

UNIVERSITY OF CALIFORNIA AT BERKELEY
COLLEGE OF ENGINEERING
DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Checkpoint 1

AC97 Audio

1.0 Motivation

One of the most difficult aspects of digital design is interfacing with a circuit or IC designed by someone else, however it is also the most important part. For this reason, a major part of the EECS150 project is interfacing your circuits both to those designed by your partner, who you can work with on debugging, and to those designed by professionals with whom you cannot interact, save through their documentation (a good reason to take E190 seriously).

In this checkpoint, for the first time we are asking you to read a datasheet, a complete specification of the operation of an integrated circuit: in this case the LM4549A AC97 audio codec. Armed with this specification, you will proceed to build a controller to initialize and then provide PCM audio data to that chip. The specification can be downloaded from <http://www-inst.eecs.berkeley.edu/~cs150/Documents/LM4549A.pdf>. It will be especially useful to look at pages 15-24 in the specification document.

At the end of the checkpoint, you should be able to record and playback audio by using the push buttons on the FPGA and use the dipswitches to toggle the volume levels.

2.0 Introduction

One of the primary reasons for building the CaLinx boards was the availability of such interesting audio and video I/O chips as the **LM4549A AC97 Audio Codec** you will be controlling in this checkpoint. This codec, and others like it, have been used in computers for years.

The **FPGA** and the **LM4549A** are connected by a relatively simple digital interface consisting of a **12.288MHz** clock generated by the Codec, **two bit serial data lines** and a **sync** signal to synchronize the data between the FPGA and Codec.

For the first part of this checkpoint your primary goal is to work on generating the **AP_RESET_**, **AP_SYNC** and **AP_SDATA_OUT** signals which go to the Codec. For more information about these signals, see the section 2.3 AC97 Interface Spec below.

