Raymond You and Shravali Reddy

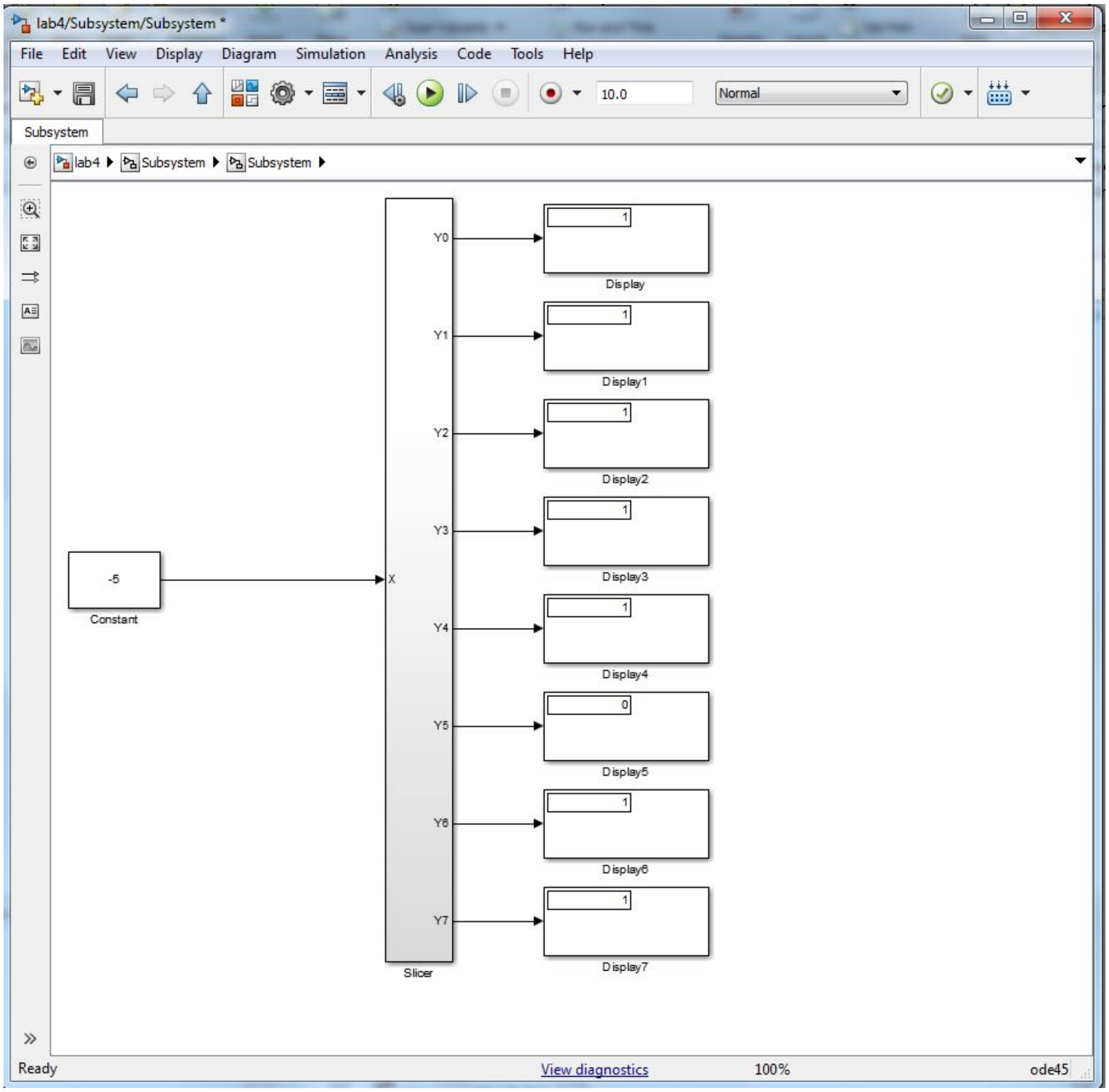
EECE 2160 – Kimani 8am

Lab 4

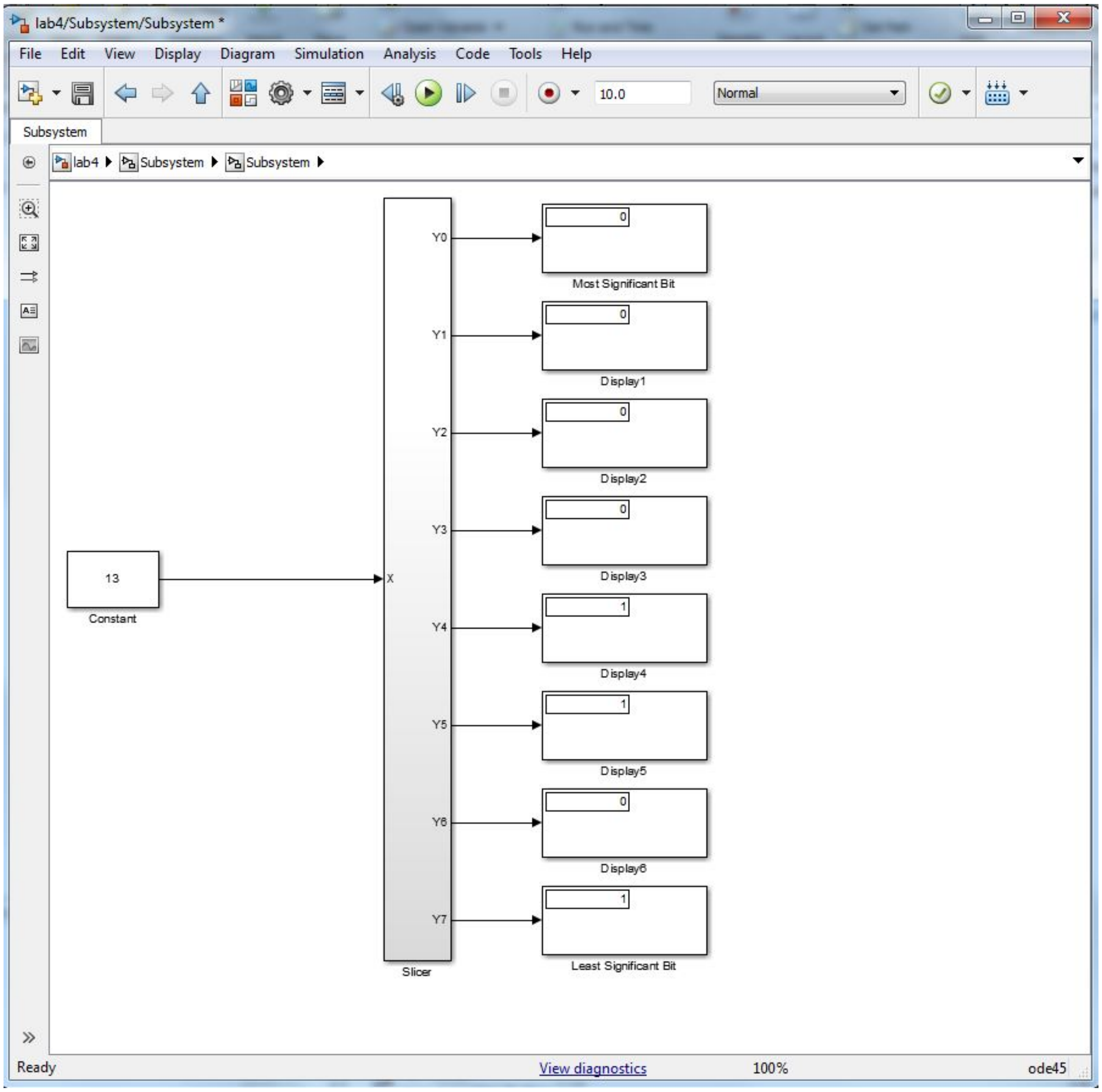
Assignment 1

a)

