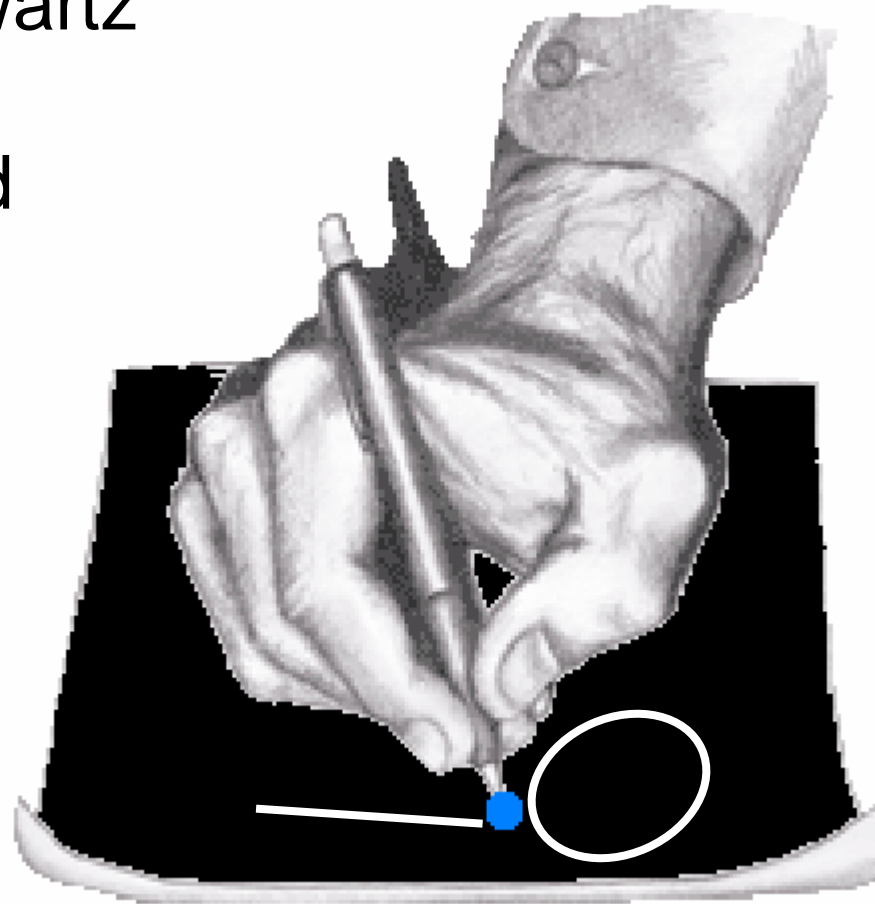
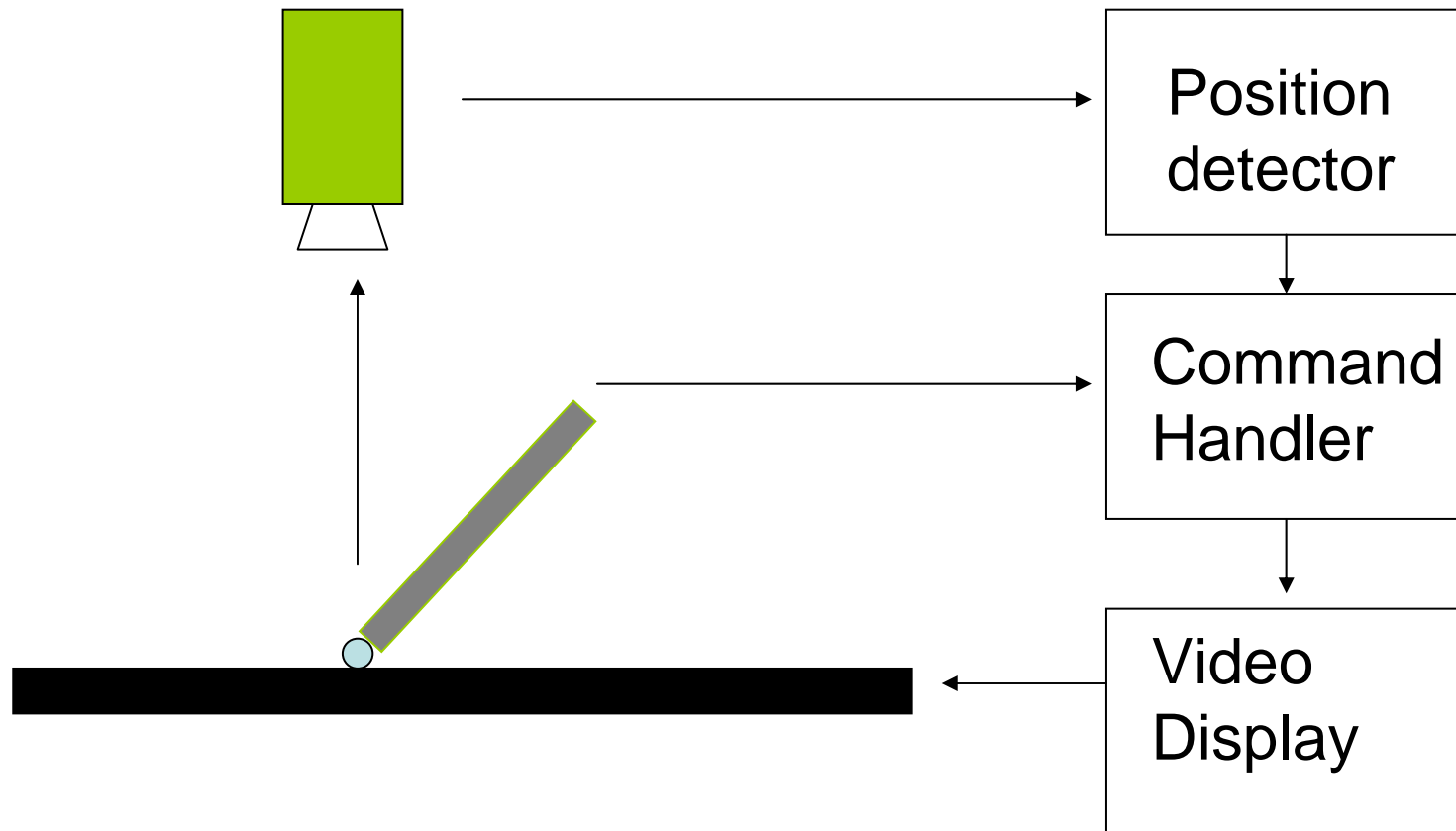


