Department of Electronic and Telecommunication Engineering

Faculty of Engineering

University of Moratuwa, Sri Lanka

EN2110 Electrnics III - Project Report

TIMING ANALYSIS OF DIGITAL CIRCUITS AND PROGRAMMABLE LOGIC DEVICES

Group 10

P.M.N.S. Bandara - 180066F

S.S. Hettiarachchi - 180237G

A.U.P.H. Athukorala - 180051F

This is submitted as a partial fulfillment for the module

EN2110: Electronics III

May 14, 2021

Abstract

In this fundamental study, the timing analysis of a defined digital circuit and an analytical approach to programmable logic devices are designed, evaluated and criticized using the fundamental theories, such that a specified set of requirements has been satisfied in each scenario. All the circuit designs and their simulations are carried out through LTSpice XVII software. The specific circuits that are proposed, as per the requirements and the corresponding simulation results are presented, evaluated and discussed using their adherence to the task in each section.

Labour Division between Team Members

- Parasitic effect in Timing Analysis: 180051F
- Programmable logic block to configure as NAND or NOR: 180237G
- Single switch matrix using six pass transistors: 180066F
- Programmable logic block to design any 3-input combinational circuit: 180066F and 180237G

List of Figures

List of Tables

Contents

Chapter 1

Timing Analysis of Digital Circuits

1.1 Introduction to Parasitic Effect

- When two conductive elements are near to each other and have different voltage levels, an intrinsic and usually undesirable capacitor is created. This is known as the parasitic capacitive effect.
- MOSFETs (Metal Oxide Semiconductor Field Effect Transistor) have a parasitic capacitance due to their structure. The MOSFET is the fundamental building block of MOS transistor-based digital integrated circuits.
- The MOSFET is described by a number of physical and electrical parameters that determine the device's electrical behavior. These parameters are known as MOSFET parameters, and depending on the MOSFET dimensions, they will have an influence on MOSFET activity and parasitic effects.
- The LTSpice software is used to monitor the MOSFET parameters.

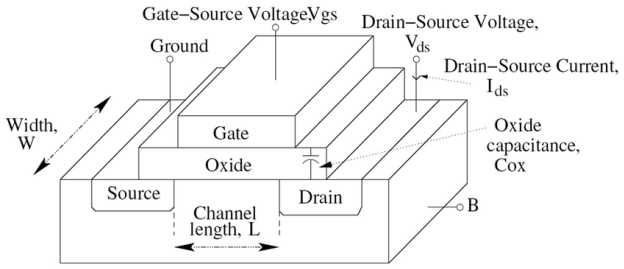
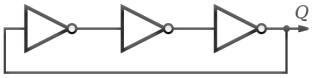


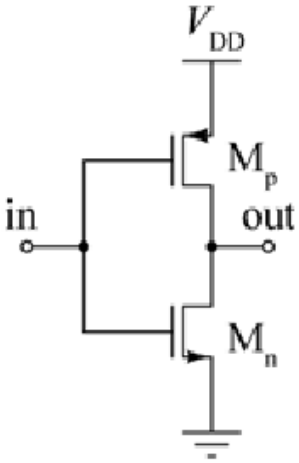
Figure 1.1: Structure and Parameters of MOSFET

1.2 Approach to the implementation

- An oscillator is an electronic circuit that generates a signal with a particular frequency and is used in digital systems to synchronize the computation method.
- A ring oscillator is a circuit composed of "an odd number of inverters connected in series with positive feedback."
- For example, if the oscillator we're using has three inverters, it's known as a three-stage ring oscillator. We can also use NOT gates instead of inverters to identify it.



(a) 3 stage ring oscillator



(b) Single inverter implemented with NMOS and PMOS MOSFETs

Figure 1.2: Pathway schematics of the oscillator design

1.3 Mathematical relations and calculations

Frequency calculation in the ring oscillator

$$f = \frac{1}{2n\tau} \tag{1.1}$$

where τ is the time delay for single inverter and n is the number of inverters in the oscillator.

Table 1.1: Observations of the Parasitic effect on the frequency of the ring oscillator using LTSpice software-CGDO/CGSO/CGBO capacitance values constant

Voltage(V)	CGDO x e-010 (f)		CGSO x e-010 (f)		CGBO x e-011 (f)		Length (m)		Width (m)		period(ns)
	NMOS	PMOS	NMOS	PMOS	NMOS	PMOS	NMOS	PMOS	NMOS	PMOS	
1	6.238	7.43	6.238	7.43	2.56	2.56	40n	40n	0.3u	0.6u	0.807
1	6.238	7.43	6.238	7.43	2.56	2.56	42n	42n	0.4u	0.8u	0.851
1	6.238	7.43	6.238	7.43	2.56	2.56	45n	45n	0.5u	1u	1.074
1	6.238	7.43	6.238	7.43	2.56	2.56	50n	50n	0.5u	1u	1.788
1	6.238	7.43	6.238	7.43	2.56	2.56	55n	55n	0.5u	1u	2.436

Table 1.2: Observations of the Parasitic effect on the frequency of the ring oscillator using LTSpice software-Length and Width values constant

Voltage(V)	Length (m)		Width (m)		CGDO x e-010 (f))		CGSO x e-010 (f)		CGBO x e-011 (f)		Period(ns)
	NMOS	PMOS	NMOS	PMOS	NMOS	PMOS	NMOS	PMOS	NMOS	PMOS	
1	50n	50n	0.5u	1u	5.5	6.5	5.5	6.5	1.5	1.5	0.851
1	50n	50n	0.5u	1u	6.238	7.43	6.238	7.43	2.56	2.56	1.788
1	50n	50n	0.5u	1u	62.38	74.3	62.38	74.3	25.6	25.6	2.403

Here, we first change the width and the length of the utilized MOSFETs keeping the CGDO/CGSO/CGBO/Length values a constant. Then we change the CGDO/CGSO/CGBO capacitances of the MOSFET keeping the width and the Length of the utilized MOSFETs a constant. To obtain the other parameters, we have included a data set in the simulation (level 3 and 54.txt). The width of the MOSFETs has always been chosen so that the PMOS width is double than the width of the NMOS.

1.4 Simulations based on LTSpice software

3 N-channel metal-oxide semiconductors, 3 P-channel metal-oxide semiconductors, 3 capacitors(100f), Conducting wires are used in the following circuit.

