ECE 278 Final Project

2 x 2 Matrix Multiplier

Chris Folden
Dane Zetouna
Jimmy Galat
Alec Spurlock

12-1-2016

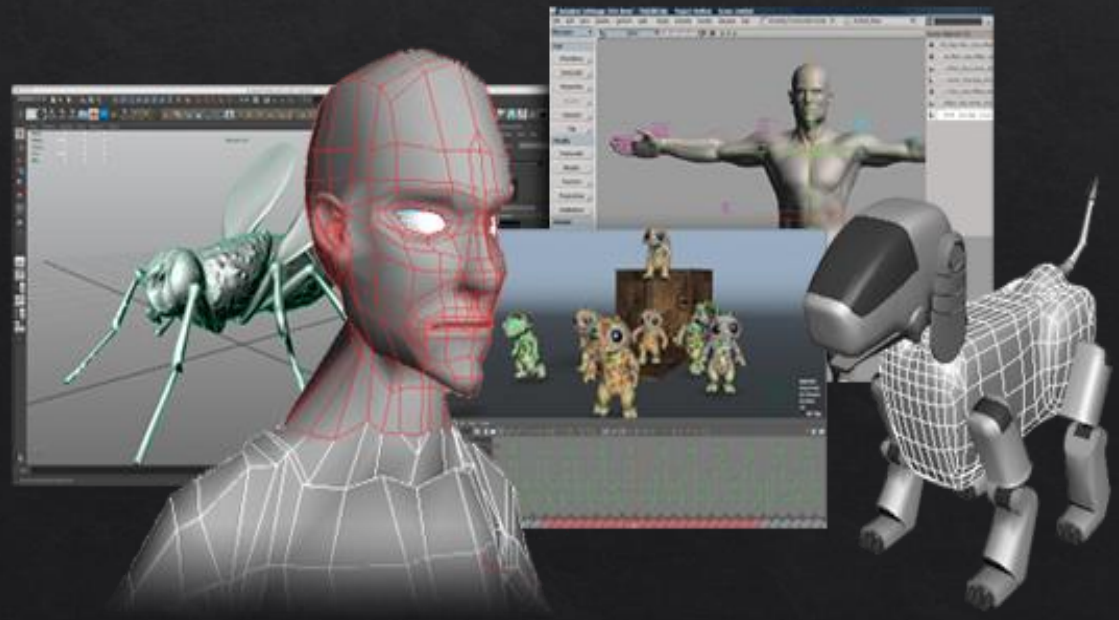
How Matrix Multiplication Works

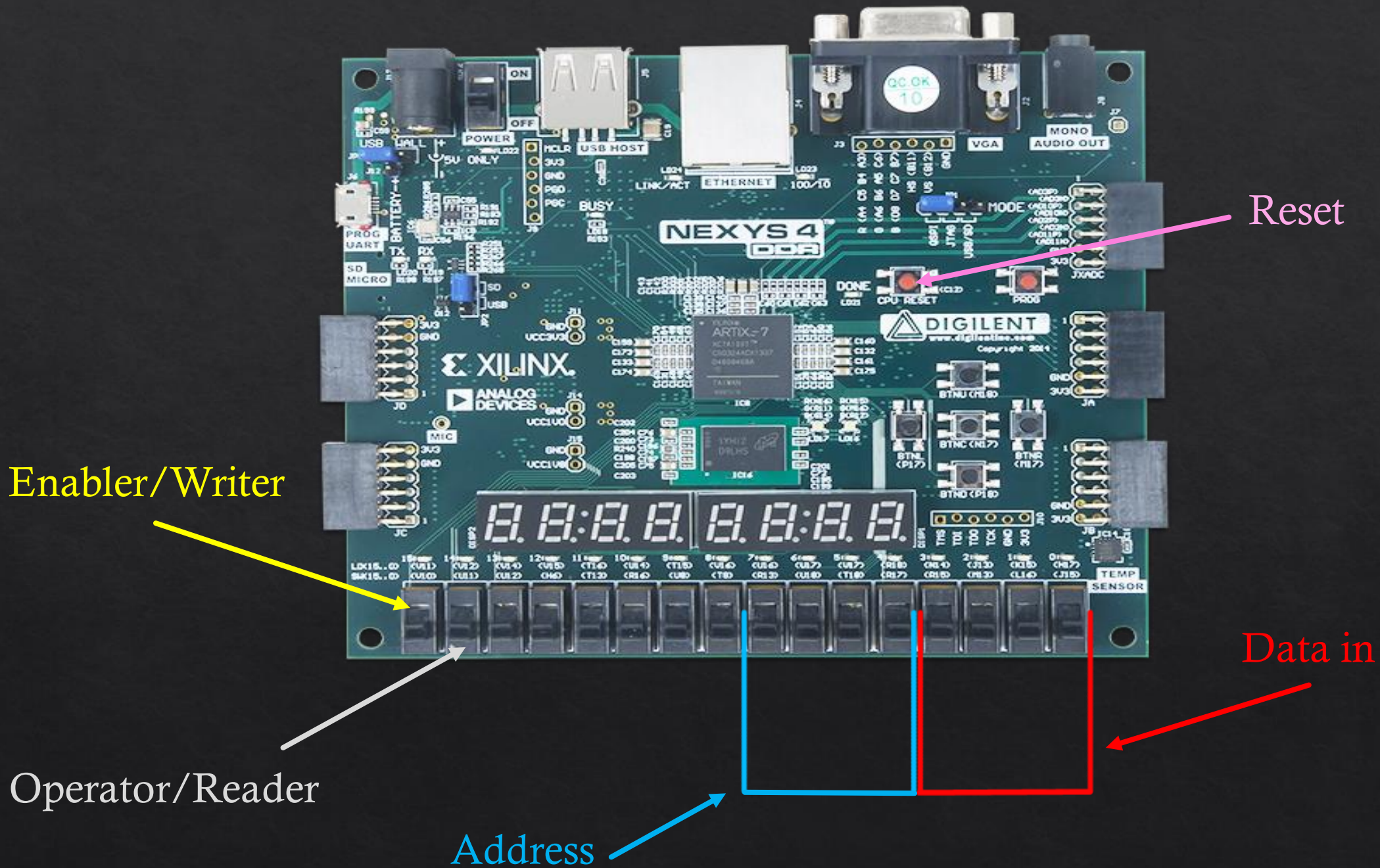
$$\begin{bmatrix} e & f \\ g & h \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} ae + cf & be + df \\ ag + ch & bg + dh \end{bmatrix}$$

- A matrix organizes a group of numbers using specified arithmetic rules.
- Matrix multiplication is a binary operation that generates a single matrix from two matrices.
- This is applicable to many different mathematical disciplines as well as several branches of science.

Applications of Matrix Multiplication

- Matrix multiplication is applicable to many different mathematical disciplines as well as several branches of science.
- One application you can relate to easily is perspective projections, which is the foundation for 3D animation.
- Another example is in the field of medicine, CAT scans and MRI's use matrices.

