Fortune Teller

Design Document ver. 1.0

Leo Gertsenshteyn (lpg2006@columbia.edu)
Russ Santillanes (res2049@columbia.edu)
Kathryn Hagan (kmh2124@columbia.edu)

March 26, 2006

I. OVERVIEW

The fortune teller will be composed of three main parts: the graphical display, the textual fortune generation and display, and the audio output. There will also be a main control program, a small C program that will run on the Microblaze CPU. It will wait for a keystroke from the user, generate a fortune when input is received (passing it to the video module to be displayed), and resume waiting for the next user input.

Figure 1 illustrates the overall system architecture.

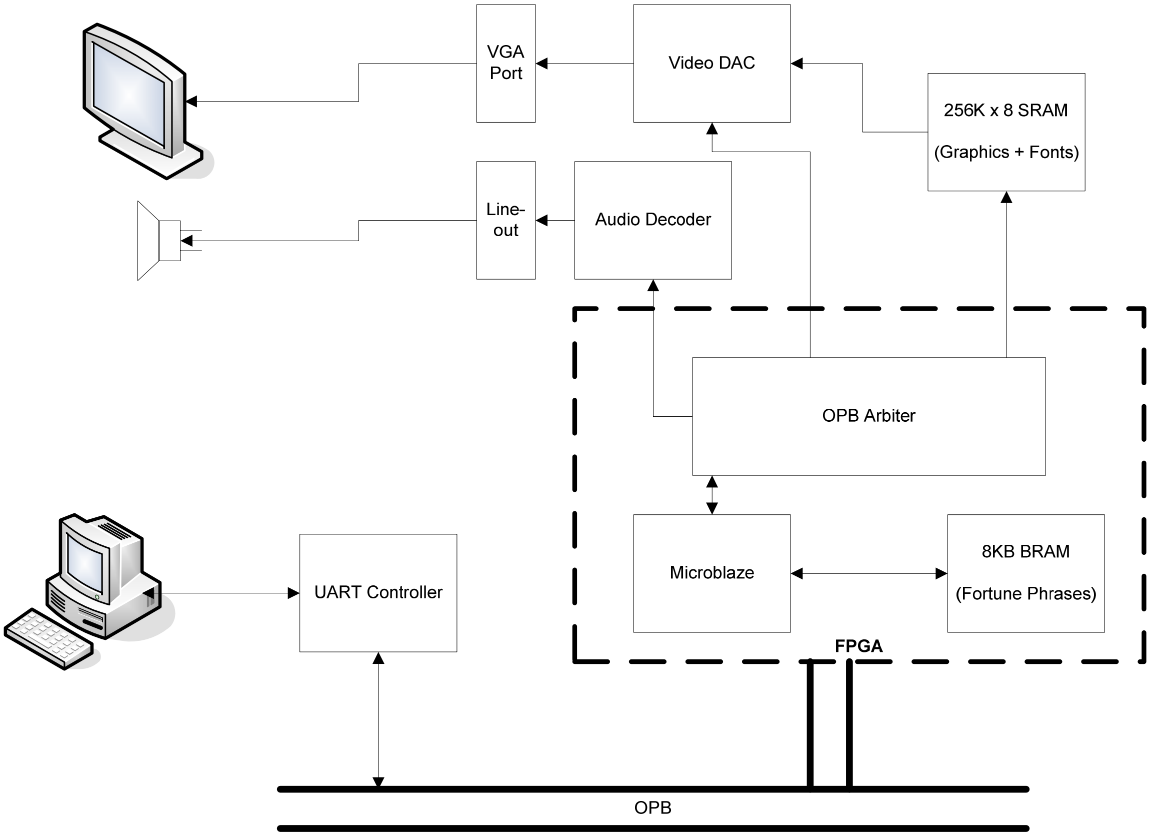


Figure 1. System architecture.

II. DESIGN PROCESS

The following is a list of project milestones, in the order they will be completed.

1. base graphic display (8-bit RGB color, no animation)
2. fortune generation and text display
3. main control program
4. animation
5. audio output

Milestones 1-3 will be complete on April 13 (the 75% mark); the entire project will be complete on May 9.

III. GRAPHICAL DISPLAY

The video module will provide an interface for the main program to write text to the screen, as well as displaying and animating the gypsy graphic. (The graphic will be based on the one on the title page of this document, a public-domain piece of internet clip-art from the following webpage: http://serp.la.asu.edu/clipart_dir/clipartidx.html.)

