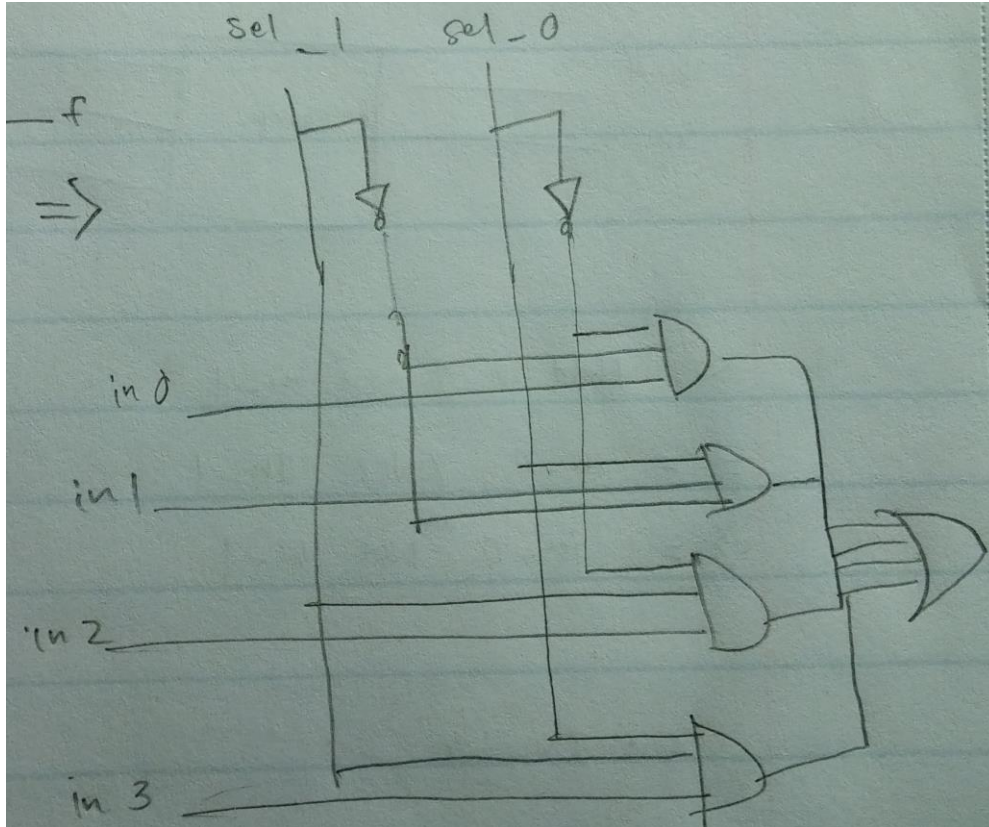


Assignment 1 Lab Report

4-to-1 Multiplexer (MUX):

The 4-to-1 multiplexer is a circuit that takes in 4 inputs and sends a signal out depending on which select gate is switched on or off.

Gate Implementation:



Truth Table:

Sel_1	Sel_0	f
0	0	In_0
0	1	In_1
1	0	In_2
1	1	In_3

Assignment 1 Lab Report

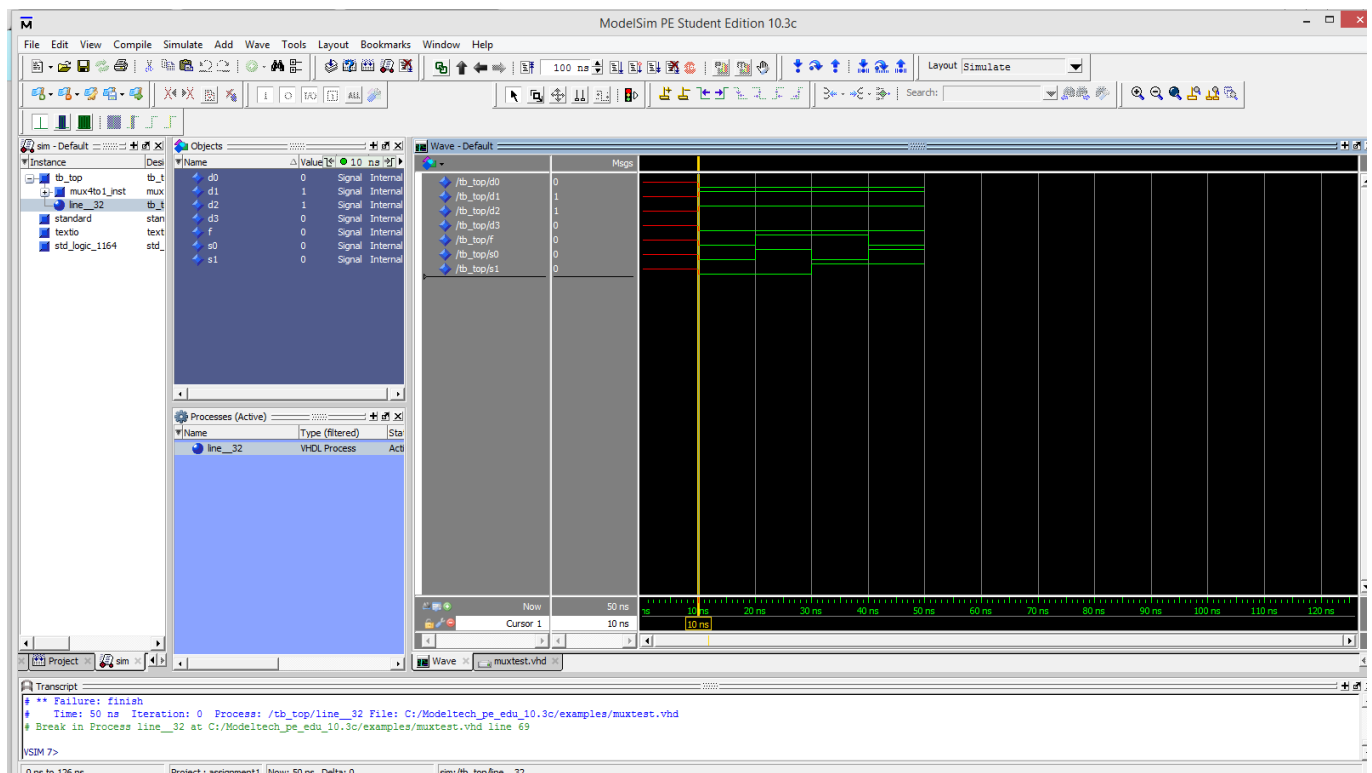
K-Map;

Sel_0	0	1
Sel_1		
0	In_0	In_1
1	In_2	In_3

Boolean Expressions:

$$f = (\text{in_0} * \text{sel_0}' * \text{sel_1}') + (\text{in_1} * \text{sel_0}' * \text{sel_1}) + (\text{in_2} * \text{sel_0} * \text{sel_1}') + (\text{in_3} * \text{sel_0} * \text{sel_1})$$

Waveforms:



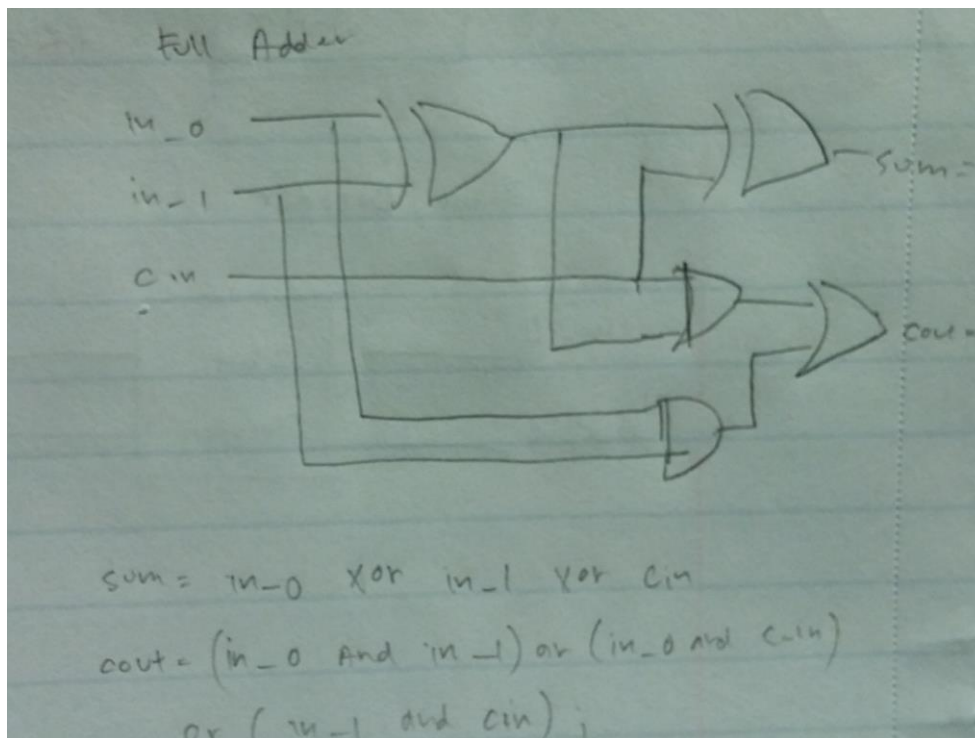
Assignment 1 Lab Report

Full Adder/Subtractor :

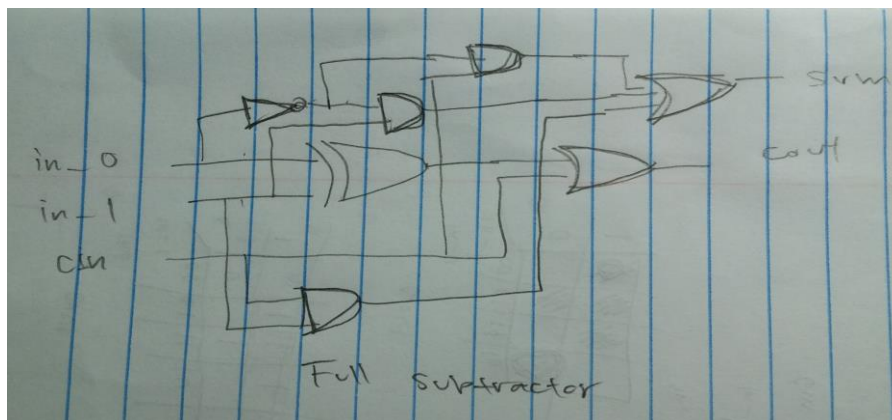
The full adder and subtractor is a circuit that takes in 2 1-bit binary numbers and either add or subtract them depending on the signal which the circuit is designed to choose. In my circuit 0 is add and 1 is subtract. What makes a full adder and subtractor different from the half adder and subtractor is that in a full adder and subtractor, carry ins and carry outs are taken in consideration during the mathematical process.

Gate Implementation:

Full Adder Gate:



Full Subtractor Gate:



Derek Yang
63118832

Assignment 1 Lab Report

Truth Table :

Adder

addorsub	in_0	in_1	cin	cout	sum
0	0	0	0	0	0
0	0	0	0	1	0
0	0	1	0	0	1
0	0	1	1	1	0
0	1	0	0	0	1
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	0	1
0	1	1	1	1	1

Subtractor

addorsub	in_0	in_1	cin	cout	sum
1	0	0	0	0	0
1	0	0	0	1	1
1	0	1	0	1	1
1	0	1	1	1	0
1	1	0	0	0	1
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	1	1

K-Map

Sum:

in_0in_1	00	01	11	10
cin 0		1		1
1	1		1	

Adder cout

in_0in_1	00	00	11	10
cin 0			1	
1		1	1	1

Subtractor cout

in_0in_1	00	01	11	10
cin 0		1		
1	1	1	1	

Derek Yang
63118832

Assignment 1 Lab Report

Boolean Equations:

Full Adder:

$$\text{Sum} = \text{in}_0 \text{ xor } \text{in}_1 \text{ xor } \text{cin}$$

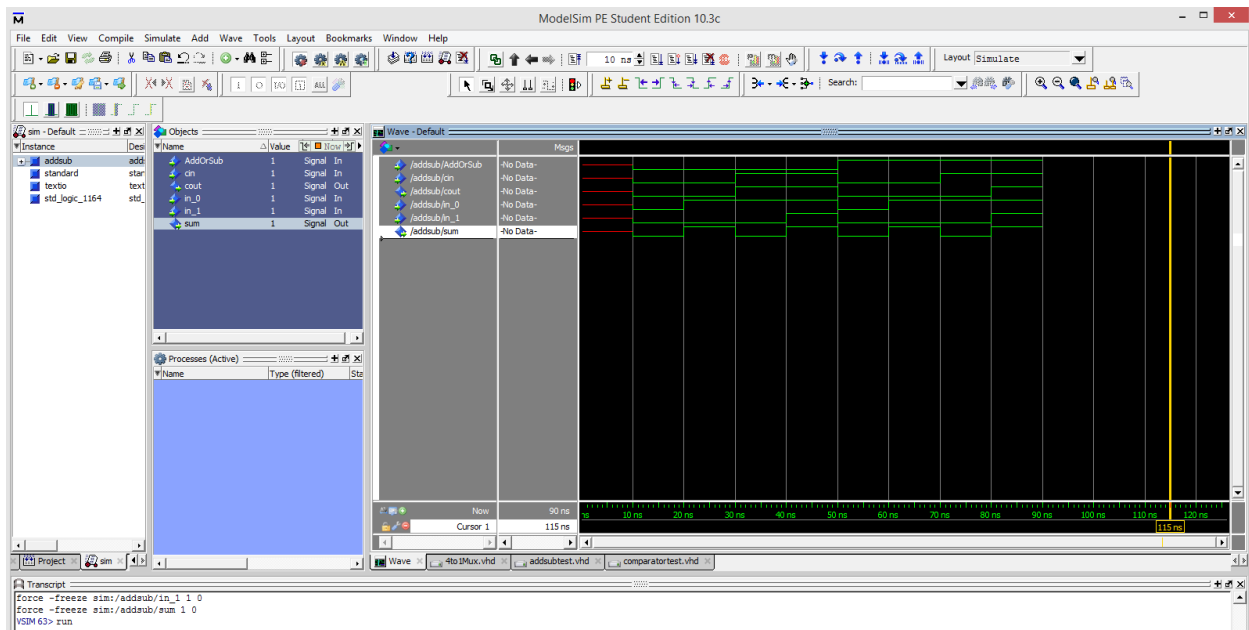
$$\text{Cout} = (\text{in}_0 * \text{in}_1) + (\text{in}_0 * \text{cin}) + (\text{in}_1 * \text{cin})$$

Full Subtractor:

$$\text{Sum} = \text{in}_0 \text{ xor } \text{in}_1 \text{ xor } \text{cin}$$

$$\text{Cout} = (\text{in}_0' * \text{in}_1) + (\text{in}_0' * \text{cin}) + (\text{in}_1 * \text{cin})$$

Wave map:



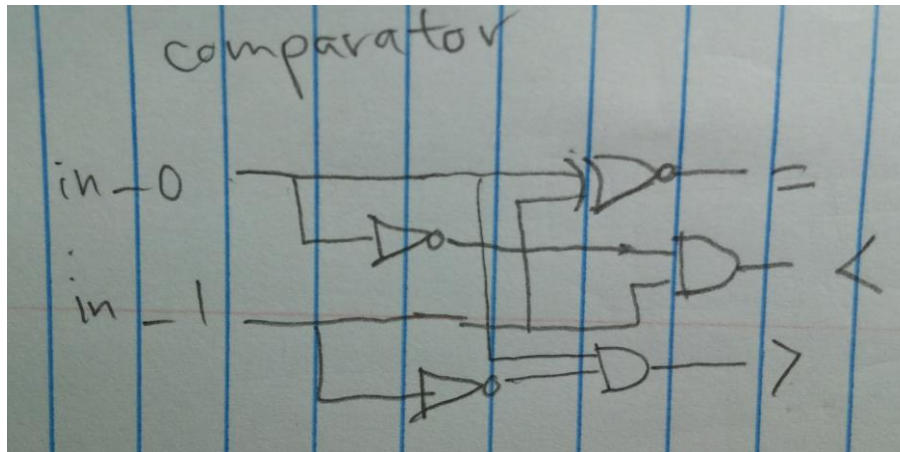
Derek Yang
63118832

Assignment 1 Lab Report

Comparator:

The comparator is a circuit that compares two 1-bit numbers and sends a signal through the gate corresponding to whether the 1st bit is either greater than, equal to, or less than the 2nd bit.

Gate Implementation:



Gate

Truth Table

K-Map

Comparator				Truth Table			K-Map		
$in-0$	$in-1$	comp.		$in-0$	$in-1$	$<$	$=$	$>$	
0	0			0	0	0	1	0	
0	1			0	1	1	0	0	
1	0			1	0	0	0	1	
1	1			1	1	0	1	0	

		$in-0$	
$in-1$	0	0	1
	1	0	0

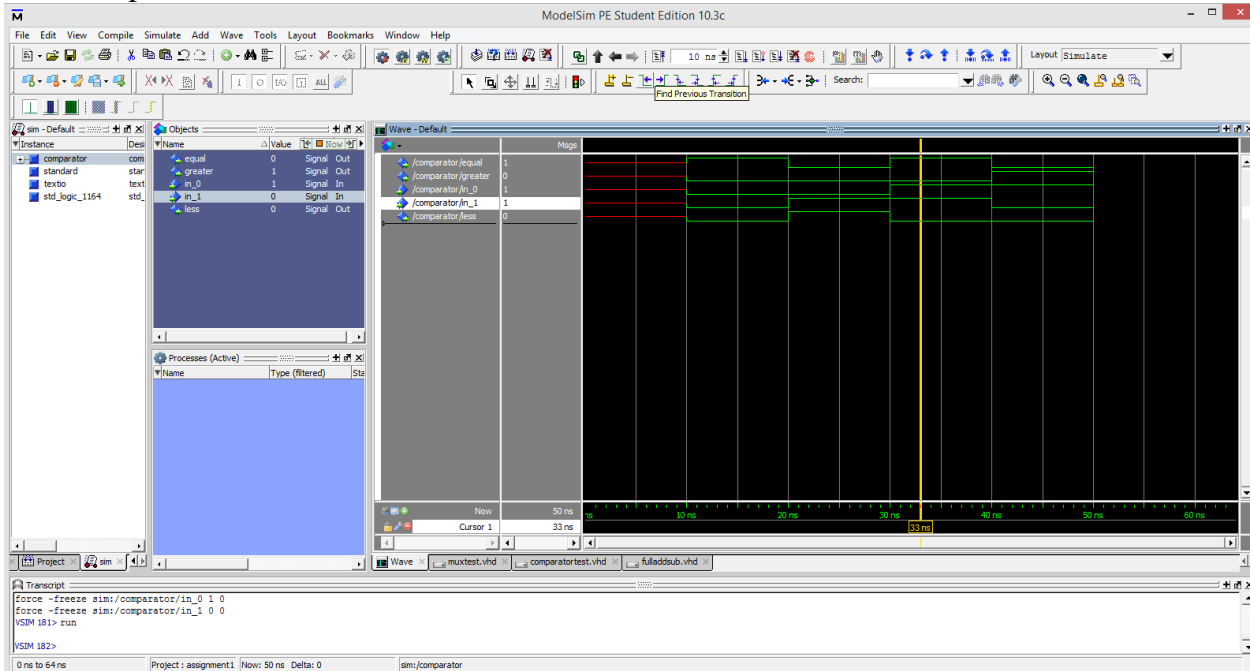
$< = \text{NOT } in-0 \text{ and } in-1$
 $= = in-0 \text{ XNOR } in-1$
 $> = in-0 \text{ and NOT } in-1$

