



Q: What is meant by CMOS device?

Complementary metal-oxide-semiconductor (CMOS) is a technology for constructing integrated circuits.

Two important characteristics of CMOS devices are :

high noise immunity and low static power consumption.

Significant power is only drawn while the transistors in the CMOS device are switching between on and off states.

CMOS also allows a high density of logic functions on a chip.

The phrase "metal-oxide-semiconductor" is a reference to the physical structure of certain field-effect transistors, having a metal gate electrode placed on top of an oxide insulator, which in turn is on top of a semiconductor material. Aluminum was once used but now the material is polysilicon.

Q Explain why and how a MOSFET works.

MOSFET is a metal oxide semiconductor field effect transistor, its unidirectional device like, formed by four terminals GATE, SOURCE, DRAIN AND SUBSTRATE. GATE is the control signal, depends upon voltage applied at gate terminal MOSFET work on three regions saturation, cutoff and active region. active region is used for

amplification, cut-off and saturation region used for switching operation, mostly digital ck+ design

Enhancement-mode, n-channel mBSFET

- * Cutoff, subthreshold, or weak-inversion mode

When $V_{GS} < V_{th}$:

- * Triode mode or linear region (also known as the ohmic mode)

When $V_{GS} > V_{th}$ and $V_{DS} < (V_{GS} - V_{th})$

- * Saturation or active mode

When $V_{GS} > V_{th}$ and $V_{DS} > (V_{GS} - V_{th})$

Body effect :The body effect describes the changes in the threshold voltage by the change in the source-bulk voltage.

Q: What are advantages of BJT over mBSFET?

BJTs have some advantages over mBSFETs for at least two digital applications.

- * Firstly, in high speed switching, they do not have the "larger" capacitance from the gate, which when

multiplied by the resistance of the channel gives the intrinsic time constant of the process. The intrinsic time constant places a limit on the speed a MOSFET can operate at because higher frequency signals are filtered out. Widening the channel reduces the resistance of the channel, but increases the capacitance by exactly the same amount.

- * The second application where BJT's have an advantage over MOSFET's stems from the first. When driving many other gates, called fanout, the resistance of the MOSFET is in series with the gate capacitances of the other FET's, creating a secondary time constant.

Q:Q: What is Process variation

Process variation is the naturally occurring variation the attributes of transistors (length, widths, oxide thickness) when integrated circuits are fabricated. It becomes particularly important at smaller process nodes ($< 65 \text{ nm}$) as the variation becomes a larger percentage of the full length or width of the device and as feature sizes approach the fundamental dimensions such as the size of atoms and the wavelength of usable

light for patterning lithography masks.

Process variation causes measurable and predictable variance in the output performance of all circuits but particularly analog circuits due to mismatch.

Q: Draw V_{ds} - I_{ds} curve for a mOSFET. Now, show how this curve changes (a) with increasing V_{gs} (b) with increasing transistor width (c) considering Channel Length modulation?

I_{ds} is directly proportional to B (beta) and B depends on W/L . Therefore I_{ds} is directly proportional to width of the gate.

Q: Explain various mOSFET capacitances and their significance.

Two significant capacitance are:

Gate(load) capacitance(logical effort) and Diffusion capacitance(parasitic delay)

fig2 and fig3

Q: What is Hot carriers effect?

As transistors switch some high energy ("hot") carriers may be injected into the gate oxide and becomes trapped there. Reducing current in NmBS and increasing current in PmBS.

Q: What happens if V_{ds} is increased over saturation?

Ans: Pinch off

In saturation V_{ds} has no control on the output current but, if we're taking channel length modulation in account then output current depends on V_{ds} . In saturation region R_{out} is infinity, but in case of channel modulation it will have some value. So, in this case if we start increasing the V_{ds} , r_{out} current will start decreasing.

Thus the current starts increasing and it may reach a breakdown called avalanche breakdown.

Increasing V_{ds} would increase dc current slightly due to channel length modulation, further increase would lead to avalanche breakdown of transistor.

Q: In the $I-V$ characteristics curve, why is the

saturation curve flat or constant?

Answer Pinch off

After saturation V_{DS} have no effect on gate current.

Condition for saturation $V_{DS} > (V_{GS} - V_{th})$

Q: What are the different regions of operation in a mos transistor? Explain.

- Cutoff, subthreshold, or weak-inversion mode

When $V_{GS} < V_{th}$:

According to the basic threshold model, the transistor is turned off, and there is no conduction between drain and source. In reality, the Boltzmann distribution of electron energies allows some of the more energetic electrons at the source to enter the channel and flow to the drain, resulting in a subthreshold current that is an exponential function of gate-source voltage. While the current between drain and source should ideally be zero when the transistor is being used as a turned-off switch, there is a weak-inversion current, sometimes called subthreshold leakage.

- Triode mode or linear region (also known as the ohmic

mode)

When $V_{GS} > V_{th}$ and $V_{DS} < (V_{GS} - V_{th})$

The transistor is turned on, and a channel has been created which allows current to flow between the drain and the source. The mQSFET operates like a resistor, controlled by the gate voltage relative to both the source and drain voltages. The current from drain to source is modeled as:

where μ_n is the charge-carrier effective mobility, W is the gate width, L is the gate length and C_{ox} is the gate oxide capacitance per unit area.

- Saturation or active mode :

When $V_{GS} > V_{th}$ and $V_{DS} > (V_{GS} - V_{th})$

The switch is turned on, and a channel has been created, which allows current to flow between the drain and source. Since the drain voltage is higher than the gate voltage, the electrons spread out, and conduction is not through a narrow channel but through a broader, two- or three-dimensional current distribution extending away from the interface and deeper in the substrate.

The onset of this region is also known as pinch-off to indicate the lack of channel region near the drain.

Q) What are the effects of the output characteristics

for a change in the beta (β) value?

Current mirror

A current mirror is a circuit designed to copy a current through one active device by controlling the current in another active device of a circuit, keeping the output current constant regardless of loading. The current mirror is used to provide bias currents and active loads to circuits.

Q) What is electron migration and how can it be eliminated?

Electromigration is the transport of material caused by the gradual movement of the ions in a conductor due to the momentum transfer between conducting electrons and diffusing metal atoms. The effect is important in applications where high direct current densities are used, such as in microelectronics and related structures. As the structure size in electronics such as integrated circuits (ICs) decreases, the practical significance of this effect increases.

Causes wearout of metal interconnect through the formation of voids. Electron wind causes metal atom to

migrate over time.

Eliminated by alloy - Al is dopped with Cu.

There is a lower limit for the length of the interconnect that will allow electromigration to occur. It is known as "Blech length", and any wire that has a length below this limit will not fail by electromigration.

It increases with temperature.

Self heating occur in a bidirection wire while electromigration occur in single direction.

Q1 Can both pmos and nmos transistors pass good 1 and good 0? Explain.

No, NmOS pass weak one. $V_{dd} - V_t$

$V_{gs} > V_t$ Therefore, $V_s < V_{dd} - V_t$ always since after that transistor will be cut off.

How do you size PMOS and NMOS transistors to increase threshold?

As a designer you can't change NMOS and PMOS size in order to change

V_t .

The only way you can change V_t as a designer is by changing the Body Bias.

What are the different methodologies used to reduce the charge sharing in dynamic logic?

- q Solution: add secondary precharge transistors
 - Typically need to precharge every other node
- q Big load capacitance C_L helps as well

Setup and hold time calculation

Ques Consider a system which works at a clock frequency of 100MHz. Suppose you have a setup violation of 0.2ns and hold violation of 0.8ns calculate the clock frequency at which the system work fine?

Hey guys, pls give me equations for t_{setup} and t_{hold}...also along with your answers.

Solution: For setup time, it's easy. The system can have

orsuvenkataKrishniah

a period about $(10\text{ns} + 0.2\text{ns}) = 10.2\text{ns}$, which means 98MHz. But for hold time violation 0.8ns, I am rather confused. Slow down the system clock has no effect on hold time since hold time compares the same clock edge. The only method to solve hold violation is adding buffers, not slow down the whole system clock.

Another Answer:

How you are missing important point . if Setup + hold are observed in same path, when you add buffer into the data to provide propagation delay of 0.8 ns for fixing hold violation, the setup violation will also increased by 0.8ns.

So the effective setup violation will be 1ns after fixing hold violation + you need to increase the clock period by 1ns to fix the effective setup violation ($10 + 1 = 11\text{ns}$).
(Reduce the clock frequency = $1/11\text{ns}$)

Solving hold time:

The only method to solve hold violation is adding buffers, not slow down the whole system clock. However, you cannot fix the hold time violation by slowing down the clock. You need to add buffers to path to make sure it

transitions later and is captured by the next rising edge.

for hold time violation, we can add some delay in fastest path to eliminate

Solving set up time

Reduce the frequency, increase the time period. If you slow down the clock frequency, you can fix the setup violation.

We can't have the data arrive exactly when the rising edge of destination clock is approaching because it would lead to metastability.

If you have setup violation then slowing down the clock will make it work. However, if you have designed the circuit for a set frequency, then you need to remove some logic from the cycle and put it in the next or previous cycle.

To fix holdtime violation add additional logic (usually buffers) between the two flops. This will delay the data change from one FF to the next "after" the holdtime of the next flop has been satisfied.

Lowering of Vdd may or may not help if you want to fix the holdtime violation as it depends on how the input circuitry to the flop has been designed.

Setup time

fixing:

- 1) reducing combinational logic delay by minimising number of logic levels
- 2) splitting the combinational logic
- 3) Implementing Pipelining
- 4) Using double synchronizer using flipflops

Hold time fixing:

- 1) Can be fixed by adding delays on input ports
- 2) adjusting clock speed

Question: If you only have 1 chance to fix a setup or hold time violation, which do you choose?

Answer is fix the hold time violation, because the setup violation can always be fixed by just slowing down the clock.

Generally hold time is not in the user control.

We try to avoid fixing hold time at the synthesis level.

Hold time is independent of clock speed, which makes it a potential design killer. No matter how much you slow your clock you may still have hold violations. They are easily fixed by inserting buffers or downsizing cells (but making sure not to create any setup violations). Also a good idea to add some extra hold.

margin.

1) What causes HOLD VIOLATIONS in D2SdGN.

Simply, data should be hold for some time (hold time) after the edge of the clock. So, if the data changes with the hold time might cause violation. In general, hold time will be fixed during backend work (during PNR) while building clock tree. If u r a frontend designer, concentrate on fixing setup time violations rather than hold violations.

2) How it effects D2SdGN.

If a chip is done with some setup violations it can work by reducing the frequency.

If a chip is done with hold violations, JUST DUMP the chip. This is how it effects at the end of the day. Hold violations needs to be fixed.

3) What changes need to be done to make D2SdGN work.

PNR tools will route and place the cells in such a way that no timing violations will occur. If still u face hold violations, u can manually work on it to fix. manually place the cells to avoid hold violations, or in the worst case, u can keep some buffers in the datapath to avoid

hold violations (but be sure setup timing is not effected.)

