**A Mini-Project Report**

**On**

**1 Bit ALU (Arithmetic Logic Unit)**

*Submitted in partial fulfilment of the requirements for the*

*Term-work of Subject*

*VLSI Design*

**Bachelor of Engineering**

*in*

**Electronics Engineering**

*by*

**Mr. Omkar Bhave(Reg. No. N-18-0303)**

**Mr. Parth Sawant(Reg. No. N-18-0273)**

*Under the Guidance of*

**Ms.V.V.Nimbalkar**



Department of Electronics Engineering

Finolex Academy of Management & Technology, Ratnagiri

YEAR 2020-2021

**ABSTRACT**

At its most fundamental level, a computer consists of a control unit, an arithmetic logic unit (ALU), a memory unit, and input/output (I/O) controllers. The ALU performs simple addition, subtraction, multiplication, division, and logic operations, such as OR & AND.

We have designed the Adder using the CMOS logic, AND & XOR using Transmission gate logic and NOR using the Pass Transistor logic. The following chapters consider every part of the ALU (i.e. Adder, AND, NOR and XOR) gates with their respective logic. The last chapter is based on the final output obtained when the above circuitry is combined to for an ALU and the obtained result is correct.

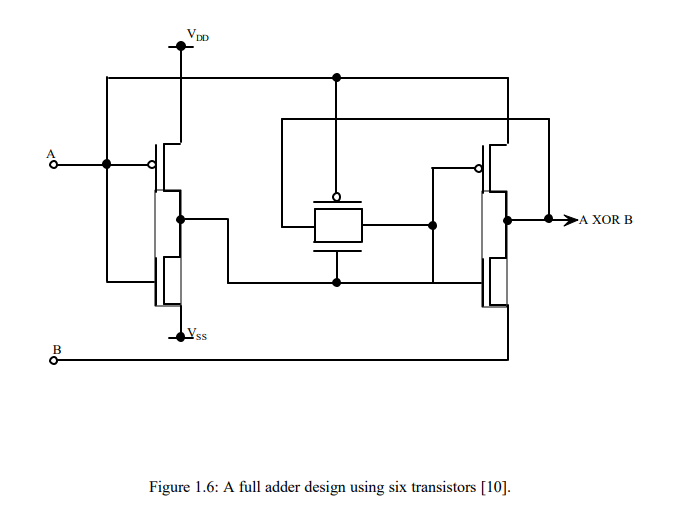
|  |  |
| --- | --- |
| **Table of Content** | |
|  |  |
| **Chapter. 1: Introduction** | …4 |
| **Chapter. 2: Project Design** | …6 |
| **Chapter. 3: Results** | …21 |
| **Chapter. 4: Applications Advantages & Future Scope** | …24 |
|  | |

# Chapter 1: Introduction

# 1.1. Problem Statement

Design 1 bit ALU (Arithmetic Logic Unit).

# 1.2. Literature Review

In the past, ALU and full adder circuits have been implemented for optimum area and delay, each with their distinct features that bring about optimum area and delay. Some of them have been briefly described below to give us an idea of earlier work and shed light on different optimization techniques. Past work related to multiple-input floating gate CMOS applications has been reviewed to give us an idea about its operation, design and simulation issues. Bui et al [9] have designed a low power 10-transistor full adder called Static Energy-Recovery Full-Adder (SERF) using 10 transistors. A novel set of XOR and XNOR gates in combination with existing ones have been used. The XOR and XNOR circuits designed by them do not directly connect to power and ground lines, respectively. It uses four transistors for XOR and XNOR gates in CMOS as shown in Fig. 1.5. Wang et al [10] have shown an improved version of XOR and XNOR gates that make use of six transistors as shown in Fig. 1.6. In another design of CMOS 1-bit full adder cell, four transistor XOR and XNOR gates have been used [11]. The cell offers higher speed and lesser power consumption than standard 1-bit full adder cell. Radhakrishnan [12] has presented the design of low power CMOS full adder circuits using transmission function theory. This design uses six transistor CMOS XOR and XNOR gates as shown in Fig. 1.7. Figure 1.8 shows a 16-bit, 2.4ns, 0.5mm CMOS ALU design [13], which consists of a logical and arithmetic unit (LAU), a magnitude comparator (CMP), an overflow detector (OVF) and zero flag detector (ZERO). The ALU employs a binary look-ahead carry (BLC) adder. All units in this design operate in parallel and high speed is achieved.

Advancement in VLSI technology has allowed following Moore’s law [1] for doubling component density on a silicon chip after every three years. Though MOS transistors have been scaled down, increased interconnections have limited circuit density on a chip. Furthermore, the size of transistor is limited by hot-carrier phenomena and increase in electric field that lead to degradation of device performance and device lifetime [2]. It has become essential to look into other methods of adding more functionality to a MOS transistor, such as, the multiple-input floating gate MOS transistor structure proposed by Shibata and Ohmi [3]. An enhancement in the basic function of a transistor has, thus, allowed for designs to be implemented using fewer transistors and reduced interconnections. In published literature [4-7], many integrated circuits have been reported which are using multi-input floating gate MOSFETs in standard CMOS process.

# 1.3. Introduction of Project

The arithmetic logic unit (ALU) is the core of a CPU in a computer. The adder cell is the elementary unit of an ALU. The constraints the adder has to satisfy are area, power and speed requirements. Some of the conventional types of adders are ripple-carry adder, carry-look ahead adder, carry-skip adder and Manchester carry chain adder [8]. The delay in an adder is dominated by the carry chain. Carry chain analysis must consider transistor and wiring delays

**1.4. Significance of Project**

1. Can be used to build a small CPU.
2. Can be used to build a calculator.
3. Since its small less power is consumed.
4. It is the building block of all the digital circuits.

# Chapter 2: Project Design

# 2.1. Block Diagram and its working

**Working**

A 1-bit ALU has been designed for 3.0 V operation in which, the full adder design has been implemented using CMOS inverters. The ALU has four stages, each stage consisting of three parts:

a) Full Adder

b) Logical Unit

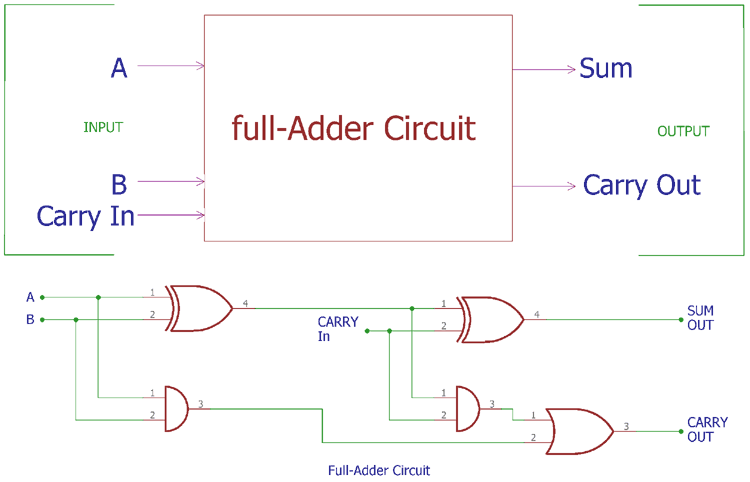
c) Output multiplexers.

