CHAPTER-1

INTRODUCTION TO VLSI

1.1 INTRODUCTION TO VLSI:

Digital systems are highly complex at their most detailed level. They may consist of millions of elements i.e., transistors or logic gates. For many decades, logic schematics served as thengua franca of logic design, but not anymore. Today, hardware complexity has grown to such a degree that a schematic with logic gates is almost useless as it shows only a web of connectivity and not functionality of design. Since the 1970s, computer engineers, electrical engineers and electronics engineers have moved toward Hardware description language (HDLs).

Digital circuit has rapidly evolved over the last twenty five years .The earliest digital circuits were designed with vacuum tubes and transistors. Integrated circuits were then invented where logic gates were placed on a single chip. The first IC chip was small scale integration (SSI) chips where the gate count is small. When technology became sophisticated, designers were able to place circuits with hundreds of gates on a chip. These chips were called MSI chips with advent of LSI; designers could put thousands of gates on a single chip. At this point, design process is getting complicated and designers felt the need to automate these processes.

With the advent of VLSI technology, designers could design single chip with more than hundred thousand gates. Because of the complexity of these circuits computer aided techniques became critical for verification and for designing these digital circuits.

One way to lead with increasing complexity of electronic systems and the increasing time to market is to design at high levels of abstraction. Traditional paper and pencil and capture and simulate methods have largely given way to the described UN synthesized approach.

For these reasons, hardware description languages have played an important role in describe and synthesis design methodology. They are used for specification, simulation and synthesis of an electronic system. This helps to reduce the complexity in designing and products are made to be available in market quickly.

The components of a digital system can be classified as being specific to an application or as being standard circuits. Standard components are taken from a set that has been

used in other systems. MSI components are standard circuits and their use results in a significant reduction in the total cost as compared to the cost of using SSI Circuits. In contrasts, specific components are particular to the system being implemented and are not commonly found among the standard components.

The implementation of specific circuits with LSI chips can be done by means of IC that can be programmed to provide the required logic.

1.2 EMERGENCE OF HARD WARE DESCRIPTION LANGUAGE

As designs got larger and complex, logic simulation assumed an important role in design process. For a long time, programming languages such as fortran, pascal & c were been used to describe the computer programs that were been used to describe the computer programs that were sequential in nature. Similarly in digital design field, designers felt the need for a standard language to describe digital circuits. Thus HDL is came in to existence. HDLs allowed the designers to model the concurrency of processes found in hardware elements. HDLs such as **VERILOG HDL & VHDL (Very high speed integrated circuit hardware description language**). Became popular.

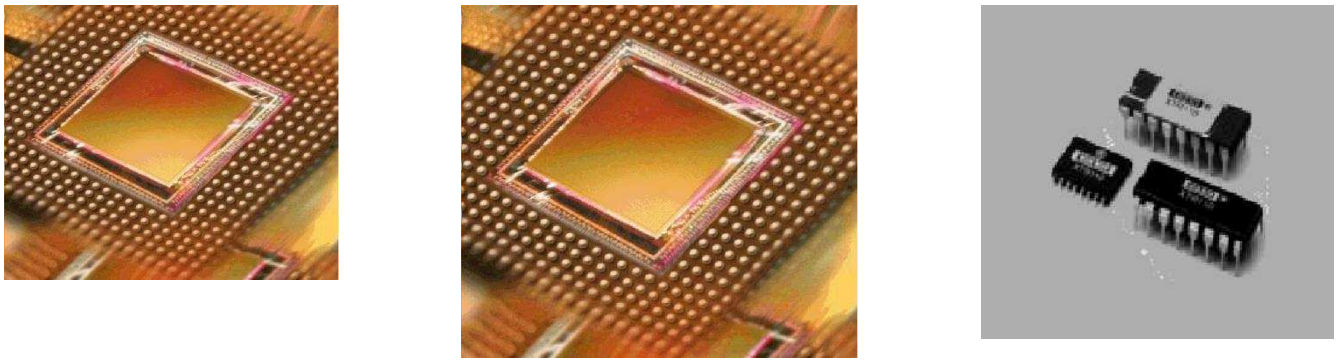


fig 1.1 Integration circuits of VLSI

1.3 DESIGN FLOW

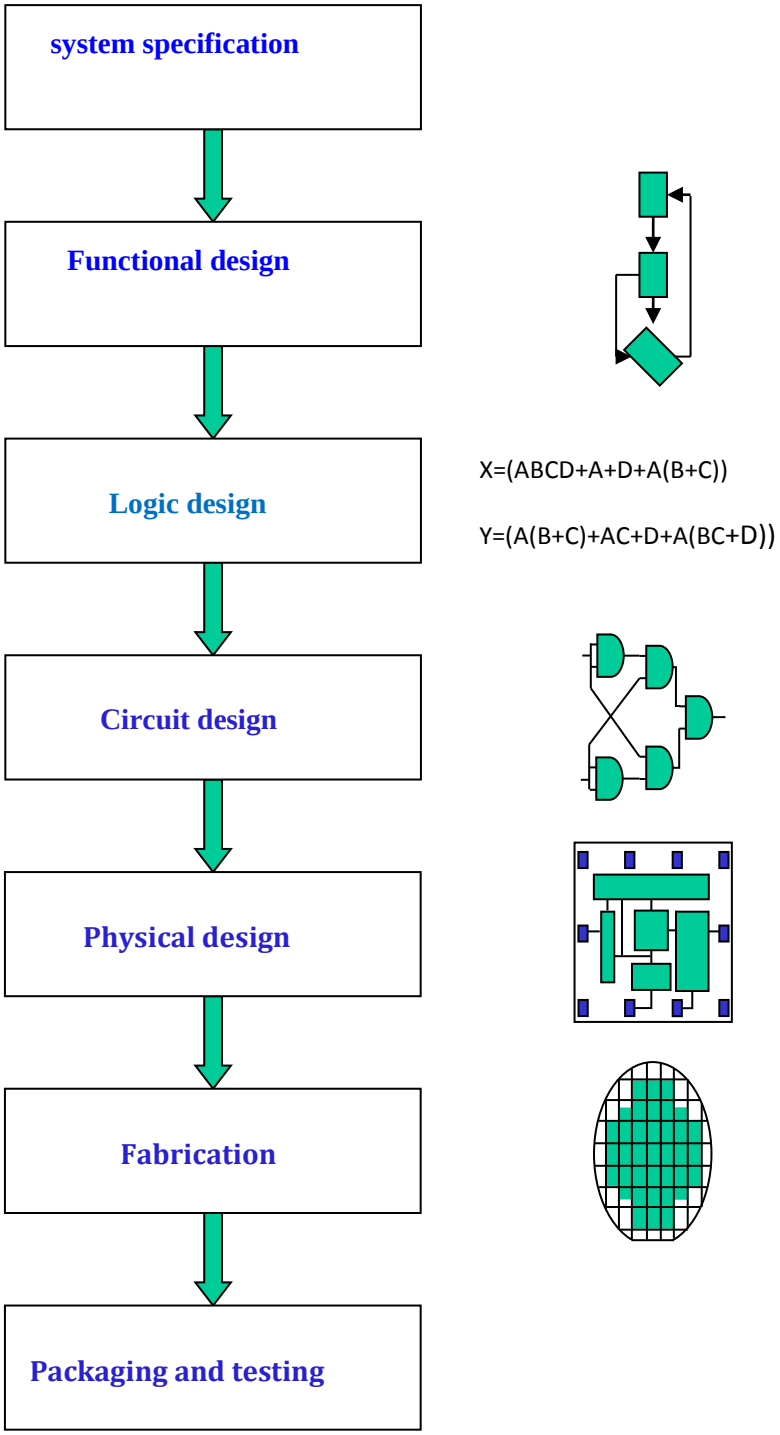


fig 1.2 VLSI Design flow

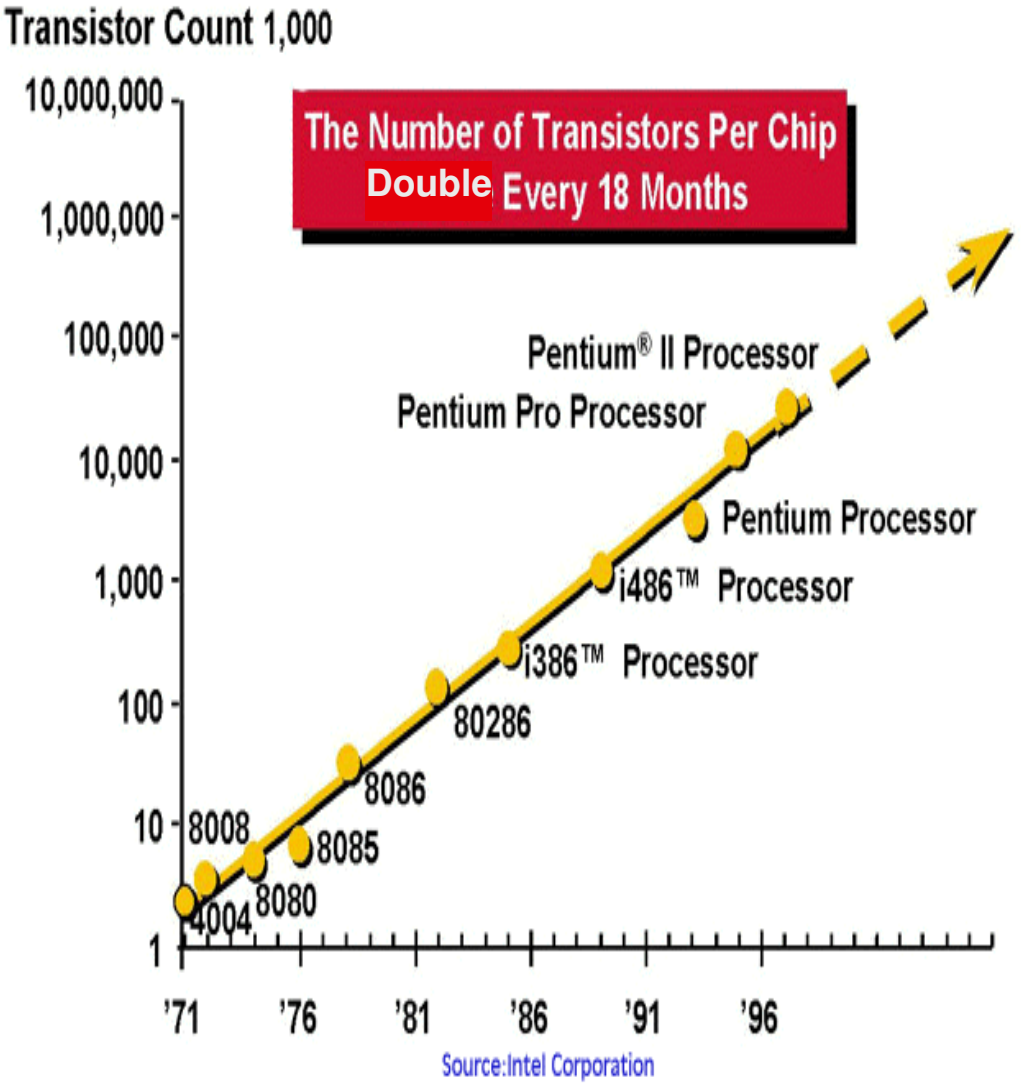


fig 1.3 Defining Moores law

1.4 POPULARITY OF VERILOG HDL

Verilog hdl has evolved as a standard hardware description language. Verilog offers many useful features for hardware design.

- Verilog is a general purpose hardware description language that is easy to learn and to use. It has thus become user friendly.
- Verilog allows different levels of abstraction to be mixed in the same model. Thus a designer can design a hardware model in terms of switches, gates, RTL or behavioral codes.
- Verilog is one language for simulation and design. Most popular logic synthesis tools support Verilog HDL. This makes it the language of choice of designers .

1.5 TYPICAL DESIGN FLOW

Typical design flow for designing VLSI circuits is shown in the tool flow diagram.

This design flow is typically used by designers who use HDLs. In any design, specification are first. Specification describes the functionality, interface and overall architecture of the digital circuit to be designed. At this point, architects need not think about how they will implement their circuit. A behavioral description is then created to analyze the design in terms of functionality, performances and other high level issues. The behavioral description is manually converted to an RTL (Register Transfer Level) description in an HDL. The designer has to describe the data flow that will implement the desired digital circuit. From this point onward the design process is done with assistance of CAD tools.

Logic synthesis tools convert the RTL description to a gate level netlist. A gate level net list is a description of the circuit in terms of gates and connections between them. The gate level net list is input to an automatic place and route tool, which creates a layout. The layout is verified and then fabricated on a chip. Thus most digital design activity is concentrated on manually optimizing the RTL description of the circuit. After the RTL description is frozen, CAD tools are available to assist the designer in further process Designing at RTL level has shrunk design cycle times from years to a few months.

1.6 LOW POWER VLSI:

The advantage of utilizing a combination of low-power components in conjunction with low-power design techniques is more valuable now than ever before. Requirements for lower power consumption continue to increase significantly as components become battery-powered, smaller and require more functionality. In the past the major concerns for the VLSI designers was area, performance and cost. Power consideration was the secondary concerned. Now a day's power is the primary concerned due to the remarkable growth and success in the field of personal computing devices and wireless communication system which demand high speed computation and complex functionality with low power consumption. The motivations for reducing power consumption differ application to application.

In the class of micro-powered battery operated portable applications such as cell phones, the goal is to keep the battery lifetime and weight reasonable and packaging cost low. For high performance portable computers such as laptop the goal is to reduce the power dissipation of the electronics portion of the system to a point which is about half of the total power dissipation.

Finally For the high performance non battery operated system such as workstations the overall goal of power minimization is to reduce the system cost while ensuring long term device reliability.

For such high performance systems, process technology has driven power to the fore front to all factors in such designs. At process nodes below 100 nm technology, power consumption due to leakage has joined switching activity as a primary power management concern. There are many techniques that have been developed over the past decade to address the continuously aggressive power reduction requirements of most of the high performance. The basic low-power design techniques, such as clock gating for reducing dynamic power, or multiple voltage thresholds (multi-Vt) to decrease leakage current, are well-established and supported by existing tools.

In the past, the major concerns of the VLSI designer were area, performance, cost and reliability; power consideration was mostly of only secondary importance. In recent years, however, this has begun to change and, increasingly, power is being given comparable weight to area and speed considerations. Several factors have contributed to this trend. Perhaps the primary driving factor has been the remarkable success and growth of the class of personal computing devices (portable desktops, audio- and video-based multimedia products) and wireless communications systems (personal digital assistants and personal communicators) which

demand high-speed computation and complex functionality with low power consumption. In these applications, average power consumption is a critical design concern. The projected power budget for a battery-powered, A4 format, portable multimedia terminal, when implemented using off-the-shelf components not optimized for low-power operation, is about 40 W. With advanced Nickel-Metal-Hydride (secondary) battery technologies offering around 65 watt W hours/kilograms, this terminal would require an unacceptable 6 kilograms of batteries for 10 hours of operation between recharges. Even with new battery technologies such as rechargeable lithium ion or lithium polymer cells, it

is anticipated that the expected battery lifetime will increase to about 90-110 watt-hours/kilogram over the next 5 years which still leads to an unacceptable 3.6-4.4 kilograms of battery cells. In the absence of low-power design techniques then, current and future portable devices will suffer from either very short battery life or very heavy battery pack.

There also exists a strong pressure for producers of high-end products to reduce their power consumption. Contemporary performance optimized microprocessors dissipate as much as 15-30 W at 100-200 MHz clock rates. In the future, it can be extrapolated that a 10 *cm2* microprocessor, clocked at 500 MHz (which is a not too aggressive estimate for the next decade) would consume about 300 W. The cost associated with packaging and cooling such devices is prohibitive. Since core power consumption must be dissipated through the packaging, increasingly expensive packaging and cooling strategies are required as chip power consumption increases. Consequently, there is a clear financial advantage to reducing the power consumed in high performance systems. In addition to cost, there is the issue of reliability. High power systems often run hot, and high temperature tends to exacerbate several silicon failure mechanisms. Every 10 °C increase in operating temperature roughly doubles a component's failure rate . In this context, peak power (maximum possible power dissipation) is a critical design factor as it determines the thermal and electrical limits of designs, impacts the system cost, size and weight, dictates specific battery type, component and system packaging and heat sinks, and aggravates the resistive and inductive volage drop problems. It is therefore essential to have the peak power under control. Another crucial driving factor is that excessive power consumption is becoming the limiting factor in integrating more transistors on a single chip or on a multiple-chip module. Unless power consumption is dramatically reduced, the resulting heat will limit the feasible packing and performance of VLSI circuits and systems.

From the environmental viewpoint, the smaller the power dissipation of electronic systems, the lower the heat pumped into the rooms, the lower the electricity consumed and hence the lower the impact on global environment, the less the office noise (e.g., due to elimination of a fan from the desktop), and the less stringent the environment/office power delivery or heat removal requirements. The motivations for reducing power consumption differ from application to application. In the class of micro-powered battery-operated, portable applications, such as cellular phones and personal digital assistants, the goal is to keep the battery lifetime and weight reasonable and the packaging cost low. Power levels below 1-2 W, for instance, enable the use of inexpensive plastic packages. For high performance, portable computers, such as laptop and notebook computers, the goal is to reduce the power dissipation of the electronics portion of the system to a point which is about half of the total power dissipation (including that of display and hard disk).

Finally, for high performance, non battery operated systems, such as workstations, set-top computers and multimedia digital signal processors, the overall goal of power minimization is to reduce system cost (cooling, packaging and energy bill) while ensuring long-term device reliability. These different requirements impact how power optimization is addressed and how much the designer is willing to sacrifice in cost or performance to obtain lower power dissipation. The next question is to determine the objective function to minimize during low power design. The answer varies from one application domain to next. If extending the battery life is the only concern, then the energy (that is, the power-delay product) should be minimized. In this case the battery consumption is minimized even though an operation may take a very long time. On the other hand, if both the battery life and the circuit delay are important, then the energy-delay product must be minimized. In this case one can alternatively minimize the energy/delay ratio (that is, the power) subject to a delay constraint. In most design scenarios, the circuit delay is set based on system-level considerations, and hence during circuit optimization, one minimizes power under user-specified timing constraints.

CHAPTER-2
POWER DISSIPATION

2.1 POWER DISSIPATION BASICS:

In a circuit three components are responsible for power dissipation: dynamic power, short-circuit power and static power. Out of these, dynamic power or switching power is primarily power dissipated when charging or discharging capacitors and is described below :

$$P_{dyn} = C_L V_{dd}^2 \alpha f$$

Where CL: Load Capacitance, a function of fan-out, wirelength, and transistor size, Vdd: Supply Voltage, which has been dropping with successive process nodes, α: Activity Factor, meaning how often, on average, the wires switch, f :Clock Frequency, which is increasing at each successive process node. Static power or leakage power is a function of the supply voltage(Vdd), the switching threshold (Vt), and transistor sizes (figure2). As process nodes shrink, leakage becomes a more significant source of energy use, consuming at least 30% of total power.

Crowbar currents, caused when both the PMOS and NMOS devices are simultaneously on, also contribute to the leakage power dissipation. Most circuit level minimization techniques focus only on Sub threshold leakage reduction without considering the effects of gate leakage. been proposed for reduction of sub-threshold leakage current in sleep mode. Figure-2 shows the various components responsible for power dissipation in CMOS.

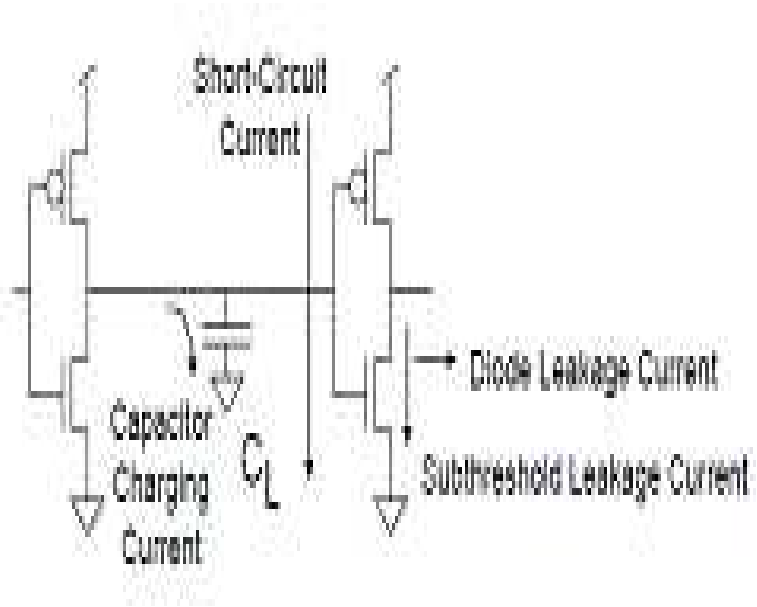


fig2.1 Power Dissipation in CMOS

2.2 SUB-THRESHOLD:

In the medium performance, medium power consumption design region, numerous optimization efforts have been made. However, not much study has been done at the two ends of the design spectrum, namely ultra low power with acceptable performance at one end, and high performance design with power within specified limit at the other end. This paper focuses on design techniques for ultra low power dissipation where performance is of secondary importance. One way to achieve this goal is by running the digital circuits in sub-threshold mode. The incentive of operating the circuit in sub-threshold mode is to able to exploit the sub-threshold leakage current as the operating drive current. The sub-threshold current is exponentially related to the gate voltage. This exponential relationship is expected to give an exponential increase in delay. The simulation results show that the reduction in power outweighs the increase in delay, and thus, giving the overall reduction in energy consumption per switching.

Sub-threshold digital circuits will be suitable only for specific application which do not need high performance, but require extremely low power consumptions. This type of applications includes medical equipments such as hearing aids and pace maker, wearable wrist watch computation, and self powered devices. Sub-threshold circuits can also be applied to application with bursty characteristics in which the circuits remain idle for an extended period of time. The original active time period T in strong inversion region (top off) is being extended throughout the idle time period T running in sub-threshold region(bottom half). The same number of operations is performed in both cases, but with much lower power consumption in the sub-threshold operation.

Sub-threshold logic operates with the power supply Vdd less than the transistors threshold voltage Vt. This is done to ensure that all the transistors are indeed operating in the sub-threshold region. We use 90nm process technology for our circuit simulation with Vt of NMOS and PMOS transistor as 0.169v and 0.178V respectively. In subthreshold region, for Vds>3kT/q, Ids becomes independent of Vds for all practical purposes. In analog design, this favorable characteristic has been extensively exploited as it provides an excellent current source that spans for almost the entire rail to rail voltage range. In digital design, circuit designers can take

advantage of this characteristic by being able to use more series connected transistors. The 3kT/q drop is practically negligible compared to the Vt drop in the normal strong inversion region.

Static CMOS is the most common logic style used in sub-threshold due to its robustness. Pseudo-NMOS has also been proposed because some of its disadvantages in strong inversion are mitigated in sub-threshold. The always-on PMOS pull up in pseudo- NMOS is less sensitive to changes in size, but more sensitive to process variations. As a result, the pseudo-NMOS logic style cannot function well in strong-PMOS technologies, because variations in the pull-up device can cause it to overpower the pull-down network despite efforts to counteract this by sizing. Specifically, the distribution of the output low logic level (VOL) can reach nearly to VDD. This sensitivity dramatically reduces the yield of pseudo- NMOS logic for strong-P processes. Static CMOS logic is more robust across different process balances in terms of functionality. However, different metrics applied to static CMOS will vary broadly as process balance changes. This has strong implications for standard cells designed to operate in sub-threshold. Characterization of standard cell libraries will vary dramatically with technology.

