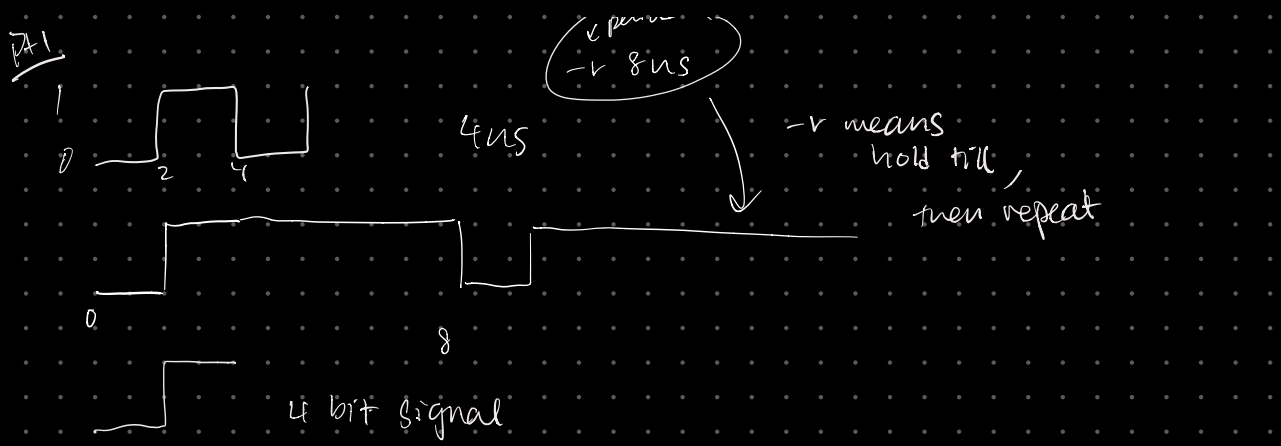
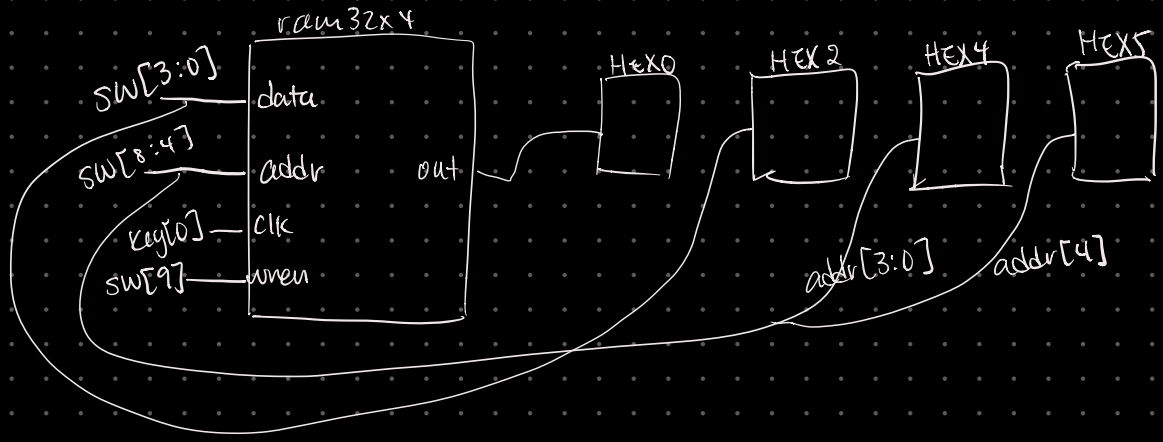
Pt 1



