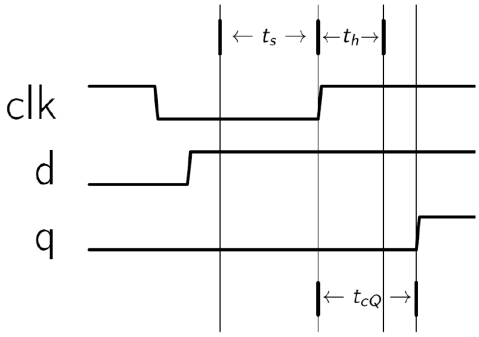
**01-1213 1. Briefly describe the function of each of the following FPGA primitive components:**
**- I/O Blocks**
**- Routing Fabric**
**- Look-Up Tables**

- I/O Blocks - Enable data to enter and exit FPGA
- Routing Fabric - Connects logic blocks and other resources
- Look-Up Tables - Implements combination logic functions

**01-1213 2. Briefly explain each of the following terms, as related to a flip flop**
**- Hold Time (Th)**
**- Clock-to-Q delay (TcQ)**

- Hold Time (Th) - input must not change for a short while after the rising edge of the clock
- Setup Time (Ts) - Desired input should arrive and be stable for a small time before rising edge
- Clock-to-Q delay (TcQ) - Time between arrival of clock edge to output change



**01-1213 3. Which type of primitive in an FPGA allow high speed calculations while saving area?**
**02-1314 / 02-1415 6. What type of FPGA Resource is well suited for implementation of multiplication and adders**
**01-1415 Briefly state 3 benefits of having hard blocks in FPGA, such as DSP Blocks and Block RAMs**

DSP Block in FPGA
- Perform faster calculation
- Wide, Fast adders and hard multipliers
- Resources can be saved and used in other functions
- saving routing space

- Hard blocks can combine for more flexible use (e.g Block RAMs)

**01-1213 4. Briefly describe 3 types of pipeline hazard**
**02-1314 7. What is a control hazard, and how is control hazard associated with branch equal instruction handled?**
- Structural Hazards - Usage of same resource two different ways at the same time
- Data Hazards - Attempt to use item before it is ready
  - Data dependency
- Control Hazards - attempt to make a decision before a condition is evaluated
  - Next instruction fetched before branch occurs

**02-1314 5. Explain briefly why a carry-lookahead adder is often preferred to a standard ripple-carry adder**

Ripple-carry adder - system only can be clocked as fast as critical path where n stages for a n-bit adder. thus, wider adder, slower system

carry-lookahead adder computes carry bits for intermediate stages of adders. This speeds up the process of not having one adder wait for the previous to be completed

**01-1415 Briefly comment on what the synthesis tools may do to simplify the implemented circuit**

FPGA minimize logic expression and store it into look up tables.
map low level circuits to existing hard blocks (e.g DSP, Multiplexers)

**01-1415 Briefly explain what is meant by *functional verification*. How and when is it performed in the design flow?**
**02-1415 What are 3 steps in FPGA design flow in other and describe what happens in each step**

- Design Entry - take design specification and map that graphically to an architecture
- Synthesize - Translates behavioural description into hardware blocks (Multiplexors, Memories)
- Mapping - Maps basic block with resources in FPGA (LUTs, Block RAMS)
- Place and Route
  - Assign each basic block to a location on FPGA
  - Configures the routing to connect this blocks
  - iterates this process to find placement and set of routes that meets timing constraints.

- <u>Algorithmic verification</u>
  - Ensures algorithm selected is suitable for desired application.
  - Even before implementation

- <u>Functional Verification (Design Entry)</u>
  - Checks design implements desired behaviour
  - test individual components, moving up the hierarchy
- <u>Synthesis Verification (Design Entry, Synthesis, Mapping)</u>
  - Check circuit fits target device
  - Initial estimate of timing performance
  - Ensure code are all synthesisable
  - No logic trimmed off unintentionally
- <u>Timing Verification (Place and Route)</u>
  - Ensure timing constraints are met
  - Analyse critical paths if a need to improve timing
  - Simulate final placed and routed design

**01-1516 Briefly explain the purpose and structure of the routing fabric on FPGAs and its impact on the timing of a circuit**

Routing fabric is a mix of wire lengths and segments to help improve performance

**01-1516 Explain why it is possible to use block RAM hard macros in Xilinx FPGAs to implement an 18 bit memory**

Block RAM has two more parity bits to use, thus increasing 16 bit ISA to 18 bits instruction

**What are two types of hazards in Asynchronous circuits**

Static Hazard - One input variable changes causing an output to change momentarily when it shouldn't
Dynamic Hazard - possibility of an output changing more than once as a result of a single input change

**What are two types of FSM**

Mealy - Output dependent on both input & state
Moore - Output dependent on state

**What are the different types of clock definitions?**

- Synchronous Signal - Same frequency with a fixed phase offset to reference clock
  - can sample data without uncertainty
- Mesochronous Signal - Same frequency but unknown phase offset to reference clock
  - Need a synchroniser (couple of register stages or delay)
- Plesiochronous Signal - similar frequency (slightly different) to reference clock
  - FIFO buffer to ensure all data received
- Asynchronous Signal - Can transition at any time
  - Handshaking protocol to ensure proper data transfer

**What are some disadvantages of Synchronous Circuits?**

- Maximum clock rate is determined by slowest path (Critical Path)
- Complex high-fan-out, timing-sensitive clock circuits with large high power clock drivers
  - Very difficult to achieve in large systems
- Nodes switch relative to clock
  - Large current consumption on rising edge
  - Electromagnetic interference
- Transistor-to-transistor variability

**What is MIPS?**

Microprocessor without Interlocking Pipeline Stages

**Why not use ROM or PLA?**

- Becomes large and slow exponentially with more inputs

**What is the difference between ISA and IS?**

Instruction Set Architecture
- Specifies the programming
- Set of all instructions
- Their format
- Specification of their operation

Instruction Set
- Specifies processors's functionality
- Operations it supports
- Storage mechanisms and how it is accessed
- Addressing modes

**What are some important instructions in testbench?**

- $display("text%b %h %o", 12);
- $write("text"); <- No New Line

- <Module> uut(.a(a), .b(b));
- initial begin <Reset Signals>… end
- $finish
- always #5 clk = ~clk;

**What are some distinct FPGA Resources?**

- Flexible Logic
  - basic combinational blocks with significant clocked elements for pipelining
- Flexible Routing
  - significant amount of chip area for wires and switchboxes
  - Mix of wire lengths and segments helps improve performance
- Flexible IO
  - Multi-standard interfacing to external pins
  - Range of speed capabilities
  - Enable data to enter or exit FPGA
- Embedded Hard Modules
  - hard blocks over soft blocks that provide speed and area

**Key Arithmetic Operations**

- $signed({1'b0, data_in}) <- unsigned to signed
- $unsigned(data_in) <- change signed to unsigned
- {8{1'b0}, data_in} <- Zero Extend
- {8{data_in[7], data_in}} <- Sign Extend

**What is data path?**

- Process large amounts of data by
  - exploiting parallelism
  - implementing low level operations