RAPID advances in digital circuit design have enabled a number of applications requiring complex sensor networks. This application space ranges widely from environmental sensing to structural monitoring to supply chain management. Highly integrated sensor network platforms would combine MEMS sensing capabilities with b digital processing and storage hardware, a low power radio, and an on-chip battery in a volume on the order of 1 mm . The design of energy-efficient data processing and storage elements is therefore paramount. Voltage scaling into the sub-threshold regime has recently been shown to be an extremely effective technique for achieving minimum energy. In previous work, we demonstrated the existence of a minimum energy voltage , where CMOS logic reaches maximum energy efficiency per operation. This occurs when leakage energy and dynamic energy are comparable. Fig shows the simulated energy consumption of a chain of 50 inverters as a function of supply
voltage in 0.13- m technology.

2.3 STANDARD CMOS INVERTER:

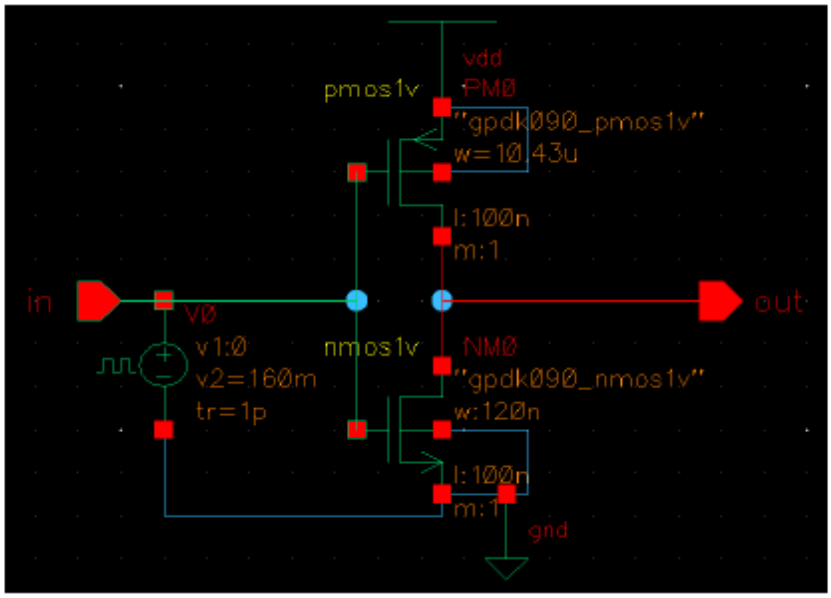


fig 2.2 Schematic of CMOS inverter using sub threshold

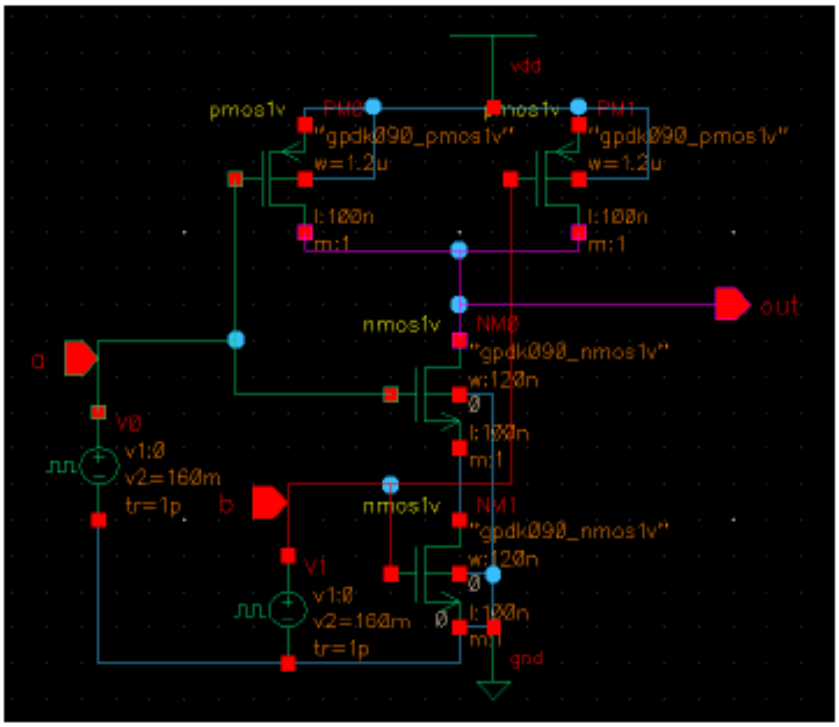


fig2.3 Schematic diagram of NAND gate

The schematic circuit of the basic NAND gate is shown in figure where the nand gate is operated in sub-threshold region. The sizing of the pmos to nmos is done by using some spice calculations. In the sub-threshold region of operation the the pmos to nmos ratio is high in order get proper outputs. The capacitance of the drain and source plays a major role in effecting the outputs. In sub-threshold region of operation the power consumption is reduced by increasing the delay of the circuit

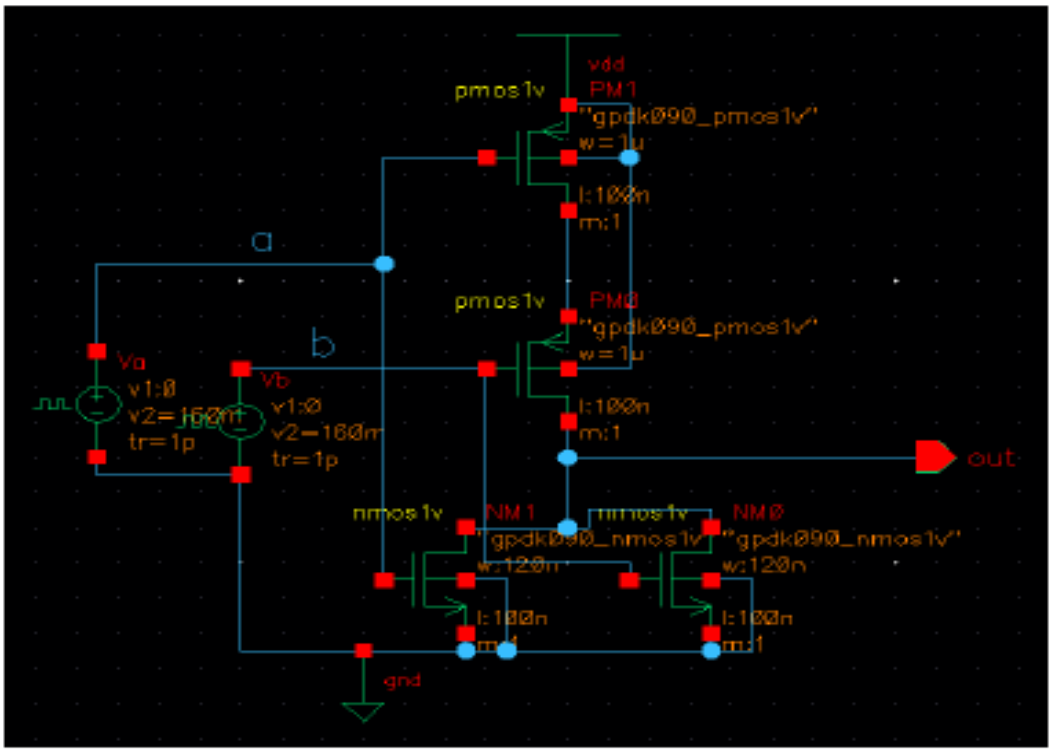


fig2.4 Schematic diagram of NOR GATE

2.4 POWER GATING LOGIC:

Power gating affects design architecture more than clock gating. It increases time delays, as power gated modes have to be safely entered and exited. Architectural trade-offs exist

between designing for the amount of leakage power saving in low power modes and the energy dissipation to enter and exit the low power modes. Shutting down the blocks can be accomplished either by software or hardware. Driver software can schedule the power down operations. Hardware timers can be utilized. A dedicated power management controller is another option.

An externally switched power supply is a very basic form of power gating to achieve long term leakage power reduction. To shut off the block for small intervals of time, internal power gating is more suitable. CMOS switches that provide power to the circuitry are controlled by power gating controllers. Outputs of the power gated block discharge slowly. Hence output voltage levels spend more time in threshold voltage level. This can lead to larger short circuit current.

Power gating uses low-leakage PMOS transistors as header switches to shut off power supplies to parts of a design in standby or sleep mode. NMOS footer switches can also be used as sleep transistors. Inserting the sleep transistors splits the chip's power network into a permanent power network connected to the power supply and a virtual power network that drives the cells and can be turned off.

Typically, high-Vt sleep transistors are used for power gating, in a technique also known as multi-threshold CMOS (MTCMOS). The sleep transistor sizing is an important design parameter. The quality of this complex power network is critical to the success of a power-gating design. Two of the most critical parameters are the IR-drop and the penalties in silicon area and routing resources. Power gating can be implemented using cell- or cluster-based (or fine grain) approaches or a distributed coarse-grained approach.

Power gating implementation has additional considerations for timing closure implementation. The following parameters need to be considered and their values carefully chosen for a successful implementation of this methodology.

1. *Power gate size*: The power gate size must be selected to handle the amount of switching current at any given time. The gate must be bigger such that there is no measurable voltage (IR) drop due to the gate. As a rule of thumb, the gate size is selected to be around 3 times the switching capacitance. Designers can also choose between header (P-

MOS) or footer (N-MOS) gate. Usually footer gates tend to be smaller in area for the same switching current. Dynamic power analysis tools can accurately measure the switching current and also predict the size for the power gate.

2. *Gate control slew rate*: In power gating, this is an important parameter that determines the power gating efficiency. When the slew rate is large, it takes more time to switch off and switch-on the circuit and hence can affect the power gating efficiency. Slew rate is controlled through buffering the gate control signal.
3. *Simultaneous switching capacitance*: This important constraint refers to the amount of circuit that can be switched simultaneously without affecting the power network integrity. If a large amount of the circuit is switched simultaneously, the resulting "rush current" can compromise the power network integrity. The circuit needs to be switched in stages in order to prevent this.
4. *Power gate leakage*: Since power gates are made of active transistors, leakage reduction is an important consideration to maximize power savings.

2.5 POWER GATING METHODS:

Fine-grain power gating:

Adding a sleep transistor to every cell that is to be turned off imposes a large area penalty, and individually gating the power of every cluster of cells creates timing issues introduced by inter-cluster voltage variation that are difficult to resolve. Fine-grain power gating encapsulates the switching transistor as a part of the standard cell logic. Switching transistors are designed by either the library IP vendor or standard cell designer. Usually these cell designs conform to the normal standard cell rules and can easily be handled by EDA tools for implementation.

The size of the gate control is designed considering the worst case scenario that will require the circuit to switch during every clock cycle, resulting in a huge area impact. Some of the recent designs implement the fine-grain power gating selectively, but only for the low Vt cells. If the technology allows multiple Vt libraries, the use of low Vt devices is minimum in the design (20%), so that the area impact can be reduced. When using power gates on the low Vt cells the output must be isolated if the next stage is a high Vt cell. Otherwise it can cause the neighboring high Vt cell to have leakage when output goes to an unknown state due to power gating.

Gate control slew rate constraint is achieved by having a buffer distribution tree for the control signals. The buffers must be chosen from a set of always on buffers (buffers without the gate control signal) designed with high Vt cells. The inherent difference between when a cell switches off with respect to another, minimizes the rush current during switch-on and switch-off.

Usually the gating transistor is designed as a high Vt device. Coarse-grain power gating offers further flexibility by optimizing the power gating cells where there is low switching activity. Leakage optimization has to be done at the coarse grain level, swapping the low leakage cell for the high leakage one. Fine-grain power gating is an elegant methodology resulting in up to 10 times leakage reduction. This type of power reduction makes it an appealing technique if the power reduction requirement is not satisfied by multiple Vt optimization alone.

Coarse-grain power gating:

The coarse-grained approach implements the grid style sleep transistors which drives cells locally through shared virtual power networks. This approach is less sensitive to PVT variation, introduces less IR-drop variation, and imposes a smaller area overhead than the cell- or cluster-based implementations. In coarse-grain power gating, the power-gating transistor is a part of the power distribution network rather than the standard cell.

There are two ways of implementing a coarse-grain structure:

1. Ring-based: The power gates are placed around the perimeter of the module that is being switched-off as a ring. Special corner cells are used to turn the power signals around the corners.
2. Column-based: The power gates are inserted within the module with the cells abutted to each other in the form of columns. The global power is the higher layers of metal, while the switched power is in the lower layers.

Gate sizing depends on the overall switching current of the module at any given time. Since only a fraction of circuits switch at any point of time, power gate sizes are smaller as compared to the fine-grain switches. Dynamic power simulation using worst case vectors can determine the worst case switching for the module and hence the size. The IR drop can also be factored into the analysis. Simultaneous switching capacitance is a major consideration in coarse-grain power

gating implementation. In order to limit simultaneous switching, gate control buffers can be daisy chained, and special counters can be used to selectively turn on blocks of switches.

Isolation cells

Isolation cells are used to prevent short circuit current. As the name suggests, these cells isolate the power gated block from the normally-On block. Isolation cells are specially designed for low short circuit current when input is at threshold voltage level. Isolation control signals are provided by the power gating controller. Isolation of the signals of a switchable module is essential to preserve design integrity. Usually a simple OR or AND logic can function as an output isolation device. Multiple state retention schemes are available in practice to preserve the state before a module shuts down. The simplest technique is to scan out the register values into a memory before shutting down a module. When the module wakes up, the values are scanned back from the memory.

Retention registers

When power gating is used, the system needs some form of state retention, such as scanning out data to a RAM, then scanning it back in when the system is reawakened. For critical applications, the memory states must be maintained within the cell, a condition that requires a retention flop to store bits in a table. That makes it possible to restore the bits very quickly during wakeup. Retention registers are special low leakage flip-flops used to hold the data of main register of the power gated block. Thus internal state of the block during power down mode can be retained and loaded back to it when the block is reactivated. Retention registers are always powered up. The retention strategy is design dependent. During the power gating data can be retained and transferred back to block when power gating is withdrawn. Power gating controller controls the retention mechanism such as when to save the current contents of the power gating block and when to restore it back.

The large magnitude of supply/ground bounces, which arise from power mode transitions in power gating structures, may cause spurious transitions in a circuit. This can result in wrong values being latched in the circuit registers. We propose a design methodology for limiting the maximum value of the supply/ground currents to a user-specified threshold level while minimizing the wake up (sleep to active mode transition) time. In addition to controlling the sudden discharge of the accumulated charge in the intermediate nodes of the circuit through the

sleep transistors during the wake up transition, we can eliminate short circuit current and spurious switching activity during this time. This is in turn achieved by reducing the amount of charge that must be removed from the intermediate nodes of the circuit and by turning on different parts of the circuit in a way that causes a uniform distribution of current over the wake up time. Simulation results show that, compared to existing wakeup scheduling methods, the proposed techniques result in a one to two orders of magnitude improvement in the product of the maximum ground current and the wake up time.

The most obvious way of reducing the leakage power dissipation of a VLSI circuit in the STANDBY state is to remove its supply voltage. Multi-threshold CMOS (MTCMOS) technology provides low leakage and high performance operation by utilizing high speed, low Vt transistors

for logic cells and low leakage, high Vt devices as sleep transistors. Sleep transistors disconnect logic cells from the power supply and/or ground to reduce the leakage in sleep mode. More precisely, this can be done by using one PMOS transistor and one NMOS transistor in series with the transistors of each logic block to create a virtual ground and a virtual power supply as depicted 2 in Figure. In practice only one transistor is necessary. Because of the lower on-resistance, NMOS transistors are usually used.

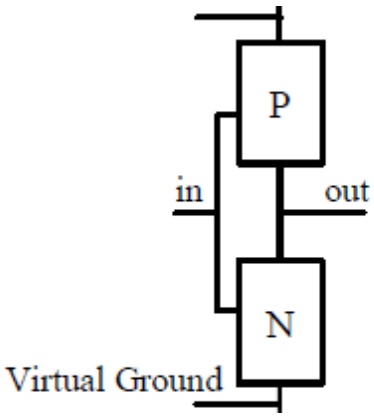


fig 2.5 Power Gating Technique

2.6 POWER GATING:

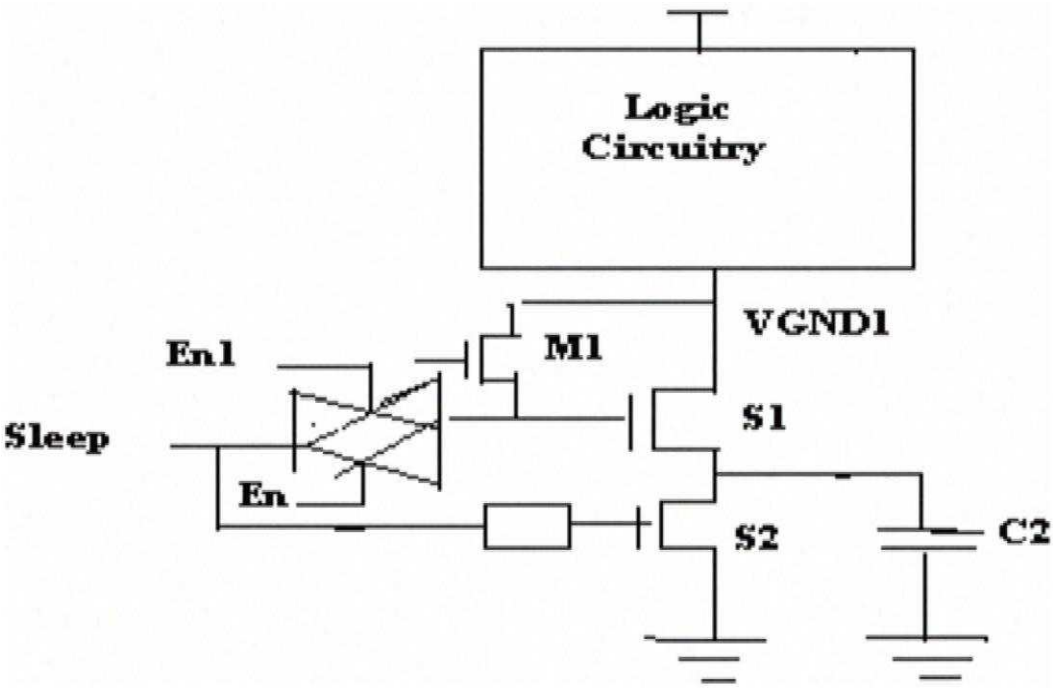


fig 2.6 logic circuit with power gating technique

In the ACTIVE state, the sleep transistor is on. Therefore, the circuit functions as usual. In the STANDBY state, the transistor is turned off, which disconnects the gate from the ground. To lower the leakage, the threshold voltage of the sleep transistor must be large. Otherwise, the sleep transistor will have a high leakage current, which will make the power gating less effective. Additional savings may be achieved if the width of the sleep transistor is smaller than the combined width of the transistors in the pull-down network. In practice, Dual VT CMOS or Multi-Threshold CMOS (MTCMOS) is used for power gating. In these technologies there are several types of transistors with different VT values. Transistors with a low VT are used to implement the logic, while high-VT devices are used as sleep transistors.

To guarantee the proper functionality of the circuit, the sleep transistor has to be carefully sized to decrease the voltage drop across it when the sleep transistor is turned on. The voltage drop decreases the effective value of the supply voltage that the logic gate receives. In addition, it increases the threshold voltage of the pull-down transistors due to the body effect. This phenomenon in turn increases the high-to-low transition delay of the circuit. The problem can be

solved by using a large sleep transistor. On the other hand, using a large sleep transistor increases the area overhead and the dynamic power consumed for turning the sleep transistor on and off. Note that because of this dynamic power consumption, it is not possible to save power for very short idle periods. There is a minimum duration of the idle time below which power saving is impossible.

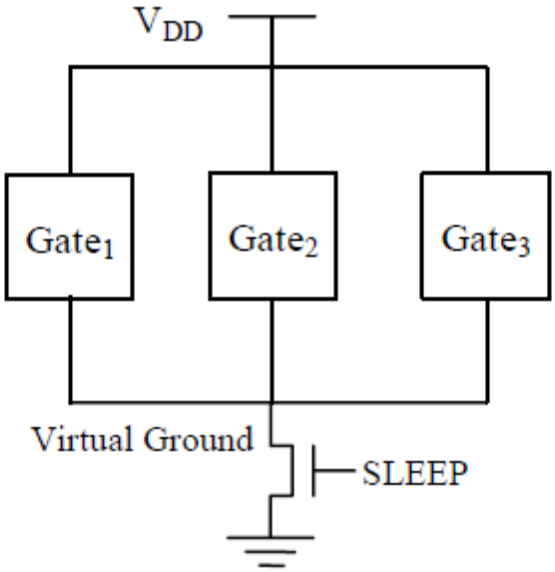


Fig 2.7: Using one sleep transistor for several gates

Since using one transistor for each logic gate results in a large area and power overhead, one transistor may be used for each group of gates as depicted in Figure. Notice that the size of the sleep transistor in this figure ought to be larger than the one used in Figure. To find the optimum size of the sleep transistor, it is necessary to find the vector that causes the worst case delay in the circuit. This requires simulating the circuit under all possible input values, a task that is not possible for large circuits.

In this technology, also called *power gating*, wake up latency and power plane integrity are key concerns. Assuming a sleep/wake up signal provided from a power management unit, an important issue is to minimize the time required to turn on the circuit upon receiving the wake up signal since the length of wake up time can affect the overall performance of the VLSI circuit. Furthermore, the large current flowing to ground when sleep transistors are turned on can become a major source of noise on the power distribution network, which can in turn adversely

impact the performance and/or functionality of the other parts of the circuit. There is trade off between the amount of current flowing to ground and the transition time from the sleep mode to the active mode.

In this paper we introduce an approach for reducing the transition time from sleep mode to active mode for a circuit part while assuring power integrity for the rest of the system by restricting the current that flows to ground during the transition. The problem is to minimize the wakeup time while constraining the current flowing to ground during the sleep to active mode transition. During the process we will also consider another important objective: limiting the number of sleep transistors.

It is a well known fact that there is no need to have both NMOS and PMOS sleep transistors to encapsulate a logic cell. In particular, NMOS sleep transistors can be used to separate the (actual) ground from the virtual ground of the logic cell. Upon entering the sleep mode, a circuit block is disconnected from the ground. This causes the voltage levels of some intermediate nodes in the circuit block to rise toward Vdd. When the circuit block is woken up, the nodes will transition to zero. This transition in turn causes the logic cells in the immediate fanout of the node to carry a potentially large amount of short circuit current as explained next. Consider the inverter chain shown in Figure, which is connected to the ground through an NMOS sleep transistor.

