Lab #2

Static Random-Access Memory

(SRAM)

Name: Chen, Kevin

Date: 03/06/2019

Course: CSC 34200-G & CSC 34300-DE

Table of Contents

Objective **4**

Functionality and Specification **5**

SR-Latch 5

Truth Table 5

Block Diagram/Schematic File 5

Control SR-Latch 6

Truth Table 6

Block Diagram/Schematic File 6

D-Latch 7

Truth Table 7

Block Diagram/Schematic File 7

SRAM 8

Master-Slave D Flip-Flop 8

Tristate Logic Buffer 9

Block Diagram/Schematic File 9

16 x 1 SRAM 10

Block Diagram/Schematic File 10

16 x 4 SRAM 11

Block Diagram/Schematic File 11

16 x 32 SRAM 12

Block Diagram/Schematic File 12

Simulation **13**

SR-Latch 13

Waveform Simulation of Block Diagram/Schematic File 13

Control SR-Latch 14

Waveform Simulation of Block Diagram/Schematic File 14

D-Latch 15

Waveform Simulation of Block Diagram/Schematic File 15

SRAM 16

Waveform Simulation of Block Diagram/Schematic File 16

Demonstration **17**
16 x 4 SRAM 17
16 x 32 SRAM 18
Conclusion **19**
Questions and Answers 19
Appendix **21**
4-to-16 Decoder 21
VHDL Code 21
Decimal to Hexadecimal Translation 22
VHDL Code 22
16 x 4 SRAM 23
Pin Assignments 23
Pin Planner of Block Diagram/Schematic File 24
4 x 8 Switches and Keys 25
VHDL Code 25
16 x 32 SRAM 26
Pin Assignments 26
Pin Planner of Block Diagram/Schematic File 30

Objective:

The goal of this lab is to create a static random-access memory chip using D-Latches, also known as SRAM. An SRAM is used to store multiple bits at various address locations. The SRAM that we will be designing will hold 16-bit wide values and could display in the 7-segment HEX displays and decimals. We will be creating 2 forms of SRAM, one 16 x 4 SRAM and one 16 x 32 SRAM. The 16 x 4 SRAM will contain 16 locations, 4 bits each and the 16 x 32 SRAM will contain 16 locations, 32 bits each. With the 16 x 32 SRAM, we will also learn how to design a way to input 32 bits to memory word using 4 times 8 switches and keys.

Functionality and Specification:

SR-Latch:

A SR-Latch is a circuit that compares two inputs and lets out a result of two outputs. It acts as two functions, set and reset, which are controlled by the inputs, “S” and “R”, respectively. When “R” is 0 and “S” is 0, then the output remains the same as the previous state. When “R” is 1 and “S” is 0, then the output resets the device, making “Q” become 0. When “R” is 0 and “S” is 1, then the output sets the device making “Q” become 1. When “R” is 1 and “S” is 1, then the output is undefined because you can’t have it both set and reset at the same time, so it will be impossible to predict the output.

The truth table, and circuit are shown below:

S	R	Q	Q’	State
0	0	-	-	No Change
0	1	0	1	Reset
1	0	1	0	Set
1	1	?	?	Undefined

Table 1: Truth Table for SR-Latch

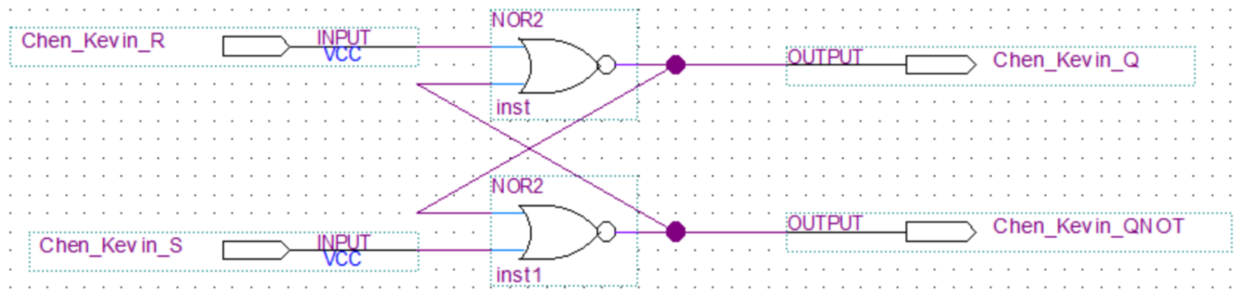


Figure 1: Block Diagram/Schematic File of SR-Latch (Chen_Kevin_SR-Latch.BDF)

Control SR-Latch:

A Control SR-Latch is a circuit that compares three inputs and lets out a result of two outputs. It acts the same as a SR-Latch except with an additional input, Control (C). The control input basically determines if the latch should change its state and take in the current inputs of "S" and "R". When "R" is 0 and "S" is 0, then the output remains the same as the previous state. When "C" is 0, then it doesn't matter what "R" and "S" is because the output will always remain the same as the previous state. When "C" is 1, then the inputs and output will act the same way as the SR-Latch.

The truth table, and circuit are shown below:

C	**S**	**R**	**Q_n**	**Q_n'**	**State**
0	X	X	Q	Q'	No Change
1	0	0	Q	Q'	No Change
1	0	1	0	1	Reset
1	1	0	1	0	Set
1	1	1	?	?	Undefined

Table 2: Truth Table for Control SR-Latch

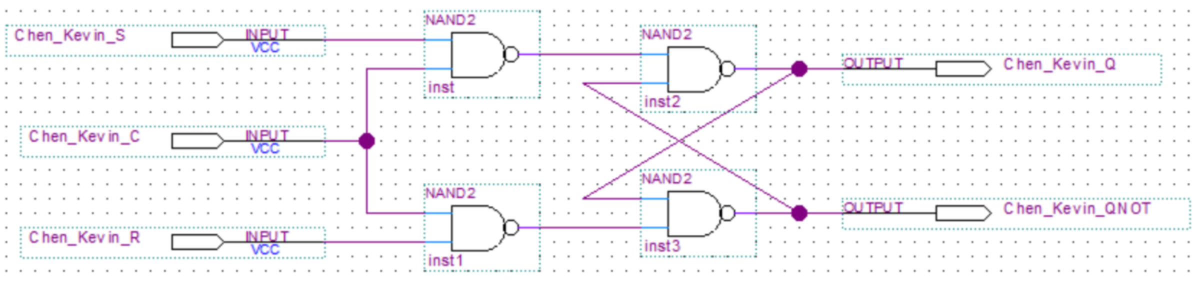


Figure 2: Block Diagram/Schematic File of Control SR-Latch

(Chen_Kevin_Control_SR-Latch.BDF)

D-Latch:

A D-Latch is a circuit that compares two inputs and lets out a result of two outputs. It acts the same as a Control SR-Latch except with a combined input, “D”. The “D” input basically combines the “S” and “R” input into one input. When “C” is 0, then it doesn’t matter what “D” is because the output will always remain the same as the previous state. When “C” is 1, then the output would depend on what “D” is. If “D” is 0, then the output resets the device, making “Q” become 0. If “D” is 1, then the output sets the device, making “Q” become 1.

The truth table, and circuit are shown below:

C	**D**	**Q_n**	**Q_n’**	**State**
0	X	Q	Q’	No Change
1	0	0	1	Reset
1	1	1	0	Set

Table 3: Truth Table for D-Latch

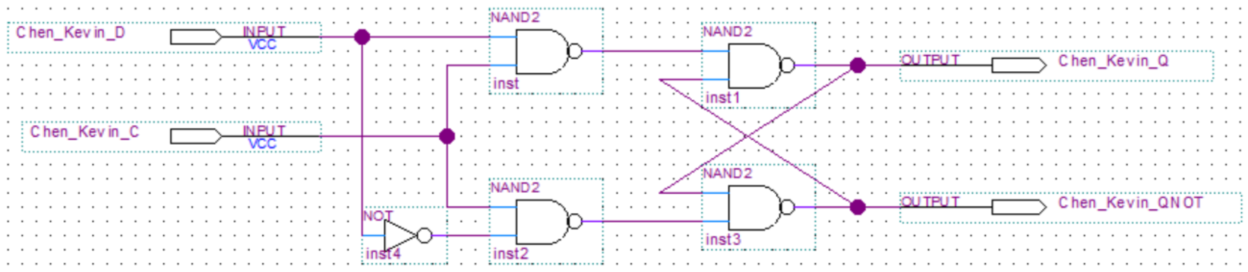


Figure 3: Block Diagram/Schematic File of D-Latch (Chen_Kevin_D-Latch.BDF)

SRAM:

A SRAM is a circuit that compares three inputs and lets out a result of one output. It's made from two D-Latches, which could also be described as a Master-Slave D Flip-Flop. A Master-Slave D Flip-Flop isn't the same a regular D-Latch because its sate change does not depend on the state of the clock signal, but the transition of the clock signal from zero to one. A SRAM is also made of a Tristate Logic Buffer, which compares two inputs and lets out a result of one output. The two inputs are "A" and "B" and it only outputs "A" when "B" is 1, otherwise the output is cut off completely and no signal is sent. The three inputs of a SRAM are "IN", "Select_Chip", and "Write_Enable". "IN" is the latch data input. "Select_Chip" is the input that activates the cell when it is high; when it's low, then the cell will not store data, and there will not be an output. "Write_Enable" is the input that allows the latch to store the data from the "IN" data input.

The truth table, and circuit are shown below:

Clock	**D_M**	**C_M**	**D_S**	**C_S**	**Q_{out}**
0	d1	1	q1	0	Q0
1	d2	0	q1	1	Q1
0	d3	1	q3	0	Q1
0	d4	1	q4	0	Q1
1	d5	0	q4	1	Q4
1	d6	0	q4	1	Q4

Table 4: Truth Table for Master-Slave D Flip-Flop

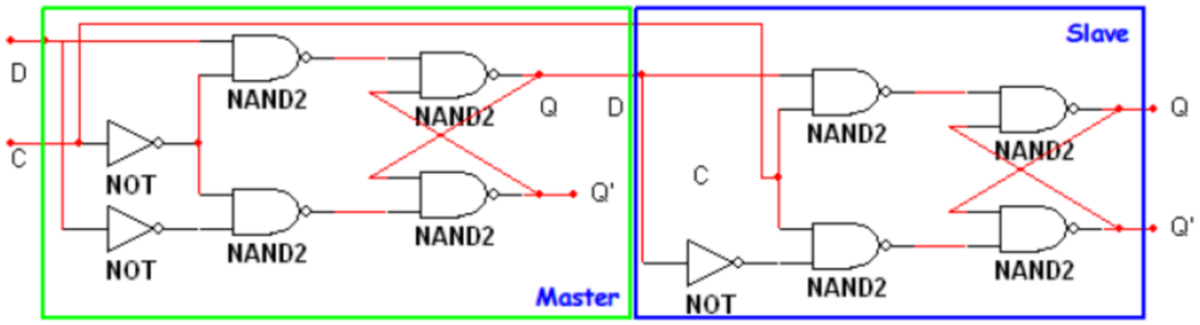


Figure 4: Master-Slave D Flip-Flop Design

A	B	C
0	1	0
1	1	1
X	0	Z

Table 5: Truth Table for Tristate Logic Buffer

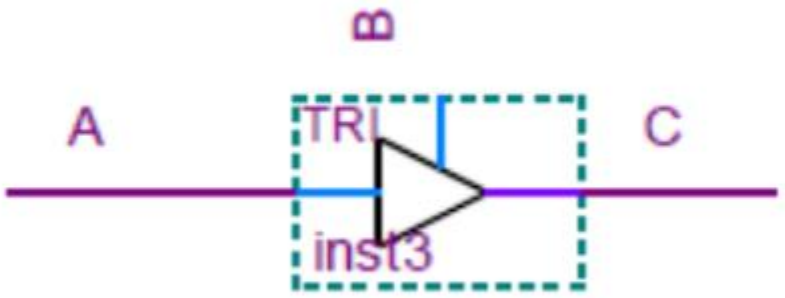


Figure 5: Tristate Logic Buffer Design

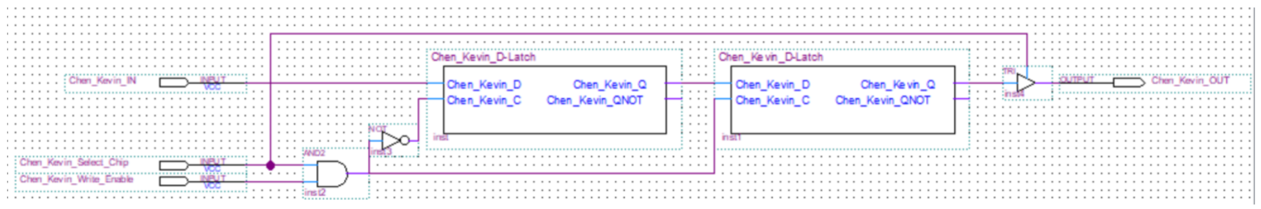


Figure 6: Block Diagram/Schematic File of SRAM (Chen_Kevin_SRAM.BDF)

<u>16 x 1 SRAM:</u>

A 16 x 1 SRAM is a circuit that's made of 16 SRAM that's all stacked into one file. It takes in the same input as a SRAM, except now, the data values are in a form of vectors. Dec[15..0] acts as "Select_Chip" from SRAM, except being a vector from 0 to 15. "IN" acts as "IN" from SRAM, and "WR" acts as "Write_Enable" from SRAM. The output of the 16 x 1 SRAM, "out", is the same as "OUT" from SRAM.

