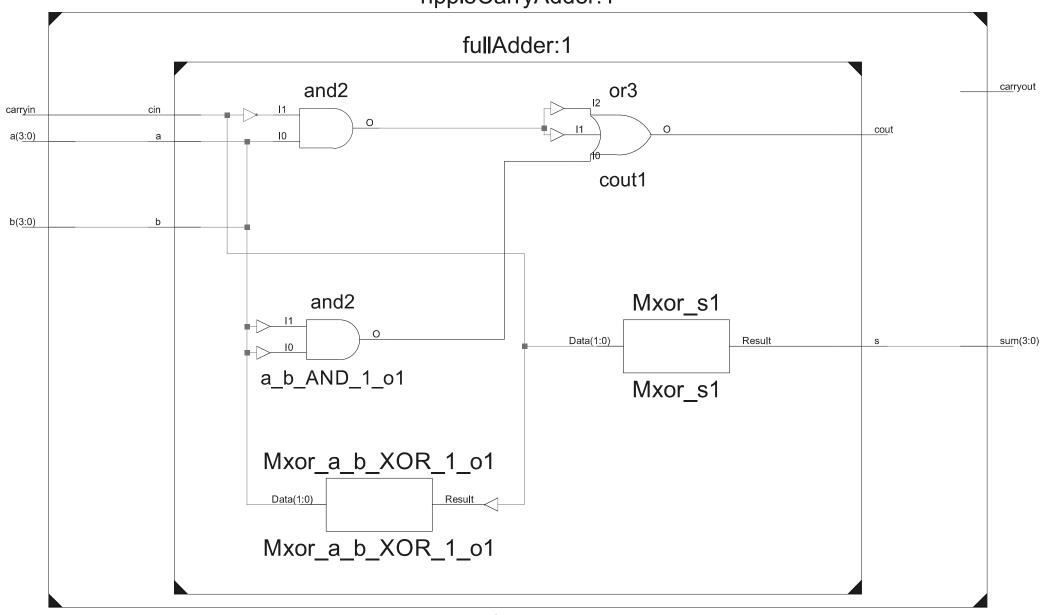
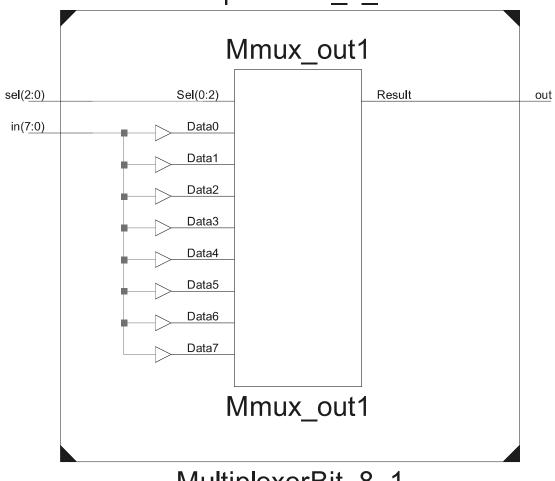
## rippleCarryAdder:1