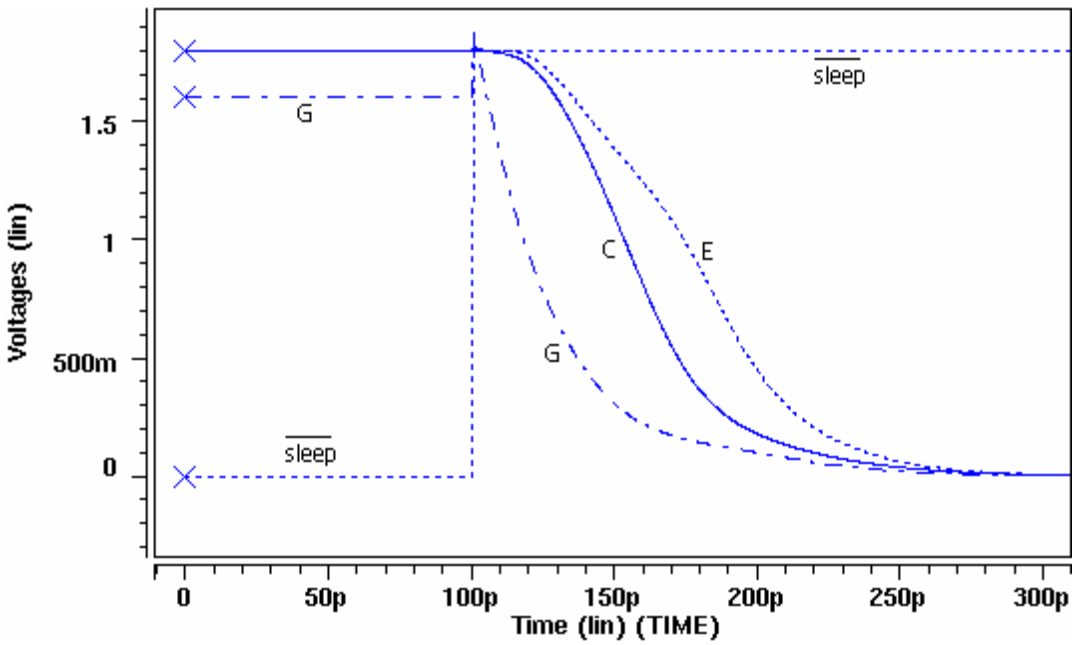


fig 2.8 graph showing the voltage waveforms of nodes C, E, and G generated by HSPICE simulation. While turning on the sleep transistor, nodes G, C and E discharges.

If the input of the circuit is low, then, in the active mode (i.e., SLEEP=0), VA=VC=VE=VG=0 and VB=VD=VDD. When entering the sleep mode, the voltages of B and D do not change, but the voltages of C, E, and G gradually increase and will be equal to VDD if the sleep period is long enough (note the driver of signal A is not controlled by the SLEEP signal). This happens because the leakage through the PMOS transistors will charge up all the floating capacitances. Figure 2.7 shows the voltage waveforms of nodes C, E, and G generated by HSPICE simulation. While turning on the sleep transistor, nodes G, C and E discharge as depicted in Figure.

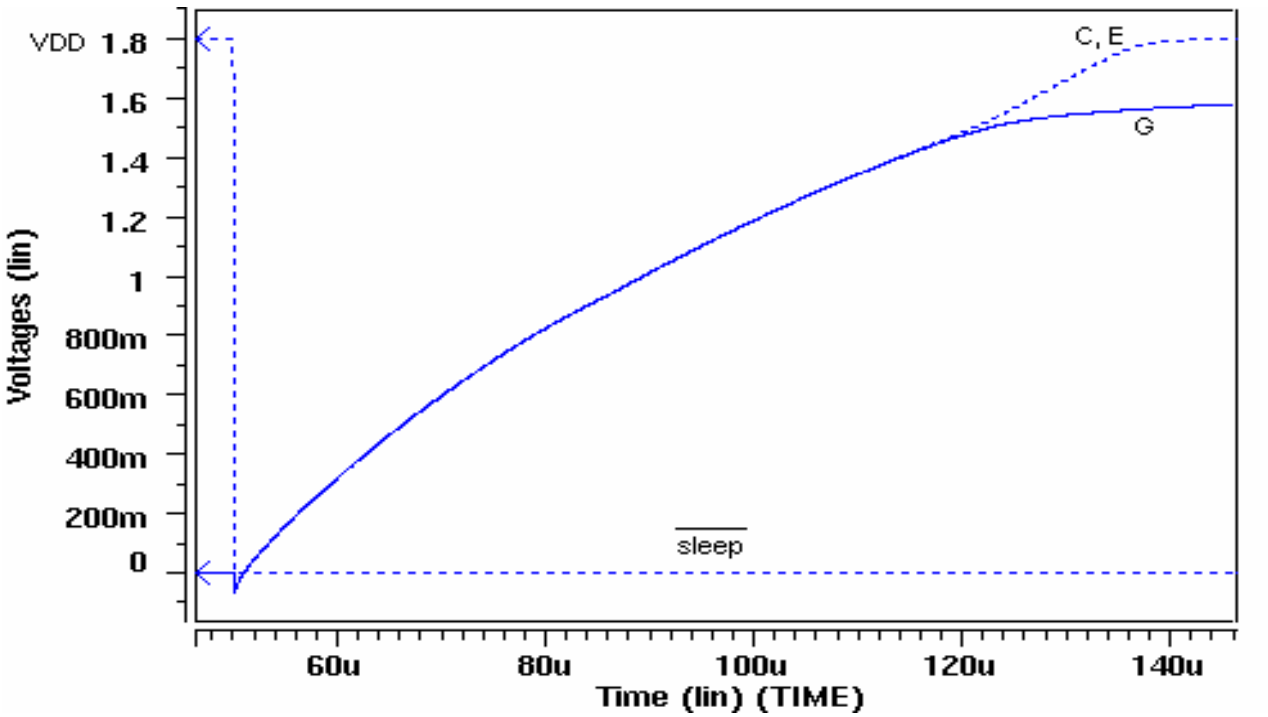


fig 2.9 graph showing the voltage waveforms of nodes C, E, and G generated by HSPICE simulation during active mode

Power gating is a circuit design technique which reduces the static power consumption of a digital circuit by inserting power switches in the supply path, and has been most widely used in industrial products. There are two implemented schemes of power gating technique: coarse-grain power gating, in which complete blocks are disconnected from the power supply and the ground through a common power switch, and fine-grain power gating, in which every standard cell contains a sleep transistor internally. Different power gating topologies for MCML circuits are depicted in Figure. The solutions (a) and (b) use a transistor to pull down the bias voltage VRFN to ground during the sleep mode. Solution (c) applies just a ON signal to the gate of the current source and connects the bulk voltage to the bias voltage *VRFN*. Options (d) and (e) consist of an additional sleep transistor in series with the current source and *V*dd, respectively.

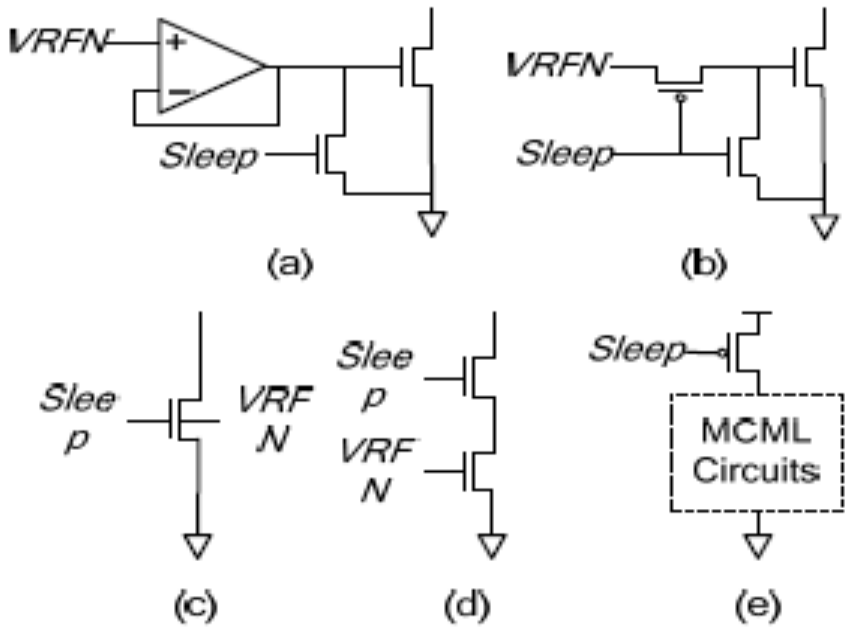


fig 2.10 Circuits with Sleepy approach

Solution (a) was discarded since it uses a follower amplifier which leads to a significant area overhead. With 500mV to 1 V voltage to modulate the bias current, solution is difficult to be implemented in practice. Option and are not suite the near-threshold circuits because the additional sleep transistors will further decease the operation voltage. For all the above reasons, we selected solution to implement the power-gating near-threshold MCML circuits.

The proposed scheme of power-gating near-threshold MCML circuits is shown in Figure

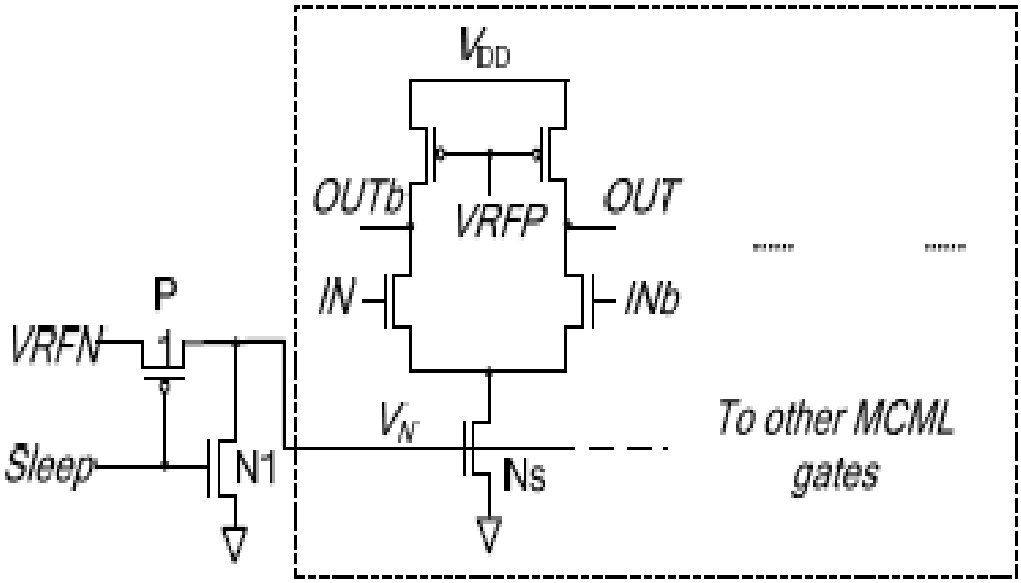


fig 2.11 Scheme of power-gating MCML circuits

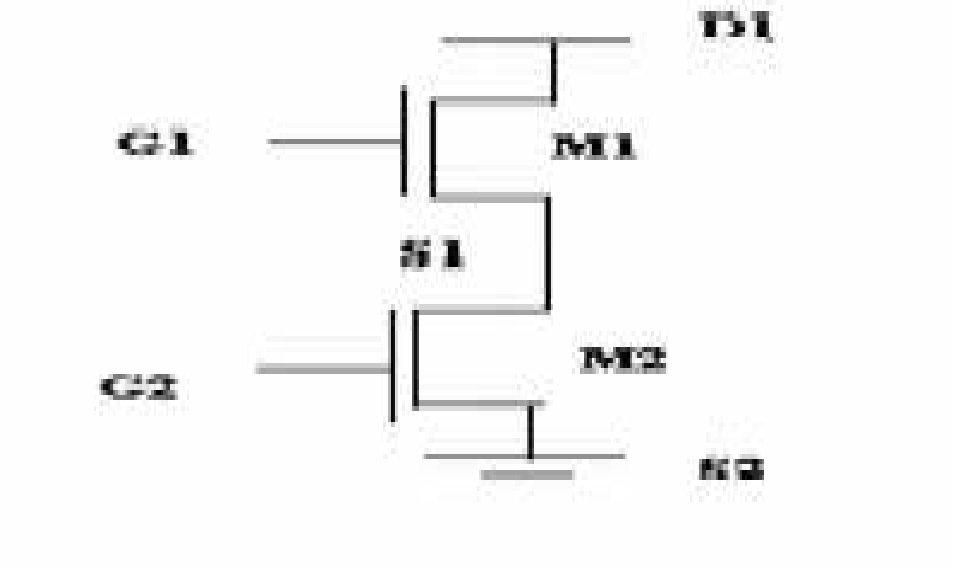


fig 2.12 Strategy to reduce leakage current in standby mode:

Here sleep transistors M 1 and M2 are stacked. When in standby mode i. e.When MI and M2 both are off. In this structure firstly, the leakage current is reduced by stacking effect, turning both M 1 and M2 sleep transistors OFF. This raises the intermediate node voltage VGN to positive values to small drain current. In the analyzed scheme, the leakage current is reduced by the stacking effect, turning both Ml and M2 sleep transistors OFF. This raises the intermediate node voltage VGND2 to positive values due to small drain current. Positive potential at the intermediate node has four effects:

CHAPTER-3

CMOS ADIABATIC CIRCUITS

3.1 CMOS adiabatic circuits:

There are some classical approaches to reduce the dynamic power such as reducing supply voltage, decreasing physical capacitance and reducing switching activity. These techniques are not fit enough to meet today's power requirement. However, most research has focused on building adiabatic logic, which is a promising design for low power applications. Adiabatic logic works with the concept of switching activities which reduces the power by giving stored energy back to the supply. Thus, the term adiabatic logic is used in low-power VLSI circuits which implements reversible logic. In this, the main design changes are focused in power clock which plays the vital role in the principle of operation. Each phase of the power clock gives user to achieve the two major design rules for the adiabatic circuit design.

- Never turn on a transistor if there is a voltage across it (VDS>0)
- Never turn off a transistor if there is a current through it (IDS≠ 0)
- Never pass current through a diode

If these conditions with regard to the inputs, in all the four phases of power clock, recovery phase will restore the energy to the power clock, resulting considerable energy saving. Yet some complexities in adiabatic logic design perpetuate. Two such complexities, for instance, are circuit implementation for time-varying power sources needs to be done and computational implementation by low overhead circuit structures needs to be followed. There are two big challenges of energy recovering circuits; first, slowness in terms of today's standards, second it requires ~50% of more area than conventional CMOS, and simple circuit designs get complicated. The basic concepts of Adiabatic logic will be introduced. "Adiabatic" is a term of Greek origin that has spent most of its history associated with classical thermodynamics. It refers to a system in which a transition occurs without energy (usually in the form of heat) being either lost to or gained from the system. In the context of electronic systems, rather than heat, electronic charge is preserved.

3.2 ADIABATIC LOGIC BASED FULL ADDER:

Energy efficient adiabatic logic:

EEAL requires only one sinusoidal power supply, has simple implementation, and performs better than the previously proposed adiabatic logic families in terms of energy consumption. As single clock circuit requires simple clock scheme, this logic style can enjoy minimal control overheads. Assuming the complementary output nodes ("out" and "outb") are initially low and supply clock ramps up from logic 0 to ("0") to logic 1("VDD") state. Now if "in"="0" and inb="1" N1, M1 will be turned off and M2,N2 and P1 will be turned ON. The OUT node is then charged by following power supply clock closely through the parallel combination of PMOS (P1) and NMOS (M2), whereas "outb" potential is kept at ground potential, as N2 is ON. When the supply clock swings from "VDD" to ground, "out" Node is discharged through the same charging path and un-driven "outb" is kept at ground potential. Resultantly full swing can be obtained in "out" node and ground potential at "outb" node. Output voltage swing for an adiabatic inverter at 100 MHz frequencies with 20pf capacitive load is shown in figure 1.

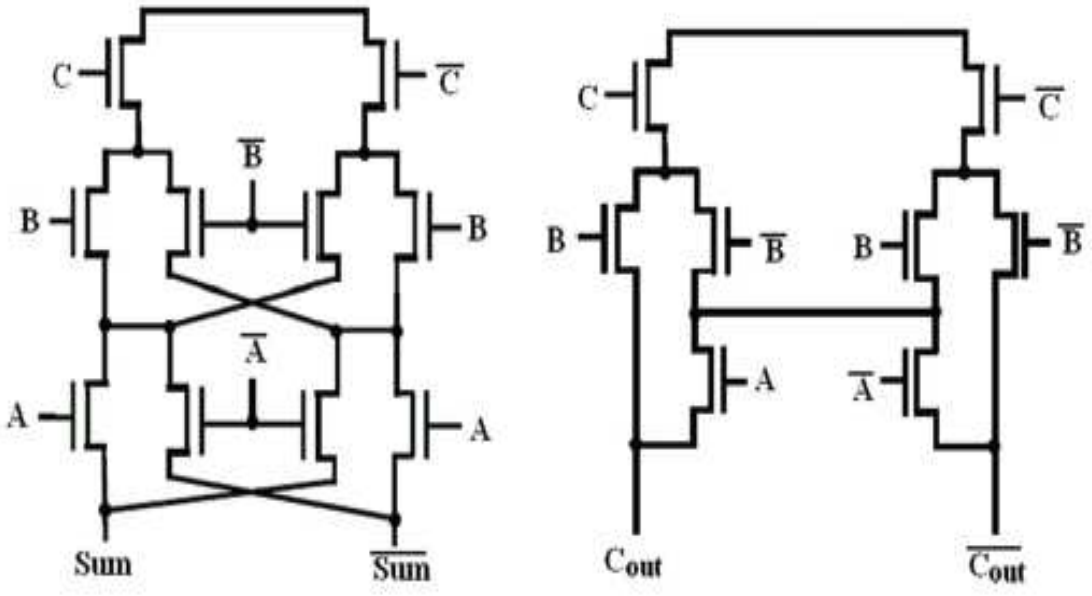


Fig3.1. FULL ADDER Circuit

The energy advantage of EEAL circuit can be readily understood by assuming a ramp type voltage source which ramps up between "0" and "VDD" and delivers the charge CLVDD over a time period T. The Dissipation through the channel resistance R is,

$$E_{diss}=\{(C_L V_{DD})/T\}^2 RT = \{(RC_L)/T\}C_L(V_{DD})^2$$

Similarly, energy consumption during charging and discharging process of the EEAL inverter/buffer can be expressed as,

$$E = \{(R_p C_L)/T\}\ C_L(V_{DD})^2 + ½\ C_L(\Delta V)^2$$

Hence Rp is the turn-on resistance of the parallel path,CL is the output load capacitances, T is the charging time and □V depends on the time, yet due to very small magnitude and the parameter is treated as constant. In Equation (2) $(½C_L(\Delta V)^2)$ measure the threshold loss which is negligibly small indeed. In EEAL as charging and discharging processes consume almost similar amount of energy, total energy dissipation for a complete cycle can be expressed as,

$$E_{load} = 2\{(RC_L)/T\}\ C_L(V_{DD})^2 + C_L(\Delta V)^2$$

3.2 CARRY LOOK AHEAD ADDER METHOD:

Carry look ahead logic uses the concepts of generating and propagating carries. Although in the context of a carry look ahead adder, it is most natural to think of generating and propagating in the context of binary addition, the concepts can be used more generally than this. In the descriptions below, the word digit can be replaced by bit when referring to binary addition.

The addition of two 1-digit inputs *A* and *B* is said to generate if the addition will always carry, regardless of whether there is an input carry (equivalently, regardless of whether any less significant digits in the sum carry). For example, in the decimal addition 52 + 67, the addition of the tens digits 5 and 6 generates because the result carries to the hundreds digit regardless of whether the ones digit carries (in the example, the ones digit does not carry (2+7=9)).

A carry-look ahead adder (CLA) is a type of adder used in digital logic. A carry-look ahead adder improves speed by reducing the amount of time required to determine carry bits. It can be contrasted with the simpler, but usually slower, ripple carry adder for which the carry bit is calculated alongside the sum bit, and each bit must wait until the previous carry has been

calculated to begin calculating its own result and carry bits (see adder for detail on ripple carry adders). The carry-look ahead adder calculates one or more carry bits before the sum, which reduces the wait time to calculate the result of the larger value bits. The Kogge-Stone adder and Brent-Kung adder are examples of this type of adder.

A ripple-carry adder works in the same way as pencil-and-paper methods of addition. Starting at the rightmost (least significant) digit position, the two corresponding digits are added and a result obtained. It is also possible that there may be a carry out of this digit position (for example, in pencil-and-paper methods, "9+5=4, carry 1"). Accordingly all digit positions other than the rightmost need to take into account the possibility of having to add an extra 1, from a carry that has come in from the next position to the right.

This means that no digit position can have an absolutely final value until it has been established whether or not a carry is coming in from the right. Moreover, if the sum without a carry is 9 (in pencil-and-paper methods) or 1 (in binary arithmetic), it is not even possible to tell whether or not a given digit position is going to pass on a carry to the position on its left. At worst, when a whole sequence of sums comes to ...99999999... (in decimal) or ...11111111... (in binary), nothing can be deduced at all until the value of the carry coming in from the right is known, and that carry is then propagated to the left, one step at a time, as each digit position evaluated "9+1=0, carry 1" or "1+1=0, carry 1". It is the "rippling" of the carry from right to left that gives a ripple-carry adder its name, and its slowness. When adding 32-bit integers, for instance, allowance has to be made for the possibility that a carry could have to ripple through every one of the 32 one-bit adders.

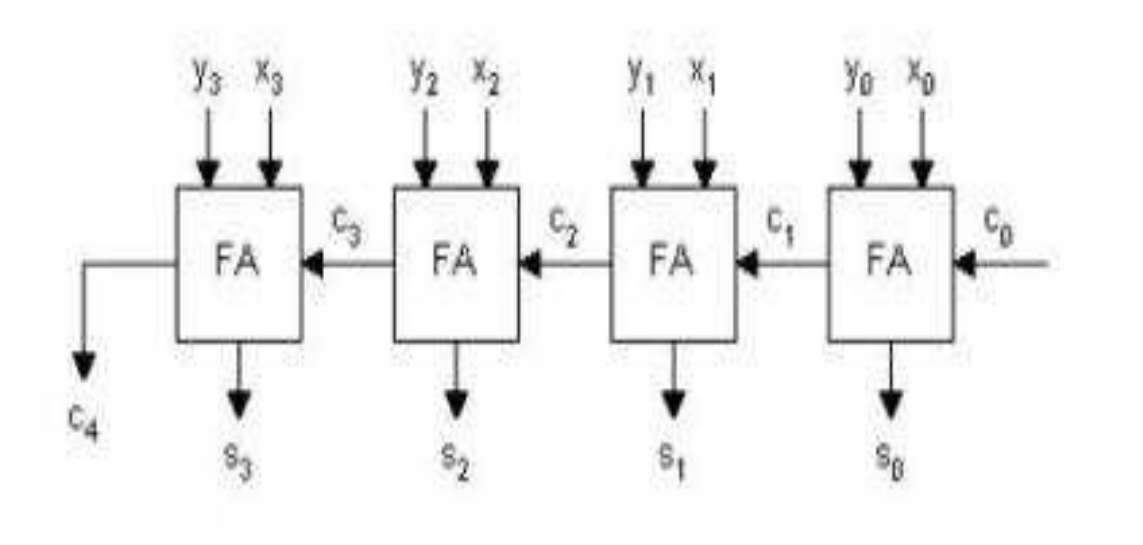


Fig 3.2. Cary look ahead adder Block Diagram

Carry look ahead depends on two things:

1. Calculating, for each digit position, whether that position is going to propagate a carry if one comes in from the right.
2. Combining these calculated values to be able to deduce quickly whether, for each group of digits, that group is going to propagate a carry that comes in from the right.

Supposing that groups of 4 digits are chosen. Then the sequence of events goes something like this:

1. All 1-bit adders calculate their results. Simultaneously, the look ahead units perform their calculations.
2. Suppose that a carry arises in a particular group. Within at most 5 gate delays, that carry will emerge at the left-hand end of the group and start propagating through the group to its left.
3. If that carry is going to propagate all the way through the next group, the look ahead unit will already have deduced this. Accordingly, before the carry emerges from the next group the look ahead unit is immediately (within 1 gate delay) able to tell the next group to the left that it is going to receive a carry - and, at the same time, to tell the next look ahead unit to the left that a carry is on its way.

The net effect is that the carries start by propagating slowly through each 4-bit group, just as in a ripple-carry system, but then move 4 times as fast, leaping from one look ahead carry unit to the next. Finally, within each group that receives a carry, the carry propagates slowly within the digits in that group.