The circuit is shown below:

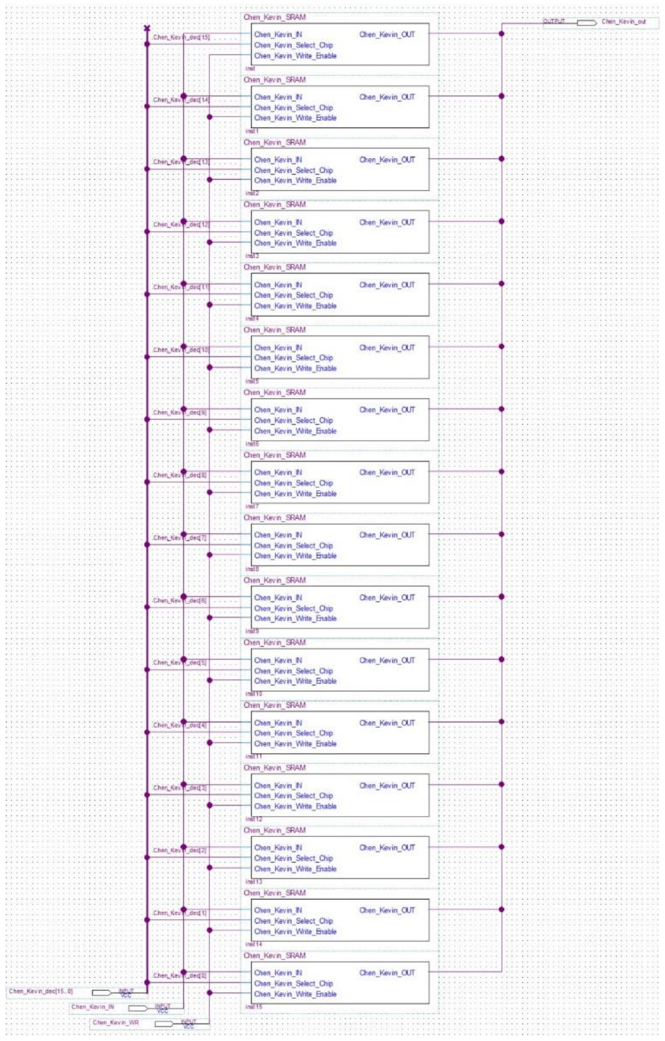


Figure 7: Block Diagram/Schematic File of 16 x 1 SRAM (Chen_Kevin_16x1SRAM.BDF)

16 x 4 SRAM:

A 16 x 4 SRAM is a circuit that's made of four 16 x 1 SRAM that's all stacked into one file. It's also made from a 4-to-16 Decoder and a Decimal to Hexadecimal Translation. A 4-to-16 Decoder is circuit that compares four inputs and lets out a result of sixteen outputs. The outputs are determined based on the inputs, and usually only one of the sixteen outputs are set to 1 and the others are set to 0, otherwise it's all set to 0. A Decimal to Hexadecimal Translation is a circuit that compares four inputs and lets out a result of seven outputs. The outputs are based on which seven-segments makes up the number corresponding to the inputs. The seven outputs would be used to display on a 7-segment-display, each output representing each segment.

The circuit is shown below:

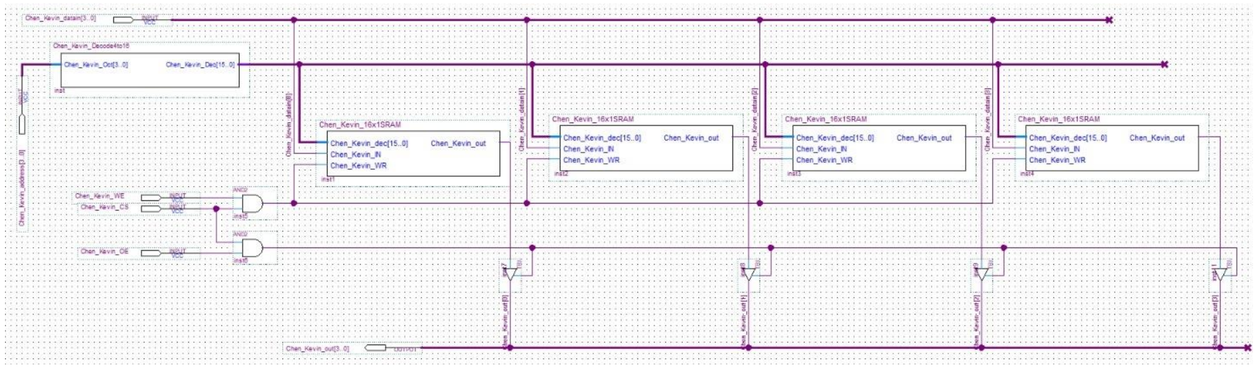


Figure 8: Block Diagram/Schematic File of 16 x 4 SRAM (Chen_Kevin_16x4SRAM.BDF)

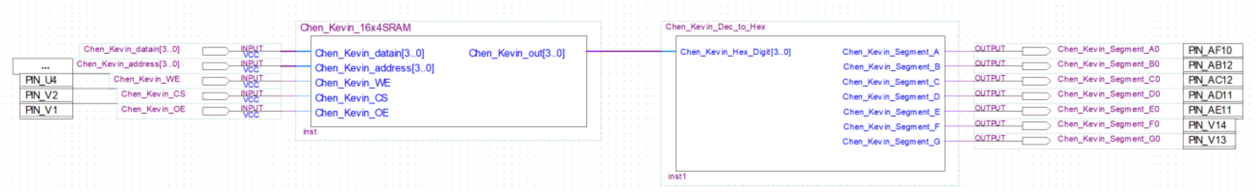


Figure 9: Block Diagram/Schematic File of Final 16 x 4 SRAM (Chen_Kevin_16x4SRAMFinal.BDF)

16 x 32 SRAM:

A 16 x 32 SRAM is a circuit that's made of eight 16 x 4 SRAM that's all stacked into one file. It's also made from 8 Decimal to Hexadecimal Translation and a 4 x 8 Switches and Keys. A 4 x 8 Switches and Keys is a circuit that compares 12 inputs and lets out a result of 32 outputs. Because we don't expect to have a pin for each 32 inputs, for the datain, this was designed to simplify the 32 inputs by using four keys and eight switches to represent each eight of the 32 inputs. For example, key[0] represents datain from 0 to 7, key[1] represents datain from 8 to 15, key[3] represents datain from 16 to 23, and key[4] represents datain from 24 to 31. The switches are from 0-7, which are used to configure each of the keys.

The circuit is shown below:

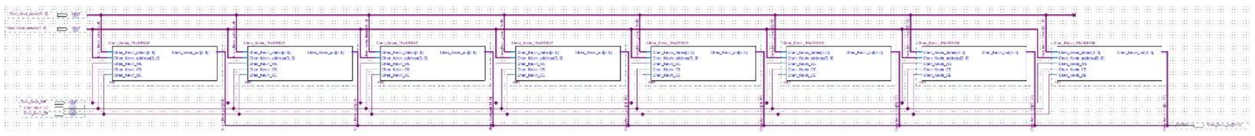


Figure 10: Block Diagram/Schematic File of 16 x 32 SRAM (Chen_Kevin_16x32SRAM.BDF)

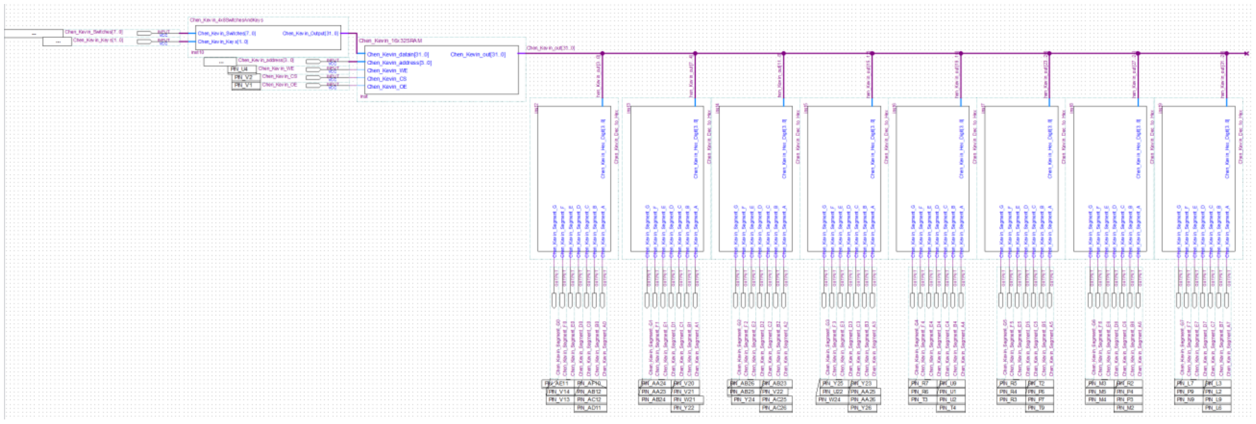


Figure 11: Block Diagram/Schematic File of Final 16 x 32 SRAM

(Chen_Kevin_16x32SRAMFinal.BDF)

Simulations:

SR-Latch:

The functionality/behavior of the SR-Latch is to follow the truth table (Table 1), which describes exactly what the output is supposed to be given a certain input.

A waveform was created using the block diagram/schematic file for SR-Latch (Figure 1) and it was given inputs to test all possible combinations. The outputs that it generated matches those from table 1, therefore, confirming the correctness of the block diagram/schematic file.

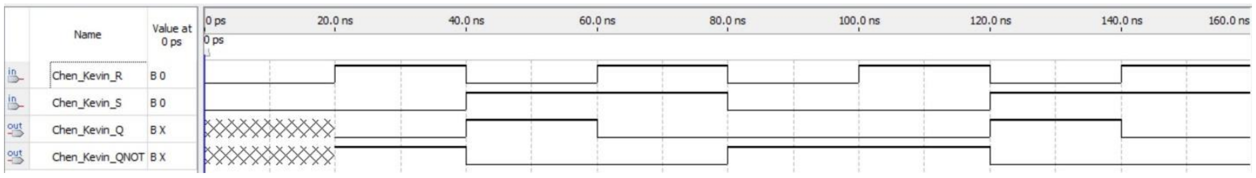


Figure 12: Waveform Simulation of Block Diagram/Schematic File of SR-Latch (Chen_Kevin_SR-LatchWaveform.VWF)

Control SR-Latch:

The functionality/behavior of the Control SR-Latch is to follow the truth table (Table 2), which describes exactly what the output is supposed to be given a certain input.

