**ECE 7349-ADVANCE TOPICS IN MICROELECTRONCIS**

**SPRING - 2014**

DESIGN LAYOUT AND SIMULATION OF FOUR BIT LINEAR FEEDBACK SHIFT REGISTER

***Primary Examiner: Professor Earl Charlson***

**Project Report Submitted By**

Sharath Chandra (PS ID: 1257931)

Srivatsan Badri Narayanan (PS ID: 1090241)

***Date: 2nd MAY 2014.***

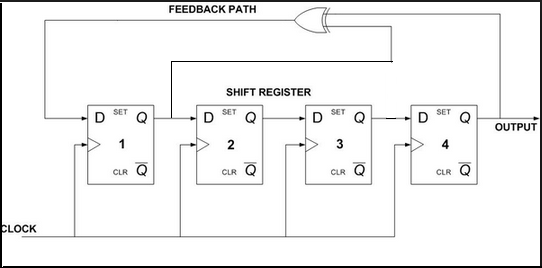
**INTRODUCTION**

An LFSR is a shift register that when clocked, advances the signal through the register from one bit to the next most-significant bit. A linear feedback shift register can be formed by performing exclusive-OR on the outputs of two or more of the flip-flops together and feeding those outputs back into the input of one of the flip-flops. The feedback causes the value in the shift register to cycle through a set of unique values. The choice of LFSR length, gate type, LFSR type, and maximum length logic and tap positions allows the user to control the implementation and feedback of the LFSR which in turn controls the sequence of repeating values the LFSR will iterate through.

*Inspiration: Linear Feedback Shift Register is used in our project “Keypad Scanner” to implement a module for password protection system using DE2 board demonstration. In this project LFSR circuit is used as a module to introduce a delay of 8ms to avoid debouncing effect of a mechanical switch. Thus LFSR circuit has a wide range of application in implementing various digital circuits and we personally had implemented, tested and verified its functionality in our above mentioned project. This experience has inspired us to design and verify the layout for LFSR circuit.*

The layouts of circuits have been drawn using MAGIC VLSI software of Ubuntu (a Linux flavor) with CMOS technology.

Block Diagram of LFSR with XOR feedback



**SPICE EXTRACTION STEPS:-**

* The layout is designed in Magic VLSI software and saved.
* "Extract all" command is used to generate all netlists of the designed layout.
* As a result, .ext file is generated.
* This .ext file is converted into spice file through command "ext2spice filename"
* A new terminal is opened & a command "vi filename.spice" is entered to view generated spice file.
* The contents of this file are copied and saved as .cir file with all voltage sources & ground typed.
* This .cir file is simulated in LtSpice software of windows version.

**Note**: Oracle VM Virtual box is used to run Ubuntu in parallel with Windows and file sharing is enabled between both of these operating systems.

CMOS TECHNOLOGY

**Complementary metal–oxide–semiconductor** (**CMOS**) is a major class of integrated circuits. CMOS technology is widely used in Digital circuits’ viz., microprocessors, microcontrollers, static RAM, and other digital logic circuits. CMOS technology is also used for a wide variety of analog circuits such as image sensors, data converters, and highly integrated transceivers in many types of communication systems.

CMOS was also sometimes referred to as **complementary-symmetry metal–oxide–semiconductor** (or COS-MOS). The words "complementary-symmetry" refer to the fact that the typical digital design style with CMOS uses complementary and symmetrical pairs of p-type and n-type metal oxide semiconductor field effect transistors (MOSFETs) for logic functions.

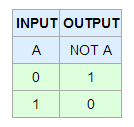
Two important characteristics of CMOS devices are high noise immunity and low static power consumption. Significant power is only drawn when the transistors in the CMOS device are switching between on and off states i.e., CMOS draws power only during transition. Consequently, CMOS devices do not produce as much waste heat as other forms of logic.

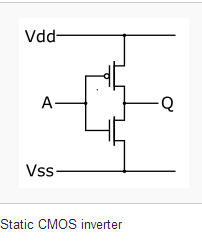
To design the layout of circuits in our project we have made use of Invertor, NAND, Ex-OR gates as a basic elements Thus it is appropriate to mention here their functionalities in brief.

Inverter Gate:

CMOS Inverter circuit is composed of two MOSFETs.  The top FET (MP) is a PMOS type device while the bottom FET (MN) is an NMOS type.  The body effect is not present in either device since the body of each device is directly connected to the device’s source.  Both gates are connected to the input line.  The output line connects to the drains of both FETs.

Truth table:-



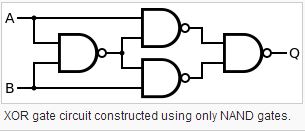
****

**EX-OR GATE**

The output of Ex-OR gate is high if one and only one of the inputs A or B is high. If either both the inputs are high or both are low the output is low. XOR can also be viewed as addition modulo 2. As a result, XOR gates are used to implement binary addition in computers. A [half adder](http://en.wikipedia.org/wiki/Half_adder) consists of an XOR gate and an [AND gate](http://en.wikipedia.org/wiki/AND_gate).

Truth Table:-

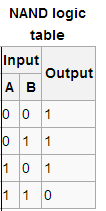
|  |  |  |
| --- | --- | --- |
| **INPUT** | | **OUTPUT** |
| A | B | A Ex-OR B |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

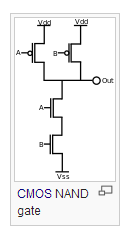


NAND GATE

The NAND gate is significant because any Boolean function can be implemented by using a combination of NAND gates. This property is called [functional completeness](http://en.wikipedia.org/wiki/Functional_completeness). NAND GATE is a [logic gate](http://en.wikipedia.org/wiki/Logic_gate) which produces an output that is false only if all its inputs are true; thus its output is [complement](http://en.wikipedia.org/wiki/Complement_(set_theory)) to that of the [AND gate](http://en.wikipedia.org/wiki/AND_gate). A LOW (0) output results only if both the inputs to the gate are HIGH (1); if one or both inputs are LOW (0), a HIGH (1) output results. It is made using transistors.

