KAUNAS UNIVERSITY OF TECHNOLOGY

THE FIRST PRPICNICPLE OF DIGITAL LOGIC

LAB 3: Registers

Student name: Zalal Youssef

Professor: KAZIMERAS BAGDONAS

Kaunas,2022

TASK: **Bit length**: 8
Shift Operations: LR2, CR1, AL1
Input information: 1
Representation: two’s complement

OBJECTIVE:

To Analyze shift register operations and the way they are executed, and to Design a universal register. It’s bit length is specified according to the individual task. Using the truth table of functions, also designing a specialized shift register witch executes shift operations, given in the individual assignment task. Using of multiplexers in the design. And finally designing tests for the schematics and test their functionality.

Theory:

In digital logic a register is a device that stores several bits of information. They are a group of flip-flops connected together using some sort of control circuit. According to the way data is written and read to and out of the registers, they can be divided into several groups:

- Parallel registers;
- Shift registers;
- Universal registers.

Each bit stored in register has it index. In this book the least significant bit (LSB) has index 0 and is considered to be rightmost. It is due to the convention of writing less significant bits further to the right.

- **Parallel Registers:**

  The transfer of new information into the register is referred to as loading the register. If all bits of the register are loaded simultaneously with a single clock pulse, we say that the loading is done in parallel. Parallel registers have n flip-flops, and control circuit for parallel data write and read. Since a flip flop can hold one bit of information, number n defines register storage capacity.

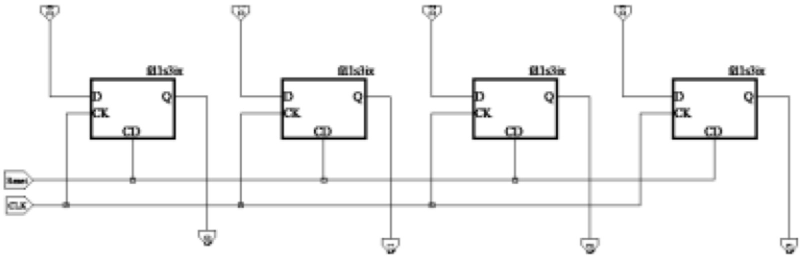


Figure 4.1: Three bit parallel register a), symbol b)

Reset input R sets the register into an initial state by writing logic 0 to all of the flip-flops. Input signal ID controls the way data is written to the register. If ID = 1 data is written in a Sign-and-Magnitude signed number representation, otherwise in a one’s complement (see 4.2). In a similar way, input signal RD controls the way data is read out from the register. If RD = 1 data is read out in a Sign-and-Magnitude representation, otherwise in one’s complement.

- **Shift Registers:**

  A shift register is a group of flip-flops connected to one another in a chain so that the output of one flipflop becomes the input of the next one. In this case every time a clock pulse is applied, each bit is moving from left to right, this operation is called logic shift to the right (LR). The schematic of the shift register is shown in Figure 4.2.

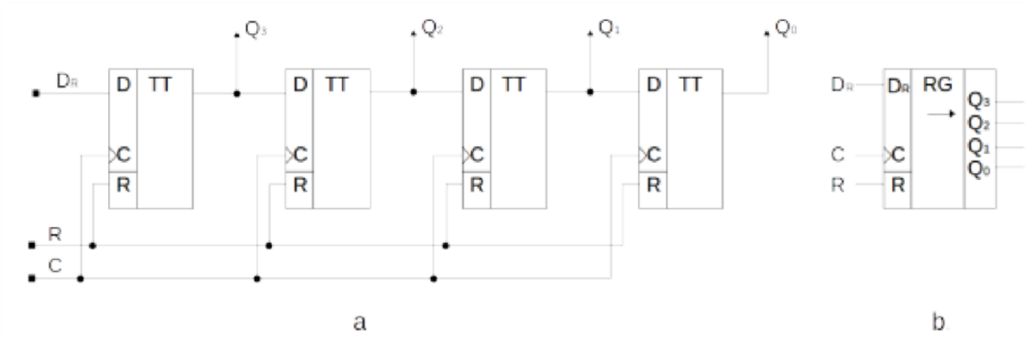
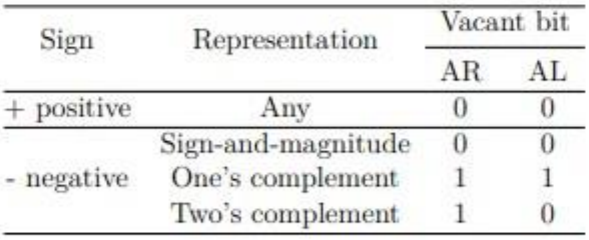


