CMR
INSTITUTE OF
TECHNOLOGY

USN | | | | | | | | | |

CMRIT

Internal Assesment Test - III

Sub:	VLSI Design							Code:	15EC63
Date:	14 / 05/ 2019	Duration:	90 mins	Max Marks:	50	Sem:	VI	Branch:	ECE- B
Answer any five Questions									

		Marks	OBE	
			CO	RBT
1.	With a relevant diagram discuss manchester carry chain operation.	[10]	CO3	L2
2.	a) Explain the carry select adder with a diagram. b) Discuss the different bus architectures.	[10]	CO3	L2
3.	Explain the architecture of Field programmable Gate Array(FPGA).	[10]	CO4	L2
4.	a) Discuss the FPGA abstractions with a diagram. b) Discuss the Ad-hoc testing.	[10]	CO4	L2
5.	Explain parity generator with basic block diagram and stick diagram.	[10]	CO3	L2
6	Discuss the architectural issues related to subsystem.	[10]	CO3	L2
7	Explain the scan design techniques.	[10]	CO3	L2

CMR
INSTITUTE OF
TECHNOLOGY

USN | | | | | | | | | |

CMRIT

Internal Assesment Test - III

Sub:	VLSI Design							Code:	15EC63
Date:	14 / 05/ 2019	Duration:	90 mins	Max Marks:	50	Sem:	VI	Branch:	ECE-B
Answer any five Questions									

		Marks	OBE	
			CO	RBT
1.	With a relevant diagram discuss manchester carry chain operation.	[10]	CO3	L2
2.	a) Explain the carry select adder with a diagram. b) Discuss the different bus architectures.	[10]	CO3	L2
3.	Explain the architecture of Field programmable Gate Array(FPGA).	[10]	CO4	L2
4.	a) Discuss the FPGA abstractions with a diagram. b) Discuss the Ad-hoc testing.	[10]	CO4	L2
5.	Explain parity generator with basic block diagram and stick diagram.	[10]	CO3	L2
6	Discuss the architectural issues related to subsystem.	[10]	CO3	L2
7	Explain the scan design techniques.	[10]	CO3	L2

1)

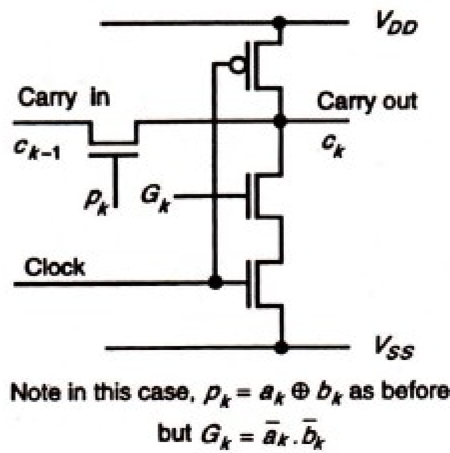


FIGURE Manchester carry-chain element.

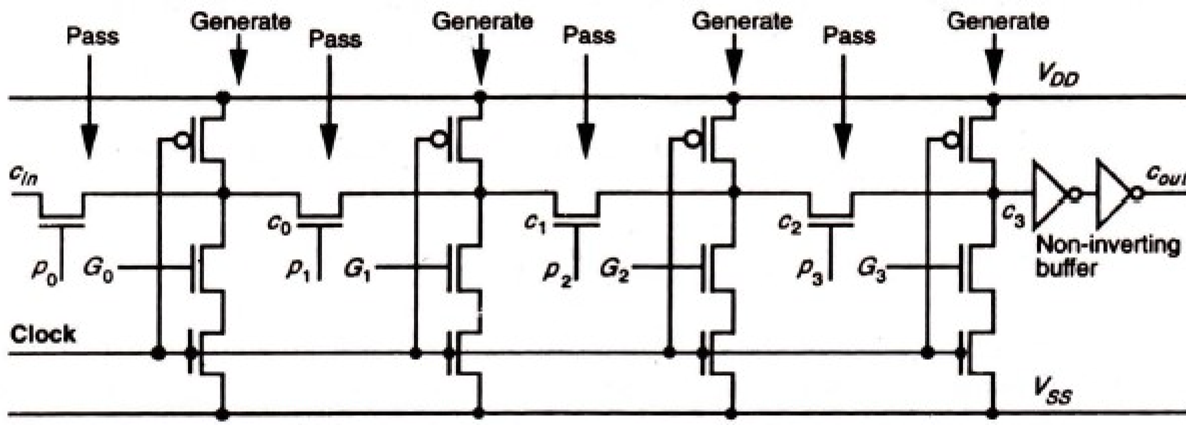


FIGURE Cascaded Manchester carry-chain elements with buffering.

Instead of the carry passing through a complete transmission gate, the carry path is precharged by the clock signal and the carry path may then be gated by a single n-type pass transistor.

Although individual Manchester carry cells are fast, care must be taken when cascading them since this effectively connects pass transistors in series. We have already seen that the delay goes up as the square of n where n is the number connected in series. Obeying the rules set out earlier to cover this situation, we must buffer after every four carry chain cells.

2) **a)Carry select adders**

For this arrangement-also referred to as a conditional sum adder-the adder is divided into blocks. Each block is composed of two adders, one with a logical 0 *carry in* and the other with a logical 1 *carry in.* The *sum* and *carry out* generated are then selected by the actual *carry in* which comes from the *carry out* output of the previous block.

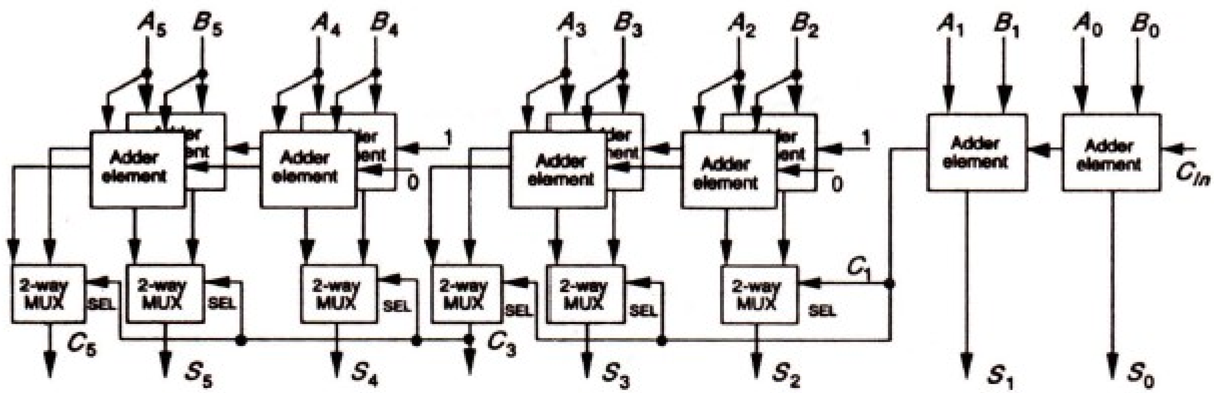


FIGURE Carry select adder structure (6-bit).

Let us consider an n-bit ripple carry adder. The computation time T is given by:

$$T = k_1 n$$

where k_1 is the delay through one adder cell.

If we now divide the adder into blocks, each with two parallel paths, then the completion time T becomes

$$T = k_1 \cdot \frac{n}{2} + k_2$$

where k_2 is the time needed by the multiplexer of the next block to select the actual output

Carry.

2)b)