Truth Table:-



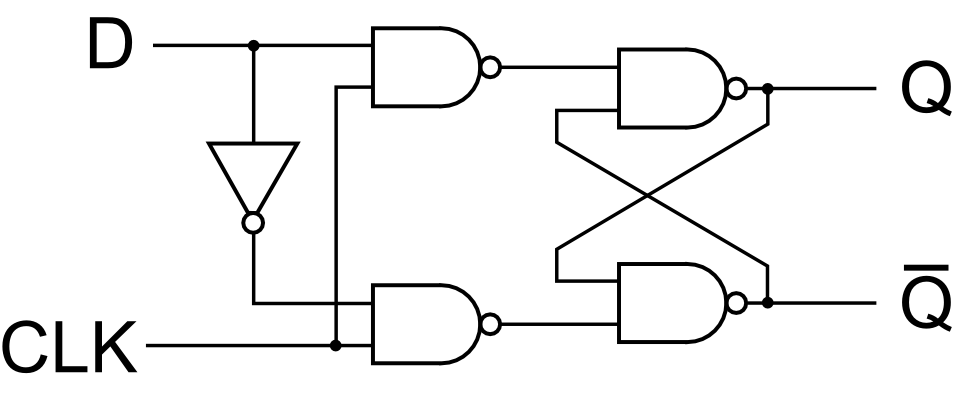


**D Flip Flop:**

A flip-flop or latch is a [circuit](http://en.wikipedia.org/wiki/Electronic_circuit) that has two stable states and can be used to store state information. The circuit can be made to change state by signals applied to one or more control inputs and will have one or two outputs. It is the basic storage element in [sequential logic](http://en.wikipedia.org/wiki/Sequential_logic).

The D flip-flop takes one data input and updates its state Q, on a clock tick, according to the table shown below. D flip flop can be implemented using the universal gates NAND and NOR.

Below is the D flip flop implementation using NAND gate.



Truth Table:-

|  |  |  |
| --- | --- | --- |
| **INPUT** | **OUTPUT** | |
| D | Q | Qbar |
| 0 | 0 | 1 |
| 1 | 1 | 0 |

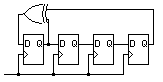
Series of D flip-flops connected together form Serial In & Parallel out Shift register when feedback mechanism is not used.

**SPECIFICATION FOR LFSR**

**Linear Feedback Shift Register**

**Linear Feedback Shift Register for a sequence of four bits can** be implemented by connecting four D flip flops sequentially and feedback from any of the D flip flops can be tapped and connected to the EXOR gate. LFSR can be implemented either using many to one or one to many feedback mechanisms.

Many to one approach of implementing a LFSR is as shown below.



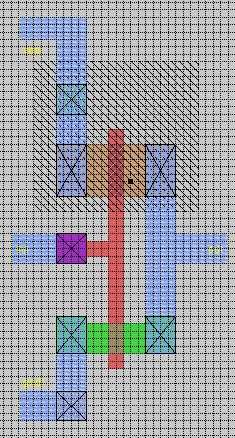
As explained above we have implemented a LFSR circuit wherein we have used a LFSR test bench. This LFSR test bench provides a provision for the user to select number of bits, type of feedback, the feedback logic gate to be used a LFSR circuit.

In the above screen shot we have selected shift register length as 4 and feedback structure as many to one and feedback logic gate as EXOR gate. The following is the pattern obtained starting with rising edge of clock cycle.

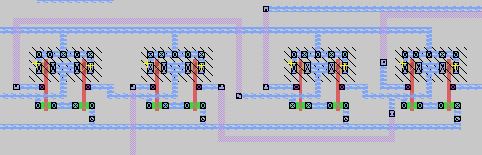
|  |  |
| --- | --- |
| Clock | Pattern Obtained |
| 1 | 1000 |
| 2 | 0001 |
| 3 | 0011 |
| 4 | 0111 |
| 5 | 1111 |
| 6 | 1110 |
| 7 | 1101 |
| 8 | 1010 |
| 9 | 0101 |
| 10 | 1011 |
| 11 | 0010 |
| 12 | 1110 |
| 13 | 1001 |
| 14 | 0010 |
| 15 | 0100 |