The more bits in a group, the more complex the look ahead carry logic becomes, and the more time is spent on the "slow roads" in each group rather than on the "fast road" between the groups (provided by the look ahead carry logic). On the other hand, the fewer bits there are in a group, the more groups have to be traversed to get from one end of a number to the other, and the less acceleration is obtained as a result. Deciding the group size to be governed by look ahead carry logic requires a detailed analysis of gate and propagation delays for the particular technology being used.

It is possible to have more than one level of look ahead carry logic, and this is in fact usually done. Each look ahead carry unit already produces a signal saying "if a carry comes in from the right, I will propagate it to the left", and those signals can be combined so that each group of (let us say) four look ahead carry units becomes part of a "super group" governing a total of 16 bits of the numbers being added. The "super group" look ahead carry logic will be able to say whether a carry entering the super group will be propagated all the way through it, and using this information, it is able to propagate carries from right to left 16 times as fast as a naive ripple carry. With this kind of two-level implementation, a carry may first propagate through the "slow road" of individual adders, then, on reaching the left-hand end of its group, propagate through the "fast road" of 4-bit look ahead carry logic, then, on reaching the left-hand end of its super group, propagate through the "superfast road" of 16-bit look ahead carry logic.

CHAPTER-4

FULL ADDER AND CARRY LOOK A HEAD ADDER USING ADIABATIC POWER GATING

4.1 ADIABATIC POWERGATING FA SCHEMATIC:

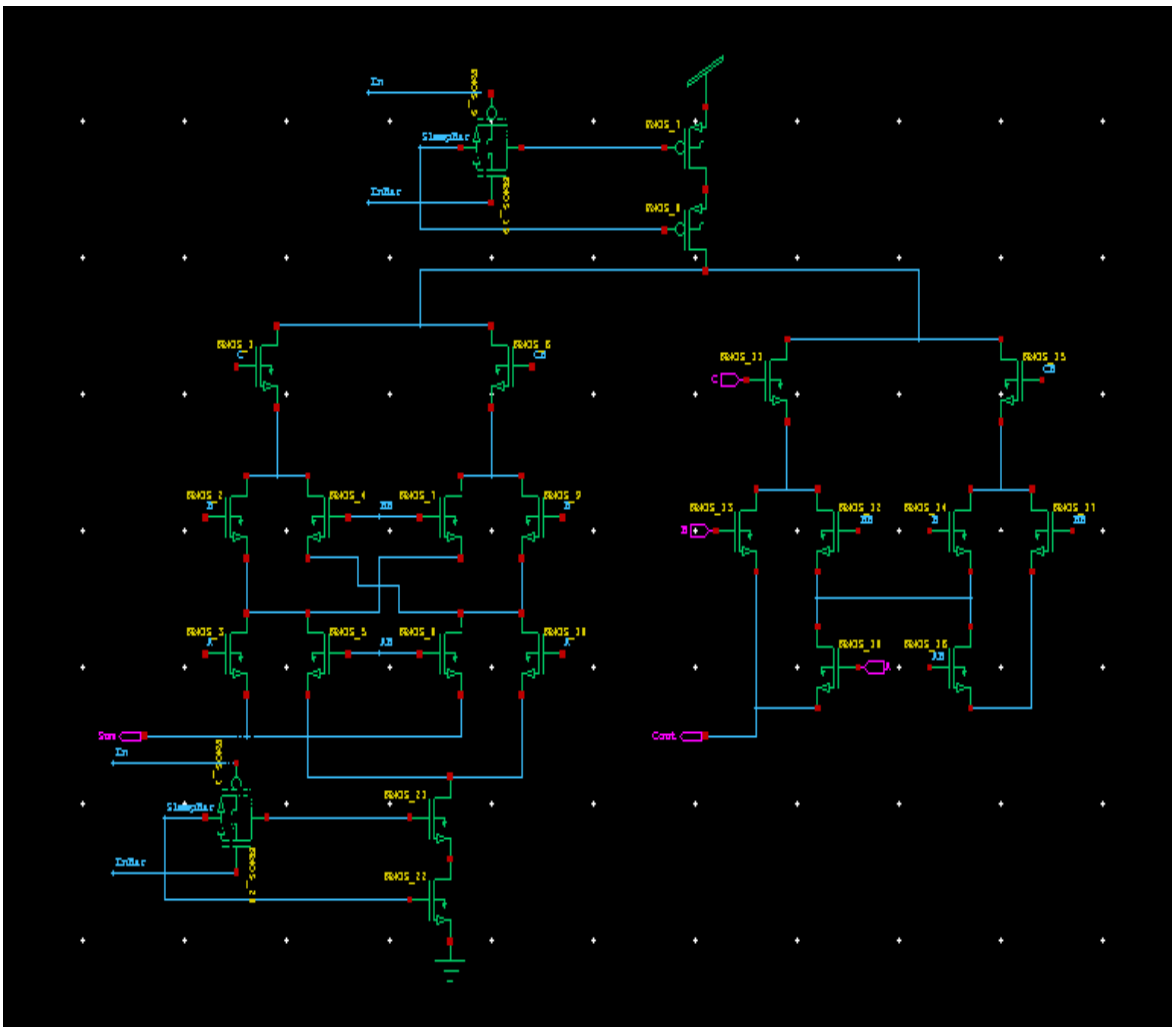


fig 4.1 Adiabatic Power gating Full adder

4.2 CLA SCHEMATIC IN TANNER:

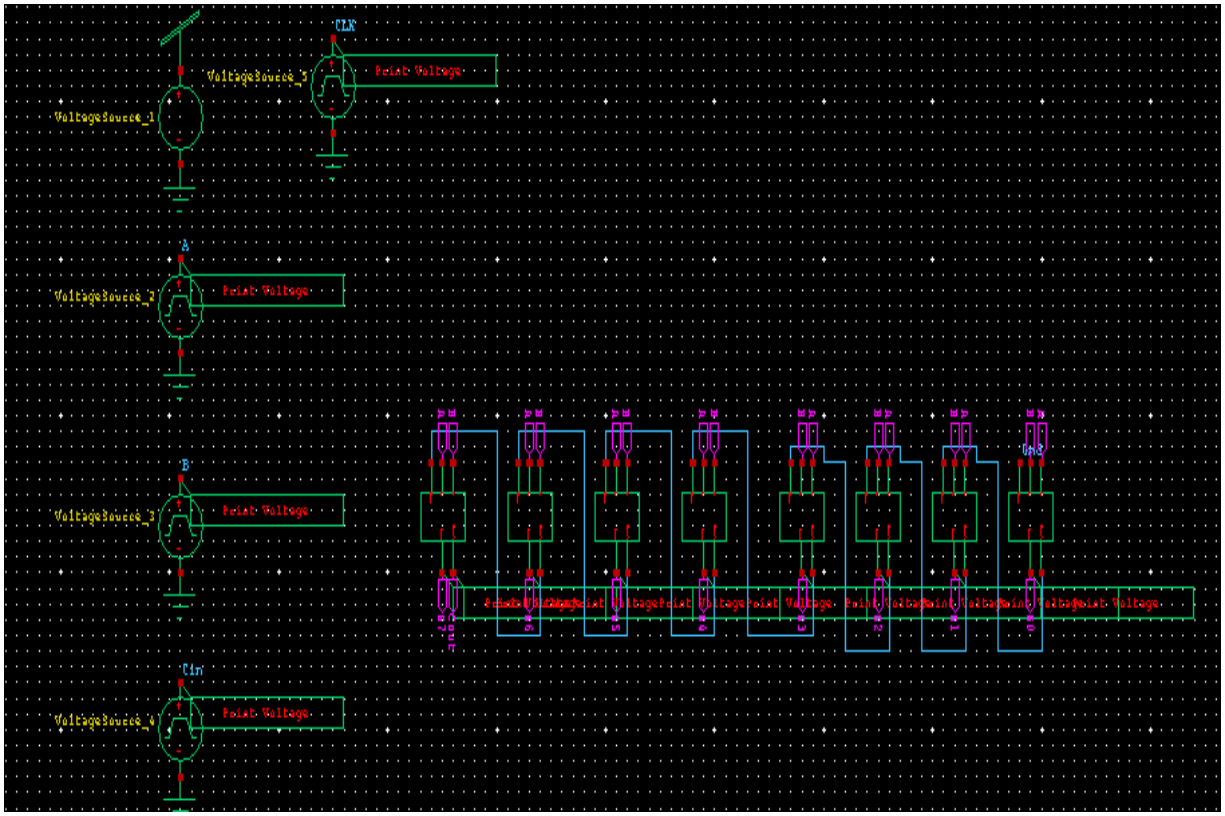


fig 4.2.1 carry look ahead adder schematic

WAVEFORM:

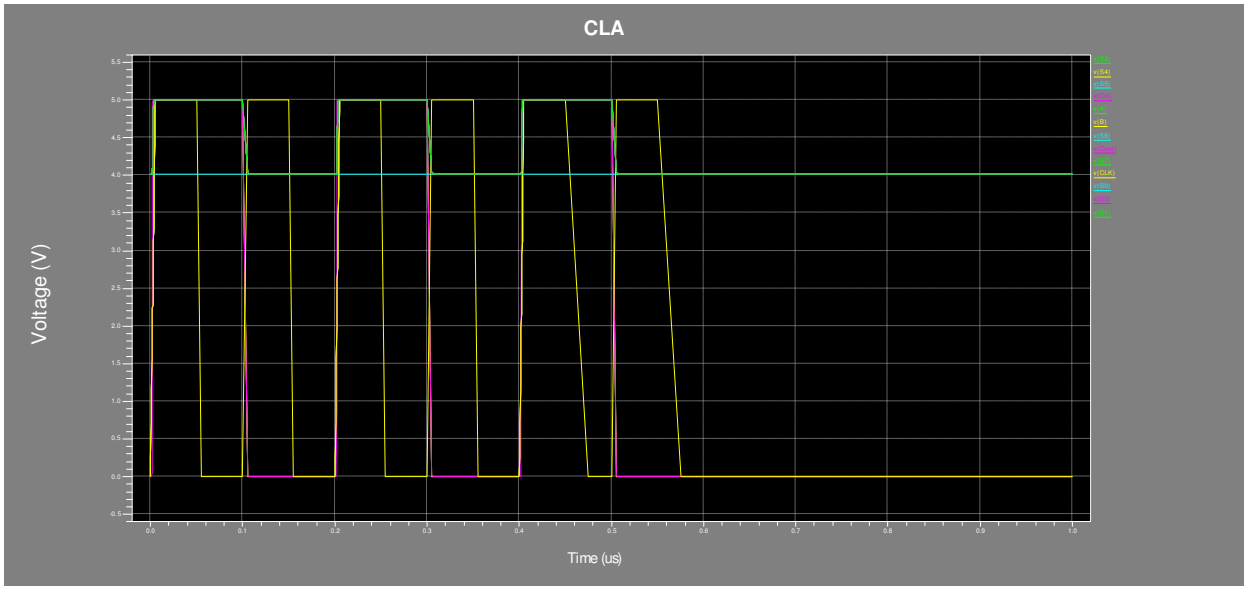


fig 4.2.2 carry look ahead adder simulation

4.3 ADIABATIC CLA SCHEMATIC:

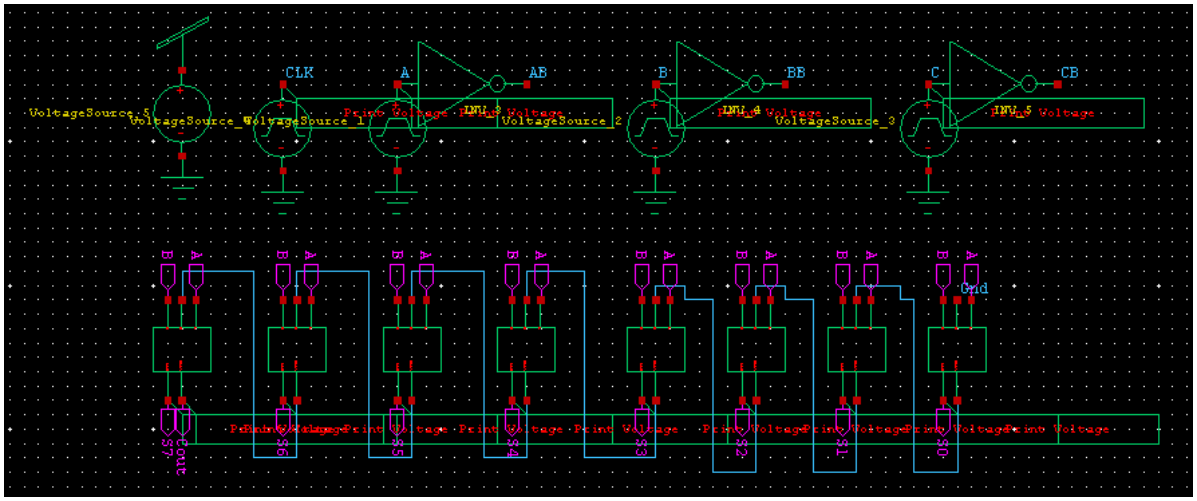


fig 4.3.1 Adiabatic CLA Schematic

4.3.2 WAVEFORM:

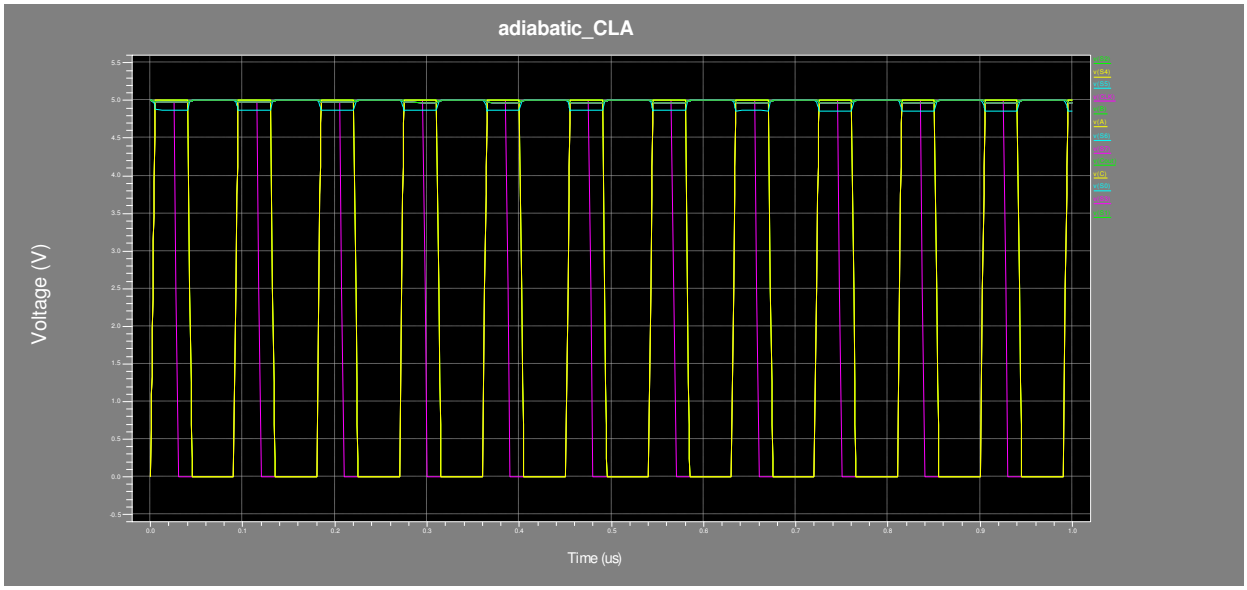


fig 4.3.2 Adiabatic CLA Simulation

4.4 POWERGATING CLA SIMULATION:

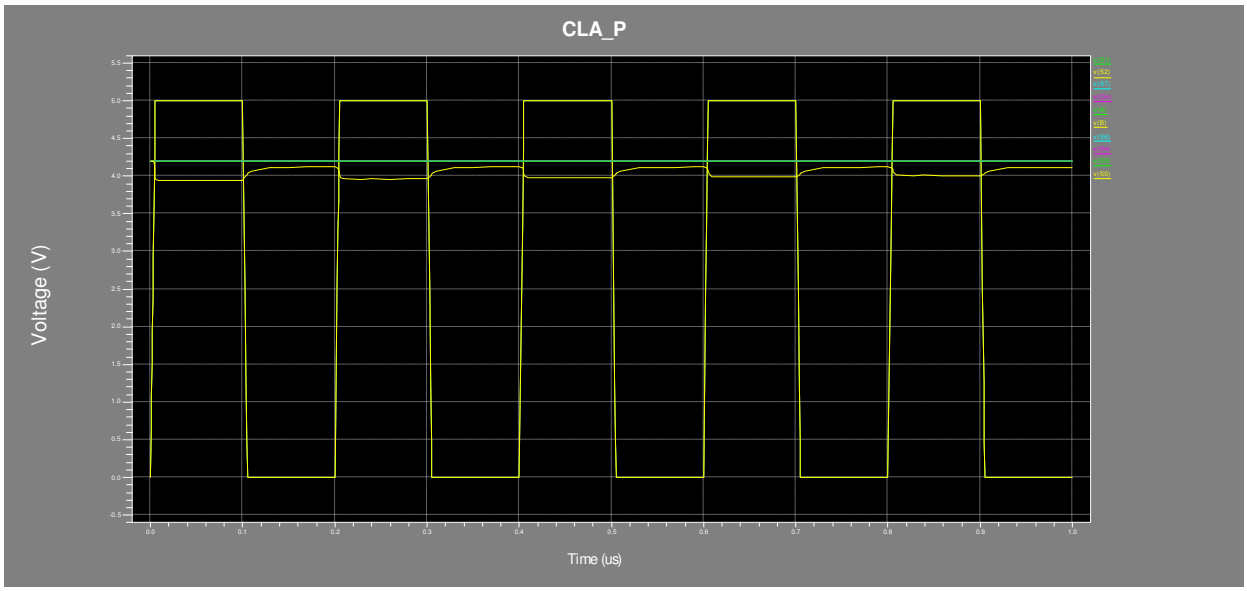


fig4.4.1 Power Gating CLA waveform

4.5 ADIABATIC CLA POWERGATING SIMULATION:

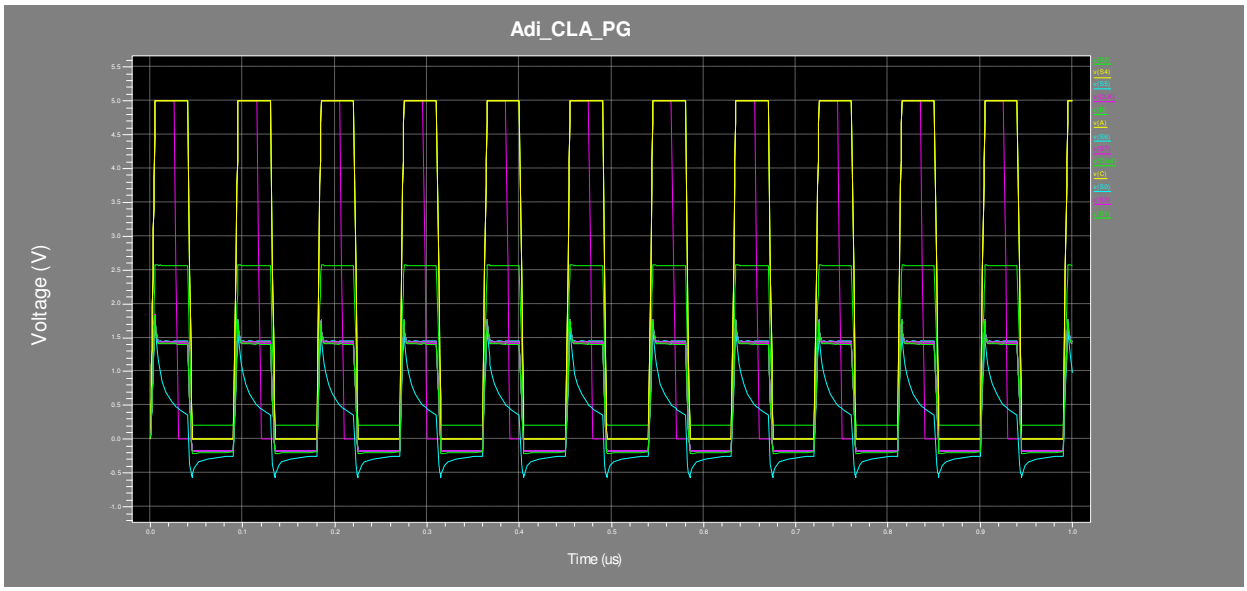


fig 4.5.1 Adiabatic CLA power gating waveform

CHAPTER-5

SOFTWARE DESCRIPTION

5.1 TANNER EDA:

Tanner EDA tools for analog and mixed-signal ICs and MEMS design offers designers a seamless, efficient path from design capture through verification. Our powerful, robust tool suite is ideal for applications including Power Management, Life Sciences / Biomedical, Displays, Image Sensors, Automotive, Aerospace, RF, Photovoltaics, Consumer Electronics and MEMS.

5.2 TANNER EDA INCLUDES:

- S-EDIT (SCHEMATIC EDIT)
- W-EDIT(WAVEFORM EDIT)
- T-SPICE

5.2.1 S-EDIT:

- S-Edit brings to front-end design capture the ease-of-use and design productivity for which Tanner Tools™ are known
- A fully user-programmable design environment allows you to remap hotkeys, create new toolbars, and customize the view to your preference; all in a streamlined GUI
- The complete user interface is available in multiple languages. S-Edit currently supports English, Japanese, Simplified and Traditional Chinese.
- S-Edit provides Unicode support. All user data can be entered in international character sets
- S-Edit imports schematics via Open Access or via EDIF from third-party tools, including Cadence, Mentor, Laker and View Draw with automatic conversion of schematics and properties for seamless integration of legacy data.

- Netlists can be exported in flexible, user-configurable formats, including SPICE and CDL variants, EDIF, structural Verilog, and structural VHDL.
- Library support in S-Edit maximizes the reuse of IP developed in previous projects, or imported from third-party vendors.
- S-Edit provides an ideal performance-to-cost ratio, allowing you to maximize the number of designers on a project.
- Since S-Edit runs on Windows® and Linux® platforms, designers can work on cost-effective workstations or laptops. This means you can take your work with you anywhere—even home—and continue working to meet time-to-market pressures.
- Available in two configurations—full schematic editor, and schematic viewer.

5.2.2 W-EDIT:

Visualizing the complex numerical data resulting from VLSI circuit simulation is critical to testing, understanding, and improving those circuits. W-Edit is a waveform viewer that provides ease of use, power, and speed in a flexible environment designed for graphical data presentation. The advantages of W-Edit include:

- Tight integration with T-Spice, Tanner EDA's circuit-level simulator. W-Edit can chart data generated by T-Spice directly, without modification of the output data files. W-Edit charts data dynamically as it is produced during the simulation.
- *f*Charts are automatically configured for the type of data being presented.
- *f*A data set is treated by W-Edit as a unit called a trace. Multiple traces from different output files can be viewed simultaneously, in single or multiple windows. You can copy and move traces between charts and windows.
- You can perform trace arithmetic or spectral analysis on existing traces to create new ones.
- *f* You can pan back and forth and zoom in and out of chart views, including specifying the exact x-y coordinate range W-Edit displays.
- You can measure positions and distances between points easily and precisely with the mouse.

- *f* You can customize properties of axes, traces, grids, charts, text, and colors.

Numerical data is input to W-Edit in the form of plain or binary text files. Header and comment information supplied by T-Spice is used for automatic chart configuration. Runtime update of results is made possible by linking W-Edit to a running simulation in T-Spice.

W-Edit saves information on chart, trace, axis, and environment settings in files with the .wdb (W-Edit Database) extension. You can load a .wdb file in conjunction with a data file to automatically apply saved formats and environment settings. The .wdb file does not contain any trace data.

You can launch W-Edit from the Windows Start menu or by using one of the W-Edit shortcuts included in T-Spice Pro applications. To launch W-Edit from the Start menu, click the Start button on the Windows taskbar and navigate to Programs > Tanner EDA > T-Spice Pro vx.y > W-Edit, where x.y is the version number of T-Spice Pro.

The display area consists of the entire W-Edit interface not occupied by the title bar, menu bar, toolbar, or status bar. W-Edit has a multiple document interface (MDI): you can open, view, edit, resize, minimize, or rearrange multiple windows in the display area. The actual area available to contain windows is much larger than the area viewable at any one time; vertical and horizontal scrollbars appear along the sides of the display area if any windows extend beyond the current viewable area.

The largest discrete unit in the display area is the window. You can have multiple windows open but only one window can be active at any given time. The title bar of the active window is highlighted. To make a window active, either click in the desired window or select it from the list of open files in the Window menu. You can scroll in windows, and they can be minimized, zoomed (maximized), resized, moved, tiled, and layered (cascaded).

Each window in W-Edit consists of one or more charts. A chart contains a rectangular x- and y-axis system, with two-dimensional data in the form of individual continuous curves called traces. Auxiliary items—grid lines, tick marks, and explanatory labels—are also included in charts.

W-Edit uses built-in ratios to determine the size of the axis system and the positions of the title, labels, and legends in a chart. When you change the chart size, W-Edit adjusts the size and position of chart elements accordingly.

Traces are formed from columns of numbers in the data file. Generally, there is one x-column and one corresponding y-column to make up the (x,y) pair of coordinates that form a trace. For information about data columns, Traces that represent individual sweeps from a .step or .alter command are grouped into trace families.

Each trace family contains all parameter sweeps or .alter simulations for a dependent variable. Traces in a family can be edited individually, but all trace family members share the same label and a single entry in the chart legend.

A W-Edit input *data file is the* file that contains numerical simulation data. The data file also contains supplementary information about units, labels, analysis types, and parameter values that W-Edit uses to construct a visible representation of the data. Data files are generated by T-Spice, or you can use a text editor or another simulation tool to create them.
W-Edit reads two types of data files: **.out** files, which contain the numeric information as text; and **.dat** files, which contain the numeric information in binary format. In general, **.out** files are created from T-Spice simulations.

5.2.3 T-SPICE:

Tanner T-Spice™ Circuit Simulator puts you in control of simulation jobs with an easy-to-use graphical interface and a faster, more intuitive design environment. With key features such as multi-threading support, device state plotting, real-time waveform viewing and analysis, and a command wizard for simpler SPICE syntax creation, T-Spice saves you time and money during the simulation phase of your design flow.

T-Spice enables more accurate simulations by supporting the latest transistor models—including BSIM4 and the Penn State Philips (PSP) model. Given that T-Spice is compatible with

a wide range of design solutions and runs on Windows® and Linux® platforms, it fits easily and cost effectively into your current tool flow.

T-Spice provides extensive support of behavioral models using Verilog-A, expression controlled sources, and table-mode simulation. Behavioral models give you the flexibility to create customized models of virtually any device. T-Spice also supports the latest industry models, including the transistor model recently selected as the next standard for simulating future CMOS transistors manufactured at 65 nanometers and below—the Penn State Philips (PSP) model. PSP will simplify the exchange of chip design information and support more accurate digital, analog, and mixed-signal circuit behavior analysis.

T-Spice also supports foundry extensions, including HSPICE® foundry extensions to models.

• Supports PSP, BSIM3.3, BSIM4.6, BSIM SOI 4.0, EKV 2.6, MOS 9, 11, 20, 30, 31, 40, PSP, RPI a-Si & Poly-Si TFT, VBIC, Modella, and MEXTRAM models.

• Includes two stress effect models, from the Berkeley BSIM4 model and from TSMC processes, in the BSIM3 model to provide more accuracy in smaller geometry processes.

• Supports gate and body resistance networks in RF modeling.

• Performs non-quasi-static (NQS) modeling.

• Supports comprehensive geometry-based parasitic models for multi-finger devices.

• Models partially depleted, fully depleted, and unified FD-PD SOI devices.

• Models self-heating and RF resistor networks.

• Performs table-based modeling for using measured device data to model a device.

• Includes enhanced diode and temperature equations to improve compatibility with many foundry model libraries.

T-Spice helps integrate your design flow from schematic capture through simulation and waveform viewing. An easy-to-use point-and-click environment gives you complete control over the simulation process for greater efficiency and productivity.

• Enables easy creation of syntax-correct SPICE through a command wizard.

• Highlights SPICE Syntax through a text editor.

• Provides Fast, Accurate, and Precise options to enable optimal balance of accuracy and performance.

• Enables you to link from syntax errors to the SPICE deck by double clicking.

• Supports Verilog-A for analog behavioral modeling, allowing designers to prove system level designs before doing full device level design.

• Provides ".alter" command for easy what-if simulations with netlist changes.

T-Spice uses superior numerical techniques to achieve convergence for circuits that are often impossible to simulate with other SPICE programs. The types of circuit analysis it performs include:

• DC and AC analysis.

• Transient analysis with Gear or trapezoidal integration.

• Enhanced noise analysis.

• Monte Carlo analysis over unlimited variables and trials with device and lot variations.

• Virtual measurements with functions for timing, error, and statistical analysis including common measurements such as delay, rise time, frequency, period, pulse width, settling time, and slew rate.

• Parameter sweeping using linear, log, discrete value, or external file data sweeps.

• 64-bit engine for increased capacity and higher performance. With T-Spice, you can:

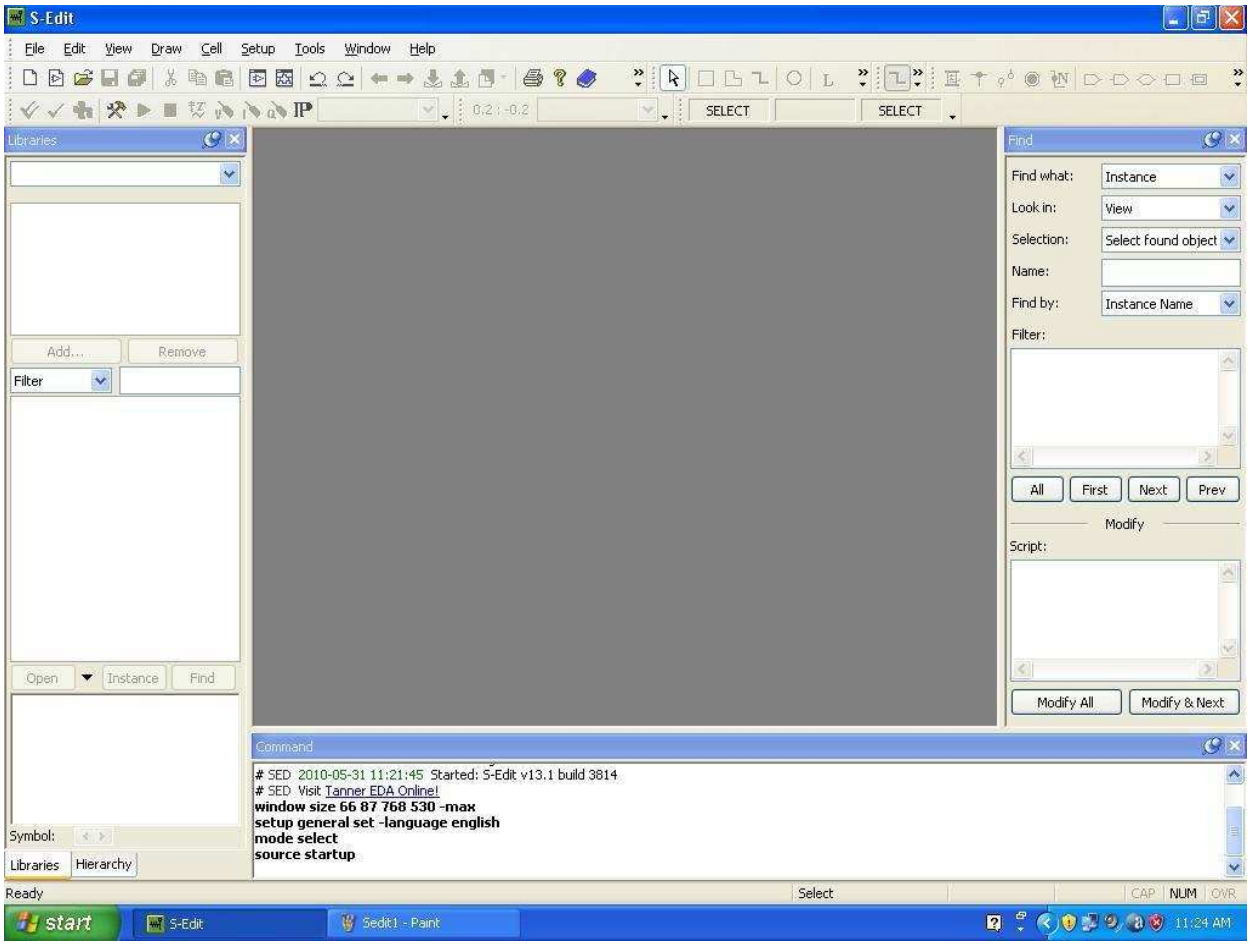
• Optimize designs with variables and multiple constraints by applying a Levenberg-Marquardt non-linear optimizer.

• Perform Safe Operating Area (SOA) checks to create robust designs.

• Use bit and bus logic waveform inputs.

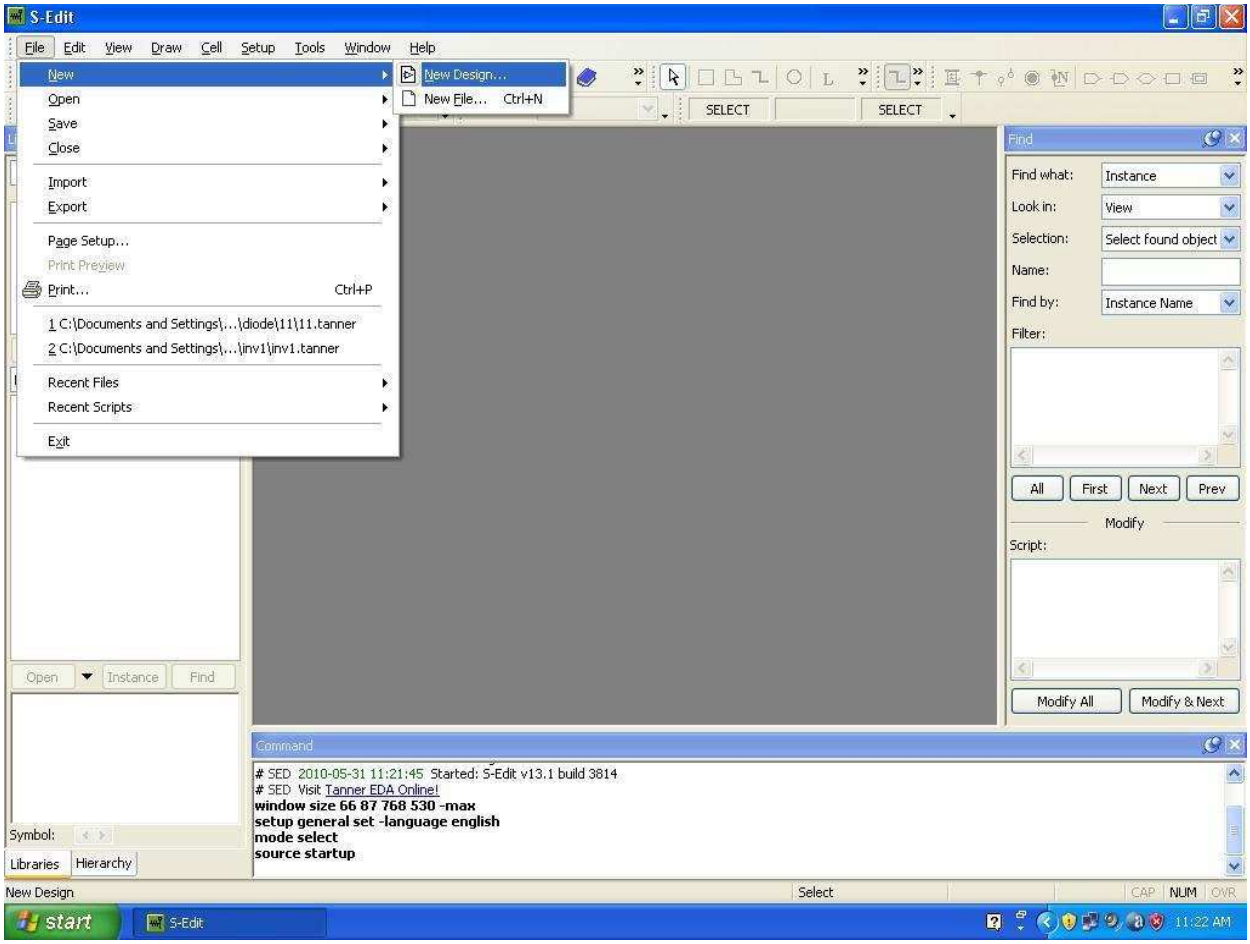
5.3 WORK FLOW OF TANNER:

A new window will open:



Go to >>file >> New >> New Design

Select New Design

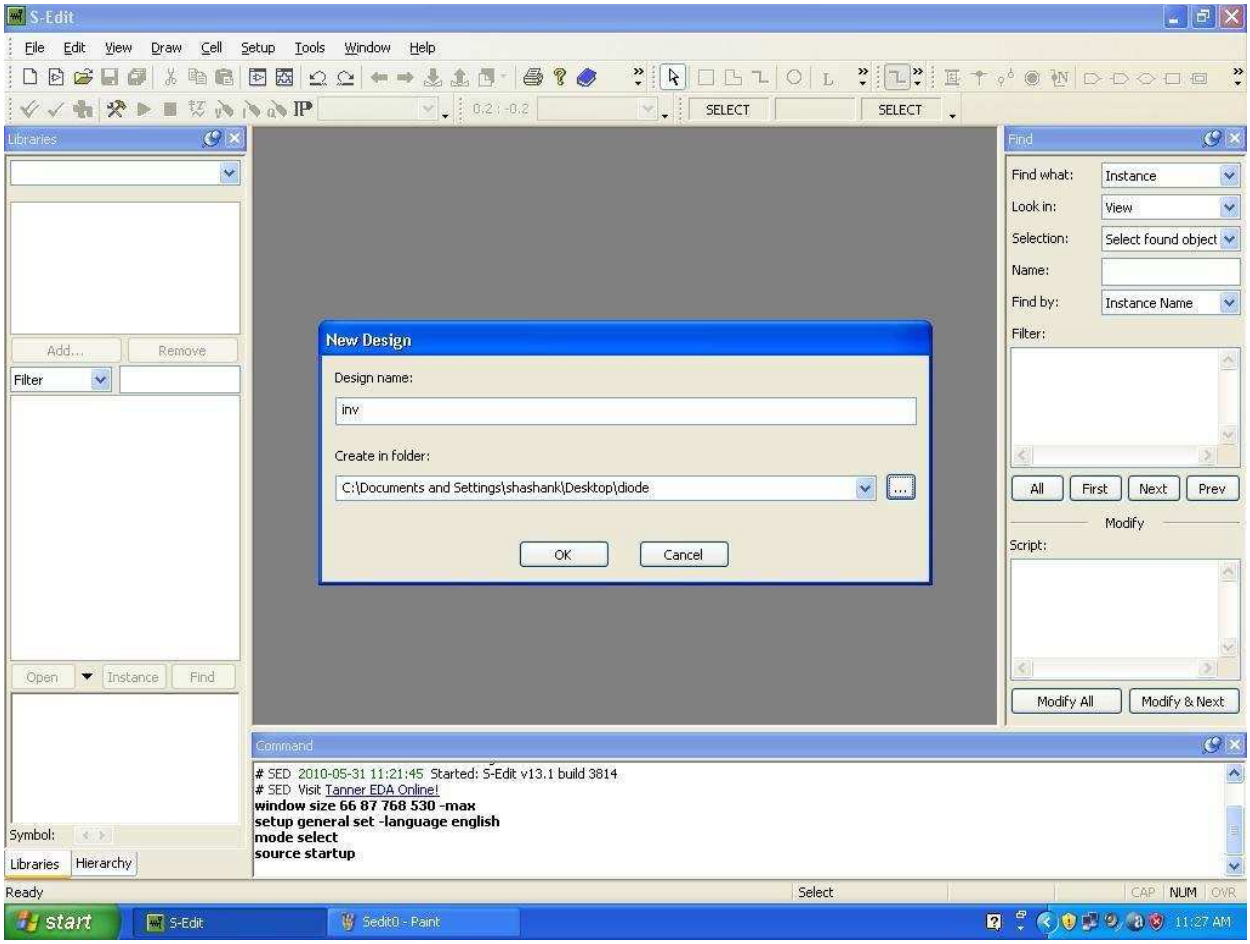


One dialog box will appear

Design Name : Give the name your design as you wish

Create a Folder : Give the path where you want to save the S-Edit Files.

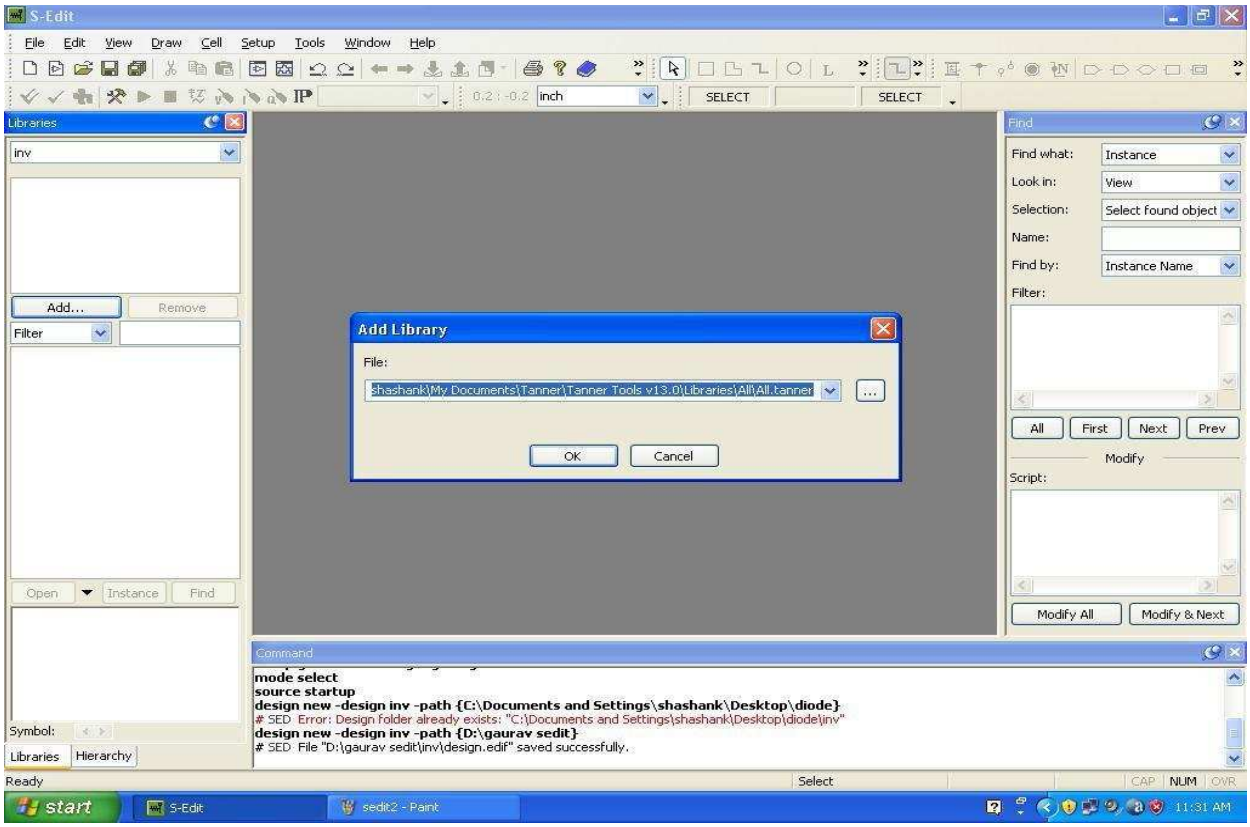
Then Click on ‘OK’



Now to add libraries in your work click on **Add ,** left on the library window.

Give the path where Libraries are stored . As for example

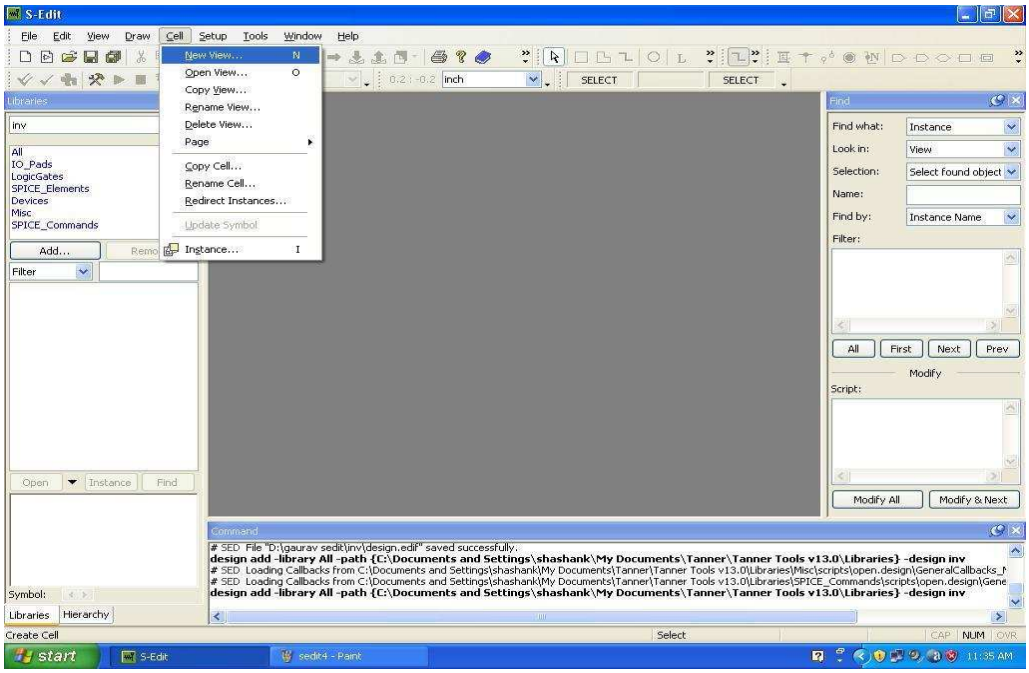
C:\Documents and Settings\Bhowmik.IIIT-3AC288AD0A\My Documents\Tanner EDA\Tanner Tools v13.0\Libraries\All\All.tanner



Now to create new cell

Go to cell menu >> New view --

Select ‘New view’



The new cell will appear like below:

Design = your design name

Cell = cell no. (cell no you can change but your design name **inv** will be same for different cell.

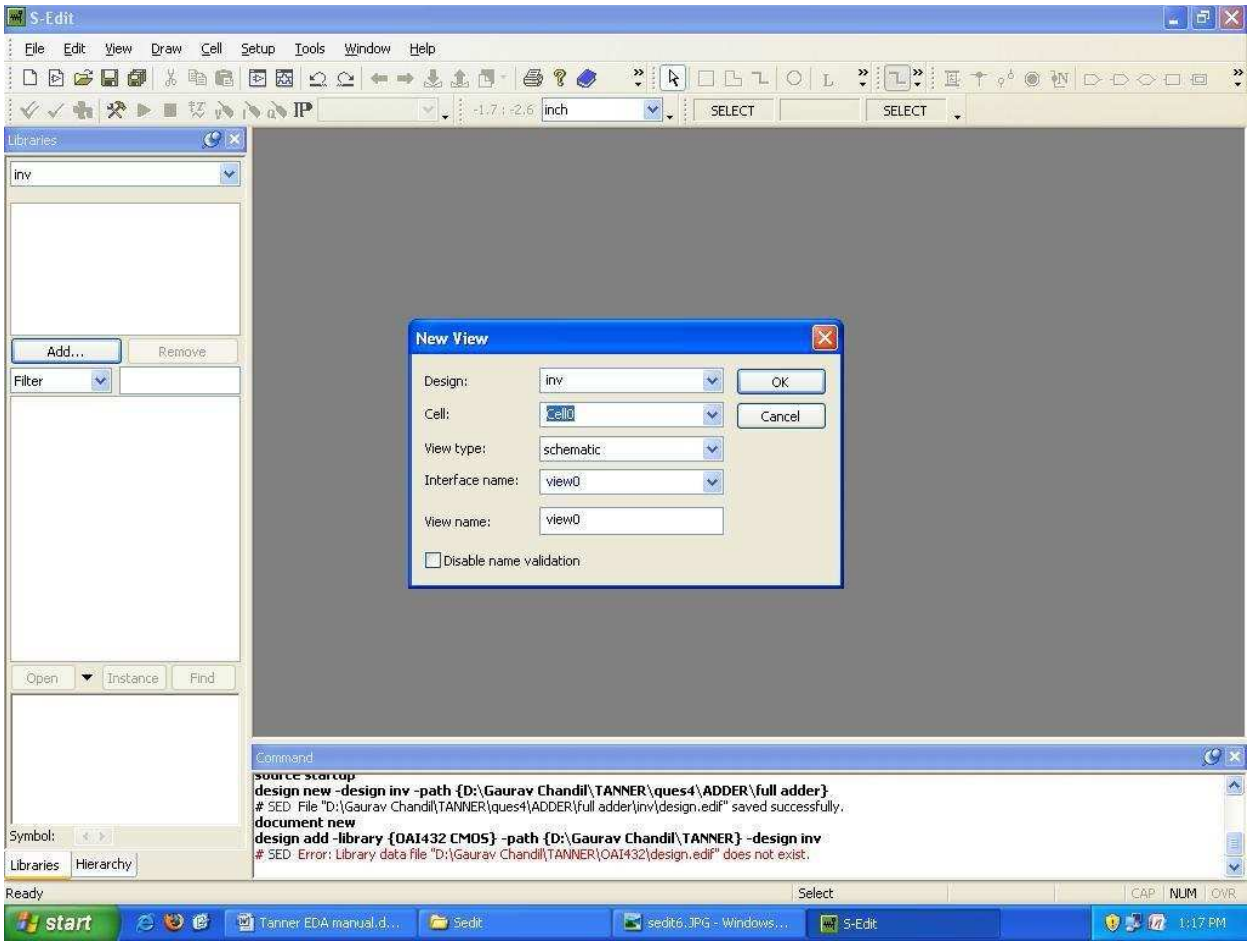
Design name should be changed only when you are going to design another circuit)

View type = schematic

Interface name = “by default”

View name = “by default”

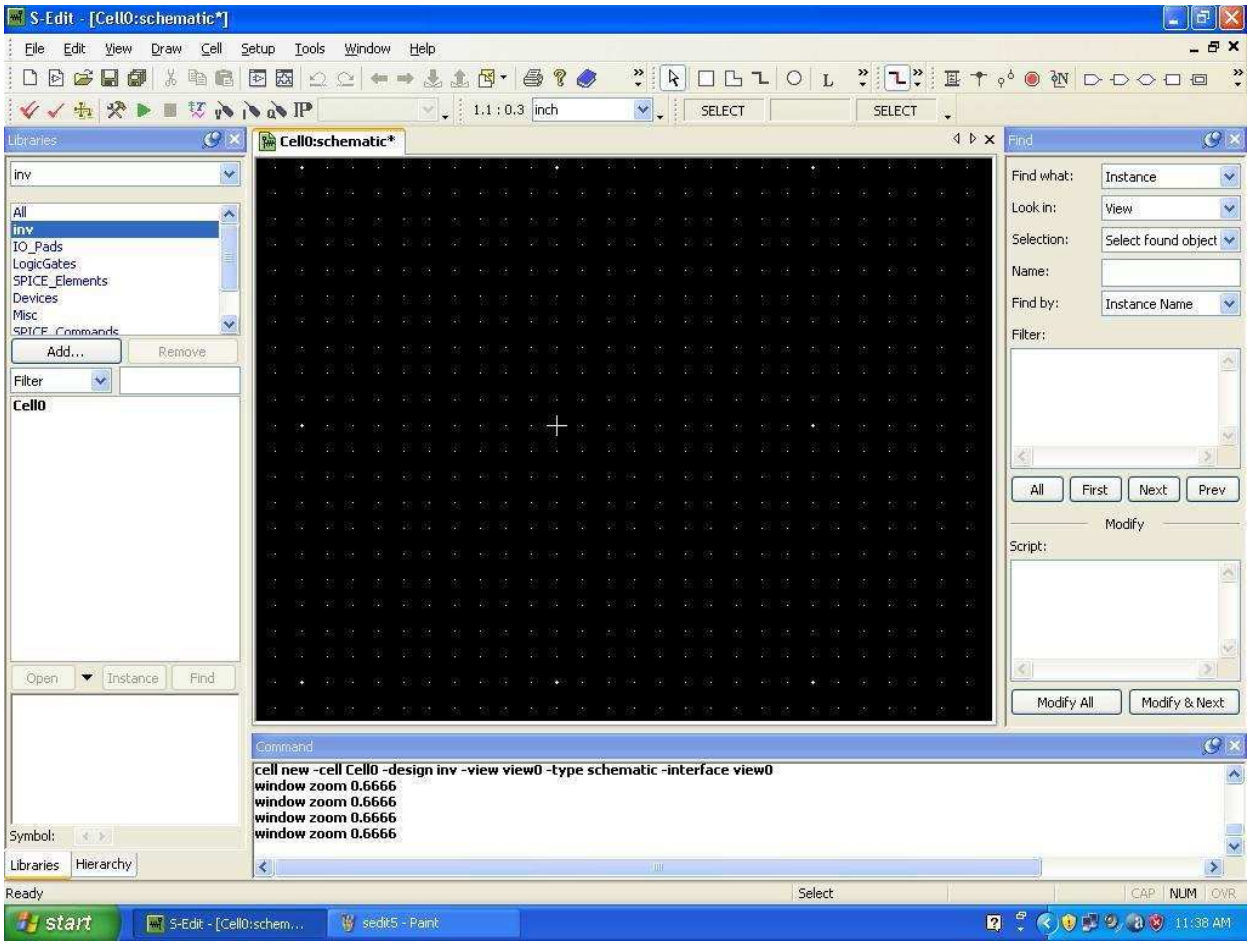
Then press “OK”.



Then a cell will be appeared where we can draw the schematic of any circuit.

In the black window you have seen some white bubble arranged in specific order. This is called grid. You can change grid distance by clicking on black screen and then scroll the mouse.

If you want your screen big enough for design space , then you can close the **Find** & **command window**. You can again bring these window from **view** menu bar.



To make any circuit schematic .

for example inverter

a) Go to >>libraries & click on device then all device will be open.

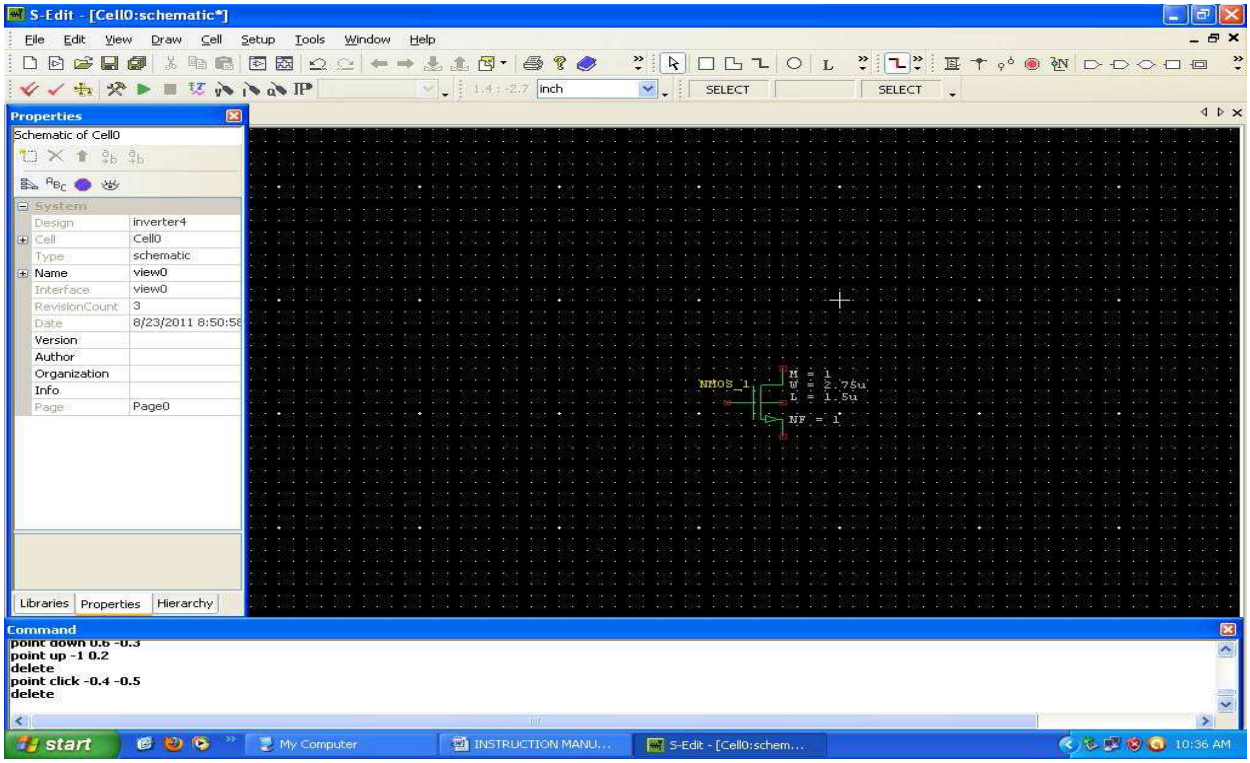
In instance cell

 You can change the values of various device parameters according to your requirements.

 Go to properties >> change the parameter values as your requirement.

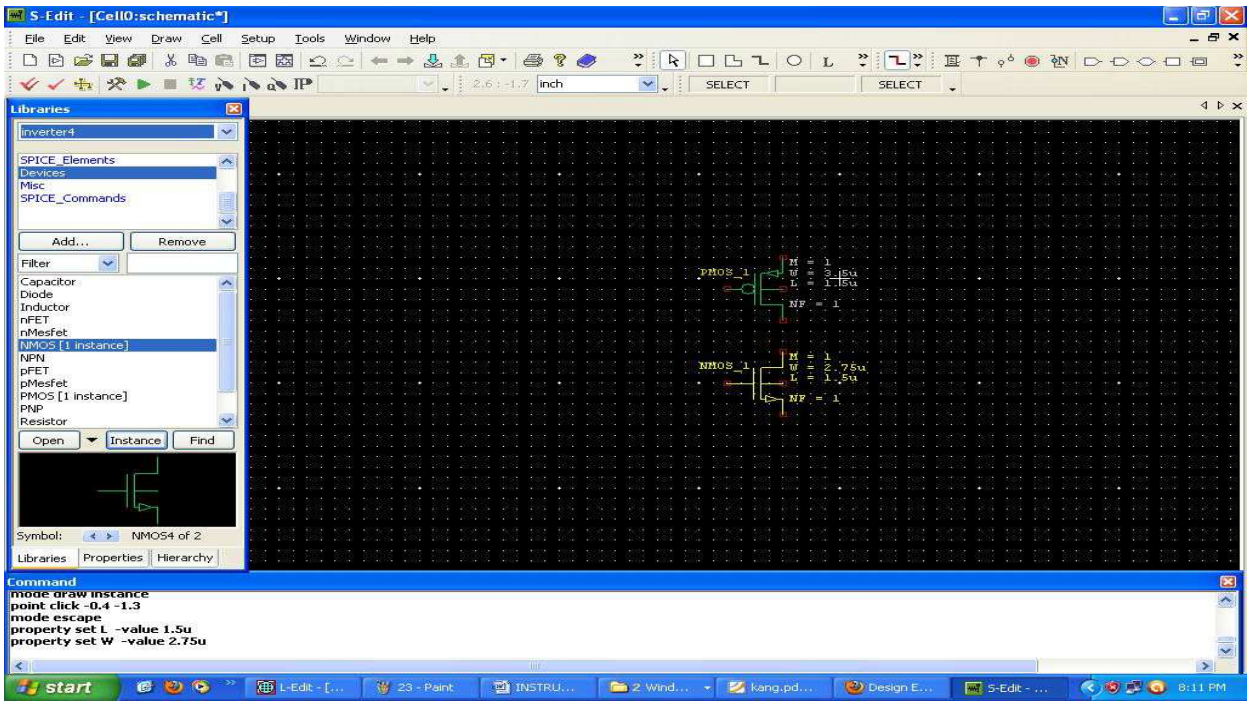
 Now before clicking **DONE** you have to DRAG the selected device into the cell and drop it where you want it to FIX .

Then click **DONE** or press **ESC.**



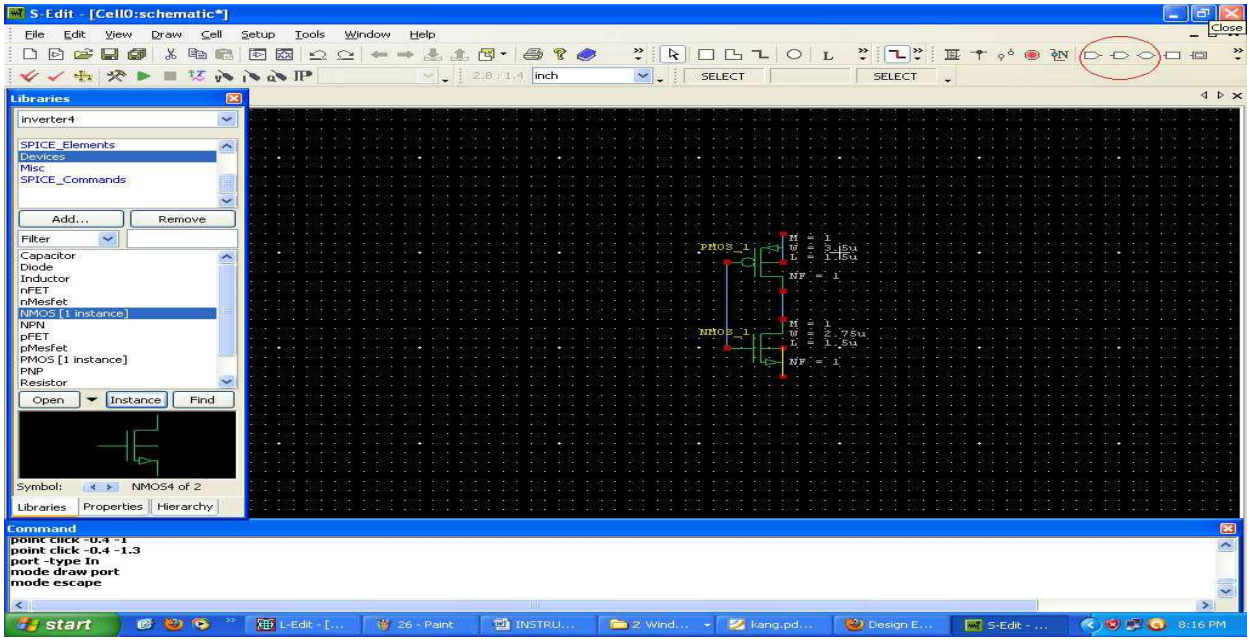
Similarly you can DRAG & DROP any device into the cell for draw your schematic circuit.

For inverter we need another Pmos.



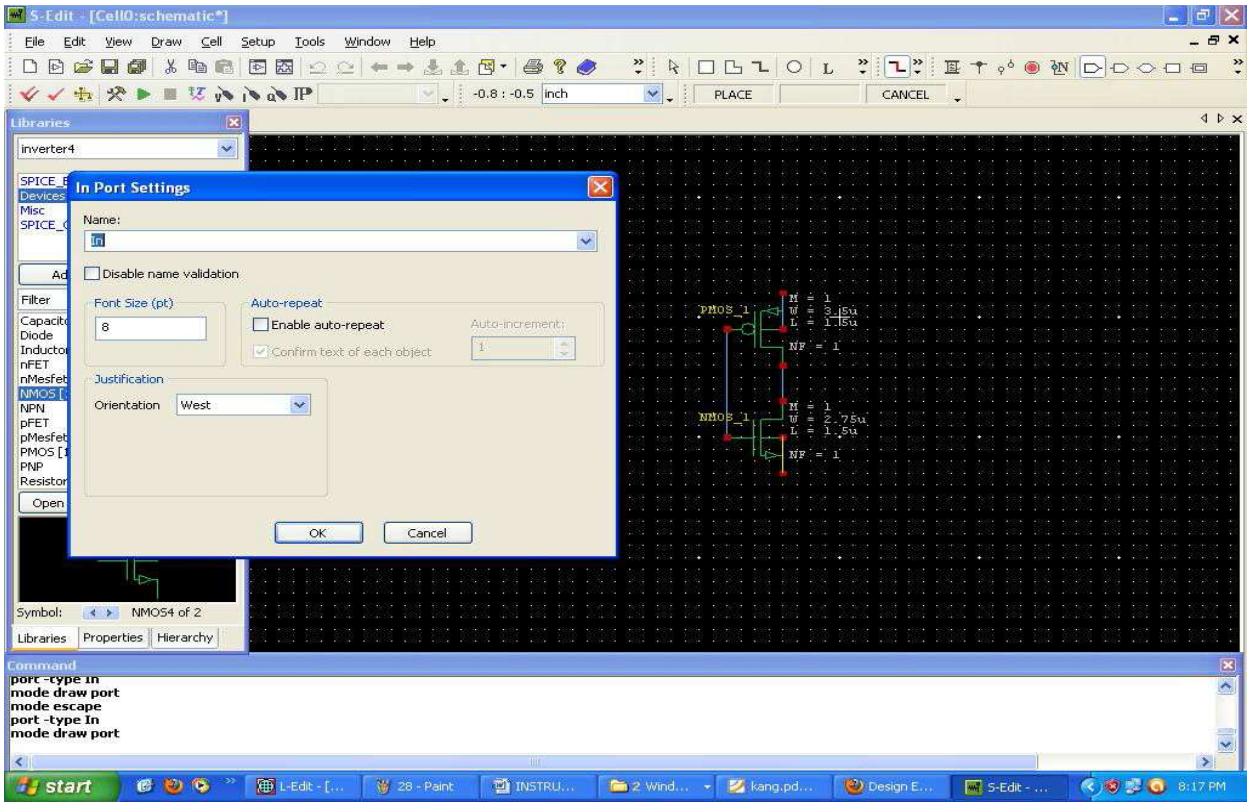
Now connect two device with wire.

Go to tool bar and select wire



Now you can give Port name as you wish in the dailog box.

Then click OK

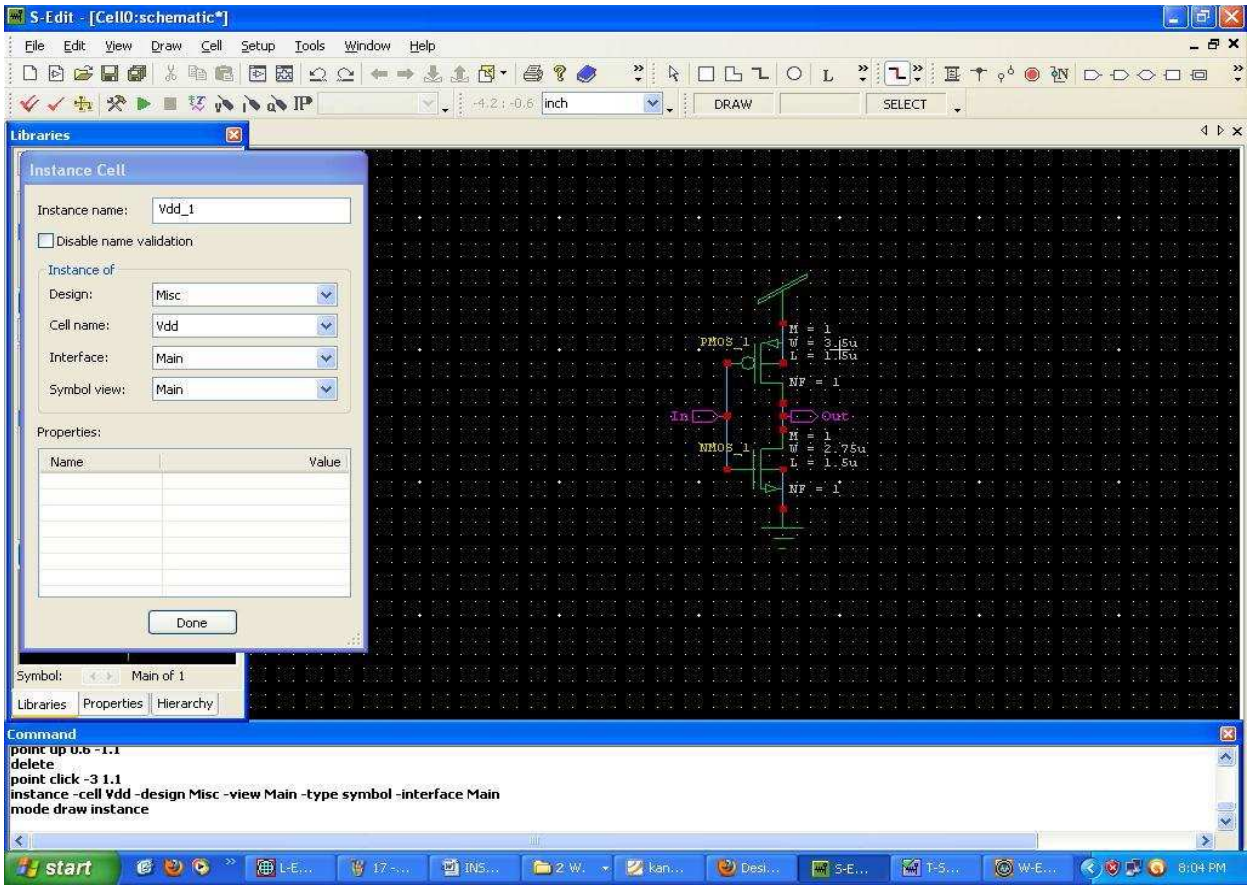


Similarly give Output Port name.

