

1. Video Encoder (35 pts)

Consider the video encoder from Checkpoint 1.

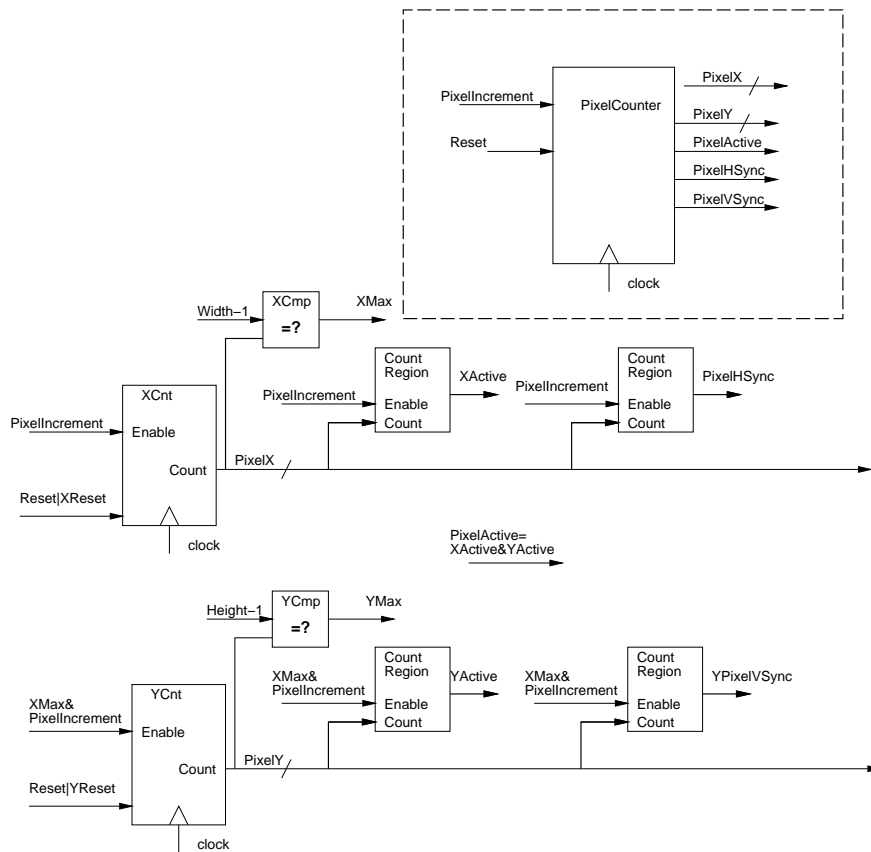
a. Describe in words the operation of the CountRegion.v module.

The CountRegion module is a comparator combined with a state machine, which monitors the Count of an external counter. Output Output is asserted if Count is within the active region between parameter Start and parameter End. Assuming EnablePre and EnablePost are true, the input Max is ignored.

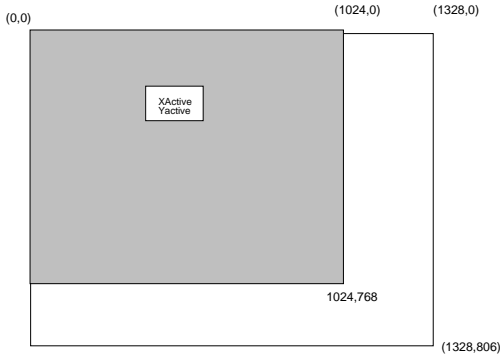
b. Describe in words the operation of the CountCompare.v module.

The CountCompare module asserts TerminalCount if the input Count is equal to the parameter Compare. This module is used to detect the maximum count for a counter.

c. Draw a detailed block diagram for the PixelCounter.v module, showing inputs, outputs, registers, counters, CountCompare, and CountRegion blocks.



d. Draw a sketch of a rectangular screen which shows the relation of XActive and YActive regions to the whole frame, including blanking regions.

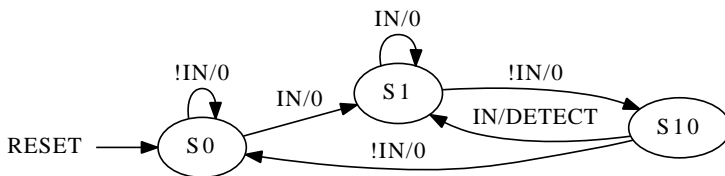


This is using the default parameters in the PixelCounter.v module. Using the FPGA_TOP_ML505.v parameters, the width is 1040 with 240 of blank region = 800, and the height is 666 with 66 of blank region = 600.

