

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

EECS 150  
Spring 2001

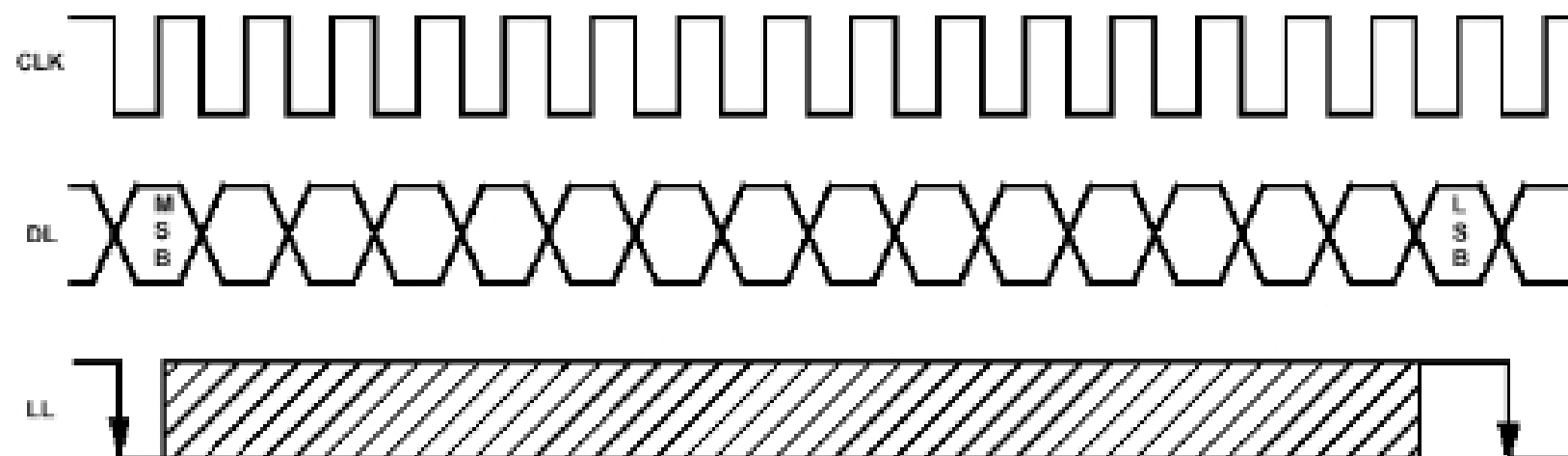
R. H. Katz  
P. Yan

**Project Checkpoint 2**  
**Audio Interface**

### DAC Overview

In order for the DreamKatz videogame console to generate sound, we need to translate the digital sound data stored in the EPROM (erasable programmable ROM) into an analog signal so that it can be output to a speaker. To accomplish this, we will use a DAC (digital to analog converter), which can be thought of as the DreamKatz's sound chip.

The AD1866 we are using is a 2-channel 16-bit DAC (we are only using one channel, though). Internally, the DAC has a 16-bit shift register, which shifts in a new bit from the DL line on each positive edge of the clock. On a negative edge of the LL line, the DAC takes the current 16-bit value in the shift register as the new value to output. The DAC only changes its output value at each negative edge of LL.



Notice that the negative edge of LL comes after the last bit has been shifted in but before the next positive edge of the clock. This is a VERY IMPORTANT detail that you should be aware of when you design the DAC module.