2.1 Checkpoint #1 Overview

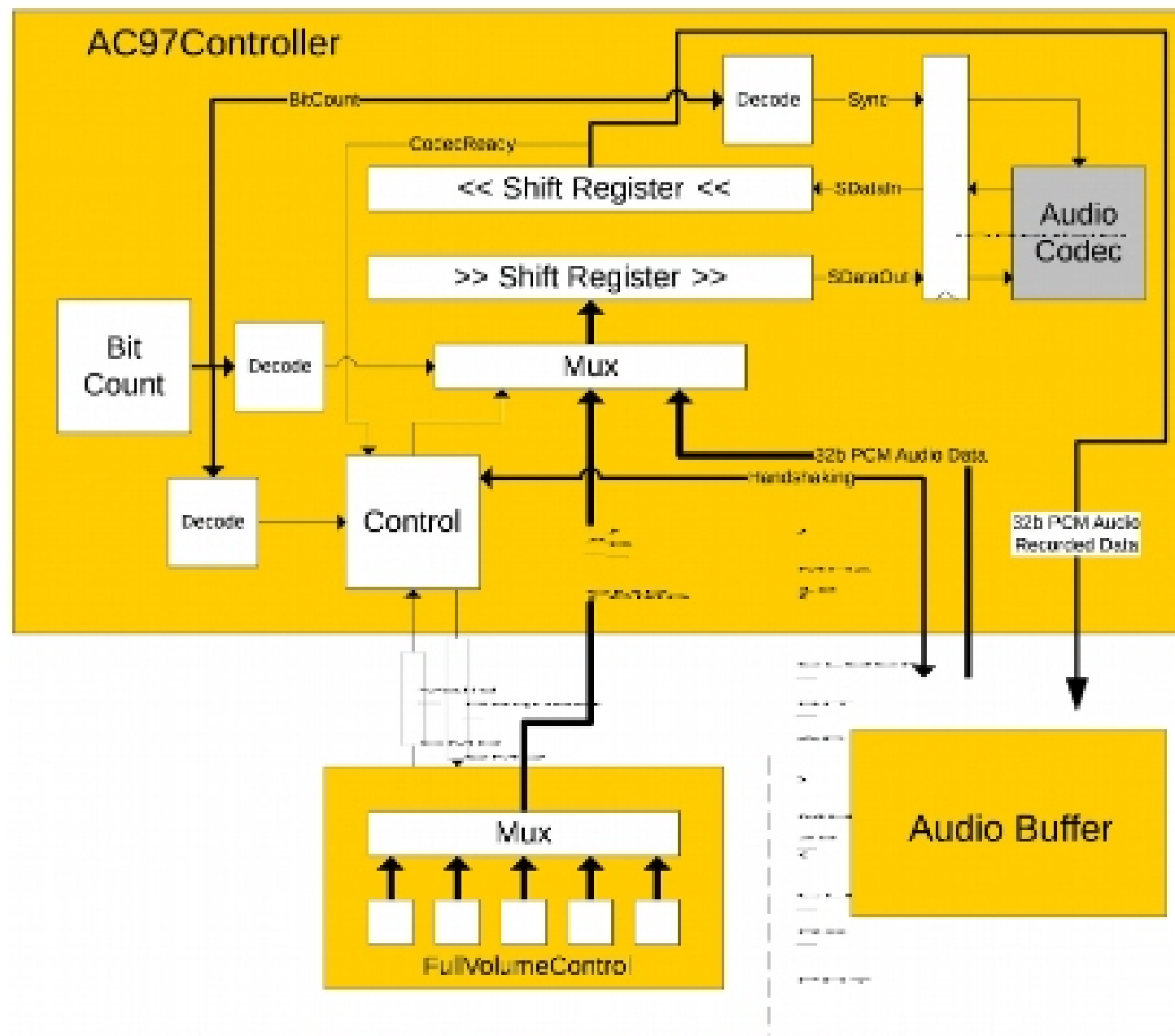


Figure 1: Checkpoint #1 Overview

Shown in Figure 1, above is the basic organization we would suggest for this checkpoint. Feel free to deviate from it, as it is neither perfect, nor optimal, it is merely a vague idea to get you started thinking for your design review.

At the core, the AC97 controller is effectively a parallel to serial converter designed to convert the 32b PCM samples into 256b AC97 frames. Its other duties include generating the proper synchronization and reset signals required for correct operation of the LM4549A.

Please note the IORegister module shown in the above diagram. This module will act like a **standard register** in that it will **delay all of your signals** between the LM4549A and your AC97Controller. However if you look inside the module you will see the text `// synthesis syn_useioff = 1`, which will cause the synthesis and PAR tools to **place that register at the very edge of the FPGA** just as the signals leave and enter the chip. **This register is required for timing reasons**, without it your design will work only occasionally and only on certain CaLinx2 boards.

2.2 AC97 Audio Codec

The AC97 Audio Codec is simply a chip meant to convert between analog audio signals such as those going to the speakers/microphone and digital PCM signals which are easy to work with on an FPGA or in a computer.

If you are interested in any of this, take a look at the analog sections of the datasheet. In this lab we will focus on how to use the chip, not what it can do or how it works.

2.2.1 Block Diagram (Page 2 of the LM4549A Datasheet)

Below is the block diagram of the audio paths in the codec. It shows the inputs, outputs, volume controls and mixers. Notice that the signals should remind you of those on a common PC volume control; however we will not be using most of them. In fact many of the analog inputs are tied to ground on the CaLinx2 boards.