The ALU performs only one arithmetic operation, ADD. The four logical operations performed are EXOR, NOR and AND. The input and output sections consist of 4 to 1 and 2 to 1 multiplexers. The multiplexers were designed using the pass transistor logic. A set of three select signals has been incorporated in the design to determine the operation being performed and the inputs and outputs being selected. Figure 3.1 shows the 4-bit ALU with the CARRY bit cascading all the way from first stage to fourth stage. In above diagram, the ALU design consisting of one 4 to 1 multiplexers, one full adder, one AND, one NOR and one XOR gate. The 1-bit ALU was designed in CMOS logic. This chapter explains in detail the 1-bit ALU design. All of the multiplexers have been implemented using pass transistors, and the full adder alone has been designed using CMOS logic. Each stage is discussed in detail in the further sections of this chapter.

# 2.2 Design of 1 bit ALU

# 2.2.1: Design of FA:

**Step1] Block Diagram**



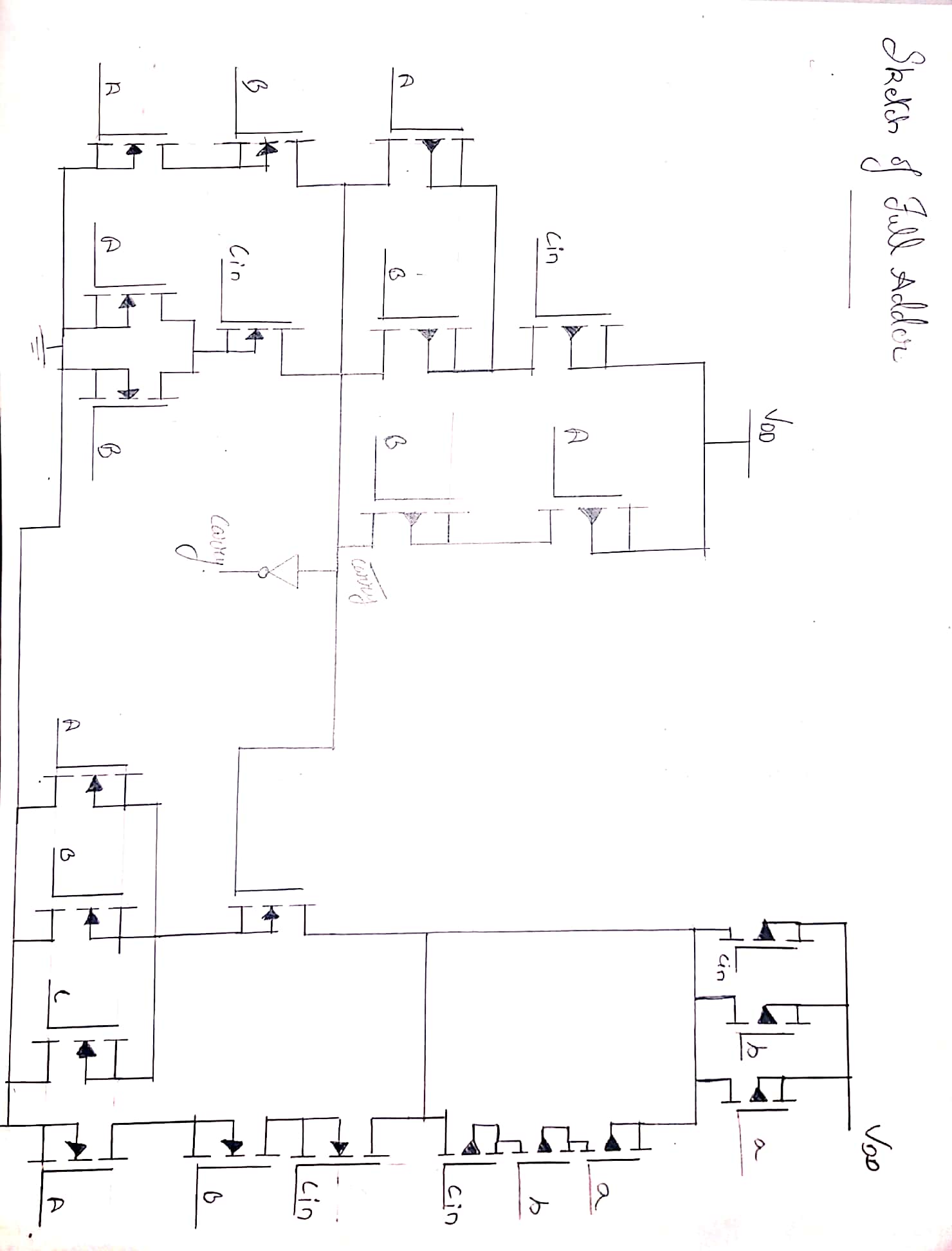
**Step2] Truth Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Carry In** | **A** | **B** | **Sum** | **Carry Out** |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

**Step3]Output Boolean equations**

Sum = ABCin + Carry Out’ (A + B + Cin)

Carry = AB + Cin (A + B)

** Step4]Circuit Diagram**

**Step5]Netlist for FA**

*\*Full Adder*

*.include "D:\Sem VI\VLSI\Project 1Bit ALU\mosmodel.txt"*

*Vdd 1 0 1.8*

*Va 4 0 PULSE 0 1.8 0 1n 1n 10u 20u*

*Vb 5 0 PULSE 0 1.8 0 1n 1n 20u 40u*

*Vcin 6 0 PULSE 0 1.8 0 1n 1n 40u 80u*

*Cl 10 0 100f*

*C2 18 0 100f*

*.model mn NMOS L = 0.18u W = 0.36u*

*.model mp PMOS L = 0.18u W = 0.72u*

*M1 9 4 0 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M2 9 5 0 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M3 2 6 9 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M4 2 4 7 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M5 7 5 0 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M6 2 4 3 1 mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M7 2 5 3 1 mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M8 3 6 1 1 mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M9 8 4 1 1 mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M10 2 5 8 1 mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M11 10 2 0 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M12 10 2 1 1 mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M13 11 4 0 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M14 11 5 0 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M15 11 6 0 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M16 14 2 11 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M17 12 4 0 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M18 13 5 12 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M19 14 6 13 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M20 14 6 15 1 mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M21 15 5 16 1 mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M22 16 4 17 1 mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M23 14 2 17 1 mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M24 17 4 1 1 mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M25 17 5 1 1 mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M26 17 6 1 1 mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M27 18 14 0 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

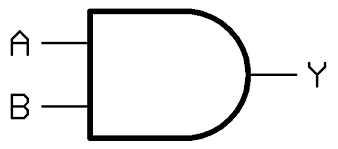
*M28 18 14 1 1 mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*.TRAN 1u 80u*

