Turbo-2D GrafixxZapper 9000

Bret Victor

bret@cco.caltech.edu

Size:	1750λ by 1780λ
Tools used in design:	`magic, makeframe`
Designed:	CS/EE 181 fall 1997
Status:	It's wired up, padded, and works fine under `cosmos`.

Functional Description

This chip is a processor/controller/driver for a 32 by 8 LED matrix display, and supports four-bit color depth (sixteen shades of intensity). The driver section takes care of multiplexing the columns of the display, leaving the lights on for appropriate amounts of time to generate the color depth. The processor section allows the user to manipulate the display. Instead of accepting specific commands, the processor lets the user define a "filter" function which describes what to do with the screen data. According to this filter, the processor will take the value of the pixel at the cursor location, modify it somehow (replace with another value, blend with another value, or add or subtract another value, clipping to 15 or 0 respectively), store this modified pixel either back in the cursor location or above, below, or to the left or right of it, and then move the cursor itself in any one of those directions. This can be automatically repeated up to 256 times. Using this filter, all kinds of various effects can be performed: drawing dots, horizontal or vertical lines, scrolling the display in any direction, filling, lightening or darkening regions, and more. It also makes it easy to upload files to the display, blend a file with the existing display, or implement a "scrolling message" display.

Implementation Strategy

At any given time, only one column of the display is active, and each light in that column is either on or off. The fraction of time that each light in the column is on is its pixel value plus one divided by sixteen. Thus, pixels with value zero are turned on one-sixteenth of the column's time, and pixels with value fifteen are left on for the column's entire time slice. By rapidly multiplexing through the columns, this gives the illusion of the higher-valued pixels being brighter and the lower-valued pixels being darker. The design takes 4096 clock cycles to go through all 32 columns. With a 1 MHz clock speed, this would give a screen refresh rate of 244 Hz, which is hopefully fast enough to avoid noticable flicker.

The processor takes user input from a four-bit parallel bus. An "input" register, the filter register, the repeat register, and the cursor register can be written to via this bus. The user sets up the registers needed to implement the desired filter, and then gives the "go" command. A busy flag will go high, and will go low again when the command has completed.

Architectural Details

The chip requires an external 256-word SRAM and a decoder for the LED display columns. The display driver works by reading in a column of data (interrupting the processor if necessary) and then continually comparing the data to a counter to generate the row outputs. The computation is done serially, but the outputs are buffered so they change in parallel and there are no flickers. As for the processor, each "filter" takes two clock cycles, a data read and a data write cycle. The processor is basically a "cursor" section and a "data" section which handle address and data lines respectively. Both consistly mostly of an adder and a couple registers and muxs.

There is no PLA in this design, nor a ROM. The control section consists of a three bit decoder for the external input bus, three state bits to control the rest of the design, and some NAND gates for next-state logic. The driver and processor work mostly asynchronously with each other, with the driver halting the processor when it needs to use the SRAM.

Project Status

It's done, and it works just dandy under `cosmos`, after fudging with the tristate pins a little bit because `cosmos` doesn't like them.