LAYOUTS

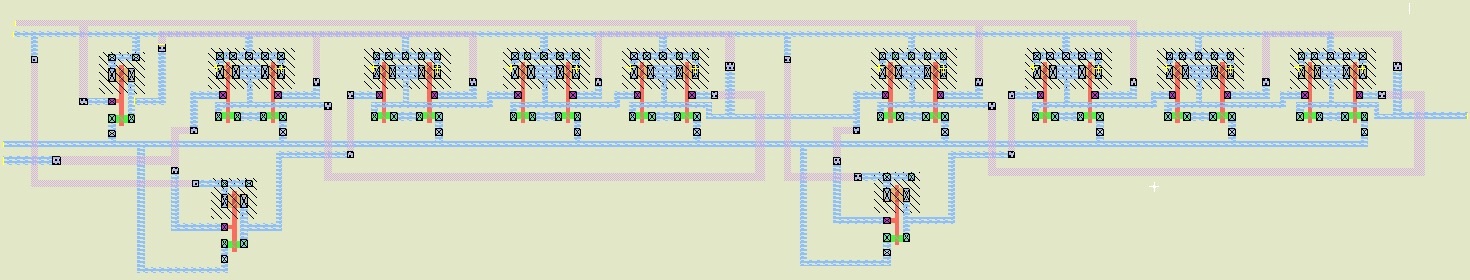
Magic VLSI layout for Inverter



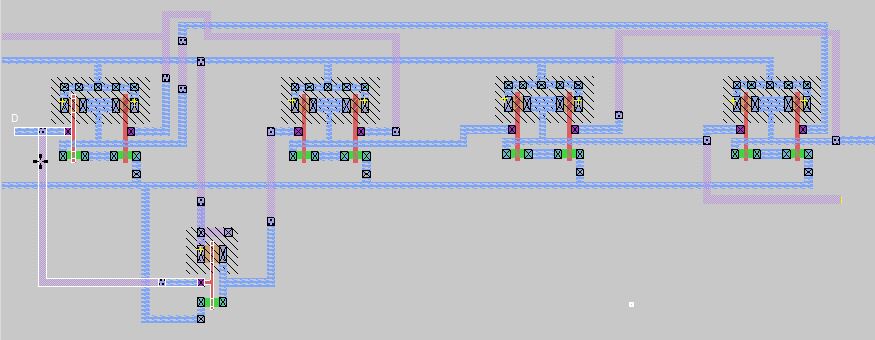
Magic VLSI Layout for Ex-OR Gate



Magic VLSI Layout of Rising Edge D- Flip Flop



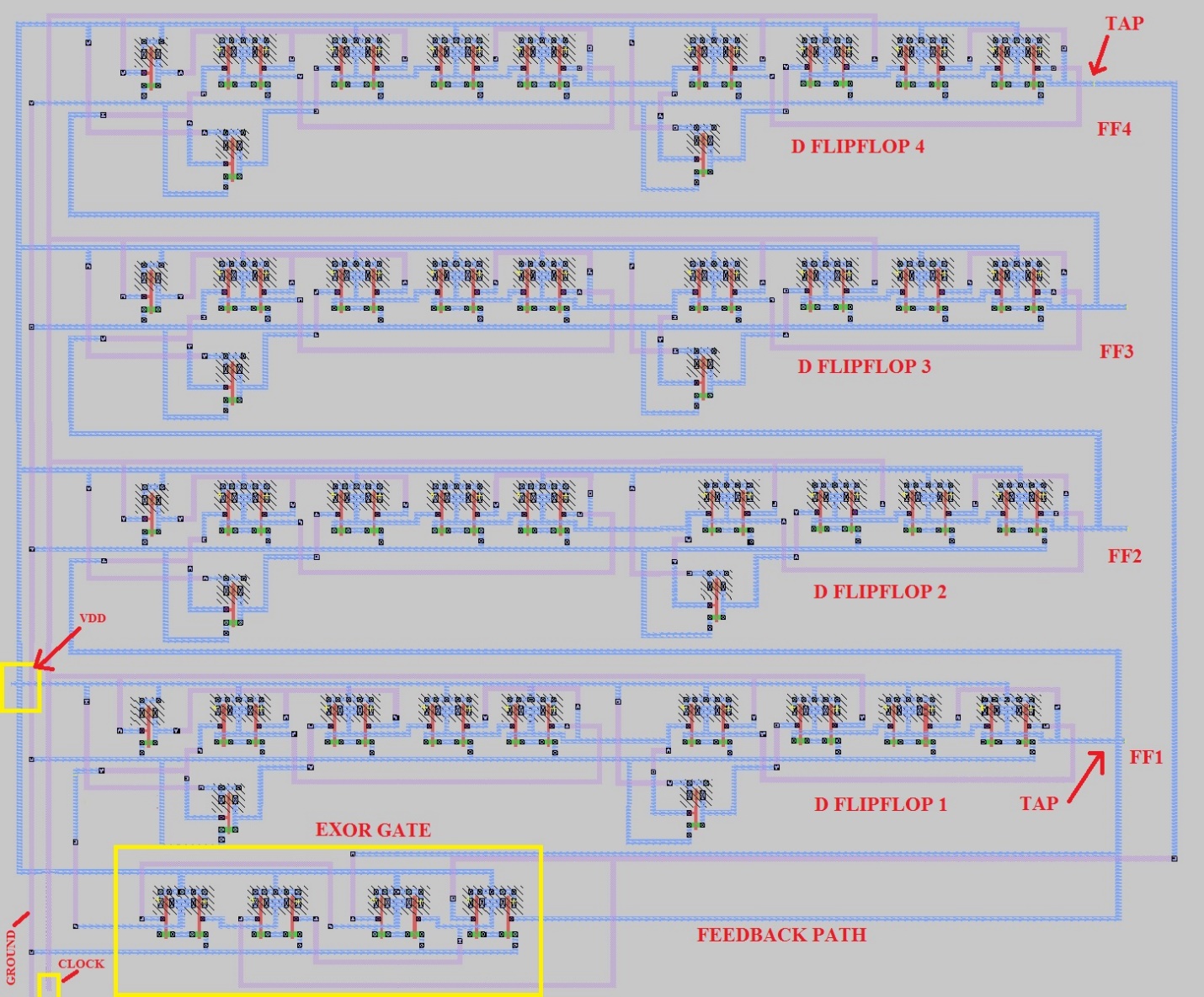
D-Flip Flop Layout



Layout for Final Design:-



Labelled Layout of Final Design:-



**SPICE CODE:**

Spice code extract for Linear Feedback Shift Register.

\* SPICE3 file created from test.ext - technology: scmos .option scale=1u

M1000 a\_n1229\_208# clock vdd vdd pfet w=8 l=2

+ ad=56 pd=30 as=4128 ps=2376

M1001 a\_n1172\_209# FF3 vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1002 vdd a\_n1229\_208# a\_n1172\_209# vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1003 a\_n1075\_209# a\_n1159\_130# vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1004 vdd a\_n1229\_208# a\_n1075\_209# vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1005 a\_n989\_209# a\_n1075\_209# vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1006 vdd a\_n954\_207# a\_n989\_209# vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1007 a\_n954\_207# a\_n989\_209# vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1008 vdd a\_n1172\_209# a\_n954\_207# vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1009 a\_n761\_209# a\_n954\_207# vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1010 vdd clock a\_n761\_209# vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1011 a\_n665\_210# a\_n748\_134# vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1012 vdd clock a\_n665\_210# vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1013 a\_n583\_209# a\_n665\_210# vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1014 vdd FF4 a\_n583\_209# vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1015 FF4 a\_n583\_209# vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1016 vdd a\_n761\_209# FF4 vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1017 a\_n1229\_208# clock gnd Gnd nfet w=4 l=2

