Experiment 5 – Digital Oscilloscope

Course title: Digital Logic Design Lab

Date: 96/2/27 and 96/3/3

Group 4

Hadi Safari
810194531
hadi.safari@ut.ac.ir

Seyed ahmad A. PouriHosseini
810194358
seyedahmadpourihosseini@yahoo.com

Abstract— **The goal of this experiment was to learn how to build a digital oscilloscope. Several RTL components were used to realize the oscilloscope circuit, two of which were actually components that we had designed in the previous labs (The VGA controller and the ADC). These components can be grouped into 4 categories: horizontal control, trigger control, vertical control, and display. A RAM is also used to allow the display part to effectively used the data generated by the other parts.**

Keywords— **digital oscilloscope, RTL design, analog to digital convertor, sampling, trigger, AC or DC switch.**

I. Horizontal Control

This part manages the tasks related to sampling the input signal, and its time related settings. It is made up of 2 modules:

** The ADC module was not implemented and instead the digital function generator from the previous lab was used, as instructed by the lab instructors, to directly input digital data to our circuit.

A. Time per div:

The role of the time per dive module is to adjust how wide or narrow the displayed signal is, i.e. it adjusts how long the period of time represented by each square on the screen is. This logic is implemented by using a clock divider and then using the divided clock instead of the normal clock in almost all other components. In this way, when the time per div module divides the clock down, the operation of the components that write into the memory is slowed down, resulting in a wider display of the signal.

B. Sampling:

This module periodically stores the data which is visible on the input of the oscilloscope inside itself. This data is then directly used to fill the RAM. This module is implemented simply by using a register that uses the divided clock as its clock input, and outputs the data directly into the RAM.

II. Trigger Control

This part manages everything related to triggering the signal, and is made up of 3 components:

A. Trigger :

This module receives the trigger level, the trigger slope, and the input signal to the scope as inputs, and issues a trig_enable signal whenever the input signal reaches the trigger level at the desired slope. A register was used inside this module to store the previous sample of the input signal in order to be able to check the slope condition.

B. Trigger mode:

This module receives the trig_enable signal from previous part, and the trigger mode as inputs and decides, according to the trigger mode, whether a trigger has actually occurred. If it is so, it issues a reset signal to the address generator module. A trigger occurs when either the trig_enable signal is issued, or if we are in auto mode and the internal counter has reached its maximum value.

C. Address Generator:

The job of this module is to provide the write address for RAM. It uses the reset signal from the previous part to decide whether or not it should start re-writing the whole RAM. It starts the re-write only if it has reached the last RAM slot, and the reset signal is issued. It is realized using a simple counter, with some extra logic on its reset input.

III. Vertical Control

This part reads its data from RAM, applies the AC/DC and volt per div settings to it and the outputs the final data to the display part. It consists of 2 components:

A. AC or DC:

This module applies the AC or DC setting on the signal. If DC mode is selected, it leaves its input data unchanged. But, if AC mode is selected it subtracts the average amount of the signal during a whole sweep from the input data before delivering it to the output. The average amount is calculated using a sum register which holds the sum of the signal values seen up to that moment, and updates itself with every clock pulse, and an avg register which holds the calculated average. Every time that a sweep is completed (which takes 640 clock pulses) the value of the sum register is divided by 640, and then assigned to the avg register as the new avg amount.

B. *Volts per div:*

Similar to the time per div part that adjusted how wide the signal is horizontally, this part adjusts how high the signal is vertically. It receives the output of the AC or DC module, and the volts per div amount, and attenuates its value as needed (The attenuation was done by shifting the input value to the right.)

IV. DISPLAY

This part receives the value that should be displayed from the vertical control part, and formats it for proper display on a VGA port. It consists of 2 modules:

A. *Comparator:*

This module receives the y coordinate of the point being displayed on the VGA controller, and the voltage amount that we want to be displayed right now, and simply checks whether the two values are the same. If so, it generated an eq signal which is fed to the VGA controller. It also checks whether the given coordinates (x and y) belong to a point on the grid lines (x == 320 or y == 240) and issues the eq signal if it is the case.

