University of California at Berkeley
College of Engineering
Department of Electrical Engineering and Computer Sciences

EECS150
Fall 2001

Richard A. Pon

Checkpoint 3

Overview

The purpose of checkpoint 3 is to create a logic block that will translate the digital audio output of a CD-ROM drive into another format that a dual 16-bit Audio DAC can recognize. The signal from the DAC is then amplified with an amplifier and is then outputted to a headphone jack. You can then attach a pair of speakers to the headphone jack and listen to cd music. **Note**: Do not attach headphones directly to the board. The volume is **extremely loud!**

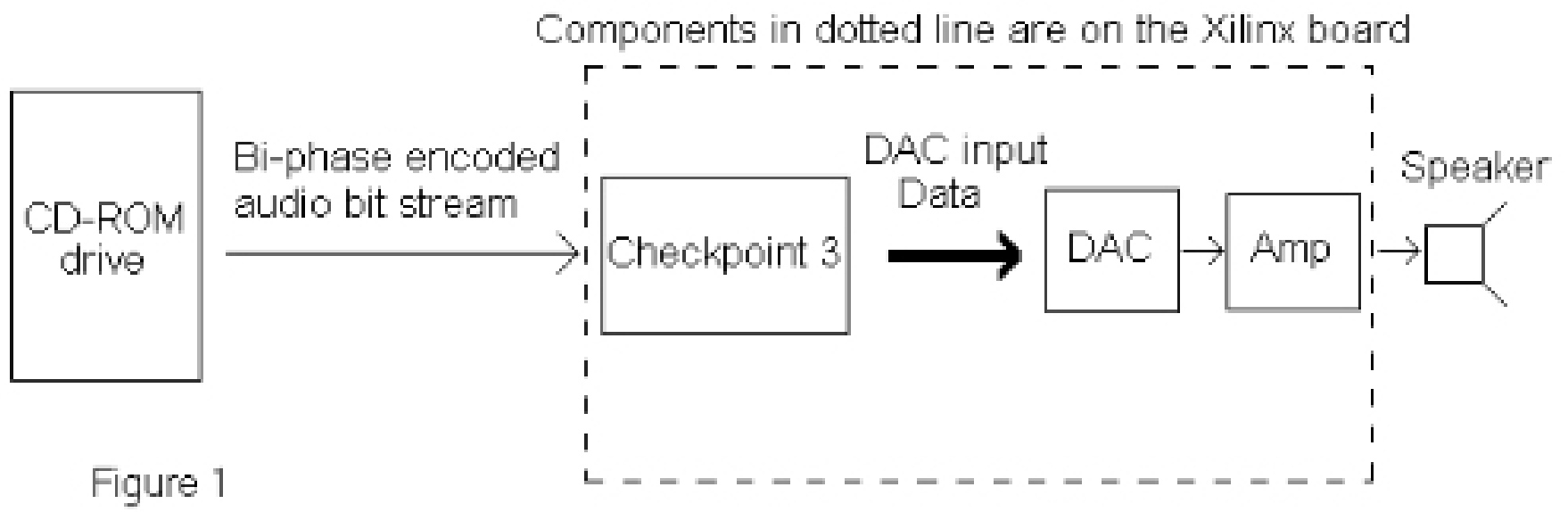

Figure 1

Figure 1 shows how checkpoint 3 is physically related to the CD-ROM drive and the rest of the Xilinx board. Additional components must be wire-wrapped to your board, a DAC, amplifier, and discrete packs containing resistors and capacitors.