+ ad=28 pd=22 as=1344 ps=1056

M1018 a\_n1163\_209# FF3 a\_n1172\_209# Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1019 gnd a\_n1229\_208# a\_n1163\_209# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1020 a\_n1066\_209# a\_n1159\_130# a\_n1075\_209# Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1021 gnd a\_n1229\_208# a\_n1066\_209# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1022 a\_n980\_209# a\_n1075\_209# a\_n989\_209# Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1023 gnd a\_n954\_207# a\_n980\_209# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1024 a\_n906\_209# a\_n989\_209# a\_n954\_207# Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1025 gnd a\_n1172\_209# a\_n906\_209# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1026 a\_n752\_209# a\_n954\_207# a\_n761\_209# Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1027 gnd clock a\_n752\_209# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1028 a\_n656\_210# a\_n748\_134# a\_n665\_210# Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1029 gnd clock a\_n656\_210# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1030 a\_n574\_209# a\_n665\_210# a\_n583\_209# Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1031 gnd FF4 a\_n574\_209# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1032 a\_n492\_209# a\_n583\_209# FF4 Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1033 gnd a\_n761\_209# a\_n492\_209# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1034 a\_n1159\_130# FF3 vdd vdd pfet w=8 l=2

+ ad=56 pd=30 as=0 ps=0

M1035 a\_n748\_134# a\_n954\_207# vdd vdd pfet w=8 l=2

+ ad=56 pd=30 as=0 ps=0

M1036 a\_n748\_134# a\_n954\_207# gnd Gnd nfet w=4 l=2

+ ad=28 pd=22 as=0 ps=0

M1037 a\_n1159\_130# FF3 gnd Gnd nfet w=4 l=2

+ ad=28 pd=22 as=0 ps=0

M1038 a\_n1229\_n64# clock vdd vdd pfet w=8 l=2

+ ad=56 pd=30 as=0 ps=0

M1039 a\_n1172\_n63# FF2 vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1040 vdd a\_n1229\_n64# a\_n1172\_n63# vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1041 a\_n1075\_n63# a\_n1159\_n142# vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1042 vdd a\_n1229\_n64# a\_n1075\_n63# vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1043 a\_n989\_n63# a\_n1075\_n63# vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1044 vdd a\_n954\_n65# a\_n989\_n63# vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1045 a\_n954\_n65# a\_n989\_n63# vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1046 vdd a\_n1172\_n63# a\_n954\_n65# vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1047 a\_n761\_n63# a\_n954\_n65# vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1048 vdd clock a\_n761\_n63# vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1049 a\_n665\_n62# a\_n748\_n138# vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1050 vdd clock a\_n665\_n62# vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1051 a\_n583\_n63# a\_n665\_n62# vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1052 vdd FF3 a\_n583\_n63# vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1053 FF3 a\_n583\_n63# vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1054 vdd a\_n761\_n63# FF3 vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1055 a\_n1229\_n64# clock gnd Gnd nfet w=4 l=2

+ ad=28 pd=22 as=0 ps=0

M1056 a\_n1163\_n63# FF2 a\_n1172\_n63# Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1057 gnd a\_n1229\_n64# a\_n1163\_n63# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1058 a\_n1066\_n63# a\_n1159\_n142# a\_n1075\_n63# Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1059 gnd a\_n1229\_n64# a\_n1066\_n63# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1060 a\_n980\_n63# a\_n1075\_n63# a\_n989\_n63# Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1061 gnd a\_n954\_n65# a\_n980\_n63# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1062 a\_n906\_n63# a\_n989\_n63# a\_n954\_n65# Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1063 gnd a\_n1172\_n63# a\_n906\_n63# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1064 a\_n752\_n63# a\_n954\_n65# a\_n761\_n63# Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1065 gnd clock a\_n752\_n63# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1066 a\_n656\_n62# a\_n748\_n138# a\_n665\_n62# Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1067 gnd clock a\_n656\_n62# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1068 a\_n574\_n63# a\_n665\_n62# a\_n583\_n63# Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1069 gnd FF3 a\_n574\_n63# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1070 a\_n492\_n63# a\_n583\_n63# FF3 Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1071 gnd a\_n761\_n63# a\_n492\_n63# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1072 a\_n1159\_n142# FF2 vdd vdd pfet w=8 l=2

+ ad=56 pd=30 as=0 ps=0

M1073 a\_n748\_n138# a\_n954\_n65# vdd vdd pfet w=8 l=2

+ ad=56 pd=30 as=0 ps=0

M1074 a\_n748\_n138# a\_n954\_n65# gnd Gnd nfet w=4 l=2

+ ad=28 pd=22 as=0 ps=0

M1075 a\_n1159\_n142# FF2 gnd Gnd nfet w=4 l=2

+ ad=28 pd=22 as=0 ps=0

M1076 a\_n1229\_n315# clock vdd vdd pfet w=8 l=2

+ ad=56 pd=30 as=0 ps=0

M1077 a\_n1172\_n314# FF1 vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1078 vdd a\_n1229\_n315# a\_n1172\_n314# vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1079 a\_n1075\_n314# a\_n1159\_n393# vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1080 vdd a\_n1229\_n315# a\_n1075\_n314# vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1081 a\_n989\_n314# a\_n1075\_n314# vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1082 vdd a\_n954\_n316# a\_n989\_n314# vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1083 a\_n954\_n316# a\_n989\_n314# vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1084 vdd a\_n1172\_n314# a\_n954\_n316# vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1085 a\_n761\_n314# a\_n954\_n316# vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1086 vdd clock a\_n761\_n314# vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1087 a\_n665\_n313# a\_n748\_n389# vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1088 vdd clock a\_n665\_n313# vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1089 a\_n583\_n314# a\_n665\_n313# vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1090 vdd FF2 a\_n583\_n314# vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1091 FF2 a\_n583\_n314# vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1092 vdd a\_n761\_n314# FF2 vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1093 a\_n1229\_n315# clock gnd Gnd nfet w=4 l=2