# User-friendly stylus and video surface CAD system

Jeremy Schwartz  
Paul Peeling  
Faraz Ahmad



# Overview



# Position detector

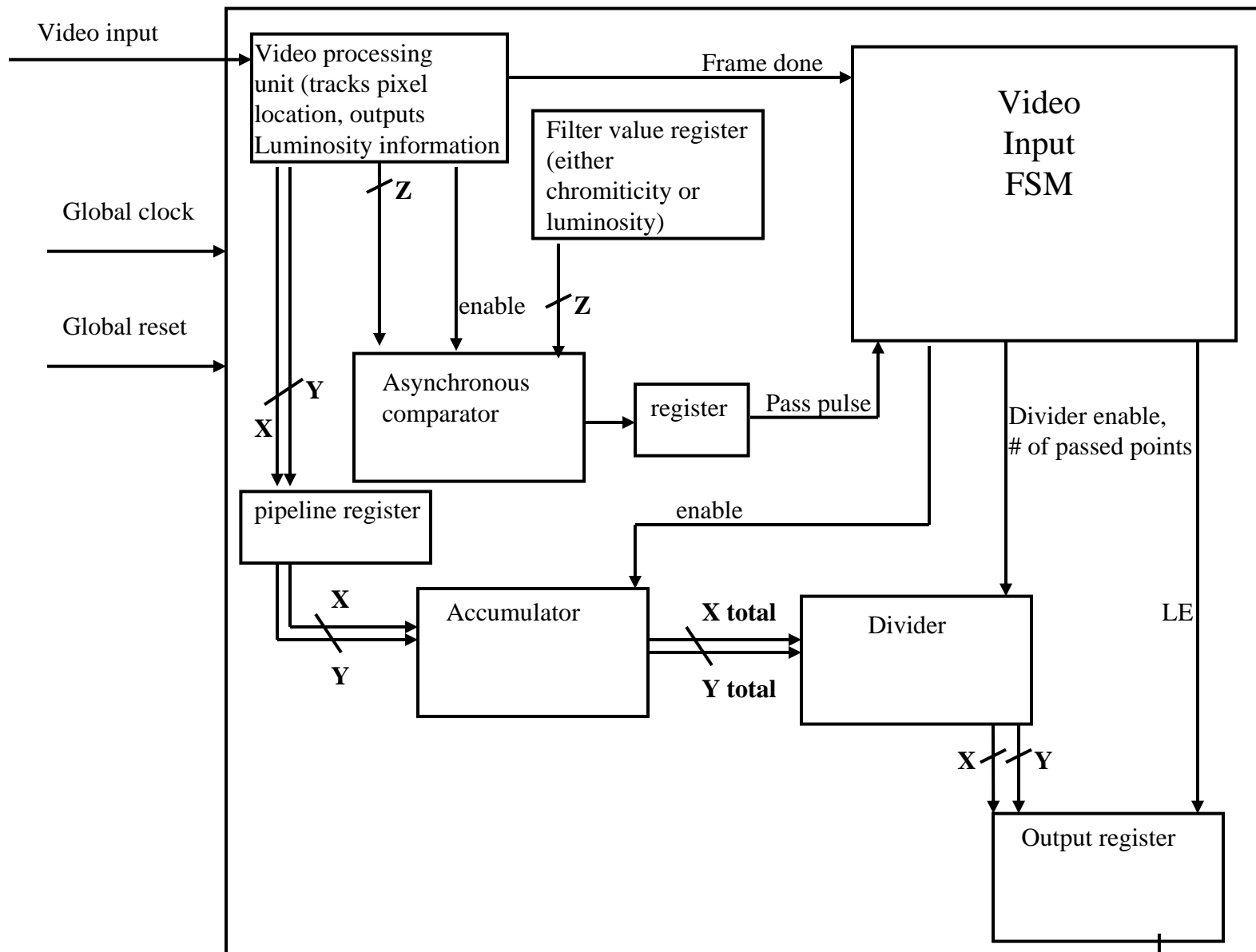
## Tasks

- Take a camera image as input
- Filter camera image for green blob
  - This step to be done on the fly!
