**CARRY LOOK AHEAD ADDER**

****

**DIGITAL VLSI DESIGN**

**PROJECT 3**

**NAME : GAUTHAM G MULAY**

**SRN :PES1UG20EC070**

**BRANCH : ECE**

**SEMESTER : 4**

**SECTION : B**

**Introduction:**

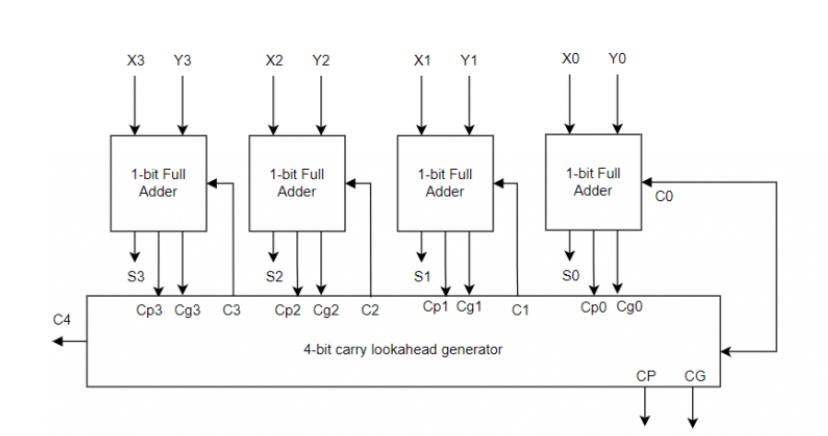
A carry-lookahead [adder](https://www.watelectronics.com/what-is-full-adder-circuit-using-basic-gates/) is called a fast adder that augments the speed required for determining carry bits. A **carry lookahead adder definition** is it is the faster circuit in performing binary addition by using the concepts of Carry Generate and Carry Propagate. A CLA is termed as the successor of a ripple carry adder. A CLA circuit minimizes the propagation delay time through the implementation of complex circuitry.

The operation of carry lookahead is based on two scenarios:

* Calculate every digit position to know whether that position is propagating a carry bit that comes from its right position.
* Then combine the calculated values to produce the output for every set of digits where the group generates a propagation bit that comes from the right position.

**Methodology**

**Logical Block diagram:**



Fig(1)

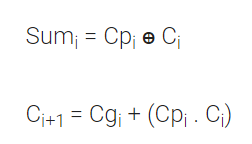
Carry lookahead adders operate by generating two bits called Carry Propagate and Carry Generate which are represented by Cp and Cg. The Cp bit gets propagated to the next stage and the Cg bit is used for generating the output carry bit and this is independent of the input carry bit. The above figure shows the 4-bit **carry lookahead adder architecture**.

For the construction of carry lookahead adder, we need two Boolean expressions which are for **carry lookahead adder formula** for carry propagate Cp and carry generate Cg.



`

With the above expressions, the sum and carry at the output can be given a



With the above fundamental equations, the Boolean expression for carry output at every stage can be known. So

C1 = Cg0 + (Cp0 . C0)

C2 = Cg1 + (Cp1 . C1) = Cg1 + (Cp1 . [Cg0 + (Cp0 . C0)])

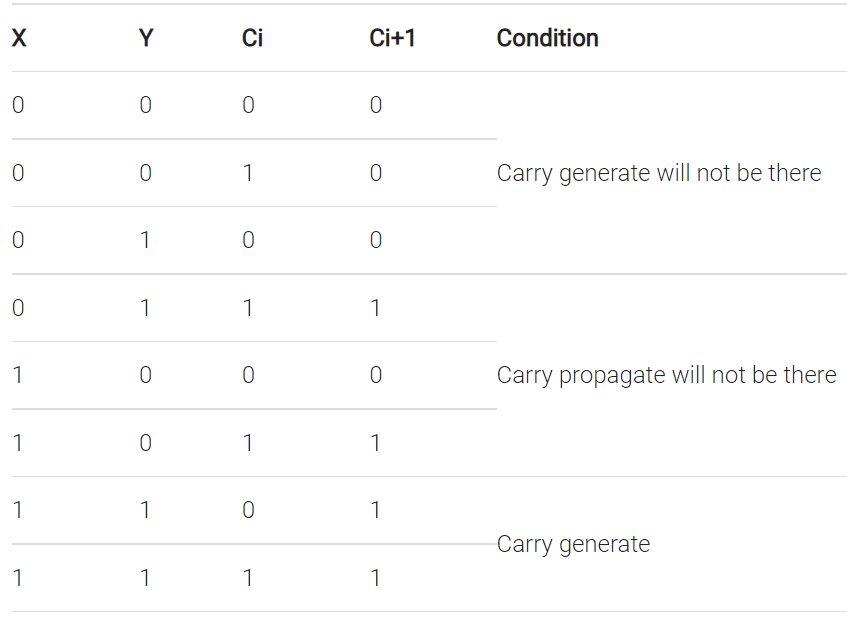
* Cg1 + Cp1 . Cg0 + Cp1 . Cp0 . C0

C3 = Cg2 + (Cp2 . C2)

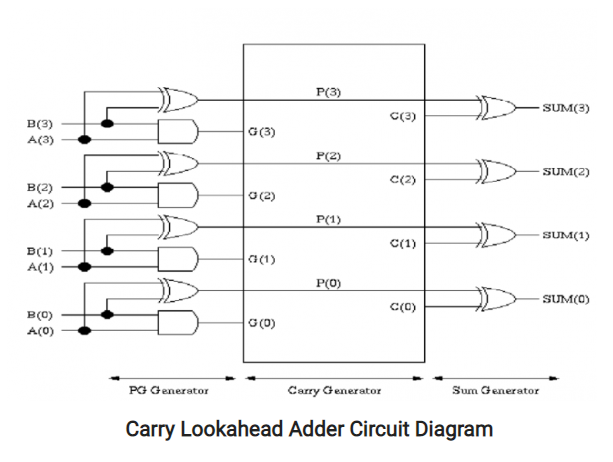
* Cg2 + (Cp2 . [Cg1 + Cp1 . Cg0 + Cp1 . Cp0 . C0])

C4 = Cg3 + (Cp3 . C3)

* Cg3 + (Cp3 . Cg2 + (Cp2 . [Cg1 + Cp1 . Cg0 + Cp1 . Cp0 . C0])



 the equations are applied using AND and OR gates which gives the **carry lookahead adder circuit diagram**.