Twos complement included



Positive Number

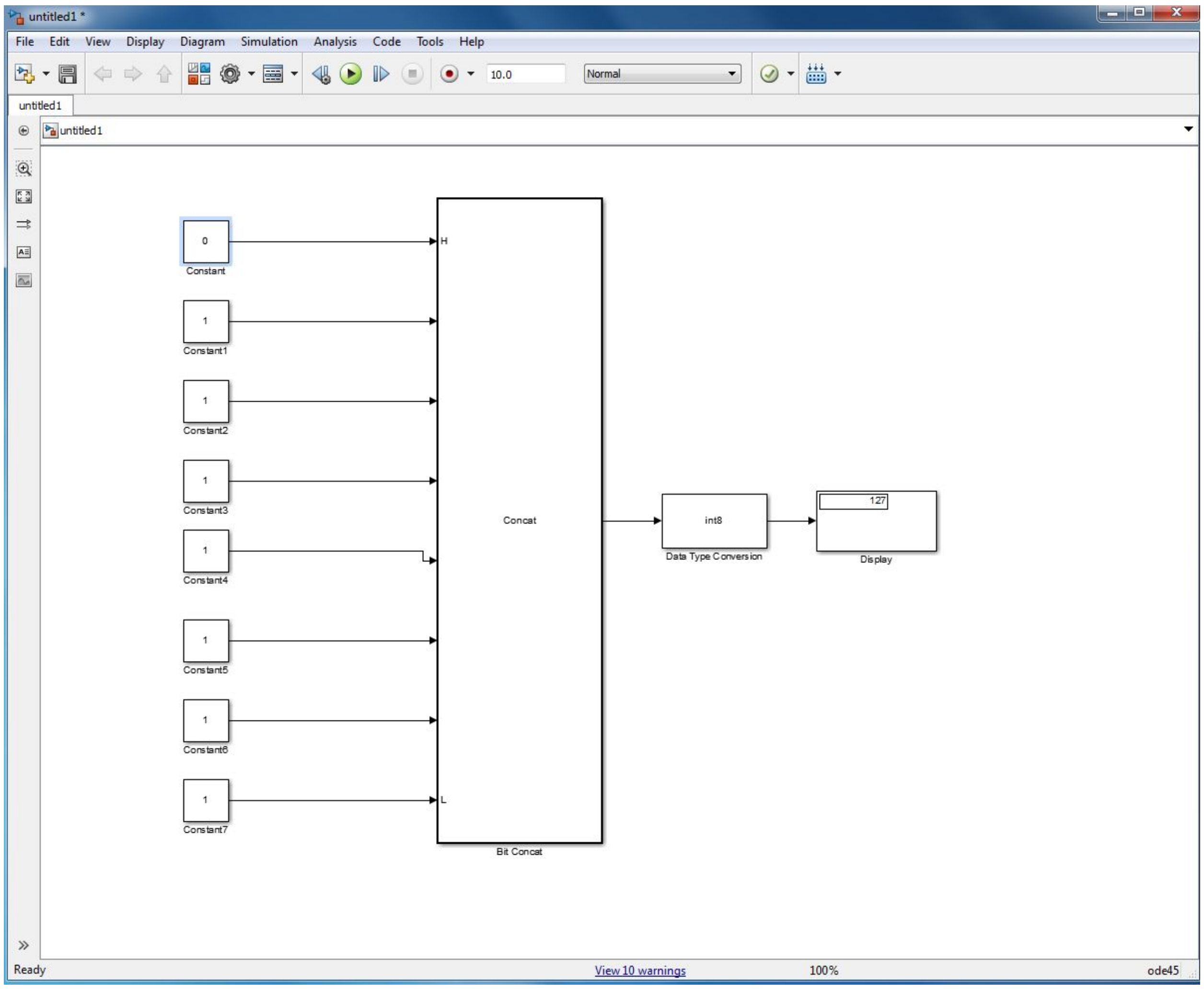


b) In the outputs displayed by Simulink, each of the outputs represent an individual bit of the number. So for example above, 13 should be 00001011, so each of the displays should show an individual bit. Overall, it converts an int into a binary number. This also works with negative numbers using twos complement.

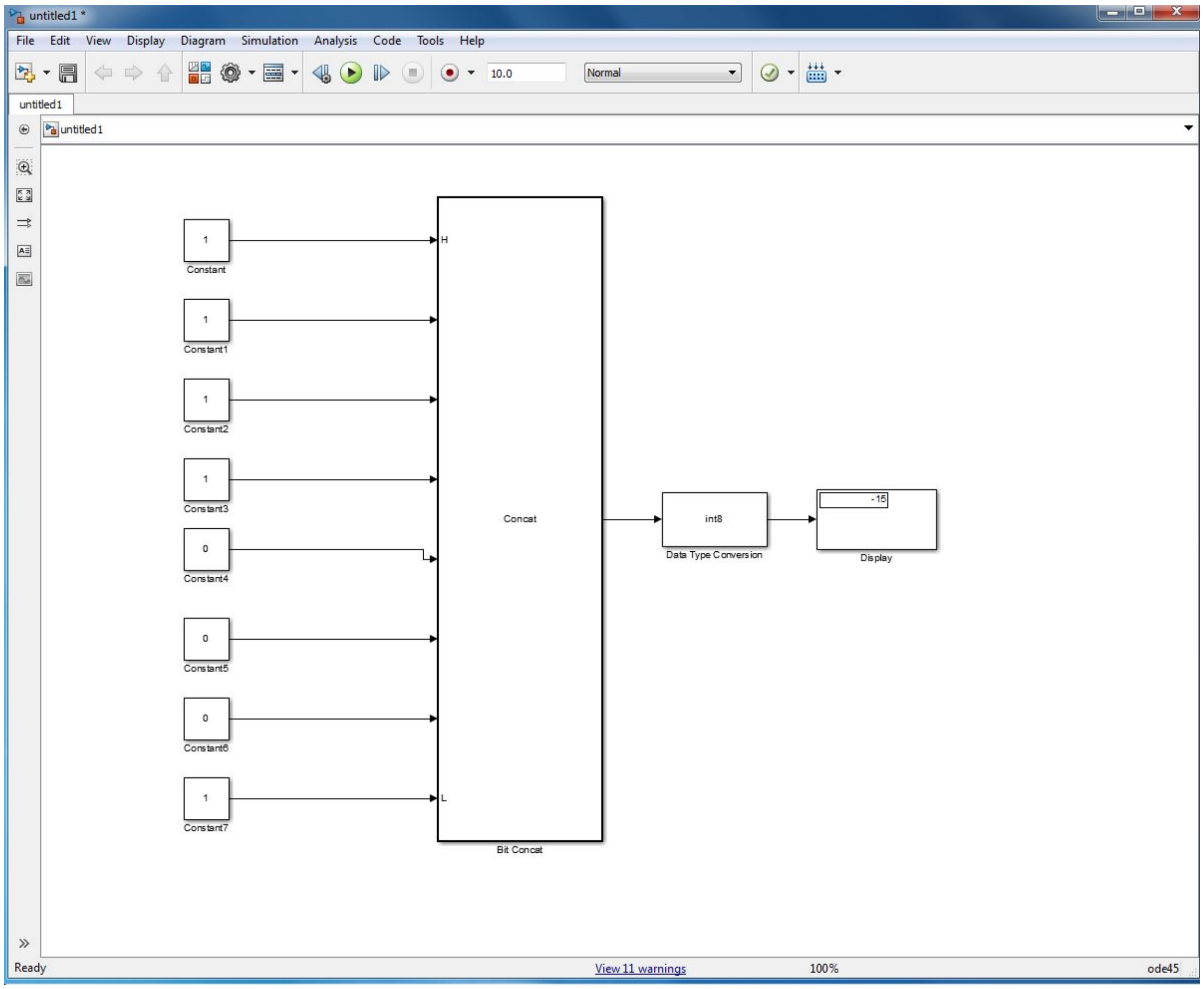
Assignment 2

a)