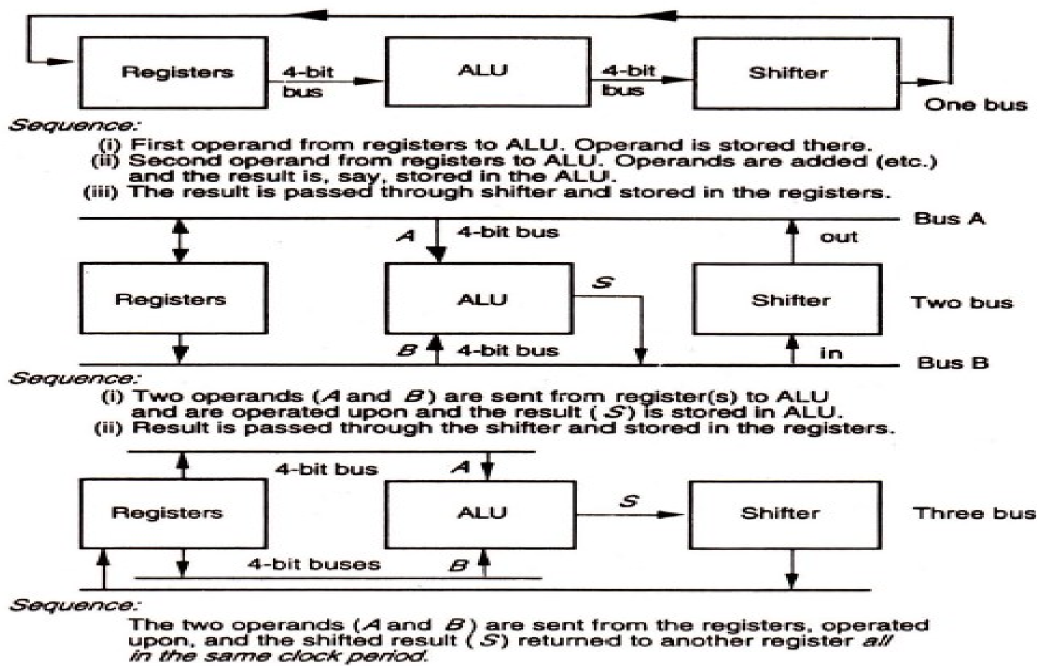


FIGURE Basic bus architectures.

3)

The combinational logic is divided into small units which is known as logic elements(LE) or combinational logic blocks(CLB).

- LE or CLB usually forms the functions of several logic gates.
- Interconnections between these logic elements are made using programmable interconnects.

 This interconnects are logically organized into channels or other units.

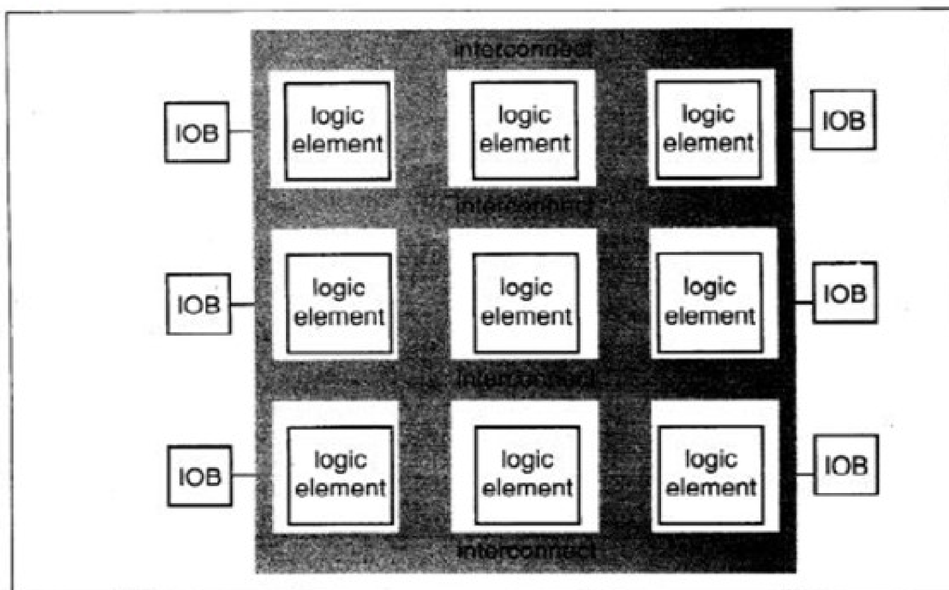
 FPGA offers several interconnects depending on the distance between CLB's that are to be connected: clock signals are provided with their own interconnection networks.

 I/O pins are referred as I/O blocks. These are generally programmable for inputs or outputs and often provides other features such as low power or high speed connection.

FPGA Architecture:

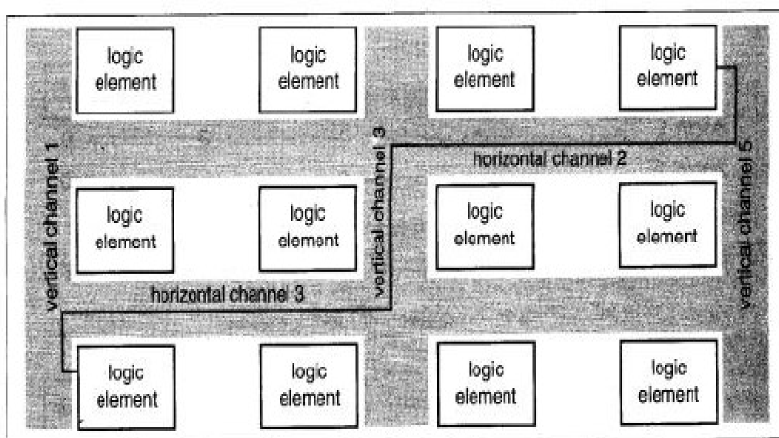
FPGA consists of three major elements.

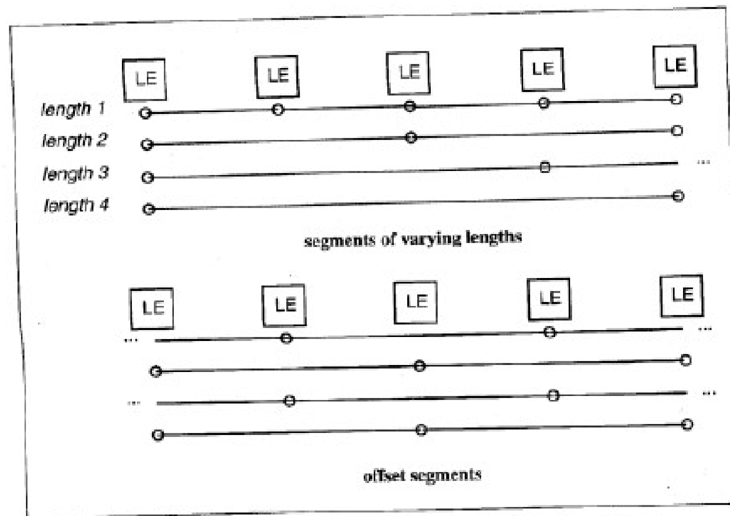
- ✓ Combinational Logic
- ✓ Interconnect
- ✓ I/O pins



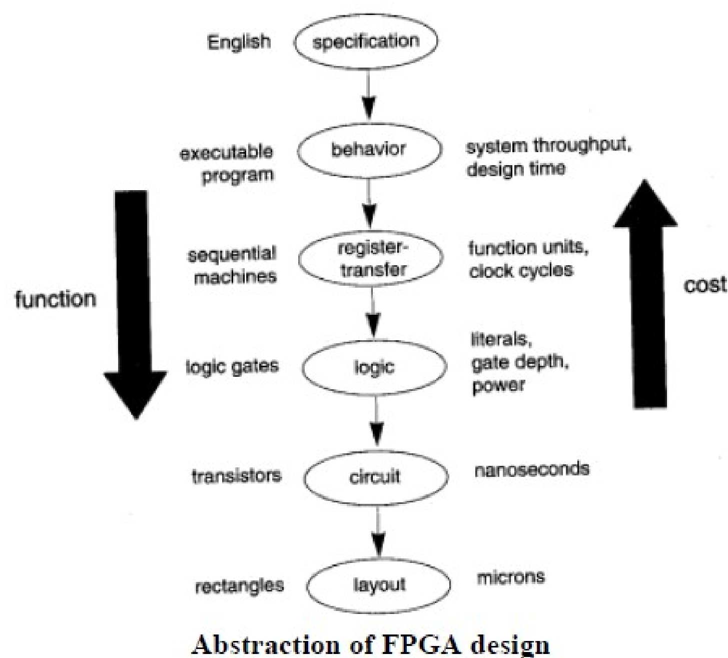
Generic structure of an FPGA fabric.

FPGA Interconnect:





4)a)



Abstraction of FPGA design

Design abstraction is critical to hardware system design.

✓ Hardware designer use multiple levels of design abstraction to manage the design process and ensure that they meet major design goals, such as speed and power consumption.

✓ Design abstraction of the FPGA is as shown in below figure.

- **Behavior:** Explains detailed, executable description of what the chip should do, but it won't describe how it should do it. Example a C program will not mimic the clock cycle-by- clock cycle behavior of the chip, but it will allow us to describe in detail what needs to be computed, error and boundary conditions etc.
- **Register transfer:** The system's time behavior is fully specified- we know the allowed input and output values on every clock cycle but the logic isn't specified as gates. The system is specified as Boolean functions stored in abstract memory elements. Only delay and area estimates can be made from Boolean functions.
- **Logic:** The system is designed in terms of Boolean logic gates, latches and flip-flops.
- **Configuration:** The logic must be placed into logic elements around FPGA and the proper connections must be made between those logic elements. Placement and routing performs these important steps.

Design always requires working down from the top of abstraction hierarchy and up from least abstract description. Work must begin by adding details to abstraction – top-down design add functional detail. Top–down design decisions are made with limited information.

 Bottom – up analysis and design percolates cost information back to higher levels of abstraction: for instance, we may use more accurate delay information from circuit design to redesign the logic. But most designs requires cycles of Top _Down design followed by bottom – up redesign.

4)b)

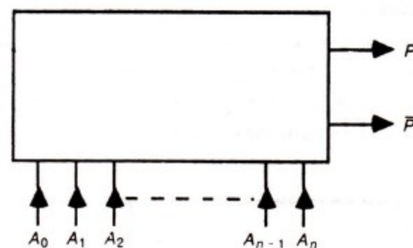
Ad hoc test techniques, as their name suggests, are collections of ideas aimed at reducing the combinational explosion of testing. They are only useful for small designs where scan, ATPG, and BIST are not available. Some of the common techniques for ad hoc testing are:

- Partitioning large sequential circuits
- Adding test points
- Adding multiplexers
- Providing for easy state reset

Some of the examples are: multiplexers can be used to provide alternative signal paths during testing. In CMOS, transmission gate multiplexers provide low area and delay overhead. Use of the bus in a bus-oriented system for test purposes. Here each register is made loadable from the bus and capable of being driven onto the bus. Here, the internal logic values that exist on a data bus are enabled onto the bus for testing purposes.

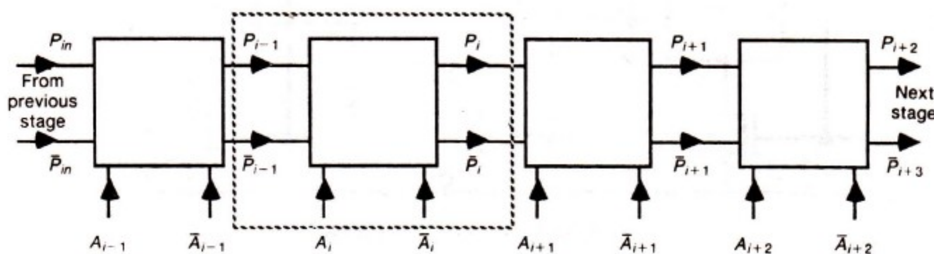
Any design should always have a method of resetting the internal state of the chip within a single cycle or at most a few cycles. Apart from making testing easier, this also makes simulation faster as a few cycles are required to initialize the chip. In general Ad hoc testing techniques represent a bag of tricks.

5) A Parity Generator:



Note: $P = \begin{cases} 1 & \text{Even number of 1s at input} \\ 0 & \text{Odd number of 1s at input} \end{cases}$

FIGURE Parity generator basic block diagram.



Note: Parity requirements are set at the left-most cell where $P_{in} = 1$ sets even and $P_{in} = 0$ sets odd parity.

FIGURE Parity generator—structured design approach.

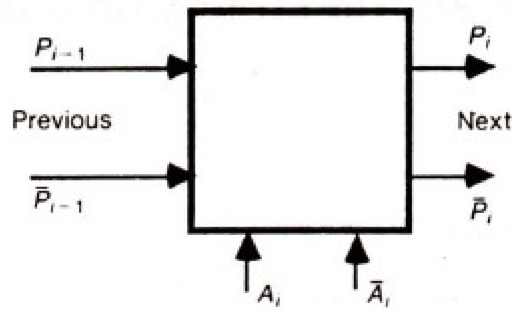


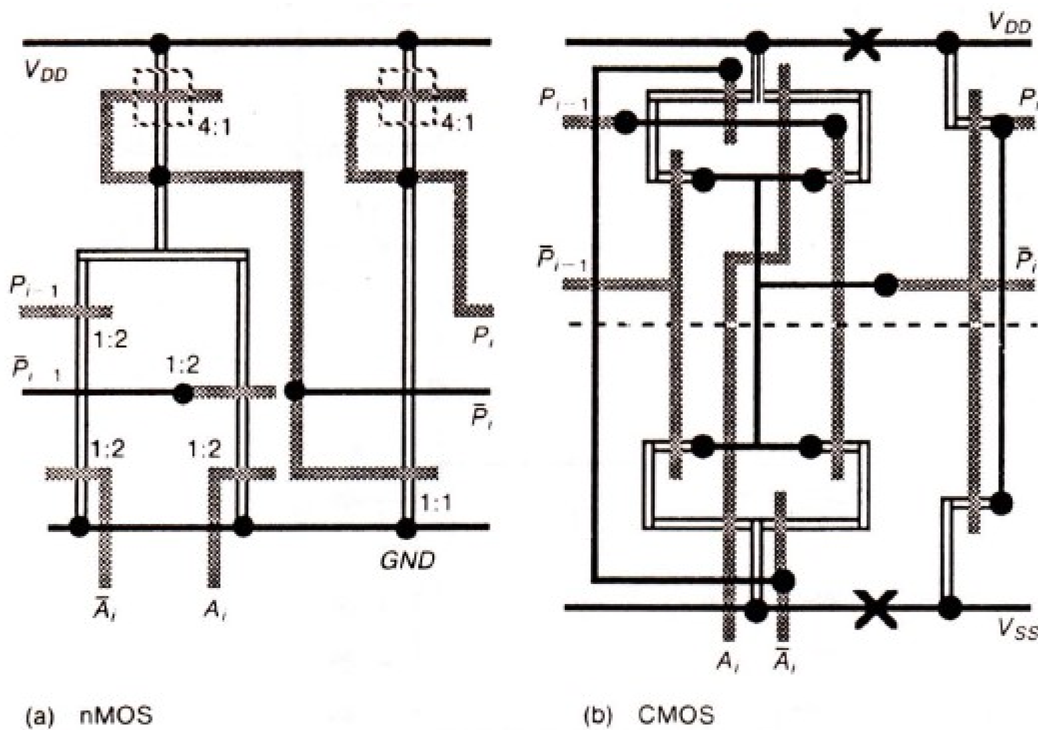
FIGURE **Parity generator—basic one-bit cell.**

A little reflection will readily reveal that the requirements are:

A_i =/ 1 parity is changed, $P_i = \overline{P}_{i-1}$
$A_i = 0$ parity is unchanged, $P_i = P_{i-1}$

A suitable arrangement for such a cell is given in stick diagram form in Figure (nMOS) and 6.18(b) (CMOS). The circuit implements the function

$$P_i = \overline{P}_{i-1} . A_i + P_{i-1} . \overline{A}_i$$



(a) nMOS (b) CMOS

FIGURE **Stick diagrams (parity generator).**

6) Architectural Issues:

As the complexity of a system increases, the design time also increases. Thus while designing we have to adopt those design methodologies which allows handling complexity with reasonable time and reasonable amount of labor.

The following are the guidelines/architectural issues that needs to be considered while designing of the system.

1. Define the requirements carefully and properly.
2. Partition the overall architecture into appropriate subsystems.

3. Communication paths should be carefully selected in order to develop sensible interrelationships between the subsystems.

4. Draw the floor plan of how the system is to map onto the silicon.

5. Aim for regular structures so that design is largely a matter of replication.

6. Draw suitable stick or symbolic diagrams of the leaf-cells of the subsystem.

7. Convert each cell into layout.

8. Carry out design rule check carefully and thoroughly.

9. Simulate the performance of each cell/subsystem.

7) Scan Design:

- The scan-design strategy for testing has evolved to provide observability and controllability at each register.
- • In designs with scan, the registers operate in one of two modes.
- • In normal mode: they behave as expected
- • In scan mode: they are connected to form a giant shift register called a scan chain spanning the whole chip.
- • By applying N clock pulses in scan mode, all N bits of state in the system can be shifted out and new N bits of state can be shifted in. Thus scan mode gives easy observability and controllability of every register in the system.
- • Modern scan is based on the use of scan registers, as shown in Fig. The scan register is a D flip-flop preceded by a multiplexer. When the SCAN signal is deasserted (made to 0), the register behaves as a conventional register, storing data on the D input. When SCAN is asserted (made to 1), the data is loaded from the SI pin, which is connected in shift register fashion to the previous register Q output in the scan chain.
- • To load the scan chain, SCAN is asserted and 8 CLK pulses are given to load the first two ranks of 4-bit registers with data. Then SCAN is deasserted and CLK is asserted for one cycle to operate the circuit normally with predefined inputs. SCAN is then reasserted and CLK asserted eight times to read the stored data out. At the same time, the new register contents can be shifted in for the next test.
- • . Testing proceeds in this manner of serially clocking the data through the scan register to the right point in the circuit, running a single system clock cycle and serially clocking the data out for observation. In this scheme, every input to the combinational block can be controlled and every output can be observed.

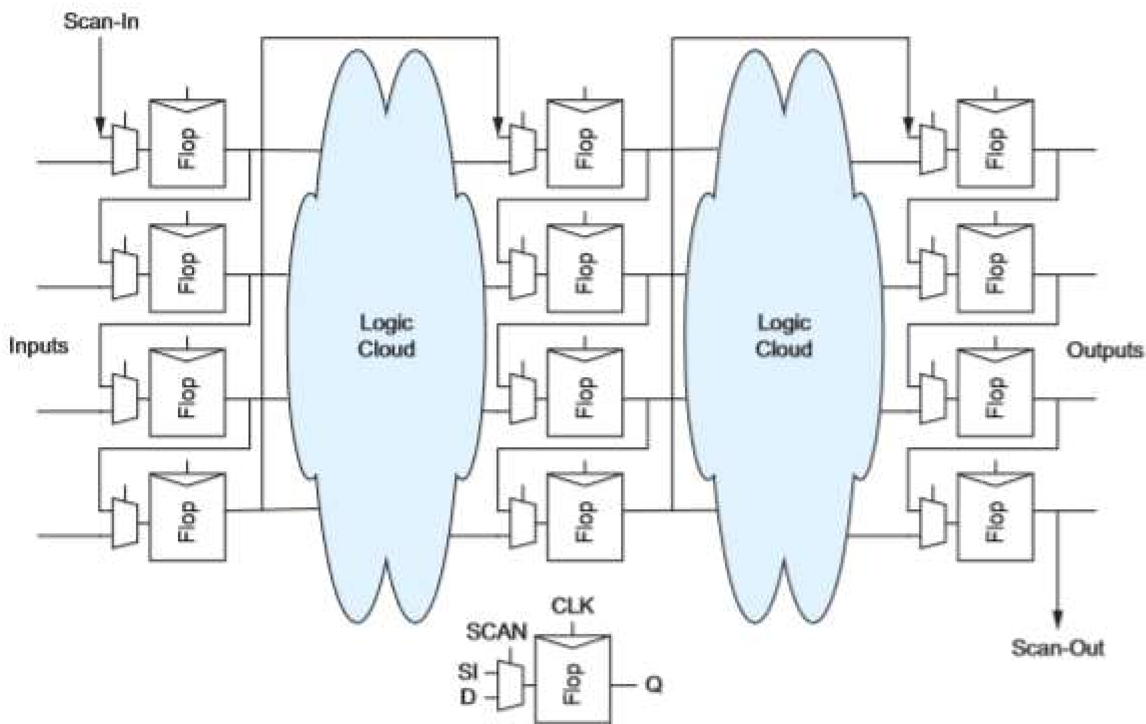


Fig. Scan based testing

Parallel Scan:

Serial scan chains can become quite long, and the loading and unloading can dominate testing time. A simple method/solution is to split the chains into smaller segments. This can be done on a module-by-module basis or completed automatically to some specified scan length. This method is called 'Random Access Scan'.

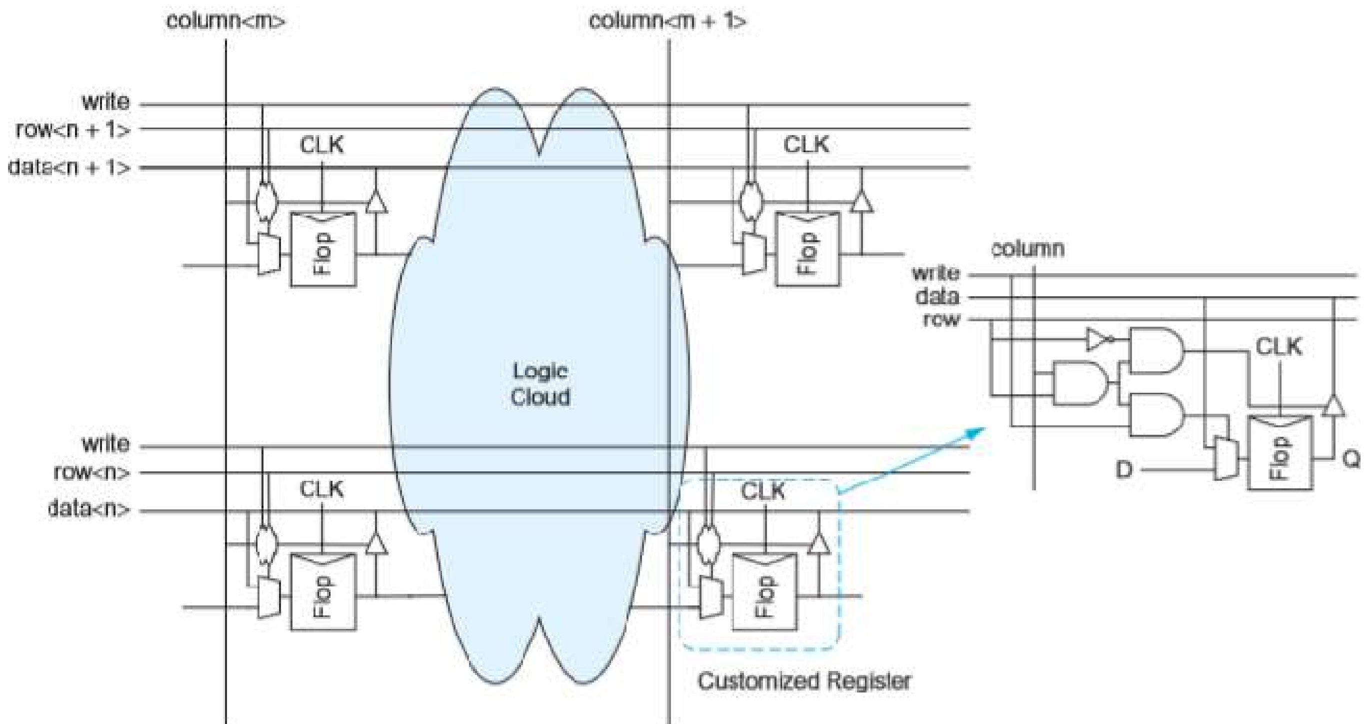


Fig. Parallel Scan - basic architecture