+ ad=28 pd=22 as=0 ps=0

M1094 a\_n1163\_n314# FF1 a\_n1172\_n314# Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1095 gnd a\_n1229\_n315# a\_n1163\_n314# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1096 a\_n1066\_n314# a\_n1159\_n393# a\_n1075\_n314# Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1097 gnd a\_n1229\_n315# a\_n1066\_n314# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1098 a\_n980\_n314# a\_n1075\_n314# a\_n989\_n314# Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1099 gnd a\_n954\_n316# a\_n980\_n314# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1100 a\_n906\_n314# a\_n989\_n314# a\_n954\_n316# Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1101 gnd a\_n1172\_n314# a\_n906\_n314# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1102 a\_n752\_n314# a\_n954\_n316# a\_n761\_n314# Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1103 gnd clock a\_n752\_n314# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1104 a\_n656\_n313# a\_n748\_n389# a\_n665\_n313# Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1105 gnd clock a\_n656\_n313# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1106 a\_n574\_n314# a\_n665\_n313# a\_n583\_n314# Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1107 gnd FF2 a\_n574\_n314# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1108 a\_n492\_n314# a\_n583\_n314# FF2 Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1109 gnd a\_n761\_n314# a\_n492\_n314# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1110 a\_n1159\_n393# FF1 vdd vdd pfet w=8 l=2

+ ad=56 pd=30 as=0 ps=0

M1111 a\_n748\_n389# a\_n954\_n316# vdd vdd pfet w=8 l=2

+ ad=56 pd=30 as=0 ps=0

M1112 a\_n748\_n389# a\_n954\_n316# gnd Gnd nfet w=4 l=2

+ ad=28 pd=22 as=0 ps=0

M1113 a\_n1159\_n393# FF1 gnd Gnd nfet w=4 l=2

+ ad=28 pd=22 as=0 ps=0

M1114 a\_n1230\_n563# clock vdd vdd pfet w=8 l=2

+ ad=56 pd=30 as=0 ps=0

M1115 a\_n1173\_n562# D vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1116 vdd a\_n1230\_n563# a\_n1173\_n562# vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1117 a\_n1076\_n562# a\_n1160\_n641# vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1118 vdd a\_n1230\_n563# a\_n1076\_n562# vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1119 a\_n990\_n562# a\_n1076\_n562# vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1120 vdd a\_n955\_n564# a\_n990\_n562# vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1121 a\_n955\_n564# a\_n990\_n562# vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1122 vdd a\_n1173\_n562# a\_n955\_n564# vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1123 a\_n762\_n562# a\_n955\_n564# vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1124 vdd clock a\_n762\_n562# vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1125 a\_n666\_n561# a\_n749\_n637# vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1126 vdd clock a\_n666\_n561# vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1127 a\_n584\_n562# a\_n666\_n561# vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1128 vdd FF1 a\_n584\_n562# vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1129 FF1 a\_n584\_n562# vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1130 vdd a\_n762\_n562# FF1 vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1131 a\_n1230\_n563# clock gnd Gnd nfet w=4 l=2

+ ad=28 pd=22 as=0 ps=0

M1132 a\_n1164\_n562# D a\_n1173\_n562# Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1133 gnd a\_n1230\_n563# a\_n1164\_n562# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1134 a\_n1067\_n562# a\_n1160\_n641# a\_n1076\_n562# Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1135 gnd a\_n1230\_n563# a\_n1067\_n562# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1136 a\_n981\_n562# a\_n1076\_n562# a\_n990\_n562# Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1137 gnd a\_n955\_n564# a\_n981\_n562# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1138 a\_n907\_n562# a\_n990\_n562# a\_n955\_n564# Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1139 gnd a\_n1173\_n562# a\_n907\_n562# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1140 a\_n753\_n562# a\_n955\_n564# a\_n762\_n562# Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1141 gnd clock a\_n753\_n562# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1142 a\_n657\_n561# a\_n749\_n637# a\_n666\_n561# Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1143 gnd clock a\_n657\_n561# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1144 a\_n575\_n562# a\_n666\_n561# a\_n584\_n562# Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1145 gnd FF1 a\_n575\_n562# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1146 a\_n493\_n562# a\_n584\_n562# FF1 Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1147 gnd a\_n762\_n562# a\_n493\_n562# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1148 a\_n1160\_n641# D vdd vdd pfet w=8 l=2

+ ad=56 pd=30 as=0 ps=0

M1149 a\_n749\_n637# a\_n955\_n564# vdd vdd pfet w=8 l=2

+ ad=56 pd=30 as=0 ps=0

M1150 a\_n749\_n637# a\_n955\_n564# gnd Gnd nfet w=4 l=2

+ ad=28 pd=22 as=0 ps=0

M1151 a\_n1160\_n641# D gnd Gnd nfet w=4 l=2

+ ad=28 pd=22 as=0 ps=0

M1152 D a\_n1220\_n721# vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1153 vdd a\_n1188\_n737# D vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1154 a\_n1188\_n737# FF4 vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1155 vdd a\_n1105\_n737# a\_n1188\_n737# vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1156 a\_n1220\_n721# FF1 vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1157 vdd a\_n1105\_n737# a\_n1220\_n721# vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1158 a\_n1105\_n737# FF4 vdd vdd pfet w=8 l=2

+ ad=96 pd=56 as=0 ps=0

M1159 vdd FF1 a\_n1105\_n737# vdd pfet w=8 l=2

+ ad=0 pd=0 as=0 ps=0

M1160 a\_n1214\_n735# a\_n1220\_n721# D Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1161 gnd a\_n1188\_n737# a\_n1214\_n735# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1162 a\_n1131\_n735# FF4 a\_n1188\_n737# Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1163 gnd a\_n1105\_n737# a\_n1131\_n735# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1164 a\_n1024\_n735# FF1 a\_n1220\_n721# Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1165 gnd a\_n1105\_n737# a\_n1024\_n735# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

M1166 a\_n941\_n735# FF4 a\_n1105\_n737# Gnd nfet w=4 l=2

+ ad=56 pd=44 as=28 ps=22

M1167 gnd FF1 a\_n941\_n735# Gnd nfet w=4 l=2

+ ad=0 pd=0 as=0 ps=0

C0 vdd gnd! 1.4fF

C1 clock gnd! 0.5fF

rgnd gnd 0 1e-6

v1 vdd gnd 5

v2 clock gnd pulse(0 5 30ns 0.5ps 0.5ps 20ns 30ns)

.trans 1ns 1000ns

.MODEL nfet NMOS ( kp=4.32e-4 vt0=0.4)

.MODEL pfet pMOS ( kp=1.12e-4 vt0=-0.4)

.print dc v(D)

.print dc v(FF1)

.print dc v(FF2)

.print dc v(FF3)

