Technology Development and Modeling of Switching Lattices Using Square and H Shaped Four-Terminal Switches

Nihat Akkan, Şerzat Safaltın, Levent Aksoy, İsmail Çevik, Herman Sedef, Csaba Andras Moritz, and Mustafa Altun

Abstract—Switching lattices formed by four-terminal switches are introduced as dense rectangular structures to implement Boolean logic functions. It is clearly shown in literature by a variety of logic synthesis algorithms including the exact one, realizing logic functions on lattices with the fewest number of four-terminal switches, as well as the heuristic ones, that switching lattices offer a significant area advantage in terms of the number of switches over the conventional CMOS implementations. Although the computing potential of switching lattices has been well justified, the same thing cannot be said for their physical implementation. There have been conceptual ideas for the technology development of switching lattices, but no concrete and directly applicable technology has been proposed yet. In this study, we show that switching lattices can be implemented using the CMOS technology. For this purpose, we propose two different four-terminal switch structures with square and H shaped gates. We construct these two structures in three dimensional technology computer-aided design (TCAD) environment satisfying the design rules of the TSMC 65nm CMOS process and perform simulations. Then, we develop Level 3 DC and AC models of these four-terminal switches in LTspice environment using the TCAD simulation data. As an experiment, we realize logic functions with the developed models using static and dynamic logic solutions. Experimental results show that the realization of logic functions using switching lattices occupy much less layout area and have competitive delay and power consumption values when compared to the conventional CMOS implementations.

Index Terms—emerging technologies, four-terminal switch, switching lattice, technology simulation, device modeling

1 Introduction

In recent years, in order to cope with the limitations of the conventional CMOS technology, novel architectures and structures in nanoelectronics have been introduced [2], [3], [4], [5]. A switching lattice, formed as a two dimensional network of four-terminal switches, is presented as a regular, dense, and area-efficient structure for the realization of logic functions [6]. As shown in Fig. 1(a), a four-terminal switch, developed especially for the cross-points of nanoarrays, has one control input x and four terminals. If its control input has the value 0, all of its terminals are disconnected (OFF). Otherwise, they are connected (ON). In a switching lattice, each four-terminal switch is connected to its vertical and horizontal neighbors. Fig. 1(b) presents a 3×3 switching lattice where $x_1, x_2, \ldots, x_9$ denote the control inputs of switches. The lattice function is formed as the sum of products of control inputs in each path between the top and bottom plates. Note that a path in a lattice with four-terminal switches is a sequence of switches connected horizontally and vertically. Thus, the lattice function evaluates to 1 if there is a path between the top and bottom plates. Fig. 1(c) shows the 3×3 lattice function $f_{3\times3}$. A lattice function is unique and does not include any redundant paths. For example, the path $x_1x_2x_5x_8$ in the 3×3 lattice is dominated by $x_2x_5x_8$.

A switching lattice can be used to realize a logic function by simply finding an appropriate assignment to the control inputs of switches from the logic function variables and also, constant logic values 0 and 1. The design complexity of a lattice is determined as the number of switches, *i.e.*, lattice size. Thus, the optimization problem is to find an $m \times n$ lattice for the realization of a given logic function such that there exists an appropriate assignment to the control inputs of the given lattice and m times n is minimum. Over the years, many efficient methods have been introduced for this problem [6], [7], [8], [9], [10], [11], [12], [13], [14].

To compare different realizations of a logic function, consider $f = \overline{a}b\overline{d} + a\overline{b}\overline{c}d + a\overline{b}c\overline{d}$ as an example. In a straight-forward implementation, f can be implemented in two-level using AND, OR, and NOT gates as shown in Fig. 2(a). Note that this implementation requires a total of 38 MOS transistors, *i.e.*, two-terminal switches[1]. However, the realization of a logic function can be optimized by using

- *N. Akkan and H. Sedef are with the Electronics and Communication Engineering Department, Yıldız Technical University, Istanbul, Turkey (e-mail: {nakkan, sedef}@yildiz.edu.tr).*
- *Ş. Safaltın is with the Metallurgical and Materials Engineering Department, Istanbul Technical University, Istanbul, Turkey (e-mail: safaltin@itu.edu.tr).*
- *L. Aksoy, İ. Çevik, and M. Altun are with the Electronics and Communication Engineering Department, Istanbul Technical University, Istanbul, Turkey (e-mail: {aksoyl, ceviki, altunmus}@itu.edu.tr).*
- *C. A. Moritz is with the Electrical and Computer Engineering Department, University of Massachusetts, Amherst, Massachusetts, USA (e-mail: andras@ecs.umass.edu).*

** A preliminary version of this article appeared in [1].*

1. In switching theory, a MOS transistor of FETs can be modeled as a two-terminal switch such that the drain and source terminals of the transistor denote the two terminals of the switch and the gate terminal of the transistor corresponds to the control input of the switch.

