Introduction to FPGA programming

Nanditha Rao

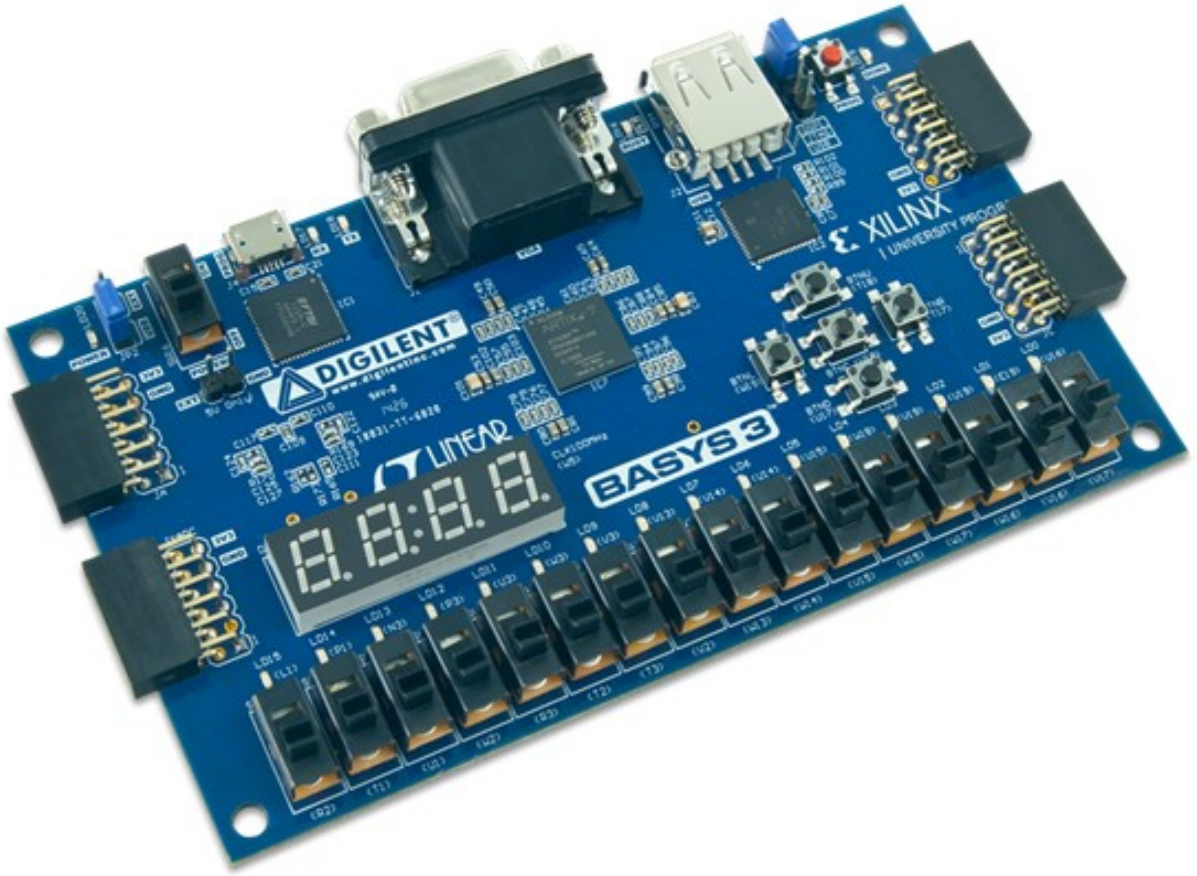
Agenda

- FPGA Introduction
- FPGA Design methodology
- Advantages
- Architecture
- FPGA programming (Verilog, Vivado)

History of programmable logic

- Programmable logic devices:
  - PLA – Programmable logic arrays
  - CPLD – Complex programmable logic device
  - FPGA
  - **Generate customisable hardware**
  - **Study the effect of area, speed, power of the digital circuits**

FPGA



www.xilinx.com

What is an FPGA?

- A “field programmable” gate array: Integrated circuit designed to be configured by a designer
- FPGA configuration is specified using HDL similar to an ASIC (application specific Integrated circuit)
- Logic design in FPGA is different
  - Uses LUTs, Flip-flops, configurable logic blocks

ASIC vs FPGA

ASIC
(Application Specific Integrated Circuit)

- Designed from RTL to layout

- Layout must be sent to semiconductor foundary for fabrication

- Cannot be re-programmed

FPGA (Field Programmable Gate Array)