- At end of each frame, calculate centroid of green blob
- Output stylus position (x, y)

# Position detector

## Block Diagram

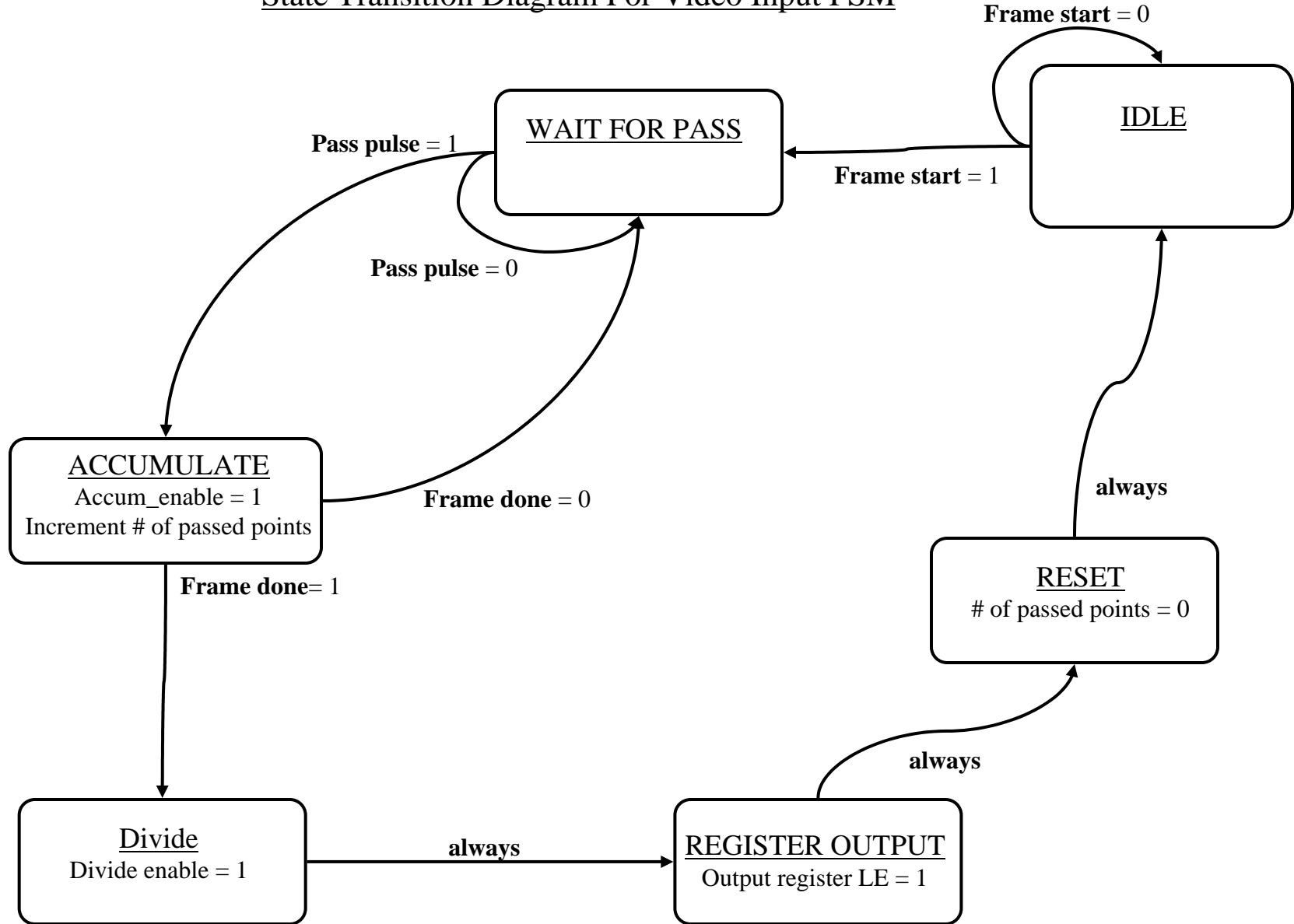
- Only record the points that register as 'green'
  - Throw everything else away
- Do this on the fly...we must pipeline!



