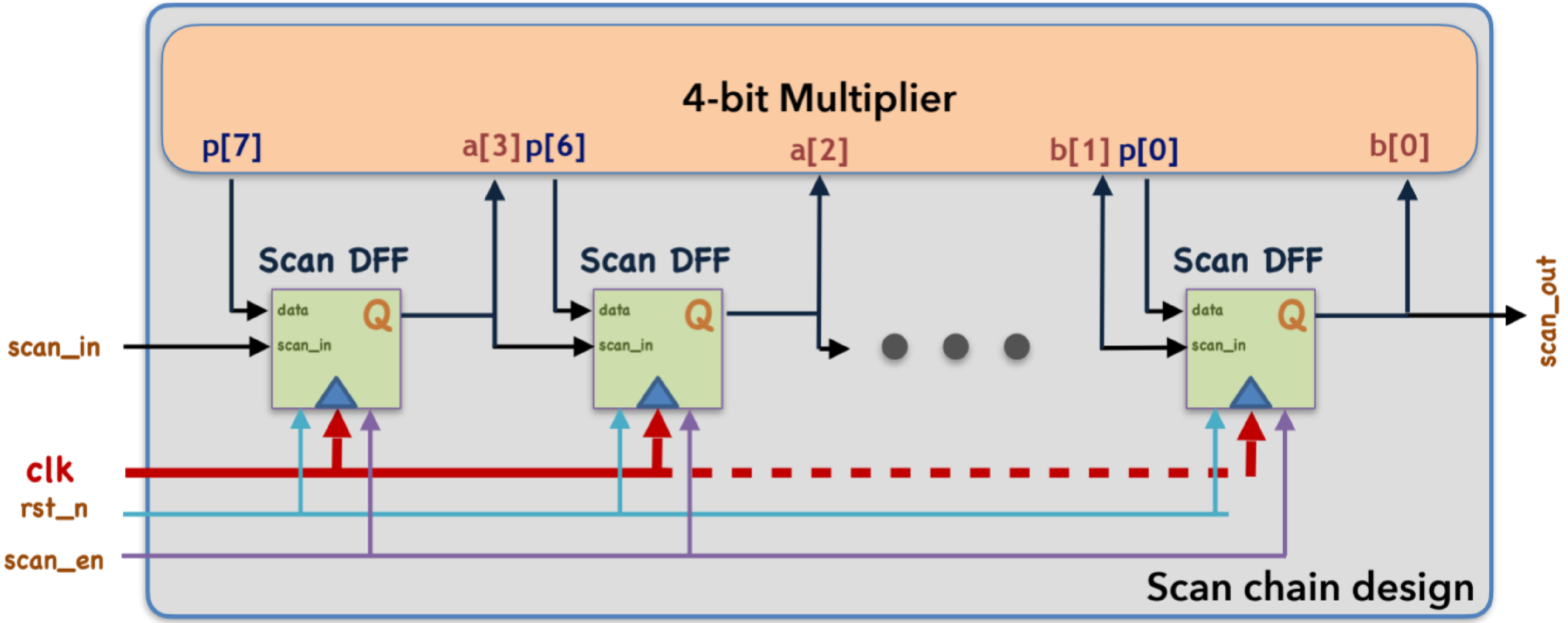
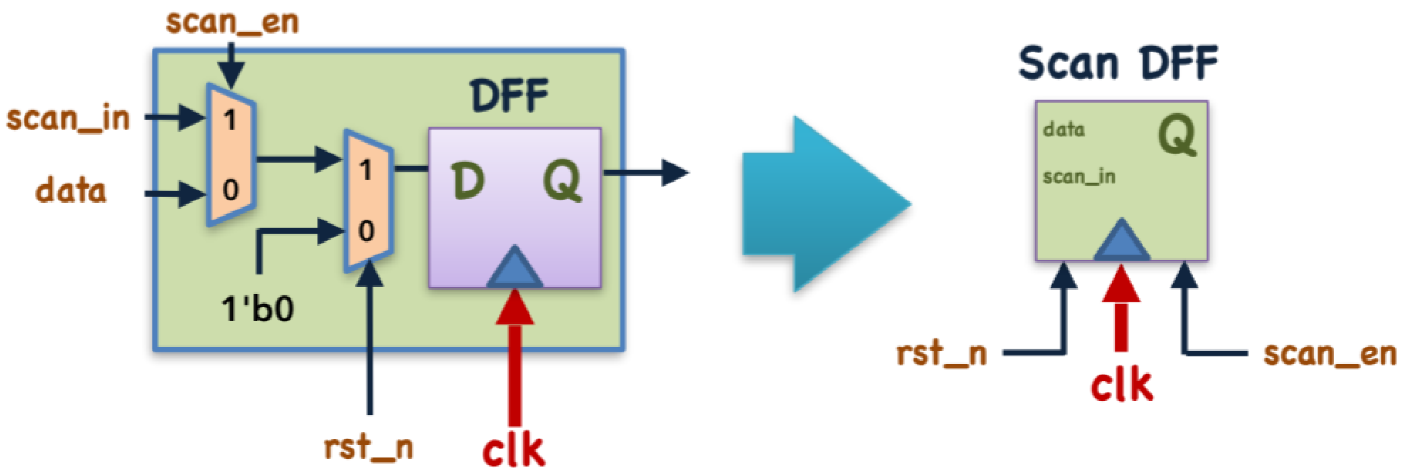
Verilog Advanced Question 2

- **Scan chain design**
- Scan chain is a technique used in design for testing. The objective is to make testing easier by providing a simple way to set and observe every flip-flop in a circuit. The structure of a scan chain is illustrated in the next page.
  - In order to achieve the above objective, the DFFs in a circuits are all replaced by a special type of DFF, called scan DFF (SDFF), which is also shown in the next page. An SDFF contains several extra ports: **scan_in** and **scan_en**, and is larger than the original DFF.
  - All the SDFFs are connected in a chain, which is called a scan chain.
- In this question, you are required to design a scan chain for a 4-bit multiplier, which is a combinational circuit and can be designed by any modeling technique.
  - Input: **clk**, **rst_n**, **scan_in**, **scan_en**
  - Output: **scan_out**
- Reset all SDFFs to **1'b0** when **rst_n** == 1'b0

Verilog Advanced Question 2 (Con't)

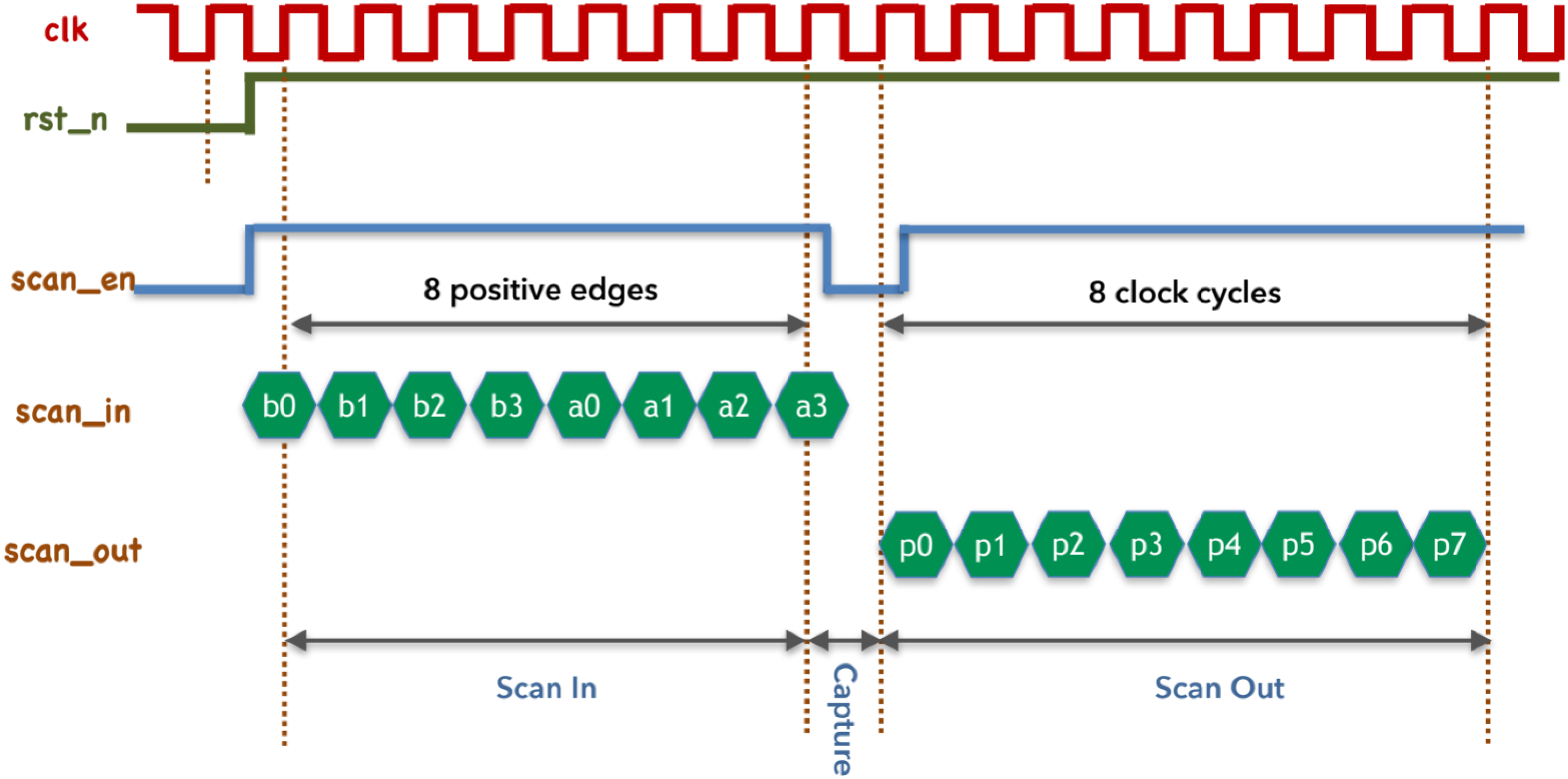




Verilog Advanced Question 2 (Con't)

- **The behavior of a scan chain**
- The behavior of a scan chain contains three phases: **scan in**, **capture**, and **scan out**.
  - **Scan in**
    - In this phase, **scan_en** is set to **1'b1**, and a test pattern is scanned (shifted) from the **scan_in** port into the scan chain bit-by-bit.
  - **Capture**
    - In this phase, **scan_en** is set to **1'b0**, and the circuit performs its original functionality.
    - The inputs of the multiplier is provided by the values stored in SDFF. The output of the multiplier is stored back to the SDFFs at the positive clock edge.
  - **Scan out**
    - In this phase, **scan_en** is set to **1'b1** again, and the values stored in the SDFFs are shifted to the **scan_out** port of the scan chain bit-by-bit.
- In TA's test bench, the **scan_en** signal is controlled **according to this three-phase behavior pattern** to test your scan chain design.
- Please refer to the next page for the example behavior waveform.

Verilog Advanced Question 2 (Con't)


clk
rst_n
scan_en
8 positive edges
8 clock cycles
scan_in
b0
b1
b2
b3
a0
a1
a2
a3
scan_out
p0
p1
p2
p3
p4
p5
p6
p7
Scan In
Capture
Scan Out