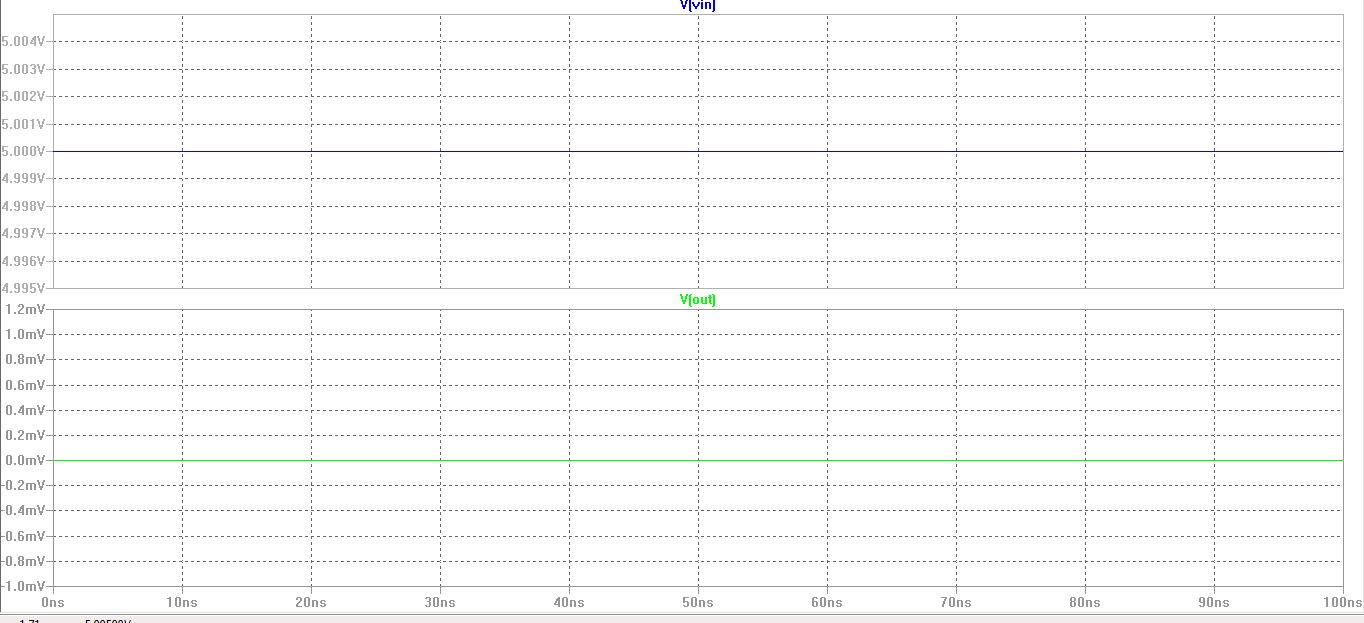
.print dc v(FF4)

.print dc v(clock)