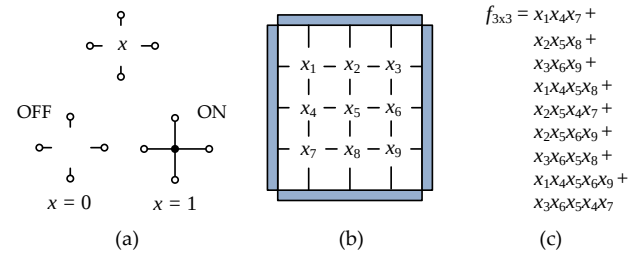

Fig. 1. (a) Four-terminal switch; (b) the 3×3 four-terminal switching network; (c) the 3×3 switching lattice function.

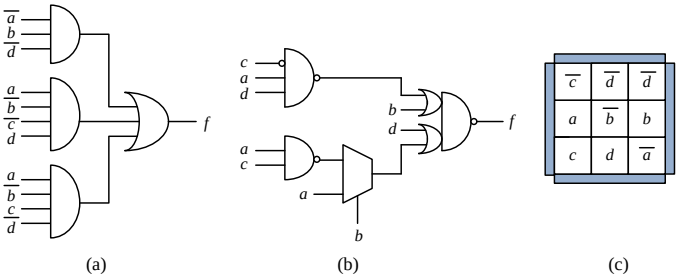

Fig. 2. Different realizations of $f = \bar{a}b\bar{d} + a\bar{b}\bar{c}d + a\bar{b}cd$: (a) two-level design; (b) optimized design; (c) switching lattice design.

a synthesis tool. Fig. 2(b) presents an optimized solution obtained using the Cadence GENUS synthesis tool with the TSMC 65nm gate library. This optimized realization needs 32 MOS transistors. On the other hand, Fig. 2(c) shows the solution of JANUS [13] developed for the optimization of the number of four-terminal switches in a lattice[2]. It requires 9 four-terminal switches and 8 MOS transistors for the inverters needed for the inputs.

This simple example, supported by many other inspiring examples and results in the literature [6], [7], [8], [9], [10], [11], [12], [13], [14], clearly shows the computing potential of switching lattices as well as their area efficiency. However, no concrete and directly applicable technology has been proposed for the implementation of switching lattices. In [6], physical formulations of nanowire and magnetic four-terminal switches are conceptually given, but they lack details on simulation and fabrication.

In this study, we introduce the implementation of two different four-terminal switches, namely, square shaped and H shaped, in a standard CMOS process. They are shown in Fig 3. We successfully generate layouts of both structures without any violations in the design rule checking (DRC) of the 65nm CMOS process. Thus, they are ready for CMOS fabrication. However, the devices cannot be simulated with the transistor device models provided by the semiconductor foundries. To overcome this challenge and to make comparison with conventional CMOS implementations, we construct these structures in three dimensional technology computer-aided design (TCAD) environment and perform simulations. Then, using the TCAD simulation data, we develop Level 3 DC and AC models of these four-terminal switches in LTspice environment. In our model of a four-terminal switch, we use 6 conventional CMOS transistors where a transistor is placed between each pair of four terminals. Finally, we realize logic functions with the developed models using the static pseudo NMOS and footed dynamic logic solutions. Experimental results show that the realizations of logic functions using switching lattices occupy much less layout area when compared to the conventional CMOS implementations. This is due to the fact that switching lattices have dense, regular, and metal-connection-free layouts. On the other hand, switching lattice realizations have delay and power consumption values close to those of the conventional CMOS implementations.