- Designed from RTL to bitstream

- Design programmed on the FPGA which is bought off-the-shelf

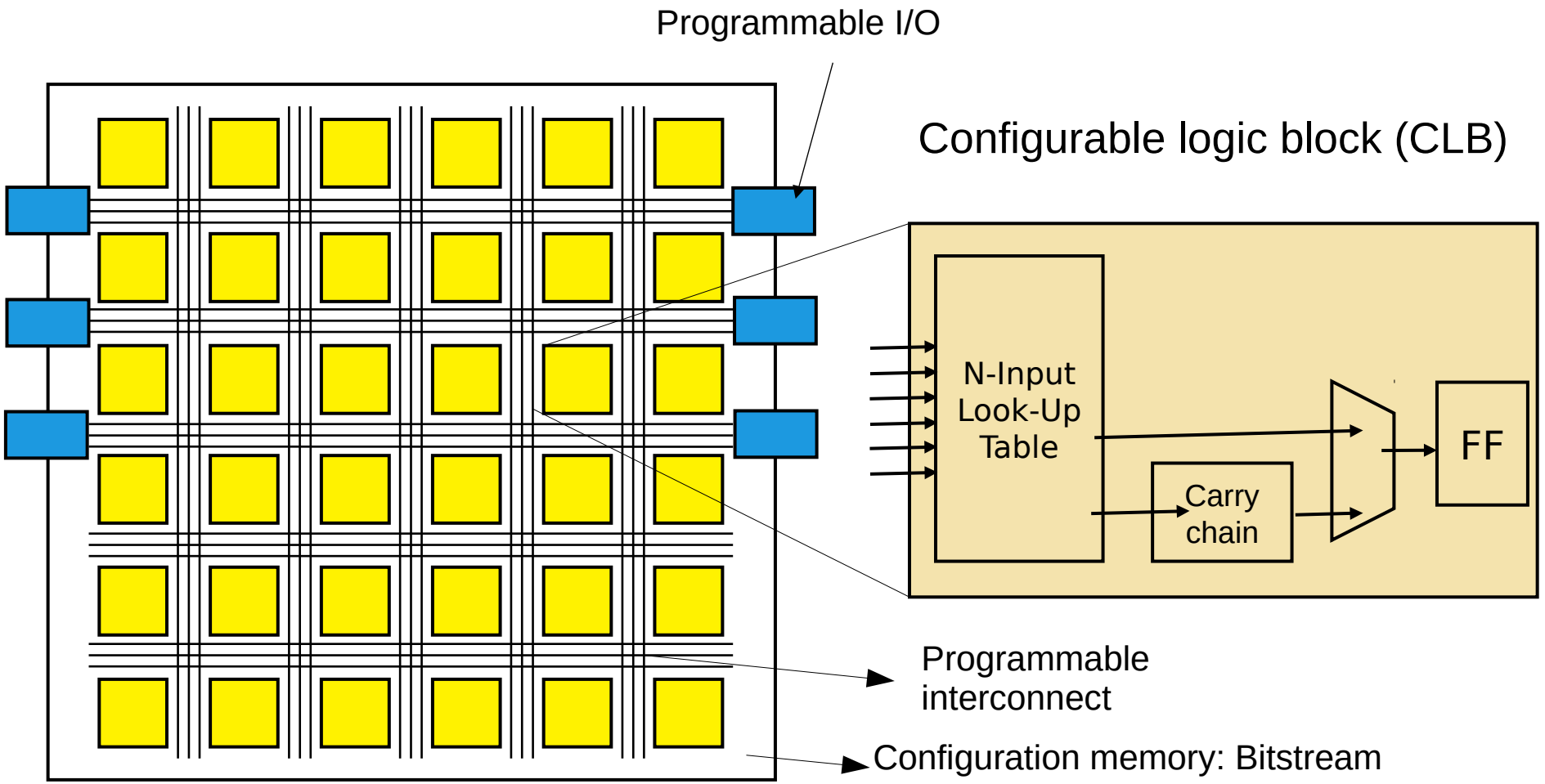
- Can be re-programmed

Applications

- Hardware acceleration
- Signal processing
- Device controllers
- Embedded systems
- Aerospace
- High performance computing
- Machine learning

FPGA architecture

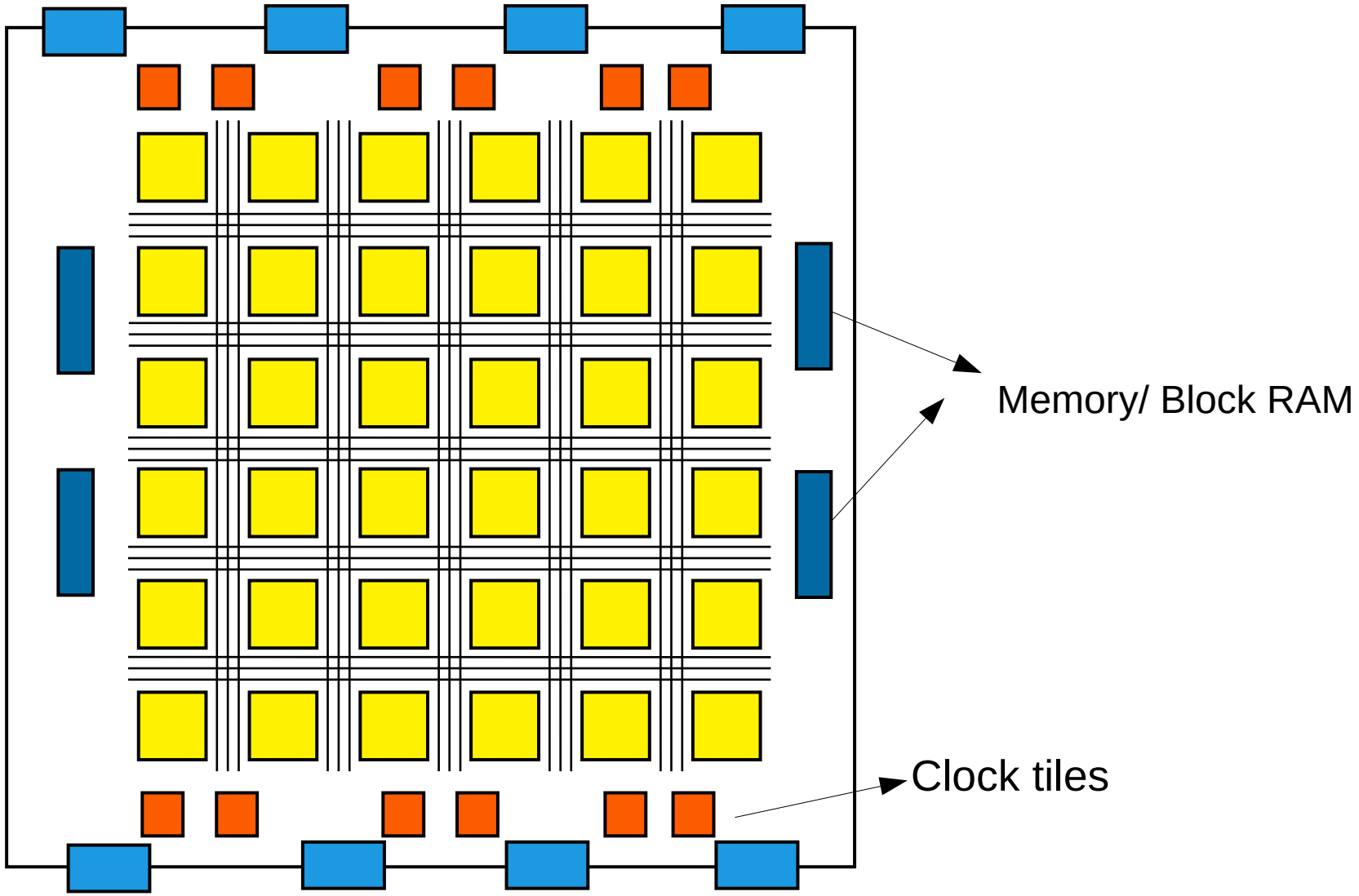
FPGA Architecture



User programs the configuration memory which defines the behavior of the design.

This includes: the connectivity between the CLBs, I/O cells, the logic to be implemented onto the CLBs

..cont

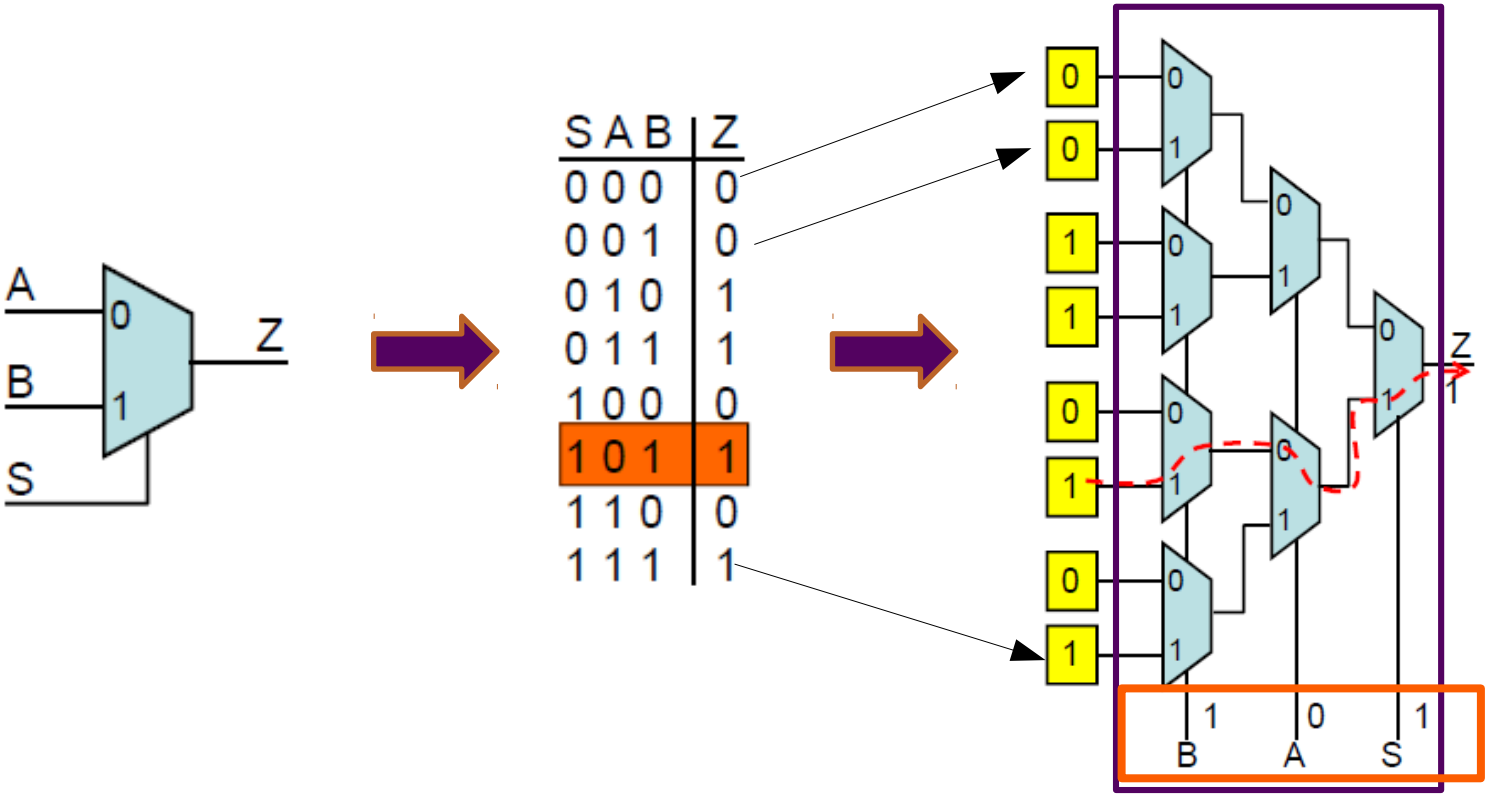


..cont

- Configurable logic blocks (CLB)-- Implement combinatorial and sequential logic. Based on LUT and Flip-flop/latches
  - Look-up Tables (LUT) which implement the logic functions- truth table
  - Carry and Control Logic- Implements arithmetic operations
  - Flip Flops (FFs)/ Latches
- Memory Elements
- Programmable I/O blocks - Configurable I/Os for external interface connections
- Programmable interconnect- Wires to connect inputs, CLBs

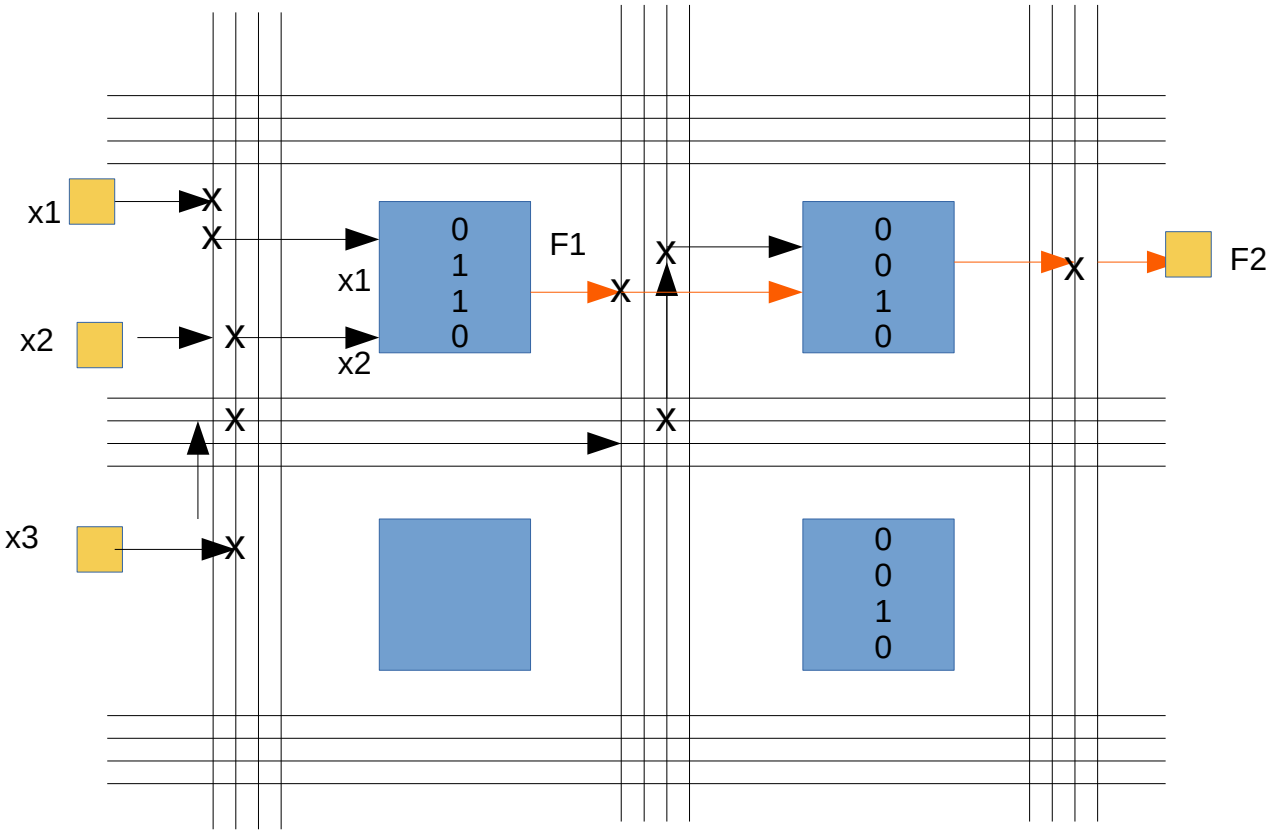
Look-up Tables (eg- 2:1 Mux)

Look-up table with N-inputs can be used to implement any combinational function of N inputs



3 input LUT

Example of FPGA programming



F1 = x1+x2
F2= x1+x2+x3

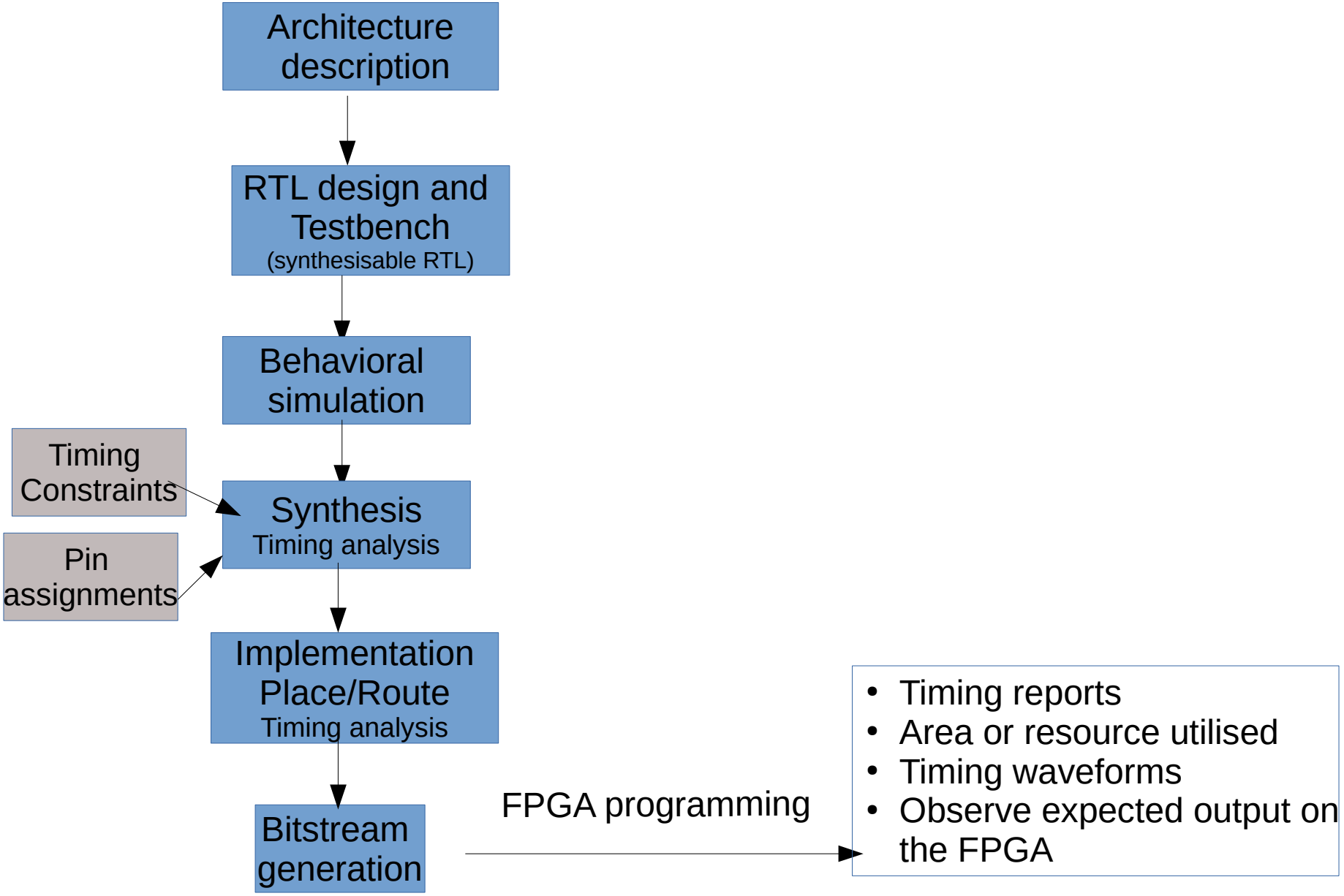
FPGA programming

The “standard” hardware description languages

Many tools provide front-ends to both VHDL/Verilog

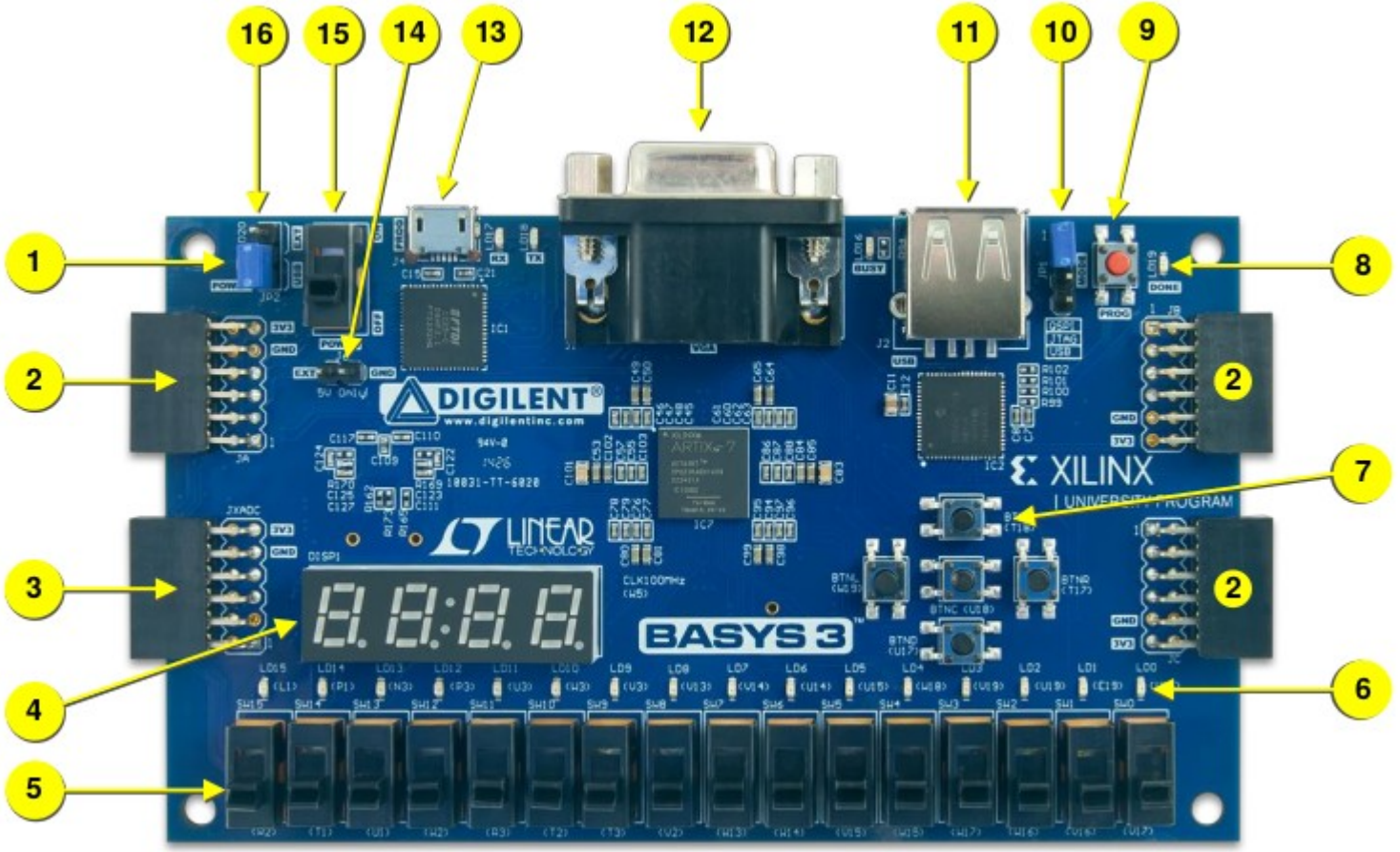
High level programming: C, C++, Python

FPGA design methodology



What cannot be synthesised

- Delays #100 --> Write counters, and generate a signal after a certain count to create a delay
- Initial block (used only in test bench)
- UDP (user defined primitives) – nmos, pmos
- Runtime/Dynamic memory allocation: Indeterminate sizes cannot be synthesised --> Convert into fixed size memory
- Infinite loops --> Convert to a finite loop



15 Power Switch
1 Power good LED
4 Four digit 7-segment display
12 VGA connector
5 Slide switches
13 Shared UART/ JTAG USB port
6 LEDs (16)
14 External power connector
7 Pushbuttons (5)
15 Power Switch
8 FPGA programming done LED

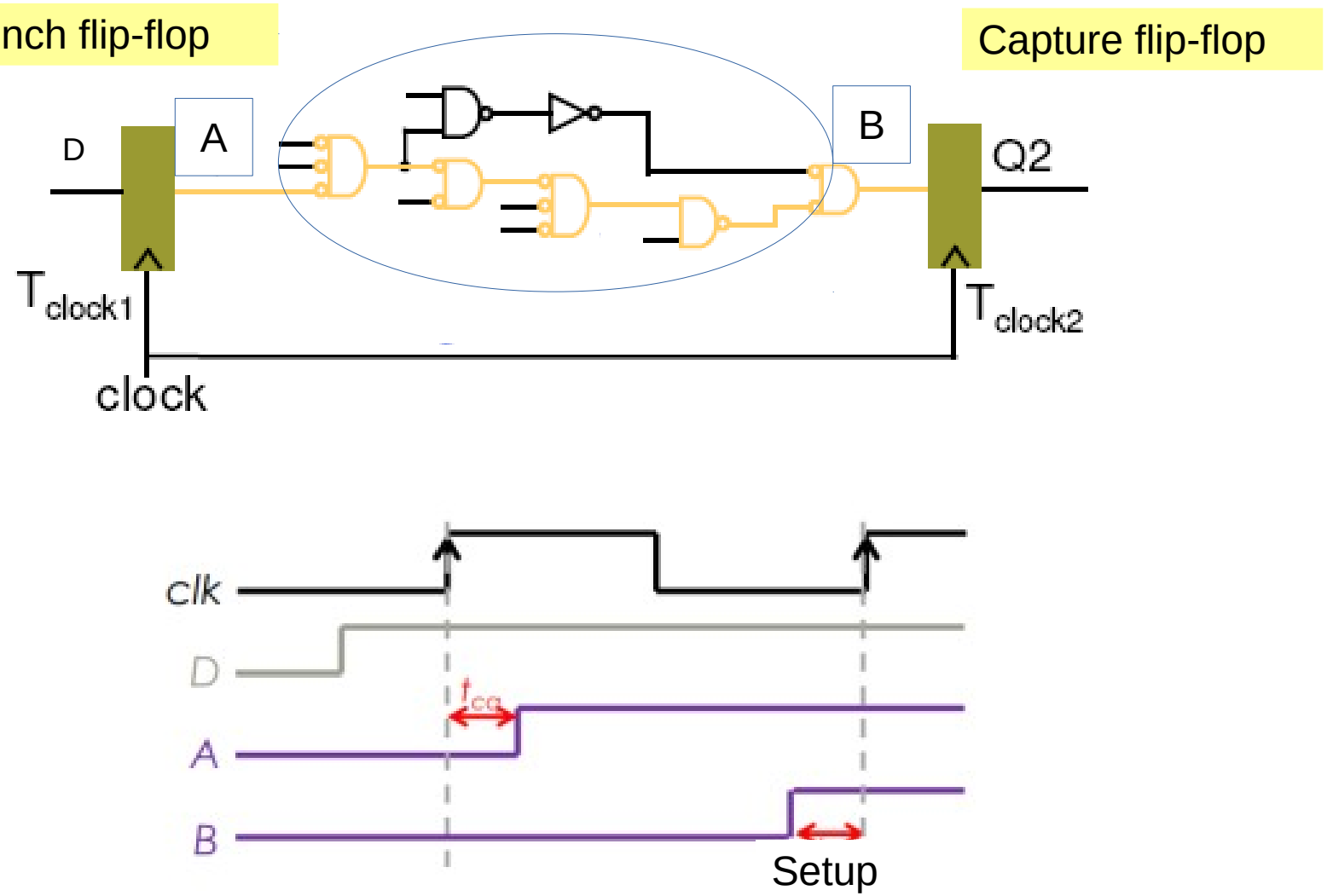
Xilinx Vivado

- Download from Xilinx website.
  - Xilinx Unified Installer 2019.1: Self Extracting Web Installer. A Xilinx signin is needed for this. You can create a login
  - Installation steps: https://www.koheron.com/support/tutorials/install-vivado-2017-1-ubuntu-16-04/

- Download the board files following the procedure at this link: https://reference.digilentinc.com/vivado/installing-vivado/start
  and install them

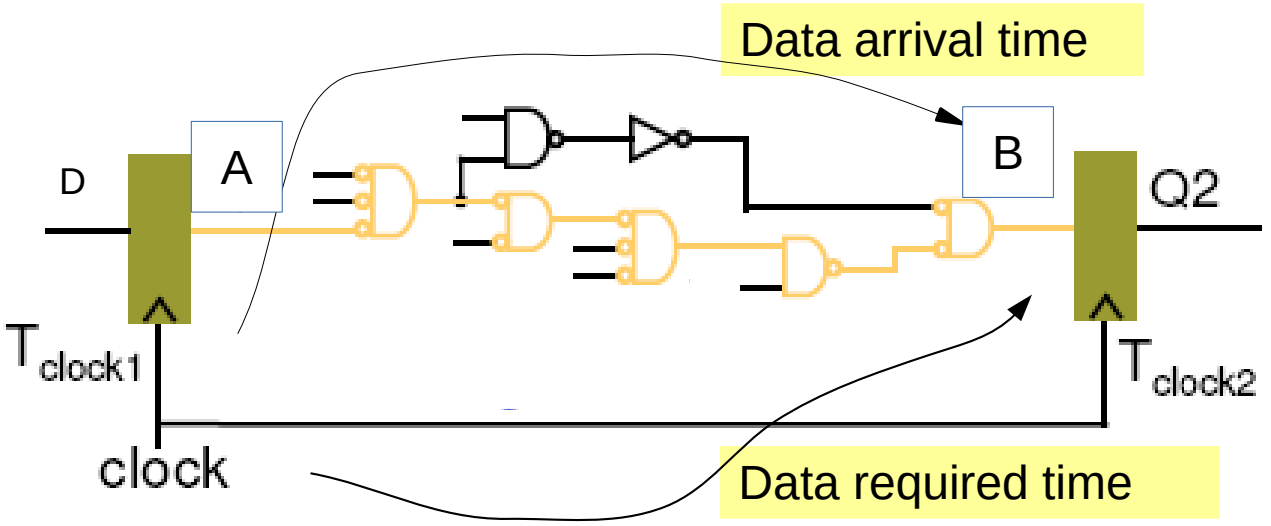
Different ways of programming

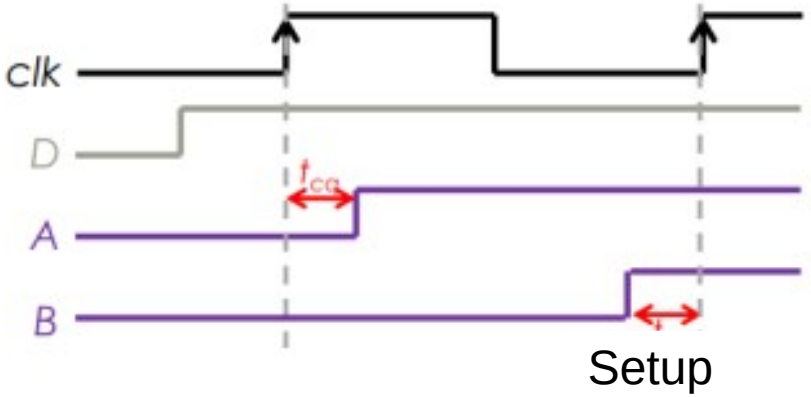
- Local programing on the Basys3 board
- Remote programing
  - Inputs through Virtual Input/Output and Outputs observed on the board
  - Inputs through Virtual Input/Output and Outputs observed on the Integrated Logic Analyzer (ILA)



Setup constraint: After the clock rises, it takes tcq for the data to propagate to point A.

Then the data goes through the delay of the logic to get to point B. The data has to arrive at point B, tsu before the next clock.





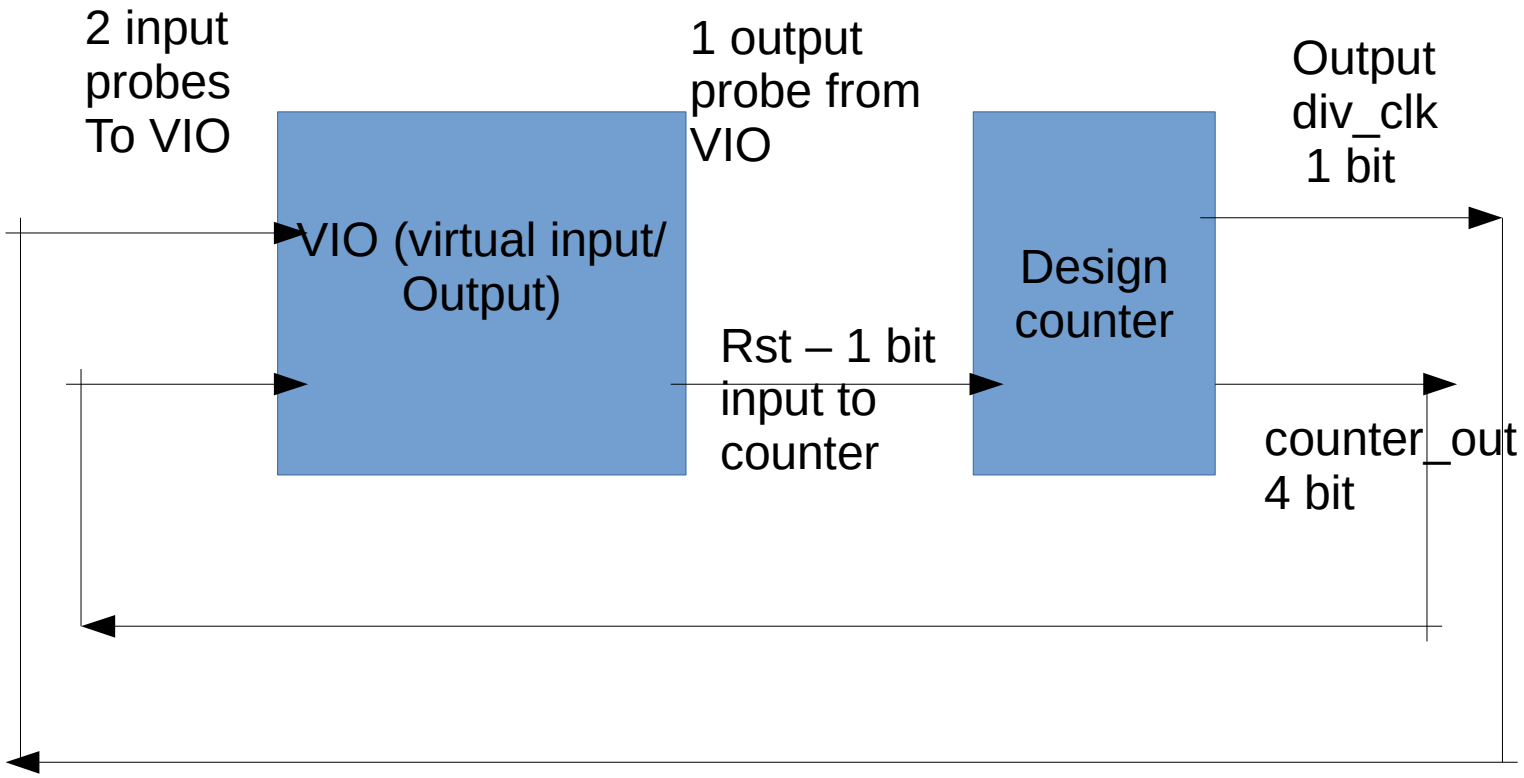
Tcq + Tlogic < (T-Tsetup)

Difference between the data required time and data arrival time is called **slack**

Setup and hold slacks are reported separately

Both need to be positive

Virtual input/output (VIO)



Inputs of the counter will be generated by the VIO. Hence they are VIO outputs

Outputs of the counter will be probed by VIO. Hence, they are VIO inputs

Steps

- Modify the counter code accordingly
  - Your module ports should be modified to make all inputs into wires and outputs as reg, except clock
- Create a VIO template using IP catalog
  - Add probes – input and output
  - Change the probe width accordingly
- Now, click on IP sources -->Instantiation tempalte--> veo (for verilog), vho (for vhdl)
- Copy paste the template inside your code
  - Now, change the module probe names of the VIO template you just copied
  - Inputs and outputs should be assigned according to the comments mentioned there
- Now, the design is ready to simulate

..cont

- After elaboration --> assign pin names.
  - We have only clk. So, assign clk pin to W5 for Basys FPGA --> save constraints
- synthesise-->generate bit stream--> program bitstream-->
- observe VIO outputs
  - Once you program bitstream, in the h/w manager, a VIO window will open.
  - Add probes- using the + symbol in the wondow.