*.PROBE*

*.END*

# 2.2.2: Design of AND:

**Step1] Block Diagram**

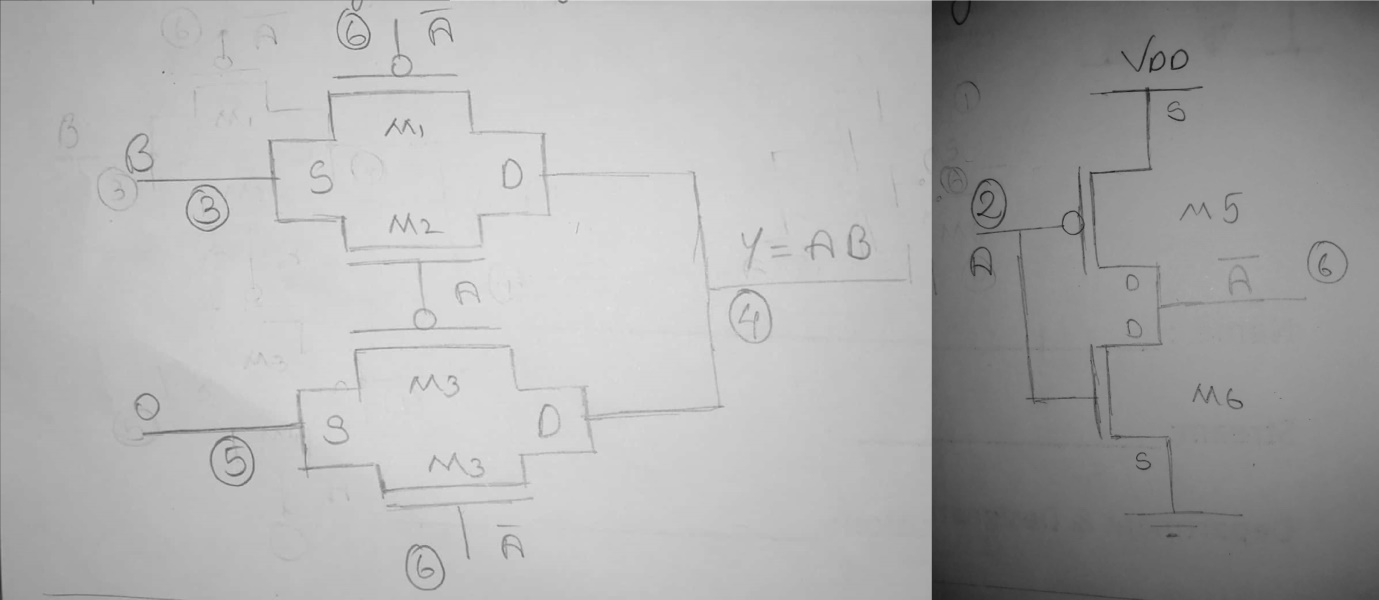
**Step2] Truth Table:**

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **Y** |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**Step3]Output Boolean equations:**

Y = A.B

**Step4] Circuit Diagram:**



**Step5] Netlist for AND:**

*\*AND gate using transmission gate*

*.include "D:\mosmodel.txt"*

*Vdd 1 0 1.8*

*Va 2 0 PULSE 0 1.8 0 1n 1n 10u 20u*

*Vb 3 0 PULSE 0 1.8 5u 1n 1n 20u 40u*

*VI0 4 0 0*

*C1 4 0 100f*

*.model mn NMOS L = 0.18u W = 0.36u*

*.model mp PMOS L = 0.18u W = 0.72u*

*M1 4 6 3 1 mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M2 4 2 3 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M3 4 2 5 1 mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M4 4 6 5 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M5 6 2 1 1 mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M6 6 2 0 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

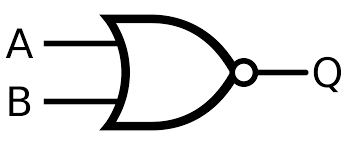
*.TRAN 1u 80u*

*.PROBE*

*.END*

# 2.2.3: Design of NOR:

**Step1] Block Diagram:**

****

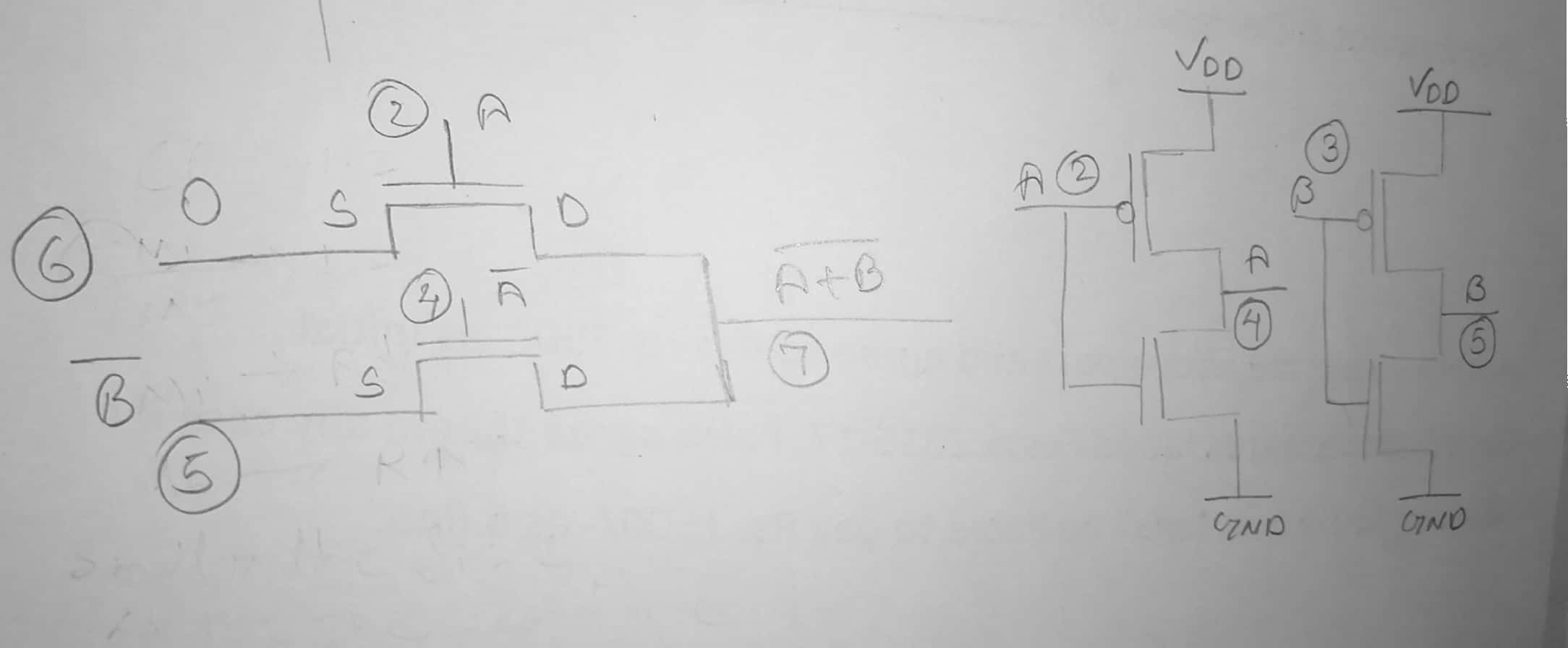
**Step2] Truth Table:**

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **Q** |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