A waveform was created using the block diagram/schematic file for Control SR-Latch (Figure 2) and it was given inputs to test all possible combinations. The outputs that it generated matches those from table 2, therefore, confirming the correctness of the block diagram/schematic file.

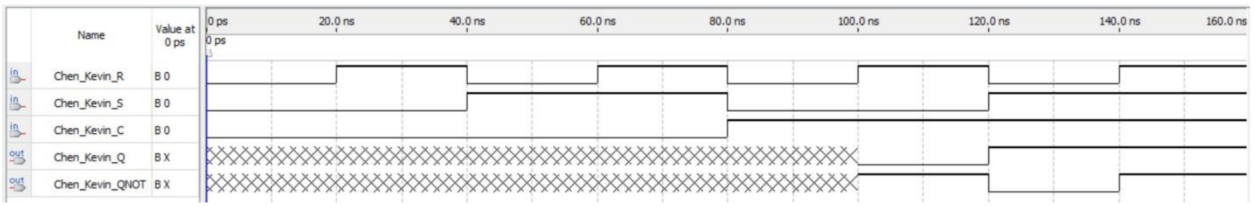


Figure 13: Waveform Simulation of Block Diagram/Schematic File of Control SR-Latch (Chen_Kevin_Control_SR-LatchWaveform.VWF)

<u>D-Latch:</u>

The functionality/behavior of the D-Latch is to follow the truth table (Table 3), which describes exactly what the output is supposed to be given a certain input.

A waveform was created using the block diagram/schematic file for D-Latch (Figure 3) and it was given inputs to test all possible combinations. The outputs that it generated matches those from table 3, therefore, confirming the correctness of the block diagram/schematic file.

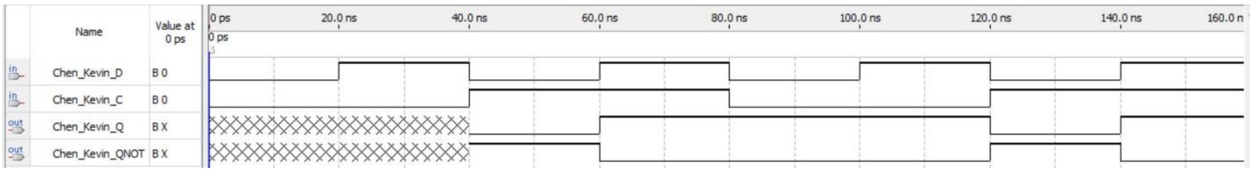


Figure 14: Waveform Simulation of Block Diagram/Schematic File of D-Latch

(Chen_Kevin_D-LatchWaveform.VWF)

SRAM:

There are three functionality/behavior of the SRAM. One functionality/behavior serves to turn the chip on or off. The second functionality/behavior serves to write the "datain" input into the "address" input. The third functionality/behavior serves to display the output that was inputted.

A waveform was created using the block diagram/schematic file for SRAM (Figure 15 and 16) and it was given inputs to test all possible combinations. The output that it generates shows that the chip is turned on when "CS" is turned on; when "WE" is turned on, it allows to write; when "OE" is on, it displays the output.

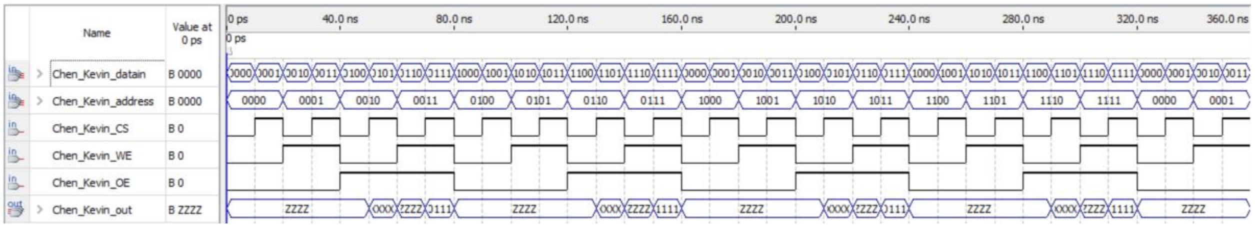


Figure 15: Waveform Simulation of Block Diagram/Schematic File of 16 x 4 SRAM

(Chen_Kevin_16x4SRAMWaveform.VWF)

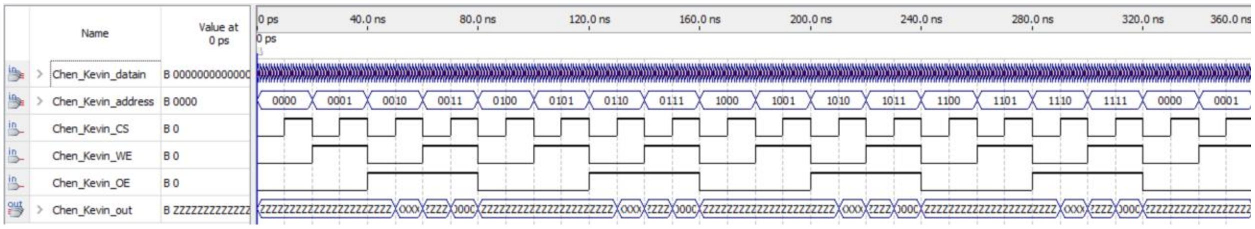


Figure 16: Waveform Simulation of Block Diagram/Schematic File of 16 x32 SRAM

(Chen_Kevin_16x32SRAMWaveform.VWF)

Demonstration:

16 x 4 SRAM:

In address 0000, I inputted the binary code 0011 and had it display on the DE2 board. After that, I swapped into address 0001, and inputted the binary code 1001 and had it display on the DE2 board.

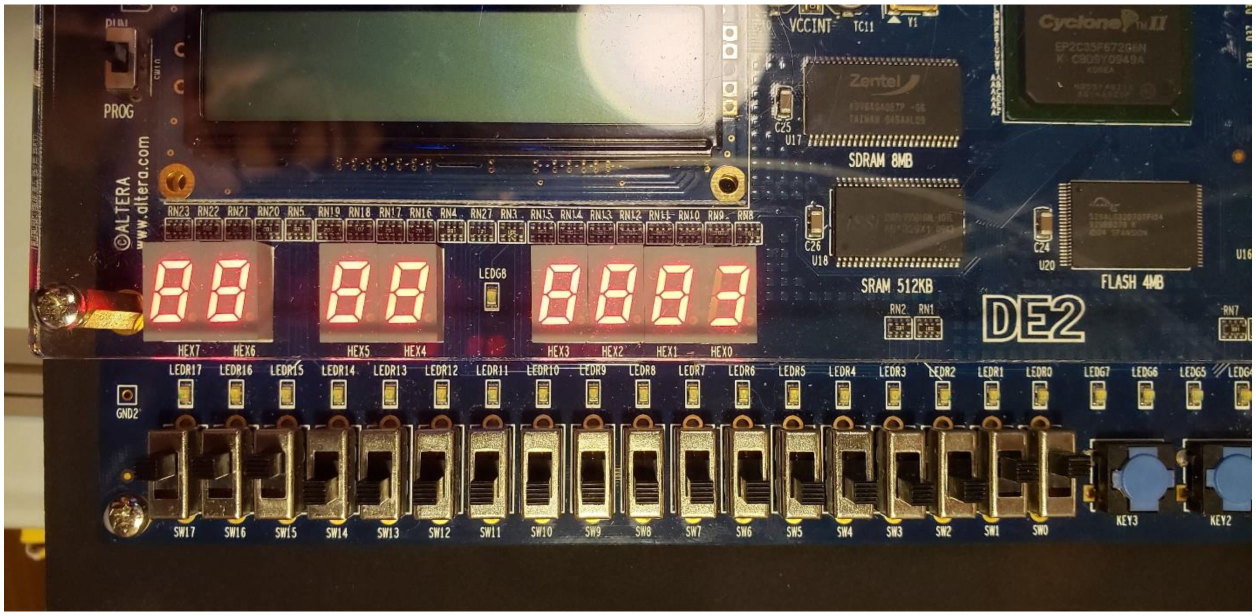


Figure 17: Input 3 Into Address 0 on DE2 Board

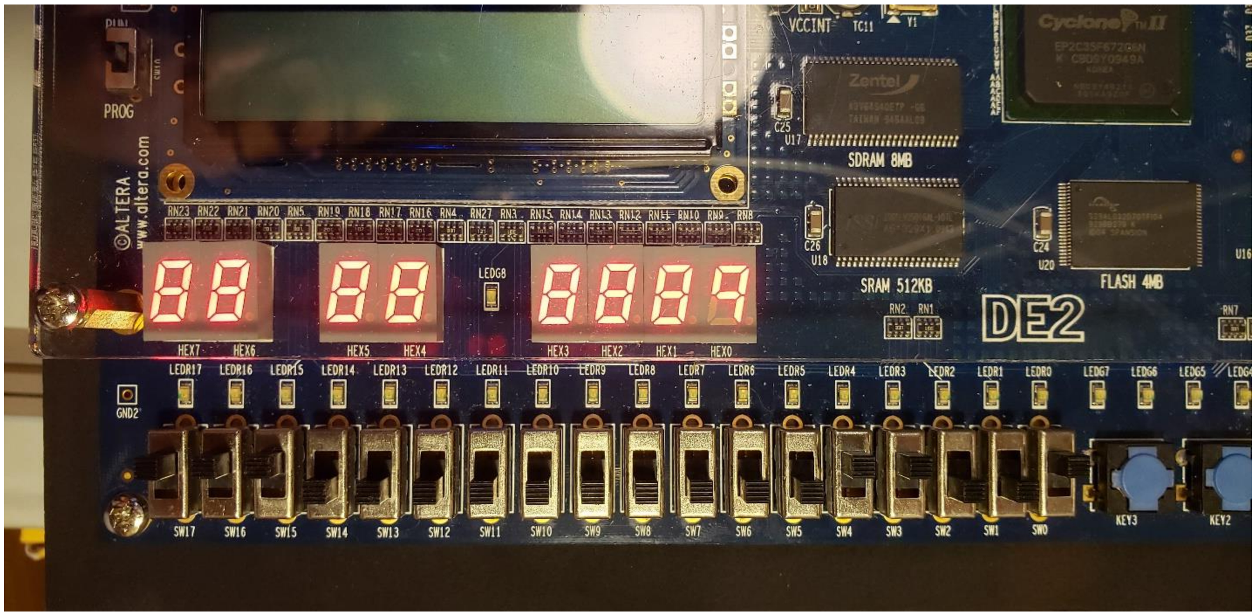


Figure 18: Input 9 Into Address 1 on DE2 Board

16 x 32 SRAM:

In address 0000, I inputted the binary code 1000 0111 0110 0101 0100 0011 0010 0001 and had it display on the DE2 board. After that, I swapped into address 0001, and inputted the binary code 1010 1011 1100 1101 1110 1111 0000 0001 and had it display on the DE2 board.

