1. Difference between unary operators and binary operators in verilog?
Ans:-unary operators are used preceded to the operands where as the binary operators are used in between the operands.
2. What is the significance of defparam in verilog?
Ans: - Parameter values can be changed in any module instance in the design with the keyword `defparam`. The hierarchical name of the module instance can be used to override parameter values. (Ref:-chapter-9.2 of palnithkar)
3. Differance between Reduction and Bitwise operators?
Ans: - The difference is that bitwise operations are on bits from two different operands, whereas reduction operations are on the bits of the same operand. Reduction operators work bit by bit from right to left. Reduction operators perform a bitwise operation on a single vector operand and yield a 1-bit result Bitwise operators perform a bit-by-bit operation on two operands. They take each bit in one operand and perform the operation with the corresponding bit in the other operand.

Verilog Answer 1

Q: What is the difference between a Verilog *task* and a Verilog *function*?

A:The following rules distinguish tasks from functions:

- A **function** shall execute in one simulation time unit;
  a **task** can contain time-controlling statements.
- A **function** cannot enable a task;
  a **task** can enable other tasks or functions.
- A **function** shall have at least one input type argument and shall **not** have an output or inout type argument;
  a **task** can have zero or more arguments of any type.
- A **function** shall return a single value;
  a **task** shall not return a value.

Q: Given the following Verilog code, what value of "a" is displayed?

```
always @(clk) begin
   a  = 0;
   a <= 1;
   $display(a);
end
```

A:This is a tricky one! Verilog scheduling semantics basically imply a four-level deep queue for the current simulation time:

```
   1: Active Events                 (blocking statements)
   2: Inactive Events               (#0 delays, etc)
   3: Non-Blocking Assign Updates   (non-blocking statements)
   4: Monitor Events                ($display, $monitor, etc).
```

Since the "**a = 0**" is an active event, it is scheduled into the 1st "queue".

The "**a <= 1**" is a non-blocking event, so it's placed into the 3rd queue.
Finally, the display statement is placed into the 4th queue.

Only events in the active queue are completed this sim cycle, so the "**a = 0**"
happens, and then the display shows a = 0. If we were to look at the value of
a in the next sim cycle, it would show 1.

Q: Given the following snipet of Verilog code,

draw out the waveforms for clk and a

```
always @(clk) begin
   a = 0;
   #5 a = 1;
end
```

A:

```
       10      30      50      70      90     110     130
      ___     ___     ___     ___     ___     ___     ___
clk ___|   |___|   |___|   |___|   |___|   |___|   |___|   |___

a   ____________________________________________________________
```

This obviously is not what we wanted, so to get closer, you could use
"**always @ (posedge clk)**" instead, and you'd get

```
       10      30      50      70      90     110     130
      ___     ___     ___     ___     ___     ___     ___
clk ___|   |___|   |___|   |___|   |___|   |___|   |___|   |___

                              ___                      ___
a   _________________________|   |____________________|   |_______
```

Q: What is the difference between the following two lines of Verilog code?

```
#5 a = b;
a = #5 b;
```

A:**#5 a = b;** Wait five time units before doing the action for "a = b;".
The value assigned to **a** will be the value of **b** 5 time units hence.

a = #5 b; The value of **b** is calculated and stored in an internal temp register.
After five time units, assign this stored value to **a**.

: What is the difference between:

```
c = foo ? a : b;
```

and

```
if (foo) c = a;
else c = b;
```

A:The ? merges answers if the condition is "x", so for instance if foo = 1'bx, a = 'b10, and b = 'b11,
you'd get c = 'b1x.

On the other hand, **if** treats Xs or Zs as FALSE, so you'd always get c = b.

Q: Using the given, draw the waveforms for the following

versions of a (each version is separate, i.e. not in the same run):

```
reg clk;
reg a;

always #10 clk = ~clk;

(1) always @(clk) a = #5  clk;
(2) always @(clk) a = #10 clk;
(3) always @(clk) a = #15 clk;
```

Now, change a to **wire**, and draw for:

```
(4) assign #5  a = clk;
(5) assign #10 a = clk;
(6) assign #15 a = clk;
```

A:

```
       10      30      50      70      90     110     130
         ___     ___     ___     ___     ___     ___     ___
clk ___|   |___|   |___|   |___|   |___|   |___|   |___|   |___

           ___     ___     ___     ___     ___     ___     ___
(1)a ____|   |___|   |___|   |___|   |___|   |___|   |___|   |_

             ___     ___     ___     ___     ___     ___     ___
(2)a ______|   |___|   |___|   |___|   |___|   |___|   |___|

(3)a __________________________________________________________
```

Since the #delay cancels future events when it activates, any delay over the actual 1/2 period time of the clk flatlines...

With changing a to a **wire** and using **assign**, we
just accomplish the same thing...

```
       10      30      50      70      90     110     130
        ___     ___     ___     ___     ___     ___     ___
clk ___|   |___|   |___|   |___|   |___|   |___|   |___|   |___

          ___     ___     ___     ___     ___     ___     ___
(5)a ____|   |___|   |___|   |___|   |___|   |___|   |___|   |_

            ___     ___     ___     ___     ___     ___     ___
(6)a ______|   |___|   |___|   |___|   |___|   |___|   |___|

(7)a __________________________________________________________
```

Q: What is the difference between a *Vera* task and a *Verilog* task?

A:

Q: What is the difference between running the following snipet of code

on Verilog vs Vera?

```
fork {
   task_one();
   #10;
   task_one();
}

task task_one() {
   cnt = 0;
   for (i = 0; i < 50; i++) {
      cnt++;
   }
}
```

A:

Q: Given $a = "5,-3,7,0,-5,12";

Write a program to find the lowest number in the string.

A:

```
// BEGIN PERL SNIPET

$a = "5,-5,-1,0,12,-3";
(@temp) = split (/,/, $a);
$lowest = $temp[0];
```

```
for ($i=0; $i<6; $i++) {
   if ($temp[$i] < $lowest) { $lowest = $temp[$i]; }
}

print "Lowest value found was: $lowest\n";
```

// END PERL SNIPET

NOTE: You could also replace the for loop with this:

```
foreach $value (@temp) {
   if ($value < $lowest) { $lowest = $value; }
}
```

Q: Write the code to sort an array of integers.

A:

/* BEGIN C SNIPET */

```
void bubblesort (int x[], int lim) {
   int i, j, temp;

   for (i = 0; i < lim; i++) {
      for (j = 0; j < lim-1-i; j++) {

         if (x[j] > x[j+1]) {
            temp = x[j];
            x[j] = x[j+1];
            x[j+1] = temp;

         } /* end if          */
      }    /* end for j       */
   }       /* end for i       */
}          /* end bubblesort */
```

/* END C SNIPET */

Some optimizations that can be made are that a single-element array does
not need to be sorted; therefore, the "for i" loop only needs to go from
0 to **lim-1**. Next, if at some point during the iterations, we go through
the entire array WITHOUT performing a swap, the complete array has been
sorted, and we do not need to continue. We can watch for this by adding
a variable to keep track of whether we have performed a swap on this
iteration.

Q: Write the code for finding the factorial of a passed integer.

Use a recursive subroutine.

// BEGIN PERL SNIPET

```
sub factorial {
   my $y = shift;
   if ( $y > 1 ) {
      return $y * &factorial( $y - 1 );
   } else {
      return 1;
   }
}
```

// END PERL SNIPET

Q: In C, explain the difference between the **&** operator and the ***** operator.

A:**&** is the address operator, and it creates pointer values.

***** is the indirection operator, and it dereferences pointers to access the object pointed to.

Example:
In the following example, the pointer **ip** is assigned the address of variable **i** (**&i**). After that assignment, the expression ***ip** refers to the same object denoted by **i**:

```
int i, j, *ip;
ip = &i;
i = 22;
j = *ip;    /* j now has the value 22 */
*ip = 17;   /* i now has the value 17 */
```

Q: Write a function to determine whether a string is a palindrome (same forward as reverse, such as "radar" or "mom").

A:

/* BEGIN C SNIPET */

```
#include <string.h>

void is_palindrome ( char *in_str ) {
   char *tmp_str;
   int  i, length;

   length = strlen ( *in_str );
   for ( i = 0; i < length; i++ ) {
      *tmp_str[length-i-1] = *in_str[i];
   }
```

```
   if ( 0 == strcmp ( *tmp_str, *in_str ) ) printf ("String is a
palindrome");
   else printf ("String is not a palindrome");
}
```

/* END C SNIPET */

Q: Write a function to output a diamond shape according to the given (odd) input.

Examples: Input is 5   Input is 7

```
    *          *
   ***        ***
  *****      *****
   ***      *******
    *        *****
              ***
               *
```

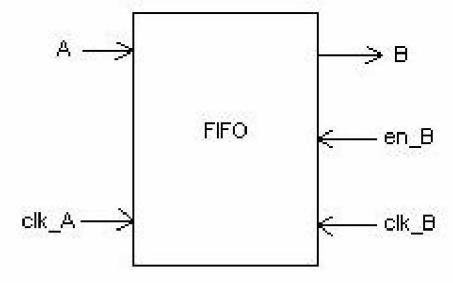
A:

BEGIN PERL SNIPET

```
for ($i = 1; $i <= (($input * 2) - 1); $i += 2) {
    if ($i <= $input) {
        $stars = $i;
        $spaces = ($input - $stars) / 2;
        while ($spaces--) { print " "; }
        while ($stars--) { print "*"; }
    } else {
        $spaces = ($i - $input) / 2;
        $stars = $input - ($spaces * 2);
        while ($spaces--) { print " "; }
        while ($stars--) { print "*"; }
    }
    print "\n";
}
```

END PERL SNIPET

Q: Given the following FIFO and rules, how deep does the FIFO need to be to prevent underflowing or overflowing?



RULES:

1) frequency(clk_A) = frequency(clk_B) / 4

2) period(en_B) = period(clk_A) * 100

3) duty_cycle(en_B) = 25%

A:

Assume clk_B = 100MHz (10ns)

From (1), clk_A = 25MHz (40ns)

From (2), period(en_B) = 40ns * 400 = 4000ns, but we only output for 1000ns,
due to (3), so 3000ns of the enable we are doing no output work.

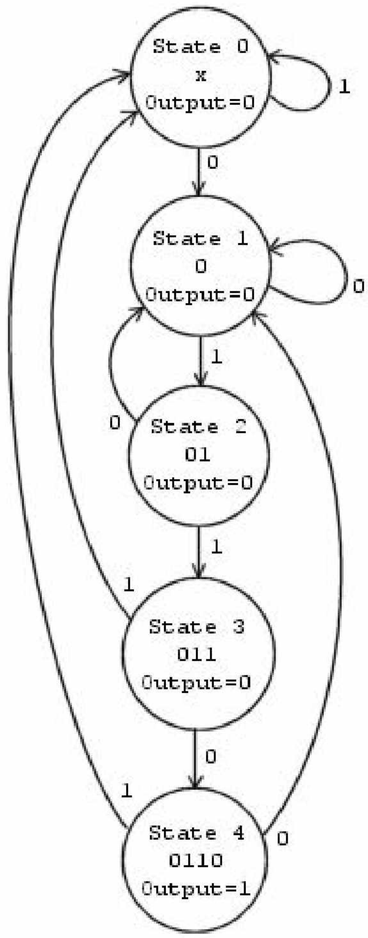
Therefore, FIFO size = 3000ns/40ns = 75 entries.

Q: Draw the state diagram to output a "1" for one cycle

if the sequence "0110" shows up (the leading 0s cannot be

used in more than one sequence).

A:



(Back)

General Answer 3

Q: Explain the differences between "Direct Mapped", "Fully Associative",

and "Set Associative" caches.

A:

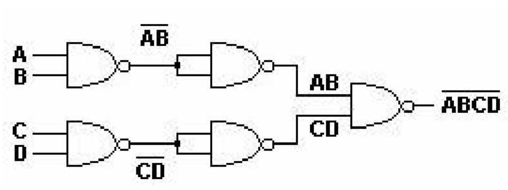
If each block has only one place it can appear in the cache, the cache is said to be ***direct mapped***. The mapping is usually (block-frame address)
modulo (number of blocks in cache).

If a block can be placed anywhere in the cache, the cache is said to be ***fully associative***.

If a block can be placed in a restricted set of places in the cache, the cache
is said to be ***set associative***. A *set* is a group of two or more blocks in the cache. A block is first mapped onto a set, and then the block
can be placed anywhere within the set. The set is usually chosen by bit
selection; that is, (block-frame address) modulo (number of **sets** in cache).
If there are *n* blocks in a set, the cache placement is
called ***n-way set associative***.

Q: Design a four-input NAND gate using only two-input NAND gates.

A:Basically, you can tie the inputs of a NAND gate together to get an inverter, so...



Q: Draw the state diagram for a circuit that outputs a "1" if the aggregate serial

binary input is divisible by 5. For instance, if the input stream is 1, 0, 1, we

output a "1" (since 101 is 5). If we then get a "0", the aggregate total is 10, so

we output another "1" (and so on).

A:We don't need to keep track of the entire string of numbers - if something

is divisible by 5, it doesn't matter if it's 250 or 0, so we can just reset to 0.
So we really only need to keep track of "0" through "4".

Synchronous and asynchronous reset.

Synchronous reset logic will synthesize to smaller flip-flops, particularly if the reset is gated with the logic generating the d-input.
But in such a case, the combinational logic gate count
grows, so the overall gate count savings may not be that significant.
The clock works as a filter for small reset glitches; however, if these glitches occur near the active clock edge, the Flip-flop could go metastable.
In some designs, the reset must be generated by a set of internal conditions. A synchronous reset is recommended for these types of designs because it will filter the logic equation glitches Between clock.
Dis advantages of synchronous reset:
Problem with synchronous resets is that the synthesis tool cannot easily distinguish the reset signal from any other data signal.
Synchronous resets may need a pulse stretcher to guarantee a reset pulse width wide enough to ensure reset is present during an active edge of the clock[
if you have a gated clock to save power, the clock may be disabled coincident with the assertion of reset. Only an asynchronous reset will work in this situation, as the reset might be removed prior to the resumption of the clock.
Designs that are pushing the limit for data path timing, can not afford to have added gates and additional net delays in the data path due to logic inserted to handle synchronous resets.

Asynchronous reset :
The biggest problem with asynchronous resets is the reset release, also called
reset removal.
Using an asynchronous
reset, the designer is guaranteed not to have the reset added to the data path.
Another advantage favoring asynchronous resets is that the circuit can be reset with or without a clock present.
Disadvantages of asynchronous reset:
ensure that the release of the reset can occur within one clock period.
if the release of the reset occurred on or near a clock edge such
that the flip-flops went metastable.

EVERY ASIC USING AN ASYNCHRONOUS RESET SHOULD INCLUDE A RESET SYNCHRONIZER CIRCUIT!!

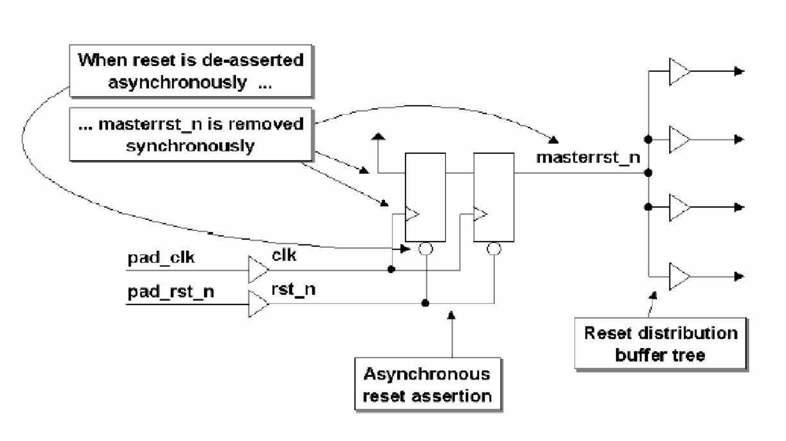
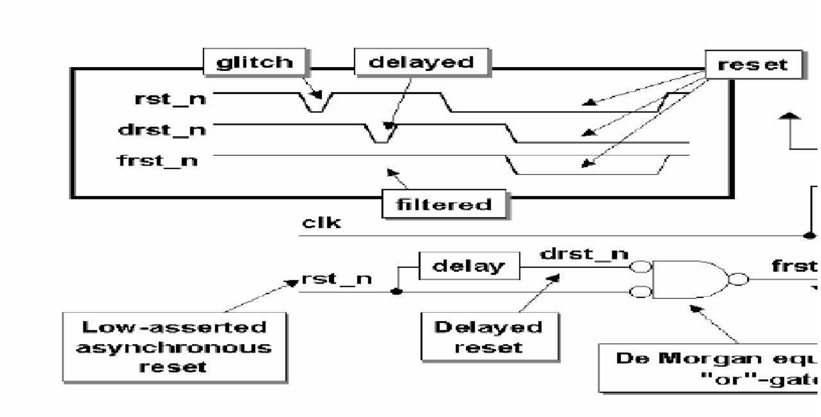


Figure 9 - Reset Synchronizer block diagram

Glitch removal in reset:



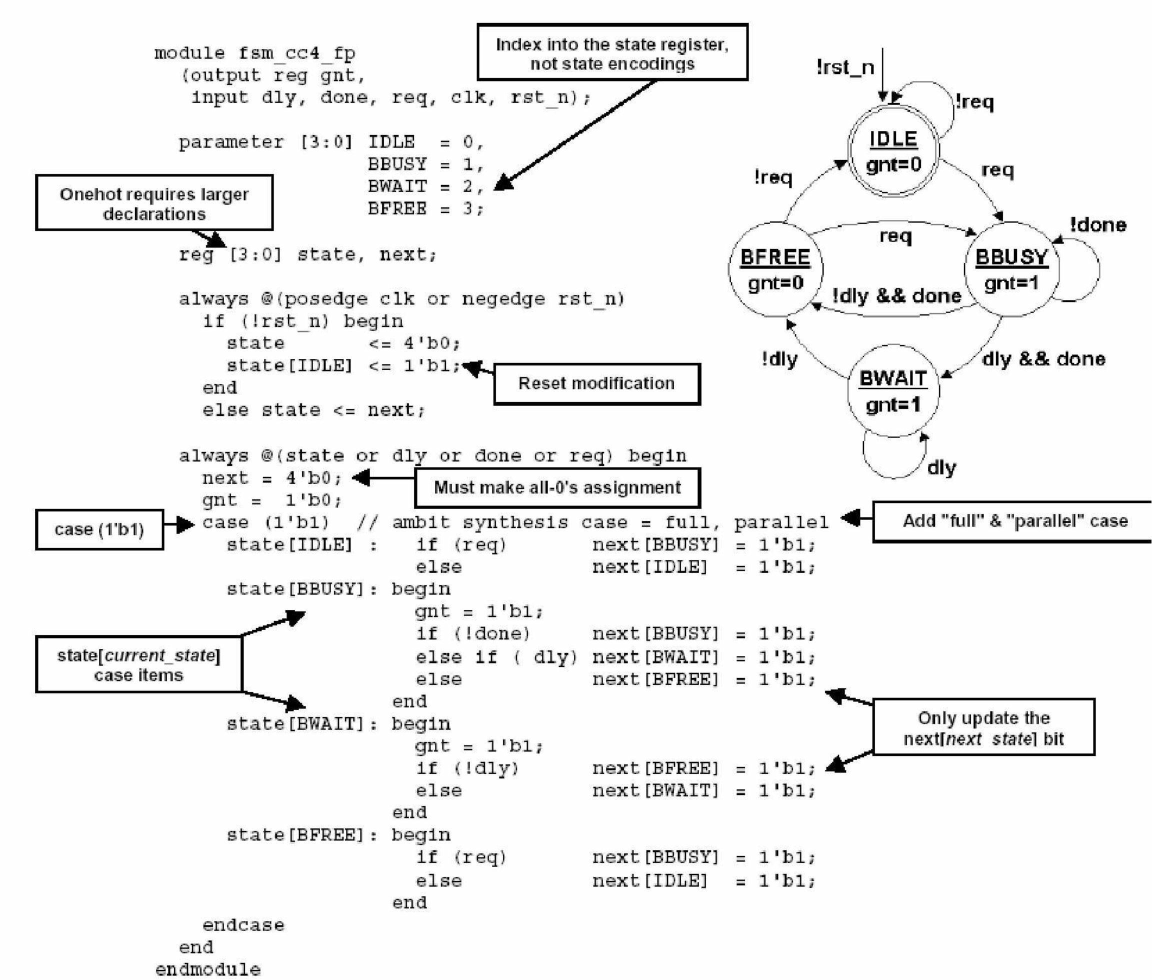
FSM:

there r mainly 4 ways 2 write fsm code
1) using 1 process where all input decoder, persent state, and output decoder r combine in one process.
2) using 2 process where all comb ckt and sequential ckt seperated in different process
3) using 2 process where input decoder and persent state r combine and output decoder seperated in other process
4) using 3 process where all three, input decoder, persent state and output decoder r seperated in 3 process.
the fsm style using 2 process where all comb ckt and sequential ckt seperated in different process in faster than all and mostly use for better performance

points:

Making default `next` equal all X's assignment
Registered FSM Outputs (Good Style)

Onehot FSM Coding Style (Good Style)



Example 7 - fsm_cc4 design - case (1'b1) onehot style - 42 lines of code

Glitch

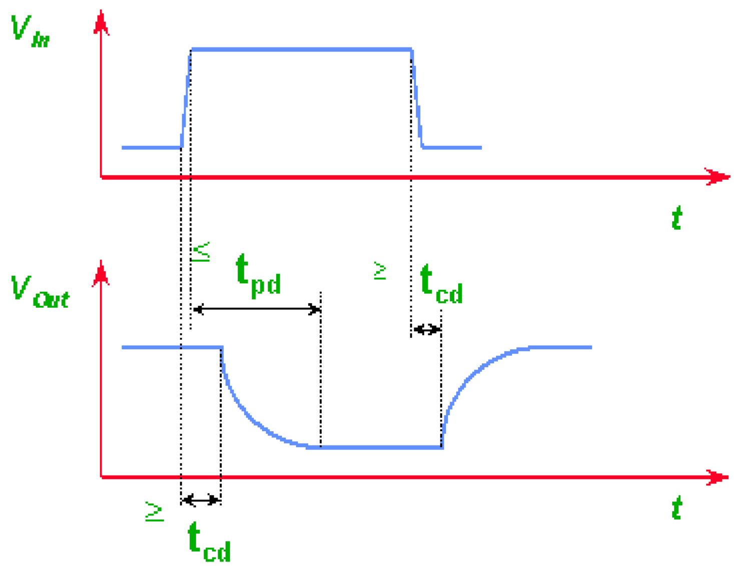
A glitch is a momentary error condition on the ouput of a circuit due to unequal path delays in a circuit. It is seen an additional pulse or pulses on the ouput. Between a time the input signals are settled and the ouput signals are being established , a glitch can occur if there is an hazard(functional or logical).

Glitches due to functional errors can occur when two input signals or more change in values at the same time. It is related to the function that is being implemented and cannot be removed by adding extra circuit.

Glitches due to logical hazard can occur only when one i/p signal changes it value. A logic hazard can be removed by add extra circuit. A logic hazard can be static or dynamic.

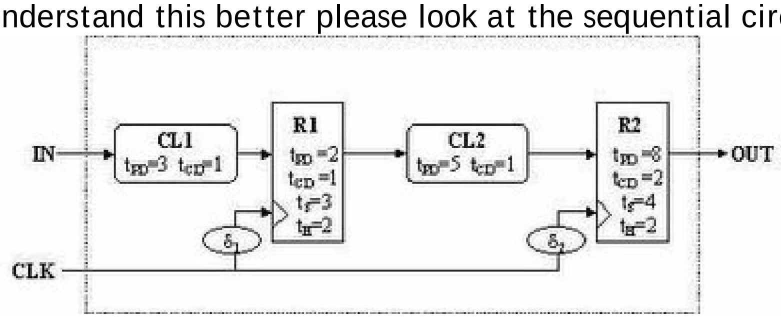
Q: What is the significance of contamination delay in sequential circuit timing?
Fact: Interestingly 70-80% of designers who deal with timing closure daily are unaware of this fact.

So, what is contamination delay anyway?
Look at the figure below. tcd is the contamination delay.



Without understanding contamination delay you "should not" complete timing estimation of any sequential circuit.

What do you mean by that?
Contamination delay tells you if you meet the hold time of a flip flop. To understand this better please look at the sequential circuit below.



The contamination delay of the data path in a sequential circuit is critical for the hold time at the flip flop where it is exiting, in this case R2.

mathematically,
th(R2) <= tcd(R1) + tcd(CL2)

Contamination delay is also called tmin and Propagation delay is also called tmax in many data sheets.

Delay Locked Loop (DLL)

Why not a PLL:

PLLs have disadvantages that make their use in high-speed designs problematic, particularly when both high performance and high reliability are required. The PLL voltage-controlled oscillator (VCO) is the greatest source of problems. Variations in temperature, supply voltage, and manufacturing process affect the stability and operating performance of PLLs.

DLLs, however, are immune to these problems. A DLL in its simplest form inserts a variable delay line between the external clock and the internal clock. The clock tree distributes the clock to all registers and then back to the feedback pin of the DLL. The control circuit of the DLL adjusts the delays so that the rising edges of the feedback clock align with the input clock. Once the edges of the clocks are aligned, the DLL is locked, and both the input buffer delay and the clock skew are reduced to zero.

Advantages:

- precision
- stability
- power management
- noise sensitivity
- jitter performance.

Fifo depth calculation

Assuming,

- F1 = frequency of the writing side.
- F2 = frequency of the reading side.
- D = data burst.

Burst duration = D/F1
Data Rec'd, Rx= (D/F1) * F2, assuming simultaneous read for the duration.
Extra storage during FULL condition, Backlog = D-Rx = D(F1-F2)/F1

To accommodate latency or response time in the receiver, T, we need additional T * F1 locations.
Receiver also needs time to read all this backlog, so the idle time between bursts must be long enough. So this minimum time is called mop-up time = backlog/F2 = D * (F1-F2)/(F1.F2)

Note: For bursts of data which are written for partial amount time for a given number of cycles and read that are happening continously "or" are also read for a partial amount of time for a given number of cycles, the calculation has to account for the next burst.

Types of Timing Verification

Dynamic timing:

1. The design is simulated in full timing mode.
2. Not all possibilities tested as it is dependent on the input test vectors.
3. Simulations in full timing mode are slow and require a lot of memory.
4. Best method to check asynchronous interfaces or interfaces between different timing domains.

Static timing:

1. The delays over all paths are added up.
2. All possibilities, including false paths, verified without the need for test vectors.
3. Much faster than simulations, hours as opposed to days.
4. Not good with asynchronous interfaces or interfaces between different timing domains.

i have two ASICs…

I have two ASICs. one has setup violation and the other has hold violation. how can they be made to work together without modifying the design?

Slow the clock down on the one with setup violations..
And add redundant logic in the path where you have hold violations.

Ways to increase frequency of operation

- Check critical path and optimize it.
- Add more timing constraints (over constrain).
- pipeline the architecture to the max possible extent keeping in mind latency req's.

When are DFT and Formal verification used?

DFT:

- manufacturing defects like stuck at "0" or "1".
- test for set of rules followed during the intitial design stage.

Formal verification:

- Verification of the operation of the design, i.e, to see if the design follows spec.
- gate netlist == RTL ???
- using mathematics and statistical analysis to check for eqivalence.

Why interrupts are active low?

This answers why most signals are active low…

If you consider the transistor level of a module, active low means the capacitor in the output terminal gets charged or discharged based on low to high and high to low transition respectively. when it goes from high to low it depends on the pull down resistor that pulls it down and it is relatively easy for the output capacitance to discharge rather than charging. hence people prefer using active low signals.

Polysilicon Vs Metal

Normally polysilicon has more resistance compared to metal. For shorter distance we go with polysilicon keeping fabrication process in mind .

Usage of metal for short distances need contacts which increases the resistance significantly.
Poly has got higher melting point and can withstand high temperature phases that follow Gate formation. So,Poly is preffered to metal, Although it has got high resistivity.

NAND or NOR design

NAND is a better gate for design than NOR becuase 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).

negative setup and hold time

A negative setup and hold condition is a very interesting proposition in static timing analysis. Support for this type of conditions was added in the Verilog LRM, only in the late 90's (using the $SETUP and $HOLD constructs).

The basic idea is something like this:
Consider a module with an ideal flop in it. Now, there

exists a data path (from primary inputs of module to D of flop) and a clock path (from primary inputs to CLK of flop). Suppose the data path delay is DD and clock path delay is 0 . Therefore, if we consider the clock pulse reaching at the primary input of the module as the reference time, the clock pulse will reach CLK pin (of flop) at 0. The data pulse will reach D pin at DD. Therefore, for setup check to be met, the data pulse must reach the primary inputs of the module, at -D, which means the setup requirement is D. Now consider a clock path delay of CD. This means that the clock pulse now reaches the flop, only after time CD. This means, the data pulse need not begin so early, and rather it has to begin at -DD+CD time(just right shifting the pulse by CD time). This means the setup requirement is now DD-CD. In this case, if CD>DD, then the setup requirement becomes negative, which means, the data pulse can reach the primary input of the module after the clock pulse has reached there.

Similarly for hold:

Consider that the data delay is 0 and the clock delay is CD. Now, the data must not change for atleast CD time, for the flop to be able to latch it. Therefore, the hold requirement is CD. Now, consider a data delay of DD. This means that, now the data need not change only for CD -DD. This is the new hold requirement. If DD>CD, then hold requirement is negative. If we analyse these results mathematically, we can see that setup relation + hold relation =0.

Physically, this implies that an infinitesimally short pulse (a delta pulse) can be captured; which is of course not possible. A more accurate model would be:setup_val<DD-

CD (for setup to be met, the time at which data begins should be atleast DD-CD before 0) hold_val < CD-DD (for hold to be met, the time for which the data should be stable should always be greater than the hold_val) Now, the model we described, regarding the module with an ideal flop, is actually a real world flop. In an actual flop, there are more than one data paths and 8 clock paths. Therefore the more accurate description would be: DDmax-CDmin >= setup_val (for setup to be met) CDmax-DDmin >= hold_val (for hold to be met) These kind of relationships, especially the ones, where a negative relations can hold cause problems in simulators. Take for example a data pulse, which rises at 0.0 and falls at 2.0. Now the clock pulse rises at 3.0 . Lets say data delay is 1.0 Assume the origin at the clock pulse (3.0) . Therefore data rise is at -3.0, fall is at -2.0 . The setup relationship may be specified as 2.0, which means data should be present at 0.0-2.0=-2.0 . Now data will arrive at -3.0+DD-CD=-3.0+1.0+0.0=-2.0 (setup OK) The hold relationship may be specified as -1.0, which means data must not change till 0.0+(-1.0)=-1.0. Now, according to our relationship, data will not change till 0.0+CD-DD=0.0-1.0=-1.0. All looks hunky dory...but... There is no problem with the timing checks, however in software, the simulator would capture the falling 2.0 edge rather than the high edge. So the simulator will get the functionally incorrect results, though timing accurate. If both setup and hold relationships were positive, then this would never have happened. So now what ? Very simple actually, instead of taking an ideal clock, the simulator takes a delayed clock. Therefore all calculations are done wrt this delayed clock (in the above example clock is delayed -1 wr.t data), so the simulator will not latch the falling edge.

What is the meaning of data setup time and data hold time? Can either of these values be zero? Can they be negative? If the answer is yes to either of the last two questions, what does it imply?

The data setup time is the time the data inputs must be valid before the clock/strobe signal occurs. The hold time is the time the data must remain valid after the clock/strobe. Both can be zero or negative. An example is that t_SHDI - the data hold time after DS* is high is 0. A zero setup time means that the time for the data to propagate within the component and load into the latch is less than the time for the clock to propagate and trigger the latch. A zero hold time means either that the moment the clock is asserted, the latch no longer looks at its inputs, or else that the clock path delay is shorter than the data path delay. A negative setup or hold time means that there is an even larger difference in path delays, so that even if the data is sent later than the clock (for setup time), it still arrives at the latch first. Typically manufacturers avoid specifying negative values since this restricts later design and manufacturing decisions, but they often specify zero values since this simplifies usage in a system.

what is the difference between a clock cycle, a bus cycle, and an instruction cycle?

The clock cycle is one period of the master clock. A bus cycle is one bus transaction, e.g. a read or write cycle, which consists of address and data transfers, and the appropriate strobe and acknowledge signals, as specified by the bus protocol. In the 68000 a bus cycle takes a minimum of 8 clock phases or states, or 4 clock cycles, longer if wait states are added. An instruction cycle is one instruction execution - instruction fetch, instruction decode, address calculation, operand fetch, execution, and store. Since the 68000 does not have a cache, and each instruction is 1 to 11 words in length, as many as 11 bus cycles must occur to read the instruction. The MOVEM instruction can save or restore as many as 16 32-bit registers, requiring as many as 32 read or write cycles. Thus an instruction cycle can be from one to many bus cycles.

Explain sizing of the inverter?

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

1. How do you size NMOS and PMOS transistors to increase the threshold voltage?
   Ans:-

2. What is Noise Margin? Explain the procedure to determine Noise Margin
   Ans: - The minimum amount of noise that can be allowed on the input stage for which the output will not be effected.
3. What happens to delay if you increase load capacitance?
   Ans:-delay increases.
4. What happens to delay if we include a resistance at the output of a CMOS circuit?
   Ans:-increases. (RC delay)
5. What are the limitations in increasing the power supply to reduce delay?
   Ans: - 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.
6. How does Resistance of the metal lines vary with increasing thickness and increasing length?
   Ans: - $R = (\rho * l) / A$.
7. For CMOS logic, give the various techniques you know to minimize power consumption
   Ans:-power dissipation=CV^2f ,from this minimize the load capacitance, dc voltage and the operating frequency.
8. What is Charge Sharing? Explain the Charge Sharing problem while sampling data from a Bus?
   Ans:-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 once. To eliminate this load capacitance must be very high compared to the input capacitance of the gates (approximately 10 times).

9. Why do we gradually increase the size of inverters in buffer design? Why not give the output of a circuit to one large inverter?
   Ans:-because it can not drive the output load straight away, so we gradually increase the size to get an optimized performance.
10. What is Latch Up? Explain Latch Up with cross section of a CMOS Inverter. How do you avoid Latch Up?
   Ans:-Latch-up is a condition in which the parasitic components give rise to the Establishment of low resistance conducting path between V_{DD} and V_{SS} with Disastrous results.
11. Give two ways of converting a two input NAND gate to an inverter
   Ans: - (1) short the 2 inputs of the nand gate and apply the single input to it,
   (2) Connect the output to one of the input and the other to the input signal.

13. What are set up time & hold time constraints? What do they signify? Which one is critical for estimating maximum clock frequency of a circuit?
Ans:-**set up time**: - the amount of time the data should be stable before the application of the clock signal, where as the **hold time** is the amount of time the data should be stable after the application of the clock. Setup time signifies maximum delay constraints; hold time is for minimum delay constraints. Setup time is critical for establishing the maximum clock frequency.

12. Give the expression for CMOS switching power dissipation?
   Ans: - CV^2_{DD} f
13. What is Body Effect?
   Ans:-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}\neq$0).Which results V_{th2}>V_{th1}.
14. Why do we need both PMOS and NMOS transistors to implement a pass gate?
   Ans:-To avoid the degraded version of ‘0’ and ‘1’ at the output. (Ref:-86 from Neil weste).
15. What is a D-latch? Write the VHDL Code for it?
   Ans: - It follows the input with some delay.
16. Differences between D-Latch and D flip-flop?
   Ans:-D-latch is level sensitive where as flip-flop is edge sensitive. Flip-flops are made up of latches.
17. What is latchup? Explain the methods used to prevent it?
   Ans: - latchup is a condition in which the parasitic components give rise to the establishment of low resistance conducting paths between V_{dd} and V_{ss} with disastrous results.

   <u>Methods to prevent</u>

   1) Increase in substrate doping levels with a consequent drop in the level of R_s
   2) Reducing R_p by controlling the fabrication parameters and by ensuring a low contact resistance to V_{ss}.
   3) Proper layout techniques.

19. What is D-FF?

Ans:-The level present at D will be stored in the F-F at the instance the positive edge transmission, i.e. the input pulse will transfer the value of input D into output of the F-F independent of the value of the output before the pulse was applied.
20. What is a multiplexer?
Ans:-Is a combinational circuit that selects binary information from one of many input lines and directs it to a single output line. (2^n =>n).
21. How can you convert an SR Flip-flop to a JK Flip-flop?
Ans:-By giving the feed back we can convert, i.e !Q=>S and Q=>R.Hence the S and R inputs will act as J and K respectively.
22. How can you convert the JK Flip-flop to a D Flip-flop?
Ans:-By connecting the J input to the K through the inverter.
23. What is Race-around problem? How can you rectify it?
Ans:-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.
24. Which semiconductor device is used as a voltage regulator and why?

Ans:-Zenor diode, It can be operated as a constant voltage device at breakdown region with adequate power dissipation capabilities.

25. Explain an ideal voltage source?

Ans:-

26. Explain zener breakdown and avalanche breakdown?

Ans:- A thermally generated carrier (part of reverse saturation current) falls down the junction barrier and acquires energy from the applied potential. This carriers collides with a crystal ion and imparts sufficient energy to disrupt a covalent bond.In addition to the original carrier, a new electron-hole pair has been generated. These carriers may also pick up sufficient energy and creates still another electron-hole pair. This cumulative process is called the **Avalanche breakdown**.

A reverse electric field at the junction causes a strong force to be applied on a bounded electron by the field to tear it out of its covalent bond. The new hole-electron pair which is created increases the reverse current, called **zener breakdown**.

27. What are the different types of filters?

Ans:-

28. What is sampling theorem?

Ans:-1)A band limited signal of finite energy ,which has no frequency components higher than W Hz, is completely described by specifying the value of the signal at instants of time separated by 1/2W seconds.
2)A band limited signal of finite energy, which has no frequency component higher than W Hz, may be completely recovered from a knowledge of its samples taken at the rate of 2W per second.

29. What is impulse response?

Ans: - The response of a system to a unit impulse or delta function δ (t).

30. What are set up time & hold time constraints? What do they signify? Which one is critical for estimating maximum clock frequency of a circuit?
Ans) (i) **Setup time** is the minimum time prior to trigerring edge of the clock pulse up to which the data should be kept stable at the flip-flop input so that data could be properly sensed at the input. **Hold time** is the minimum time after the clock edge upto which the data should be kept stable in order to trigger the flip flop at right voltage level. Setup time is required in order to find the maximum clock frequency of a circuit.
(ii)**Setup time**: It is the minimum time before the clock edge the input should be stable. This is due to the input capacitance present at the input. It takes some time to charge to the particular logic level at the input.
Hold time: It is the minimum time the input should be present stable after the clock edge. This is the time taken for the various switching elements to transit from saturation to cut off and vice versa. So basically set up and hold time is the window during which the input should be stable. Any changes in the input during the window period may lead to voltage levels which are not recognized by the subsequent stages and the circuit may go to metastable stage.

31. How do you detect if two 8-bit signals are same?
Ans) XOR each bits of A with B (for e.g. A[0] xor B[0]) and so on.the o/p of 8 xor gates are then given as i/p to an 8-i/p nor gate. if o/p is 1 then A=B.

32. How is the FPGA configured?

Ans. The FPGA can be configured in any one of three ways: from FLASH during power up, under host processor control of via JTAG. Each of the VPF1's FPGA node configuration is supplied by its attached PowerPC processor: the configuration is stored in the PowerPC CPU's FLASH. Transtech development tools facilitate configuring the FPGA in both development and run-time environments. JTAG can also be used for FPGA configuration during development. An onboard battery (optional) is available so that encrypted keys can be stored for secure FPGA configurations.

33. In what way are QDR SRAMs useful?

Ans. Fast banks of SRAM are ideal for lookup tables, local data buffers and DSP operations such as dealing with concurrent (MAC) Multiply- ACcumulate data streams.

Although FPGAs include an amount of on-chip memory, this is not sufficient for applications that need to support large datasets, such as large FFTs, or framestores for image processing. Without external memories, the performance could be severely compromised or it may not even be possible to implement the algorithm.Each VPF1 FPGA node includes four banks of 2Mx 18-bit QDR SRAM.

34. How many flip flops you require for modulo 33 counter.
Ans. 6 f/f

35. 7 bit ring counter's initial state is 0100010. After how many clock cycles will it return to the initial state.
Ans. 6 cycles

36. some boolean expression of the form x'y'z' + yz +.. (something like this) find the simplified expression
Ans. z(x+y)

37. A signed no is stored in 10-bit register, what is the max and min possible value of the number.
Ans. 2^10 - 1.....max -2^10 ......min

Q1

What is Instrumentation Amplifier(IA) and what are all the advantages?

Ans1)
An instrumentation amplifier is a differential op-amp circuit providing high input impedances with ease of gain adjustment by varying a single resistor

Q2)

Two Nmos Transistors are connected in series and the gates of both the transistors are connected to 5V. One end of the Transistor is connected to a 10V supply and the Vth=1V . What is the voltage at the other end ?

PS:This was a quick interview question asked by a company called Alcatel, to short list one of my friend for the interview

Ans: Q2:

The output voltage is 4V.

Consider a single NMOS as a switch.
The max voltage at the other end can reach max of VG - Vt, after that NMOS will be off.
So if the voltage at one end is less than VG-Vt it passes that value to the other end, but if it is more, it reaches VG-Vt at the end and stops there bcoz after that the MOS switch will be off.

So in this case, first NMOS which has 12v, at the input, gives 4v out at its source, the other Transistor which has 4v at the input transmits samething to the other end as it is.

So final voltage is 4V.

Q3

What are the important characteristics of a Source Follower ?

Source follower need not be for an opamp.
1.Give the input to the gate and take the output at the source of a Mosfet, we get the configuration called Source follower.The gain of such a stage is very close to 1.
2.It acts as a voltage buffer
3.Some of the drawbacks of this are non-linearity due to body effect,voltage headroom consumption due to level shift, and poor driving capability.

Q5) 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 dont loose our current into the substrate..

Q4) What is Miller capacitance.
It is an unwanted effective capacitance between input and output of an amplifier which often effects input impedance and hamper its frequency response.

We all are aware of miller's effect right!.. If there is an impedance between input and output of an Amplifier(gain=A). this particular impedance(Z) is seen from Input as Z/(1+A). if this impedance happens to be a capacitor(i.e Z= 1/[jwC]; where 'w' is the frequency in s-domain) then input capacitance is increased by a factor of (1+A). i.e it is equivalent to have a capacitor (with an effective capacitance of 1/[(1+A)*(jwC)]) hanging between input node and ground. So.. effective it acts as a LOW pass Filter.

Now replace the earlier amplifier with a MOS for say NMOS in common source configuration...fine..

Now here.. the input is *Gate* and output is *Drain*. So the efective capacitance Cgd between gate and drain shall hamper the frequency response of this CS-amplifier. High frequency component in a singal that is fed at gate shall be void in the output at drain, and also effecting its input impedance. It amplifies only the part of input signal that falls in lower part of the spectrum.

The same can be argued with Common Emitter amplifier.. with base as input and Collector as output...

If we compare both these transistors' performance in this situation.. NMOS has better performance over n-p-n BJT, as such NMOS actual impedance at (gate)input is very high.. due to Oxide acting as good insulator.

Q5) What is thefundamental difference between a MOSFET and BJT ?

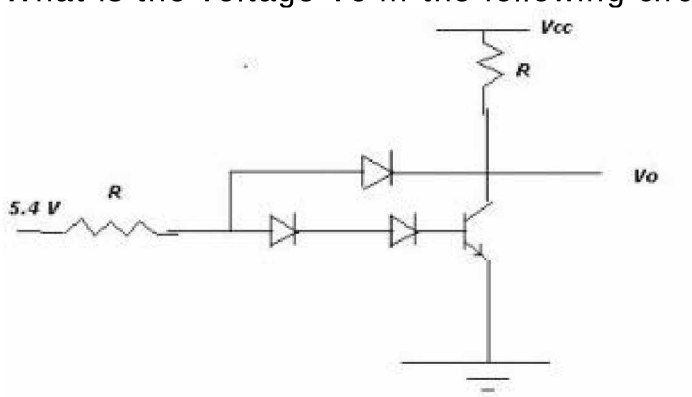
In MOSFET, current flow is either due to electrons(n-channel MOS) or due to holes(p-channel MOS)
- In BJT, we see current due to both the carriers.. electrons and holes.
BJT is a current controlled device and MOSFET is a voltage controlled device.

What is Hall effect and its application?

Q7 What are MEMS.?

Ans7
Micro-Electro-Mechanical Systems (MEMS) is the integration of mechanical elements, sensors, actuators, and electronics on a common silicon substrate through microfabrication technology. While the electronics are fabricated using integrated circuit (IC) process sequences (e.g., CMOS, Bipolar, or BICMOS processes), the micromechanical components are fabricated using compatible "micromachining" processes that selectively etch away parts of the silicon wafer or add new structural layers to form the mechanical and electromechanical devices.

Q8:
All diode drops are 0.7 V .The Vbe voltage is 0.7V if forward biased. What is the voltage Vo in the following circuit:



In the circuit the voitage

output Vo should be 0.7 since the, the voltage across the series diodes is 1.4 volts.The above diode is in off state because it is not in forward bias condition. The voltage given as input to the base is 4 volts. Therfore the transistor turns on.. Hence the voltage drop of 0.7 volts i.e, Vbe appears across the output..
So Vo is 0.7 volts.

Ans Q8:

As the transistor is on, voltage at base of 0.7V because emiter is grounded

So the input voltage at the two doides is 2.1V

2.1v is sufficient enough to make the other diode on, which gives an O/P DC voltage of 1.4V

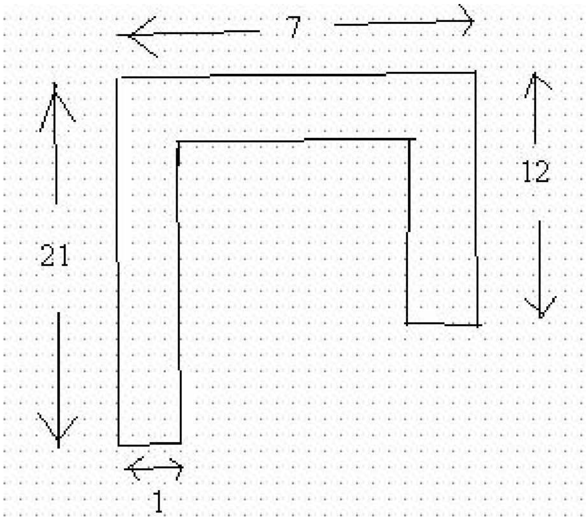
So Voltage at Vo is 1.4V

Q9:
A DAC has the following equation 0.5b2 + 0.20 b1 + 0.175 b0. Find its maximum error with respect to the ideal DAC.
Ans:Q9:
ideal DAC: 0.5b2 + 0.25b1 + 0.125 b0

Worst case error will both b0 and b1 are 1.

So error = 0.05 + 0.05 = 0.1
Q10:
The following n-substrate has shown dimensions(micro meter). The resistance is 25ohm/sq micrometer and the corner squares contribute resistance which is only 0.5 times the non-corner ones. What is its

sheet resistance?



I wil give a try to this question though not sure if i am right..
36 micrometer square is the area of the squares.
and 2 corner squares.

so
36 * 25 + 2 * 12.5
925 ohms..

Q12

which transistor has higher gain. BJT or MOS and why?

Ans12)BJT has higher gain bcoz it has higher transconductance.This is because the current in bjt is exponentially dependent on input where as in mosfet it is square law...This is what I could recall from my undergrad bjt stuff since i have been completely out of touch with bjt now...

Q13)
Why are OPAMPs called opamps?

A13)Operational Amplifiers because they formed the basis of how we could perform mathematical operations like addition,subtraction,integration,differentiation etc using circuits.

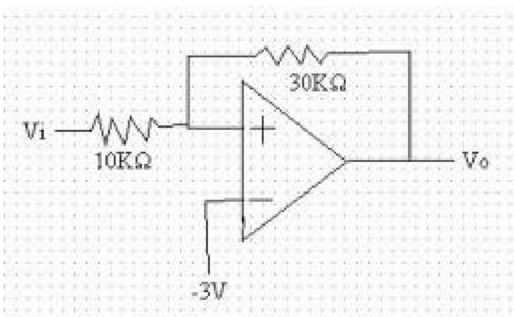
Q14)
How do you use opamp as comparator?

Ans:Q14:
If you use OPAMP in open loop, if goes to either +ve saturation or -ve saturation based on the voltage at the + and -ve termianls.

I mean, if voltage at + terminal is more, it goes to + and similarly if volatege at -vs is more, it goes to -ve

We can use this principle and use OPAMP as a comparator.
Am I correct.
Q15)What is a Schmitt Trigger and why would you use it?How would you use an op-amp as a Schmitt trigger?

Q15:
For this Opamp, the saturation voltages are +15V and -15V. Find VUT and VLT this ckt.



Q17)
Guys, one more interview questions from my phone interview
What happens to the current flow in a mosfet as temperature decreases.

Q18)
What are Guard Rings and why are they used?

Ans to Q18)

Guard rings are nothing but substrate/well contacts placed close to the source connection of the N-MOS and P-MOS transistor.They are used to reduce the high current which has occured due to latch up.Internally they will be used to reduce the gain of the pnpn thyristor like devices which in turn will reduce the current.

guard rings basically reduce your Resistance between substrate and power supplies so that the gain of the pnpn transistor loop is small.

Q19

List the different ways in which a MOSfet can breakdown ?

Q 20. a) How does channel length modulation effect the CMOS inverter characterstics or VTC of CMOS inverter.

b) what parameters has to be varied to obtain early pinch off in MOS transistor

q 21)Which device is fast BJT or MOS? Why ?

Q22 :

what kind of analog buffers do we use if we have very high Cload on a common source amplifier.

1) Explain why & how a MOSFET works
2) Draw Vds-Ids curve for a MOSFET. Now, show how this curve changes (a) with increasing Vgs (b) with increasing transistor width (c) considering Channel Length Modulation
3) Explain the various MOSFET Capacitances & their significance
4) Draw a CMOS Inverter. Explain its transfer characteristics
5) Explain sizing of the inverter

6) How do you size NMOS and PMOS transistors to increase the threshold voltage?
7) What is Noise Margin? Explain the procedure to determine Noise Margin
8) Give the expression for CMOS switching power dissipation
9) What is Body Effect?
10) Describe the various effects of scaling
11) Give the expression for calculating Delay in CMOS circuit
12) What happens to delay if you increase load capacitance?
13) What happens to delay if we include a resistance at the output of a CMOS circuit?
14) What are the limitations in increasing the power supply to reduce delay?
15) How does Resistance of the metal lines vary with increasing thickness and increasing length?
16) You have three adjacent parallel metal lines. Two out of phase signals pass through the outer two metal lines. Draw the waveforms in the center metal line due to interference. Now, draw the signals if the signals in outer metal lines are in phase with each other
17) What happens if we increase the number of contacts or via from one metal layer to the next?
18) Draw a transistor level two input NAND gate. Explain its sizing (a) considering Vth (b) for equal rise and fall times
19) 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 NMOS inputs A & B, which one would you place near the output?
20) Draw the stick diagram of a NOR gate. Optimize it
21) For CMOS logic, give the various techniques you know to minimize power consumption
22) What is Charge Sharing? Explain the Charge Sharing problem while sampling data from a Bus
23) Why do we gradually increase the size of inverters in buffer design? Why not give the output of a circuit to one large inverter?
24) In the design of a large inverter, why do we prefer to connect small transistors in parallel (thus increasing effective width) rather than lay out one transistor with large width?
25) Given a layout, draw its transistor level circuit. (I was given a 3 input AND gate and a 2 input Multiplexer. You can expect any simple 2 or 3 input gates)
26) Give the logic expression for an AOI gate. Draw its transistor level equivalent. Draw its stick diagram
27) Why don't we use just one NMOS or PMOS transistor as a transmission gate?
28) For a NMOS transistor acting as a pass transistor, say the gate is connected to VDD, give the output for a square pulse input going from 0 to VDD
29) Draw a 6-T SRAM Cell and explain the Read and Write operations
30) Draw the Differential Sense Amplifier and explain its working. Any idea how to size this circuit? (Consider Channel Length Modulation)
31) What happens if we use an Inverter instead of the Differential Sense Amplifier?
32) Draw the SRAM Write Circuitry
33) Approximately, what were the sizes of your transistors in the SRAM cell? How did you arrive at those sizes?

34) How does the size of PMOS Pull Up transistors (for bit & bit- lines) affect SRAM's performance?
35) What's the critical path in a SRAM?
36) Draw the timing diagram for a SRAM Read. What happens if we delay the enabling of Clock signal?
37) Give a big picture of the entire SRAM Layout showing your placements of SRAM Cells, Row Decoders, Column Decoders, Read Circuit, Write Circuit and Buffers
38) In a SRAM layout, which metal layers would you prefer for Word Lines and Bit Lines? Why?
39) How can you model a SRAM at RTL Level?
40) What's the difference between Testing & Verification?
41) For an AND-OR implementation of a two input Mux, how do you test for Stuck-At-0 and Stuck-At-1 faults at the internal nodes? (You can expect a circuit with some redundant logic)
42) What is Latch Up? Explain Latch Up with cross section of a CMOS Inverter. How do you avoid Latch Up?
43) Insights of an inverter. Explain the working?
44) Insights of a 2 input NOR gate. Explain the working?
45) Insights of a 2 input NAND gate. Explain the working?
46) Implement F= not (AB+CD) using CMOS gates?
47) Insights of a pass gate. Explain the working?
48) Why do we need both PMOS and NMOS transistors to implement a pass gate?
49) What does the above code synthesize to?
50) Cross section of a PMOS transistor?
51) Cross section of an NMOS transistor?
52) What is a D-latch? Write the VHDL Code for it?
53) Differences between D-Latch and D flip-flop?
54) Implement D flip-flop with a couple of latches? Write a VHDL Code for a D flip-flop?
55) What is latchup? Explain the methods used to prevent it?
56) What is charge sharing?
57) While using logic design, explain the various steps that r followed to obtain the desirable design in a well defined manner?
58) Why is OOPS called OOPS? (C++)
59) What is a linked list? Explain the 2 fields in a linked list?
60) Implement a 2 I/P and gate using Tran gates?
61) Insights of a 4bit adder/Sub Circuit?
62) For f = AB+CD if B is S-a-1, what r the test vectors needed to detect the fault?
63) Explain various adders and diff between them?
64) Explain the working of 4-bit Up/down Counter?
65) A circuit has 1 input X and 2 outputs A and B. If X = HIGH for 4 clock ticks, A = 1. If X = LOW for 4 clock ticks, B = 1. Draw a state diagram for this Spec?
66) Advantages and disadvantages of Mealy and Moore?
67) Id vs. Vds Characteristics of NMOS and PMOS transistors?
68) Explain the operation of a 6T-SRAM cell?

69) Differences between DRAM and SRAM?
70) Implement a function with both ratioed and domino logic and merits and demerits of each logic?
71) Given a circuit and asked to tell the output voltages of that circuit?
72) How can you construct both PMOS and NMOS on a single substrate?
73) What happens when the gate oxide is very thin?
74) What is setup time and hold time?
75) Write a pseudo code for sorting the numbers in an array?
76) What is pipelining and how can we increase throughput using pipelining?
77) Explain about stuck at fault models, scan design, BIST and IDDQ testing?
78) What is SPICE?
79) Differences between IRSIM and SPICE?
80) Differences between netlist of HSPICE and Spectre?
81) What is FPGA?
82) Draw the Cross Section of an Inverter? Clearly show all the connections between M1 and poly, M1 and diffusion layers etc?
83) Draw the Layout of an Inverter?
84) If the current thru the poly is 20nA and the contact can take a max current of 10nA how would u overcome the problem?
85) Implement F = AB+C using CMOS gates?
86) Working of a 2-stage OPAMP?
87) 6-T XOR gate?
88) Differences between blocking and Non-blocking statements in Verilog?
89) Differences between Signals and Variables in VHDL? If the same code is written using Signals and Variables what does it synthesize to?
90) Differences between functions and Procedures in VHDL?
91) What is component binding?
92) What is polymorphism? (C++)
93) What is hot electron effect?
94) Define threshold voltage?
95) Factors affecting Power Consumption on a chip?
96) Explain Clock Skew?
97) Why do we use a Clock tree?
98) Explain the various Capacitances associated with a transistor and which one of them is the most prominent?
99) Explain the Various steps in Synthesis?
100) Explain ASIC Design Flow?
101) Explain Custom Design Flow?
102) Why is Extraction performed?
103) What is LVS, DRC?
104) Who provides the DRC rules?
105) What is validation?
106) What is Cross Talk?
107) Different ways of implementing a comparator?
108) What r the phenomenon which come into play when the devices are scaled to the sub-micron lengths?

109) What is clock feed through?
110) Implement an Inverter using a single transistor?
111) What is Fowler-Nordheim Tunneling?
112) Insights of a Tri-state inverter?
113) If an/ap = 0.5, an/ap = 1, an/ap = 3, for 3 inverters draw the transfer characteristics?
114) Differences between Array and Booth Multipliers?
115) Explain the concept of a Clock Divider Circuit? Write a VHDL code for the same?
116) Which gate is normally preferred while implementing circuits using CMOS logic, NAND or NOR? Why?
117) Insights of a Tri-State Inverter?
118) Basic Stuff related to Perl?
119) Have you studied buses? What types?
120) Have you studied pipelining? List the 5 stages of a 5 stage pipeline. Assuming 1 clock per stage, what is the latency of an instruction in a 5 stage machine? What is the throughput of this machine ?
121) How many bit combinations are there in a byte?
122) For a single computer processor computer system, what is the purpose of a processor cache and describe its operation?
123) Explain the operation considering a two processor computer system with a cache for each processor.
124) What are the main issues associated with multiprocessor caches and how might you solve them?
125) Explain the difference between write through and write back cache.
126) Are you familiar with the term MESI?
127) Are you familiar with the term snooping?
128) Describe a finite state machine that will detect three consecutive coin tosses (of one coin) that results in heads.
129) In what cases do you need to double clock a signal before presenting it to a synchronous state machine?
130) You have a driver that drives a long signal & connects to an input device. At the input device there is either overshoot, undershoot or signal threshold violations, what can be done to correct this problem?
131) What are the total number of lines written by you in C/C++? What is the most complicated/valuable program written in C/C++?
132) What compiler was used?
133) What is the difference between = and == in C?
134) Are you familiar with VHDL and/or Verilog?
135) What types of CMOS memories have you designed? What were their size? Speed?
136) What work have you done on full chip Clock and Power distribution? What process technology and budgets were used?
137) What types of I/O have you designed? What were their size? Speed? Configuration? Voltage requirements?

138) Process technology? What package was used and how did you model the package/system? What parasitic effects were considered?
139) What types of high speed CMOS circuits have you designed?
140) What transistor level design tools are you proficient with? What types of designs were they used on?
141) What products have you designed which have entered high volume production?
142) What was your role in the silicon evaluation/product ramp? What tools did you use?
143) If not into production, how far did you follow the design and why did not you see it into production?

1. What happens if Vds is increased over saturation?
2. In the I-V characteristics curve, why is the saturation curve flat or constant?
3. What happens if a resistor is added in series with the drain in a mos transistor?
4. What are the different regions of operation in a mos transistor?
5. What are the effects of the output characteristics for a change in the beta (â) value?
6. What is the effect of body bias?
7. What is hot electron effect and how can it be eliminated?
8. What is latchup problem and how can it be eliminated?
9. What is channel length modulation?
10. What is the effect of temperature on threshold voltage?
11. What is the effect of temperature on mobility?
12. What are the different types of scaling?
13. What is stage ratio?
14. What is charge sharing on a bus?
15. What is electron migration and how can it be eliminated?
16. Can both pmos and nmos transistors pass good 1 and good 0? Explain.
17. Why is only nmos used in pass transistor logic?
18. What are the different methodologies used to reduce the charge sharing in dynamic logic?
19. What are setup and hold time violations? How can they be eliminated?
20. Explain the operation of basic sram and dram.
21. Of Read and Write operations, which ones take more time? Explain.
22. What is meant by clock race?
23. What is meant by single phase and double phase clocking?
24. If given a choice between NAND and NOR gates, which one would you pick? Explain.
25. What are stuck-at faults?
26. What is meant by ATPG?
27. What is meant by noise margin in an inverter? How can you overcome it?
28. Why is size of pmos transistor chosen to be close to three times of an nmos transistor?
29. Explain the origin of the various capacitances in the mos transistor and the physical reasoning behind it.

30. Why should the number of CMOS transistors that are connected in series be reduced?
31. What is charge sharing between bus and memory element?
32. What is crosstalk and how can it be avoided?
33. Two inverters are connected in series. The widths of pmos and nmos transistors of the second inverter are 100 and 50 respectively. If the fan-out is assumed to be 3, what would be the widths of the transistors in the first inverter?
34. In the above situation, what would be the widths of the transistors if the first inverter is replaced by NAND and NOR gates?
35. What is the difference between a latch and flip-flop? Give examples of the applications of each.
36. Realize an XOR gate using NAND gate.
37. What are the advantages and disadvantages of Bi-CMOS process?
38. Draw an XOR gate with using minimal number of transistors and explain the operation.
39. What are the critical parameters in a latch and flip-flop?
40. What is the significance of sense amplifier in an SRAM?
41. Explain Domino logic.
42. What are the differences between PALs, PLAs, FPGAs, ASICs and PLDs?
43. What are the advantages of depletion mode devices over the enhancement mode devices?
44. How can the rise and fall times in an inverter be equated?
45. What is meant by leakage current?
46. Realize an OR gate using NAND gate.
47. Realize an NAND gate using a 2:1 multiplexer.
48. Realize an NOR gate using a 2:1 multiplexer.
49. Draw the layout of a simple inverter.
50. What are the substrates of pmos and nmos transistors connected to and explain the results if the connections are interchanged with the other.
51. What are repeaters in VLSI design?
52. What is meant by tunneling problem?
53. What is meant by negative biased instability and how can it be avoided?
54. What is Elmore delay algorithm?
55. What are false and multi cycle paths?
56. What is meant by metastability?
57. What are the various factors that need to be considered while choosing a technology library for a design?
58. What is meant by clock skew and how can it be avoided?
59. When stated as 0.13µm CMOS technology, what does 0.13 represent?
60. What is the effect of Vdd on delay?
61. What are the various limitations in changing the voltage for less delay?
62. What is the difference between testing and verification?
63. While trying to drive a huge load, driver circuits are designed with number of stages with a gradual increase in sizes. Why is this done so? What not use just one big driver gate?
64. What is the effect of increase in the number of contacts and vias in the

interconnect layers?
65. How does the resistance of the metal layer vary with increasing thickness and increasing length?
66. What is the effect of delay, rise and fall times with increase in load capacitance?
67. In a simple inverter circuit, if the pmos in the Pull-Up Network is replaced by an nmos and if the nmos in the Pull-Down Network is replaced by a pmos transistor, will the design work as an non-inverting buffer? Justify your answer.

Q1) Convert D-FF into divide by 2. (not latch)
What is the max clock frequency the circuit can handle, given the following information?
T_setup= 6nS
T_hold = 2nS
T_propagation = 10nS

Q1: Ans

Circuit:
Connect Qbar to D and apply the clk at clk of DFF and take the O/P at Q. It gives freq/2.

