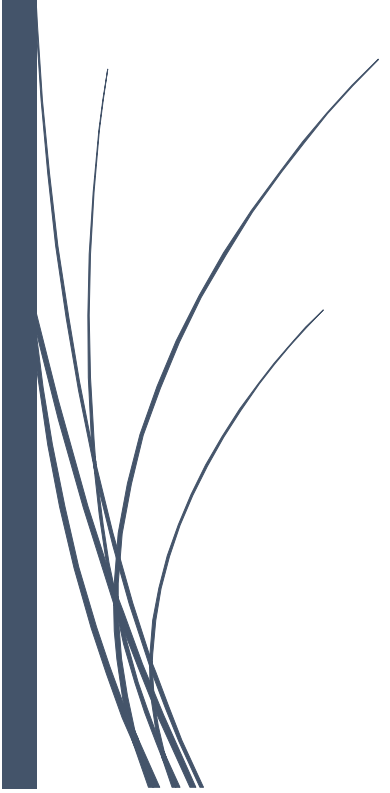
DSPLab hardware has two USB ports. One at the back for JTAG connection and one on the top of the box, which is connected directly to the USB pins of the TMS320VC5509A. This chapter explains how to use the second USB port.

In this chapter, first, USB loads a program into DSP memory (similar to JTAG), and then USB is used to transfer packets between DSP and PC.

2) Use USB as a Bootloader

TI has created a software called 'USBIOAPP.EXE' to load the code into DSP memory. This program first finds the DSP board connected to the USB port and then loads the application from the computer hard disk to the board. The 'USBIOAPP.EXE' simply sends the code from the computer to the USB port. On the other side, there is a code running in TMS320VC5509A that receives the file from the USB port and then copies it into DSP memories. This process is described step by step.

Appendix A
CCS3.3 JTAG connection



DSP Gig Co.
Kyanoosh Shafaei

The DSPLab hardware has an embedded JTAG and can not connect another JTAG to it. This Appendix is for using an old version of JTAGs with any available hardware. Many old JTAGs can not be used with new version of CCS(CCS4 and above).

1) JTAG connection for CCS 3.3 and older

In this Appendix, the JTAG setup process for an older version of CCS (CCS3.3 and older) is explained.

Before buying a new JTAG, first, make sure it is compatible with the older version(e.g., CCS3.3). Usually, many new JTAGs are not compatible with the older version anymore. Also, it is important to install the SETUP file that comes with the JTAG before proceeding to the next step. For non-TI JTAG, usually, there is a Setup file.

Then, connect the JTAG to the computer via a USB cable. When the 'Found New Hardware' window appears, specify the JTAG driver from the JTAG CD/DVD.

Now make sure Windows detects the new hardware correctly and finds all drivers.

1.1)Start 'Setup' processes inside CCS3.3

Run the CCS3.3 software Setup[1] by clicking on the following icons.

Figure 1: The icon for running Setup in CCS version 3.3 and older

[1] - This is the CCS Setup software and is different than the Setup file(*.exe) which is for JTAG installation and is part of the JTAG setup process.