This module will be a custom video controller, based on the one discussed in class. Graphics will be buffered in the SRAM (8-bit/pixel depth RGB @ 640 x 480); the SRAM will also store fonts and a few bitmaps for animation.

As seen in the figure below, the display will be split between a large area for the main graphic and a text area that will display the fortune.

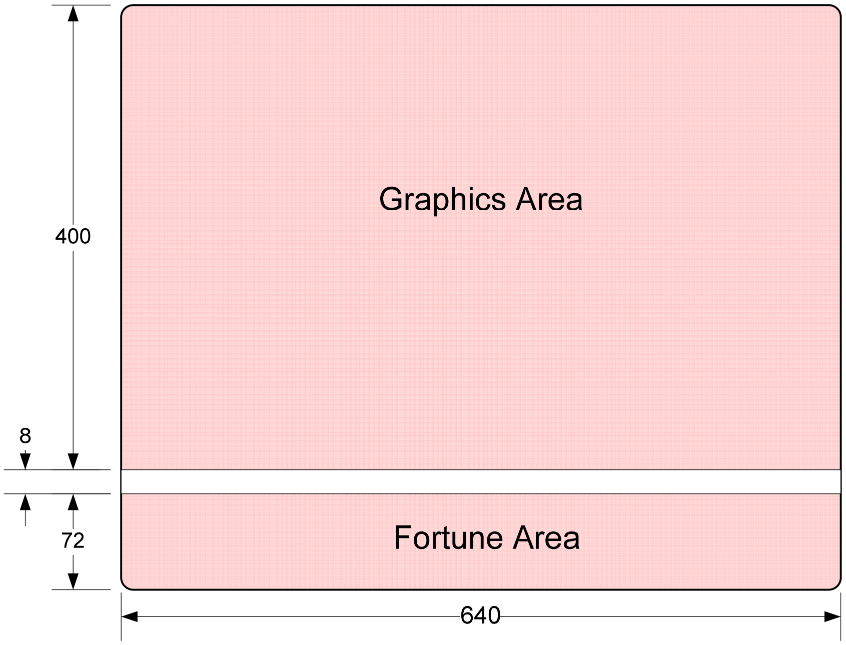


Figure 2. Division of screen real estate.

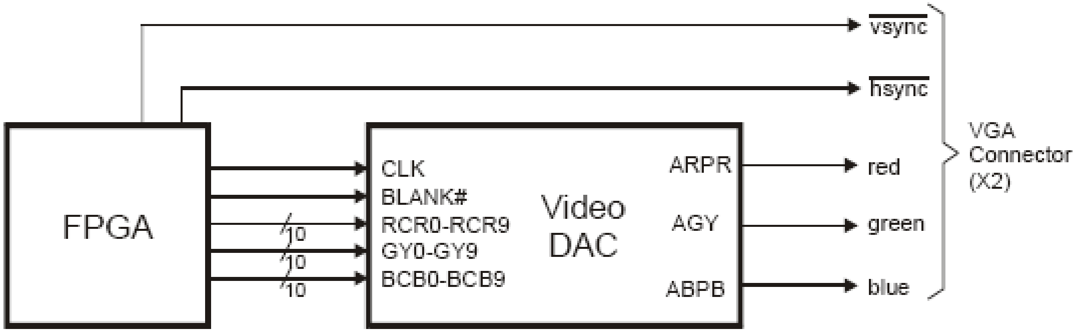
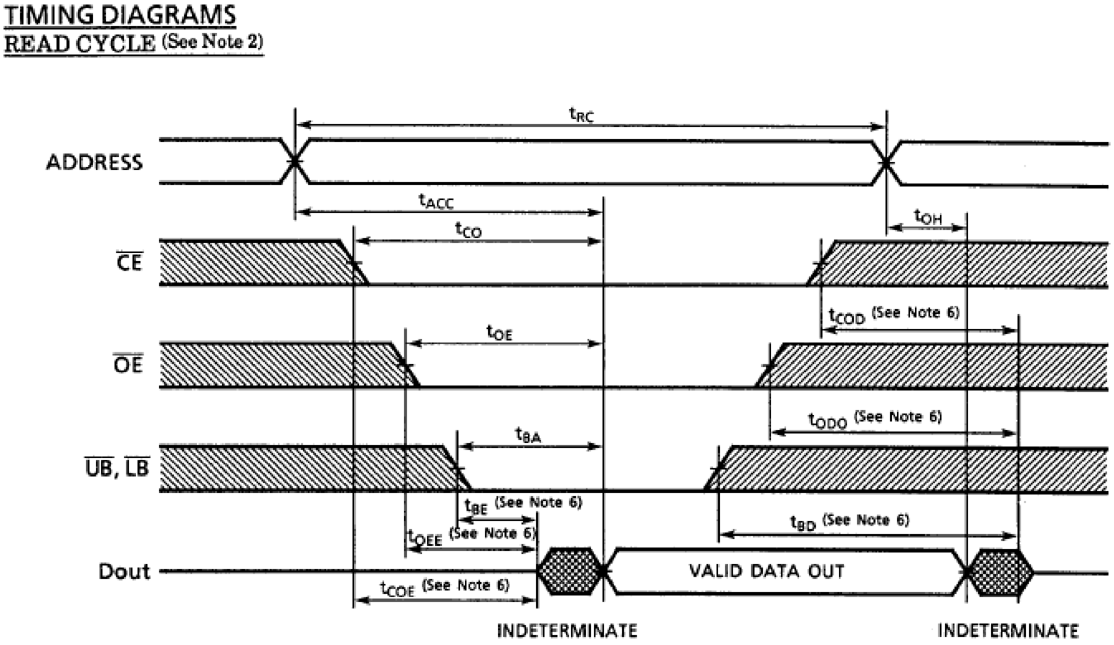