Max. Freq of operation:
1/ (propagation delay+setup time) = 1/16ns = 62.5 MHz

Question(2)
Why do we gradually increase the size of inverters in buffer design when trying to drive a high capacitive load? Why not give the output of a circuit to one large inverter
Ans2)
We cannot use a big inverter to drive a large output capacitance because, who will drive the big inverter? The signal that has to drive the output cap will now see a larger gate capacitance of the BIG inverter.So this results in slow raise or fall times .A unit inverter can drive approximately an inverter thats 4 times bigger in size. So say we need to drive a cap of 64 unit inverter then we try to keep the sizing like say 1,4,16,64 so that each inverter sees a same ratio of output to input cap. This is the prime reason behind going for progressive sizing.
Question(3)
Why don?t we use just one NMOS or PMOS in a transmission gate?
ANSWER : Q3

PMOS degrades Logic 0 & NMOS degrades logic 1
To restore the logic levels to full, both NMOS & pMOS will be used together in TG

Ans(3)
Using only an nmos will result in an poor 1. Why is it so? Assume the gate voltage on NMOS is 5V. If we connect Drain to 5V, and the source is initially at 0, NMOS will turn on as long as there Vgs >Vth, this means, once the source reaches 4.3V (Assuming Vth=0.7), the nmos will turn off and there will be no more increase in source voltage.Similarly the opposite happens with PMOS, it doesn't give us a clean 0, but it can give a full 5V. So we use a combination of both NMOS and PMOS so that our signal doesn't get degraded by Vth on either side of VDD and GND.

Question(4)
Is there in Hold violation in the Circuit of Q1?
Describe clearly when there will be Hold violation wrt to the given data and how we can solve it in circuit level?

Q(5)

In CMOS technology, in digital design, why do we design the size of pmos to be higher than the nmos.What determines the size of pmos wrt nmos. Though this is a simple question try to list all the reasons possible..

Ans(5)
In PMOS the carriers are holes whose mobility is less[aprrox half] than the electrons, the carriers in NMOS. That means PMOS is slower than an NMOS. In CMOS technology, nmos helps in pulling down the output to ground ann PMOS helps in pulling up the output to Vdd. If the sizes of PMOS and NMOS are the same, then PMOS takes long time to charge up the output node. If we have a larger PMOS than there will be more carriers to charge the node quikly and overcome the slow nature of PMOS . Basically we do all this to get equal rise and fall times for the output node.

Q3: Why PMOS and NMOS are sized equally in a Transmission Gates?

In Transmission Gate, PMOS and NMOS aid each other rather competing with each other. Thats the reason why we need not size them like in CMOS.
In CMOS design we have NMOS and PMOS competing which is the reason we try to size them proportional to their mobility.

Q(6)

How many unique boolean fucntions can be there for n number of inputs?
Number of unique boolean function for n variables is 2^(2^n).
I did this by taking n=1,2 and applied the theory of induction.
Srikanth, If you can explain it ,would be great... A good question this one, since at first look I thought the answer to be 2^n....

For n number of inputs, the possible number of min terms or max terms, k = 2^n

To form any bollean function, we can take any combination of these.
So possible boolean functions = kc0 + kc1 + kc2 ...+kck = 2^k = 2^2^n

Q7

Design a sequential circuit which cuts the every second pulse from the input(clk)?

Hint: If we think in other way, it is nothing but frequency devider by 2 , But with 25% Duty cycle!!!

First do a Divide by 2 counter, ie connect Qbar to D input of FF.
Connect the Q output and CLK to a 2 input AND gate, this will gives us a divide by 2 clock with 25% duty cycle.

Question8

Guys this is the basic question asked most freqently. Design all the basic gates(NOT,AND,OR,NAND,NOR,XOR,XNOR) using 2:1 Multiplexer.
Ans8
Using 2:1 Mux, (2 inputs, 1 output and a select line)
(a) NOT
Give the input at the select line and connect I0 to 1 & I1 to 0. So if A is 1, we will get I1 that is 0 at the O/P.
(b) AND
Give input A at the select line and 0 to I0 and B to I1. O/p is A & B
(c) OR
Give input A at the select line and 1 to I1 and B to I0. O/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 I0 and ~B at I1. ~B can be obtained from (a)
(g) XNOR
A at the select line B at I1 and ~B at I0

Q9

N number of XNOR gates are connected in series such that the N inputs (A0,A1,A2......AN-1) are given in the following way:
A0 & A1 to first XNOR gate and
A2 & O/P of First XNOR to second XNOR gate
and so on..... Nth XNOR gates output is final output. How does this circuit work? Explain in detail?
Ans9

If N=Odd, the circuit acts as even parity detector, ie the output will 1 if there are even number of 1's in the N input...This could also be called as odd parity generator since with this additional 1 as output the total number of 1's will be ODD

If N=Even, just the opposite, it will be Odd parity detector or Even Parity Generator...

Can any one explain me the significance of current mirror? Thanks in advance.

Current mirrors are the most widely use analog circuit. Most of the transistors in an analog integrated circuit are parts of current mirrors.

Various Applications:
1. Current mirrors are used as current sources in ICs. An ideal current source has an infinite output impedance. That is, the output current does not change, even for large swings in output voltage. delta I/delta V = 0. That's high impedance.
2. The current mirroes are used for biasing and as loads in case of Amplifiers. A current source is equal to a very high resistive load(as mentioned in 1), If u use the same value resistor, it occupies too much of the area.

A typical current mirror cicuit can be designed by using either BJTs or MOSFETs

Q10)
All of us know how an inverter works. What happens when the PMOS and NMOS are interchanged with one another in an inverter?
I have seen similar Qs in some of the discussions.
If the source & drain also connected properly...it acts as a buffer.
But suppose input is logic 1 O/P will be degraded 1
Similarly degraded 0;
Q11)
A good question on Layouts...
Give 5 important Design techniques you would follow when doing a Layout for Digital Circuits
Ans11
But I remember slight rules in making the standard cells...
like
1. Total height some 80, above 40 Pull up network below 40 NMOS,pulldown
2. VDD, GND Standard widths
3. The parallel PMOS nearer to the power supply etc..

2) Use as much less poly as possible, since resistance of poly is 100 times more than that of the metal layers.
3) Avoid routing spaces by using different metal layers available now. Also Bit wise design approach can save lot of routing wires.
4) Distribute Vdd, gnd and clk are distributed uniformly throughout the layout.

1.In digital design, decide the height of standard cells you wanna layout.It depends upon how big your transistors will be.Have reasonable width for VDD and GND metal paths.Maintaining uniform Height for all the cell is very important since this will help you use place route tool easily and also incase you wanna do manual connection of all the blocks it saves on lot of area.
2. Use one metal in one direction only, This does not apply for metal 1. Say you are using metal 2 to do horizontal connections, then use metal 3 for vertical connections, metal4 for horizontal, metal 5 vertical etc...
3.Place as many substrate contact as possible in the empty spaces of the layout.
4.Do not use poly over long distances as it has huge resistances unless you have no other choice.
5.Use fingered transistors as and when you feel necessary.
6.Try maintaining symmetry in your design. Try to get the design in BIt Sliced manner.

Q12)
What is Latch Up? How do you avoid Latch Up?

Coming to latchup problem,
If you look at the cross cectional view of any inverter, there is a positive feedback between a NPN and PNP transistor which results in latch up problem.This positive feedback results in excessive current which ultimately destroys the device. These NPN and PNP transistors are formed from the p+/n+ source/drains, p/n well and substrates
Q13)

Implement the following circuits:
(a) 3 input NAND gate using min no of 2 input NAND Gates
(b) 3 input NOR gate using min no of 2 inpur NOR Gates
(c) 3 input XNOR gate using min no of 2 inpur XNOR Gates
Assuming 3 inputs A,B,C

3 input NAND:
Connect :
i) A and B to the first NAND gate
ii) Output of first Nand gate is given to the two inputs of the second NAND gate (this basically realises the inverter functionality)
iii) Output of second NAND gate is given to the input of the third NAND gate, whose other input is C

((A NAND B) NAND (A NAND B)) NAND C

Thus, can be implemented using '3' 2-input NAND gates. I guess this is the minimum number of gates that need to be used.

3 input NOR:
Same as above just interchange NAND with NOR
((A NOR B) NOR (A NOR B)) NOR C

3 input XNOR:
Same as above except the inputs for the second XNOR gate, Output of the first XNOR gate is one of the inputs and connect the second input to ground or logical '0'

((A XNOR B) XNOR 0)) XNOR C

Name the logical gates for which the 3input implementation can not be obtained from 2 2input gates? Explain.
For this Qs also the answer is same.

Where as AND, OR XOR 3 inputs can be easily obtained from 2 2input gates.

Q14:
An assembly line has 3 fail safe sensors and one emergency shutdown switch.
The line should keep moving unless any of the following conditions arise:
(i) If the emergency switch is pressed
(ii) If the senor1 and sensor2 are activated at the same time.
(iii) If sensor 2 and sensor3 are activated at the same time.
(iv) If all the sensors are activated at the same time
Suppose a combinational circuit for above case is to be implemented only with NAND Gates. How many minimum number of 2 input NAND gates are required?

No of 2-input NAND Gates required = 6
You can try the whole implementation also
Q15:
What is race-around condition? Explain in it in case of R-S Latch and solution to avoid that?

The race around conidtion is : O/P osciallting between 0s & 1s.
This problem will occur in Latches especially if the clock is high for long time.

I can explain in case of J-K Latch
Suppose J=K=1 & O/P = Compliment of prev state (Q(t+1) = Q(t)')
So as far as clock is high, O/P oscialltes between 0 & 1

To avoid this,
they use Master-Slave configuration.
Q16:

What is metastability? When/why it will occur? Different ways to avoid this?
Ans : Q16

Metastable state: A un-known state in between the two logical known states.
This will happen if the O/P cap is not allowd to charge/discharge fully to the required loigcal levels.

One of the cases is: If there is a setup time violation, metastability will

occur,
To avoid this, a series of FFs is used (normmly 2 or 3) which will remove the intermediate states.

Q17:

Give the basic schematic of Set-Reset Latch with NOR gates. Explain the functionality with truth tables. Which input combination is not allowed.

Change the same to provide clock enable.

Try with NAND gate also. Which I/P combination is not allowed?

Latches
How can we make a circuit out of gates that is not combinatorial? The answer is feed-back, which means that we create loops in the circuit diagrams so that output values depend, indirectly, on themselves. If such feed-back is positive then the circuit tends to have stable states, and if it is negative the circuit will tend to oscillate.

A latch has positive feedback. Here is an example of a simple latch:

This latch is called SR-latch, which stands for set and reset.

It is not practical to use the methods that we have used to describe combinatorial circuits to describe the behavior of the SR-latch. Later, we will show a method for describing flip-flops and clocked sequential circuits. For now, we just rely on our intuition to describe how latches work.

The SR-latch is meant to have at most one of its inputs equal to 1 at any time. When both of its inputs are 0 it has two different stable states possible. Either x is 0, in which case we have the following signal values:

or else x is 1, in which case we have the following signal values:

The actual value depends on the history of input values as we will show next.

Now suppose that s is 1 (and therefore r is 0 since we allow at most one input to be 1 at any time). We get the following signal values:

The 1 on the s input makes sure the output of the upper nor-gate is 0, and the two 0s on the input of the lower nor-gate make sure the x output is 1.

Now suppose the s input goes from 1 to 0, while the r input remains at 0. The second input of the upper nor-gate is 1, so the transition from 1 to 0 of the s input, does not make any difference. The x output remains at 1. In this case, if the s and r inputs are both 0, there is only one possible stable state, the one that gives x the value 1.

Conversely, suppose that r is 1 (and therefore s is 0 since we allow at most one input to be 1 at any time). We get the following signal values:

The 1 on the r input makes sure the x output is 0, and the two 0s on the input of the upper nor-gate make sure the output of the upper nor-gate is 0.

Now suppose the r input goes from 1 to 0, while the s input remains at 0. The second input of the lower nor-gate is 1, so the transition from 1 to 0 of the r input, does not make any difference. The output of the upper nor-gate remains at 1. In this case, if the s and r inputs are both 0, there is only one possible stable state, the one that gives x the value 0.

From the discussion above, we conclude that the SR-latch is able to remember the last state of the inputs, in the sense that it remembers which of the two inputs, s or r, last had the value of 1.
Q18:

Give the state machine for a serial two's complimenter? Then design the complete circuit using DFF?
Hint: If you observe a binary number and its 2's compliment, the 0s will be retained until the first 1 occurs (from LSB side) and the first 1 also will be retained after that compliment all the follwoing bits.

It is very interseting & simple Qs to check the knowledge of state M/Cs
Q18:

The state M/C will have only two states. State A & B. Stay in State A as far as you are getting 0's and O/P is also 0. If 1 comes go to state B and O/P is 1.
If you are in state B, whether I/P is 1 or 0 stay in B only and O/P is compliment of input.

State Table is as follows:
PS x NS O/P
0 0 0 0
0 1 1 1
1 0 1 1
1 1 1 0

To obtain the circuit using DFF,
OR x & Q of FF and give at the I/P of FF
XORing of Q&x will give O/P(2's compliment)
Q19:
In a PLL, what elements(digital blocks) can be used to design the phase detector?
Ans: 19

1. XOR Gate
2. S-R Latch
3. PFD(Phase/freq detector) : It is designed from FFs & some NAND Gate COnnected to resetes)

[b]Q20:[/b]
Describe a finite state machine that will detect three consecutive coin tosses (of one coin) that results in heads.

Assume state A : no heads has occured
state B: only one head has occured
state C: more than 2 heads has occured

So intial state is A

PS I/P NS O/P
A Tail A 0
A Head B 0
B Tail A 0
B Head C 0
C Tail A 0
C Head C 1

[b]Q21:[/b]

What is Moore model & Mealy model? Explain.

Ans: Q21

A state machine consists of set of states, initial state, input symbols and transition function that maps input symbols and current state to next state.

Mealy machine: machines having outputs associated with transition of input and the current state. So in this case, the O/P will vary as soon as the input varies..O/P can't be held until clock pulses.

Moore machine: machines having outputs associated with only states. The O/P will vary only after the states are varied.So the cahgnes in the O/P will be only during clock transitions.

Adv & Disadv:
In Mealy as the output variable is a function both input and state, changes of state of the state variables will be delayed with respect to changes of signal level in the input variables, there are possibilities of glitches appearing in the output variables.
Moore overcomes glitches as output dependent on only states and not the input signal level.

All of the concepts can be applied to Moore-model state machines because any Moore state machine can be implemented as a Mealy state machine, although the converse is not true.

Moore machine:
the outputs are properties of states themselves... which means that you get the output after the machine reaches a particular state, or to get some output your machine has to be taken to a state which provides you the output.

The outputs are held until you go to some other state

Mealy machine:
Mealy machines give you outputs instantly, that is immediately upon receiving input, but the output is not held after that clock cycle.
Q22)

How many minimum number of MOS transistors are required to implement a Full Adder using CMOS technology?

Q23:[/color]
(a) Show all the possible ways to convert a 2-input NAND Gate into an

inverter?
(b) Show the implementaion of XOR Gate using minimum number of 2-input NAND Gates?

Ans: Q23:
(a) A 2 input NAND gate can be converted into an inverter in two ways: one way is by tieing up the two inputs and give the input, second give make one of the two inputs permanatly high and give the input at the other input.

(b) A XOR B = AB' + A'B = (AB)' A + (AB)' B
So one 2-input NAND is needed to generate (AB)' 3 other to implent the rest of boolean function. So total we need 4 2input NAND Gates.

Q24:
It is required to connect a Master, which generates data @ 200 Mega Samples/sec to a Slave which can receive the data @ 10 Mega Samples/Sec. If the data lost in 10Micro Sec, what is the optimal size of FIFO to be used to avoid lose of data?
Ans : (200-10) * 10 = 1900 samples is the size of FIFO

It is given that data will lost in 10micro sec otherwise, that is if data comes continuosly,fifo size will be infinity.

What is the difference between inertial and transport delays?
What are the applications?

The inertial delay model is specified by adding an after clause to the signal assignment statement. To model this delay in the SR latch example, we could replace the two signal assignments with the following two statements.

```
q <= r nor nq after 1ns;
nq <= s nor q after 1ns;
```

Now during simulation, say signal r changes and will cause the signal q to change, rather than schedule the event on q to occur during the next round, it is scheduled to occur 1ns form the current time. Thus the simulator must maintain a current time value. When no more events exist to be processed at the current time value, time is updated to the time of the next earliest event and all events scheduled for that time will be processed.

Note that the change will not occur in q until 1ns after the change in r.

Likewise the change in nq will not occur until 1ns after the change in q. Thus, the "after 1ns" models an internal delay of the nor gate.

Typically, when a component has some internal delay and an input changes for a time less than this delay, then no change in the output will occur. This is also the case for the inertial delay model.

Although most often the inertial delay is desired, sometimes all changes on the input should have an effect on the output. For example, a bus experiences a time delay, but will not "absorb" short pulses as with the inertial delay model. In these cases, we can use [color=red]transport delay model[/color]. The transport delay model just delays the change in the output by the time specified in the after clause. You can elect to use the transport delay model instead of the inertial delay model by adding the keyword transport to the signal assignment statement.

The SR latch example could be modified to use the transport delay model by replacing the signal assignments with the following two statements.

```
q<=transport r nor nq after 1ns;
nq<=transport s nor q after 1ns;
```

Following simple example can illustrate the concept.

```
module delay(in,transport,inertial);
input in;
output transport;
output inertial;

reg transport;
wire inertial;

// behaviour of delays
always @(in)
begin
transport <= #10 in;
end

assign #10 inertial = in;

endmodule // delay
```

The timing Diagram for input and outputs

```
_______ __
in _____| |_____||_______

_______ __
transport _________| |_____||_____

_______
inertial _________| |____________
```

Non blocking assignment gives you transport delay. Whenever input changes, output is immediately evaluated and kept in a event queue and assigned to output after specified "transport" delay.

In Continuous assign statement the latest event overrides the earlier event in the queue.

Here is a rudimentary testbench and its output. Hope this helps.

```
module test;
reg in;
wire transport, inertial;

// instantiate delay module
delay my_delay(in,transport,inertial);

// apply inputs
initial
begin
in = 0;
#20 in = 1;
#20 in = 0;
#30 in = 1;
#5 in = 0;
#30 in = 1;
#30 $finish;
end
// monitor signals
initial
begin
$monitor($time," in = %b transport = %b inertial = %b",
in,transport, inertial);
```

```
end

endmodule // test
```

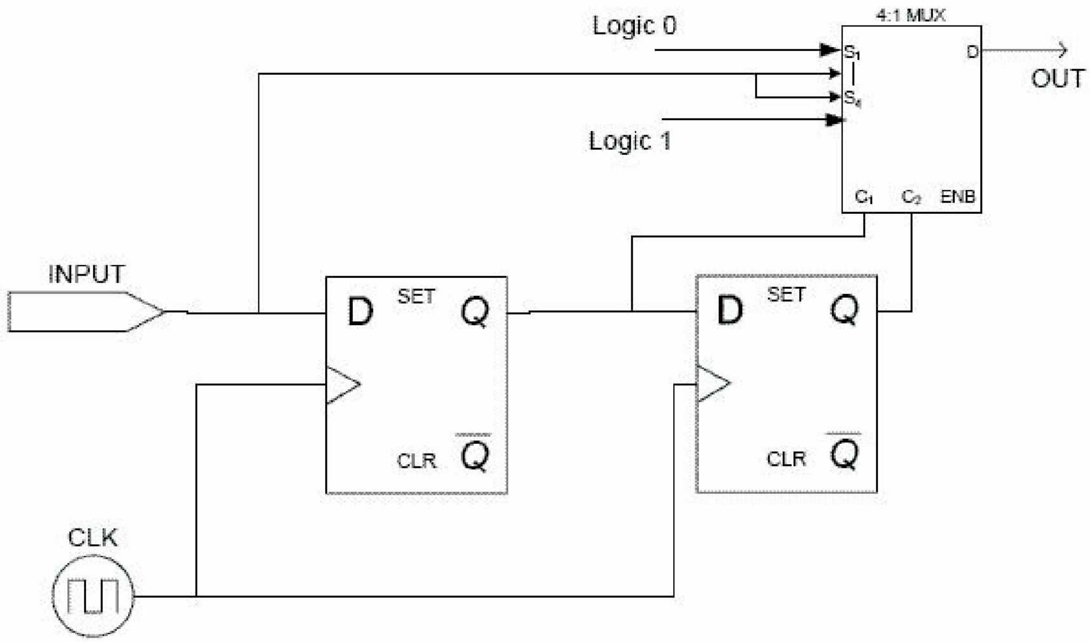
log file
Compiling source file "delay.v"
Highest level modules:
test

0 in = 0 transport = x inertial = x
10 in = 0 transport = 0 inertial = 0
20 in = 1 transport = 0 inertial = 0
30 in = 1 transport = 1 inertial = 1
40 in = 0 transport = 1 inertial = 1
50 in = 0 transport = 0 inertial = 0
70 in = 1 transport = 0 inertial = 0
75 in = 0 transport = 0 inertial = 0
80 in = 0 transport = 1 inertial = 0
85 in = 0 transport = 0 inertial = 0
105 in = 1 transport = 0 inertial = 0
115 in = 1 transport = 1 inertial = 1
L35 "delay.v": $finish at simulation time 135
81 simulation events

Q27:

Design a synchronous sequential circuit to check the highest number of ones and zeros in the last 3 input samples. Your ckt should give 1 at the O/P if the last 3 samples at the input has more 1's similarly 0 when the no. of zeros is high.
Eg:
IN : 001110110000
OUT: 0111111000

Constraints:
1) You are supposed to use only Multiplexers and DFFs for your design. No external gates. To be specific, 1 4:1 Mux only.
2) Design should be optimized one.
3) Only one clock is available to you. And it is given that the input is sampled at that clock rate only.



Q28 What is overflow? How can you detect overflow in signed and unsigned numbers?

Ans: Q28:

case1 : Unsigned numbers:
In N-bits, we can represent numbers from 0 to (2^N) - 1. Suppose if we are adding 2 N bit unsigned numbers and if the result is greater than (2^N) - 1 , overflow will occur. To detect this, check whether the MSB addition (Nth bit) + Carry generated from N-1bit addition is genrating any carry or not. If there is carry out, there is overflow.

case2 : Signed numbers:
In N-bits, we can represent numbers from -(2^(N-1)) to (2^(N-1)) - 1. Suppose if we are adding 2 N bit signed numbers and if the result is not in the above range , overflow will occur. To detect overflow in this case : if two numbers with the same sign (either positive or negative) are added and the result has the opposite sign, an overflow has occurred.

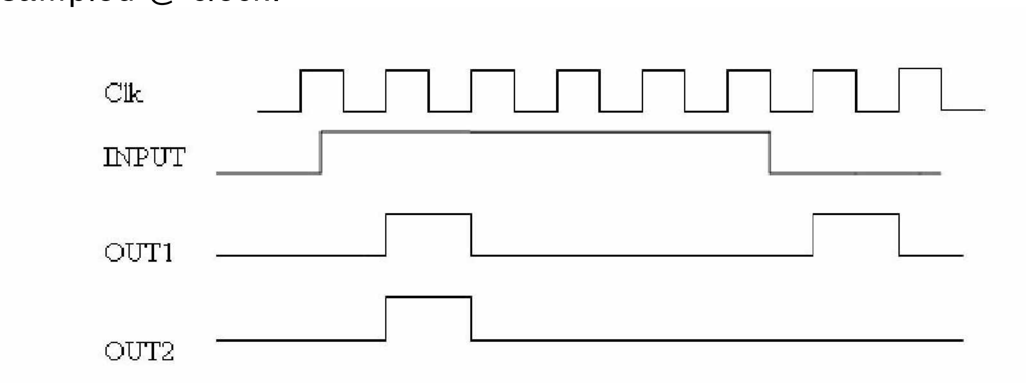
more cases in case of signed...
So overall, the conidtions to detect overflow are here again:

In unsigned arithmetic a carry out of the most significant digit means that there has been an overflow

A signed overflow has occurred in an addition or subtraction if:

* the sum of two positive numbers is negative;
* the sum of two negative numbers is positive;
* subtracting a positive number from a negative one yields a positive result; or
* subtracting a negative number from a positive one yields a negative result.

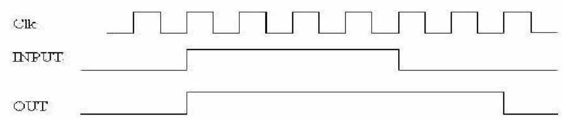
To solve this type of sequential problems, using synchronous methods, we should be having the input sampled @ clk. If that is not the case, it is our responsibilty to make it align to the clock.
If you observe the following waveforms, the [b]INPUT is not alinged with the rising edge of clock, So we will use one Flip flop to make it proper. So the purpose of the first flip flop, is to make the INPUT proper. In few cases this may not be required. Suppose if we are working with FSMs, the main assumption itself is the data is sampled @ clock.

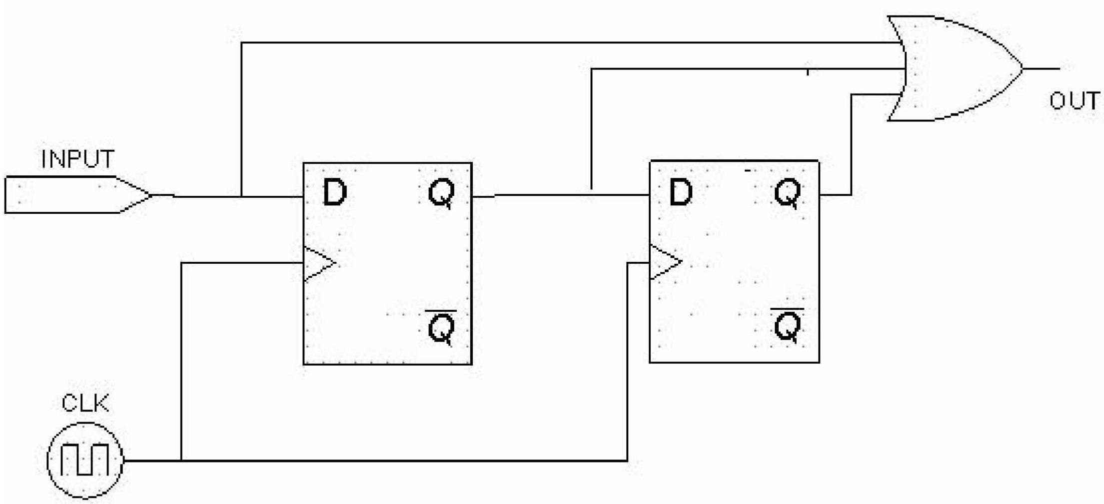


Now to get the Output, of you observe OUT1 is changing at both the rising edge and falling edge of the input where as OUT2 is changing only at the rising edge. What this effectively mean is, OUT1 has to be 1 for one cycle, when the current sample and prev sample of INPUT are opposite whereas OUT2 will be 1 if the current input is 1 and prev is 0.

So toget prev sample...I have to store it...so need second flop...
So first flops o/p will give current value whereas second flops out will give prev value...
Using these to we build the combinational logic.[/img]

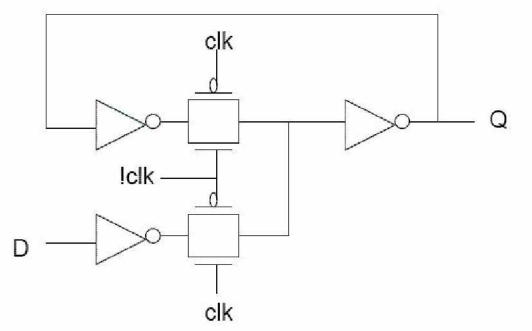
Q29:
Give the circuit to extend the falling edge of the input by 2 clock pulses. The waveforms are shown in the following figure.

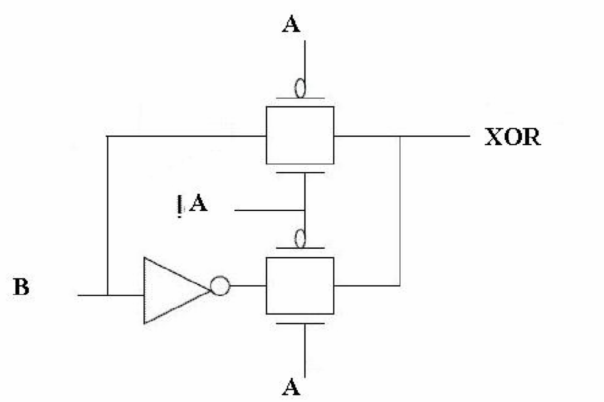




Q30)
(a) Draw a Transmission Gate-based D-Latch.
(b) Design a Transmission Gate based XOR. Now, how do you convert it
to XNOR? (Without inverting the output)

NOTE on TG: Transmission Gate has one NMOS & one PMOS (pass transistors). In the symbol bubble indicates PMOS and other side is NMOS. To select TG, We need to give 0 to PMOS and 1 to NMOS. In this case whatever is there at the input will be connected to the output. In the other case, that is 0 is given NMOS and 1 is given to PMOS, output will be just hanging. We need keep these things in solving TG based problems. The answers to above mentioned problems will be updated shortly.





Q31,Q32,Q33:

Design the Digital Circuit which gives
(Q 31) fout = (1/2) fin
(Q 32) fout = (1/3) fin
(Q 33) fout = (2/3) fin (3 different circuits)

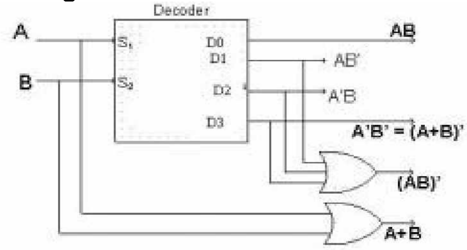
NOTE:
(a) fout is O/P freq and fin is I/P freq
(b) Duty cycles are also not mentioned..so its okay to design for any duty cycle.
(c) All the ckts design using DFFs and min no of external gates

Show all the waveforms also.....

Q34:
You are given a 2:4 decoder. 2 input OR gate and one 3 input OR gate. Using these Components design the following system whcih takes A & B as inputs and generates the 4 O/Ps : AB, (AB)' , A+B, (A+B)' .

2:4 decoder will have 4 O/Ps which are the minterm/maxterms of the 2 inputs. So the O/P are AB, AB', A'B, A'B'. So AB and (A+B)' = A'B' are directly the O/P s of decoder. Whereas A+B can be obtained using

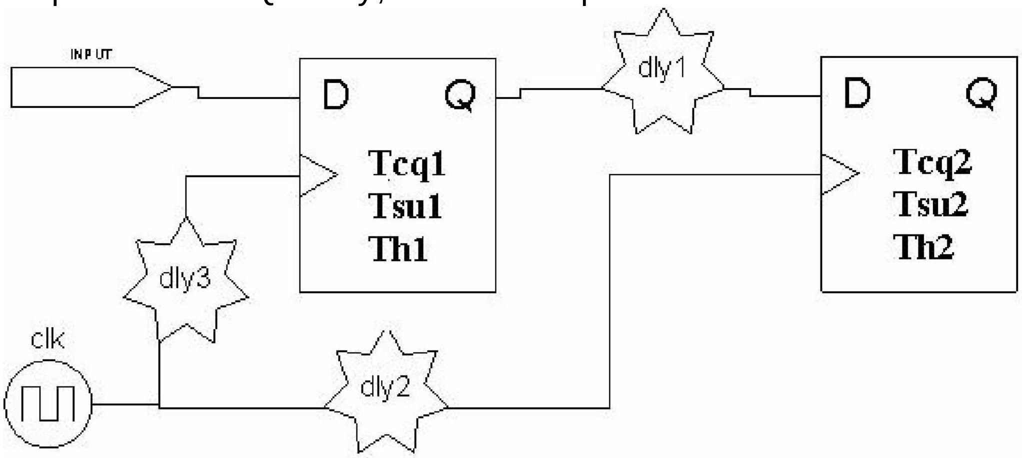
2 input OR gate(which is given). So only O/P that is needed is (AB)' = A' + B' = A(B+B') + B(A+A') = AB' + A'B + A'B'. That is , use 3-input OR gate for this. The whole design is shown below.



Q35:

The following digital circuit shows two flops with a logic delay (dly1) in between and two clock buffer delays (dly2, dly3). Derive the conditions in terms of (dly1,dly2,dly3) to fix setup and hold timing violations at the input of second FF?

Tcq -- Clock to Q delay, Tsu -- Setup time and Th -- hold time.



The above waveforms show the CLK, CLK1 and CLK2. The input waveform at FF1 is assumed and the input of FF2 is shown accordingly with all the given delays and clock-to-Q delays.

From the waveforms it is clear that, to avoid setup time violation,

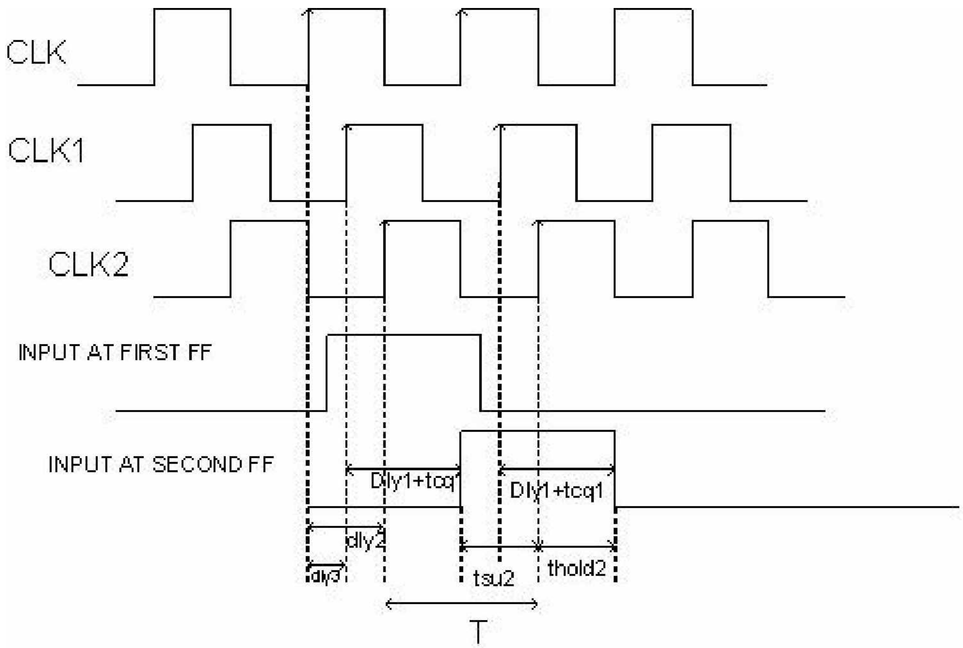
T >= (Tsu2 + Tcq1 + dly1 +delta) where delta = dly2-dly3 (assuming +ve skew) ---> (1)

From this equation we can get maximum freq of operation.

To avoid hold time violation,

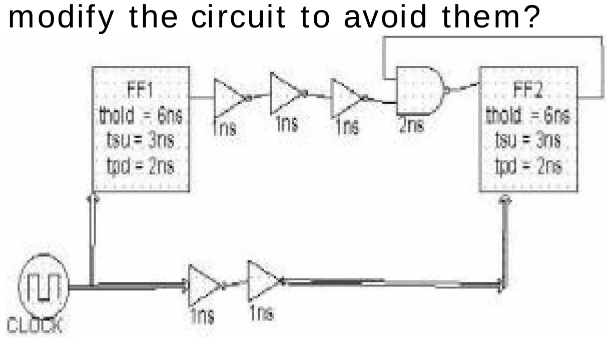
Th2 <= Tcq1 + dly1 +delta ---> (2)

These two equations can be used as generalized equations to solve setup time/hold time problems. This works only for synch circuits. If one clock works at pos edge and other is negative edge we need to derive one more set of equations. That also we will at later section.



You have to obtain the value of dly using hold time violation eq and substitute that value in setup time equation to get the maximumfreq of operation.

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



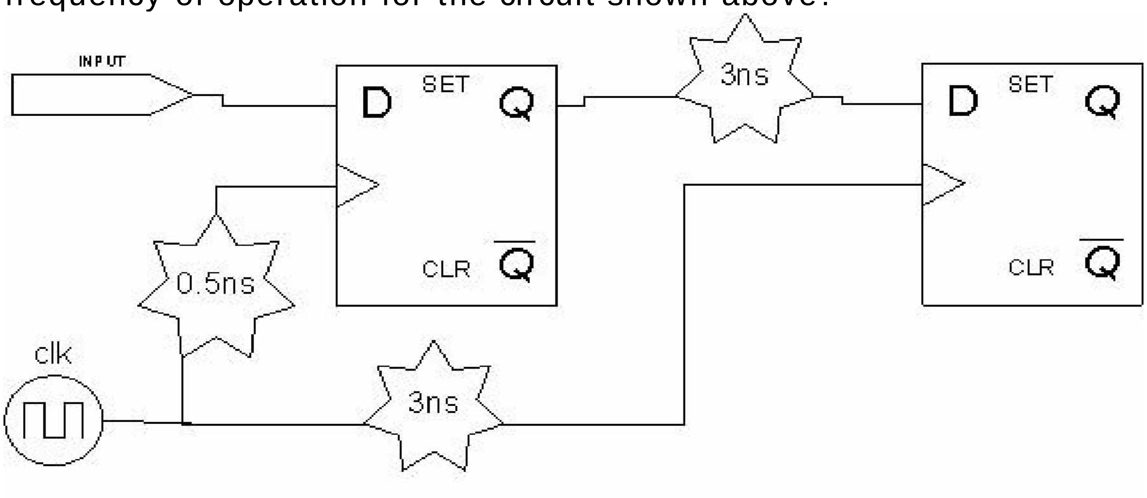
Ans36
The minumum time period = 3+2+(1+1+1) = 8ns
Maximum Frequency = 1/8n= 125MHz

There is a hold violation in th ecircuit.You can avoid it by giving the input to the AND gate through two inverters.

Ans: Q36
In this diagram,
dly3 = 0
dly2 = 2ns
so,delta = 2ns
tsu2 = 3ns, tcq1 = 2ns, dly1 = 5ns

Putting all these values in Eq(1) ,
T >= Tcq1 + dly1 + Tsu2 - delta
so, T >= 2 + 5 + 3 - 2, T >= 8ns, f <= 1/8
Max freq of operation is 125MHz

And there is a hold time violation in the circuit, bcoz of feedback, if you observe, tcq2+andgate delay is less than thold2,To avoid this as mentioned in Q1 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.
Q37: If both the flip flops have same clock-to-Q delay of 2.5ns,setup time of 2ns and a hold time violation of 1ns, what is the maximum frequency of operation for the circuit shown above?



Ans: Q37:

Tcq1 = Tcq2 = 2.5ns
Tsu1 = Tsu2 = 2ns
Thold1 = Thold2 = 1ns
delta = clock_skew = 3 - 0.5 = 2.5ns

Equation for hold-violation is,

Thold <= dly + Tcq1 - delta (equation 2)
1 <= dly + 2.5 - 2.5
So dly >= 1ns

To obtain maximum freq, fixing it to lowest possible value, so *dly = 1ns*

Using this value and Eqauation (1) of setup time, we can obtain max freq.

T >= Tcq1 + dly +Tsu2 - delta
T >= 2.5 + 1 + 2 - 2.5
So T >= 3ns

Max freq of operation = 1/3ns = 333.33 MHz
Q38)

A simple question...
What is meant by CMOS Design ?

Ans38
CMOS design means complimentary metal oxide semiconductor design which involves the use of CMOS and PMOS in realizing the logic design. This is the dominant technology now a days because of its ten fold reduction of power dissipation which outweighs 30-50% speed reduction and size increase.
ANS 38. Gain Factor (read *Beta* as the symbol is not appearing)

Beta = µ.Cox (W/L)

= µ * Epsilon (again symbol not appearing) / tox (W/L)

where µ is effective surface mobility of the carriers in the channel, 'epsilon' is the permittivity of the gate insulator (SiO2), tox is oxide thickness of the gate insulator, W is width and L is Length of the channel. The gain factor *Beta* thus consists of a process dependent factor µ.Cox, where Cox (oxide capacitance) is given by 'epsilon'/tox. So in the above equation a designer of CMOS couldn't control over process dependent factor. Eventually what remain in the hands of the designer is W and L. Therefore, from designers point of view, 'designing' W and L is what i describe as CMOS Design
Q39) Two NMOS transistors are connected in series. The gate of each transistor is connected to 5v and the drain of one transistor is

connected to 12v. What is the voltage at the other end of the transistor if the threshold voltage of each is 1v?

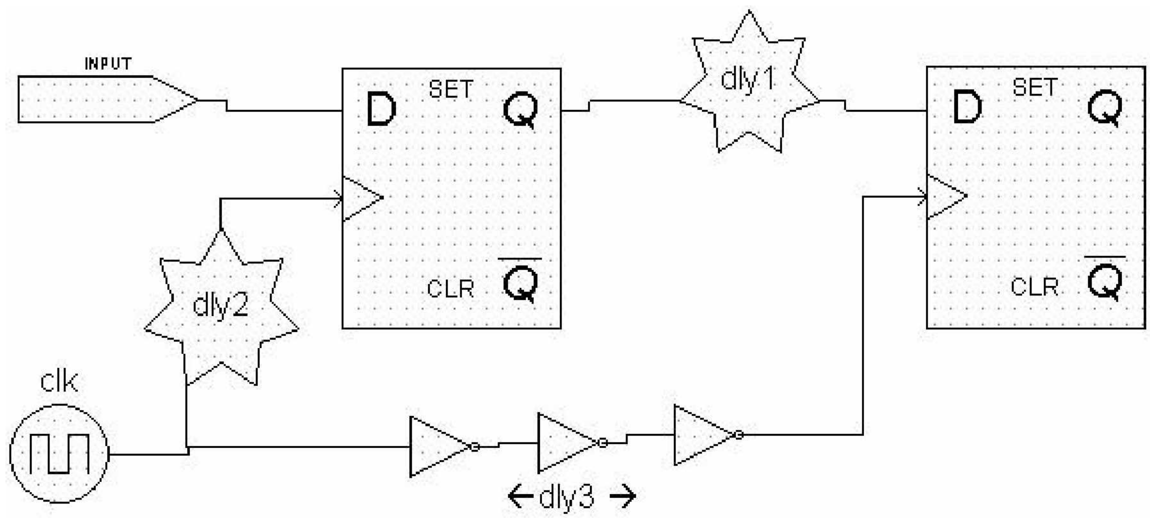
Basic equation is Vs=min of (vd,vg-vth) by papi.
The output voltage is 4V.

Consider a single NMOS as a switch.
The max voltage at the other end can reach max of VG - Vt, after that NMOS will be off.
So if the voltage at one end is less than VG-Vt it passes that value to the other end, but if it is more, it reaches VG-Vt at the end and stops there bcoz after that the MOS switch will be off.

So in this case, first NMOS which has 12v, at the input, gives 4v out at its source, the other Transistor which has 4v at the input transmits samething to the other end as it is.
Q40) Here is an interesting design question. There is a room which has two doors one to enter and another to leave. There is a sensor in the corridor at the entrance and also there is sensor at the exit. There is a bulb in the room which should turn off when there is no one inside the room. So imagine a black box with the inputs as the ouputs of sensors. What should the black box be?

The black box can be an up/down counter which can count number of people inside.
For 200 people, we need 8 bit counter.
So The O/P of entrance sensor will be used as enable for UP count and the other sensor at exit will be used for DOWN count, whenever the counter's O/P is 0, we can make the BULB OFF, Otherwise ON.

This is the easiest I can think about
Q41) Design a 2 bit up/down counter with clear.
Derive setup time/hold time vilation equations for the follwoing circuit? Assume Tcq1 Clock to Q delay, Tsu1 -- Setup time and Th1 hold time for first FF and similarly Tcq2,Tsu2,Th2 for second FF.



Ans:Q42:

Setup time :
(T/2) + delta >= Tcq1 + dly1 + Tsu2

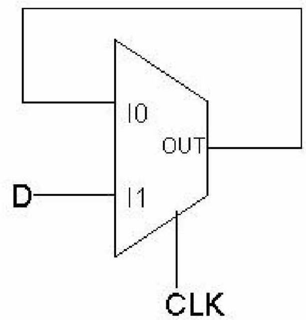
Hold time:
Th2 <= delta + Tcq1 + dly1

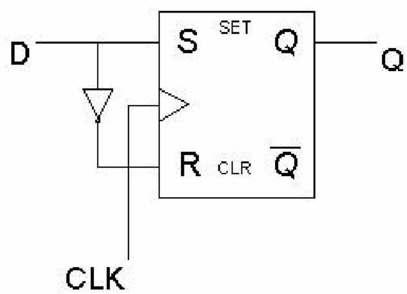
where delta = dly3 - dly2, assuming positive skew
and T is clock period.

Note: The procedure is same as that of Q37.Just draw the waveforms with proper delays, you will get above equations.

Q43:

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





Q44:

Suppose A & B are two unsigned n-bit numbers, how many minimum number of bits required for Y = A + B + (A*B). Here + is for addition and * is for multiplication. All are unsigned operations only

We need 2n bits for the operation. Take n=1,2,3,4 and take the maximum n-bit number and calculate Y , we end up with 2n bits for it.

Q45:

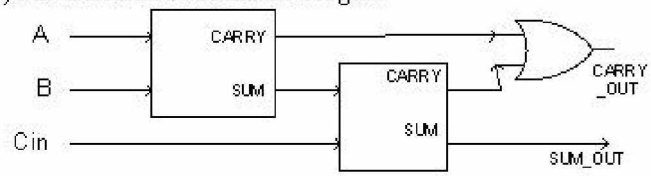
(a) Give the truth table of a Half Adder?

(b) Design a full adder from HA's? (You can use Min no. of external gates)

(a) Truth table of a HA:

A	B	Cout	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

(b) Full adder from 2 HA and one OR gate:



Q46:

(a) How will you count the number of 1's that are present in a given 3-bit input using full adder?
(b) If input is a 7-bit vector, how many minimum number of full adders are required to count the number of 1s?
i feel in this if we could look at the output of Full Adder as combined result.i.e.
if a,b,c are inputs to FA and S,C are outputs then the number of 1's is equal to the concatinated result of S and C in the order CS.
Q47) Came across this question from a friend.
Design a circuit that calculates the square of a number? It should not use any multiplier circuits. It should use Multiplexers and other logic

Assume its n-bit input. I guess there should be some general topology for it.
if not try doing it for 8 bit input.
A47)This is interesting....
1^2=0+1=1
2^2=1+3=4
3^2=4+5=9
4^2=9+7=16
5^2=16+9=25
and so on
See a pattern yet?To get the next square, all you have to do is add the next odd number to the previous square that you found.See how 1,3,5,7 and finally 9 are added.Wouldn't this be a possible solution to your question since it only will use a counter,multiplexer and a couple of adders?It seems it would take n clock cycles to calculate square of n.
Q. 46 What is the purpose of DRC?
This was asked in an interview.

Coming to the answer- DRC is used to check whether the particular schematic and corresponding layout(especially the mask sets involved) cater to a pre-defined rule set depending on the technology used to design. They are parameters set aside by the concerned semiconductor manufacturer with respect to how the masks should be placed , connected , routed keeping in mind that variations in the fab process does not effect normal functionality. It usually denotes the minimum allowable configuration.

q.47
What is LVs and why do we do that. What is the difference between LVS and DRC?

The layout must be drawn according to certain strict design rules.
DRC helps in layout of the designs by checking if the layout is abide by those rules.
After the lalyout is complete we extract the netlist.
LVS compares the netlist extracted from the layout with the schematic to ensure that the layout is an identical match to the cell schematic.

Q48: What is DfT ?

A 48
DFT means design for testability.
'Design for Test or Testability' - a methodology that ensures a design works properly after manufacturing, which later facilitates the failure analysis and false product/piece detection
Other than the functional logic,you need to add some DFT logic in your design.
This will help you in testing the chip for manufacturing defects after it come from fab.
Scan,MBIST,LBIST,IDDQ testing etc are all part of this.
(this is a hot field and with lots of opportunities)

Q49. What is MBIST, and LBIST?

MBIST(memory bulit in self test) is a wrapper logic around memory.This will test the memory for any manufacturing defects.
It will generate write and read transactions and check the data.(am i right gopi??)
MBIST is for only testing memories whereas LBIST(Logic BIST) is used for testing any logic.
LBIST can also be run at speed.Thats one advantage of it i guess(right gopi???)

PS:
LBIST was started by an ex bitsian.

MBIST(Memory Built-In Self Test), as its name implies, is used specifically for testing memories. It typically consists of test circuits that *apply*, *read*, and *compare* test patterns designed to expose defects in the memory device.
There now exists a variety of industry-standard MBIST algorithms, such as the "March" algorithm, the checkerboard algorithm, and the varied pattern background algorithm.

LBIST(Logic Built-In Self Test), is designed for testing random logic. It typically.. employs a PRPG-pseudo-random pattern generator to generate *input patterns* that are applied to the device's *internal scan chain*, and a MISR- multiple input signature register for obtaining the response of the device to these test input patterns. An incorrect MISR output indicates a defect in the device.

LBIST is started by BITSian.. this is something I have learnt from you. Shall propagate the same with in my circle

Q50: What is boundary scan??

Q51....can anybody explain in detail about +ve hold time and -ve hold time.I get quite confused with this always
A51:

Hold time can be negative meaning that data can be changed even before clock edge and still previous value will be stored

Negative hold time is existing particlarly in the case of clock skew.

Consider two FFs with a *clock skew* i.e FF1 lags behind FF2 and suppose FF1's output is fed to FF2's input passing through some logic cloud (or directly) *then* by the time FF1 is active the FF2 would have done its sampling the FF1's output.. there by no violation of hold time.

Some information regarding negative and setup time that might be of interest to you..

-A zero setup time means that the time for the data to propagate within the component and load into the latch is less than the time for the clock to propagate and trigger the latch.

-A zero hold time means either that the moment the clock is asserted, the latch no longer looks at its inputs, or else that the clock path delay is shorter than the data path delay.
-A *negative* setup or hold time means that there is an even larger difference in path delays, so that even if the data is sent later than the clock (for setup time), it still arrives at the latch first.

Typically manufacturers avoid specifying negative values since this restricts later design and manufacturing decisions, but they often specify zero values since this simplifies usage in a system.
Q52) What is the basic difference between Analog and Digital Design?
Digital design is distinct from analog design. In analog circuits we deal with physical signals which are continuous in amplitude and time. Ex: biological data, sesimic signals, sensor output, audio, video etc.

Analog design is quite challenging than digital design as analog circuits are sensitive to noise, operating voltages, loading conditions and other conditions which has severe effects on performance. Even process technology poses certain topological limitations on the circuit.
Analog designer has to deal with real time continuous signals and even manipulate them effectively even in harsh environment and in brutal operating conditions.
Digital design on the other hand is easier to process and has great immunity to noise.
No room for automation in analog design as every application requires a different design. Where as digital design can be automated.

Analog circuits generally deal with instantaneous value of voltage and current(real time). Can take any value within the domain of specifications for the device.consists of passive elements which contribute to the noise(thermal) of the circuit . They are usually more sensitive to external noise more so because for a particular function a analog design
uses lot less transistors providing design challenges over process corners and temperature ranges. deals with a lot of device level physics and the state of the transistor plays a very important role

Digital Circuits on the other hand deal with only two logic levels 0 and 1(Is it true that according to quantum mechanics there is a third logic level?) deal with lot more transistors for a particular logic, easier to design complex designs, flexible logic synthesis and greater speed although at the cost of greater power. Less sensitive to noise. design and analysis of such circuits is dependant on the clock. challenge lies in negating the timing and load delays and ensuring there is no set up or hold violation.

Q53) What is difference between Static Logic and Dynamic Logic

Static logic is when you provide a low resistance path from VDD or GND to the output. Basically in static logics the output is pulled high or low through a low resistance path
In case of Dynamic logic, as you sais an intermediate node is charged uo or down and that state is maintained via high impedance path...

Q54) In CMOS deisgn, given a choice between implementing a logic in NOR and NAND implementation, which one would you prefer and why?
Ans 54. If it is a complementary CMOS Nand is preffered over NOR as NOR has PMOS in series which slows it down.
If it is a pseudo-NMOS, NOR is preferred as it has Transistors in parallel.

Q55) What are the conditions for obtaining worst case set up and hold times?
My intention in asking this question is how can we obtain worst case set up and worst case hold times i.e. w.r.t process variations .
We obtain worst case set up time when it is a slow process, at high temperature and low voltage.
Similarly, the worst case hold time can be obtained if it is a fast process, at low temperature and high voltage.
It will be great if someone can interpret these answers.
Q56) How can we obtain equal rise and fall times for a CMOS inverter? Similarly how can we obtain the same for NAND, NOR gates?

Equal rise and fall times: As to my understanding goes rise and fall times depends on the relative strengths of the PMOS and NMOS as well as the fan out(number of input gates connected to the output of the gate concerned) of the circuit. A greater fan-out can degrade the rise and fall times. For equal times we need to ensure that the relative strengths of each transistor is same. As such a NMOS is stronger that PMOS(wrt to similar sizing based on the mobility of electrons and holes). Thus a PMOS is sized 2.36:1.

Q57) What are the advantages and disadvantages of static CMOS logic?

Ans57

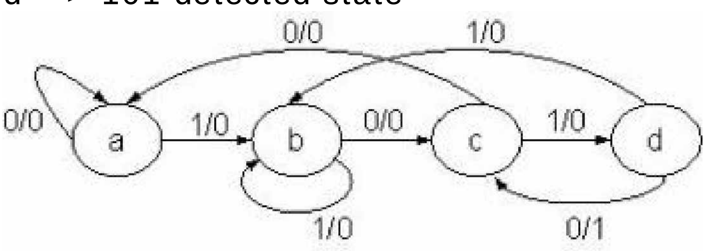
Adv:
> Proper/Full logic levels
> O/P node is connected to either VDD/Gnd: no floating nodes

Dis ADv:
> needs constand voltage supply
> More power dissipation (clk distb n/w)

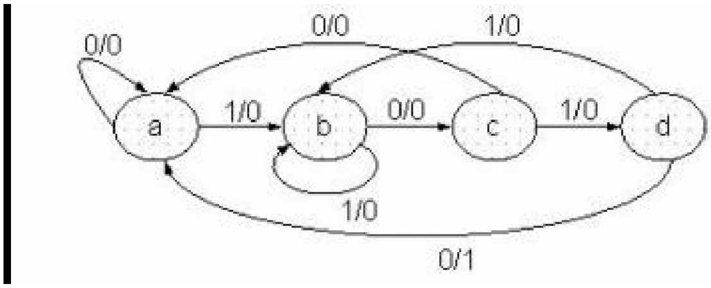
if we compare to Dyanmic, static has the following disadvantages:
1. considerable time delay
2. Large number of transistors
3.no need to refresh

Q57:
Give the State Machine for detecting the sequence "1010" from a serially coming data for both (a) Overlapping & (b) Non-overlapping cases.

Ans Q57:
Part (a) Overlapping case
a --> cont zeros(intial state)
b --> 1 detected state
c --> 10 detected state
d --> 101 detected state



Ans Q57:
Part (b) Non-overlapping case
a --> cont zeros(intial state)
b --> 1 detected state
c --> 10 detected state
d --> 101 detected state



Note the difference between the two FSMs:
This one is going back to a fter detection of 1010 and where are in earlier case it is going to c that 10 detected state

Q58:
Sender sends data at the rate of 80 words / 100 clocks
Receiver can consume at the rate of 8 words / 10 clocks
Calculate the depth of FIFO so that no data is dropped.
Assumptions:
There is no feedback or handshake mechanism.
Occurrence of data in that time period is guaranteed but exact place in those clock cycles is indeterminate

A58) : In the worst case,sender would send 80 words in 80 clock cycles.In this time period,receiver would only be able to receive 8*8=64 words in those 80 clock cycles.Therefore FIFO size should be 16 words.
I may well be mistaken but I will fall back on the old adage that it is only through making mistakes that you will learn something

But what we missed was, " the data may occur anywhere..
the sender may send the data in the last 80 clocks..of first 100 clocks and initial 80 clocks in the next 100 clocks.
So this is the worst case....

So now want to give one more try
Ya Srikanth,now I realised that I erred in considering the 100 clock cycles in isolation....so is it a FIFO depth of 32 words then?

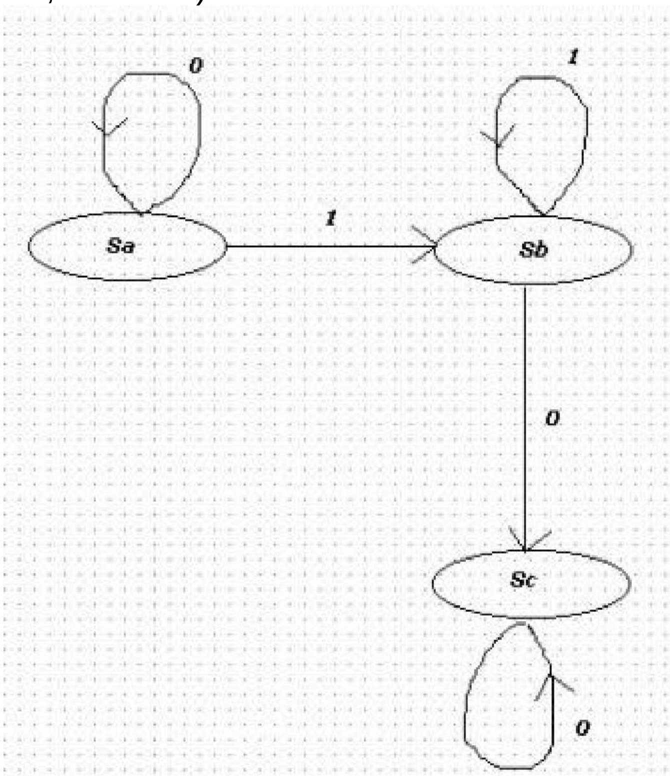
A58)As Srikanth pointed out,worst case is when 80 words come in the last 80 clock cycles of a 100 clock cycle period and 80 words come in the first 80 clock cycles of the next 100 clock cycle period.So now reduce the problem as to how the receiver will handle 160 words in 160 clock cycles.
Receiver can only receive 8 words in 10 clock cycles,so remaining 2 words would have to be stored in a FIFO.At the end of 20 clock

cycles,receiver would have been able to receive 16 of the 20 words it was supposed to i.e 4 to be stored in the FIFO.Thus 2 additional words must be stored in every 10 clock cycle period.So for 16 of these 10 clock cycle periods,16*2=32 words would need to be stored in a FIFO. Hope this was helpful rather than confusing matters further!

Q59:

A simple Qs to understand FSM flow.(asked in TIs apti test)

A state diagram is shown in the following figure: (States are named as Sa, Sb & Sc)



The system is initially in state Sa. If * represents zero or more occurrence of a logic level and + represents one or more occurrence of a logic level which of the following best describes the sequence of states the system could have gone through if it is finally in state Sc.

a) 0* --> 1+ --> 0 +

b) 0* --> 1* --> 1*

c) 0* --> 0* --> 1

d) 0+ --> 1+ --> 0*

A59

(a) 0* --> 1+ --> 0+

Reason: since '*' refers to atleast zero occurence and '+' refers to atleast one occurence of a specific logic level.. IN the worst case.. the following happens.. "1-->0"

so the..first occurence of '1' will take state Sa to Sb and next '0' takes to State Sc from Sb.

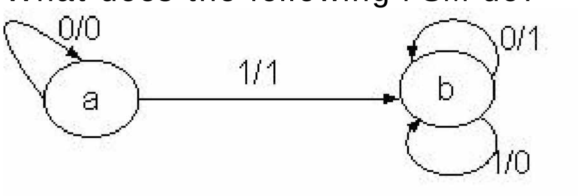
But the worst case in all other options is ..in(b) final state is Sa.. in (c) and (d) final state is Sb.

So the one that best suits the transition of state from Sa to Sc as final state is (a)

Q60:

One more intersting Q on FSM

What does the following FSM do?



Its serial 2′s complimentar

.62. How is a depletion mode NMOS different from a normail NMOS. How is it different in the physical construction ?

Q63:

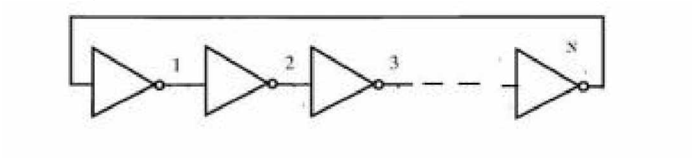
What is ring oscillator? And derive the freq of operation?

A 63. Ring oscillator circuit is a coupled inverter chain with the output being connected to the input as feedback. The number of stages(inverters) is always odd to ensure that there is no single stable state(output value). sometimes one of the stages consists of a logic gate which is used to initialise and control the circuit.

The total time period of operation is the product of 2*number of gates and gate(inverter) delay. And frequency of operation will be inverse of time period.(hope I have looked at the question in the right way)

Application: used as prototype circuits for modeling and designing new semiconductor processes due to simplicity in design and ease of use. Also forms a part of clock recovery circuit.

Ans:Q63



The above diagram shows N number of inverters connected, where N should be odd number.
Let td be the delay of each inverter.

Total delay from in to out is , N * td
So half period = N *td

Freq of oscillation = 1/ 2 *N * td

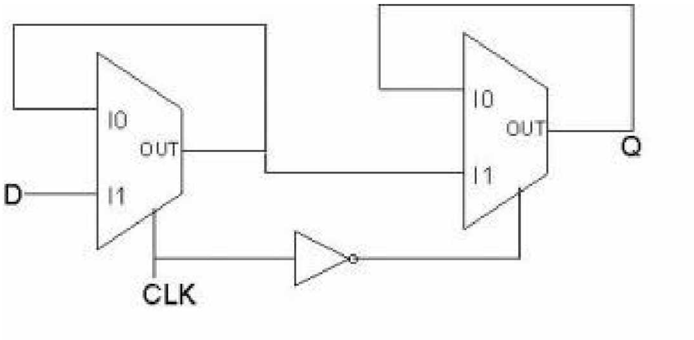
Q65) Is it possible to use Even number of stages instead of Odd in the ring oscillator? If so how can you do it.

Ans65) We can use a differential inverter and in such case we can keep connecting the terminals criss cross(ie + - go to -+) .In this style we then need only even stages
Q: Can anyone try to explain the difference between full-custom design and semi-custom design? Which one would you employ for what and why..?!

Full Custom : Start to finish is your own blood,sweat and tears including designing and laying out lowest level cells (eg AND,OR gates etc) and then placing and routing entire design.
Semi Custom : Using predesigned and layed out standard library cells and then doing place and route yourself.
Full Custom would give you slightly higher performance I guess,provided you do it well.So slightly less work and hopefully slightly higher performance.

Q66)
How will you implement a Master Slave flip flop using a 2 to 1 mux?

Ans: Q66:



Q67:
Using DFFs and minimum no. of 2×1 MUXs, implement the following XYZ flip-flop.

X Y Z Qnext
0 0 0 1
0 0 1 0
0 1 0 0
0 1 1 1
1 0 0 Q
1 0 1 Q`
1 1 0 Q`
1 1 1 Q

ANS 67

Hi Srikanth,

This is what i think could be a way to solve the problem.

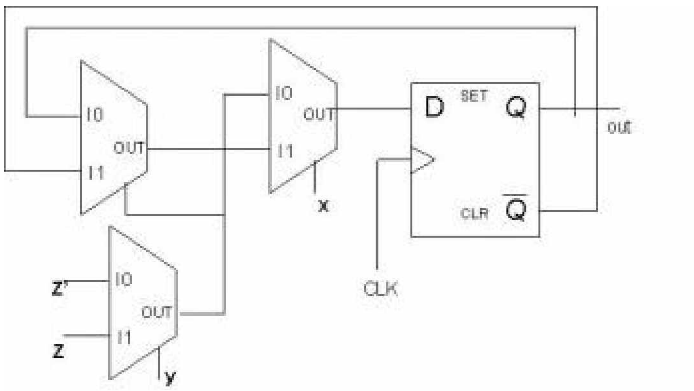
Divide it into 2 parts.
For the first four..
A 2*1 mux will have a enable signal as x.
If x= 0, then Io= 1 wil be used to activate the 4* 1 mux.
4*1 mux wil have 2 select signals y and z. and 4 inputs. being Ino= 1, in1=0,in2=0 in3=1.
The output of the 4* 1 mux is sent to a D flip flop.

The lower four combinations of x,y,z can be implemented,
when the select signal of the 2* 1 mux x=1 then I1= Q is selected.
which wil be sent as the input to the second D-flip flop.
The Qnext of both the flip flops are sent to a 2*1 mux. whose enable signal is x.

the final output Q_next wil be sent as a input to the I1 pin of the 2*1 mux.

it can be implemented using 3 mux and two flip flops.