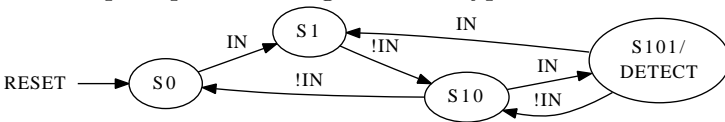
2. Finite State Machines, Moore vs. Mealy (35 pts)

Consider a sequence detector which detects the possibly overlapping pattern “101” in a synchronized input sequence IN. A global reset brings the sequence detector to the initial state corresponding to a “0” input.

a. Draw a state diagram for a Mealy type FSM which outputs DETECT for one clock cycle every time “101” is detected on IN.

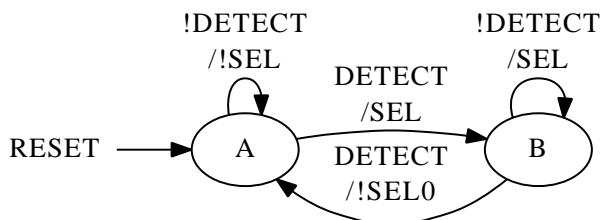


b. Repeat part a. using a Moore type FSM.

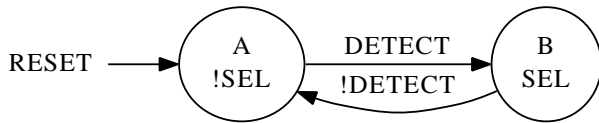


Consider two possible synchronized input streams A and B. A mux with 1 bit select line SEL is used to select source A (SEL=0) or source B (SEL=1) for the sequence detector.

c. Draw a state diagram for a 2 state Mealy-type FSM whose output SEL alternates between SEL=0 and SEL=1 as soon as input DETECT is asserted and maintains the current output SEL until the next assertion of input DETECT.



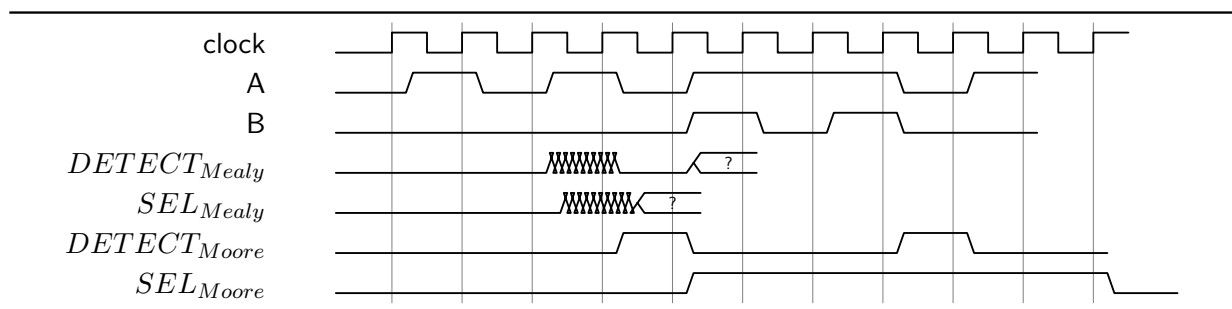
d. Repeat part c. but for a Moore type FSM.



Consider serial streams A=0101011101 and B=0000010100.

e. Consider the Mealy FSM sequence detector of part a. with an input mux controlled by the Mealy FSM of part c. Draw a timing diagram showing clock, A, B, DETECT, SEL.

f. Consider the Moore FSM sequence detector of part b. with an input mux controlled by the Mealy FSM of part c. Draw a timing diagram showing clock, A, B, DETECT, SEL.



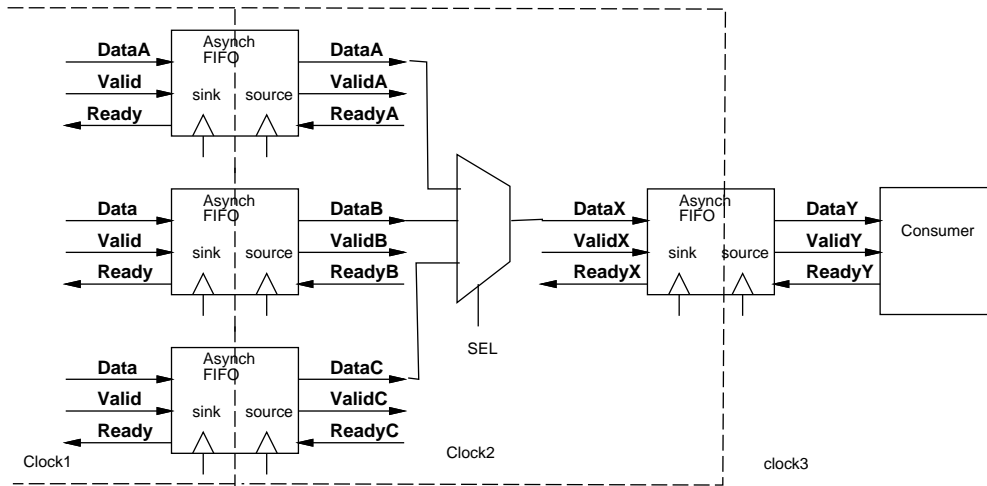
g. For part e. and f., explain any operational differences or problems found.

