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:

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

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

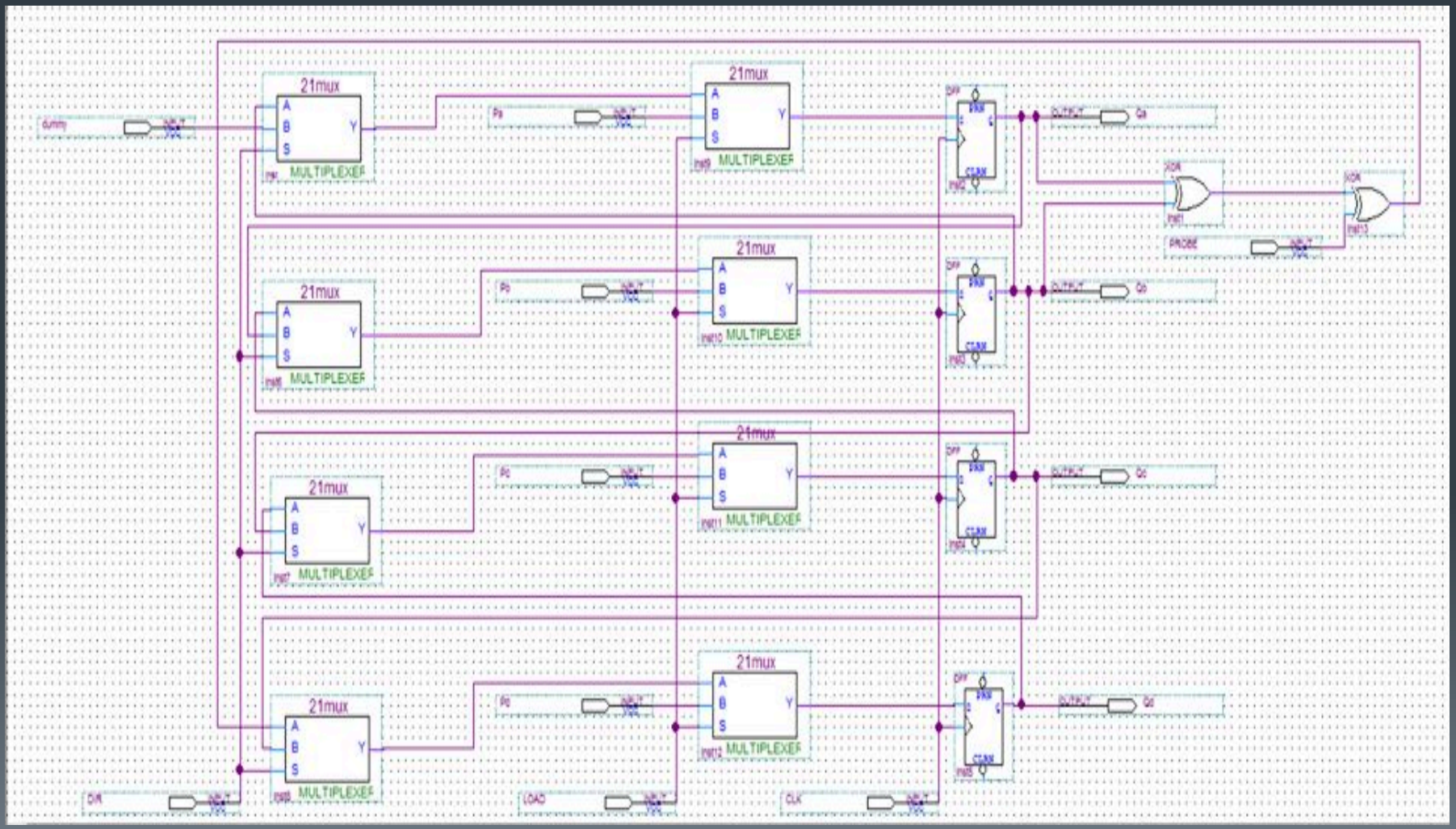
Equipment

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