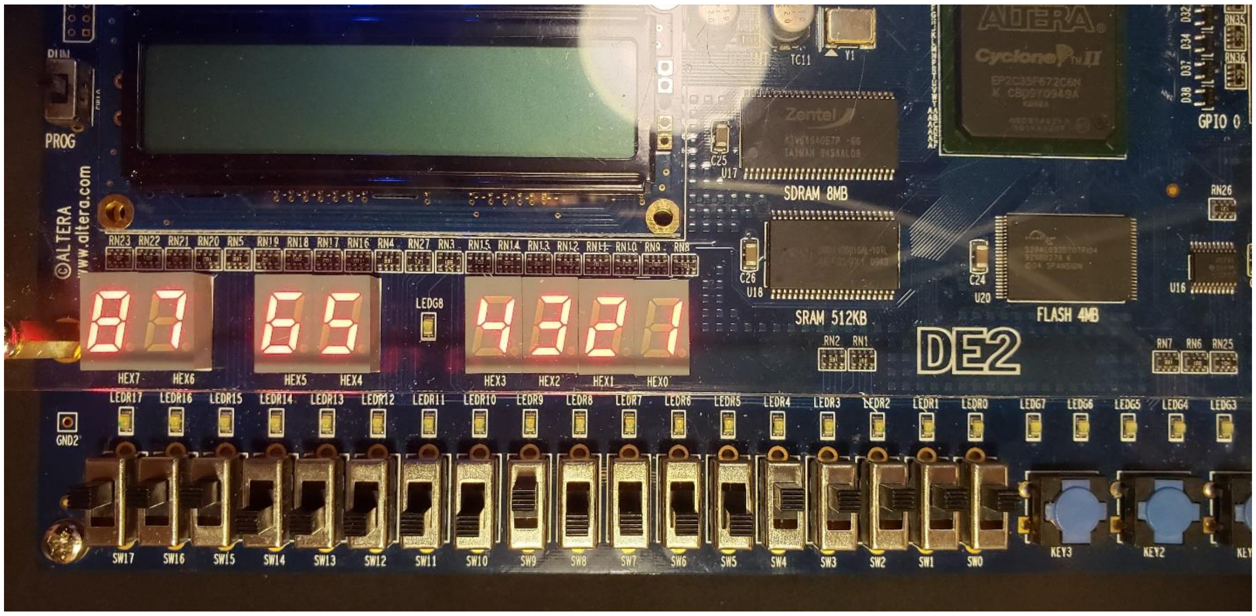


Figure 19: Input 8 7 6 5 4 3 2 1 Into Address 0 on DE2 Board

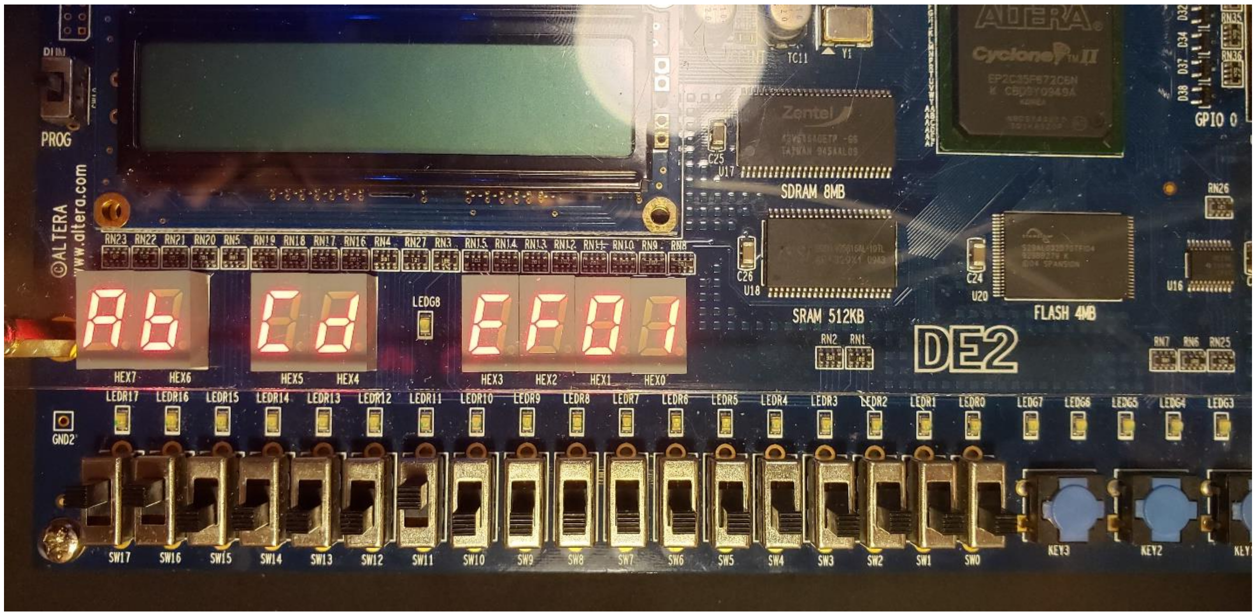


Figure 20: Input A b C d E F 0 1 Into Address 1 on DE2 Board

Conclusion:

This lab taught me how create a static random-access memory chip using D-Latches. It also taught me how to allow the output of the SRAM to be connected to a 4-to-7 decoder that outputs to the 7-segment-display. I was able to view how a SR-Latch could become into a Control SR-Latch, then into a D-Latch, then into a Master-Slave D Flip-Flop, then into a SRAM.

1. What is the difference between a latch and a flip-flop?

   The difference between a latch and a flip-flop is that a latch is level triggered so the output can change anytime based on the inputs, and a flip-flop is edge triggered so the output can only change when the clock changes.

2. Explain the behavior of a SR-Latch

   The behavior of a SR-Latch is either give the same outputs as the previous output or to set an output to 1 or reset an output to 0.
3. Explain how the gated (control) SR-Latch is a modified SR-Latch

   The gated (control) SR-Latch is a modified SR-Latch because it performs the same action as a SR-Latch except with an additional control input. When the control input is not active, then nothing changes, but when it is active, it acts the same as a SR-Latch.
4. Explain how the D-Latch is a modified gated SR-Latch

   The D-Latch is a modified gated SR-Latch by how it uses the same control input and combines the S and R input into one input, called “D”. The “D” input acts the same way as a SR-Latch except now you can’t have both set and reset; it’s either or.

5. Explain how to build a Master-Slave flip-flop.

   A Master-Slave Flip-Flop is built by connecting two latches in series. Figure 4 shows how it's built.

6. Explain how to build SRAM cells from flip-flops.

   A SRAM cell is built by using a Master-Slave D Flip-Flop and connecting it to a few logic gates and a tristate logic buffer. Figure 6 shows how it's built.

7. Why do we want to avoid the scenario when S=1 and R=1 in an SR-Latch? Could this scenario happen in a D-Latch? Why or why not?

   We want to avoid the scenario when S=1 and R=1 in an SR-Latch because we can't have a device both set and reset at the same time. This scenario can't happen in a D-Latch because a D-Latch combines the set and reset inputs into one input, which makes it so it can either be set or reset.

8. What is the difference between edge-triggered and level triggered devices? Is the flip-flop used for the SRAM cell in this lab level or edge triggered?

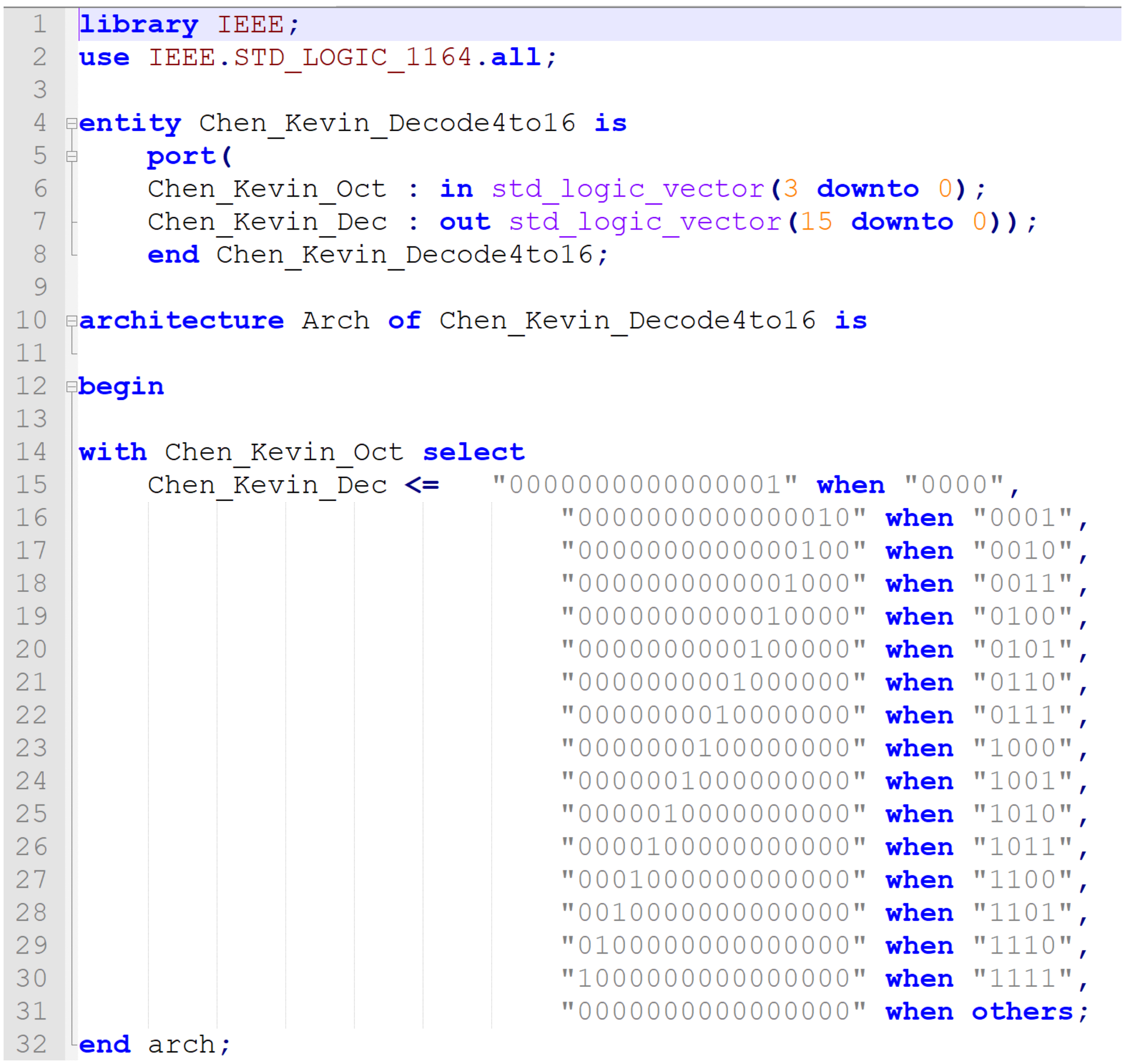
   The difference between edge-triggered and level triggered devices is that edge-triggered devices depend on the clock signal for a change in output, while level triggered devices depend on a change in inputs for a change in output. The flip-flop used for the SRAM cell in this lab is edge triggered.

9. Can you explain how an SRAM works?

   An SRAM works by first enabling the "Chip_Select", which turns the SRAM on. Then by giving it a specific input and a specific address, where the input will be stored in, the input would be written into the address when the "Write_Enable" is enabled. By enabling the "Output_Enable", it's able to display what was written into the address.