$Z$  represents width of information about one pixel  
 $Y$  represents height of screen (in pixels)  
 $X$  represents width of screen (in pixels)

Stylus position

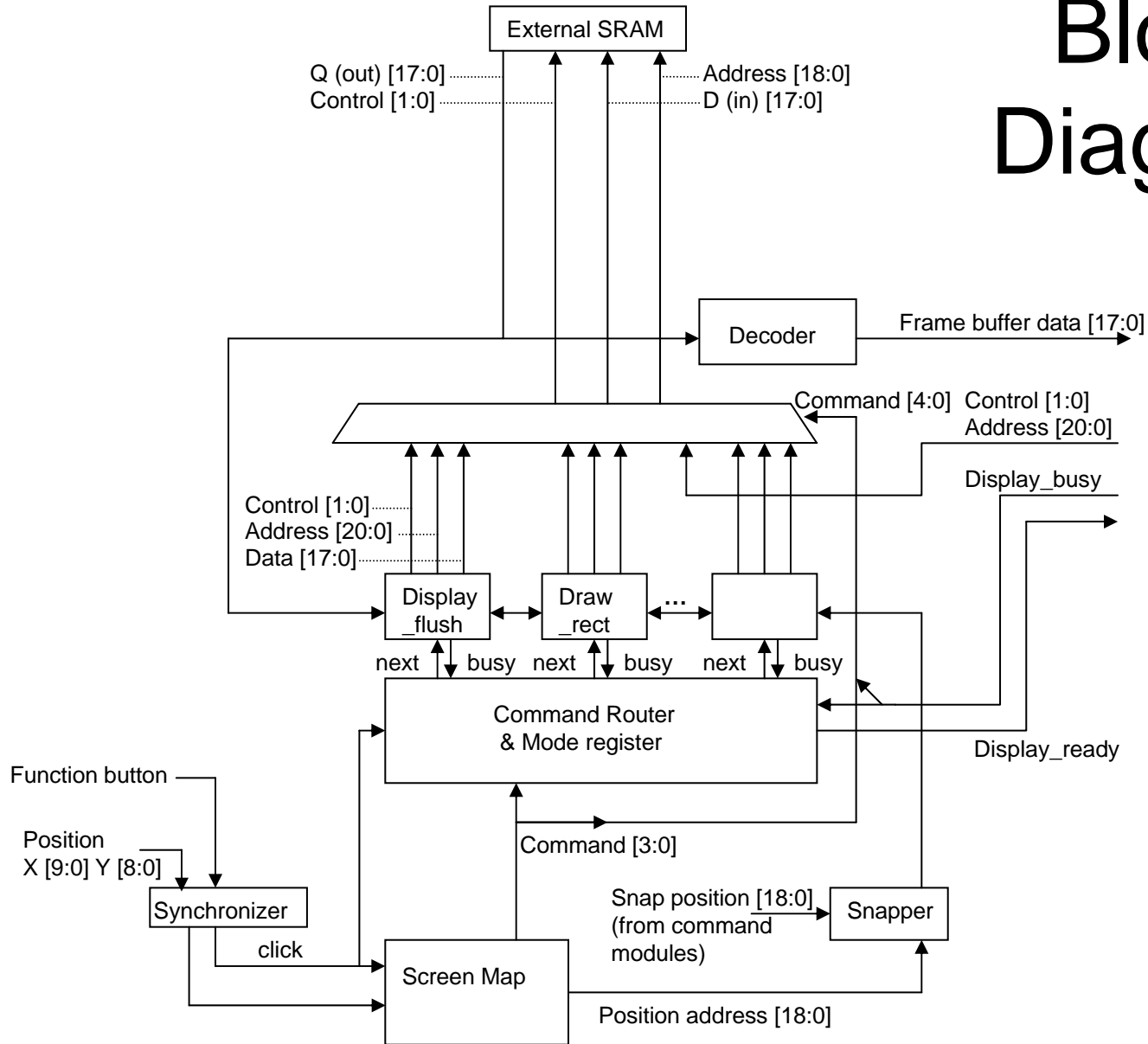
# State Transition Diagram For Video Input FSM