Ans:Q67:
I could come with a solution of 3 2:! Mux and 1 DFF
If compliments are not availble, we need one more 2:! Mux for finding z'



Q68) What are the advantages and disadvatages of Dynamic logic?
Ans68)
Advantages of dynamic logic:
1. Less power consumption due to less capacitance because only pull up or pulldown network is available.
2.Faster because we have precharge phase, so need time for only evaluate phase.
3.Reduced transistors compared to static logic.
4.There is no short circuit current since pull up path is not turned on in evaluate phase.

Disadvantages:
1.The output node is dynamic node, ie high impedance node, there would be charge leakage and this might affect the logic after some time.
2.charge sharing between output node and internal nodes of pull down network
3.clock feedthrough is one more drawback. This casues the pn junctions of evaluate pmos to turn on when clock goes high.
4.Major drawback is , adds lot of load onto clock because precharge and evaluate transistors are connected to clock.

Q 69) How will you implement a Full subractor from a Full adder?

all the bits of subtraend 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

If we assume full adder as a block where changes can be made only to inputs .. not to the intermediate signals.. Dont you think the xor method which i mentioned should work.... All i have to change is negate the subtraebd and make carry as 1. This should give the output of a full subtractor.
What is wrong according you ??

Q69:
In what cases do you need to double clock a signal before presenting it to a synchronous state machine?
A 69
When the signal is asynchronous (Probably coming from a different clock domain)
Q70)a)
Why did you connect the bulk of NMOS transistor to source?
b)If the bulk of NMOS is connected to some voltage instead of connecting to source what will happen to the NMOS transistor?

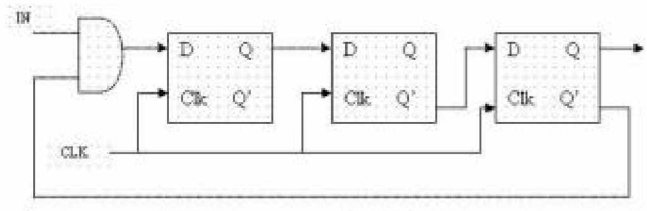
Q71 What are the advantages of BJT over CMOS?

Q71) A very good interview question...
What is difference between setup and hold time. The interviewer was looking for one specific reason , and its really a good answer too..The hint is hold time doesn't depend on clock, why is it so...

Ans71
Setup violations are related to two edges of clock, i mean you can vary the clock frequency to correct setup violation. But for hold time, you are only concerned with one edge and does not basically depend on clock frequency.

Q72:
You have three delay elements D1, D2, D3 that delay a clock by 25%,50% and 75% respectively. Design a frequency doubling (fout = 2 * fin) circuit that uses these delay elements along with any combinational logic.
I think we can double the frequency by just using the delay element D1 and an XOR gate. Just pass the input signal through D1. Now the delayed signal and the original signal are input to a 2-i/p XOR.

Q73:
What are all the test patterns needed at the input, IN, to detect Stuck at zero problem at the input of the first FF in the Following Figure? Assume that initially all DFFs are reset to logic 0.



Can you explain what struck at zero means?

These stuck-at problems will appear in ASIC. Some times, the nodes will permanatly tie to 1 or 0..bcoz of some fault.
To avoid that, we need to provide..Testability in RTL.

If it is permanantly 1 it is called stuck-at-1
If it is permanantly 0 it is called stuck-at-0.

Q74:I was given the layouts of a pmos device with one,two and three fingers.Which one offers better performance and why?(in terms of capacitance at the output node,on resistance and also power dissipation
Ans64)
Isn't is a straight forward answer??? The one with 3 fingers would be a better one overall...
3 fingers offers you the least capacitance at the output, resulting in less power dissipation..
on resistance would be more or less the same for all 3 of the pmos..

from what they had tried to explain to me,one dimension of your source/drain would be fixed (determined by DRC rules for contacts etc)...lets say it is of size x.
now for 2- finger pmos,width of o/p node would be w/2 and hence area would be wx/2.
for 3-finger pmos you would have to tie two drains together...hence area of o/p node would be wx/3+wx/3=2wx/3.this would be more capacitance at the o/p node as compared to 2-finger.
is this correct?

How can I draw a state diagram that asserts a 1 everytime it sees a bit stream that is divisible by 5? For eg: if the bit stream is:
101 output=1 (since 101=5);
1010 output=1 (since 1010=10);
1011 output=0 (since 1011=11) and so on.

SO=remainder 0
S1=remainder 1
ans so and so forth....

First point to note is, according to the Qs, the first bit that is coming is MSB..we need to accumulate the bits that has come till now.

Now to get the multiples of 5, we need to trace the reminders and if the reminder is 0 O/P is 1 else it is zero
So the possible reminders are 0,1,2,3, and 4.

So we need 5 states....Whenever we reach state0, we make the O/P 1 bcoz the reminder is zero.

Initial state will also be 0 only.

Now if one 1 comes reminder will be 1..so goto state1..

In state1, if 0 comes reminder will be 2 and if 1 comes reminder will be 3...

Like this analyse for all the states and go to the next state accordingly..you will reach the solution..now take any input sequence you will get proper Output

Q)How is metastability related to setup time. How does the 2 correlate.
Kindly explain with an example.
Take a DFF. Assume that there is a transition at the input from 0-->1
It will take some considerable time to reach 1..(the duration is called rise time)
The O/P cap will starts charging slowly. But if we disturb the I/P in between,
the cap will stop at some intermediate point..if you tap the O/P from this point u will get some unknown state.. To avoid this...you should allow the I/P to charge to its full level..we shouldn't change it in between..This time is called Setup time...

75]
input A:n bit, B: n bit C:1 bit

if C=1 then out=A-B;
if C=0 then out=A+B;

Q77:
(a) Define: SOP from and POS form?
(b) When is a SOP/POS form is called standard or canonical?

1) Sum-of-Products (SOP) - Example : AB+AC+BC
2) Product-of-Sums (POS) - Example : (A+B)(A+C)(B+C)

If each term in SOP or POS contains all the literals, then it is called a Standard or Canonical form SOP/POS.

Q78:
If F(A,B,C,D,E) = BE, how many terms will be there in the standard or canonical SOP representation of F?

BE=(A+!A)(C+!C)(D+!D)BE

there will be 8 terms

Q79:
In C-N (Change-No change) flip flop, there won't be any change in output as far as N is 0, irrespective of C. If N=1, then if C = 0 output will change to zero else if C =1 output will be the compliment of previous output.
(a) Write the characteristic table ?
(b) Design this using J-K flip-flop?

Ans79:
a) Characteristic Table:

C N Q(t+1)
0 0 Q(t)
0 1 0
1 0 Q(t)
1 1 Q`(t)

b) Design:

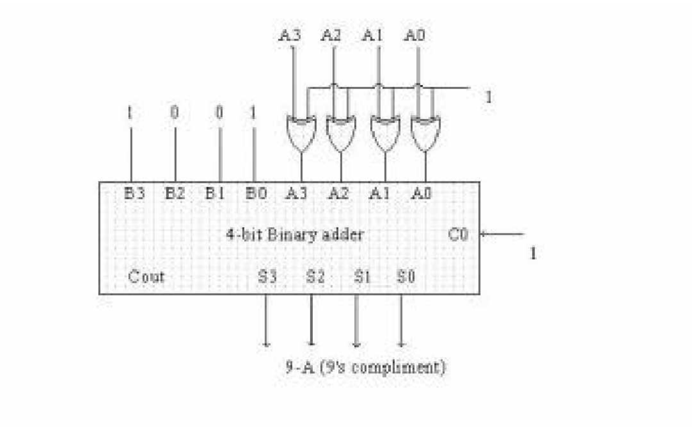
J = C.N ; K = N; Q = O/P
2)why do we require setup and hold times.
setup time models the time required for the data to propagate within the component and load into the latch before the latch is triggered by the clock.

Q80)How will u interface a 16-bit microprocessor to a 2 1-K RAM.Describe the read and write operations with the necessary signals?Guys this is a basic question asked in one of my interview.

Q81)

Design a circuit for finding the 9's compliment of a BCD number using 4-bit binary adder and some external logic gates.

Ans to Q81)
9's compliment is nothing but subracting the given no from 9.So using a 4 bit binary adder we can just subract the given binary no from 1001(i.e. 9).Here we can use the 2's compliment method addition.Is my answer correct.
Ans:81:



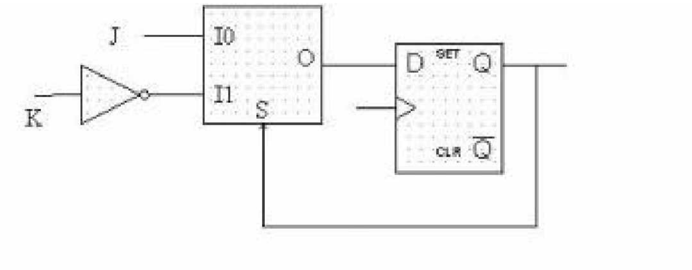
Q82:

Design a J-K flip flop using DFF,2:1 Mux and an-inverter?

Q82: Connect J i/p to MUX S1 pin, Kbar i/p(Inverter o/p of K) to MUX S0 pin & Qbar (of the DFF) to the MUX select pin --- Give the MUX o/p to DFF i/p.

DFF o/p will be the desired o/p.

Ans:82:



Q83:

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

Ans83)
2(power n)-2n 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

q86) What is an LFSR .List a few of its industry applications.(Interview q for an internship

Ans 86) LFSR is a linear feedback shift register where the input bit is driven by a linear function of the overall shift register value.

coming to industrial applications, as far as i know, it is used for encryption and decryption and in BIST(built-in-self-test) based applications..

Q87

Design a divide by 3 clock. It was asked in one of my friends interview. I guess this has complex logic to it. In a phone interview how do you explain all that stuff, or is there a simple design for it..

Can u elaborate more on BIST?

there r 3 strategies for testing a circuit.
ad hoc techniques, scan based technique, and self-test techniques.
BIST(built in self test) is the self-test techniques
in this technique parts of the circuit are used to test the circuit itself.
BIST requires two circuit modules called
1. PRPG(pseudo random pattern generator)
2. ORA(output response analyzer)
these two modules can be built using LFSR.

Q.87
How do we design a 1Mb SRam? Give the architecture of the same?

guyz i couldnt solve this Q.

8 Consider an alternate binary number representation scheme, wherein the number of ones M, in a word of N bits, is always the same. This scheme is called the M-out-of-N coding scheme. If M=N/2, and N=8, what is the efficiency of this coding scheme as against the regular binary number representation scheme? (As a hint, consider that the number of unique words represent able in the latter representation with N bits is 2^N. Hence the efficiency is 100%)
Thanks
Ans: Q88)

If N=8, M=4

That is we have only combinations which has excalty 4 number of 1's

The possible binary combination in this number system = The 8 bit binary numbers which has exactly 4 number of 1s = 8C4 = 70

So eff = (70/256) * 100 = 27.34%

Q89)

2 bit gray code sequence (00,01,11,10) is coming serially. Design the circuit to detect the false sequence.

What is the difference between MAC n Normal Mulltiplier

Normal multipler just multiplies the two inputs,
OUT = IN1 * IN2

Where as MAC implements the following function:
OUT = PREV_OUT + (IN1 * IN2)

This is major block in many DSP processors as it helps in realization of filters.
For example for FIR filter, y[n] = c0 * x0 + c1 * x1 + c2 * x2 and so on

First MAC multiplies c0 and x0 and stores the results. then multiplies c1 * x1 and adds that to prev_out and stores this value and so on

Q) How to increase the noise margin of a given CMOS circuit.

Q90.
If u want to convert X xor Y to X or Y ,what should Xand Y be?

1)Antonyms
2)SYNONYMS
3)twos compliment of each other
4)ones compliment

Q91)how to obtain a 3 input OR gate using 2 input NAND gates?how many nand gates are used?

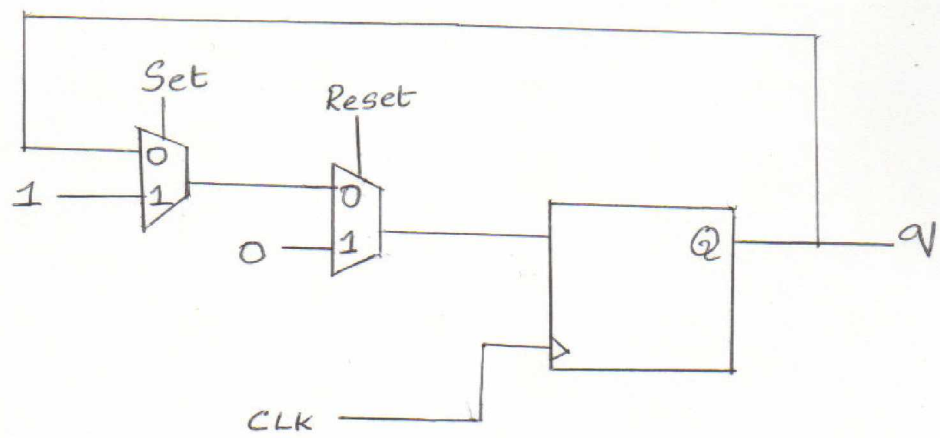
We can get 3input OR gate ffrom 6 2input NAND gates

Q91
This is the one of the interiview question sireesha was asked for asic design job
there is a block with 3 i/ps. and 1 o/p
block is +ve edge triggered.
the conditions are
if reset=1,q=0,
else if set=1,q=1,
else q=q.

Ans91

First do the truth table for the conditions given
For R=1, and all other 4 combinations of S,Q(t) , Q(t+1) = 0

Now we need to fill the other 4 combo's left when R=0(y only 8 combinations , it is since we have only 3 variables giving 2^3=8 combos)
In this when S=1, irrespective of Q(t) , Q(t+1) = 1,
in other S=0, when Q(t) = 1, we have Q(t+1) =1...
draw the kmap and get the equation for Q(t=1)
This is what I get
Q(t+1) = Rbar[S+ Sbar*Q]
The simplified expression,
Q(t+1) = R' (S+Q)



Ans Q91:

Q92

Does combinational circuits have setup and hold times???Why
think As in sequential circuits setup and hold times are with respect to clock,in combinational circuits they are with respect to output ?If they are w.r.t output are they equal to propagation delay, as input should be maintained constant til propogation delay,right?
i m also not much sure about this qiestion.
how much maximum frequency clock i can apply to input of inverter having propagation delay Tpd ns?

1/2Tpd

Q: Can a circuit have both setup and hold violations? Is it possible to have Setup and hold violations together on the same path?

guess a circuit cannot have both set up and hold time violations together...

bcoz if there is setup time violation then for that data then the data is not at all sampled by the flip flop,so there wouldnt be any hold time for that data which is not sampled not any condition to violate.

and if there is hold time violation that means that data was successfully sampled...so there was no setup time violation.

So,answer for ure first part is not possible to ahve both set up and hold violations for a ckt at same time and for same data.

95)

Given a 8 bit number how would you check whether it is a palindrome or not???...

Ans 95.
DO the XNOR of bits 1,8 ; 2,7; 3;6; 4,5
Then do an AND of the outputs of the 4 XNOR gates...
The output is 1 for the palindromes

q 96)A 4 bit shift register has _______ number of states.

ans 96) i guess it is 16 states
The number of states of a shift register are dependant on which configuration it is connected.
Suppose last fliop flfops output is connected to the input of flip lfop(Ring counter), the possible states are only 4

q 97)Given/using a Positive Trigger as input generate Square wave.

q99)Swap two 8-bit registers without using another register.
Ans99)
Connect the registers as shift registers. Also connect the output of last register in set2 to the input of register in set1 and after 8 clock cycles you will have the register values swapped..Is there any other way venkat?

Many other sol are avil
q100)In what cases do you need to double clock a signal before presneting it to a Synchronous machine?

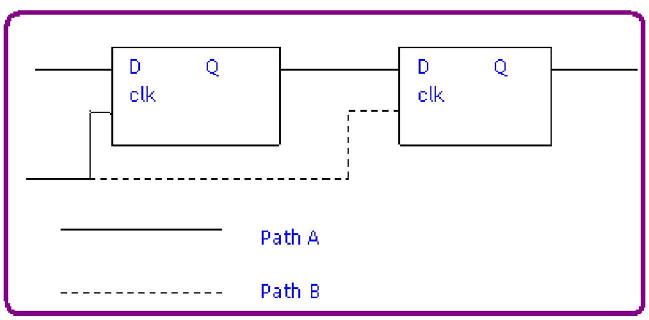
A100:
If the signal is moving from one clock-domain to another,
that if the signal is asynchornous..
we need to double clock the signal.
The extra flop that is used for this purpose is called synchronizer.

The synchornizer avoids metastability.

Q101)You have a driver that drives a long signal and connects to an i/p device. At the i/p there is either overshoot or undershoot or signal threshold violations. What can be done to correct this problem?
Ans101)

Do u mean long interconnect between driver and input device[ie load device]..
If so, in such cases we try to use buffers at equal intervals to boost the signanal and reduce delay, this would as well take care of overshoots or undershoots too...

Referring to the diagram below, briefly explain what will happen if the propagation delay of the clock signal in path B is much too high compare to path A. How do we solve this problem if the propagation delay of path B can not be reduced ?

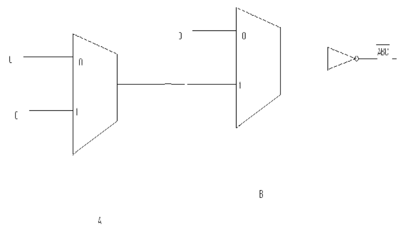


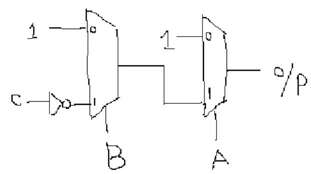
This is fast path condition and to rectify this you need to put some redundant conbinational logic(which does not alter the Q output value, say shorted ip AND) between the Q of first ff to D of 2nd ff just to delay the internal signal a bit.

Q 102 Why do we call as pmos passes good 1 & poor 0 and nmos passes good 0 and poor 1? please explain in detail.

103)what are advantages of latch over flip flop?

Q104) how would u design 3 i/p NAND gate using 2no. of 2:1 muxes and 1 inverter? i know the answer, but let people try first then i will post the answer.





Q105:
List the differences between SRAM and DRAM?
ANS 105. Though simple, remained unanswered for a long time. This Q was asked to me in Cypress Semiconductors telephonic interview. Answer as follows:

1. SRAM faster than DRAM
2. SRAM consumes less power as compared to DRAM (reason: DRAM needs periodic refreshing of capacitor used to store a bit)
3. SRAM requires 6 transistors where as DRAM is built using 1 transistor and a capacitor
4. SRAM have low density as compared to DRAM, in fact DRAM is typically 1/4 the silicon area of SRAM or less.
5. SRAM expensive than DRAM, in fact DRAM much cheaper than SRAM
6. SRAM compatibale with CMOS technology where as DRAM not typically compatible with CMOS technology.
7. SRAM gives differential outputs (BL and !BL) where as DRAM gives single ended outputs, BL only (where BL = Bit Line)
8. SRAM is used where speed is imp for ex. Cache memory where as DRAM is mainly used as Main memory
9. SRAM uses sense amplifiers for performance where as DRAM need sense amplifiers for correct operation

Q106:
Match the following:
(a) PROM (i) Programmable AND Array and programmable OR array
(b) PAL (ii) Fixed AND array and programmable OR array
(c) PLA (iii) Programmable AND Array and fixed OR array

A 106)

(a)PROM -> Fixed AND array and programmable OR array
(b)PAL -> Programmable AND Array and fixed OR
(c)PLA -> Programmable AND Array and programmable OR array
Q 107) How do you count the no. of 1s present in an 8 bit register without using counter or adding bit by bit?

Q108)

A D FF has its D i/p from a MUX. MUX input0 is connected to external i/p and MUXi input1 is connected to output of D FF (Q) through combo block(i.e: feedback of o/p to i/pthru combo block). If Mux delay is 0 ns and
Tsetup = 3ns, Thold = 2ns , TClock-to-Q = 1ns
What is the max frequency of the circuit with and without feedbak?

I am getting a hold violation with feedback and without feedback the clock frequency is 1/1ns....correct me if I am wrong and plz tell me ure answers

Ans:108:
Before going to the actual solution, I just want to mention one point about the given data.
In the given data The hold time(2ns) is greater than the Clock-to-Q delay(1ns) of the flop.
That means the data is available at the output at 1ns after the clock edge but the input should not change till 2ns. It doesn't look logical for me.
For most of the flip flops, Thold is always less than TCq. This condition is essential for shift registers where as for circuits it may not be compulsory.
This Qs has two parts: (a) no feedback (b) feedback.

(a) If there is no feedback, that is assuming both external inputs and they meet the setup time of the flop, THe maximum clcok freq = 1/1ns

(b) If there is a feedback, to avoid hold violation, the "dly" has to be atleast 1ns. With violations there is no meaning for maximum freq. Becasue the circuit won't be functional at all.
So we have to take combo + Mux dly together >= 1ns. As MUx delay

= 0, combo delay has to be 1ns.
Thold <= Tcq + dly, that implies, dly >= 1ns

T >= Tcq + combo_dly + Tsu = 1 + 1 + 3 = 5ns

F <= 1/5ns

Q 108.

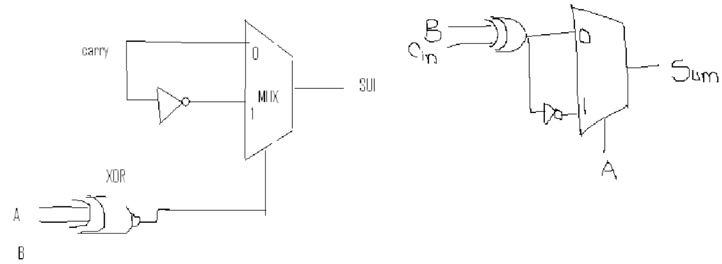
What is the source of set up and hlod time time violations ? what exactly happens inside the Flip-flops ? and What is inside the Flip-flope which makes it edge-triggering ?

Q 109. Explain Electromigration.

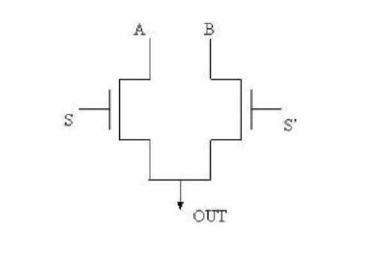
ANS 109. Electromigration refers to the gradual displacement of the metal atoms of a conductor as a result of the current flowing through that conductor. The process of electromigration is analogous to the movement of small pebbles in a stream from one point to another as a result of the water gushing through the pebbles.

Because of the mass transport of metal atoms from one point to another during electromigration, this mechanism leads to the formation of voids at some points in the metal line and hillocks or extrusions at other points. It can therefore result in either: 1) an open circuit if the void(s) formed in the metal line become big enough to sever it; or 2) a short circuit if the extrusions become long enough to serve as a bridge between the affected metal and another one adjacent to it.

Electromigration is actually not a function of current, but a function of current density. It is also accelerated by elevated temperature. Thus, electromigration is easily observed in Al metal lines that are subjected to high current densities at high temperature over time.
Q110) impliment the full adder ckt. using 2:1 mux
and 2 i/p x-or gate?



Q111:
(a) What is the functionality of the following circuit?
(b) Show the boolean equation?
(c) What is name of logic that is used in implementing the circuit?
(d) Mention the advantages and diadvantages of this method. Also suggest improvements, if any, to overcome the disadvantages?



Ans111)
a.)OUT=AS+BS'
b.)PASS TRANSISTOR LOGIC
c.)
advantages:faster since nmos is used
disadvantages:Logic Degradation of 1.

Q112
What happens when we increase the number of contacts or via from one metal layer to the next one?
QWhat happens if we use an inverter instead of a differential sense amplifier?

Ans112)
The parasitic capacitance increases , but the contact resistance reduces , overall it speeds up the circuit..also in designs its better to have multiple contacts because 1 in 1000 contacts do not make proper

contacts, in that scenario have multiple contacts ensures that u do not have unnecessary opens..

Q 113

How to decrease the propagation delay in a gate ?

Ans113)

. Transistor Sizing:- The most obvious solution is to increase the overall transistor size. This lowers the resistance of devices in series and lowers the time constant. However, increasing the transistor size, results in larger parasitic capacitors, which do not only affect the propagation delay of the gate in question, but also present a larger load to the preceding gate. This technique should, therefore, be used with caution.

2. Progressive Transistor Sizing:- An alternate approach to uniform sizing (in which each transistor is scaled up uniformly), is to use progressive transistor sizing.

3. Input Re-Ordering:- Regarding Critical Signal and Critical Path (refer ANS. 120 for more details)

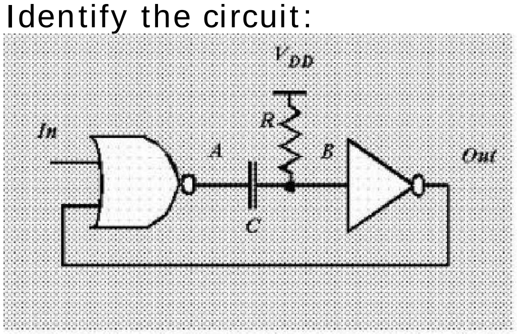
4. Logic Restructuring:- Manipulating the logic equations can reduce the fan-in requirements and hence reduce the gate delay as illustrated in Figure. Partitioning the NOR-gate into two three input gates results in a significant speed-up.

Q114

How to convert Ex-or gate to AND gate.

Q115:

Identify the circuit:



Q116:
(a) Will hold time effect the max freq of operation?
(b) If hold time is negative, will it effect the maximum frequency of operation?

Ans to Q116)
(a) we dont have control over the hold time,But if we chose a device with higher hold time .the frequency of operation will reduce
since for a single flip flop

freq=1/T pd
Tpd=Tsetup+Thold - CLKskew

In our case Thold increases ,Tpd also increases so frequency decreases.

b)
note : a hold time has to follow the condition ie
Thold< T clk-q+Tcd
If it does not follow it goes to metastability.

so if the hold time decreases the frequency increases.

May be I put my sentence in some other way, if there is a setup time violation, we can change the clock freq and adjust this.
And is it possible to adjust the hold violation with clock freq?
And yes I do agree hold time also effects the max clock freq indirectly..
Becoz to fix hold time violation, we need to calculate the delay and then use that value, for cal of freq. But again that purely depends on the configuration that you are using.

Coming to negative hold time, if you just consider a flop it may not be negative.
But the negatiev hold time comes into picture, bcoz of various reasons in asic desing.
I can give one example.
Say suppose consider a memory cell. Where after clock the input takes some considerable delay to reach the flop, so in that case, even if you change input before the clock edge also..no issues..
because this clock is only for our reference.
Did you get my point..

But most of the cases, abs(Th) < Tsu..
So still max freq will be effected by Tsu only

Q117) can anybody please explain what is meant by
CLOCK TREE SYNTHESIS ? why it is used?
what r its advantages?

Q118:

Design a sequential circuit that produces a logic 1 at the output when the input has been 1 for eight or more consecutive clock pulses using a counter(shown below) and minimum number of basic gates.

Q)119

Construct a half subtractor using a full adder.
Hint:Make use of gates & block diagram of full adder.
It was asked Sasken Interview

Connect Cin of full Adder to Ground(Permanent Zero) as it is not needed
Connect A to one Input of the Full Adder
Connect B to one input of the Full Adder
Consider the Sum output of the Full Adder as the Difference O/p.
To generate Borrow, Use a 2 i/p AND gate, whose i/ps are B and Difference.

can any tel me how to find depth of fifo?
I don't think there is some fixed formula or something to calculate the depth of FIFO.
But FIFO's will be needed when the Master and Slave are working at different clock freqs
The two points that we need to keep in mind are:
1. We shouldn't miss any data.(No lose of data)
2. It should be minimum cost or optimized

For Eg: Assume sender sends the data for 10us cont at the rate of 200 M samples/sec. But the receiver can receive it @10M samples/sec.
In this case, depth of FIFO = (200M - 10M) * 10us = 1900 samples
But if data duration is not finite in this case, depth of FIFO will be infinite.

Q120:
Let A and 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 NMOS inputs A and B which one would you place near to the output?
Ans120)
The late coming signals are to be placed closer to the output node ie A should go to the nmos that is closer to the output.

Q121. In cadence, how are rise, fall and delay time calculated?
A121 I the extracted netlist we ues . measure statment to find rise, fall and delay.

example
.measure tran tr trig v(O) val=0.1 rise=2 targ v(O) val=0.9 rise=2
.measure tran tf trig v(O) val=0.9 fall=2 targ v(O) val=0.1 fall=2
.measure tran prdelay trig v(A1) val=0.5 rise=2 targ v(O) val=0.5 fall=2

when simulated u hspice will generate .mt0 file. which will have the rise fall and delay time.
Q122. When we shrink from 130nm technology to 90nm, how does it effect the resistance?
we know R= row *(L/A). which shows that R is proportional to L and inversly proportional to (w)square. if L and w both r reduced, w will have more effect and R will increase.
Q123. A) In the 2-input NAND, between the gates of the 2 NMOS, to which would you give a critical signal and why?
B) In the same condition, if given a chance to increase the size of the NMOS, which one should you increase and why?
A 123 in the NAND gate, between the gates of the 2 nmos, to which would u give a critical signal and why?

critical signal is one which probably has more delay. hench should be given to the nmos close to the output, so that the effctive cap across it is also less.

10. in the same condition , if given a chance to increase the size of the

nmos, which one should u increase and why?

the size of the nmos conncted to the ground should be increased, so that the cap across it decreases.

Q124. Given three logic inputs, A, B and C. Design a logic circuit which gives as an output the three signals NOT(A), NOT(B) and NOT(C). Your circuit may contain any number of AND and OR gates, but no more than 2 NOT gates.
--> You can't use feedback at all

--> You can use AND and OR gates with more than 2 inputs, but nothing changes, since for every circuit with gates having more than 2 inputs, another circuit can be designed using only 2-inputs gates which implements the same logic function.

Q 125] How many Radix 2 Booth's multiplier do you need to multiply 24 bits by 20 bits?

Q126. Explain Fowler-Nordheim tunneling. (This Q was asked in Cypress Semiconductors interview)

ANS 126. When the gate oxide is very thin, a current can flow from gate to source or drain by electron tunneling through the gate oxide. This current is proportional to the area of the gate of the transistor. This effect limits the thickness of the gate oxide as processes are scaled. However, it is of great use in electrically alterable Programmable Logic Devices (PLDs).

Q127. Draw a 6-T SRAM Cell and explain the Read and Write operations.

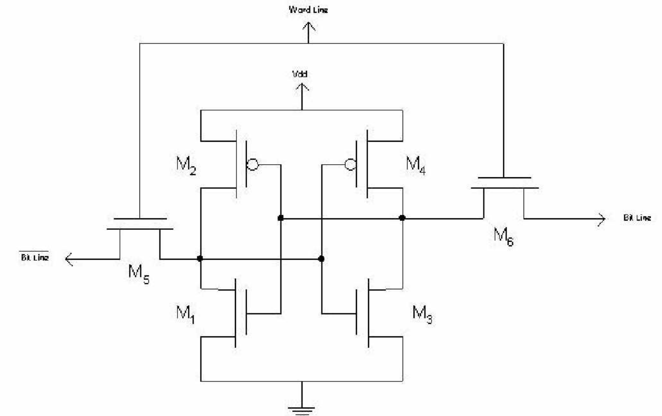
ANS 127. A single SRAM memory cell is presented in a block diagram below. As can be noted, six total transistors are required for our design. Two NMOS and two PMOS transistors are used to construct a simple latch to store the data, plus two more pass NMOS transistors are controlled by Word Line to pass Bit Line and Bit Line into the cell.

Write and Read operations are performed by executing a sequence of actions that are controlled by the outside circuit.

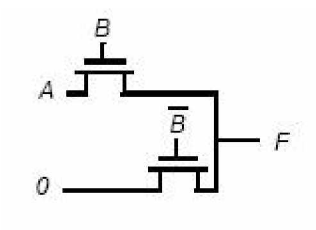
<u>Write Operation:</u> A Write operation is performed by first charging the Bit Line and Bit Line with values that are desired to be stored in the

memory cell. Setting the Word Line high performs the actual write operation, and the new data is latched into the circuit.

<u>Read Operation:</u> A Read operation is initiated by pre-charging both Bit Line and Bit Line to logic 1. Word Line is set high to close NMOS pass transistors to put the contents stored in the cell on the Bit Line and Bit Line.



Q128. Refer the diagram shown below. F = ? (write the equation)



= AB

Q129. This is a very interesting Q. Consider a n-MOS. Now tell what is the <u>difference between</u> a PN Junction that exists in a BJT or diode (or between Source or Drain and Substrate) <u>and</u> the inversion layer - substrate junction? (inversion layer is nothing but n-channel below gate)
[Hint: Think in terms of how n-type conductivity is bought.]

ANS 129. The <u>difference between</u> a PN Junction that exists in a BJT or diode (or between Source or Drain and Substrate of NMOS) <u>and</u> the inversion layer - substrate junction in NMOS is that in PN junction the n-type conductivity is brought about by a metallurgical process; i.e. the electrons are introduced into the semiconductors by the introduction of donor ions. Where as in an inversion layer substrate junction, the n-type layer is induced by the electric field E applied to the gate of NMOS. Thus, this junction instead of being metallurgical is a *field induced* junction.

)if rise time of the circiut is 1/100ns then what is thge bandwidth

cpu which is running faster than memory what logic shuold be in between cpu and memory in order to synchronize these two?

ANS 132. Front Side Bus (FSB) is the most imp bus when we are talking about the synchronization between CPU and Main memory and performance of the computer. The FSB connects the CPU to Main memory. The faster the FSB the faster you get the data to CPU and the faster the CPU processes the data the faster is computer. The speed of the FSB depends on CPU, Motherboard chip-set and the System Clock.

1) draw a ckt to multiply a digit with two?
2)draw a ckt to divide a digit with two?
3)to get 25mhz from 800mhz how many d f/fs are required?
4)what r the ways to code FSM which technique is faster?

ans4)
there r mainly 4 ways 2 write fsm code
1) using 1 process where all input decoder, persent state, and output decoder r combine in one process.
2) using 2 process where all comb ckt and sequential ckt seperated in different process
3) using 2 process where input decoder and persent state r combine and output decoder seperated in other process
4) using 3 process where all three, input decoder, persent state and output decoder r seperated in 3 process.
the fsm style using 2 process where all comb ckt and sequential ckt seperated in different process in faster than all and mostly use for better performance

Q131:
What is volatile memory? Give an example.

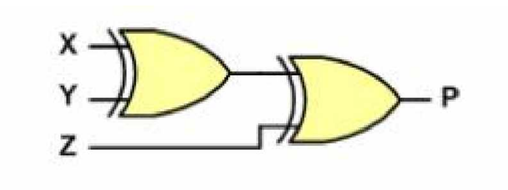
ANS 131. Volatile memory is one whose contents or data is destroyed or vanished when it's power is switched off.
For ex: RAM, be it SRAM or DRAM.

Where as Non-Volatile memory is one whose data or content/s are retained even after the power is switched-off.
For ex: ROM

Q133)
1)a sequence is given and asked to calculate odd parity for that?
2)what is the starting & ending Address of48kb and 1Mb?how to calc

ANS 133. Parity generation, insertion and detection are three different things. What i understand from your Q i.e. calculating odd-parity is to detect whether there are odd or even number of 1's in the sequence.

1) For calculating odd-parity, if there are n-bits you need n-1 XOR gates. Connect the i/p's as shown in the figure and you get odd-parity at the o/p P i.e. P will be 1 if there are odd no. of 1's in the sequence else will be 0.



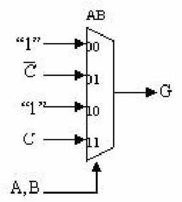
For even-parity simply invert o/p P in the shown figure and you get even-parity i.e. P will be 1 if there are even no. of 1's in the sequence else will be 0..

2) a) 48 KB = 48*1024 Bytes = 49152 Bytes, but hang on. We address from 0, therefore we need to minus 1 from 49152 to get the final byte of 48th KB, which comes to 49151. Therefore 49151 is the last byte of 48th KB. Similarly 49151-1023 = 48128 is the 1st byte of 48th KB. Convert this 1st and last byte of 48th KB into HEX and you get starting and ending address of 48th KB as follows:
Starting address --> BC00
Ending address --> BFFF

b) Similarly for 1MB
Starting address --> FFC00
Ending address --> FFFFF

Please correct if I'm worng.

Q134. Refer diagram below. Write minimal SOP expression for G.

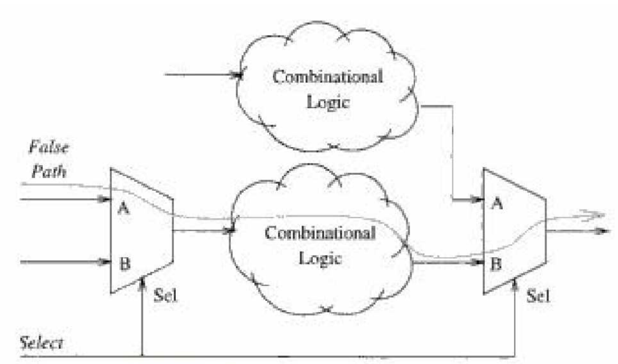


A134:
G = B' + AC + A'C′

Q135
what is false path? how it determine in ckt? what the effect of false path in ckt
ANS 135. By timing all the paths in the circuit the timing analyzer can determine all the critical paths in the circuit. However, the circuit may have false paths, which are the paths in the circuit which are never exercised during normal circuit operation for any set of inputs.

An example of a false path is shown in figure below. The path going from the input A of the first MUX through the combinational logic out through the B input of the second MUS is a false path. This path can never be activated since if the A input of the first MUX is activated, then Sel line will also select the A input of the second MUX.

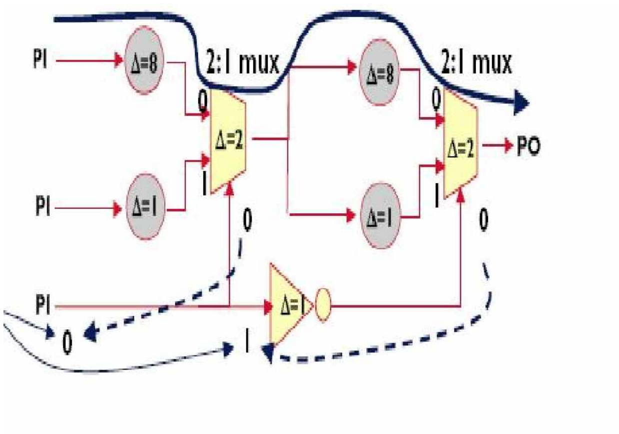
STA (Static Timing Analysis) tools are able to identify simple false paths; however they are not able to identify all the false paths and sometimes report false paths as critical paths.
Removal of false paths makes circuit testable and its timing performance predictable (sometimes faster)



An example of a false path (i.e., a path which can never be activated).

Q136
what are topological and logical timing analysis?

how we analyze timing behavior of give ckt in fig. with respect to logical and topological timing analysis.



A136:

Topological analysis: We don't bother about the funcitonality of the gates..We just blindly add all the delays and identify the longest delay.

Logical analysis: We actually see the functionality of the circuit...then make the sensitized paths and then identify the longest delay.

For example, in the given circuit, from PI to PO, the topological analysis gives the longest delay as 20: 8+2+8+2

But if we use logical analysis, the two 2:1 muxes select lines are connected through an inverter. So both can't be 0 at the same time. So if S of first Mux is 0, then second mux's select line must be 1. That makes the longet delay as: 13

Q137. Consider two similar processors, one with a clock skew of 100ps and other with a clock skew of 50ps. Which one is likely to have more power? Why?

ANS 137. Clock skew of 50ps is more likely to have clock power. This is because it is likely that low-skew processor has better designed clock tree with more powerful and number of buffers and overheads to make skew better.

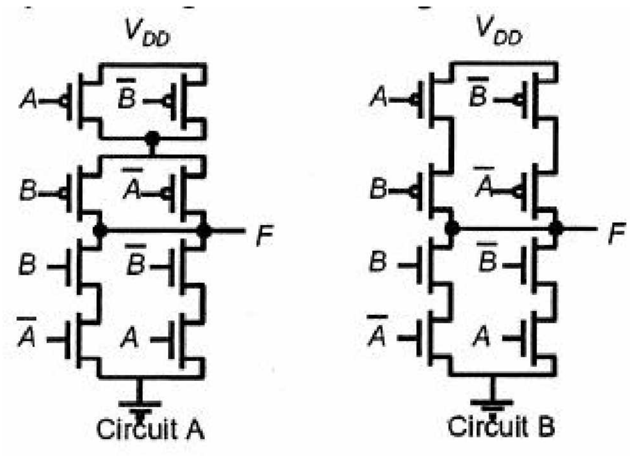
Q138. Refer the diagram below:

a) What is the logic function of the circuits A and B in the figure?

b) Which one is a dual network and which one is not?

c) Is the non-dual network is still a valid static logic gate?

d) List the primary advantage of one configuration over the other



A138)
(a) Both are implementing XNOR gate
F = (AB' + A'B)'

(b) Circuit A is dual.. Circuit B is modified version of circuit A..can be considered as dual.

(c) Yes Circuit B is also valid static logic

(d) Circuit B takes the adavatage of A & A' can not be 1 or 0 at the same time. SImilarly B also. This makes the layout simpler.

Q139. Explain:
a) Static Timing Analysis (STA)
b) Dynamic Timing Analysis (DTA)
139:

(a) STA: No input patterns are needed as we don't look for the funcitonality. We will just check for the timings. This gives worst case timing. Useful in synch desing. Doesn't worry about multicycle and false paths
(b) DTA: Checks functionality and timing at the same time. Dynamically input vectors needs to be varied in order to check for the timing. Useful in asynch designs. Worries about multicycle paths and false paths much slower

____
Q140. What are multi-cycle paths?
_________________ANS 140. Multi-cycle paths are paths between

registers that intentionally take more than one clock cycle to become stable.

For ex. Analysing the design shown in fig below shows that the output SIN/COS requires 4 clock-cycles after the input ANGLE is latched in. This means that the combinatorial block (the Unrolled Cordic) can take up to 4 clock periods (25MHz) to propagate its result.

Place and Route tools are capable of fixing multi-cycle paths problem.

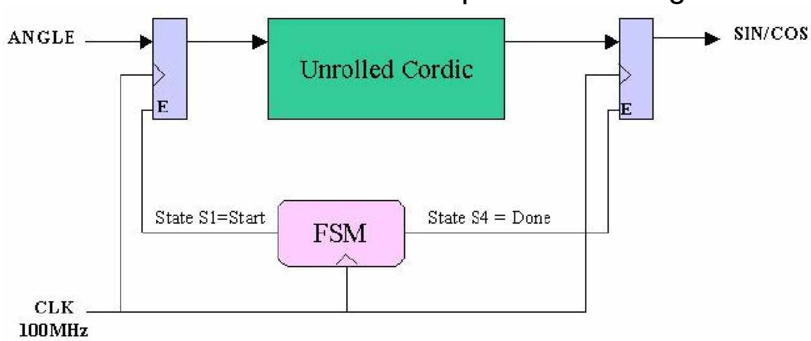
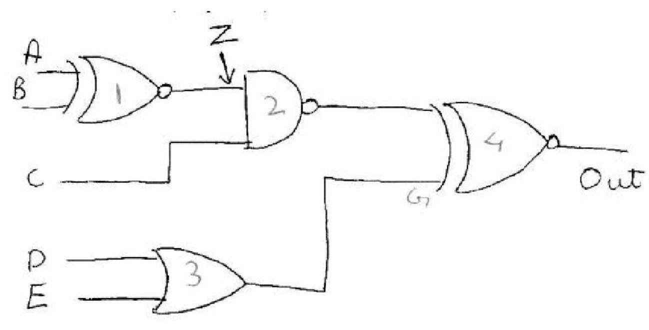


Figure 1 Example Design

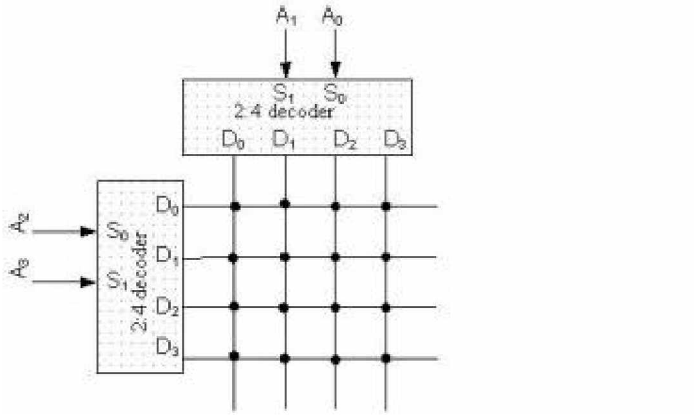
Q141. Generate the complete set of test vectors to check for Z stuck at 1 in fig. below.



A141:
Using single fault model and path sensitized test,
A=1,B=0,C=1 or A=0,B=1,C=1 are needed. (In both the cases D and E can be anything)
SO the test set = { 101xx, 011xx }

Q142;
How many number of 16X8 size memories are needed to obtain a memory of size 256X16?
ANS 142. 32 number of 16X8 size memories are needed to obtain a memory of size 256X16.

Q143:
Using the binary cell as block box, two-dimensional decoding structure of a memory block is shown below:



(a) What is the size of the memory?
(b) What is address of the basic cell at location: Row3, Column3?
(c) Which cell will get selected for the address: C (in hex) ?

ANS 143.

a) 16 bits

b) A (in HEX)

c) Row 1, Column 4

Q144. What is difference between Contamination Delay tcd and Propagation Delay tpd?

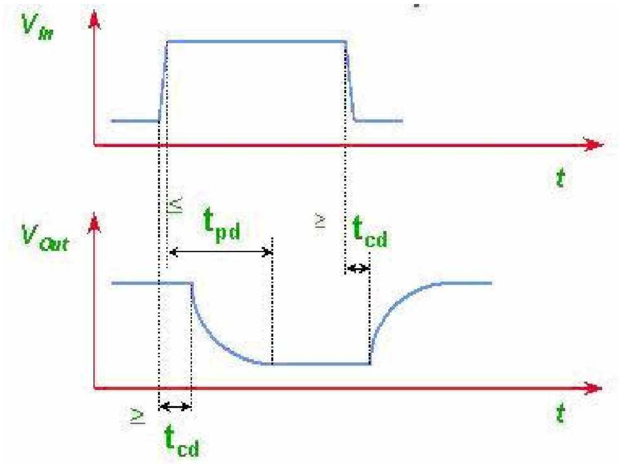
____
I think, therefore I am

ANS 144.

Contamination Delay tcd – Time from first input change until first output change. (Input contaminated to output contaminated) OR Best case delay from invalid inputs to invalid outputs.

Propagation Delay tpd – Time from last input change until last output change. (Input at steady state to output at steady state.) OR Worst case delay from new valid inputs to new valid outputs.

Refer fig. attached for better understanding.



Q145
In what cases do you need to double clock a signal before presenting it to a synchronous state machine?

ANS 145. If the signal is moving from one clock-domain to another, i.e. if the signal is Asynchronous.

Q146. What is clock-feed-through (CFT)? Explain in detail.
Q147. You have two counters counting upto 16, built from negedge D-FF . First circuit is synchronous and second is "ripple" (cascading). Which circuit has a less propagation delay? Why?
A147:
The synchronous counter will have lesser delay as the input to each flop is readily avaible before the clock edge.
Whereas the cascade counter will take long time as the output of one flop is used as clock to the other. So the delay will be propagating.

For Eg: 16 state counter = 4 bit counter = 4 Flip flops
Let 10ns be the delay of each flop

The worst case delay of ripple counter = 10 * 4 = 40ns
The delay of synchronous counter = 10ns only.(Delay of 1 flop)
Q148. You have a circuit operating at 20 MHz and 5 Volt supply. What would you do to reduce the power consumption in the circuit- reduce the operating frequency of 20Mhz or reduce the power supply of 5 Volts and why?

A148:

The power dissipation is f*C*V^2, that is proportional to VDD^2
So reducing VDD is more effective.

Q149. What is the minimum clock freq for a CMOS to work? What are the parameters for determinig this min. freq? [DRDO interview Q]

Q150. When will you use a latch and a flipflop in a sequential design?

Q1: Given the following Verilog code,
what value of "a" is "displayed"?

```
always @(clk) begin
  a  = 0;
  a <= 1;
  $display(a);
end
```

Q2: Given the following snipet of Verilog code,
draw out the waveforms for "clk" and "a".

```
always @(clk) begin
  a = 0;
  #5 a = 1;
end
```

Q3: What is the difference between the following two lines of Verilog code?

```
#5 a = b;
a = #5 b;
```

Q4: Write the Verilog code to provide a divide-by-3 clock from the standard clock.

Q5: What is the difference between:

```
c = foo ? a : b;
```

and

```
if (foo) c = a;
else c = b;
```

Q6: Using the given, draw the waveforms for the following
versions of a (each version is separate, i.e. not in the same run):

```
reg clk;
reg a;
always #10 clk = ~clk;

(1) always @(clk) a = # 5 clk;
(2) always @(clk) a = #10 clk;
(3) always @(clk) a = #15 clk;
```

Now, change to a wire, and draw for:

(4) assign #5 a = clk;
(5) assign #10 a = clk;
(6) assign #15 a = clk;

q7]Y=a?p:q;
If a=1'bx then whats the value of y?

Q8]Whats the notation to see hirarcy in display?
%m
Q9]What is the basic difference between verilog HDL and VHSIC-HDL
Coming to the difference between VHDL and verilg HDL ..The configure statement in VHDL is very useful which is not found in Verilog.
VHDL's high-level constructs (such as configuration, generate, generic and unconstrained arrays) have no equivalency in verilog. One the other hand, verilog has very good low-level constructs for gate-level modeling.

Swapna.. u r right.. verilog is very easy to learn when compared to VHDL. VHDL being a designer's choice with high HDL modeling capability, is chosen over verilog considering the bulkness of the design and its re-usability. That doesnt mean that verilog is overlooked. Small units in the large design are manged with Verilog coding and large units with VHDL.

Now a days we are having standard IPs available in mixed form. Later they are integrated to make a System on Chip. Depending on the complexity and integrator's choice VHDL wins...

However.. Verilog is syntesis guy's choice(at gate-level).. as it gives maximum flexibility with good constructs for modelling the cell-primitives. Down the grade at Gate and logic level, verilog certainly wins over VHDL.

We use mentor graphics' MODEL SIM for verilog and VHDL.. also we use NCsim for fewer designs.. The design I work on is a mixed one(verilog and VHDL).. so for mixed design simulations.. Modelsim is the best option..
Q10]. There is a language, which is a derivative of verilog, used to model analog and Mixed signal designs..?
Verilog AMS
Q11]. What is the difference between wire and reg?
Net types: (wire,tri)Physical connection between structural elements. Value assigned by a continuous assignment or a gate output.
Register type: (reg, integer, time, real, real time) represents abstract data storage element. Assigned values only within an always statement or an initial statement.

the main difference between wire and reg is wire cannot hold (store) the value when the no connection between a and b like
a-------------b, if there is no connection in a nad b, wire loose value.
but reg can hold the value even if there in no connection.
Default values:wire is Z,reg is x.

Q12]Between the if-else and case statements which is usually preferred?I heard that case is better for synthesis..Iam not sure.I have not done any synthesis part.
Case is better from synthesis point of view.
if else will be synthesised to a priority encoder.
Whereas case will be synthesised to a normal encoder.
Priority encoder has more gates and also timing is affected.
So,case is usually preferred.
There are switches that design compiler(synopsys synthesis tool) provides to synthesize case statement either way.

Q13)
In verilog,what is the basic difference between a function and a task?
1. Fucntions must have atleast one input arguement...Tasks can have zero or more arguments.
2. Functions return always a single value..Tasks do not return with a value but can pass multiple values through output and inout arguments
3. Functions must not have delay,event or timing control statements..Tasks may contain delay,event or timing control statements.
4. Function can enable another function only but not another task whereas Task can enable other functions and tasks.
Q14]What is delta delay concept in verilog?

Q15]What are Intertial and Transport Delays ??
The inertial delay model is specified by adding an after clause to the signal assignment statement. To model this delay in the SR latch example, we could replace the two signal assignments with the following two statements.

q <= r nor nq after 1ns;
nq <= s nor q after 1ns;
Now during simulation, say signal r changes and will cause the signal q to change, rather than schedule the event on q to occur during the next round, it is scheduled to occur 1ns form the current time. Thus the simulator must maintain a current time value. When no more events exist to be processed at the current time value, time is updated to the time of the next earliest event and all events scheduled for that time will be processed.

Note that the change will not occur in q until 1ns after the change in r. Likewise the change in nq will not occur until 1ns after the change in q. Thus, the "after 1ns" models an internal delay of the nor gate.

Typically, when a component has some internal delay and an input changes for a time less than this delay, then no change in the output will occur. This is also the case for the inertial delay model.

Although most often the inertial delay is desired, sometimes all changes on the input should have an effect on the output. For example, a bus experiences a time delay, but will not "absorb" short pulses as with the inertial delay model. In these cases, we can use transport delay model. The transport delay model just delays the change in the output by the time specified in the after clause. You can elect to use the transport delay model instead of the inertial delay model by adding the keyword transport to the signal assignment statement.

The SR latch example could be modified to use the transport delay model by replacing the signal assignments with the following two statements.

q<=transport r nor nq after 1ns;
nq<=transport s nor q after 1ns;

What is the difference between the following verilog statements, Explain:
#10 a = b + c;
a = #10 b + c;

Assume timescale of 1 ns/ 1 ps

#10 a = b+ c : This delay model is called regular delay control. Regular delays defer the execution of the entire assignment. That is, in this eg, after 10ns only, b+c will be calculated and assigned to a.

a = # b + c : This is Intra-assignment delay control.Intra-assignment delays compute the RHS expression at the current time and defer the assignment of the computerd value to the LHS variable. Here b+c will be calculated with current values of b& c but a will get that value only after 10ns

Q16]Design of a 4:1 mux in Verilog is shown below in two styles of coding:

(i) Using if-else statements
```
if(sel_1 == 0 && sel_0 == 0) output = I0;
else if(sel_1 == 0 && sel_0 == 1) output = I1;
else if(sel_1 == 1 && sel_0 == 0) output = I2;
else if(sel_1 == 1 && sel_0 == 1) output = I3;
```

(ii) Using case statement
```
case ({sel_1, sel_0})
00 : output = I0;
01 : output = I1;
10 : output = I2;
11 : output = I3;
default : output = I0;
endcase
```
Answer the following Questions:

(a) What are the advantages / disadvantages of each coding style shown above?
(b) How Synthesis tool will give result for above codes?
(c) What happens if default statement is removed in case statement?
(d) What happens if combination 11 and default statement is removed?

(a) Functionality wise and simulation point of view, both the implementaions are the same and no extra advantage/disadvantage over the other, except case statement makes code more readable and simple.

(b) Coming to synthesis, if the conditions are mutually exclusive, as shown in this example, case will synthesize it parallely and if-else synthesizes to a priority encoder, which is exrtra logic. How ever, if the conditions are not mutually exclusive, even case synthesizes it to priority encoder. So if we want parallel in that condition also, we need to use parallel_case for synthesis as shown below:
```
case(1) : // parallel_case
sel_a : out <= a;
sel_b : out <= b;
end
```

(c) Even if you remove default, nothing will happen as all other possible input combinations are defined properly.

(d) If both default and 11 are removed then latch will be formed. Suppose, if 11 comes at input, as nothing matches O/P will not change, that means it is holding the prev value..which is Latch. To avoid this we need default always

Q17
What is the difference between blocking and non blocking statements?
Give some examples.

Blocking Statements:A blocking statement must be executed before the execution of the statements that follow it in a sequential block. They will be executed in the same seq order as they are entered.
= is used

Nonblocking Statements: The nonblocking statements allows you to schedule assignments without blocking the procedural flow. You can use the nonblocking procedural statement whenever you want to make several register assignments within the same time step without regard to order or dependence upon each other. It means that nonblocking statements resembles the actual hardware more then the Blocking assignments.

Example:

```
module block_nonblock();
reg a, b, c, d , e, f ;

// Blocking assignments
initial begin
a = #10 1'b1; // The simulator assigns 1 to a at time 10
b = #20 1'b0; // The simulator assigns 0 to b at time 30
c = #40 1'b1; // The simulator assigns 1 to c at time 70
end

// Nonblocking assignments
initial begin
d <= #10 1'b1; // The simulator assigns 1 to d at time 10
e <= #20 1'b0; // The simulator assigns 0 to e at time 20
f <= #40 1'b1; // The simulator assigns 1 to f at time 40
end

endmodule
```

18]What is the difference between $display and $monitor is verilog?

Both $monitor and $display both are used to display the varibales.
Whereas $monitor displays O/P whenever there is some change in the signal. And should be invoked only once. If u use more than once, only last one will be executed all others will be overwritted.

Whereas $display diaplays the signal continiously. It can be used any number of times. Very similar to printf in C.

19]What does `timescale 1 ns/ 1 ps signify in a verilog code?

'timescale directive is a compiler directive.It is used to measure simulation time or delay time.
Usage : `timescale <reference_time_unit>/ <time_precision>

reference_time_unit : Specifies the unit of measurement for times and delays.
time_precision: specifies the precision to which the delays are rounded off

20]wots the diff bet synchronous and asynchronous FIFO in the application point of view
When your read and write clocks are synchronous, you use synchronous fifos.
If they are in different clock domains,asynchronous fifos are used.
21]
What is the difference between === and == ?[/b]

Ans)
output of "==" can be 1, 0 or X.

output of "===" can only be 0 or 1.

When you are comparing 2 nos using "==" and if one/both the numbers have one or more bits as "x" then the output would be "X" . But if use "===" outpout would be 0 or 1.

e.g A = 3'b1x0
B = 3'b10x
A == B will give X as output.
A === B will give 0 as output.

"==" is used for comparison of only 1's and 0's .It can't compare Xs. If any bit of the input is X output will be X

"===" is used for comparison of X also...

22]What is the difference among case,casex and casez?
case treats only 0 or 1 values in case alternative and is is not dealing with don't care condition.

casex treats all x and z values in case alternative or case expession as a don't care.

casez treats all z values in case alternatives. all bit positions with z can treat as a don't care
23]what is full case and parallel case.

24]how i declare array for input port.
like input [1:0] in [7:0]; reg is not allowed 4 input port and if i declare wire its give error.

25Q)1.can we use case statement within case statement in verilog
if we synthesize what hardware will it infer.
does any one has used like this in there code

I feel that it is quite possible to have nested case statements.
Suppose consider the following verilog code:

```
always @(s0,s1,i0,i1,i2,i3)
begin
case(s1)
0: case(s0)
0: out <= i0;
1: out <= i1;
endcase
1: case(s0)
0: out <= i2;
1: out <= i3;
endcase
endcase
end
```

This code works properly in simulations..funcitonality wise
And coming to synthesis part,
I feel it will be similar to the implementation of a 4:1 mux using 2 2:1 muxes...

Q26. In how many ways we can write a 2:1 or 4:1 Mux?

Using Gate Level Modelling --> and, not and or gate used:

```
// 4-to-1 multiplexer.

module mux4_to_1 (out, i0, i1, i2, i3, s1, s0);

// Port declarations from the I/O diagram
output out;
input i0, i1, i2, i3;
input s1, s0;

// Internal wire declarations
wire s1n, s0n;
wire y0, y1, y2, y3;

// Gate instantiations

// Create s1n and s0n signals.
not (s1n, s1);
not (s0n, s0);

// 3-input and gates instantiated
and (y0, i0, s1n, s0n);
and (y1, i1, s1n, s0);
and (y2, i2, s1, s0n);
and (y3, i3, s1, s0);

// 4-input or gate instantiated
or (out, y0, y1, y2, y3);

endmodule
```

~~~~~~

2) Using Gate Level Modelling --> bufif constructs used

```
module mux2_to_1(out, a, b, control);
output out;
input a, b, control;
tri out;
wire a, b, control;

//drives a when control = 0; z otherwise
bufif0 b1(out, a, control);

//drives b when control = 1; z otherwise
bufif1 b2(out, b, control);

endmodule
```
~~~~~~

27) What is the difference between if-else statement
and case statement? which is used more? why?

The main difference between if-else and case is if-else is more powerful in the sence that the expressions need not be of the same type where as in case the expression must be of same type.

Take a example of simple 4:1 mux explained above in ANS 27 using case and if-else. Now consider an example where expressions or scenarios are different from each other and does not belong to the same black box like mux and according to the o/p of the expression i need to execute statement/s. In this case you can't use case where as if-else works perfect.

But, if you want to write a 16:1 mux or something like that then case always comes handy where as if-else becomes unwieldy if there are too many alternativies.

Q28. Interesting one.

```
always @(clk)
begin
a = 0;
a <= 1;
$display(a);
end
```

what value of 'a' is displayed?
The value will be compiler dependant. The compiler keeps the various executable statements in Queue...
say
at delta, instr1
at 2 delta, instr2 etc..

based on this Queue, the last statement that got executed will have that value.

Q29]
in conditional operator if logical statement is true then it will execute first expression else it will execute 2nd expression. my question is
what happen if the result of logical statement is don't care i.e. x?which expession it will execute?

Q30
how i generate sine wav using verilog coding stle, can u hav any idea regarding this, or using lut based for sine and cosine wave.
A: The easiest and efficient way to generate sine wave is using CORDIC Algorithm.

Q31. Given the following code:

```
reg clk;
reg a;
always #10 clk = ~clk;
```

draw the waveforms for 'a':

```
always @(clk) a = #15 clk;
```

Srikanth will it be a flat '0' or do we get some values at 'a'?? As it's a intra-assignment delay control where the value of the R.H.S is computed at current simulation time and is assigned after the stipulated delay, 15 time units in this case, we should get some values for 'a'..what you say??

Differentiate between Inter assignment Delay and Inertial Delay ?

What are the different State machine Styles ? Which is better ? Explain Disadv. and Advan. ?

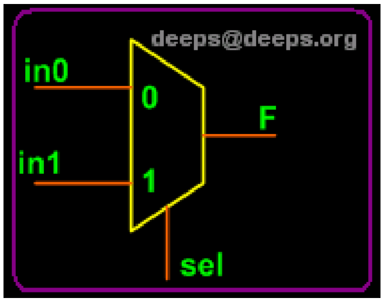
What is the difference between the following lines of code ?

- reg1<= #10 reg2 ;
- reg3 = # 10 reg 4 ;

What is value of Var1 after the following assignment ?

```
reg Var1;
initial begin
  Var1<= "-"
end
```

Consider a 2:1 mux , what will be the output F if the Select (sel) is "X" ?



What is the difference between blocking and non-blocking assignment?

What is the difference between wire and a reg data type?

Write code for async reset D-Flip-Flop.

Write code for 2:1 MUX using different coding methods.

Write code for parallel encoder and priority encoder.

What is the difference between === and == ?

Why is defparam used for ?

What is the difference between unary operator and logical operator ?

What is the difference between task and function ?

What is the difference between transport and inertial delays

What is the difference between casex and case statements ?

What is the difference between $monitor and $display ?

What is the difference between compiled, interpreted, event based and cycle based simulator ?

Verilog mostly asked in interviews: -

Each Verilog simulation time step is divided into 4

Q1 — *(in any order)* :

- Evaluate RHS of all non-blocking assignments
- Evaluate RHS and change LHS of all blocking assignments
- Evaluate RHS and change LHS of all continuous assignments
- Evaluate inputs and change outputs of all primitives
- Evaluate and print output from $display and $write

Q2 — *(in any order)* :

- Change LHS of all non-blocking assignments

Q3 — *(in any order)* :

- Evaluate and print output from $monitor and $strobe
- Call PLI with reason_synchronize

Q4:

- Call PLI with reason_rosynchronize

--------------------------------------- Blocking / Nonblocking related ---------------------------
----

Blocking assignment "=" -- use for combinational logic

One-step process:

1. Evaluate the RHS and update the LHS of the blocking assignment without interruption from any other Verilog statement.

A blocking assignment "blocks" trailing assignments in the same always block from occurring until after the current assignment has been completed

Nonblocking assignment "<=" -- use for secq. logic
Two-step process:
1. Evaluate the RHS of nonblocking statements at the beginning of the time step.
2. Update the LHS of nonblocking statements at the end of the time step.