**Step3] Output Boolean equations:**

Q = (A + B)’

**Step4] Circuit Diagram:**

****

**Step5] Netlist for NOR:**

*\*NOR gate*

*.include "D:\mosmodel.txt"*

*Vdd 1 0 1.8*

*Va 2 0 PULSE 0 1.8 0 1n 1n 10u 20u*

*Vb 3 0 PULSE 0 1.8 5u 1n 1n 20u 40u*

*VI0 6 0 0*

*C1 7 0 100f*

*.model mn NMOS L = 0.18u W = 0.36u*

*.model mp PMOS L = 0.18u W = 0.72u*

*M1 7 2 6 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M2 7 4 5 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M3 5 3 1 1 mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M4 5 3 0 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M5 4 2 1 1 mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M6 4 2 0 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

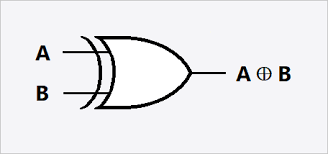
*.TRAN 1u 50u*

*.PROBE*

*.END*

# 2.2.4: Design of XOR:

**Step1] Block Diagram**

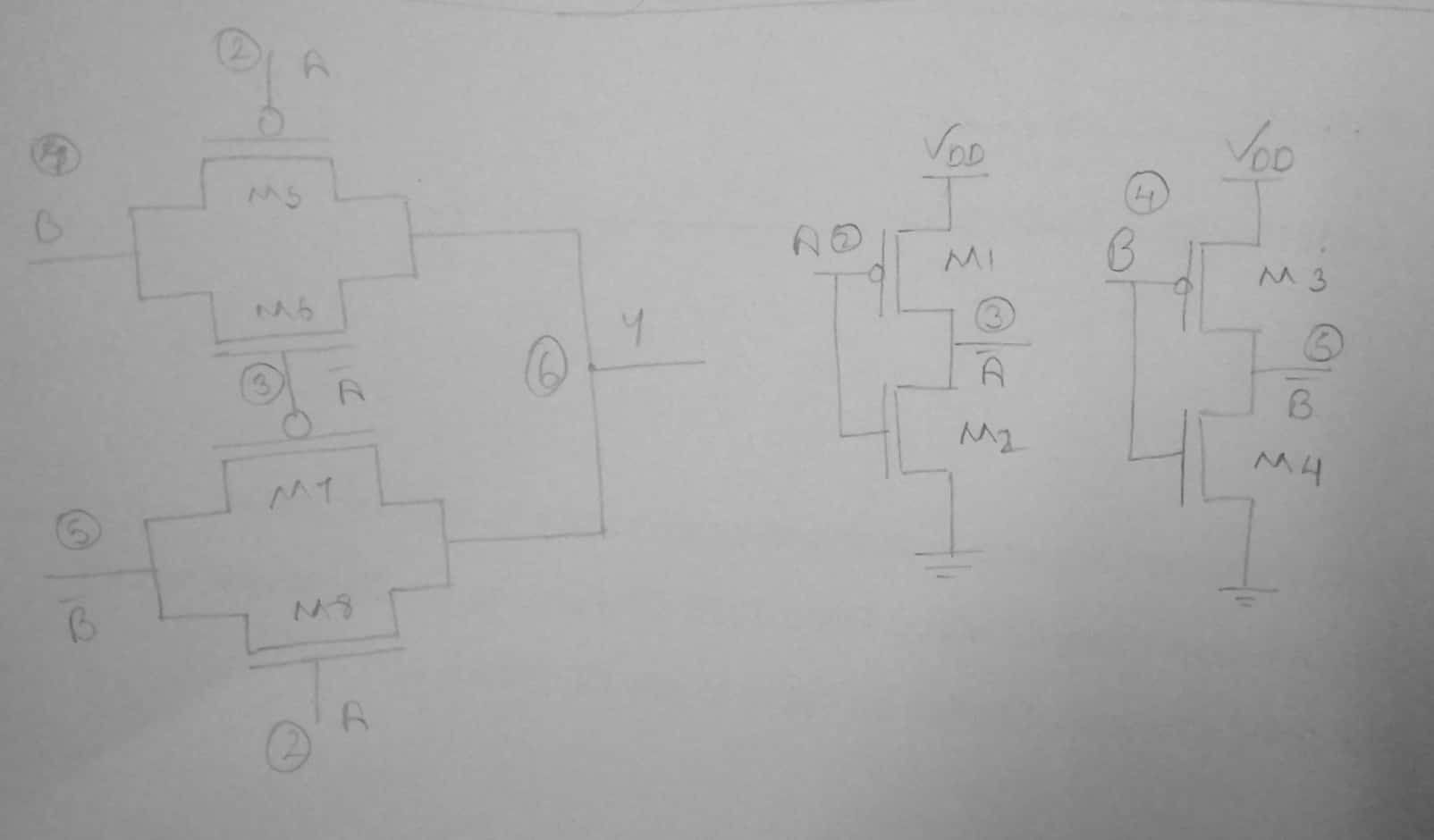
****

**Step2] Truth Table:**

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **Y** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**Step3] Output Boolean equations:**

Y = A’B + AB’

**Step4] Circuit Diagram:**

**Step5] Netlist for XOR:**

*\*XOR Gate using Transmission gate:*

*.include "D:\mosmodel.txt"*

*VDD 1 0 1.8V*

*Va 2 0 PULSE 0 1.8 0 1n 1n 10u 20u*

*Vb 4 0 PULSE 0 1.8 0 1n 1n 20u 40u*

*Cl 6 0 100f*

*.model mn NMOS L=0.18u W=0.36u*

*.model mp PMOS L=0.18u W=0.72u*

*M1 3 2 1 1 mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M2 3 2 0 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M3 5 4 1 1 mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M4 5 4 0 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M5 6 2 4 1 mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M6 6 3 4 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M7 6 3 5 1 mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

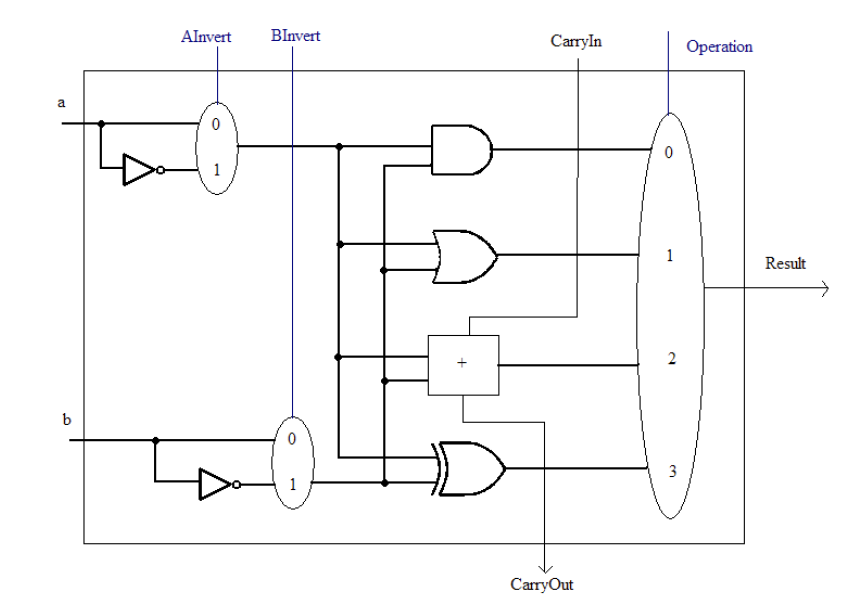
*M8 6 2 5 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*.TRAN 1u 80u*