2. Keeping the same order in the products and variables of $f_{3\times3}$ in Fig. 1(c), the function realized by the lattice can be given as $g = \bar{c}ac + \bar{d}b\bar{d} + \bar{d}b\bar{a} + \bar{c}a\bar{b}d + \bar{d}\bar{b}ac + \bar{d}\bar{b}b\bar{a} + \bar{d}\bar{b}b\bar{d} + \bar{c}a\bar{b}b\bar{a} + \bar{d}\bar{b}\bar{b}ac$. After the application of Boolean algebra laws, it can be written as $g = \bar{a}b\bar{d} + a\bar{b}\bar{c}d + a\bar{b}cd$.

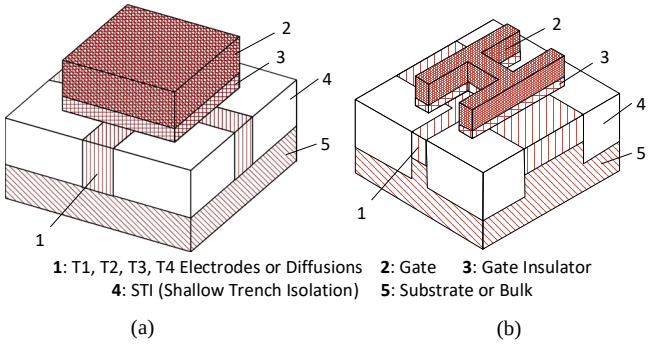

Fig. 3. Three dimensional structures of the proposed devices behaving as a four-terminal switch: (a) square shaped; (b) H shaped.

This article is organized as follows. Section 2 describes the technology development of four-terminal switches and switching lattices. Section 3 introduces the modeling of square and H shaped four-terminal switches using Level 3 model parameters. Realization of logic functions on switching lattices using static pseudo NMOS logic and footed dynamic logic is described in Section 4. Section 5 presents the experimental results and finally, Section 6 concludes the article.

2 Technology Development of Switching Lattices

In this section, we introduce the developed technology for the proposed square and H shaped four-terminal switches with TCAD simulation results. While developing the technology, we consider three main criteria: i) current-voltage relationship of terminals should satisfy four-terminal switch behaviour; ii) a single gate should control all current paths between terminal pairs; and iii) area and current conducting capability of devices should be optimized. Since there are 6 possible terminal pairs, computed as $C(4, 2)$, where C stands for the combinations, we have 6 different current paths or channels. Note that in a conventional CMOS transistor, there is a single current path between its source and drain terminals.

We consider two different types for the realization of a four-terminal device, *i.e.*, square shaped gate and H shaped gate, as shown in Fig 3. These structures have been developed under the requirement for satisfaction of the design rules in the 65nm CMOS technology. While the square shaped device is the minimum area solution, the H shaped device offers more current conducting capability at the cost of a slightly larger area when compared to the square shaped device. Note that the square and H shaped devices are enhancement n-type based on NMOS transistors. Their p-type versions based on PMOS transistors can also be generated easily.

2.1 Design in a CMOS Process

Previously, we introduced two CMOS-compatible four-terminal switch devices, namely the square-shaped and the cross-shaped, in [1]. These structures demonstrated that a four-terminal switch could be built in a CMOS process. However, CMOS design rules bring some challenges while building four terminal switches in a CMOS process. The biggest challenge is due to the gate extension rule. Design rules mandate the gate electrode to extend beyond the source and drain regions. The mandatory extension on each side of the channel is larger than the minimum channel width. This means that the width of the gate electrode must be larger than 3 times the minimum channel width for a minimum sized transistor. For instance, in the 65nm process used in this study, the minimum transistor channel width is 120nm while the minimum extension 140nm in both sides of the channel results in 400nm minimum gate electrode width.

A four-terminal switch has four source/drain regions positioned at 90° angles around the gate. This means that the transistor channel is along the x direction at the boundary of two drain/source regions and along y direction at the boundary of the other two. Each boundary is considered as a channel width by the design rules. Therefore, gate electrode must have the extensions around all four terminals. Therefore, gate electrode is significantly large.

