Professor Fearing EECS150/Problem Set 6 Fall 2003

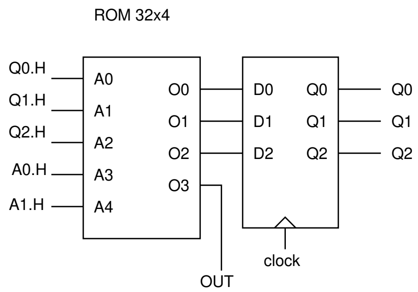
Due at 10 am, Fri. Oct. 17 in box outside 125 Cory Hall

Reading Ciletti: 8.2.7, "Video in a NutShell", project spec

(This problem set may be done individually or by a group of two people maximum. A group of two should hand in a single copy of the solution with both names affixed).

1. ROM Based FSM (20 pts)

Using the ROM based FSM shown below, with ROM contents as listed in Table 1, draw a state table and state diagram for the FSM. Explain why this is a Moore or Mealy type FSM. Consider synchronized inputs A1.H and A0.H. The following data is presented in order at A[1:0]: 2,1,0,1,1,1,3,2,0,0. Draw a timing diagram showing CLOCK, A1.H, A0.H, Q[2:0] and OUT.H.



Adr	Data	Adr	Data	Adr	Data	Adr	Data
0	8	8	2	10	4	18	6
1	9	9	3	11	5	19	7
2	A	A	4	12	6	1A	0
3	B	B	5	13	7	1B	1
4	C	C	6	14	0	1C	2
5	D	D	7	15	3	1D	5
6	E	E	0	16	2	1E	4
7	F	F	0	17	1	1F	2

2. Synchronization (35 pts)

The ADV7185 video decoder on the Calinx board sends an 8 bit stream of video on inputs P2[9:2]. Video level 0xFF is reserved for indicating timing information such as StartofField. An 8 input AND gate on P2[9:2] could be used to detect 0xFF, which would be input to an FSM which controls frame buffer row and column counters. The ADV7185 generates a nominal 27 MHz clock LLC1, while the FSM runs on a different 27.00 MHz clock VE_Clock. Explain, with the aid of a timing diagram, why each of the following schemes a-c would fail to guarantee proper detection of the 0xFF by the FSM. (Consider routing delays to the AND gate if appropriate.)

a. P2[9:2] connected to AND gate, AND gate output directly to FSM input.

b. P2[9:2] connected to AND gate, with synchronizing FF on AND gate output.

c. P2[9:2] connected to 8 bit D register running on VE_Clock, then AND gate on output of register connnected to FSM input.

d. Draw a block diagram of a system which would guarantee proper detection of the 0xFF. (Hint: async FIFO.)

3. Block Motion Estimation Algorithm Design (45 pts)

In this problem, you will prototype a block motion estimation algorithm to use in the final project. (It is much simpler to develop and debug this algorithm using a high level language without any of the timing, space, and real-time debugging limitations of the FPGA.) A set of test images have been captured and are accessible from the class homepage. The camera moves slightly between each frame. Each frame is represented as an array of `unsigned char image0[NROWS][NCOLS]` . Your algorithm should estimate the motion of an 8×8 block at 8 locations for the upper left element of this block: (124,4), (124,12), (124,20), (124,28), (124,32), (124,40), (124,48), (124,56). (Compare the 8×8 block within a 16x16 block from the next frame and find location with minimum absolute error.)

a. Explain the algorithm, e.g. mathematics and pseudo-code.

b. Show C-code and print motion estimate results for the test images at the 8 locations. What is a reasonable threshold for determining motion outside the 16x16 block?

c. Outline how your algorithm might be realized in Xilinx. In particular, how many adders do you need, how many clock cycles, and how much memory is used.