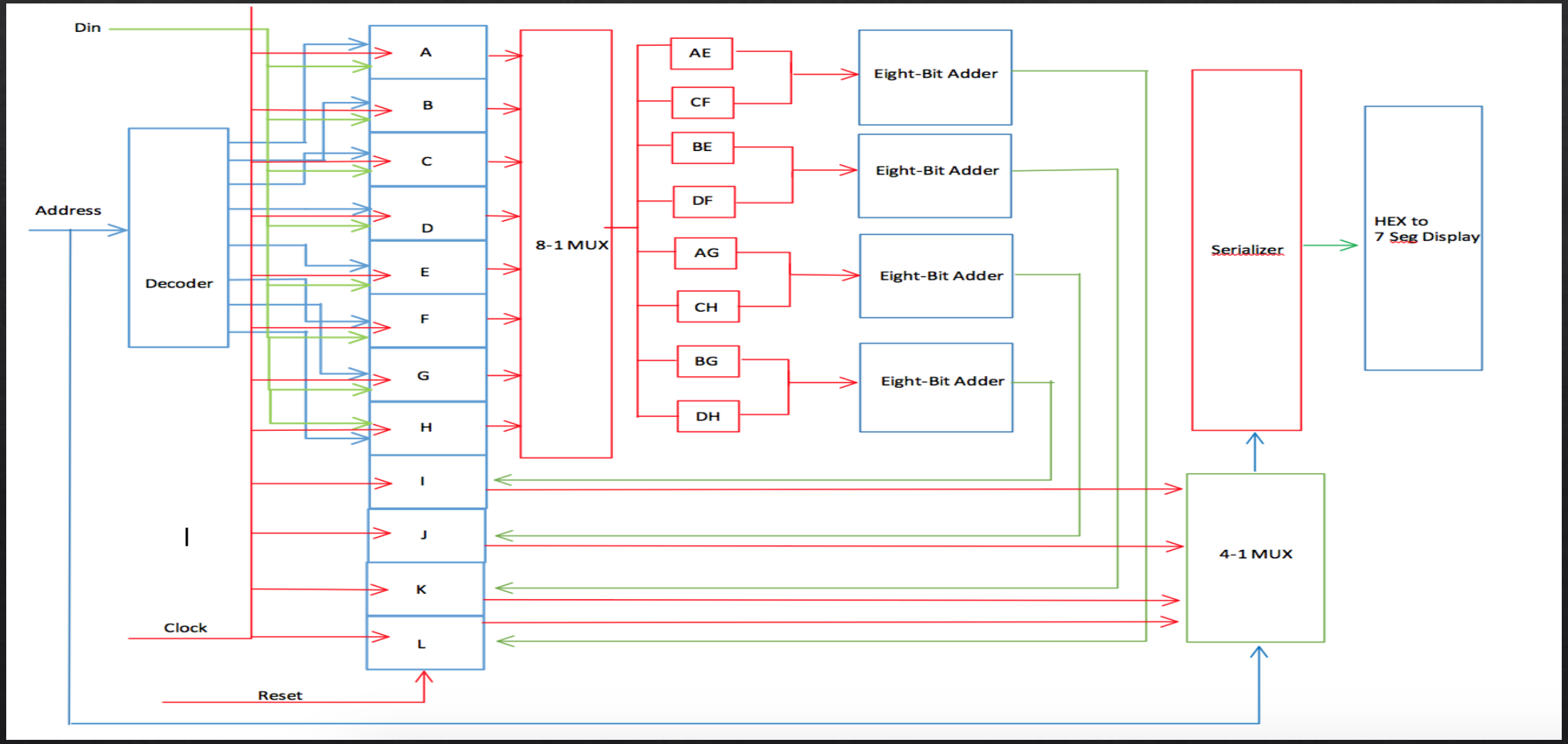



Reset
Enabler/Writer
Operator/Reader
Data in
Address
NEXYS 4
DDR
XILINX
DIGILENT
ANALOG DEVICES
ARTIX-7
CPU RESET
DONE
MODE
VGA
ETHERNET
USB HOST
POWER
MONO AUDIO OUT
TEMP SENSOR

Layout of the Project

- After many ideas were discussed, the plan was to have a 4-bit address input into the decoder.
- This will select which of the 8 registers to write into and which registers to read the data from.
- From there, the data will be taken from the first 8 registers and then multiplied and added together in the format shown in the first slide.