Figure 4.2: 4 bit logic right shift register a), symbol b)

- **Signed number Representation:**
  In computing, signed number representations are required to encode negative numbers in binary number systems. In mathematics, negative numbers are represented by prefixing them with a ”-” sign. However, in digital logic, numbers are represented in bit vectors only, without extra symbols. Three best-known methods to represent signed numbers are:
  1. Sign-and-magnitude;
  2. Ones’ complement;
  3. Two’s complement.

  At present the most widely used representation is two’s complement.

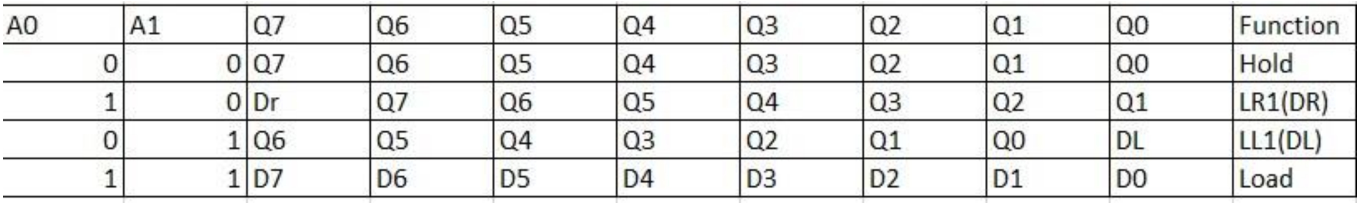
  Table of the signed numbers:



Sign	Representation	Vacant bit	
		AR	AL
+ positive	Any	0	0
- negative	Sign-and-magnitude	0	0
	One’s complement	1	1
	Two’s complement	1	0

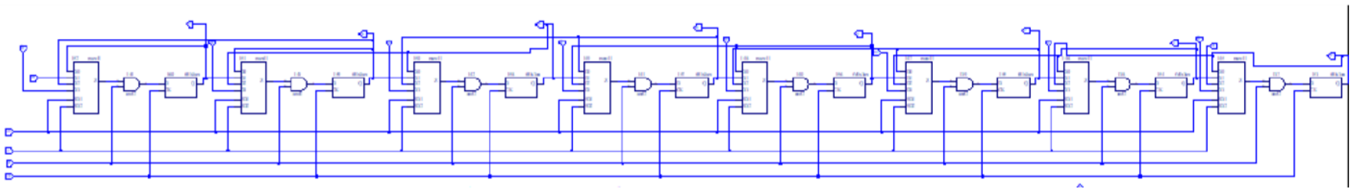
PROCEDURE:

We design a function table for the register:

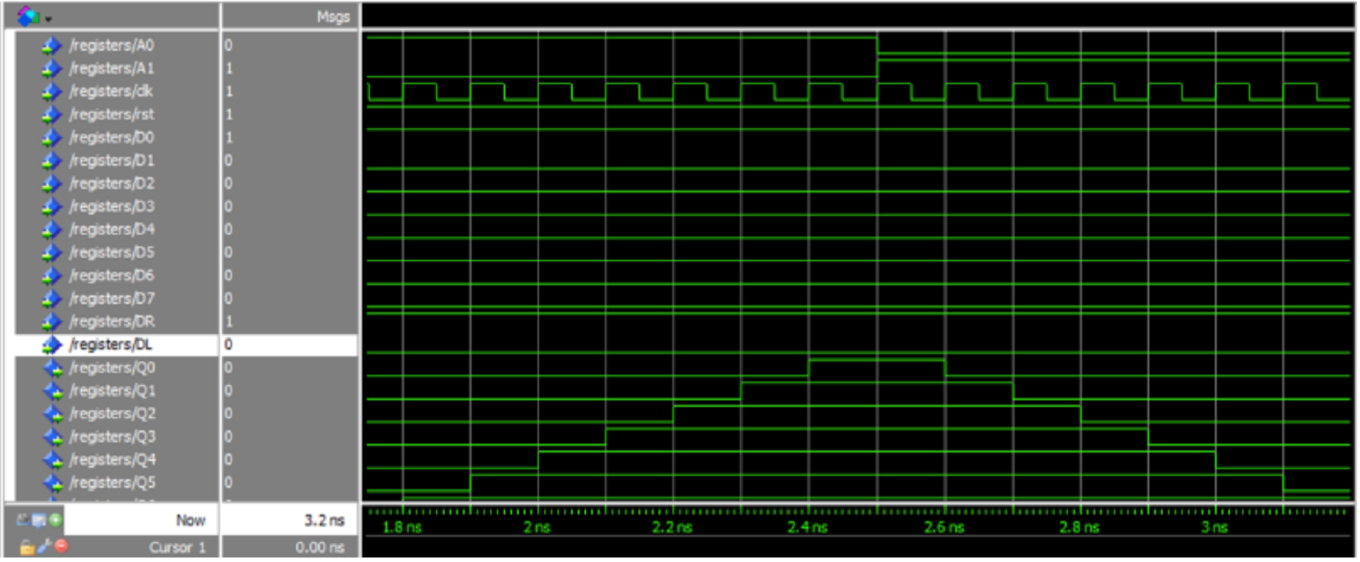


A0	A1	Q7	Q6	Q5	Q4	Q3	Q2	Q1	Q0	Function
0	0	Q7	Q6	Q5	Q4	Q3	Q2	Q1	Q0	Hold
1	0	Dr	Q7	Q6	Q5	Q4	Q3	Q2	Q1	LR1(DR)
0	1	Q6	Q5	Q4	Q3	Q2	Q1	Q0	DL	LL1(DL)
1	1	D7	D6	D5	D4	D3	D2	D1	D0	Load

The register has to execute 4 shift operations, therefore we will use multiplexers. Signals A1, A0 are connected to the multiplexer selector inputs. Data inputs are connected to signals according to the register function table.
For the asynchronous reset, use D Flip-Flop Asynchronuos clear fd1s3dx
For asynchronous reset only the appropriate flip-flops are required. The schematic of the universal register is presented below:



We design a test for testing all of the register’s functions. We get timing diagrams, presented below:

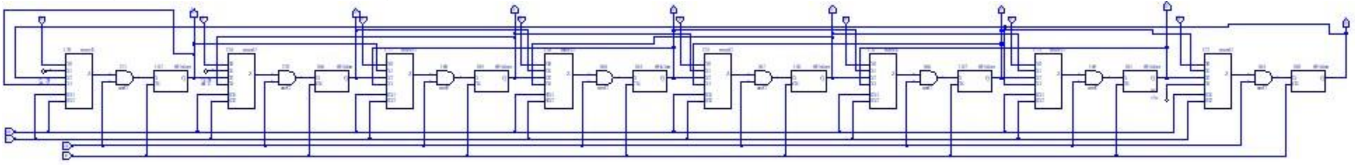


	Msgs
/registers/A0	0
/registers/A1	1
/registers/clk	1
/registers/rst	1
/registers/D0	1
/registers/D1	0
/registers/D2	0
/registers/D3	0
/registers/D4	0
/registers/D5	0
/registers/D6	0
/registers/D7	0
/registers/DR	1
/registers/DL	0
/registers/Q0	0
/registers/Q1	0
/registers/Q2	0
/registers/Q3	0
/registers/Q4	0
/registers/Q5	0
Now	3.2 ns
Cursor 1	0.00 ns

