Questions on Logic Synthesis

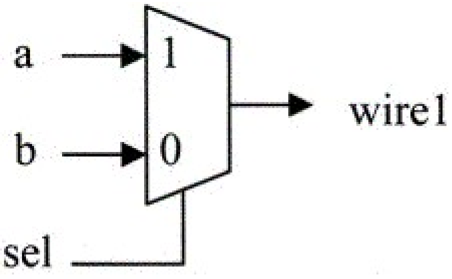
What logic is inferred when there are multiple assign statements targeting the same wire?

It is illegal to specify multiple assign statements to the same wire in a synthesizable code that will become an output port of the module. The synthesis tools give a syntax error that a net is being driven by more than one source.
However, it is legal to drive a three-state wire by multiple assign statements.

What do conditional assignments get inferred into?

Conditionals in a continuous assignment are specified through the “?:” operator. Conditionals get inferred into a multiplexor. For example, the following is the code for a simple multiplexor

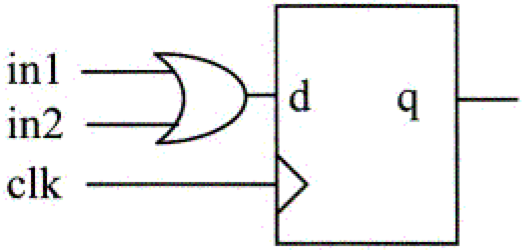
```
assign wire1 = (sel==1'b1) ? a : b;
```



What value is inferred when multiple procedural assignments made to the same reg variable in an always block?

When there are multiple nonblocking assignments made to the same reg variable in a sequential always block, then the last assignment is picked up for logic synthesis. For example

```
always @ (posedge clk) begin
out <= in1^in2;
out <= in1 &in2;
out <= in1|in2;
```



In the example just shown, it is the OR logic that is the last assignment. Hence, the logic synthesized was indeed the OR gate. Had the last assignment been the “&” operator, it would have synthesized an AND gate.

1) What is minimum and maximum frequency of dcm in spartan-3 series fpga?

Spartan series dcm’s have a minimum frequency of 24 MHZ and a maximum of 248

2)Tell me some of constraints you used and their purpose during your design?

There are lot of constraints and will vary for tool to tool ,I am listing some of Xilinx constraints
a) Translate on and Translate off: the Verilog code between Translate on and Translate off is ignored for synthesis.
b) CLOCK_SIGNAL: is a synthesis constraint. In the case where a clock signal goes through combinatorial logic before being connected to the clock input of a flip-flop, XST cannot identify what input pin or internal net is the real clock signal. This constraint allows you to define the clock net.
c) XOR_COLLAPSE: is synthesis constraint. It controls whether cascaded XORs should be collapsed into a single XOR.
For more constraints detailed description refer to constraint guide.

3) Suppose for a piece of code equivalent gate count is 600 and for another code equivalent gate count is 50,000 will the size of bitmap change?in other words will size of bitmap change if gate count changes?

The size of bitmap is irrespective of resource utilization, it is always the same,for Spartan xc3s5000 it is 1.56MB and will never change.

4) What are different types of FPGA programming modes?what are you currently using ?how to change from one to another?

Before powering on the FPGA, configuration data is stored externally in a PROM or some other nonvolatile medium either on or off the board. After applying power, the configuration data is written to the FPGA using any of five different modes: Master Parallel, Slave Parallel, Master Serial, Slave Serial, and Boundary Scan (JTAG). The Master and Slave Parallel modes

Mode selecting pins can be set to select the mode, refer data sheet for further details.

5) Tell me some of features of FPGA you are currently using?

I am taking example of xc3s5000 to answering the question .