Positive numbers

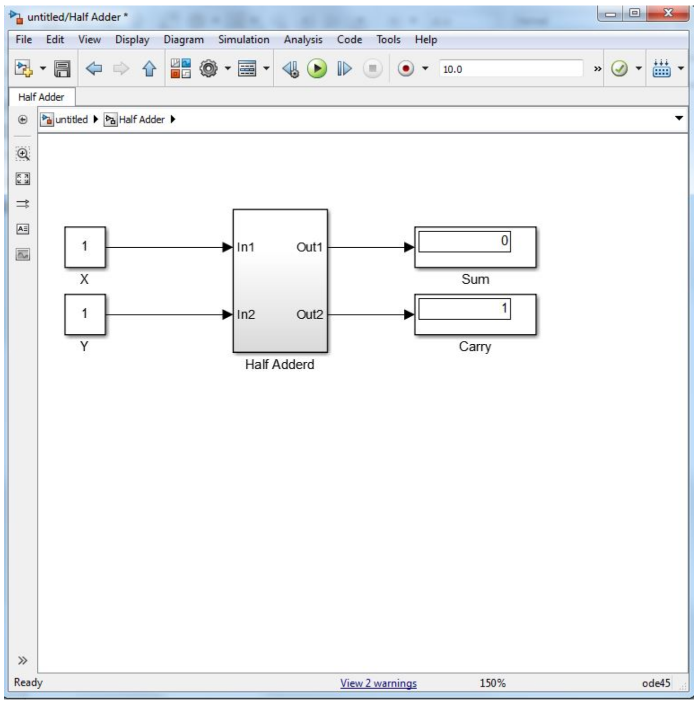


Negative Numbers



b) The bit concat takes in the individual bits for an 8-bit binary number and converts it into an int8 value. So for example above, 11110001 should be -15 in binary number which is shown above, verifying the correctness of the program and output. The data type converter is essential to properly represent negative 8-bit twos complement numbers.