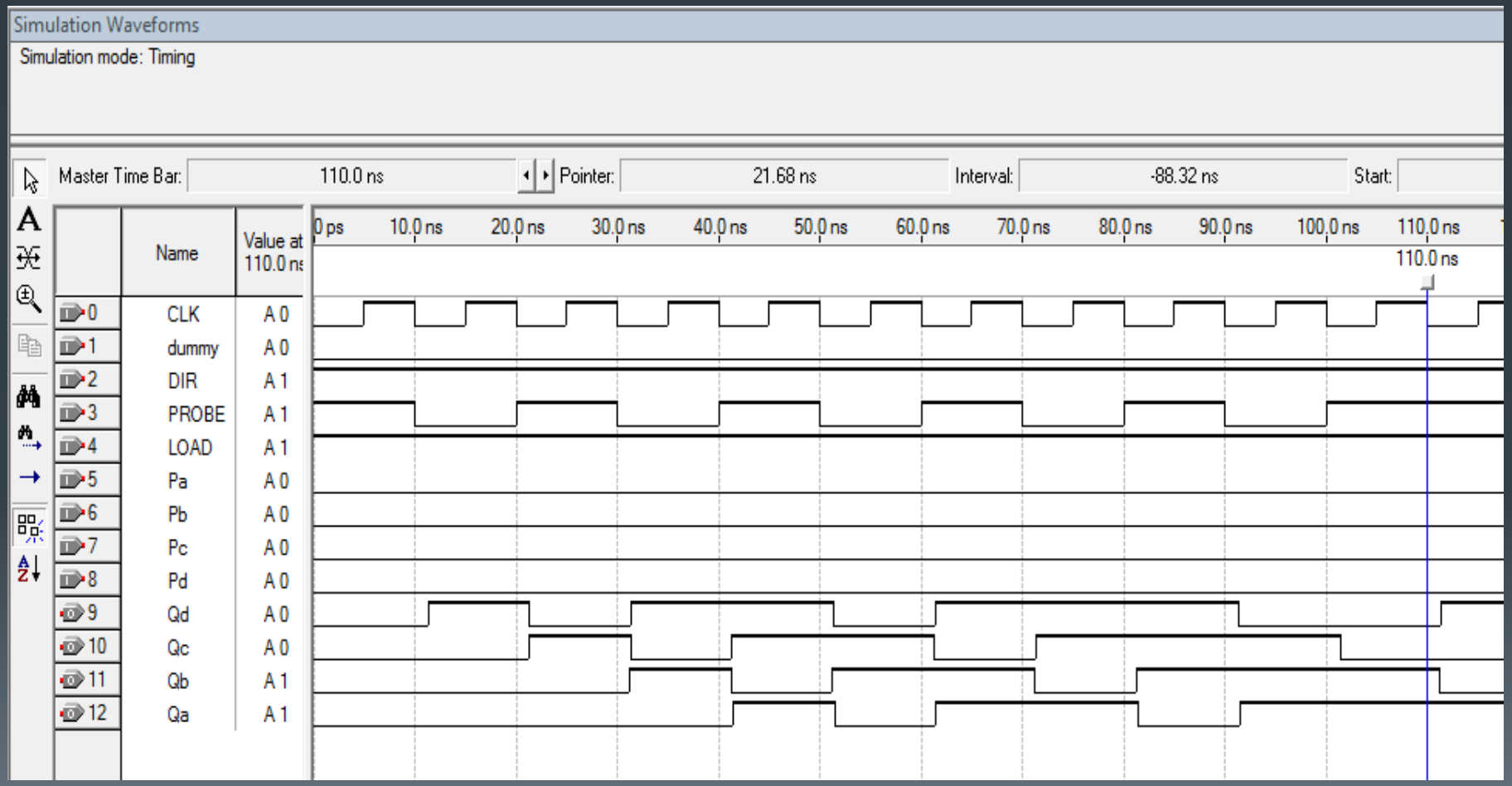
Implementation

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

Sequence 1

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



Sequence 2

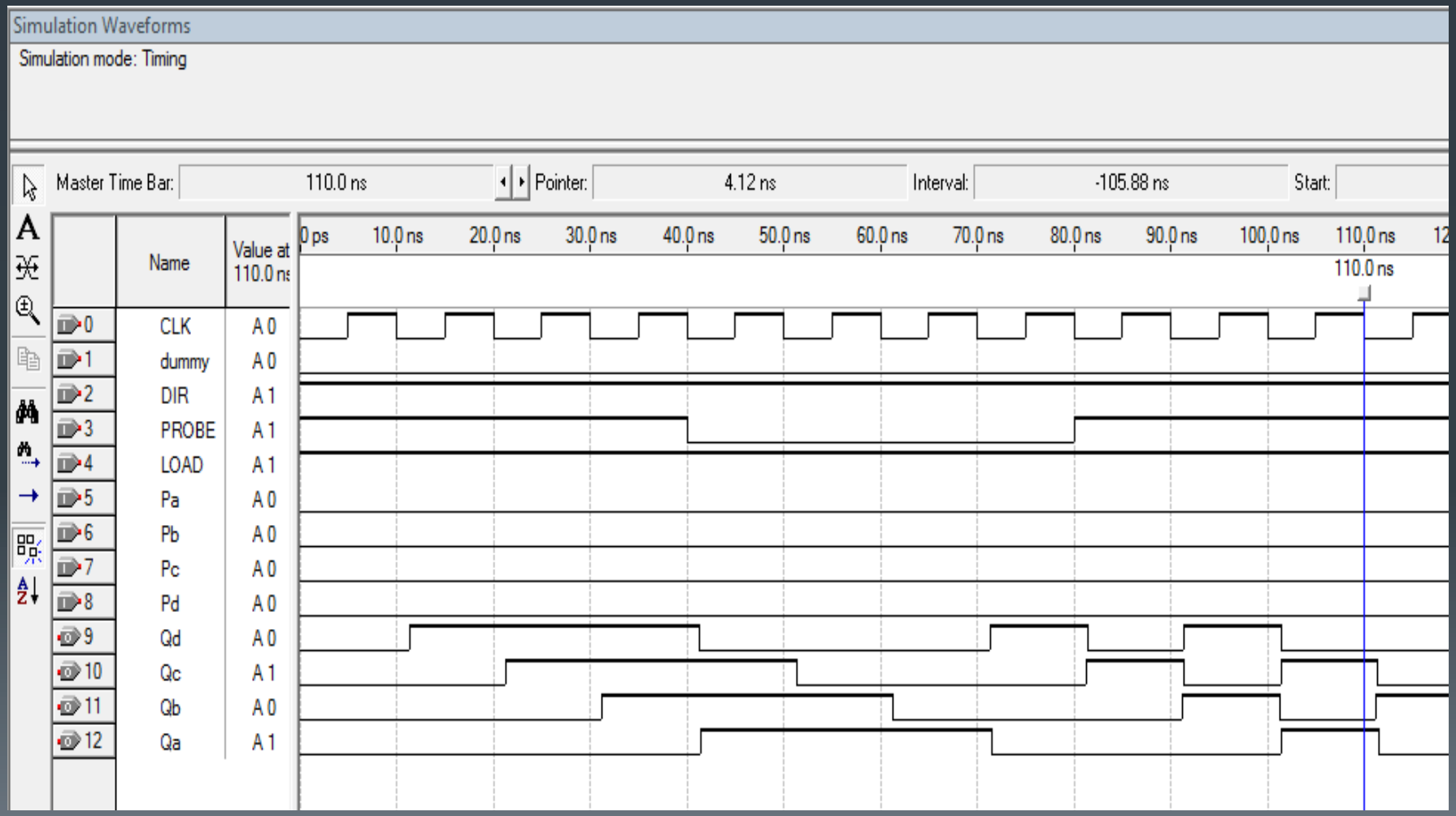
Qd	Qc	Qb	Qa	PROBE(INPUT)	Qd+	Qc+	Qb+	Qa+
0	0	0	0	1	1	0	0	0
1	0	0	0	1	1	1	0	0
1	1	0	0	0	0	1	1	0
0	1	1	0	0	1	0	1	1
1	0	1	1	1	1	1	0	1
1	1	0	1	1	0	1	1	0
0	1	1	0	0	1	0	1	1
1	0	1	1	0	0	1	0	1
0	1	0	1	1	0	0	1	0
0	0	1	0	1	**0**	**0**	**0**	**1**