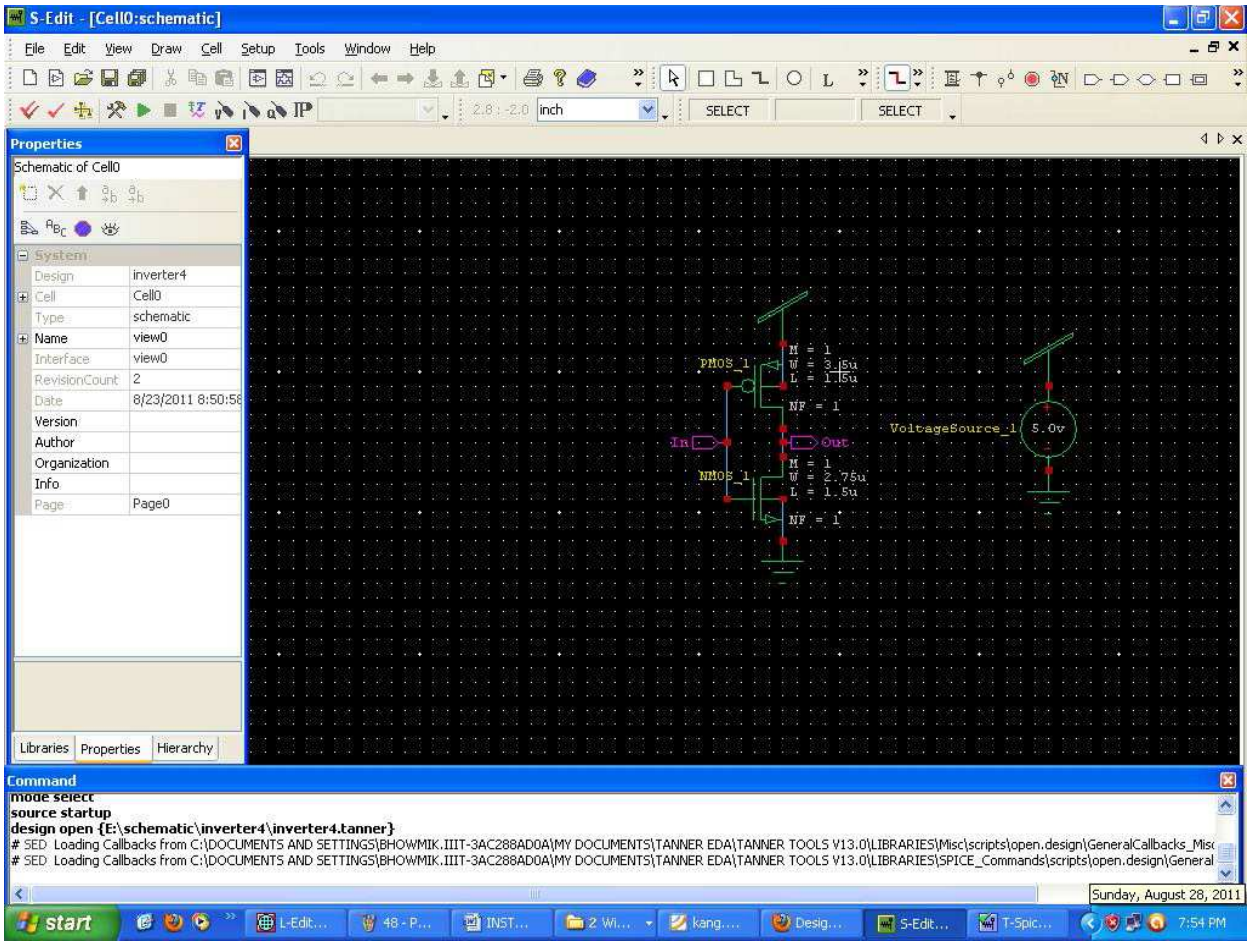
NOTE : you can rotate the port (short cut key “R”).

Now, after completed these steps, you should give the supply (VDD) & ground (GND).

For that Go to liberaries >> MISC >>Select VDD or GND



Now you have to create a source of VDD. For that go to libraries >>spice_element >> and then select voltage source of type DC . you can give any value in vdd .lets take vdd =5v.

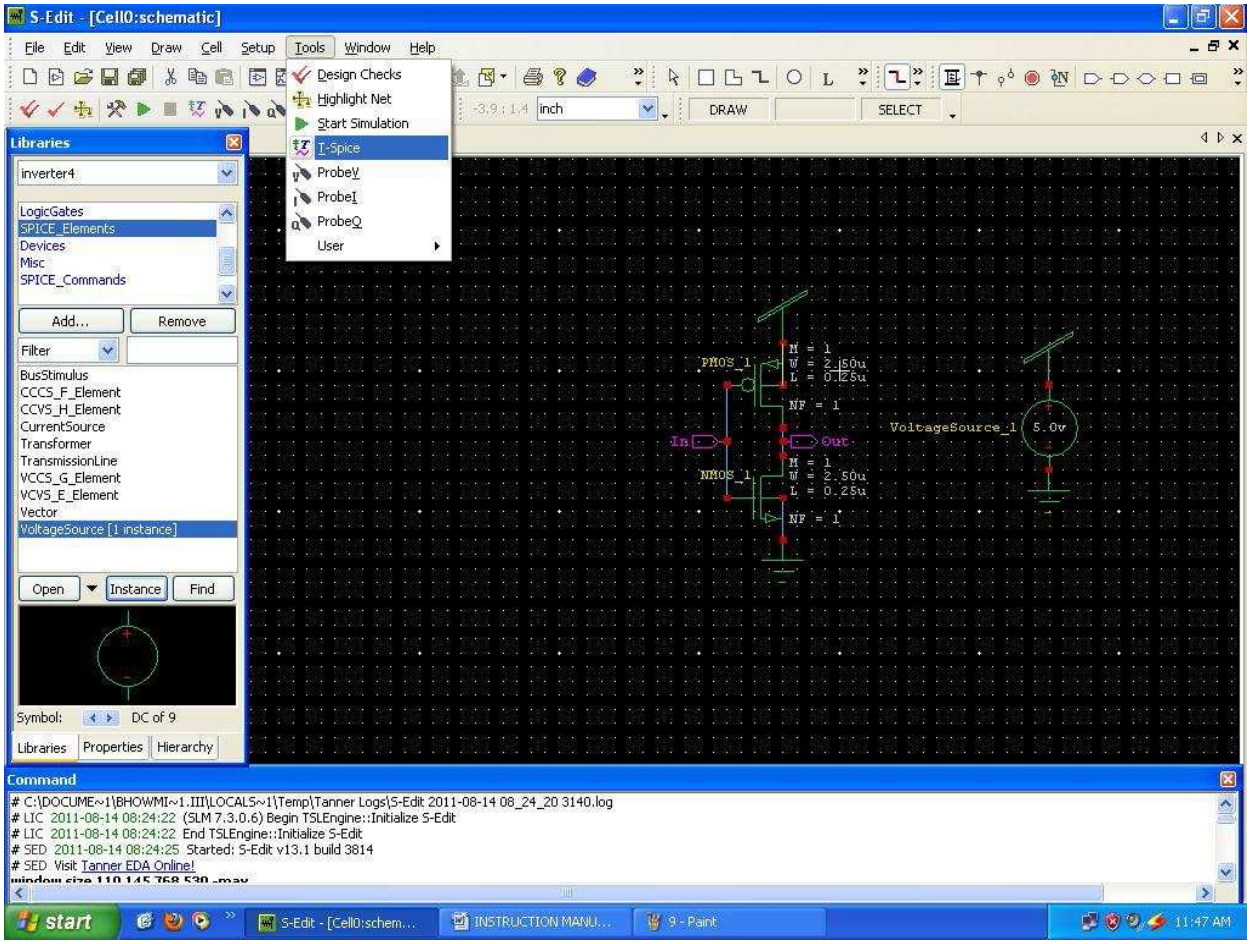


By doing all the above steps you have completed schematic of Inverter

Pre layout simulation

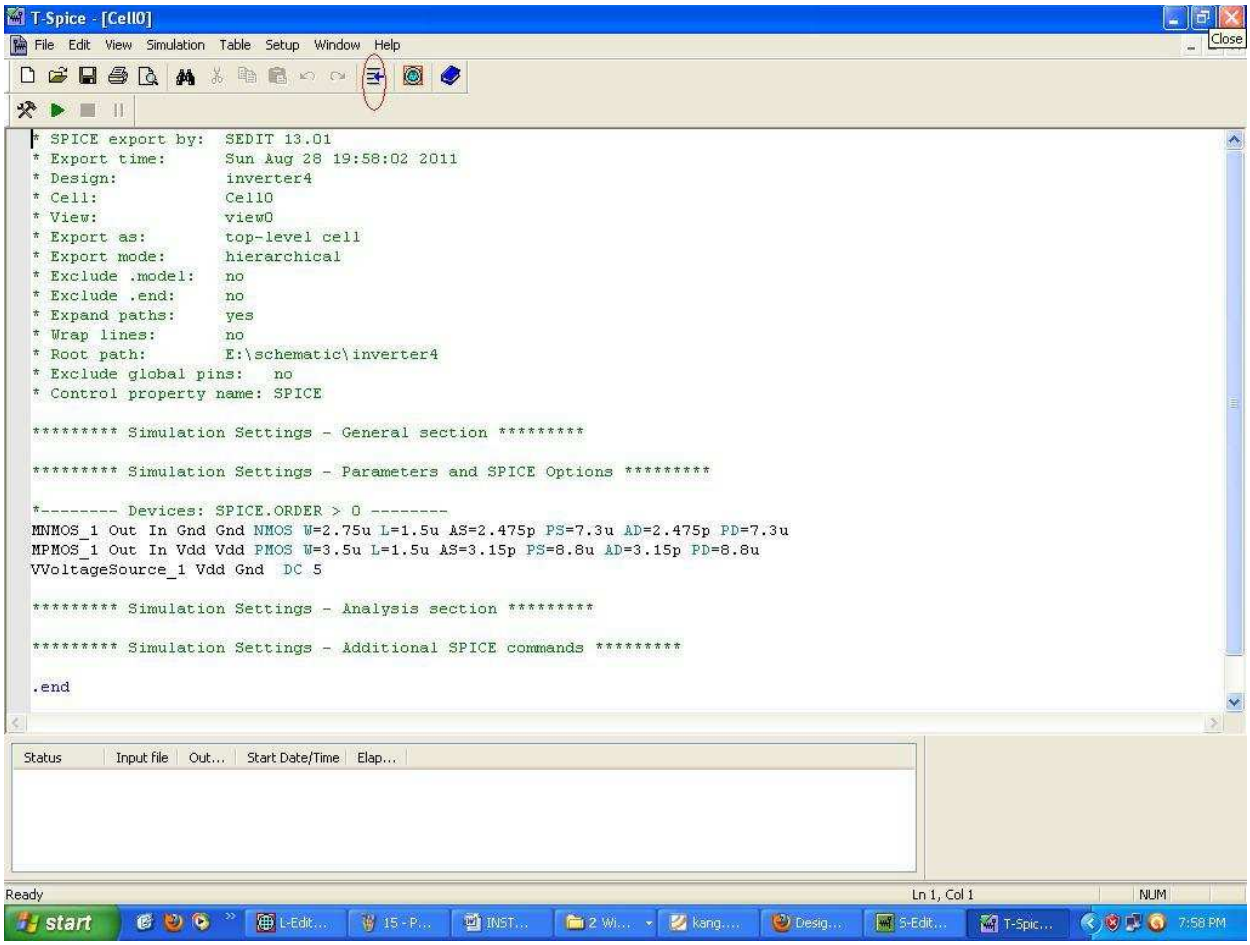
After schematic design you have to check whether your design match with the specification required or not. That's why you need to simulate the design which is called Pre layout simulation.

For simulation go to>> tools>> T-spice>> 'ok'



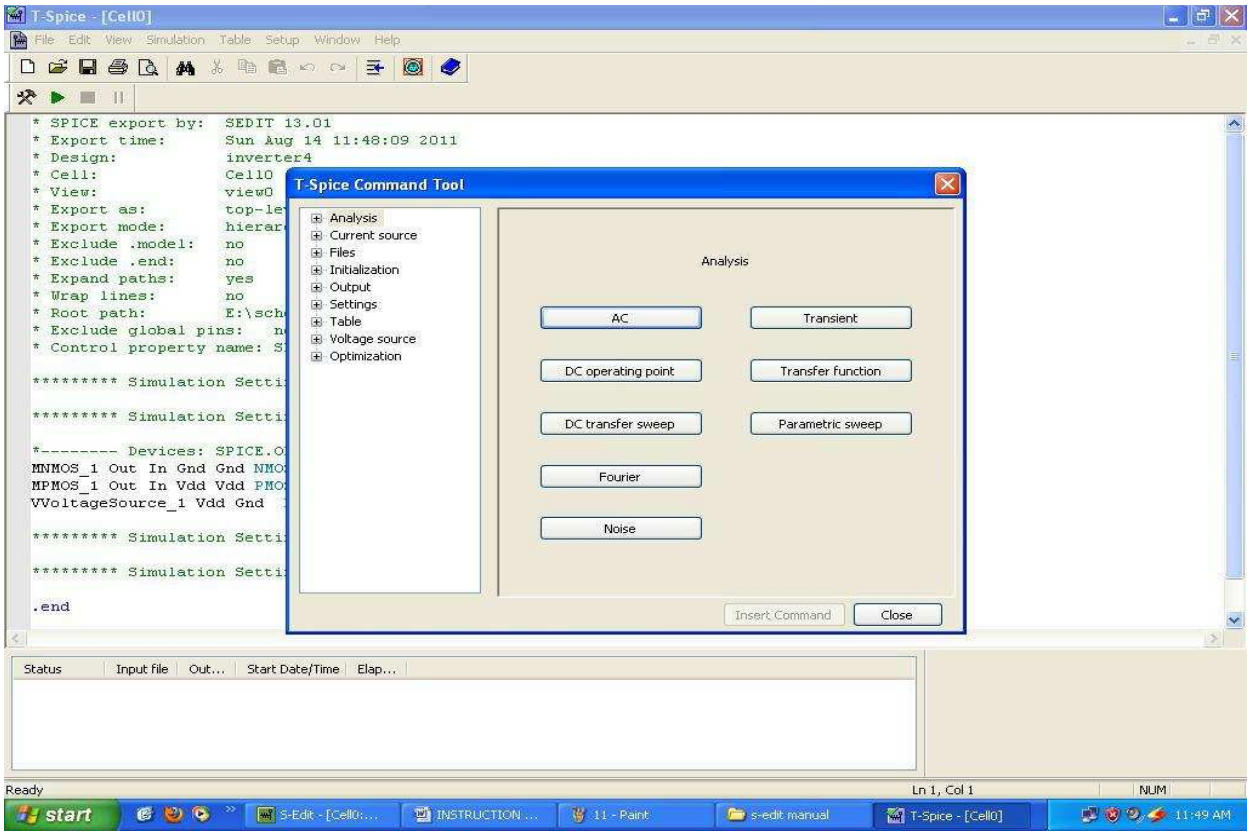
A T-spice window will open.

Then click on the bar shown by red ellipse



A “T-spice command Tool “ dialog box will open as shown beow.

On the



On the T-spice command you can see in the left hand side

Analysis,

Current source

Files

Initialization,

Output

Settings

Table

Voltage source

Optimization

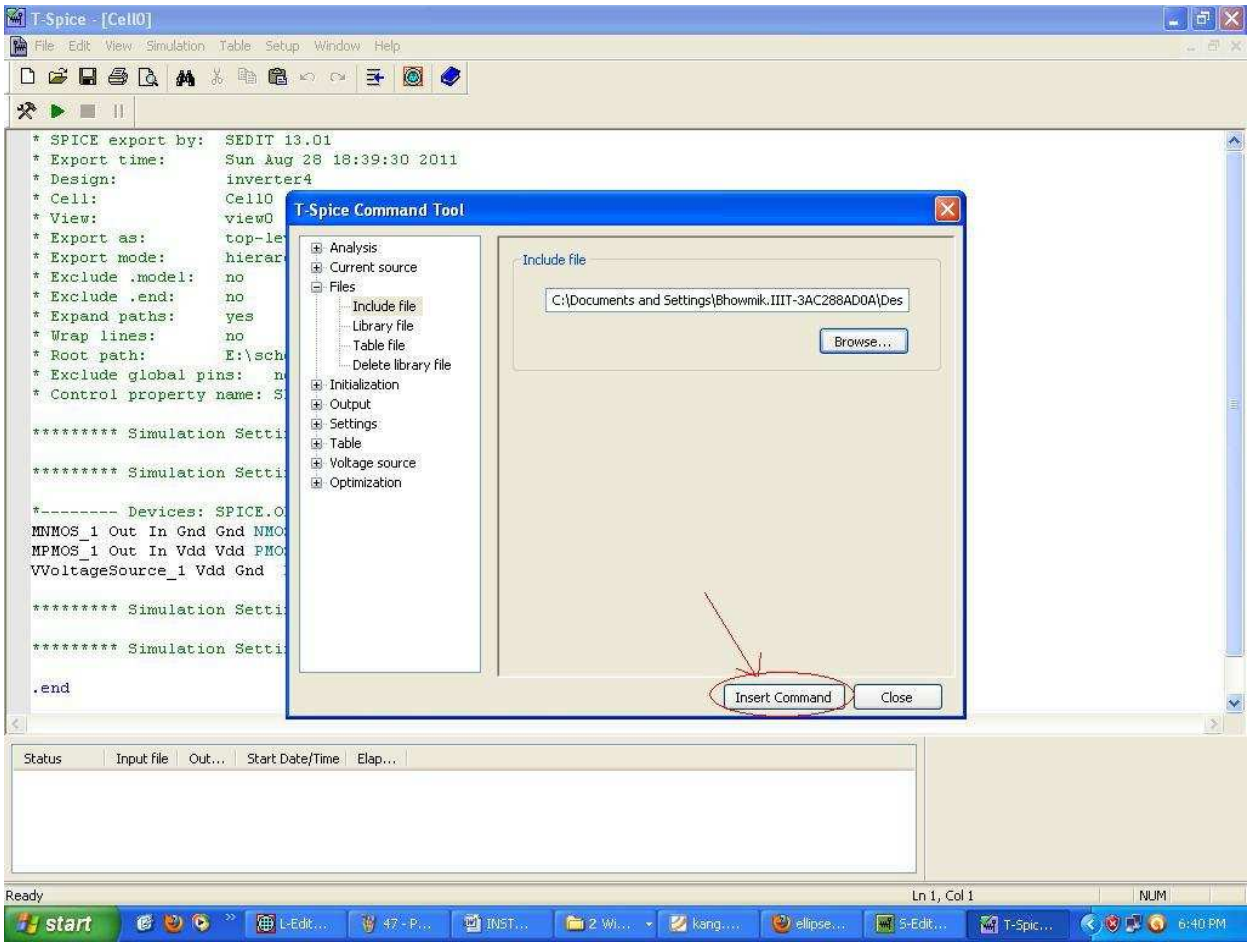
Lets start doing transient analysis of Inverter.

Step 1 : You have to include TSMC 0.18 µm Technology file .

For that

Go to >> T-spice command tool >> Files >> Include >> browse TSMC .18µm files >> Insert command.

C:\Documents and Settings\Bhowmik.IIIT-3AC288AD0A\Desktop\TSMC 0.18um\MODEL_0.18.md



Step2 : Then to give Input

T-spice command tool >> Voltage source >> select type of input you want to give(lets take **bit**) >> Insert command

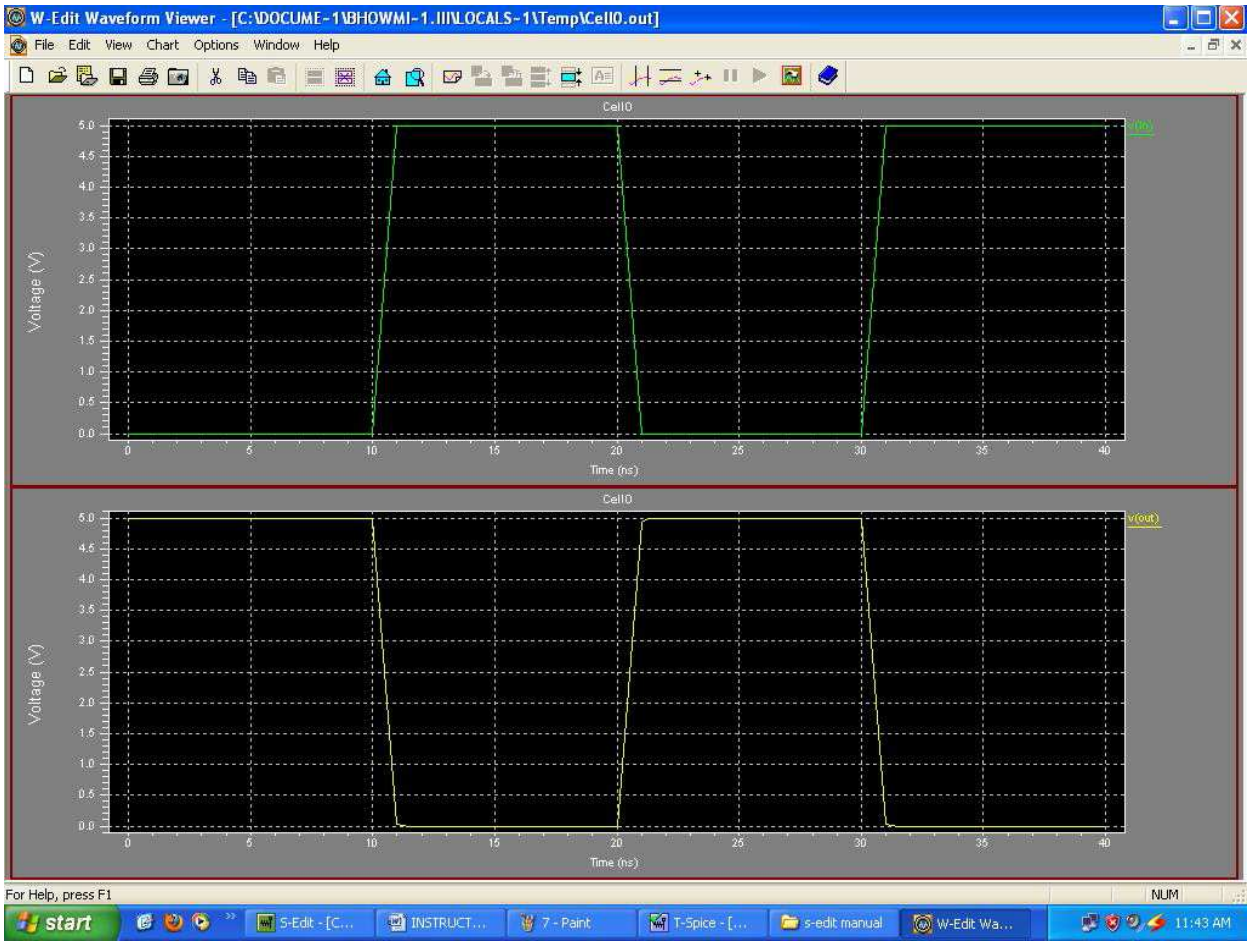
Step 3: Analysis

T-spice command tool >> Analysis >> select type of analysis you want to give(lets take **transient**) >> Insert command