Equation for Setup Time

$$T_{clk} > T_{clk+q} + T_{logic} + T_{setup} + T_{skew} + T_{jitter}$$

Equation for Hold Time

$$T_{clk+q} + T_{logic} - T_{skew} > T_{hold}$$

Note that Hold Time equation is independent of clk frequency (i.e Time period T_{clk})

Key things to note from above equations a) once the silicon comes back if u have setup time problem u can increase the clock period (T_{clk}) to fix it whereas if u have hold time problem its a more serious problem and u will need a new metal fix tapeout (Bt u can still test the current chip using Low supply voltage, or High temperature or SS corner part that decrease hold time violation)

Hold violations are more critical than setup. becoz for solving setup violation u can change the operating frequency and can make ur chip work. but if there is hold violations then the functionality cant be achieved.

the chip function will be failed.

Setup and hold time violation for a latch

Latch Setup and Hold window appears on falling edge if it's high level sensitive latch or vice versa.

Once again, there are only recovery and removal times for a latch, not setup and hold times.

Question : There are 100 setup and 100 hold violations in ur design. In order to have a successful chip which one do u consider first and why ?

Answer 1) setup time violations can be taken care of by reducing the clock frequency, but the hold time violation is due to unnecessary delays on the clock tree.

(Unwanted clock skew due to bad clock tree design or place and route). therefore removing the hold time violation is a preferred option. this might as well require some changes in the design netlist...

Answer 2) first hold time violation should be sorted out. even if u satisfy setup time requirements for a particular frequency, ur system will land up in metastable state if hold is not met.

This hold time should be checked at every flop stage for

proper operation of the ck+.

RAC2 condition

I think it is because there is a sufficient amount time for the inputs to settle down ($T_{CLK}/2$) before the output voltage is allowed to change (after $T_{CLK}/2$). i.e if all the inputs cannot arrive at the processing unit at the same time, u should not manipulate output with only few inputs.

i.e read everything -then process the output.

race condition and hold time violation is two different thing!

hold time violation and setup time violation mean you can't meet timing request!

race condition mean the code you write will cause uncertain for some state!

If " $T_{CLK-q} + T_{comb} < T_{hold}$ " leads to a Race condition.

T_{CLK-q} = Clock to Q path Delay

T_{comb} = Combinational path delay between two flip-flop

T_{hold} = Flip-flop hold

time

Race condition can lead to a Hold violation.

Race is of hardware and one race is in verilog

Race condition in Verilog context refers to a condition wherein the results are unpredictable. For e.g. consider:

`always @ (a)`

`b = a;`

`always @ (a)`

`display ("a %b b %b ",a,b);`

Race can be avoided in verilog using non-blocking stmts appropriately.

Race condition: defines a condition when a device's output depends on two [or more] nearly simultaneous events to occur at the input(s) of a device and cause the device's output to switch. One way to design in a race condition is to use the same signal to perform two [opposing] functions. In some cases a race condition could also be a logic hazard. Another case for a race condition could be caused from floating inputs.

Two kinds of

racing

in JK latch, the output is feedback to the input, and therefore change in the output causes the change in inputs. due to this in the positive half of the clock pulse if J and K both are high then the output toggles simultaneously. this condition is known as race around condition.

to overcome this problem we should use master slave FF.
r u asking abt race around problem in flip flop or in the code u r writing?

if its a flipflop then the above answers r correct...

if its in the code, u will get the race condition, when the variable value is changing at a same time by two different tasks and the result value of that variable is undefined ...

Definition

An undesirable condition which may exist in a system when two or more inputs change simultaneously. If the final output depends on which input changes first, then a race condition arise.

In the JK latch, the output is feedback to the input, and therefore change in

the

output results change in the input. Due to this in the positive half of the clock pulse

if J and K are both high then output toggles continuously.
This condition is known as race around condition.

there can be conditions where the output is unpredictable when the input changes the state. this depends on the timing of the input. if the input delays in changing state then it can anomalous behaviours on the output. this is race around condition. to avoid this the input must have tolerable delay in its change of state.

Race condition occurs when two or more sequential circuits are connected back to back without/less intervening logic if the hold time is high, then in a single clock edge data may pass through two (or more) consecutive flops.. this is called a race condition or minimum delay failure or hold failure ..

Solutions:

insert logic/delay elements in between the sequential elements ..

Hold failures are very difficult to modify if found after

orsuvenkatakrishniah

tape-out and usually involves a very costly re-designing..
See fig :

dmp

Let me add a "rule for the beginner":

Verilog blocking signal assignment ("=") is similar to VHDL variables,

while non-blocking signal assignment ("<=") is similar to VHDL signals.

What is the effect of body bias?

How a body bias can intentionally be applied to alter the threshold voltage, permitting tradeoffs between performance and sub-threshold leakage current.
Forward body bias (higher V_t) increase performance
Reverse body bias reduce leakage.

HOT electron effect and how it can be overcome?

hot electron means...

because of temperature or any other effects the electrons or holes jump from channel region to gate oxide region...this is called hot electron effect...

effect :

These electrons impact the drain, dislodging holes. If these electrons enter gateoxide, it will cause gate current. This may lead to degradation of device parameters like threshold current, subthreshold current

and transconductance.

Reducing current in NmBS and increasing current in PmBS

avoiding hot electron effect:

We have to keep diode in reverse bias at gate terminal....

What is latchup problem and how can it be eliminated?

Latchup is a term used in the realm of integrated circuits (ICs) to describe a particular type of short circuit which can occur in an improperly designed circuit. more specifically it is the inadvertent creation of a low-impedance path between the power supply rails of a MOSFET circuit, triggering a parasitic structure which disrupts proper functioning of the part and possibly even leading to its destruction due to overcurrent.

During a latchup when one of the transistors is conducting, the other one begins conducting too. Similar to thyristor or SCR.

Prevention:

It is possible to design chips that are latchup-resistant, where a layer of insulating oxide (called a trench) surrounds both the NmBS and the PmBS

transistors.

Another possibility for a latchup prevention is the Latchup Protection Technology circuit. When a latchup is detected, the LPT circuit shuts down the chip and holds it powered-down for a preset time.

What is channel length modulation?

One of several short channel effects in mOSFET scaling, channel length modulation (CLm) is a shortening of the length of the inverted channel region with increase in drain bias for large drain biases. The result of CLm is an increase in current with drain bias and a reduction of output resistance.

To understand the effect, first the notion of pinch-off of the channel is introduced. The channel is formed by attraction of carriers to the gate, and the current drawn through the channel is nearly a constant independent of drain voltage in saturation mode.

However, near the drain, the gate and drain jointly determine the electric field pattern. Instead of flowing in a channel, beyond the pinch-off point the carriers flow in a subsurface pattern made possible because the drain and the gate both control the current. In the figure at the right, the channel is indicated by a

dashed line and becomes weaker as the drain is approached, leaving a gap of uninvited silicon between the end of the formed inversion layer and the drain (the pinch-off region).

As the drain voltage increases, its control over the current extends further toward the source, so the uninvited region expands toward the source, shortening the length of the channel region, the effect called channel-length modulation. Because resistance is proportional to length, shortening the channel decreases its resistance, causing an increase in current with increase in drain bias for a mOSFET operating in saturation.

Threshold voltage

While,

t_{ox} is the oxide thickness, therefore V_t is directly proportional to oxide thickness.

As with the case of oxide thickness affecting threshold voltage, temperature has an effect on the threshold voltage of a CMOS

device

What is the effect of temperature on mobility?

With increase in temperature carrier mobility decrease
3.

Threshold voltage and hence performance also decrease with increase in temperature

Junction leakage increases with temperature.

Interconnect capacitance

Traditionally, switching time was roughly proportional to the gate capacitance of gates. However, with transistors becoming smaller and more transistors being placed on the chip, interconnect capacitance (the capacitance of the metal-layer connections between different parts of the chip) is becoming a large percentage of capacitance.^[30] ^[31] Signals have to travel through the interconnect, which leads to increased delay and lower performance.

Any two adjacent conductors can be considered a capacitor, although the capacitance will be small unless

the conductors are close together for long. This (often unwanted) effect is termed "stray capacitance". Stray capacitance can allow signals to leak between otherwise isolated circuits (an effect called crosstalk), and it can be a limiting factor for proper functioning of circuits at high frequency.

What are the different types of scaling?

Over the past decades, the MOSFET has continually been scaled down in size; typical MOSFET channel lengths were once several micrometres, but modern integrated circuits are incorporating MOSFETs with channel lengths of tens of nanometers. Intel began production of a process featuring a 32 nm feature size (with the channel being even shorter) in late 2009. Smaller MOSFETs are desirable for several reasons. The main reason to make transistors smaller is to pack more and more devices in a given chip area. This results in a chip with the same functionality in a smaller area, or chips with more functionality in the same area. Since fabrication costs for a semiconductor wafer are relatively fixed, the cost per integrated circuits is mainly related to the number of chips that can be produced.

per wafer. Moore's law the number of transistor doubles every two years.

It is also expected that smaller transistors switch faster. For example, one approach to size reduction is a scaling of the mOSFET that requires all device dimensions to reduce proportionally. The main device dimensions are the transistor length, width, and the oxide thickness, each (used to) scale with a factor of 0.7 per node.

Scaling :

Dimensions

Device voltage

Doping concentrations

Stage Ratio

I am not sure this is it, but it might be the ratio of capacitance driven by a cell to the capacitance at its input. This is also known as electrical fanout (as opposed to logical fanout, which is just the number of gates driven). Rule of thumb is to make this number around 4-5.

$fd = g \cdot h$ g = logical effort fd is known as stage

effort.

$$f(\text{Optimized}) = g_{hi} = Fl/n$$

Charge sharing in bus

Say that the bus carries a single bit a certain value, to read this value, you're going to connect the input of another circuit to the bus via a switch.

Let's assume that the value on the bus is one and has a known capacitance associated with it. And also assume that your input circuit has a known parasitic cap at its output and its initially discharged.

The instance you're going to connect the two caps together, charge from the bus will go to the input parasitic cap, until both caps have the same potential.

If you apply the law of charge conservation, you can predict how much drop to expect on the data bus.

In CMOS

charge sharing is a phenomenon found in clocked cmos logic and dynamic cmos logic.

in the precharge mode the output capacitance is charged to Vdd. now if the logic block has top transistors in the on state and the bottom ones (those which are nearer to Vss) in off state then during the

evaluate mode the charge on the output capacitance is shared among the capacitances of transistors which are in the on state. This is charge sharing due to which the output voltage degrades below the logic threshold sometimes.

$$\text{where } v_x = v_y = c_y (V_{ex} + t_{ey}) * V_{dd}$$

SRAM

Static random access memory (SRAM) is a type of semiconductor memory where the word static indicates that, unlike dynamic RAM (DRAM), it does not need to be periodically refreshed, as SRAM uses bistable latching circuitry to store each bit. SRAM exhibits data remanence, but is still volatile in the conventional sense that data is eventually lost when the memory is not powered.

Standby

If the word line is not asserted, the access transistors m_3 and m_6 disconnect the cell from the bit lines. The two cross coupled inverters formed by $m_1 - m_4$ will continue to reinforce each other as long as they are

connected to the supply.

Reading

Assume that the content of the memory is a 1, stored at Q. The read cycle is started by precharging both the bit lines to a logical 1, then asserting the word line WL, enabling both the access transistors. The second step occurs when the values stored in Q and \bar{Q} are transferred to the bit lines by leaving BL at its precharged value and discharging \bar{BL} through m_1 and m_3 to a logical 0. On the BL side, the transistors m_4 and m_6 pull the bit line toward VDD, a logical 1.