Assignment 3



Truth table half adder

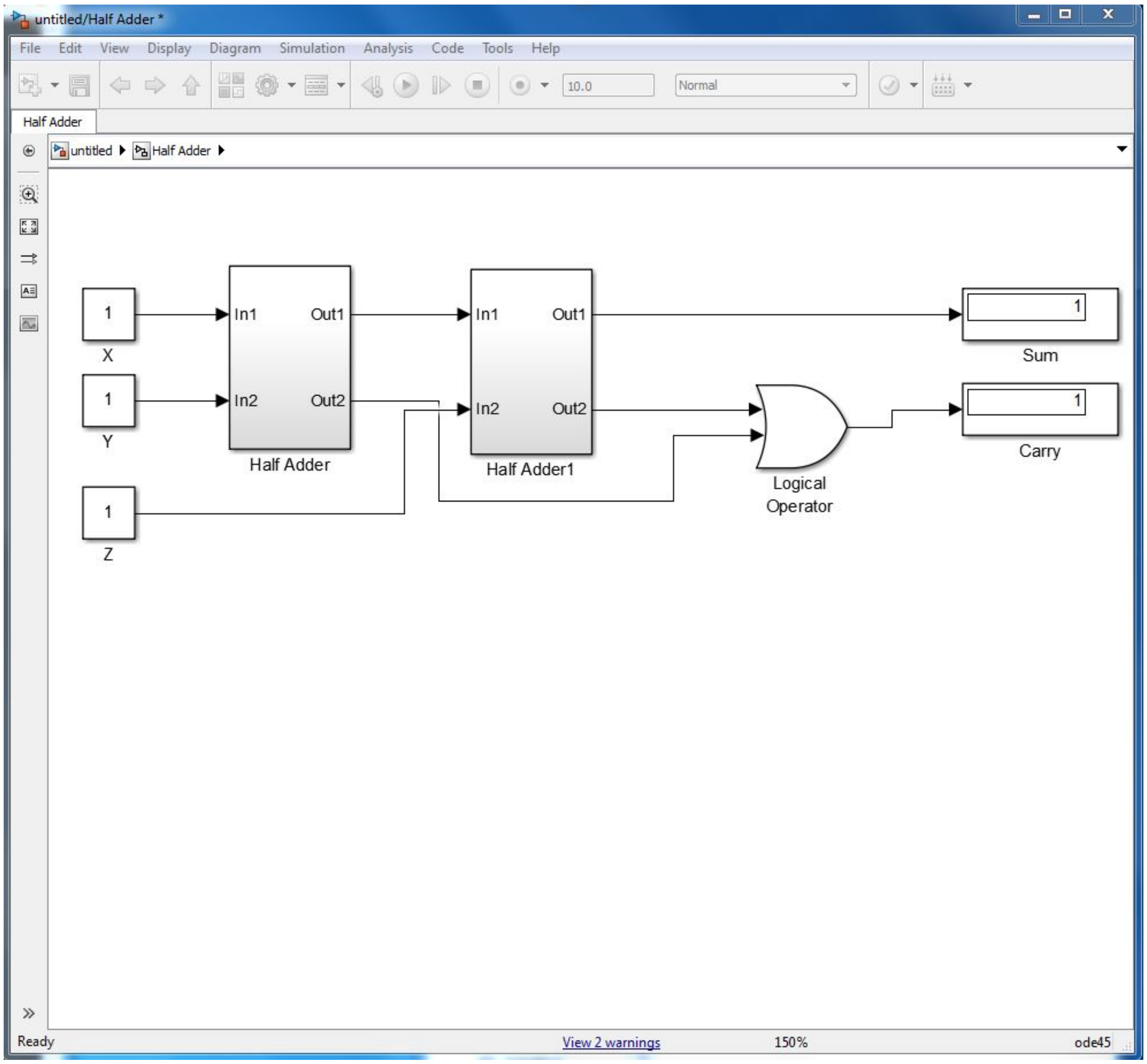
Inputs **Outputs**

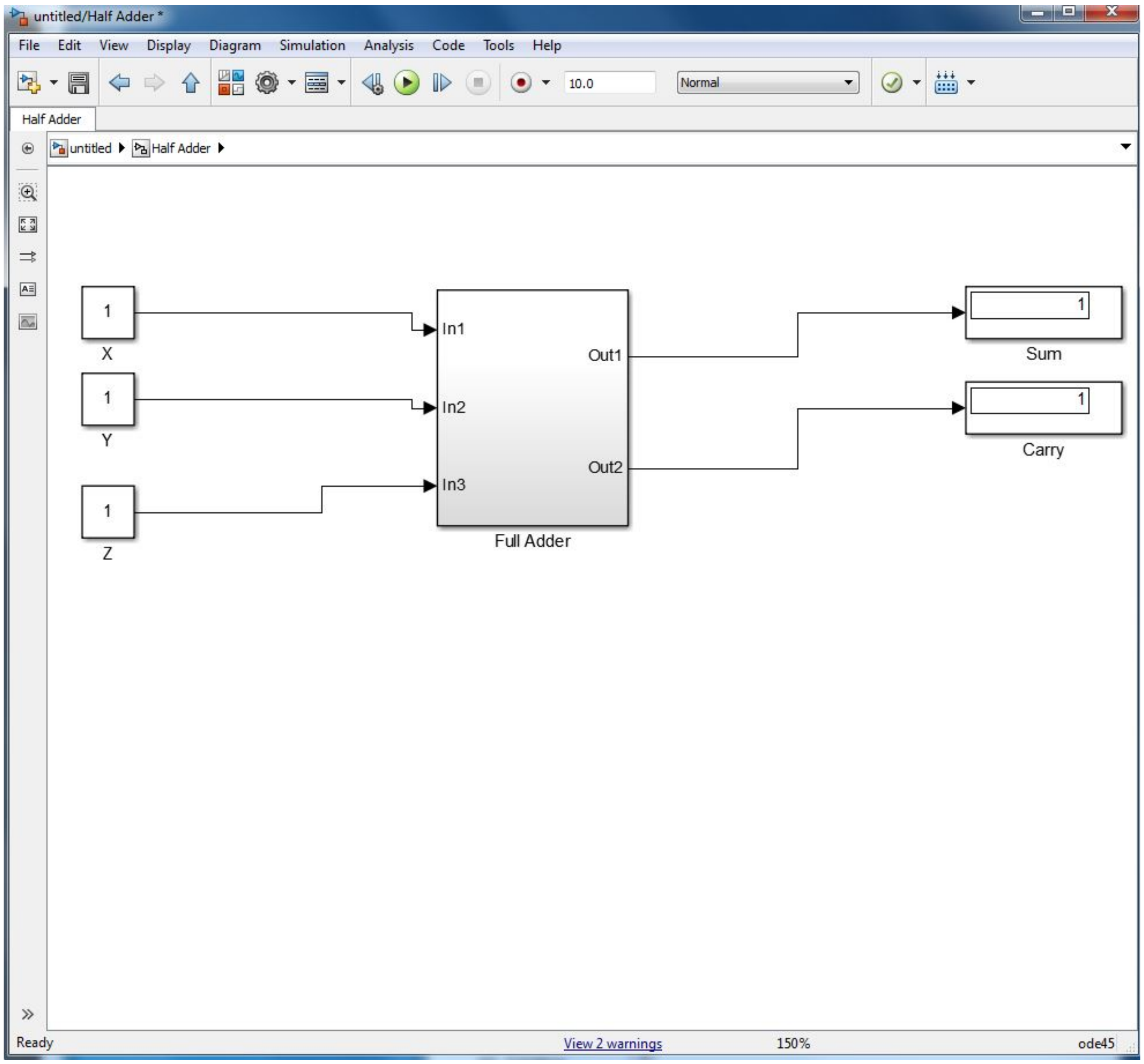
A	B	SUM	CARRY
1	1	0	1
1	0	1	0
0	1	1	0
0	0	0	0

So we have two inputs of 1-bit numbers. The half adder solves the problems of adding two 1-bit numbers by having two outputs, the sum which represents the remainder of FirstBit + SecondBit / 2, while the carry is the result.

Assignment 4

Screenshots:


untitled/Half Adder *
File Edit View Display Diagram Simulation Analysis Code Tools Help
10.0
Normal
Half Adder
untitled
Half Adder
1
X
1
Y
1
Z
In1
Out1
In2
Out2
Half Adder
In1
Out1
In2
Out2
Half Adder1
Logical Operator
1
Sum
1
Carry
Ready
View 2 warnings
150%
ode45