*.PROBE*

*.END*

# 2.2.5: Design of 1 Bit ALU:

** Step1] Block Diagram/Circuit Diagram:**

**Step2] Netlist for 1 Bit ALU:**

*\*1bit ALU using different design styles*

*.include "D:\mosmodel.txt"*

*.global Vdd*

*.subckt FA A B Cin CA SU*

*Cl CA 0 100f*

*C2 SU 0 100f*

*.model mn NMOS L = 0.18u W = 0.36u*

*.model mp PMOS L = 0.18u W = 0.72u*

*M1 9 A 0 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M2 9 B 0 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M3 2 6 9 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M4 2 A 7 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M5 7 B 0 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M6 2 A 3 Vdd mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M7 2 B 3 Vdd mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M8 3 6 Vdd Vdd mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M9 8 A Vdd Vdd mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M10 2 B 8 Vdd mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M11 CA 2 0 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M12 CA 2 Vdd Vdd mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M13 11 A 0 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M14 11 B 0 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M15 11 Cin 0 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M16 14 2 11 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M17 12 A 0 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M18 13 B 12 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M19 14 Cin 13 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M20 14 Cin 15 Vdd mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M21 15 B 16 Vdd mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M22 16 A 17 Vdd mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M23 14 2 17 Vdd mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M24 17 A Vdd Vdd mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M25 17 B Vdd Vdd mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M26 17 Cin Vdd Vdd mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M27 SU 14 0 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M28 SU 14 Vdd Vdd mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*.end FA*

*.subckt ANDGATETGL A B O1*

*C1 O1 0 100f*

*.model mn NMOS L = 0.18u W = 0.36u*

*.model mp PMOS L = 0.18u W = 0.72u*

*M1 O1 6 B Vdd mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M2 O1 A B 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M3 O1 A 5 Vdd mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M4 O1 6 5 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M5 6 A Vdd Vdd mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M6 6 A 0 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*.end ANDGATETGL*

*.subckt NORGATE A B O2*

*C1 O2 0 100f*

*.model mn NMOS L = 0.18u W = 0.36u*

*.model mp PMOS L = 0.18u W = 0.72u*

*M1 O2 A 6 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M2 O2 4 5 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M3 5 B Vdd Vdd mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M4 5 B 0 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M5 4 A Vdd Vdd mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M6 4 A 0 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*.end NORGATE*

*.subckt XORGATETGL A B O3*

*Cl O3 0 100f*

*.model mn NMOS L=0.18u W=0.36u*

*.model mp PMOS L=0.18u W=0.72u*

*M1 3 A Vdd Vdd mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M2 3 A 0 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M3 5 B Vdd Vdd mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M4 5 B 0 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M5 O3 A B Vdd mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M6 O3 3 B 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M7 O3 3 5 Vdd mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M8 O3 A 5 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*.end XORGATETGL*

*.subckt 4MUX A1 B1 C1 D1 S0 S1 O4*

*Cl O4 0 100f*

*.model mn NMOS L = 0.18u W = 0.36u*

*.model mp PMOS l = 0.18u W = 0.72u*

*M1 9 S1 Vdd Vdd mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M2 9 S1 0 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M3 10 S0 Vdd Vdd mp AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M4 10 S0 0 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M5 11 9 A1 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M6 11 10 O4 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M7 12 9 B1 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M8 12 S0 O4 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M9 13 S1 C1 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M10 13 10 O4 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M11 14 S1 D1 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*M12 14 S0 O4 0 mn AS = 0.648p AD = 0.648p PS = 3.24u PD = 3.24u*