If the content of the memory were a 0, the opposite would happen and BL would be pulled toward 1 and \bar{BL} toward 0. Then these BL and BL-bar will have a small difference of delta between them and then these lines reach a sense amplifier, which will sense which line has higher voltage and thus will tell whether there was 1 stored or 0. the higher the sensitivity of sense amplifier is faster is the speed of read operation of SRAM.

Writing

The start of a write cycle begins by applying the value to be written to the bit lines. If we wish to write a 0, we would apply a 0 to the bit lines, i.e. setting BL to 1 and \bar{BL} to 0. This is similar to applying a reset pulse to a SR-latch, which causes the flip flop to change state. A

I is written by inverting the values of the bit lines. WL is then asserted and the value that is to be stored is latched in. Note that the reason this works is that the bit line input-drivers are designed to be much stronger than the relatively weak transistors in the cell itself, so that they can easily override the previous state of the cross-coupled inverters. Careful sizing of the transistors in an SRAM cell is needed to ensure proper operation.

DRAM

Dynamic random access memory (DRAM) is a type of random access memory that stores each bit of data in a separate capacitor within an integrated circuit. Since real capacitors leak charge, the information eventually fades unless the capacitor charge is refreshed periodically. Because of this refresh requirement, it is a dynamic memory as opposed to SRAM and other static memory.

The main memory (the "RAM") in personal computers is Dynamic RAM (DRAM), as is the "RAM" of home game consoles (PlayStation, Xbox 360 and Wii), laptop, notebook and workstation computers.

The long lines connecting each row are known as word lines. Each column is actually composed of two bit lines, each one connected to every other storage cell in the

column.

To read a bit from a column, the following operations take place:

The sense amplifier is switched off and the bit lines are precharged to exactly matching voltages that are intermediate between high and low logic levels. The bit lines are constructed symmetrically to keep them balanced as precisely as possible.

The precharge circuit is switched off. Because the bit lines are very long, their capacitance will hold the precharge voltage for a brief time. This is an example of dynamic logic.

The selected row's word line is driven high. This connects one storage capacitor to one of the two bit lines. Charge is shared between the selected storage cell and the appropriate bit line, slightly altering the voltage on the line. Although every effort is made to keep the capacitance of the storage cells high and the capacitance of the bit lines low, capacitance is proportional to physical size, and the length of the bit lines means that the net effect is a very small perturbation of one bit line's voltage.

The sense amplifier is switched on. The positive feedback takes over and amplifies the small voltage difference until one bit line is fully low and the other is

fully high. At this point, the row is "open" and a column can be selected.

Read data from the DRAM is taken from the sense amplifiers, selected by the column address. many reads can be performed while the row is open in this way.

While reads proceed, current is flowing back up the bit lines from the sense amplifiers to the storage cells. This restores (refreshes) the charge in the storage cell. Due to the length of the bit lines, this takes significant time beyond the end of sense amplification, and overlaps with one or more column reads.

When done with the current row, the word line is switched off to disconnect the storage capacitors (the row is "closed"), the sense amplifier is switched off, and the bit lines are precharged again.

To write to memory, the row is opened and a given column's sense amplifier is temporarily forced to the desired state, so it drives the bit line, which charges the capacitor to the desired value. Due to the positive feedback, the amplifier will then hold it stable even after the forcing is removed. During a write to a particular cell, the entire row is read out, one value changed, and then the entire row is written back in, as illustrated in the figure to the right.
manufacturers specify that each row should be

refreshed every 64 ms or less

Why Random Access

RAM is called "random access" because any storage location can be accessed directly.

The amount of time that RAM takes to write data or to read it once the request has been received from the processor is called the access time. Typical access times vary from 9 nanoseconds to 70 nanoseconds, depending on the kind of RAM.

Well, the biggest advantage is that it is non-volatile. Both SRAM and DRAM lose data when power is removed.

DRAM loses it in every read operation! So that's where flash is helpful. Disadvantage is that it is slower.

Question what does the Burst length in DDR SDRAM refer

to.....

DDR1 has a Burst length of 2

DDR2 has a Burst length of 4

DDR3 has a Burst length of 8

Solution $\#t$ indicates how many data bytes are written or read from DDR sdram when write or read command is given along with row and column address.

Sense amplifier in SRAM

A Sense Amplifier is an essential circuit in designing memory chips.

Due to large arrays of SRAM cells, the resulting signal, in the event of a Read operation, has a much lower voltage swing. To compensate for that swing a sense amplifier is used to amplify voltage coming off Bit Line and Bit Line. The voltage coming out of the sense amplifier typically has a fully swing (0 - 2.5V) voltage. Sense amplifier also helps reduce the delay times and power dissipation in the overall SRAM.

chip.

There are many versions of sense amplifiers used in memory chips. The one that we will use in our design is called a Cross-coupled Sense Amplifier demonstrated on a block diagram below.

During a read sequence, Bit Line and Bit Line are directed into X and X inputs. Once S2 (Sense Enable) has been set to logic 1, the amplifier turns on, and gives Y and Y as its outputs.

What happens if we use an inverter instead of the differential sense amplifier??

Common mode doesn't get rejected!

Difference Between SRAM and DRAM

- Categorized under Objects, Technology
- Difference Between SRAM and DRAM
- ddr_ram SRAM vs DRAM
 - There are two types of Random Access memory or RAM, each has its own advantages and disadvantages compared to the other. SRAM (Static RAM) and DRAM (Dynamic RAM) holds data but in a different ways. DRAM requires the data to be refreshed periodically in order to retain the data. SRAM does not need to be

refreshed as the transistors inside would continue to hold the data as long as the power supply is not cut off. This behavior leads to a few advantages, not the least of which is the much faster speed that data can be written and read.

The additional circuitry and timing needed to introduce the refresh creates some complications that makes DRAM memory slower and less desirable than SRAM.

One complication is the much higher power used by DRAM memory, this difference is very significant in battery powered devices. SRAM modules are also much simpler compared to DRAM, which makes it easier for most people to create an interface to access the memory.

This makes it easier to work with for hobbyists and even for prototyping.

Structurally, SRAM needs a lot more transistors in order to store a certain amount of memory. A DRAM module only needs a transistor and a capacitor for every bit of data where SRAM needs 6 transistors.

Because the number of transistors in a memory module determine its capacity, a DRAM module can have almost 6 times more capacity with a similar transistor count to an SRAM module. This ultimately boils down to price, which is what most buyers are really concerned with.

Because of its lower price, DRAM has become the mainstream in computer main memory despite being slower and more power hungry compared to SRAM. SRAM memory is still used in a lot of devices where speed is more crucial than capacity. The most prominent use of SRAM is in the cache memory of processors where speed is very essential, and the low power consumption translates to less heat that needs to be dissipated.

Even hard drives, optical drives, and other devices that need cache memory or buffers use SRAM modules.

Summary:

1. SRAM is static while DRAM is dynamic
2. SRAM is faster compared to DRAM
3. SRAM consumes less power than DRAM
4. SRAM uses more transistors per bit of memory compared to DRAM
5. SRAM is more expensive than DRAM
6. Cheaper DRAM is used in main memory while SRAM is commonly used in cache memory

Q) Difference between latch + flip flop

§ Latches are level sensitive i.e. the output captures the input when the clock signal is high, so as long as the

clock is logic 1, the output can change if the input also changes.

§ Flip-Flops are edge sensitive i.e. flip flop will store the input only when there is a rising or falling edge of the clock.

§ A positive level latch is transparent to the positive level(enable), and it latches the final input before it is changing its level(i.e. before enable goes to '0' or before the clock goes to -ve level.)

§ A positive edge flop will have its output effective when the clock input changes from '0' to '1' state ('1' to '0' for negative edge flop) only.

Advantages of latch design

Latches are faster, flip flops are slower. §

Latches take less gates (less power) to implement than flip-flops. §

Latch facilitate time borrowing or cycle stealing whereas flip flops allow §
synchronous logic.

latch timings (Recovery and Removal)

Recovery Time

Recovery time is the minimum length of time an asynchronous control signal (eg.preset) must be stable

before the next active clock edge. The recovery slack time calculation is similar to the clock setup slack time calculation, but it applies asynchronous control signals.

Removal Time

Removal time is the minimum length of time an asynchronous control signal must be stable after the active clock edge. Calculation is similar to the clock hold slack calculation, but it applies asynchronous control signals

Time borrowing in latches

Time borrowing is a concept that is used in latch based pipelines in which you typically have 2 stages of combinatorial surrounded by latches. If the first combinational piece of logic has a much longer delay than the second one, you can borrow some of the time of the second part to the first part

WHY CLOCK

The whole reason that we need clocks is that we want the output to depend on more than just the inputs, we want it to depend on previous outputs too.

These

previous outputs are the state bits of the Fsm, and are the signals that cause lots of problems.

The state bits cause problems because we now need some sort of policy to define what previous and next mean. This is almost always done with the help of a clock, to provide reference points in time.

Benefit of two phase clock

It will be a level sensitive design

- no race, glitch, or hazard problems
- no skew problems

One sided timing constraints

- Logic can't be too fast

Unfortunately, there is a serious problem with using flip-flops in complicated designs which include multi-level logic (i.e. designs where outputs from one flip-flop become inputs to another flip-flop). This problem is called a race condition. Take the following circuit, where the values of A, B and C are initially

O:

When A is (asynchronously) set to 1, what happens to C after the first rising clock edge?

Well, on one hand, just before the first rising clock edge, the value of B is 0 (because it was initially equal to 0), so after the first rising clock edge, the value of C should still be 0. But on the other hand, just before the first rising clock edge, the value of A is 1, so on the first rising clock edge this value is propagated to B. If the propagation is fast enough, this value of 1 might also get propagated through the second flip-flop to C.

Two phase clocking designs

A two phase clocking design fixes this problem. It uses latches as opposed to flip-flops and features two non-overlapping clocks (i.e. when one clock is high, the other clock is guaranteed to be low even with pretty high clock skew). Essentially, we avoid race conditions by following a few rules that state that only certain types of signals can go into certain types of latches. Let's explain this more carefully. Let $\underline{1}$ and $\underline{2}$ be the two clocks. The diagram below shows the waveform of both

clocks:

Call the latch that has $\underline{1}$ as clock input the " $\underline{1}$ " latch, and the latch that has $\underline{2}$ as clock input the " $\underline{2}$ " latch. Moreover, call any signal that is guaranteed not to change while $\underline{1}$ is high the "s1" signal, and any signal that is guaranteed not to change while $\underline{2}$ is high the "s2" signal. In other words, s1 signals are those suitable to be used in $\underline{1}$ latches, and s2 signals are those suitable to be used in $\underline{2}$ latches.

Therefore, the normalised delay of a two-input NAND gate driving an identical copy of itself (such that the electrical effort is 1) is

$$d = gh + p = (4 \cdot B)(1) + 2 = 10 \cdot B$$

and for a two-input NOR gate, the delay is

$$d = gh + p = (5 \cdot B)(1) + 2 = 11 \cdot B$$

CmOS

testing-----

As described in the previous module that there are various types of physical faults which are modeled as logical fault. Logical fault enables us to evaluate performance of a faulty circuit quite easily. Logical fault models are of two types :

Stuck at fault As described earlier in a faulty circuit, the lines, many times, assume some potential either zero(ground) or Vdd(power supply). Therefore it is natural to assume that that faulty lines have a permanent potential value which is either 0 or 1 (i.e. high or low). Since they are of constant value irrespective of the signals that are applied they are called stuck at faults. Stuck at faults are of two types: stuck-at-0 and stuck-at-1. There may be a situation where fault value or signal value may be fluctuating or varying; under these conditions it is referred as stuck-at-u.