Sequence 2 Simulation

Sequence 3

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**

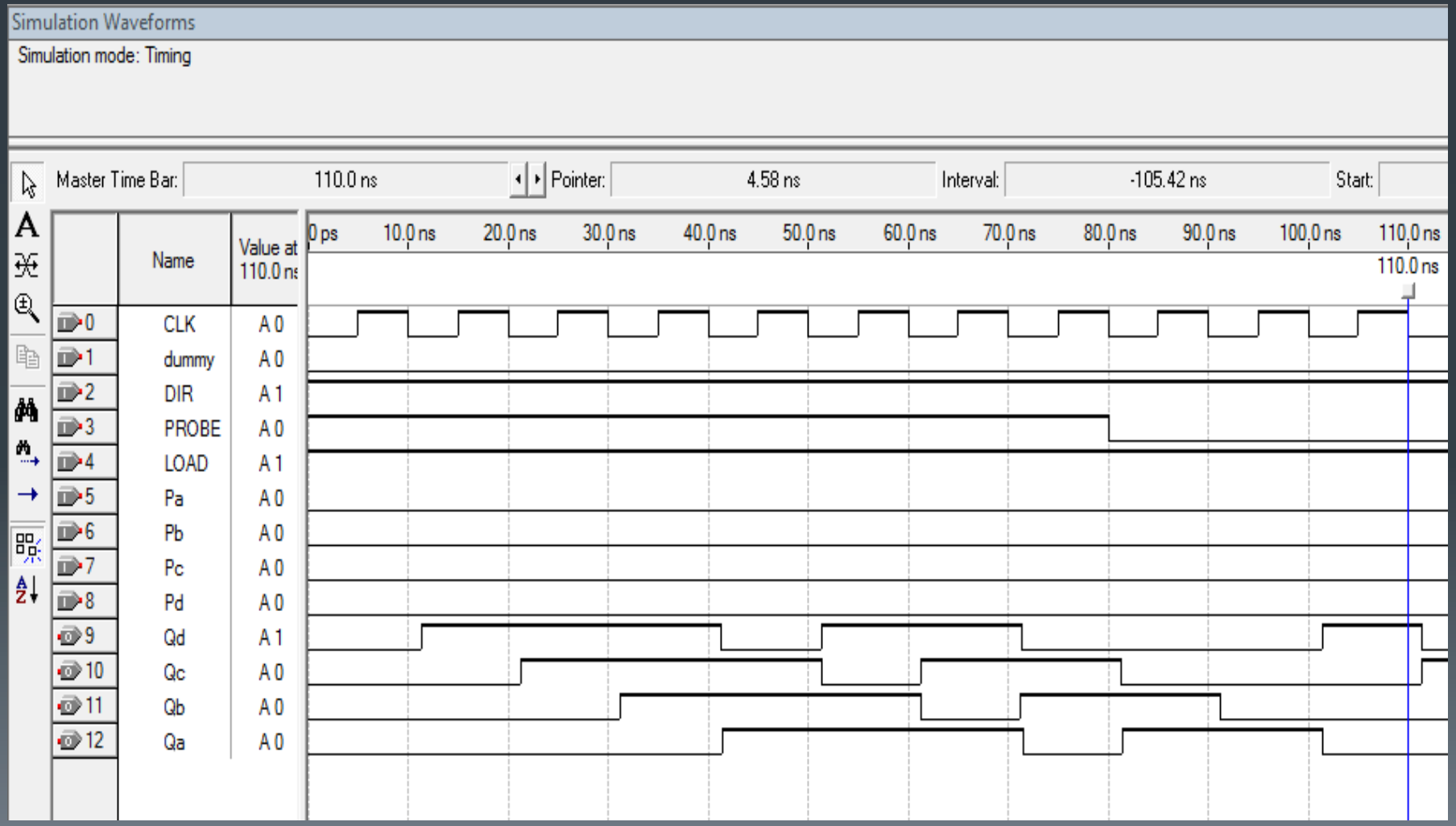
Sequence 3 Simulation



Sequence 4

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



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.