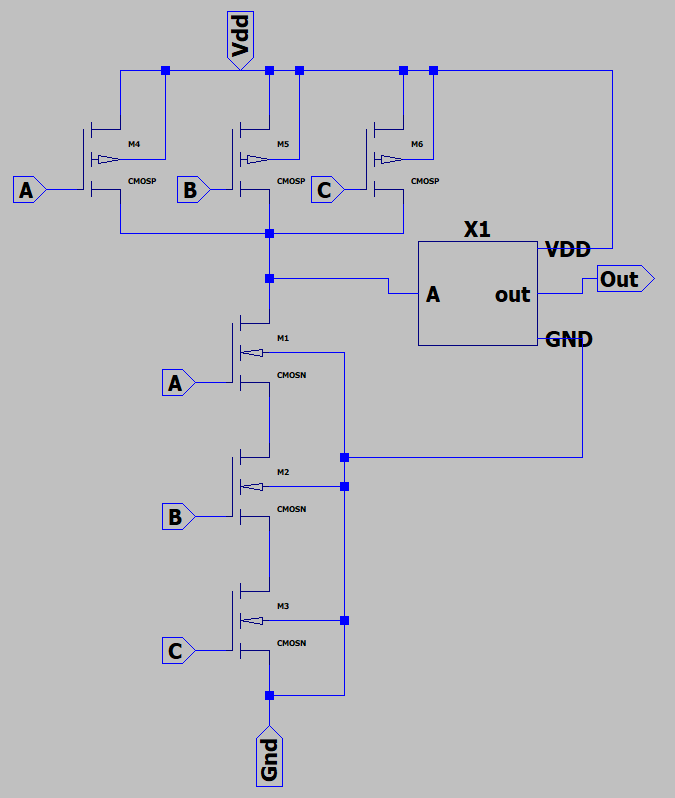


Fig(2)

**Components:**

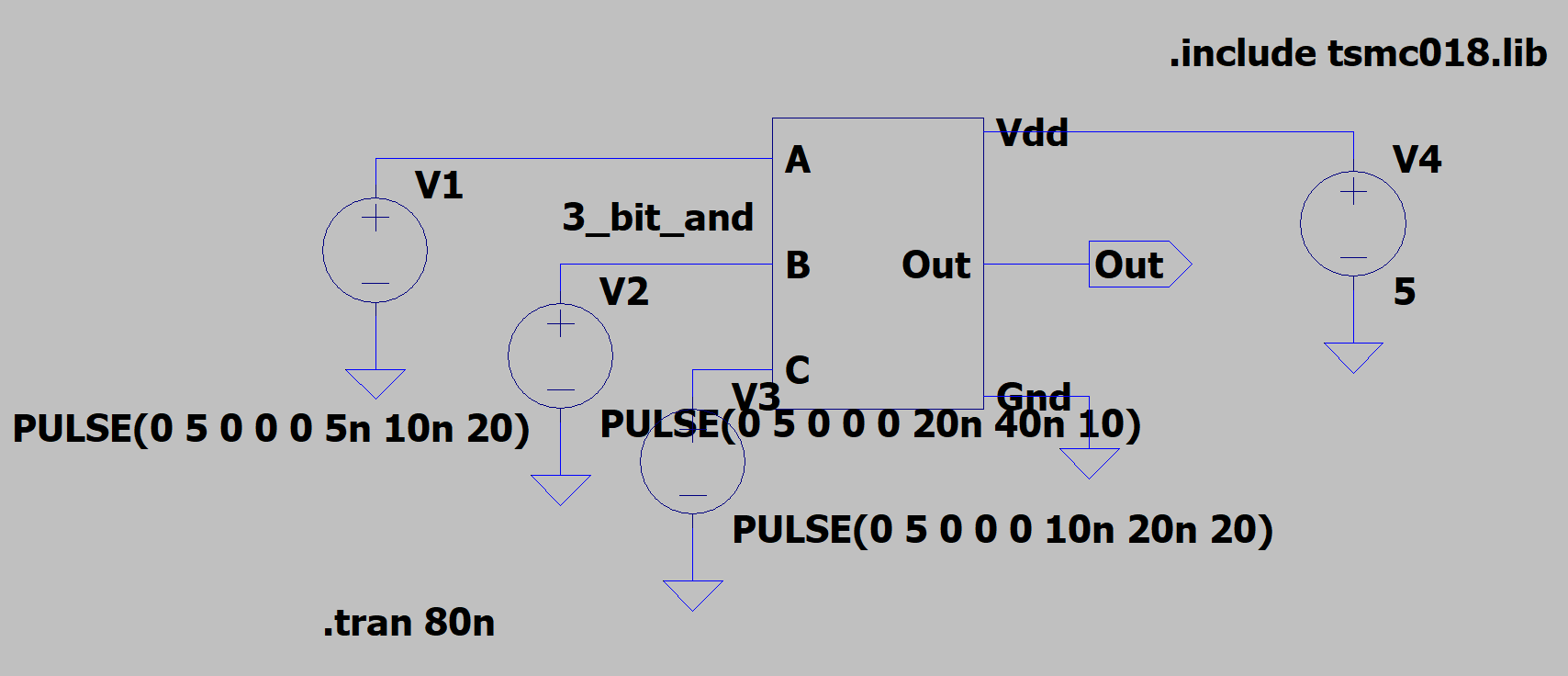
* 1. 3 bit AND:

Design:



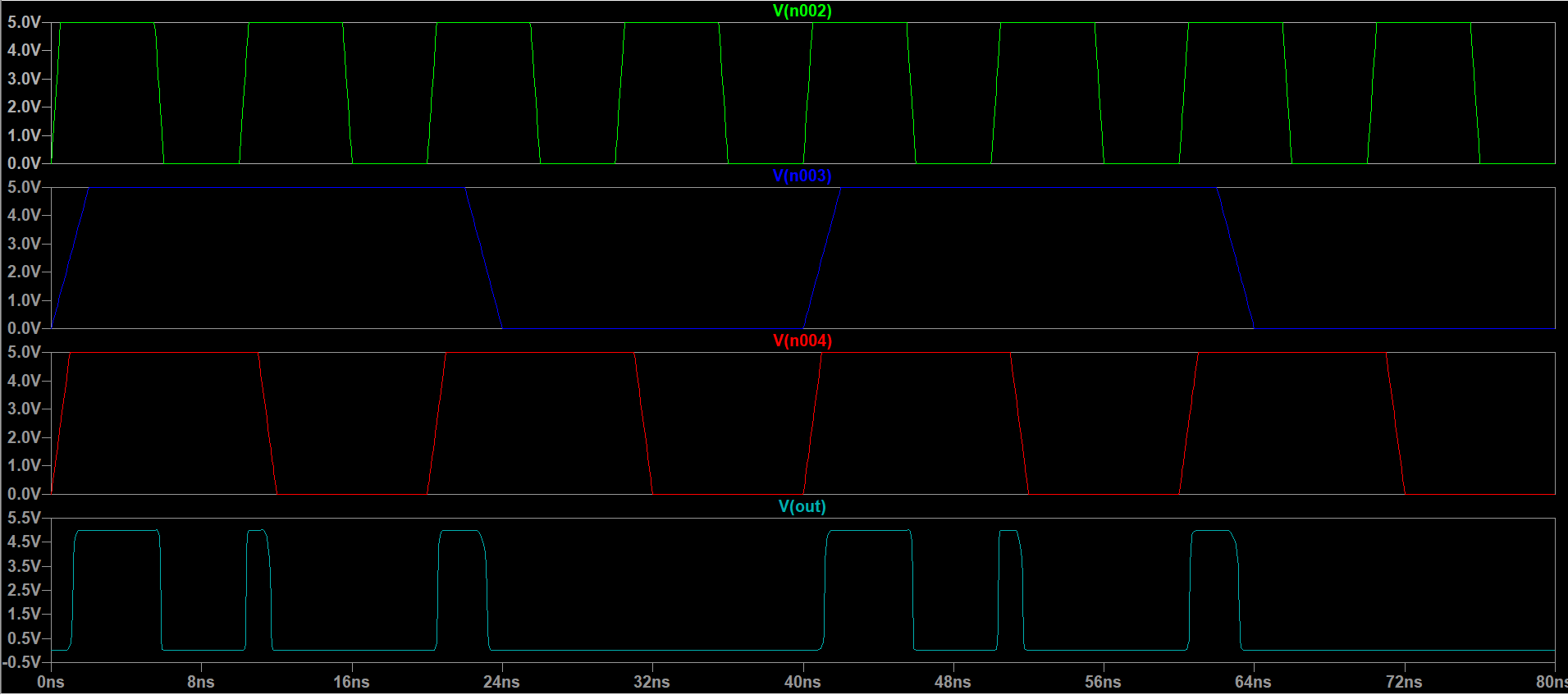
Fig(3)

Simulation:



Fig(4)

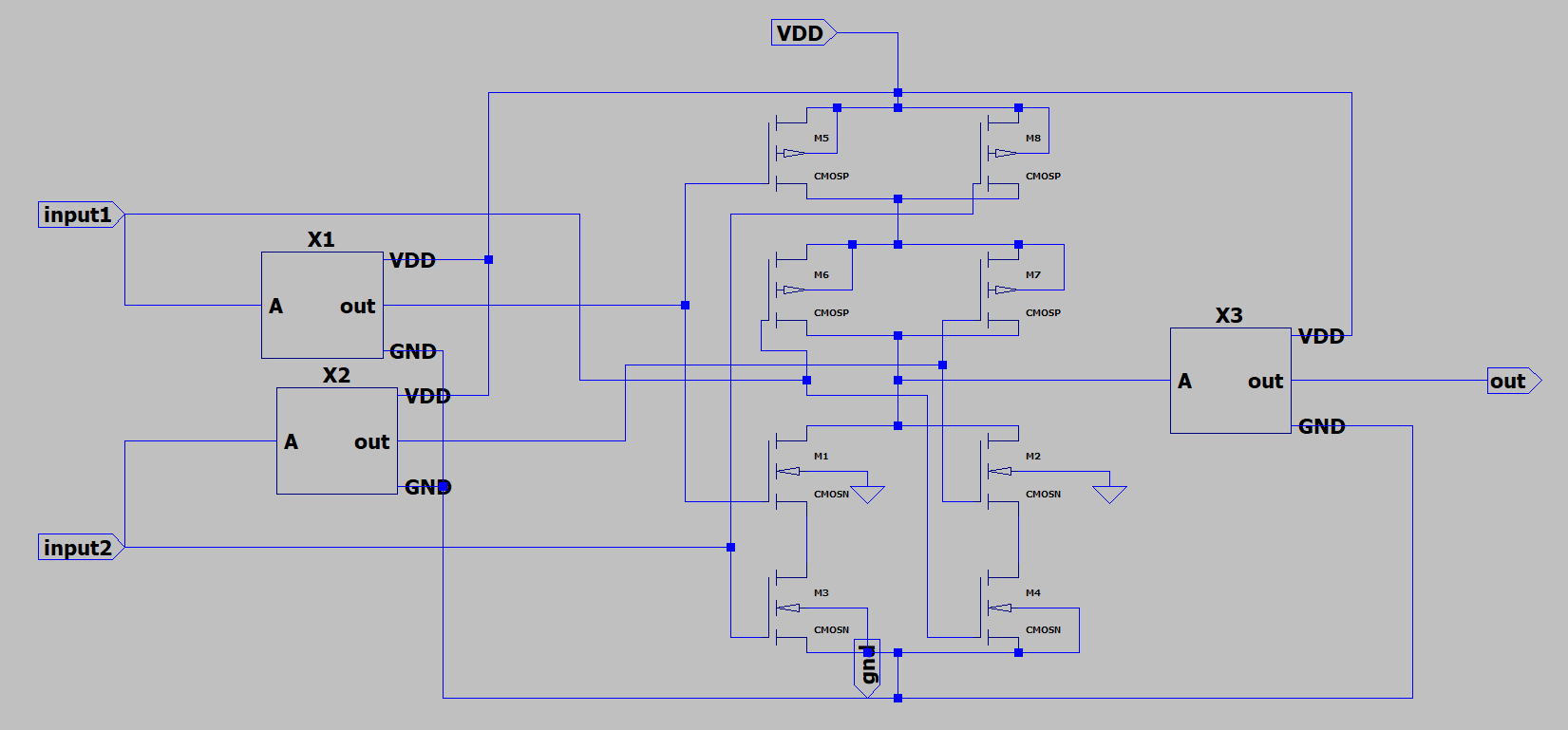
Result:



Fig(5)

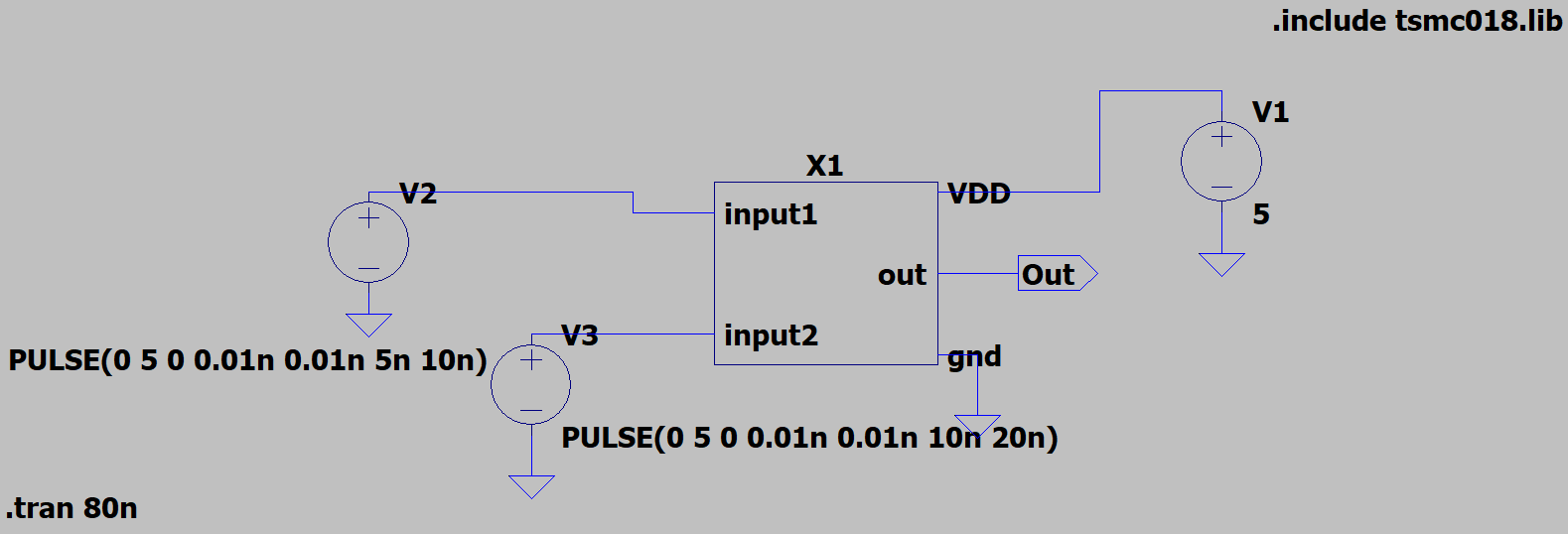
* 1. 2 bit XOR:

Design:



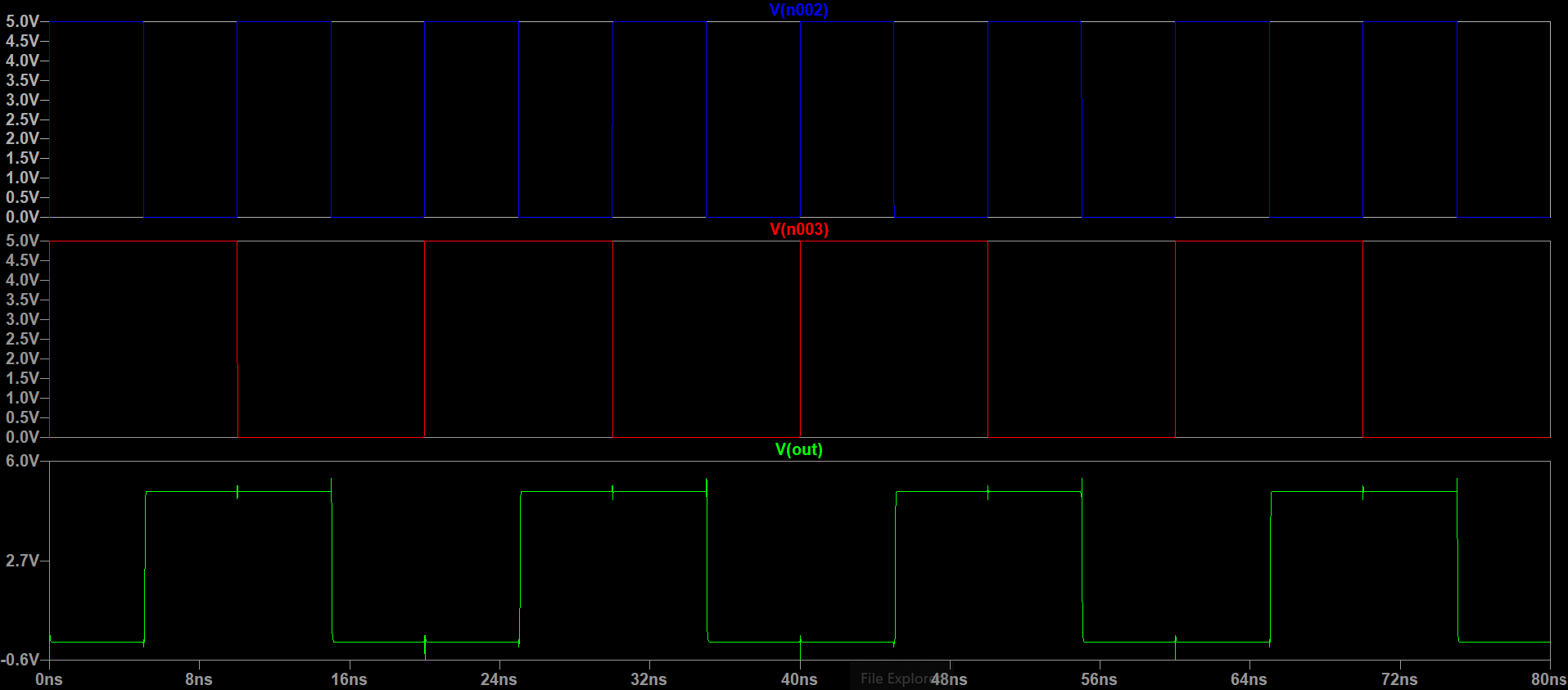
Fig(6)

Simulation:



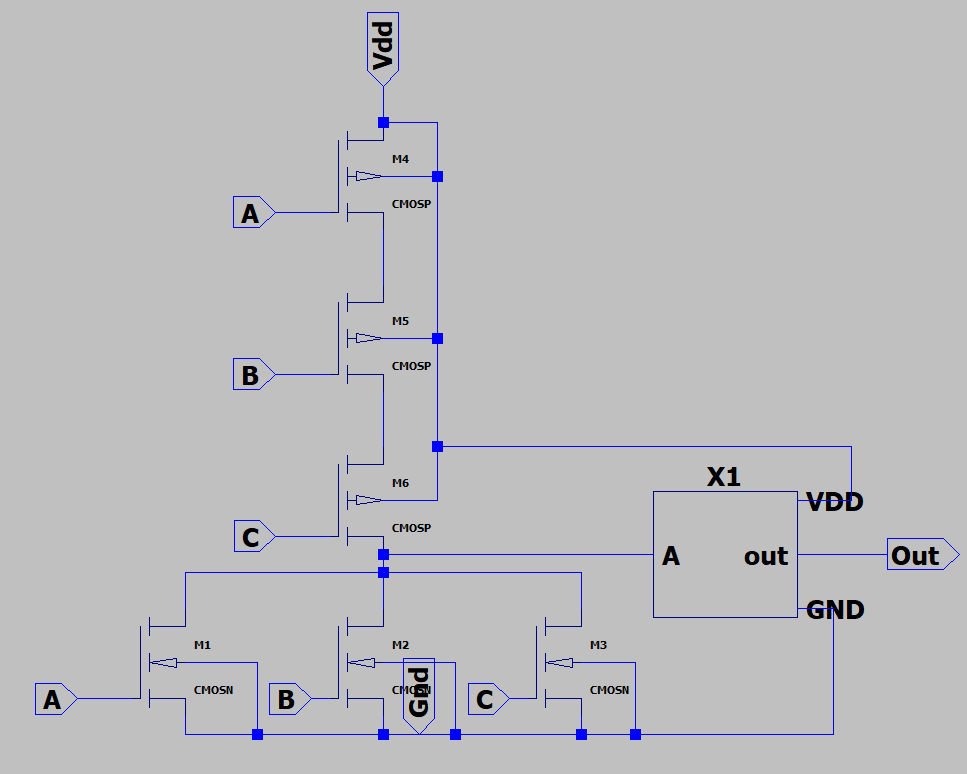
Fig(7)

Result:



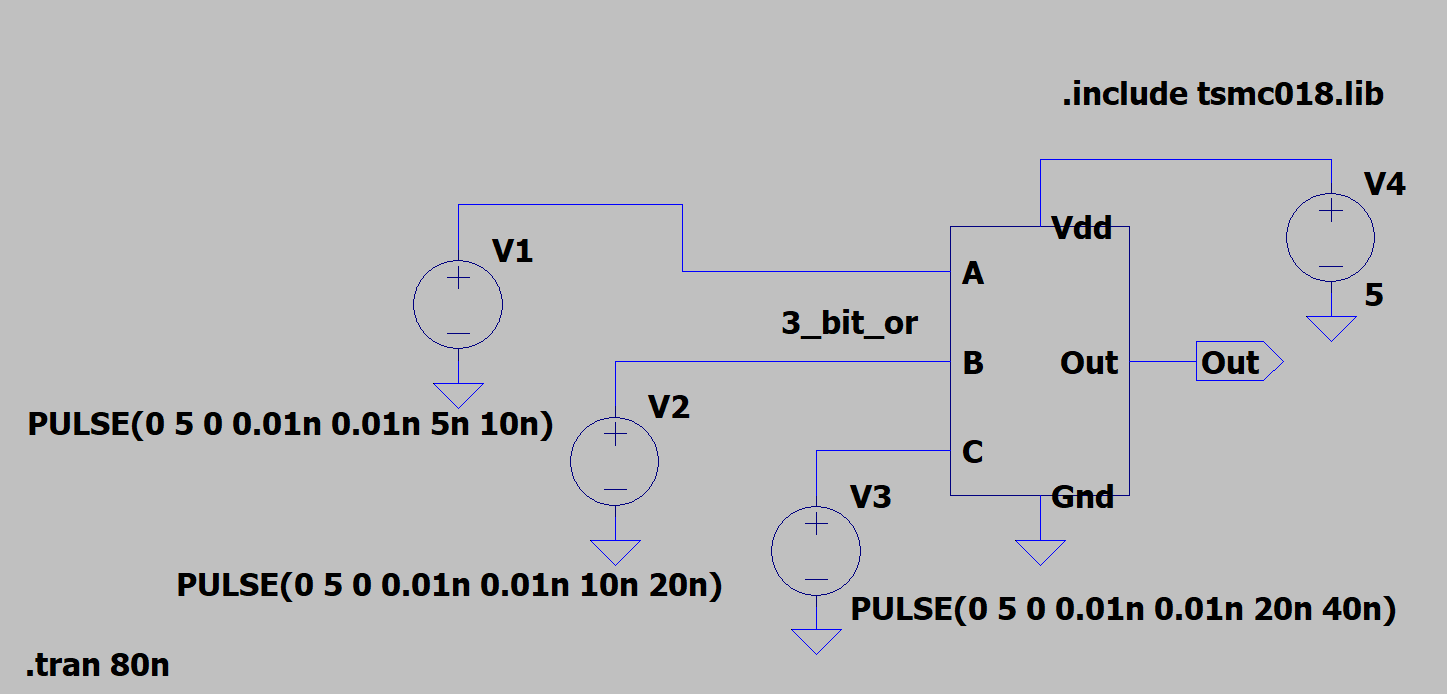
* 1. 3 bit OR:

Design:



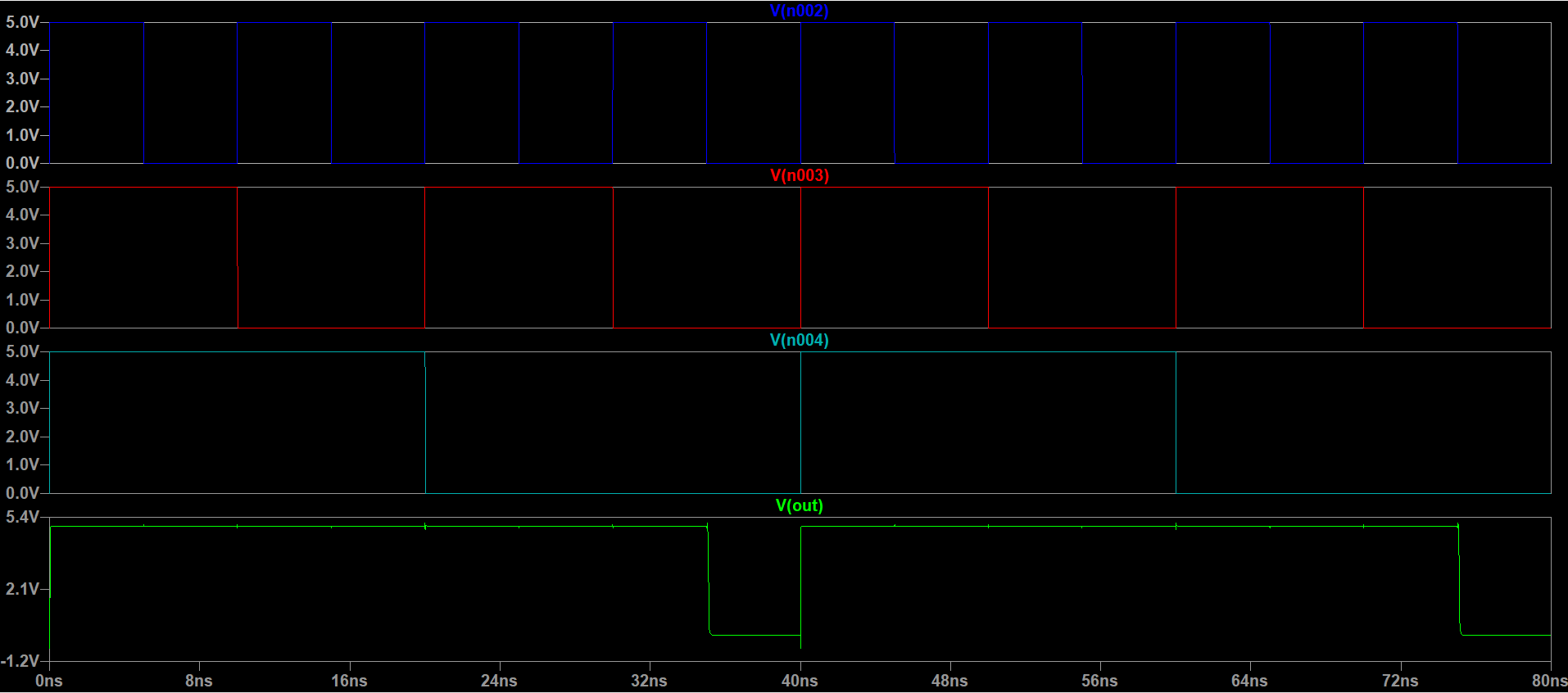
Fig(9)

Simulation:

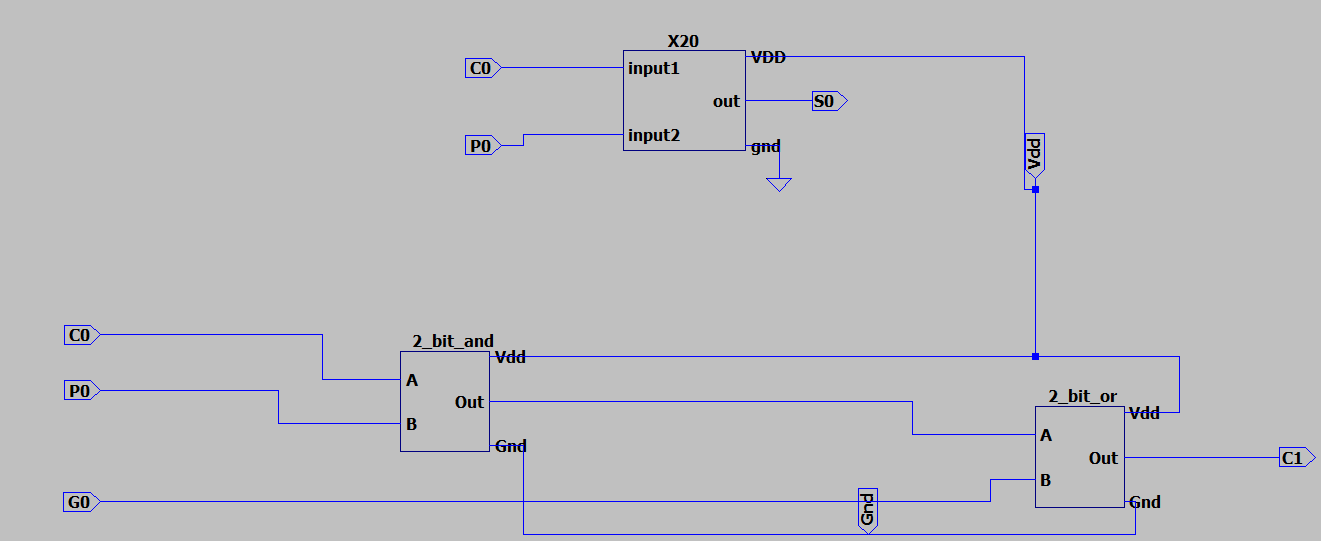


Fig(10)

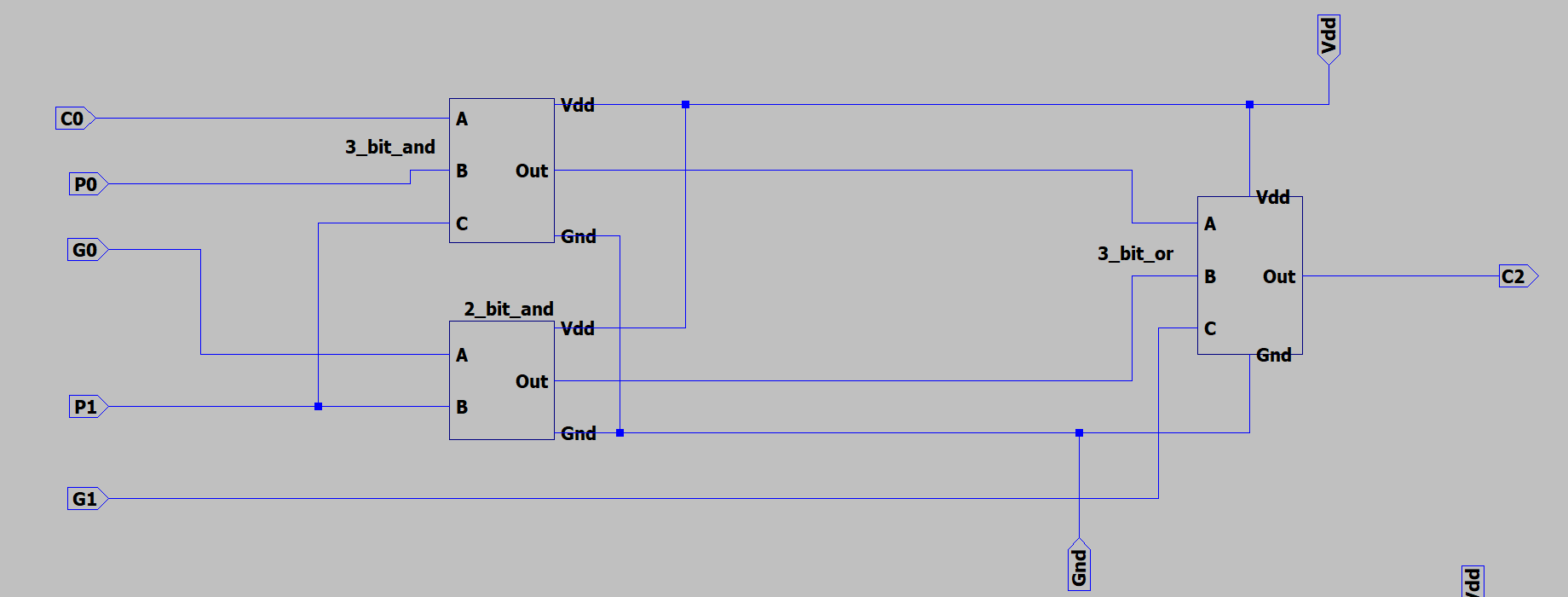
Result:



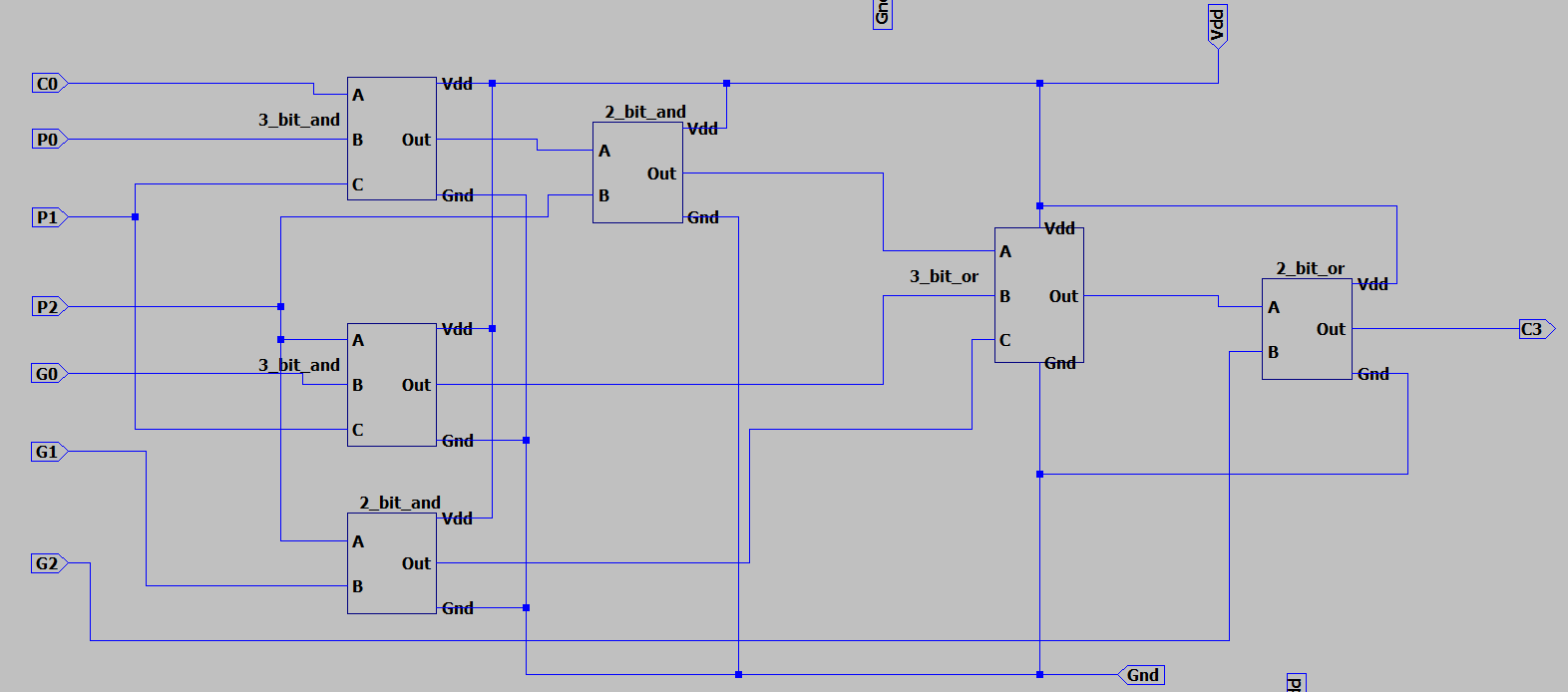
**Carry calculations:**



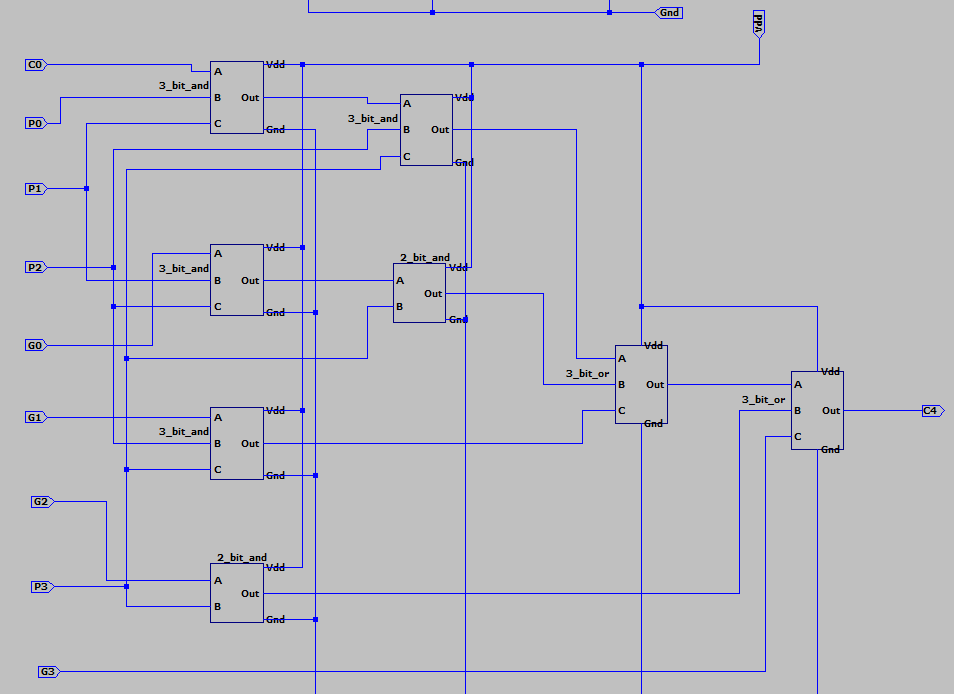
Fig(11)



Fig(12)

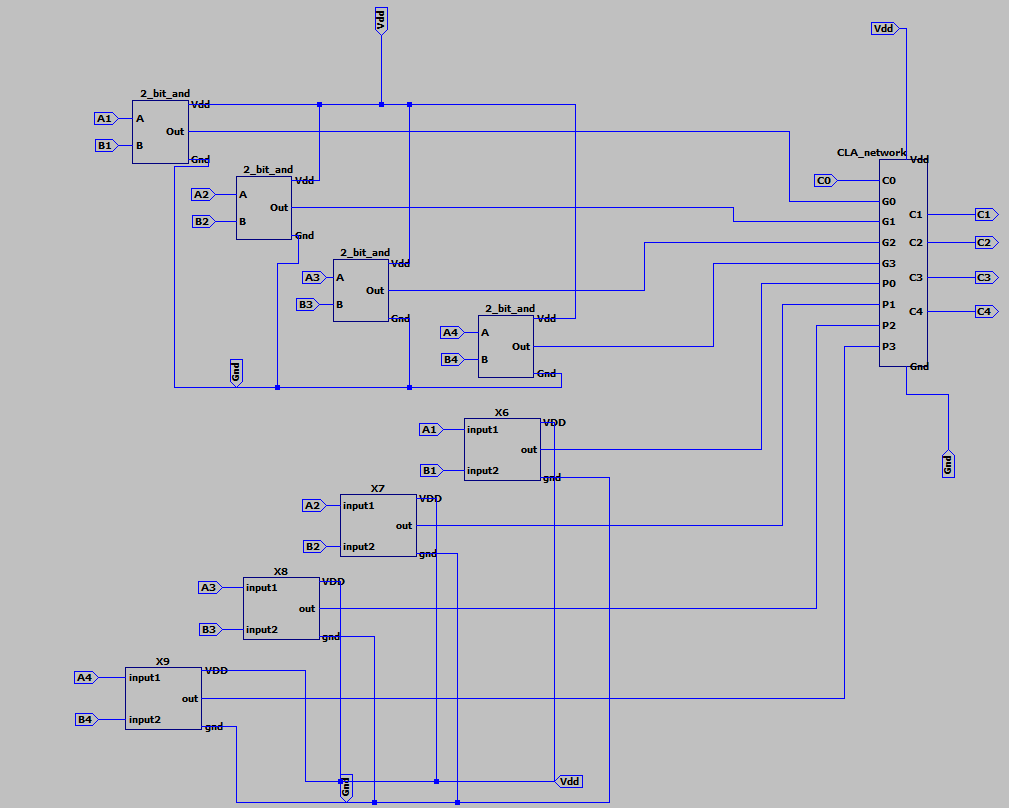


Fig(13)



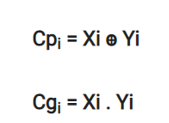
Fig(14)

Here the figures c to f are used the calculate the carryout C1 to C4 respectively