Stuck at fault cover 90% of manufacturing defect.

Bridging faults: One of the manufacturing defects that occur frequently in chips and on boards is short between two unconnected lines. The two signal lines become equi-potential because of short. Bridging fault is a logical

fault representing a short between two normally unconnected signal lines which creates a new logic function.

* Iddq testing is a method for testing CMOS integrated circuits for the presence of manufacturing faults. It relies on measuring the supply current (I_{DDQ}) in the quiescent state (when the circuit is not switching and inputs are held at static values). The current consumed in the state is commonly called I_{DDQ} for I_{DD} (quiescent) and hence the name.

Iddq testing uses the principle that in a correctly operating quiescent CMOS digital circuit, there is no static current path between the power supply and ground, except for a small amount of leakage. Many common semiconductor manufacturing faults will cause the current to increase by orders of magnitude, which can be easily detected. This has the advantage of checking the chip for many possible faults with one measurement. Another advantage is that it may catch faults that are not found by conventional stuck-at fault test vectors.

Drawback: Compared to scan testing, Iddq testing is time consuming, and then more expensive, since is achieved by current measurements that take much

more time than reading digital pins in mass production.
what do you mean by Automatic test pattern generation
and describe its algorithms

ATPG (Acronym for both Automatic Test Pattern Generation and Automatic Test Pattern Generator) is an electronic design automation method/technology used to find an input (or test) sequence that, when applied to a digital circuit, enables testers to distinguish between the correct circuit behavior and the faulty circuit behavior caused by defects.

The effectiveness of ATPG is measured by the amount of modeled defects, or fault models, that are detected and the number of generated patterns. These metrics generally indicate test quality (higher with more fault detections) and test application time (higher with more patterns).

Algorithm in ATPG

- The D Algorithm was the first practical test generation algorithm in terms of memory requirements. The D Algorithm introduced D Notation which continues to be used in most ATPG algorithms.
- Path-Oriented Decision making (POD²m) is an improvement over the D Algorithm. POD²m was created in 1981 when shortcomings in D Algorithm became evident

when design innovations resulted in circuits that D Algorithm could not realize.

- * Fan-Out Oriented (FAN Algorithm) is an improvement over PQDlm. It limits the ATPG search space to reduce computation time and accelerates backtracing.
- * methods based on Boolean satisfiability are sometimes used to generate test vectors.
- * Pseudorandom test generation is the simplest method of creating tests. It uses a pseudorandom number generator to generate test vectors, and relies on logic simulation to compute good machine results, and fault simulation to calculate the fault coverage of the generated vectors.

what do you mean by Noise margin

It is the difference between input low and the output low.

Then the Noise margin for low signals is:

$$Nml = V_{oh} - V_{ol}$$

and for high signals

or suvenkata Krishniah

is:

$$N_{MOS} = V_{DH} - V_{SLH}$$

Re: Size P_{MOS} 2 or 3 times larger than N_{MOS} in the inverter?

The main idea is that ur rise time and fall time of your output voltage signal are the same. And for this the resistance of the nmos and pmos should be the same.

This can be achieved only by sizing the pmos 3 times to the nmos sizing.

For example, if u are designing an inverter or buffer and have a spec on the Duty Cycle, u will need to match ur rise and fall time as much as u can

And also the delay of the following stages depends on the rise and fall time of ur output signal and so it's some times beneficial to have symmetrical delays in order to increase the clock frequency

This scaling factor varies with the technology.

Sweep the input and watch the output with varying the scaling factor, the correct scaling factor is one which gives the threshold voltage of the inverter to be

odd/2.

To be exact, P_{mOS} should be 2.5 or 3 (if not 2.7) times larger than N_{mOS} because electron mobility is 2.7 faster than hole mobility. Larger is actually not a good one. In fact, it should be longer in Gate Width because only increasing the width decreases the resistance. And we can reduce the Gate Length, as this is determined by the CMOS technology used in the design.

Sizing affects rise time and fall time, as well as V_{TH} of CMOS inverter, but this is not yet the end of the story.

BTW, someone mentioned Noise margin.

CMOS has one undefined voltage band between upper and lower margin for logic 1 and 0 respectively. This is due to the negative gain, which is not an infinite gain, of the $V_{dN}-V_{SUT}$ curve. It is impossible to get an infinite gain that results such undefined voltage band.

The actual rise and fall time is still determined by the logic gates in the digital circuit. Having CMOS inverter nicely tuned is only the tip of an

iceberg.

These days, all CMOS inverters are based on minimum-sized NMOS transistors. By means of Lambda scale to Gate Length, this is easily done in Cadence tools.

Therefore focus should emphasize more on trying to balance the logic gates to ensure fast rise and fall times in various logic conditions so as to eliminate race conditions and static hazards.

asymmetrical rise and fall times leads to duty cycle variations which leads to jitter... And trust me clock jitter is hell...

one method is to look into the model and find out the mobilities.

the other (and the better) method is create an inverter and fine tune the ratios until you get perfect transition.. i mean for i/p of ' $\text{odd}/2$ ' u shud get an output of ' $\text{odd}/2$ '. This is the exact ratio of the mobilities...

of course this would change wrt process variations. but i

would say, one has to design for typical and ensure that it works more or less ok for other combinations...

μ_n need not be always greater than 3 times μ_p ... it entirely depends on the fab and technology.. so with change in fab, and change in technology ur basic or perfect inverter ratio changes..

Leakage control-----

Among the different leakage currents in the nanometer CMOS, the subthreshold and gate leakage are the most dominant. While the latter is mainly due to electron tunnelling from the gate to the substrate, the former is caused by many other factors. As a result, the leakage control techniques to be discussed will focus more on subthreshold currents. Over the years, many techniques have been developed to reduce the subthreshold currents in both the active and standby modes in order to minimize the total power consumption of CMOS circuits. While the standby leakage currents are wasted

currents when the circuit is in idle mode where no computation takes place, the active leakage currents are wasted current when the circuit is in use. Generally, reduction of leakage currents involves application of different device and circuit level techniques. At the device level, it involves controlling the doping profiles and physical dimensions of transistors while at the circuit level, it involves the manipulation of threshold voltage (V_{th}) and source biasing of the transistor.

A. Circuit Level Leakage Control Techniques

i) multi V_{th} Techniques

This technique involves fabrication of two types of transistors, high V_{th} and low V_{th} transistors, on a chip. The high V_{th} is used to lower the subthreshold leakage current, while the low V_{th} is used to enhance performance through faster operation. Obtaining these different types of transistors is done through controlled channel doping, different oxide thickness, multiple channel lengths or multiple body biases. Notwithstanding, with technology scaling and continuous decrease in the supply voltage, the implementation of the high V_{th} transistor will become a major practical challenge.

Dual threshold method

In logic circuits, leakage current can be reduced by assigning higher V_{th} to devices in non-critical paths, while

maintaining performance with low V_{th} in the critical paths. This technique is applicable to both standby and active mode leakage power dissipation control. It ensures that the circuit operates at a high speed and reduced power dissipation.

multi-Threshold Voltage method

This method uses a high V_{th} device to gate supply voltage from a low V_{th} logic block thereby creating a virtual power rail instead of directly connecting the block to the main power rail. The high V_{th} switches are used to disconnect the power supplies during the standby state, resulting in very low leakage currents set by the high V_{th} of the series logic block. In active mode operation, the high V_{th} transistors are switched on and the logic block, designed with low V_{th} , operates at fast speed.

This enables leakage current reduction via the high V_{th} and enhanced performance via the low V_{th} block.

Alternatively, this system could be implemented with a high V_{th} NMOS transistor connected between the GND and the low V_{th} block. The NMOS transistor insertion is preferred to the PMOS since it has a lower ON-resistance at the same width and consequently can be sized smaller. The use of these transistors increases circuit delay and area. Besides, to retain data during

standby mode, extra high V_{th} memory circuit is needed.

Variable V_{th} method

This is a method mainly used to reduce standby leakage currents by using a triple well process where the device V_{th} is dynamically adjusted by biasing the body terminal. Through application of maximum reverse biasing during the standby mode, V_{th} is increased and the subthreshold leakage current minimized. In addition, this method could be applied in active mode operation to optimize circuit performance by dynamically tuning the V_{th} based on workload requirements. Through this tuning capability, the circuit is able to operate at the minimal active leakage power.

Dynamic V_{th} method

This is a method used in active mode operation to control the leakage current in a circuit based on the desired frequency of operation. The frequency is dynamically adjusted through a back-gate bias in response to workload of a system. At low workload, increasing the V_{th} reduces the leakage power.

ii) Body Bias Control

Body biasing a transistor is an effective way of reducing both the active and standby leakage through its impact of increasing the threshold voltages of the mOS transistors. By applying a reverse body bias, the V_{th} is

increased and subsequently reduces the subthreshold leakage currents. This could be done during standby mode by applying a strong negative bias to the NMOS bulk and connecting the PMOS bulk to the VDD rail. Body biasing is also used to minimize DdBL effect and V_{th} -Rolloff associated with SC2. The Variable Threshold CMOS technique described above utilises body biasing to improve circuit performance. It is important to note that the V_{th} is related by the square root of the bias voltage implying that a significant voltage level would be needed to raise the V_{th} . This could be a potential challenge in the UDSm where the supply has been severely scaled down.

iii) minimum Leakage Vector method.

The fundamental concept in this technique is to force the combinational logic of the circuit into a low-leakage state during standby periods. This state enables the largest number of transistors to be turned off so as to reduce leakage and make use of multiple off transistors in stacks.

iv) Stack Effect-based method

The "stacking effect" is the reduction in subthreshold current when multiple transistors connected in series (in a stack) are turned off. The transistor stacking increases the source bias of the upper transistors in the

stack as well as lowers the gate-source voltage (V_{gs}) of these transistors. All these effects contribute to lower subthreshold leakage current in the circuit. Minimizing leakage through transistor stacking depends on the pattern of the input sequence during standby periods as it determines the number of OFF transistors in the stack.

Finding the low leakage input vector involves either a complete enumeration of the primary inputs or random search of the primary inputs. While the former is used for small circuits, the latter is applied for more complex circuits. The idea is to use the input vector to determine the combination that results to the least leakage current. When the input vector is obtained, the circuit is evaluated and if necessary, additional leakage control transistors are inserted in series at the non-critical paths to be turned OFF during the standby mode.

B. Device Level Leakage Control Techniques

Silicon-on-insulator (SOI): This is a non-bulk technology that builds transistors on top of insulating layer instead of a semiconductor substrate. Using insulating layer reduces parasitic capacitance, which results to higher operational speed and lower dynamic power dissipation in

integrated circuits. Though the early SOI used crystals like sapphire, emerging technologies favour the use of silicon wafer, making it economically attractive. The ITRS 2003 projects the use of Ultra-thin body (UTB) SOI by 2008 to manage the increasing effects of leakage.