Very low cost, high-performance logic solution for
high-volume, consumer-oriented applications
- Densities as high as 74,880 logic cells
- Up to 784 I/O pins
- 622 Mb/s data transfer rate per I/O
- 18 single-ended signal standards
- 6 differential I/O standards including LVDS, RSDS
- Termination by Digitally Controlled Impedance
- Signal swing ranging from 1.14V to 3.45V
- Double Data Rate (DDR) support

• Logic resources
- Abundant logic cells with shift register capability
- Wide multiplexers
- Fast look-ahead carry logic
- Dedicated 18 x 18 multipliers
- Up to 1,872 Kbits of total block RAM
- Up to 520 Kbits of total distributed RAM

• Digital Clock Manager (up to four DCMs)
- Clock skew elimination

• Eight global clock lines and abundant routing

6) What is gate count of your project?

Well mine was 3.2 million, I don't know yours.!

7) Can you list out some of synthesizable and non synthesizable constructs?

not synthesizable->>>>
initial
ignored for synthesis.
delays
ignored for synthesis.
events
not supported.
real
Real data type not supported.
time
Time data type not supported.
force and release
Force and release of data types not supported.

fork join
Use nonblocking assignments to get same effect.
user defined primitives
Only gate level primitives are supported.

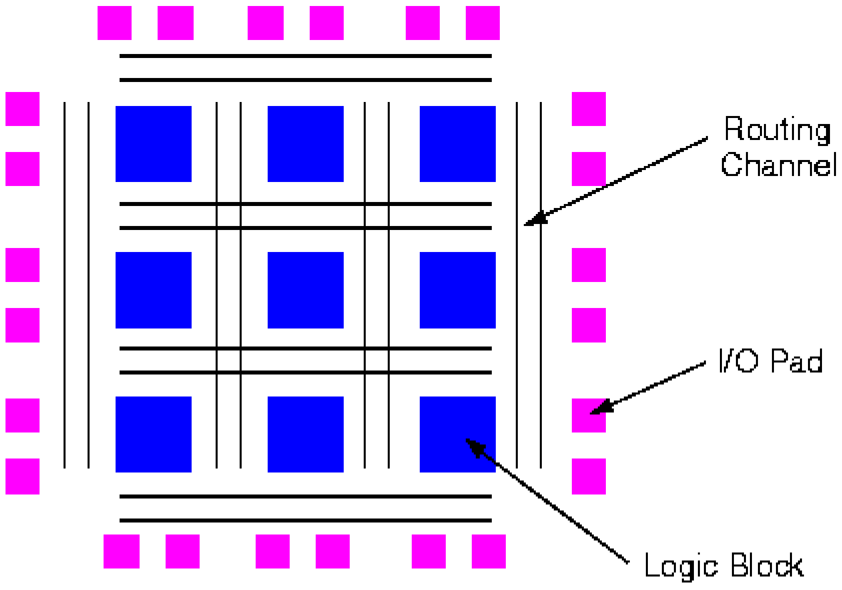
synthesizable constructs->>
assign,for loop,Gate Level Primitives,repeat with constant value...

8)Can you explain what struck at zero means?

These stuck-at problems will appear in ASIC. Some times, the nodes will permanently tie to 1 or 0 because of some fault in manufacturing/layout. To avoid that, we need to provide testability in RTL. If it is permanently 1 it is called stuck-at-1 If it is permanently 0 it is called stuck-at-0.

9) Can you draw general structure of fpga?

FPGA structure



10) Difference between FPGA and CPLD?

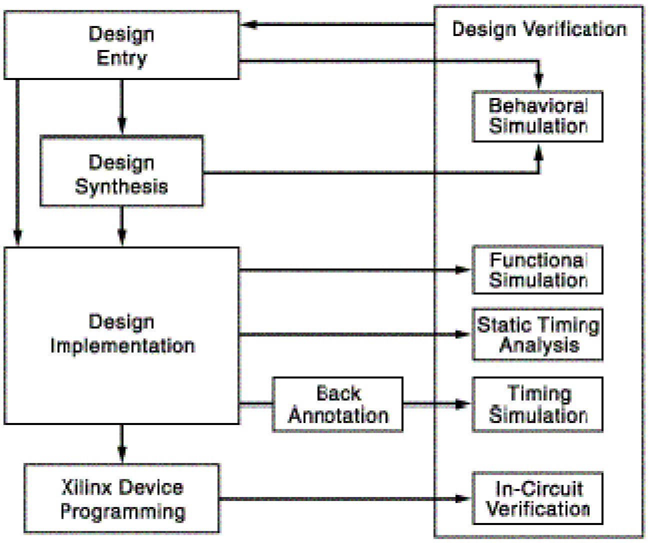
FPGA:
a)SRAM based technology.
b)Segmented connection between elements.
c)Usually used for complex logic circuits.
d)Must be reprogrammed once the power is off.
e)Costly