Boolean Equations

Derek Yang
63118832

Assignment 1 Lab Report

Wave Map:

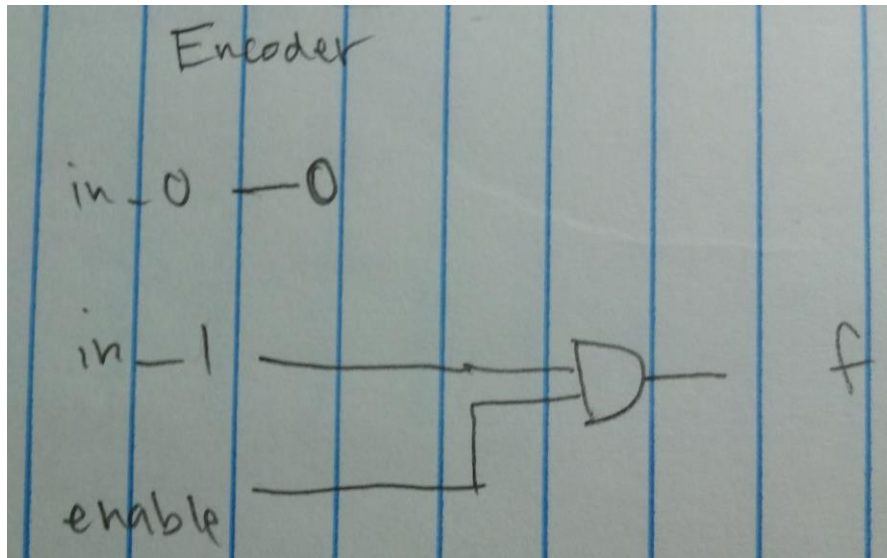


Assignment 1 Lab Report

Encoder:

The encoder is a circuit that takes in a decimal number and transcribes it into a binary number.

Gate Implementation:



Gate

Truth Table

2-bit Encoder

enable	in_0	in_1	f	in_3	in_2	in_1	in_0	f ₁	f ₀
0	x	x	0	0	0	0	1	0	0
1	0	0	0	0	0	1	x	0	1
1	0	1	0	0	1	x	x	1	0
1	1	0	1	1	x	x	x	1	1
1	1	1	1						