ii) Schematic of circuit

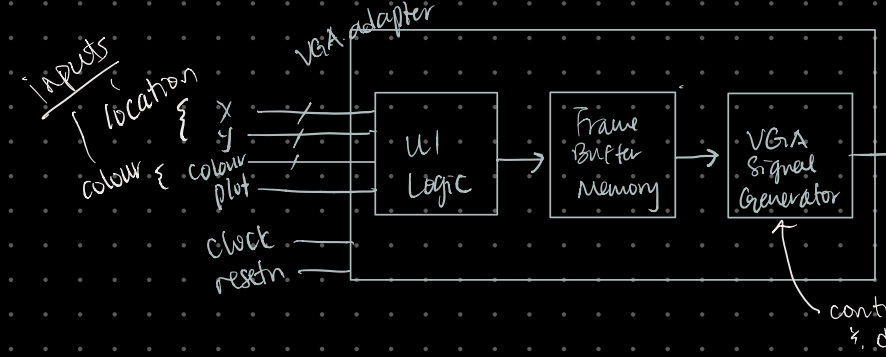


Pt 2

Display simple images on VGA display

→ Design circuit to draw a filled square on the screen @ any location in any colour

VGA adapter module → accepts a set of (x,y) coords (pixel) and a colour



RGB → (1,0,1) = Red

(0,1,1) = additive G,B colour

KEY[0]: active low reset

SW[9:7]: specify the colour

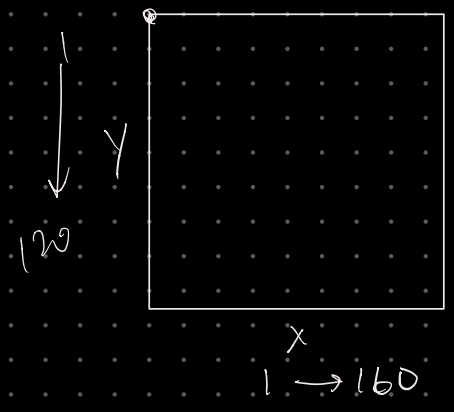
SW[6:0]: use to input (X,Y)

load: set SW[6:0] to val then press KEY[3] to hold

X = 8 bits → register 8-bit to hold X val (register w/ X val)

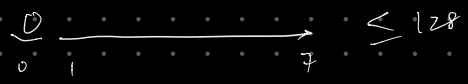
Y = 7 bits → access only 1st 128 cols

else goes to Y

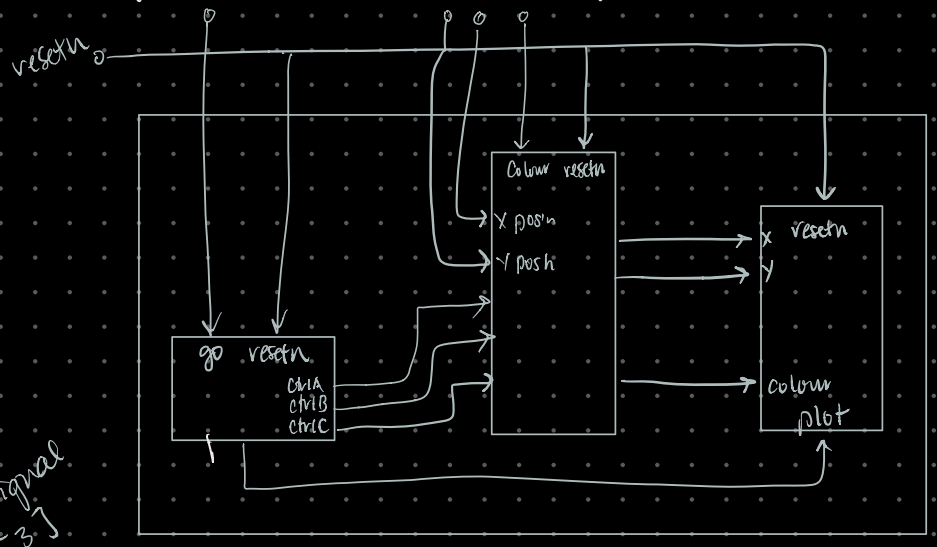


$2^7 = 128 > 120$

stores X → only need 7 bits for inputs (from SW[6:0] signal)



Draw filled square when KEY[1] pressed (go)



enable signal KEY[3] for the register that holds coord X.