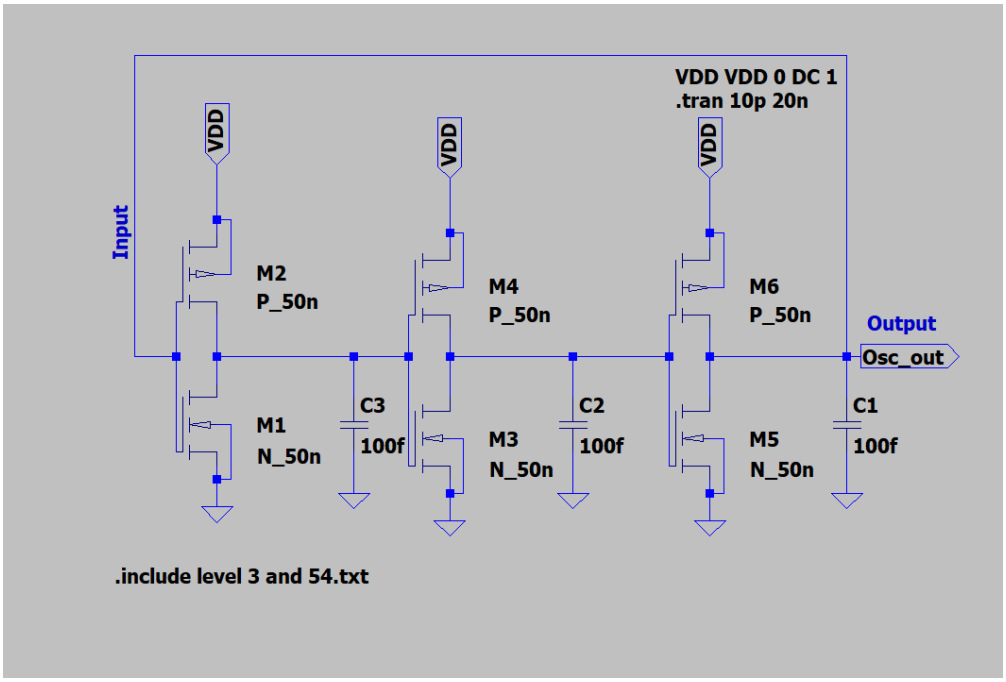


Figure 1.3: Designed Circuit

1.5 Waveform results from the simulations

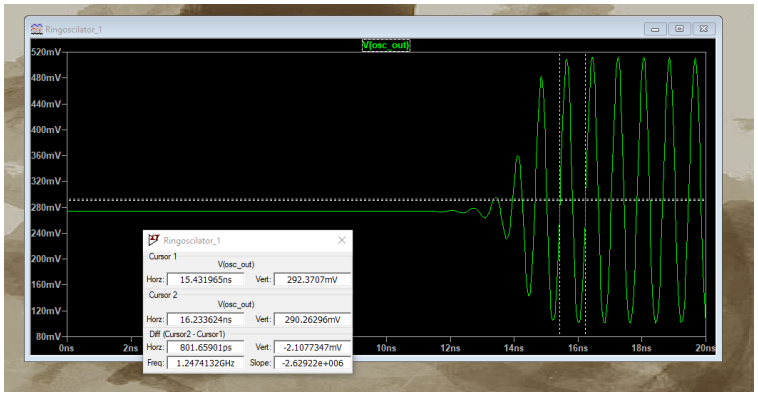


Figure 1.4: Resulting waveform - P_50n l=40n w=0.6u / N_50n l=40n w=0.3u

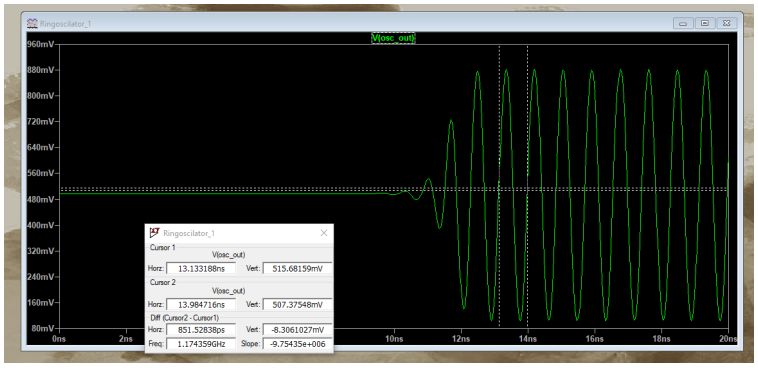


Figure 1.5: Resulting waveform - P_50n l=42n w=0.8u / N_50n l=42n w=0.4u

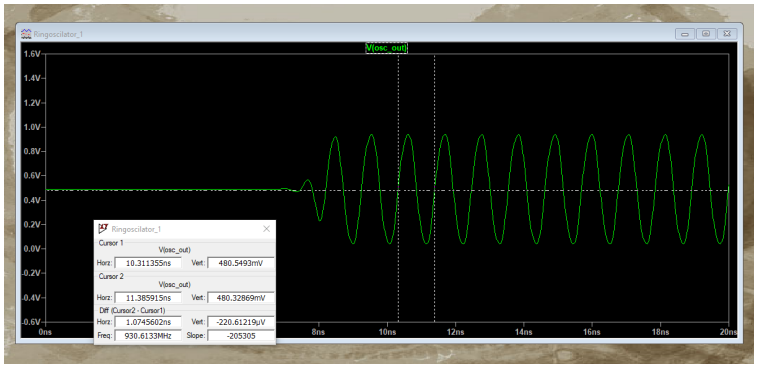


Figure 1.6: Resulting waveform - P_50n l=45n w=1u / N_50n l=45n w=0.5u

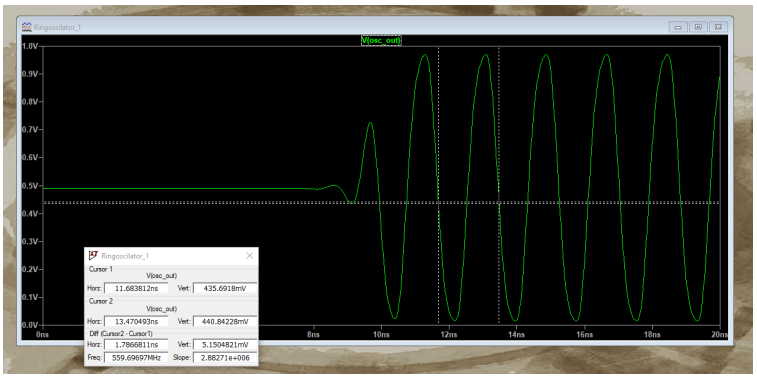


Figure 1.7: Resulting waveform - P_50n l=50n w=1u / N_50n l=50n w=0.5u

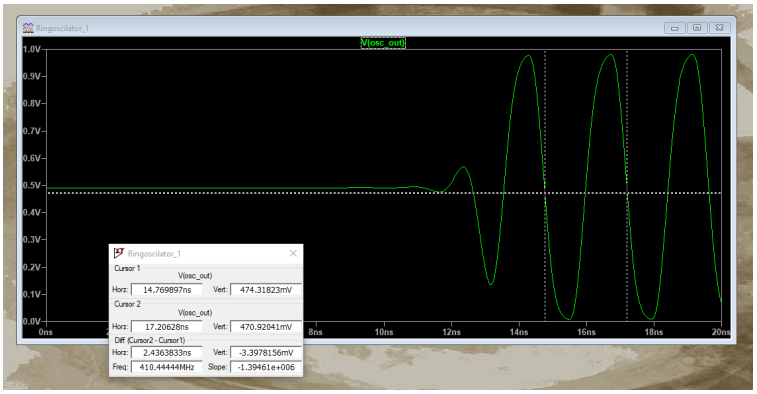


Figure 1.8: Resulting waveform - P_50n l=55n w=1u / N_50n l=55n w=0.5u

1.6 Discussion

According to the table 1.1, when we keep the CGDO/CGSO/CGBO capacitance values of the MOSFET a constant, and increase the width and the length of the MOSFET gradually, we can observe that the period of the ring oscillator is increasing gradually. This suggests that the frequency of the ring oscillator has a possible inverse relationship with the increment of width and length of the utilized MOSFETs.

According to the table 1.2, when we keep the channel length and the width of the MOSFET a constant, and decrease the CGDO/CGSO/CGBO capacitance values of the MOSFET , we can observe that the period of the ring oscillator is decreasing and when we keep the channel length and the width of the MOSFET a constant, and increase the CGDO/CGSO/CGBO values of the MOSFET , we can observe that the period of the ring oscillator is increasing.This suggests that the frequency of the ring oscillator has a possible linear relationship with the internal junction capacitances of the utilized MOSFETs.

This ensures that the parasitic affect of the ring oscillator is increasing with it's parasitic capacitance.

Chapter 2

Programmable Logic Devices

2.1 Programable logic block to configure it as a 'NAND' or a 'NOR' gate using a single selection bit

2.1.1 Approach to the implementation

The main objective of this section is to design a Programmable Logic block that can be configured as a NAND gate or a NOR gate using a single selection bit.