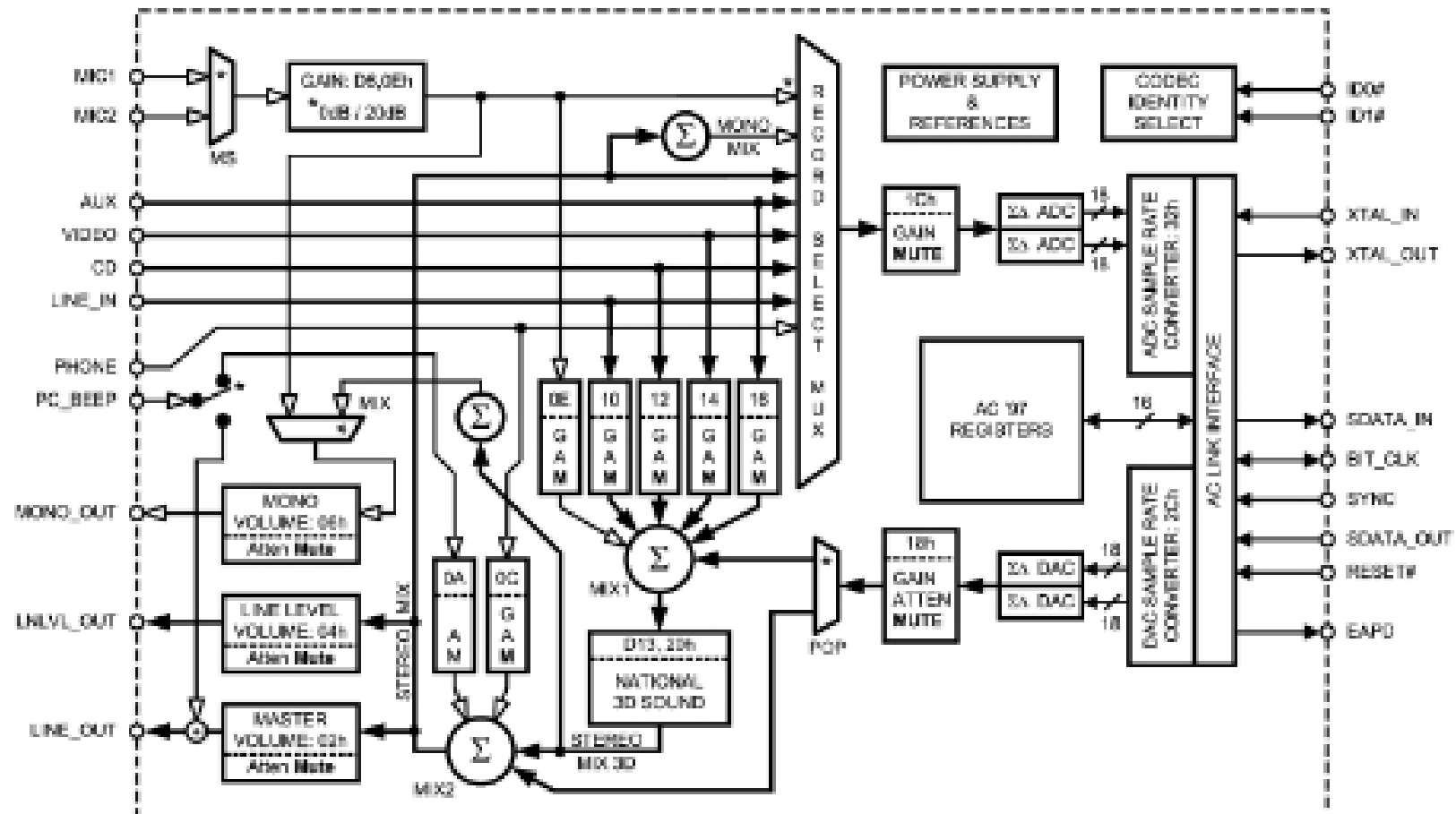


Figure 2: LM4549A Block Diagram

2.2.2 Register Map (Page 15, 22-24 of the LM4549A Datasheet)

Most of programming the **LM4549A** consists of writing the control registers to change volume settings and PCM audio rates. For this you will send **register write commands** using **Slot1** and **2** in the output data stream. **Slot1** will contain the **address** of the register to write and **Slot2** will contain the **data** to write.