State machine

Datapath

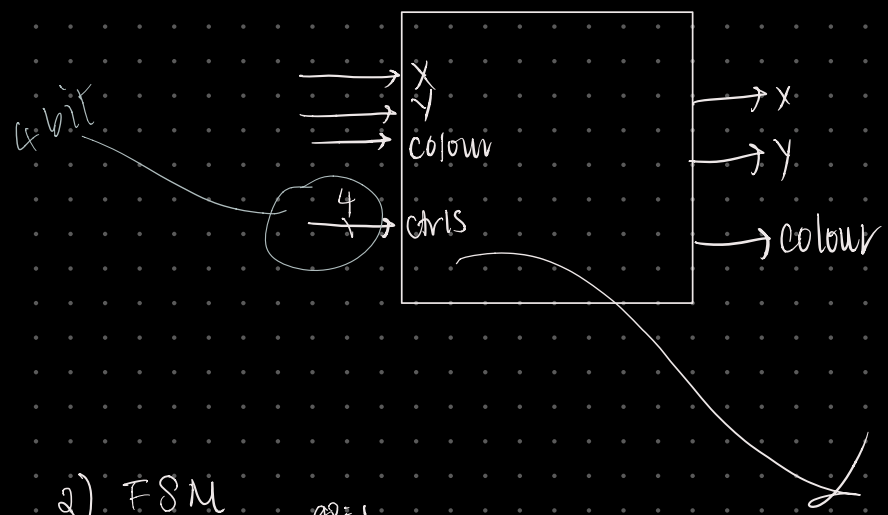
VGA Adapter

*modify the # of controllers (ctrlA/B/C etc) based on datapath

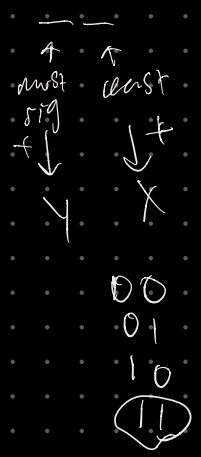
1) Datapath:

Inputs: X, Y, colour, ctrl signals

output: (x, y) to paint

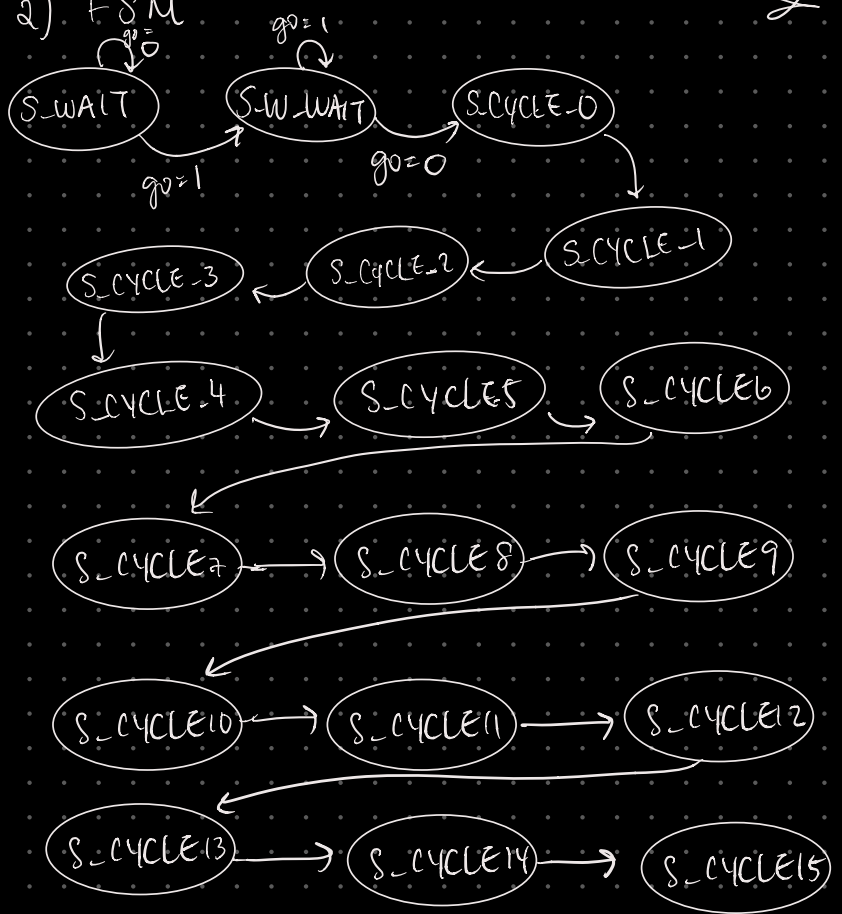


2x2



X,Y	X+1,Y
X,Y+1	X+1,Y+1

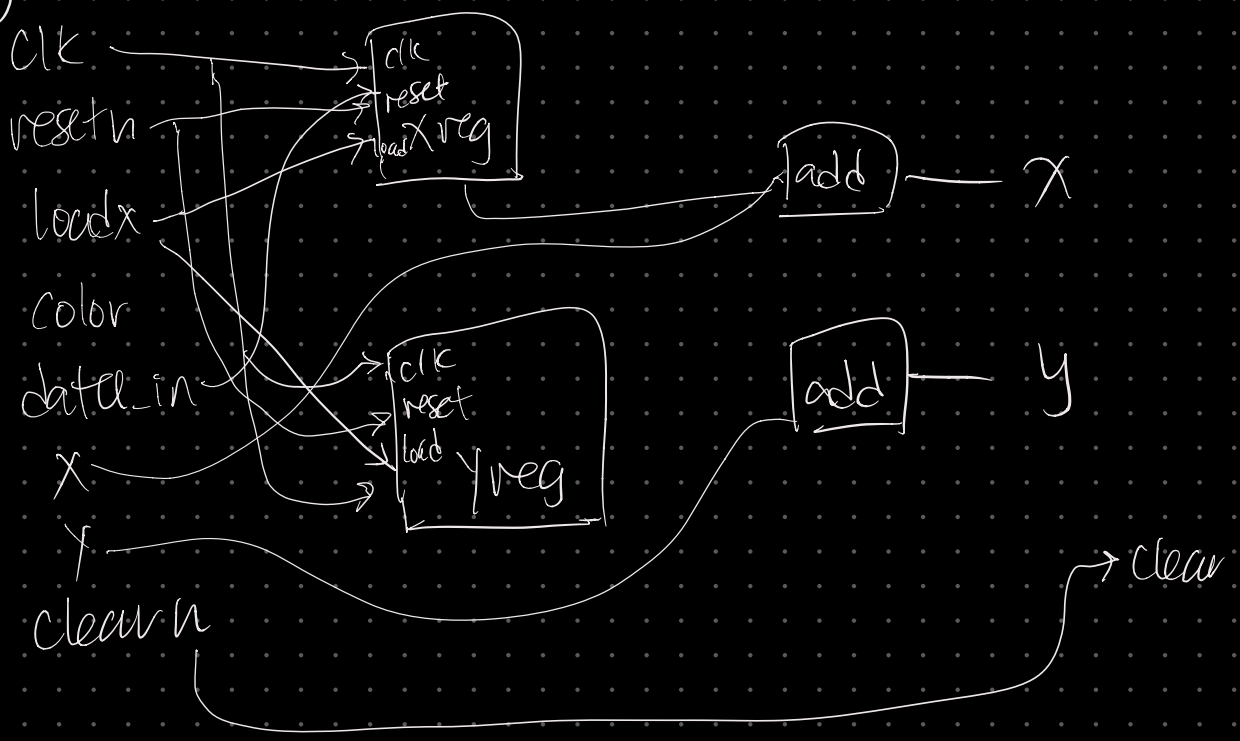
2) FSM



4-bit

X,Y	X+1,	X+2	X+3
X,Y+1			
Y+2			
Y+3			

1)


clk
resetn
loadx
color
data_in
x
y
clear_n
clk
reset
load
x reg
clk
reset
load
y reg
add
x
add
y
clear