If a square shaped switch is designed in a CMOS process as shown in Figs. 4(a-b), each side of the gate electrode is equal to channel width plus two gate extension (G_{ext}) which is at least 400nm in the 65nm process used for this study. When current flows through any two terminals, channel width is equal to the drawn channel width which is the width of the source/drain regions. However, the effective channel length is equal to the edge of the gate electrode. The effective width to length ratio of the switch is $W/(W + 2G_{ext})$. This ratio will always be less than unity. In the 65nm process, minimum sized switch has an effective width to length ratio of 120nm/400nm. The resulting conductivity of the equivalent switch channel is less than one sixth of that in a minimum sized NMOS transistor having a width to length ratio of 120nm/60nm. If the channel width is doubled, gate electrode edge becomes 520nm and effective width to length ratio becomes 240nm/520nm.

This issue is even more problematic in the cross shaped switch introduced in our previous work [1]. Each arm of the cross shaped gate electrode must extend beyond the drain/source regions, meaning that each arm is at least 400nm wide. The intersection of the two arms is a $400nm \times 400nm$ square. The length of each arm is longer than 400nm as expected. An L shaped gate must extend at least 120nm beyond the corner in the given processes. This rule brings the minimum length of each arm of the cross shaped gate electrode to 640nm. Once again, the channel length is equal to the length of each arm of the cross. Therefore, the width to length ratio of the cross shaped switch is smaller than that of a square shaped switch. Since current conducting capability is proportional to width to length ratio, minimum sized square-shaped switch has higher current conducting capability. Moreover, it occupies smaller chip area and it is easier to manufacture when compared to the cross-shaped one. Hence, we decided not to further investigate the cross shaped switch in this study.

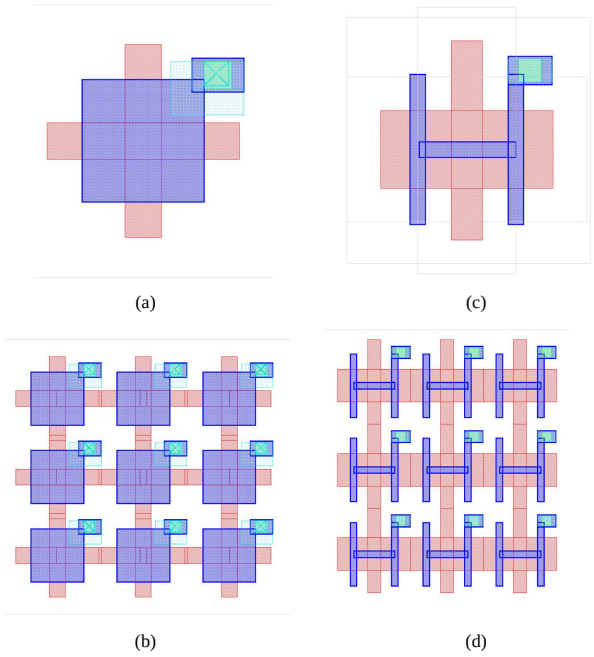

Fig. 4. Layouts in 65nm CMOS process: (a) square shaped four-terminal switch; (b) 3×3 switching lattice with square shaped switches; (c) H shaped four-terminal switch; (d) 3×3 switching lattice with H shaped switches.

A four-terminal switch with an H shaped gate presented in Figs. 4(c-d) is introduced in this study as an alternative to the cross shaped gate. The H shaped gate separates four diffusion regions acting as drain/source regions. Four vertical and one horizontal channels are formed between these four regions. The length of a channel is the thickness of edges of the H shaped gate. Channel length can be picked as the minimum channel length allowed by the process, *i.e.*, 60nm. Yet, the channel width is dictated by the design rules. Two parallel edges of the H shaped gate have to be 320nm which makes the width of the horizontal channel 320nm. Diffusion regions have to extend at least 120nm beyond the corner of an L shaped gate. Therefore, the vertical edges of the H shaped gate have to extend 120nm on each side of the horizontal edge. Total length of a vertical edge is 300nm, meaning that the four vertical channels have an effective channel width equal to 150nm.

A switching lattice can be easily built using these switches by connecting them with shared diffusion regions without a need of metal contacts. The only metal connection in the switch structure is the gate contact shown in the upper right corner of gate. Horizontal and vertical pitch of both four-terminal switches are dictated by distance requirements between polysilicon layers.

2.2 Design and Simulation in TCAD

Using the design rules and properties of the 65nm CMOS process explained above, we construct minimum sized square and H shaped four-terminal switches in TCAD. The features of four-terminal devices are summarized in Table 1. Here, all these values and information agree with the 65nm CMOS process including gate oxide material and thickness, as well as doping concentrations.