<u>Appendix:</u>

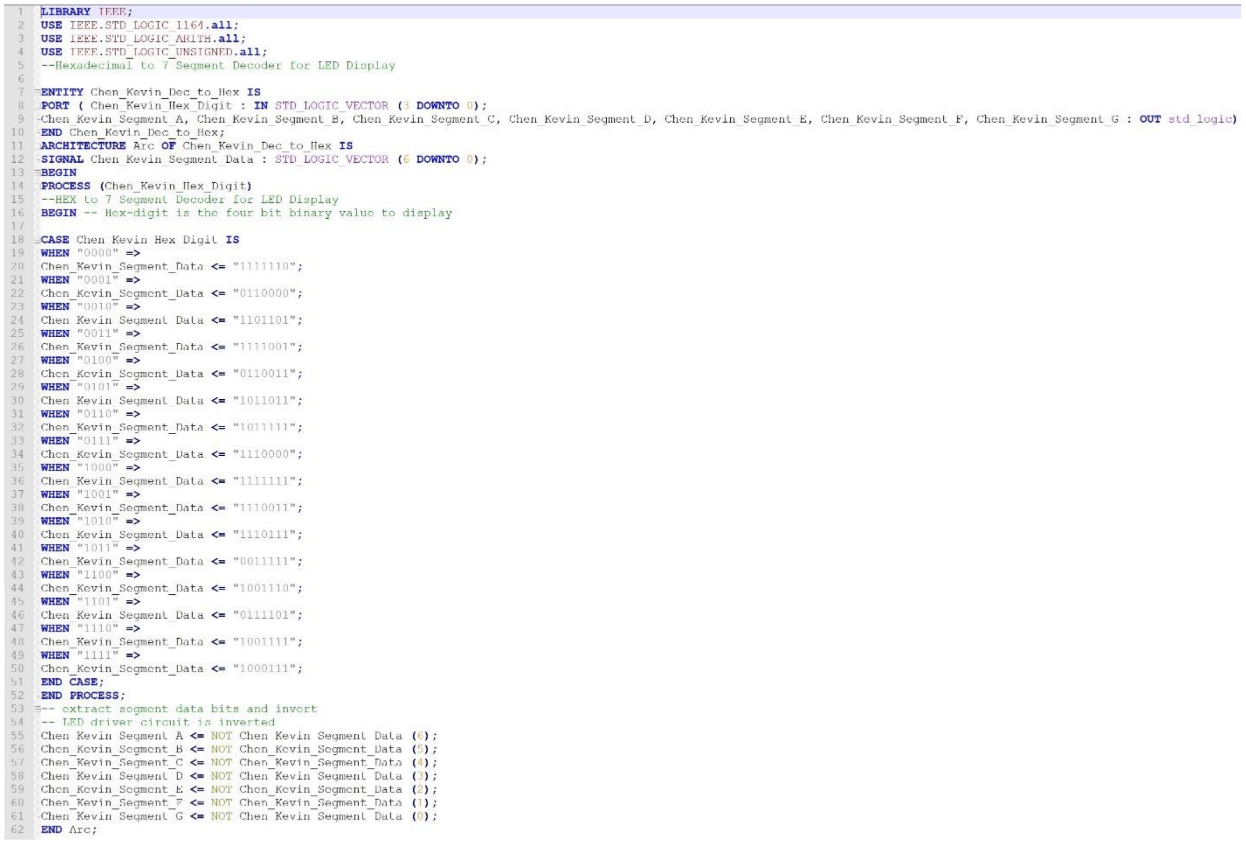
<u>4-to-16 Decoder:</u>



```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity Chen_Kevin_Decode4to16 is
    port(
    Chen_Kevin_Oct : in std_logic_vector(3 downto 0);
    Chen_Kevin_Dec : out std_logic_vector(15 downto 0));
    end Chen_Kevin_Decode4to16;

architecture Arch of Chen_Kevin_Decode4to16 is

begin

with Chen_Kevin_Oct select
    Chen_Kevin_Dec <=    "0000000000000001" when "0000",
                            "0000000000000010" when "0001",
                            "0000000000000100" when "0010",
                            "0000000000001000" when "0011",
                            "0000000000010000" when "0100",
                            "0000000000100000" when "0101",
                            "0000000001000000" when "0110",
                            "0000000010000000" when "0111",
                            "0000000100000000" when "1000",
                            "0000001000000000" when "1001",
                            "0000010000000000" when "1010",
                            "0000100000000000" when "1011",
                            "0001000000000000" when "1100",
                            "0010000000000000" when "1101",
                            "0100000000000000" when "1110",
                            "1000000000000000" when "1111",
                            "0000000000000000" when others;
end arch;
```

Figure 21: VHDL Code for 4-to-16 Decoder (Chen_Kevin_Decode4to16.VHD)

Decimal to Hexadecimal Translation:



```vhdl
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.all;
USE IEEE.STD_LOGIC_ARITH.all;
USE IEEE.STD_LOGIC_UNSIGNED.all;
--Hexadecimal to 7 Segment Decoder for LED Display

ENTITY Chen_Kevin_Dec_to_Hex IS
PORT ( Chen_Kevin_Hex_Digit : IN STD_LOGIC_VECTOR (3 DOWNTO 0);
Chen_Kevin_Segment_A, Chen_Kevin_Segment_B, Chen_Kevin_Segment_C, Chen_Kevin_Segment_D, Chen_Kevin_Segment_E, Chen_Kevin_Segment_F, Chen_Kevin_Segment_G : OUT std_logic)
END Chen_Kevin_Dec_to_Hex;
ARCHITECTURE Arc OF Chen_Kevin_Dec_to_Hex IS
SIGNAL Chen_Kevin_Segment_Data : STD_LOGIC_VECTOR (6 DOWNTO 0);
BEGIN
PROCESS (Chen_Kevin_Hex_Digit)
--HEX to 7 Segment Decoder for LED Display
BEGIN -- Hex-digit is the four bit binary value to display

CASE Chen_Kevin_Hex_Digit IS
WHEN "0000" =>
Chen_Kevin_Segment_Data <= "1111110";
WHEN "0001" =>
Chen_Kevin_Segment_Data <= "0110000";
WHEN "0010" =>
Chen_Kevin_Segment_Data <= "1101101";
WHEN "0011" =>
Chen_Kevin_Segment_Data <= "1111001";
WHEN "0100" =>
Chen_Kevin_Segment_Data <= "0110011";
WHEN "0101" =>
Chen_Kevin_Segment_Data <= "1011011";
WHEN "0110" =>
Chen_Kevin_Segment_Data <= "1011111";
WHEN "0111" =>
Chen_Kevin_Segment_Data <= "1110000";
WHEN "1000" =>
Chen_Kevin_Segment_Data <= "1111111";
WHEN "1001" =>
Chen_Kevin_Segment_Data <= "1110011";
WHEN "1010" =>
Chen_Kevin_Segment_Data <= "1110111";
WHEN "1011" =>
Chen_Kevin_Segment_Data <= "0011111";
WHEN "1100" =>
Chen_Kevin_Segment_Data <= "1001110";
WHEN "1101" =>
Chen_Kevin_Segment_Data <= "0111101";
WHEN "1110" =>
Chen_Kevin_Segment_Data <= "1001111";
WHEN "1111" =>
Chen_Kevin_Segment_Data <= "1000111";
END CASE;
END PROCESS;
-- extract segment data bits and invert
-- LED driver circuit is inverted
Chen_Kevin_Segment_A <= NOT Chen_Kevin_Segment_Data (6);
Chen_Kevin_Segment_B <= NOT Chen_Kevin_Segment_Data (5);
Chen_Kevin_Segment_C <= NOT Chen_Kevin_Segment_Data (4);
Chen_Kevin_Segment_D <= NOT Chen_Kevin_Segment_Data (3);
Chen_Kevin_Segment_E <= NOT Chen_Kevin_Segment_Data (2);
Chen_Kevin_Segment_F <= NOT Chen_Kevin_Segment_Data (1);
Chen_Kevin_Segment_G <= NOT Chen_Kevin_Segment_Data (0);
END Arc;
```

Figure 22: VHDL Code for Decimal to Hexadecimal Translation

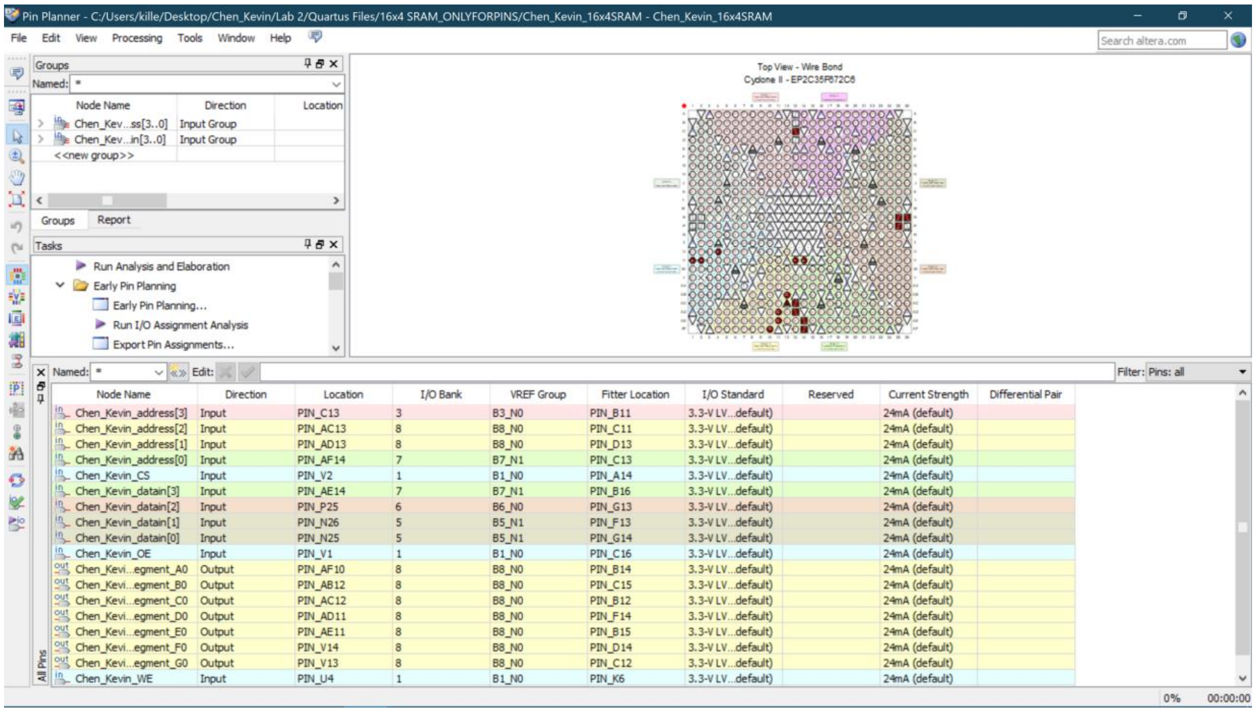
(Chen_Kevin_Dec_to_Hex.VHD)

<u>16 x 4 SRAM Pin Assignments:</u>

Variable Name	**Variable Type**	**Signal Name**	**FPGA Pin No.**
Chen_Kevin_datain[0]	Input	SW[0]	PIN_N25
Chen_Kevin_datain[1]	Input	SW[1]	PIN_N26
Chen_Kevin_datain[2]	Input	SW[2]	PIN_P25
Chen_Kevin_datain[3]	Input	SW[3]	PIN_AE14
Chen_Kevin_address[0]	Input	SW[4]	PIN_AF14
Chen_Kevin_address[1]	Input	SW[5]	PIN_AD13
Chen_Kevin_address[2]	Input	SW[6]	PIN_AC13
Chen_Kevin_address[3]	Input	SW[7]	PIN_C13
Chen_Kevin_WE	Input	SW[15]	PIN_U4
Chen_Kevin_OE	Input	SW[16]	PIN_V1
Chen_Kevin_CS	Input	SW[17]	PIN_V2
Chen_Kevin_Segment_A0	Output	HEX0[0]	PIN_AF10
Chen_Kevin_Segment_B0	Output	HEX0[1]	PIN_AB12
Chen_Kevin_Segment_C0	Output	HEX0[2]	PIN_AC12
Chen_Kevin_Segment_D0	Output	HEX0[3]	PIN_AD11
Chen_Kevin_Segment_E0	Output	HEX0[4]	PIN_AE11
Chen_Kevin_Segment_F0	Output	HEX0[5]	PIN_V14
Chen_Kevin_Segment_G0	Output	HEX0[6]	PIN_V13

Table 6: Pin Assignments for 16 x 4 SRAM

16 x 4 SRAM Pin Planner:



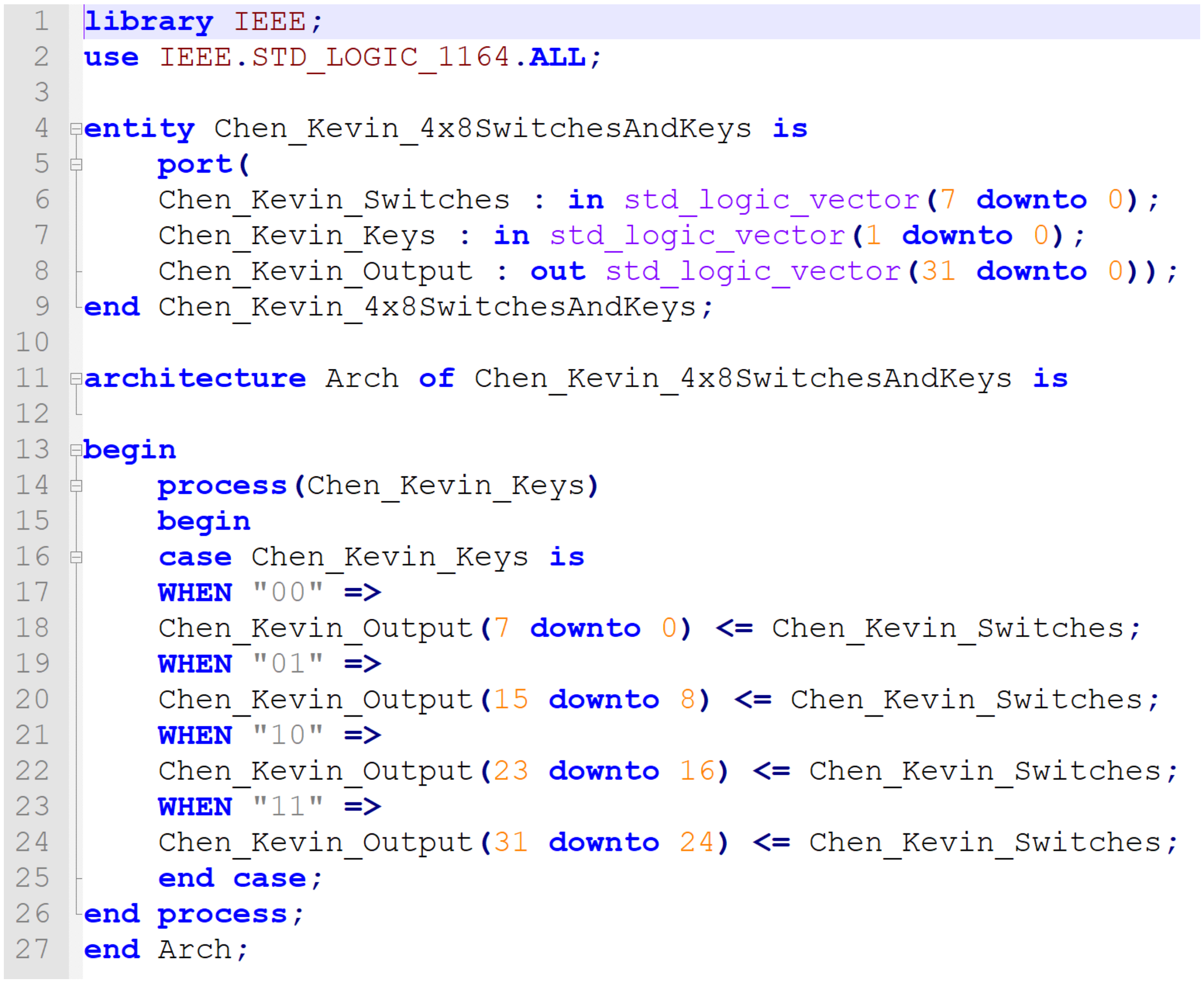
Pin Planner - C:/Users/kille/Desktop/Chen_Kevin/Lab 2/Quartus Files/16x4 SRAM_ONLYFORPINS/Chen_Kevin_16x4SRAM - Chen_Kevin_16x4SRAM

Node Name	Direction	Location
Chen_Kev...ss[3..0]	Input Group	
Chen_Kev...in[3..0]	Input Group	
<<new group>>		

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Differential Pair
Chen_Kevin_address[3]	Input	PIN_C13	3	B3_N0	PIN_B11	3.3-V LV...default)		24mA (default)	
Chen_Kevin_address[2]	Input	PIN_AC13	8	B8_N0	PIN_C11	3.3-V LV...default)		24mA (default)	
Chen_Kevin_address[1]	Input	PIN_AD13	8	B8_N0	PIN_D13	3.3-V LV...default)		24mA (default)	
Chen_Kevin_address[0]	Input	PIN_AF14	7	B7_N1	PIN_C13	3.3-V LV...default)		24mA (default)	
Chen_Kevin_CS	Input	PIN_V2	1	B1_N0	PIN_A14	3.3-V LV...default)		24mA (default)	
Chen_Kevin_datain[3]	Input	PIN_AE14	7	B7_N1	PIN_B16	3.3-V LV...default)		24mA (default)	
Chen_Kevin_datain[2]	Input	PIN_P25	6	B6_N0	PIN_G13	3.3-V LV...default)		24mA (default)	
Chen_Kevin_datain[1]	Input	PIN_N26	5	B5_N1	PIN_F13	3.3-V LV...default)		24mA (default)	
Chen_Kevin_datain[0]	Input	PIN_N25	5	B5_N1	PIN_G14	3.3-V LV...default)		24mA (default)	
Chen_Kevin_OE	Input	PIN_V1	1	B1_N0	PIN_C16	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_A0	Output	PIN_AF10	8	B8_N0	PIN_B14	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_B0	Output	PIN_AB12	8	B8_N0	PIN_C15	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_C0	Output	PIN_AC12	8	B8_N0	PIN_B12	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_D0	Output	PIN_AD11	8	B8_N0	PIN_F14	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_E0	Output	PIN_AE11	8	B8_N0	PIN_B15	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_F0	Output	PIN_V14	8	B8_N0	PIN_D14	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_G0	Output	PIN_V13	8	B8_N0	PIN_C12	3.3-V LV...default)		24mA (default)	
Chen_Kevin_WE	Input	PIN_U4	1	B1_N0	PIN_K6	3.3-V LV...default)		24mA (default)	

Figure 23: Pin Planner of Block Diagram/Schematic File of 16 x 4 SRAM

4 x 8 Switches and Keys:



```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity Chen_Kevin_4x8SwitchesAndKeys is
    port(
    Chen_Kevin_Switches : in std_logic_vector(7 downto 0);
    Chen_Kevin_Keys : in std_logic_vector(1 downto 0);
    Chen_Kevin_Output : out std_logic_vector(31 downto 0));
end Chen_Kevin_4x8SwitchesAndKeys;

architecture Arch of Chen_Kevin_4x8SwitchesAndKeys is

begin
    process(Chen_Kevin_Keys)
    begin
    case Chen_Kevin_Keys is
    WHEN "00" =>
    Chen_Kevin_Output(7 downto 0) <= Chen_Kevin_Switches;
    WHEN "01" =>
    Chen_Kevin_Output(15 downto 8) <= Chen_Kevin_Switches;
    WHEN "10" =>
    Chen_Kevin_Output(23 downto 16) <= Chen_Kevin_Switches;
    WHEN "11" =>
    Chen_Kevin_Output(31 downto 24) <= Chen_Kevin_Switches;
    end case;
end process;
end Arch;
```

Figure 24: VHDL Code for 4 x 8 Switches and Keys (Chen_Kevin_4x8SwitchesAndKeys.VHD)

<u>16 x 32 SRAM Pin Assignments:</u>

Variable Name	**Variable Type**	**Signal Name**	**FPGA Pin No.**
Chen_Kevin_Keys[0]	Input	SW[0]	PIN_N25
Chen_Kevin_Keys[1]	Input	SW[1]	PIN_N26
Chen_Kevin_Switches[0]	Input	SW[2]	PIN_P25
Chen_Kevin_Switches[1]	Input	SW[3]	PIN_AE14
Chen_Kevin_Switches[2]	Input	SW[4]	PIN_AF14
Chen_Kevin_Switches[3]	Input	SW[5]	PIN_AD13
Chen_Kevin_Switches[4]	Input	SW[6]	PIN_AC13
Chen_Kevin_Switches[5]	Input	SW[7]	PIN_C13
Chen_Kevin_Switches[6]	Input	SW[8]	PIN_B13
Chen_Kevin_Switches[7]	Input	SW[9]	PIN_A13
Chen_Kevin_address[0]	Input	SW[10]	PIN_N1
Chen_Kevin_address[1]	Input	SW[11]	PIN_P1
Chen_Kevin_address[2]	Input	SW[12]	PIN_P2
Chen_Kevin_address[3]	Input	SW[13]	PIN_T7
Chen_Kevin_WE	Input	SW[15]	PIN_U4
Chen_Kevin_OE	Input	SW[16]	PIN_V1
Chen_Kevin_CS	Input	SW[17]	PIN_V2

Table 7a: Pin Assignments for 16 x 32 SRAM

Variable Name	**Variable Type**	**Signal Name**	**FPGA Pin No.**
Chen_Kevin_Segment_A0	Output	HEX0[0]	PIN_AF10
Chen_Kevin_Segment_B0	Output	HEX0[1]	PIN_AB12
Chen_Kevin_Segment_C0	Output	HEX0[2]	PIN_AC12
Chen_Kevin_Segment_D0	Output	HEX0[3]	PIN_AD11
Chen_Kevin_Segment_E0	Output	HEX0[4]	PIN_AE11
Chen_Kevin_Segment_F0	Output	HEX0[5]	PIN_V14
Chen_Kevin_Segment_G0	Output	HEX0[6]	PIN_V13
Chen_Kevin_Segment_A1	Output	HEX1[0]	PIN_V20
Chen_Kevin_Segment_B1	Output	HEX1[1]	PIN_V21
Chen_Kevin_Segment_C1	Output	HEX1[2]	PIN_W21
Chen_Kevin_Segment_D1	Output	HEX1[3]	PIN_Y22
Chen_Kevin_Segment_E1	Output	HEX1[4]	PIN_AA24
Chen_Kevin_Segment_F1	Output	HEX1[5]	PIN_AA23
Chen_Kevin_Segment_G1	Output	HEX1[6]	PIN_AB24
Chen_Kevin_Segment_A2	Output	HEX2[0]	PIN_AB23
Chen_Kevin_Segment_B2	Output	HEX2[1]	PIN_V22
Chen_Kevin_Segment_C2	Output	HEX2[2]	PIN_AC25
Chen_Kevin_Segment_D2	Output	HEX2[3]	PIN_AC26
Chen_Kevin_Segment_E2	Output	HEX2[4]	PIN_AB26
Chen_Kevin_Segment_F2	Output	HEX2[5]	PIN_AB25
Chen_Kevin_Segment_G2	Output	HEX2[6]	PIN_Y24