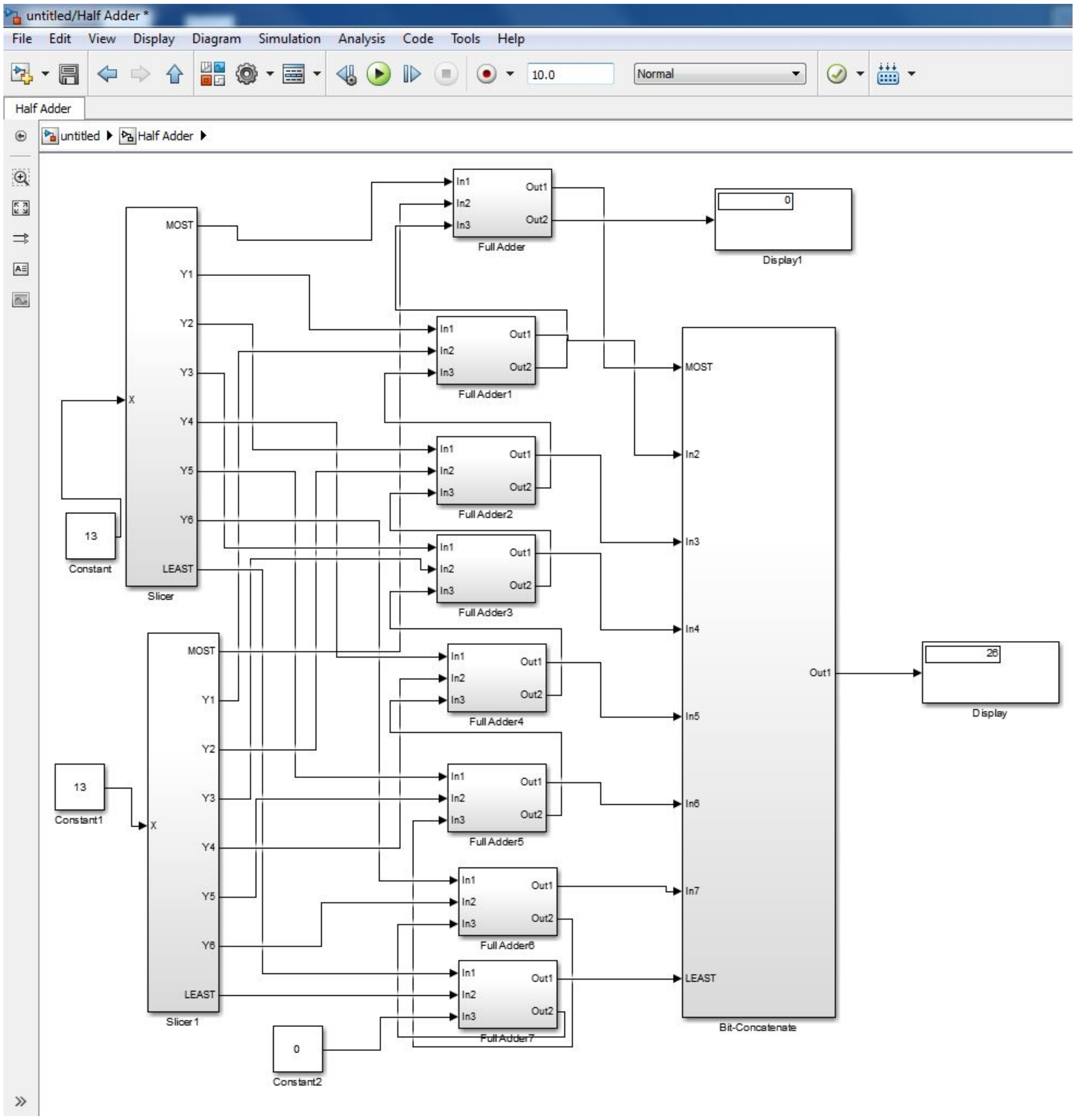


The full-adder adds 3 one-bit binary numbers such as X, Y, Z and outputs 2 one-bit binary numbers which consist of a sum and carry. X and Y are the operands, and Z is a bit carried in from the previous less-significant stage. A full adder is constructed from two half adders by connecting X and Y to the input of one half adder, connecting the sum from that to an input to the second adder, connecting C to the other input and OR the two carry outputs. If any of the half adder logic produces a carry, there will be an output carry which is why the OR gate is used.

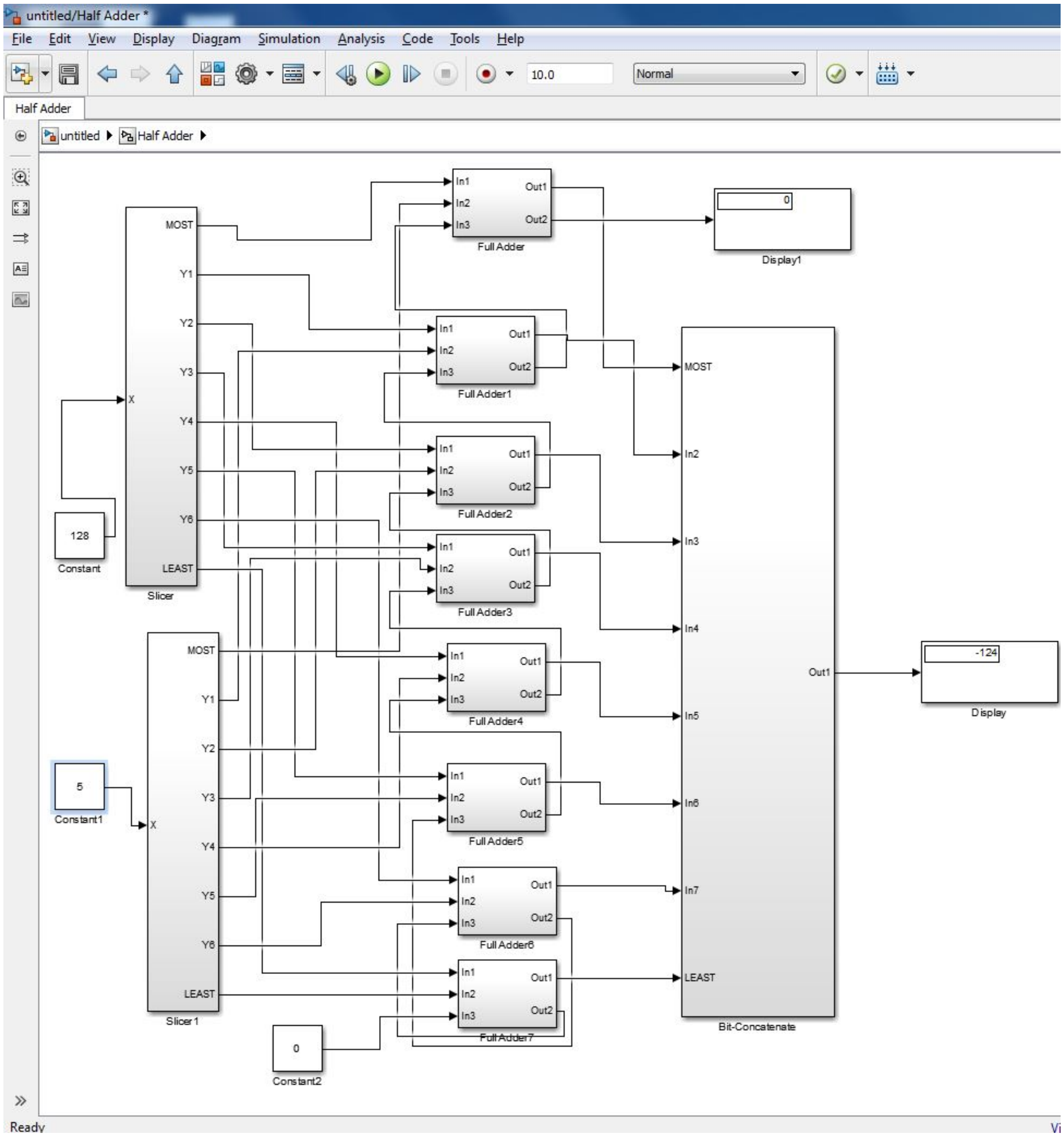
Assignment 5

It is an overflow when you add two positive numbers and the sum is greater than 127 and you get a negative sum since we are dealing with int8. Also if you add two negative numbers and you get a positive sum. In our design, it takes in two int8 inputs and it goes through the slicer which converts it into one bit parts of the binary number. Each of these individual bit parts go into a full adder as one of the inputs. The other input of the full adder is the output of the overflow from the low bit to the high bit, also known as a "ripple-carry" adder. The sum from each of the full adder is then sent into a bit-concatenate that sums of the bits and converts it back into an int8 number and outputs the result into the display. Then there is another display that shows the overflow bit. The purpose of output sgnal Cout is the sum of the two int8 inputs.

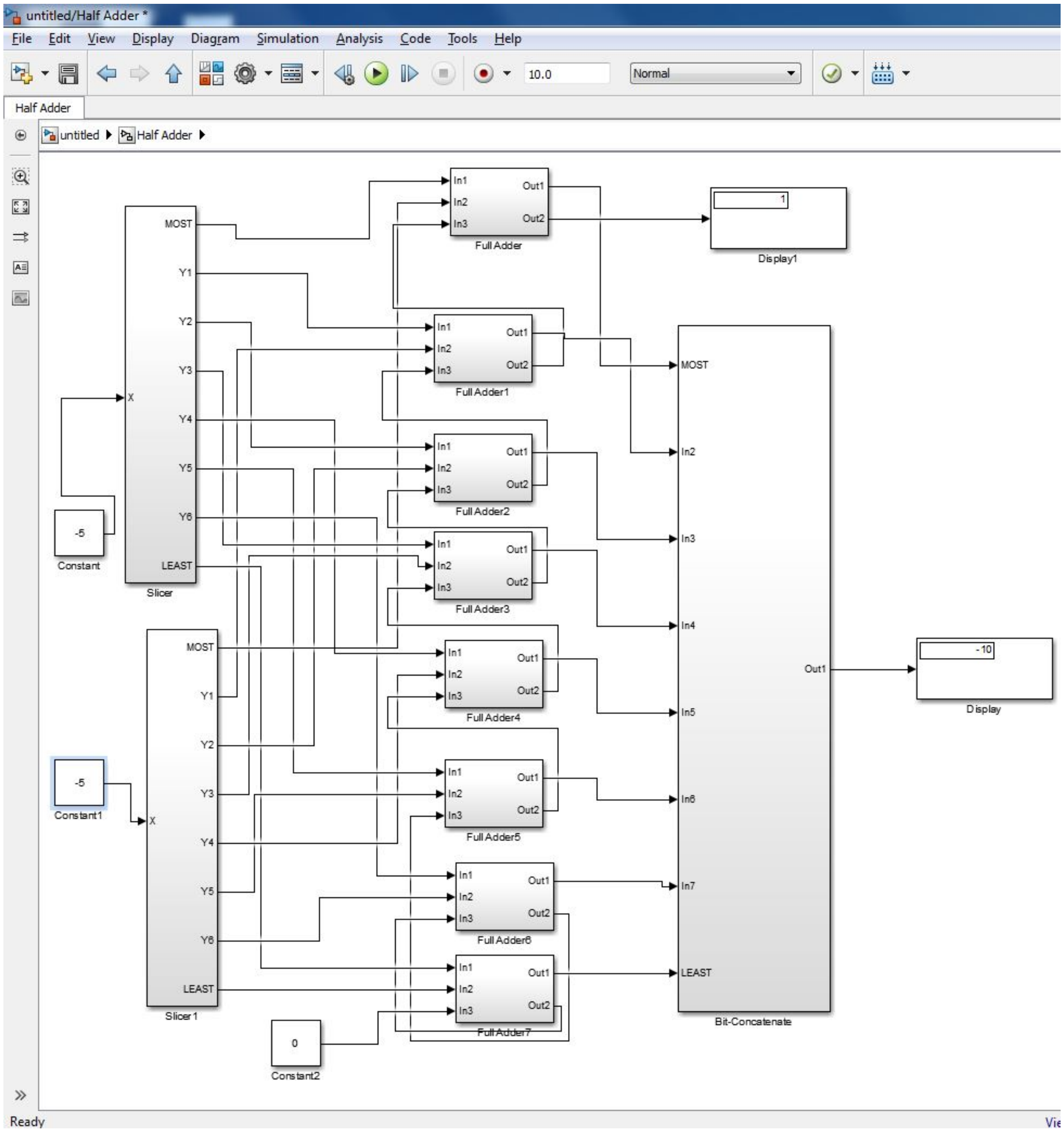
Two positive values that add up to less than 127: It should produce the sum in the sum display and 0 for the carry out display.