.end

**WAVEFORMS**

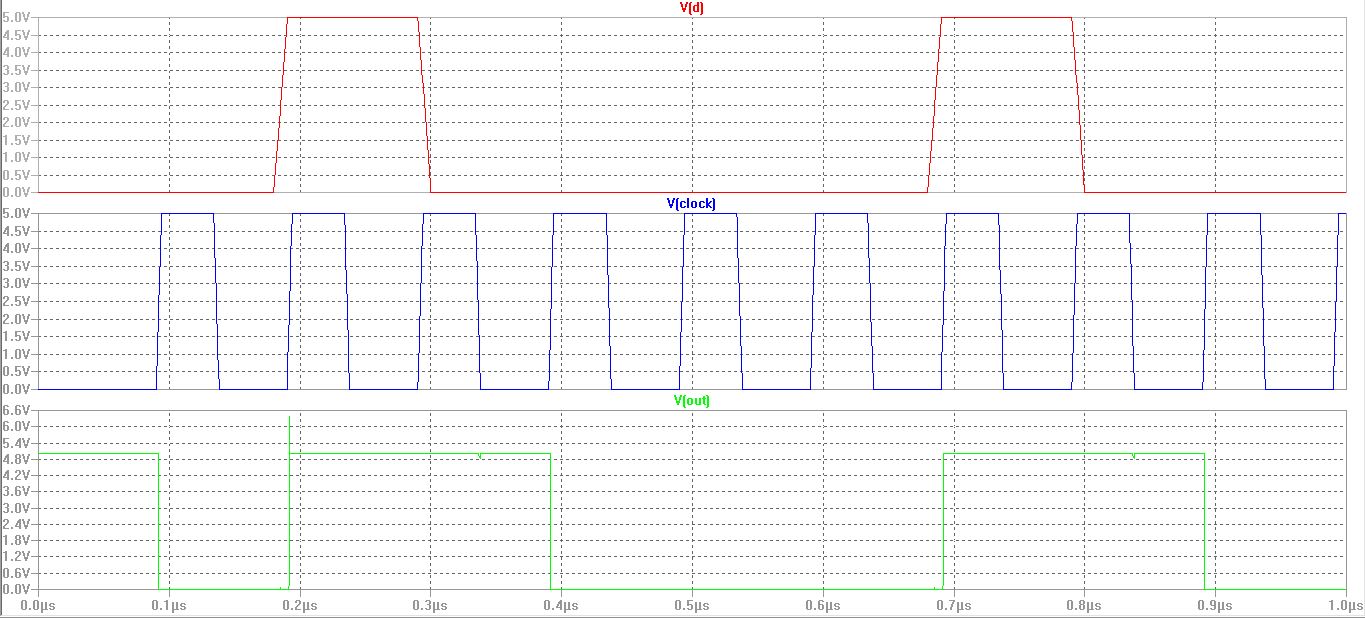
Output Waveform of Inverter



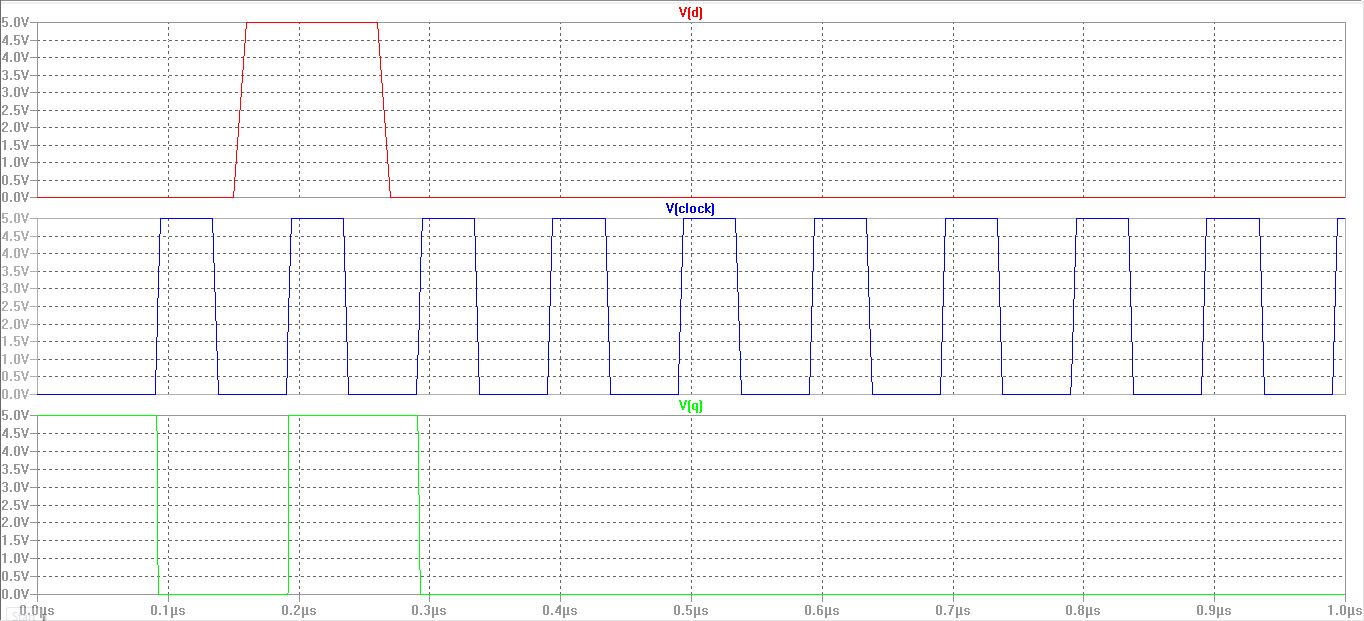
Output waveform of EX-OR Gate



Output of D-Flip Flop Rising Edge



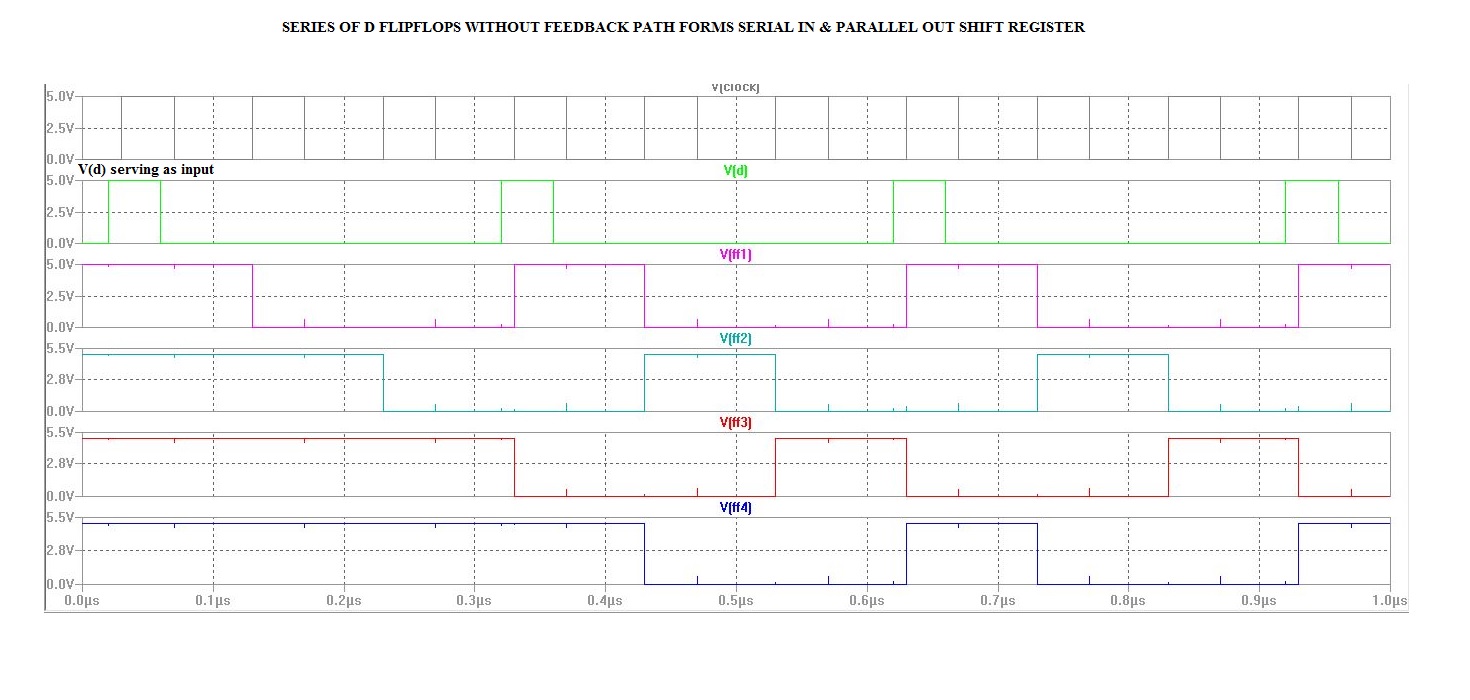
Output of D-latch with clock



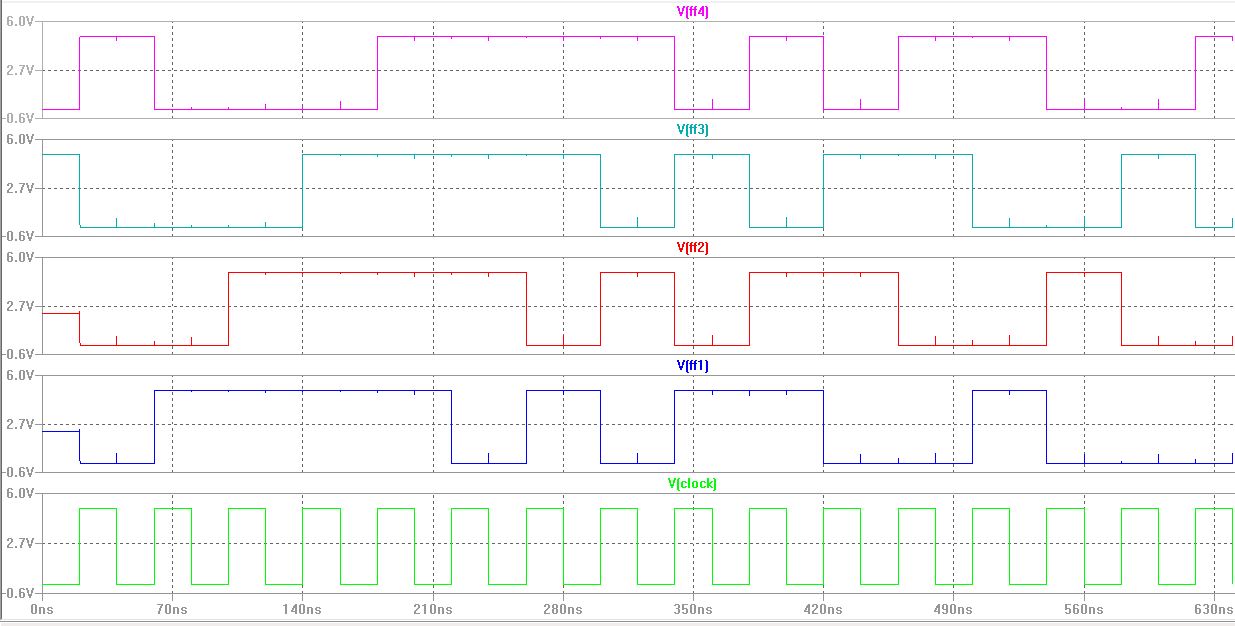
Output of D-latch with inverted clock



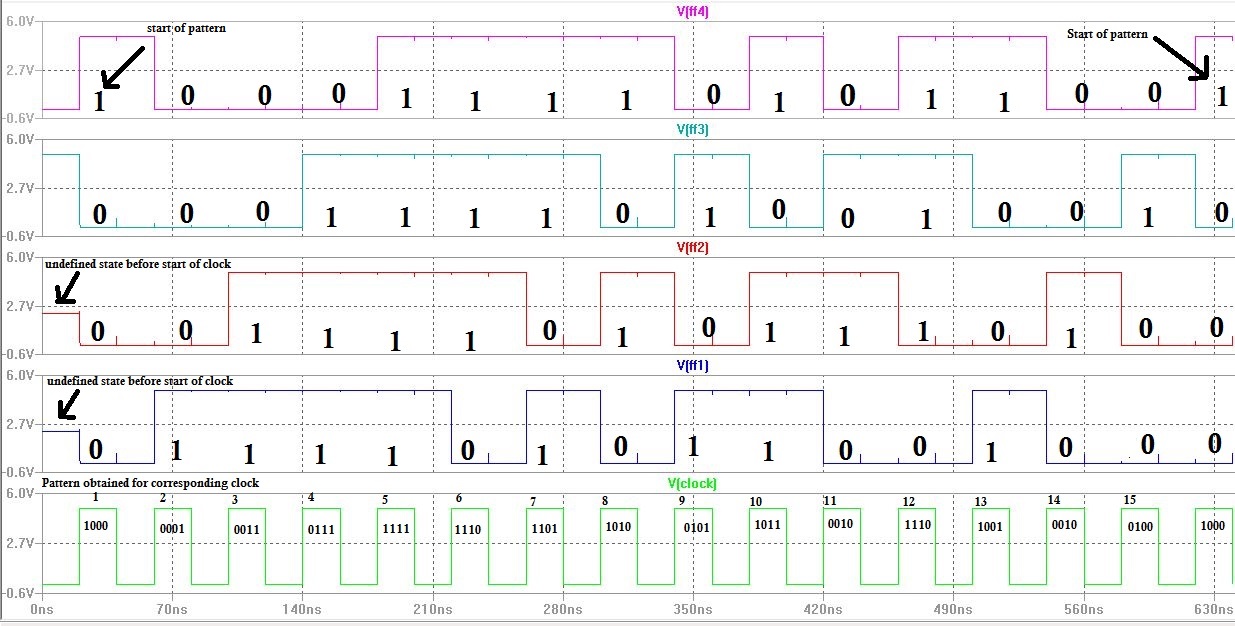
Output of series of D-Flip flops without feedback



Output waveform of LFSR



Labelled Output waveform of LFSR



**CONCLUSION:**

The Magic VLSI layout for 4 bit Linear Feedback Shift Register is successfully drawn using Magic VLSI software and the circuit’s spice extract is simulated using LtSpice. The simulation results obtained successfully verifies the operation of 4 bit Linear Feedback Shift Register.

This project includes implementation of logics such as Ex-OR, Inverter and D flip flop using universal gates NAND and NOR. So these concepts can be extended to a wide area of digital logic circuits implementation such as multipliers, MAC (Multiplier and Accumulator) unit, and memory units viz., RAM, ROM, SRAM, DRAM and several other digital processors’ units.

REFERENCES:

1) N. A. Doshi, S. B. Dhobale, and S. R. Kakade," LFSR Counter Implementation in CMOS VLSI", World Academy of Science, Engineering and Technology 48 2008173

2) <http://waset.org/publications/4463/lfsr-counter-implementation-in-cmos-vlsi>

**ACKNOWLEDGEMENTS:**

1. ECC lab and facilities.
2. LtSpice online tutorials.
3. Prof. E. Joe Charlson.