step 4: Output

T-spice command tool >> Output >> which output you want to see >> Insert Command

Output of Pre layout simulation of Inverter



CHAPTER-6

RESULTS AND POWER ANALYSIS

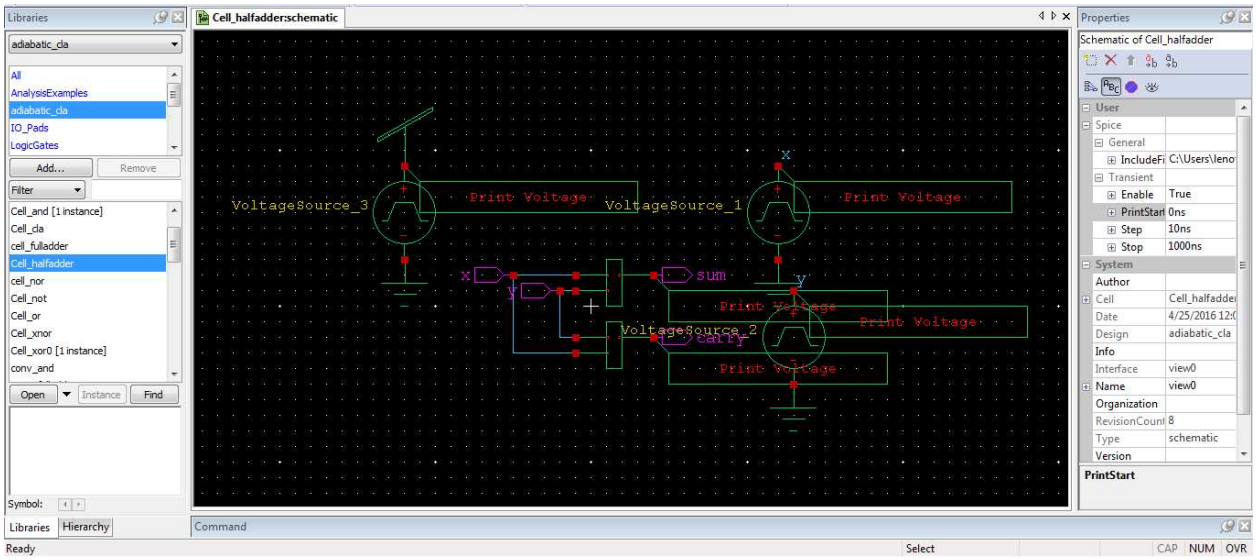


fig 6.1 Half adder schematic

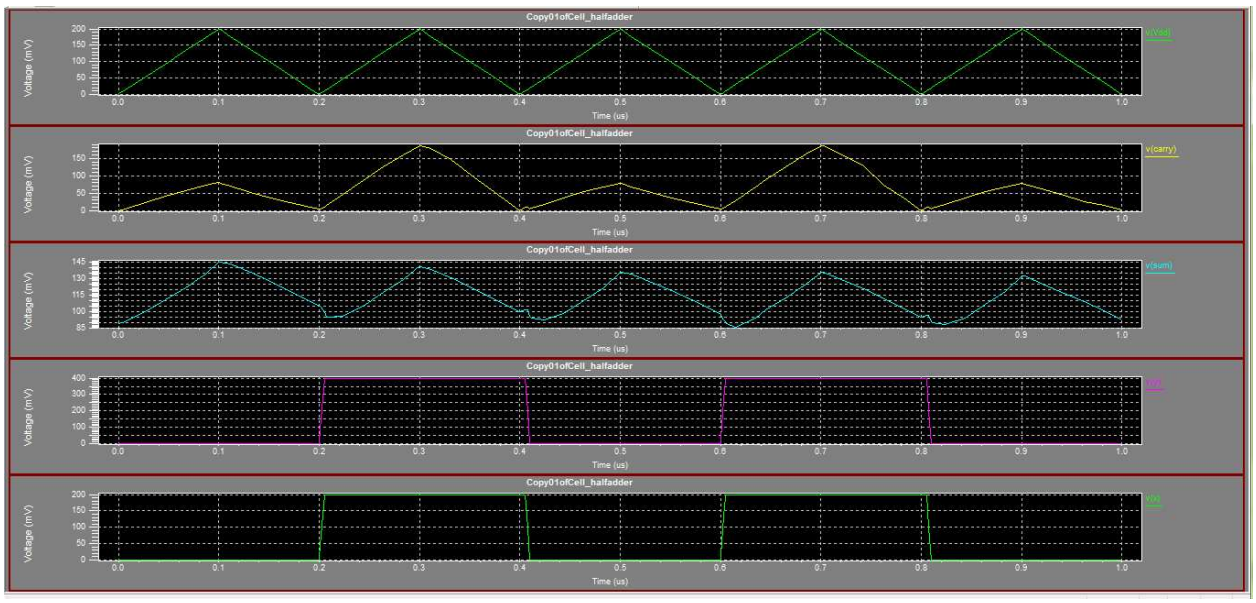


fig 6.2 Half adder simulation

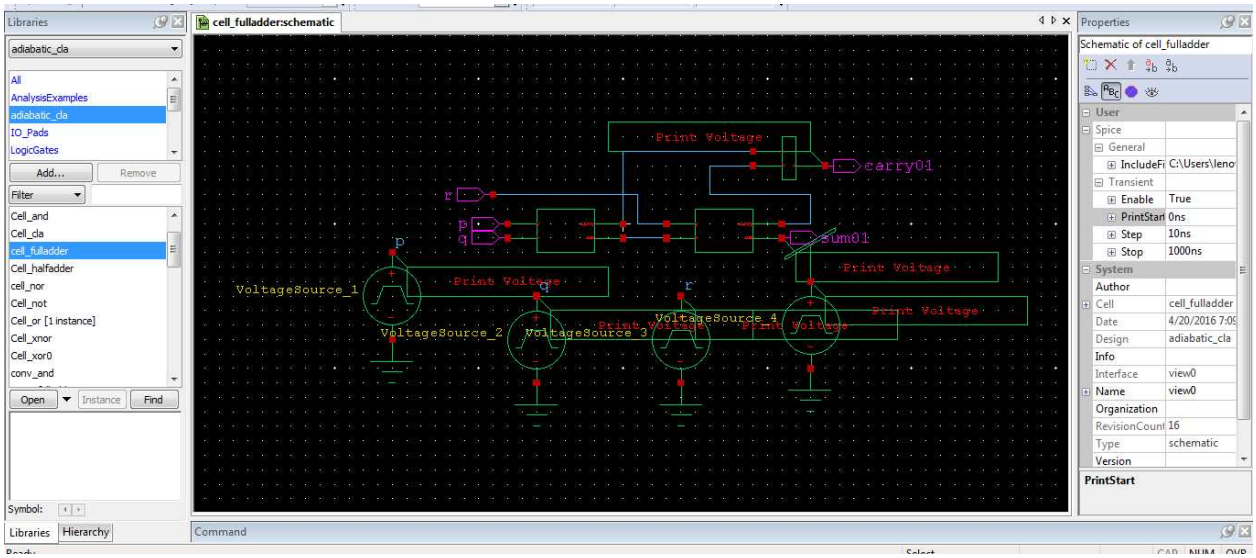


fig 6.3 full adder schematic

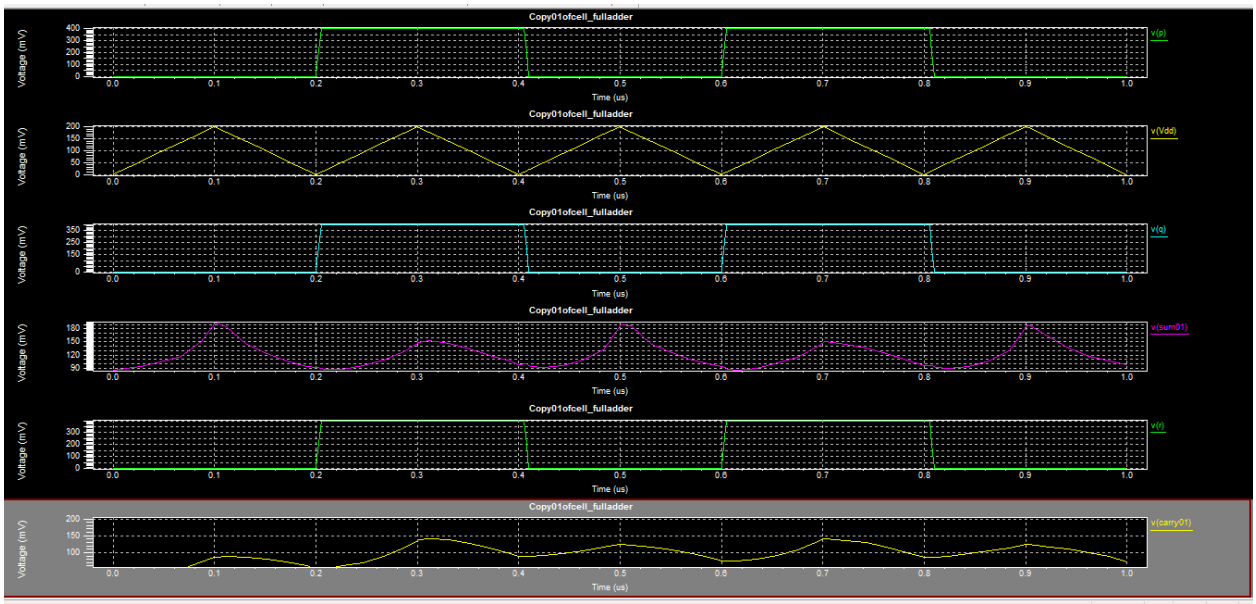


fig 6.4 full adder simulation

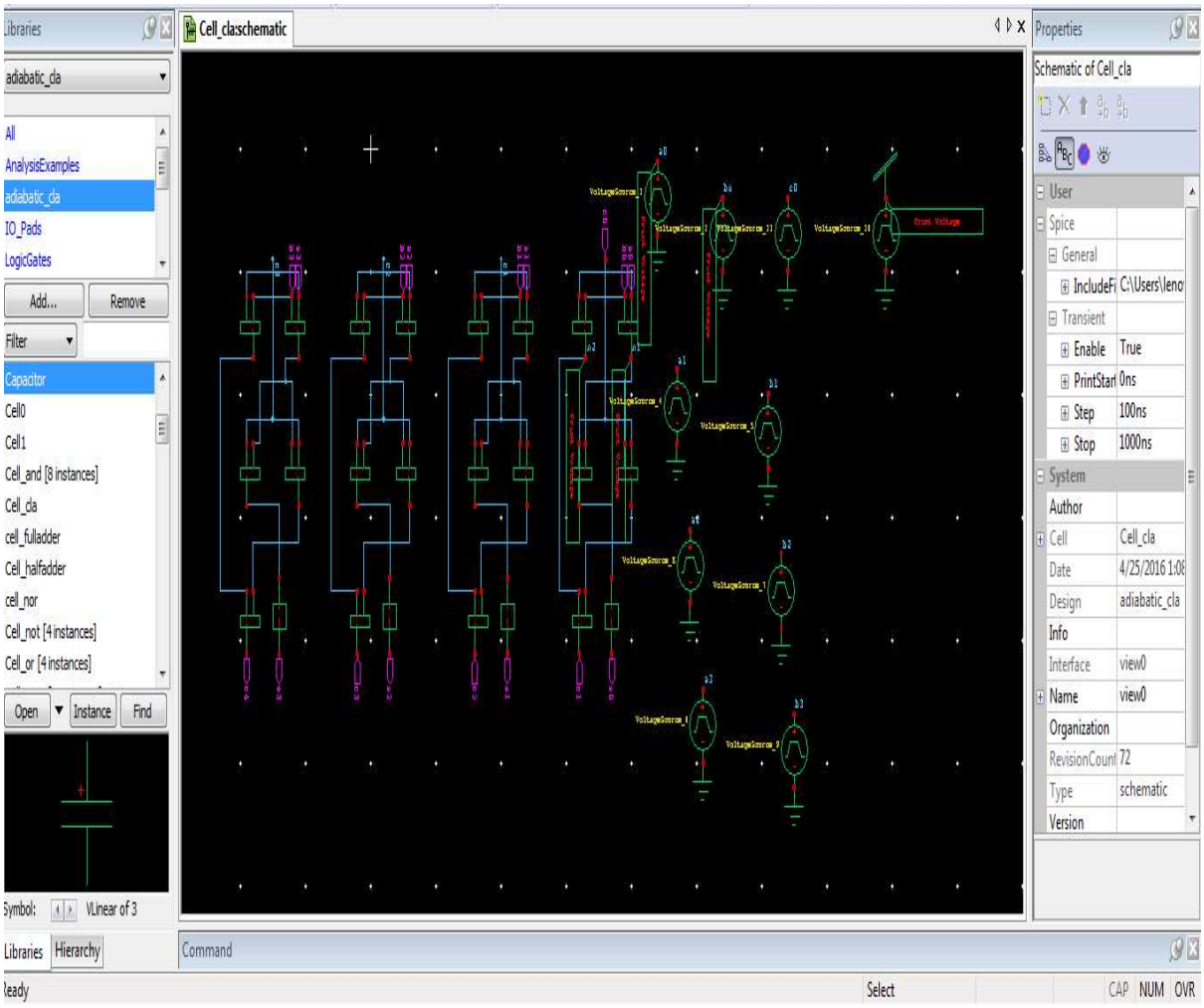


fig 6.5 Carry look ahead adder schematic

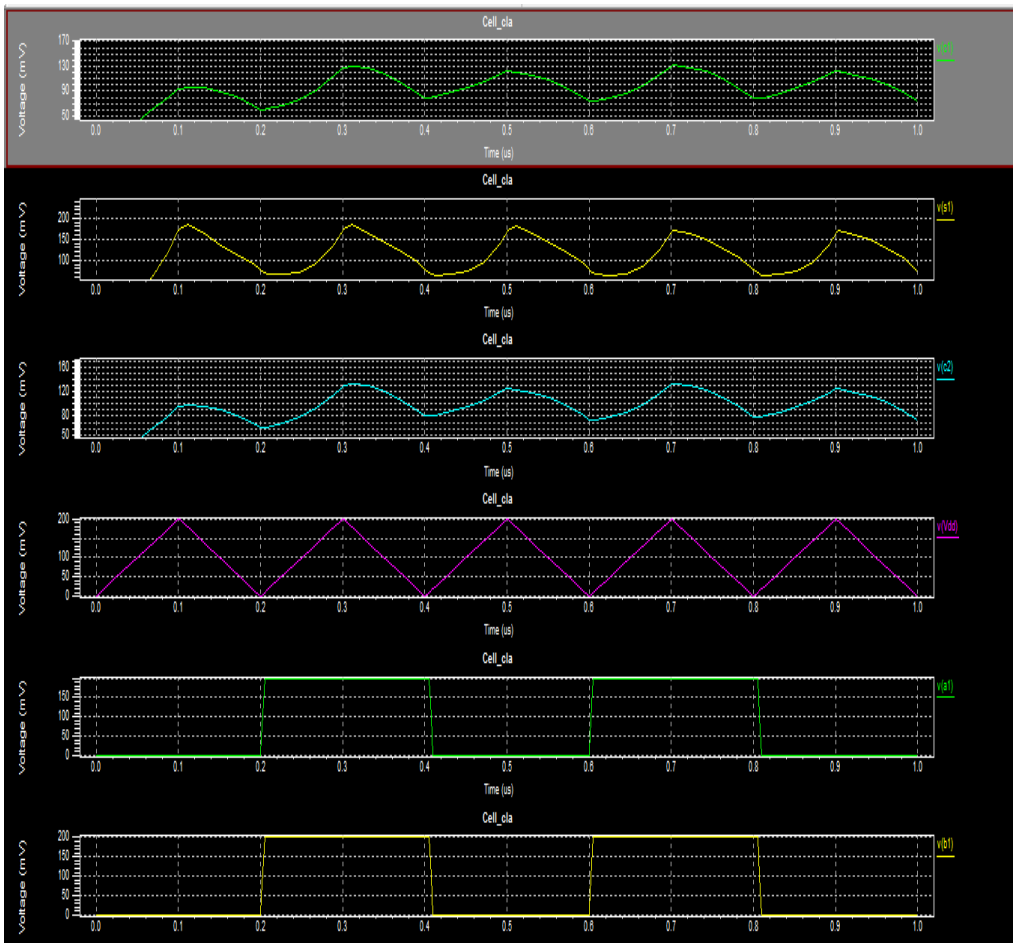


fig 6.6 Carry look ahead adder simulation

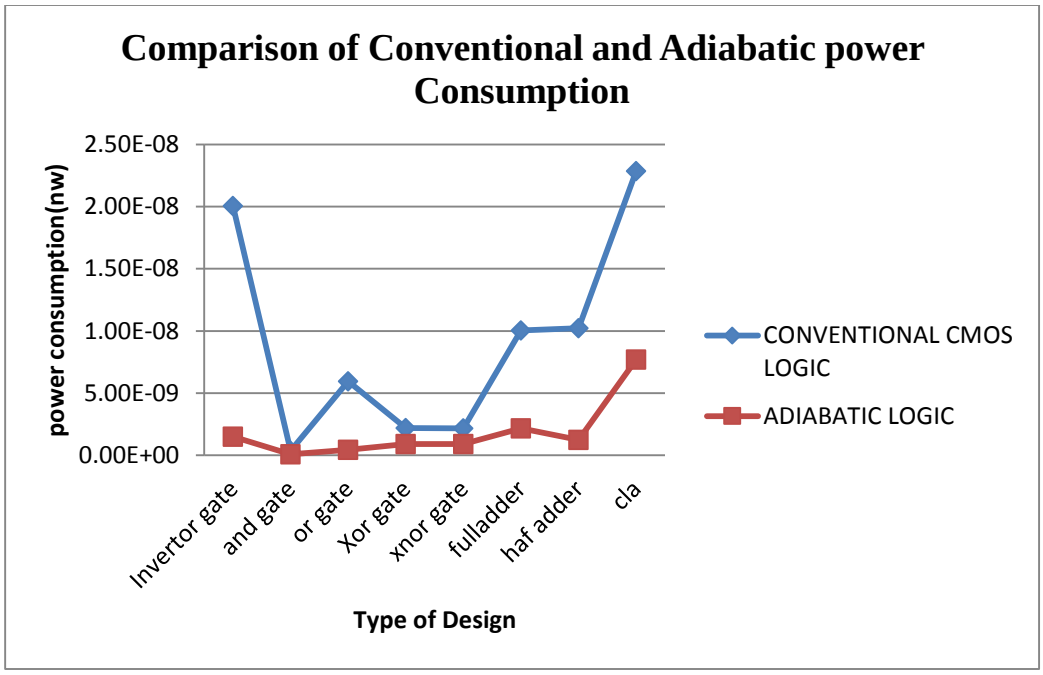


fig 6.7 Comparing Conventional and Adiabatic power Consumption

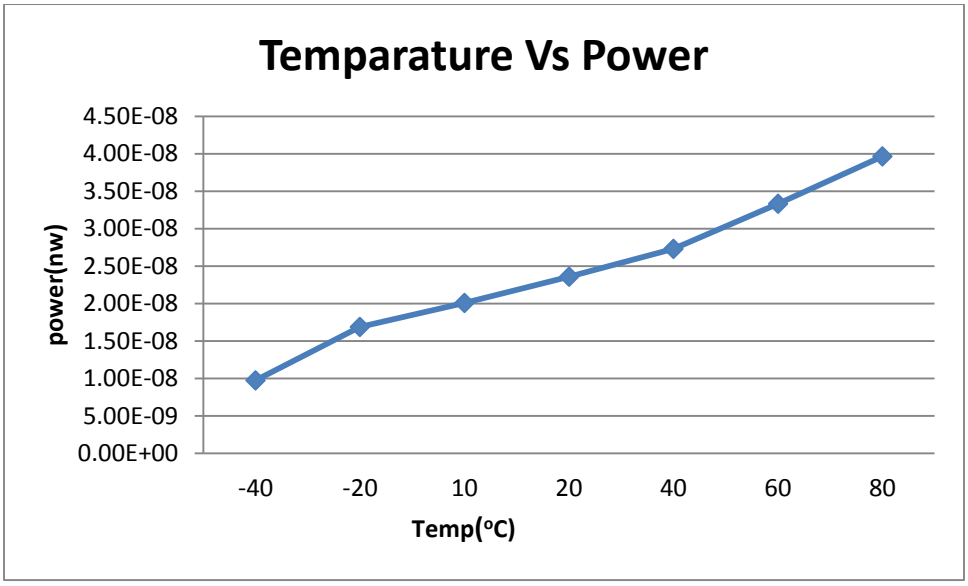


fig 6.8 Power dissipation of 4-bit Adiabatic CLA for different temperatures.

S.NO.	TYPE OF DESIGN	CONVENTIONAL CMOS		ADIABATIC CMOS	
		POWER (nw)	NO OF TRANSISTORS	POWER (nw)	NO OF TRANSISTORS
1.	Inverter gate	2.01E-08	2	1.50E-09	1
2.	and gate	3.78E-10	6	8.44E-11	2

3.	or gate	5.96E-09	6	4.41E-10	2
4.	Xor gate	2.20E-09	8	9.15E-10	4
5.	Xnor gate	2.16E-09	8	9.15E-10	4
6.	Full adder	1.00E-08	28	2.18E-09	12
7.	haf adder	1.02E-08	14	1.25E-09	6
8.	cla	2.29E-08	36	7.71E-09	15

fig 6.8Comparison of Conventional and Adiabatic CMOS Designs

REFERENCES:

[1] M. Alioto, "Ultralow power VLSI circuit design demystified and explained: A tutorial," *IEEE Trans. Circuits Syst. I*, vol. 59, no. 1, pp. 3–29, Jan. 2012.

[2] D. Bol, R. Ambroise, D. Flandre, and J. D. Legat, "Analysis and minimization of practical energy in 45 nm subthreshold logic circuits," in *Proc. IEEE Int. Conf. Comput. Design*, Oct. 2008, pp. 294–300.

[3] D. Markovic, C. C. Wang, L. P. Alarcon, and J. M. Rabaey, "Ultra low power design in near-threshold region," *Proc. IEEE*, vol. 98, no. 2, pp. 237–252, Feb. 2010.

[4] B. Zhai, S. Hanson, D. Blaauw, and D. Sylvester, "Analysis and mitigation of variability in subthreshold design," in *Proc. Int. Symp. Low Power Electron. Design*, Aug. 2005, pp. 20–25.

[5] N. Verma, J. Kwong, and A. P. Chandrakasan, "Nanometer MOSFET variation in minimum energy subthreshold circuits," *IEEE Trans. Electron Devices*, vol. 55, no. 1, pp. 163–174, Jan. 2008.

[6] R. Swanson and J. Meindl, "Ion-implanted complementary MOS transistors in low-voltage circuits," *IEEE J. Solid-State Circuits*, vol. 7, no.2, pp. 146–153, Apr. 1972.

[7] B. Zhai, L. Nazhandali, J. Olson, A. Reeves, M. Minuth, R. Helfand, S. Pant, D. Blaauw, and T. Austin, "A 2.60 pJ/Inst subthreshold sensor processor for optimal energy efficiency," in *Symp. VLSI Circuits, Dig. Tech. Papers*, 2006, pp. 154–155.

[8] W. M. Pensey and L. Lau, *MOS Integrated Circuits*. New York: Van Nostrand, 1972, pp. 260–282.

[9] H. Soeleman, K. Roy, and B. Paul, "Sub-domino logic: Ultralow power dynamic sub-threshold digital logic," in *Proc. 14th Int. Conf. VLSI Design*, 2001, pp. 211–214.

[10] H. Yamada, T. Hotta, T. Nishiyama, F. Murabayashi, T. Yamauchi, and H. Sawamoto, "A 13.3 ns double-precision floating-point ALU and multiplier," in *Proc. IEEE Int. Conf. Comput. Design: VLSI Comput. Process.*, Oct. 1995, pp. 466–470.

[11] H. Razak, *High Performance ASIC Design: Using Synthesizable Domino Logic in an ASIC Flow*. Cambridge, U.K.: Cambridge Univ. Press, 2008.

[12] B. H. Calhoun, A. Wang, and A. Chandrakasan, "Modeling and sizing for minimum energy operation in subthreshold circuits," *IEEE J. Solid-State Circuits*, vol. 40, no. 9, pp. 1778–1786, Sep. 2005.

[13] Y. Pu, J. P. de Gyvez, H. Corporaal, and Y. Ha, "An ultralowenergy/ frame multi-standard JPEG co-processor in 65 nm CMOS with sub/near-threshold power supply," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2009, pp. 146–147.

[14] J. Kwong and A. P. Chandrakasan, "Variation-driven device sizing for minimum energy sub-threshold circuits," in *Proc. Int. Symp. Low Power Electron. Design*, Oct. 2006, pp. 8–13.

[15] E. Seevinck, F. List, and J. Lohstroh, "Static-noise margin analysis of MOS SRAM cells," *IEEE J. Solid-State Circuits*, vol. 22, no. 5, pp.748–754, Oct. 1987.