# Command Handler

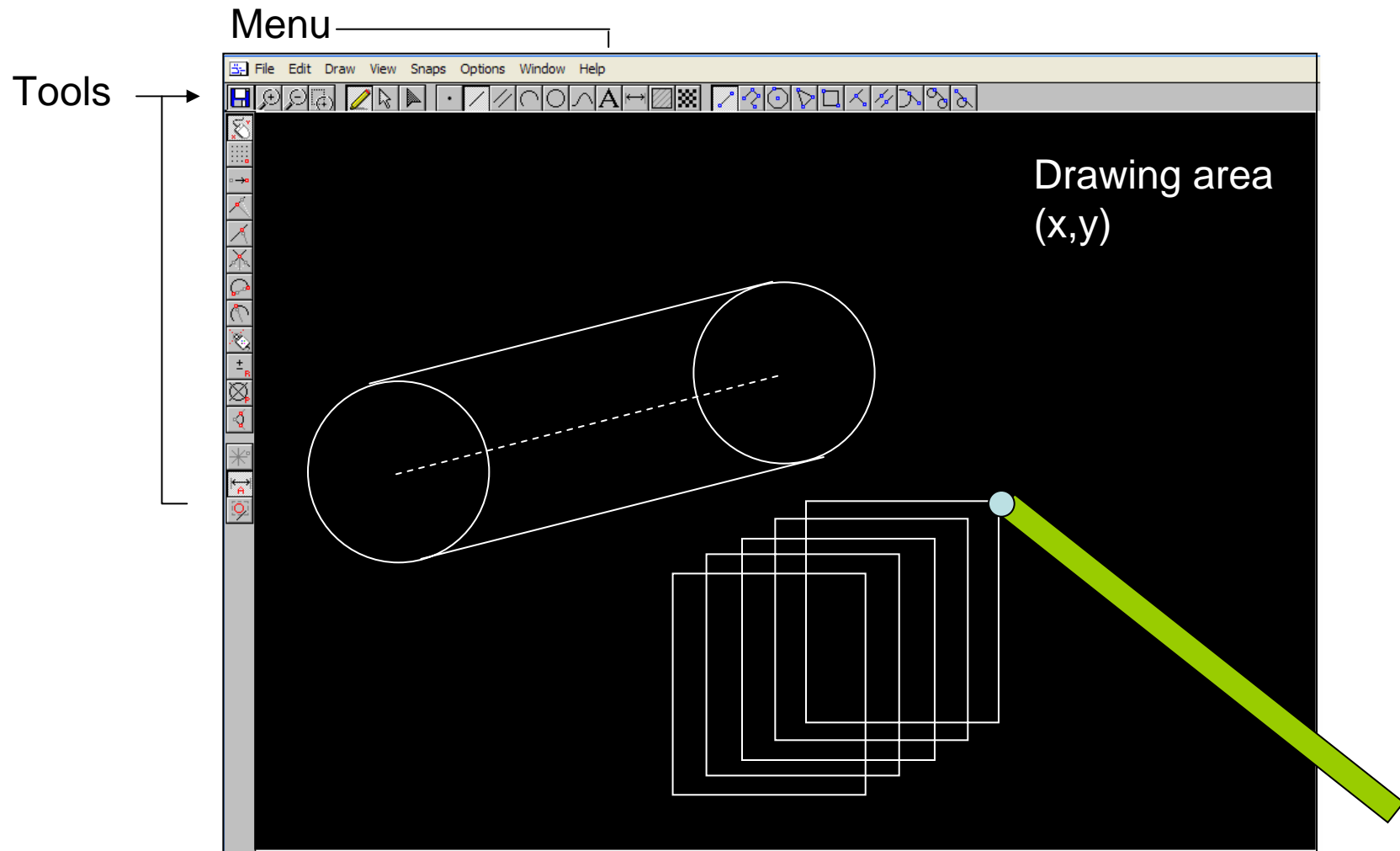
- Screen position and button click input
- Maps the screen position to either a drawing area position, or a menu / toolbar command
- Major/Minor FSM structure controls flow of drawing algorithms
- Output to a frame buffer containing information about the objects
- Frame buffer decoded into RGB