enable	in_0	f
0	0	0
0	1	0
1	0	1
1	1	1

in_1 enable

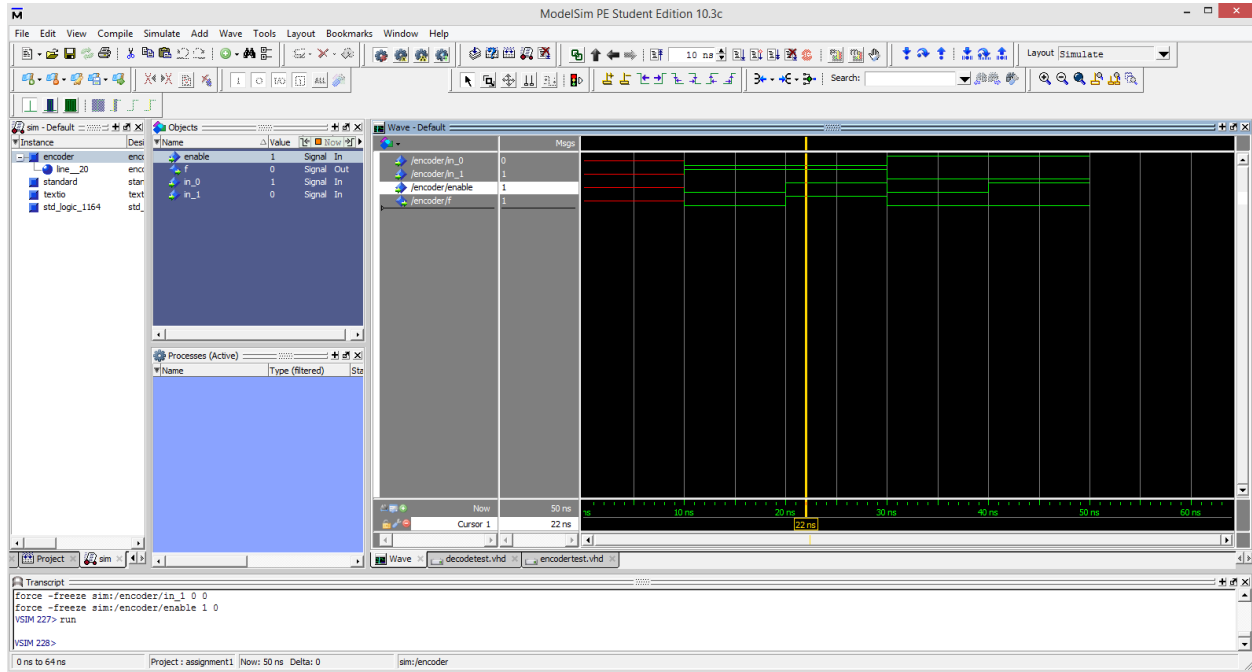
$f = 0$ and in_1

K- Map

Boolean Equation

Assignment 1 Lab Report

Wave Map:

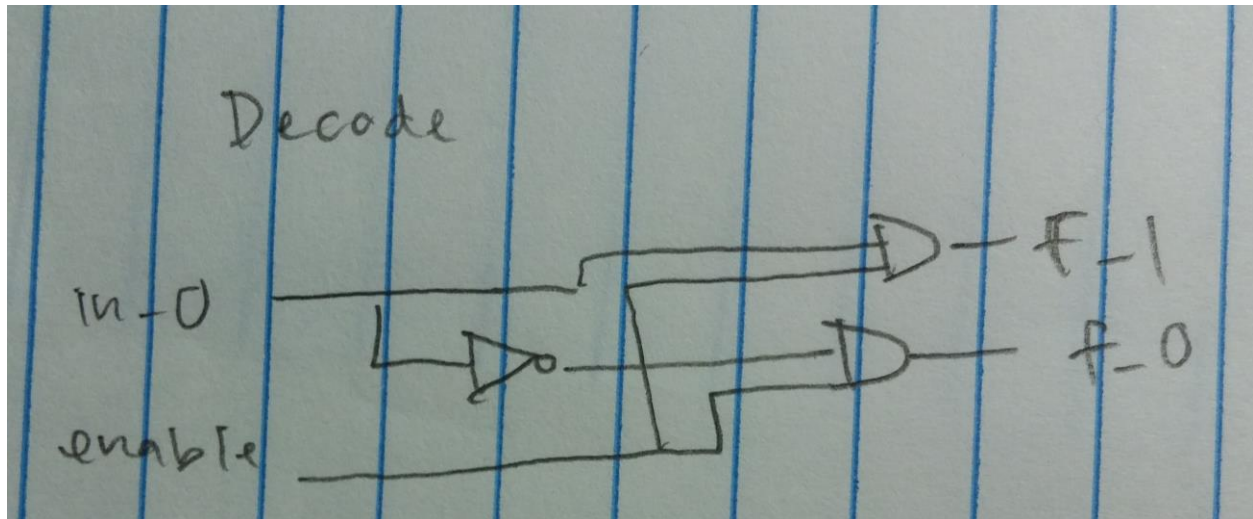


Assignment 1 Lab Report

Decoder:

The decoder is a circuit that takes a binary number and transcribes it into a decimal number, It is the opposite of an encoder.

Gate Implementation:



Gate

Truth Table

2 bit Decoder

enable	in-0	f-0	f-1
0	x	0	0
1	0	1	0
1	1	0	1

f-0

enable	0	1
in-0	0	1
0	0	1
1	0	0

↓

$f_0 \leftarrow \text{enable and } (\text{not } in_0)$

f-1

enable	0	1
in-0	0	1
0	0	0
1	0	1

↓

$f_1 \leftarrow \text{enable and } in_0$

K- Map Boolean Equation

Derek Yang
63118832

Assignment 1 Lab Report

Wave Map:

