Chapter 7
Field-Programmable-Gate-Array (FPGA)

7.1 Introduction

To be able to implement large-scale SOC designs, minimizing overall power dissipation is a critical [1]. The primary objective of this chapter is to present the results of silicon nanowire technology in a widely utilized prototyping platform called Field-Programmable Gate Array (FPGA). The proposed FPGA architecture in this chapter uses cluster blocks, each of which includes several Look-Up-Tables (LUT) to configure any logic functionality. Each LUT can be configured as a combinatorial logic block or part of a state machine. This flexible configuration is achieved by scan chains implemented inside the cluster block to define the interconnectivity between LUTs and to determine the logic functionality for each LUT. After describing the architectural aspects of the LUT and the cluster, circuit simulation were performed using BSIMSOI SNT models. The chapter reports the results worst-case propagation delays and power dissipation figures of various FPGA circuits and shows typical LUT and the cluster layouts.

7.2 Brief Description of Transistor Design and Modeling

Chapters 1, 2 and 3 show the 3D structure of the device and the complete design process for undoped SNTs. The device design goals were set to achieve minimum static and dynamic power dissipations and minimum intrinsic transient times as discussed in Chapter 3. This criterion resulted in a 2 nm diameter and 10 nm channel length SNTs and the generation of extrinsic BSIMSOI device models to be used for all the circuit simulations in this chapter.

A. Bindal, S. Hamedi-Hagh, *Silicon Nanowire Transistors*,
DOI 10.1007/978-3-319-27177-4_7

7.3 FPGA Architecture

7.3.1 Cluster Architecture

FPGA can be considered as one of the most important application platforms for the SNT technology to test its true potential. An FPGA consists of configuration logic blocks called clusters which contain several LUTs [2]. To program each LUT, scan chains are used. Scan chains are basic shift registers that transmit serial data to configure the logic functionality for each LUT. A typical block diagram of a cluster is shown in Fig. 7.1.

In this figure, an 8-bit wide intercluster bus, line0 through line7, sustains continuous data exchange among clusters. The intercluster data is routed in any available direction using switch boxes placed at four corners of each cluster.

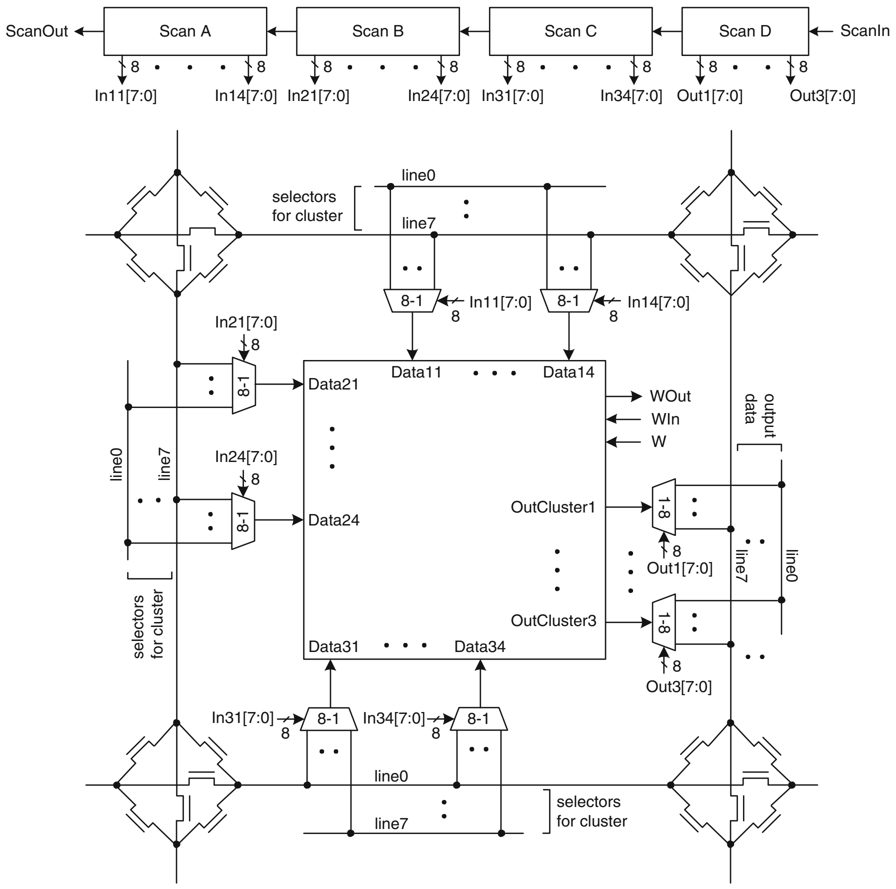


Fig. 7.1 The proposed FPGA architecture with global scan chains

Each bit of the 8-bit intercluster bus is connected to a different switch box composed of six SNTs programmed to guide the incoming data, establishing a configurable path between clusters. A total of 12 data inputs ports, Data11 through Data34, accept incoming signals from neighboring clusters; each input port is connected to the 8-bit wide intercluster bus using an 8-1 MUX. Each cluster has three output ports, OutCluster1, OutCluster2, and OutCluster3, which transfer the cluster data to other clusters for data processing. A particular cluster output can be routed to any one of the eight bits of the intercluster bus using 1-8 DEMUX blocks.

There are two types of scan chains embedded in each cluster. The outputs of the scan chains A, B and C are used as the selector inputs for each 8-1 MUX connected to input ports of the cluster. Once the 8-1 MUX selector is programmed by a scan chain, then incoming signals from neighboring clusters are routed to a specific cluster input for processing. Similarly, scan chain D outputs are used as selectors for each 1-8 DEMUX at each cluster output. Once the selector for each 1-8 DEMUX is programmed, the processed data in a particular cluster is distributed among neighboring clusters for further processing. The serial ports, WIn and WOut, are used to program each LUT of a particular cluster. Serial ports that belong to neighboring clusters are cascaded such that WOut of one cluster is connected to WIn of the other to propagate programming data to all clusters without needing more than one wiring channel. W is another input port that programs each LUT, cluster selector tree and bypass path in each cluster.

Each cluster in Fig. 7.1 is composed of three 4-input Look-Up-Tables (4-LUT) as shown in Fig. 7.2. Each 4-LUT receives external input data, Data11 through Data34, through twelve 4-1 MUXes and can be programmed to implement a primitive or complex logic function requiring up to four inputs. The number of 4-LUTs in each cluster and the intercluster bus width are determined according to the study in [3] to maximize 4-LUT usage and increase available wire utilization between clusters.

There is an additional internal scan chain in each cluster as shown in Fig. 7.2. The primary purpose of this chain is to program all 24 selector inputs, s111 through s342, to control the flow of data into the cluster and to generate program inputs for the three bypass paths in each LUT, bypass1 through bypass3.

7.3.2 4-Input Look-Up-Table (4-LUT)

Each 4-LUT is composed of a 16-bit scan chain at the input and a 16-1 pass-gate MUX tree as shown in Fig. 7.3.

The 16-bit scan chain is used to program a specific logic function for each 4-LUT in the cluster. Each 4-LUT has programming input and output ports, PIn and POut, respectively. PIn of the first LUT, PIn1, receives the programming data directly from WIn of the cluster. POut of the first 4-LUT, POut1, is connected to PIn2 of the second 4-LUT to program the second 4-LUT. Similarly, POut2 of the second 4-LUT is connected to PIn3 to program the third 4-LUT. The serial data at

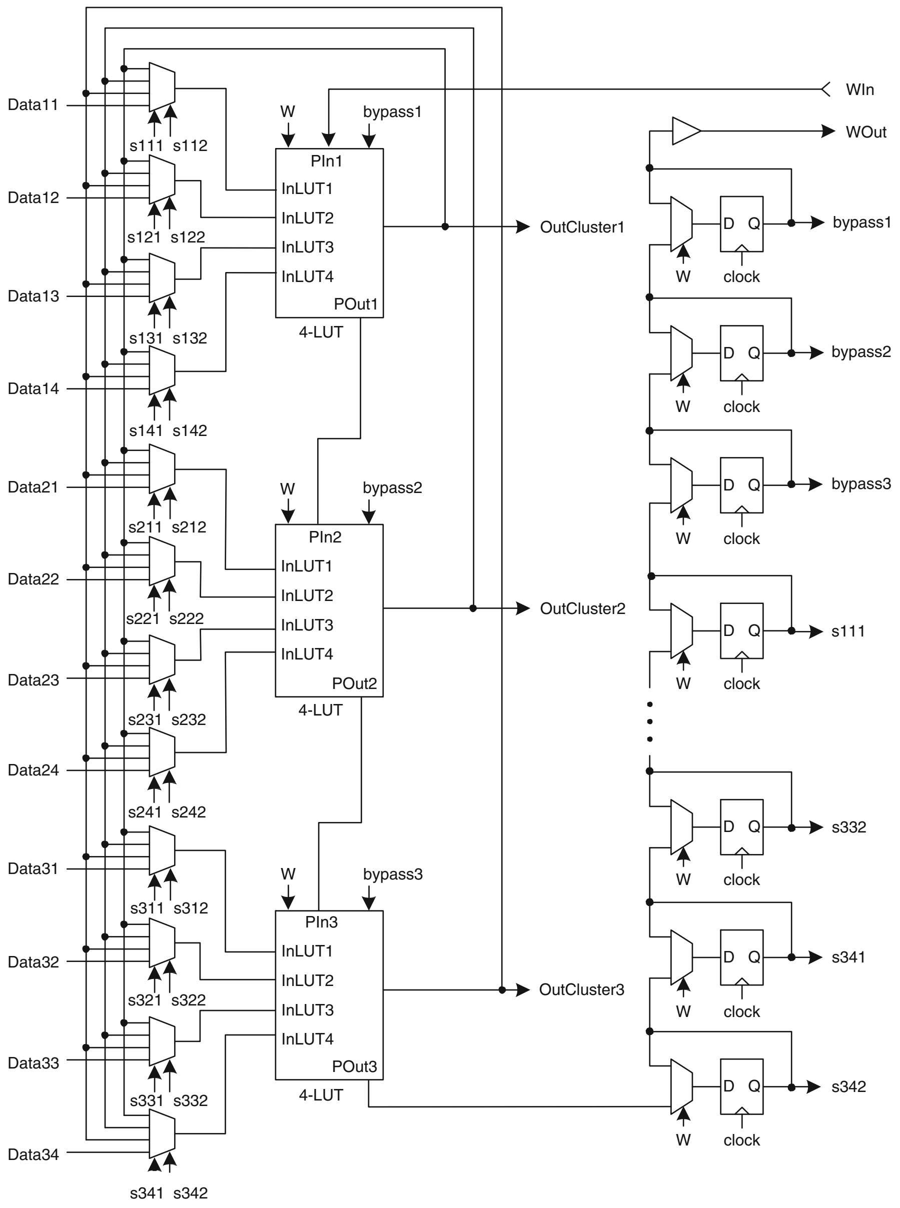
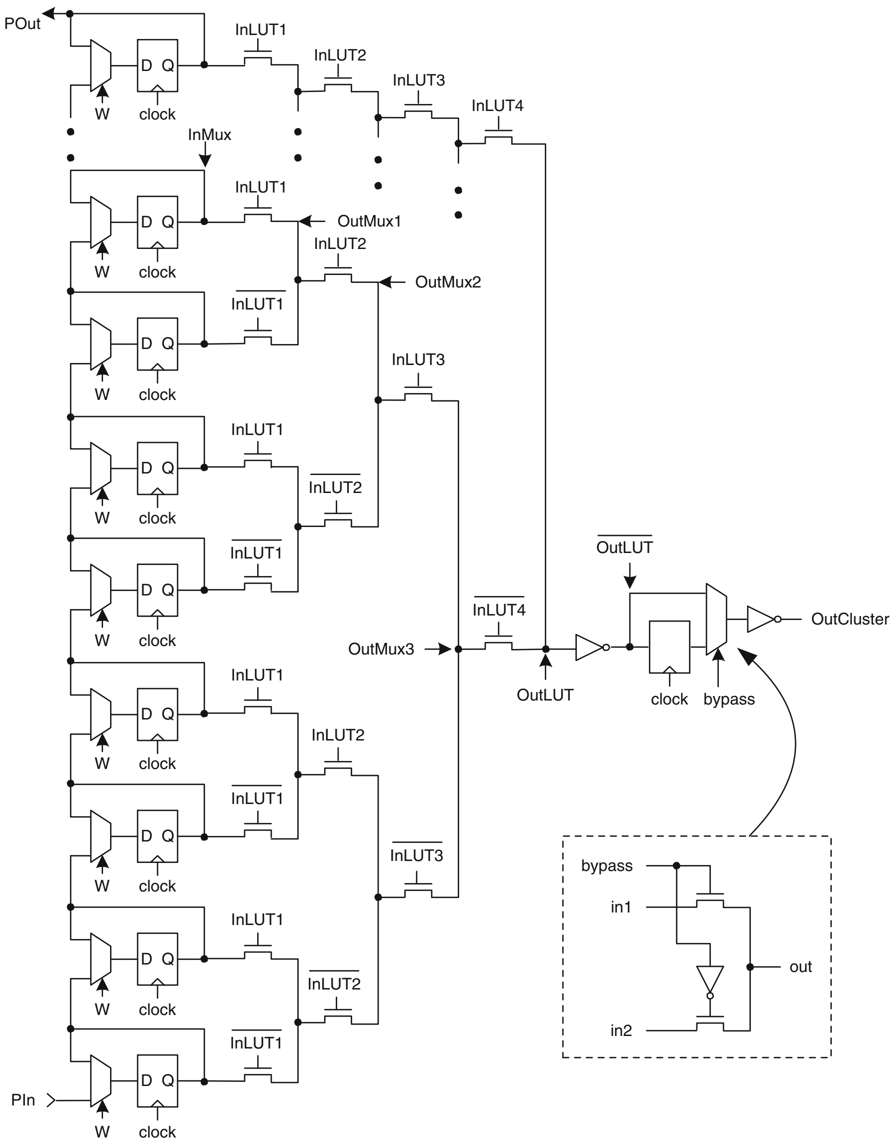


Fig. 7.2 The cluster architecture containing three 4-LUT

the PIn input propagates through the scan chain at the positive edge of clock while the global write signal, W, is kept at logic 1. When scan is finished to program the logic function for each 4-LUT, W is lowered and kept at logic 0 during normal FPGA operation.



POut
InLUT1
InLUT2
InLUT3
InLUT4
W
clock
InMux
D Q
OutMux1
OutMux2
OutMux3
$\overline{\text{InLUT1}}$
$\overline{\text{InLUT2}}$
$\overline{\text{InLUT3}}$
$\overline{\text{InLUT4}}$
$\overline{\text{OutLUT}}$
OutLUT
OutCluster
bypass
in1
in2
out
PIn

Fig. 7.3 The 4-LUT circuit containing 1 × 16 Look-Up-Table and 16-1 pass-gate MUX

Each selector input to the 16-1 MUX, InLUT1 through InLUT4, is formed either by the cluster inputs routed to this specific 4-LUT (e.g., Data11 through Data14 if referred to the first 4-LUT) or from the outputs of each 4-LUT in the cluster, OutCluster1 through OutCluster3. The output of 16-1 MUX is either registered or routed directly to the output of the 4-LUT via a bypass path.

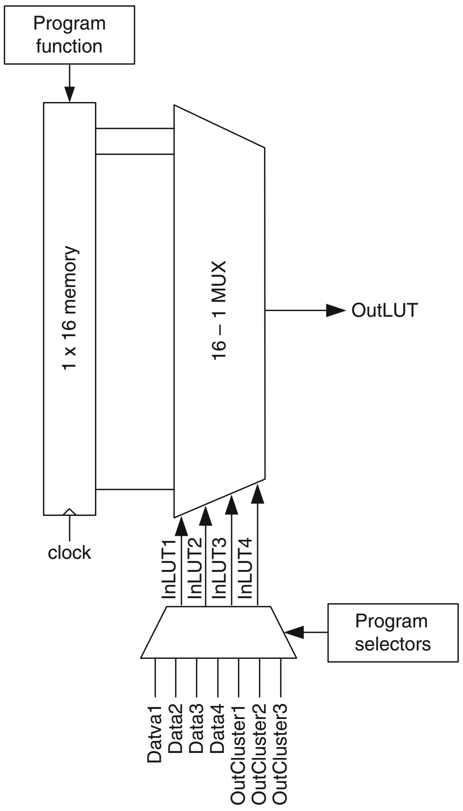


Fig. 7.4 Functional representation of the 4-LUT

The registered 4-LUT outputs are either routed to the inputs of neighboring clusters to form pipelined structures or fed back to the 4-LUT inputs within the same cluster to implement state machines. This flexibility of logic configuration creates an environment to implement any hybrid logic block composed of combinatorial and sequential logic functions.

The functional representation of each 4-LUT is shown in Fig. 7.4. In this figure, 1×16 memory block represents 16-bit input scan chain in Fig. 7.3 responsible for storing a particular logic function. An output is generated for each 4-LUT using a 16-1 pass-gate MUX.

7.3.3 An Example: A 3-bit Carry-Ripple Adder (CRA)

This simple 3-bit Carry-Ripple Adder (CRA) example demonstrates the capability of the proposed FPGA architecture to implement a combinatorial block. A total of six 4-LUTs in two separate clusters are used to implement the 3-bit CRA. The truth

Fig. 7.5 Truth table of a full adder using 4-LUT

A	B	Cin	Data4	Sum	Cout
0	0	0	$\boxed{0}$ X	0	0
0	0	0	$\boxed{1}$	0	0
0	0	1	0	1	0
0	0	1	1	1	0
0	1	0	0	1	0
0	1	0	1	1	0
0	1	1	0	0	1
0	1	1	1	0	1
1	0	0	0	1	0
1	0	0	1	1	0
1	0	1	0	0	1
1	0	1	1	0	1
1	1	0	0	0	1
1	1	0	1	0	1
1	1	1	0	1	1
1	1	1	1	1	1

table to generate sum (Sum) and carry-out (Cout) bits of each full adder (FA) in the CRA is shown in Fig. 7.5.

Since each full adder in the CRA has the A, B and carry-in (Cin) inputs, the fourth input to the full adder receives a "don't care" entry, which makes each Sum and Cout output repeat twice in the truth table. The implementation of the 3-bit CRA is shown in Fig. 7.6. The Sum0 output corresponding to the least significant bit of the full adder is generated by storing the Sum column of the truth table in Fig. 7.5 into the 1 × 16 memory array of the 4-LUT in Fig. 7.6, and programming the cluster scan chain to produce the selector inputs, s111 = s112 = s121 = s122 = s131 = s132 = s141 = s142 = 0. This allows the inputs, Data11 through Data14, to be routed directly to the selector inputs, InLUT1 through InLUT4, of the first 4-LUT, respectively. Therefore, selector inputs for 4-LUT become InLUT1 = Data11 = A0, InLUT2 = Data12 = B0, and InLUT3 = Data13 = Cin0. InLUT4 = Data14 takes a "don't care" input, which converts the 16-1MUX in Fig. 7.3 into two identical 8-1 MUXes. Only one 8-1 MUX output is connected to the OutLUT node in Fig. 7.3 because the InLUT4 input has a "don't care" value. The other 8-1 MUX output stays unconnected. The same approach applies to generate the Cout0 output corresponding to the least significant full adder. The Cout column in Fig. 7.5 is stored in the 1 × 16 memory array of the second 4-LUT in the cluster. The selector inputs, s211 through s242, are programmed to be equal to 0 such that InLUT1 = Data21 = A0, InLUT2 = Data22 = B0, and InLUT3 = Data23 = Cin0. InLUT4 = Data24 again takes a "don't care" input. The Cout0 output is directly routed to the third 4-LUT in the same cluster and also to the first 4-LUT of the second cluster to generate the Sum1 and Cout1 outputs of the next significant full adder bit of the 3-bit CRA.

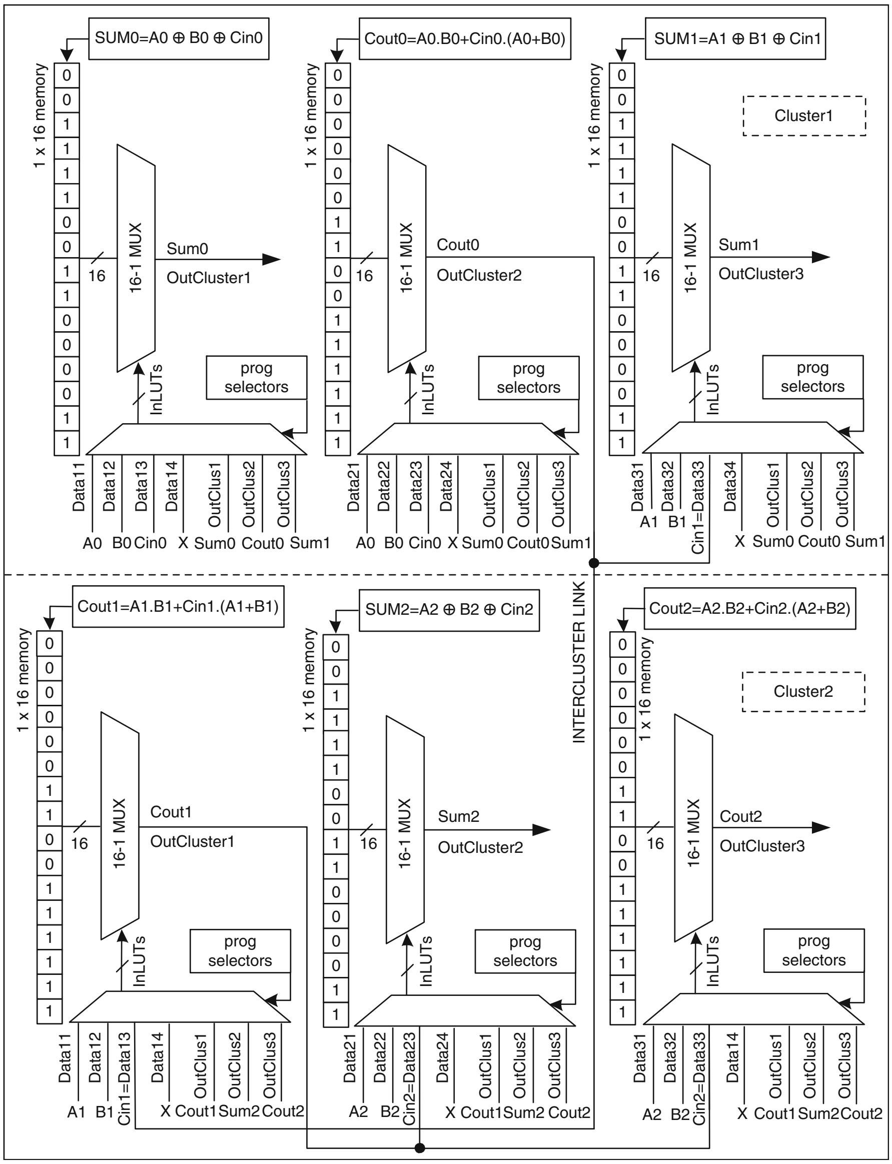


Fig. 7.6 3-bit Carry-Ripple Adder circuit using two separate clusters

Similarly, the Cout1 output is routed to the inputs of the second and third 4-LUTs of the second cluster to generate the Sum2 and Cout2 outputs, respectively. The Bypass1, Bypass2, and Bypass3 inputs are programmed to be at logic 1 to make all three bits of the CRA purely combinatorial and unregistered.

7.4 FPGA Circuit Characteristics

7.4.1 4-LUT Worst-Case Propagation Delays

Figure 7.7 shows the worst-case propagation delays at the internal nodes of the 4-LUT. In this figure, all uncomplemented InLUT inputs, InLUT1 through InLUT4, are assumed to be connected to logic 1. When the signal reaches the internal node, OutMUX1, the logic 1 level at that node drops by an amount equal to NMOS threshold voltage, V_{TN}, as expected. This trend continues at the subsequent nodes from OutMUX2 to OutLUT until normal logic levels are restored at the OutLUT node by an inverter. The OutCluster node shows a slower rise time due to a 100 aF load capacitor. Typical worst-case delays in Fig. 7.7 are 4.7 ps from the clock input to the InMUX node, 8.9 ps from the clock input to the OutLUT node, and 20.2 ps from the clock input to the OutCluster node.

Figure 7.8 shows the worst-case delay from the clock input to the OutCluster node as a function of load capacitor and it is expressed as $T_{CLK\text{-}OUTCLUSTER} = 10.8 + 0.09$ CLOAD in ps.

7.4.2 Intercluster Propagation Delays

The intercluster delays are explained in Fig. 7.9. In this figure, a worst-case signal path is shown when two clusters are placed in a diagonal fashion. A typical intercluster signal path starts from the OutCluster port of the source cluster and travels through the pass-gate transistor in the 1-8 DEMUX stage, an intercluster wire along the *y*-axis of the source cluster, a pass-gate transistor at the first switch box, the second intercluster wire along the *y*-axis of the destination cluster, two pass-gate transistors in the second switch box, the third intercluster wire half the

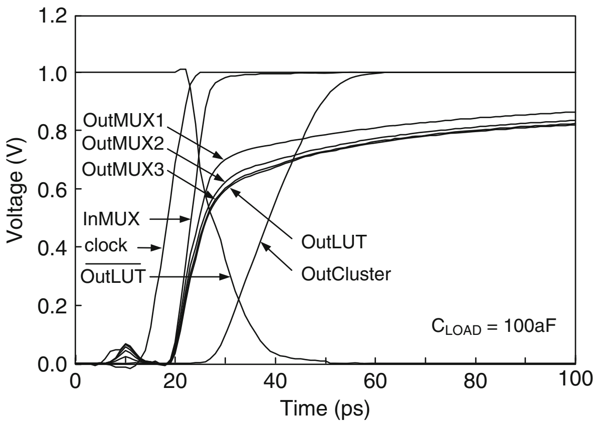


Fig. 7.7 Post-layout, worst-case propagation delays through 4-LUT

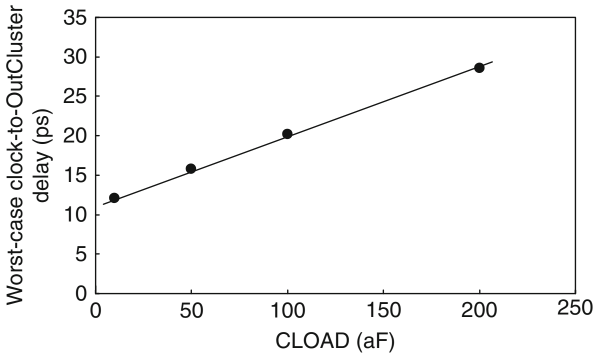


Fig. 7.8 Post-layout, worst-case propagation delay from clock input to 4-LUT output as a function of output load

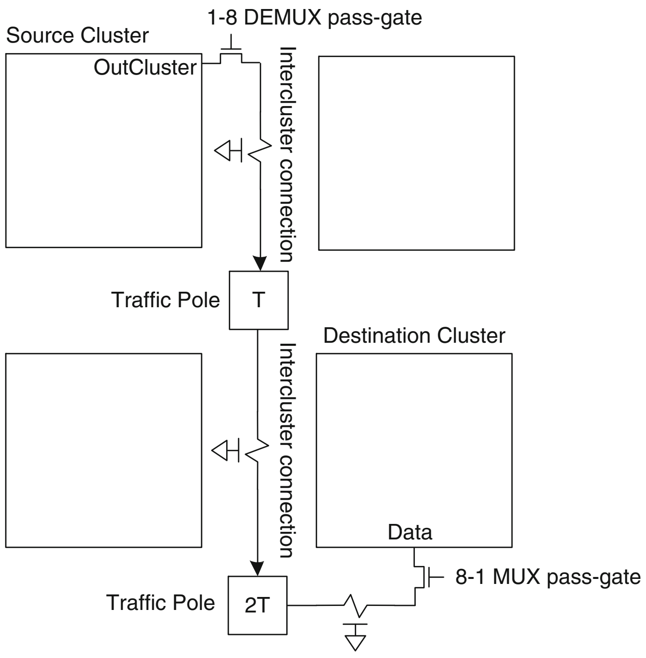


Fig. 7.9 Intercluster wiring topology

length along the x-axis of destination cluster, and a pass-gate transistor in the 8-1 MUX stage before arriving at the data input of the destination cluster.

Typical worst-case delay waveforms at the OutCluster port of the source cluster, the Data port, and the InLUT1 nodes of the destination cluster are shown in Fig. 7.10. In this figure, fan-out is equal to one because only one destination cluster is connected to the source cluster. The worst-case intercluster delay as a function of fan-out is shown in Fig. 7.11 where fan-out is defined as the number of destination clusters. In this figure, the delay analysis is performed on a diagonally placed source and destination clusters two and four cluster lengths apart.

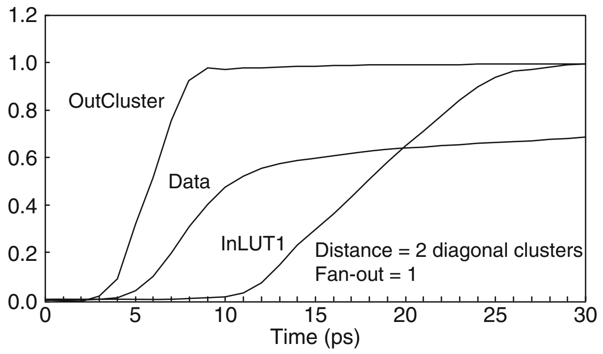


Fig. 7.10 Post-layout, worst-case intercluster wiring delay between two neighboring clusters diagonally placed two clusters apart

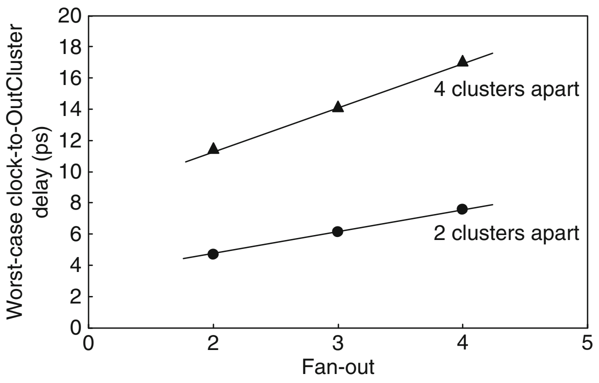


Fig. 7.11 Post-layout, worst-case intercluster wiring delay between two diagonally placed neighboring clusters as a function of fan-out (the number of destination clusters)

7.4.3 4-LUT Power Dissipation

Since 16-1 input MUX of the 4-LUT is configured using only pass-gate transistors, it is quite possible that the InLUT (InLUT1 through InLUT4) and the $\overline{\text{InLUT}}$ ($\overline{\text{InLUT1}}$ through $\overline{\text{InLUT4}}$) inputs may not arrive at the same time and the overlap between these signals may develop varying dynamic power dissipations for 4-LUT. Figure 7.12 shows the worst-case dynamic power dissipation as a function of signal overlap for 2 GHz, 4 GHz, and 10 GHz frequencies. To obtain the 2 GHz data, a 200 ps pulse with a 500 ps period is applied to all InLUT and $\overline{\text{InLUT}}$ inputs. However, the pulse applied to InLUT inputs is forced to overlap with the pulse applied to $\overline{\text{InLUT}}$ inputs by 20 ps, 40 ps, 60 ps, 80 ps and 100 ps intervals. Subsequently, total dynamic power dissipation is measured and averaged within 500 ps window for each case. The same procedure is applied to obtain the 4 GHz and 10 GHz data. However, the pulse width and period are both decreased to 100 ps and 250 ps for 4 GHz, and to 50 ps and 100 ps for 10 GHz, respectively. Both the level and the slope of power dissipation are observed to increase with increasing frequency. However, when there is no signal overlap, the power dissipation in 4-LUT (1 μW at 2 GHz and 2.8 μW at 10 GHz) is found to be independent of any linear frequency-power relationship.

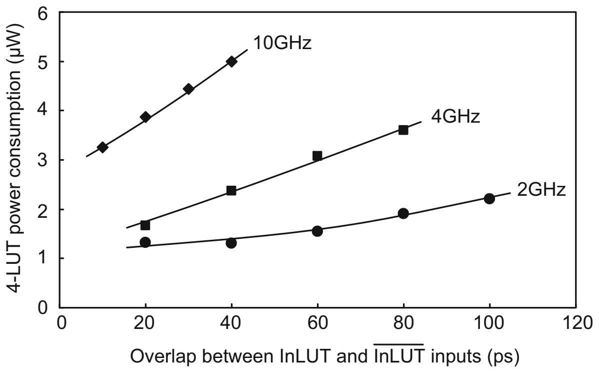


Fig. 7.12 Post-layout, worst-case dynamic power dissipation of 4-LUT for different frequencies of operation

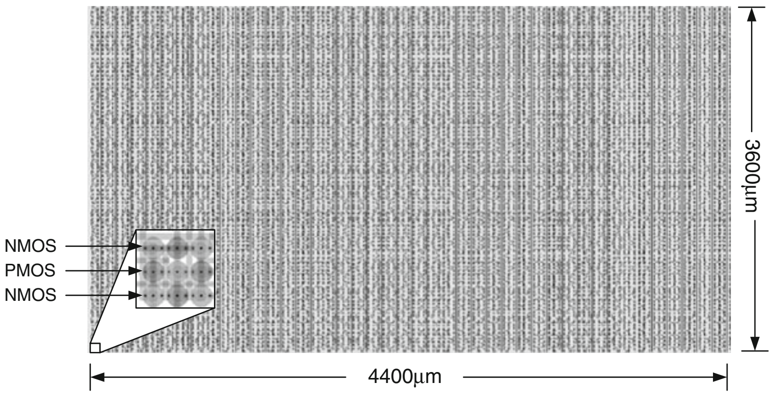


Fig. 7.13 The cluster layout

7.4.4 Flip-Flop Characteristics

Each cluster contains 93 D-type flip-flops, which are used for both scanning control inputs and logic configuration. A typical flip-flop used in this design has 5 ps set-up time, 1 ps hold time, and 11.4 ps clock-to-output delay at a load capacitor of 10 aF. Its worst-case power dissipation is 1.2 μW at 10 GHz.

7.4.5 Cluster Layout

Figure 7.13 shows a cluster layout that contains three 4-LUTs. Note that this layout contains alternating NMOS-PMOS placement methodology in a fabric-like matrix as shown in the subpicture of Fig. 7.13. Every NMOS transistor is surrounded by 4 PMOS transistors and vice versa in a cross-bar form. This way, fabrication-related issues such as uniform vertical crystal growth of transistor bodies, Reactive Ion Etch (RIE) lag etc. may be minimized. Each 4-LUT input is placed at a different cluster boundary: the first 4-LUT inputs are placed at the top, the second at the left, and the third at the bottom. The cluster outputs and control signals are placed at the right boundary of the cluster.

7.5 Summary

This study investigates the outcome of using silicon nanowire transistor circuits in FPGAs containing scan chains. Each FPGA cluster in this architecture is composed of three 4-LUTs. Each cluster receives input data from other clusters through its twelve data inputs and sends data to neighboring clusters from its three outputs. Each 4-LUT in the cluster can be programmed to implement a specific logic function with a maximum of four inputs or can be configured as part of a state machine. Scan chains are used to program the data path in the cluster, to determine the logic function for each 4-LUT and most importantly to minimize wiring channels in the entire FPGA array. Intercluster communication is established by an 8-bit wide bus architecture interconnected with switch boxes that contain six transistors and are placed at the corners of each cluster. Post-layout, worst-case propagation delay for a 4-LUT is 20.8 ps from the rising edge of the clock signal to its output. The worst-case intercluster wire delay is 4.8 ps between two diagonally placed adjacent clusters and increases to 11.2 ps between two neighboring clusters diagonally placed at four cluster lengths away. The average worst-case dynamic power dissipation of a 4-LUT is 1 μW at 2 GHz and 2.8 μW at 10 GHz if there is no overlap between uncomplemented and complemented selector pulses for the 4-LUT. If there is an overlap between the selector pulses, power dissipation increases by 12.5 nW/ns of the overlap at 2 GHz and 60 nW/ns of the overlap at 10 GHz. Cluster layout contains vertical silicon nanowire transistors placed in a fabric matrix where each NMOS (PMOS) transistor has four neighboring PMOS (NMOS) transistors.

References

1. Choudhary P, Marculescu D (2009) Power management of voltage/frequency island-based systems using hardware-based methods. IEEE Trans VLSI Syst 17(3):427–438
2. Bindal A, Hamedi-Hagh S, Ogura T (2008) Silicon nanowire technology for applications in the field programmable gate array architectures. J Nano Opto 3:113–122
3. Ahmed E, Rose J (2004) The effect of LUT and cluster size on deep-submicron FPGA performance and density. IEEE Trans VLSI Syst 12(3):288–298