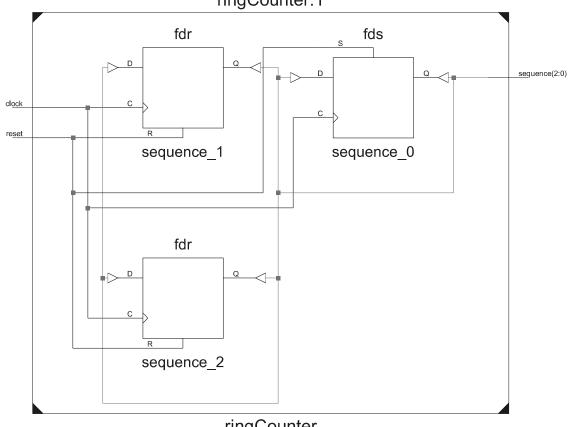
rippleCarryAdder

## MultiplexerBit\_8\_1:1



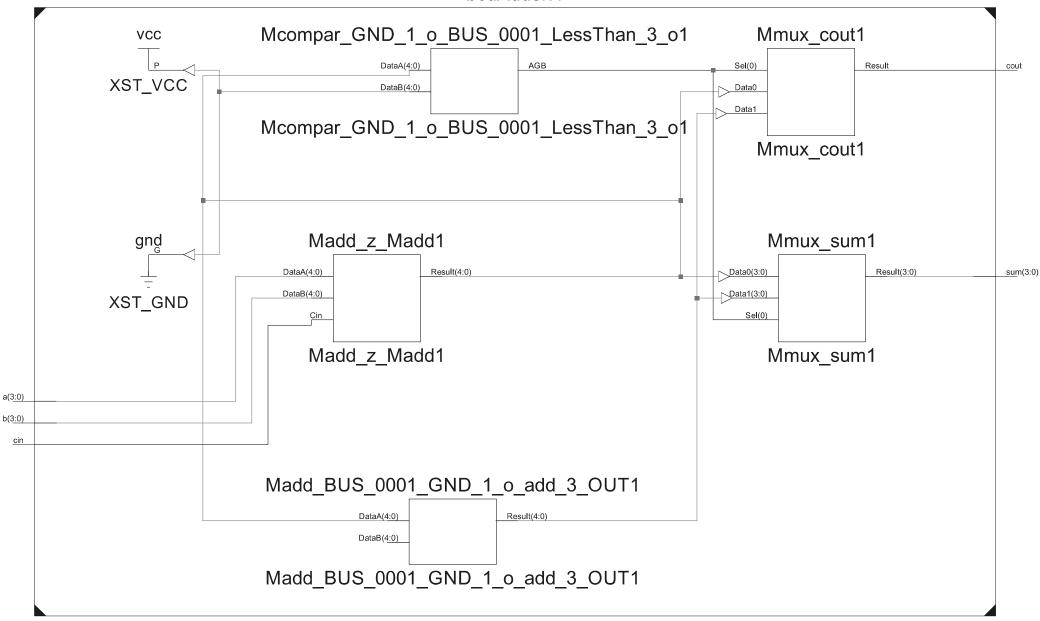
MultiplexerBit\_8\_1

## ringCounter:1

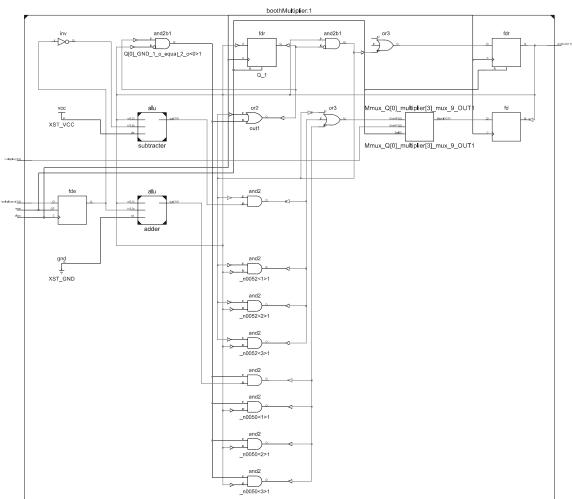


ringCounter

bcdAdder:1

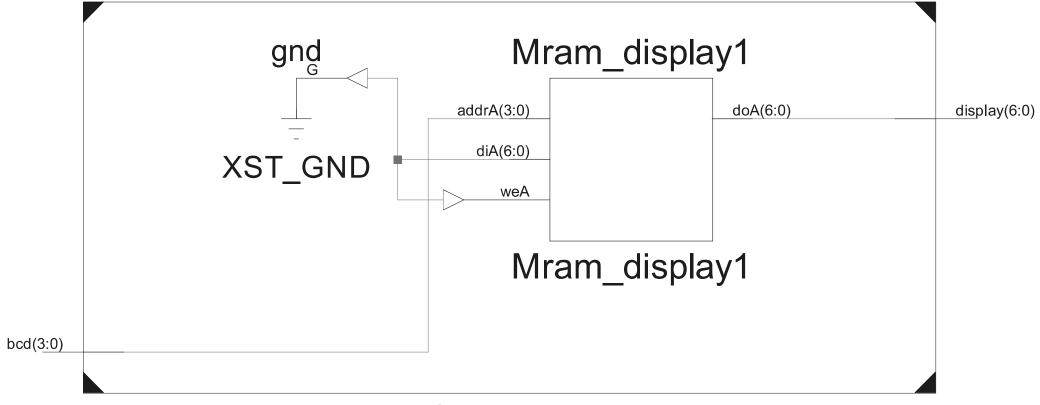


bcdAdder



boothMultiplier

## sevenSegmentDisplay:1

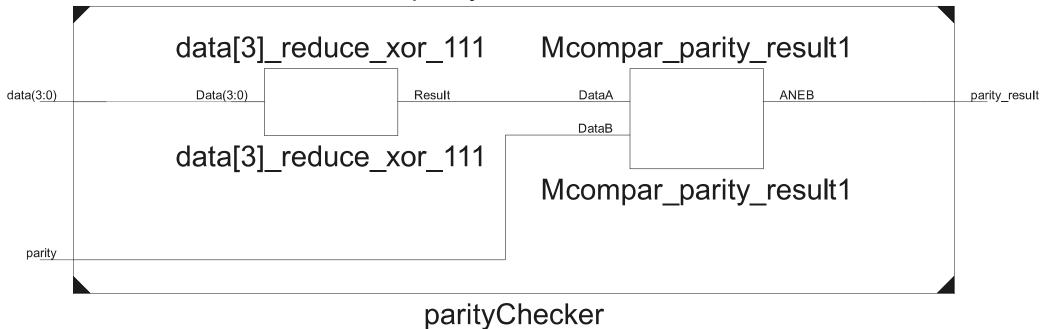


sevenSegmentDisplay

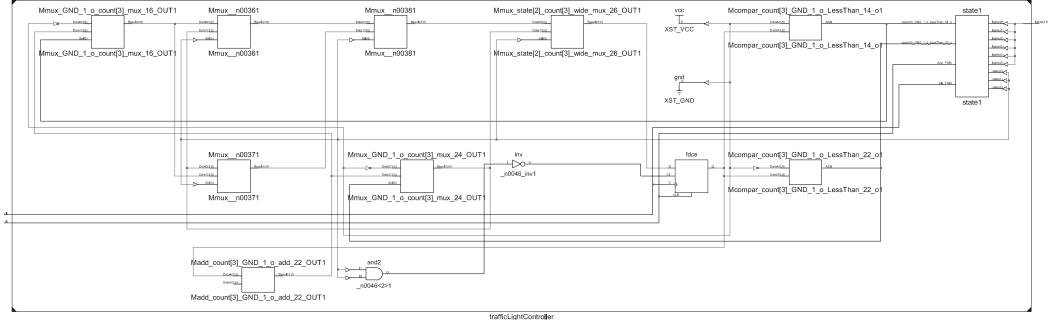
# binaryToGrayCode:1 Mxor\_gray<0>1 binary(3:0) Data(1:0) Result Mxor\_gray<0>1 gray(3:0) Mxor\_gray<1>1 Data(1:0) Result Mxor\_gray<1>1 Mxor\_gray<2>1 Data(1:0) Result Mxor\_gray<2>1

binaryToGrayCode