In TCAD simulations, we use three different simulation setups: i) obtain the drain source current (I_{ds})-gate source

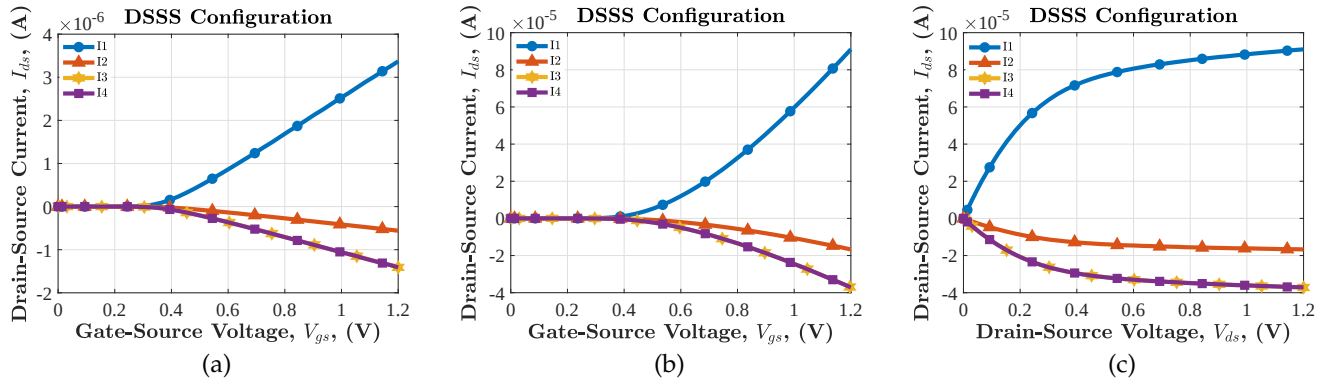

Fig. 5. TCAD results on square shaped device using DSSS: (a) I_{ds}-V_{gs} with V_{ds}=10mV; (b) I_{ds}-V_{gs} with V_{ds}=1.2V; (c) I_{ds}-V_{ds} with V_{gs}=1.2V.

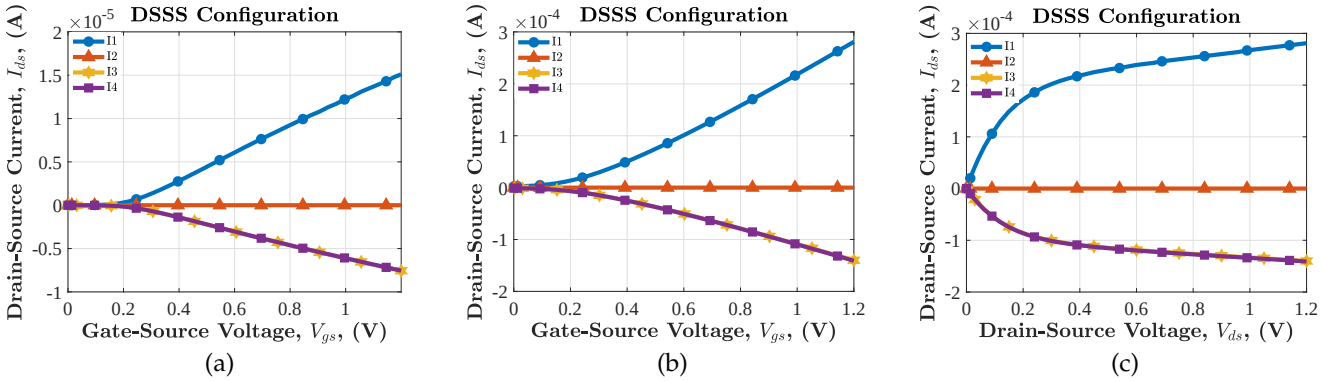

Fig. 6. TCAD results on H shaped device using DSSS: (a) I_{ds}-V_{gs} with V_{ds}=10mV; (b) I_{ds}-V_{gs} with V_{ds}=1.2V; (c) I_{ds}-V_{ds} with V_{gs}=1.2V.

TABLE 1
Features of four-terminal devices used in TCAD simulations.