*.END 4MUX*

*x1 A B Cin CA SU FA*

*x2 A B O1 ANDGATETGL*

*x3 A B O2 NORGATE*

*x4 A B O3 XORGATETGL*

*x5 O3 O1 O2 SU S0 S1 O4 4MUX*

*Vdd Vdd 0 1.8*

*Vs0 S0 0 Pulse 0 1.8 0 1n 1n 10u 20u*

*Vs1 S1 0 Pulse 0 1.8 5u 1n 1n 10u 20u*

*Va A 0 Pulse 0 1.8 0 1n 1n 10u 20u*

*Vb B 0 Pulse 0 1.8 0 1n 1n 20u 40u*

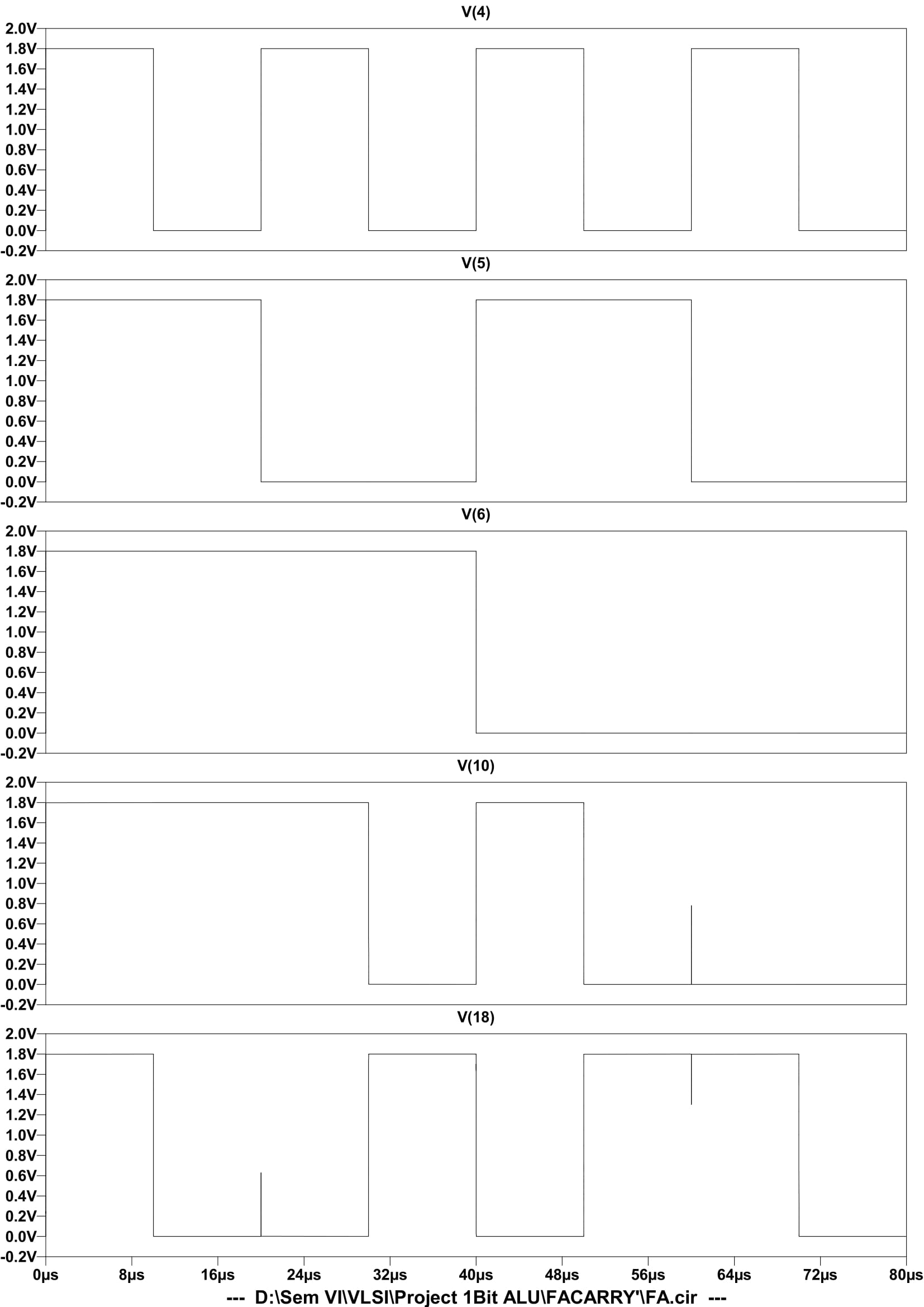
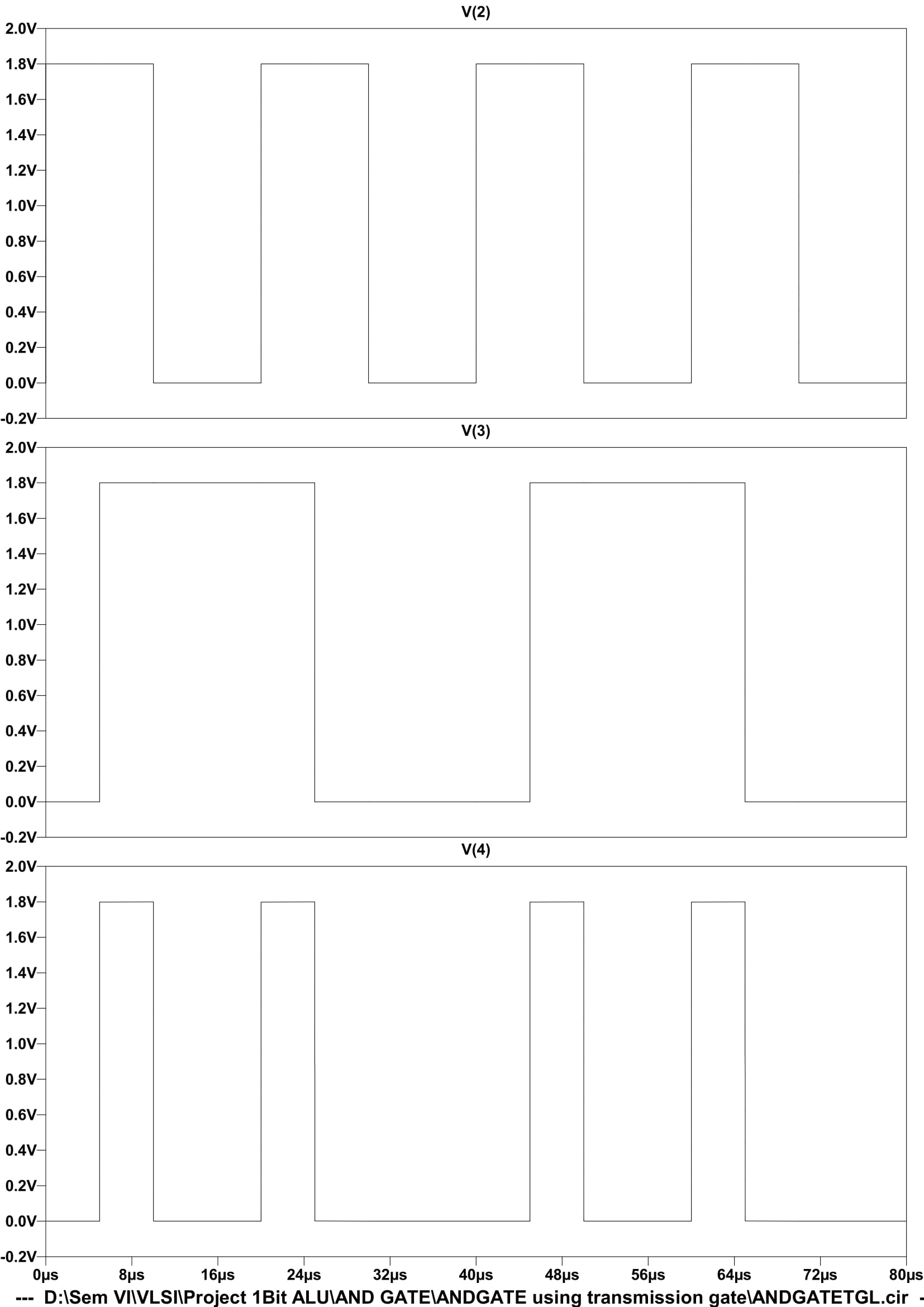
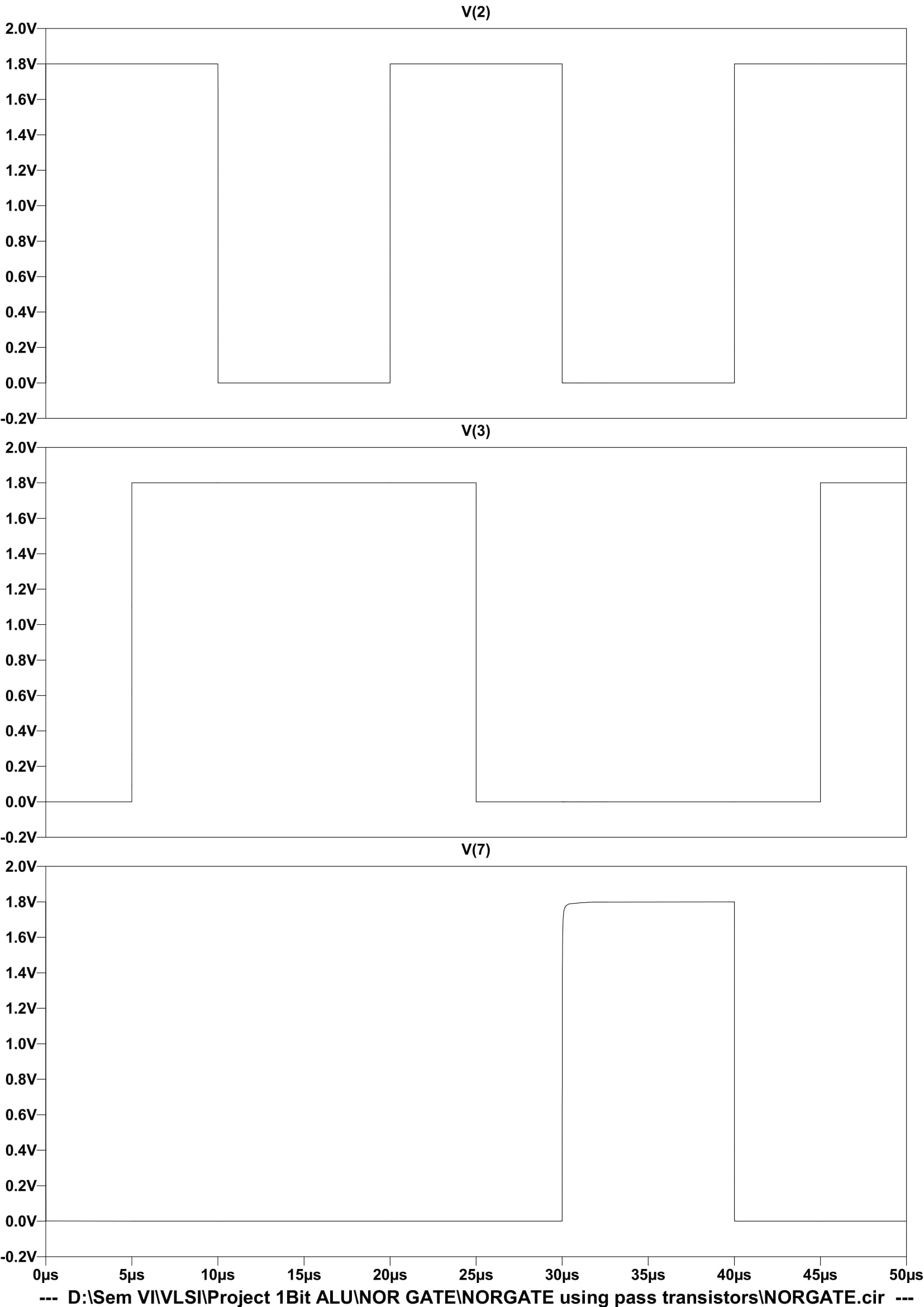
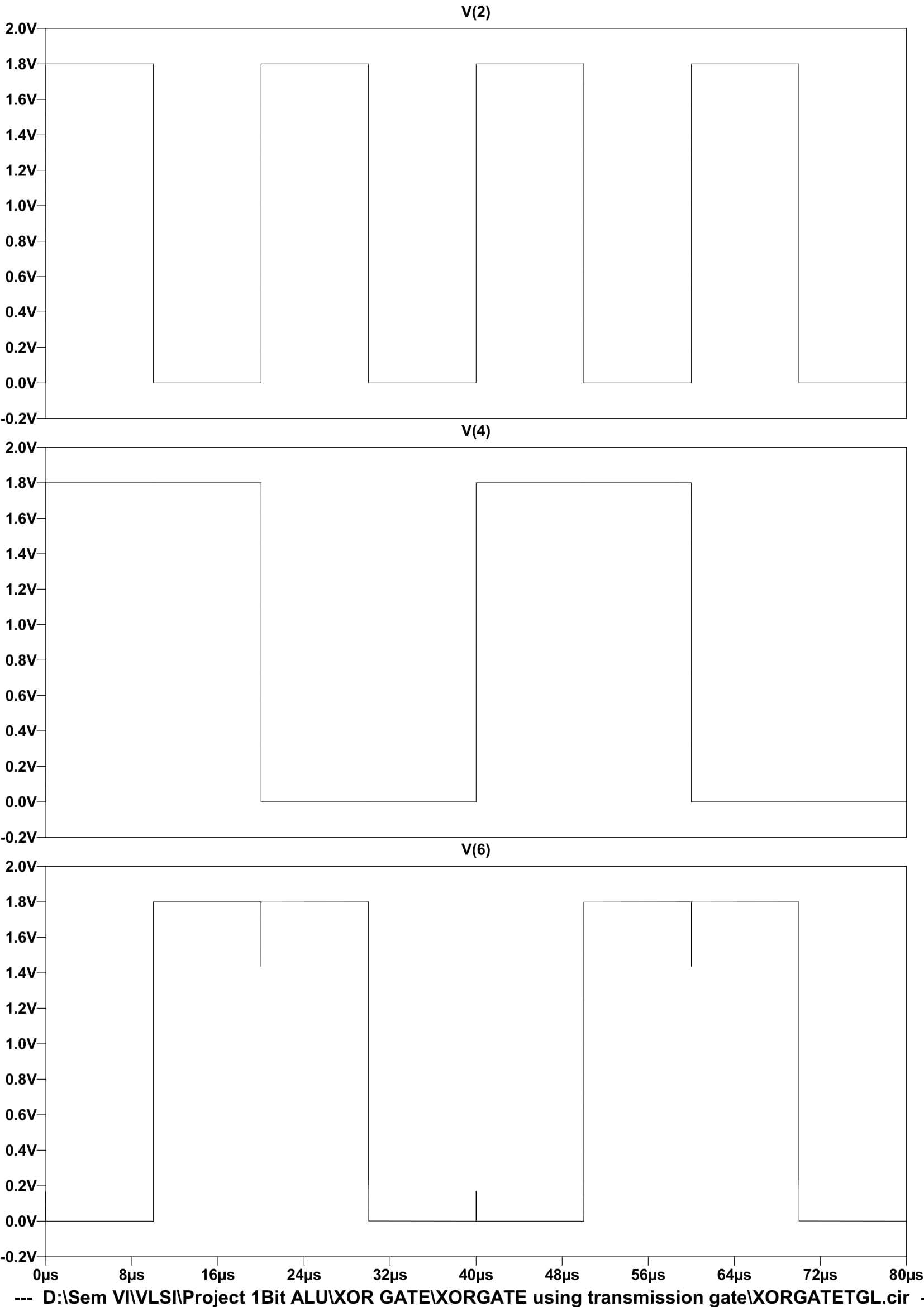
*Vcin Cin 0 Pulse 0 1.8 0 1n 1n 40u 80u*