CPLD:
a)Flash or EPROM based technology.
b)Continuous connection between elements.
c)Usually used for simpler or moderately complex logic circuits.
d)Need not be reprogrammed once the power is off.
e)Cheaper

11) What are dcm's?why they are used?

Digital clock manager (DCM) is a fully digital control system that uses feedback to maintain clock signal characteristics with a high degree of precision despite normal variations in operating temperature and voltage.
That is clock output of DCM is stable over wide range of temperature and voltage , and also skew associated with DCM is minimal and all phases of input clock can be obtained . The output of DCM coming form global buffer can handle more load.

12) FPGA design flow?



Also,Please refer to presentation section synthesis ppt on this site.

13)what is slice,clb,lut?

I am taking example of xc3s500 to answer this question

The Configurable Logic Blocks (CLBs) constitute the main logic resource for implementing synchronous as well as combinatorial circuits.
CLB are configurable logic blocks and can be configured to combo,ram or rom depending on coding style

CLB consist of 4 slices and each slice consist of two 4-input LUT (look up table) F-LUT and G-LUT.

14) Can a clb configured as ram?

YES.

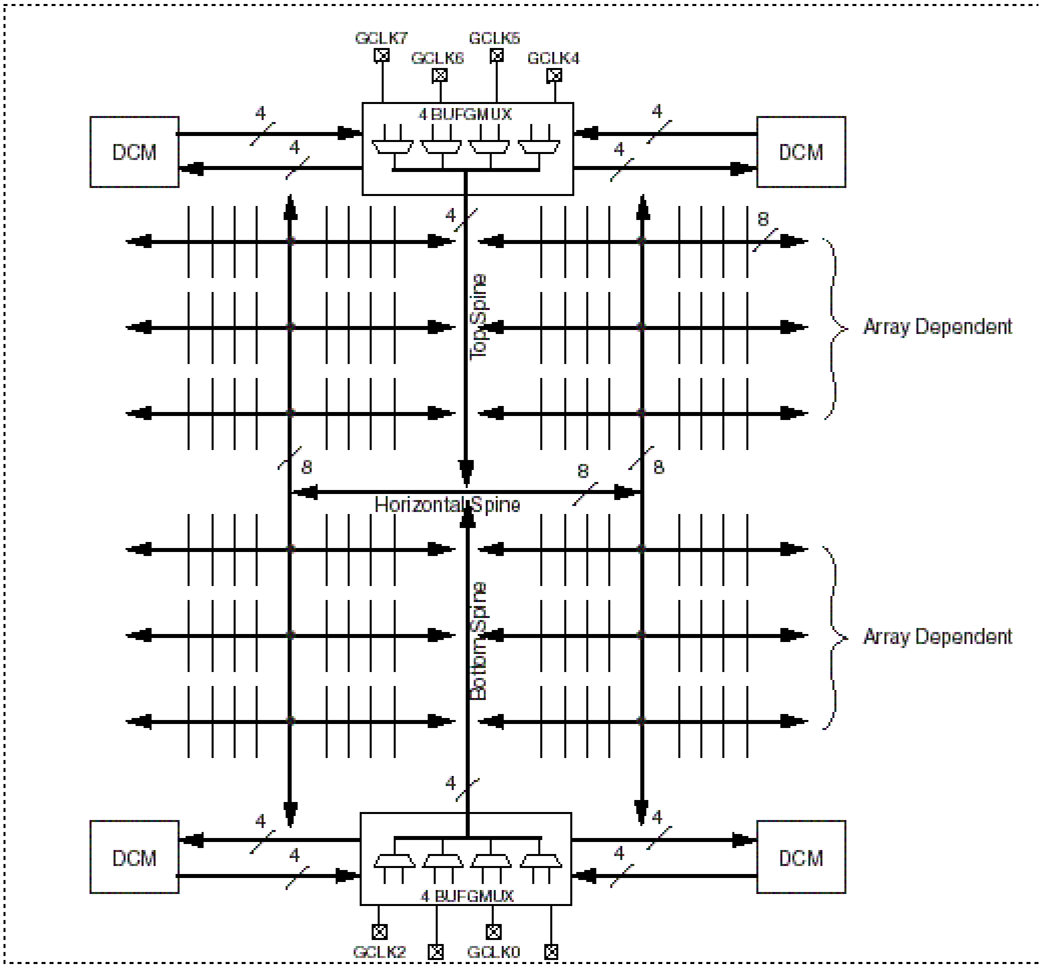
The memory assignment is a clocked behavioral assignment, Reads from the memory are asynchronous, And all the address lines are shared by the read and write statements.