Features	Device Type	
	Square Shaped	H Shaped
Width of electrodes (nm)	120	320 and 65
Distance/length between opposing electrodes (nm)	400	65 and 450
Gate insulator thickness (nm)	2.6	2.6
Doping profile (cm^{-3})	Substrate: Boron, 6.5×10^{17} Electrodes: Phosphorus, 10^{20}	Substrate: Boron, 6.5×10^{17} Electrodes: Phosphorus, 10^{20}
Gate insulator material	SiO_2	SiO_2
Electrode material	n-type Si	n-type Si
Substrate material	p-type Si	p-type Si

voltage (V_{gs}) values when the drain-source voltage (V_{ds}) is 10mV; ii) obtain I_{ds}-V_{gs} values when V_{ds} is 1.2V; iii) obtain I_{ds}-V_{ds} values when V_{gs} is 1.2V. For these setups, the source voltage is always set to 0V.

The configuration of a typical run is based on the applied voltage and current vectors over the four-terminal electrodes. If the current is towards device, it is referred as drain (D). Otherwise, it is called as source (S). We note that the four-terminal electrodes have fixed locations and are named as T1, T2, T3, and T4. We explore 16 common different cases for both four-terminal switches in the symmetric and non-symmetric operating conditions where the terminals are used as drain (D) or source (S) electrodes or left as floating (F) meaning that a terminal is connected to nothing. Additionally, the H shaped four-terminal switch has one more different case. These cases can be given as 1 drain-1 source (DSFF, SFDF, additionally FFDS for H shaped device), 1 drain-3 sources (DSSS, SDSS, SSDS, SSSD), 2 drain-2 sources (DDSS, SDDS, DSDS, DSSD, SDSD, SSDD) and 3 drains-1 source (DDDS, SDDD, DDSD, DSDD). As an example, the DSSS case means that T1 is drain, the other terminals are source.

Figs. 5 and 6 present the TCAD simulation results showing the transfer and output characteristic curves of each terminal for the DSSS case on the enhancement type square and H shaped devices, respectively. In TCAD simulation results, we observe expected transistor characteristic curves. Threshold voltages are determined around 0.3V and 0.2V, and the on and off current ratios are calculated as 222 and 128 for the square and H shaped devices, respectively. While the maximum current is 281μA for the H shaped device, it is 91μA for the square shaped device. On the other hand, the off current of the H shaped device is larger than that of the square shaped device.

3 Modeling of Switching Lattices

In this work, we focus on the modeling of the square and H shaped devices. In our previous work [1], we used the Level 1 MOSFET equations while modeling the devices proposed in [1] and extracted the model parameters to be used in SPICE simulations. However, the performance of the Level 1 MOSFET model was not promising. In this article, we use the SPICE Level 3 model. The Level 3 model has a semi-emprical nature and its parameters can be extracted directly. It is also faster and has less convergence problems than previous levels. Short channel and narrow channel effects

TABLE 2
The Level 3 Model Parameter Set.

Parameter	Units	Description
TOX	m	Thickness of gate oxide
UO	$cm^2/V.s$	Zero bias low field mobility
VTO	V	Threshold voltage
GAMMA	$V^{1/2}$	Body factor
PHI	V	Bulk fermi-potential
THETA	V^{-1}	Gate field induced mobility reduction parameter
TPG	—	Type of the gate material
RS	ohm	Source series resistance
RD	ohm	Drain series resistance
LD	m	Lateral diffusion
DELTA	—	Narrow channel effect on the threshold voltage
NSUB	cm^{-3}	Effective substrate doping
XJ	m	Short channel correction to the substrate sensitivity
VMAX	m/s	Maximum carrier velocity
ETA	—	Drain-induced barrier lowering (DIBL) coefficient
KAPPA	V^{-1}	Channel length modulation effect on the drain current
NFS	cm^{-2}	Subthreshold region fitting parameter
CGSO	F/m	Zero bias gate-source capacitance
CGDO	F/m	Zero bias gate-drain capacitance
CGBO	F/m	Zero bias gate-bulk capacitance
CJ	F/m^2	Zero bias junction capacitance
MJ	—	Bulk junction bottom grading coefficient

are included in the Level 3, introducing new parameters. A list of SPICE Level 3 model parameters are given in Table 2.