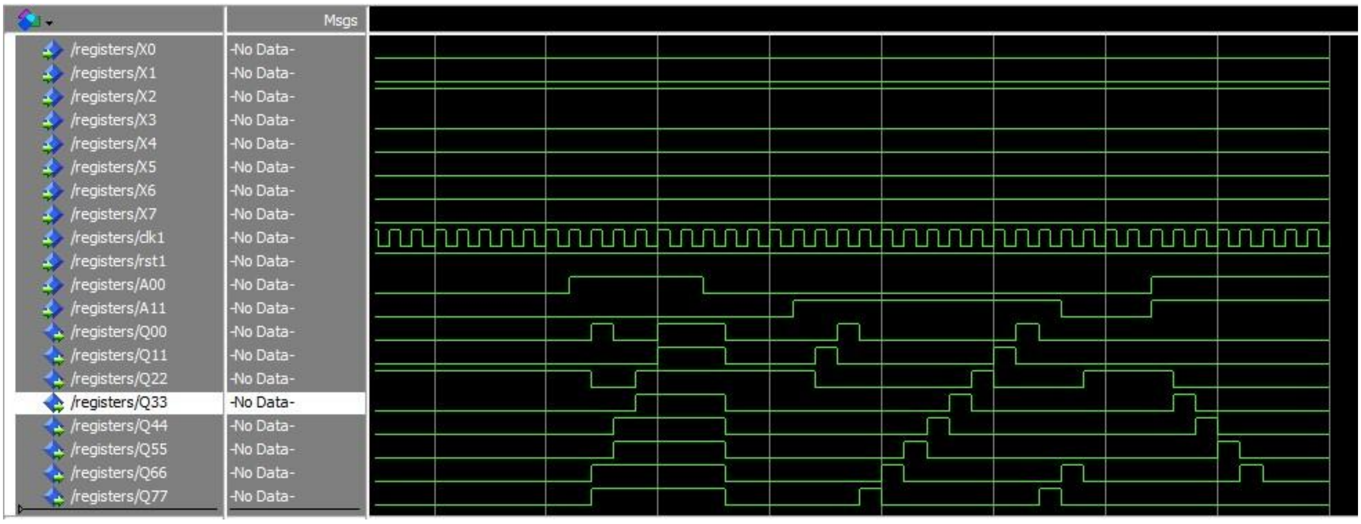
When designing a specialized register, which executes the following shift operations LR2, input data: 1, CR1 and AL2, we will use multiplexers with 4 inputs and flip-flops. Multiplexers are controlled by signals A0 and A1. These inputs control, which operation is being executed. We draw a table with the information needed to design the schematic

A0	A1	D7	D6	D5	D4	D3	D2	D1	D0	Function
0	0	x7	x6	x5	x4	x3	x2	x1	x0	Data Load
0	1	1	1	Q7	Q6	Q5	Q4	Q3	Q2	LR2(1)
1	0	Q0	Q7	Q6	Q5	Q4	Q3	Q2	Q1	CR1
1	1	Q7	Q5	Q4	Q3	Q2	Q1	Q0	0	AL1(2's)

This design is implemented using multiplexers, and presented below:



We design a test for testing all of the register’s functions. We get timing diagrams, presented below:



	Msgs
/registers/X0	-No Data-
/registers/X1	-No Data-
/registers/X2	-No Data-
/registers/X3	-No Data-
/registers/X4	-No Data-
/registers/X5	-No Data-
/registers/X6	-No Data-
/registers/X7	-No Data-
/registers/clk1	-No Data-
/registers/rst1	-No Data-
/registers/A00	-No Data-
/registers/A11	-No Data-
/registers/Q00	-No Data-
/registers/Q11	-No Data-
/registers/Q22	-No Data-
/registers/Q33	-No Data-
/registers/Q44	-No Data-
/registers/Q55	-No Data-
/registers/Q66	-No Data-
/registers/Q77	-No Data-

By examining the timing diagrams, we note that the register executes all of the operations required: LL2, CR1, AL2.

CONCLUSION:

To conclude, a register is a device that stores several bits of information. They are a group of flip-flops connected together using some sort of control circuit. According to the way data is written and read to and out of the registers.

They perform many types of operations, and they are used in for several purposes, as in computers, to quickly accept, store, and transfer data and instructions that are being used immediately by the CPU.

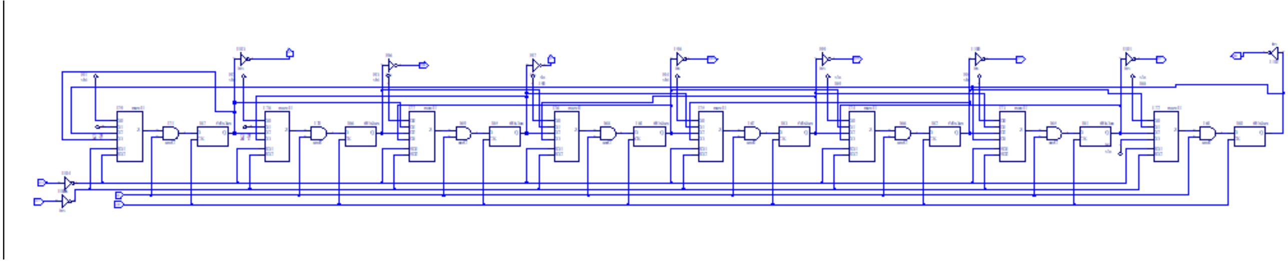
FPGA implementation:

In order for your circuits to function on FPGA board some minor modifications must be performed. The list of modifications is provided below.

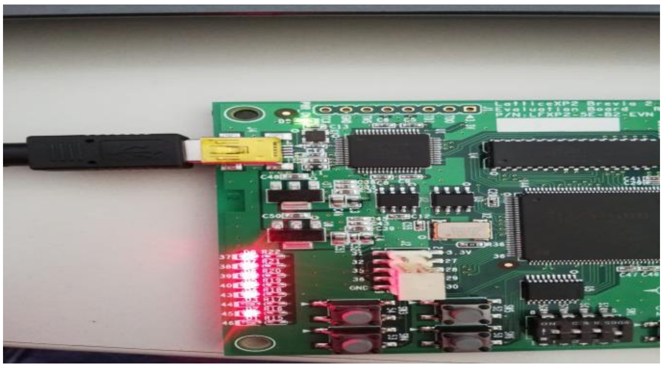
1. All outputs must be inverted
2. Asynchronous reset signal must be inverted, synchronous must not be inverted
3. All parallel data inputs must be exchange into Vhi or Vlo components

Modified schematics:

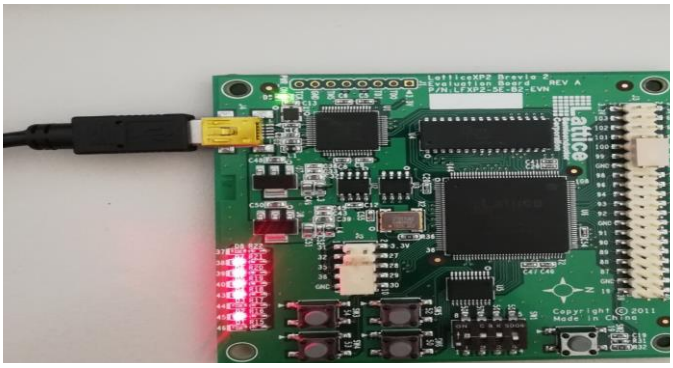
Specialized:



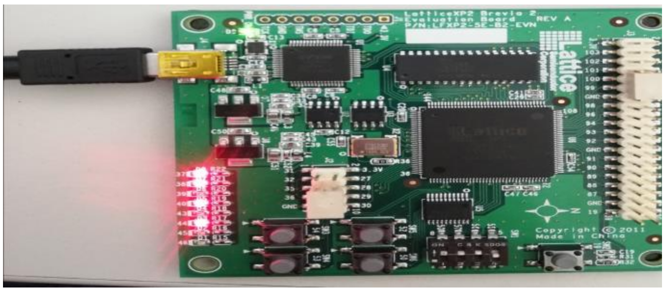
FPGA board results:



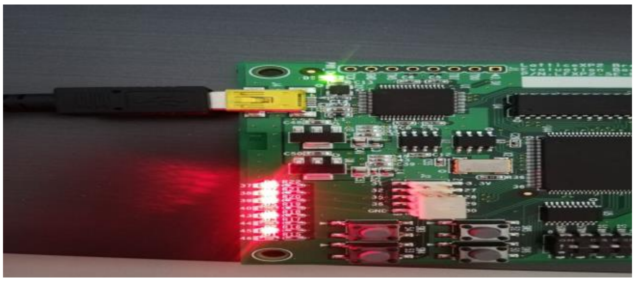
LR2



CR1

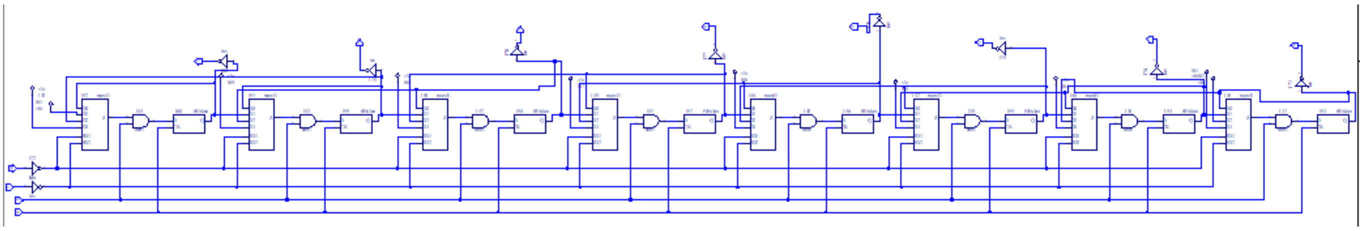


AL1

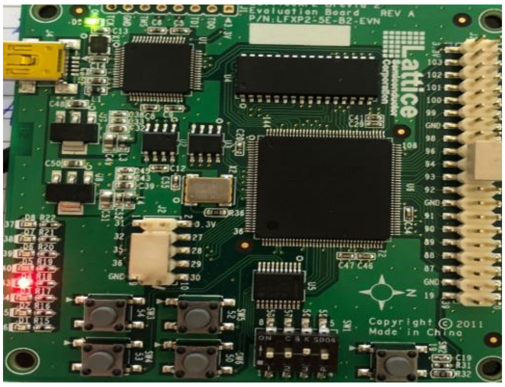


Data LOAD

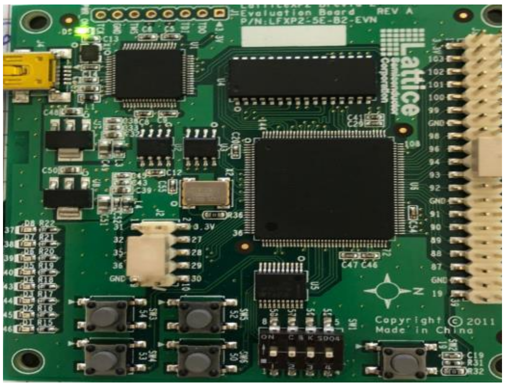
Universal modified schematic:



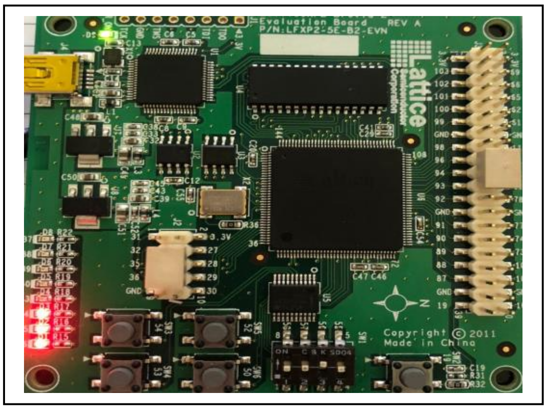
FPGA board results:



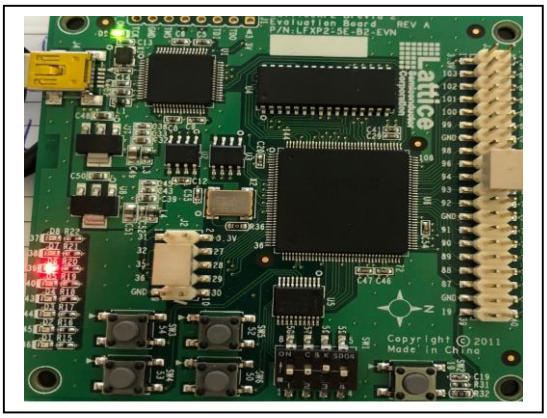
LL2



SAVE


Evaluation Board REV A
P/N:LFXP2-5E-B2-EVN
Lattice
Copyright © 2011
Made in China

LR1


Evaluation Board REV A
P/N:LFXP2-5E-B2-EVN
Lattice
Copyright © 2011
Made in China

LOAD