Double Gate mOSFET (DG-mOS): In traditional bulk and SOI devices, immunity from SC2 like V_{th}-rolloff and DdBL requires increasing the channel doping to enable reduction of the depletion depth in the substrate. The inherent drawbacks to this approach are increased substrate-bias sensitivity and degraded subthreshold swing. By replacing the substrate with another gate to form a double gate mOSFET, short channel immunity is achieved with an ideal subthreshold swing.

Separation by Implantation of Oxygen (SIMOX): This is a more modern and elegant technique for making the SOI structure by implanting heavy doses of oxygen directly into a silicon substrate. The wafer is then annealed at very high temperatures, which induces oxide growth below the wafer surface and pushes a top layer of silicon on the top. The resulting SOI consumes lesser power than the bulk technologies. Other methods used in device level control include retrograde doping and halo

doping.

In addition to the two techniques discussed above, system and architectural level techniques are also used in leakage reduction. This technique could involve designing the system architecture so that it operates at low voltage. The underlining strategy is that when the system operates at low voltage, it reduces both the static and dynamic power consumption and consequently minimizes the leakage power. One of the ways of doing this is to design the system using pipeline architecture. With pipelining, it is possible to operate the system at lower voltage without performance degradation.

The penalty for this technique is extra hardware required for pipelining. Another method is threshold voltage hopping. This involves the use of software to dynamically control the threshold voltage of transistors based on the workloads of the system. By adjusting the threshold voltage in this way, high percentage power savings could be realised in a system. Furthermore, reduction in supply voltage is also a good technique to reduce leakage power. By lowering supply voltage, the source-drain voltage is reduced. This has the effect of minimizing DsBL, gate and subthreshold leakage.

currents.

Re: Details of Crosstalk delay and Crosstalk noise
If two signal or clock nets are closer enough to each other, the effect of coupling capacitance between them leads to crosstalk.

Crosstalk delay: Let's say one net is switching at faster rate (time taken for signal level to rise from 0 to 1 is less) and other net switching at slower rate (time taken for signal level to rise from 0 to 1 is less), due to coupling cap...faster switching net effects in speeding up of slower net.

Let say one net is switching at faster rate from 0 to 1 and other adjacent ramps down from 1 to 0 at slower rate. The effect of first net will delay ramping down of slower net further.

Think of these kind of situations on the chip for crosstalk delays....I will not elaborate further on this...just imagine...

Crosstalk noise: When one net is idle and its adjacent net is switching from 0 to 1, there is possibility of unwanted signal transition for finite time induced in idle

net due to coupling cap. If this unwanted transition is within noise margin of your technology then it will corrupt the logics.

Remedies to encounter these two effects:

1. move interacting nets apart in layout. (If your layout is not congested)
2. Shield the nets using proper shielding metal
3. Re-route the interacting nets.

- 1) double spacing
- 2) reduce parallelism
- 3) switch to other metal layer
- 4) insert buffer in victim net
- 5) upsize driver of victim
- 6) downsize driver of aggressor
- 7) shielding the victim net

You can add to this list....if you have any more remedies....

Crosstalk delay analysis: PT-SDF tool does this based on timing constraints. It needs SPDF and incremental SDF to do this. I can elaborate on this...let me know if you want

orsuvenkataKrishniah

to...

Crosstalk noise analysis: I believe libs should have noise models to do this. Don't have much idea on this. Please let me know if you know something on this.

Inputs: PT-Sd license, SPZF and inc SDF, Proper timing constraints, noise models.

Outputs: Violating timing paths.

Advantage of latch and flip flop

Latches are asynchronous, which means that the output changes very soon after the input changes. Most computers today, on the other hand, are synchronous, which means that the outputs of all the sequential circuits change simultaneously to the rhythm of a global clock signal.

A flip-flop is a synchronous version of the latch. To complicate the situation even more, there are several fundamental types of flip-flops. Here, we shall only consider a type called master-slave flip-flop.

Latch is a level sensitive device while flip-flop is an edge sensitive device.

Latch is sensitive to glitches on enable pin, whereas

flip-flop is immune to glitches.

Latches take less gates (also less power) to implement than flip-flops.

Latches are faster than flip-flops.

FF in counter, shift register

APPLICATION OF S-R LATCH

Digital Logic + Design

Digital systems use switches to input values and to control the output. For example, a Keypad uses 10 switches to enter decimal numbers 0 to 9. When a switch is closed the switch contacts physically vibrate or 'bounce' before making a solid contact. The switch bounce causes the voltage at the output of the switch to vary between logic low and high for a very short duration before it settles to a steady state. Figure 23.1a. The variation in the voltage causes the digital circuit to operate in an erratic manner. An S-R latch connected between the switch and the digital circuit prevents the varying switch output from reaching the digital circuit.

Figure 23.1b.

Latches are also put the value in a register, along with enable.

BiCMOS

Advantages:

Higher output currents than CMOS transistors of equal input capacitance.

Low logical effort.

Good for driving large capacitive loads.

Difficulties

BiCMOS may never offer the (relatively) low power consumption of CMOS alone. But since CMOS is already ideal for pure digital logic, this is only a serious issue where it is desirable to put logic circuits together on the same chip with other circuits that are not strictly logic: either for the purpose of a mixed-signal application, or simply to reduce the chip count in an electronic product by combining two chips into one, in order to reduce cost, size, and/or

weight.

BiCmBS as a fabrication process is not currently as commercially viable for some applications, such as microprocessors, as either BGT or CmBS fabrication. Unfortunately, many of the improvements to CmBS fabrication, for example, do not transfer directly to BiCmBS fabrication. An inherent difficulty arises from the fact that fine tuning of both the BGT and mBS components of the process is impossible without adding many extra fabrication steps and consequently increasing the process cost.

metastability [Wd]

Whenever there are setup and hold time violations in any flip-flop, it enters a state where its output is unpredictable: this state is known as metastable state (quasi stable state); at the end of metastable state, the flip-flop settles down to either '1' or '0'. This whole process is known as metastability. In the figure below T_{SU} is the setup time and T_H is the hold time. Whenever the input signal D does not meet the T_{SU} and T_H of the given D flip-flop, metastability

occurs.

When a flip-flop is in metastable state, its output oscillate between '0' and '1' as shown in the figure below (here the flip-flop output settles down to '0') How long it takes to settle down, depends on the technology of the flip-flop.

What are the cases in which metastability occurs?

As we have seen that whenever setup and hold violation time occurs, metastability occurs, so we have to see when signals violate this timing requirement:

- When the input signal is an asynchronous signal.

When the clock skew/slew is too much (rise and fall time are more than the tolerable values).

When interfacing two domains operating at two different frequencies or at the same frequency but with different phase.

When the combinational delay is such that flip-flop data input changes in the critical window (setup/hold window). So how do I avoid metastability?

- In reality, one cannot avoid metastability and increased clock-to-Q delays in synchronizing asynchronous inputs, without the use of tricky self-timed circuits. So a

more appropriate question might be "How do we tolerate metastability?"

- * In the simplest case, designers can tolerate metastability by making sure the clock period is long enough to allow for the resolution of quasi-stable states and for the delay of whatever logic may be in the path to the next flip-flop. This approach, while simple, is rarely practical given the performance requirements of most modern designs.
- * The most common way to tolerate metastability is to add one or more successive synchronizing flip-flops to the synchronizer. This approach allows for an entire clock period (except for the setup time of the second flip-flop) for metastable events in the first synchronizing flip-flop to resolve themselves. This does, however, increase the latency in the synchronous logic's observation of input changes.

Clock gating

Clock tree consume more than 50 % of dynamic power.

The components of this power are:

- 1) Power consumed by combinatorial logic whose values are changing on each clock edge
- 2) Power consumed by flip-flops

and

- 3) The power consumed by the clock buffer tree in the design.

There are two types of clock gating styles available.

They are:

- 1) Latch-based clock gating
- 2) Latch-free clock gating.

Latch free clock gating

The latch-free clock gating style uses a simple AND or OR gate (depending on the edge on which flip-flops are triggered). Here if enable signal goes inactive in between the clock pulse or if it multiple times then gated clock output either can terminate prematurely or generate multiple clock pulses. This restriction makes the latch-free clock gating style inappropriate for our single-clock flip-flop based design.

Latch free clock gating

Latch based clock gating

The latch-based clock gating style adds a level-sensitive latch to the design to hold the enable signal from the active edge of the clock until the inactive edge of the clock. Since the latch captures the state of the

enable signal and holds it until the complete clock pulse has been generated, the enable signal need only be stable around the rising edge of the clock, just as in the traditional ungated design style.

Latch based clock gating

Specific clock gating cells are required in library to be utilized by the synthesis tools. Availability of clock gating cells and automatic insertion by the EDA tools makes it simpler method of low power technique. Advantage of this method is that clock gating does not require modifications to RTL description.

XOR gate using transistors

Transmission gate

Only size transistor $\rightarrow AB + A'B$

Using mUX

$$Y = S'DO + S'DI$$

Therefore put DO as A and DI as A'

And S as B.

[8 transistor] Therefore the mUX can be build using

transmission gate causing only 4 transistors and 4 transistor for inverting.

[12 transistor] mUX can be made as CMOS causing only 8 transistor + 4 inverting.

Using pass (NMOS) only 6 transistor.

[16 transistor] Using NAND gates

Causes 16 transistors

Q) Gates using Diode

AND gate

OR gate

NOT gate

XOR gate

Dynamic logic

Dynamic logic requires a minimum clock rate fast enough that the output state of each dynamic gate is used before it leaks out of the capacitance holding that state, during the part of the clock cycle that the output is not being actively driven.

Static logic has no minimum clock rate—the clock can

be paused indefinitely. While it may seem that doing nothing for long periods of time is not particularly useful, it leads to two advantages:

being able to pause a system at any time makes debugging and testing much easier, enabling techniques such as single stepping.

being able to run a system at extremely low clock rates allows low-power electronics to run longer on a given battery.

Advantages

Dynamic logic (properly designed) is over twice as fast as normal logic.

most electronics running at over 2 GHz these days uses dynamic logic, although some manufacturers, such as Intel, have completely switched to static logic to save on power[2]. However, dynamic logic too has techniques for reducing power consumption. A dynamic logic circuit running at $1/2$ voltage could consume $1/4$ the power of normal.

Smaller area

NAND gate using Dynamic logic.

The dynamic logic circuit requires two phases. The first

phase, when Clock is low, is called the setup phase or the precharge phase and the second phase, when Clock is high, is called the evaluation phase. In the setup phase, the output is driven high unconditionally (no matter the values of the inputs A and B). The capacitor, which represents the load capacitance of this gate, becomes charged. Because the transistor at the bottom is turned off, it is impossible for the output to be driven low during this phase.

During the evaluation phase, Clock is high. If A and B are also high, the output will be pulled low. Otherwise, the output stays high (due to the load capacitance).

Problem

Dynamic logics, this logic is subject to the charge-sharing problem

monotonicity \rightarrow solve by domino

In practical use, however, dynamic logic still greatly increases the number of transistors that are switching at any given time, which greatly increases power consumption over static CMOS.

Only non inverting

Domino logic

Dynamic logic has a few potential problems that static

logic does not. For example, if the clock speed is too slow, the output will decay too quickly to be of use.