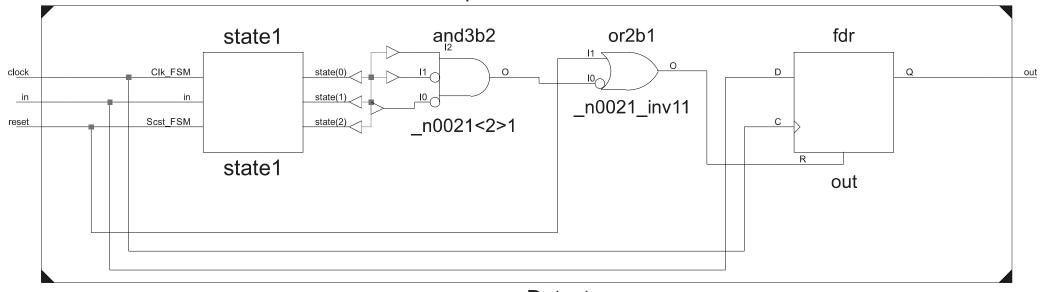
## parityChecker:1



#### trafficLightController:1



#### sequencDetector:1



sequencDetector

400 ns	420 ns	440 ns	460 ns	480 ns	500 ns	520 ns	54
1010 10	011 1100 110	1110 1111	0000 0001	0011 0100	0101 0110	0111 \ 1000	X
							$\perp$
 		0010		X	0011		
1000 10	001 1010 10:	1100 1101	1110 1111	0000 0001	0010 0011	0100 0101	
0000010 0000	0000010\000000	0010 0000010 0000010.	\0000010\0000010	. 0000011 0000011	0000011 0000011	. 0000011 0000011	\0
[							

sum[3:0]

carryout

a[3:0]

b[3:0]

carryin

i[10:0]