Table 7b: Pin Assignments for 16 x 32 SRAM

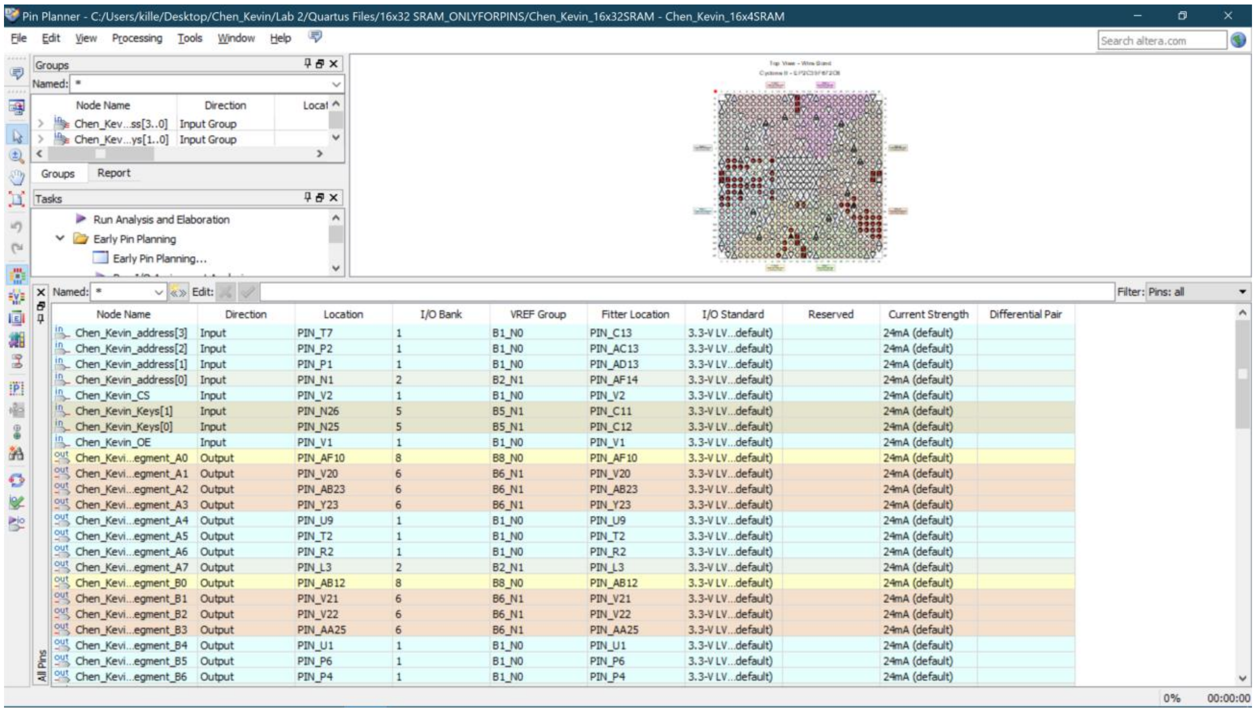
Variable Name	**Variable Type**	**Signal Name**	**FPGA Pin No.**
Chen_Kevin_Segment_A3	Output	HEX3[0]	PIN_Y23
Chen_Kevin_Segment_B3	Output	HEX3[1]	PIN_AA25
Chen_Kevin_Segment_C3	Output	HEX3[2]	PIN_AA26
Chen_Kevin_Segment_D3	Output	HEX3[3]	PIN_Y26
Chen_Kevin_Segment_E3	Output	HEX3[4]	PIN_Y25
Chen_Kevin_Segment_F3	Output	HEX3[5]	PIN_U22
Chen_Kevin_Segment_G3	Output	HEX3[6]	PIN_W24
Chen_Kevin_Segment_A4	Output	HEX4[0]	PIN_U9
Chen_Kevin_Segment_B4	Output	HEX4[1]	PIN_U1
Chen_Kevin_Segment_C4	Output	HEX4[2]	PIN_U2
Chen_Kevin_Segment_D4	Output	HEX4[3]	PIN_T4
Chen_Kevin_Segment_E4	Output	HEX4[4]	PIN_R7
Chen_Kevin_Segment_F4	Output	HEX4[5]	PIN_R6
Chen_Kevin_Segment_G4	Output	HEX4[6]	PIN_T3
Chen_Kevin_Segment_A5	Output	HEX5[0]	PIN_T2
Chen_Kevin_Segment_B5	Output	HEX5[1]	PIN_P6
Chen_Kevin_Segment_C5	Output	HEX5[2]	PIN_P7
Chen_Kevin_Segment_D5	Output	HEX5[3]	PIN_T9
Chen_Kevin_Segment_E5	Output	HEX5[4]	PIN_R5
Chen_Kevin_Segment_F5	Output	HEX5[5]	PIN_R4
Chen_Kevin_Segment_G5	Output	HEX5[6]	PIN_R3

Table 7c: Pin Assignments for 16 x 32 SRAM

Variable Name	**Variable Type**	**Signal Name**	**FPGA Pin No.**
Chen_Kevin_Segment_A6	Output	HEX6[0]	PIN_R2
Chen_Kevin_Segment_B6	Output	HEX6[1]	PIN_P4
Chen_Kevin_Segment_C6	Output	HEX6[2]	PIN_P3
Chen_Kevin_Segment_D6	Output	HEX6[3]	PIN_M2
Chen_Kevin_Segment_E6	Output	HEX6[4]	PIN_M3
Chen_Kevin_Segment_F6	Output	HEX6[5]	PIN_M5
Chen_Kevin_Segment_G6	Output	HEX6[6]	PIN_M4
Chen_Kevin_Segment_A7	Output	HEX7[0]	PIN_L3
Chen_Kevin_Segment_B7	Output	HEX7[1]	PIN_L2
Chen_Kevin_Segment_C7	Output	HEX7[2]	PIN_L9
Chen_Kevin_Segment_D7	Output	HEX7[3]	PIN_L6
Chen_Kevin_Segment_E7	Output	HEX7[4]	PIN_L7
Chen_Kevin_Segment_F7	Output	HEX7[5]	PIN_P9
Chen_Kevin_Segment_G7	Output	HEX7[6]	PIN_N9

Table 7d: Pin Assignments for 16 x 32 SRAM

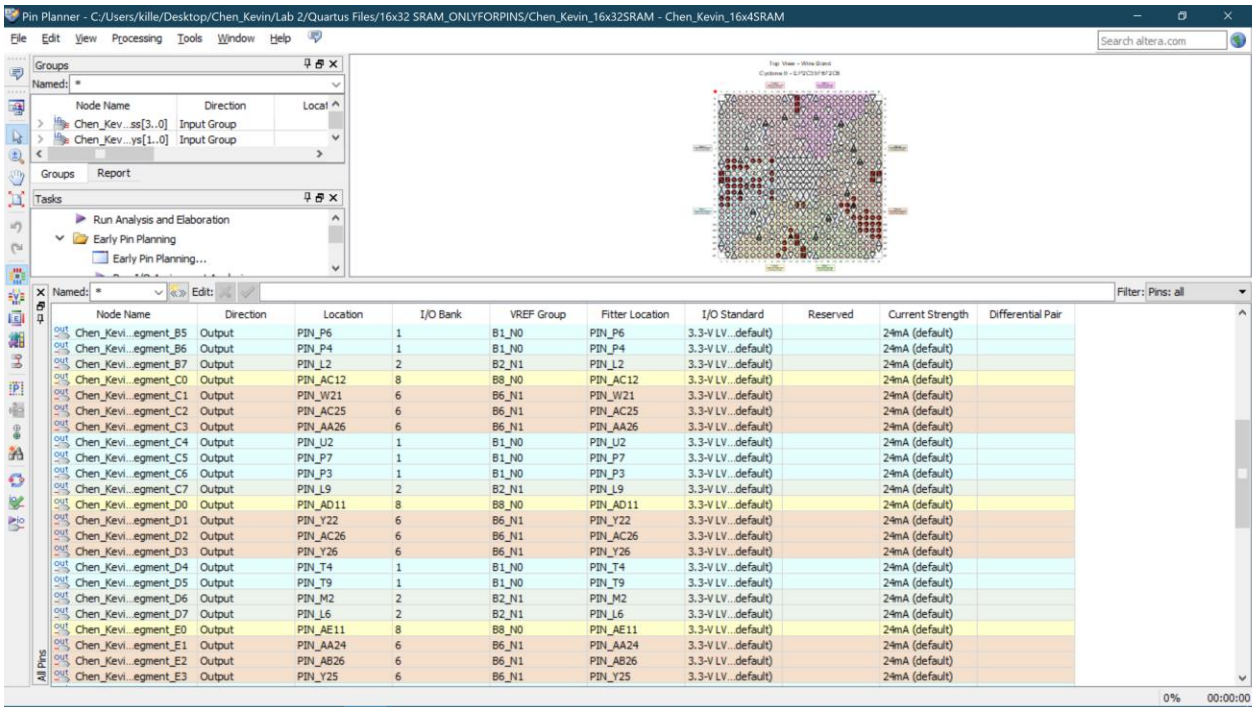
16 x 32 SRAM Pin Planner:



Node Name	Direction	Locat
Chen_Kev...ss[3..0]	Input Group	
Chen_Kev...ys[1..0]	Input Group	

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Differential Pair
Chen_Kevin_address[3]	Input	PIN_T7	1	B1_N0	PIN_C13	3.3-V LV...default)		24mA (default)	
Chen_Kevin_address[2]	Input	PIN_P2	1	B1_N0	PIN_AC13	3.3-V LV...default)		24mA (default)	
Chen_Kevin_address[1]	Input	PIN_P1	1	B1_N0	PIN_AD13	3.3-V LV...default)		24mA (default)	
Chen_Kevin_address[0]	Input	PIN_N1	2	B2_N1	PIN_AF14	3.3-V LV...default)		24mA (default)	
Chen_Kevin_CS	Input	PIN_V2	1	B1_N0	PIN_V2	3.3-V LV...default)		24mA (default)	
Chen_Kevin_Keys[1]	Input	PIN_N26	5	B5_N1	PIN_C11	3.3-V LV...default)		24mA (default)	
Chen_Kevin_Keys[0]	Input	PIN_N25	5	B5_N1	PIN_C12	3.3-V LV...default)		24mA (default)	
Chen_Kevin_OE	Input	PIN_V1	1	B1_N0	PIN_V1	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_A0	Output	PIN_AF10	8	B8_N0	PIN_AF10	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_A1	Output	PIN_V20	6	B6_N1	PIN_V20	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_A2	Output	PIN_AB23	6	B6_N1	PIN_AB23	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_A3	Output	PIN_Y23	6	B6_N1	PIN_Y23	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_A4	Output	PIN_U9	1	B1_N0	PIN_U9	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_A5	Output	PIN_T2	1	B1_N0	PIN_T2	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_A6	Output	PIN_R2	1	B1_N0	PIN_R2	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_A7	Output	PIN_L3	2	B2_N1	PIN_L3	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_B0	Output	PIN_AB12	8	B8_N0	PIN_AB12	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_B1	Output	PIN_V21	6	B6_N1	PIN_V21	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_B2	Output	PIN_V22	6	B6_N1	PIN_V22	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_B3	Output	PIN_AA25	6	B6_N1	PIN_AA25	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_B4	Output	PIN_U1	1	B1_N0	PIN_U1	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_B5	Output	PIN_P6	1	B1_N0	PIN_P6	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_B6	Output	PIN_P4	1	B1_N0	PIN_P4	3.3-V LV...default)		24mA (default)	

Figure 25a: Pin Planner of Block Diagram/Schematic File of 16 x 32 SRAM

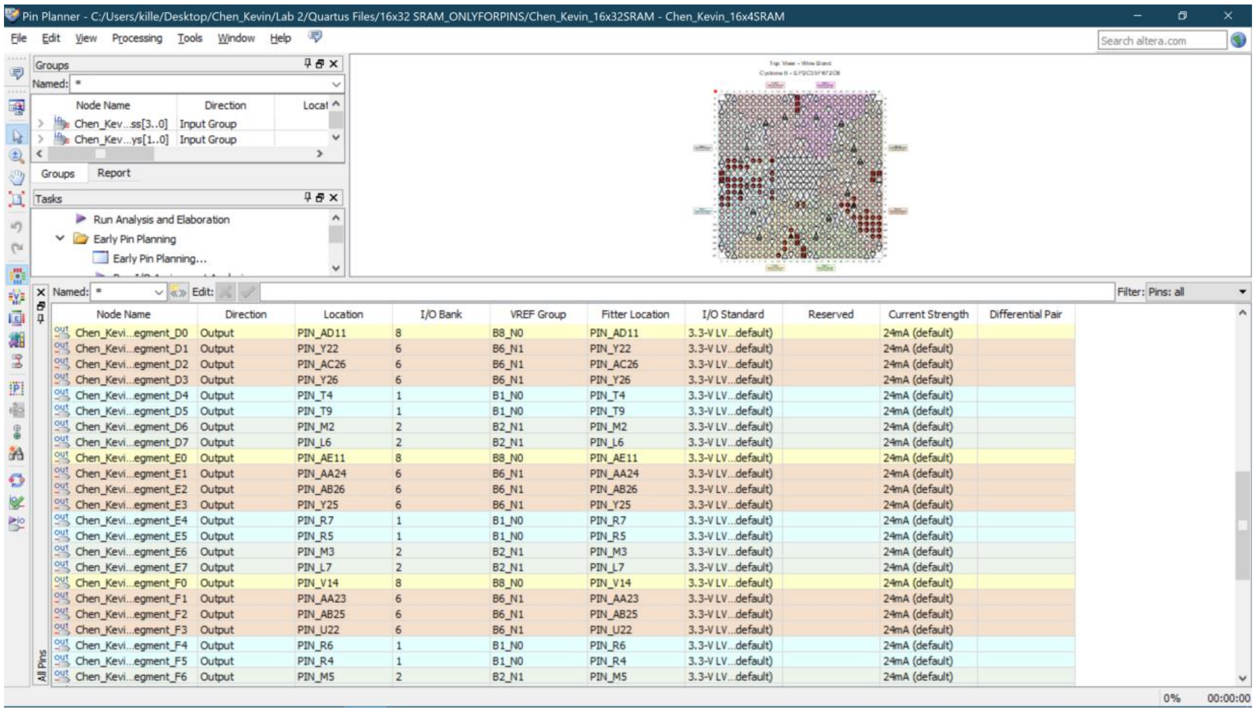


Pin Planner - C:/Users/kille/Desktop/Chen_Kevin/Lab 2/Quartus Files/16x32 SRAM_ONLYFORPINS/Chen_Kevin_16x32SRAM - Chen_Kevin_16x4SRAM

Node Name	Direction	Locat
Chen_Kev...ss[3..0]	Input Group	
Chen_Kev...ys[1..0]	Input Group	

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Differential Pair
Chen_Kevi...egment_B5	Output	PIN_P6	1	B1_N0	PIN_P6	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_B6	Output	PIN_P4	1	B1_N0	PIN_P4	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_B7	Output	PIN_L2	2	B2_N1	PIN_L2	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_C0	Output	PIN_AC12	8	B8_N0	PIN_AC12	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_C1	Output	PIN_W21	6	B6_N1	PIN_W21	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_C2	Output	PIN_AC25	6	B6_N1	PIN_AC25	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_C3	Output	PIN_AA26	6	B6_N1	PIN_AA26	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_C4	Output	PIN_U2	1	B1_N0	PIN_U2	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_C5	Output	PIN_P7	1	B1_N0	PIN_P7	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_C6	Output	PIN_P3	1	B1_N0	PIN_P3	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_C7	Output	PIN_L9	2	B2_N1	PIN_L9	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_D0	Output	PIN_AD11	8	B8_N0	PIN_AD11	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_D1	Output	PIN_Y22	6	B6_N1	PIN_Y22	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_D2	Output	PIN_AC26	6	B6_N1	PIN_AC26	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_D3	Output	PIN_Y26	6	B6_N1	PIN_Y26	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_D4	Output	PIN_T4	1	B1_N0	PIN_T4	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_D5	Output	PIN_T9	1	B1_N0	PIN_T9	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_D6	Output	PIN_M2	2	B2_N1	PIN_M2	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_D7	Output	PIN_L6	2	B2_N1	PIN_L6	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_E0	Output	PIN_AE11	8	B8_N0	PIN_AE11	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_E1	Output	PIN_AA24	6	B6_N1	PIN_AA24	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_E2	Output	PIN_AB26	6	B6_N1	PIN_AB26	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_E3	Output	PIN_Y25	6	B6_N1	PIN_Y25	3.3-V LV...default)		24mA (default)	

Figure 25b: Pin Planner of Block Diagram/Schematic File of 16 x 32 SRAM

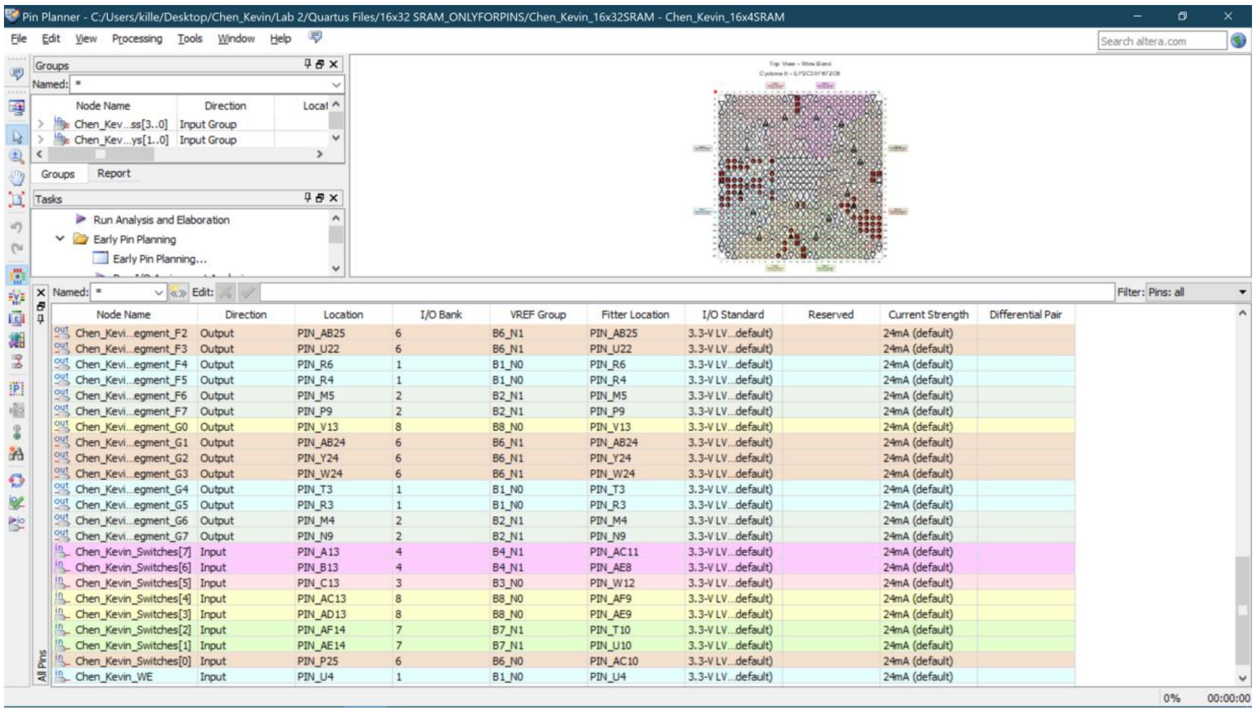


Pin Planner - C:/Users/kille/Desktop/Chen_Kevin/Lab 2/Quartus Files/16x32 SRAM_ONLYFORPINS/Chen_Kevin_16x32SRAM - Chen_Kevin_16x4SRAM

Node Name	Direction	Locat
Chen_Kev...ss[3..0]	Input Group	
Chen_Kev...ys[1..0]	Input Group	

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Differential Pair
Chen_Kevi...egment_D0	Output	PIN_AD11	8	B8_N0	PIN_AD11	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_D1	Output	PIN_Y22	6	B6_N1	PIN_Y22	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_D2	Output	PIN_AC26	6	B6_N1	PIN_AC26	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_D3	Output	PIN_Y26	6	B6_N1	PIN_Y26	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_D4	Output	PIN_T4	1	B1_N0	PIN_T4	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_D5	Output	PIN_T9	1	B1_N0	PIN_T9	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_D6	Output	PIN_M2	2	B2_N1	PIN_M2	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_D7	Output	PIN_L6	2	B2_N1	PIN_L6	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_E0	Output	PIN_AE11	8	B8_N0	PIN_AE11	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_E1	Output	PIN_AA24	6	B6_N1	PIN_AA24	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_E2	Output	PIN_AB26	6	B6_N1	PIN_AB26	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_E3	Output	PIN_Y25	6	B6_N1	PIN_Y25	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_E4	Output	PIN_R7	1	B1_N0	PIN_R7	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_E5	Output	PIN_R5	1	B1_N0	PIN_R5	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_E6	Output	PIN_M3	2	B2_N1	PIN_M3	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_E7	Output	PIN_L7	2	B2_N1	PIN_L7	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_F0	Output	PIN_V14	8	B8_N0	PIN_V14	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_F1	Output	PIN_AA23	6	B6_N1	PIN_AA23	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_F2	Output	PIN_AB25	6	B6_N1	PIN_AB25	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_F3	Output	PIN_U22	6	B6_N1	PIN_U22	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_F4	Output	PIN_R6	1	B1_N0	PIN_R6	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_F5	Output	PIN_R4	1	B1_N0	PIN_R4	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_F6	Output	PIN_M5	2	B2_N1	PIN_M5	3.3-V LV...default)		24mA (default)	

Figure 25c: Pin Planner of Block Diagram/Schematic File of 16 x 32 SRAM



Pin Planner - C:/Users/kille/Desktop/Chen_Kevin/Lab 2/Quartus Files/16x32 SRAM_ONLYFORPINS/Chen_Kevin_16x32SRAM - Chen_Kevin_16x4SRAM

Groups — Named: *

Node Name	Direction	Locat
Chen_Kev...ss[3..0]	Input Group	
Chen_Kev...ys[1..0]	Input Group	

Tasks: Run Analysis and Elaboration; Early Pin Planning; Early Pin Planning...

Named: * — Filter: Pins: all

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Differential Pair
Chen_Kevi...egment_F2	Output	PIN_AB25	6	B6_N1	PIN_AB25	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_F3	Output	PIN_U22	6	B6_N1	PIN_U22	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_F4	Output	PIN_R6	1	B1_N0	PIN_R6	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_F5	Output	PIN_R4	1	B1_N0	PIN_R4	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_F6	Output	PIN_M5	2	B2_N1	PIN_M5	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_F7	Output	PIN_P9	2	B2_N1	PIN_P9	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_G0	Output	PIN_V13	8	B8_N0	PIN_V13	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_G1	Output	PIN_AB24	6	B6_N1	PIN_AB24	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_G2	Output	PIN_Y24	6	B6_N1	PIN_Y24	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_G3	Output	PIN_W24	6	B6_N1	PIN_W24	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_G4	Output	PIN_T3	1	B1_N0	PIN_T3	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_G5	Output	PIN_R3	1	B1_N0	PIN_R3	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_G6	Output	PIN_M4	2	B2_N1	PIN_M4	3.3-V LV...default)		24mA (default)	
Chen_Kevi...egment_G7	Output	PIN_N9	2	B2_N1	PIN_N9	3.3-V LV...default)		24mA (default)	
Chen_Kevin_Switches[7]	Input	PIN_A13	4	B4_N1	PIN_AC11	3.3-V LV...default)		24mA (default)	
Chen_Kevin_Switches[6]	Input	PIN_B13	4	B4_N1	PIN_AE8	3.3-V LV...default)		24mA (default)	
Chen_Kevin_Switches[5]	Input	PIN_C13	3	B3_N0	PIN_W12	3.3-V LV...default)		24mA (default)	
Chen_Kevin_Switches[4]	Input	PIN_AC13	8	B8_N0	PIN_AF9	3.3-V LV...default)		24mA (default)	
Chen_Kevin_Switches[3]	Input	PIN_AD13	8	B8_N0	PIN_AE9	3.3-V LV...default)		24mA (default)	
Chen_Kevin_Switches[2]	Input	PIN_AF14	7	B7_N1	PIN_T10	3.3-V LV...default)		24mA (default)	
Chen_Kevin_Switches[1]	Input	PIN_AE14	7	B7_N1	PIN_U10	3.3-V LV...default)		24mA (default)	
Chen_Kevin_Switches[0]	Input	PIN_P25	6	B6_N0	PIN_AC10	3.3-V LV...default)		24mA (default)	
Chen_Kevin_WE	Input	PIN_U4	1	B1_N0	PIN_U4	3.3-V LV...default)		24mA (default)	

0% 00:00:00

Figure 25d: Pin Planner of Block Diagram/Schematic File of 16 x 32 SRAM