15)What is purpose of a constraint file what is its extension?

The UCF file is an ASCII file specifying constraints on the logical design. You create this file and enter your constraints in the file with a text editor. You can also use the Xilinx Constraints Editor to create constraints within a UCF(extention) file. These constraints affect how the logical design is implemented in the target device. You can use the file to override constraints specified during design entry.

16) What is FPGA you are currently using and some of main reasons for choosing it?

17) Draw a rough diagram of how clock is routed through out FPGA?



18) How many global buffers are there in your current fpga,what is their significance?

There are 8 of them in xc3s5000
An external clock source enters the FPGA using a Global Clock Input Buffer (IBUFG), which directly accesses the global clock network or an Input Buffer (IBUF). Clock signals within the FPGA drive a global clock net using a Global Clock Multiplexer Buffer (BUFGMUX). The global clock net connects directly to the CLKIN input.

19) What is frequency of operation and equivalent gate count of u r project?

20)Tell me some of timing constraints you have used?

21)Why is map-timing option used?

Timing-driven packing and placement is recommended to improve design performance, timing, and packing for highly utilized designs.

22)What are different types of timing verifications?

Dynamic timing:
a. The design is simulated in full timing mode.
b. Not all possibilities tested as it is dependent on the input test vectors.
c. Simulations in full timing mode are slow and require a lot of memory.
d. Best method to check asynchronous interfaces or interfaces between different timing domains.
Static timing:
a. The delays over all paths are added up.
b. All possibilities, including false paths, verified without the need for test vectors.
c. Much faster than simulations, hours as opposed to days.
d. Not good with asynchronous interfaces or interfaces between different timing domains.

23) Compare PLL & DLL ?

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.

24) Given 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.

25)Suggest some ways to increase clock frequency?

· 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.

26)What is the purpose of DRC?

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.

27)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 layout 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.

28)What is DFT ?

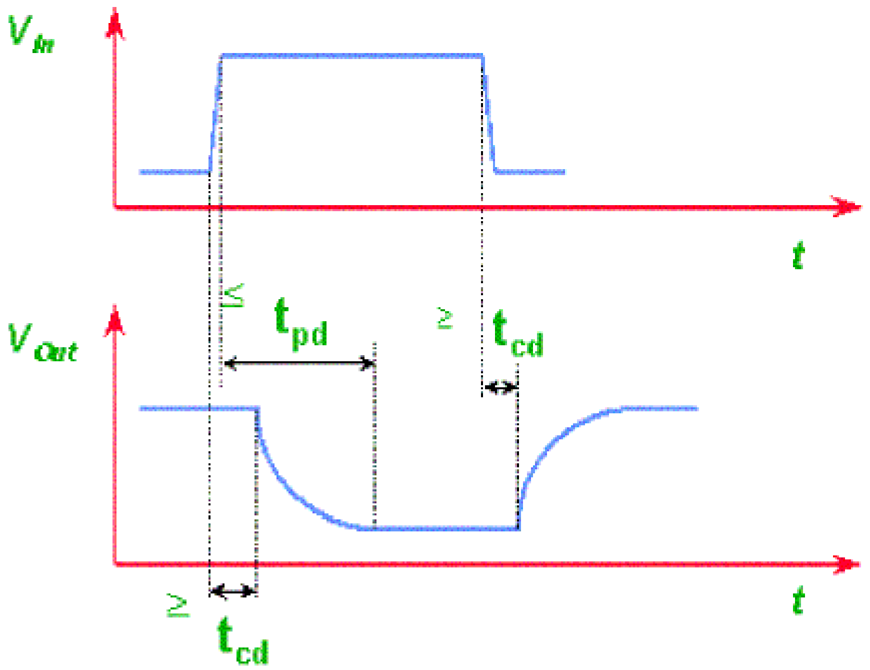
DFT means design for testability. 'Design for Test or Testability' - a methodology that ensures a design works properly after manufacturing, which later facilitates the failure analysis and false product/piece detection
Other than the functional logic,you need to add some DFT logic in your design.This 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)