	30 ns	40 ns	50 ns	60 ns	70 ns
<b>¼</b> out <b>减</b> d[7:0]			01101001		
■ d[7:0] ■ sel[2:0]	011	100	101	110	111
■ i[3:0]	0011	0100	0101	0110	0111
<b>■</b> [3.0]	0011	0100	0101	0110	0111

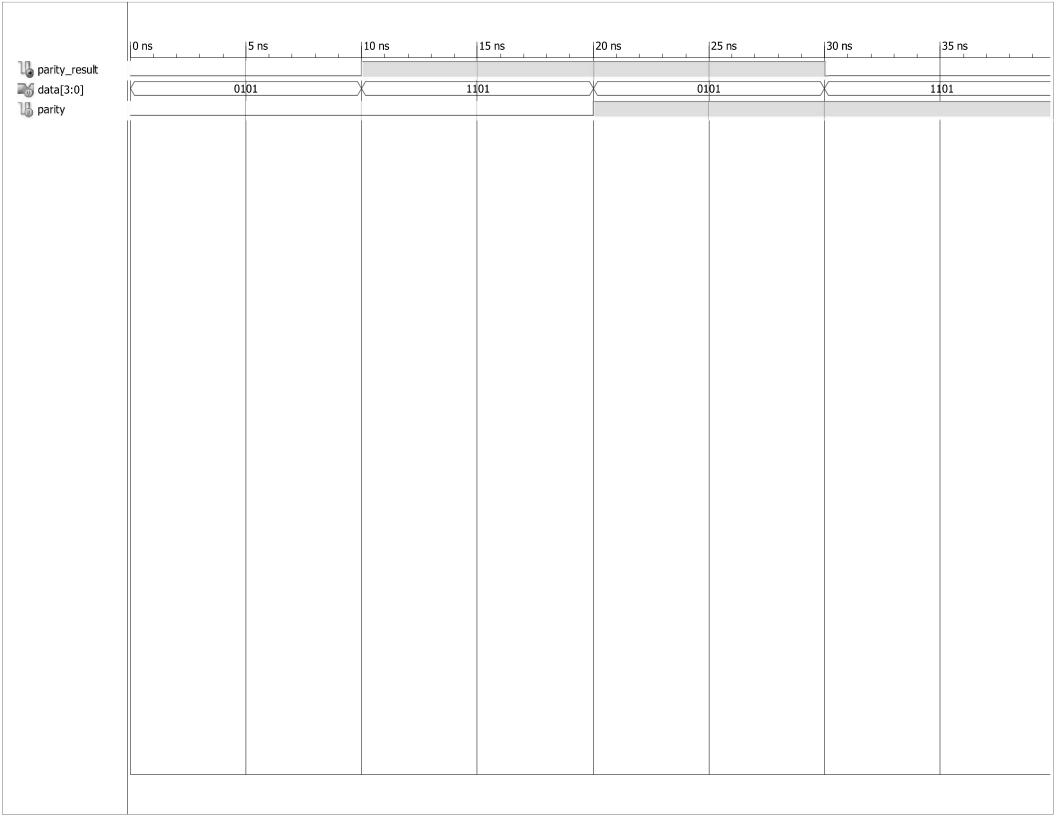
	0 ns	100 ns	200 ns	300 ns	400 ns	500 ns
sequence[2:0]	001	010	100	001	010	100
clock						
reset						
) reset						