Nonblocking assignments are only made to register

Verilog race conditon - When 2 statements are sheduled to execute in one time step, & give different results when order of executtion is changed.

example -

```
always @(posedge clk or posedge rst)
        if (rst) y1 = 0; // reset
        else y1 = y2;
always @(posedge clk or posedge rst)
        if (rst) y2 = 1; // preset
        else y2 = y1;
```

will give y1 & y2 eq 1 if first block executes first. will give both 0 if second block exe first.

self triggering blocks -

```
module osc2 (clk);
output clk;
reg clk;
initial #10 clk = 0;
always @(clk) #10 clk <= ~clk;
endmodule
```

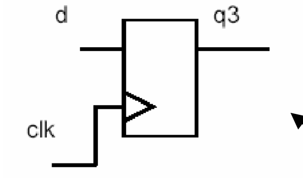
After the first @(clk) trigger, the RHS expression of the nonblocking assignment is evaluated and the LHS value scheduled into the nonblocking assign updates event queue. Before the nonblocking assign updates event queue is "activated," the @(clk) trigger statement is encountered and the always block again becomes sensitive to changes on the clk signal. When the nonblocking LHS value is updated later in the same time step, the @(clk) is again triggered.

```
module osc1 (clk);
output clk;
reg clk;
initial #10 clk = 0;
```

always @(clk) #10 clk = ~clk;
endmodule

Blocking assignments evaluate their RHS expression and update their LHS value without interruption. The blocking assignment must complete before the @(clk) edge-trigger event can be scheduled. By the time the trigger event has been scheduled, the blocking clk assignment has completed; therefore, there is no trigger event from within the always block to trigger the @(clk) trigger.

Bad modeling: - (using blocking for seq. logic)



always @(posedge clk) begin
q1 = d;
q2 = q1;
q3 = q2;
end

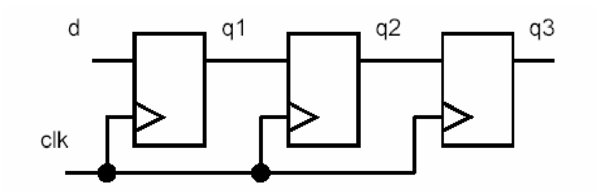
always @(posedge clk) q1=d;
always @(posedge clk) q2=q1;
always @(posedge clk) q3=q2;

Race condition

always @(posedge clk) q2=q1;
always @(posedge clk) q3=q2;
always @(posedge clk) q1=d;

always @(posedge clk) begin
q3 = q2;
q2 = q1;
q1 = d;
end

Bad still works



Good modeling: -

always @(posedge clk) begin
q1 <= d;

```
q2 <= q1;
q3 <= q2;
end

always @(posedge clk) begin
q3 <= q2;
q2 <= q1;
q1 <= d;
end

always @(posedge clk) q1<=d;
always @(posedge clk) q2<=q1;
always @(posedge clk) q3<=q2;

always @(posedge clk) q2<=q1;
always @(posedge clk) q3<=q2;
always @(posedge clk) q1<=d;
```

No matter of sequence for Nonblocking

Good Combinational logic :- (Blocking)

```
always @(a or b or c or d) begin
tmp1 = a & b;
tmp2 = c & d;
y = tmp1 | tmp2;
end
```

Bad Combinational logic :- (Nonblocking)

```
always @(a or b or c or d) begin
tmp1 <= a & b;
tmp2 <= c & d;
y <= tmp1 | tmp2;
end
```

will simulate incorrectly…
need tmp1, tmp2 insensitivity

Mixed design: -

Use Nonblocking assignment.

In case on multiple non-blocking assignments last one will win.

Timing Control / Delays:

A timing control before assignment -> Evaluation is delayed for the amount of time.

#5 A = 1; ---- delay for 5, then evaluate and assign
#5 A = A + 1; -----delay 5 more, then evaluate and assign
B = A + 1; -----no delay; evaluate & assign

An *intra-assignment delay* places the timing control after the assignment
The right-hand side is evaluated before the delay. The left-hand side is assigned after the delay

always @(A) ------------The **Bold** letters indicate intra assignment delays
B = **#5** A; -- A is evaluated at the time it changes, but assigned to B after 5 time units

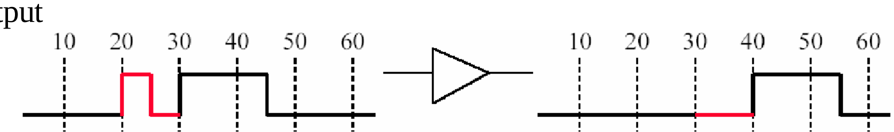
always @(negedge clk) --- D is evaluated at the negative edge of is CLK,,
Q <= **@(posedge clk)** D; --- Q is changed on the positive edge of CLK

Intra-Assignment Delays
With Repeat Loops
An edge-sensitive intra-assignment timing control permits a special use of the repeat loop

- The edge sensitive time control may be repeated several times before the delay is completed
- Either the blocking or the non-blocking assignment may be used

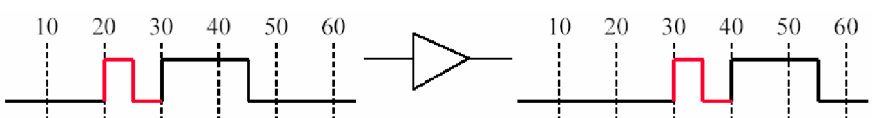
always @(IN)
OUT <= **repeat (8) @(posedge clk)** IN;
The value of IN is evaluated when it changes, but is
not assigned to OUT until after 8 clock cycles
Hardware has two primary **propagation delay methods**:

Inertial delay models devices with finite switching speeds; input glitches do not propagate to the output

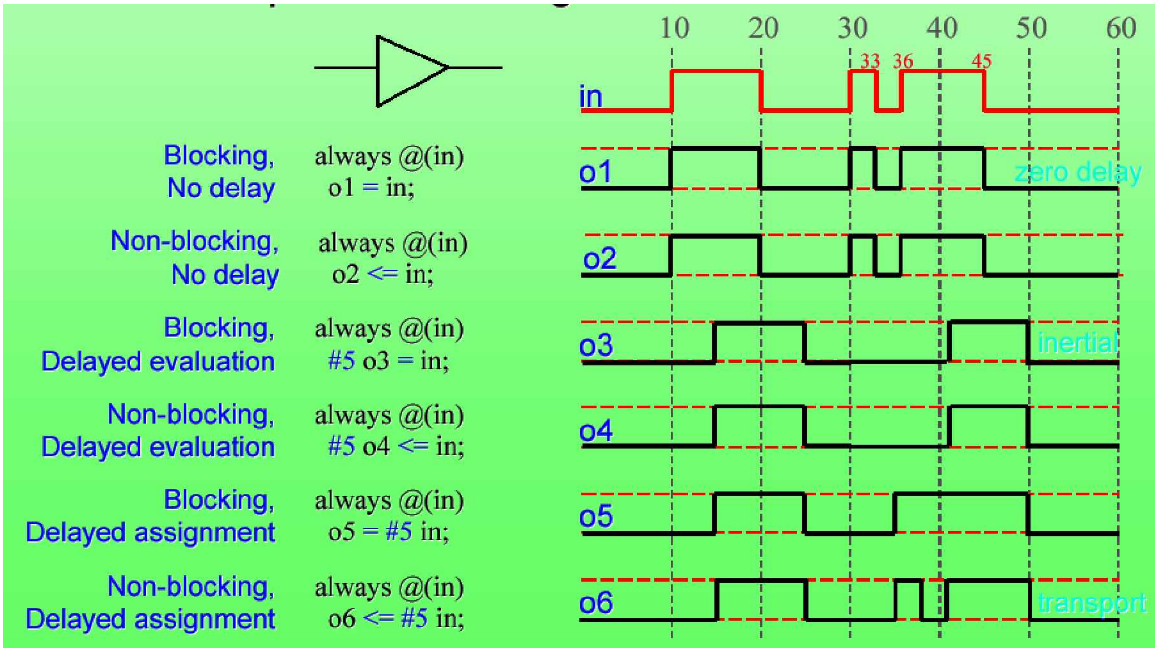


Buffer with a 10 nanosecond propagation delay

Transport delay models devices with near infinite switching speeds; input glitches propagate to the output



Buffer with a 10 nanosecond propagation delay



Combinational Logic:
No delays: Use blocking assignments (a = b;)
Inertial delays: Use delayed evaluation blocking assignments (#5 a = b;)
Transport delays: Use delayed assignment non -blocking assignments (a <= #5 b;)

Sequential Logic:
No delays: Use non blocking assignments (q <= d;)
With delays: Use delayed assignment non -blocking assignments (q <= #5 d;)

First, once the **always** block is entered due to a change on the sensitivity list variable **in**, subsequent changes on **in** will not cause re-entry until the **always** block is exited 65 time units later. Second, after a delay of 25 time units, the current value of **in** is read, inverted, and assigned to **out1**. After an additional 40 time units, **in** will again be read, inverted, and assigned to **out2**. During the timing delays, all other events on **in** will be ignored. The outputs will not be updated on every input change if changes happen more frequently than every 65 time units. The post-synthesis gate-level model will simulate two inverters while the pre-synthesis RTL code will miss multiple input transitions. Placing delays on the left side of **always** block assignments does not accurately model either RTL or behavioral models.

```
module code11 (out1, out2, in);
output out1, out2;
input in;
reg out1, out2;
always @(in) begin
#25 out1 = ~in;
```

```
#40 out2 = ~in;
end
endmodule
```

Write a procedure for an adder (combinational logic) that assigns C the sum of A plus B with a 7ns propagation delay.

```
always @(A or B)
#7 C = A + B;
```

Write a Verilog procedure for a “black box” ALU pipeline that takes 8 clock cycles to execute an instruction. The pipeline triggers on the positive edge of clock. The “black box” is represented as call to a function named ALU with inputs A, B and OPCODE

```
always @(posedge clk)
alu_out <= repeat(7) @(posedge clk) ALU(A,B, OPCODE);
```

Functions:
Functions always synthesize to combinational logic. The problem occurs when engineers make a mistake in the combinational function code and create simulation code that behaves like a latch.

```
always @(a or nrst or en)        // Infers a latch with asynchronous low-true
if (!nrst) o = 1'b0;             // nrst and transparent high latch enable "en"
else if (en) o = a;
always @(a or nrst or en)
o = latch(a, nrst, en);

function latch;          // Infers three input and gate
input a, nrst, en;
if (!nrst) latch = 1'b0;
else if (en) latch = a;
endfunction
```

casex & casez
The use of **casex** statements can cause design problems. A **casex** treats ‘X’s as "don't cares" if they are in either the case expression or the case items. The problem with **casex** occurs when an input tested by a **casex** expression is initialized to an unknown state.
pre-synthesis -> unknown input as a "don't care"
post-synthesis -> propagate ‘X’s through the gate-level

Guideline: Do not use casex for RTL coding.

The use of **casez** statements can cause the same design problems as **casex**, but these problems are less likely to be missed during verification. With **casez,** a problem would occur if an input were initialized to a high impedance state.

Assigning 'X' initialization

The 'X' assignment is interpreted as an unknown by the Verilog simulator (with the exception of casex), but is interpreted as a "don't care" by synthesis tools.

FSM designs, 'X' assignment to the state variable can help debug bogus state transitions. This is done by defaulting the next state registers to 'X' prior to entering the **case** statement, resulting in 'X' for any incorrect state transitions.

```
always @(a or b or c or s) begin
  y = 1'bx;
  case (s)
    2'b00: y = a;
    2'b01: y = b;
    2'b10: y = c;
  endcase
end
```

```
always @(a or b or c or s)
  case (s)
     2'b00:         y = a;
     2'b01:         y = b;
     2'b10, 2'b11: y = c;
   endcase
```

// Note: the second example synthesizes to a smaller and faster implementation than the first example.

// **Hiding the initialization of variables** from the synthesis tool is a very dangerous practice!!
// synopsys translate_off
initial y1 = 1'b1;
// synopsys translate_on

1.TTL :Transistor-Transistor Logic.
DTL :Diode-Transistor Logic.
ECL :Emitter-Coupled Logic.
2.TTL : Invented by "TEXAS INSTRUMENTS" ,in 1964. Mostly used in SSI and MSI ICs.
3.ECL :Fastest logic device,used for high speed aplications.
4.Basic technologies, used for digital ICs are :-
a>BIPOLAR :Prefferred for SSI and MSI and is faster.
b>MOS Technology :Fabricates MOSFET transistors on a chip ,

prefferred for LSI as more MOSFETs can be packed in the same chip area.

5.Three levels of integration are :-
a>SSI :Has less than 12 transistors per chip.
b>MSI :Has more than 12 ,but less than 100 transistors per chip.
c>LSI :Has more than 100 transistors per chip.

6.Families in MOS categories :-
a>PMOS :P-Channel MOSFETs ::oldest ,slowest but now obsolete.
b>NMOS :N-Channel MOSFETs ::mostly used for general purposes, in microprocessors and memories. Dominates the LSI field.
c>CMOS :A push-pull arrangement of n- and p- channel MOSFETs, extensively used where low power consumption is needed like calculators ,digital wrist-watches etc.

7.7400 DEVICES :: TTL series of SSI AND MSI chips only.
It consists of basically a number of catagories,like :- a>STANDARD TTL :These are the logical circuits ,in which a combination of transistors are taken ,to produce the outputs ,depending on the inputs.The output is generrally taken across a TOTEM-POLE or across an OPEN-COLLECTOR OUTPUT.
Different TTL devices generally used are :-

7400 : NAND GATE (Cheapest).
7402 : NOR GATE (A 8-input TTL is not available).
7404 : NOT GATE.
7408 : AND GATE.
7432 : OR GATE.
7486 : XOR GATE.
74147 : BCD Encoder.
74154 : Binary to Hexadecimal Decoder.
7446/7447 : Seven-Segment Decoder.
7414 : Schmitt Trigger.
74150 : 16 to 1 Multiplexer.It is also called "UNIVERSAL LOGIC CIRCUIT" as it can be used to develop any truth table in the world.
DEVICE NUMBERING SYSTEM :7400,7402,7486 etc.
POWER DISSIPITATION :10 mw.
PROPAGATION DELAY TIME :10 ns.

b>HIGH-SPEED TTL :The resistance in standard TTL is decreased. =>

The RC-output time-constant gets lowered.
=> The PROPAGATION DELAY TIME gets decreased.
=> High power dissipitation due to small resistance.
DEVICE NUMBERING SYSTEM :74H00,74H02,74H86 etc.
POWER DISSIPITATION :22 mw.
PROPAGATION DELAY TIME :1 ns.
c>LOW-POWER TTL :The resistance in standard TTL is increased.
=> The RC-output time-constant getsincreased.
=> The PROPAGATION DELAY TIME gets increased.
=> Low power dissipitation due to high resistance.
DEVICE NUMBERING SYSTEM :74L00,74L02,74L86 etc.
POWER DISSIPITATION :1 mw.
PROPAGATION DELAY TIME :35 ns.

d>SCHOTTKY TTL :In this case ,a schottky diode is connected across the base and collector of each transistor so that the "SATURATION DELAY TIME" virtually gets eliminated.
As its forward voltage is 0.4 V ,it prevents the transistor from saturating fully ,thus eliminating "S.D.T." fully.
DEVICE NUMBERING SYSTEM :74S00,74S02,74S86 etc.
POWER DISSIPITATION :20 mw.
PROPAGATION DELAY TIME :3 ns.

e>LOWER-POWER SCHOTTKY TTL :Both resistance is increased and SCHOTTKY DIODE is used.
DEVICE NUMBERING SYSTEM :74LS00,74LS02,74LS86 etc.
POWER DISSIPITATION :2 mw.
PROPAGATION DELAY TIME :35 ns.

FOR ALL 7400 SERIES TTLs:-
TEMPERATURE RANGE : 0 DEGREE TO 70 DEGREE CELCIUS.
VOLTAGE RANGE :4.75 V TO 5.25 V

8.5400-DEVICES: Used for military purposes. Highly expensive.
TEMPERATURE RANGE :-55 DEGREE TO 125 DEGREE CELCIUS.
VOLTAGE RANGE :4.50 V TO 5.50 V
9.PROPAGATION DELAY TIME :It is the amount of time ,it takes for the output of a gate to change after a input has changed.
10.SATURATION DELAY TIME :It is the time required by a transistor to bring back all its extra/stray charges to there original eras ,so that the transistor can change its state spountaneously.
It is virtually eliminated when a SCHOTTKY DIODE is used in Emitter-Base-MODE with the transistor due to a forward voltage of 0.4 V.
11.FLOATING POINT INPUT :It is the open-end of a wire in any

electronic device ,which is not attached to any specific device and thus keep floating in air.It acts like an high input ,and can be used in a TTL device ,but is avoided in a CMOS device ,as it increases the total heat dessipitation in the CMOS too much.

12.WORST CASE PARAMETERS FOR TTL DEVICES :-

Maximum input current ,minimum output voltage etc. measured under worst conditions of maximum temperature and minimum voltage for som parameters like minimum temperature and maximum voltage for others.

13.WINDOW PROFILE :

INPUT OUTPUT
5.0 V ------------------------------------5.0 V

2.4 V ---] 0.4 V
--2.0 V

--0.8 V] 0.4 V
0.4 V ---

14.NOISE MARGIN :Iis the protective range of voltage difference ,created between the optimum value and the actual value due to the presence of stray charge particles ,upto which ,the '0' or '1' state will not toggle.

It is equal to 0.4 V for TTL = 2.4 - 2.0 = 0.8 - 0.4 = 0.4 V.

15.FAN-OUT :It is the maximum number of TTL devices which can be safely and relaibly connected to the output of a TTL under certain , specified temperature.

For standard TTL ,FANOUT = 10.

For LOW-POWER SCHOTTKY TTL ,FANOUT = >10.

16.FAN-IN :It is the maximum number of TTL devices which can be safely and relaibly connected to the input of a TTL under certain , specified temperature.

17.BUFFER :It is a device that isolates two other devices ,due to its high input impedence and low output impedence.

18.SCHMITT TRIGGER :It is a circuit ,designed to produce a rectangular outout ,regardless of the nature of the input waveform.

______ ______
______/ ______ => S. T. => _______| |_______

19.OPEN-COLLECTOR OUTPUT :It is the output mode ,utilised in a few

TTL devices ,in which only one transistor of the TOTEM-POLE output is utilised and the output is taken across the open collector of the lower transistor only.
20.MULTIPLEXER : " MANY to ONE ".It is a circuit with many inputs and only one output.
It is used as a :-
1>DATA-SELECTOR.
2>BOOLEAN FUNCTION-GENERATOR.
3>WORD SELECTOR.

****

REGISTERS AND COUNTERS

****

1.REGISTERS :These are a group of memory elements that work together as a unit and perform simple tasks of storing a binary word ,modifying it , left /right shifting it ,loading or read-enabling and a number of other operations on binary words.
2.DIFFERENT TYPES OF REGISTERS ARE :-
1>BUFFER REGISTER :Used to simply store a binary word and contains only D-flip-flops.
2>SHIFT-LEFT REGISTER :Shifts the bit-contents to left by one ,thus deleting the "MSB" and taking '1' or '0' as input for "LSB".D-flip-flops are used with each "Q" inputted to successive "D".
3>SHIFT-RIGHT REGISTER :Shifts the bit-contents to right by one ,thus deleting the "LSB" and taking '1' or '0' as input for "MSB".D-flip-flops are used with each "Q" inputted to previous "D".
4>PARALLEL or BROADSIDE-LOADING :In this case ,same clock is entered in parallel ,to each D-flip-flop ,thus only one clock pulse is required to store a digital word.

3.COUNTERS :These are special type of registers used to count the numbers of clock pulses arriving at its input.

4.DIFFERENT TYPES OF COUNTERS ARE :-
1>RIPPLE COUNTER :Built with JK-flip-flops.
All "J-K" are kept at common high.
"Q" is transferred to successive clock as negative triggered pulse.
Counts total no. of clock pulses arrived in binary format.
Timing-diagram is of ripple form.

NOTE :- As each flip-flop divides the frequency by two ,they are also called "divide-by-2 circuits" and if there are N flip-flops in "ripple counter" then ,it will be a "divide-by-2**n circuit".

2>PROGRAM COUNTER :Same as ripple counter ,only "COMMON HIGH" can be "switched at wish" and us named as "COUNT". Used to keep track of the instruction being executed.
3>SYNCHRONOUS COUNTER :Used to avoid "RIPPLE-DELAY PROBLEM".
Same as "RIPPLE COUNTER" except :-
a>CLOCK is connected to each flip-flop ,exclusively.
b>AND combination of "Q0" and "Q1" are inputted to "J2" and "K2".
c>AND combination of "Q1" and "Q2" are inputted to "J3" and "K3".
d>"Q0" is inputted to "J1" and "K1".
e>"HIGH" is inputted to "J0" and "K0".
4>RING COUNTER :Instead of counting a binary number ,a "RING COUNTER" uses a word that has only one "HIGH" bit.
All the "Qs" are inputted to successive "Ds" and "Q3" is inputted to "D0".
"D-flip-flops" are used.
For "N" flip-flops ,it can count count only from "0" to "N-1" in decimals.
Used when at a precise time ,one and only one register is to be made active ,out of a handful of registers.

5>MODULUS COUNTER :"MODULUS" is the no. of outputs a counter has.
EXAMPLE :-
"mod-10 COUNTER " or "divide-by-10 COUNTER" ,counts from "0" to "9" in binary ,and then gets reset to "0" again.
Used as a "BCD-GENERATOR".
Same as RIPPLE COUNTER ,only the NANDed output of "Q3" and "Q1" are ANDed to the negative "CLEAR" pulse.
6>DOWN COUNTER :Counts from the highest to lower number and at last gets preset to highest number again.
Simply ,the " Q' "s are inputted as the negative clocks for the successive "JK" flip-flops.
7>PRESETTABLE COUNTER :In this case ,the count starts from a number ,greater than zero and ends before the maximum possible value has been reached.

5.EXAMPLES OF "TTL COUNTERS" ARE :-
7490 Decade.
7492 Divide-by-12.
7493 Divide-by-16.

6.THREE-STATE SWITCH :Invented in 1970s. Ideal for "BUS-ORGANIZED COMPUTERS".

ENABLE	Din	Dout
0	X	Open
1	0	0
1	1	1

It has three states of outputs :-
1>LOW.
2>HIGH.
3>FLOATING or HIGH-IMPEDENCE STATE.

It is of two types :-
1>NORMALLY OPEN SWITCH :To close or activate it ,high "ENABLE" is to be applied.
2>NORMALLY CLOSE SWITCH :To close or activate it ,low "ENABLE" is to be applied.

7.BUS-ORGANIZED COUNTERS :These are actually a combination of a large number of registers ,connected through the same channel or set of wires so that ,the word passing through it can be utilised by any/some/all of the registers in parallel to make the job more efficient. Most popular COUNTER ,used in nowadays "COMPUTER ARCHITECTURE".

8.SAP :"SIMPLE-AS-POSSIBLE".It is the name of computer, generally used to understand the basic architecture of computers.

****

MEMORIES

****

1.MEMORY :It is the location where the program and data are stored ,before the calculations begin.
2.Two different types of memory are :-
1>ROM :Read Only Memory.

2>RAM :Random Access Memory.
3.ROM :Provides non-volatile storage of programs and data.
It retains the stored data even when the power to the device is shut off.
Its different types are :-
1>PROMs :PROGRAMABLE ROMs :These allow the user to store data using a PROM programmer by "BURNING IN" .Once written,it can't be erased.
2>EPROMs :ERASABLE PROMs :Their data can be erased using "ULTRA-VIOLET RADIATIONs" and then can be rewritten.
3>EEPROMs :ELECTRICALLY EPROMs :Non-volatile like "EPROM" but can be erased completely or partially using electrical pulses.Writing to EEPROM is slower than writing to RAM ,so it can't be used in high-speed circuits.It gets weared out after a few thousands of "ERASE" cycle.

4.ACCESS TIME :The access time of memory is the time it takes to read a stored word after applying address bits.
Bipolar memories are faster but more than MOSFETs.The access time for a general Bipolar PROM is 80 ns while that of a general MOSFET EPROM is 450 ns.

5.RAMs :Random Access Memory or Read-Write Memory :Equivalent to a group of addressable registers.
These are allways volatile i.e. They lose the stored data whenever power is turned off.
Different types of RAM-MEMORIES are :-
1>core RAM :"Non-volatile" ,used in earlier computers but expensive and harder to work than semiconductor memories.

2>semiconductor RAMs :These may be of the following two types:-
a>STATIC RAM :It uses Bipolar or MOSFET flip-flops ,data is retained indefinitely as long as power is applied to the flip-flops.
These act as "TRANSISTOR LATCHES" and can thus store data for indefinitely long time.
It is preferred to DYNAMIC RAMs.

b>DYNAMIC RAM :It uses MOSFETs and CAPACITORs to store data.
Becouse the capacitor charge leaks off ,the stored data must be refreshed (recharged) after every few milli-seconds.

Dynamic RAM has more memory locations than a static RAM of same physical size as in this case only a single MOSFET and CAPACITOR are needed to store a bit.

Main disadvantage of the DYNAMIC RAM is the need to refresh the capacitor every few milliseconds which complicates the design problem because more circuitry is needed.

6.GENERAL SIGNALS ,INPUTTED TO RAMs ARE :-
1>WE!-signal :WRITE-ENABLE SIGNAL.
LOW :WRITING WILL BE DONE.
HIGH :READING WILL BE DONE.
2>CE!-signal :CHIP-ENABLE SIGNAL.
LOW :THE MEMORY LOCATION WILL BE CAPABLE OF RECIEVING WE-signal.
HIGH :HOLDING OF MEMORY WILL TAKE PLACE i.e. NOTHING CAN BE DONE ON THAT MEMORY LOCATION.
3>ADDRESS signal :SELECTS THE SPECIFIED MEMORY LOCATION.
4>Din signal :INPUTS or FEEDS THE DATA INTO THE MEMORY-LOCATION.

7.BUBBLE MEMORY :It sandwiches a thin film of magnetic materials between two permanent bias magnets.NON-VOLATILE and CAPABLE OF STORING HUGE AMOUNT OF DATA IN A VERY SMALL AREA ,but it has SLOW ACCESS TIME.

8.DIFFERENT MODES OF SENDING INSTRUCTIONS TO THE MEMORY SO THAT A SPECIFIC TASK CAN BE PERFORMED ARE :-
1>HARDWIRED CONTROL MATRIX :In this case ,microinstructions for each instruction ,is generated in each execution cycle through a set of "CONTROL MATRICES" ,soldered together.

2>MICROPROGRAMMING :In this case ,the microinstructions are stored in ROM ,instead of being produced in each step.
THREE STEPS REQUIRED FOR ACCESSING ANY ROUTINE ARE :-
A>Knowing the starting address of the routine.
B>Stepping through the routine addresses.
C>Applying the addresses to the control ROM.

9.ACCUMULATOR :The memory locations ,where answers to airthmetic and logic operations are accumulated.

10.B REGISTER :An auxiliary register that stores the data to be added or subtracted from the accumulator.

11.LDA :Mnemonic for "LOAD THE ACCUMULATOR".

12.MACHINE CYCLE :All states generated by the ring counter.

13.MAR :Memory Address Register.

14.NOP :No Operation.

15.HAND-SHAKING :Interaction between a CPU and a peripheral device that takes place during an I/O operation.

****

D / A AND A / D CONVERSION

****

1.DIGITAL SIGNALs :Signals with discrete voltage levels.

2.ANALOG SIGNALs :Signals with continous voltage levls.
Note :-
** ABSOLUTE ANALOG SIGNAL IS IMPOSSIBLE TO GENERATE.
** THE SIGNAL WHICH IS ANALOG AT ONE SCALE ,WILL BECOME A DIGITAL SIGNAL AT A SMALLER SCALE.
"ADC" (ANALOG to DIGITAL CONVERTER) is often called "ENCODING DEVICE" since it is used to encode signals for entry into a digital system.

THE BASIC PROBLEM OF CONVERTING A DIGITAL SIGNAL INTO AN EQUIVALENT SIGNAL IS TO CHANGE THE "n" DIGITAL VOLTAGE LEVELS INTO ONE EQUIVALENT ANALOG VOLTAGE BY USING THE CONCEPT OF "EQUIVALENT BINARY WEIGHT".

3.MILLMAN'S THEOREM :"The voltage appearing at any node in a resistive network is equal to the summation of currents entering the node (found by assuming "ZERO" voltage at the NODE) divided by the summation of the "CONDUCTANCES" connected to the node."

$$V = \frac{V1/R1 + V2/R2 + V3/R3 + \ldots\ldots\ldots\ldots}{1/R1 + 1/R2 + 1/R3 + \ldots\ldots\ldots\ldots}$$

4."DAC"(DIGITAL TO ANALOG CONVERTER) :It can basically be formed by using two different types of circuits :-
1>RESISTIVE DIVIDER :It utilizes the "MILLMAN'S DIVIDING THEOREM" and the "VOLTAGE-DIVIDING" capability of two resistances in series.

CRITERIAS THAT CAN BE APPLIED TO RESISTIVE DIVIDER ARE :-
1>There must be one inut resistor for each digital bit.
2>Beginning with LSB ,each following resistor value is one-half of the previous resistor.
3>The full-scale output voltage is equal to the positive voltage of the digital input signal.
4>The LSB has a weight of [1 / (2**n - 1)].
5>The change in output voltage due to change in LSB is equal to

$$\frac{V}{[2^{n} - 1]} .$$

6>The output voltage "Va" can be found for any digital input signal by using the following modified form of "MILLMAN'S THEOREM" :-

$$Va = \frac{V0 * 2^{0} + V1 * 2^{1} + V2 * 2^{2} + ... + Vn\text{-}1 * 2^{(n-1)}}{(2^{n} - 1)} .$$

WHERE, V0,V1,V2,..,Vn-1 are the digital input voltage levels.

ITS TWO MAJOR DRAWBACKS ARE :-
A>As each resistance in the network has a preciese and different value ,they add extra costs in getting exact resistances of a number of different values.
B>The resistor used for the MSB is required to handle much greater current (abt. 500 times greater) than the current through the LSB resistor.

2>BINARY LADDER :It is a resistive network ,whose output voltage is a properly weighted sum of the digital inputs.It is constructed of resistors that have only two values(one double of another).
CHARACTERISTIC PROPERTY OF THIS BINARY LADDER IS THAT ,THE TOTAL RESISTANCE LOOKING FROM ANY NODE BACK TOWARD THE TERMINATING RESISTOR OR OUT TOWARD THE DIGITAL INPUT IS

"2R".

OUTPUT VOLTAGE AT THE "nth" "MSB" CAN BE CALCULATED BY THE FORMULA :-

Va = V / (2**n).

THE NET OUTPUT VOLTAGE IS GIVEN BY THE FORMULA :-

Va = [V/2 + V/4 + V/8 + V/16 + V/32 +...+ V/2**n].

5.UNITY GAIN AMPLIFIER :It is an unity-gain noninverting voltage amplifier (i.e. Vout = Vin) ,used for the holding purposes of data and instrutions for some time-interval ,so that the data path may get cleared ,till it arrives.
ALSO CALLED "OPERATIONAL AMPLIFIER (O.A.).

```
___________________
| |
| |\ |
|_____|-\ |
| \__________|_______ Vout = Vin .
| /
Vin ________|+/
|/
```

6.SAMPLE-AND-HOLD CIRCUIT :It is a combination of two "UNITY GAIN AMPLIFIERS" ,connected together with a CAPACITOR in the middle and a switch.When the switch is closed ,the capacitor charges to the D/A converter output voltage. When the switch is opened ,the capacitor holds the voltage level untill the next sampling time.

THE OPERATIONAL AMPLIFIER PROVIDES A LARGE INPUT IMPEDENCE SO AS NOT TO DISCHARGE THE CAAPACITOR APPRECIABLY AND AT SAME TIME OFFERS GAIN TO DRIVE EXTERNAL CIRCUITS.

USED IN MULTI-PLEXER CIRCUITS TO HOLD A DATA TILL THE PROCESSING OF ITS SUCCESSIVE DATA GETS COMPLETED.

```
| | | |
| |\ | |___|\ |
|_____|-\ | / |-\ |
| \__|____/ ________________| \______|________ Va . | / | | /
Vin ________|+/ | |+/
|/ --- |/
---
|
___|___
_____
___
-
```

7.THERE ARE TWO METHODS OF DECODING TWO OR MORE "DIGITAL SIGNALS" SIMULTANEOUSLY ,AS FOLLOWS :-
1>To use one D/A converter for each signal.
IT HAS THE ADVANTAGE THAT EACH SIGNAL TO BE DECODED IS HELD IN ITS REGISTER AND THE ANALOG OUTPUT VOLTAGE IS THEN HELD FIXED.

2>To use a "MULTIPLEXER" circuit along with a "SAMPLE-AND-HOLD" circuit and thus using only one D/A converter and switching its output.

8.TWO TESTS THAT CAN BE PERFORMED TO CHECK THE PROPER FUNCTIONING OF THE D/A CONVERTER ARE :-
1>STEADY-STATE-ACCURACY-TEST :It involves setting a known digital number in the input register ,measuring the analog output with an accurate meter ,and comparing it with the "THEORITICAL VALUE".

2>MONOTONOCITY TEST :It involves checking that the output voltage voltage increases regularly as the input signal increases.The output curve obtained in the "OSCILLOSCOPE" during this test must be a perfect "STAIR-CASE WAVEFORM".

It does not ensures "complete absence of error" but "absence of error greater than 1 LSB".

9.ACCURACY :It is a measure of how close the actual output voltage is to the theoretical output value.

10.RESOLUTION :It defines the smallest increment in voltage that can be determined by the "LSB".

11.DIFFERENT A/D CONVERTER METHODS :-
1>SIMULTANEOUS METHOD :It is based on the use of a number of "COMPARATOR CIRCUITs" (generally 3).

The analog signal to be digitalized serves as one of the inputs to each comparator.
The second input is a standard reference voltage (usually V/4 ,V/2 ,3V/4) making the system capable of accepting a analog signal voltage between "0 V" to "+V V".
If the analog input voltage exceeds the reference voltage to any comparator ,that comparator turns on.
" n "
"2 - 1 " comparators will be required to convert analog input signal to a digital signal that has "n" bits.

ALSO CALLED "FLASH CONVERTER" AS IT IS TOO "FAST".
Its disadvantage is that for 3,4 or higher number of levels in digital signal ,the number of "COMPARATORS" required will be very large "2**n -1".

2>COUNTER METHOD :It is composed of a D/A converter (the counter ,level amplefier and the binary ladder) ,one comparator ,a clock ,and the gate and control circuitry.
It is simpler to SIMULTANEOUS METHOD
BUT the CONVERSION TIME is longer.

AS THE COUNTER ALWAYS BEGINS WITH "ZERO" AND COUNTS THROUGH ITS NORMAL BINARY SEQUENCE ,AS MANY AS "2**n" COUNTS MAY BE NECESSARY BEFORE THE COUNT IS COMPLETE.

The "AVERAGE CONVERSION TIME" is "2**(n-1)" which is the time required to reach "ONE-HALF FULL SCALE" for a large number of conversions.

THIS TYPE OF CONVERTORS OPERATE AT MAXIMUM RATE ,IFF THE NEW "START" SIGNAL IS GENERATED IMMEDIATELY AFTER EACH CONVERSION IS COMPLETE.

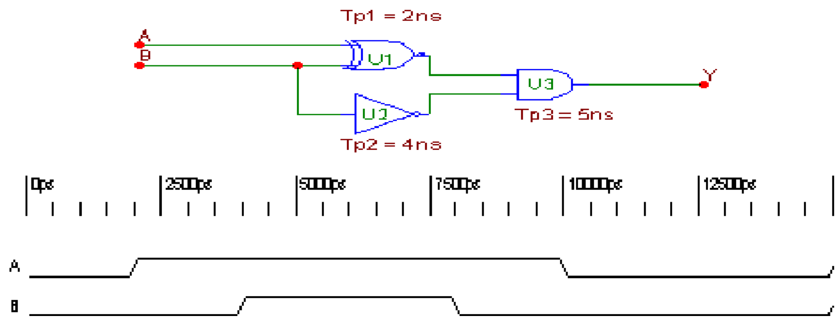
THE "CONVERSION TIME" FOR CONVERTORS ARE NOT CONSTANT BUT DEPEND UPON THE AMPLITUDE OF THE INPUTED SIGNAL.

3>CONTINUOUS METHOD :It is the method of using an "UP-DOWN" counter instead of normal counter ,so that we don't have to start counting from "ZERO" each time and thus number of net counts in the "COUNTER METHOD" is reduced a lot.
In this case ,the "UP" output will not go high unless the ladder voltage is more than "(1/2)LSB" below the analog voltage.
This is called the "CENTERING ON THE LSB". 4>SUCCESSIVE APPROXIMATION METHOD :Constructed on a single "MSI" chip which is also called "SUCCESSIVE APPROXIMATION REGISTRATION (SAR)"
It is a method of approximating the analog voltage by trying "1" bit-at-a-time ,beginning with "MSB".

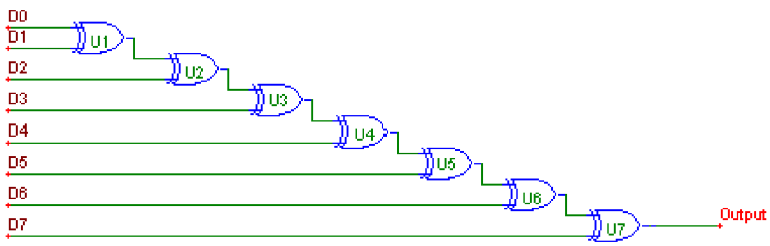
5>SECTION COUNTER :In this method the whole signal is distributed into a number of different sections.

. Sample Digital Questions Asked in Interviews

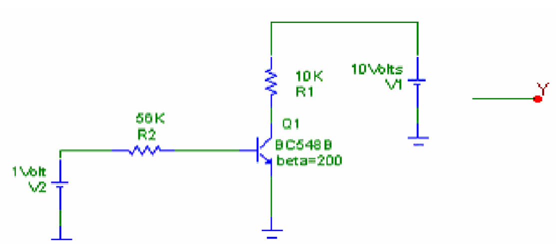
1. What is the output of AND gate in the circuit below, when A and B are as in waveform? Where, Tp is gate delay of respective gate



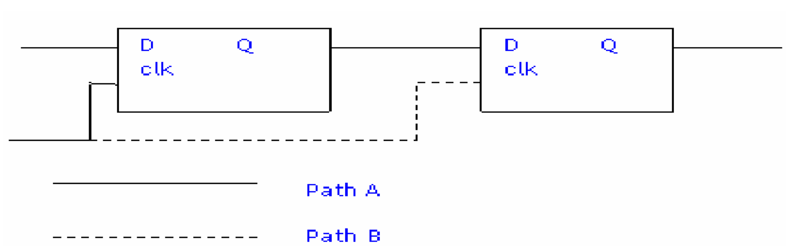
2. Identify the below circuit, and its limitation ?



3. What is the current though the resistor R1 (Ic) ?

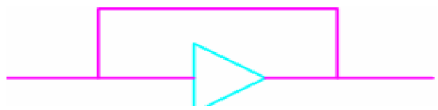


4. Referring to the diagram below, briefly explain what will happen if the propagation delay of the clock signal in path B is much too high compare to path A. How do we solve this problem if the propagation delay of path B can not be reduced ?

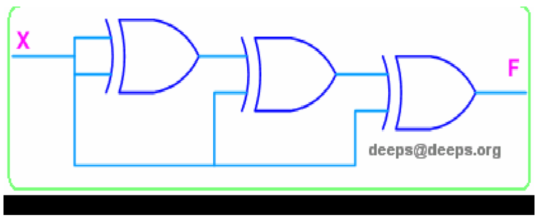


4. What is the function of a D flip-flop, whose inverted output is connected to its input ?
5. Design a circuit to divide input frequency by 2 ?
6. Design a divide-by-3 sequential circuit with 50% duty cycle.?
7. What are the different types of adder implementation ?

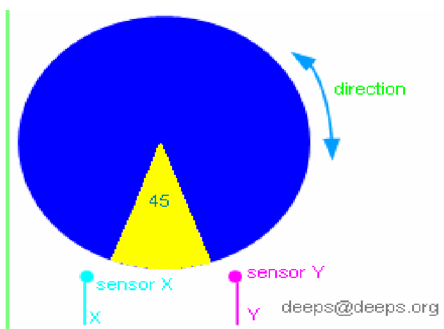
8. Draw a Transmission Gate-based D-Latch ?
9. Give the truth table for a Half Adder. Give a gate level implementation of the same
10. What is the purpose of the buffer in below circuit, is it necessary/redundant to have buffer ?



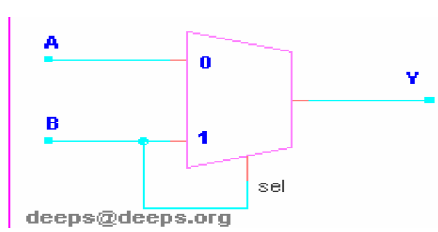
11. What is output of the below circuit, assuming that value of 'X' is not known?



12. Consider a circular disk as shown in figure below with two sensors mounted X, Y and blue shade painted on the disk for a angle of 45 degree. Design a circuit with minimum number of gates to detect the direction of rotation.



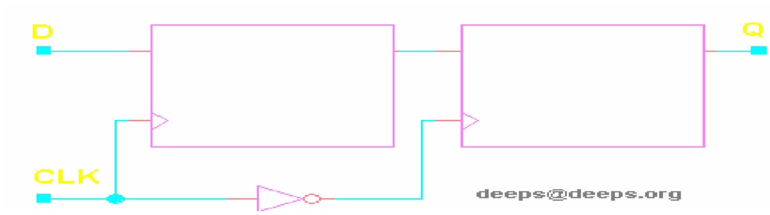
13. Design a OR gate from 2:1 MUX.



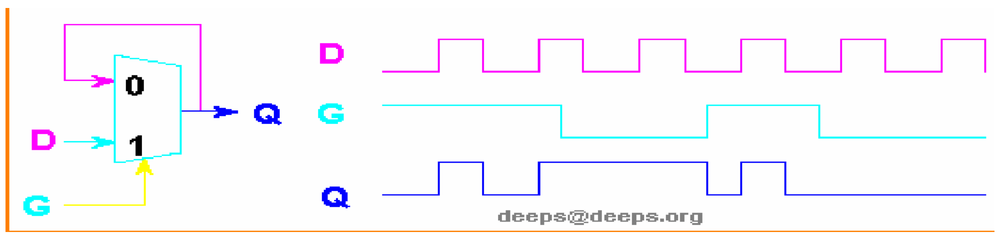
14. What is the difference between a LATCH and a FLIP-FLOP ?

Latch is a level sensitive device and flip-flop is edge sensitive device. Latch is sensitive to glitches on enable pin, where as flip-flop is immune to glitches. Latches take less gates (also less power) to implement then flip-flops. Latches are faster then flip-flops

15. Design a D Flip-Flop from two latches.



16. Design a 2 bit counter using D Flip-Flop.
17. What are the two types of delays in any digital system ? **Wire Delay and Gate Delay**
18. Design a Transparent Latch using a 2:1 Mux.



19. Design a 4:1 Mux using 2:1 Mux's and some combo logic
20. What is metastable state ? How does it occur ?
21. Design a FSM to detect sequence "101" in input sequence
22. Convert NAND gate into Inverter, in two different ways.
23. Design a D and T flip flop using 2:1 mux, use of other components not allowed, just the mux.
24. Design a divide by two counter using D-Latch.
25. Design D Latch from SR flip-flop
26. Define Clock Skew , Negative Clock Skew, Positive Clock Skew?
27. What is Race Condition?
28. Design a 4 bit Gray Counter ?
29. Design 4-bit Synchronous counter, Asynchronous counter ?
30. Design a 16 byte Asynchronous FIFO?
31. What is the difference between a EEPROM and FLASH ?
32. What is the difference between a NAND-based Flash and NOR-based Flash ?
33. You are given a 100 MHz clock , Design a 33.3 MHz clock with and without 50 % duty cycle?
34. Design a Read on Reset System ?
35. Which one is superior Asynchronous Reset or Synchronous Reset, Explain ?
36. Design a State machine for Traffic Control at a Four point Junction ?

QUESTIONS from Code Converters, Multiplexers, and Demultiplexers
Multiple Choice

1.

A magnitude comparator determines:

A B and if A B or A >> B
A B and if A > B or A < B
A = B and if A > B or A < B
A B and if A < B or A > B

2.

The expansion inputs to a comparator are used for expansion to a:

4-bit system
8-bit system
BCD system
counter system

3.

How many Exclusive-NOR gates would be required for an 8-bit comparator circuit?

4
6
8
10

4.

What do the mathematical symbols A<B and A>B mean?

A<B means A is greater than B. A>B means A is less than B.
A>B means A is less than B. A<B means A is greater than B.
A<B means A is less than B. A>B means A is greater than B.

5.

For the following conditions on a 7485 magnitude comparator, what will be the state of each of the three outputs?

A=B=0, A<B=0, A>B=1
A=B=0, A<B=1, A>B=0
A=B=1, A<B=0, A>B=0
A=B=0, A<B=0, A>B=0

6.

Which digital system translates coded characters into a more intelligible form?

encoder
display
counter
decoder

7.

How many inputs are required for a 1-of-10 BCD decoder?

4
8
10
1

8.

A principle regarding most IC decoders is that when the correct input is present, the related output will switch:

active-HIGH
to a high impedance
to an open
active-LOW

9.

A BCD decoder will have how many rows in its truth table?

10
9
8
3

10.

A truth table with output columns numbered 0-15 may be for which type decoder IC?

hexadecimal 1-of-16
dual octal outputs
binary-to-hexadecimal
hexadecimal-to-binary

11.

What control signals may be necessary to operate a 4-to-16 line decoder?

flasher circuit control signal
a LOW on all gate enable inputs
input from a hexadecimal counter

a HIGH on all gate enable circuits

12.

How many inputs are required for a 1-of-16 decoder?

2
4
8
16

13.

How many outputs are on a BCD decoder?

4
16
8
10

14.

How can the active condition (high or low) or the decoder output be determined from the logic symbol?

A bubble indicates active high.
A bubble indicates active low.
A square indicates active high.
A square indicates active low.

15.

What is the purpose of an enable input to a decoder?

to apply or disconnect Vcc
to apply or disconnect ground
to turn the circuit on or off

16.

How many possible outputs would a decoder have with a 6-bit binary input?

16
32
64
128

17.

A circuit that responds to a specific set of signals to produce a related digital signal output is called a/an:

BCD matrix
display driver
encoder
decoder

18.

When two or more inputs are active simultaneously, the process is called:

first-in, first-out processing
priority encoding
ripple blanking
both A and B

19.

How is an encoder different from a decoder?

The output of an encoder is a binary code for 1-of-N input.
The output of a decoder is a binary code for 1-of-N input.

20.

How many inputs will a decimal-to-BCD encoder have?

4
8
10
16

21.

If two inputs are active on a priority encoder, which will be coded on the output?

the higher value
the lower value
neither of the inputs
both of the inputs

22.

From the following list of input conditions, determine the state of the 5 output leads on a 74148 octal-to-binary encoder.

GS=L, A0=L, A1=L, A2=H, EO=H
GS=L, A0=H, A1=L, A2=L, EO=H
GS=L, A0=L, A1=H, A2=L, EO=H
GS=L, A0=H, A1=H, A2=L, EO=H

23.

One way to convert BCD to binary using the hardware approach is:

with MSI IC circuits
with a keyboard encoder
with an ALU
UART

24.

Which of the following is not a weighted value positional numbering system:

hexadecimal
binary-coded decimal
binary
octal

25.

In a BCD-to-seven-segment converter, why must a code converter be utilized?

to convert the 4-bit BCD into 7-bit code
to convert the 4-bit BCD into 10-bit code
to convert the 4-bit BCD into Gray code
No conversion is necessary.

26.

In a Gray code, each number is 3 greater than the binary representation of that number.

True
False

27.

A binary code that progresses such that only one bit changes between two successive codes is:

nine's-complement code
8421 code
excess-3 code
Gray code

28.

The primary use for Gray code is:

coded representation of a shaft's mechanical position
turning on/off software switches
to represent the correct ASCII code to indicate the angular position of a shaft on rotating machinery
to convert the angular position of a shaft on rotating machinery into hexadecimal code

----

29.

Which type of error was eliminated through the use of the Gray code?

decoding

timing
encoding
conversion

30.

Use the weighting factors to convert the following BCD numbers to binary.

0101 0011 0010 0110 1000

01010011 001001101000
11010100 100001100000
110101 100001100
101011 001100001

31.

How many 74184 BCD-to-binary converters would be required to convert two complete BCD digits to a binary number?

8
4
2
1

32.

Why is the Gray code more practical to use when coding the position of a rotating shaft?

All digits change between counts.
Two digits change between counts.
Only one digit changes between counts.

33.

One application of a digital multiplexer is to facilitate:

data generation
serial-to-parallel conversion
parity checking
data selector

34.

A basic multiplexer principle can be demonstrated through the use of a:

single-pole relay
DPDT switch
rotary switch
linear stepper

35.

One multiplexer can take the place of:

several SSI logic gates
combinational logic circuits
several Ex-NOR gates
both A and B

36.

How many select lines would be required for an 8-line to 1-line multiplexer?

2
3
4
8

37.

What is the function of an enable input on a multiplexer chip?

to apply Vcc
to connect ground
to active the entire chip
to active one half of the chip

38.

What is the status of the inputs S0, S1, and S2 of the 74151 eight-line multiplexer in order for the output Y to be a copy of input I5?

S0=0, S1=1, S2=0
S0=0, S1=0, S2=1
S0=1, S1=1, S2=0
S0=1, S1=0, S2=1

39.

The inputs/outputs of an analog multiplexer/demultiplexer are:

bidirectional
unidirectional
even parity
binary-coded decimal

40.

Most demultiplexers facilitate which type of conversion?

decimal-to-hexadecimal
single input, multiple outputs
AC to DC
odd parity to even parity

41.

Why is s demultiplexer called a data distributor?

The input will be distributed to one of the outputs.
One of the inputs will be selected for the output.
The output will be distributed to one of the inputs.

42.

Why can a CMOS IC be used as both a multiplexer and a demultiplexer?

It cannot be used as both.
CMOS uses bidirectional switches.

43.

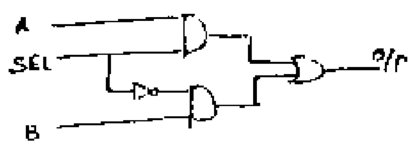
A microcontroller differs from a microprocessor in that it has several _____ ports and _____ built into its architecture, making it better suited for _____ applications.

communication, PROMS, control
parallel, logic gates, processing
input/output, memory, control

data, memory, decoding

[illegible] is selected [illegible] If the next i/p [illegible] according to the [illegible] pattern. When the pattern recognition is completed FOUND should go high and SEQ should go low and system should start searching for the pattern again. In the absence of pattern recognition FOUND should remain low.

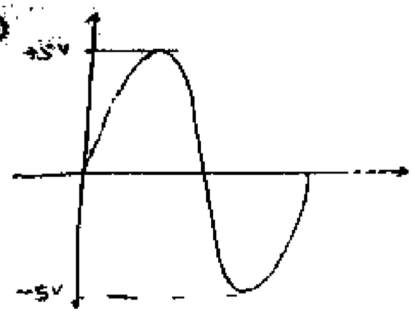
(2) This is the diagram of a multiplexer. All the gates have a propagation delay of 1.5 ns.

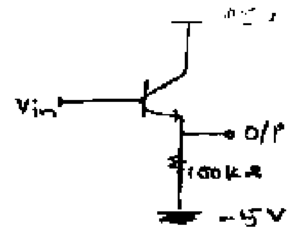


(1) Find the time and width of the glitch in the o/p.

(2) Suggest a scheme so that there is no glitch in the o/p while it is performing the same function.

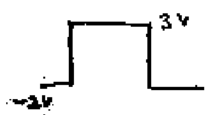
(3)

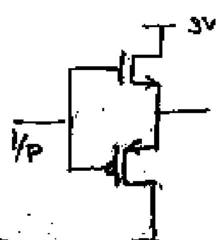




If above is the input to the circuit draw the o/p.

(4) If it is the i/p to the circuit, draw the o/p waveform considering $V_{tn} = V_{tp} = 1V$.

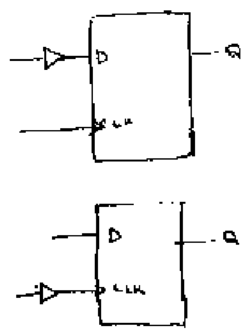




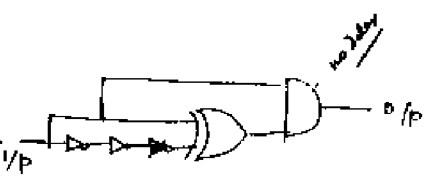
(7) fin and fout are the frequencies of i/p and o/p, design the combinational circuit so that

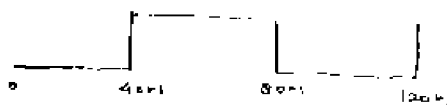
$$f_{out} = 2 f_{in}$$

(8) If the setup time of D flip flop is 0.5 ns and hold time is 0.3 and delay of buffer is 1 ns calculate the setup and hold time for both the circuits.



(9) If the delay time of all the gates is 10 ns draw the o/p waveform for the given i/p waveform



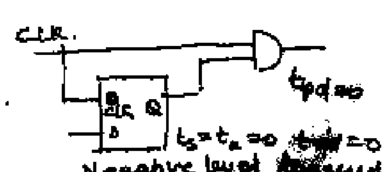


(10) a) Transparent latch using 2:1 MUX

b) Realize XOR using a MUX and an inverter

11) [illegible] problem — [illegible] calculation

12) What will be the o/p of the following ckt.



What will be [illegible] is replaced by [illegible]

waht are corner effects ?
Leakage current increases in the edges of the active area degrades MOSFET performance.The reason is that gate wrap-around of a sharp trench corner leads to the undesirable double-hump in the subthreshold IV characteristics,poor gate oxide integrity, and inverse narrow width effect
waht are end overlap rules?
what are there purposes?
what is metal strap programmability and via programmability? where each of technoques is used..

1. What is the gain of common emitter amplifier, asked me to derive from Ic, Ib equations.

Gain of a comon emitter amplifier is given by Vc/Vb. Given by
-Rc/(re + Re)
where.
Rc= resistance connected between supply and the collector of the transistor..
Re= resistance connected between ground and emitter..
re= resistance offered by the transisitor..
2.Gain of common source amplifier, but thats not all, the poles and zeroes. I was asked t oexplain intuitively why we get a zero because of the miller capacitance and stuff like that.
3. Common source with cascode amplifier , advantages , disadvantages, and also why dont we have miller effect here
4. given Id , vds curves for a mosfet , how do we design a power amplifier t oextrat maximum power out of a class A amplifier. basically i was asked to calculate the value of load resistance given Imax and VDD.
5.Intuitively explain why do we use inductor degeneration in Low noise amplifiers
6.I did a project using HLFF(hybrid latch flip flop) . was asked the detail operation of the flip flop and also what the maximum speed we can get from such style of FF
7.Opamp design, basically differences between telescopic and folded cascode opamp...
8. what kind of analog buffers do we use if we have very high Cload on a common source amplifier.
Basic Questions (usually asked in phone interview)

1.Tell me a little bit about semiconductors (what is conductance and valence band? Fermi level?
For n type semiconductor, what is the doping? Do you know how to say P and As in English?)
2.How does a pn junction works? (I know you know it, but could you tell other people clearly?
Try it!!!, They ask you this question!). What is the depletion region? What is the build-in potential?
What is the relation between these parameters with doping concentration and temperature?
Remember the tempo of the build in potential is about –2mV/K.
3.Tell me how MOSFET works. (Write it down in your own words and remember it !!!).
4.Tell me how BJT works. (Should I write down and remember it? Sure! But it is less asked than MOSFET). How does Vbe and Ic change with temperature?
5.Threshold voltage: If the substrate doping concentration increase, or temperature increases, how about Vt? it increase or decrease?
6.Tell me what is Channel length modulation, what is Early effects and their physical origin.
7.Tell me what is short Channel effect.
8.For a 0.18um and 0.8um technology MOSFET, which has a higher cutoff frequency?
9.How does a Bandgap Voltage reference work?
10.What is the ideal input and output resistance of a current source?
How about voltage source? How to improve these parameters?
(Cascode topology, use long channel transistors)
11.Tell me the parameters as many as possible you know that used to character an amplifier.

12.What are the two types of noise of MOSFET, how to eliminate them?(Thermal and Flicker).

ABOUT DIGITAL CIRCUITS:

Though you are applying for an analog circuits position, they often ask you one or two digital questions. To my experience, XOR (XNOR) circuits is very important. Sometimes they ask you to use NAND NOR gates to realize the XOR function, some times they even ask you to use 8 transistor to realize this function.
To my experience, you should read the whole textbook by Thomas A. DeMassa " Digital Integrated Circuits", including those parts that talk about Flip-Flop. They often ask
you a small question about State Machine or State Diagram. If you do not has the basic concepts of these concepts, read some books or take a course.
A very good company once asked me to use gates to realize a traffic light control problem.

Q: What are the factors to be considered for selecting the operating point?

A: While selecting the operating point the following factors are to be considered
Dc and AC loads at the output of the stages
The maximum transistor rating.
The available power supply.
The peak signal execution to be handled by the amplifier.
The tolerance distortion.
1) Design a full adder with 5 number of 2:1 Muxes.

2) design an AND gate with a single CMOS inverter(Just for theoritical purpose).

3) Design an EX_OR gate with two CMOS inverters(remeber it should work practically also).

4) what is the difference between simulation, verification and testing.

5)what is the main use of LATCHES in Digital circuits, though we r having Flip-Flops.

6)What is need of Programmable BIST in todays IC design.

```
7)always@(posedge clk or posedge reset)
  &nbspif(reset)
    &nbspq<=d1;
    else
    &nbspq<=d2;
```

How this always block will be synthesized.

```
8) always@(posedge clk)
  &nbspif(reset)
    &nbspq<=d1;
    else
    &nbspq<=d2;
```

How this always block will be synthesized.

What is the best way to count the number of 1's in an 8-bit register? Cannot add bit by bit.
How do I write verilog rtl to generate a signal when there are 5 '1' in an 8 bit register. For example, output = 1 when this 8 bit register equals 8'b000_11111 or 8'b11111_000 or 8'b10101011... etc.
1)you are given a Push button and a 4 leds , each time the pushbutton is clicked the color of the LED muct change from yellow to blue then to Red and finally green in sequence. Design the circuit.

Ans : use sate machine

2) Suppose instad of changing from yello to blue the signal changes to red ...what is the reason?

ans: glitch in the signal.

3) How do you divide a clk?
-- T ff

4) Design a OR gate using 2x1 MUX.
-- use 1 2x1 mux

5) you have 2 bulbs 100W each connected in parallel what is the total power dissipation?
how much will the power consumption vary if they are connected in series?

- 4 times

how to design static RAM with 4 input AND gate

give me detail information on design of DPLL(Digital Phase Locked Loop)?

fingering done to cmostransistors

why fingering done to cmostransistors

what is silicidation in cmos

standared cell design concepts

what is stick diagram, explain with one example?

The main reason why do we need fingering is because it reduces the side wall capacitances and also the junction wall capacitance.

Is there any way that we control Hold time of a register?
what are the challenges in implementing a FIFO contoller?

Xilinx Interview Questions
1.How do you implement DCM ??
2.What is the feature of Virtex 4?
3.What are the contraints?
4.What is Sopc Builder?
5.How do u implement the GCLK when ther is lack of Source?

1. (from my project)
list some of the routing algorithms u studied.

2. what is noise margin?

ans1:
we take a simple example of inverter, from the rise to fall , we draw a tangent, from the tangent we deduce tph and tpl. the diffrence is called noise margin.(see any digital book)

ans2:
the max noise which circuit can tolerate.

3. what is RTL?

we take an example from VHDL

a <= b;

when we synthesise a will be asigned as register and b will be assigned as register.
so there s register to register transfer done.this level of abstract model is called RTL.

4. which method of coding u prefer?

depends on reqirement

for synthesise it is better to code n lower abstaction.

5. which circuit family has high noise margin?

CMOS

6. why?

as n CMOS when pull up conducts, pull down does not conduct ,so there s high noise margin.(because it is static circuit)

7. why CMOS takes low power?(static power)
same as previous ans,

8. tell me about ASIC design flow

9. what is procedural and continuous assignment(verilog question)

10 . where we have to use blocking and non blocking?
(i don't know)

11. what is diffrence between RAM and FIFO?

FIFO does not have address lines

12. draw state diagram of small sequence detector and code it in verilog or vhdl

13. tell me about steps involved in chip manufacturing

what is diffrence between RAM and FIFO?

ram is used for storage purpose where as fifo is used for synchronization purpose i.e. when two peripherals are working in different clock domains then we will go for fifo.

1. lee maze routing algorithm.
7. CMOS has very less static leakag power.
9. continous assigment:
  the statements outside of always statement
  like eg:-

```
assign a=b;
assign d=s;
```

procedural statement:
the statemnts inside aways statement
always@(clk)
begin
a=b; -- procedural assignment
10.
guidelines from cliff ford cummins

guideline 1: USE BLOCKING FOR COMBINATIONAL CIRCUIT.

Guideline 2:USE NONBLOCKING FOR SEQUENTIAL CIRCUIT.

design FSM so that when ip X=0 modulo-3 counter , when x=1 modulo 4 counter.

CG-Coreel Interview Questions
1. Why we are using GREY code in K-maps.
2. Design T-Flip flop using D-Flipflop.
3 Design 3-bit synchrnous counter using D flip flop and write a code in VHDL

1)Design a circuit which does 2's complement of a number with out using memory element?

2)Vhdl is Concurrent language, but your system is serial, how do u use this concurrent language in a serial system?

3)I have a bit file, but i dont have FPGA to download it, how do you check it works on FPGA?

1. what ciruit u you used in u r project.

asn: mealy

2. why?

ans: mealy has less number of states.

3. draw digram of inverter.

4. why PMOS has higher width?

ans: in PMOS holes are main carrier ,holes speed is around 700,compared to 1000 of electrons,
so PMOS is bigger.

5. if there r 6000 bytes address spaces of RAM, how will u interface? how many address lines u need to uniquely link them?

6. draw the D FF using pass transistors.

7. write a small program to change column to row.(C language)

8. if 'case' is good for synthesis and 'if' is bad for synthesis then if u have 4:1 MUx and 2:1 Mux n standard cell .then how will they synthesise.

10.in microprocessor write a command to move a byte of data from accumulator to a register.

11.(VHDL question) what s use of package decalartion?

it can be used to declare all the typed variables in a single block, then it can be called easily or used with another entity so that it can be used in another architecture with same entity.(some what confusing) see the bhasker book.

12.what s bus contention?

wen two signals try to gain control of bus then thsi condition is know as bus contention.

contention-fighting

13. what s SLEW rate?

the metals lines running along can pick up with adjacent metalines dielectric which is cause of capacitance leads to slowing in rise time of a signal.this is slew rate
texas interview question
here v have ring osclr which is made up of three inverters,the third inv o/p is connected back to firstinv i/p.(osclr)and they given tphl=2ns,tplh=1ns.and asked for duty cycle for the output waveform?

1. what s difference b/w blocking and non blocking assignments?
where will u use it?

2. what s racing condition?

3. if u replace latch enable signal by clock wat will be the difference?

4. how latch takes less power and ff takes more power?

5. what's diffrence b/w mealy and moore ckt?

6. how to overcome metastability?

7. if u have 'case' stmt and 'if' stmt, and 'case' s good for synthesis then wat s advantage of 'if' stmt?

8. how will u write d ff using variables alone?