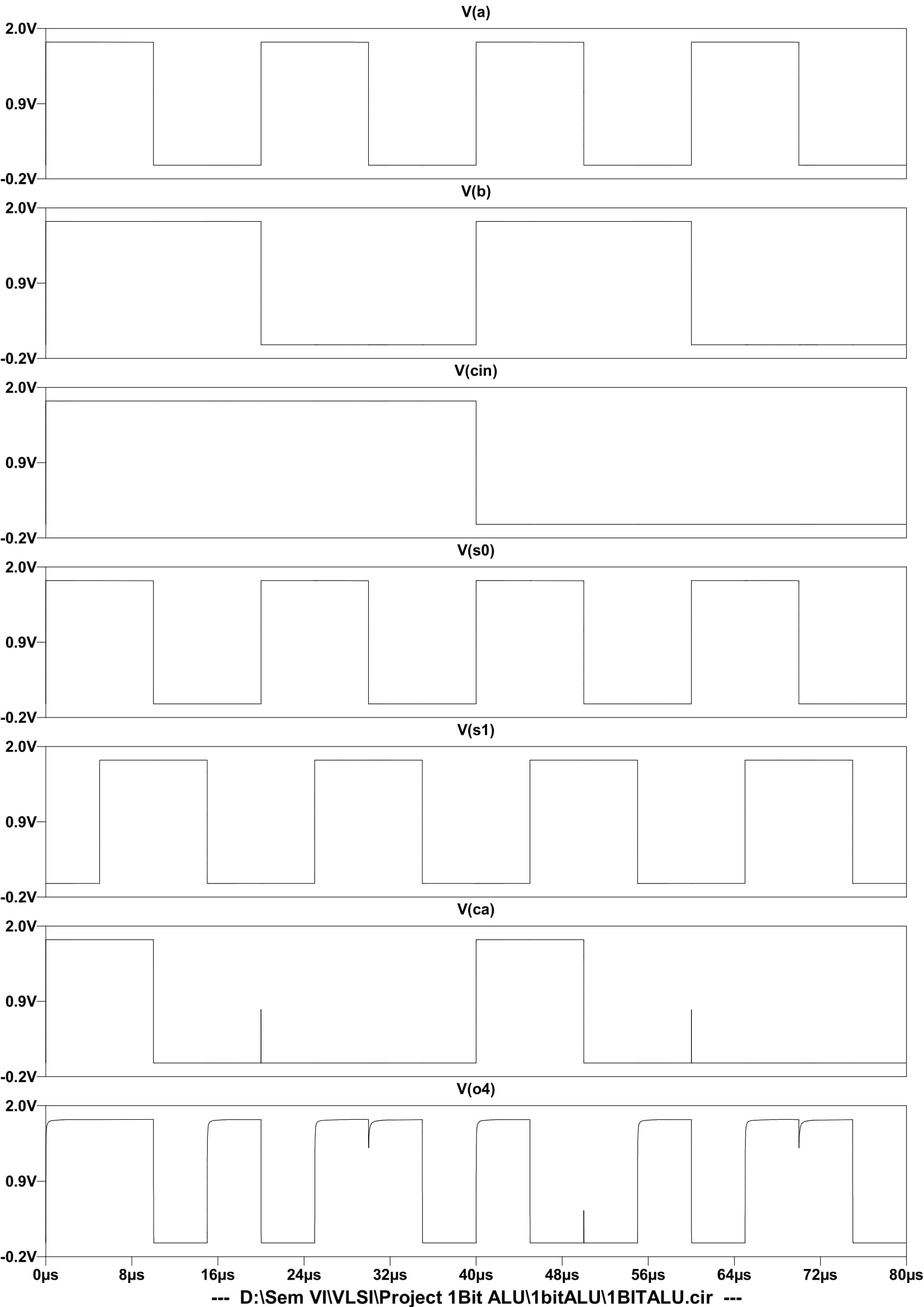
*.TRAN 1n 80u*

*.PROBE*

*.END*

# Chapter 3: Results

* 1. **Output Simulated waveform of FA:**
  2. **Output Simulated waveform of AND:**
  3. **Output Simulated waveform of NOR:**
  4. **Output Simulated waveform of XOR:**
  5. **Output Simulated waveform of 1 Bit ALU:**

****

# Chapter 4: Applications Advantages & Future Scope

**4.1 Applications:**

1. ALU is used in all the digital circuits.
2. ALU is the fundamental part of Computer CPU.
3. It can perform simple arithmetic calculations to complex arithmetic calculations like integration.
4. In the case of logical calculations, it performs them using the concept of Gates.
5. Everything that is given as input is converted to 0’s and 1’s and does the required calculations.
6. Left shifting a bit, right shifting bits, and carry forward concepts are also performed using the Arithmetic Logic Unit (ALU).
7. Registers are small storage spaces on the CPU which is used to store intermediate results.
8. Stacks are used to store details of the performed which have been performed recently and it uses “Last in First Out” concept.
9. Most of the operations that are performed by the CPU are being performed by the ALU.
10. The Working mode of CPU functions well only if the ALU works properly.

**4.2 Advantages & Future Scope:**

1. It can perform on a very large set of instructions.
2. They are spaced so evenly that they never part in between.
3. They remain uniform through the entire operation
4. There is no memory wastage with ALU.
5. It is very fast in general and results can be obtained so easily.
6. It has no sensitivity issues.
7. They minimize the logic gate requirements.

**10.1 References:**

http://web.engr.uky.edu/~elias/projects/10.pdf

CMOS Logic Circuit Design by John P. Uyemura

Design of Analog CMOS Integrated Circuits Second Edition Behzad Razavi

[1] W. Wolf, Modern VLSI Design- Systems On Silicon, Prentice Hall, 1998.

[2] Y. Tsividis, Operation and Modeling of The MOS Transistor, Mc Graw-Hill, 1999.

[3] T. Shibata and T. Ohmi, “A functional MOS transistor featuring gate-level weighted sum and threshold operations”, IEEE Trans. on Electron Devices, vol. 39, Issue 6, pp.1444-1455, June 1992.

[4] A. Srivastava, H.N. Venkata and P.K. Ajmera, “A novel scheme for a higher bandwidth sensor readout”, Proc. of SPIE, vol. 4700, pp. 17-28, 2002.

[5] W. Weber, S.J. Prange, R. Thewes, E. Wohlrab and A. Luck, “On the application of the neuron MOS transistor principle for modern VLSI design”, IEEE Trans. on Electron Devices, vol. 43, pp.1700 – 1708, Oct. 1996.

[6] L. Yin, S.H.K. Embadi and E. Sanchez-Sinencio, “A floating gate MOSFET D/A converter”, Proc. of IEEE International symposium on Circuits and Systems, vol. 1, pp. 409-412, June 1997.

[7] E. Rodriguez-Villegas, G. Huertas, M.J. Avedillo, J. M. Quintana and A. Rueda, “A practical floating-gate Muller-C element using vMOS threshold gates”, IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing, vol. 48, pp. 102-106, Jan. 2001.

[8] J.M. Rabaey, Digital Integrated Circuits- A Design Perspective, Prentice Hall, 1996.

[9] H.T. Bui, Y. Wang and Y. Jiang, “Design and analysis of low-power 10- transistor full adders using novel XOR-XNOR gates”, IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing, vol. 49, pp. 25 –30, Jan. 2002.

[10] J.M. Wang, S.C. Fang and W.C. Fang, “New efficient designs for XOR and XNOR functions on transistor level”, IEEE J. of Solid State Circuits, vol. 29, pp. 780-786, July 1994.

* 1. **Website:**

https://teachcomputerscience.com/arithmetic-logic-unit/

https://www.britannica.com/technology/arithmetic-logic-unit

https://computersciencewiki.org/index.php/Functions\_of\_the\_arithmetic\_logic\_unit\_(ALU)