VLSI & ASIC Digital design interview questions

Verilog FAQ

Synthesis FAQ

<u>Digital FAQ</u> <u>Digital interview questions Page 1</u> <u>Page 2</u>

<u>Timing FAQ</u> Page 3 Page 4 Page 5

ASIC FAQ

Cmos FAQ

Digital design interview questions & answers.

Misc FAQ

Home

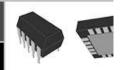
11) What are different ways to synchronize between two clock domains?

Click to view solution

The First Model is Free



Search Now >



1) Explain about setup time and hold time, what will happen if there is setup time and hold tine violation, how to overcome this?

Set up time is the amount of time before the clock edge that the input signal needs to be stable to guarantee it is accepted properly on the clock edge.

Hold time is the amount of time after the clock edge that same input signal has to be held before changing it to make sure it is sensed properly at the clock edge.

Whenever there are setup and hold time violations in any flip-flop, it enters a state where its output is unpredictable: this state is known as metastable state (quasi stable state); at the end of metastable state, the flip-flop settles down to either '1' or '0'. This whole process is known as metastability

2) What is skew, what are problems associated with it and how to minimize it?

Specialise in Data Science, AI, Computer Engineering, IT, Mechanical, Civil & more. Join

In circuit design, clock skew is a phenomenon in synchronous circuits in which the clock signal (sent from the clock circuit) arrives at different components at different times.

This is typically due to two causes. The first is a material flaw, which causes a signal to travel faster or slower than expected. The second is distance: if the signal has to travel the entire length of a circuit, it will likely (depending on the circuit's size) arrive at different parts of the circuit at different times. Clock skew can cause harm in two ways. Suppose that a logic path travels through combinational logic from a source flip-flop to a destination flip-flop. If the destination flip-flop receives the clock tick later than the source flip-flop, and if the logic path delay is short enough, then the data signal might arrive at the destination flip-flop before the clock tick, destroying there the previous data that should have been clocked through. This is called a hold violation because the previous data is not held long enough at the destination flip-flop to be properly clocked through. If the destination flip-flop receives the clock tick earlier than the source flip-flop, then the data signal has that much less time to reach the destination flip-flop before the next clock tick. If it fails to do so, a setup violation occurs, so-called because the new data was not set up and stable before the next clock tick arrived. A hold violation is more serious than a setup violation because it cannot be fixed by increasing the clock period.

Clock skew, if done right, can also benefit a circuit. It can be intentionally introduced to decrease the clock period at which the circuit will operate correctly, and/or to increase the setup or hold safety margins. The optimal set of clock delays is determined by a linear program, in which a setup and a hold constraint appears for each logic path. In this linear program, zero clock skew is merely a feasible point.

Clock skew can be minimized by proper routing of clock signal (clock distribution tree) or putting variable delay buffer so that all clock inputs arrive at the same time

3) What is slack?

Verilog FAQ NEW

Synthesis FAQ

Digital FAQ

Timing FAQ

ASIC FAQ

Cmos FAQ

Misc FAQ

Home

'Slack' is the amount of time you have that is measured from when an event 'actually happens' and when it 'must happen'.. The term 'actually happens' can also be taken as being a predicted time for when the event will 'actually happen'.

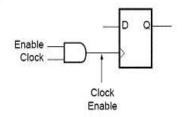
When something 'must happen' can also be called a 'deadline' so another definition of slack would be the time from when something 'actually happens' (call this Tact) until the deadline (call this Tdead). Slack = Tdead - Tact.

Negative slack implies that the 'actually happen' time is later than the 'deadline' time...in other words it's too late and a timing violation....you have a timing problem that needs some attention.

4) What is glitch? What causes it (explain with waveform)? How to overcome it?

The following figure shows a gated clock. The gated clock's corresponding timing diagram shows that this implementation can lead to clock glitches, which can cause the flip-flop to clock at the wrong time.

a) Gated Clock



b) Corresponding Timing Diagram

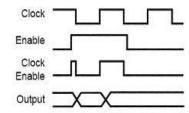


Figure 2-12: Gated Clock

The following figure shows a synchronous alternative to the gated clock using a data path. The flip-flop is clocked at every clock cycle and the data path is controlled by an enable. When the enable is Low, the multiplexer feeds the output of the register back on itself. When the enable is High, new data is fed to the flip-flop and the register changes its state

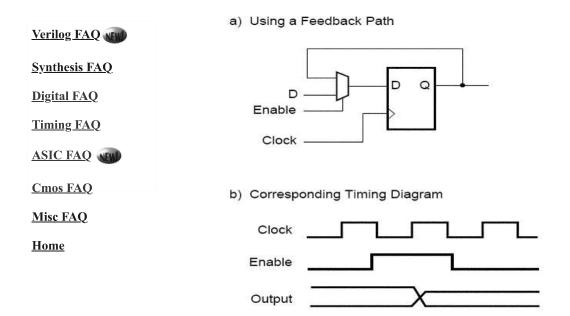


Figure 2-13: Synchronous Design Using Data Feedback

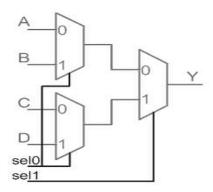
5) Given only two xor gates one must function as buffer and another as inverter?