The Mealy FSMs form a combinational loop. After detecting “101” on the third clock edge, DETECT is asserted. However, DETECT goes through the SEL output and changes the MUX to stream “B” which has a 0 input, causing an oscillation. Thus the Mealy FSM could end up in state S1 or S10 (for a non-LUT gray code state assignment) or any random state depending on when the next clock edge arises. Basically, the Mealy to Mealy connection of FSMs will make a combinational loop and must be avoided. The Moore-Moore connection of FSMs works, but note that DETECT is delayed by one clock cycle, and then SEL is delayed by one more clock cycle. If B was B=0000010100 the “101” sequence on B would not have been detected. A Moore and a Mealy machine could have been connected here and would not have made a combinational loop.

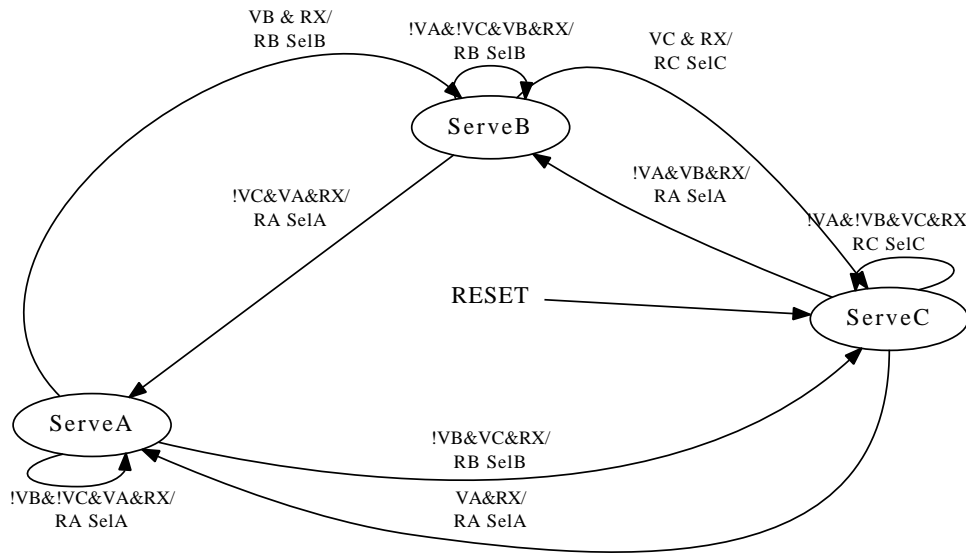
3. Round-Robin Arbiter and FIFOs (30 pts)

Consider the data path below.

a. Draw a state diagram for a round robin arbiter which, when ReadyX is asserted, will read data from FIFOs in order A,B,C,A,... if a FIFO has data ready. If a read FIFO is not asserting Valid, the scheduler should skip ahead to the next source in order. The FSM should be in clock domain 2 and generate all needed control signals, i.e. explain how to calculate the control signals from present state and inputs.



The state diagram is shown below, using a Mealy machine. Abbreviations: VA = ValidA, RA = ReadyA, VB = ValidB, RB = ReadyB, VC = ValidC, RC = ReadyC, VX = ValidX, RX = ReadyX. Control signals: The data select function is simply $DataX = ReadyA \& DataA \mid ReadyB \& DataB \mid ReadyC \& DataC$ where the Ready outputs come from the FSM. $ValidX = ReadyA \mid ReadyB \mid ReadyC$. Note that the FSM starts in state ServeC after Reset, so that FIFO A will be served first. (Note the implicit self transitions when !ReadyX.) A Moore type FSM could also be used, but would introduce a one clock delay from idle, and extra states.



b. Consider that at reset FIFO A,B,C contain respectively 3,2, and 1 data words. Clock2 runs at 10 MHz, and clock3 runs at 45 MHz, and Consumer can process one word every clock cycle. Draw a timing diagram showing clock2, Data, Ready, and Valid for A,B,C,X,Y. For timing diagram, data should be labelled A0, A1, etc. Assume the FIFO is setup in Read w/FWFT mode.

Note that the problem specifies First Word Fall Through, thus the FIFO has the first value already present (this saves a clock cycle). For reference, FIFO read with FWFT is shown below:

