Midterm Exam ECE 448
Spring 2020
Friday, March 6
15 points

Instructions:
Zip all your deliverables into an archive <last_name>.zip and submit it through Blackboard no later than Friday, March 6, 11:25 AM EST.

The EXAM circuit implements an averaging filter for a grayscale image:

1. Averaging filter

Averaging filter is used to de-noise a grayscale image. A grayscale image is a matrix of 8-bit unsigned integers, in which each number represents a pixel. Pixel values are integers that range from 0 (black) to 255 (white). These values represent the amount of gray intensity to be displayed for that particular portion of the image.

Input:
M: a grayscale image of dimensions DIM rows by DIM columns. Before the beginning of processing, this image is stored in the internal RAM at locations 0..DIM^2-1 (which for DIM=6 is the range 0..35). An address at which a pixel with coordinates (x, y) is stored is given by DIM·x+y, where x=0..DIM-1 and y=0..DIM-1, as shown in figure below for DIM=6.

Output:
D: a grayscale image of dimensions DIM-2 rows by DIM-2 columns. After the end of processing, this image is stored in the internal RAM at locations DIM·x+y+DIM^2, where x=1..DIM-2 and y=1..DIM-2. For DIM=6, these locations are given by 6·x+y+36, where x=1..4 and y=1..4, as shown in the figure below.

x \ y	0	1	2	3	4	5
0	0	1	2	3	4	5
1	6	7	8	9	10	11
2	12	13	14	15	16	17
3	18	19	20	21	22	23
4	24	25	26	27	28	29
5	30	31	32	33	34	35

ADDRESSES
OF PIXELS
IN THE INPUT IMAGE

ADDR= 6x + y
x=0..5, y=0..5

36	37	38	39	40	41
42	43	44	45	46	47
48	49	50	51	52	53
54	55	56	57	58	59
60	61	62	63	64	65
66	67	68	69	70	71

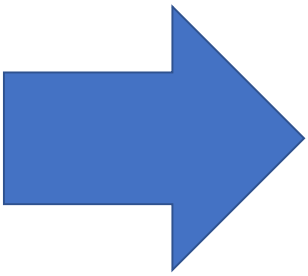
CORRESPONDING
ADDRESSES
OF PIXELS
IN THE OUTPUT IMAGE
(SHOWN IN GRAY)

ADDR= 6x + y + 36
x=1..4, y=1..4

Algorithm:
Averaging filter calculates the average value of 8 pixels neighboring with the pixel with dimensions x, y. It then averages this value with the value of a pixel itself.
In other words, the algorithm performs the following calculation:
S = M(x-1, y-1)+ M(x-1, y)+ M(x-1, y+1)+ M(x, y-1)+M(x, y+1)+ M(x+1, y-1)+ M(x+1, y)+ M(x+1, y+1)
Z = (S/8 + M(x, y))/2.
The division by 8 and 2 is assumed to be an integer division (with truncation), which is equivalent to a shift by 3 and 1, respectively.
Thus,
Z = (S>>3 + M(x, y))>>1.
This procedure is illustrated below.

Calculating 1st output pixel:
((10 + 22 + 80 + 6 + 98 + 255 + 12 + 200)/8 + 77)/2 = 81

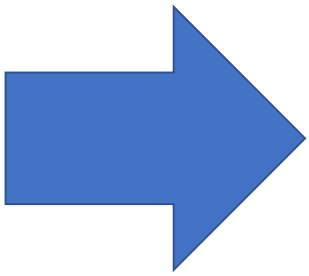
10	22	80	48	51	1
6	77	98	90	10	0
255	12	200	100	50	5
40	0	20	150	250	4
90	30	70	60	120	20
33	25	255	180	200	0



81			

Calculating 2nd output pixel:
((22 + 80 + 48 + 77 + 90 + 12 + 200 +100)/8 + 98)/2= 88

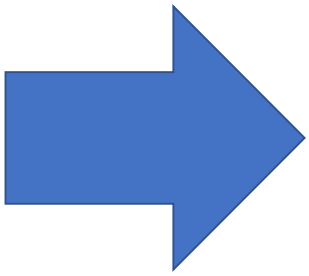
10	22	80	48	51	1
6	77	98	90	10	0
255	12	200	100	50	5
40	0	20	150	250	4
90	30	70	60	120	20
33	25	255	180	200	0



81	88		

Calculating 4th output pixel:
((48 + 51 + 1 + 90 + 0 + 100 + 50 + 5)/8 + 10)/2= 26

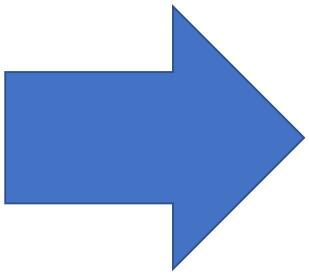
10	22	80	48	51	1
6	77	98	90	10	0
255	12	200	100	50	5
40	0	20	150	250	4
90	30	70	60	120	20
33	25	255	180	200	0



81	88	78	26

Calculating 5th output pixel:
((6 + 77 + 98 + 255 + 200 + 40 + 0 + 20)/8+12)/2= 49

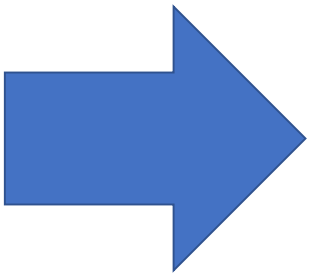
10	22	80	48	51	1
6	77	98	90	10	0
255	12	200	100	50	5
40	0	20	150	250	4
90	30	70	60	120	20
33	25	255	180	200	0



81	88	78	26
49			

Calculating 16th output pixel:
((150 + 250 + 4 + 60 + 20 + 180 + 200 + 0)/8 + 120)/2= 114

10	22	80	48	51	1
6	77	98	90	10	0
255	12	200	100	50	5
40	0	20	150	250	4
90	30	70	60	120	20
33	25	255	180	200	0



81	88	84	26
49	134	104	63
44	48	129	156
48	80	107	114

Similarly, we can calculate values of the remaining output pixels.

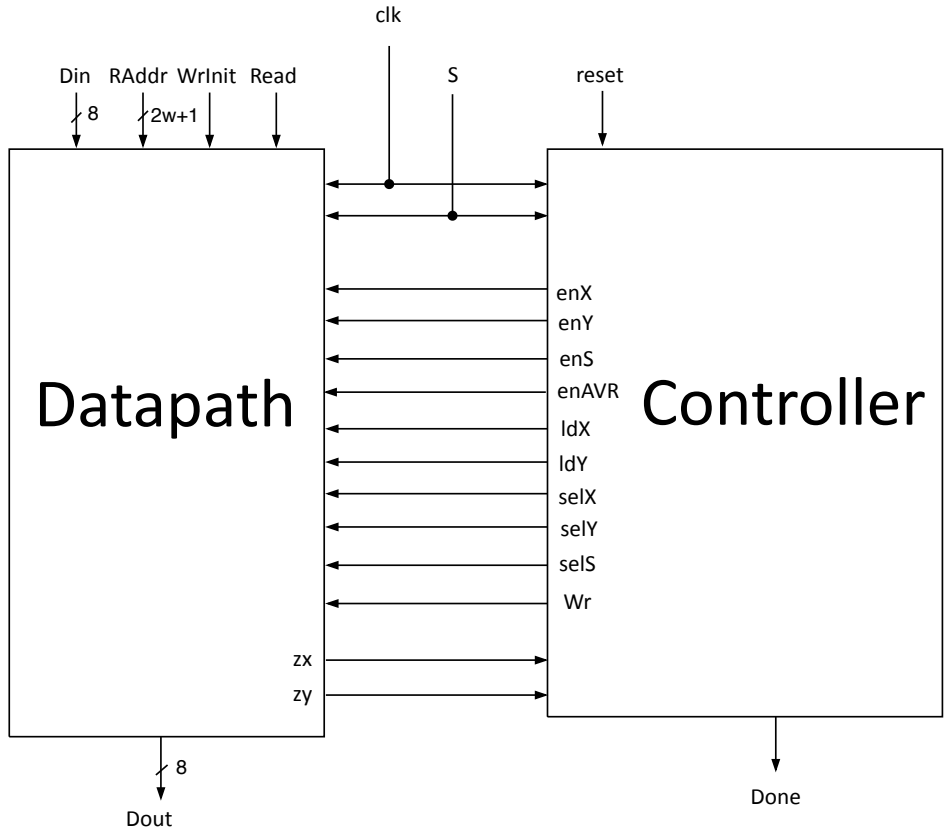
The circuit is specified below using its

- Table of input/output ports
- Interface with the division into the Datapath and Controller
- Block diagram of the Datapath
- ASM chart of the Controller.

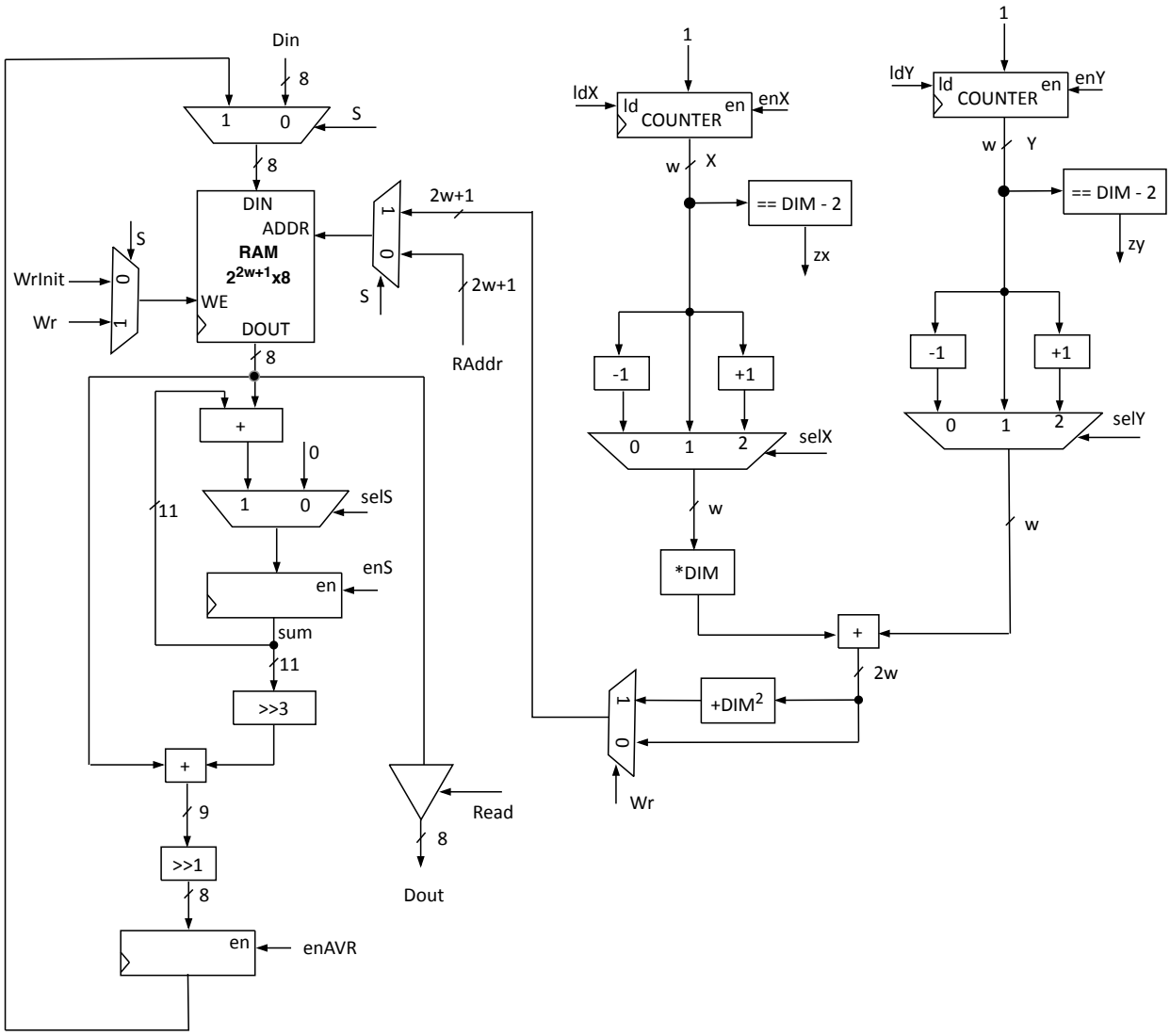
Table of input/output ports

Port	Width	Meaning
clk	1	System clk
reset	1	System reset
S	1	Operating mode: 0 = initialization or reading results, 1 = computations.
Din	8	Input data bus
Dout	8	Output data bus
RAddr	2w+1	Address of the internal memory where input data is stored
WrInit	1	Synchronous write control signal used during the initialization of internal memory
Read	1	Read enable: 0 = high impedance on the output bus, 1 = valid output on the output data bus.
Done	1	Output control signal indicating that the filter operation is complete

