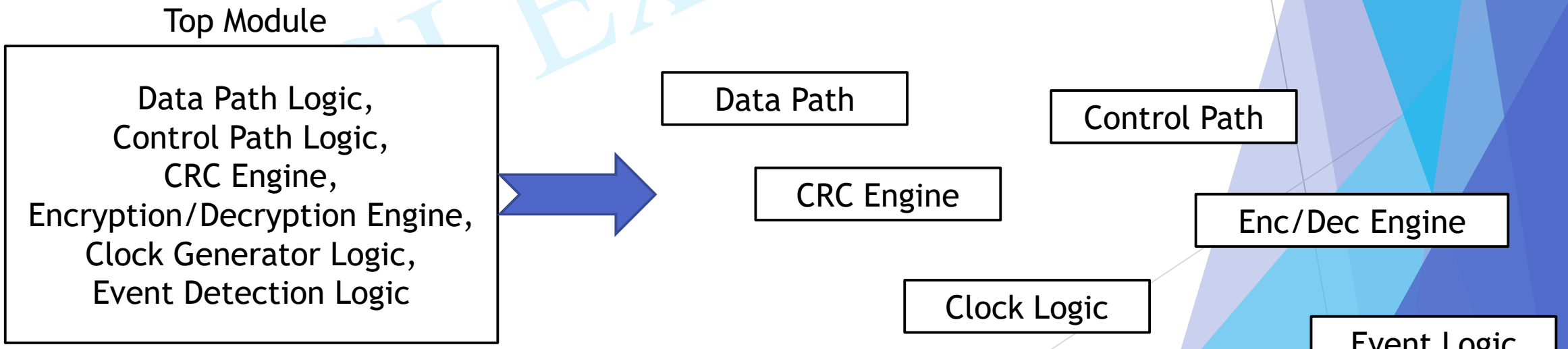
5 RTL Design Best Practices

1

1. Always Partition Your Design into Small Blocks :

- Partition the design into subsystems which is easy to design and test individual

- Use Major/Minor FSMs whenever appropriate



5 RTL Design Best Practices

2. Always Try to Use Same Edge Triggered Flip Flops:

- If there are multiple clock domain , careful about passing the information from one clock domain to another clock domain

- Proper clock domain crossing need to take place in order to not corrupt any data passing between two clock domains

5 RTL Design Best Practices

3. Design Should be "Glitch" Free:

- There is always glitch problem with combinations circuits

- Never drive any critical asynchronous control signal (Clock , Write Enable etc) directly from the output of combinational logic

- Whenever needed, always try to create glitch free signals by registering them

- Always ensure a stable combinational logic output before it is sampled by clock

5 RTL Design Best Practices

4. Synchronize all Asynchronous Control Signals:

- For Example, Synchronize by using N-DFF (Ex : 2-DFF Synchronizer)

- Always have reset synchronizer in your design for external asynchronous reset

5 RTL Design Best Practices

5. Always Avoid use of Tri-State Logic by Design

- Tristate logic is not allowed in RTL code.
- Tristate buffers are large and consume a large amount of power.
- The synthesis, DFT, static timing analysis and place-and-route tools do not handle tristate logic.
- Unidirectional mux logic has been the industry best practice.

Best Free VLSI Content

1. Verilog HDL Crash Course – Link
2. Static Timing Analysis (STA) – Theory Concepts – Link
3. Static Timing Analysis (STA) – Practice/Interview Questions – Link
4. Low Power VLSI Design – Theory Concepts – Link
5. Low Power VLSI Design (LPVLSI) – Practice/Interview Questions – Link
6. Digital ASIC Design Verilog Projects – Link

Please Like, Comment, Share & Subscribe My Channel in Order to Reach Out the Content to a Larger Audience.

Thanks !!