9. design a some gates using mux

10. draw a state diagram for sequence detector.

11. how to overcome racing condition?

12. for a small example draw simlation time for non blocking assignment, blocking assignment.

13. some questins on interfacing.
1. what s difference b/w blocking and non blocking assignments?
where will u use it?

ans: 1.blocking stmt
 these stmt blocks the other stmt from being executed.ie first these stmt gets executed before it lets other stmt execute.
 2.Non blocking stmt
 these stmt executes all parallely.withe designated delayes.

2. what s racing condition?

when s=1 and r=1, then both the signals fight against each other to determine the output.if this occurs output of the FF is not determined.

how to overcome?

1. we can reduce the noise.
2. (i think we can put master-slave FF)

3. if u replace latch enable signal by clock wat will be the difference?
(i think if we put clock ,then power consumption will be more)

4. how latch takes less power and ff takes more power?
(clock routing may take power)

5. what's diffrence b/w mealy and moore ckt?

mealy
it has less number of states.

it more prone to noise.
moore
it has more number of states.
it is less prone to noise.

6. how to overcome metastability?
by adding another FF

7. if u have 'case' stmt and 'if' stmt, and if 'case' s good for synthesis then wat s advantage of 'if' stmt?

inside case stmt expression cannot be used like

case (x=x*5+y-10)

(x): z=5;
(y): f=10;
end case;

8. how will u write d ff using variables alone?(VHDL question)
(i don't know)
but i think using variable and signals we can write D FF

9. design a some gates using mux
refer any digital book
10. draw a state diagram for sequence detector.
refer any digital book

12. for a small example draw simlation time for non blocking assignment, blocking assignment.
13. some questions on interfacing.

14. draw a simple circuit of D FF using pass gate transistor
 refer any CMOS book

15. which universal gate do u prefer(NAND or NOR)?
 ans: NAND gate
16. why?
 ans: it takes less power,....(i think less arear)

17. draw the schematic of NAND gate.
 see the book.

18. what determines drive strength of a gate?
 width of the gate determines drive strength of the gate.
19. what s clock skew?
 when clock s routed through the chip it is subjected to parasitic capacitance,so the current is absorbed by the circuit, so clock is delayed by some amount.which is called clock skew.
skew-- means delay
eqn:- r1+r1*c1+r1(c1+c2)+... like that (see any CMOS book)

20. how to reduce the frequency by half?(very very important question!!!)

 ans i know:
 just put a T FF.
if u put a T FF u reduce the freq by half.
if u put 2 power n FF, u reduce the frequency by n times.
 they may ask u to code it in Verilog or VHDL.

21. how to double the frequency by two times.

Q1)
1) Consider a NMOS where drain is connected to Vdd and output is taken from source(Vo1). Input applied to gate.
2) Consider a PMOS where source is connected to Vdd and output is taken from drain(Vo2). Input applied to gate.

if Vdd=5v, tell me what are Vo1 and Vo2 when Vin is 5V, 3V, 2.5V and 0V(calculate for each transistor)

Q2) Consider 2 NMOS transistors stacked one on another as Q1(bottom transistor, Vb1 is the gate input) and Q2(top transistor, Vb2 is the gate input). The drain of Q2 is connected to Vdd and the threshold voltage of the transistor is 0.7V. Vb1=1v, Vb2=2v, When Vdd change from 5V to 0V, draw the current flow through the transistors VS Vdd.

2 find out when the transistors switches from stauration to linear region. Once you do that all you have to do is draw a V-I characterstic curve for the transistors.

how to design a MUX based D-Flip Flop. whats the logic behind it? and also can tell me what a Super buffer is???any help would really be great..

give feedback to select 0's input and the other to D and give clock to select.if u connect the ckt in master slave type appropriately it will result in dff else d latch.super buffer has good buffer characteristics with less delay and less loading and other paramaeters i donno to a very good extent abt that.u can find that on many books.bye

1.Output of an inverter is connected to a bidirectional cmos transmission gate (driven by EN at nmos gate and its complement at pmos gate). Construct a truth table:

I/p en o/p
0 0 ?
0 1 1
1 0 ?
1 1 1
plz explain as well!
2.
In an inverter the pmos in connected to Vdd through another pmos with gate and drain tied.
The nmos is connected to gnd through another nmos with gate and drain tied.
Vtp=-1.1v
Vtn=1 v
What are the output voltage levels if a clock signal is fed to the inverter that zero corresponds to zero Volts and 1 corresponding to 5 V?

conditions r like there,
if G0<=100KG that means if weight in lift is less than equal to 100kg than gear G0 will work.
if G1<=200kg than G1 work
same way,
G3<=300kg and G4<=400kg
now if weight is more than 400kg lift will stop
can anybody design

how canu make a positve logic inverter using pnp transistor

IT IS IMPOSIBLE TO MAKE POSITIVE LOGIC WITH PNP TRANSISTER, TELL INTERVIEWER DIRECTLY.
BUT IF THEY TELL IT IS POSIBLE THEN TELL IMPOSIBLE

ONE WAY IT IS POSIBLE,(IF WE USE ANOTHER COMPONET OTHERWISE NOT..)

APPLY INPUT VOLGATE INTO FIST NPN TRANSISTER AND IF VOLAGE IS POSITIVE THEN O/P IS REDUSED TO ZERO(COMMON EMITER) THEN APPLED THIS O/P TO PNP TRANSISTER SO O/P IS POSITIVE...AND VICE-

MEANS POSITIVE VOLTAGE O/P POSITIVE
NEGETIVE VOLTAGE O/P NEGETIVE.........

1.Implement 3x8 decoder using 1x2 decoders..

2. Design a clock that will half the input frequency using only combinational logic ckt...

3.Implement 4 input OR gate using 2X1 mux...

4.Draw logic ckt for 4-bit Binary to Reflected code....

5.F=ab+cd+ef implement using 2 input NAND gates only........

6.Prove that 2x1 mux work as Universal logic gate..

7.Design a ckt that will devide a clock 3/2 with 50% duty cycle..

8. write a code in verilog ,,, T=40nsec and DC=25%
 Generate a clock....

discuss about differerent coverages in verification

1. Code Coverage
2. Functional Coverage.

Code Coverage:
It will show which part of the DUT Code has been coverage by the Testbench(es).

Code Coverage has following Metrics
Line Coverage -> Lines covered by the Testbench(es),
Conditional Coverage --> Conditions Covered ...
FSM Coverage --> Reports states of FSM covered ...
also there is toggle Coverage which shows if the Registers, nets have toggled or not. etc

Disadvantages of Line Coverage: It will only show which part of the DUT Code has not been covered. If a particular Functionalty itself is not coded in DUT it will not be able to catch it.

The support for monitoring Code Coverage is there in Simulators.

2. Functional Coverage:

It monitors if a particular functionality has been generated by the Testbench(es).
The Verification Engineer has to Code up a Coverage Model in the TestBench.
Here the Code in most of the Cases the DUT Code is not used for calculating Coverage.

It is very useful along with Constrained Random Testing wherin data, Test Vectors or Scenarios are randomly generated and Functional Coverage helps in knowing whether the randomly generated Test has Covered the Required Functionality or not!

Supported in Hardware Verification Languages like OpenVera, E and now in SVTB too.
A INPUT TO AND GATE
B INPUT TO AND GATE

O/P OF BOTH WITH AND = F
B INPUT TO XOR GATE
C INPUT TO XOR GATE

HOW THIS CIRCUIT WORK
1.integrator
2.diferentiator
3.Antilog dfferentiator
4.Log differentiator

1) Implement Y1=ax+by; and Y2=ay+cx using only 2 multipliers. You can any number of adder, subtractor, MUX etc..,

2) They had given a D-FF with q(bar) connected to D input. What is the maximum frequency at which it can operate. (Is it fmax = 1/(Tc-q+Tcombo+Tsu ????)

3)Identify the sequence 01110 from the incoming serial pattern. Take care of overlapping sequences. Draw state diagram for this.

4) Implement 4 input AND gate using multiple 2 input AND gate atleast by two methods and give the advantages and disadvantages of one over the other.

5) For a 4-bit ripple counter, what is the maximum operating frequency. (the setup time, hold time and FF delay is given)

6) What is the maximum operating frequency for the following 2 cases:
i) Output of first D-FF goes as the input to the second FF through a combo logic. The clock is given directly to the first FF but through a NOT gate to the second FF.
ii) Same as (i) but clk is fed directly to FF2 but through an inverter to the FF1.

7) How do you design a 3-bit comparator (A>B) if A and B are signed numbers.

Above ans
1. can you tell me the widths of a,b and x,y. These are bit arrays or bits. Multiplication of vectors is nothing but shift and add na.
2. U are right. One more thing is Tc-q+Tcombo+Tsu > Thold also.
3. Its a very generic detector. You can make out if we have some basic design.
4. One way is ((AB)(CD)) and the other is (((AB)C)D). In first one the path delay on each i/p to o/p is the same, but its not in second one.
5. In that ripple counter U can look for the critical path which wil gives you the max. freq of operation.
6. In this case the clk freq is more than the second case.
In first case Tclk = Tcq+Tcombo+Tsu - Tinv
In second case Tclk = Tcq+Tcombo+Tsu + Tinv
7. It is there in Digital desing Morris mano Black pad book. We can do this by simple combo logic. Using AB, ~AB and A+B gates only.

1.there is a series combination of : -- pmos---pmos--- pmos----nmos
leftmost pmos is connected to Vdd nmos to gnd and output is taken from between 2nd and 3rd pmos. When an input combination is such that all devices are turned on will the output be pulled up or pulled down?

2.How do u size nmos and pmos transistors to increase the threshold voltage?

3.What happens to delay if add a resistance to the output of a cmos ckt?

4.There r three adjacent parallel metal lines. Two out of phase signals pass through the outer two lines what are the waveforms in the center line. What will be the waveform if signal in outer lines are in phase with each other?

5.What happens if we increase the no of contacts or via from one metal layer to the next?

6.In the design of a large inverter y do we prefer to connect small transistors in parallel(thus increasing effective width) rather than make one transistor withlarge width?

7.Suppose u have a combinational ckt between two flip flops(registers)?
What will happen if delay of combinational ckt is greater than clock signal? How will u correct it? (u cant resize combinational ckt transistors)

Ans
1.0001 and op depends on the sizes or simply depends on the resistances of mosfets.cant comment without that.

2.Vm=(Vdd-modulus(Vtp)+(sqrt(N/P))Vtn)/(1+sqrt(N/P))

N=Kn'w/l similarly for P

3.delay increases due to RC

4.
5.diffusion capacitance increases and resistance too

6.use of fingers facilitates in sharing the drain or source of the parallel trans. so diffusion capacitance is reduced hugely

7. one way is to skew the clock.so that setup time is met.hold time is met due to the delay of combi.skew shd be appropriate so that both violations are averted.

How do we design a divide by 3 cicuit with 50% duty cycle.?

Design a divide by 5 sequential ckt with 50% duty cycle?

·How does the size of pmos pull up transistors affect the SRAM's performance?

·In sram layout which metal layers would you prefer for word lines and bit lines?

·What is the differnce between testing and verification?

·Why is the transconductance of a BJT greater than that of a mosfet?

What technology do we use in cmos… NAND OR NOR? WHY?

·What technology do we use in nmos nand or nor? Why?

An Opamp has it's noninverting i/p grounded.
and input is applied to inverting i/p.

Also o/p is fedback to the input i.e: invering input directly without any element.(direct connection between inv i/p and o/p.

So, o/p is directly connected to input and also OPAMP is driven at it's inv i/p with input....

IT acts as __________________????????

he circuit will act as unity gain buffer.because it has low output resistance,it is generally used with the circuits which drive large capacitive loads.

(1) In a SRAM circuit, how do you design the precharge and how do you size it?
(2) In a PLL, what elements(like XOR gates or Flipflops) can be used to design the phase detector?
(3) While synthesis of a design using synopsys design compiler, why do you specify input and output delays?
(4) What difference do you see in the timing reports for a propogated clock and an ideal clock?
(5) What is timeborrowing related to Static timing anaylsis in Primetime?

What is charge sharing between bus and memory element??

What are the effects of making PMOS stronger than NMOS (by increasing W/L ratio of PMOS) and vice-versa in CMOS Inverter?

How does the circuit behaves for 0->1 and 1->0 transition in each case?

How is the Transistor threshold voltage is affected by Process Parameters like Oxide thickness, diffusion depth and impurity concentrartion density (substrate doping density)?? Explain in detail by semiconductor physics.

how DFF written in VHDL/Verilog is implemented in FPGA internally???

One way of implementing it, is the FPGA architecture has block called CLB (Configurable logic block). This is the block that holds your logic in the FPGA. It can either have multiplexer based logic or something called as Look Up table (LUT). The LUT is basically a small memory element that holds the truth table for the logic. So the inputs to the LUT are the primary inputs to the logic. The output of the LUT is then routed to the final output if the FPGA chip

Is skew an advantage or disadvantage?? How the skew is related with Setup Time and Hold Time??

For a circuit Source Register ---- Combinational Logic ---- Destination register, Clock skew (clock at destination arrives later than source register) affects the circuit as:
1. It increases setup time at the destination register
2. It decreases the hold time at the destination register.

So, if the combinational logic is very small, then clock skew is a disadvantage.

If the combinational logic is very large, then the clock skew can work to an advantage. In large combo blocks like multipliers / ALUS, clock is skewed at destination clock to allow the huge combo logic to be executed and still meet setup w.r.t destination register.

To infer a FlipFlop using Verilog coding style we use,...
always@(posedge clk)
,....,statemetns
This always block infers flop.
So, to have an asynchronous flop, we have to write,...,
always@(posedge clk or negedge reset)
,...,statements follows.
'negedge' reset is for the codition not to mix edge and level sensitives in the sensitivity list.
So, for this how does the Hardware look like,...Does it infer 2 latches or 1 latch only. If it is only one latch, How does the h/w recognizes the edge of the reset.

1.Implement 2:1 AND GATE function using 2:1 Mux

2.Implement 2:1 AND GATE using m inputs AND Gate
(Tricky and simple ?)

3.What is the value of Last digit of the factorial 36.

Write verilog code to detect a 64bit pattern if the pattern can be expressed in power of 2.

Example expected pattern,
64'd1;, can be expressed as 2power'0'
64'd2; 2power1;
....
64'd128; 2power7;
and so on..

```
module pat_det (
  &nbspdata_in,
  &nbsppatDetected
 );

  &nbspinput [31:0] data_in;
  &nbspoutput patDetected;

&nbspwire [4:0] patSum = data_in[0] + data_in[1] + data_in[2] +
                   data_in[3] + data_in[4] + data_in[5] +
                   data_in[6] + data_in[7] + data_in[8] +
                   data_in[9] + data_in[10] + data_in[11] +
                   data_in[12] + data_in[13] + data_in[14] +
                   data_in[15] + data_in[16] + data_in[17] +
                   data_in[18] + data_in[19] + data_in[20] +
                   data_in[20] + data_in[21] + data_in[22] +
                   data_in[23] + data_in[24] + data_in[25] +
                   data_in[26] + data_in[27] + data_in[28] +
                   data_in[29] + data_in[30] + data_in[31] ;
wire patDetected = (patSum <= 1)? 1'b1: 1'b0;

endmodule
```

1> implement 7 bit unsigned adder using 4 bit 2's compliment adder (as many as u like) ?

2> circuit for positive ecge and neg edge detection ?

3> at every clock a 32 bit data is coming , u have to design a crcuit for the addition of all even numbers coming from outside ?

4> hardware ckt for fibinoccci sequence genaration ?

5> hardware ckt for factorial of a given number ?

1) What r all the steps in ASIC flow backend?

2)what is need for VLSI technology in backend?

3)what is the difference between floor planning and place ment?

4) how does powerplanning of a chip depends?

5)what is difference between the IO pads and IO pins?

6) what is difference between the Routing and prerouting?
Design a 2 input AND gate using 2 input XOR gate?

a) difference between sdram and ddram.
SD Ram is actually SDR SDRAM, and DD Ram is actually DDR SDRAM, though the former does not usually have the SDR in front of it. The letters SDR and DDR stand for Single Data Rate and Double Data Rate, and the letters SDRAM stand for Synchronous Dynamic Random Access Memory. The main difference between SDR and DDR memory is speed. There are a lot of little differences, but the main one (IMHO) that affects the user is speed: DDR can transfer data at roughly twice the speed of SDR. More speedy data rates = better performance. Just remember, the motherboard you are using must include the appropriate chipset to support the different RAM types. They are not interchangeable. SDR SDRAM comes in three main flavors: PC66, PC100 and PC133. Each successive number refers to the bus speed of the RAM in MHz, thus PC66 runs at 66 MHz, PC100 runs at 100 MHz, etc. SDR SDRAM has 168 pins at the connector. DDR SDRAM has 184 pins at the connector, which is one reason you can't just use DDR instead of SDR, and comes in many different flavors; PC2100 which runs at 266 MHz, PC2700 which runs at 333 MHz, PC3200 which runs at 400 MHz, etc. In order to know which type of RAM you need, you must know what your motherboard supports.
b) difference between data processors and dsp
A:
In data processing or information processing, a Data Processor or Data Processing Unit or Data Processing System is a system which processes data which has been captured and encoded in a format recognizable by the data processing system or has been created and stored by another unit of an information processing system.

On the other hand DSP is a specialized digital microprocessor used to efficiently and rapidly perform calculations on digitized signals that were originally analog in form (eg voice). The big advantage of DSP lies in the programmability of the processor, allowing parameters to be easily changed.
c) tell about harvard architecture
The term Harvard architecture originally referred to computer architectures that used physically separate storage and signal pathways for their instructions and data (in contrast to the von Neumann architecture). The term originated from the Harvard Mark I relay-based computer, which stored instructions on punched tape (24-bits wide) and data in relay latches (23-digits wide). These early machines had very limited data storage, entirely contained within the data processing unit, and provided no access to the instruction storage as data (making loading, modifying, etc. of programs entirely an offline process).

How to use a NPN transistor to become an logic inverter?

1) According to Clein, what has been one of the main reasons why CAD tools have failed to be successful among IC layout engineers?

2) With respect to CAD tools, what are some of the advantanges and disadvantages to being a small IC design house?

3) What is an IC design flow? Why do IC design teams operate within the constraints of design flows?

4) Why are PMOS transistor networks generally used to produce high (i.e. 1) signals, while NMOS networks are used to product low (0) signals?

5) On IC schematics, transistors are usually labeled with one, or sometimes two numbers. What do each of those numbers mean?

6) Why is the number of gate inputs to CMOS gates (e.g. NAND or NOR gates) usually limited to four?

7) What is meant by static and dynamic power with respect to the operation of a CMOS gate? Why do CMOS gates dissipate close to zero static power? Why is the static power not exactly zero?

What is a transmission gate, and what is it used for typically? Why are transmission gates made with both PMOS and NMOS transistors?

9) What are the major factors that determine the speed that a logic signal propagates from the input of one gate to the input of the next driven gate in the signal's path?

10) What are some of the major techniques that are usually considered when one wants to speed up the propagation speed of a signal?

11) What is the difference between a mask layer and a drawn layer in an IC layout? Why do layout designers usually only specify drawn layers?

12) In an IC layout, what is a polygon and what is a path? What are the advantages and disadvantages of each?

13) What is the difference between a contact and a via? What is a "stacked" via process?

14) Why is it that NMOS transistors can be created directly in a P-type substrate, whereas PMOS transistors must be created in an N-type well?

15) Why must transistors be provided with "bulk" connections? What voltage levels are connected to a p-type substrate and an n-type well through these connections, and why?

16) What are process design rules? What is their major purpose? How are design rules created?

17) What are width rules, space rules, and overlap rules?

1 What is a "vertical connection diagram"? What is it used for?

19) The routing strategies for the power grid and global signals are usually defined at the start of planning a new chip floorplan. Why?

20) What are the major advantages of hierarchical IC design?

21) Define what is meant by the terms design rules checking, layout versus schematic, and electrical rules check? Are all three procedures required in every chip design?

22) What is meant by the term "porosity"? Why is it desirable for a cell or macro to have high porosity?

23) What are the main differences in priorities between microprocessor design, ASIC design, and memory design? How are those differences reflected in the corresponding design flows?

24) What is an "application-specific memory", according to Clein? What are some specific examples of this part type?

25) What is the difference between a soft IP block (soft core) and a hard IP block (hard core)?

26) In ASIC design, what are the main advantages of expressing the design using a hardware description language, such as VHDL or Verilog?

27) Why are memory layouts designed primarily from the bottom up, instead of from the top down, like other ICs?

2 With respect to a memory layout, what is meant by "array efficiency"?

29) What is "pitch-limited layout"? What are some of the major circuits in a memory layout that must meet pitch-limited constraints?

30) What are some of the typical kinds of cells that one would expect to

find in a library of standard cells?

31) The layout of standard cells is constrained to simplify the job of place & route tools. Give several examples of these constraints.

32) Why did older cell libraries include so-called feedthrough cells? Why are such cells no longer required in cell libraries for modern processes?

33) What is electromigration? How does electromigration affect the design of a standard cell based design?

34) What is a gate array? Why are main advantages of using gate arrays to implement an IC? What are some of the main disadvantages, with respect to custom design or standard cell based design?

35) Why might one want to use some gate array based design inside an otherwise custom IC design, according to Clein's experience?

36) What are some of the major similarities and differences of standard cells and datapath cells?

37) How is the problem of driving a clock node different from that of designing a regular signal node? What are the key goals when laying out a clock node?

3 What is a "pad frame"? What are "staggered" pads?

39) Why are 90 degree corners usually avoided in the layout of pad cells?

40) In the layout of output pad driver transistors, why is the gate length often lengthened at both ends of the gate?

41) Why is the pad ring provided with power supply connections that are separate from those of the core design?

42) What are so-called friendly cells in a DRAM core design? Why and where these cells included in a memory design?

43) Why are metal straps used along with polysilicon wordlines in memory designs?

44) Why are wordline driver circuits very long and narrow?

45) Describe some of the alignment keys that are included in IC layouts.

46) Why is the power supply interconnect layout layout planned out before other elements? Similarly, why are busses, differential signals, and shielded signals routed before other general signals?

47) What are the root and resistance styles of power supply layout?

4 What are some of the main reasons why clock skew minimization is such a major design challenge?

49) What are the major advantages and disadvantages of using a single clock tree conductor driven by one big buffer?

50) In ASIC design flows, why are clock trees inserted after the logic cells have been placed? In such clock trees, how is clock skew minimized at the leaves of the tree?

51) What is a routing channel? Why are routing channels used in IC layouts?

52) Why is the estimated area for routing channels increased by 10% during early stages of layout planning?

53) When routing a signal interconnect, why is it desirable to minimize layer changes through vias?

54) Interconnect resistance is usually minimized in IC layouts. Give at least four situations where a deliberably large, but controlled, resistance is usually required?

55) Why should minimum-width paths be avoided in the design of deliberate resistances?

56) Usually one wishes to minimize the capacitance of electrical nodes in an IC design. Give four examples of circuits where one would wish a larger, but controlled, capacitance at a node?

57) The capacitance on a node is the sum of several components. What is meant by fringe capacitance? How does reducing the width of a conductor affect the fringe capacitance?

5 How can the parasitic capacitance between two signal nodes possibly cause the signal transition on one of the nodes to be unexpectedly sped up?

59) How can a layout designer help ensure that the propagation delay along two conductors is very similar?

60) List four situations where it may be desirable to have 45 degree corners in the interconnect.

61) Explain what is meant by electromigration. What are some possible

consequences of unexpectedly high electromigration? How is electromigration controlled in IC layout design?

62) Why are wide metal conductors, such as those in the power rings, provided with slits? What constraints must be followed when positioning these slits?

63) When placing multiple vias to connect two metal conductors, why is it better to space the vias far apart from each other?

64) Why would a DRAM layout be verified against two or more different sets of design rules?

65) What is the antenna effect, and how can it cause problems in an IC design? What are two layout techniques that can be used to reduce vulnerability to the antenna effect?

66) What is the purpose of minimum area design rules?

67) What is the purpose of end overlap rules?

6 What is the phenomenon of latch-up? Why is it a serious concern in CMOS layout design?

69) Describe six different layout strategies that are commonly used to minimize the possibility of latch-up.

70) Why is it wise to plan designs to make it easier to change details later?

71) What is meant by metal strap programmability and via programmability? Give one example where each techniques is commonly used.

72) What is the difference between test pads and probe pads?

73) Dan Clein advocates the use of contact and via cells, which is not a common design practice. What are his reasons?

74) In which situation should one avoid using the minimum allowed feature sizes allowed by the design rules?

75) What fundamental factors limits the speed with which detected design errors can be corrected?

76) When floorplanning a chip at the start of the IC layout process, what are the main goals in deciding how to arrange the major blocks in the design?

77) How is block floorplanning different from chip floorplanning?

7 What is a silicon compiler? How is it different from a tiler?

79) What is the difference between a channel router and a maze router? Which type of router will tend to produce higher utilization factors?

80) What is a chip assembly tool? What kind of routing should a chip assembly tool provide to have maximum flexibility?

81) At IBM, it has been found to be advantageous to sacrifice performance when migrating a chip design in one process into a second process. Process migration is facilitated by the use of "migratable design rules". What is the major benefit that can be obtained by such rules to offset the loss in potential chip performance?

82) At IBM a design methodology has been developed that makes the layout of standard cells very similar to that of gate array cells. What is the potential benefit of intermixing such cells in the same chip design?

83) In its ASIC design flow, IBM uses a formal verification tool that performs a technique called Boolean equivalence checking. What is the primary potential benefit of using formal verification methods in design verification? What is the conventional way of verifying the equivalence of different implementations of the same function?

84) IBM has standardized its logic design on the use of pulse-triggered latches, whereas the rest of the industry has tended to adopted design based on edge-triggered flip-flops. What is the strategy that IBM has adopted to be able to accommodate designers from other companies who wish to have ASICs fabricated through IBM?

85) Why are terminator cells sometimes used when clock trees are inserted into a block of placed standard cells?

86) When constructing a clock tree with distributed buffers, why is it very desirable to keep the buffers lightly loaded near the root of the clock distribution tree? Why can leaf nodes of the clock tree can be loaded more heavily? Why does one aim to have a balanced clock tree?

87) What is the difference between two- and three-dimensional analysis of interconnect capacitance.

8 Guard bands are usually built into the timing estimates employed by logic synthesis, cell placers, and other CAD tools. What is lost when the guard bands are relatively large? What could be gained if the timing estimates could be made more accurate?

89) Full 3-D capacitance calculations are generally extremely timing consuming. How can the technique of tunneling be used to make such calculations efficient enough to use in large IC designs?

89) The output of a 3-D field solver is a charge distribution over the signal net under consideration, and a charge distribution over the surrounding passive nets. Generally the signal net is assumed to be at a potential of 1 volt while the other nets are held at 0 volts. How can the signal net's self-capacitance and coupling capacitance then be computed?

90) Moore's Law predicts a doubling in the number of transistors per chip every two to three years. The major factor supporting Moore's Law is improvements in lithographic resolution that permit finer features. What are the two other major factors that Moore believes have allowed Moore's Law to hold? Even if physical factors allow for further increases in per-chip component density, what other factors could slow or even stop Moore's Law in practice?

91) What is meant by the term "dual damascene process"? How has the availability of this type of process simplified the creation of multiple interconnected metal layers?

92) In processes that have multiple layers of metal interconnect, why is it common to make the upper wires thicker than the lower layers? (The use of fat wires is sometimes called "reverse scaling".) In which situations would one be willing to use reverse scaling and hence appear to throw away the possible advantages of thinner wires?

93) What are some of the important reasons why DRAM technology has been a pioneer for semiconductor technology advances?

94) Briefly explain what are planar DRAM cells, trench capacitor DRAM cells, and stacked capacitor DRAM cells. Which type of cell is becoming dominant in embedded DRAMs? Why is this so?

95) There are numerous technological challenges and additional costs with embedded DRAM. Describe three of the main potential advantages that could be gained with embedded DRAM. What are characteristics of an application that could benefit from using embedded DRAM?

96) What are the three most common process solutions to providing embedded DRAM? Discuss some of the important trade-offs that must be made when selecting a process strategy for embedded DRAM.

87) What is the difference between two- and three-dimensional analysis of interconnect capacitance.

Ans: In 2d, we study the thickness and the width of the materials. Top consist of positive charge and bottom consist of negetive charge. In between is the insulator. CApacitance do affected by the amount of charge and thicness(in between top and bottom).

As in 3D we also a matters of length. Picture it like this. 2d analysis give u capacitance on area(at one point). The 3D give u the integration of the point from 0 to the L that u draw or simple word the volume capacitance.

66) What is the purpose of minimum area design rules?
Ans: LAyout will be fabricated, study shown that if the area of some subtrate is to small and/or the gap between them is less than a studied length, problem will accour in devices like overlapping or diffusion of two materials. Device will not working. Another this less area=less materials use=small device(wafer is still the same size)=many die = many devices per wafer = more money.

39) Why are 90 degree corners usually avoided in the layout of pad cells?
Ans: Current is by the movement of electrons in one direction, if there is a corner, electron pilled up will happen at the corner, resulting delay. THat corner also is easy to brake down in less time than straight line.

33) What is electromigration? How does electromigration affect the design of a standard cell based design?
Ans: Refer to ans 39, it got some thing to do with it.

31) The layout of standard cells is constrained to simplify the job of place & route tools. Give several examples of these constraints?
Ans: example, primitive cell=NAND, four NAND can be X-or, using X-OR, NAND and Inverter can produce ADDER. Instead or redraw, just take the primitive cell and route them.

14) Why is it that NMOS transistors can be created directly in a P-type substrate, whereas PMOS transistors must be created in an N-type well?
ANs: This for the thechnology that uses P-subsstrate..Easily said just look at the diffrent type..n-type is easyly doped into p-Substrate but for PMOS , the substrate it self is a p-type. Well we can use a high doppant of p-type materials but it's not cost effective and quit hard to do. N-type less doppan is the n-well and it's easy to diffuse them on p-type. Well now the p-type can be diffuse to the n-well easily will susal dop concentration.

11) What is the difference between a mask layer and a drawn layer in an IC layout? Why do layout designers usually only specify drawn layers?
Ans: like picture and the film negetive. MAsk layer is the opposite of drawn layer. Layout Eng. draw according to the circuit and trun it into the art of implementing materials. From circuit to drawn layout is straight work. THe opposite of that(area that is not drawn) is the mask layer.
LAyout Eng. draw-->Computer reverse the picture-->MAsk layer-->Fabrication

2)advantages:good integration of tools and service cut off their time wasted on tools
disadvantages:too expensive
3)Constrain in a flow in order to integrating different part of a system and with expected results

76)power line, noise, clock tree?

89)if you can solve this, you will be one of the chairman of synopsys or cadence. This is a good question for CADers.

4) Why are PMOS transistor networks generally used to produce high (i.e. 1) signals, while NMOS networks are used to product low (0) signals?

This should be a question on circuit, it is because there should be a voltage difference between the source and gate of transistors to make it work, so PMOS will generate a week 0 and NMOS will generate a week 1 too.

ans18vertical connection diagram illustrates the relative position, going vertically, of all the drawn layers. Such diagrams are especially useful in complex processses, such as DRAM processes.

Electromigration is generally considered to be the result of momentum transfer from the electrons, which move in the applied electric field, to the ions which make up the lattice of the interconnect material.

25) What is the difference between a soft IP block (soft core) and a hard IP block (hard core)?

Answer:
Softcore
- most flexible
- exist either as a gate-netlist or RTL.

Hardcare
- best for plug and play
- less portable and less flexible.
- physical manifestations of the IP design.

6) Why is the number of gate inputs to CMOS gates (e.g. NAND or NOR gates) usually limited to four?

Answer:
To limit the height of the stack.

As we all know, the number of transistor in the stack is usually equal to the number of input. The higher the stack the slower it will be.

16) What are process design rules? What is their major purpose? How are design rules created?

The purpose of the design rules is to ensure the greatest possibility of successful fabrication.The design rules are a set of requirements and advisement that are defined by the limits of the process(ie athe stable process window)which in turn is defined by the capabilities of the individual process steps.
In general minimum design rules are defined by the resolution and alignment capabilities of the lithographic systems.

12) In an IC layout, what is a polygon and what is a path? What are the advantages and disadvantages of each?
A polygon is a polygon and a pad is a pad. A pad can be easily edited and reshaped, however, it's off grid with 45 degree angle. Polygon is always on-grid, unless it's a copy and flip. However, polygon is hard to edit and work with.

59) How can a layout designer help ensure that the propagation delay along two conductors is very similar?

13) What is the difference between a contact and a via? What is a "stacked" via process?
Via: a contact between two conductive layers.
Contact:Opening in an insulating film to allow contact to an underlying electronic device.

13) What is a "stacked via process?
The placement of vias directly over the contacts or other,lower vias is known as stacked via.

6 What is the phenomenon of latch-up? Why is it a serious concern in CMOS layout design?

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)

An SCR is a 3-terminal 4-layered p-n-p-n device that basically consists of a PNP transistor and an NPN transistor as shown in Figure 1. An SCR is 'off' during its normal state but will conduct current in one direction (from anode to cathode) once triggered at its gate, and will do so continuously as long as the current through it stays above a 'holding' level. This is easily seen in Figure 1, which shows that 'triggering' the emitter of T1 into conduction would inject current into the base of T2. This would drive T2 into conduction, which would forward bias the emitter-base junction of T1 further, causing T1 to feed more current into the base of T2. Thus, T1 and T2 would feed each other with currents that would keep both of them saturated.

Fig. 1. A parasitic thyristor that can result in latch-up

A parasitic SCR is a pseudo-SCR device that is formed by parasitic bipolar transistors in the active circuit. These parasitic bipolar transistors, in turn, result from various p-n junctions found in the circuit. Latch-up is more widely associated with CMOS circuits because CMOS structures tend to contain several parasitic bipolar transistors which, depending on their lay-out, can form a parasitic SCR by chance.

Examples of parasitic bipolar transistors that may be found in CMOS circuits are as follows: 1) vertical PNP transistors formed by a p-substrate, an n-well, and a p-source or p-drain; and 2) lateral NPN transistors formed by an n-source or n-drain, a p-substrate, and an n-well. These parasitic PNP and NPN transistors may be coupled with point-to-point stray resistances within the substrate and the wells, completing the SCR configuration in Figure 1. If such an SCR device is formed from these parasitic transistors and resistors, then latch-up can occur within the device.

Events that can trigger parasitic thyristors into latch-up condition include: excessive supply voltages, voltages at the I/O pins that exceed the supply rails by more than a diode drop, improper sequencing of multiple power supplies, and various spikes and transients. Once triggered into conduction, the amount of current flow that results would depend on current limiting factors along the current path. In cases where the current is not sufficiently limited, EOS damage such as metal burn-out can occur.

The best defense against latch-up is good product design. There are now many design-for-reliability guidelines for reducing the risk of latch-up, many of which can be as simple as putting diodes in the right places to prevent parasitic devices from turning on. Of course, preventing a device from being subjected to voltages that exceed the absolute maximum ratings is also to be observed at all times.

15) Why must transistors be provided with "bulk" connections? What voltage levels are connected to a p-type substrate and an n-type well through these connections, and why?
Ans:To make the parasitic diodes reverse biased.p type substrstrate is generally connected to the most negative supply and n well is connected to the most positive supply of the circuit

20) What are the major advantages of hierarchical IC design

Concurrent design
· Design reuse
· Predictable schedule

26) In ASIC design, what are the main advantages of expressing the design using a hardware description language, such as VHDL or Verilog?

The main reason for using high level hardware design like VHDL or Verilog is easy generating hundred of million gate counts chip better than schematic entry design.

4.) PMOS is used to drive 'high' because of the thresholdvoltage-effect
The same is true for NMOS to drive 'low'.
A NMOS device cant drive a full '1' and PMOS cant drive full '0'
Maximum Level depends on vth of the device. PMOS/NMOS aka CMOS gives you a defined rail to rail swing
5.) The numbers you see there are usually the width and the length of the devices (channel dimensions drawn in the layout)
If given only one number it's the width combined with a default length
6.) This is only for gates that are connected in series (NOR PMOS-Part NAND NMOS)
Rootcause is the channelresistance and the delay of the corresponding edge
7.) Static-Power is the part that is comsumed by leakage mechanisms while the dynamicpart is caused switchingfreqency. Staticpower cant be zero cause you cant turn of a device completely.
8.) A transmissiongate is usually used as a latch. Biggest advantage of TG's is the small areaconsumption. It consists of one N and one PMOS, connected in source/source drain/drain configuration. Gateconnections: NMOS CLK PMOS inverted_CLK or the other way round.
9.) Loadcapacitance/Resistance and driverstrength (drivecurrent)
10.) Use Redrivers, reduce resistance, increase driversize change architecture
11.) Masklayers are used to produce IC's. You can have much more drawinglayers than masklayers. Some masklayers are the results of boolean operations applied to two or more drawinglayers. For example: you have design that uses thick and thinoxide transistors. So you need a difference between thin and thickoxide gates. You can achive that by drawing an extra layer afterwards you generate your thinkoxide mask by AND'ing the Gatelayer with the thickoxide-identification layer
12.) This depends on the tool you're using. AFAIK Path and Polys are only used in Cadence, sorry if I'm wrong.
Polygons can have any form while Paths are more like streets.
Paths are defined by width/layer and the centerline.
13.) contact and via means basically the same. Usually VIA is used if metallayers are connected while connections to source/drain/gate are named contacts
Stacking means that your technology allows to place vias/contacts directly over another
14.) NMOS need to form a channel with minority charges. That only works

in a p-Substrate because electrons are minority carriers there.
The the is true for PMOS and a NWELL/n-Substrate.
Note that this is only true if you want to have normally off(depletion) device.
15.) MOS does not need Bulks, if you take a look on SOI-Technologies
For normal technologies the bulk is used to control the backbias of the devices and to improve latchup behaviour.
The voltagelevels of the bulks have to be choosen in such a way that you dont forwardbias your diodes between source/drain and the bulk
16.) Designrules specify what you are allowed to draw and what not.
Metal width/spacings contact overlaps wellspacings implatoverlaps and stuff like that. The creation of designrules is more or less black magic
You run experiments which look quite promising, afterwards you get problems on your products and readjust)))))
Designrules specify what the fab is able to manufacture.
If you don't follow them you're fucked.
17.) Widthrules specify how wide a specific geometry has to be
There are max and min values sometimes. Spacingrules specifiy the distance between two shapes. Nowadays you even have spacings that depend on the width of the drawn shapes. Makes the job a bit more challenging Overlaprules specify the overlap of interconnectinglayers and their contactholes.
19.) Because you have to fix Power and critical signals first to make sure that you have them in and that they are dimensioned properly.
Typically you get more and more signals the longer the project is running and then you're running out of wirespace.
20.) Imagine you have a design in which you use 100 identical flipflops
You'll be faster is you draw one instance of the flipflop and copy it 99 times.
21.) If you miss any signal check of your list your 100% fucked.
DRC is needed to make sure that your design fits to the designrules you can easly get shorts by spacings being to small and so on ...
LVS is needed to verify that your drawn structures match the function of your schematics. Without that you end up with a totally different function (if it's working at all)
ERC looks for highohmic shorts in wells and in the substrate, finds forewardbiased diodes and stuff like that.
If you're confident enough you might skip ERC and DRC in a small metal redesign.
23.) Well, microprocessorguys need fast devices, while memoryguys need devices with a very very low leakage current. I dont know what the ASIC-guys do but I think they like taking the best of both worlds
26.) You can use the code for different technologies without problems.
27.) You have a lot of regular structures. Hierarchical layout is perfect for those structures. Therefore you start with the smallest leaf and build up hierarchy bottomup. Nevertheless TopDown design is off course used

at higher hierarchies.
28.) Total Area (Chip) divided by Cellarray Area
29.) If you design a DRAM you specify the wordlinepitch as a main-key parameter of the array. The rowdecoder you have to draw for that array has to fit to that wordlinepitch.
Circuits which have to meet such constrains are SenseAmps,Rowdecoder Columndecoder and Fuses
30.) INV,TRINV,NAND,NOR,Flipflops all with different driversizes
33.) Electromigration discribes a transporteffect caused by to much current in a wire. The wire starts to flow, getting thicker at one end and thinner at the other. Ends up in a fail wire. Happens only to Al/Cu Layers. Tungsten is unaffected.
37.) Clocks are usually distributed over the hole chip. Normally you would like your clocksignal to arrive everywhere at the same time to have best possible timing. You can achive that by many different clocktree-architectures In the layout you should try to shield clocks and try to reduce paraitic loads
40.) You mean hammerheads ? Never drawn a hammerhead at an outputdriver in 10 years. I think it has something to do with the cornerdevice
42.) Doing 10 years of DRAM-Layout and I have never heard of that ...
43.) Straped wordlines are a bit outdated nowadays. In the past they were used to reduce wordline resistance
44.) Because they have to fit in the pitch (see 29)
45.) Hmm, as far as I know alignmentmarks are only placed in the kerf The alignmentmarks in the Chip itself are mostly used for process control
46.) You want to make sure that your Voltagedrop is not to high.
Voltagedrop at VDD together with a rising VSS can cause serious trouble in a Design. Busses are timingcritical, differential signals are also critical so are shielded signals like biasvoltages and stuff.
Everything else is unimportant compared to the others
53.) Contactresitances are quite big.
55.) Litho-effects are worst for minimum geometries
5nm per edge is much for a 140nm wire but not for a 400nm
56.) Bias-nodes,supplyvoltages, compensation caps in analogcircuits delay-cells
58.) Capacitive coupling in the right moment could cause a small speedup
59.) Make sure that both signals 'see' the exactly same neighbourhood
60.) 45Degrees can be used everywhere when the technology allows it.
It reduces currentdensity in the corners
61.) See 33
You usually have electromigration guidelines. Draw your layout the fulfill those needs. You can also use Tools like Simplex to extract Powernets and to an currentdensity-simulation
62.) Powerlines don't need slits
63.) Cause a Via cant carry as much current as the wire without Via.
If you place to much contacts you introduce a weak spot in your wiring
64.) Cause you have diffent areas on the chip. You have highly regular

structures in the Array/SenseAmp Area -> Arrayrules really tight
Then you have the logicpart -> peripheral rules quite relaxed
65.) In the production process (usually metal-layers dd-processes) big wires might collect charges. If this charge finds a way to Gate the Gate might break. Possible Workarounds are tiedowndiodes or the reduction of the wirearea.
66.) Small chips, high yield, low cost, loads of work for layouters
67.) Line-End-Shortening.
Due to Lithoeffects all lineends print out significantly shorter than drawn
68.) Parasitic tyristors that you get automatically in CMOS ignite and destroy your hardware. It's getting serious because of the shrinking dimensions of modern technologies. Therefore the ignitionvoltage is reduced and voila, you just killed your silicon
Can be controlled by designrules, substrate and wellcontacts, dopings
69.) see 68
70.) Metall or fuseoptions are far cheaper than Gate or ActiveArea Masks
74.) I would say whereever possible
75.) Where is the error is the most important question ?
Is there a simple workaround ? Is there enough room for the fix ?
Which layers are affected ? Is there sparelogic to use ?
87.) Accuracy, Runtime
94.) Nobodys using planar DRAM-Cells anymore because you need to much area to get your cellcapacitance.
Stack builds up little trees from substratelevel while the Trench is digging a hole in the substrate. I think that Trench will become dominant because you have no planarisation problems between array and logic

) According to Clein, what has been one of the main reasons why CAD tools have failed to be successful among IC layout engineers?

I don't completely agree with that statement. The only thing I can say from my experience and from what the layout engineers told me is that some of the features of the cad tools are not very "user friendly". What I mean is that it will be very hard to use only predefined cells(p-cells) when you have to do a full-custom, low offset, good matching layout for an analog aplication. It's much easier a lot of times to make your one cells/blocks

What is retiming & Register balancing?..

what is diffrence between buffer and transistor?

what is diffrence between FETand BJT?

I gathered these quest ions f rom several emails, sent to me by students who at tended on- site interviews at varios dif f erent companies. I shall t ry to add more of them in near f uture.

1. What is t he dif f er ence bet ween a lat ch and a f lip f lop. For t he same input
,
how
would t he out put look f or a lat ch and f or a f lip-f lop.
2. Finit e st at e machines:
(2.1)Design a st at e-machine (or dr aw a st at e-diagr am) t o give an out put ' 1' when t he # of A' s ar e even
and # of B' s ar e odd. The input is in t he f orm of a ser ial-st r eam (one-bit per clock cycle). The input s could be of t he t ype A, B or C. At any given clock cycle, t he out put is a ' 1' , pr ovided t he # of A' s ar e even and # of B' s ar e odd. At any given clock cycle, t he out put is a ' 0' , if t he above condit ion is not sat isf ied.
(2.2). To det ect t he sequence "abca" when t he input s can be a b c d.
3. minimize a boolean expr ession.
4. Dr aw t r ansist or level nand gat e.
5. Dr aw t he cr oss-sect ion of a CMOS inver t er .
6. Der iving t he vect or s f or t he st uck at 0 and st uck at 1 f ault s.
7. Given a boolean expr ession he asked me t o implement j ust wit h muxes but not hing else.
8. Dr aw I d Vds cur ves f or mosf et s and explain dif f er ent r egions.
9. Given t he t r ansf er char act er ist ics of a black box dr aw t he cir cuit f or t he black box.
10. Given a cir cuit and it s input s dr aw t he out put s exact t o t he t iming.
11. Given an inver t er wit h a par t icular t iming der ive an inver t er using t he pr evious one but wit h t he r equir ed t iming ot her t han t he pr evious one.
12. Change t he r ise t ime and f all t ime of a given cir cuit by not changing t he t r ansist or sizes but by using cur r ent mir r or s.
13. Some pr oblems on clamping diodes.

What is meant by D-FF?

1. What is the basic difference between Latches and Flip flops?
2. What is a multiplexer?
3. How can you convert an SR Flip-flop to a JK Flip-flop?
4. How can you convert an JK Flip-flop to a D Flip-flop?
5. What is Race-around problem? How can you rectify it?
6. Which semiconductor device is used as a voltage regulator and why?
7. What do you mean by an ideal voltage source?
8. What do you mean by zener breakdown and avalanche breakdown?

9. What are the different types of filters?
10. What is the need of filtering ideal response of filters and actual response of filters?
11. What is sampling theorem?
12. What is impulse response?
13. Explain the advantages and disadvantages of FIR filters compared to IIR counterparts.
14. What is CMRR? Explain briefly.
15. What do you mean by half-duplex and full-duplex communication? Explain briefly.
16. Which range of signals are used for terrestrial transmission?
17. What is the need for modulation?
18. Which type of modulation is used in TV transmission?
19. Why we use vestigial side band (VSB-C_3F) transmission for picture?
20. When transmitting digital signals is it necessary to transmit some harmonics in addition to fundamental frequency?
21. For asynchronous transmission, is it necessary to supply some synchronizing pulses additionally or to supply or to supply start and stop bit?
22. BPFSK is more efficient than BFSK in presence of noise. Why?
23. What is meant by pre-emphasis and de-emphasis?
24. What do you mean by 3 dB cutoff frequency? Why is it 3 dB, not 1 dB?
25. What do you mean by ASCII, EBCDIC?

MICROPROCESSOR QUESTIONS

1. Which type of architecture 8085 has?
2. How many memory locations can be addressed by a microprocessor with 14 address lines?
3. 8085 is how many bit microprocessor?
4. Why is data bus bi-directional?
5. What is the function of accumulator?
6. What is flag, bus?
7. What are tri-state devices and why they are essential in a bus oriented system?
8. Why are program counter and stack pointer 16-bit registers?
9. What does it mean by embedded system?
10. What are the different addressing modes in 8085?
11. What is the difference between MOV and MVI?
12. What are the functions of RIM, SIM, IN?
13. What is the immediate addressing mode?
14. What are the different flags in 8085?
15. What happens during DMA transfer?
16. What do you mean by wait state? What is its need?
17. What is PSW?
18. What is ALE? Explain the functions of ALE in 8085.

19. What is a program counter? What is its use?
20. What is an interrupt?
21. Which line will be activated when an output device require attention from CPU?

QUESTION NO. 1 :

Next State Equations for 3 bit up counter using only AND, OR, NOT, and XOR gates is as given below.

Equations:

$$\mathbf{Q0 = not\ (Q0)}$$
$$\mathbf{Q1 = Q1 \oplus Q0}$$
$$\mathbf{Q2 = Q2 \,.\, not\ (Q0) + Q2 \,.\, not\ (Q1) + not\ (Q2) \,.\, Q1 \,.\, Q0}$$

Will the circuit have any problems with respect to set up and hold violations? Check and explain…

Note :

Assume 3 input AND & OR are available . And 3 input and 2 input AND gates have same delays.

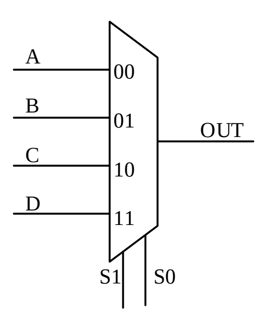
2ns < TpINV < 4ns --- - propagation delay for INVERTER
3ns < TpAND < 6ns --- propagation delay for AND GATE
3ns < TpOR < 5ns --- propagation delay for OR GATE
3ns < TpXOR < 4ns --- propagation delay for XOR GATE
3ns < Tcko < 8ns --- CLOCK TO OUTPUT DELAY OF FLIP FLOP
Tsetup = 4ns
Thold = 6ns
CLK = 40 MHz

QUESTION NO. 2 :

Design a 2 Bit Down Counter that is implemented with T FFs.

QUESTION NO. 3 :

Use 3 BUFTs and 3 BUFEs to build a 4-1 MUX.



Note : BUFT- Tri State Buffer with Active Low Enable
BUFE – Tri State Buffer with Active High Enable

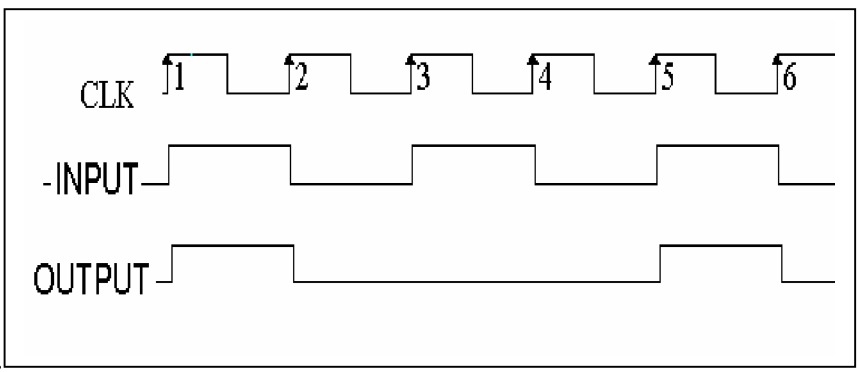
QUESTION NO. 4 :

Design a 3 bit binary counter with synchronous pre-load and synchronous reset using D f/fs. The counter has an input called load. If load is high, at the high clock edge, the counter is loaded with count provided by the user on the data-in inputs of the counter. The counter has another input called reset. If reset is high on the clock edge, then the counter is loaded with a value "000". The reset has higher priority than load.

Note : Can use 3 bit SYN counter Next State Equations given in QUESTION NO. 1 above.

QUESTION NO. 5 :

Draw only state diagram for the following input /output relationship.

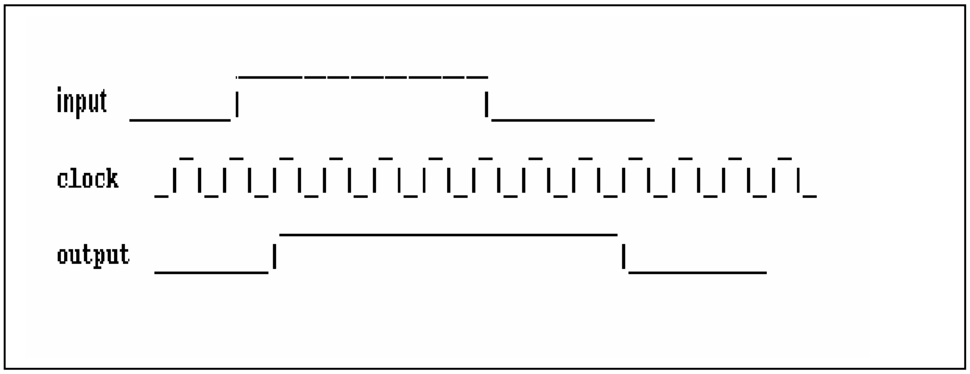


.

QUESTION NO. 6 :

Draw only the state diagram for the circuit which delays the negative edge of the input signal by

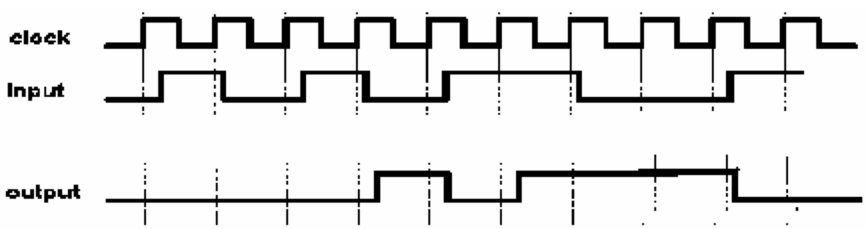
2 clock cycles as shown below.



QUESTION NO. 7 :

Draw only state diagram for FSM which detects more than one "1"s in last 3 samples as shown.

For example: If the input sampled at clock edges is 0 1 0 1 0 1 1 0 0 1 then output should be 0 0 0 1 0 1 1 1 0 0 as shown in timing diagram.



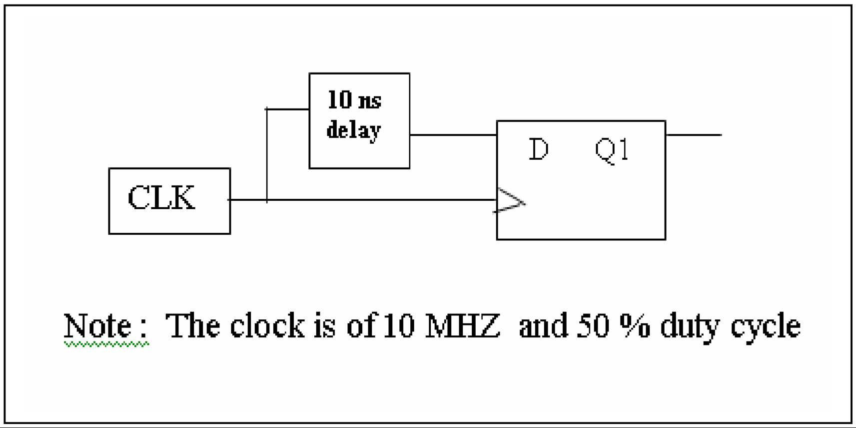
QUESTION NO. 8 :

Obtain a frequency divider circuit using a D latch and additional gates.

QUESTION NO. 9 :

Draw the output wave form at Flip flop output.

(Assume, D FLIP- FLOP is **dual edge triggered .**



QUESTION NO. 10 :

Input clock signal is of 20 MHZ , and 80 % Duty Cycle .

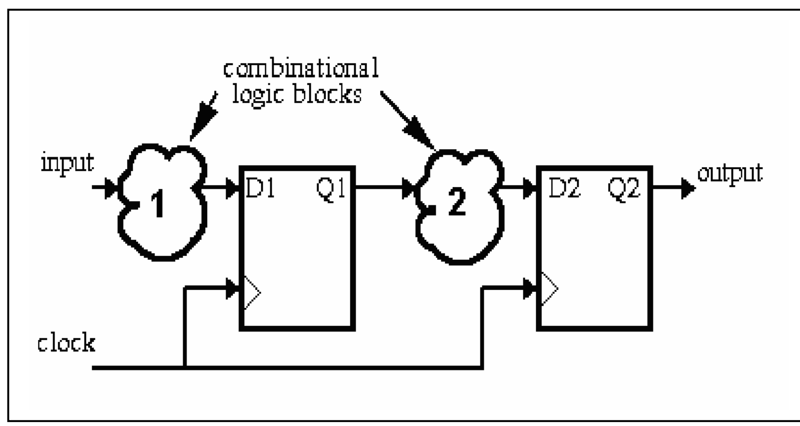
Obtain output clock of 10 MHZ, and 50 % Duty Cycle . Use D FF . and additional logic .

Note : Do not use delay element block as above.

QUESTION NO. 11 :

For the adjacent circuit,

a. **T clock to Q = 5 ns**
b. **T setup = 3ns**
c. **T hold = 1 ns**
d. **Combo logic 1 = 3 ns**
e. **Combo logic 2 = 4 ns**
f. **Positive clock skew = 1ns**



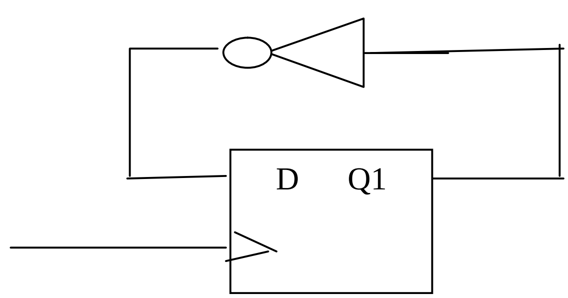
Find the **maximum frequency** of operation.

..

QUESTION NO. 12 :

For the circuit shown,

a. T setup = 3ns
b. T hold = 2ns



c. T inv delay = 3ns
d. T clk to Q1 = 4ns
e. Find Max Frequency of operation.

QUESTION NO. 13 :

Implement **A + BC + (Abar . Cbar)** using **only two** 2: 1 MUXes .

(No additional gates to be used.)

QUESTION NO. 14 :

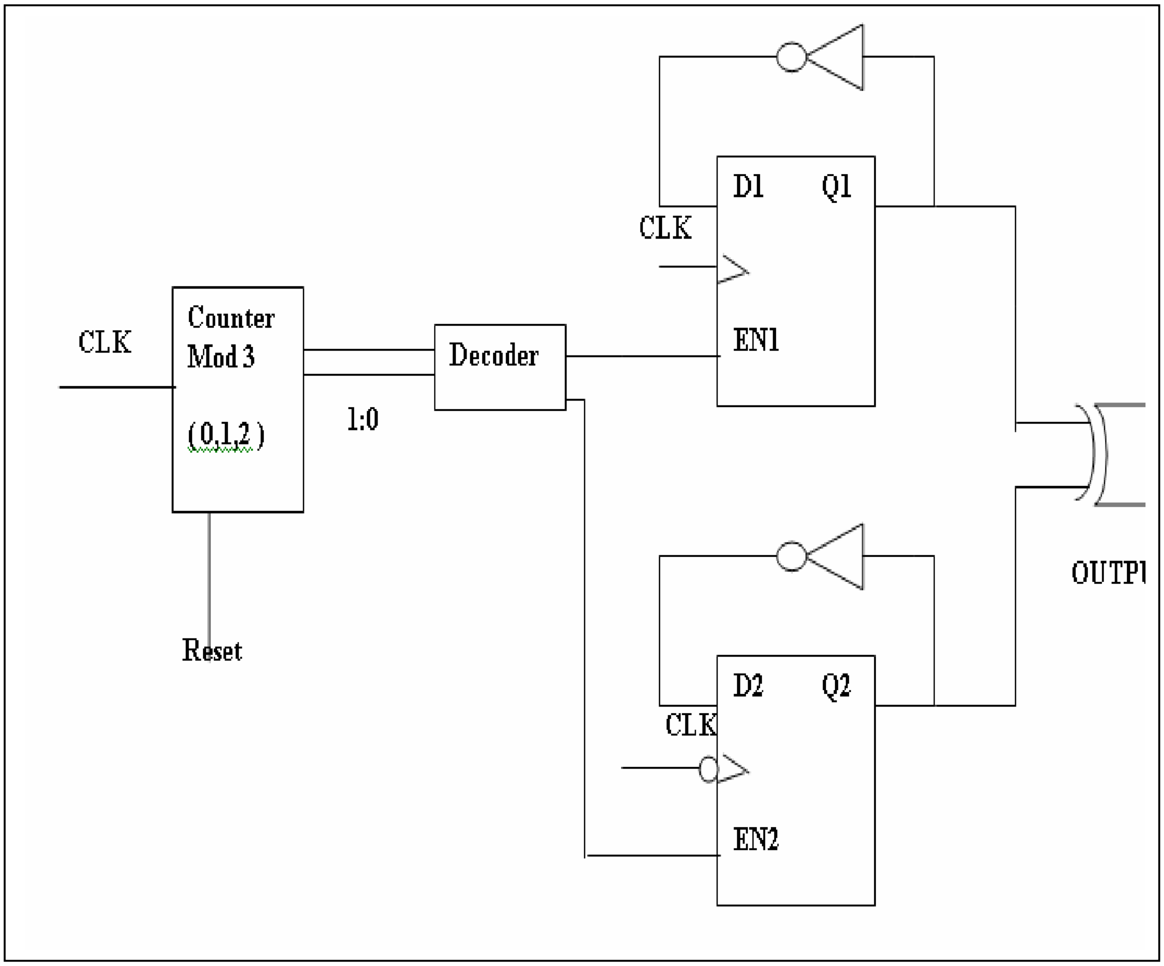
Following circuit. It has 2 flip- flops one Positive edge triggered and other negative edge triggered.

The Counter is synchronous MOD 3 running through states 0, 1, 2,…& repeat.. with 2 bit output.

EN 1 (of Positive edge Triggered FF) = 1 , When Counter output = 0
EN 2 (of Negative edge Triggered FF) = 1 , When Counter output = 2
Decoder does this decoding for EN 1 and EN 2.

CLK is the clock input to counter and both the Flip Flops.

Draw the timing diagram for two flip – flop outputs and final XOR GATE output with respect to the CLK input and deduce relationship between CLK input and XOR OUTPUT.



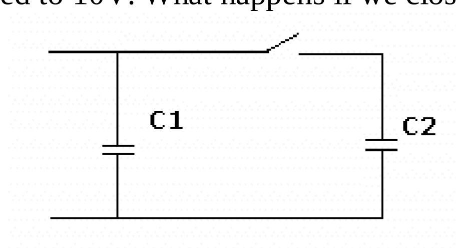
-
- What is the difference between compiled, interpreted, event based and cycle based simulator ?

Given the following multiplexor, and the four possible input signals A, A*, B and B*, hook up the inputs to the muxes input pins and select pins such that the output is the exlusive-or of A, B:

1. Divide by 3, with 50% duty cycle
2. Swap A & B without using third temp variable
3. Define Rest removal and recovery time.
4. Define Top-Down and Bottom-Up approach in Synthesis
5. Implement two input Nand gate using 2:1MUX
6. Define ESD and how can it be reduced
7. Tri-State Bus.
8. Clock Gating: Adv & Disad
9. How to bypass gated reset in Scan mode
10. Effect on Delay with respect to Load capacitance
11. Two major factors for increase in Power consumption
12. Implement JKFF using DFF
13. Blocking and Non Blocking in Veriog
14. Diff b/w $Display and $Monitor
15. Soft and Hard Macro
16. What is the possible maximum frequency for the clock for a DFF whose Q output is connected back to D-input.
17. One ASIC has Set up violation, and another ASIC has Hold time violation, how can we use both without redesigning.
18. Diff Latches and FF
19. Transistor level diagram for two input NAND gate

ANSWERS TO ELECT:-

1.Q: Two capacitors are connected in parallel through a switch. C1= 1uF, C2=0.25uF. Initially switch is open,C1 is charged to 10V. What happens if we close the switch?



No loss in the wires and capacitors.

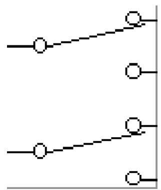
A:
Since no loss in the circuit the charge remains the same:
U1C1 + U2C2 = U3(C1+C2)
U3 = (U1C1+U2C2)/(C1+C2) = (10*1 + 0*0.25)/1+0.25 = 8
U3= 8V

2.A

```
<---------------( lamp )------o___o------
                                    |
```

```
                                        o            |
                                        |            |
                                        |            |
                                        |            |
                                        |            |
                                        o      S2    |
                                         ___    |
      <----------------------------o   o------
```

3.A



use extra DPDT switch

4A

Q:What will be the voltage between the 2 capacitors connected in series between Vdd and GND?

A:
Q1=C1U1; Q2=C2U2
U2=C1U/(C1+C2)=4v

5. Q: You work on a specification of a system with some digital parameters. Each parameter has Min, Typ and Max colomns.
What colomn would you put setup and hold time?

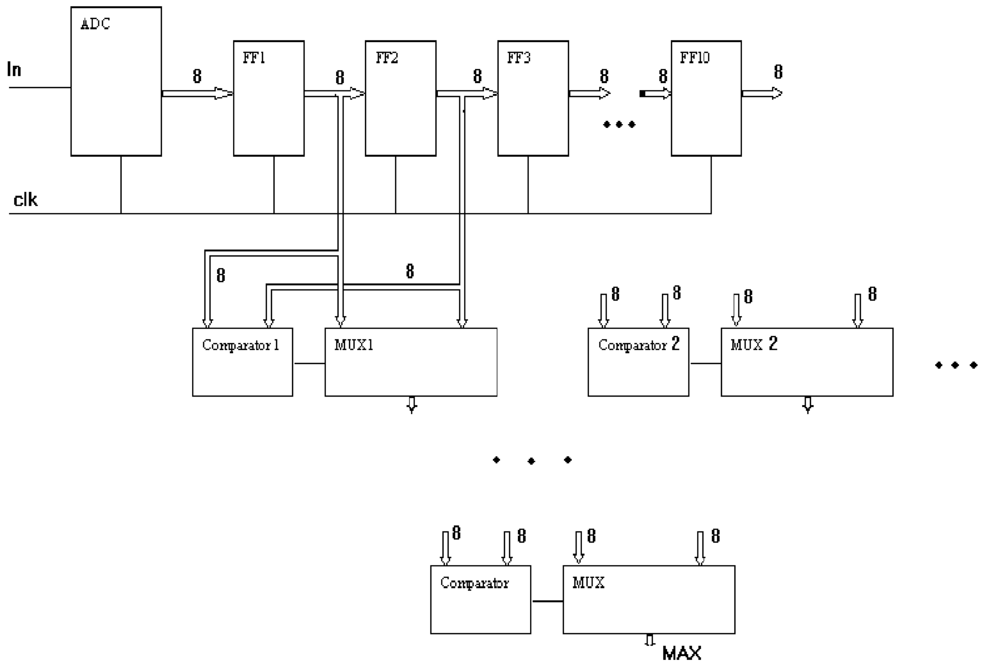
A: put SETUP time into the Min colomn, put HOLD time into the Min colomn too.
Example:
usually the data must be set at least (minimum) X nS before clock and being held at least Y nS after the clock. You need to specify Min setup and Min hold time.

5.A

```
Q: Design a simple circuit based on combinational logic to
double the
Output frequency.

A:
                                    ______________
    ____      ____      ____      |            |      _   _
_   _
```

```
   |    |    |    |    |    |    |           |     |      |
| | | | | |
  _|    |____|    |____|    |__  |           |    _| |_| |_|
|_| |_
                          |           |
  ------------------------------>|           |--------------
------>
                          |           |
                          |______________|



          |\            |\
          | \           | \
     _____|  \o_________|  \o_______
    |   |  /         |  /       |
    |   | /          | /      |
    |   |/ not        |/ not       |
    |                         |
    |                         |   \\-----.
    |                         +----\\     \
    |                               ||      )--------+
---+------------------------------//     /
                               //-----'
                              XOR
```

6A

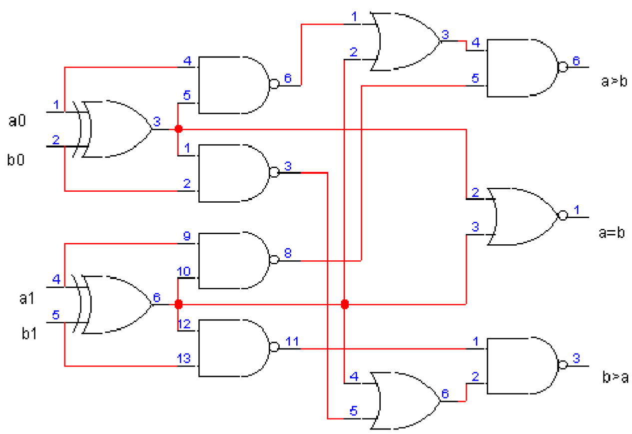


Since we need to find MAX of every 10 samples, we are going to place after ADC a FIFO 8 bit wide and 10 words deep.
It will require 8*10 flip flops.
Every two stages of FIFO are forwarded to comparator and multiplexer. The comparator compares two 8 bit numbers
and enables a multiplexer to choose the maximum of these two numbers.
It will require 9 pairs of comparator/multiplexer to find the MAX number.
So far every new clock there will be new MAX number.

8A
Q: Implement comparator using combinational logic, that compares two 2-bit numbers A and B. The comparator should have 3 outputs: A > B, A < B, A = B.

A: A1	A0	B1	B0	A>B	B>A	A=B
0	0	0	0	0	0	1
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	0	1	0
0	1	0	0	1	0	0
0	1	0	1	0	0	1
0	1	1	0	0	1	0
0	1	1	1	0	1	0

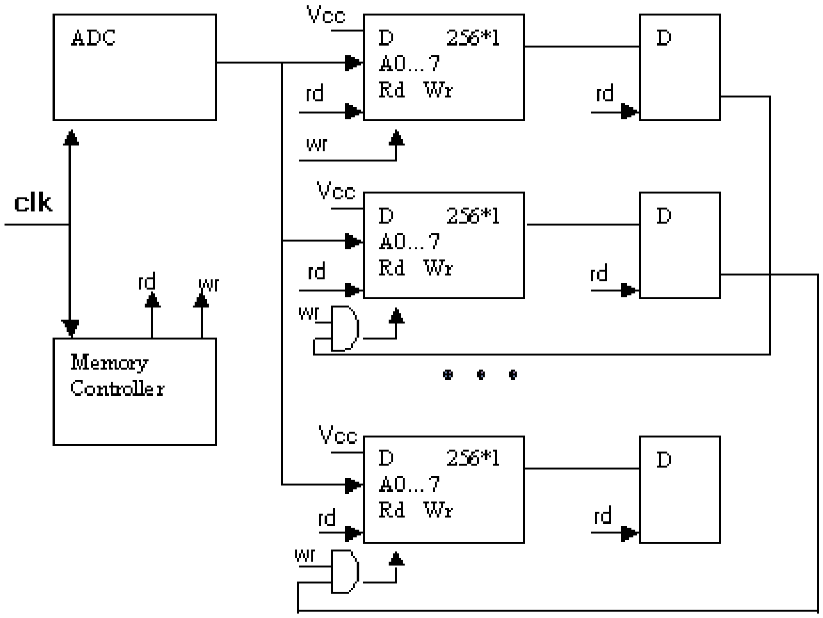
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	0	1
1	0	1	1	0	1	0
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	0	1



Here is what the behavioral model of the comparator looks like:

```
module comp0 (y1,y2,y3,a,b); input [1:0] a,b;

 output y1,y2,y3; wire y1,y2,y3; assign y1= (a >b)? 1:0;

 assign y2= (b >a)? 1:0; assign y3= (a==b)? 1:0;endmodule
```

9.Q: You have 8 bit ADC clocking data out every 1mS. Design a system that will sort the output data and keep a statistics how often each binary number appears at the output of ADC.



The diagram showes a basic idea of possible solution: using RAM to store a statistic data.
The digital number that is to be stored is considered as RAM address.
Once digital data at the output of ADC becomes available, memory controller generates RD signal,
and the contence of the memory cell addressed by ADC output latches into D register.
"1" in the D-register enables WR signal to the next memory cell. To calculate how many times a certain number appeared at the output of ADC it necessary to summarize the contence of all memory cells.

11.A

```
Q:Implement D- latch from
a) RS flip flop;
b) multiplexer

       _______________________
b)    |                   |
      |                   |
      |     |\            |
      |     | \           |
      |_____|  \          |
            | o \         |
            |   |         |  out
  in        |   |_________|_______
 ______________| 1 |
            |   /
            |  /
            | /|
            |/ |
  clock               |
```

```
 ___________________|
```

```
 This is a 2-input mux implements D-latch. It works this
way:
  At every clock =1    out=in,
     when clock =0    out= last in
```

12.Q:How to convert D-latch into JK-latch and JK-latch into D-latch?

Compare the truth tables of D-latch and JK-latch accordingly:

```
clk   D     Q
==================
+     0     0
+     1     1
```

```
clk   J     K     Q
=========================
+     0     0     hold
+     0     1     0
+     1     0     1
+     1     1     switch to opposite
```

```
Using these truth tables it is easy to convert JK-latch
into D-latch.
              _______
     D       |      |   Q
     ___________|J |----
       |     |      |
     __|__     > clk      |    _
     \   /   |      |   Q
      \ /    ___|K |----
       v    |      |_______|
       o    |
       |____|
```

```
To convert D-latch into JK-latch some extra logic is
required.
```

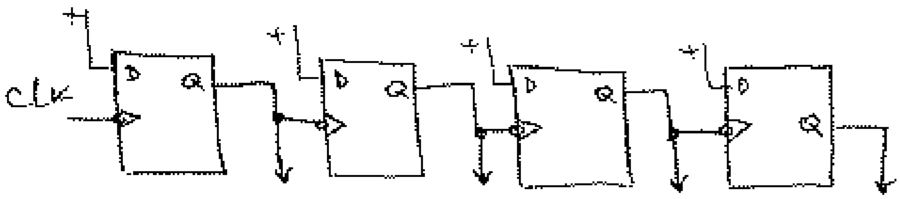
The following table shows the relation between J,K and D

```
J      K      D
=================
0      0      Q
0      1      0
1      0      1
1      1      !Q
```

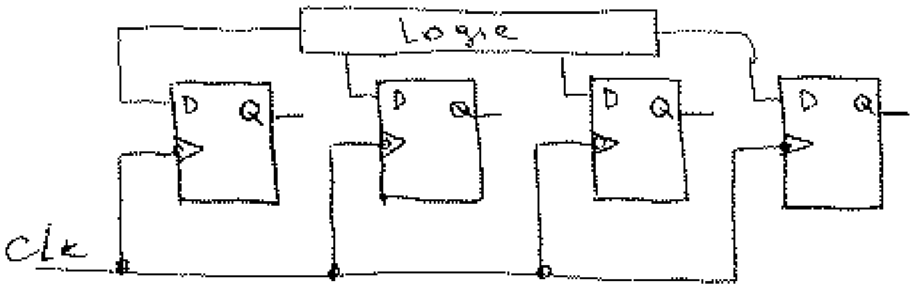
```
             _______________________________
           |                                |
              ___|___            _______    |
       |     |           |     |   |
    _J__|       |________________|D    Q |---'
       | A   |           |     |
       |     |           |     |
     K |     |           |clk  _      |
    ----|    |          --|     Q |--.
       |_______|         |_______|  |
          |                        |
          |________________________________|
```

Looking at the drawing and the table it is not a problem to implement block A.
Probably the easiest way is to use a MUX.
J and K are control signals and 1,0,Q,!Q are data inputs.

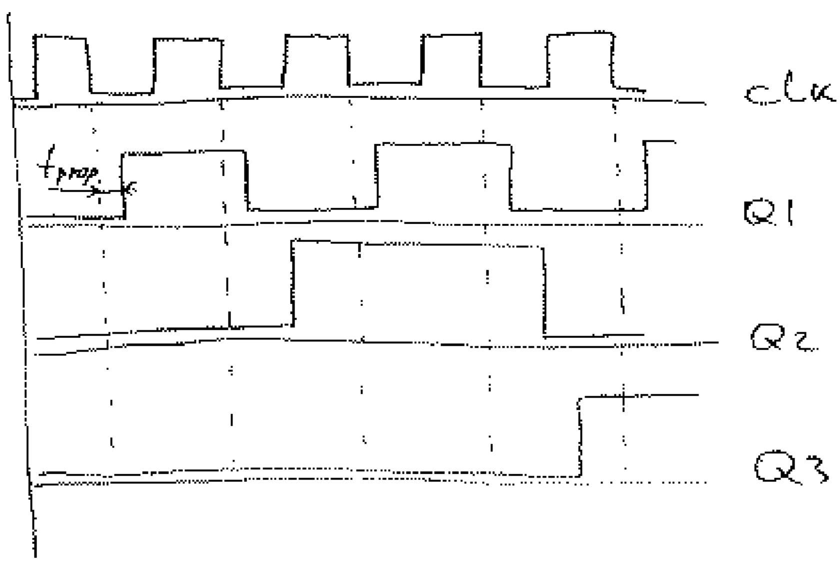
13.Q:You have two counters to 16 built from negedge D- FF . First circuit is synchronous and second is "ripple" (cascading). Which circuit has less delay?



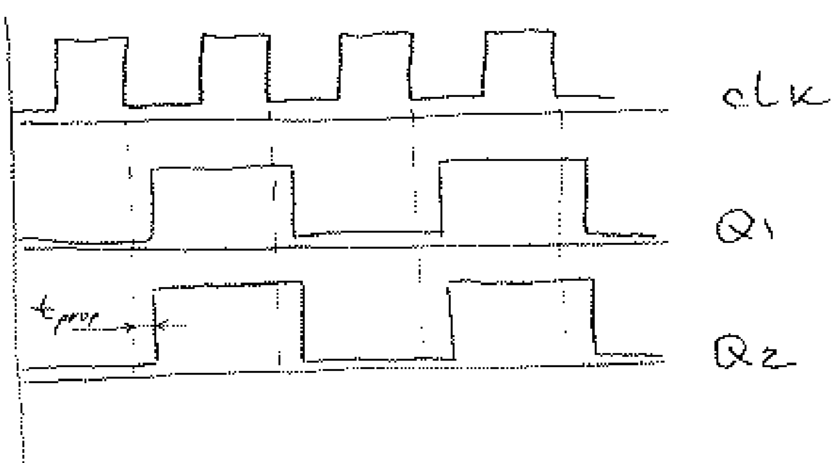
1



2



3



4

1 - is ripple counter;
2 - synchronous.
Both consist of 4 FF, synchronous counter also has some logic to control it's operation. From diagram 3 (for ripple) and 4 (for synchronous) it is seen that propagation delay of ripple counter is 4* t_prop , while synchronous counter has only 1* t_prop.

15.A
Q:What is the difference between flip-flop and latch?

```
        _______                    _______
   D   |      |   Q         D     |    |   Q
   -----|     |----         -----|     |----
  clock |     |            enable |     |
   ----->     |   _          -----|     |   _
       |      |   Q              |      |   Q
       |      |----              |      |----
       |_______|                 |_______|
```

The example shows D-latch and D-FF.
The simplest form of data storage is latch.It's output responds immediately to changes at the input and the input state will be remembered, or "latched" onto.While "enable" input is active the input of the latch is transparant to the output,
once "enable" is disactivated the output remains locked.
Flip flops use clock as a control input. The transition in output Q occurs only at the edge
of the clock pulse. Input data must present T_setup time before
clock edge and remain T_hold time after.

* RESET input, while it is not shown, is present in most FF.

```
module DFF (Q,_Q,D,clk,rst);
output Q,_Q;
input D,clk,rst;
reg Q,_Q;

always @(posedge clk or posedge rst)
 begin
  if (rst) Q  <= 0;
  else     Q  <= D;
  _Q <= !Q;
```

```
 end

endmodule

module DLatch (Q,_Q,D,en,rst);
output Q,_Q;
input D,en,rst;
reg Q,_Q;

always @(en or D or posedge rst)
 begin
  if (rst)      Q  <= 0;
  else if (en) Q  <= D;
  _Q <= !Q;
 end

endmodule
```

=>

DAC parameters

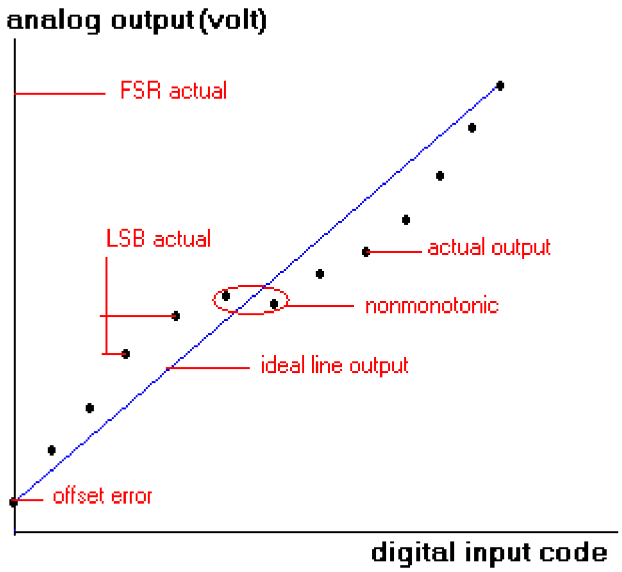


Fig.1

Resolution - normally given in bits.
Resolution indicates the smallest increment of its output,
corresponding to a 1 LSB input code change. For example for 10 bit DAC, 2 power 10 = 1024 codes,
so the resolution is 1/1024 of the output range.

Full scale range (FSR) - maximum output signal for the DAC,specified as current or voltage (ma or V).
Can be negative ,positive or both.

Offset error - difference between an ideal and actual DAC output when zero digital code applied to the input.

Gain error - the difference between an ideal and actual output when full scale digital code applied to the input.
Strongly depends on VREF stability.

Differential Nonlinearity - (see the drawing below) is measured whith a ramp code applied to the input of DAC.
The step between every pair of the adjacent codes should not exceed 1 LSB.
1LSB is calculated from gain and offset measurements:

1 LSB = FSR/N-1

Differential Nonlinearity represents the error from 1 LSB for every step,
where N=(2 power of bit number) .

Integral Nonlinearity - shows how the output differs from an ideal line. It is also measured in LSB. Usually 1LSB is an exeptable value for nonlinearity.
Fig.2 and Fig.3 show the difference between Integral and Differential Nonlinearity.
Offset and Gain are absolute measurements,DNL and INL are referenced to the zero and full scale outputs of the DAC.

SNR - signal to noise ratio. Measured with digital code representing sine wave applied to the input. Fundamental
and harmonic components of the sine wave are filtered out. Any remaining signal at the output of the DAC
is considered as a noise.
SNR is a ratio of the full scale sine wave output to noise level.
Some companies specify **SINAD** parameter. SINAD - is a signal to noise and distortion ratio. It is similar to SNR, but the harmonic signal components are not removed. Specified in dB.

THD - total harmonic distortion - measured with digital code representing sinewave,applied to the DAC input continuously.
The output is analyzed in the frequency domain to find harmonic components related to the fundamental output signal.Specified in dB.

IM - intermodulation distortion - non harmonic product terms that appear in the output signal due to nonlinearity
of the DAC. Measured with summed 2 sinusoid tone applied to the input.
The output is tested on harmonic components,appearing due to modulation affect on non linear characteristics within DAC.
The harmonic components are (F1+F2), (F1-F2) , (2F1+/-F2), (F1+/-2F2) ...

Max conversion rate - it is actually max input signal frequency the DAC can handle.The worst case is when
input signal changes from zero to max,and the output should reach the max level and settle.

Settling time - the time required for the output to reach the final value and remain within +-1 LSB
after overshoot.

PSRR - power supply rejection ratio .The output signal should remain within limits while power supply
voltage changes from Min to Max.

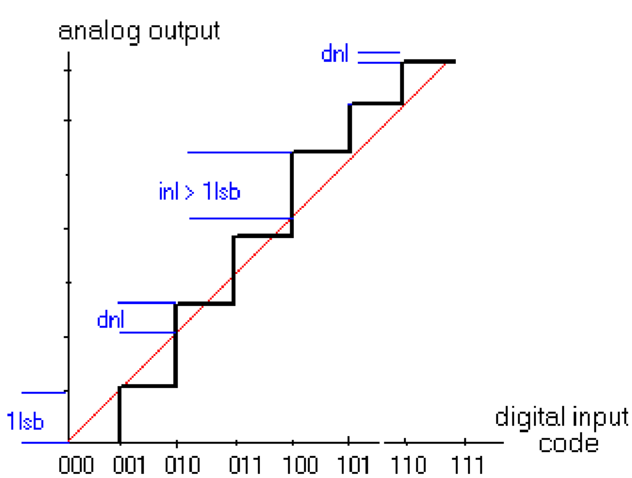


Fig.2

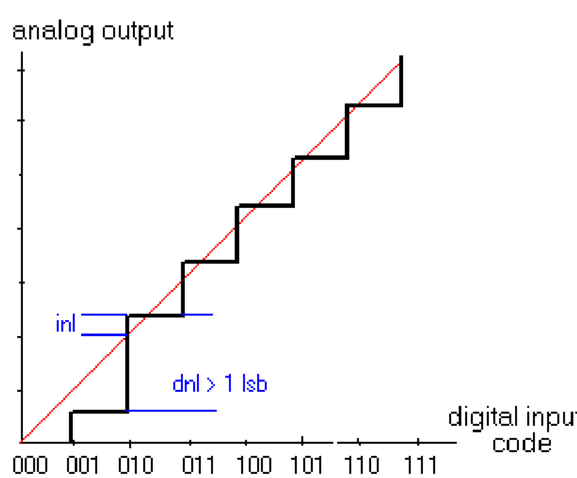


Fig.3

Fig.2 shows the case when DNL is within limits (<1 lsb) and INL is above 1lsb; Fig.3 - the opposite case.
If continuously increasing code from 0000 to 11111111 applied to the DAC input,the

output should represent
a rising staircase with equal steps. If staircase is smoothed, a perfectly straight line should result.
The INL defines the overall straightness of this line, whereas the DNL describes differences in amplitude
between adjacent steps in the staircase waveform.

Which of these two parameters is more important depends on the application.
In the imaging application, it may be necessary to distinguish between slightly different color densities
in adjacent areas of an image. Here DNL is more important .
But in an application in which a widely varying parameter like speed must be continuously monitored,
INL is usually more important.

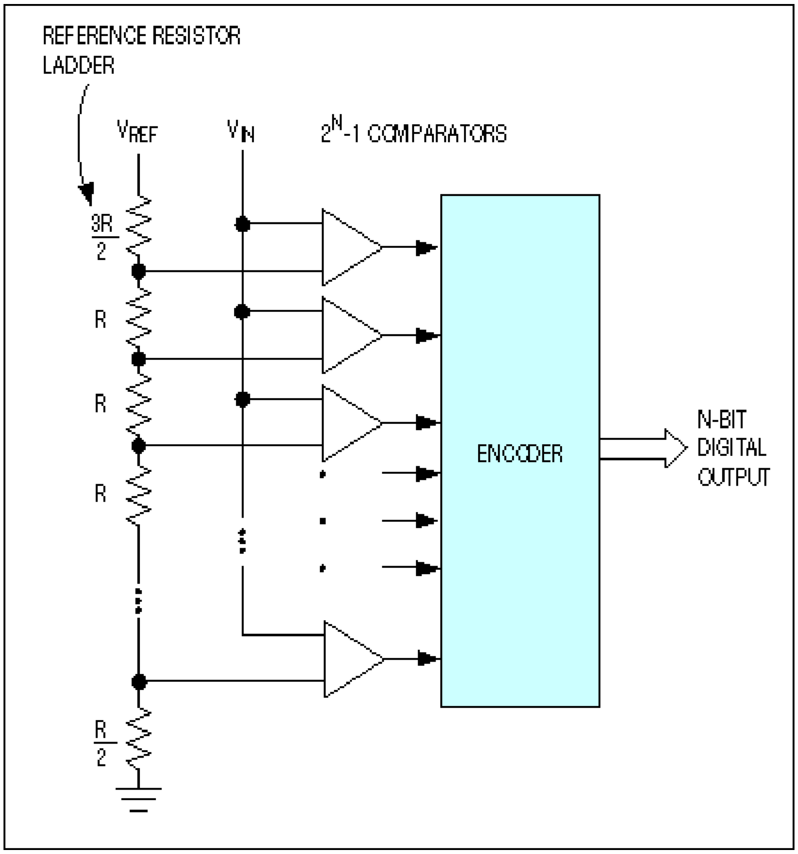
DNL must be less than 1lsb to garantee DAC's monotonic behavior (Fig.1).
A Monotonic DAC has an output that changes in the same direction (or remains constant) for each increase
in the input code. The quality of monotonicy is important if DAC is used in feedback loop.
When a non monotonic device is used in a feedback loop, the loop can get stuck and DAC will toggle forever between 2 input codes.

If you find an error in this article or would like to add something , please send an e-mail to eclub.

HITEQUEST Apr 2003	**A/D converters**

Direct-conversion ADCs

Direct conversion is also known as a flash conversion. ADCs based on this architecture are extremely fast with a sampling rate of upto 1GHz . However, their resolution is limited because of the large number of comparators and reference voltages required.

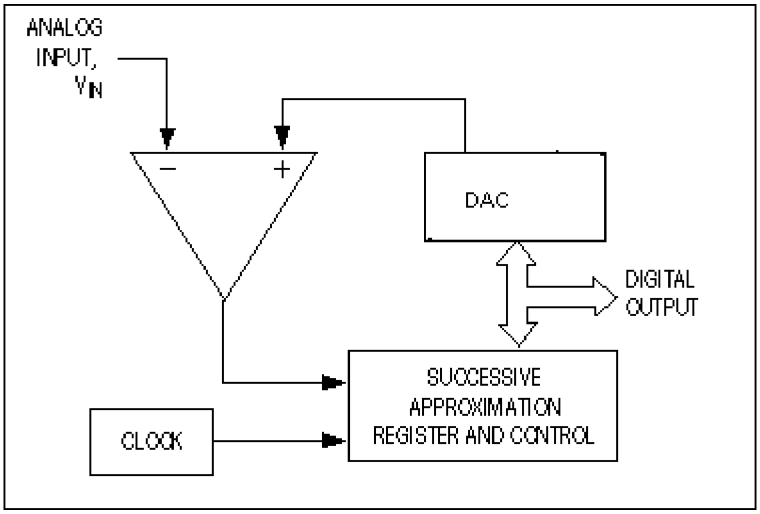


The input signal is fed simultaneously to all comparators. A priority encoder generates a digital output that corresponds with the highest activated comparator. The identity of comparators is important, any mismatch can cause a static error.
Flash ADCs have a short *aperture* interval - the time when the comparators' outputs are latched.

Successive-approximation ADCs

The conversion technique based on a successive-approximation register (SAR), also known as bit-weighing conversion, employs a comparator to weigh the applied input voltage against the output of an N-bit digital-to-analog converter (DAC). Using the DAC output as a reference, this process approaches the final result as a sum of N weighting steps, in which each step is a single-bit conversion.

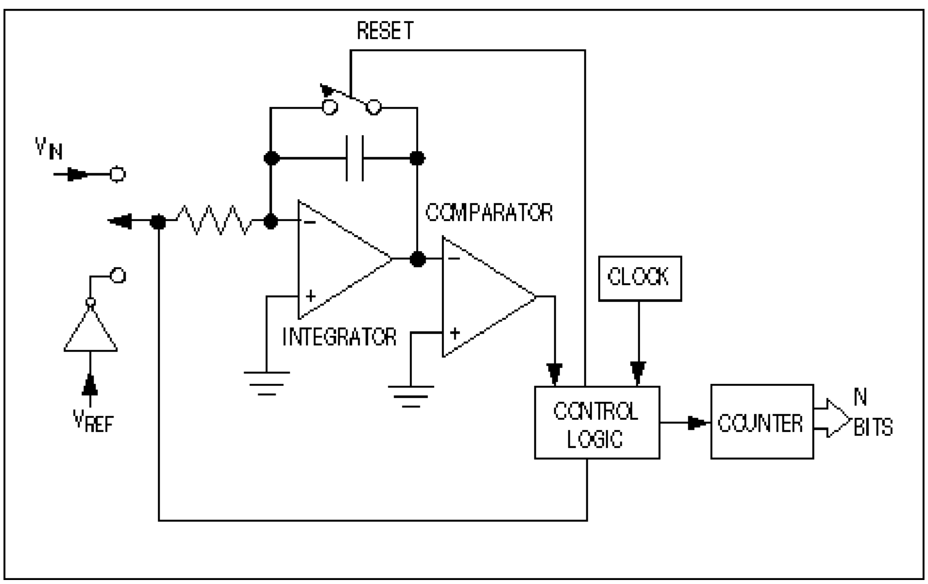
Initially all bits of SAR are set to 0. Then, beginning with the most significant bit, each bit is set to 1 sequentially. If the DAC output does not exceed the input signal voltage, the bit is left as a 1. Otherwise it is set back to 0. It is kind of a binary search. For an *n*-bit ADC, *n* steps are required.



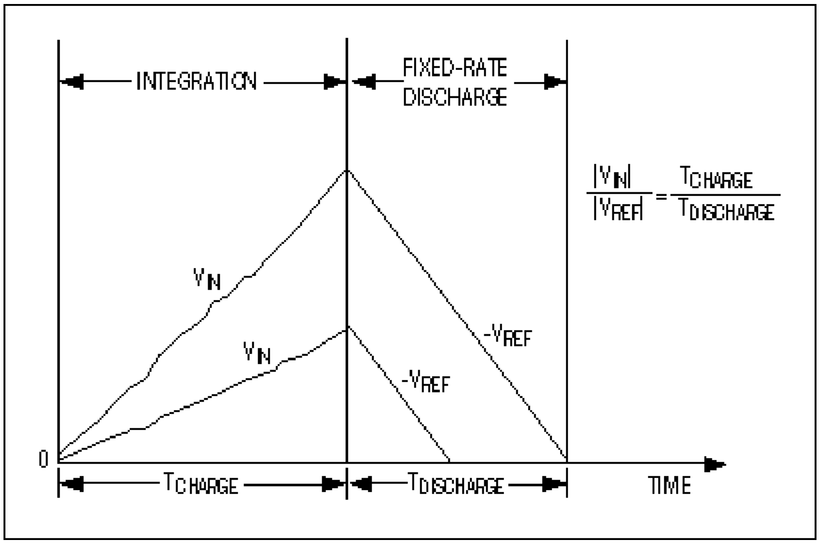
SAR converters sample at rates to 1Msps, draw low supply current, and offer the lowest production cost.

Integrating ADCs

A classical dual-slope converter is shown at the drawing.



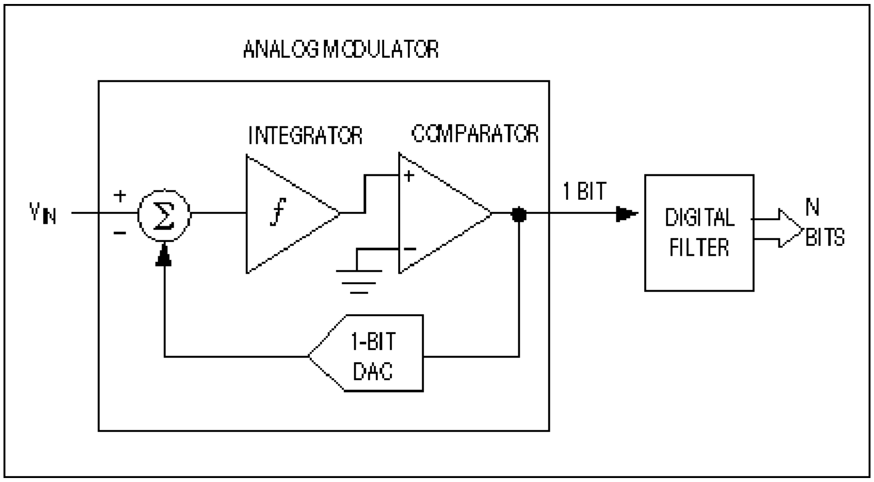
A current, proportional to the input voltage, charges a capacitor for a fixed time interval *Tcharge*. At the end of this interval the device resets its counter and applies an opposite-polarity (negative) reference voltage to the integrator input. With this opposite-polarity signal applied the cap is discharged by a constant current until the voltage at the output of the integrator reaches zero again. The time *Tdischarge* is proportional to the input voltage level and used to enable a counter. The final count provides the digital output, corresponding to the input level.



Note that even the clock frequency does not have to have high stability, because both ramp-up and ramp down time are measured with the same clock.If the clock slows down 10%, the initial ramp will go 10% higher than normal, requiring 10% longer ramp down time resulting in the same final count.
Only the discharge current produced by precise Vref has to be of high stability.
Integrating ADCs are extremely slow devices with low input bandwidths. But their ability to reject high-frequency noise and fixed low frequencies such as 50Hz or 60Hz makes them useful in noisy industrial environments and applications . Provide 10-18 bit resolution. A coversion time for a medium speed 12 bit integrating ADC is about 20mS. This type of ADC is most commonnly used in multi-meters.

Sigma-delta ADCs

Sigma-delta converters , also called oversampling converters, consist of 2 major blocks: modulator and digital filter . The modulator, whose architecture is similar to that of a dual-slope ADC, includes an integrator and a comparator with a feedback loop that contains a 1-bit DAC. The modulator oversamples the input signal, transforming it to a serial bit stream with a frequency well above the required sampling rate. The output filter then converts the bit stream to a sequence of parallel digital words at the sampling rate.
The delta-sigma converters perform high-speed, low resolution (1-bit) A/D conversions, and then remove the resulting high-level quantization noise by passing the signal through analog and digital filters.
Features: high resolution , high accuracy , low noise, low cost.
Good for applications with a bandwidth upto 1MHz, such as speech, audio.

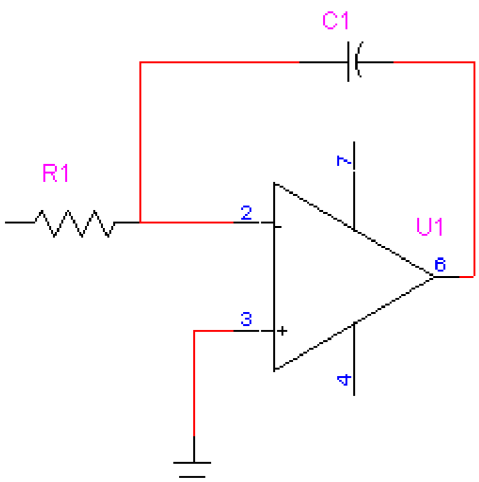


References:
1.The scientist and engineer's guide to digital signal processing,Second Edition,Prentice Hall,1999
2.Horowitz P.,Hill W., The art of electronics , Second edition,1997,Cambridge
3.Websites: www.analog.com , www.national.com , www.maxim.com , www.intersil.com

---

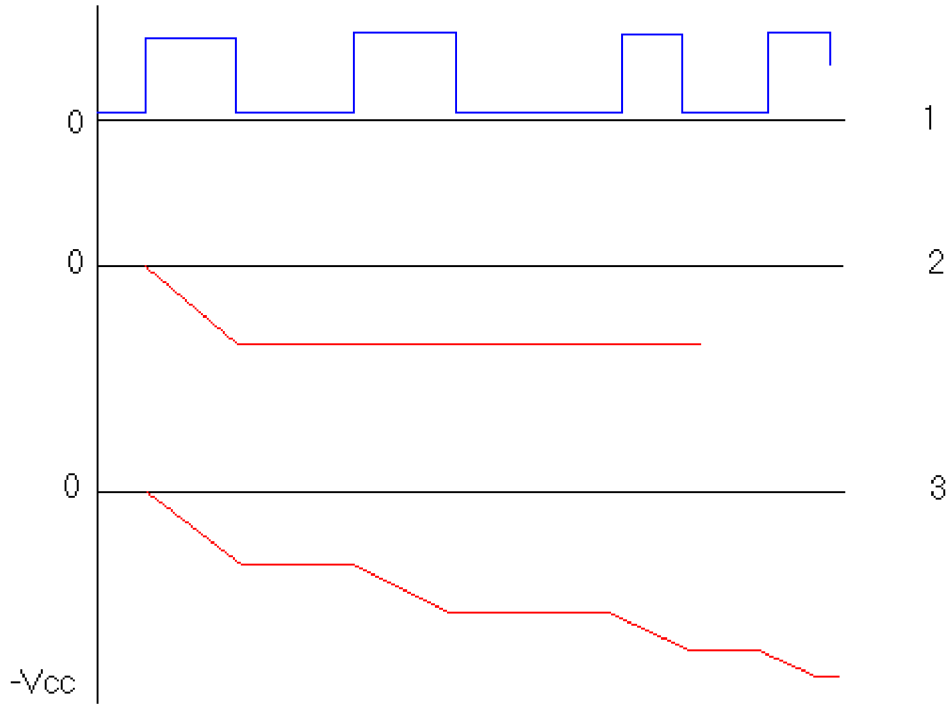
HITEQUEST
Apr 2003

=>
Q:How will the output signal of an ideal integrator look like after
- a positive pulse is applied to the input;
- a series of 10 positive pulses ?



Graph 2 showes the output of integrator after one pulse applied to the input, at the graph

3 the output of integrator changes every new coming pulse until it reaches -Vcc.

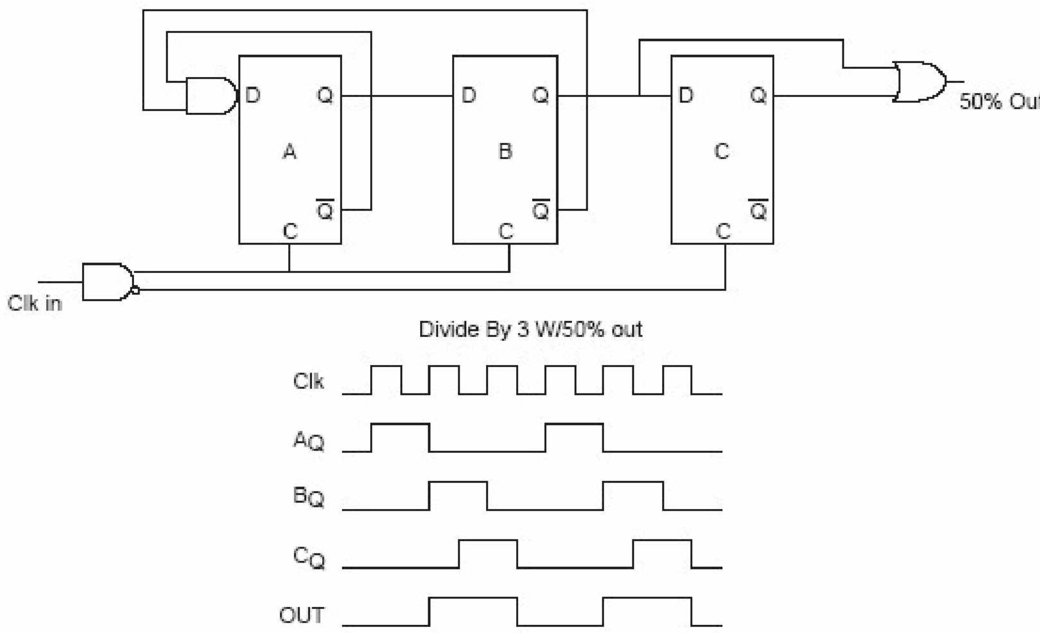


Does the solution for divide by 3 counter by Badri work? I tried it and it is not working for me.

I have attached my solution for divide by-3 circuit which i think is simple enough. This is how the circuit works.
We add a gate on the clock to get differential Clock and Clock bar, a flip flop that triggers on the Clock Bar rising edge (Clock Neg.) to shift the output of "B" by 90 degrees and a gate to OR the outputs of FF "B" and FF "C" to produce the 50% output.

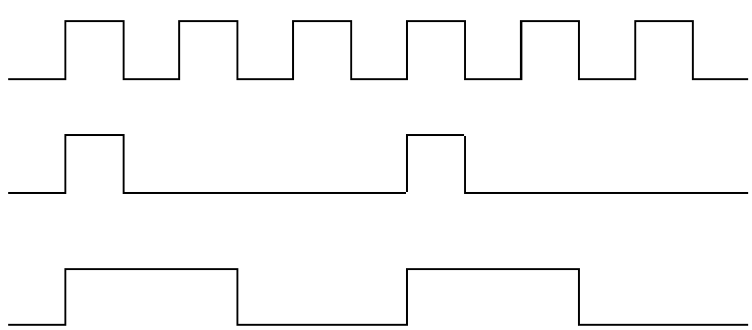
I have attached a .jpg file of the schematic

Satish B.



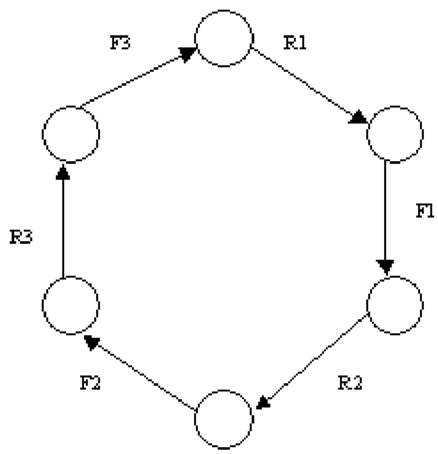
Q: how to design a divide-by-3 counter with equal duty cycle ?

Here is one of the solutions...
Start with the timing diagram. It shows an input clock, an output of a regular divide-by-3 counter and an output of divide-by-3 counter with 50% duty cycle.



It is obvious from the diagram, that we have to use both rising and falling edges of the

input clock.
The next drawing is a state diagram.

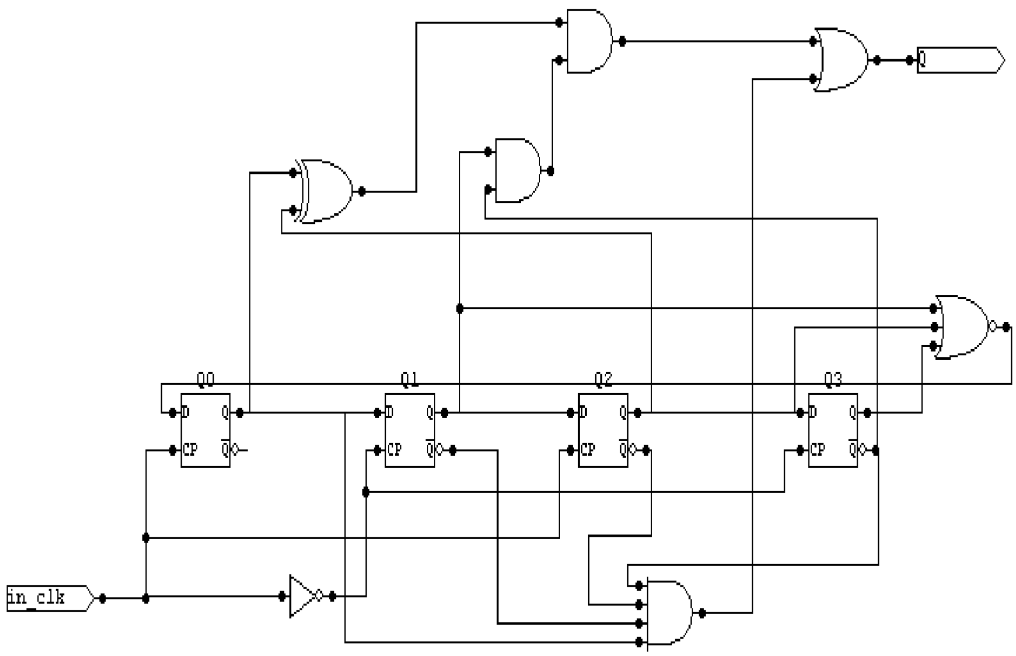


On this diagram R - is a rising edge of input clock, and F - is a falling edge.
How many FF do we need to implement 6 states? At least 3. In this example I am going to use 4 D-type FF just to simplify the design.
Now, look at the table below. Q0 ... Q3 are the outputs of FFs. Q - is the output of the devider with 50% duty cycle. In the first raw the outputs are in the initial state: 0000. In the second raw - the data after the first rising edge and so on.The status of the FFs' ouputs is changing on every rising or falling edge of the input clock according to the information on D-inputs. So, D-inputs should have the data before the clock edge.

in_clk	**Q**	**Q0**	**Q1**	**Q2**	**Q3**	**D0**	**D1**	**D2**	**D3**
R1	1	1	0	0	0	1	1	0	0
F1	1	1	1	0	0	0	1	1	0
R2	1	0	1	1	0	0	0	1	1
F2	0	0	0	1	1	0	0	0	1
R3	0	0	0	0	1	0	0	0	0
F3	0	0	0	0	0	1	0	0	0

These equations are resulting from the table analysis:
D1 = Q0
D2 = Q1
D3 = Q2
D0 = (Q1+Q2+Q3)'
Q = Q0*Q1'*Q2'*Q3'+Q0*Q1*Q2'*Q3'+Q0'*Q1*Q2*Q3' =
Q1*Q3'(Q0*Q2'+Q0'*Q2)+Q0*Q1'*Q2'*Q3'

Now it is the time for the circuit diagram:



This design is not optimal, but illustrates the concept quite well.

Din contributed to this article. Your comments are welcome at eclub@hitequest.com
We will gladly post your design as well.

Q:I swapped transistors in CMOS inverter (put n-transistor at the top and p-transistor at the bottom).
Can I use this circuit as a noninverting buffer?

Yuri M, principal device engineer at Nasional Semiconductor :

My opinion is that the output of non-inverting buffer will follow the input level, but I wouldn't recommend this circuit as a noninverting buffer.
Try to analyse the circuit with the input signal slowly rising from GND to VDD.
Transistors should switch from open to close state. The conditions for fast switching are worse for transistors of the non-inverting buffer. That means longer time interval when both transistors are open which may result in short circuit and possible damage of the transistors.

Q: Convert D-latch into divider by 2.

What is the max clock frequency the circuit can handle ?

T_setup= 6nS
T_hold = 2nS
T_propagation = 10nS

A:

```
    _____________________
   |   _______         |
   |  |     |   Q    |
   `----|D   |----    |
  clock |   |        |
   ----->   |   _  |
      |     |   Q    |
      |     |--------'
      |_______|
```

```
          ___     ___     ___
         |   |   |   |   |   |
clock ___|   |___|   |___|   |___
            _______
           |      |
Q      ____|      |___

       ____        ___
_          |      |
Q          |_______|
```

Any system with clock should meet setup and hold time conditions.
Besides since there is a feedback from !Q to D, we should take care of
D input timing: the data on D input should not change while clock is high!
Otherwise the results are unpredictable.
To meet these conditions:

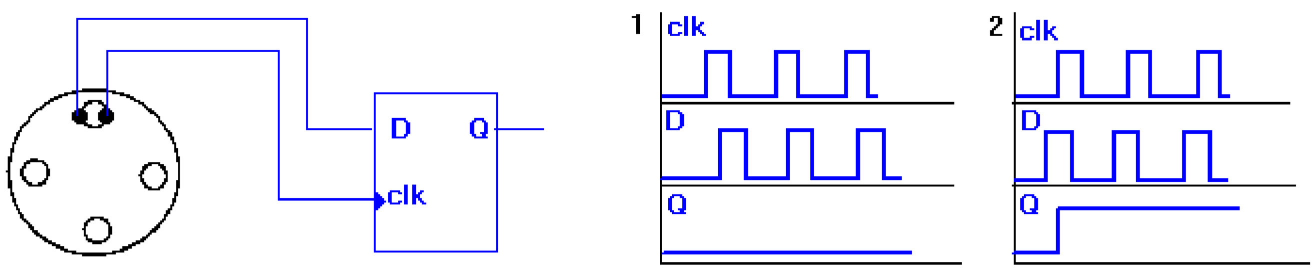
t_clock_high <= T_prop

t_clock_low >= T_setup

```
        T_hold <= T_prop

For example if we take t_clock_high= t_clock_low = 6nS
Then clock period = 12nS,i.e max Freq = 80MHz
```

Q: The circle can rotate clockwise and back. Use minimum hardware to build a circuit to indicate the direction of rotating.

A: 2 sensors are required to find out the direction of rotating.
They are placed like at the drawing. One of them is connected to the data input of D flip-flop,
and a second one - to the clock input. If the circle rotates the way clock sensor sees the light
first while D input (second sensor) is zero - the output of the flip-flop equals zero, and if D input
sensor "fires" first - the output of the flip-flop becomes high.



Q:A device draws higher current when temperature gets:
- higher
- lower

A device draws higher current when temperature gets lower.

=>

When load impedance Z_load equals Z_transmission_line , the impedance of transmission line matches load impedance. The reflection coefficient= 0.
Voltage is devided between Z_source and Z_load.

module fifo1 (full,empty,clk,clkb,ain,bout,rst_N)
output [7:0] bout;
input [7:0] ain;

```
input clk,clkb,rst_N;
output empty, full;
reg [3:0] wptr, rptr;
...

endmodule
```

	0	<-- rptr
	1	
	2	
	3	
	4	
	5	
wptr -->	6	
	7	
	8	
	9	
	10	
	11	
	12	

Multiple clocks add complexity to this design. We need to define conditions for Empty and Full signals, take care of WR and RD pointers. Here is one of the solutions.

Empty and Full flags:

```
assign empty=((wptr == rptr) && (w_flag == r_flag);
assign full=((wptr == rptr) && (w_flag == ~r_flag);
```

where w_flag is set when wptr =12 (end of FIFO). After that wptr is reset to 0. The same for r_flag and rptr.

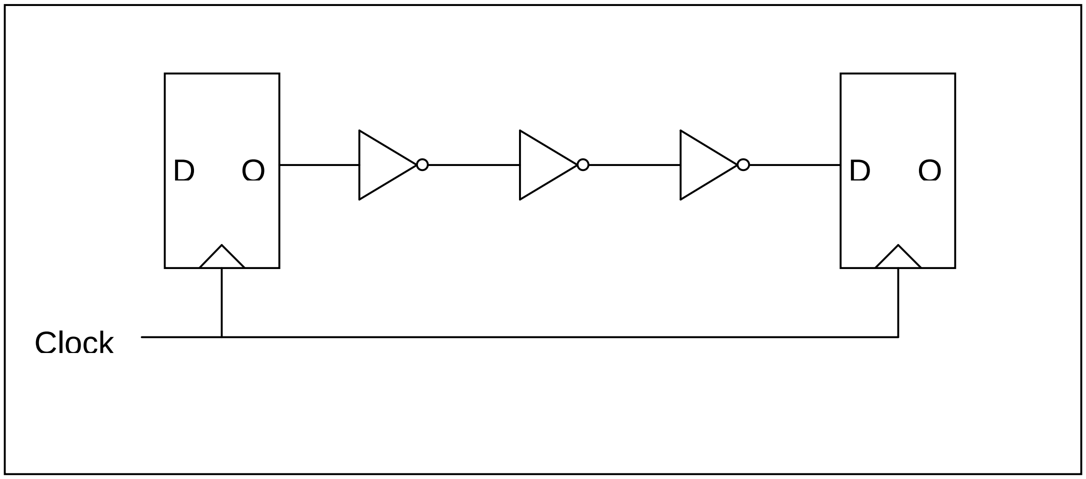
Pointer handling:

```
if (wptr == 12) {w_flag,wptr} <= {~w_flag,4'b0000};
else wptr <= wptr+1;
if (rptr == 12) {r_flag,rptr} <= {~r_flag,4'b0000};
else rptr <= rptr+1;
```

=>

Someone thinks the FIFO should be 16 words deep. Our opinion is that the FIFO should be 32words deep.

1. What is the maximum frequency that the following circuit will operate, given the parameters below, and assuming no wire delay?



Parameters:

inverter propagation delays:
- LH = 150 picoseconds
- HL = 200 picoseconds

flip-flop clock-to-out
- LH = 130 picoseconds
- HL = 170 picoseconds

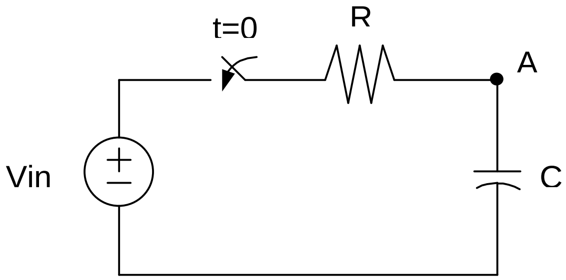
flip-flop setup time
- LH = 90 picoseconds
- HL = 110 picoseconds

flip-flop hold time
- LH = -20 picoseconds
- HL = -30 picoseconds

clock jitter, cycle-to-cycle = 200 picoseconds

clock jitter, long-term = 250 picoseconds

2. Examine the following circuit. Assume the voltage at node A is zero before the switch is closed.



a) at t=0-, what is the current through the resistor?
b) at t=0+, what is the current through the resistor?
c) after a long time, what is the voltage at node A?
d) what is the equation for the voltage at node A as a function of time

4. Add the following unsigned, binary numbers, and provide the answer in binary form.

```
   010110011
   111011101
   011111011
   110101101
+  111011101
------------
   sum = ?
```

6. For the following Karnaugh map, write the logical expression for the output in a <u>minimized</u>, sum-of-products form.

CD \ AB	00	01	11	10
00	1	0	0	1
01	0	1	0	1
11	0	1	1	0
10	1	0	0	1

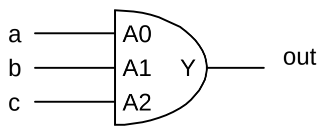
8. Assume you have the following library of generic elements available

```
    AND (Y, A0, A1, ... , An);
    NAND (Y, A0, A1, ... , An);
    INVERT (Y, A0);
    OR (Y, A0, A1, ... , An);
    NOR (Y, A0, A1, ... , An);
    DFLIPFLOP (Y, A0);  (clock is free running, and an
implied connection)
```

where A0, A1, ... through An are inputs (however many are needed for the function), and Y is the output for each element.

Example:

To describe the connections to the simple circuit shown below,



Type = “AND”

use the following code snippet:

```
    AND (out, a, b, c);
```

Problem:

Assume the state table below specifies a sequential circuit.

Current State	Value of outputs y_1y_0 in Current State	Next State if $x = 0$	Next State if $x = 1$
A	00	B	D
B	01	A	C
C	10	D	A
D	11	D	C

Write the text description that will instantiate and connect the necessary elements to implement the sequential machine.

VLSI Questions

1. what types of CMOS memories have you designed? What were their size? Speed? Configuration Process technology?
2. What work have you done on full chip Clock and Power distribution? What process technology and budgets were used?
3. What types of I/O have you designed? What were their size? Speed? Configuration? Voltage requirements? Process technology? What package was used and how did you model the package/system? What parasitic effects were considered?
4. What types of high speed CMOS circuits have you designed?
5. What transistor level design tools are you proficient with? What types of designs were they used on?
6. What products have you designed which have entered high volume production?
7. What was your role in the silicon evaluation/product ramp? What tools did you use?
8. If not into production, how far did you follow the design and why did not you see it into production?

(2) Define Clock jitter and differentiate skew and jitter.How clock jitter effects the system?

2) Jitter can be defined as a cause that varies the period from cycle to cycle where as Clock Skew can be defined as difference in arrival time of clock signal between two sequentially adjacent registers. Clock skew can be positive or negative, depending on which clock lags. Jitter makes the system unstable.

Q. Design a logic which mimics a infinite width register. It takes input serially 1 bit at a time. Output is asserted high when this register holds a value which is divisible by 5.

For example:

Input	Sequence	Value	Output
1	1	1	0
0	10	2	0
1	101	5	1
0	1010	10	1
1	10101	21	0

(Hint: Use a FSM to create this)

Q.Design a block which has 3 inputs as followed.

1. system clock of pretty high freq
2. asynch clock input P
3. asynch clock input Q

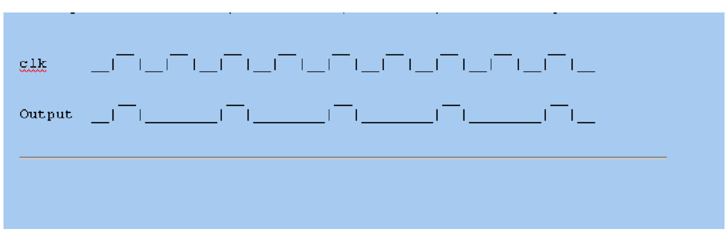
P and Q clocks have 50% duty cycle each. Their frequencies are close enough and they have phase difference. Design the block to generate these outputs.

1. PeqQ : goes high if periods of P and Q are same
2. PleQ : goes high if P's period is less than that of Q.
3. PgrQ : goes high if P's period is greater than that of Q.

Q. What's the difference between a latch and a flip-flop? Write Verilog RTL code for each. (This is one of the most common questions but still some EE's don't know how to explain it correctly!)

Q. Design a black box whose input clock and output relationship as shown in diagram.

```
           __    __    __    __    __    __    __    __
__
clk      __|  |__|  |__|  |__|  |__|  |__|  |__|  |__|  |
|__|  |__

           __          __          __          __
__
Output   __|  |_________|  |_________|  |_________|
|_________|  |__
```



Q. Design a digital circuit to delay the negative edge of the input
signal by 2 clock cycles.

```
                    ______________________
  input     ________|                      |______________
             _    _    _    _    _    _    _    _    _    _    _    _
_
  clock   _| |_| |_| |_| |_| |_| |_| |_| |_| |_| |_| |_| |_|
|_| |_
                   ______________________________
  output  __________|                              |____________
```

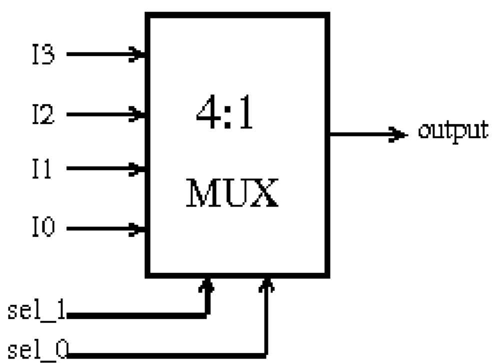
Q. Design a Pattern matching block

- Output is asserted if pattern "101" is detected in last 4 inputs.
- How will you modify this design if it is required to detect same "101" pattern anywhere in last
8 samples?

Questions:

---

1. Design a 4:1 mux in Verilog.



- Multiple styles of coding. e.g.
  Using **if-else** statements

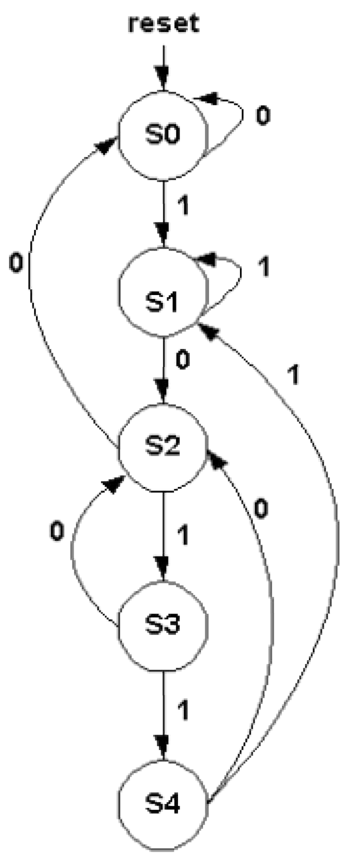
  if(sel_1 == 0 && sel_0 == 0) output = I0;
  else if(sel_1 == 0 && sel_0 == 1) output = I1;
  else if(sel_1 == 1 && sel_0 == 0) output = I2;
  else if(sel_1 == 1 && sel_0 == 1) output = I3;

Using **case** statement

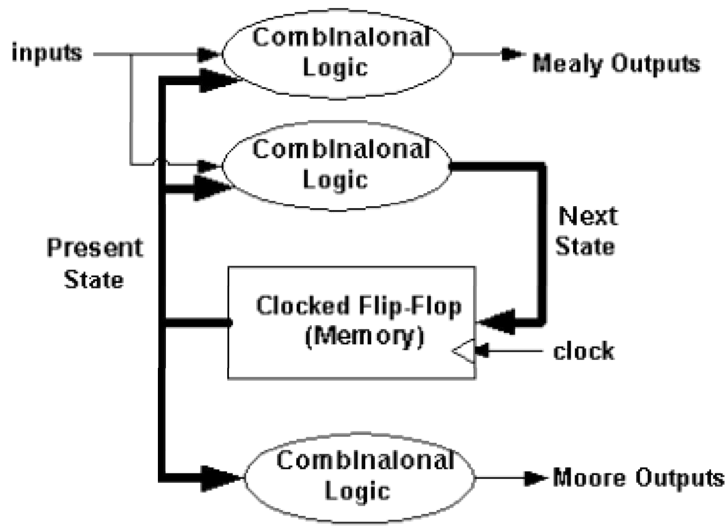
```
case ({sel_1, sel_0})
     00 : output = I0;
     01 : output = I1;
     10 : output = I2;
     11 : output = I3;
     default : output = I0;
endcase
```

- What are the advantages / disadvantages of each coding style shown above?
- How Synthesis tool will give result for above codes?
- What happens if default statement is removed in case statement?
- What happens if combination 11 and default statement is removed? (Hint Latch inference)
  (Comments : Though this questions looks simple and out of text books, the answers to supporting questions can come only after some experience / experimentation

Design a FSM (Finite State Machine) to detect a sequence 10110.



State Transition Diagram



FSM Block Diagram

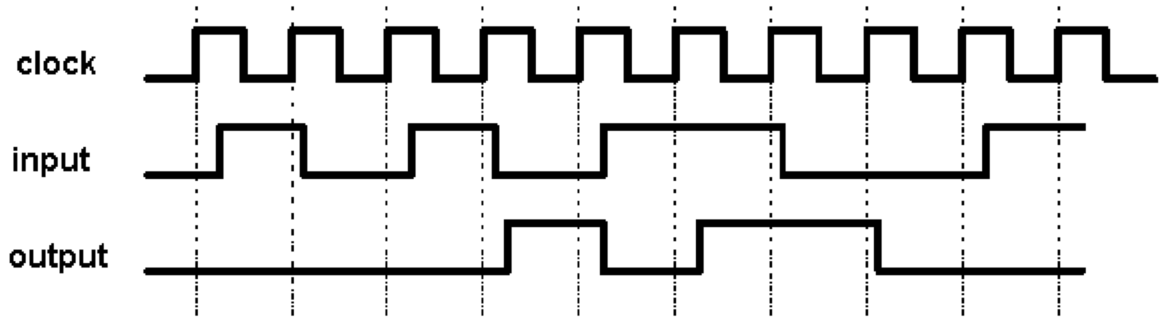
- Have a good approach to solve the design problem.
- Know the difference between Mealy, Moore, 1-Hot type of state encoding.
- Each state should have output transitions for all combinations of inputs.
- All states make transition to appropriate states and not to default if sequence is broken. e.g. S3 makes transition to S2 in example shown.
- Take help of FSM block diagram to write Verilog code.

---

3. One more sequence detector:

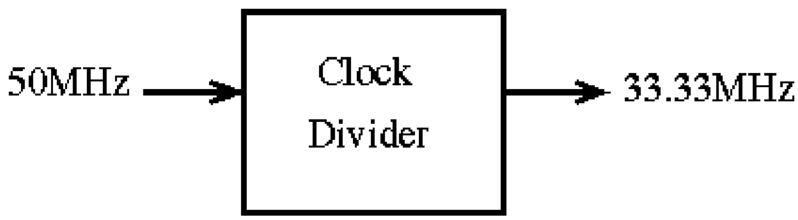
Design a FSM (Finite State Machine) to detect more than one "1"s in last 3 samples.
For example: If the input sampled at clock edges is 0 1 0 1 0 1 1 0 0 1
then output should be 0 0 0 1 0 1 1 1 0 0 as shown in timing diagram.

And yes, you have to design this FSM using not more than 4 states!!



---

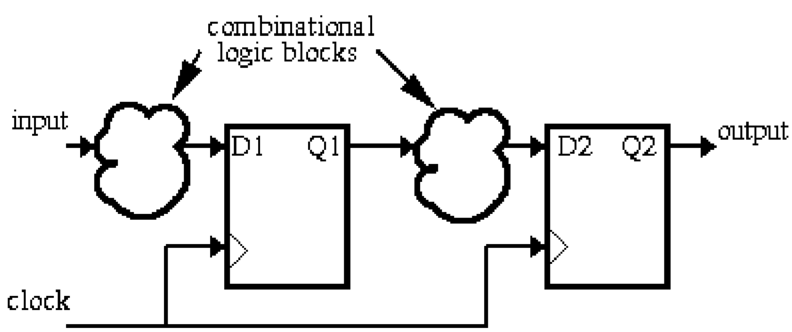
4. Design a state machine to divide the clock by 3/2.