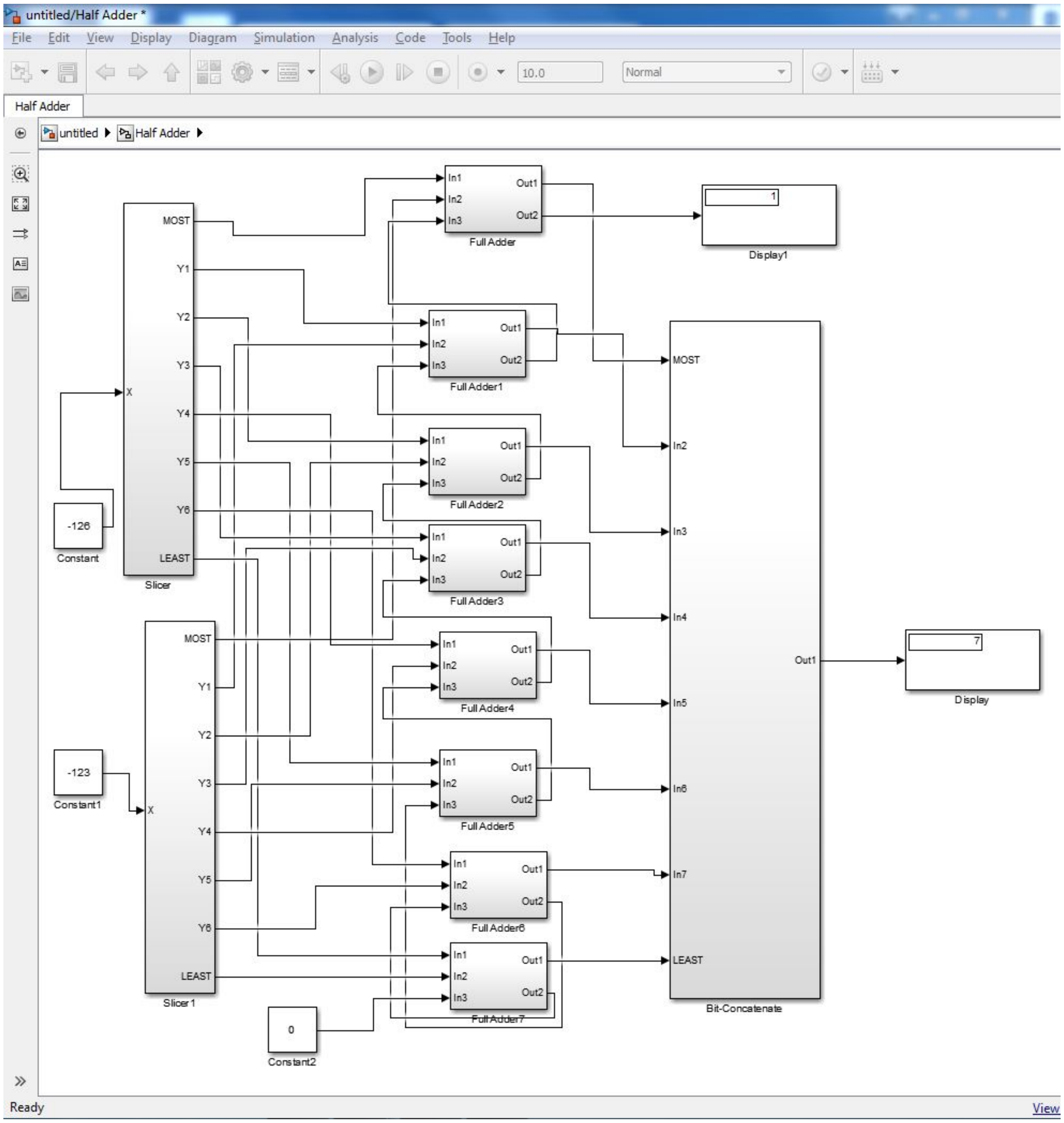
Two positive values that add up to more than 127: It should produce a negative sum in the sum display and a 0 for the carry out display.



Two negative values that add up to greater than -127: It should produce the sum of the two negative numbers in the sum display and a 1 in the carry out display due to twos complement.



Two negative values that add up to less than -127: It should display a positive sum in the sum display and a 1 in the carry out display.


untitled/Half Adder *
File Edit View Display Diagram Simulation Analysis Code Tools Help
10.0
Normal
Half Adder
untitled
Half Adder
MOST
Y1
Y2
Y3
Y4
Y5
Y6
LEAST
X
Slicer
-126
Constant
-123
Constant1
Slicer1
0
Constant2
In1
In2
In3
Out1
Out2
Full Adder
Full Adder1
Full Adder2
Full Adder3
Full Adder4
Full Adder5
Full Adder6
Full Adder7
1
Display1
MOST
In2
In3
In4
In5
In6
In7
LEAST
Out1
Bit-Concatenate
7
Display
Ready
View