Inputs and outputs

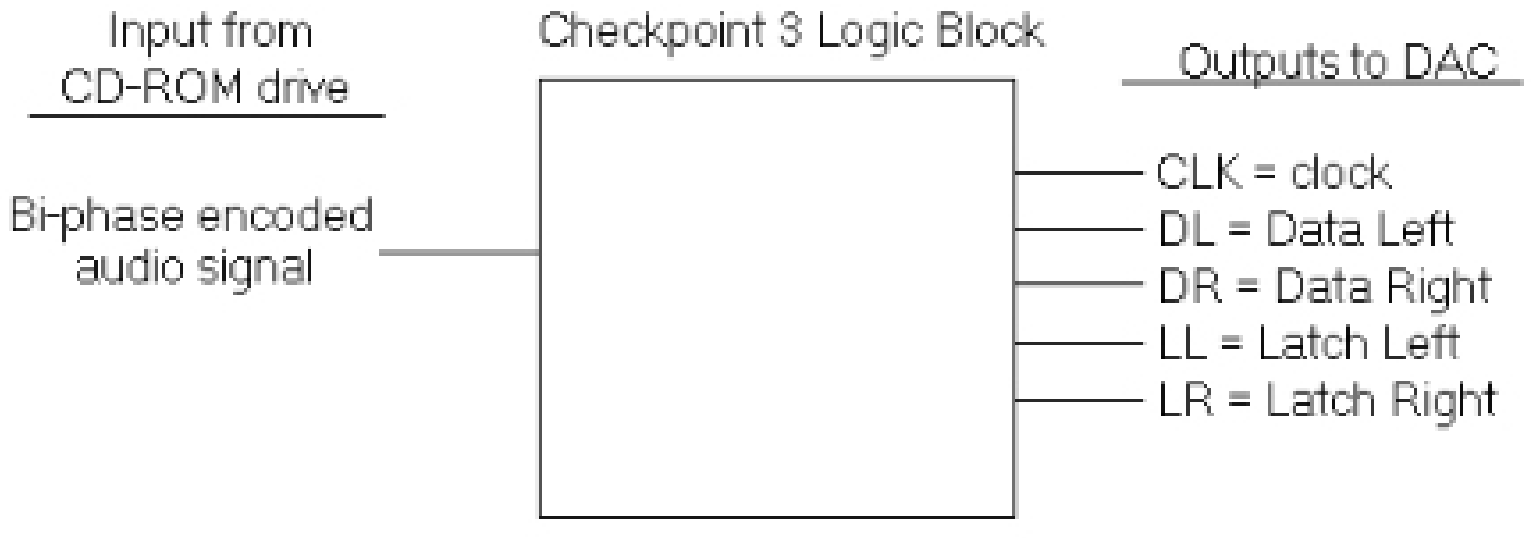

Figure 2 - Graphical representation of inputs and outputs

Figure 2 shows the inputs and outputs of the checkpoint2 logic block. There is one input, the digital audio output from the CD-ROM drive. There are 5 output signals, clock(CLK), Data Left(DL), Data Right(DR), Latch Left(LL), and Latch Right(LR). **Note**: We will also use a 16MHz clock not shown in the figure.

Inputs

Bi-phase encoding

Bi-phase encoding is a method for encoding data. The basic building is a cell. Each data bit is composed of two cells. A 1 data bit is made up of two different cells, either a 1 and a 0, or a 0 and a 1. A 0 data bit is composed of two of the same cells, either two 1's or two 0's.

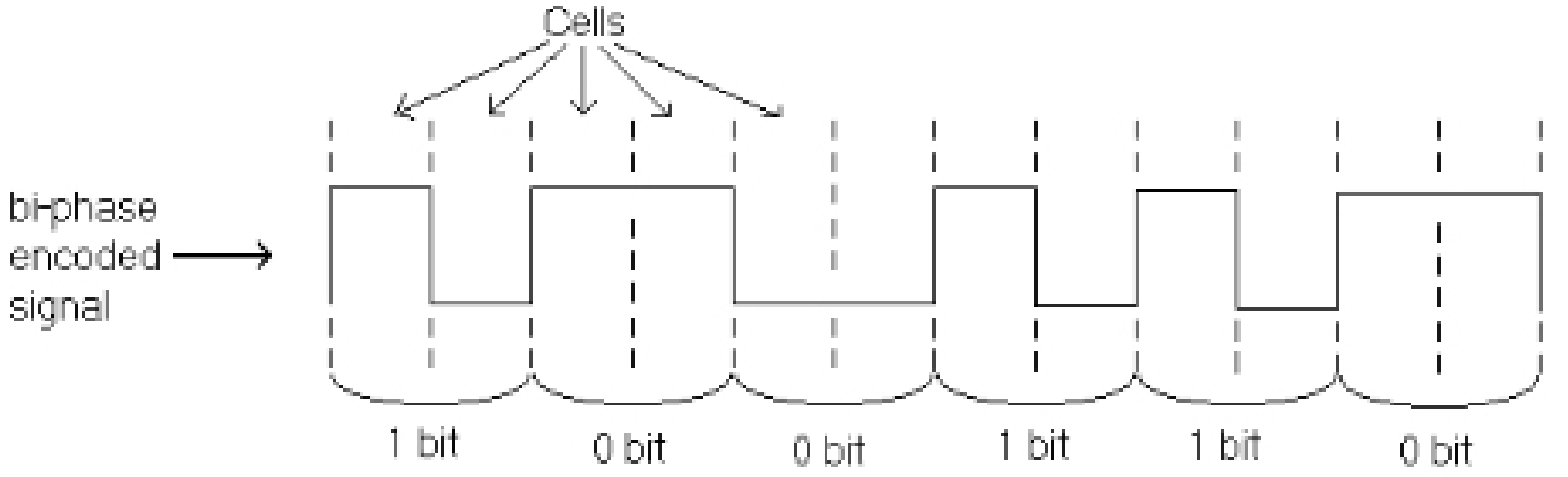

Figure 3

Figure 3 shows a section of a bi-phase encoded signal. This section is composed of 12 cells in the sequence "101100101011" which represents 6 data bits, "100110".

Note: No more than two consecutive cells can have the same polarity, (except preambles: please continue reading)

Frames and Preambles

How do we extract the audio data from the bi-phase encoded bit stream? The bi-phase encoded stream is composed of 32-bit frames. Each frame contains one 16-bit audio sample. Each frame begins with a preamble. There are three types of preambles.

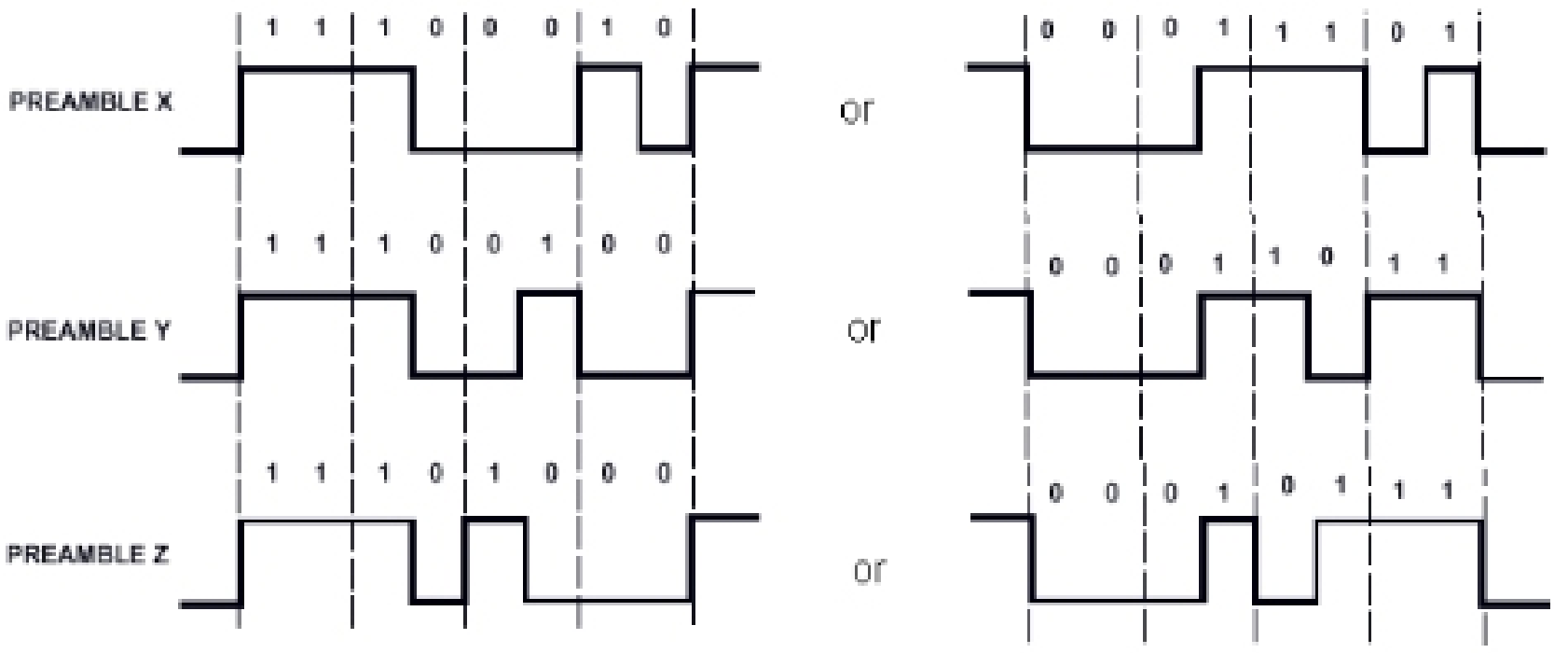

Figure 4

Figure 4 shows the three different Preambles. Please note that:

- Each Preamble is made up of 8 cells. (not data bits!)
- Notice that each Preamble has two representations. They are inverses of each other. This means that the Preamble X can be either "11100010" or "00011101". (← These are **cells**)
- Preambles are distinguished from regular data as they contain 3 consecutive cells of the same polarity. Preambles begin with a "111" or "000" sequence of cells.
- Preambles begin with "111" if the previous cell was a "0" and begin with "000" is the previous cell was a "1".

What do these different preambles mean?

Remember that each frame starts with a preamble and contains one 16-bit audio sample. The preamble tells you which channel the sample is for, either left or right. (We have stereo audio so we have both left and right channels)

	BIPHASE PATTERNS	CHANNEL
X	11100010 OR 00011101	LEFT
Y	11100100 OR 00011011	RIGHT
Z	11101000 OR 00010111	LEFT

Figure 5

From figure 5, we see that preamble X and Z are used to denote the left channel, and preamble Y denotes the right channel.

Note: You should still distinguish between preamble X and Preamble Z. We may use preamble X for future uses. For now, preamble X and Z will do the same thing.