29) 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?

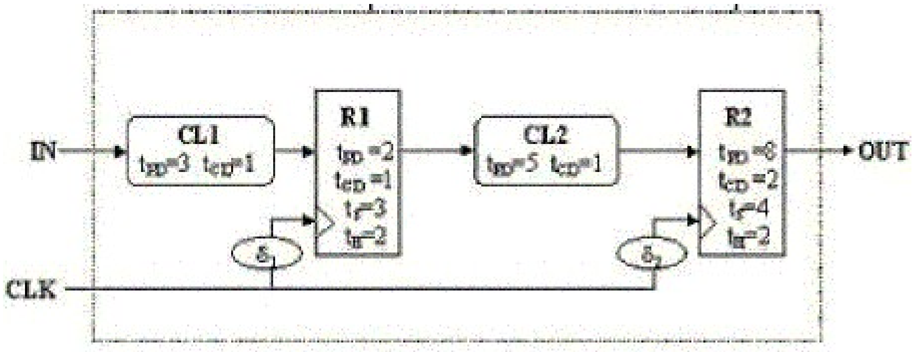
A hard processor core is a pre-designed block that is embedded onto the device. In the Xilinx Virtex II-Pro, some of the logic blocks have been removed, and the space that was used for these logic blocks is used to implement a processor. The Altera Nios, on the other hand, is a design that can be compiled to the normal FPGA logic.

30)What is the significance of contamination delay in sequential circuit timing?

Look at the figure below. tcd is the contamination delay.



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.

31)When are DFT and Formal verification used?

DFT:
· manufacturing defects like stuck at "0" or "1".
· test for set of rules followed during the initial 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 equivalence.

32)What is Synthesis?

Synthesis is the stage in the design flow which is concerned with translating your Verilog code into gates - and that's putting it very simply! First of all, the Verilog must be written in a particular way for the synthesis tool that you are using. Of course, a synthesis tool doesn't actually produce gates - it will output a netlist of the design that you have synthesised that represents the chip which can be fabricated through an ASIC or FPGA vendor.

33)We need to sample an input or output something at different rates, but I need to vary the rate? What's a clean way to do this?

Many, many problems have this sort of variable rate requirement, yet we are usually constrained with a constant clock frequency. One trick is to implement a digital NCO (Numerically Controlled Oscillator). An NCO is actually very simple and, while it is most naturally understood as hardware, it also can be constructed in software. The NCO, quite simply, is an accumulator where you keep adding a fixed value on every clock (e.g. at a constant clock frequency). When the NCO "wraps", you sample your input or do your action. By adjusting the value added to the accumulator each clock, you finely tune the AVERAGE frequency of that wrap event. Now - you may have realized that the wrapping event may have lots of jitter on it. True, but you may use the wrap to increment yet another counter where each additional Divide-by-2 bit reduces this jitter. The DDS is a related technique. I have two examples showing both an NCOs and a DDS in my File Archive. This is tricky to grasp at first, but tremendously powerful once you have it in your bag of tricks. NCOs also relate to digital PLLs, Timing Recovery, TDMA and other "variable rate" phenomena