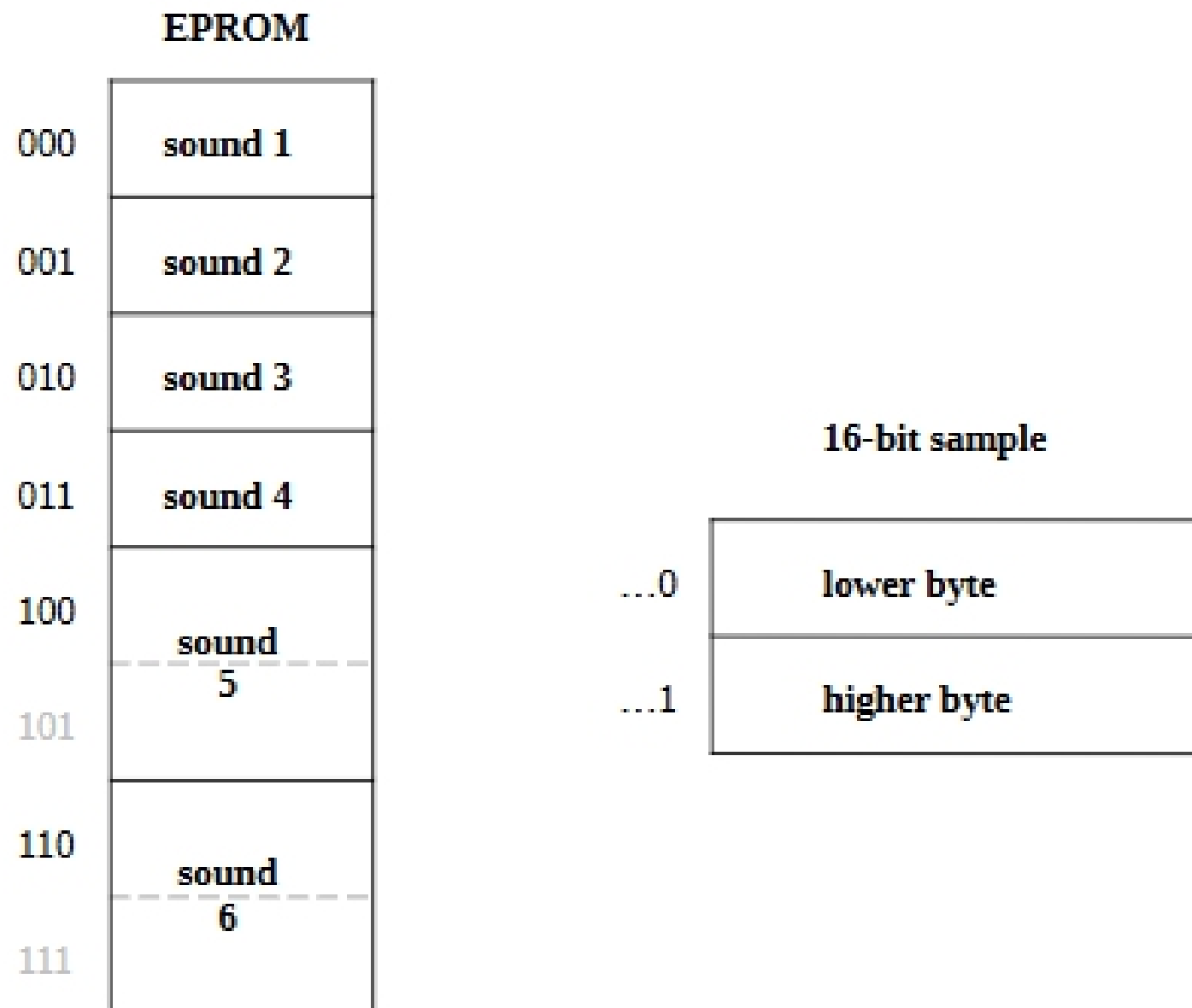
### EPROM Overview

We will use the EPROM to store sound data for sound effects. The EPROM uses 19-bit addresses, which reference 8-bit words. The only interfaces with the EPROM are the address lines (19 bits) and the data lines (8 bits). To read a word, all that is necessary is to drive the address lines with an address, and the word at that address will be output to the data lines (after a small delay, of course).

### EPROM Data Format

The sound samples stored in the EPROM are 16-bit (two's complement) sampled at 16kHz. Since the EPROM is 512K, this gives us a total of 16 seconds of sound. To organize the sounds, the EPROM is partitioned into 8 segments each holding 2 seconds of sound; the upper (most significant) three bits are used to refer to each partition. These 8 partitions are used to store six sound effects. The first four

partitions each hold one sound. The fifth and sixth partitions hold a double-long (4 seconds) sound, as do the seventh and eighth partitions.



Since each sound sample is 16-bit, we need two words of the EPROM to hold each sample. The samples in the EPROM are all aligned in little endian format. Each sample resides in two consecutive addresses, where the lower byte is contained at an address with LSB = 0, and the higher byte is contained at an address with LSB = 1. Thus, you can refer to a sample by the upper 18 bits of the address and just toggle the LSB to get the lower and higher bytes.

## Prelab Preparations

This checkpoint requires these hardware components:

- M27C4001 EPROM
- AD1866 DAC
- LM4862 amplifier
- the two discrete packages

There is A LOT of wire wrapping to do. All of the components listed above need to be wire wrapped. It will probably be helpful to place wrap-IDs on the pin array for each chip, which can be found at <U:\cs150\projectdocs\wrapid.doc>. Also, it is a good idea to place components close to other components that they will be connected to (i.e. place the EPROM close to the Xilinx 4010XL chip).

## EPROM Pin Connections

**connection**

**connection**

VCC	Vpp	1	32	VCC	VCC
P58	A16	2	31	A18	P60
P57	A15	3	30	A17	P59
P65	A12	4	29	A14	P67
P10	A7	5	28	A13	P66
P9	A6	6	27	A8	P14
P8	A5	7	26	A9	P18
P7	A4	8	25	A11	P62
P6	A3	9	24	$\bar{G}$	GND
P5	A2	10	23	A10	P61
P4	A1	11	22	$\bar{E}$	GND
P3	A0	12	21	Q7	P84
P77	Q0	13	20	Q6	P83
P78	Q1	14	19	Q5	P82
P79	Q2	15	18	Q4	P81
GND	VSS	16	17	Q3	P80

**DAC and LM4862 Amplifier Pin Connections**