# Block Diagram





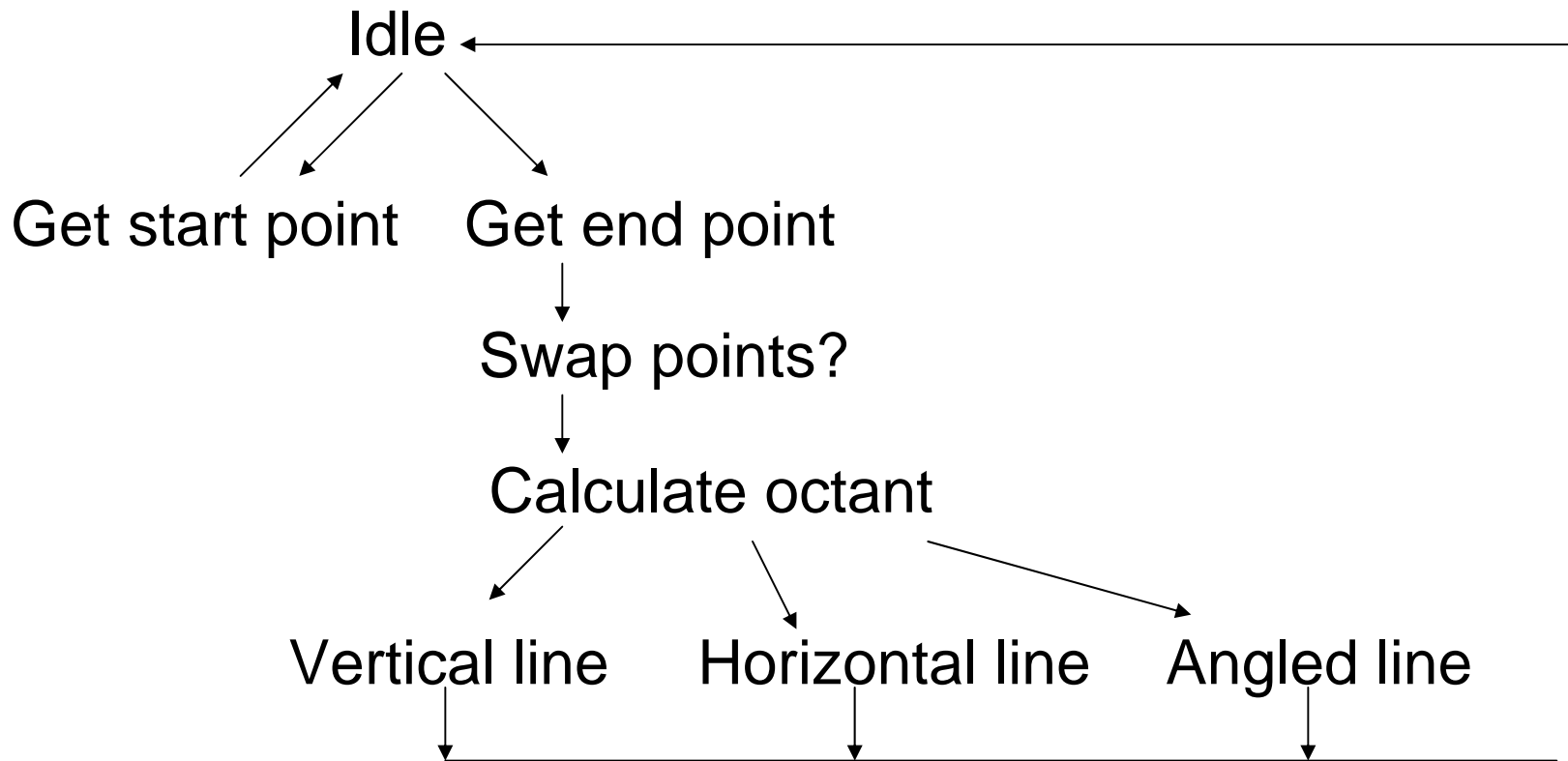
# Screen Mapping



# Command Router

- Stores the current state of a drawing operation (e.g. `got_centre_of_circle`) in the mode register.
- Uses 'next' signals to tell the minor FSMs to complete the next operation given the new position information from the input
- 'Next' and 'busy' signals are muxed to result in simple command-independent structure

# Example: Line-drawing FSM



Bresenham's Algorithm

# Storage of objects

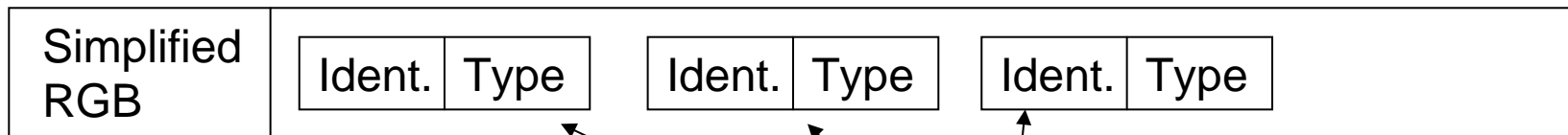
ZBT SRAM storage: 19 address lines, 36 data lines

Address: 

X coord [9:0]	Y coord [8:0]
---------------	---------------

640x480 resolution frame buffer

Data: object identifiers and information

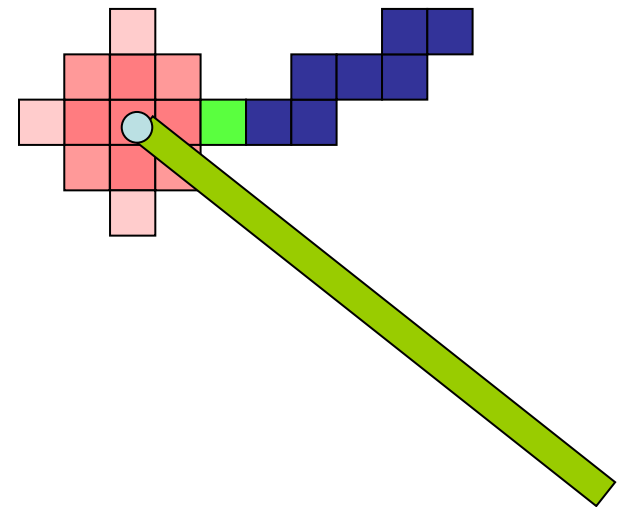
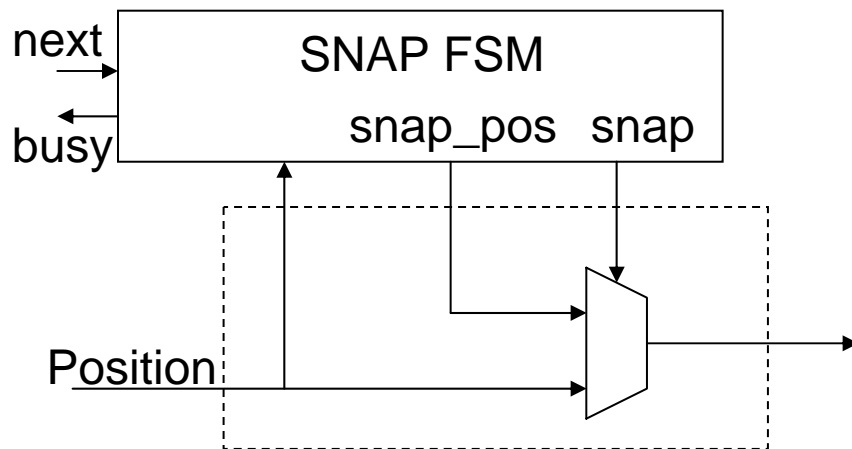


↓  
To video display  
through decoder

Also stored in a lookup table  
for easy access to related objects,  
e.g. the other end of a line

# Snapping

- Snapping to grid, other objects etc.

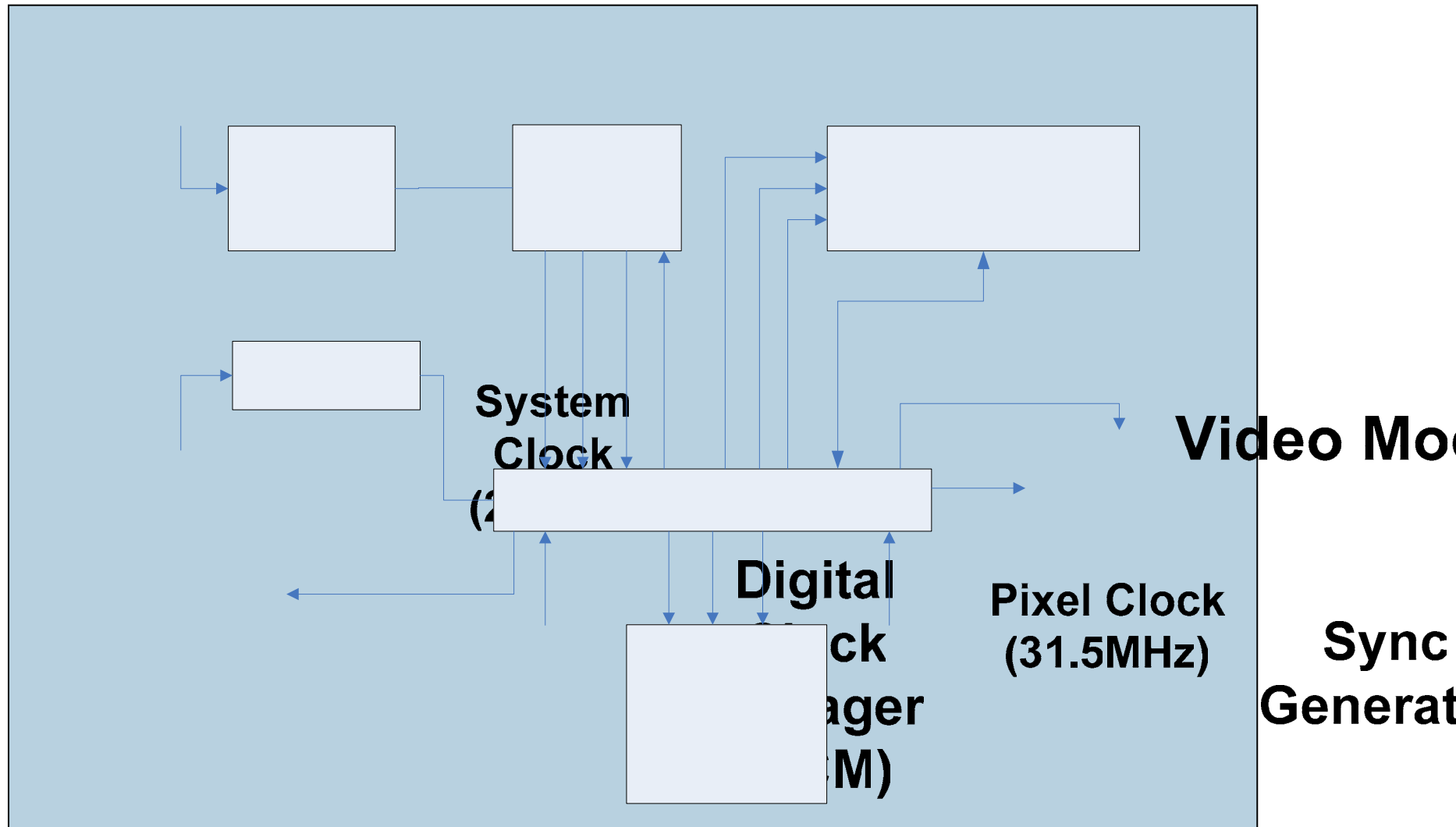


# Video Display

The objectives for the Video Display are:

- Display Command Toolbar
- Display Drawing Image
- Be able to superimpose this onto a background

# Implementation of Video Display



# Milestones

April 26	Point drawing (to SRAM) Camera input
April 29	Lines, circles, polygons B&W display
May 2	Snapping Command & Video integration
May 6	Pointer input Color display
May 9	Deletion & Movement System integration