Tie one of xor gates input to 1 it will act as inverter. Tie one of xor gates input to 0 it will act as buffer.

6) What is difference between latch and flipflop?

The main difference between latch and FF is that latches are level sensitive while FF are edge sensitive. They both require the use of clock signal and are used in sequential logic. For a latch, the output tracks the input when the clock signal is high, so as long as the clock is logic 1, the output can change if the input also changes. FF on the other hand, will store the input only when there is a rising/falling edge of the clock.

7) Build a 4:1 mux using only 2:1 mux?



Difference between heap and stack?

The Stack is more or less responsible for keeping track of what's executing in our code (or what's been "called"). The Heap is more or less responsible for keeping track of our objects (our data, well... most of it - we'll get to that later)

Think of the Stack as a series of boxes stacked one on top of the next. We keep track of what's going on in our application by stacking another box on top every time we call a method (called a Frame). We can only use what's in the top box on the stack. When we're done with the top box (the method is done executing) we throw it away and proceed to use the stuff in the previous box on the top of the stack. The Heap is similar except that its purpose is to hold information (not keep track of execution most of the time) so anything in our Heap can be accessed at any time. With the Heap, there are no constraints as to what can be accessed like in the stack. The Heap is like the heap of clean laundry on our bed that we have not taken the time to put away yet - we can grab

what we need quickly. The Stack is like the stack of shoe boxes in the closet where we have to take off the top one to get to the one underneath it.

Verilog FAQ

9) Difference between mealy and moore state machine?

Synthesis FAQ

Digital FAQ

Timing FAQ

ASIC FAQ

Cmos FAQ

Misc FAQ

Home

A) Mealy and Moore models are the basic models of state machines. A state machine which uses only Entry Actions, so that its output depends on the state, is called a Moore model. A state machine which uses only Input Actions, so that the output depends on the state and also on inputs, is called a Mealy model. The models selected will influence a design but there are no general indications as to which model is better. Choice of a model depends on the application, execution means (for instance, hardware systems are usually best realized as Moore models) and personal preferences of a designer or programmer

B) Mealy machine has outputs that depend on the state and input (thus, the FSM has the output written on edges) Moore machine has outputs that depend on state only (thus, the FSM has the output written in the state itself.

Adv and 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.

10) Difference between onehot and binary encoding?

Common classifications used to describe the state encoding of an FSM are Binary (or highly encoded) and One hot.

A binary-encoded FSM design only requires as many flip-flops as are needed to uniquely encode the number of states in the state machine. The actual number of flip-flops required is equal to the ceiling of the log-base-2 of the number of states in the FSM.

A onehot FSM design requires a flip-flop for each state in the design and only one flip-flop (the flip-flop representing the current or "hot" state) is set at a time in a one hot FSM design. For a state machine with 9-16 states, a binary FSM only requires 4 flip-flops while a onehot FSM requires a flip-flop for each state in the design

FPGA vendors frequently recommend using a onehot state encoding style because flip-flops are plentiful in an FPGA and the combinational logic required to implement a onehot FSM design is typically smaller than most binary encoding styles. Since FPGA performance is typically related to the combinational logic size of the FPGA design, onehot FSMs typically run faster than a binary encoded FSM with larger combinational logic blocks

12) How to calculate maximum operating frequency?

13) How to find out longest path?

You can find answer to this in timing.ppt of presentations section on this site

14) 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)?

Click to view solution

15) How to achieve 180 degree exact phase shift?

Never tell using inverter

- a) dcm's an inbuilt resource in most of fpga can be configured to get 180 degree phase shift.
- b) Bufgds that is differential signaling buffers which are also inbuilt resource of most of FPGA can be used.

16) What is significance of ras and cas in SDRAM?

SDRAM receives its address command in two address words.

It uses a multiplex scheme to save input pins. The first address word is latched into the DRAM chip with the row address strobe (RAS).

Following the RAS command is the column address strobe (CAS) for latching the second address word. Shortly after the RAS and CAS strobes, the stored data is valid for reading.

17) Tell some of applications of buffer?

a)They are used to introduce small delays

c)They are used to support high fanout,eg:bufg

18) Implement an AND gate using mux?

b) They are used to eliminate cross talk caused due to inter electrode capacitance due to close routing.

Verilog FAQ NEW

Synthesis FAQ



Digital FAQ

This is the basic question that many interviewers ask, for and gate, give one input as select line, incase if u r giving b as select line, connect one input to logic '0' and other input to a.

Timing FAQ



ASIC FAQ NEW

Cmos FAQ Misc FAQ

Home

19) What will happen if contents of register are shifter left, right?

It is well known that in left shift all bits will be shifted left and LSB will be appended with 0 and in right shift all bits will be shifted right and MSB will be appended with 0 this is a straightforward answer

What is expected is in a left shift value gets Multiplied by 2 eg:consider 0000 1110=14 a left shift will make it 0001 110=28, it the same fashion right shift will Divide the value by 2.

20)Given the following FIFO and rules, how deep does the FIFO need to be to prevent underflow or overflow?

RULES:

- 1) frequency(clk A) = frequency(clk B) / 4
- 2) $period(en_B) = period(clk_A) * 100$
- 3) duty cycle(en B) = 25%

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.

www.asic.co.in/Index_files/Digital_interview_questions1.htm