B. *VGA controller:*

This part was implemented in the previous lab. Its job is to specify the x and y coordinates of the point it is currently at, then receiving the RGB values for that point from the input, and then displaying that specific color at that given point.

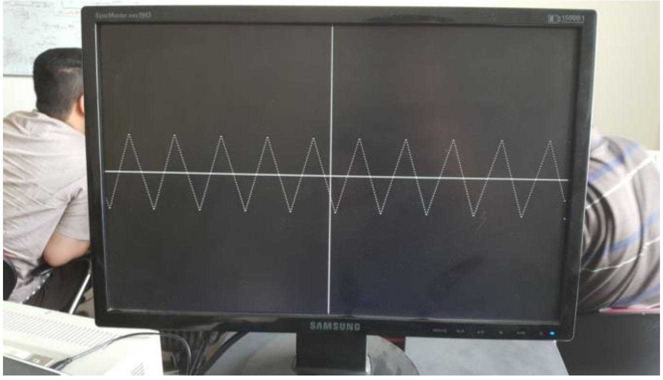

Figure 1. Output of the circuit for a triangular input wave

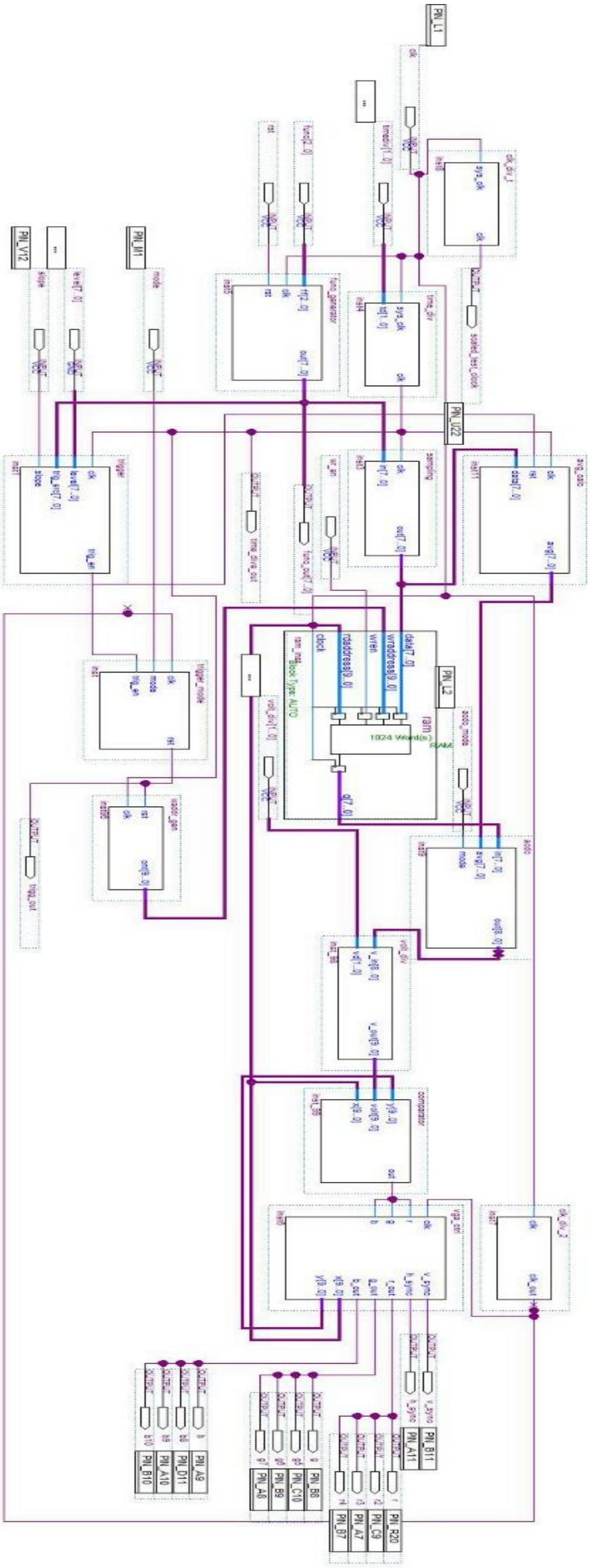

Figure 2. Schematic view of the whole circuit