A popular implementation is domino logic.

Domino logic is a CMOS-based evolution of the dynamic logic techniques which were based on either PMOS or NMOS transistors. It allows a rail-to-rail logic swing. It was developed to speed up circuits.

In Dynamic Logic, a problem arises when cascading one gate to the next. The precharge "1" state of the first gate may cause the second gate to discharge prematurely, before the first gate has reached its correct state. This uses up the "precharge" of the second gate, which cannot be restored until the next clock cycle, so there is no recovery from this error.

There are various solutions to the problem of how to cascade dynamic logic gates. One solution is Domino Logic, which inserts an ordinary static inverter between stages.

Important Domino Logic features include:

They have smaller areas than conventional CMOS logic (as does all Dynamic Logic).

Parasitic capacitances are smaller so that higher operating speeds are possible.

Operation is free of glitches as each gate can make only one

transition.

Only non-inverting structures are possible because of the presence of inverting buffer.

Charge distribution may be a problem.

General Domino logic

PAL

PAL devices consisted of a small PROM (programmable read-only memory) core and additional output logic used to implement particular desired logic functions with few components. March 1978

PAL devices have arrays of transistor cells arranged in a "fixed-OR, programmable-AND" plane used to implement "sum-of-products" binary logic equations for each of the outputs in terms of the inputs and either synchronous or asynchronous feedback from the outputs.

PLA

A programmable logic array (PLA) is a programmable

device used to implement combinational logic circuits. The PLA has a set of programmable AND gate planes, which link to a set of programmable OR gate planes, which can then be conditionally complemented to produce an output. This layout allows for a large number of logic functions to be synthesized in the sum of products (and sometimes product of sums) canonical forms.

PLDs

A programmable logic device or PLD is an electronic component used to build reconfigurable digital circuits. Unlike a logic gate, which has a fixed function, a PLD has an undefined function at the time of manufacture. Before the PLD can be used in a circuit it must be programmed, that is, reconfigured.

Before PLDs were invented, read-only memory (ROM) chips were used to create arbitrary combinational logic functions of a number of inputs.

PLD types can classified into the following groups

- PROMs (Programmable Read Only memory) - offer high speed and low cost for relatively small designs
- PLAs (Programmable Logic Array) - offer flexible

features for more complex designs

- PAL/GALs (Programmable Array Logic/Generic Array Logic) - offer good flexibility and are faster and less expensive than PLAs

FPGAs

A field-programmable gate array (FPGA) is an integrated circuit designed to be configured by the customer or designer after manufacturing—hence "field-programmable". The FPGA configuration is generally specified using a hardware description language (HDL), similar to that used for an application-specific integrated circuit (ASIC) (circuit diagrams were previously used to specify the configuration, as they were for ASICs, but this is increasingly rare).

microcode

- (1) The lowest-level instructions that directly control a microprocessor. A single machine-language instruction typically translates into several microcode instructions. In modern PC microprocessors, the microcode is

hardwired and can't be modified. Some RISC designs go one step further by completely eliminating the microcode level so that machine instructions directly control the processor. At the other end of the spectrum, some mainframe and minicomputer architectures utilize programmable microcode. In this case, the microcode is stored in EPROM, which can be modified. This is called microprogramming.

Microcode simplified the job by allowing much of the processor's behaviour and programming model to be defined via microprogram routines rather than by dedicated circuitry. Even late in the design process, microcode could easily be changed, whereas hard-wired CPU designs were very cumbersome to change, so this greatly facilitated CPU design.

Example Jump and Add, block memory transfer
Always used in CISC, but RISC don't use microcode.

Depletion-mode MOSFETs

There are depletion-mode MOSFET devices, which are less commonly used than the standard enhancement-mode devices already described. These are MOSFET

devices that are doped so that a channel exists even with zero voltage from gate to source. In order to control the channel, a negative voltage is applied to the gate (for an n-channel device), depleting the channel, which reduces the current flow through the device. In essence, the depletion-mode device is equivalent to a normally closed (on) switch, while the enhancement-mode device is equivalent to a normally open (off) switch.[1] Due to their low noise figure in the RF region, and better gain, these devices are often preferred to bipolars in RF front-ends such as in TV sets. Depletion mode is shown by solid lines.

Depletion mode

The depletion mode mOSFET can operate in a depletion mode, where a negative gate voltage increases the layer and hence reduces the source-drain current, or in an enhancement mode, where a positive gate voltage reduces the depletion layer and hence increases the source-drain current.

Leakage current

There are 3 types of leakage:

Higher subthreshold conduction

As mQSFET geometries shrink, the voltage that can be applied to the gate must be reduced to maintain reliability. To maintain performance, the threshold voltage of the mQSFET has to be reduced as well. As threshold voltage is reduced, the transistor cannot be switched from complete turn-off to complete turn-on with the limited voltage swing available; the circuit design is a compromise between strong current in the "on" case and low current in the "off" case, and the application determines whether to favor one over the other. Subthreshold leakage (including subthreshold conduction, gate-oxide leakage and reverse-biased junction leakage), which was ignored in the past, now can consume upwards of half of the total power consumption of modern high-performance VLSI.

chips.

Increased gate-oxide leakage

The gate oxide, which serves as insulator between the gate and channel, should be made as thin as possible to increase the channel conductivity and performance when the transistor is on and to reduce subthreshold leakage when the transistor is off. However, with current gate oxides with a thickness of around 1.2 nm (which in silicon is 5 atoms thick) the quantum mechanical phenomenon of electron tunneling occurs between the gate and channel leads to gate-oxide leakage, leading to increased power consumption.

Increased junction leakage

To make devices smaller, junction design has become more complex, leading to higher doping levels, shallower junctions, "halo" doping and so forth,[27][28] all to decrease drain-induced barrier lowering (see the section on junction design). To keep these complex junctions in place, the annealing steps formerly used to remove damage and electrically active defects must be curtailed[29] increasing junction leakage. Heavier doping is also associated with thinner depletion layers and more recombination centers that result in increased leakage current, even without lattice

damage.

Inverter layout

$$L_{min} = 2 * \lambda$$

180nm then $L_{min} = 180\text{nm}$ and $\lambda = 90\text{nm}$,

Typically we use P substrate

Tunneling

The gate oxide, which serves as insulator between the gate and channel, should be made as thin as possible to increase the channel conductivity and performance when the transistor is on and to reduce subthreshold leakage when the transistor is off. However, with current gate oxides with a thickness of around 1.2 nm (which in silicon is 5 atoms thick) the quantum mechanical phenomenon of electron tunneling occurs between the gate and channel, leading to increased power consumption.

All else equal, a higher dielectric thickness reduces the quantum tunneling current through the dielectric between the gate and the

channel

2lmore delay

2lmore delay is a simple approximation to the delay through an RC network in an electronic system. It is often used in applications such as logic synthesis, delay calculation, static timing analysis, placement and routing, since it is simple to compute (especially in tree structured networks, which are the vast majority of signal nets within FPGAs) and is reasonably accurate.

2lmore delay[2] is a simple approximation, often used where speed of calculation is important but the delay through the wire itself cannot be ignored. It uses the R and C values of the wire segments in a simple calculation. The delay of each wire segment is the R of that segment times the downstream C. Then all delays are summed from the root. (This assumes the network is tree structured, true of most nets in chips. In this case the 2lmore delay can be calculated in time $O(N)$ with two tree traversals. If the network is not tree structured the 2lmore delay can still be computed, but

involves matrix calculations.)

What is multicycle path and false path?

A multicycle path is one where the generation of valid output takes more than one clock cycle. ex: pipelined FullR filter.

multicycle paths Usually false and multicycle paths comes in STA. An STA analyzer analyzes any path in one clock cycle. If suppose the path takes more than one cycle then we must specify the tool that how many extra cycles we must provide to analyze that path.

False Paths There may be some paths in your design where the data does not traverse. But STA analyzer will not be knowing and hence it may be analyzing that path. Hence we make that path as false path so that STA should not analyze it. Normal signals flow never pass through. false paths is the path that you don't care timing.

What is "Scan"

?

§ Scan Insertion and ATPG helps test ASiCs (e.g. chips) during manufacture. If you know what QTAG boundary scan is, then Scan is the same idea except that it is done inside the chip instead of on the entire board. Scan tests for defects in the chip's circuitry after it is manufactured (e.g. Scan does not help you test whether your Design functions as intended). ASiC designers usually implement the scan themselves and occurs just after synthesis. ATPG (Automated Test Pattern Generation) refers to the creation of "Test Vectors" that the Scan circuitry enables to be introduced into the chip. Here's a brief summary:

- Scan Insertion is done by a tool and results in all (or most) of your design's flip-flops to be replaced by special "Scan Flip-flops". Scan flops have additional inputs/outputs that allow them to be configured into a "chain" (e.g. a big shift register) when the chip is put into a test mode.
- The Scan flip-flops are connected up into a chain (perhaps multiple

chains)

- The ATPG tool, which knows about the scan chain you've created, generates a series of test vectors.
- The ATPG test vectors include both "Stimulus" and "Expected" bit patterns. These bit vectors are shifted into the chip on the scan chains, and the chip's reaction to the stimulus is shifted back out again.
- The AT2 (Automated Test Equipment) at the chip factory can put the chip into the scan test mode, and apply the test vectors. If any vectors do not match, then the chip is defective and it is thrown away.
- Scan/ATPG tools will strive to maximize the "coverage" of the ATPG vectors. In other words, given some measure of the total number of nodes in the chip that could be faulty (shorted, grounded, "stuck at 1", "stuck at 0"), what percentage of them can be detected with the ATPG vectors? Scan is a good technology and can achieve high coverage in the 90% range.
- Scan testing does not solve all test problems. Scan testing typically does not test memories (no flip-flops!).

needs a gate-level netlist to work with, and can take a long time to run on the AT2.

- FPGA designers may be unfamiliar with scan since FPGA testing has already been done by the FPGA manufacturer. ASoC designers do not have this luxury and must handle all the manufacturing test details themselves.

Standard cell library

A typical standard-cell library contains two main components:

Timing Abstract - This is generally in the Synopsys Liberty format, and provides functional definitions, timing, power, and noise information for each cell.

Layout Abstract - Common formats that are in use are the Cadence L2F format, and the Synopsys milkyway format, which contain reduced information about the cell layouts, sufficient for automated "Place and Route" tools.

A standard-cell library may also contain the following additional

components:

A full layout of the cells

Spice models of the cells

Verilog models or VHDL Vital models

Parasitic Extraction models

DRC rule decks

Re: clock-skew

Definition

clock skew is the difference in the arrival times of clock at the clock pin of the flops.

- early arrival (positive skew) can cause set up violation and
- late arrival (negative skew) can cause hold violation.

basically at every next clock there should be transfer from one flop to another flop. that difference value is called as slack accordingly positive n negative.....for a design to run correctly there should be positive slack taking care of the margins as well.

Clock skew is difference from highest insertion delay and lowest insertion delay of clock tree.

positive skew ---helps to meet the setup

negative skew---helps to meet the hold

clock balancing is done to reduce skew.... that is to

orsuvenkataKrishniah

divide unequal delays in data paths made to almost equal so that operating frequency can be fixed to higher value

You should build the clock-tree better to increasing clock-skew. Don't separate the clock tree if possible! Or you can search CTS distribution from this forum for your details.

