

Lab 2: Etch a Sketch

Computer Architecture, Fall 2022

Lilo Heinrich and Emma Mack

- Pulse generator
- PWM Module
- Triangle Generator
- One of:
 - SPI Controller for Display
 - I2C Controller for Touchscreen
- Learning from Professional Code: In your own words, describe the FSMs in:
 - ili9341_display_controller.sv
 - ft6206_controller.sv
- Design and implement a main FSM that interfaces with a video RAM.

Pulse Generator, PWM Module, Triangle Generator

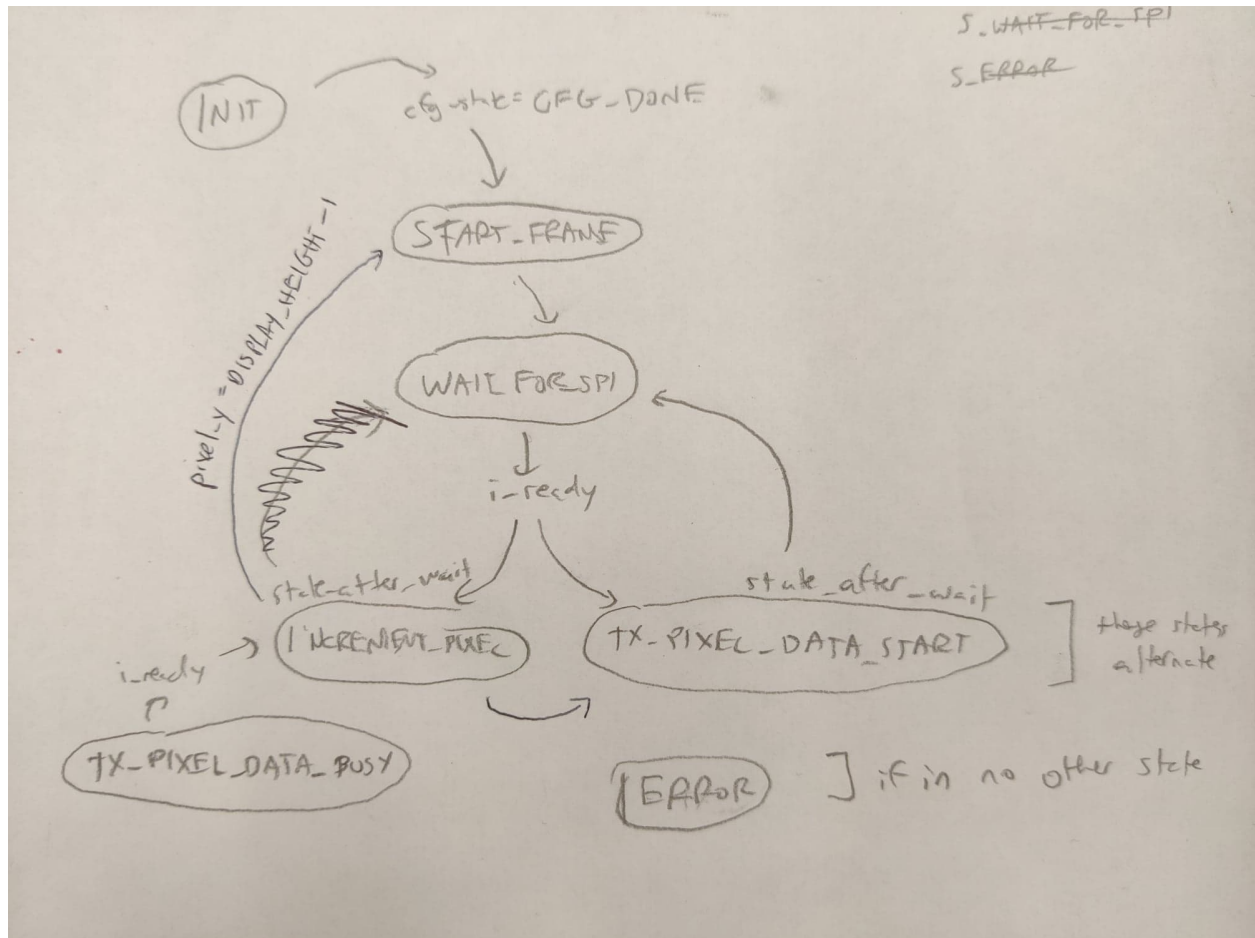
Saved in /hdl as pulse_generator_ours.sv, pwm_ours.sv, and triangle_generator_ours.sv. These were the generators that we made for homework 5. When we were debugging in later stages, we switched over to the solutions to eliminate possibilities for error, and continued using the solutions through our finished product.

SPI Controller

Saved in /hdl as spi_controller_our_attempt.sv. It worked in simulation, printing out the values we expected. We were not able to load it onto the fpga because of the ili9341_init.memh error. By the time the ili9341_init.memh error was fixed, we started using the solution SPI controller.