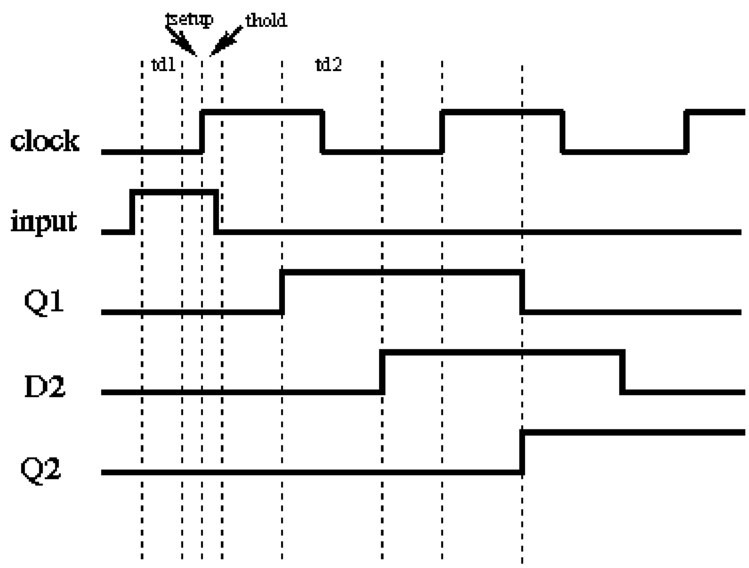
(Hint: 2 FSMs working on posedge and negedge)

---

5. Draw timing diagrams for following circuit.



Schematic of two cascaded flip–flops

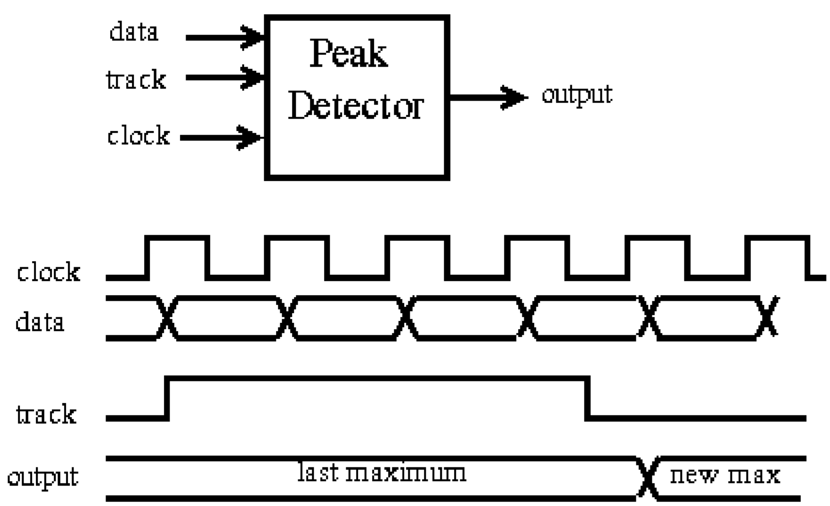


Timing Diagram

- What is the maximum frequency at which this circuit can operate?
- What is the minimum width of input pulse and position?
- Problem can be given interesting twist by specifying all delays in min and max types.

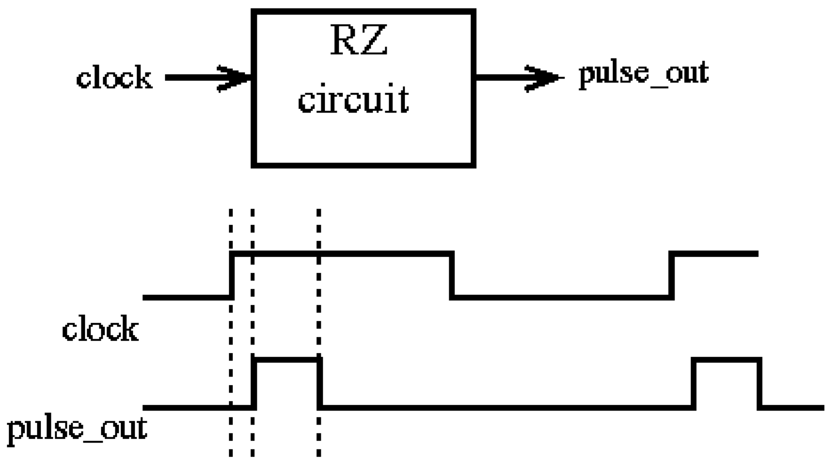
---

6. Design a Digital Peak Detector in Verilog.



---

7. Design a RZ (return to zero)circuit. Design a clock to pulse circuit in Verilog / hardware gates.



---

8. Miscellaneous Basic Verilog Questions:

- What is the difference between Behavior modeling and RTL modeling?
- What is the benefit of using Behavior modeling style over RTL modeling?
- What is the difference between blocking assignments and non-blocking assignments ?
- How do you implement the bi-directional ports in Verilog HDL
- How to model inertial and transport delay using Verilog?
- How to synchronize control signals and data between two different clock domains?

) Give two ways of converting a two input NAND gate to an inverter

2) Given a circuit, draw its exact timing response. (I was given a Pseudo Random Signal Generator; you can expect any sequential ckt)

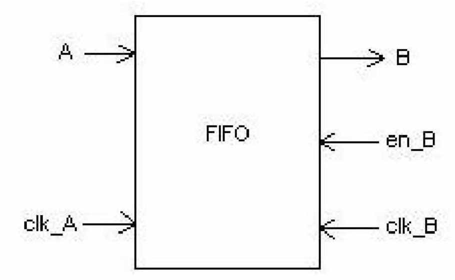
3) What are set up time & hold time constraints? What do they signify? Which one is critical for estimating maximum clock frequency of a circuit?

4) Give a circuit to divide frequency of clock cycle by two

5) Design a divide-by-3 sequential circuit with 50% duty circle. (Hint: Double the Clock)

6) Suppose you have a combinational circuit between two registers driven by a clock. What will you do if the delay of the combinational circuit is greater than your

Q: Given the following FIFO and rules, how deep does the FIFO need to be to
prevent underflowing or overflowing?



RULES:
1) frequency(clk_A) = frequency(clk_B) / 4
2) period(en_B) = period(clk_A) * 100
3) duty_cycle(en_B) = 25%

A:

Assume clk_B = 100MHz (10ns)

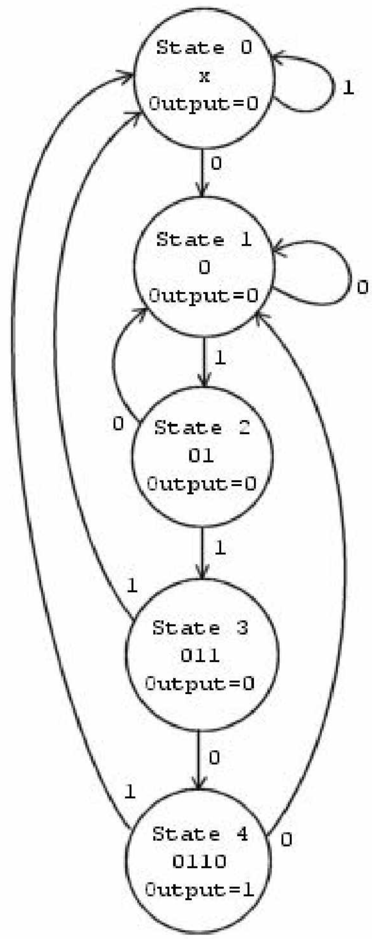
From (1), clk_A = 25MHz (40ns)

From (2), period(en_B) = 40ns * 400 = 4000ns, but we only output for 1000ns, due to (3), so 3000ns of the enable we are doing no output work.Therefore, FIFO size = 3000ns/40ns = 75 entries.

General Answer 2

Q: Draw the state diagram to output a "1" for one cycle
if the sequence "0110" shows up (the leading 0s cannot be
used in more than one sequence).

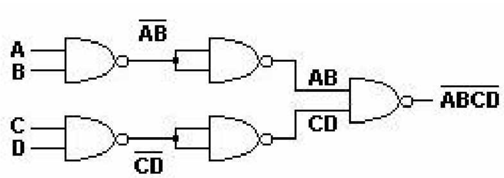
A:



General Answer 3

Q: Design a four-input NAND gate using only two-input NAND gates.

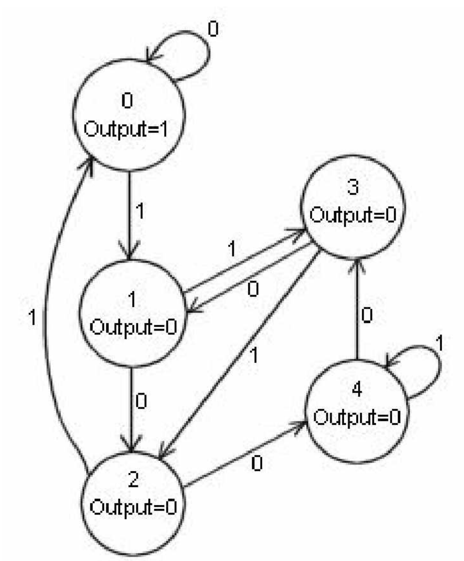
A: Basically, you can tie the inputs of a NAND gate together to get an inverter, so...



Q: Draw the state diagram for a circuit that outputs a "1" if the aggregate serial binary input is divisible by 5. For instance, if the input stream is 1, 0, 1, we output a "1" (since 101 is 5). If we then get a "0", the aggregate total is 10, so we output another "1" (and so on).

A: We don't need to keep track of the entire string of numbers - if something is divisible by 5, it doesn't matter if it's 250 or 0, so we can just reset to 0.

So we really only need to keep track of "0" through "4".



7) The answer to the above question is breaking the combinational circuit and pipelining it. What will be affected if you do this?

8) What are the different Adder circuits you studied?

9) Give the truth table for a Half Adder. Give a gate level implementation of the same.

10) Draw a Transmission Gate-based D-Latch.

11) Design a Transmission Gate based XOR. Now, how do you convert it to XNOR? (Without inverting the output)

12) How do you detect if two 8-bit signals are same?

13) How do you detect a sequence of "1101" arriving serially from a signal line?

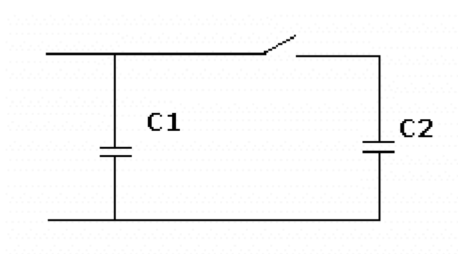
14) Design any FSM in VHDL or Verilog.

1. Have you studied buses? What types?
2. Have you studied pipelining? List the 5 stages of a 5 stage pipeline. Assuming 1 clock per stage, what is the latency of an instruction in a 5 stage machine? What is the throughput of this machine ?
3. How many bit combinations are there in a byte?
4. For a single computer processor computer system, what is the purpose of a processor cache and describe its operation?
5. Explain the operation considering a two processor computer system with a cache for each processor.
6. What are the main issues associated with multiprocessor caches and how might you solve them?
7. Explain the difference between write through and write back cache.
8. Are you familiar with the term MESI?
9. Are you familiar with the term snooping?
10. Describe a finite state machine that will detect three consecutive coin tosses (of one coin) that results in heads.
11. In what cases do you need to double clock a signal before presenting it to a synchronous state machine?
12. You have a driver that drives a long signal & connects to an input device. At the input device there is either overshoot, undershoot or signal threshold violations, what can be done to correct this problem?
13. What are the total number of lines written by you in C/C++? What is the most complicated/valuable program written in C/C++?
14. What compiler was used?
15. What is the difference between = and == in C?
16. Are you familiar with VHDL and/or Verilog?
17. What types of CMOS memories have you designed? What were their size? Speed?
18. What work have you done on full chip Clock and Power distribution? What process technology and budgets were used?
19. What types of I/O have you designed? What were their size? Speed? Configuration? Voltage requirements?
20. Process technology? What package was used and how did you model the package/system? What parasitic effects were considered?
21. What types of high speed CMOS circuits have you designed?
22. What transistor level design tools are you proficient with? What types of designs were they used on?

23. What products have you designed which have entered high volume production?
24. What was your role in the silicon evaluation/product ramp? What tools did you use?
25. If not into production, how far did you follow the design and why did not you see it into production?

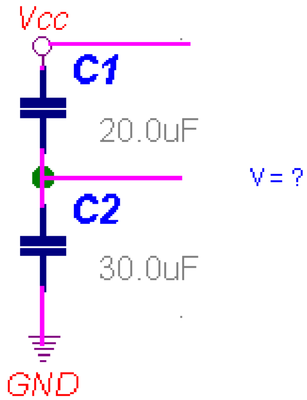
1. What types of CMOS memories have you designed? What were their size? Speed? Configuration Process technology?
2. What work have you done on full chip Clock and Power distribution? What process technology and budgets were used?
3. What types of I/O have you designed? What were their size? Speed? Configuration? Voltage requirements? Process technology? What package was used and how did you model the package/system? What parasitic effects were considered?
4. What types of high speed CMOS circuits have you designed?
5. What transistor level design tools are you proficient with? What types of designs were they used on?
6. What products have you designed which have entered high volume production?
7. What was your role in the silicon evaluation/product ramp? What tools did you use?
8. If not into production, how far did you follow the design and why did not you see it into production?

- Two capacitors are connected in parallel through a switch. C1= 1uF, C2= 0.25uF. Initially switch is open,C1 is charged to 10V. What happens if we close the switch?No loss in the wires and capacitors.



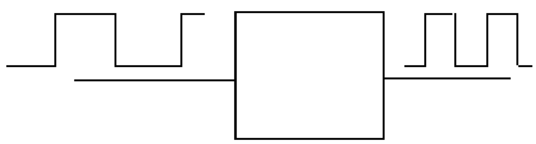
- You have 2 switches to control the light in the long corridor. You want to be able to turn the light on entering the corridor and turn it off at the other end. Do the wiring circuit.
- This question is based on the previous one, but there are 3 switches that can turn on and off a light in the room. How to wire them up?

- What will be the voltage level between the 2 capacitors? The Vcc = 10v DC.
Sent by Tanh, VLSI engineer

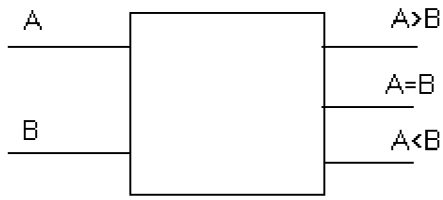


Hint from Hitequest

- Suppose, you work on a specification for a system with some digital parameters. Each parameter has Min,Typ and Max colomns. In what column would you put a Setup time and a Hold time?
- Design a simple circuit based on combinational logic to double the output frequency.



- 8bit ADC with parallel output converts input signal into digital numbers. You have to come up with the idea of a circuit , that finds MAX of every 10 numbers at the output of the ADC.
- Implement comparator that compares two 2-bit numbers A and B. The comparator should have 3 outputs: A > B, A < B, A = B. Do it two ways:
- using combinational logic;
- using multiplexers. Write HDL code for your schematic at RTL and gate level.



- You have 8 bit ADC clocking data out every 1mS. Design a system that will sort the output data and keep a statistics how often each binary number appears at the output of ADC.
  What types of flip-flops do you know?

- Implement D- latch from
  - RS flip flop;
  - multiplexer.
- How to convert D-latch into JK-latch and JK-latch into D-latch?

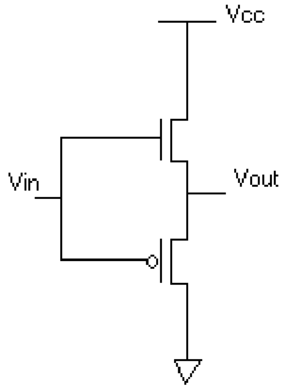
- You have two counters to 16, built from negedge D- FF . First circuit is synchronous and second is "ripple" (cascading). Which circuit has a less propagation delay?
- what is the difference between flip-flop and latch?
  Write an HDL code for their behavioral models.

  Describe the operation of DAC? What are the most important parameters of DAC? Do we really need both INL and DNL to estimate linearity?

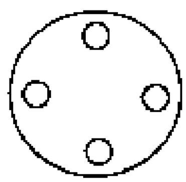
- Compare briefly all types of ADC,that you know .
- How will the output signal of an ideal integrator look like after
  - a positive pulse is applied to the input;
  - a series of 10 positive pulses ?
- How to design a divide-by-3 counter with equal duty cycle ?
  question from Anonymous

- For an 8-bit flash A/D converter with an input range from 0V to 2.55V, describe what happens when the input voltage changes from 1.27V to 1.28V
- Your system has CPU, ALU and two 8bit registers. There is no external memory. Can you swap the contence of the registers ?

- I swapped 2 transistors in CMOS inverter (put n-transistor at the top and p-transistor at the bottom). Can this circuit work as a noninverting buffer? (By E.Martovetsky,design eng from Transmeta)



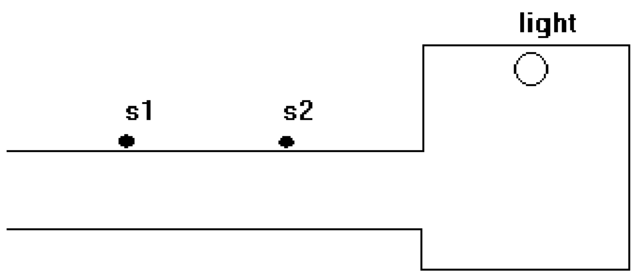
- Convert D-latch into divider by 2.
  What is the max clock frequency the circuit can handle ?
  T_setup= 6nS
  T_hold = 2nS
  T_propagation = 10nS
- The circle can rotate clockwise and back. Use minimum hardware to build a circuit to indicate the direction of rotating.



- For ATE engineers (semiconductor test):
  Draw a shmoo plot of two parameters: Clock period Tclk and setup time Tsetup.
- For chip design/test/product engineers :
  An IC device draws higher current when temperature gets:
  - higher
  - lower

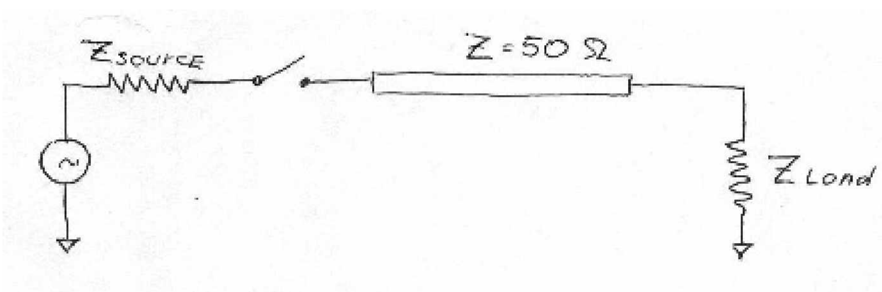
Hint from Hitequest

- enter the office people have to pass through the corridor. Once someone gets into the office the light turns on. It goes off when noone is present in the room. There are two registration sensors in the corridor. Build a state machine diagram and design a circuit to control the light.



- A voltage source with internal impedance Z_source = 50 OHm is connected to a transmission line with Z = 50 OHm. Z_load is also 50 OHm.
  The voltage source generates a single voltage step 1V.
  What will be the voltage level on the load:

  a) 2V , because the reflected signal will be in-phase with the incident signal;
  b) 0.33V , because the voltage is devided between Z_source , Z_load and Z_transm.line;
  c) 0.5V , because the voltage is devided between Z_source and Z_load.



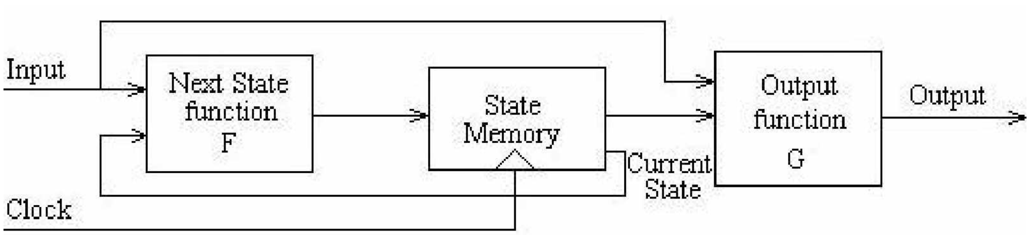
- What does CMOS stand for? VLSI? ASIC?
  This was in the series of quick questions in the interview at Analog Devices. We use these abbreviations daily, but not everyone remembers what they stand for.
- Design a COMBINATIONAL circuit that can divide the clock frequency by 2.
- Design a 2bit up/down counter with clear using gates. (No verilog or vhdl)

we have a circular wheel with half painted black and the other half painted white. There are 2 censors mounted 45 degree apart at the surface of this wheel(not touching the wheel) which give a "1" for black and "0" for white passing under them. Design a circuit to detect which way the wheel is moving. Can not assume any fixed position for start.

We have a fifo which clocks data in at 100mhz and clocks data out at 80mhz. On the input there is only 80 data in any order during each 100 clocks. In other words, a 100 input clock will carry only 80 data and the other twenty clocks carry no data (data is scattered in any order). How big the fifo needs to be to avoid data over/under-run.

- Instead of specifying SETUP and HOLD time, can we just specify a SETUP time for '1' and a SETUP time for '0'?
- Here some hardware digital design specific questions, offered by Suhas:
  (1) When will you use a latch and a flipflop in a sequential design?
  (2) Design a 1-bit fulladder using a decoder and 2 "or" gates?
  (3) You have a circuit operating at 20 MHz and 5 volt supply. What would you do to reduce the power consumption in the circuit- reduce the operating frequency of 20Mhz or reduce the power supply of 5Volts and why?
  (4) In a nmos transistor, how does the current flows from drain to source in saturation region when the channel is pinched off?
  (5) In a SRAM circuit, how do you design the precharge and how do you size it?
  (6) In a PLL, what elements(like XOR gates or Flipflops) can be used to design the phase detector?
  (7) While synthesis of a design using synopsys design compiler, why do you specify input and output delays?
  (8) What difference do you see in the timing reports for a propogated clock and an ideal clock?
  (9) What is timeborrowing related to Static timing anaylsis in Primetime?

A clocked synchronous state machine is built as shown below.



The next state logic F is a combinational circuit with a maximum propagation delay of tF. The state memory is an edge-triggered register with a minimum clock-to-output propagation delay of 0 and a maximum propagation delay of tR, a setup time of tS, and a hold time of tH. The out put logic G is a combinational circuit with maximum propagation delay of tG.

What is the maximum clock frequency at which the machine will cycle through its states properly?
What is the minimum propagation delay required for the proper operation of the next state logic F?
What is the maximum delay from triggering clock edge until the output is valid?
Is this a Mealy or a Moore finite state machine?

1. Draw state diagrams and verify JK, D and T flip-flops.

2. Design and verify, the mode controlled mod – 8 UP/DOWN synchronous counter. (M = 0, UP AND M =1. DOWN)

3. Design a parity generator. This block accepts a serial data stream. It gives out a data stream as it is and also gives an output indication of parity of all bits it has received after the de assertion of reset.

4. Information bits are encoded on single line x so as to synchronize with a clock. Bits are encoded so that two or more consecutive 1s or 4 or more consecutive 0s should never appear on line x. An error indicating sequential circuit is to be designed to indicate an error by generating 1 on output z coinciding with the 4^{th} of every sequence of four zeros or the second of every sequence of two 1s. e.g. if two consecutive 1s appear, the output remains 1 for the second and subsequent clock cycles till bit 1 is received. Design such an error detector.

5. Design a circuit, which samples an input line on each rising edge of clock. This system is to take four samples before it returns to its initial state. Further, this system is only to initiate an output if the 0111 sequence is detected. Make certain your system returns to the initial state on the rising edge of the fourth clock.

6. Design non-overlapped and overlapped sequence detector to detect a sequence 1010 of serial bits, outputting '1' when the sequence is detected with Moore and mealy machine styles.

7. Design a synchronous circuit that has single input x and single output z. The input data is received serially. The first output bit after reset should be the same as the first input bit. After that the output should change only when three consecutive input bits have the same value.

8. Design a sequential machine, that will assert output Z if and only if any of the two input sequences 1010 or 0010 occur on a serial input X. Assume that the serial input is synchronous.

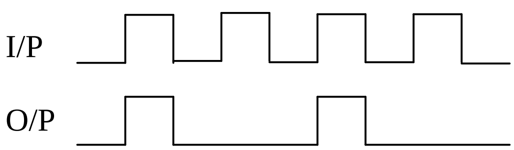
9. Generate a control signal Z, having word time equal to 4 clock pulses as shown. Design the circuit as mealy machine. External input is assumed to be x, x = '1' will be a command to generate Z.

10. A finite string recognizer has one input X and one input Z. The output is asserted whenever the input sequence…..010….has been observed, as long as the sequence 100 has never been seen.

11. Construct, the state diagram for a machine which is to produce an output y =1 whether the last string of five inputs contain exactly three 1s and the string starts with two 1s , analysis of the next string will not start until the end of this string of five, whether it produces 1 output or not.

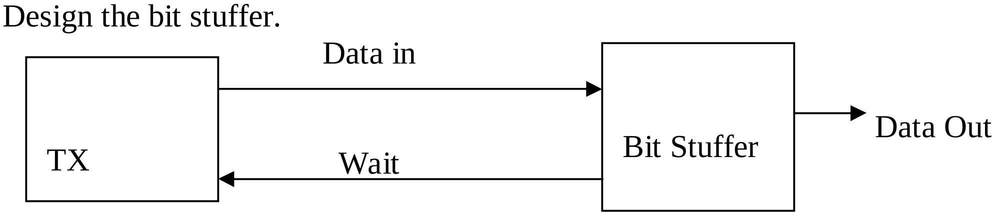
12. Design a FSM, which monitors an input data line four consecutive times and displays three – bit output Y2 Y1 Y0 after the fourth sample. The output is to be binary equivalent of the no of 1 sample taken in the four – sample sequence.

13. Design a pulse gulping circuit as shown below.



14. Design the FSM for the following specifications. The circuit has an input x and two outputs Z1 and Z2. The circuit consists of two concurrent mealy machines. The Z1 output becomes '1' when a "1011" sequence is found on the input and Z2 becomes '1' when "110" sequence is found on the input X.

15. Design the bit stuffer.



It accepts a serial bit stream. The data, which is received serially, is given out as a serial bit stream. If there are more than 5 consecutive 1s or five consecutive 0s, then an extra 0 or 1, respectively is inserted into the output stream following the five consecutive bits. Whenever an extra bit is inserted in the output stream, a signal wait is asserted for one bit period. The entity TX generates the bit stream using a shift register. It will use the wait signal as a shift enable. The bit stuffer should generate a wait signal such that no data is missed due to the extra stuffed bits. The design has to be purely synchronous.

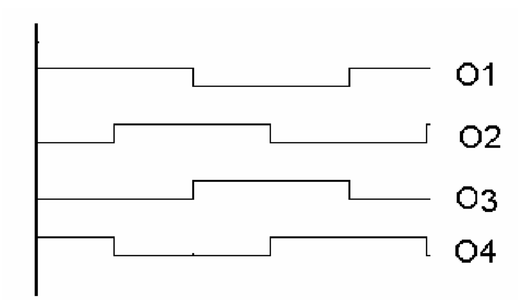
1. Generate a control signal Z,
   a) Which is high for 4 clock pulses when X (serial command input) goes from low to high.

b) Which is high for one clock cycle, low for two clock cycle and high for one clock cycle when X (serial command input) goes from high to low.

Note: Command signal X going from low to high overrides (in priority) the command Signal X going from high to low ie. Control signal generation initiated by high to low command can be interrupted in between by low to high command to start new pulse as in (a) and not vice-versa.

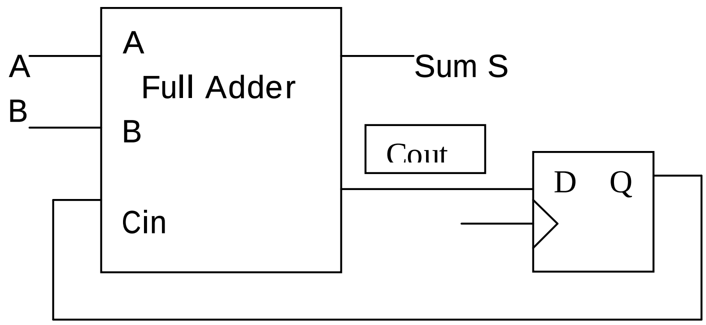
2. Design a sequential machine which will generate output "Match", which is asserted active high only if any of the three input sequences 0110 or 1001 or 0101 occur on a serial input. System has to take 4 samples before it returns to initial stage Assume that the serial input is synchronous.

3. Design a stepper motor sequence generator. The circuit has 4 outputs 01, 02, 03, 04. Refer to the timing diagram below. The circuit accepts 2 inputs CW & CCW (clk-wise & counter – clk – wise) if both are asserted or de asserted, then motor holds its position i.e., o/p maintain their current value.

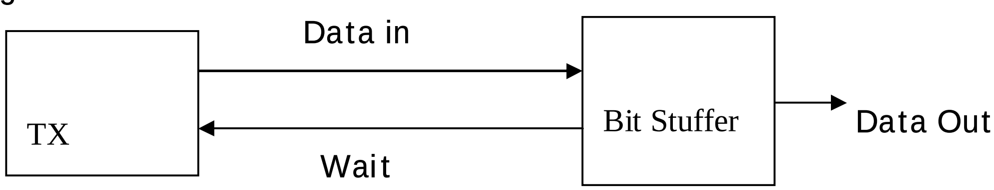


4. There are 3 entities (en1, en2, en3), trying to gain control of one bus. Before they use the bus, they have to give a request to the bus controller and they can use the bus only when they get a grant from the controller. The entity driving the bus should keep its request high as long as it wants to use the bus. Once its request becomes zero, the bus will be released. Once a grant has been given to any entity, all other requests will be ignored until the request of the entity driving the bus is removed. The bus controller gives highest priority to en1 & lowest priority to en3 in case of contention.

5. Draw the state machines for the following circuit. And verify output sequence for first 5 clock signals.



6. Design the bit stuffer.



It accepts a serial bit stream. The data, which is received serially, is given out as a serial bit stream.

If there are more than two 1s i.e. if 3rd '1' is detected on 3rd rising clock edge, then , in the place of this 3 '1' , bits "01" or "00" are stuffed alternately (e.g. this time for 3rd '1' detected, "01" is stuffed, next time if 3rd '1' is detected, "00" is stuffed..and next time again "01" and so on...) in the output stream for 3rd and 4th clock pulse respectively.

If there are more than two 0s i.e. if 3rd '0' is detected on 3rd rising clock edge, then , in the place of this 3 '0' , bits "10" or "11" are stuffed alternately (e.g. this time for 3rd '0' detected, "10" is stuffed, next time if 3rd '0' is detected, "11" is stuffed..and next time again "10" and so on...) in the output stream for 3rd and 4th clock pulse respectively.

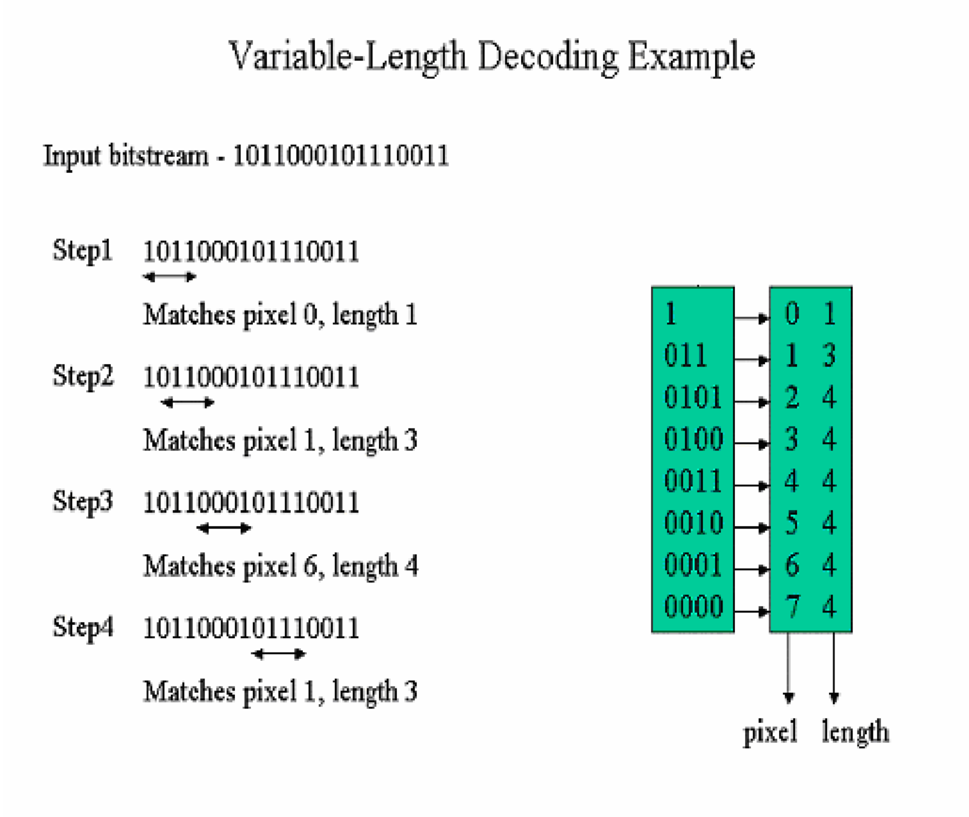
Whenever these extra 2 bits are inserted in the output stream, a signal wait is asserted for two bits period. The entity TX generates the bit stream using a shift register. It will use the wait signal as a shift enable. The bit stuffer should generate a wait signal such that no data is missed due to the extra stuffed bits. The design has to be purely synchronous.

7. Design a state m/c for the following specification. M/c has single synchronous input X and serial output Z. After every sample of two bits it asserts or de-asserts the output as follows:
    i. For pattern "11" or "00" -> o/p toggles as long as pattern "01" is not seen on the serial input else o/p remains same as previous o/p.
    ii. For pattern "01" -> o/p resets.
    iii. For pattern "10" -> o/p sets.
    iv. Between two bits there is no change in the output.

   Note: m/c checks for the next pattern after receiving every two bits only.

   Sample I/O bit pattern: X – 00 10 00 10 00 11 01 11 00 10 01
   Z – 01 11 10 01 10 01 10 00 00 01 10
8. A system receives input bit stream containing variable length bit patterns one after another. The figure below shows 8 bit patterns that the input bit stream may contain in any order. On detection of any of these patterns (in continuously receiving input bit stream) respective 2 integer outputs named pixel and length are to be generated as shown in the table.

   Devise a scheme with detailed architecture for such a system.



9. A protocol demands the transfer of data from the host to receiver in the form of packets where by there has to be a distinct signal for start of packet (SOP) to identify the beginning of a packet and a distinct signal for the end of Packet (EOP) to signify the end of packet. Data between these two signals (SOP & EOP) is the clock information for the receiver to synchronize to the host clock for correct sampling of data. The end of this clock signal is a distinct bit pattern called ENDOFSYNC (end of synchronization field - "KK").
   There are three signals X, Y & Z which are the only means of communication between the HOST and the receiver.
   There are three states on the input lines (X, Y, Z), namely:
   J-state – X='1', Y='1', Z='0' (idle state)
   K-state – X='0', Y='0', Z='1'
   Bus low state (L) – X='0', Y='0', Z='0'
   Assume: Unless and until you get a SOP signal (J-state to K-state) the X, Y, Z line remains in the J-state.

Following are the signal conditions for X, Y and Z on the bus for identification of EOP, SOP, ENDSYNC, IDLE and RESET conditions on the receiver side.
Use these conditions to design a state m/c to generate the above mentions five signals …

a. idle (J): Whenever machine goes from any state to idle state(J-state),
initial state), then the idle signal will be asserted stating the bus is in the idle state.
b. SOP (JK): 1st cycle => 110 (J-state), 2nd cycle => 001 (K-state).
c. ENDSYNC (KJKK): K-state à J- state -> (K-state-> J-state)…any times
repeat -> K-state-> K-state

JJJJJJ K JK(JKJK…)KK LLJ JJJJJJ
Idle………SOP…sync field…EOP… idle…….
JJJJJJJJ K JK LLLL JJJJJJJJJJJJ
Idle………….. SOP…sync…reset…idle……….

(Two continuous "KK" state is the end of synchronization field. Before K-K state sequence, at least once K-J state sequence should have appeared…then ENDSYNC will be asserted and machine will remain in the same state with ENDSYNC asserted waiting for RESET OR EOP before going back to idle state and then de-assert the ENDSYNC).
d. EOP (LLJ): 1st cycle => 000, 2nd cycle => 000, 3rd cycle=> 110(J-state/idle state).

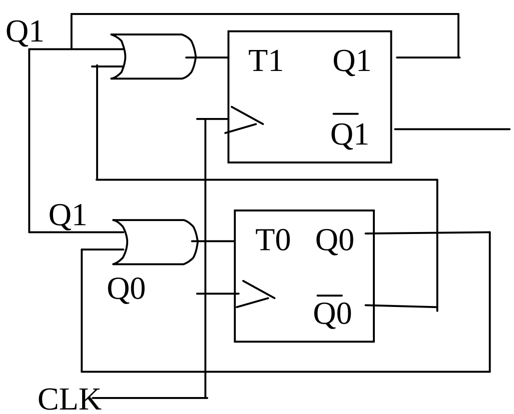
e. Reset (LLLL): XYZ is 000 for 4 clock cycles and on the fourth clock edge
machine goes to idle state asserting reset and idle … both the signals. Assume the reset (XYZ="000") can come at any of the states

10.NRZI ENCODER

11.MANCHESTOR ENCODING

12.ROUND ROBBIN

13.Draw state diagram for following sequential circuit and write the output sequence for 5 clock pulses. The flip-flops used are T-FFs.



1. **What's a PCB?** Printed Circuit Board.
2. **Name some band definitions**. L-Band - 1-2 GHz, S-Band - 2-4 GHz, C-Band - 4-8 GHz, X-Band - 8-12 GHz, Ku-Band - 12-18 GHz.
3. **What's the definition of gain?** The signal going into the device is smaller than the signal going out.
4. **What's a definition of an amplifier?** It's a device that exhibits gain.
5. **Insertion loss?** The signal going in is greater than the signal that's coming out.
6. **Thermal impedance?** A measure of how hot the device gets with varying electrical input.
7. **What's the relation between dB and change of signal?** +3 dB is 2 times bigger, +10 dB is 10 times bigger.
8. **What's an octave?** The bandwidth characterized by higher band being twice as much as lower band (100-200 MHz).
9. **What's a decade?** A bandwidth, where the higher band is 10x the lower band (100 MHz - 1 GHz).

10. **Skin effect?** The signal is reported on the surface of the object, but not inside, e.g. a solid brick wall.
11. **VSWR?** Voltage Standing Wave Ratio. The numeric representation of the device leak, written like X:1 where X is greater when the leak is greater and 1:1 is the absolute no-leakage VSWR.
12. **Describe an impedance matching circuit**. A circuit capable of changing the impedance with different values for incoming and outgoing impedance.
13. **Describe circular polarization**. The signal travels both vertically and horizontally.
14. **What's the job of the amplifier?** To increase the incoming signal.
15. **What is noise figure?** The level of noise that a low-noise amplifier (LNA) is capable of discriminating.
16. **What's a transfer curve?** It is a graph of the output power versus the input power of an amplifier.
17. **What's the 911 point?** The point where the input power results in flames on the other end.
18. **What's saturation point?** The point where the amplifier loses its linearity of the transfer curve.
19. **What does balanced amplifier consist of?** Two amplifiers stuck together.
20. **What is the filters function?** To filter out all the unwanted radio signals.
21. **What filter types are there?** Low-pass (all frequencies below are allowed to pass in), high-pass, bandpass (all frequencies of a given bandwidth are allowed) and band-reject.
22. **What are saw filters used for?** Very low frequency filtering.
23. **What are mixers used for?** To change the frequency of the signal.
24. **What's a superheterodyne receiver?** It's a receiver that uses two mixers in a row.
25. **Describe VCO**. Voltage-controlled oscillators change the frequency of the produced sine wave depending on the voltage they receive.
26. **Who is responsible for PLLs?** Phase-locked loops are done by synthesizer, that samples the sine wave and if it's not good enough, changes it.
27. Have you studied buses? What types?
28. Have you studied pipelining? List the 5 stages of a 5 stage pipeline. Assuming 1 clock per stage, what is the latency of an instruction in a 5 stage machine? What is the throughput of this machine ?
29. How many bit combinations are there in a byte?
30. For a single computer processor computer system, what is the purpose of a processor cache and describe its operation?
31. Explain the operation considering a two processor computer system with a cache for each processor.
32. What are the main issues associated with multiprocessor caches and how might you solve them?
33. Explain the difference between write through and write back cache.
34. Are you familiar with the term MESI?
35. Are you familiar with the term snooping?

36. Describe a finite state machine that will detect three consecutive coin tosses (of one coin) that results in heads.
37. In what cases do you need to double clock a signal before presenting it to a synchronous state machine?
38. You have a driver that drives a long signal & connects to an input device. At the input device there is either overshoot, undershoot or signal threshold violations, what can be done to correct this problem?
39. What are the total number of lines written by you in C/C++? What is the most complicated/valuable program written in C/C++?
40. What compiler was used?
41. What is the difference between = and == in C?
42. Are you familiar with VHDL and/or Verilog?
43. What types of CMOS memories have you designed? What were their size? Speed?
44. What work have you done on full chip Clock and Power distribution? What process technology and budgets were used?
45. What types of I/O have you designed? What were their size? Speed? Configuration? Voltage requirements?
46. Process technology? What package was used and how did you model the package/system? What parasitic effects were considered?
47. What types of high speed CMOS circuits have you designed?
48. What transistor level design tools are you proficient with? What types of designs were they used on?
49. What products have you designed which have entered high volume production?
50. What was your role in the silicon evaluation/product ramp? What tools did you use?
51. If not into production, how far did you follow the design and why did not you see it into production?
52. **What is a Microprocessor?** - Microprocessor is a program-controlled device, which fetches the instructions from memory, decodes and executes the instructions. Most Micro Processor are single- chip devices.
53. **Give examples for 8 / 16 / 32 bit Microprocessor?** - 8-bit Processor - 8085 / Z80 / 6800; 16-bit Processor - 8086 / 68000 / Z8000; 32-bit Processor - 80386 / 80486.
54. **Why 8085 processor is called an 8 bit processor?** - Because 8085 processor has 8 bit ALU (Arithmetic Logic Review). Similarly 8086 processor has 16 bit ALU.
55. **What is 1st / 2nd / 3rd / 4th generation processor?** - The processor made of PMOS / NMOS / HMOS / HCMOS technology is called 1st / 2nd / 3rd / 4th generation processor, and it is made up of 4 / 8 / 16 / 32 bits.
56. **Define HCMOS?** - High-density n- type Complimentary Metal Oxide Silicon field effect transistor.
57. **What does microprocessor speed depend on?** - The processing speed depends on DATA BUS WIDTH.
58. **Is the address bus unidirectional?** - The address bus is unidirectional because the address information is always given by the Micro Processor to address a memory location of an input / output devices.

59. **Is the data bus is Bi-directional?** - The data bus is Bi-directional because the same bus is used for transfer of data between Micro Processor and memory or input / output devices in both the direction.
60. **What is the disadvantage of microprocessor?** - It has limitations on the size of data. Most Microprocessor does not support floating-point operations.
61. **What is the difference between microprocessor and microcontroller?** - In Microprocessor more op-codes, few bit handling instructions. But in Microcontroller: fewer op-codes, more bit handling Instructions, and also it is defined as a device that includes micro processor, memory, & input / output signal lines on a single chip.
62. **What is meant by LATCH?** - Latch is a D- type flip-flop used as a temporary storage device controlled by a timing signal, which can store 0 or 1. The primary function of a Latch is data storage. It is used in output devices such as LED, to hold the data for display.
63. **Why does microprocessor contain ROM chips?** - Microprocessor contain ROM chip because it contain instructions to execute data.
64. **What is the difference between primary & secondary storage device?** - In primary storage device the storage capacity is limited. It has a volatile memory. In secondary storage device the storage capacity is larger. It is a nonvolatile memory. Primary devices are: RAM / ROM. Secondary devices are: Floppy disc / Hard disk.
65. **Difference between static and dynamic RAM?** - Static RAM: No refreshing, 6 to 8 MOS transistors are required to form one memory cell, Information stored as voltage level in a flip flop. Dynamic RAM: Refreshed periodically, 3 to 4 transistors are required to form one memory cell, Information is stored as a charge in the gate to substrate capacitance.
66. **What is interrupt?** - Interrupt is a signal send by external device to the processor so as to request the processor to perform a particular work.
67. **What is cache memory?** - Cache memory is a small high-speed memory. It is used for temporary storage of data & information between the main memory and the CPU (center processing unit). The cache memory is only in RAM.
68. **What is called "Scratch pad of computer"?** - Cache Memory is scratch pad of computer.
69. **Which transistor is used in each cell of EPROM?** - Floating –gate Avalanche Injection MOS (FAMOS) transistor is used in each cell of EPROM.
70. **Differentiate between RAM and ROM?** - RAM: Read / Write memory, High Speed, Volatile Memory. ROM: Read only memory, Low Speed, Non Voliate Memory.
71. **What is a compiler?** - Compiler is used to translate the high-level language program into machine code at a time. It doesn't require special instruction to store in a memory, it stores automatically. The Execution time is less compared to Interpreter.
72. **Which processor structure is pipelined?** - All x86 processors have pipelined structure.
73. **What is flag?** - Flag is a flip-flop used to store the information about the status of a processor and the status of the instruction executed most recently

74. **What is stack?** - Stack is a portion of RAM used for saving the content of Program Counter and general purpose registers.
75. **Can ROM be used as stack?** - ROM cannot be used as stack because it is not possible to write to ROM.
76. **What is NV-RAM?** - Nonvolatile Read Write Memory, also called Flash memory. It is also know as shadow RAM.
77. **What are the various registers in 8085?** - Accumulator register, Temporary register, Instruction register, Stack Pointer, Program Counter are the various registers in 8085 .
78. **In 8085 name the 16 bit registers?** - Stack pointer and Program counter all have 16 bits.
79. **What are the various flags used in 8085?** - Sign flag, Zero flag, Auxillary flag, Parity flag, Carry flag.
80. **What is Stack Pointer?** - Stack pointer is a special purpose 16-bit register in the Microprocessor, which holds the address of the top of the stack.
81. **What is Program counter?** - Program counter holds the address of either the first byte of the next instruction to be fetched for execution or the address of the next byte of a multi byte instruction, which has not been completely fetched. In both the cases it gets incremented automatically one by one as the instruction bytes get fetched. Also Program register keeps the address of the next instruction.
82. **Which Stack is used in 8085?** - LIFO (Last In First Out) stack is used in 8085.In this type of Stack the last stored information can be retrieved first.
83. **What happens when HLT instruction is executed in processor?** - The Micro Processor enters into Halt-State and the buses are tri-stated.
84. **What is meant by a bus?** - A bus is a group of conducting lines that carriers data, address, & control signals.
85. **What is Tri-state logic?** - Three Logic Levels are used and they are High, Low, High impedance state. The high and low are normal logic levels & high impedance state is electrical open circuit conditions. Tri-state logic has a third line called enable line.
86. **Give an example of one address microprocessor?** - 8085 is a one address microprocessor.
87. **In what way interrupts are classified in 8085?** - In 8085 the interrupts are classified as Hardware and Software interrupts.
88. **What are Hardware interrupts?** - TRAP, RST7.5, RST6.5, RST5.5, INTR.
89. **What are Software interrupts?** - RST0, RST1, RST2, RST3, RST4, RST5, RST6, RST7.
90. **Which interrupt has the highest priority?** - TRAP has the highest priority.
91. **Name 5 different addressing modes?** - Immediate, Direct, Register, Register indirect, Implied addressing modes.
92. **How many interrupts are there in 8085?** - There are 12 interrupts in 8085.
93. **What is clock frequency for 8085?** - 3 MHz is the maximum clock frequency for 8085.
94. **What is the RST for the TRAP?** - RST 4.5 is called as TRAP.
95. **In 8085 which is called as High order / Low order Register?** - Flag is called as Low order register & Accumulator is called as High order Register.

96. **What are input & output devices?** - Keyboards, Floppy disk are the examples of input devices. Printer, LED / LCD display, CRT Monitor are the examples of output devices.
97. **Can an RC circuit be used as clock source for 8085?** - Yes, it can be used, if an accurate clock frequency is not required. Also, the component cost is low compared to LC or Crystal.
98. **Why crystal is a preferred clock source?** - Because of high stability, large Q (Quality Factor) & the frequency that doesn't drift with aging. Crystal is used as a clock source most of the times.
99. **Which interrupt is not level-sensitive in 8085?** - RST 7.5 is a raising edge-triggering interrupt.
100. **What does Quality factor mean?** - The Quality factor is also defined, as Q. So it is a number, which reflects the lossness of a circuit. Higher the Q, the lower are the losses.
101. **What are level-triggering interrupt?** - RST 6.5 & RST 5.5 are level-triggering interrupts.

1. **What are the flags in 8086?** - In 8086 Carry flag, Parity flag, Auxiliary carry flag, Zero flag, Overflow flag, Trace flag, Interrupt flag, Direction flag, and Sign flag.
2. **What are the various interrupts in 8086?** - Maskable interrupts, Non-Maskable interrupts.
3. **What is meant by Maskable interrupts?** - An interrupt that can be turned off by the programmer is known as Maskable interrupt.
4. **What is Non-Maskable interrupts?** - An interrupt which can be never be turned off (ie.disabled) is known as Non-Maskable interrupt.
5. **Which interrupts are generally used for critical events?** - Non-Maskable interrupts are used in critical events. Such as Power failure, Emergency, Shut off etc.,
6. **Give examples for Maskable interrupts?** - RST 7.5, RST6.5, RST5.5 are Maskable interrupts
7. **Give example for Non-Maskable interrupts?** - Trap is known as Non-Maskable interrupts, which is used in emergency condition.
8. **What is the Maximum clock frequency in 8086?** - 5 Mhz is the Maximum clock frequency in 8086.
9. **What are the various segment registers in 8086?** - Code, Data, Stack, Extra Segment registers in 8086.
10. **Which Stack is used in 8086?** - FIFO (First In First Out) stack is used in 8086.In this type of Stack the first stored information is retrieved first.
11. **What are the address lines for the software interrupts?** -

RST 0	**0000 H**

RST1	**0008 H**
RST2	**0010 H**
RST3	**0018 H**
RST4	**0020 H**
RST5	**0028 H**
RST6	**0030 H**
RST7	**0038 H**

12. **What is SIM and RIM instructions?** - SIM is Set Interrupt Mask. Used to mask the hardware interrupts. RIM is Read Interrupt Mask. Used to check whether the interrupt is Masked or not.
13. **Which is the tool used to connect the user and the computer?** - Interpreter is the tool used to connect the user and the tool.
14. **What is the position of the Stack Pointer after the PUSH instruction?** - The address line is 02 less than the earlier value.
15. **What is the position of the Stack Pointer after the POP instruction?** - The address line is 02 greater than the earlier value.
16. **Logic calculations are done in which type of registers?** - Accumulator is the register in which Arithmetic and Logic calculations are done.
17. **What are the different functional units in 8086?** - Bus Interface Unit and Execution unit, are the two different functional units in 8086.
18. **Give examples for Micro controller?** - Z80, Intel MSC51 &96, Motorola are the best examples of Microcontroller.
19. **What is meant by cross-compiler?** - A program runs on one machine and executes on another is called as cross-compiler.
20. **What are the address lines for the hardware interrupts?** -

RST 7.5	**003C H**
RST 6.5	**0034 H**

RST 5.5	**002C H**
TRAP	**0024 H**

21. **Which Segment is used to store interrupt and subroutine return address registers?** - Stack Segment in segment register is used to store interrupt and subroutine return address registers.
22. **Which Flags can be set or reset by the programmer and also used to control the operation of the processor?** - Trace Flag, Interrupt Flag, Direction Flag.
23. **What does EU do?** - Execution Unit receives program instruction codes and data from BIU, executes these instructions and store the result in general registers.
24. **Which microprocessor accepts the program written for 8086 without any changes?** - 8088 is that processor.
25. **What is the difference between 8086 and 8088?** - The BIU in 8088 is 8-bit data bus & 16- bit in 8086.Instruction queue is 4 byte long in 8088and 6 byte in 8086.

1. Insights of an inverter. Explain the working?
2. Insights of a 2 input NOR gate. Explain the working?
3. Insights of a 2 input NAND gate. Explain the working?
4. Implement F= not (AB+CD) using CMOS gates?
5. Insights of a pass gate. Explain the working?
6. Why do we need both PMOS and NMOS transistors to implement a pass gate?
7. What does the above code synthesize to?
8. Cross section of a PMOS transistor?
9. Cross section of an NMOS transistor?
10. What is a D-latch? Write the VHDL Code for it?
11. Differences between D-Latch and D flip-flop?
12. Implement D flip-flop with a couple of latches? Write a VHDL Code for a D flip-flop?
13. What is latchup? Explain the methods used to prevent it?
14. What is charge sharing?
15. While using logic design, explain the various steps that r followed to obtain the desirable design in a well defined manner?
16. Why is OOPS called OOPS? (C++)
17. What is a linked list? Explain the 2 fields in a linked list?
18. Implement a 2 I/P and gate using Tran gates?
19. Insights of a 4bit adder/Sub Circuit?
20. For f = AB+CD if B is S-a-1, what r the test vectors needed to detect the fault?
21. Explain various adders and diff between them?
22. Explain the working of 4-bit Up/down Counter?

23. A circuit has 1 input X and 2 outputs A and B. If X = HIGH for 4 clock ticks, A = 1. If X = LOW for 4 clock ticks, B = 1. Draw a state diagram for this Spec?
24. Advantages and disadvantages of Mealy and Moore?
25. Id vs. Vds Characteristics of NMOS and PMOS transistors?
26. Explain the operation of a 6T-SRAM cell?
27. Differences between DRAM and SRAM?
28. Implement a function with both ratioed and domino logic and merits and demerits of each logic?
29. Given a circuit and asked to tell the output voltages of that circuit?
30. How can you construct both PMOS and NMOS on a single substrate?
31. What happens when the gate oxide is very thin?
32. What is setup time and hold time?
33. Write a pseudo code for sorting the numbers in an array?
34. What is pipelining and how can we increase throughput using pipelining?
35. Explain about stuck at fault models, scan design, BIST and IDDQ testing?
36. What is SPICE?
37. Differences between IRSIM and SPICE?
38. Differences between netlist of HSPICE and Spectre?
39. What is FPGA?
40. Draw the Cross Section of an Inverter? Clearly show all the connections between M1 and poly, M1 and diffusion layers etc?
41. Draw the Layout of an Inverter?
42. If the current thru the poly is 20nA and the contact can take a max current of 10nA how would u overcome the problem?
43. Implement F = AB+C using CMOS gates?
44. Working of a 2-stage OPAMP?
45. 6-T XOR gate?
46. Differences between blocking and Non-blocking statements in Verilog?
47. Differences between Signals and Variables in VHDL? If the same code is written using Signals and Variables what does it synthesize to?
48. Differences between functions and Procedures in VHDL?
49. What is component binding?
50. What is polymorphism? (C++)
51. What is hot electron effect?
52. Define threshold voltage?
53. Factors affecting Power Consumption on a chip?
54. Explain Clock Skew?
55. Why do we use a Clock tree?
56. Explain the various Capacitances associated with a transistor and which one of them is the most prominent?
57. Explain the Various steps in Synthesis?
58. Explain ASIC Design Flow?
59. Explain Custom Design Flow?
60. Why is Extraction performed?
61. What is LVS, DRC?
62. Who provides the DRC rules?

63. What is validation?
64. What is Cross Talk?
65. Different ways of implementing a comparator?
66. What r the phenomenon which come into play when the devices are scaled to the sub-micron lengths?
67. What is clock feed through?
68. Implement an Inverter using a single transistor?
69. What is Fowler-Nordheim Tunneling?
70. Insights of a Tri-state inverter?
71. If an/ap = 0.5, an/ap = 1, an/ap = 3, for 3 inverters draw the transfer characteristics?
72. Differences between Array and Booth Multipliers?
73. Explain the concept of a Clock Divider Circuit? Write a VHDL code for the same?
74. Which gate is normally preferred while implementing circuits using CMOS logic, NAND or NOR? Why?
75. Insights of a Tri-State Inverter?
76. Basic Stuff related to Perl?
77. Have you studied buses? What types?
78. Have you studied pipelining? List the 5 stages of a 5 stage pipeline. Assuming 1 clock per stage, what is the latency of an instruction in a 5 stage machine? What is the throughput of this machine ?
79. How many bit combinations are there in a byte?
80. For a single computer processor computer system, what is the purpose of a processor cache and describe its operation?
81. Explain the operation considering a two processor computer system with a cache for each processor.
82. What are the main issues associated with multiprocessor caches and how might you solve them?
83. Explain the difference between write through and write back cache.
84. Are you familiar with the term MESI?
85. Are you familiar with the term snooping?
86. Describe a finite state machine that will detect three consecutive coin tosses (of one coin) that results in heads.
87. In what cases do you need to double clock a signal before presenting it to a synchronous state machine?
88. You have a driver that drives a long signal & connects to an input device. At the input device there is either overshoot, undershoot or signal threshold violations, what can be done to correct this problem?
89. What are the total number of lines written by you in C/C++? What is the most complicated/valuable program written in C/C++?
90. What compiler was used?
91. What is the difference between = and == in C?
92. Are you familiar with VHDL and/or Verilog?
93. What types of CMOS memories have you designed? What were their size? Speed?

94. What work have you done on full chip Clock and Power distribution? What process technology and budgets were used?
95. What types of I/O have you designed? What were their size? Speed? Configuration? Voltage requirements?
96. Process technology? What package was used and how did you model the package/system? What parasitic effects were considered?
97. What types of high speed CMOS circuits have you designed?
98. What transistor level design tools are you proficient with? What types of designs were they used on?
99. What products have you designed which have entered high volume production?
100. What was your role in the silicon evaluation/product ramp? What tools did you use?
101. If not into production, how far did you follow the design and why did not you see it into production?

1. What is pipelining?
2. What are the five stages in a DLX pipeline?
3. For a pipeline with ‘n’ stages, what’s the ideal throughput? What prevents us from achieving this ideal throughput?
4. What are the different hazards? How do you avoid them?
5. Instead of just 5-8 pipe stages why not have, say, a pipeline with 50 pipe stages?
6. What are Branch Prediction and Branch Target Buffers?
7. How do you handle precise exceptions or interrupts?
8. What is a cache?
9. What’s the difference between Write-Through and Write-Back Caches? Explain advantages and disadvantages of each.
10. Cache Size is 64KB, Block size is 32B and the cache is Two-Way Set Associative. For a 32-bit physical address, give the division between Block Offset, Index and Tag.
11. What is Virtual Memory?
12. What is Cache Coherency?
13. What is MESI?
14. What is a Snooping cache?
15. What are the components in a Microprocessor?
16. What is ACBF(Hex) divided by 16?
17. Convert 65(Hex) to Binary
18. Convert a number to its two’s compliment and back
19. The CPU is busy but you want to stop and do some other task. How do you do it?
20. Give two ways of converting a two input NAND gate to an inverter
21. Given a circuit, draw its exact timing response. (I was given a Pseudo Random Signal Generator; you can expect any sequential ckt)
22. What are set up time & hold time constraints? What do they signify? Which one is critical for estimating maximum clock frequency of a circuit?

23. Give a circuit to divide frequency of clock cycle by two
24. Design a divide-by-3 sequential circuit with 50% duty circle. (Hint: Double the Clock)
25. Suppose you have a combinational circuit between two registers driven by a clock. What will you do if the delay of the combinational circuit is greater than your clock signal? (You can’t resize the combinational circuit transistors)
26. The answer to the above question is breaking the combinational circuit and pipelining it. What will be affected if you do this?
27. What are the different Adder circuits you studied?
28. Give the truth table for a Half Adder. Give a gate level implementation of the same.
29. Draw a Transmission Gate-based D-Latch.
30. Design a Transmission Gate based XOR. Now, how do you convert it to XNOR? (Without inverting the output)
31. How do you detect if two 8-bit signals are same?
32. How do you detect a sequence of "1101" arriving serially from a signal line?
33. Design any FSM in VHDL or Verilog.
34. Explain RC circuit’s charging and discharging.
35. Explain the working of a binary counter.
36. Describe how you would reverse a singly linked list

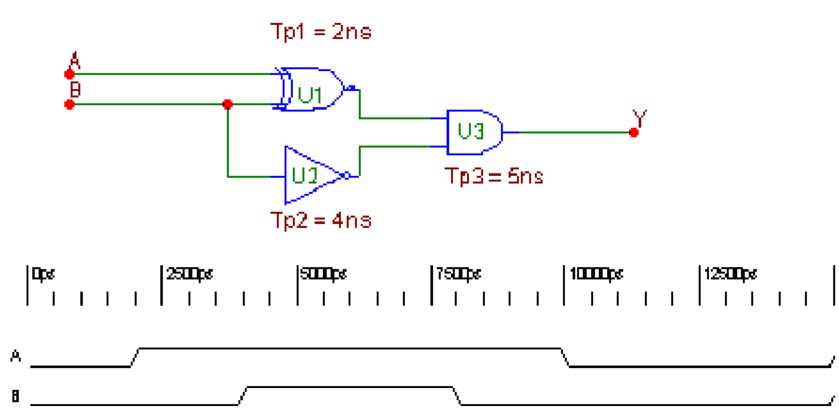
DIGITAL QUESTIONS

OCT-17-2001

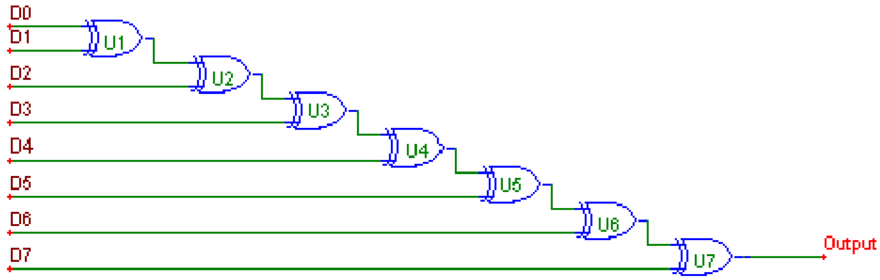
Digital
Tutorial
Questions
Tools
Books
Links
Home
Disclaimer

Sample Digital Questions Asked in Interviews

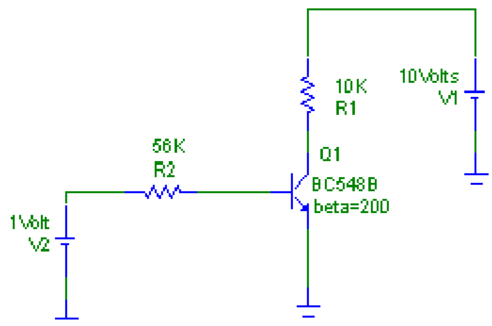
- What is the output of AND gate in the circuit below, when A and B are as in waveform? Where, Tp is gate delay of respective gate.



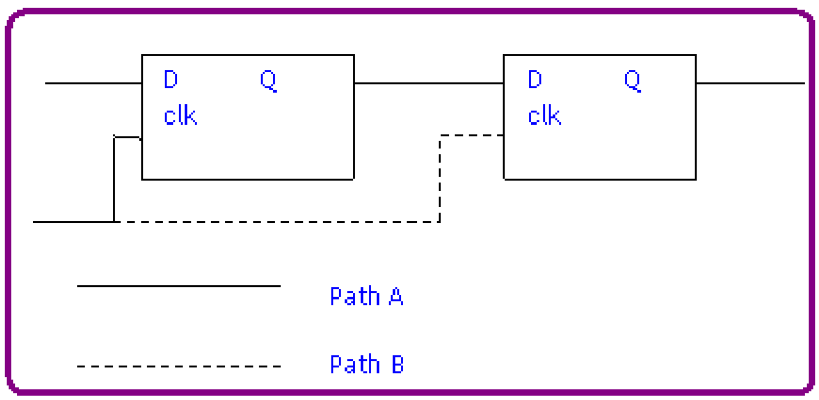
- Identify the below circuit, and its limitation ?