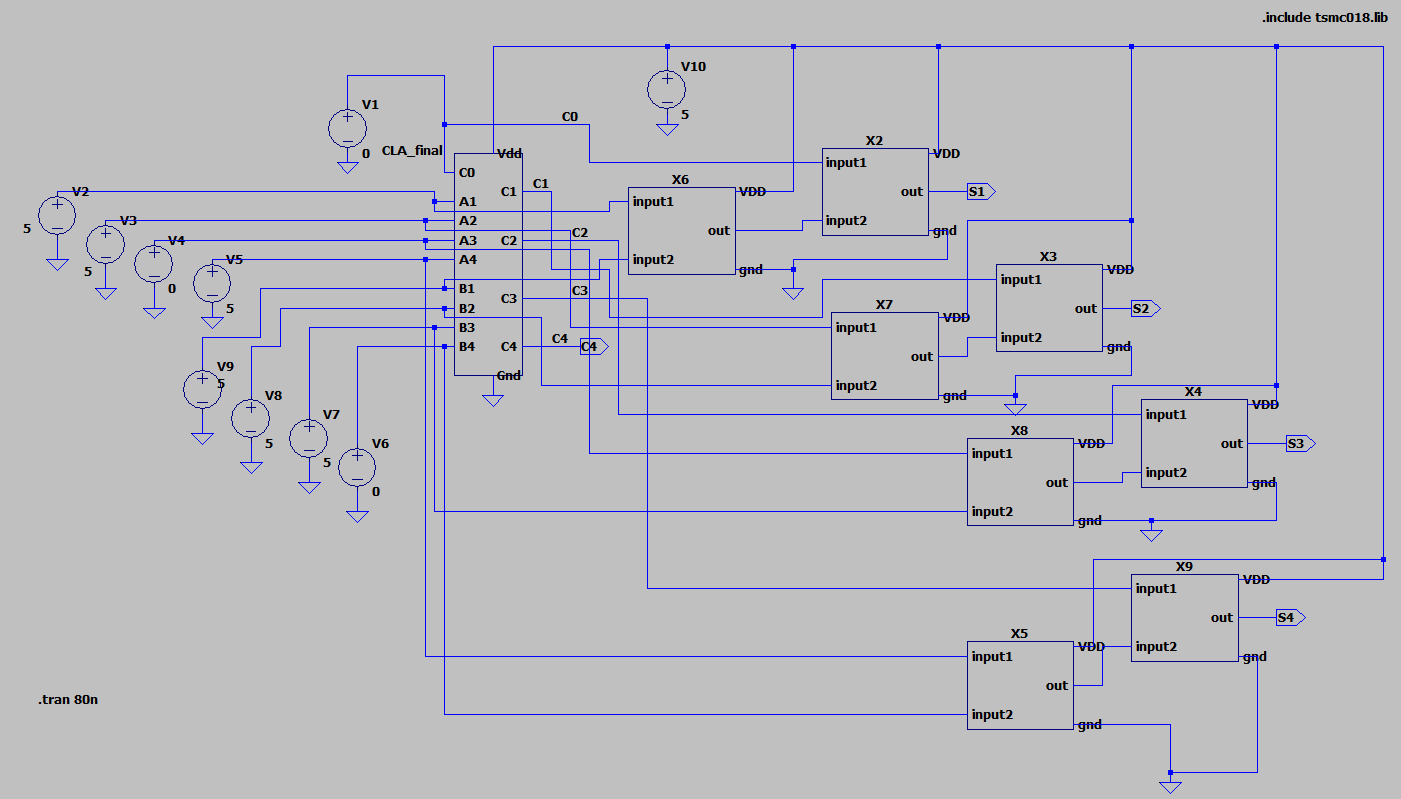


Fig(15)

The figure G implements the formulae for generate and propagate which is given below:



**Final circuit diagram:**



Fig(16)

**Simulation results:**

* 1. CASE 1:

A=1001

B=1100

C0=0



From the simulation:

S=0101

Carryout=1

* 1. CASE 2

A=1111

B=1111

C0=1



From the above simulation:

S=1111

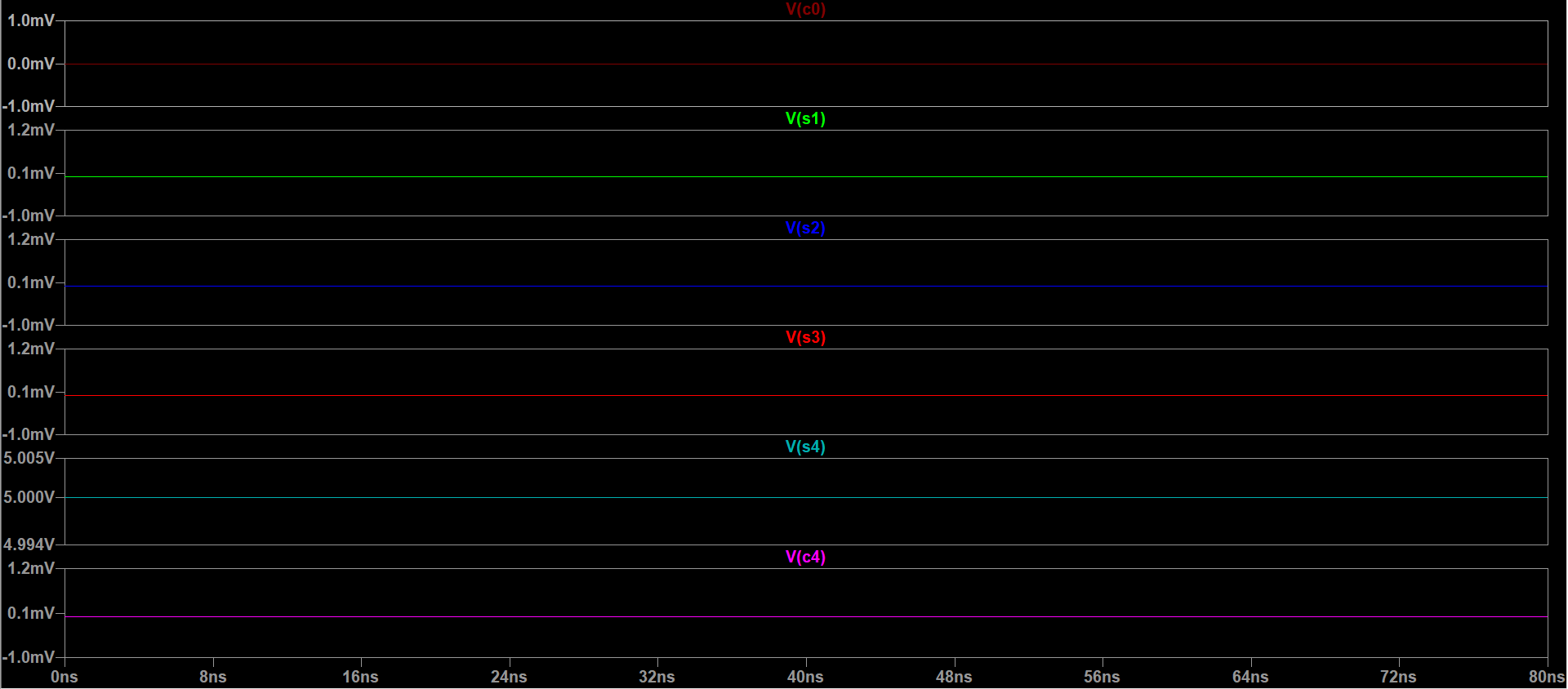
Carryout=1

* 1. CASE 3:

A=0110

B=0010

C0=0



From the above simulation:

S=1000

Carryout=0

* 1. CASE 4:

A=1110

B=0110

Co=0



From the above simulation result

S=0100

Carryout=1

* 1. CASE 5

A=0010 B=0111 Co=1



From the above result

S=1010 Carryout=0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Simulation No. | A | B | C0 | Sum | Carry out |
| 1 | 1001 | 1100 | 0 | 0101 | 1 |
| 2 | 1111 | 1111 | 1 | 1111 | 1 |
| 3 | 0110 | 0010 | 0 | 1000 | 0 |
| 4 | 1110 | 0110 | 0 | 0100 | 1 |
| 5 | 0010 | 0111 | 1 | 1010 | 0 |

Table(1)

**Application:**

High-speed Carry Look-ahead Adders are used as implemented as IC’s. Hence, it is easy to embed the adder in circuits. By combining two or more adders’ calculations of higher bit Boolean functions can be done easily. Here the increase in the number of gates is also moderate when used for higher bits.

In this adder, the propagation delay is reduced. The carry output at any stage is dependent only on the initial carry bit of the beginning stage. Using this adder it is possible to calculate the intermediate results. This adder is the fastest adder used for computation.

**Conclusion:**

A carry look ahead adder is designed and implemented successfully using LTSpice.

Number of transistors:372

Width/Length: 180n/1u

Average power: 29.129nW