- As the first step, defining the selection bit (0 for NAND gate and 1 for NOR gate) and preparing a truth table considering the inputs and outputs, are implemented.
- Then drawing the Karnaugh map is done as the next approach referring to the truth table. Karnaugh map can be used to build the simplified Boolean algebraic expression that is required to design the logic block.
- Afterwards, using NOT gates, AND gate plane and OR gate plane, required Programmable Logic block can be designed. Gate planes consists with gates(most likely AND gates and OR gates) and these gates can be switched or fixed. In our design we have used programmable AND plane, and fixed OR plane.

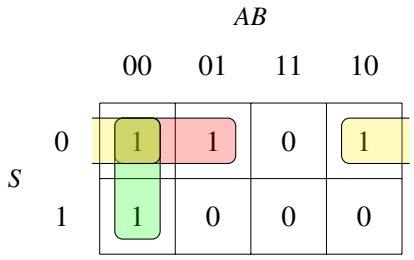
2.1.2 Mathematical relationships and calculations

As previously mentioned, preparing the truth table is required to obtain the simplified Boolean expression. A, B and, S are used as inputs, and for 3 inputs 8 combinations can be obtained. A and B are used as the main inputs for the circuit and S is used as the selection bit of the circuit that decides the gate of the circuit. In this approach, S=0 is used for NAND operation and S=1 is used for NOR operation. Following table 2.1 is shows the calculated truth table.

Table 2.1: Truth-table of the Programmable Logic Block

A	B	S	Output
0	0	0	1
1	0	0	1
0	1	0	1
1	1	0	0
0	0	1	1
1	0	1	0
0	1	1	0
1	1	1	0

Using the truth table, the Karnaugh map can be drawn to obtain the simplified boolean expression. Here, a sum-of-products expression (SOP) can be implemented using AND gates in the AND plane feeding into an OR gate. The designed circuit contain programmable AND plane and, fixed OR plane. In AND plane all of three AND gates can be switched and outputs of the AND gates are directly connected with the OR gate.



$$Out\,put = \overline{SA} + \overline{S}\overline{B} + \overline{A}\overline{B} \tag{2.1}$$

2.1.3 Simulations based on LTSpice Software

Simulation has been implemented using LTsipce XVII software. For the circuit design, AND gates, OR gates and, inverter gates are used. To confirm the precision of the circuit design, four simulations are implemented. The following figure shows the Logic gate circuit design for the simulations.

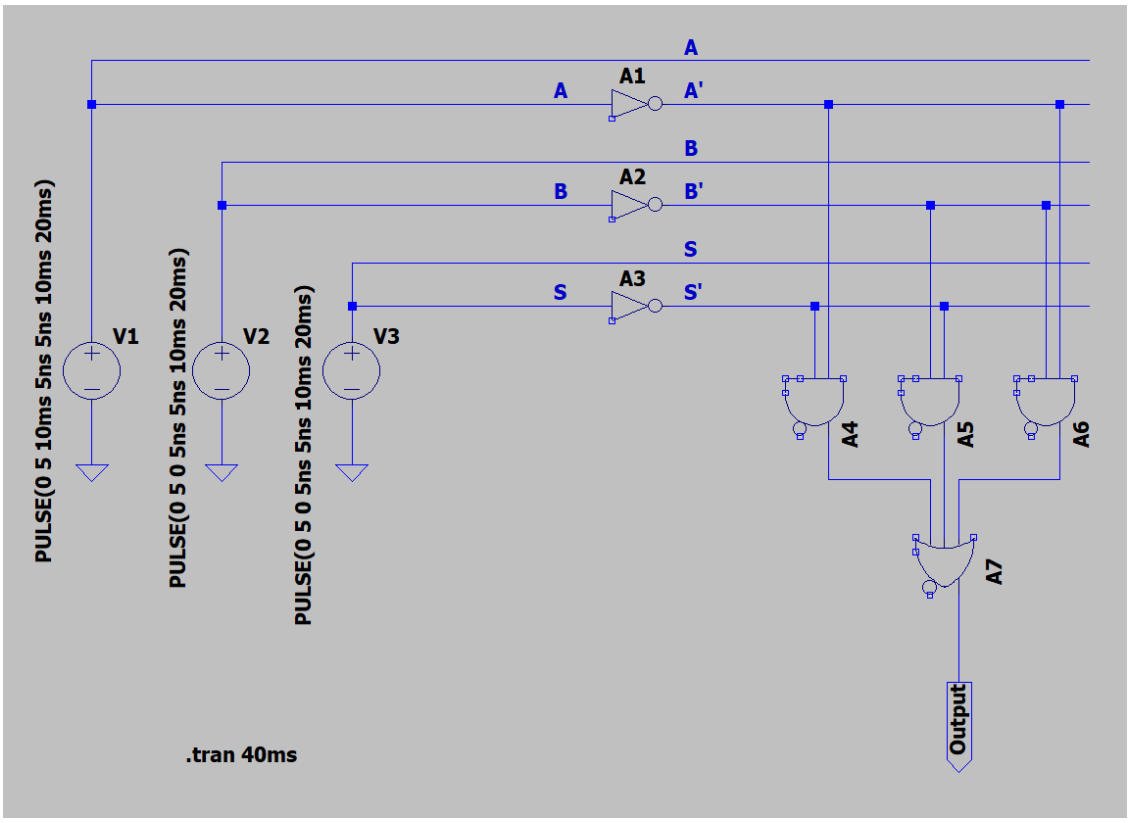


Figure 2.1: Designed Circuit

The following specifications are used for the input waveforms. Input signals are square waved pulse signal with two states. In higher state signal reaches to 5V magnitude peak value and in lower state signal reaches to 0V magnitude.

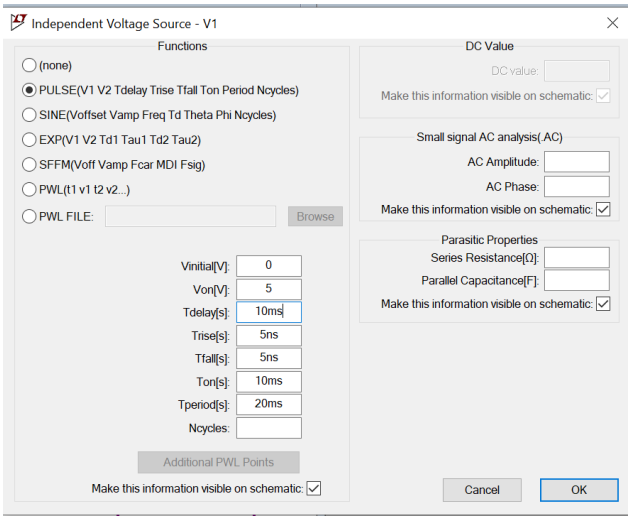


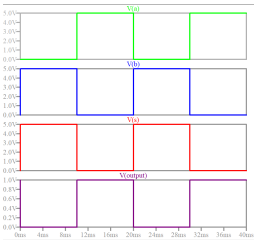
Figure 2.2: Specifications of wave forms

2.1.4 Waveform results from the simulations

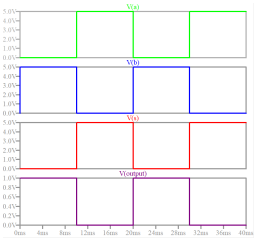
For each simulation, relevant input waveforms and output waveforms are shown below.

(a) Simunation 1

(b) Simulation 2



(c) Simulation 3



(d) Simulation 4

Figure 2.3: Results from the simulations

2.1.5 An alternative approach

As a alternative approach we designed a circuit using four AND gates, two OR gates, three NOT gates, and MUX with the selection bit. This design contain programmable AND plane and programmable OR plane.

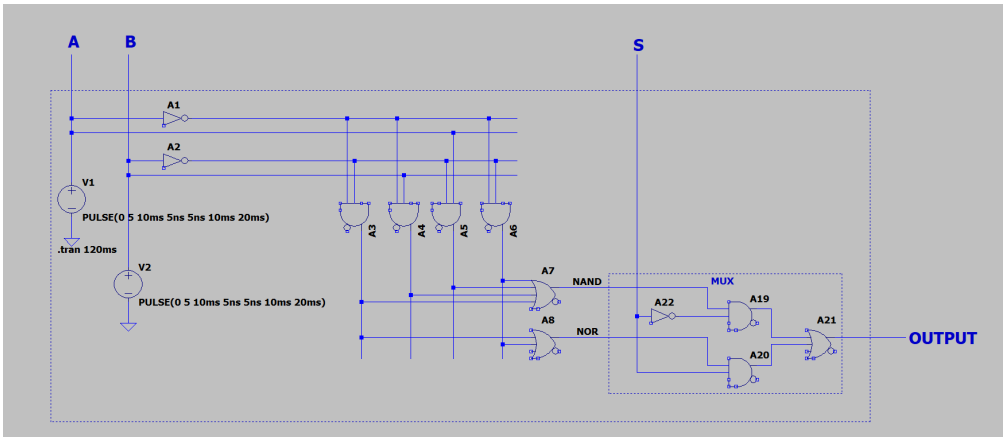


Figure 2.4: Designed circuit of the alternative approach

2.1.6 Discussion

The expected requirement can also be acquired in several other ways like above mentioned alternative approach. In our approach, we tried several circuits and they do not have any problem with the ideal scenario but in practical scenarios. More problems may occur. The use of more gates leads to use of more transistors and that cause to occur propagation delays, additional impedance and space issues. Therefore, the circuit design that is designed with the help of Karnaugh maps, is the optimized circuit design.

This circuit can be defined as a programmable logic device either as a PAL (Programmable Array Logic) or a PLA (Programmable Logic Array). PAL logic devices contained fixed OR planes and programmable AND planes and PLA logic devices contain programmable gate planes (Both AND and OR planes are programmable), but compare to PAL, PLA more complex and more expensive. Therefore PAL could be the better method to create the circuit as our requirement is not complicated. PAL can be divided into two parts reversible PAL and non-reversible PAL. Using reversible PAL may be additionally helpful as it can be used to design another circuit.

We implemented four simulations as we can increase the precision of the results from the designed circuit. Using a square waved pulse signal is a more effective way rather than a step signal as pulse signal contains two state that can be used to obtain 2 of 8 combinations. There may be any other pulse signals which can be cover all the 8 combinations with different periods. As our suggested method is simple and easy to implement we selected this method. Used all square waved pulse signal has the same period (20ms) as mentioned in the specifications. The only change that did to the square wave pulse signal, to do other simulations, is adding a 10ms delay. Input pulse signals' upper-level voltage is 5V and lower-level voltage is 0V but output pulse signals' upper level is lower than 5V as logic gate contains impedances. However propagation delays are not considering in the simulations and clearly identifiable that scenarios can not be founded as the LTspice default logic gates propagation delays are zero. Gates are not connected to a power source as in LTspice default settings gate has been powered.

2.2 Single switch matrix using six pass transistors

2.2.1 Introduction

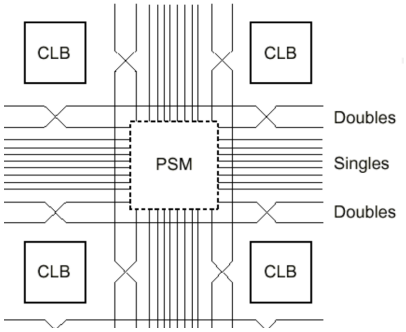
Programmable interconnect technology is considered to be the key component which determines the programmable features of any type of PLD. Therefore, it is often said that PLDs are 90% routing and 10% logic such that the programmable routing necessarily includes the the manner in which the programmable wires are connected and the circuit design of the programmable switches themselves. This technology in PLD has been evolved over the years by introducing several ways to complement the target of programmable interconnection such as:

- *Fuse*: An electrical device that has a low initial resistance and is designed permanently to break an electrically conducting path when current through it exceeds a specified limit. Fuse uses bipolar technology and is nonvolatile and one-time programmable.
- *Antifuse*: An electrical device with a high initial resistance and is designed permanently to create an electrically conducting path typically when voltage across it exceeds a certain level. Antifuse uses CMOS technology and consists of an insulating layer sandwiched between two conducting layers in which, in the unprogrammed state, the insulating layer isolates the top and bottom conducting layers. When programmed, the insulating layer is transformed into a low-resistance link.
- *Floating-gate transistor switch*: This is based on the principle of placing a floating-gate transistor between two wires in such a way as to facilitate a WIRE-AND function such as in EPROM or EEPROM devices.
- *Static-RAM controlled programmable switch*: This is based on having a non-volatile memory in order to control not only the gate nodes but also the select inputs of multiplexers that drive the logic block inputs. SRAM controlled programmable switch matrix (PSM) may use pass transistors or multiplexers as per the requirement.

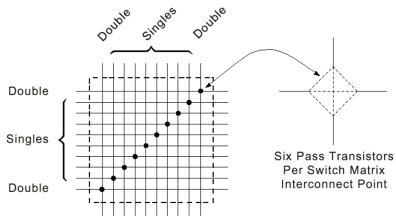
In order to facilitate the electronically ”wired” nature of the PLD logic circuit, a switching matrix is implemented where a switch is implemented at each intersection of the conductor grid that could electrically connect a defined column (the conductor or the wire which is connected to the input signals of the grid) to a defined row (the conductor or the wire which is connected to the output signals of the grid). By deciding the connection status of each interconnection, it is expected to ”program” this programmable array in order to enable the programmable interconnect technology.

2.2.2 Approach to the implementation

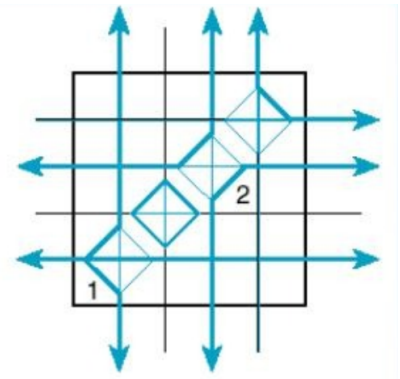
In order to support all possible interconnections at a grid intersection point, at least six switches are needed as illustrated below.



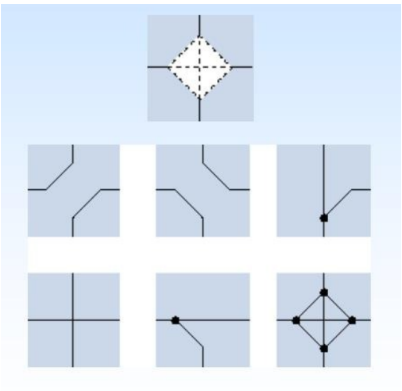
(a) PSM in a typical FGPA configuration (Xilinx 4000)



(b) Switch pass transistor configuration in a switch matrix interconnect point



(c) Possible paths at a single switch matrix



(d) Possible connections at a single switch matrix

Figure 2.5: The need for six pass transistors at a switch matrix interconnect point

Thus, for a single switch matrix interconnect point, six switches are required in order to establish a programmable switch matrix. Pass transistors is considered to be a viable option for a switch in these cases since they are bidirectional type of switches, they occupy less area (comparative to buffers) and validated to be faster at short wiring paths (passing through small number of switches).

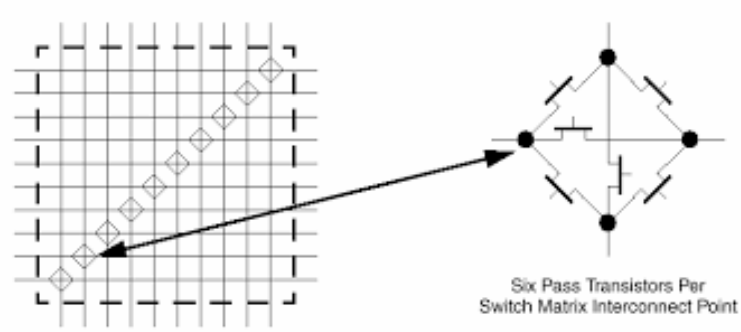


Figure 2.6: The simplest form of single switch matrix using six pass transistors

2.2.3 Mathematical relationships and calculations

Pass transistor logic for NMOS

This could be simply illustrated using a AND gate, designed using NMOS transistors.

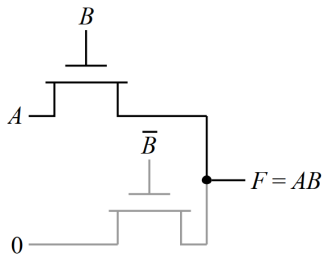


Figure 2.7: NMOS-based AND gate to demonstrate the activity of pass transistors

When B is "1" (logic high), top device turns on and copies the input A to output F. When B is low, bottom device turns on and passes a "0" (logic low) (The presence of the switch driven by $\overline{B}$ is essential to ensure that the gate is static: a low-impedance path must exist to supply rails).

Howsoever, when NMOS is used as the pass transistor:

When the source voltage is zero;

$$V_{GS} = V_G - V_S = V_{dd} \tag{2.2}$$

So that, it satisfies,

$$V_{GS} \geq V_{tn} \tag{2.3}$$

But, when the source voltage is V_{dd};

$$V_{GS} = 0 \tag{2.4}$$

Therefore, the maximum voltage that could passed from source terminal to drain terminal is $V_{dd} - V_{tn}$.

In summary, NMOS is effective at passing a 0, but poor at pulling a node to V_{dd}. When the pass transistor a node high, the output only charges up to $V_{dd} - V_{tn}$. Further, this becomes worse due to the body effect. The node will be charged up to $V_{dd} - V_{tn}$.

2.2.4 Simulation Circuit 01

Simulations based on LTSpice Software

All the simulations are designed and complied using LTSpice XVII software. The initial simulation is based on the circuit design introduced in Figure 2.6 where six NMOS transistors are configured as pass transistors. For the simulation, six monolithic MOSFETs are utilized each has four terminals including one terminal for the transistor body (p-type substrate). All terminals that are connected to transistor bodies are grounded

(or connected to the lowest possible voltage) in order to ensure the reverse-biased requirement of the Gate-Source junction.

In the following simulation, it is intended to pass the signal from top to left and the signal is a pulse signal with two states: 0V and 5V and a period of 200ms. The gate of the pass transistor between the top and left is connected to a dc voltage of logic high (5V) while all the other gates of the transistors are grounded.

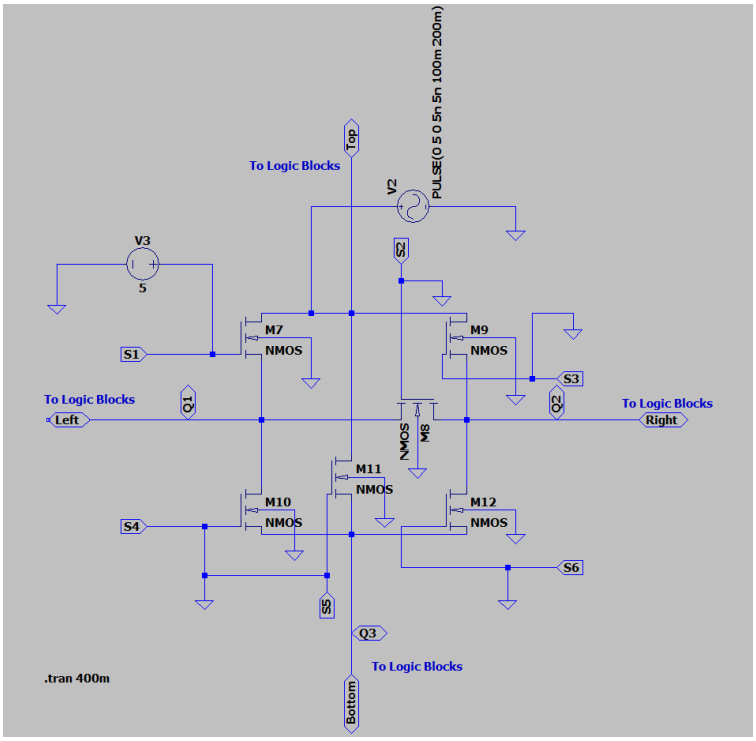
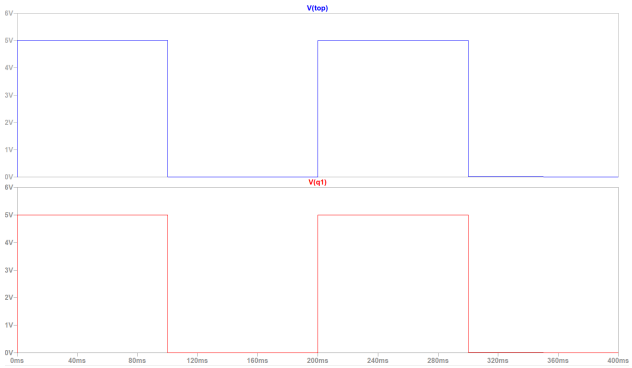


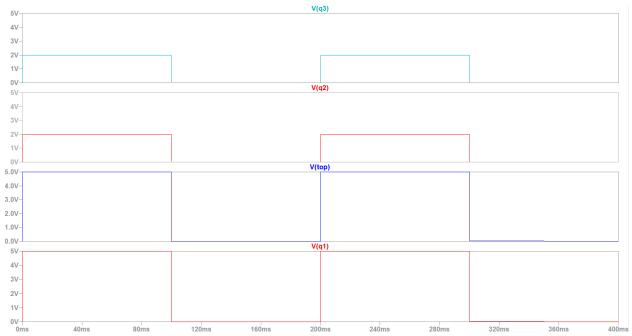
Figure 2.8: Proposed simulated circuit based on LTSpice XVII software

Waveform results from the simulations

This circuit is capable of handling the intended purpose as the following result waveforms suggest. Howsoever, a leakage voltage is observed in the right and bottom connectors due to the internal capacitance of the transistors.



(a) Input (top) signal and the passed (output - q1) signal from the left



(b) Imperfect voltage leakages in the right (q2) and bottom (q3) connectors

Figure 2.9: Resulting waveforms from the proposed circuit

2.2.5 Simulation Circuit 02

Simulations based on LTSpice Software

In order to deal with the input impedance and leakage voltage of the NMOS transistors, the following circuit is proposed and simulated using the same software and most of the circuit connections are similar to that of Simulation 01 (the signal is passed from left to bottom) unless the fact that this circuit is combined with four pull-up external capacitors.

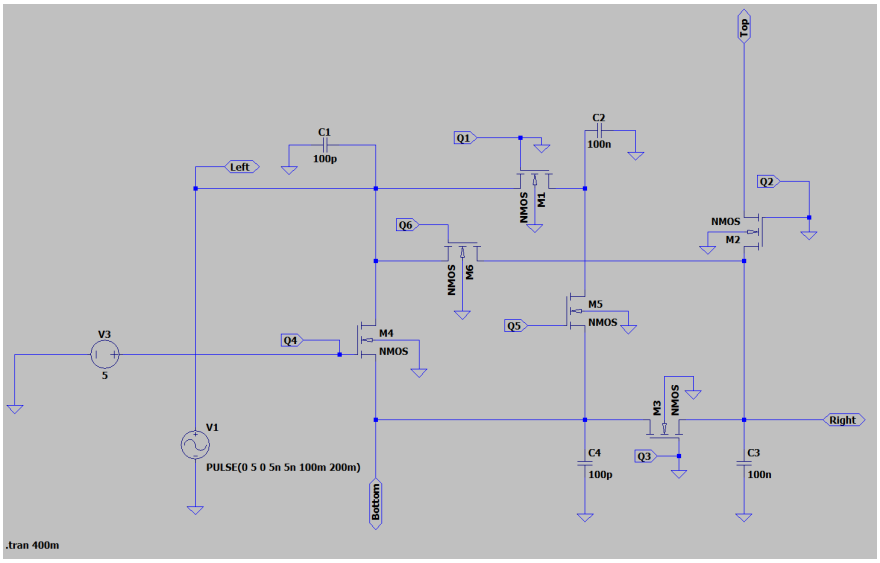
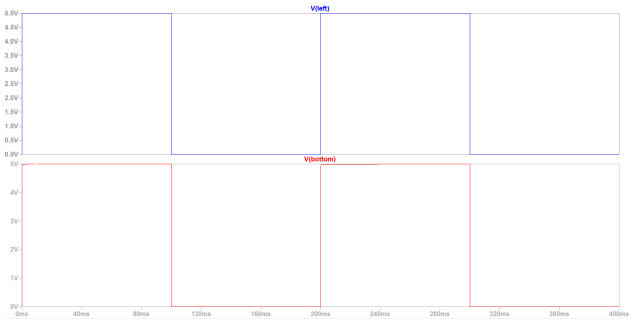


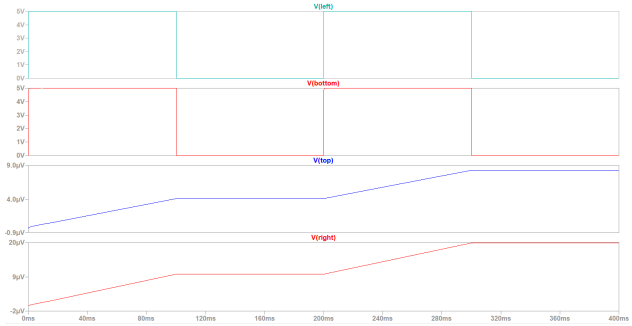
Figure 2.10: Proposed modified simulated circuit based on LTSpice XVII software

Waveform results from the simulations

The following resulted waveforms obviously convey the efficiency of this implementation which drastically reduce the leakage voltage into μV level while preserving the intended functionality.



(a) Input (left) signal and the passed (output - bottom) signal from the left

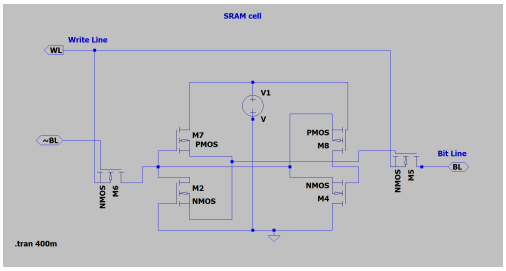


(b) Imperfect μV voltage leakages in the right and top connectors

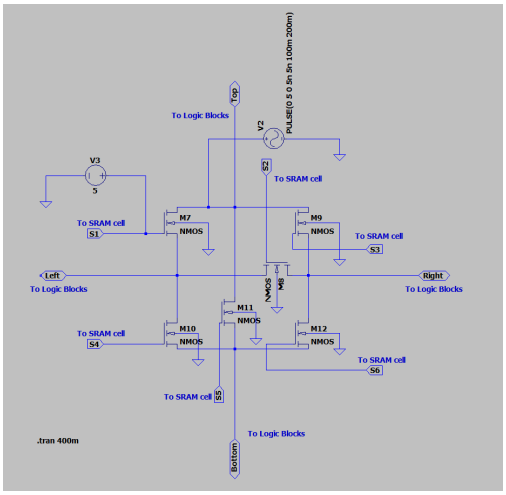
Figure 2.11: Resulting waveforms from the proposed modified circuit

2.2.6 Extended circuit using SRAM-controlled gates of NMOS transistors

In order to facilitate the po control ssibility to program this switch matrix, it is needed to control/set the status of the gates of the pass transistors accordingly. SRAM-controlled gates are considered to be an efficient option for this scenario. Typically, a SRAM with m address lines and n data lines is referred to as a 2m × n memory and is capable of storing 2m n-bit words. The following basic SRAM cell comprising six MOSFET switches, with four of them connected as cross-coupled inverters. A basic SRAM cell can store one bit of information. The reading operation is carried out by precharging both the bit lines (BL and BL to logic '1' and then asserting the WL line. The writing operation is done by giving the desired logic status to the BL line and its complement to the BL line and then asserting the WL line. Besides, the intended circuit based on SRAM-controlled pass transistors (in LTSpice) is illustrated in Figure 2.12 (b).



(a) The basic SRAM cell



(b) SRAM-controlled programming switch

Figure 2.12: Modified SRAM-controlled circuits

2.2.7 Discussion

The simplified logic behind the single switch matrix could be presented as in the following figure.

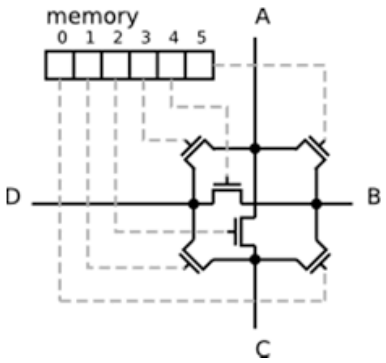


Figure 2.13: Programming configuration of the single switch matrix

The programming of this interconnect matrix could be understood and configured using a six-bit memory cell where every pass transistor is either at 0 (gate is at logic high) or at 1 (gate is at logic low). This ensures that there are $2^6 = 64$ possible distinct connections that could be handled through this switch matrix with distinct programming configurations. The simplest forms would be where only one pass transistor is enabled for passing the signal such as in 000001 or 100000 or 010000 etc. configurations.

The implemented single switch matrix using six pass transistors in simulation 01 is capable of enabling the interconnect matrix into programming by allowing and facilitating various possible connections between combinational logic blocks and input/output pins. Howsoever, this initial circuit has several drawbacks including the leakage voltages presented in other connectors. This circuit which is only with NMOS pass transistors, is enhanced using defined external capacitances to mitigate the above drawbacks in the simulation 01 and the results of the simulation 02 circuit ensures a more efficiency since it reduces the leakage in to μV level while preserving the intended switched passing.

Howsoever, in the above both proposed circuits, the functionality of the single switch matrix is performed under static conditions where only the gate of the required pass NMOS transistor is connected to a constant DC voltage source while all the other gates are being grounded. Therefore, in order to enable the potential to switch between ”on” (gate is at logic high) and ”off” (gate is at logic low), the gates of the pass transistors must be relevantly addressed and the extended circuit of SRAM-controlled pass transistors provides a possible framework for that purpose while being memorizing one bit through basic SRAM cell.

2.3 Programmable logic device that can be used to design any 3 input combinational circuit

2.3.1 Approach to the implementation

The objective of this section is to design a programmable logic device which could be utilized to build any 3 input combinational circuit. Since any 3-input combinational logic could be represented as a sum-of-products (or sum of min terms), the required circuit should adhere the capability to be adjusted to implement sum-of-products using necessary types and number of logic gates: AND and OR gates.

Therefore, our approach includes a programmable logic array where both AND planes and OR planes are capable of being programmed as per the required combinational logic. The implemented circuit consists of three input channels, eight AND gates and two OR gates where final output is obtained through OR planes as shown in the simulation circuit.

2.3.2 Mathematical relationships and calculations

Since the implemented combinational logic circuit is intended to obtain 3 inputs, there are eight possible combinations which may be resulted using these inputs. Hence, to satisfy these combinations, eight AND gates are required where the connections are established (in order to program) to the input lines via the AND plane. The outputs from the AND plane are fed into the two OR gates via the OR plane since the utilized OR gates are a type of 5-inputs, and thus are sufficient to perform the intended operation.

An example combinational logic which could be obtained through this proposed PLA would be described as follow. If the required performance of the combinational logic is conveyed through the following truth table:

Table 2.2: Required performance in a truth-table

A	B	C	Output
0	0	0	1
1	0	0	1
0	1	0	1
1	1	0	1
0	0	1	1
1	0	1	1
0	1	1	1
1	1	1	0

This truth-table could be simplified using the Karnaugh map as shown below, in order to obtain the simplest boolean expression in this scenario.

C \ AB	00	01	11	10
0	1	1	1	1
1	1	1	0	1

So that, the obtained boolean expression would be, when the output is P and inputs are A, B and C,

$$P = \overline{A} + A\overline{C} + A\overline{B} \tag{2.5}$$

Howsoever, in order to implement this in the proposed PLA, it is required to convert this boolean expression into an expandable format where it is expressed as a sum of min-terms. Therefore, the output would be,

$$P = AB\overline{C} + A\overline{B}C + A\overline{B}\overline{C} + \overline{A}BC + \overline{A}B\overline{C} + \overline{A}\overline{B}C + \overline{A}\overline{B}\overline{C} \tag{2.6}$$

The corresponding circuit design to the above combinational logic, using the proposed PLA is illustrated in the next section.

2.3.3 Simulations based on LTSpice Software

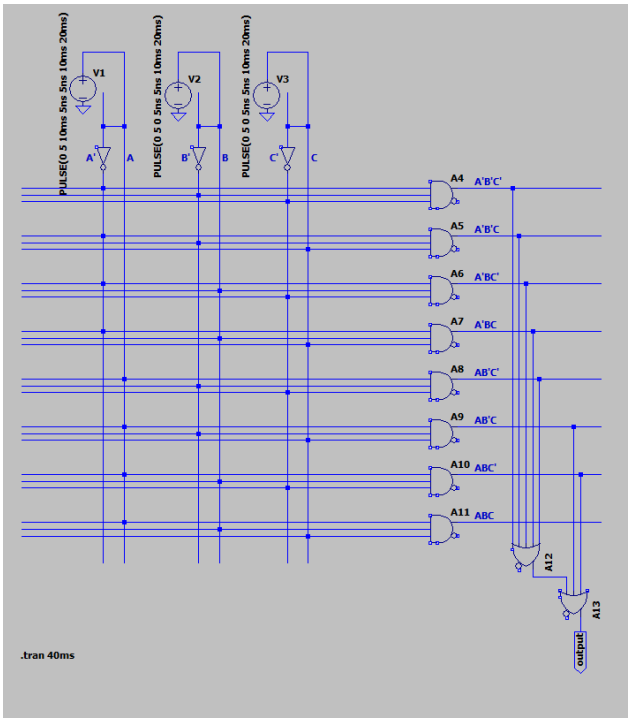


Figure 2.14: Designed circuit based on LTSpice XVII software

2.3.4 Waveform results from the simulations

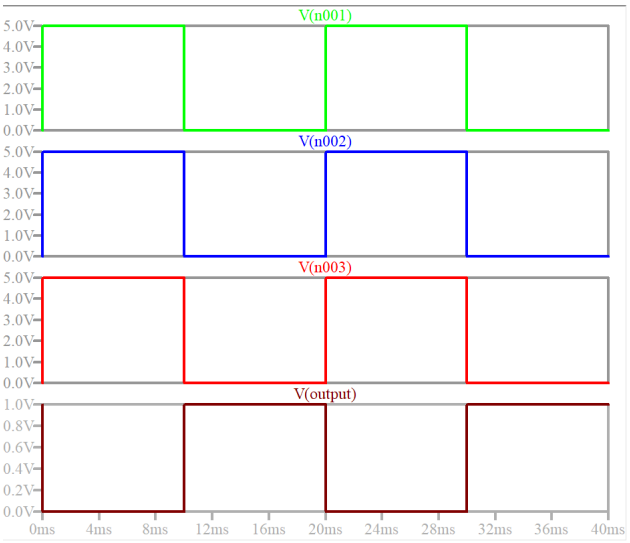


Figure 2.15: Results from the simulation 1

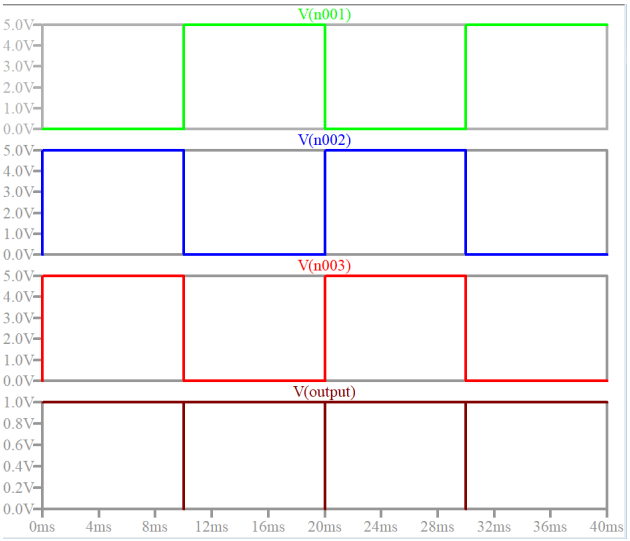


Figure 2.16: Results from the simulation 2

2.3.5 Discussion

The proposed programmable logic device (which is a programmable logic array) is capable of implementing any 3 input combinational logic circuit since it comprises of an AND plane and an OR plane where both planes are programmable. Following the fact that any combinational logic circuit could be expressed as a sum of min-terms, the proposed PLD is capable of designing any 3-input combinational logic circuit using the potential of programming the connections. The example combinational logic circuit demonstates and validates the above argument in a practical manner.

Howsoever, it has to be noted that, in order to allow for the extreme flexibility in combinations, the implemented logic circuit in PLD is not sufficiently optimized (as seen in the above example) in the boolean logic. Further, the proposed PLD is also complex (compared to most of other 3-input programmable logic devices), needs more space (since of having more gates and thus, more transistors), has comparatively more propagation delays and additional impedences in operation.