- What is the current though the resistor R1 (Ic) ?



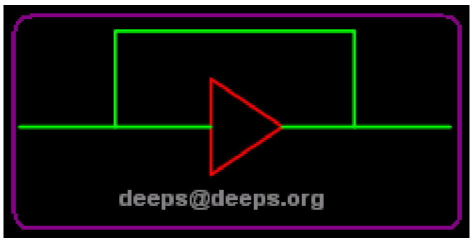
- Referring to the diagram below, briefly explain what will happen if the propagation delay of the clock signal in path B is much too high compare to path A. How do we solve this problem if the propagation delay of path B can not be reduced ?



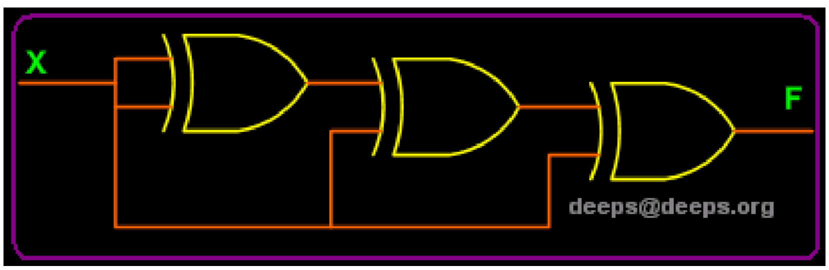
- What is the function of a D flip-flop, whose inverted output is connected to its input ?

- Design a circuit to divide input frequency by 2 ?

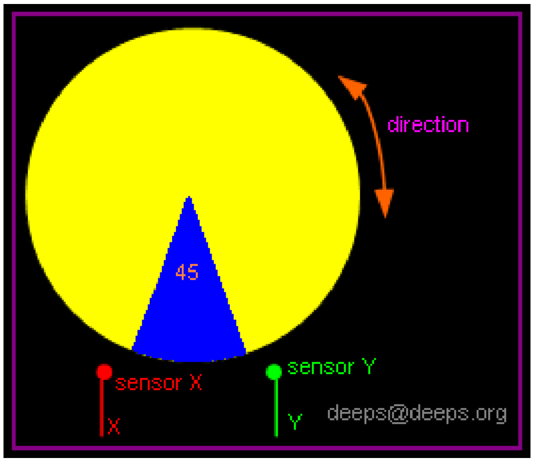
- Design a divide-by-3 sequential circuit with 50% duty cycle.?
- What are the different types of adder implementation ?
- Draw a Transmission Gate-based D-Latch ?
- Give the truth table for a Half Adder. Give a gate level implementation of the same.
- What is the purpose of the buffer in below circuit, is it necessary/redundant to have buffer ?



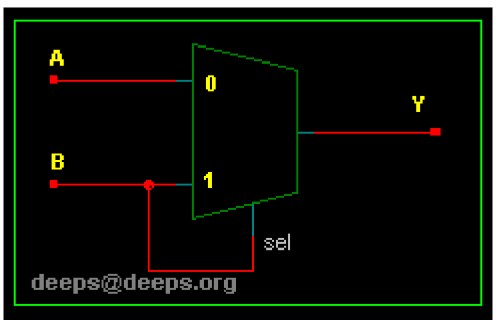
- What is output of the below circuit, assuming that value of 'X' is not known ?



- Consider a circular disk as shown in figure below with two sensors mounted X, Y and blue shade painted on the disk for a angle of 45 degree. Design a circuit with minimum number of gates to detect the direction of rotation.

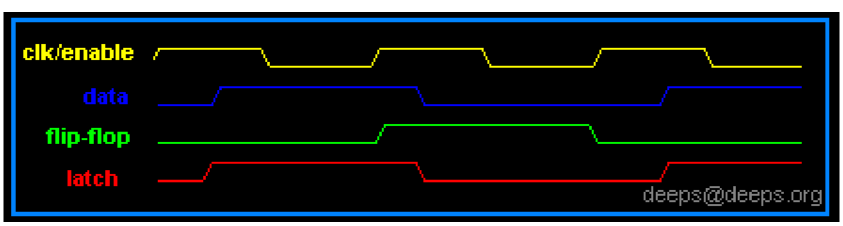


Design a OR gate from 2:1 MUX.

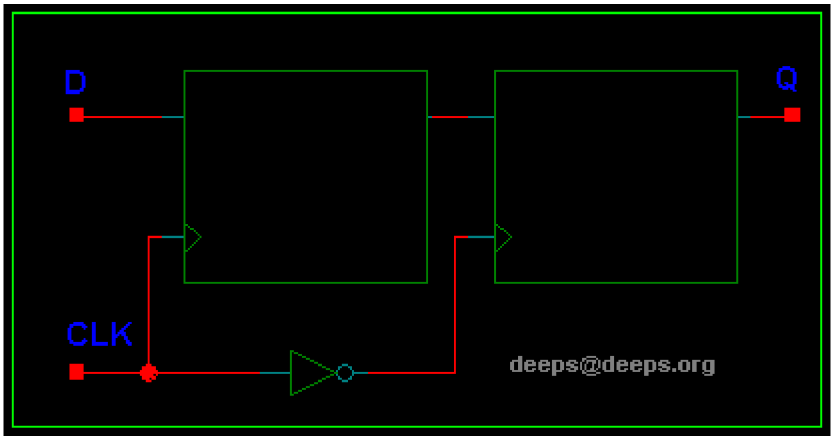


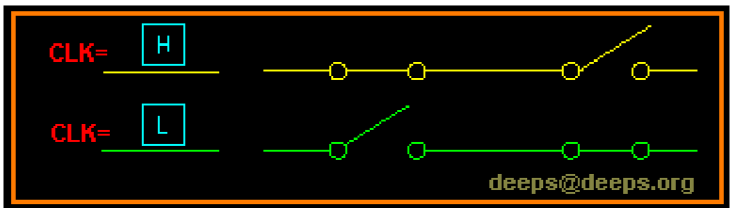
What is the difference between a LATCH and a FLIP-FLOP ?

- Latch is a level sensitive device and flip-flop is edge sensitive device
- Latch is sensitive to glitches on enable pin, where as flip-flop is immune to gltiches.
- Latches take less gates (also less power) to implement then flip-flops
- Latches are faster then flip-flops



Design a D Flip-Flop from two latches.





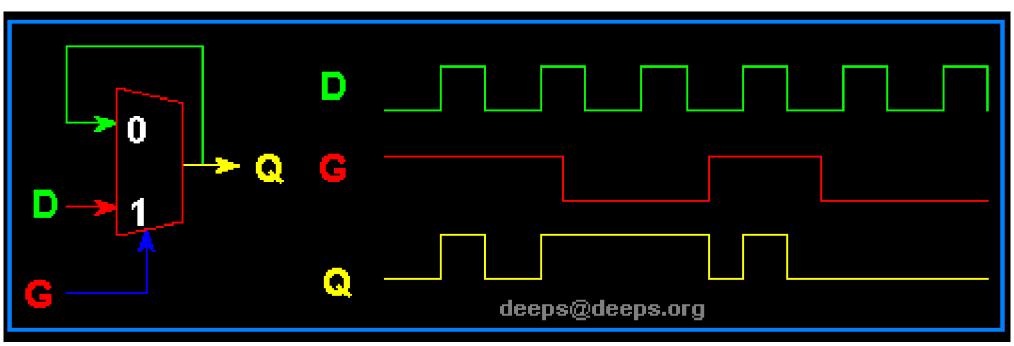
Design a 2 bit counter using D Flip-Flop.

Transition Table ➤ Next State Transistion Table ➤ K-map ➤ circuit

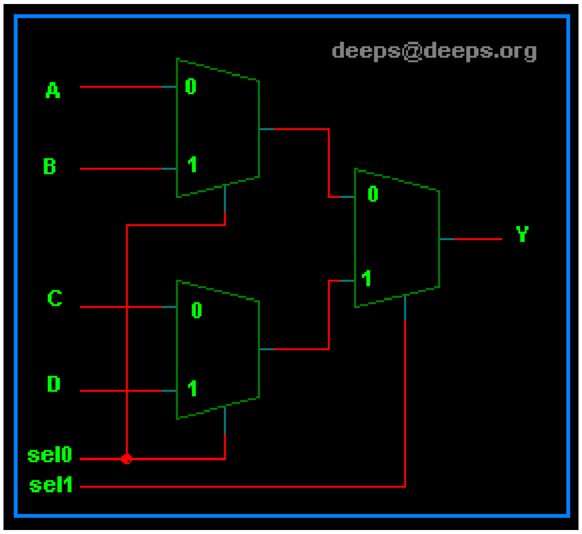
What are the two types of delays in any digital system ?

Wire Delay and Gate Delay

Design a Transparent Latch using a 2:1 Mux.



Design a 4:1 Mux using 2:1 Mux's and some combo logic.



- What is metastable state ? How does it occur ?

  Refer Tidbits section

- What is metastablity ?

  Refer Tidbits section

- Design a 3:8 decoder

- Design a FSM to detect sequence "101" in input sequence.

- Convert NAND gate into Inverter, in two different ways.

- Design a D and T flip flop using 2:1 mux, use of other components not allowed, just the mux.

- Design a divide by two counter using D-Latch.

- Design D Latch from SR flip-flop.

- Define Clock Skew , Negative Clock Skew, Positive Clock Skew ?

- What is Race Condition ?
- Design a 4 bit Gray Counter ?
- Design 4-bit Synchronous counter, Asynchronous counter ?
- Design a 16 byte Asynchronous FIFO?
- What is the difference between a EEPROM and FLASH ?
- What is the difference between a NAND-based Flash and NOR-based Flash ?
- You are given a 100 MHz clock , Design a 33.3 MHz clock with and without 50 % duty cycle?
- Design a Read on Reset System ?
- Which one is superior Asynchronous Reset or Synchronous Reset, Explain ?
- Design a State machine for Traffic Control at a Four point Junction ?

STATE MACHINES

1. Describe a finite state machine that will detect three consecutive coin tosses (of one coin) that results in heads.
2. In what cases do you need to double clock a signal before presenting it to a synchronous state machine?
3. Design a state-machine (or draw a state-diagram) to give an output '1' when the # of A's are even and # of B's are odd. The input is in the form of a serial-stream

(one-bit per clock cycle). The inputs could be of the type A, B or C. At any given clock cycle, the output is a '1', provided the # of A's are even and # of B's are odd. At any given clock cycle, the output is a '0', if the above condition is not satisfied.

4. To detect the sequence "abca" when the inputs can be a b c d.
5. Design a finite state machine to give a modulo 3 counter when x=0 and modulo 4 counter when x=1.
6. How do you detect a sequence of "1101" arriving serially from a signal line?
7. Design any FSM in VHDL or Verilog.

Logic design:

1. Draw the transistor level CMOS #input NAND or NOR gate. After drawing it lot of qestions on that ckt will be asked.

2. Transistor sizing for given rise time and fall time. How do you size it for equal rise and fall time.

3. Given a function whose inputs are dependent on its outputs. Design a sequential circuit.

4. Design a finite state machine to give a modulo 3 counter when x=0 and modulo 4 counter when x=1.

5. Given a boolean equation minimize it.

6. Given a boolean equation draw the transistor level minimum transistor circuit.

7. What is the function of a D-flipflop, whose inverted outputs are connected to its input ?

8. What will you do if you want to drive a large capacitance ?

Layout related questions:

1. asked me to layout the 3 input nand gate.

2. Later he asked me to modify it to consume as much less space as we can.

3. He also asked me about the transistor sizing.

---

1. He asked me to draw the cross section of an inverter and asked me to show all the capacitances on it and reason for those capacitances.

2. Describe the latchup effect.

3. More about the tristate buffers.

3. What will be the voltage at the output node of a triostate buffer in its high impedence state. He gave a waveform for the input and asked me to draw the output waveform for that.

4. Posed a lot of questions on charge sharing problems and keeper circuits.

5. Asked me to draw the Id Vds curves for mosfet. Asked me to explain the regions and some couses for that curve like channel width modulation.

6. He asked me about the electron migration effect and methods to avoid it.

7. Asked me to draw the dynamic logic of a particular gate and then posed lots of tricky questions from the previous discussion.

8. He asked me to draw the 6 transistor contemporary sram cell and asked me to explain how the reading and writing is done in it.

9. Something about trip point.

Computer Architecture Questions:

1. Explain what is DMA?
2. what is pipelining?
3. what are superscalar machines and vliw machines?
4. what is cache?
5. what is cache coherency and how is it eliminated?

6. what is write back and write through caches?
7. what are different pipelining hazards and how are they eliminated.
8. what are different stages of a pipe?
9. eplain more about branch prediction in controlling the control hazards
10. Give examples of data hazards with pseudo codes.
11. Caluculating the number of sets given its way and size in a cache?
12. How is a block found in a cache?
13. scoreboard analysis.
14. What is miss penalty and give your own ideas to eliminate it.
15. How do you improve the cache performance.
16. Different addressing modes.
17. Computer arithmetic with two's complements.
18. About hardware and software interrupts.
19. What is bus contention and how do you eliminate it.
20. What is aliasing?
21) What is the difference between a latch and a flip flop?
22) What is the race around condition? How can it be overcome?
23) What is the purpose of cache? How is it used?
24) What are the types of memory management?

The following questions are from the prescreening interview at INTEL:

COMPUTER ARCHITECTURE QUESTIONS
1. For a single computer processor computer system, what is the purpose of a processor cache and describe its operation?
2. Explain the operation considering a two processor computer system with a cache for each processor.
What are the main issues associated with multiprocessor caches and how might you solve it?
3. Explain the difference between write through and write back cache.
4. Are you familiar with the term MESI?
5. Are you familiar with the term snooping?

STATE MACHINE QUESTIONS
1. Describe a finite state machine that will detect three consecutive coin tosses (of one coin) that results in heads.
2. In what cases do you need to double clock a signal before presenting it to a synchronous state machine?

SIGNAL LINE QUESTIONS

1. You have a driver that drives a long signal & connects to an input device. At the input device there is either overshoot,
undershoot or signal threshold violations, what can be done to correct this problem?

VALIDATION QUESTIONS:
What are the total number of lines written in C/C++? What is the most complicated/valuable program written in C/C++?
What compiler was used?
Have you studied busses? What types?
Have you studied pipelining? List the 5 stages of a 5 stage pipeline. Assuming 1 clock per stage, what is the latency of an instruction in a 5 stage machine? What is the throughput of this machine ?
How many bit combinations are there in a byte?
What is the difference between = and == in C?
Are you familiar with VHDL and/or Verilog?

MEMORY, I/O, CLOCK AND POWER QUESTIONS
1. What types of CMOS memories have you designed? What were their size? Speed? Configuration Process technology?
2. What work have you done on full chip Clock and Power distribution? What process technology and budgets were used?
3. What types of I/O have you designed? What were their size? Speed? Configuration? Voltage requirements?
Process technology? What package was used and how did you model the package/system?
What parasitic effects were considered?
4. What types of high speed CMOS circuits have you designed?
5. What transistor level design tools are you proficient with? What types of designs were they used on?
6. What products have you designed which have entered high volume production?
What was your role in the silicon evaluation/product ramp? What tools did you use?
7. If not into production, how far did you follow the design and why did not you see it into production?

1. Explain why & how a MOSFET works

2. Draw Vds-Ids curve for a MOSFET. Now, show how this curve changes (a) with increasing Vgs (b) with increasing transistor width © considering Channel Length Modulation
3. Explain the various MOSFET Capacitances & their significance
4. Draw a CMOS Inverter. Explain its transfer characteristics
5. Explain sizing of the inverter
6. How do you size NMOS and PMOS transistors to increase the threshold voltage?
7. What is Noise Margin? Explain the procedure to determine Noise Margin
8. Give the expression for CMOS switching power dissipation
9. What is Body Effect?
10. Describe the various effects of scaling
11. Give the expression for calculating Delay in CMOS circuit
12. What happens to delay if you increase load capacitance?
13. What happens to delay if we include a resistance at the output of a CMOS circuit?
14. What are the limitations in increasing the power supply to reduce delay?
15. How does Resistance of the metal lines vary with increasing thickness and increasing length?
16. You have three adjacent parallel metal lines. Two out of phase signals pass through the outer two metal lines. Draw the waveforms in the center metal line due to interference. Now, draw the signals if the signals in outer metal lines are in phase with each other
17. What happens if we increase the number of contacts or via from one metal layer to the next?
18. Draw a transistor level two input NAND gate. Explain its sizing (a) considering Vth (b) for equal rise and fall times
19. 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 NMOS inputs A & B, which one would you place near the output?
20. Draw the stick diagram of a NOR gate. Optimize it
21. For CMOS logic, give the various techniques you know to minimize power consumption
22. What is Charge Sharing? Explain the Charge Sharing problem while sampling data from a Bus
23. Why do we gradually increase the size of inverters in buffer design? Why not give the output of a circuit to one large inverter?
24. In the design of a large inverter, why do we prefer to connect small transistors in parallel (thus increasing effective width) rather than lay out one transistor with large width?
25. Given a layout, draw its transistor level circuit. (I was given a 3 input AND gate and a 2 input Multiplexer. You can expect any simple 2 or 3 input gates)
26. Give the logic expression for an AOI gate. Draw its transistor level equivalent. Draw its stick diagram
27. Why don't we use just one NMOS or PMOS transistor as a transmission gate?
28. For a NMOS transistor acting as a pass transistor, say the gate is connected to VDD, give the output for a square pulse input going from 0 to VDD
29. Draw a 6-T SRAM Cell and explain the Read and Write operations

30. Draw the Differential Sense Amplifier and explain its working. Any idea how to size this circuit? (Consider Channel Length Modulation)
31. What happens if we use an Inverter instead of the Differential Sense Amplifier?
32. Draw the SRAM Write Circuitry
33. Approximately, what were the sizes of your transistors in the SRAM cell? How did you arrive at those sizes?
34. How does the size of PMOS Pull Up transistors (for bit & bit- lines) affect SRAM's performance?
35. What's the critical path in a SRAM?
36. Draw the timing diagram for a SRAM Read. What happens if we delay the enabling of Clock signal?
37. Give a big picture of the entire SRAM Layout showing your placements of SRAM Cells, Row Decoders, Column Decoders, Read Circuit, Write Circuit and Buffers
38. In a SRAM layout, which metal layers would you prefer for Word Lines and Bit Lines? Why?
39. How can you model a SRAM at RTL Level?
40. What's the difference between Testing & Verification?
41. For an AND-OR implementation of a two input Mux, how do you test for Stuck-At-0 and Stuck-At-1 faults at the internal nodes? (You can expect a circuit with some redundant logic)
42. What is Latch Up? Explain Latch Up with cross section of a CMOS Inverter. How do you avoid Latch Up?

TEXAS INSTRUMENTS: TECHNICAL TEST

Date: 20th December 2003
Venue: TAJ LANDS END, BANDRA-WEST, MUMBAI
Note: Out of 100 candidates only 7 were short-listed)

1. For a CMOS inverter, the transition slope of Vout vs Vin DC characteristics can be increased (steeper transition) by:

   a. Increasing W/L of PMOS transistor
   b. Increasing W/L of NMOS transistor
   c. Increasing W/L of both transistors by the same factor
   d. Decreasing W/L of both transistor by the same factor

Ans: c
2. Minimum number of 2-input NAND gates that will be required to implement the function: Y = AB + CD + EF is
   a. 4
   b. 5
   c. 6
   d. 7

ans: c

3. Consider a two-level memory hierarchy system M1 & M2. M1 is accessed first and on miss M2 is accessed. The access of M1 is 2 nanoseconds and the miss penalty (the time to get the data from M2 in case of a miss) is 100 nanoseconds. The probability that a valid data is found in M1 is 0.97. The average memory access time is:
   a. 4.94 nanoseconds
   b. 3.06 nanoseconds
   c. 5.00 nanoseconds
   d. 5.06 nanoseconds

ans: a

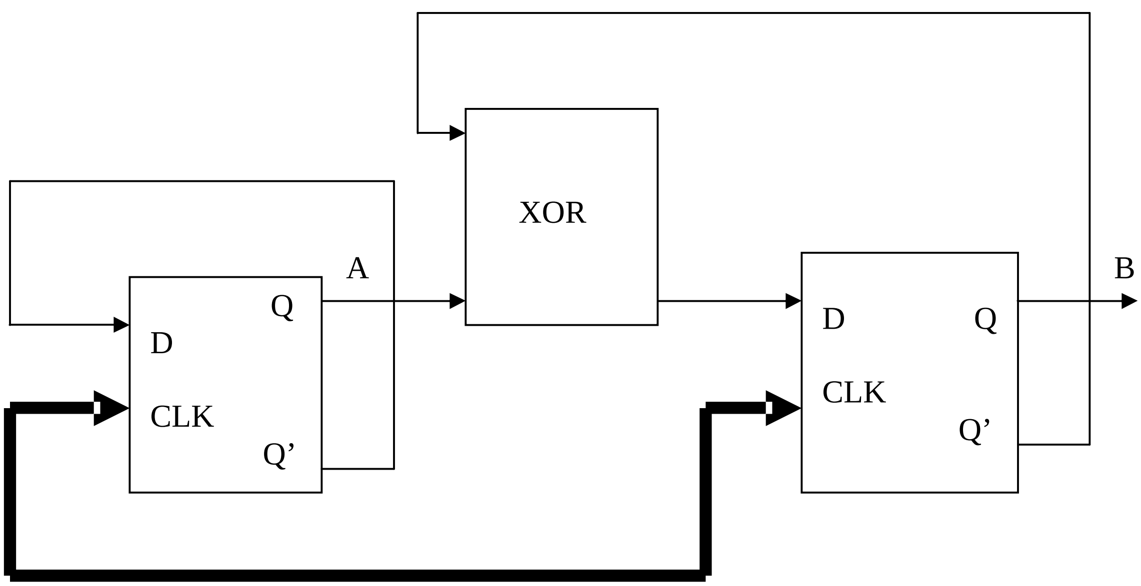
4. Interrupt latency is the time elapsed between:
   a. Occurrence of an interrupt and its detection by the CPU
   b. Assertion of an interrupt and the start of the associated ISR
   c. Assertion of an interrupt and the completion of the associated ISR
   d. Start and completion of associated ISR

Ans: d (not confirmed)

5. Which of the following is true for the function (A.B + A'.C + B.C)
   a. This function can glitch and can be further reduced
   b. This function can neither glitch nor can be further reduced
   c. This function can glitch and cannot be further reduced
   d. This function cannot glitch but can be further reduced

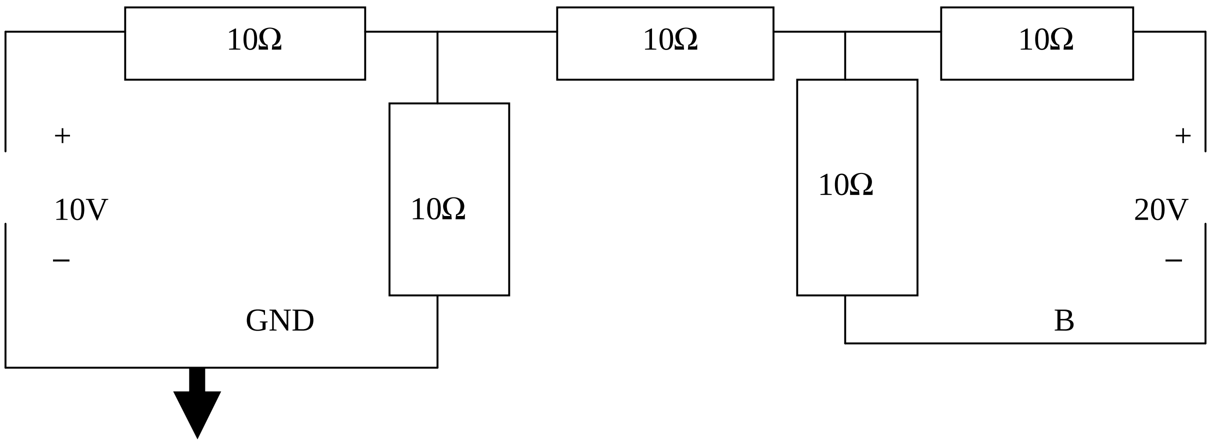
Ans: c **This can be reduced further using K-map, don't know abt glich, but it should not glitch**

6. For the two flip-flop configuration below, what is the relationship of the output at B to the clock frequency?
   a. Output frequency is 1/4th the clock frequency, with 50% duty cycle
   b. Output frequency is 1/3rd the clock frequency, with 50% duty cycle
   c. Output frequency is 1/4th the clock frequency, with 25% duty cycle
   d. Output frequency is equal to the clock frequency



Ans: a

7. The voltage on Node B is:
   a. 0
   b. 10
   c. –10
   d. –5



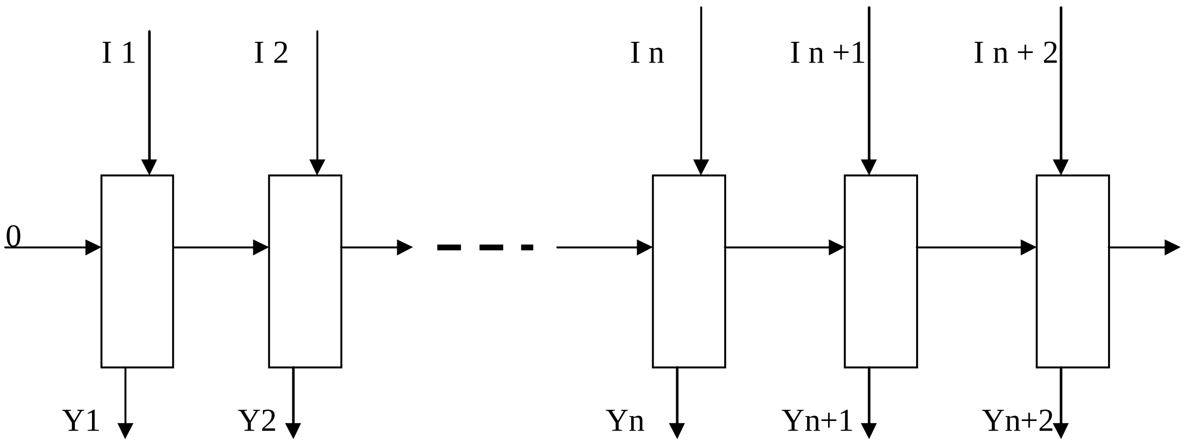
Ans: d

8. A CPU supports 250 instructions. Each instruction op-code has these fields:
   - The instruction type (one among 250)
   - A conditional register specification
   - 3 register operands
   - Addressing mode specification for both source operands

The CPU has 16 registers and supports 5 addressing modes. What is the instruction op-code length in bits?
   a. 32
   b. 24
   c. 30
   d. 36

ans: don't know

9. In the iterative network shown, the output Yn of any stage N is 1 if the total number of 1s at the inputs starting from the first stage to the Nth stage is odd. (Each identical box in the iterative network has two inputs and two outputs). The optimal logic structure for the box consists of:
   a. One AND gate and one NOR gate
   b. One NOR gate and one NAND gate
   c. Two XNOR gates
   d. One XOR gate



Ans: d

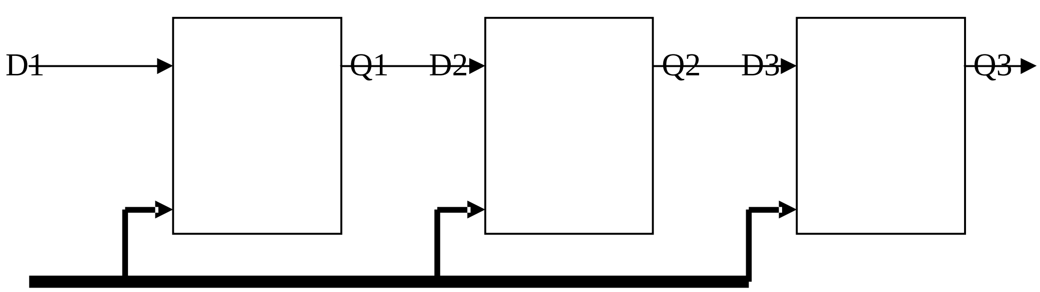
10. Consider a circuit with N logic nets. If each net can be stuck-at either values 0 and 1, in how many ways can the circuit be faulty such that only one net in it can be faulty, and such that up-to all nets in it can be faulty?

   a. 2 and 2N
   b. N and 2^N
   c. 2N and 3^N-1
   d. 2N and 3N

ans: 2N and 2^N (no match) see it .

sorry , no idea abt this

11. In the circuit shown, all the flip-flops are identical. If the set-up time is 2 ns, clock->Q delay is 3 ns and hold time is 1 ns, what is the maximum frequency of operation for the circuit?



CLOCK SIGNAL

a. 200 MHz
b. 333 MHz
c. 250 MHz
d. None of the above

Ans: a

12. Which of the following statements is/are true?
    I. Combinational circuits may have feedback, sequential circuits do not.
    II. Combinational circuits have a 'memory-less' property, sequential circuits do not.
    III. Both combinational and sequential circuits must be controlled by an external clock.

a. I only
b. II and III only
c. I and II only
d. II only

Ans: d

13. Consider an alternate binary number representation scheme, wherein the number of ones M, in a word of N bits, is always the same. This scheme is called the M-out-of-N coding scheme. If M=N/2, and N=8, what is the efficiency of this coding scheme as against the regular binary number representation scheme? (As a hint, consider that the number of unique words represent able in the latter representation with N bits is 2^N. Hence the efficiency is 100%)

a. Close to 30%
b. Close to 50%
c. Close to 70%
d. Close to 100%

Ans: a

14. A CPU supports 4 interrupts- I1, I2, I3 and I4. It supports priority of interrupts. Nested interrupts are allowed if later interrupt is higher priority than previous one. During a certain period of time, we observe the following sequence of entry into and exit from the interrupt service routine:
I1-start---I2-start---I2-end---I4-start---I3-start---I3-end---I4-end---I1-end
From this sequence, what can we infer about the interrupt routines?

a. I3 > I4 > I2 > I1
b. I4 > I3 > I2 > I1
c. I2 > I1; I3 > I4 > I1
d. I2 > I1, I3 > I4 > I2 > I1

Ans: c

15. I decide to build myself a small electric kettle to boil my cup of tea. I need 200 ml of water for my cup of tea. Assuming that typical tap water temperature is 25 C and I want the water boiling in exactly one minute, then what is the wattage required for the heating element?
[Assume: Boiling point of water is 100 C, 1 Calorie (heat required to change 1 gm of water by 1 C)= 4 joules, 1 ml of water weighs 1 gm.]

a. Data given is insufficient
b. 800 W
c. 300 W
d. 1000 W
e. 250 W

ans: d

16. The athletics team from REC Trichy is traveling by train. The train slows down, (but does not halt) at a small wayside station that has a 100 mts long platform. The sprinter (who can run 100 mts in 10 sec) decides to jump down and get a newspaper and some idlis. He jumps out just as his compartment enters the platform and spends 5 secs buying his newspaper that is at the point where he jumped out. He then sprints along the platform to buy idlis that is another 50 mts. He spends another 5 secs buying the idlis. He is now just 50 mts from the other end of the platform where the train is moving out. He begins running in the direction of the train and the only other open door in his train is located 50 mts behind the door from where he jumped. At what(uniform) speed should the train be traveled if he just misses jumping into the open door at the very edge of the platform? Make the following assumptions

- He always runs at his peak speed uniformly
- The train travels at uniform speed
- He does not wait (other than for the idlis & newspaper) or run baclwards

a. Data given is insufficient
b. 4 m/s
c. 5 m/s
d. 7.5 m/s
e. 10 m/s

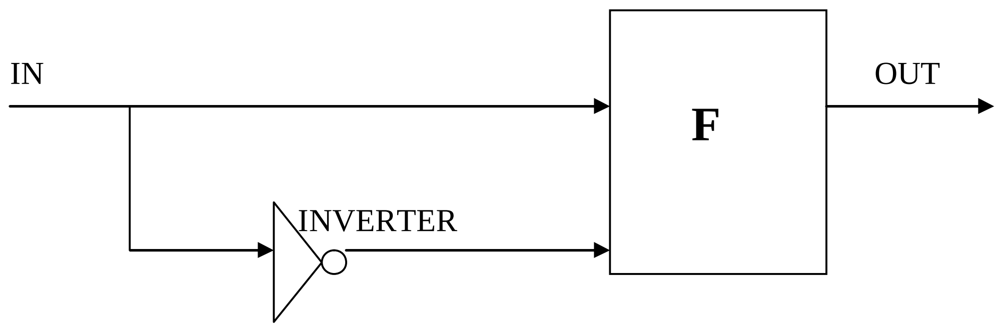
ans: c

17. State which of the following gate combinations does not form a universal logic set:

a. 2-input AND + 2-input OR
b. 2-to-1 multiplexer
c. 2-input XOR + inverter
d. 3-input NAND

ans: a

18. For the circuit shown below, what should the function F be, so that it produces an output of the same frequency (function F1), and an output of double the frequency (function F2).



a. F1= NOR gate and F2= OR gate
b. F1=NAND gate and F2= AND gate
c. F1=AND gate and F2=XOR gate
d. None of the above

Ans: c

19. The FSM (finite state machine) below starts in state Sa, which is the reset state, and detects a particular sequence of inputs leading it to state Sc. FSMs have a few characteristics. An autonomous FSM has no inputs. For a Moore FSM, the output depends on the present state alone. For a Mealy FSM, the output depends on the present state as well as the inputs. Which of the statements best describes the FSM below?

a. It has two states and is autonomous
b. The information available is insufficient
c. It is a Mealy machine with three states
d. It is a Moor machine with three states

0

1

0

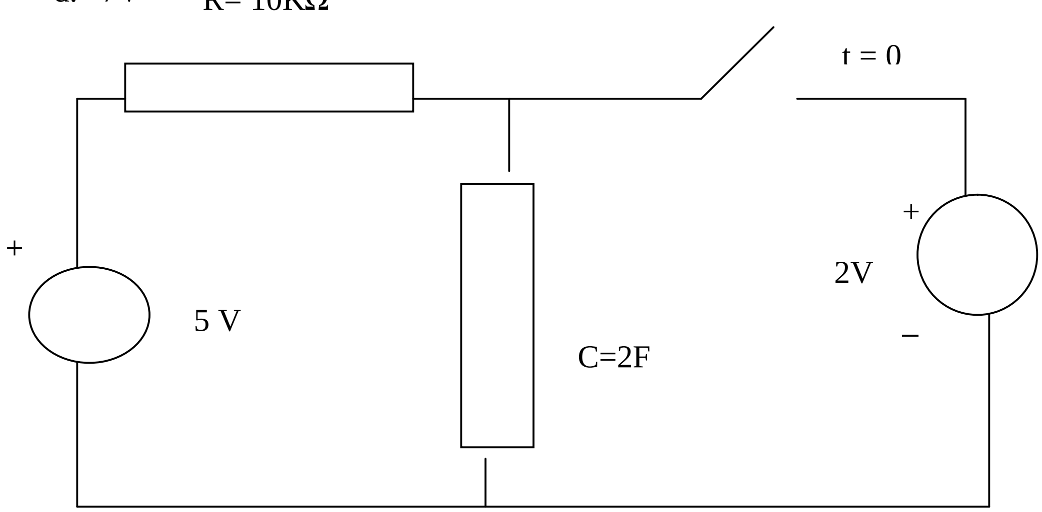
1

0

Ans :d

20. In the circuit given below, the switch is opened at time t=0. Voltage across the capacitor at t=infinity is:

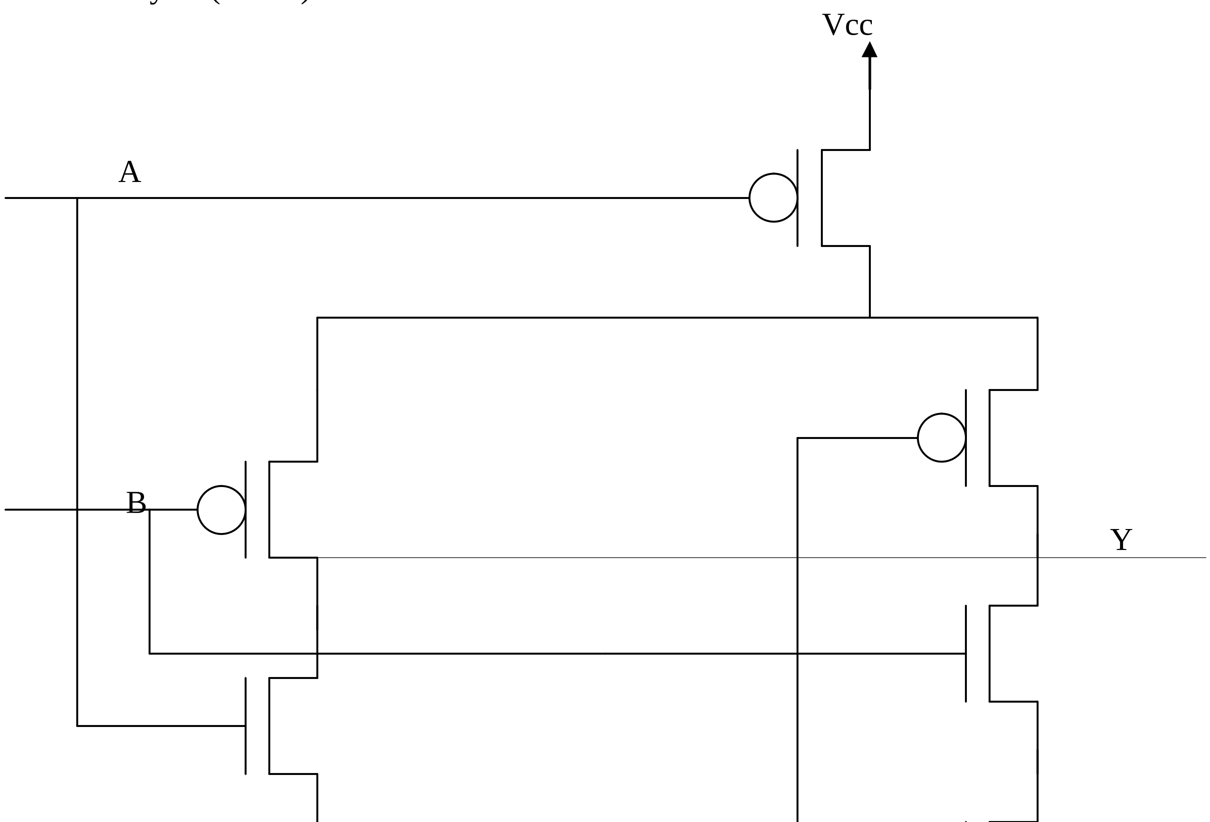
a. 2V
b. 3V
c. 5V
d. 7V



–

Ans: c

21. What is the functionality represented by the following circuit?
    a. y= ! (b+ac)
    b. y= ! (a+bc)
    c. y= ! (a(b+c))
    d. y= ! (a+b+c)



C

Ans: b

22. The value (0xdeadbeef) needs to stored at address 0x400. Which of the below ways will the memory look like in a big endian machine:

	0x403	0x402	0x401	0x400
a.	be	ef	de	ad
b.	ef	be	ad	de
c.	fe	eb	da	ed
d.	ed	da	eb	fe

ans: don't know
ans should be (b), just check with some CS guy, little endian is Intel type, Big-endian is perhaps Motorola type

23. In a given CPU-memory sub-system, all accesses to the memory take two cycles. Accesses to memories in two consecutive cycles can therefore result in incorrect data transfer. Which of the following access mechanisms guarantees correct data transfer?

a. A read operation followed by a write operation in the next cycle.
b. A write operation followed by a read operation in the next cycle.
c. A NOP between every successive reads & writes
d. None of the above

Ans: c(not confirm)
I'm also not sure.

24. An architecture saves 4 control registers automatically on function entry (and restores them on function return). Save of each registers costs 1 cycle (so does restore). How many cycles are spent in these tasks (save and restore) while running the following un-optimized code with n=5:

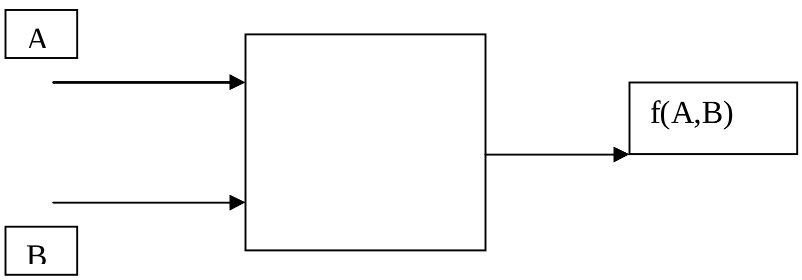
```
Void fib(int n)
{
        if((n==0) || (n==1)) return 1;
        return(fib(n-1) + fib(n-2));
}
```

a. 120
b. 80
c. 125
d. 128

ans: a

25. The maximum number of unique Boolean functions F(A,B), realizable for a two input (A,B) and single output (Z) circuit is:

a. 2
b. 6
c. 8
d. None of the above



Ans: 2*(2*2)=16 ie d

1. Draw the circuit and explain the operation of standard TTL circuit in high and low state. Give specifications of standard 74/54 TTL.
2. Draw the transfer characteristic of Std TTL and derive the co-ordinates of break points
3. Draw the circuit for HTTL circuit and explain its different features with the advantages of active pull-down. Give specifications of HTTL.
4. Write short note
    1. Noise margin
    2. Fan out
    3. Wire anding , Totem Pole and open collector
    4. Tri-state TTL
    5. Speed, Power and Speed Power product
    6. TTL unused Inputs
    7. Clamping diodes.

1) What is j to the power j?

2) What is Normal Distribution? Where is the Mean and Median on the graph for Normal Distribution?

3) Draw a simple RC-Low pass circuit.

1.two transistors are connected Vbe is 0.7volts .this is simple ckt.one

>transistor is diode equivalent. & asked the o/p across the 2 nd transistor.

>2.simple k map ans is Bbar.

>3.

>

> Emitter

>---R-------transistorbase| --

> | ---

> collector

> in above capacitor is connected parallel with resistance

>r.capacitor is not shown

> in fig.capacitor is used for in this ckt:

>

>

> ans:a.speedupb.active bypass c.decoupling

> 4.

>

> -----R------I----------o/p

> |___R_____ |

> in above r is resistence.I is cmos inverter.

> then ckt is used for:

>

>

> a.schmitt trigger b.latch c.inverter

>d.amplifier

>

>

> 5.simple amplifier ckt openloop gain of amplifier is 4.V in

>=1v.asked for V x?

> amplifdier + is connected to base. - is connected to i/p in between

>5k is connected.

> from o/p feedback connected to - of amplifier with 15k.this is ckt.

>

>

> 6.resistence inductot cap are serially connected to ac voltage 5

>volts.voltage across

> inductor is given.R I C values are given & asked for

> voltages across resistence & capacitor.

> 7.

> ___ R_____

> | |

> ---R------OPAMP ----------

> |---

> R1 R1 is for wjhat i mean what is the purpose of R1.

> |

>

> ground

>

>

> 8.asked for Vo at the o/p.it is like simple cmos realization that is n

>block is above

> & p block is below.Vdd is 3 volts at supply.V threshold 5 volts.

> 9.2 d ffs are connected in asyncro manner .clock 10 MEGAHZ.gate delay

>is 1 nanosec.

> A B are the two given D FFs.asked for AB output is:

>

>

> a.updown

> b.up c. updown glitching like that (take care abt glitching word)

>

> 10.

>

>

>   ----------------| subtractor|---------o/p

> |___HPF____|

>

>   the ckt is LPF ,HPF or APF ?

>

>   11.in a queue at the no of elements removed is proportional to no of

>elements in

>   the queue.then no of elements in the queue:

>   a.increases decreases exp or linearly(so these are the 4 options given

>choose 1 option)

>   12.with 2 i/p AND gates u have to form a 8 i/p AND gate.which is the

>fastest in the

>   following implementations.

>   ans we think ((AB)(CD))((EF)(GH))

>   13.with howmany 2:1 MUX u can for   8:1 MUX.answer is 7.

>   14. there are n states then ffs used are log n.

>   15.cube each side has r units resistence then the resistence across

>diagonal of cube.

>   16.op amp connections asked for o/p

>   the answer is (1+1/n)(v2-v1).check it out.practise this type of model.

>   17.

>   ______________ supply

```
>   ---|__          ___|
>  Ii    >_________ |___    Tranistot
>                    > ________Vo
>                     |
>                     |
>                     R      |
>                     |     |  Io
>                     ground.
```

>

>

>

>

> asked for Io/Ii=? transistor gain is beta.

>

>

> a.(1+beta)square b.1+beta c. beta

>

>

> 18.y=kxsquare. this is transfer function of a block with i/p x & o/p

>y.if i/p is

> sum of a & b then o/p is :--

>

> a. AM b.FM c. PM

```
>   19.
>              ------MULTIPLIER--- |
>              |                   |
>      _____R__|__OPAMP______________________Vo
>              ---
>              |
>              ground.
```

> v in = -Ez then o/p Vo =?

> answer is squareroot of -Ez.multiplier i/ps are a & b then

>its o/p

> is a.b;

>

>

>

--texas instruments paper .

in this paper there was 20 questions as follows in 60 minutes .

second part consists of 36 que. in 30 minutes all questions are

diagramatical.(figurs)..

1. if a 5-stage pipe-line is flushed and then we have to execute 5 and 12 instructions respectively then no. of cycles will be

a. 5 and 12

b. 6 and 13

c. 9 and 16

d.none

2. k-map

ab

----

c 1 x 0 0

1 x 0 x

solve it

a. A.B

B. ~A

C. ~B

D. A+B

Question 1
a. Design a muxed based Flip-Flop to have a setup and hold time of .6 ns with Q of the slave driving a load of 20fF. (Hand Calculations only.
b. Draw a test vector and Q of the slave and master latch of a set-up error.
c. Draw a test vector and Q of the slave and master latch of a hold error.
Since we are given set up and hold times of .6ns we just divide the time by the number of stages in each latch. In this case the rise and fall time should be .3ns for the nand and the driver mux.
Using the excell worksheet:
Note1: I used the layout values for the Master driver mux.
Note2: If we were using this for the project the Cg of the master driver mux would be 300fF which is much larger than what we assumed in our logic equations. This circuit is not really practical as the W values are greater than 50xL of the process.
What do we do about it?
We could try to steal time from other parts of the FF. This will not work since the prop delay is kind of fast for this circuit.
We could try and steal time from the logic part of the circuit.
We could try to se if we really needed a set pin (drop nand and use an inverter).
We could notice that the circuit designed in the sample notes although designed for a setup and hold of .779ns it worked at around .6ns in schematic, not layout! (Which would be faster).
Why? Remember that we break up the transitions into two, which is an over estimation. We could probably over lap the times and use .32 or longer to reduce the widths.
In all practicality we could just layout the DFF in the notes and it would be fast enough.
The easiest way to draw a wave from form with a set up error is to have D rise or fall at the same time at the triggering edge changes or within tsetup before the triggering edge. The hold time is to have the non- triggering edge change less than the hold time after the triggering edge.
Question 2
Draw a stick diagram of you project circuit shown how you will distribute the power and the clock. Do not show the details of each cell, just how you would connect power, gnd, ck, reset, and route the major signals.
Question 3
Design a chain of inverters to drive a clock signal with minimum delay. The propagation delays must be symmetric. The load that this driver has to drive is comprised of 100 DFF, which have a Cin each of 14fF.
What is the power dissipation of this driver?
alfa is still 4

Stage Wn Wp
1 (extra for logic) 1.5 3
2 6 12
3 24 48
4 96 192
To make sure the logic is not inverted we add 1 stage of min inverter to beginning.

Question 4): Is it possible to reduce clock skew to zero? Explain your answer. NO BS!
Even though there are clock layout strategies (H-tree) that can in theory reduce clock skew to zero by having the same path length from each flip-flop from the pll, process variations in R and C across the chip will cause clock skew as well as a pure H-Tree scheme is not practical (consumes too much area).
Revised: 22 Jan 99, 1150 hrs

Review Questions:
Basic Principles of
Design and Construction
of Digital Computers

1. List the major components of a generic computer system, and briefly describe the function of each.
2. What electronic device was used as a switch in the earliest electronic computers?
3. What is the derivation of the word *transistor?*
4. What are the principal types of transistors?
5. What are the major categories of circuit packages?
6. What are the major constituents of a typical Personal Computer or Workstation?
7. What is the function of the Clock Generator?
8. What is meant by the term "active-low"?
9. Define the term "bus". What are the four major constituents of a bus? Briefly describe the function
of each, and indicate which ones in the "Beboputer" are unidirectional and which are bidirectional.
10. Name several different kinds of data that can be represented in memory locations in a digital
computer.

Review Questions: Basic Principles of Design and Construction of Digital Computers

11. What is the difference in the electrical connections between ROM and RAM?
12. Explain how various memory modules (both RAM and ROM) can be organized and connected to
construct the totality of a computer's semiconductor primary memory.
13. Explain the memory map of the "Beboputer" **and** of the IBM PC.
14. Explain how circuit components can be connected and configured to constitute an Input Port or an
Output Port of specified address.
15. Draw a diagram and explain how a single bit of Magnetic Core memory works.

16. Why is it that some instructions occupy only one byte of memory, while others occupy two or three
or more bytes?
17. Define each of the following terms:
a. Hardware
b. Software
c. Firmware
d. Wetware
e. Vaporware
f. Freeware
g. Shareware

Answers to Selected Questions:

1. List the major components of a generic computer system, and briefly describe the function of each.
Answer: See *Bebop Bytes Back,* Figure 1.1
Inputs
Processor
Outputs
Memory
2. What electronic device was used as a switch in the earliest electronic computers?
Answer: the Vacuum Tube (triode or pentode)
3. What is the derivation of the word *transistor?*
Answer: transfer resistor
4. What are the principal types of transistors?
Answer:
point-contact transistors
Bipolar Junction Transistors (BJTs)
Field-Effect Transistors
5. What are the major categories of circuit packages?
Answer:
Discrete Components
Printed Circuit Boards (PCBs)
Integrated Circuits (ICs)
Small-Scale Integration (SSI)
Medium-Scale Integration (MSI)
Large-Scale Integration (LSI)
Very Large-Scale Integration (VLSI)
6. What are the major constituents of a typical Personal Computer or Workstation?
Answer:
Processor
(Main or Primary) Semiconductor Memory
Secondary Store: Hard Disk
Removable Store:
Floppy Disk
CD-ROM
Monitor

Keyboard
Mouse
Printer
Network Card and/or Modem Card
Other (optional) devices and accessories:
Sound card
Scanner
Surge Suppressor
Power Conditioner
"Uninterruptible" Power Supply
7. What is the function of the Clock Generator?
Answer: See page 2-4 in *Bebop Bytes Back.*
8. What is meant by the term "active-low"?
Answer: See text and sidebar on page 2-5 in *Bebop Bytes Back.*
9. Define the term "bus". What are the four major constituents of a bus? Briefly describe the function
of each, and indicate which ones in the "Beboputer" are unidirectional and which are bidirectional.
Answer: Data Bus; Address Bus; Control Bus; and Power Bus. Further explanation is to be
found in *Bebop Bytes Back*, pages 2-6 through 2-9.
10. Name several different kinds of data that can be represented in memory locations in a digital
computer.
Answer:
Instructions
Numbers of different types, of which the simplest type is Non-Negative Integers
Text characters (ASCII, extended ASCII, and EBCDIC)
Bit patterns
11. What is the difference in the electrical connections between ROM and RAM?
Answer: See Figure 2.20 and accompanying text in *Bebop Bytes Back.*
12. Explain how various memory modules (both RAM and ROM) can be organized and connected to
construct the totality of a computer's semiconductor primary memory.
Answer: See **both** Figures 2.21 and 2.22, and also the accompanying text, in *Bebop Bytes Back.*
13. Explain the memory map of the "Beboputer" **and** of the IBM PC.
Answer: For the "Beboputer", see Figure 2.23 and the accompanying text in *Bebop Bytes Back.*
For the IBM PC, either search the internet or find a reference in your local library. Be sure
that you understand and can describe and give memory address ranges for each of the following memory areas:
Conventional Memory Area
Upper Memory Area
Video RAM area

Video adapter card BIOS area
BIOS areas for other adapter cards
System BIOS area
High Memory Area (NOTE: Understanding how this area came into existence, and both the what and the why of its precise dimensions, is an excellent opportunity
to put to use your newly-found expertise in understanding binary and
hexadecimal numbers.)
Extended Memory Area
14. Explain how circuit components can be connected and configured to constitute an Input Port or an
Output Port of specified address.
Answer: See Figures 2.24 and 2.25 **and accompanying text** in *Bebop Bytes Back.*
15. Draw a diagram and explain how a single bit of Magnetic Core memory works.
Answer: See Figure 3.2 and accompanying text in *Bebop Bytes Back.* Note the importance of
the row wires, the column wires, and the sense wire, as explained in the text and illustrated in the right-hand component of Figure 3.2.
16. Why is it that some instructions occupy only one byte of memory, while others occupy two or three
or more bytes?
Answer: See the bottom paragraph on page 3-18, and all of page 3-19, in *Bebop Bytes Back.*
17. Define each of the following terms:
h. Hardware
i. Software
j. Firmware
k. Wetware
l. Vaporware
m. Freeware
n. Shareware
Answer: See page 3-31 in *Bebop Bytes Back.*

Here are some sample questions with answers. There are a few important things to note. First, it always looks much easier when the answers are in front of you. Second, although this is *somewhat* representative of the types of questions you might expect on the exams, these are only example questions. You might find questions that are significantly easier or harder on the exam.
1. Suppose you are designing a digital camera. One of the tasks performed when encoding a
picture is Huffman Encoding. In Huffman encoding, pixel values are each replaced by a code.
The code for pixel values that occur frequently are shorter than those that occur rarely.
We are considering the following options to implement this module:
1. Use a simple microcontroller

2. Use an embedded processor and change the instruction set to suit the application
3. Design custom hardware to implement the module
4. Use a Field-Programmable Gate Array (FPGA)

Rank the above options in terms of performance (the inverse of the time to execute the module on
each platform):

My Answer:
Slowest: 1
Second-Slowest: 2
Third-Slowest: 4
Fastest: 3
You could ask the same thing about design time, cost, power, etc.

2. There are two major FPGA companies: Xilinx and Altera. Xilinx tends to promote its hard processor cores and Altera tends to promote its soft processor cores. What is the difference between a hard processor core and a soft processor core?

My Answer:
A hard processor core is a pre-designed block that is embedded onto the device. In the Xilinx Virtex II-Pro, some of the logic blocks have been removed, and the space that was used for these logic blocks is used to implement a processor. The Altera Nios, on the other hand, is a design that can be compiled to the normal FPGA logic.

3. What does it mean if a reconfigurable device (FPGA or coarse-grained array) is partially-reconfigurable? Why would this be advantageous?

My Answer:
A partially-reconfigurable device is one in which part of the FPGA can be configured while the rest is running. In a multi-tasking system, this might allow us to swap in and out pieces of hardware while the remainder of the "hardware tasks" continue to run.

Distinguish between a real-time system and an embedded system.

Answer: A real-time system is a system in which the ***time*** *that a result of a computation is available is important. An embedded system is a computer that is part of a larger system that controls some physical process. In general, most real-time systems are embedded systems and vice-versa.*

4. List some differences between a real-time operating system and a regular operating system:

Answer:
- *In a regular operating system, the goal is to make everything fast. In a real-time operating system, the goal is to make all times predictable*
- *In a real-time operating system, processes and tasks have deadlines; these deadlines can be hard or soft.*
- *Virtual memory is more useful in a regular operating system than a real-time operating system*
- *Schedulers in a real-time operating system must schedule processes in such a manner as to meet deadlines. This leads to scheduling algorithms like FPS and EDF. In addition, real-time operating systems need some way of avoiding priority inversion, as much as possible.*

• *Real-Time operating systems need more extensive time-management routines.*
5. Why is virtual memory less useful in a real-time operating system than in a regular operating system?
Virtual memory leads to uneven memory access times. If a page is not in memory, it must be swapped in, and this takes time. Since, in a real-time system, we are always concerned with "worst case" delays, we would have to consider the swapping time in any timing analysis. In a regular operating system we are only really concerned with average case delays, so virtual memory works well.
*However, some real-time operating systems do support virtual memory. One feature of virtual memory that *is* useful in a real-time system is protection. By providing protection for memory pages from processes that shouldn't access them, it is easier to verify that a system is correct.*
6. What is POSIX.4?
POSIX.4 is an API that defines a set of routines that an application can call. By calling POSIX.4 routines, an application is theoretically portable across any system supporting POSIX.4. In POSIX.4, the routines primarily have to do with real-time tasks, such as timers, etc.
7. What is the difference between a "bursty" event stream and a "bounded" event stream?
A "bounded" event stream is one in which the minimum (and perhaps maximum) inter-arrival time is known. A "bursty" event stream is one in which the maximum event density is known.
8. In a periodic event stream, what is jitter?
Answer: Jitter is the deviation from the ideal periodic arrival time of an event. Consider an event stream in which an event is supposed to arrive every T time units. Suppose an event that should arrive at nT arrives at nT+ Ä. Then, Ä is the jitter.
9. Suppose you are designing a Rover that will be used to explore Venus. Your rover needs to perform the following tasks:
a) Every 7 sec, you must check the temperature. If the temperature is more than 100 degrees, you must turn on the fan. If it is less than 80 degrees, you should turn off the fan. If it is between 80 and 100 degrees, you should not do anything.
b) Every 10 sec, you must check the tire pressure. If the pressure is less than 40 psi, you should pump in some more air by opening a valve. If the pressure is more than 40 psi, you should close the valve.
c) You must flash a light on and off. The light must be on for 2 sec, then off for 2 sec (and repeat).
You have the following routines available:
get_temperature( ) - gets the temperature in degrees
get_pressure( ) - gets the tire pressure in psi
turn_on_fan( ) - turns on the fan. If the fan is already on, this routine does nothing.
turn_off_fan( ) - turns off the fan. If the fan is already off, this routine does nothing.
open_valve( ) – opens the valve to pump more air into the tire. If the valve is already open, this routine does nothing.
close_valve() – closes the valve to stop putting area into the tire. If the valve is already closed, this routine does nothing
turn_on_light ( ) – turns on the light. If the light is already on, this routine does nothing.
turn_off_light ( ) – turns off the light. If the light is already off, this routine does nothing.

In addition, you have all the POSIX routines available.
You are to write this controller using a *cyclic executive*. Show pseudo-code for your controller. On an exam, you don't have to get the POSIX calls exactly right, but they should be close enough that I can understand which ones you are calling.
Answer:

```
main()
{
count = 0;
lighton = FALSE;
while(1) {
count ++;
if (count % 7 == 0) {
temp = get_temperature();
if (temp < 80) turn_off_fan();
else if (temp > 100) turn_on_fan();
}
if (count % 10 == 0) {
temp = get_pressure();
if (temp < 40) open_Valve();
else close_valve();
}
if (count % 2 == 0) {
if (lighton) {
turn_off_light();
} else {
turn_on_light();
}
}
sleep(1);
}
}
```

10. Explain the difference between mutual exclusion and condition synchronization.
Mutual exclusion deals with protecting a critical section from concurrent access whereas condition synchronization acts to synchronize multiple processes according to a certain condition; condition synchronization can also be used as an ordering mechanism for multiple processes.
11. State a distinct advantage and a distinct disadvantage of a process/thread scheduling framework compared to using a cyclic scheduling framework.
+ easier to handle asynchronous (non-harmonic, sporadic, aperiodic) events
+ easier to handle new tasks
+ faster, uses less CPU
– behaviour not as deterministic
– may lead to resource overloading or starvation
– harder to deal with priority (priority inversion problem)
12. Assume that an operating system provides system calls for performing both synchronous (blocking) I/O and asynchronous (non-blocking) I/O. Further assume that you are required to implement an application using a cyclic executive scheduling

framework where one of the tasks must execute an I/O system call during its execution. Should you use blocking I/O or non-blocking I/O?

Answer:

The key observation is that with a cyclic executive you do not want any task to block since you need predictable timing. Thus, it is important to use non-blocking I/O calls for the task that needs I/O so it will always return in a bounded amount of time.

14. We have been using a non-real-time version of Linux. There have been efforts to develop Real-time versions of Linux. To do this, they divide the OS into two parts: regular Linux and a Real-Time kernel. They then replace all hardware interrupts with software emulated interrupts. Rather than letting regular Linux handle hardware interrupts directly, the interrupts are passed to the Real-Time kernel, which then can decide whether or not to pass the interrupts onto the operating system. Why do you suppose they would do this?

Answer: Suppose the interrupt is from a clock. This clock may be used to invoke the scheduler to run a real-time task. It would be bad if regular Linux somehow received this clock interrupt and did something else with it, or blocked it, or even delayed it, before calling the RT-kernel to schedule the run-time task. By ensuring that this interrupt goes directly to the RT-kernel, better real-time behaviour can be expected.

Pulse Clipper

This interview question is to design a black box that receives clock and cuts every seco
pulse.

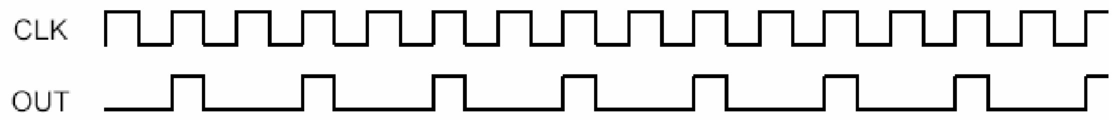


Figure 1: Pulse clipper

As we can see from Figure 1, our black box is a simple clock divider with 25% c
cycle. So, to achieve this we need a controlled clock divider. To avoid glitches, the ou
must be locked in low position when counter changes its state. That means we need
negative-edge trigger as clock divider and one "AND" gate. The timing diagram is sho
on Figure 2.

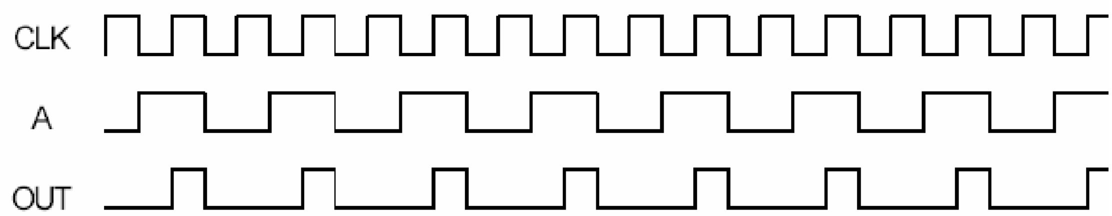


Figure 2: Timing diagram

Implementation of the divider is shown in Figure 3.

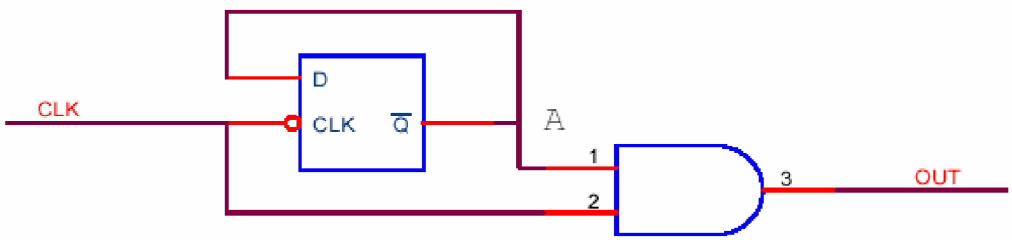


Figure 3: The Solution

24) What ar e t he t ypes of memor y management ?
STD & STT
Draw a Moore STD for a circuit that does the same thing as the circuit in problem 5 and then
make it’ s STT.