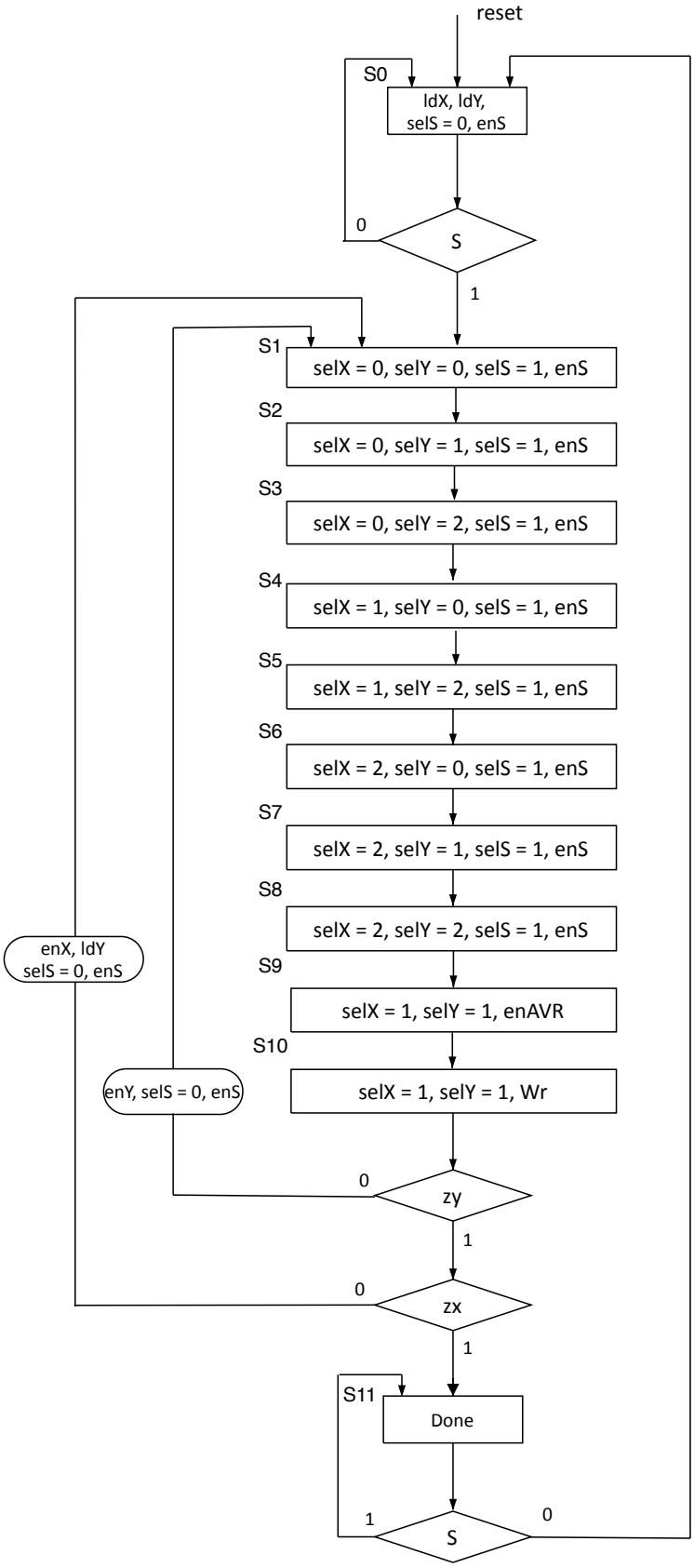
Top-level circuit interface



Block diagram of the Datapath



ASM Chart


reset
S0
ldX, ldY,
selS = 0, enS
0
S
1
S1
selX = 0, selY = 0, selS = 1, enS
S2
selX = 0, selY = 1, selS = 1, enS
S3
selX = 0, selY = 2, selS = 1, enS
S4
selX = 1, selY = 0, selS = 1, enS
S5
selX = 1, selY = 2, selS = 1, enS
S6
selX = 2, selY = 0, selS = 1, enS
S7
selX = 2, selY = 1, selS = 1, enS
S8
selX = 2, selY = 2, selS = 1, enS
S9
selX = 1, selY = 1, enAVR
S10
selX = 1, selY = 1, Wr
enX, ldY
selS = 0, enS
enY, selS = 0, enS
0
zy
1
0
zx
1
S11
Done
1
S
0

Tasks:

Perform the following tasks:

1. Write a synthesizable VHDL code representing the circuit.
2. Write a testbench verifying the operation of your circuit for DIM=6, w=3, and the input image given in the algorithm description on pages 3 and 4.
3. Perform functional simulation of your circuit and use it to debug your VHDL code.
4. Synthesize your circuit.
5. Implement your circuit using
    a. FPGA family: Artix-7
    b. Part name: XC7A35TCPG236-1
    c. Speed Grade: -1
6. Based on the implementation reports, determine the number of CLB slices, LUTs, flip-flops, and pins used by the circuit.

Deliverables:

1. VHDL code of your entire circuit.
2. VHDL code of your testbench.
3. Timing waveforms from the functional simulation demonstrating outputs generated by your circuit.
4. FPGA resource utilization (as defined in Task 6 above).