Matrix Multiplication Top-File



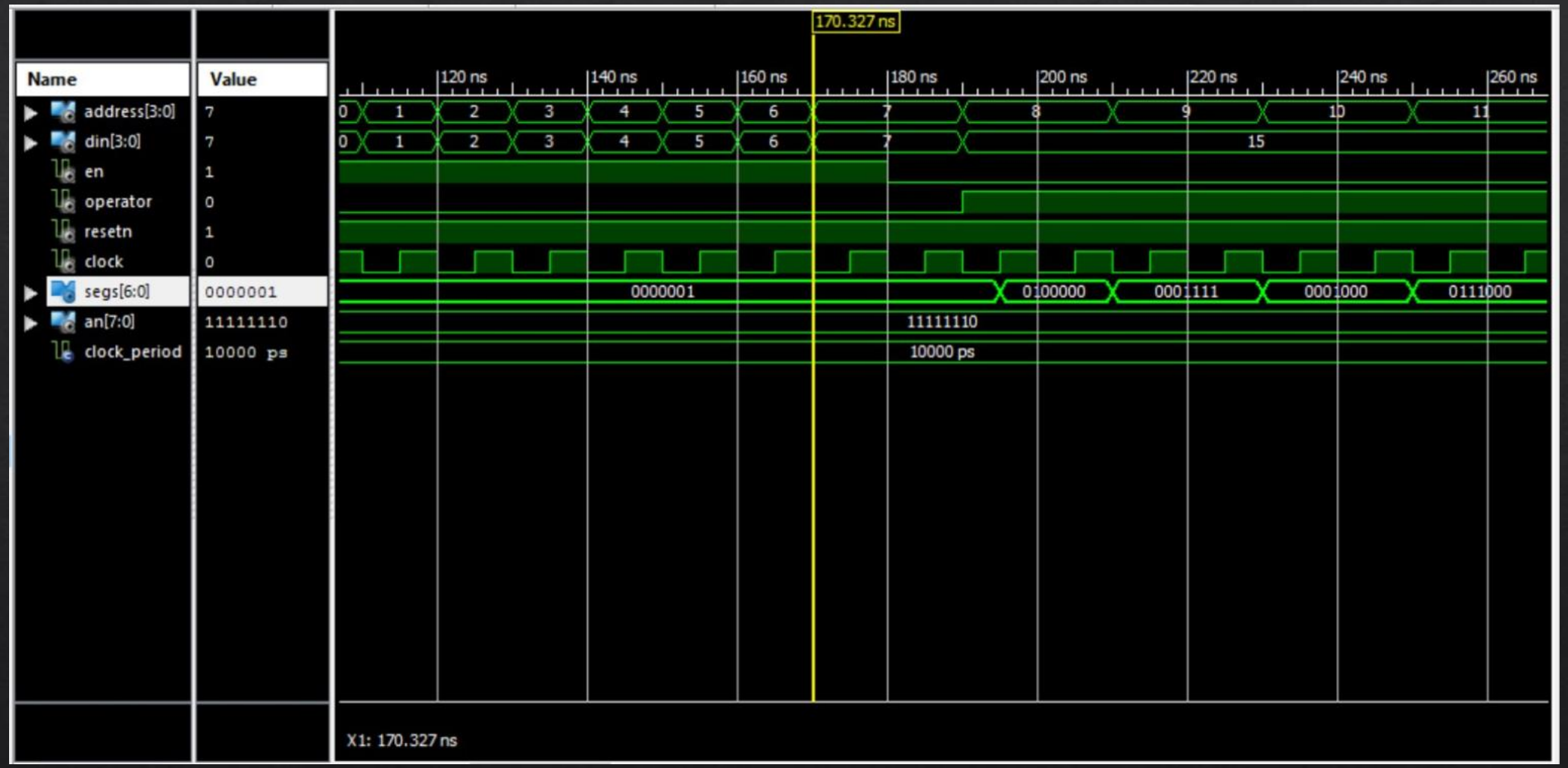
Code Sample

```
a20: Multiplexer port map (mux0 => qmux0, mux1 => qmux1, mux2 => qmux2, mux3 => qmux3, mux4 => qmux4, mux5 => qmux5, mux6 => qmux6, mux7 => qmux
mux_out => h, address => "0111", inv_E => Operator);

a21: Multiplexer2 port map (mux8 => qmux8, mux9 => qmux9, mux10 => qmux10, mux11 => qmux11, mux_out => MatrixOut, address => address, inv_E => O

f0: Array_Multiplier port map (a => a, b => e, p => ae);-- multiplies a and e
f1: Array_Multiplier port map (a => b, b => g, p => bg);-- multiplies b and g
f2: Array_Multiplier port map (a => a, b => f, p => af);-- multiplies a and f
f3: Array_Multiplier port map (a => b, b => h, p => bh);-- multiplies b and h
f4: Array_Multiplier port map (a => c, b => e, p => ce);-- multiplies c and e
f5: Array_Multiplier port map (a => d, b => g, p => dg);-- multiplies d and g
f6: Array_Multiplier port map (a => c, b => f, p => cf);-- multiplies c and f
f7: Array_Multiplier port map (a => d, b => h, p => dh);-- multiplies d and h

f8: Five_Bit_Adder port map (Cin => '0', x => ae, y => bg, s => ae_bg, Cout => C_ae_bg); -- Adding ae + bg
ele1 <= C_ae_bg & ae_bg;
```


170.327 ns
Name
Value
120 ns
140 ns
160 ns
180 ns
200 ns
220 ns
240 ns
260 ns
address[3:0]
7
din[3:0]
7
en
1
operator
0
resetn
1
clock
0
segs[6:0]
0000001
an[7:0]
11111110
clock_period
10000 ps
0000001
0100000
0001111
0001000
0111000
11111110
10000 ps
X1: 170.327 ns


Name
Value
address[3:0]
0111
din[3:0]
8
en
1
operator
0
resetn
1
clock
0
element1[8:
0
element2[8:
0
element3[8:
0
element4[8:
0
clock_perio
10000 ps
160 ns
180 ns
200 ns
220 ns
240 ns
260 ns
270.897 ns
280 ns
0000
0001
0010
0011
0100
0101
0110
0111
0
1
2
3
4
5
6
7
8
19
22
43
50
10000 ps

Demonstration Time!!

Problems and Possible Improvements

- Signed Numbers

- Displaying in Decimal

- Bigger Matrices