If you are not meeting skew then your cts tool is crap and you will need to manually fix the skew by adding or deleting buffers. Either way it is not something you want to do manual.

So, if we increase the positive skew, we can avoid the setup violation with respect to the maximum delay path. But this may affect the hold requirement

it is overcome by using various routing techniques. positive skew can improve the performance of your circuit by increasing the clock frequency... however the circuit can become susceptible to race condition...i.e it might result in hold time violation...

fixing setup violation using clock skew

Reduce data path delay as much as possible. There are few techniques for that like

Up/down sizing cells

Adding/removing

orsuvenkatakrishniah

buffers

Changing placement of cells (except F/F after CTS)

Decreasing crosstalk delay by spacing/widening routing

Re: hi plz can u tell the differe for 180 nm and 90 nm techn

yes it is the gate length

that is it is the minimum length of the channel it can be fabricated.

When we r talking about 180nm or 90 nm or any nm then it means that we r talking about the feature size so 90 nm meter the min size of the length of the transistor in that device is 90nm..

gate delay $\tau = C \Delta V$
(ΔV is the Voltage swing)

to decrease gate delay (τ)

a) decrease C - reduce output loading (fanout) newer

process like 45nm etc ...

b) decrease ΔV - reduce voltage swing etc ...

c) increase λ - bigger transistors (bigger W) smaller L etc ...

note that reducing V_{dd} than process nominal V_{dd} increases gate delay becoz

- ΔV decreases proportional to V_{dd} hence tries to decrease τ

- λ decreases proportional to $(V_{dd} - V_t)$ a where a is between 1 and 2 hence tries to increase τ the combined effect will increase τ i.e higher gate delay.

Difference between testing and verification

Testing: A manufacturing step that ensures that the physical device manufactured from the synthesized design, has no manufacturing defect.

Testing is same as Validation where you just verify the performance of the system without actually getting down to the

subsystems.

Verification: Predictive analysis to ensure that the synthesized design, when manufactured, will perform the given I/O function.

Verification is process that improves the confidence in Validation by analysing the circuits down to the transistor level.

Verification takes place before validation

1. Verification: A method which ensures the formal proofs of correctness in a DC.

2. Validation:/testing: A method which improves the confidence in the correctness, such as Simulation.

Verification is the Offchip checking of correctness of the design

Validation is the Onchip checking of the correctness of the design

Testing is provided after verification.

testing is related to applying test patterns to the chip and verifying its function, for this the chip is designed based on DFT, BIST etc. whereas verification is done

at hdI or schematic level using simulators applying test vectors through test benches

Resistance of interconnect:

$$R = (k \cdot \text{length}) / (L \cdot w)$$

Capacitance is directly proportional to $w \cdot l$

Wire capacitance add delay by loading

While long wire add delay bcs of resistance.

What is the effect of delay, rise and fall times with increase in load capacitance?

Everything increases.

Switching power in CMOS

Multiply by the switching frequency on the load

capacitances to get the current used, and multiply by voltage again to get the characteristic switching power dissipated by a CMOS device: $P = CV^2f$

Delay = $d = f + p$

Where The stage effort is then simply: $f =$

gh

G is logical effort and h is load

And p is parasitic delay

Let A + B be two inputs of the NAND gate. Say signal A arrives at the NAND gate later than signal B. To optimize delay, of the two series NmQS inputs A + B, which one would you place near the output? Why?

Solution B should be at bottom closer to ground and A should be closer to Output, then 2 more delay is simply ωRC ,

In otherwise, delay becomes $7RC$.

Why the size of inverters in buffer design gradually increased? Why not give the output of a circuit to one large inverter?

that concept is called as Gate Sizing.

Gate capacitance.

One large transistor pair would have a large gate capacitance, and the weakly driven signal would have trouble driving it. This is why we use a chain of devices with increasing dimensions.

So the best practice is to increase buffer size gradually,

$\times 4$ at a stage. Books on logic effort proves this. Because it can not drive the output load straight away, so we gradually increase the size to get an optimized performance.

The concept is that crosstalk is caused by coupling capacitance of two wire running parallel for long distances. If the parallel run is broken by inserting buffer(s), which reduces coupling capacitance, then the crosstalk is reduced.

ABd

Logic operations

ABd gates perform one or more AND operations followed by a OR operation and then an inversion.

Construction of ABd cells is particularly efficient using CMOS technology where the total number of transistor gates can be compared to the same construction using NAND logic or NOR logic.

For example, a 2-1 ABd gate can be constructed with 6 transistors in CMOS compared to 10 transistors using a

2-input NAND gate (4 transistors), an inverter (2 transistors), and a 2-input NOR gate (4 transistors).

For PMOS

$$V_{tp} = -0.7 \text{ V}$$

NOT S and D should be interchanged.

Q) What is a Diode Clamping Circuit?

What is a Diode Clamping Circuit?

A diode clamping circuit consists of a combination of one or more diodes and series resistors. It is used to control the amount of input voltage to a sensitive circuit.

Significance

Perhaps you have a circuit you wish to protect from voltage spikes due to static electricity. Alternatively, you may have a circuit where you wish to limit the voltage to a particular input or output range, in order to either guard against accidental destruction or to make it compatible with a circuit with a lower voltage. The addition of a diode clamp helps solve problems such as these where you need to constrain the amount of voltage in a circuit.

Digital logic circuits nearly always feature the use of

diode clamps to provide protection to their inputs.

The clamping diodes on the inputs are to offer some protection against ESD as shown in figure:

Electrostatic discharge (ESD) is the sudden and momentary electric current that flows between two objects at different electrical potentials caused by direct contact or induced by an electrostatic field.[1]

The term is usually used in the electronics and other industries to describe momentary unwanted currents that may cause damage to electronic equipment.

A tri-state buffer is a useful device that allows us to control when current passes through the device, and when it doesn't.

Here's two diagrams of the tri-state buffer.

A tri-state buffer has two inputs: a data input x and a control input c . The control input acts like a value. When the control input is active, the output is the input. That is, it behaves just like a normal buffer. The "value" is open.

When the control input is not active, the output is "Z".

The "value" is open, and no electrical current flows through. Thus, even if x is 0 or 1, that value does not flow through.

Three-state outputs are implemented in many registers, bus drivers, and flip-flops in the 7400 and 4000 series as well as in other types, but also internally in many integrated circuits. Other typical uses are internal and external buses in microprocessors, memories, and peripherals. Many devices are controlled by an active-low input called OE (Output Enable) which dictates whether the outputs should be held in a high-impedance state or drive their respective loads (to either 0- or 1-level).

Keeper

Dynamic circuit suffers from charge leakage as it is charged high in precharge high and left floating.

Increase Noise margin \rightarrow bcs when input is slightly greater than V_t still the output is kept still high bcs of keeper.

Re: what is ground bounce and how to eliminate it?

ground bounce means the OV power net is not OV in your chip/board, some noise on your power net.
using de-coupling cell in your

chip

Ground bounce is due to rapid current sink/source thru bonding wire between ground pad and package pin. $dV = L(dI/dt)$

SO the methods to mitigate this ground bouncing effect are:

(1) lower the inductance(L):

for example; double/triple bonding for ground pin. or using flip-chip packaging, etc.

(2) lower the current spike(dI/dt):

for example; using slew-rate control output buffer, decoupling cap in internal power-ground net, well controlled slew-rate of internal signal with adequate buffer size, etc.

The problem is cause by the large current flow through the ground pin which

develops a voltage drop over the lead inductance. This voltage drop on the ground line creates two main problems; first it rises the chip off ground [0 volts] potential which increases

the devices input threshold level, and increases the voltage level on an output pin which is not switching.

Static hazards

A static hazard is the situation where, when one input variable changes, the output changes momentarily before stabilizing to the correct value. There are two types of static hazards:

Static-1 Hazard: the output is currently 1 and after the inputs change, the output momentarily changes to 0 before settling on 1

Static-0 Hazard: the output is currently 0 and after the inputs change, the output momentarily changes to 1 before settling on 0

In properly formed two-level AND-OR logic based on a Sum Of Products expression, there will be no static-0 hazards. Conversely, there will be no static-1 hazards in an OR-AND implementation of a Product Of Sums expression.

Dynamic hazards

A dynamic hazard is the possibility of an output changing more than once as a result of a single input change.

Dynamic hazards often occur in larger logic circuits where there are different routes to the output (from the input). If each route has a different delay, then it

quickly becomes clear that there is the potential for changing output values that differ from the required / expected output. e.g. A logic circuit is meant to change output state from 1 to 0, but instead changes from 1 to 0 then 1 and finally rests at the correct value 0. This is a dynamic hazard.

As a rule, dynamic hazards are more complex to resolve, but note that if all static hazards have been eliminated from a circuit, then dynamic hazards cannot occur.

What are Synchronous and Asynchronous?

Synchronous and asynchronous transmissions are two different methods used in digital systems for synchronization of transmitter and receiver during data transfer. Synchronous transmissions are synchronized by an external clock, while asynchronous transmissions are synchronized by special signals along the transmission medium.

What is the Need of Synchronization?

Whenever an electronic device transmits digital (and sometimes analog) data to another electronic device, there must be a certain rhythm established between the two devices, i.e., the receiving device must have some way of knowing, within the context of the fluctuating

signal that it's receiving, where each unit of data begins and where it ends.

For example, a television transmitter produces a continuous stream of data in which each horizontal line of image must be distinguishable from the preceding and succeeding lines, so that a TV will be able to distinguish between them upon reception.

So the signal must be synchronized in a way that the receiver can understand the data format and receive the data correctly from the transmitter.

What are the Advantages of Asynchronous data transfer?

Simple. It doesn't require synchronization of both communication sides

Cheaper. Timing is not as critical as for synchronous transmission, therefore hardware can be made cheaper. Set-up is very fast, so well suited for applications where messages are generated at irregular intervals, for example data entry from the Keyboard.

What are the Limitations of Asynchronous data transfer?

Large relative overhead, a high proportion of the transmitted bits are uniquely for control purposes and thus carry no useful information

What are the Advantages of Synchronous data

transfer?

Lower overhead and thus, greater throughput

Synchronous designs eliminate the problems associated with speed variations through different paths of logic. By sampling signals at well-defined time intervals, fast paths and slow paths can be handled in a simple manner.

Synchronous designs work well under variations of temperature, voltage and process. This stability is key for high-volume manufacturing.

many designs must be portable—that is, they must be easy to migrate to a new and improved technology (say, moving from 6 micron to 3.5 micron). The deterministic behavior of synchronous designs makes them much more straightforward to move to a new technology.

Interfacing between two blocks of logic is simplified by defining standardized synchronous behavior. Asynchronous interfaces demand elaborate handshaking or token passing to ensure integrity of information; synchronous designs with known timing characteristics can guarantee correct reception of data.

What are the Limitations of Synchronous data transfer?

Slightly more complex

Hardware is more

expensive

Latch is sensitive to input whereas flip-flop is sensitive to clock transition.

Latch is a (clock/enable) level sensitive device while flip-flop is an clock edge sensitive device.

Latch is sensitive to glitches on enable pin, whereas flip-flop is immune to glitches.

Latches take less gates (also less power) to implement than flip-flops.

Latches are faster than flip-flops.

