ISSN: 2454-9940

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN:2454-9940 www.ijsem.org

Vol 20, Issuse.2 April 2026

A Design of Low Power and High-Speed Multiplier Circuits by Re-Evaluating High Speed Design Values

[1] V. Nagalakshmi,[2] Pentyala Ashwitha,[3] Thalari Anjaiah,[4] A.V. Vamshi Krishna,[5] Sanda Shiva Shankar,

[1]Assistant Professor, Department of ECE, Narsimha Reddy Engineering Collage, Maisammaguda(V), Kompally, Telangana.

[2,3,4,5] Student, Department of ECE, Narsimha Reddy Engineering Collage, Maisammaguda(V), Kompally, Telangana.

ABSTRACT

Research is now the primary focus because to the fast advancements in CMOS technology that have decreased power consumption, increased chip speed, and decreased space requirements. The most difficult part of CMOS VLSI design, power dissipation and latency, are addressed in this article by using Transmission Gate Logic (TGL). The performance is maintained while the power consumption is decreased. An important consideration in the design of such devices is the balance between power consumption and speed. Using the suggested TGL, the multipliers in this work are aimed to remove trade-off difficulties to a larger extent. The AND Gate reduces the number of transistors by four. Despite adding two extra transistors to the suggested design, the power consumption of the OR gate is cut in half. With CADENCE VIRTUOSO, the circuits are constructed using 45nm technology, and the performance characteristics are assessed.

Transmission gate, multiplier, AND gate, OR gate are all important terms.

I. INTRODUCTION

Thanks to advancements in very large scale integration (VLSI) technology, we can now manufacture devices with millions of transistors. Accordingly, there is a strong need for low power VLSI technologies in the present day. Circuit, architectural, layout, and process technology are the four tiers of design for very large scale integration technologies. The circuit level design has become much more power efficient by choosing an appropriate logic style for designing logic circuits [1]. Because it uses less power than bipolar junction transistors, complementary metal-oxide semiconductors (CMOS) integrated circuits are now in use. Compared to CMOS logic, the circuit's power consumption will be lower when a transmission gate is used.Translation Gate Logic may therefore be used to build logic circuits, as an alternative to the conventional CMOS pull-up and pull-down networks [2]. Using cadence software, this research will investigate the 45nm technology's encoder and decoder performance in terms of speed and power in transmission gate logic. Section A: The Transporter The PMOS and NMOS transistors work in tandem to form the Transmission Gate. Power MOS and NMOs transistors may both operate in tandem. A low-resistance channel between the input and output terminals is provided by TG over the whole input voltage range [3]. The gate input voltage determines which input signals are sent on to the output by the TG. A three-or four-terminal NMOS transistor is available. The three terminals of a three-terminal NMOS transistor are the gate, drain, and source, while the fourth terminal is the substrate. A p-substrate, an n-type drain, and an n-type source comprise an NMOS transistor. The NMOS transistor uses electrons as a carrier. In order to turn an NMOS on, you need to add a high voltage to its gate input; conversely, turning it off requires a low voltage. In comparison to PMOS transistors, NMOS transistors are quicker due to the fact that electrons travel at double the speed of holes. Smaller than PMOS ICs, the NMOS IC is what you may expect. Compared to a PMOS transistor, it only supplies half the impedance. One way that NMOS transistors work is by creating an inversion layer on top of a p-type transistor. To establish an n-channel, a voltage is applied to the gate terminal. There are three or four terminals on a PMOS transistor. Whereas in a four-

terminal PMOS transistor the substrate is an extra terminal, in a three-terminal PMOS transistor the terminals are source, drain, and gate. The PMOS transistor has an n-substrate, p-type drain, and p-type source. Carriers in a PMOS transistor are holes. In a PMOS transistor, turning it on requires a low voltage applied as the gate input, and turning it off requires a high voltage applied as the gate input. Compared to NMOS transistors, PMOS transistors have better noise immunity. An inversion layer is created in the body of an n-type transistor for the PMOS transistor to function. To form a pchannel, a voltage is applied to the gate terminal. The input, output, and controlling gate terminals of a transmission gate are shown in Fig. 1.

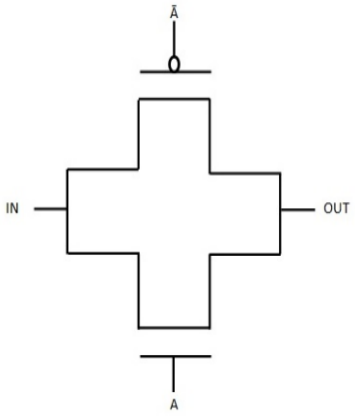


Fig 1. Symbol of Transmission gate.

Multipliers

Digital Signal Processing (DSP), embedded control systems, cryptography, high performance computing, image and video processing, machine learning accelerators, and many more digital systems rely on multipliers as an essential mathematical component. Designing high-speed, low-power multipliers has become a key issue in current VLSI design because to the increased throughput and energy efficiency demands of these applications. Multipliers multiply two binary values by iteratively adding and bitwise shifting them. The idea is straightforward, but as the operand size increases, the complexity of the hardware implementation climbs. Optimizing the multiplier is crucial for system performance as a whole since its delay, power, and area grow non-linearly with bit width. 1. Multipliers In VLSI Systems And Their Significance Performance essential datapaths often include multipliers. Multiplication may happen millions of times per second in today's CPUs and digital signal processors, and any efficiency boost boosts system performance and power consumption. Important factors to consider while improving multipliers are: • Lots of internal switching: Multipliers have a lot of switching, which makes them very dynamically powerful. Conventional CMOS multipliers need a high number of gates for the creation and addition of partial products, which in turn increases the space required for the device. • Dominance of critical paths: The multiplier is often the node in a processor's datapath that has the greatest latency. By executing the same code over and over again, even little improvements in latency or power consumption add up to a big difference. 2. Binary Multipliers' Basic Structure The three main components of a binary multiplier are: 1. Generating Parts of the Product The bits of the multiplier are ANDed with the bits of the multiplicand. By doing so, a matrix of partial products is formed. 2. Minimizing Some Product Features Adders, compressors, or parallel reduction trees are used to lower the matrix in order to lessen the propagation latency. 3. The Last Subtraction The final output is the result of applying a high-speed adder to the last two rows of the reduced matrix. Standard layouts include four-phase compressors, full adders, half adders, and complementary metal-oxide semiconductor (CMOS) AND gates. Nevertheless, when used in sub micron technologies, they experience increased leakage and capacitance, which results in excessive power consumption and delay. 3. Differences Between Hardware Multipliers Different multiplier topologies aim to maximize speed, power, or regularity, among other criteria. a) Multiplier to Array • Applicable to very large scale integration (VLSI) • Area is high since there are a lot of adders. • Performance is moderate; the delay grows in a linear fashion with the breadth of the operand. b) A tree multiplier called Wallace • Utilizes compressors' parallel reduction tree quick because of the depth of logarithmic reduction Complicating routing is the irregular structure. b) The Dadda Trifecta • Wallace tree with optimizations • Lessens the need for adders • Reduces hardware utilization by only reducing when necessary c) Multiplier for Booth Using encoding, the number of incomplete products is reduced. Signed multiplication is handled efficiently. • Relatively slow, superior space conservation Partially generating and reducing products are the same basic actions across all architectures, while each has its own set of advantages and disadvantages.

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN:2454-9940 www.ijsem.org

Vol 20, Issuse.2 April 2026

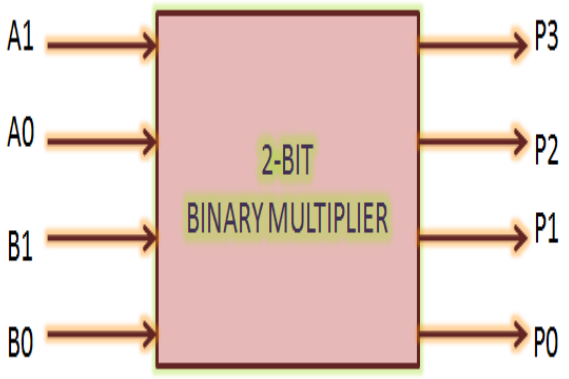


Fig 2. Multiplier

EXISTING SYSTEM

One of the most fundamental building elements of digital very large scale integration (VLSI) systems is the multiplier, and there are several designs to choose from depending on factors including power consumption, silicon area, and processing speed. The current setup generates partial products, reduces them, and then adds them using traditional CMOS-based multiplier topologies. High dynamic power dissipation, higher latency from lengthy critical pathways, and greater transistor counts from reliance on sophisticated CMOS gate layouts are some of the problems with these conventional designs, despite their widespread usage. 1. Initial Product Generation using Conventional CMOS Techniques Common CMOS AND gates, which use 6-8 transistors per gate, are used to create partial products in a traditional binary multiplier. A total of N^2 AND operations are needed for a N×N multiplier. Therefore, as the operand size grows, so does the number of transistors. As a consequence, The parasitic capacitance is enhanced. A greater degree of dynamic switching a bigger surface area of silicon • Decreased efficiency The pull-up and pull-down networks used by each CMOS AND gate are complimentary. Area grows with many pMOS pull-up transistors, while propagation delay grows with series stacking of nMOS devices. 2. Standard Reduction Trees and Operators Full adders, half adders, and compressor circuits are used to conduct summing after partial product creation. Within the current framework: • Full adders in standard CMOS technology need 28 to 32 transistors. 14–16 transistors are needed for half adders. • These adders are used in compressors on numerous levels (4-2, 5-2). As a result, both power consumption and the total critical path latency are increased. Multiplier forArrays: A regular grid of complete adders is used. • Efficient but sluggish design due to linear growth of the critical path with operand size Extremely high power consumption because to the array's high switching activity Multiplier for Wallace Trees: quick compared to array multipliers • Partial products undergo simultaneous grouping and reduction. • Complicated routing and increased connection capacitance result from an irregular structure. Multiplier by Dadda: Compared to Wallace tree, hardware utilization is somewhat lower. On the other hand, CMOS logic is still widely used, which results in high power consumption. 3. CMOS Multipliers' Static and Dynamic Power Problems Understanding complementary metal-oxide-semiconductor (CMOS) circuits • Subthreshold leaking causes static power Dynamic power is generated when internal nodes are charged or discharged. • More internal nodes in large transistor networks The delay is increased due to higher output loading. This power loss adds up quickly since multipliers are used so often in CPU and DSP pipelines. Problems with the Current System Summarized • A high number of transistors • Propagation latency is increased. • Excessive static and dynamic power • Enhanced space for design • Advanced routing in multipliers of higher order Because of these restrictions, a space-saving, time-saving, and power-saving implementation method like Transmission Gate Logic (TGL) is required.

II. PROPOSED SYSTEM

The suggested method employs Transmission Gate Logic (TGL) to rethink the whole multiplier architecture, including partial product creation, reduction tree, and final addition. Power consumption, space, and operation speed are all improved because to the TG-based implementation, which takes the role of traditional CMOS gates. First, the Production of Partial Products Based on TG Instead of an 8-transistor CMOS implementation, the AND operations needed for partial product production are carried out in TG logic utilizing a 4-transistor TG-AND gate. Advantages: • Decrease in transistor count by 50% The parasitic capacitance is reduced. • Quicker production of incomplete goods Because there is no direct channel leakage, static power is reduced. Because both NMOS and PMOS devices are active during switching, the TG-AND gate offers full-swing output with low latency. 2. Compressors and Adders Based on TG The multiplier's adding phases have been rethought

by incorporating: Full TG adders • Half-adders TG TG-XOR/XNOR gates • Compressors based on TG (4-2) and 5-2 These blocks fare far better than their CMOS counterparts. All-Inclusive TG Adder: • Less transistors are needed (10–14), as opposed to 28–32 Dynamic switching is minimized. Few pathways for leaking • Less internal nodes means faster operation TG Compressors: Wallace/Dadda multipliers rely on compressors. Compressors based on TG: Logic levels should be minimized. The propagation latency is reduced. • Supply outputs that are both symmetric and full-swing • Less transistors are needed compared to CMOS versions In high-speed multiplier designs, this improves the efficiency of reduction trees. 3. Establishing a Multiplier Architecture Based on TG The following designs are compatible with the proposed multiplier: a) TG-Based Array Multiplier Maintaining the original layout • Significantly decreased power usage • Quicker as a result of using TG-FA b) A Wallace Tree Multiplier Based on TG • Makes use of turbochargers Obtains a decrease in logarithmic delay • Well-suited for complex digital signal processing tasks b) A Dadda Multiplier Based on TG reduces the need for hardware Reduced transistor count is achieved by TG-based methods. 4. Using TG to Reduce Power The use of transmission gate logic lowers dynamic and static power consumption: Because PMOS and NMOS work in tandem, TG has almost little power loss even when turned off. o Fewer switching nodes lead to dynamic power reduction. - Reduced capacitances - Preservation of signal quality Compared to PTL and CMOS, TG offers better power-delay performance because it integrates the advantages of pass transistor logic with full-swing operation. 5. Improving the Proposed TG Multiplier with Minimal Delay There is less wait time because of: • Decreased strain on essential routes Logic levels that cascade are reduced. When turned on, the conductive route has low resistance. • Characteristics of symmetrical switching A fast multiplier, well-suited to high-speed applications such as communication processors and digital signal processing pipelines, is the end product.

Total Transistor Count Comparison

Component	CMOS Transistors	TG Transistors	Reduction
AND gate	6–8	4	33–50%
Full Adder	28–32	10–14	50–65%
XOR Gate	6–10	4–6	40–60%
4-2 Compressor	30–40	16–20	40–55%

For an 8×8 multiplier, the reduction is significant across all stages.

Benefits of the Proposed TG Multiplier

- **Lower power dissipation** (static + dynamic)
- **Higher switching speed**
- **Lower transistor count**
- **Reduced area**
- **Improved power–delay product (PDP)**
- Full-swing outputs ensure compatibility with CMOS stages
- More robust performance in scaled technologies (45 nm and below)

Literature Survey

Digital signal processors, hardware for processing images and videos, accelerators for machine learning, embedded controllers, and communication subsystems all rely on multipliers as a fundamental computing component. Design criteria like as power consumption, latency, leakage currents, and silicon area are becoming more important as semiconductor technology advances into deep-submicron dimensions. Power consumption and critical path latency in very large scale integration (VLSI) datapaths are often dominated by multipliers due to their high switching activity and huge number of arithmetic operations. This literature review delves into different multiplier designs, optimization methods, logic styles, and new developments, with a focus on Transmission Gate Logic (TGL)—a new alternative to conventional CMOS logic—and its recent achievements. The objective is to showcase TGL's capabilities in developing high-speed multiplier architectures that are both power-efficient

and appropriate for technologies that use 45 nm or less. 2. Typical CMOS Multiplier Architectures In order to generate partial products, use adders or compressors to reduce them, and then produce the product using fast adders, traditional multipliers depend substantially on CMOS static logic. There are a number of drawbacks to using CMOS, despite its benefits (such as resilience to noise and durability). 2.1.1 Static CMOS's Restrictions on Multiplier Architecture Scientists like Weste and Harris point out CMOS logic's shortcomings in fast-processing mathematical circuits: • Full adder, AND, XOR, and OR circuits need a large number of transistors. The dynamic power consumption is enhanced due to large internal capacitances. • Significantly increasing leakage currents in nanoscale technologies • Slower switching due to more stacking transistors • Large space and connective tissue Because switching activity is dense over several bit locations, these constraints are made more apparent by the dynamic behavior of multipliers. 3. Literature on Multiplier Architectures Speed, area, and power efficiency have been the primary concerns of the many multiplier designs that have been advanced over the years. 3.1 Multipliers for Arrays The array multiplier, an early design, is simple and very regular. Nevertheless, it has issues with: • Extending essential routes • The regular grid of adders results in a large area. • Heavy use of electricity because of several switches Although array multipliers are simple to design, researchers have found that they struggle to handle high-speed tasks. Triplets of the Wallace Tree 3.2 Hierarchical reduction was first introduced by the Wallace tree multiplier: The 4-2 and 5-2 compressors are used in parallel. • Significantly lowers the partial product matrix's height • Makes a huge difference with delay On the other hand, wire capacitance rises and routing complexity is introduced by the Wallace tree's uneven topology. 3.3 Multipliers for Dadda By postponing compression until absolutely required, the Dadda multiplier improves upon the Wallace tree and thereby decreases hardware consumption: • A reduced amount of adders quick compared to array multipliers • Not as fast as Wallace, but much smaller According to research, Dadda architecture successfully combines speed with area. 3.4 Multipliers for Booths The amount of incomplete products is decreased via Booth encoding: - Works well with signed multiplication • Decreases the number of incomplete products, which in turn decreases switching activity • Apt for high-performance digital signal processor blocks Nevertheless, more logic and latency are introduced by Booth recoding. 4. Methods for Optimizing Multipliers at the Logic Level Many researchers have looked at low-power logic approaches, in addition to architectural adjustments. 4.1 PTL, or Pass Transistor Logic PTL decreases capacitance and the number of transistors, yet it has drawbacks such as: • Voltage drop at the threshold Limitation on background noise Decreasing signals in a series of related processes For greater multipliers, PTL is not a good fit due to these restrictions. 4.2 Logic for Complementary Pass Transistors (CPL) CPL offers full swing outputs, however, it is Static power is high. • The amount of transistors has increased The connection capacitance is large. Research shows that deep-submicron multipliers can't handle CPL's power requirements. Chapter 4.3: The GDI Gate The use of GDI lowers power consumption and the number of transistors, however: • Needs unique production procedures Not suitable for use with conventional CMOS foundries Because of this, it has not been widely used in industry. 5. A More Efficient Option: ransmission Gate Logic (TGL) One of the main reasons why TGL is starting to take the place of CMOS logic is because it can: • Give complete swing performance 2. Decrease both static and dynamic power • Keep internal node capacitances to a minimum • Significantly decrease the number of transistors • Keep noise margins strong 5.1% TG-Based AND, OR, and XOR Gates Evidence from several research shows that TG implementations of fundamental gates: • Cut down on transistor use by 40–60% • Provide CMOS-like switching speeds at a higher rate Lessen power loss and fluctuating power consumption Issues such as Vth drop in pass transistor logic should be eliminated. The AND and XOR gates are the backbone of multiplier circuits, which use these enhancements to generate and add partial products. Full adders and compressors based on TG (section 5.2) The use of adders and compressors is crucial to the reduction tree in multipliers (Wallace/Dadda). The study's authors state: • CMOS complete adders need 28–32 transistors, whereas TG-based ones only need 10–14. • Compressors based on TG lower switching levels TG-XOR/XNOR circuits expedite the summing process. • Multipliers for TG-Wallace trees with reduced critical path delays 5.3 Technology Gain Accelerators for Large-Scale Systems For technologies using 45 nm and smaller, TG logic provides: Reducing the number of stacked transistors results in lower gate leakage. DIBL (drain-induced barrier lowering) effects are minimized.

Compared to PTL and CPL, it is more resistant to process changes. • Less gadgets mean more efficient arrangement Modern VLSI multipliers are well-suited to TGL because of this. 6. Exploring Current Techniques for Low-Power Multiplier Design 6.1 Partial Product Generators That Use Less Energy Several studies have looked at ways to make partial product creation use less energy: enhanced array of AND gates • TG-logic-enabled booth recoders • PTL-GDI combined ignition systems Though, when it comes to speed-power trade-off, TGL is till superior. Architectures for High-Speed Compressors (6.2) A lot of effort goes into fast compressors in modern multiplier designs, particularly Wallace tree variations. New research suggests: four or two TG compressors 2-and-a-half TG compressors • CMOS-TG hybrid compressors for optimal speed These streamline the adding process and enhance timing efficiency. 6.3 Designs for Low-Power Multipliers in DSP and AI 6.3 The use of multipliers is on the rise in: Engines that use convolution • Multiple-accumulate (MAC) units • Processors for neural networks The FFT and IFFT blocks Designing low-power multipliers is a straightforward way to make ML and DSP accelerators that are energy efficient. 7. Areas Where Research Is Deficient A number of gaps persist despite much research on optimal multipliers: 1. Few studies have investigated hybrid dadda and booth-wallace algorithms, which combine TG logic with sophisticated multiplier architectures. 2. For high operand sizes, there is a lack of progress on comprehensive TG-based reduction trees. 3. There are currently no design automation tools available for TGL arithmetic circuits. 4. Emerging technologies (FinFET, FDSOI) do not adequately investigate TG-based multipliers. 5. Very little research on how TG multipliers react to noise and temperature. Filling these deficiencies, this project develops a comprehensive TG-based multiplier that includes: • Partially constructed TG-AND products TG complete adders compressors used by TG Summation stages that are optimized for TG 8. Final Thoughts and Remarks Despite their decades-long dominance, conventional CMOS multiplier designs are becoming more and more unfit for low-power, high-speed applications in contemporary nanoscale technology, as shown by this literature study. There have been many attempts at optimization, however the majority of these methods suffer from either incompatibilities with manufacturing processes or power consumption. A strong substitute, Transmission Gate Logic, arises as a result of: • Decrease attenuation • Decreased switching frequency • Minimal static electricity • Behaviour including a full swing • TGL's high-speed functioning is a great step toward creating efficient multipliers for use in VLSI systems of the future.

INTODUCTION TO VLSI TECHNOLOGY

Together, advancements in integrated circuit (IC) technology have enabled a plethora of novel devices and systems that have fundamentally altered our daily lives. Neither semiconductors nor personal computers would have achieved their current levels of significance without the integrated circuit. Winners of the Nobel Prize in Physics in 2000 were Jack Killby and Robert Nonce, for their work on the coordinated circuit. Very large scale integration (VLSI) systems save space and energy compared to the separate components utilized to construct electronic frameworks prior to the 1960s. Thanks to resolution, we can include a lot more transistors into our systems, which means we can apply a lot more processing power to a problem. However, discrete reason frameworks are more efficient than PCs for the principal activity, and coordination circuits are simpler to design and build than discrete frameworks. This allows for the creation of separate reason frameworks.

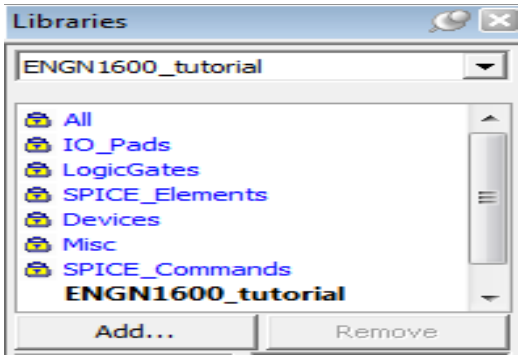
INTRODUCTION TO SOFTWARE

In order to record, go to the following menu: Begin > Electrical > EDA tanner > Tanner Tools v7.2 > S-Edit v7.2 64-bit. This is a list of the tools and hardware requirements for 6.2. Inverter with symbols and setup. Choose "Fresh," then "Fresh file," and finally "New style" to start constructing a fresh idea.Make sure to include both the folder and the name of the template you want to create. According to their designation, the "ENGN1600 tutorial" will be located on the left side of the Library navigator.Automated library installation is not possible on shared PCs. Get the Tanner Library folder and import it into your program. To access all of the tanner app libraries, either use the "Connect..." key on the S-EDIT Library page or click to App > Open > Connect Library. From there, you may search the archive folder. When you open the file in Libraries, you'll see a bunch of sub-libraries, which are IO pads and devices.Upon opening the file, you will be able to see the following libraries in the Libraries tab:

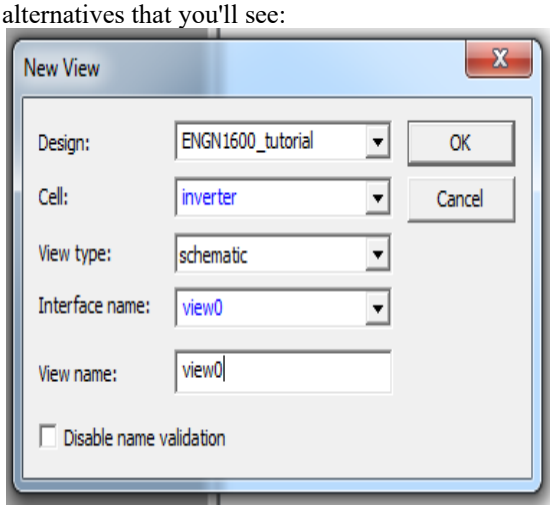
INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN:2454-9940 www.ijsem.org

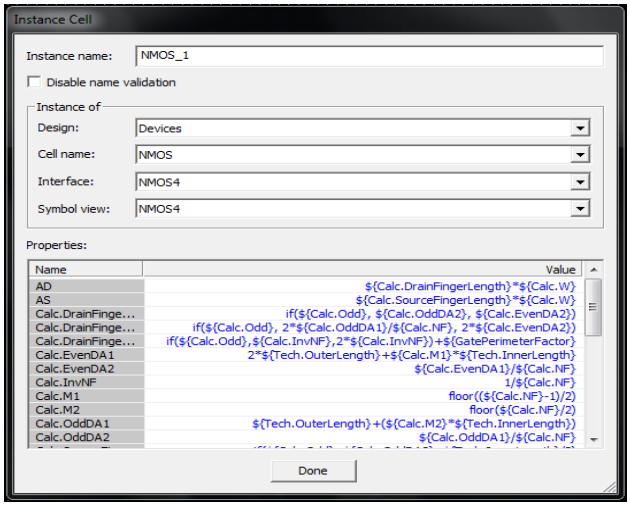
Vol 20, Issuse.2 April 2026



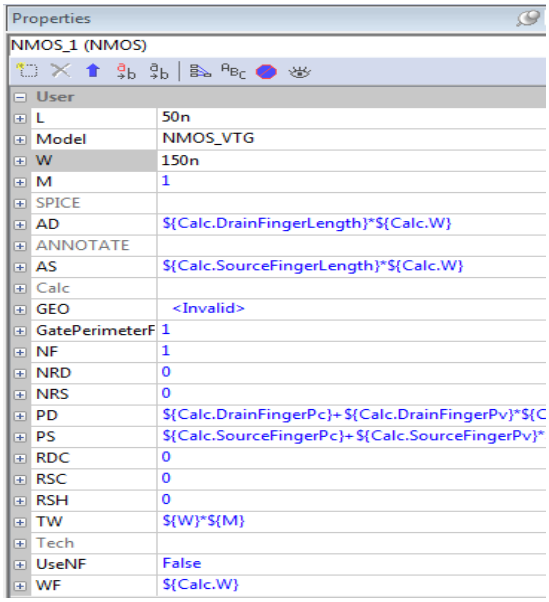
Structures for construction Launch a new layout cell by selecting Cell > New View. Here are the alternatives that you'll see:



Click OK once you've entered the name of the cell you want to construct. The door will soon open to a new cell perspective.A CMOS inverter provides In / Out, Vdd, Gnd, NMOS, and PMOS connections. The library software is located on the left side of the Libraries tab. When you click on it, the symbol artifacts will be shown in the browser below. Looking at the graph, think about the NMOS. Simply drag and drop your computer into the display area, or choose it and then select the "Instance" button. Do the same with the PMOS, Vdd, and Gnd symbols. A popup similar to the one on the left side of the screen may display before you release the case.



After the instant period is dropped, a second window, similar to the one on the right, will show on the right side of the device. Here, we'd want to apply the property, which includes the diameter and depth. Once the case is in place, we usually click on it to edit its properties in the Navigator Properties panel on the right. We will pretend that the length and width of NMOS devices are 50n and 150n, and that the duration and breadth of PMOS devices are 50n and 300n, respectively. Please be informed that in order to do SPICE experiments on our libraries, it is necessary to change the settings from NMOS to NMOS VTG.



To make the model look the way you want it to, you'll have to reorganize the pieces. When you click on a certain part of the design, you'll see the THREE-

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

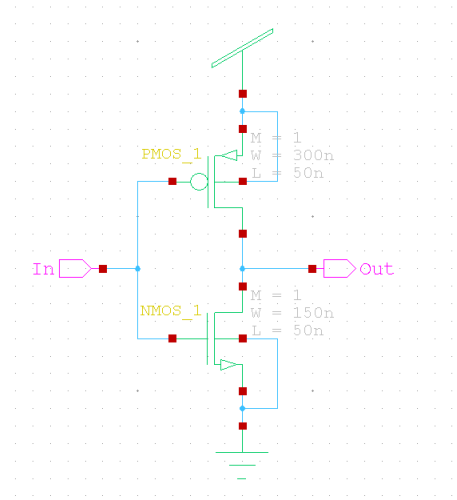
ISSN:2454-9940 www.ijsem.org

Vol 20, Issuse.2 April 2026

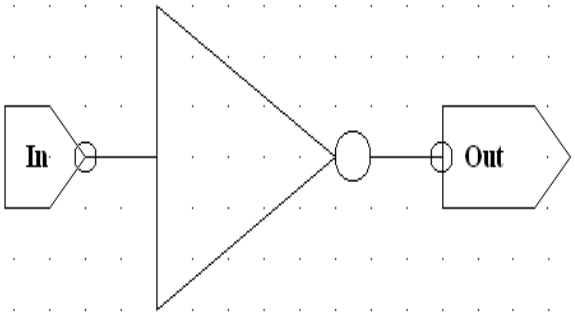
BUTTON Cursor icon anywhere in the user interface, looking like this:

•At this point, we should go to the visual part of the process to choose an element from the LEFT or RIGHT mouse trap, and then use the MIDDLE capture to move it. Alternatively, you might press the part, hold down the ALT key, and then push it to a designated spot on the screen. Zoom in or out using the View screen, the Scroll wheel cursor, or the plus sign (+) to zoom in and minus sign (-) to zoom out. It is now time to include I/O ports and connect the components, assuming we have already set up the basic ones. Just give each port a unique name and choose the knowledge port from the list of yield ports on the diagram.

The next step is to make the proper terminal connections using the wiring tool. The cable starts with a SINGLE MOUSE CLICK when we push on the device. Simply press the DOUBLE CLICK button twice to end the cable at a different location.At both the input and output ports of your completed inverter, you should see something similar.



The connections were generated in that way. Modifying the width and duration of the two MOS transistors to meet the requirements is the next step. 6.2.1 Inverter Symbol - To produce the symbol, develop a perspective. Either create a new cell with a unique aspect on the symbol or use Cell > existing perspective to start again with the existing cell. Give the icon the title and choose the device you want to link it to. Next, apply the w ports. It may be helpful when creating a hierarchical structure if the ports are named the same as the ones you're using in the creating the symbol.



SPICE NETLIST WILL BE CREATED FROM THE SCHEMATIC

- Navigate to Import > Tab > Export Spice.Give the name of the reference file. The produced SPICE file might be included in several additional SPICE files using the.end option in the Test Exclude.endin Options.Send the message.A tab for schematics stores the SPICE file. Information like this is included in the file: - MNMOS 1 Out NMOS VTG with 150n and 50n layers, AS=135f, PS=2.1u, AD=135f, and PD=2.1u in GndGnd. - MPMOS 1 Out PMOS with 300n and 50n layers, AS=270f, PS=2.4u, AD=270f, and PD=2.4u in Vdd to Vddd. The spice layout is often incomplete. It is merely an enumeration of the proper roof that is solely made by us. Several components, including MOS, may be added to the simulation in order to create a model of the circuit by feeding it feedback signals.Here is the condensed version of the spice that we use:The Smrtspice circuit may be repeated.After you've created a spice portfolio using L-Edit or S-Edit, the next step is to figure out how to conclude the simulation.Before you go on to using the 45 nm model with MOS prototypes, be sure you save the simulation when it finishes. Return to the folder's opening screen by entering the supplied password.S-Edit stores the following information in an alternator-style SPICE Net database.

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN:2454-9940 www.ijsem.org

Vol 20, Issuse.2 April 2026

```
* SPICE export by:  S-Edit 15.22
* Export time:      Wed Jan 30 16:39:09 2013
* Design:           ENGN1600_tutorial
* Cell:             inverter
* Interface:        view0
* View:             view0
* View type:        connectivity
* Export as:        top-level cell
* Export mode:      flat
* Exclude empty cells: no
* Exclude model:    yes
* Exclude end:      yes
* Exclude simulator commands:    yes
* Expand paths:     yes
* Wrap lines:       no
* Root path:        C:\Users\Marco\Desktop\EN1600_test\ENGN1600_tutorial
* Exclude global pins:
* Exclude instance locations: yes
* Control property name: SPICE

MNMOS_1 Out In GndGnd NMOS_VTG W=150n L=50n AS=135f PS=2.1u AD=135f PD=2.1u
MPMOS_1 Out In VddVdd PMOS_VTG W=300n L=50n AS=270f PS=2.4u AD=270f PD=2.4u
```

The result was a SPICE file that included the basic schematic of the circuit. See the spice sideboard for information on how to use the model library to include the specified parts, power supply, various simulation variables, research design, and stimulus input. Everything we would want the alternator to look at is in this file.

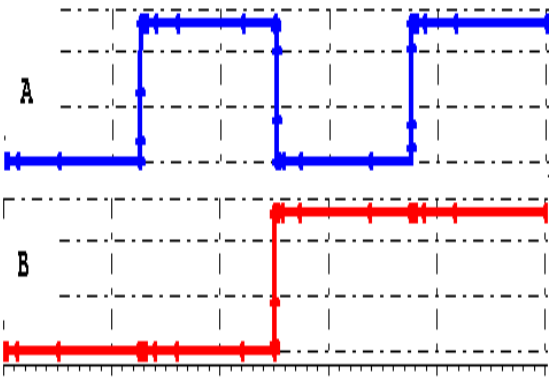
```
*Main                          inverter                        file
.param                                                         Supply=1.1
.include                                                       '../models/hspice_nom.include'
.options                                                       post

*                    Define                power                supply
.global                                                        VddGnd
VddVddGnd                                                      'Supply'

*Set                          Transient                        Analysis
.tran                          100ps                           500ns

***                  CIRCUIT                 NETLIST                  ***
*                    .include                <name_of_extracted_spice_file>
.include                                                       'inverter.sp'

***                  INPUT                   STIMULS                  ***
*          add             any          input          vector          here

vin1   In   Gnd   pulse(0   Supply   10n   1n   1n   50n   100n)
* if you want to see the VTC for the inverter add a DC analysis
.DC                vin1              0              Supply              0.1
.END
```

The blue line will remain unchanged when we change styles. The short-term research needs tweaking. The date has to be changed. as a result of customizing the system for research purposes. The invertsp, which changes the spice-dependent file according to the Tanner approach, may be seen by referring to the first red line. The third line was used to play the labels. The green lines show how to utilize a DC examination to generate a VTC for the inverter, and they are also used to construct the dc analysis that builds and modifies the T voltage transfer functionality. Please submit the Green Line Statement in the event that we do not conduct DC research.And we'll begin with SPICE. Page 9 of this list starts with a more thorough description of a SPICE DECK.Go back to the associated database and find the pattern index; then, duplicate the spice deck.sp list in the design registry. Incorporating the many promoters into the declaration that may display all the instructions is crucial.We will make the whole spike using dazzling log. The resourceful filespice.The source system is the first step in documenting data. Move the online window deck After making changes to the documents, choose the source capture and then save the folder by selecting the capture and providing the correct data sources. SmartSpice might display paper inputs. You should be able to see any mistakes.To conduct an inquiry, choose the analysis option and then run the alternative data box. You can also use the shortcut Ctrl-R to keep running it.Check out the tale details that seem to be bwellneeds. Choose the vectors for capture. We go on to another explanation since there are several layouts to choose from and columns may be viewed in and of themselves. Press the plan to reveal tran1. Additional steps are required to get the required results and data sources. At that moment, we bolded the sentences to indicate how well it worked and how confidently we knew it was down to the control key we picked.Reload it when the reproduction is complete to discover various We denote types of vectors that may be seen as tran1, tran2, tran3, etc., regardless of whether the input text for Spice is modified after the algorithm is run, loaded, and run again. Be careful to disable your previous plots before selecting a new one; else, new, informative ones will appear.It is important to observe all the inputs and outputs of the reaction in this case.if you want to make a two-entry switch that looks like the real thing, you have to make sure the entrance fits in each of the four potential knowledge combinations (00, 01, 10, and 11). Assume that An and B are the names of the two categories of knowledge.This is due to the fact that the two heartbeats will repeat all four mixes in a single run: 0 Supply 10n 1n 50n 100n AGndpulse vin1, and 0 Supply 10n 1n 100n 200n = vin2 B Gndpulse.

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

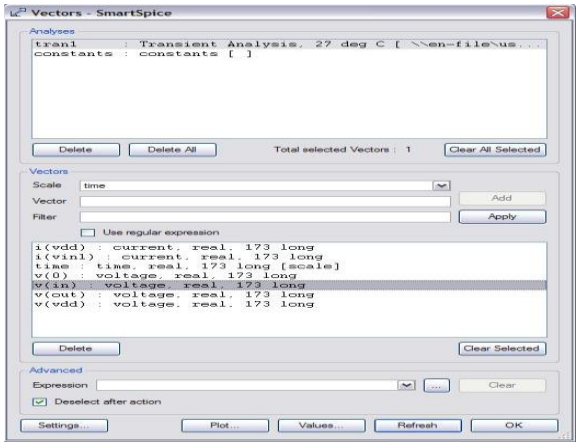
ISSN:2454-9940 www.ijsem.org

Vol 20, Issuse.2 April 2026

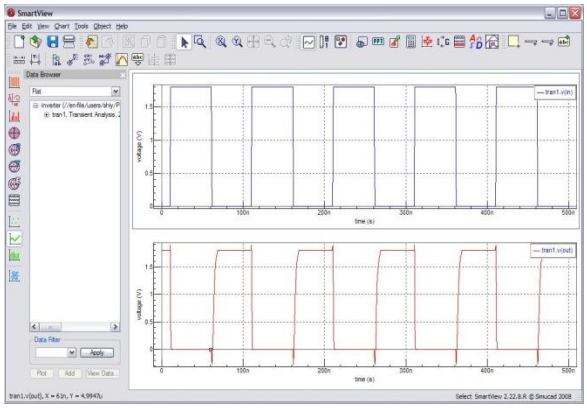


SmartSpice Plots

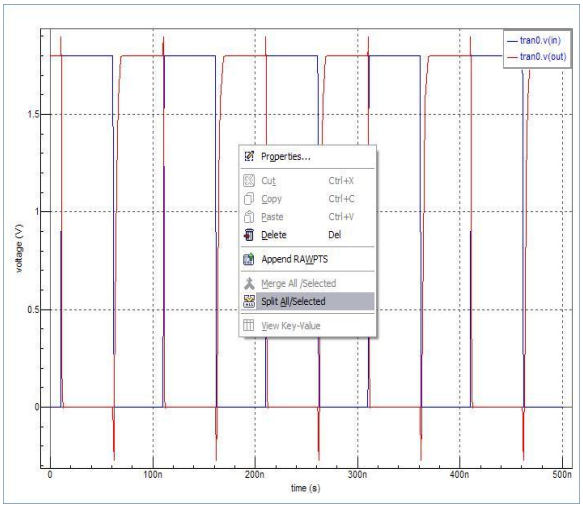
After the complete analysis, click on  from the toolbar to launch the vectors window.

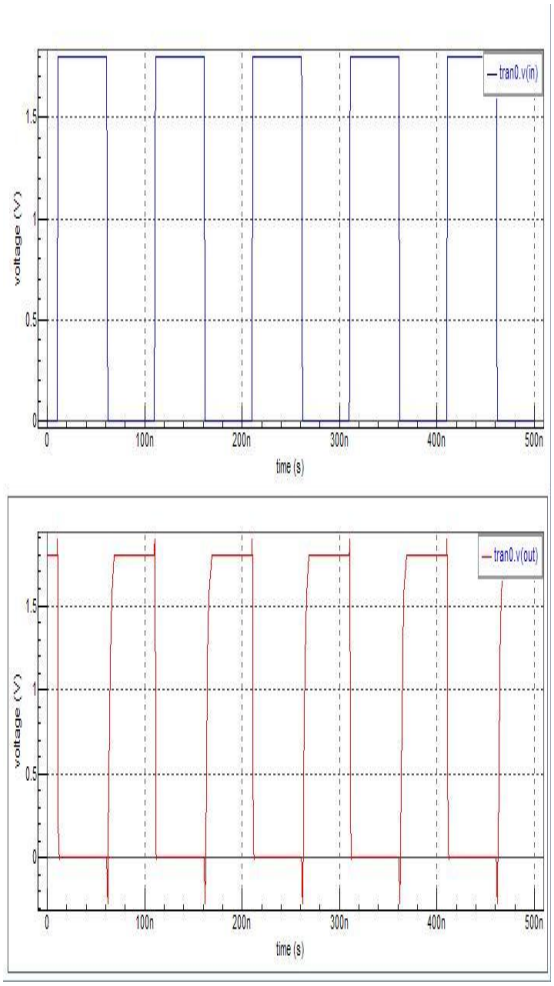


To print a certain voltage, tap on its name. Choose Plot. When it's plotted in SmartView, you may start a new plot area by tapping on the left. Press the second name of the voltage in SmartSpice again. Choose Plot. Continue this process until we have the data and can visualize the yield. This will bring about every one of the sources of info and yield voltages on an indistinguishable plot from subplots.



