SPI Controller Notes:

ili9341_display_controller FSM

- Goal is to read vram data on to display using SPI communication between vram and display controller
- Two sets of states:
    - CFG (config) states, governs how the controller acts when state INIT is active
    - Main FSM states, governs how controller acts after initializing
- Keeps track of both current state, and what state the FSM should be in after waiting for current state to complete.
    - Keeps track of where it's at in current command by a bit counter that counts down each posedge of clk until it hits the end of the transmitted data
- Uses variable current command to store what the controller should be doing
- Block from 144 to 176 draws all pixels from vram according to color and address unless enable test pattern is active
- After colors are sent to controller, main fsm begins
    - Checks if rst is active and sets state to INIT again to essentially wipe all vram
    - If rst is not active, and ena is, checks if FSM is in INIT state and runs through CFG states to check what initial conditions should be
    - If not INIT, runs through normal FSM states to update vram and states accordingly

ft6206_controller FSM

- Loading touch data from the i2c controller into touch0 is the goal
- We start with S_INIT and transition to S_SET_THRESHOLDs to send the touch threshold of the display so that only touches that are above the configured thresh are valid
- Afterwards we are in the S_IDLE state where we wait for the i2c to say it is ready to send data back and we then wait for the write phase to pass and finally read the output data from the i2c
    - This is the S_GET_REG_DONE state where we cycle through the important registers and store data about the touch locations and validity
- Next we switch to the S_TOUCH_DONE state where we load our touch buffers into the touch0 and touch1 output wires
- Finally we jump back to S_IDLE and repeat

LAB 2 Block Diagram

Components:

touch controller

I²C device that takes capacitive signals in fingers and converts to digital representation in x and y coord.

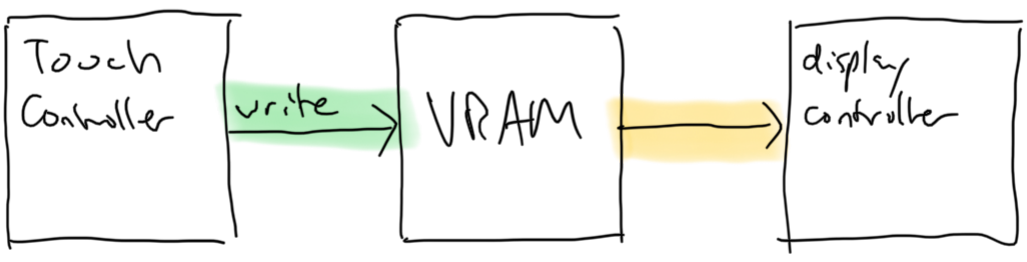
display controller

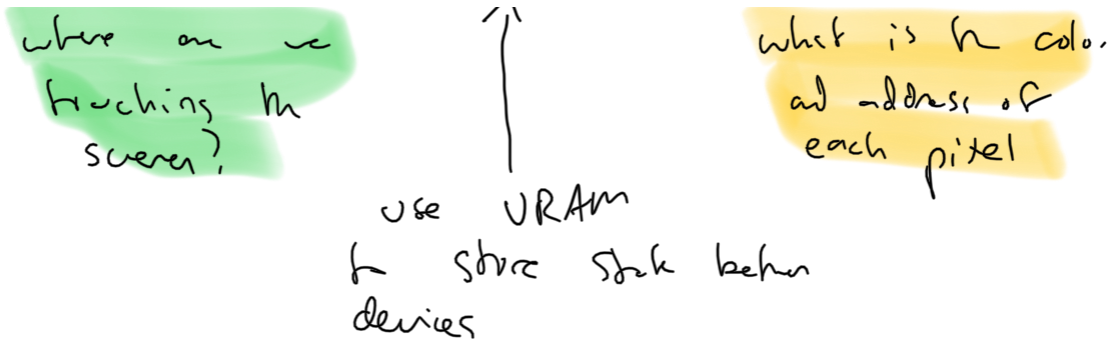
SPI device that reads a pixel color and current read address in display or pixel color to the screen. Iterates through each pixel to update it.

VRAM

Video RAM, or block of nxm registers that hold the pixel color at the address. The address of the VRAM is used as the address on the display controller.

Diagram




where on we touching the screen?
what is the color address of each pixel
use VRAM to store state behind devices