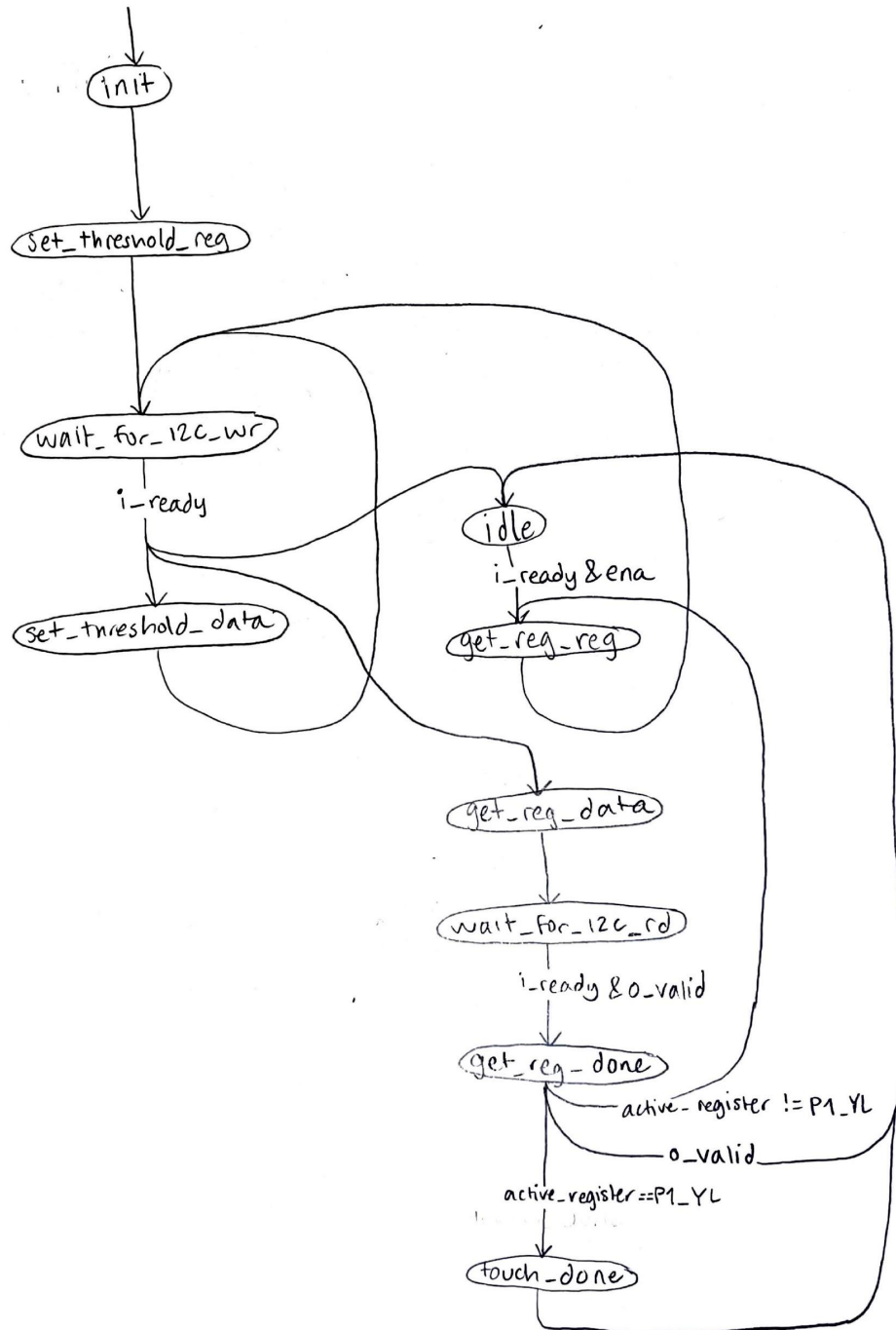
Display Controller FSM

Upon initialization, there's a sub-FSM that controls configuration. Once that is complete, we enter start frame, where things that need to be initialized every frame are initialized. Then, we alternate between waiting for spi to send, tx_pixel_data_start, and incrementing which pixel we are "drawing" until the last pixel has been drawn, at which point we return to the start frame.



Capacitive Touch Controller FSM

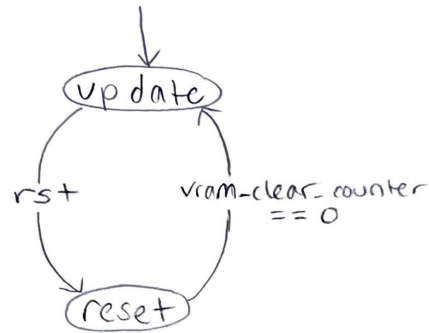
There are two main steps in the capacitive touch sensor controller FSM in `ft6206_controller.sv`. First, it starts by setting the touch detection threshold (see the states named “`set_threshold`”). Secondly, after it goes into the idle mode for the first time it starts looping through a cycle of reading the register for touch data (see “`get_reg`” states). If a touch is received (state “`touch_done`” is reached) or if the output is valid (“`o_valid`”), then the loop starts over again.



Main FSM

Our main FSM is fairly simple with only two states: reset and update. The state starts at update. If the reset button is pressed during the update state, it would enter the reset state and clear the vram. Otherwise, it writes the received touch data into the corresponding pixel in vram, thereby

allowing the display controller to read it out from vram to show the previously touched pixels on screen in a different color from the background.



To show we successfully got the etch a sketch to run, here is a masterful finger drawing we made!