Simple for you to imagine and model. By right-clicking the plot and selecting "Split All/Selected," you may divide a single plot containing two signals into two separate ones. An stacked view will be shown thereafter.





The default foundation color is going to be dark. Changing it to a white base with dark tomahawks is as simple as go to Edit > Preferences > Colors.

Conclusion

Using a transmission gate and a supply voltage of Vdd (1.8V), this research investigates multiplier circuits. Design and simulation of circuits are carried out using 45 nm technology. In comparison to the CMOS transistor, the bidirectional element (the transmission gate) used here is thought to be the element that switches more quickly. Power and latency are both decreased in the suggested design, according to the simulation and analysis findings.

REFERENCES

[1] C. N. Marimuthu and P. Thangaraj, "Transmission Gate based High Performance Low Power Multiplier," *Journal of Applied Sciences*, vol. 10, no. 23, pp. 3051–3059, Dec. 2010. docsdrive.com+1

[2] Seera Reshma and Velaga Suryakala, "Design of Area Efficient, Low Power, High Speed 8-Bit Array Multipliers by Using Transmission Gate Logic," *International Journal of Science, Engineering and Technology*, vol. 12, no. 5, 2024. ijset.in

[3] Sombathini Rachana and Nandyala Ramanjulu, "Design of Area Efficient and Low Power 4-Bit Multiplier Based on TG," *International Journal of Advance Scientific Research and Engineering Trends*, vol. 5, no. 11, pp. 17–22, Nov. 2020. ijasret.com

[4] Flavio Carbognani, Felix Buergin, Norbert Felber, Hubert Kaeslin and Wolfgang Fichtner, "A Low-Power Transmission-Gate-Based 16-Bit Multiplier for Digital Hearing Aids," *Analog Integrated Circuits and Signal Processing*, vol. 56, pp. 5–12, Aug. 2008. SpringerLink

[5] P. Deepika and N. Shylashree, "Design of an Array Multiplier for Computation in Memory Architecture," *Engineering, Technology & Applied Science Research*, vol. 15, no. 5, pp. 27285–27292, 2025. etasr.com

[6] Sahil Garg, "Low Power and Area Efficient Multiplier Layout Using Transmission Gate," *International Journal of Electrical & Electronics Engineering*, vol. 2, special issue 1, 2015. www.slideshare.net

[7] S. S. Mahant-Shetti, P. T. Balsara and C. Lemonds, "High Performance Low Power Array Multiplier Using Temporal Tiling," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 7, no. 1, pp. 121–124, Mar. 1999. SpringerLink

[8] Kuan-Hung Chen and Y.-S. Chu, "A Low-Power Multiplier With the Spurious Power Suppression Technique," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 7, pp. 846–850, Jul. 2007. computer.org+1

[9] Sumit Vaidya and Deepak Dandekar, "Delay–Power Performance Comparison of Multipliers in VLSI Circuit Design," *International Journal of Computer Networks & Communications (IJCNC)*, vol. 2, no. 4, pp. 47–56, 2010. www.slideshare.net

[10] S. Jagadeesh and S. Swapanthi, "Low Power Multiplier Design for Polyphase Filter with Reduced

Area and High Speed Design," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 3, no. 12, pp. 13860–13871, Dec. 2014. ijareeie.com

[11] Yihe Yu, Wanyuan Pan, Chengcheng Tang, Ningyuan Yin and Zhiyi Yu, "Design of a High-Speed, Low-Power PTL-CMOS Hybrid Multiplier Using Critical-Path Evaluation Model," *Electronics*, vol. 13, no. 7, 1284, Apr. 2024. MDPI

[12] P. Kiran Kumar and P. Srikanth, "Design of Low Power High Speed Hybrid Full Adder," *International Journal of Electronics & Communication Technology*, vol. 6, no. 4, pp. 14–18, Oct.–Dec. 2015. ResearchGate

[13] Alvin Joseph J. Tang and Joy Alinda Reyes, "Comparative Analysis of Low Power Multiplier Architectures," in *Proc. Asia Modelling Symposium (AMS)*, 2011. ijareeie.com

[14] Andrew D. Booth, "A Signed Binary Multiplication Technique," *The Quarterly Journal of Mechanics and Applied Mathematics*, vol. 4, no. 2, pp. 236–240, 1951. ijset.in+1

[15] Christopher S. Wallace, "A Suggestion for a Fast Multiplier," *IEEE Transactions on Electronic Computers*, vol. EC-13, no. 1, pp. 14–17, Feb. 1964. ijset.in+1

[16] Charles R. Baugh and Bruce A. Wooley, "A Two's Complement Parallel Array Multiplication Algorithm," *IEEE Transactions on Computers*, vol. C-22, no. 12, pp. 1045–1047, Dec. 1973. ijset.in

[17] C. N. Marimuthu, P. Thangaraj and Aswathy Ramesan, "Low Power Shift and Add Multiplier Design," *International Journal of Computer Science and Information Technology*, vol. 2, no. 3, pp. 62–69, Jun. 2010. ijareeie.com

[18] M. Mottaghi-Dastjerdi, A. Afzali-Kusha and M. Pedram, "BZ-FAD: A Low-Power Low-Area Multiplier Based on Shift-and-Add Architecture," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 2, pp. 302–306, Feb. 2009.