Figure 3. Block diagram for the video controller.

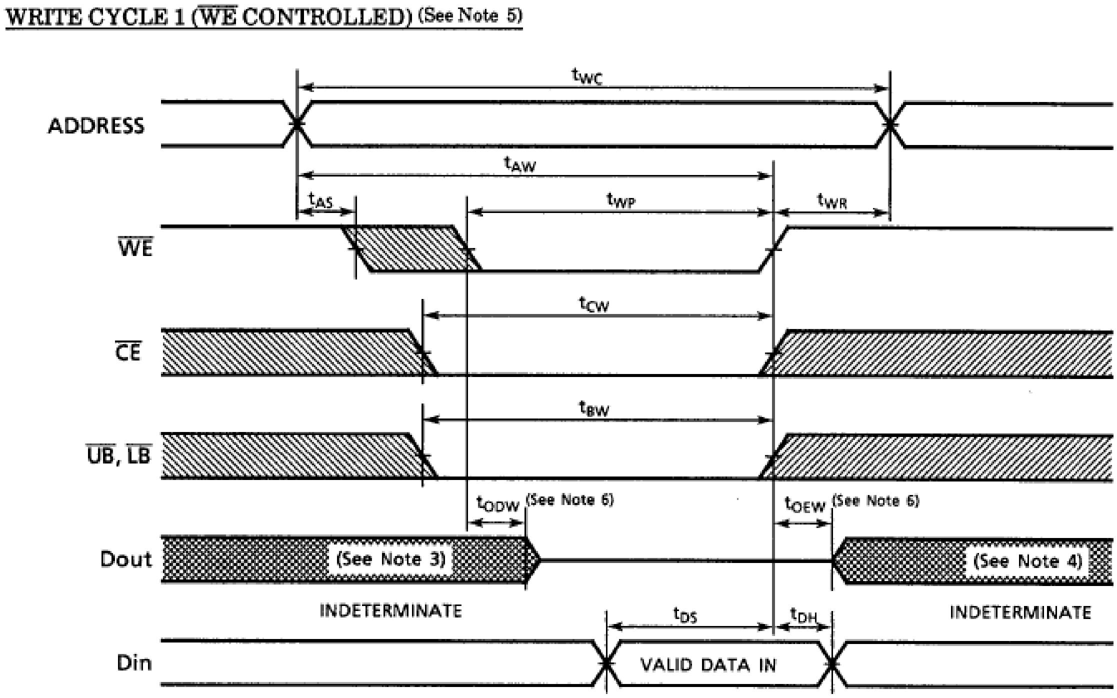
SRAM timing diagrams (from the data sheets):



TIMING DIAGRAMS

READ CYCLE (See Note 2)

Figure 4.



WRITE CYCLE 1 ($\overline{WE}$ CONTROLLED) (See Note 5)

Figure 5.

IV. FORTUNE GENERATION AND DISPLAY

Fortunes will be randomly constructed from lists of sentence fragments stored in the BRAM. This will be done by the main control program,.

Text writing will utilize a peripheral similar to that which was used in lab 2 (the simple terminal emulator); this peripheral will take a C string as input and write it to the text area of the screen.

V. SOUND

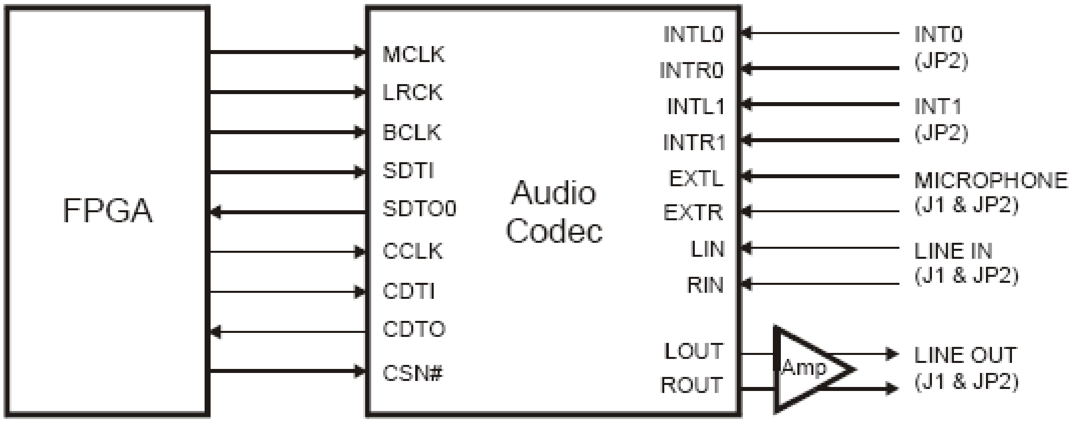


Figure 6. Block diagram of the audio codec.

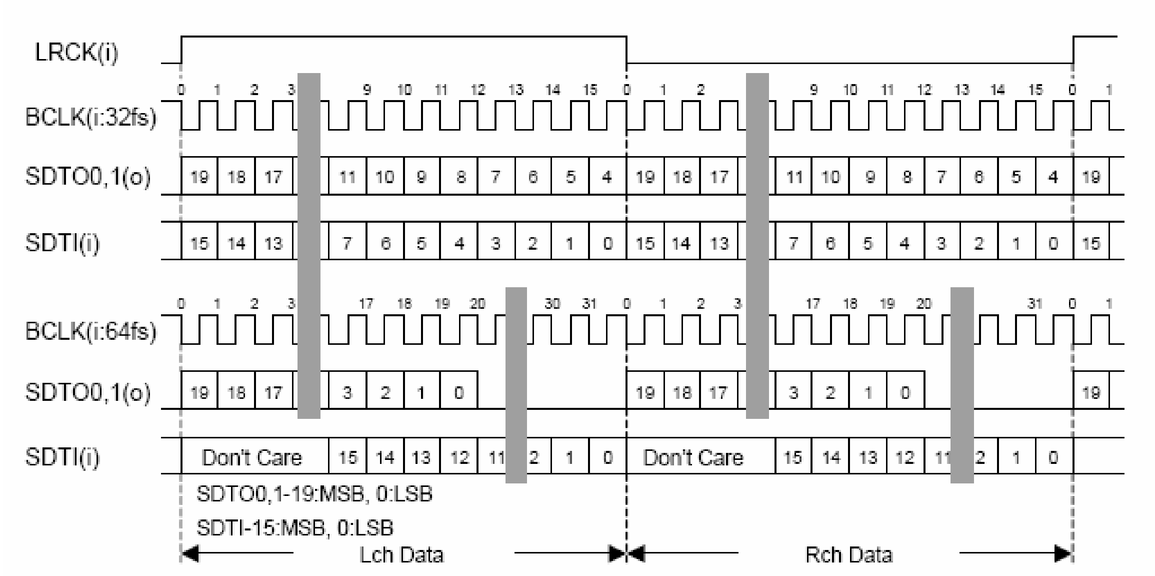


Figure 9. Audio Data Timing (No.0)

Figure 7. Audio timing diagram.