Latch Vs Flip-Flop

Both latches and flip-flops are circuit elements whose output depends not only on the present inputs, but also on previous inputs and outputs.

They both are hence referred as "sequential" elements.

In electronics, a latch, is a kind of bistable multivibrator, an electronic circuit which has two stable states and thereby can store one bit of information.

Today the word is mainly used for simple transparent storage elements, while slightly more advanced non-transparent (or clocked) devices are described as flip-flops. Informally, as this distinction is quite new, the

two words are sometimes used interchangeably.

In digital circuits, a flip-flop is a kind of bistable multi-vibrator, an electronic circuit which has two stable states and thereby is capable of serving as one bit of memory. Today, the term flip-flop has come to generally denote non-transparent (clocked or edge-triggered) devices, while the simpler transparent ones are often referred to as latches.

A flip-flop is controlled by (usually) one or two control signals and/or a gate or clock signal.

Latches are level sensitive i.e. the output captures the input when the clock signal is high, so as long as the clock is logic 1, the output can change if the input also changes.

Flip-Flops are edge sensitive i.e. flip flop will store the input only when there is a rising or falling edge of the clock.

A positive level latch is transparent to the positive level(enable), and it latches the final input before it is changing its level(i.e. before enable goes to '0' or before the clock goes to -ve level.)

A positive edge flop will have its output effective when the clock input changes from '0' to '1' state ('1' to '0' for negative edge flop) only.

Latches are faster, flip flops are

slower.

Latch is sensitive to glitches on enable pin, whereas flip-flop is immune to glitches.

Latches take less gates (less power) to implement than flip-flops.

D-FF is built from two latches. They are in master slave configuration.

Latch may be clocked or clock less. But flip flop is always clocked.

For a transparent latch generally D to Q propagation delay is considered while for a flop clock to Q and setup and hold time are very important.

Synthesis perspective: Pros and Cons of Latches and Flip-Flops

In synthesis of HDL codes inappropriate coding can infer latches instead of flip flops. Eg. "if" and "case" statements. This should be avoided as latches are more prone to glitches.

Latch takes less area, Flip-flop takes more area (as flip flop is made up of latches).

Latch facilitate time borrowing or cycle stealing whereas flip flops allow synchronous logic.

Latches are not friendly with DFT tools. minimize inferring of latches if your design has to be made testable. Since enable signal to latch is not a regular

clock that is fed to the rest of the logic. To ensure testability, you need to use OR gate using "enable" and "scan enable" signals as input and feed the output to the enable port of the latch.

most EDA software tools have difficulty with latches. Static timing analyzers typically make assumptions about latch transparency. If one assumes the latch is transparent (i.e. triggered by the active time of clock, not triggered by just clock edge), then the tool may find a false timing path through the input data pin. If one assumes the latch is not transparent, then the tool may miss a critical path.

If target technology supports a latch cell then race condition problems are minimized. If target technology does not support a latch then synthesis tool will infer it by basic gates which are prone to race condition. Then you need to add redundant logic to overcome this problem. But while optimization redundant logic can be removed by the synthesis tool! This will create endless problems for the design team.

Due to the transparency issue, latches are difficult to test. For scan testing, they are often replaced by a latch-flip-flop compatible with the scan-test shift-register. Under these conditions, a flip-flop would actually be less expensive than a

latch.

Flip flops are friendly with DFT tools. Scan insertion for synchronous logic is hassle free.

Band Gap

Band Gap in semiconductor of 2eV which get reduce to 0.045eV after adding impurity (0.001% which increases 10^{17} electrons).

Fermi level: It is the energy level in between of Covalent Band to Valence Band. Either, to change the energy level band from covalent to valence (or) valence to covalent band.

Q8) How can you convert an SR Flip-flop to a JK Flip-flop?

By giving the feed back we can convert, i.e. $Q \Rightarrow S$ and $Q \Rightarrow R$. Hence the S and R inputs will act as J and K

respectively.

29) How can you convert the JK Flip-flop to a D Flip-flop?

By connecting the J input to the K through the inverter.

30) What is Race-around problem? How can you rectify it?

The clock pulse that remains in the 1 state while both J and K are equal to 1 will cause the output to complement again and repeat complementing until the pulse goes back to 0, this is called the race around problem. To avoid this undesirable operation, the clock pulse must have a time duration that is shorter than the propagation delay time of the F-F, this is restrictive so the alternative is master-slave or edge-triggered construction.

34) Guys this is the basic question asked most frequently. Design all the basic gates (NOT, AND, OR, NAND, NOR, XOR, XNOR) using 2:1 multiplexer?

Using 2:1 mux, (2 inputs, 1 output and a select line)
(a)

(a) NOT

Give the input at the select line and connect $\bar{A}O$ to $I + \bar{A}I$ to O . So if A is 1, we will get \bar{A} that is 0 at the Q/P.

(b) AND

Give input A at the select line and O to $\bar{A}O$ and B to $\bar{A}I$. Q/P is $A + B$

(c) OR

Give input A at the select line and I to $\bar{A}I$ and B to $\bar{A}O$.

Q/P will be $A + B$

(d) NAND

AND + NOT implementations together

(e) NOR

OR + NOT implementations together

(f) XOR

A at the select line B at $\bar{A}O$ and B at $\bar{A}I$. B can be obtained from (a) (g) XNOR

A at the select line B at $\bar{A}I$ and B at $\bar{A}O$

40) In a 3-bit Johnson's counter what are the unused states?

$2^L - 2^n$ is the one used to find the unused states in Johnson counter.

So for a 3-bit counter it is $8 - 6 = 2$. Unused states = 2. The two unused states are 010 and

101

40) In a 3-bit Johnson's counter what are the unused states?

$2^n - 2$ is the one used to find the unused states in Johnson counter.

So for a 3-bit counter it is $8 - 6 = 2$. Unused states = 2.
the two unused states are 010 and 101

38) How will you implement a Full subtractor from a Full adder?

All the bits of subtrahend should be connected to the xor gate. Other input to the xor being one. The input carry bit to the full adder should be made 1. Then the full adder works like a full subtractor

51) Design a FSM (Finite State machine) to detect a sequence 10110?

52) Convert D-FF into divide by 2. (not

latch)?

53) What is the max clock frequency of the circuit , given the following information?

$$T_{\text{setup}} = 6 \text{ ns}$$

$$T_{\text{hold}} = 2 \text{ ns}$$

$$T_{\text{propagation}} = 10 \text{ ns}$$

Circuit:

Connect \bar{Q} to D and apply the clk at clk of DFF and take the Q/P at Q. It gives freq/2.

max. Freq of operation:

$$1 / (T_{\text{propagation}} + T_{\text{setup}}) = 1 / 16 \text{ ns} = 62.5 \text{ MHz}$$

54) For the Circuit Shown below, What is the maximum Frequency of Operation? Are there any hold time violations for FF2? If yes, how do you modify the circuit to avoid them?

$$\text{The minimum time period} = 3 + 2 + (1+1+1) = 8 \text{ ns}$$

maximum Frequency = $1/8n = 125\text{mHz}$.

And there is a hold time violation in the circuit, because of feedback, if you observe, $t_{cq2} + \text{AND gate delay}$ is less than t_{hold2} . To avoid this we need to use even number of inverters(buffers). Here we need to use 2 inverters each with a delay of 1ns . then the hold time value exactly meets.

55) Design a D-latch using (a) using 2:1 mux (b) from S-R Latch ?

$$\text{slack} = R\bar{T} - A\bar{T}$$

if $A\bar{T}$ is greater than $R\bar{T}$ then negative slack occur

S
.....

simply put, it means that the timing constraints that you have given for your design, are not met.

mostly Asked Questions:-

1/ What is latch up?

Latch-up pertains to a failure mechanism wherein a parasitic thyristor (such as a parasitic silicon controlled rectifier, or SCR) is inadvertently created within a

circuit, causing a high amount of current to continuously flow through it once it is accidentally triggered or turned on. Depending on the circuits involved, the amount of current flow produced by this mechanism can be large enough to result in permanent destruction of the device due to electrical overstress (EOS).

2) Why is NAND gate preferred over NOR gate for fabrication?

NAND is a better gate for design than NOR because at the transistor level the mobility of electrons is normally three times that of holes compared to NOR and thus the NAND is a faster gate.

Additionally, the gate-leakage in NAND structures is much lower. If you consider t_{phl} and t_{plh} delays you will find that it is more symmetric in case of NAND (the delay profile), but for NOR, one delay is much higher than the other (obviously t_{plh} is higher since the higher resistance p mos's are in series connection which again increases the resistance).

3) What is Noise margin? Explain the procedure to determine Noise margin

The minimum amount of noise that can be allowed on the input stage for which the output will not be

effected.

4) Explain sizing of the inverter?

In order to drive the desired load capacitance we have to increase the size (width) of the inverters to get an optimized performance.

5) How do you size NMOS and PMOS transistors to increase the threshold voltage?

(e) What is Noise margin? Explain the procedure to determine Noise margin?

The minimum amount of noise that can be allowed on the input stage for which the output will not be effected.

7) What happens to delay if you increase load capacitance?

delay increases.

[What happens to delay if we include a resistance at the](http://www.edaboard.com/images/smilies/icon_cool.gif)

output of a CMOS circuit?

Increases. (RC delay)

Q) What are the limitations in increasing the power supply to reduce delay?

The delay can be reduced by increasing the power supply but if we do so the heating effect comes because of excessive power, to compensate this we have to increase the die size which is not practical.

10) How does Resistance of the metal lines vary with increasing thickness and increasing length?

$$R = L \cdot \frac{A}{t}$$

11) For CMOS logic, give the various techniques you know to minimize power consumption?

Power dissipation = CV^2f from this minimize the load capacitance, dc voltage and the operating frequency.

12) What is Charge Sharing? Explain the Charge Sharing problem while sampling data from a Bus?

In the serially connected NMOS logic the input capacitance of each gate shares the charge with the load capacitance by which the logical levels drastically mismatched than that of the desired one. To eliminate

this load capacitance must be very high compared to the input capacitance of the gates (approximately 10 times).

13) Why do we gradually increase the size of inverters in buffer design? Why not give the output of a circuit to one large inverter?

Because it can not drive the output load straight away, so we gradually increase the size to get an optimized performance.

14) What is Latch Up? Explain Latch Up with cross section of a CMOS inverter. How do you avoid Latch Up?

Latch-up is a condition in which the parasitic components give rise to the establishment of low resistance conducting path between VDD and VSS with disastrous results.

15) Give the expression for CMOS switching power

dissipation?

CVD

16) What is Body Effect?

In general multiple MOS devices are made on a common substrate. As a result, the substrate voltage of all devices is normally equal. However while connecting the devices serially this may result in an increase in source-to-substrate voltage as we proceed vertically along the series chain ($V_{Sb1}=0$, $V_{Sb2} > 0$). Which results $V_{th2} > V_{th1}$.

17) Why is the substrate in NMOS connected to Ground and in PMOS to VDD?

We try to reverse bias not the channel and the substrate but we try to maintain the drain,source junctions reverse biased with respect to the substrate so that we don't lose our current into the substrate.

http://www.edaboard.com/images/smilies/icon_cool.gif

What is the fundamental difference between a MOSFET and BJT?

In MOSFET, current flow is either due to