Signature Analyzer

Presented By : Andrew O'Neil-Smith, Amy Zamon, Austin Clifton EEN 315 – Group 2

For : Sayan Maity, TA University of Miami November 18, 2013

Project Goals

- Understand Shift Registers
- Reinforce sequential design
- Introduce concepts of testing, signature generation, and pseudo random binary sequence generation

Move away from physical breadboard implementation to digital design

Background Information

A Flip-Flop is the basic memory unit of a sequential circuit
Types – Set/Reset, D, JK, Edge Triggered (clock) vs. Latch (signal level)

Signatures are used to verify correct transfer of data

•Basic logic diagrams, truth table construction, Next State Tables

Understanding of multiplexors

Background Information

- Basic understanding of truth tables, logic diagrams, transition tables, and next state tables is essential to understanding logic derivations
- Truth table for a JK flip-flop:

	Q(t+1)	ĸ	J
No Change	Q(t)	0	0
Reset	0	1	0
Set	1	0	1
Complement	Q'(t)	1	1

- Equation for a JK flip-flop: Q⁺ = JQ' + K'Q
- Bit shifting: 0110 shifted left = 1100, 1011 shifted right = 0101, etc.

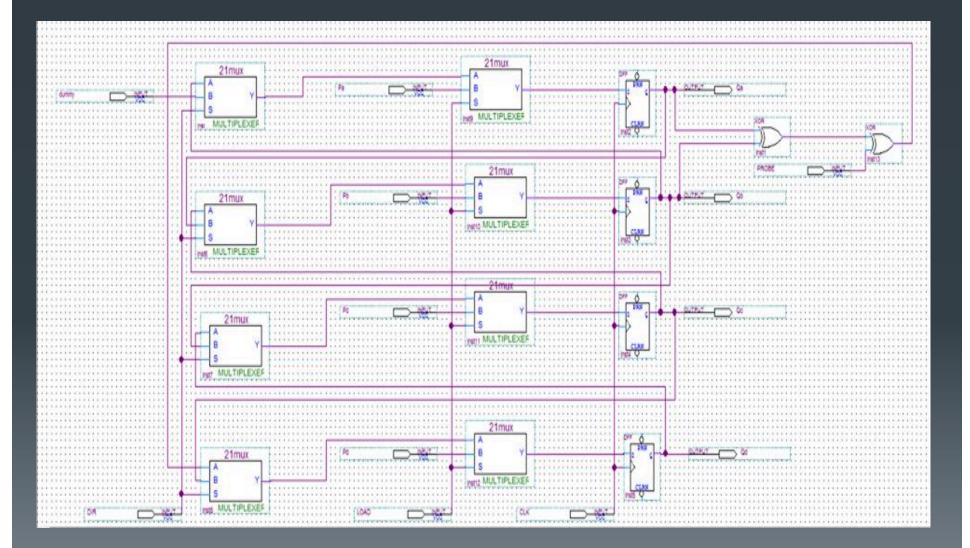
Equipment

DESCRIPTION	QUANTITY
2:1 Multiplexer	8
D Flip Flop	4
XOR Gate	2

Design Process

- Each Flip Flop represents a bit in the register, output of one is input of the next
- 2:1 MUX for each Flip Flop determines Parallel or Serial Input
- 2:1 MUX for each Flip Flop determines Right or Left Shift
- Parallel output is the output of the Flip Flops
- Signature Analyzer: XOR the two least significant bits Qd+ = (Qa XOR Qb) XOR PROBE Qc+ = Qd Qb+ = Qc Qa+ = Qb
- After 10 clock cycles, will have final signature

Logic Diagram



Implementation

- Design Block Diagram File
- Compilation
- Input values for Vector Waveform File
- Simulation

Qd	Qc	Qb	Qa	PROBE(INPUT)	Qd+	Qc+	Qb+	Qa+
0	0	0	0	1	1	0	0	0
1	0	0	0	0	0	1	0	0
0	1	0	0	1	1	0	1	0
1	0	1	0	0	1	1	0	1
1	1	0	1	1	0	1	1	0
0	1	1	0	0	1	0	1	1
1	0	1	1	1	1	1	0	1
1	1	0	1	0	1	1	1	0
1	1	1	0	1	0	1	1	1
0	1	1	1	0	0	0	1	1

Sequence 1 Simulation

Simu	ulation W	/aveforms																				
Simu	ulation mo	de: Timing																				
A	Master T	ime Bar:		110.0 n	\$		• • Po	pinter:		21.	68 ns			Interval:		-8	8.32 ns			Star	t	
A ⊛		Name	Value at 110.0 ns	0 ps	10.0 ns	20.(0 ns	30.0 ns	40.	0 ns	50.0 n	s	60.0 ns	70.	0 ns	80.0 ns	90.	0 ns	100	0 ns	110 _/ 110.(
€ ₽	₽ 0 ₽ 1	CLK dummy	A 0 A 0																			
#	₽ 2	DIR	A 1																			
M_	→3 →4	PROBE LOAD	A 1 A 1	<u> </u>]]												
→	₽ 5	Pa	A 0																			
暻	₽ 6	Pb	A 0	<u> </u>																		
₽↓	₽7	Pc Pd	A 0 A 0	<u> </u>																	-	
	@ 9	Qd	A 0																			
	 10 11 	Qc Qb	A 0 A 1	<u> </u>							г									1	_	
	12	Qa	A1																		-	

Qd	Qc	Qb	Qa	PROBE(INPUT)	Qd+	Qc+	Qb+	Qa+
0	0	0	0	1	1	0	0	0
1	0	0	0	1	1	1	0	0
1	1	0	0	0	0	1	1	0
0	1	1	0	0	1	0	1	1
1	0	1	1	1	1	1	0	1
1	1	0	1	1	0	1	1	0
0	1	1	0	0	1	0	1	1
1	0	1	1	0	0	1	0	1
0	1	0	1	1	0	0	1	0
0	0	1	0	1	0	0	0	1

Sequence 2 Simulation

Master T	ime Bar:					 Image: Pointer: 				.02 ns		Interval:		-10)2.98 ns		Start:		
	Name	Value at 110.0 ns	0 ps	10.0 ns	20.	0 ns	30.0 ns	s 4().0 ns	50.() ns	60.0 ns	70.0 ns	80.0 ns	90.0 ns	100,0 n		0 ns 0 ns	1
₽0	CLK	A 0	\mid																
1 ₪	dummy	A 0	<u> </u>			1													
<mark>⊪</mark> 2	DIR	A 1																	
₽3	PROBE	A 1																	
₫≥4	LOAD	A 1																	
₽5	Pa	A 0				1													
₽6	Pb	A 0																	
₽7	Pc	A 0																	
₽8	Pd	A 0																	_
@ 9	Qd	A 0																	_
🐵 10	Qc	A 0																	
@11	Qb	A 0																	
12	Qa	A 1																	

Qd	Qc	Qb	Qa	PROBE(INPUT)	Qd+	Qc+	Qb+	Qa+
0	0	0	0	1	1	0	0	0
1	0	0	0	1	1	1	0	0
1	1	0	0	1	1	1	1	0
0	1	1	0	1	0	1	1	1
0	1	1	1	0	0	0	1	1
0	0	1	1	0	0	0	0	1
0	0	0	1	0	1	0	0	0
1	0	0	0	0	0	1	0	0
0	1	0	0	1	1	0	1	0
1	0	1	0	1	0	1	0	1

Sequence 3 Simulation

		aveforms																								
Simu	mulation mode: Timing																									
								_ , ,								_	_						_	_		=
\mathbf{k}															Inter	val:		-1	05.88 n	ns Start:						
Α			Value at	0 ps	1(0.0 ns	2	20.0 ns	30).0 ns	40	0 ns	50.0	ns	60.0	ns	70.0 n	s	80.0 ns	90).0 ns	100	0 ns	110,	0 ns	12
Æ		Name	110.0 ns																					110.0	0 ns	
Ð,	₽0	CLK	A O			1				1		1 1		Γ		Γ					1		1 1	Ŧ	<u>י</u> ר	=
Þà	⊡ 1	dummy	A 0												1	_										
桷	₽ 2	DIR	A 1																					_		
	₫>3	PROBE	A 1																							
<u>w</u>	₫≥4	LOAD	A 1																							
→	₽5	Pa	A 0	⊨																						_
巺	₽ 6	РЬ	A 0	⊢																						_
₽↓		Pc	A O	⊢																						-
	₽8 ₽	Pd Qd	A 0 A 0	⊢						-													-			-
	10	Qc	A1	\vdash		-		_						7											7	
	@11	Qb	AO					+								1							Ë,			=
	12	Qa	A 1																					_	í.	

Qd	Qc	Qb	Qa	PROBE(INPUT)	Qd+	Qc+	Qb+	Qa+
0	0	0	0	1	1	0	0	0
1	0	0	0	1	1	1	0	0
1	1	0	0	1	1	1	1	0
1	1	1	0	1	0	1	1	1
0	1	1	1	1	1	0	1	1
1	0	1	1	1	1	1	0	1
1	1	0	1	1	0	1	1	0
0	1	1	0	1	0	0	1	1
0	0	1	1	0	0	0	0	1
0	0	0	1	0	1	0	0	0

Sequence 4 Simulation

Simulation Waveforms Simulation mode: Timing

	Master T	ime Bar:		110.0 n	s	••	Pointer:	4	.58 ns	lr	nterval:	-105	i.42 ns	Star	t
A ⊛		Name	Value at 110.0 ns	0 ps	10.0 ns	20.0 ns	30.0 ns	40.0 ns	50.0 ns	60.0 ns	70.0 ns	80.0 ns	90.0 ns	100 <u>.</u> 0 ns	110 <u>,</u> 0 ns 110.0 ns
€ ħ	⊪ 0 ⊪ 1	CLK dummy	A 0 A 0												
М	2 →2 3	DIR PROBE	A 1 A 0												
# <u></u> , →	→4 →5 →	LOAD Pa	A 1 A 0												
際 ₽↓		Pb Pc Pd	A 0 A 0 A 0												
		Qd Qc	A 1 A 0								1				
	 11 12 	Qb Qa	A O A O												

Conclusions

- Learned how to use Quartus software to design and implement circuits with simulations.
- Mistakes made initially due to lack of familiarity with Quartus software were easily found and fixed as we continued to accustom to wiring a circuit digitally, and we eventually found that wiring digitally is even easier than wiring manually on a bread board.
- Successfully implemented a shift register and shift analyzer without any major issues.