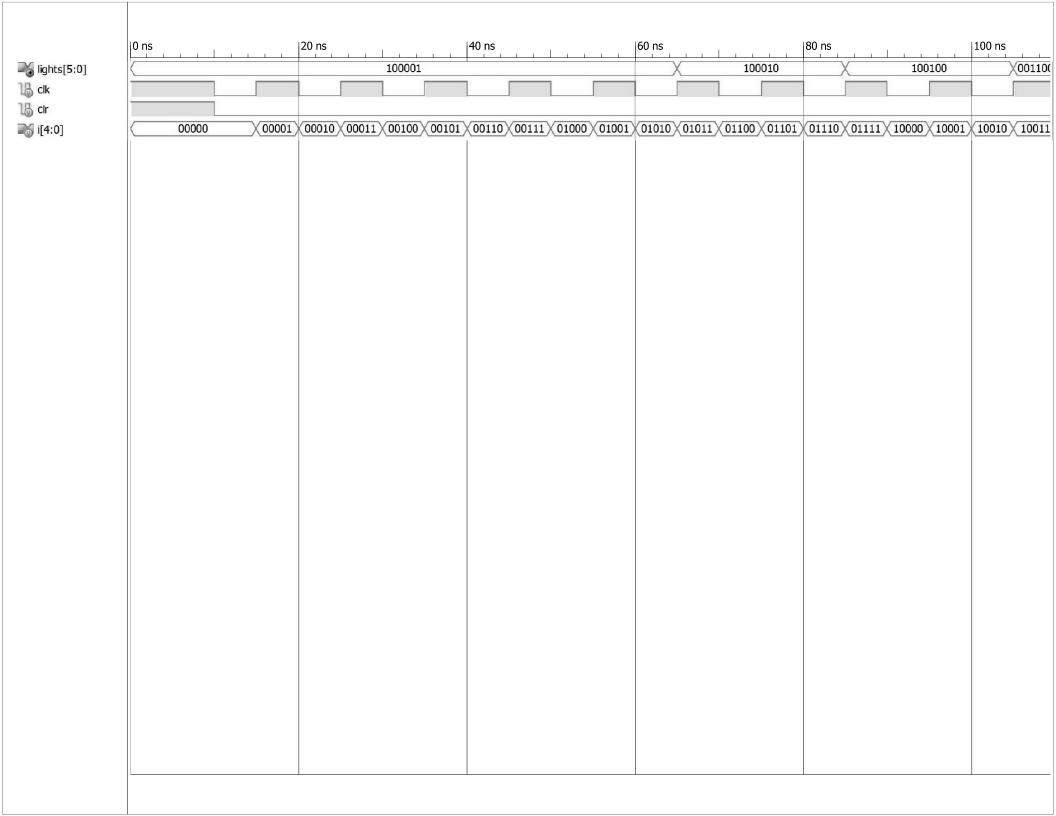
	80 ns	100 ns	120 ns	140 ns	160 ns	180 ns	200 ns
■ sum[3:0]	1000 1001	0001 0010	0011 0100	0101 0110	0111 1000	1001 0000	0010
<b>¼</b> cout							
a[3:0]	0000	0000 \ 0001	0010	0001		(1000 \ 1001	0010
■ b[3:0]	1000 \ 1001	0000 0001	0010 0011	0100 \ 0101	0110 0111	1000 1001	0000
i[3:0]	0000			0001		>	0010
■ j[3:0]	1000 \ 1001	0000 0001	0010 0011	0100 \ 0101	0110 0111	1000 1001	0000
		ı	1	I .	I .	ı	

	0 ns	1	20 ns		40 ns		60 ns	1	80 ns	1
fproduct[7:0]	00000010	X 00000001	11101000	X 00001100	00000110	00000111	11001011	X 11100101	X 11110010	X 0011000
multiplicand[3:0]			0011			X		0111		
🖏 multiplier[3:0]			0010			X		0111		
clock										
a reset										
	H		1						1	

	20 ns	30 ns	40 ns	50 ns	60 ns
M disular (F. 01	101101		110011	011011	011111
display[5:0]		111001			
■ bcd[3:0]	0010	0011	0100	0101	0110
■ i[3:0]	0010	0011	0100	0101	0110

		60 ns		80 ns	1	100 ns	
<b>a</b> gray[3:0]	0111	0101	0100	1100	1101	1111 X	1110
■ binary[3:0]	0101	0110	0111	1000	1001	1010	1011
■ i[4:0]	00101	00110	00111	01000	01001	01010	01011





	20 ns		40 ns	60 ns	
U out		_			
l∰ in l∰ reset					
1 clock					