The drain current for Level 3 model can be expressed in a general form for both the linear and saturation regions including channel length of modulation as follows:

$$I_{ds} = \frac{\mu_{eff} C_{ox} W_{eff}}{L_{eff} - L'} \times \left\{ (V_{gs} - V_t) V_{ds}^{'} - \frac{{V_{ds}^{'}}^2}{2} \left[1 + \frac{f_s . \gamma}{4(2\phi_f - V_{bs})^{\frac{1}{2}}} + f_n \right] \right\}$$

where $V_{ds}^{'} = min(V_{ds}, V_{dsat})$ and μ_{eff} stands for effective mobility, C_{ox} is oxide capacitance, W_{eff} and L_{eff} are respectively effective channel width and length, $L^{'}$ is the distance between pinch-off point and drain terminal, V_{gs} is the gate to source voltage, V_{ds} is the drain to source voltage, V_{bs} is the bulk to source voltage, V_{dsat} is the drain saturation voltage, V_t is the threshold voltage, f_s indicates short channel effects, f_n denotes narrow channel effects, γ is the body effect factor, and $2\phi_f$ is the surface potential. The comprehensive details of Level 3 drain-current equation and all the effects of model parameters can be found in [15].

According to the TCAD simulations, we model the square and H shaped devices using NMOS transistors as shown in Fig. 7. Although model configurations are the same, two different NMOS transistors, called as Type A and Type B, are used for the square shaped device and three different NMOS transistors, called as Type A, Type B and Type C, are used for the H shaped device due to their geometries. We tried to keep transistor sizes minimum considering the 65nm process technology. The device geometries are given in Table 3.

The proposed devices are modeled using SPICE Level 3 model and its parameters are extracted by fitting the model curves to TCAD simulation data. For DSSS and DSDS cases, the output curves generated from TCAD and LTspice simulations are presented in Fig 8. In DSSS case, the first terminal (T1) is swept from 0 to 1.2V and other three terminals are grounded while V_{gs} is 1.2V. In DSDS case, the first and third terminals (T1 and T3) are swept from 0 to 1.2V and other two terminals are grounded while V_{gs} is 1.2V. In these configurations, TCAD simulations are

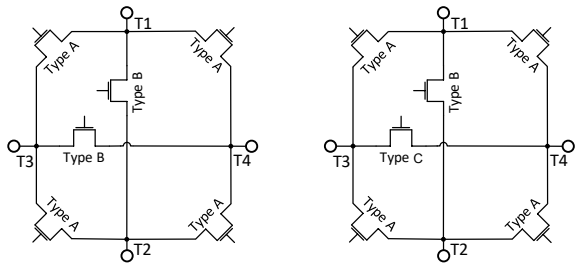

Fig. 7. Models of four-terminal switches using six NMOS transistors: (a) for square shaped device; (b) for H shaped device.

TABLE 3
Transistor Sizes for Square and H Shaped Devices.

Transistors	Square shaped		H shaped	
	W (nm)	L(nm)	W(nm)	L(nm)
Type A	120	285	120	65
Type B	120	400	65	450
Type C	—	—	320	65

performed and $I_{ds} - V_{ds}$ output curves are obtained for each terminal. Then, the Level 3 parameters are determined using our metaheuristics-based optimization algorithm [16] for each NMOS transistor in the square and H shaped four-terminal switch models. Finally, the found model parameters are included in LTspice and the output curves are generated. All the parameters determined for the square and H shaped four-terminal devices are given in Figs. 9 and 10, respectively. These parameters are extracted in three steps as follows:

1) *TOX* and *NSUB* process parameters are already determined in TCAD designs. When these parameters are known, *GAMMA* and *PHI* can be calculated as given in [17, eqs. (5.11) and (5.12)].
2) The calculated parameters in Step 1 are taken as constants. All the other DC model parameters are determined by our metaheuristics-based optimizer [16] using TCAD experimental data.
3) Level 3 already uses the Meyer's capacitance model [18]. We calculate parasitic capacitances, *CGSO* and *CGDO*, using [15, eq. (13.1)] and junction capacitance, *CJ*, using [15, eq. (3.37)]. Although there is no overlap in our TCAD designs, fringing capacitances are taken into account for *CGSO* and *CGDO*. Normally, *CGBO* is smaller than *CGSO* and *CGDO* and can be neglected. Circuit simulators may calculate a model parameter automatically if it is not given in the model file. Therefore, *CGBO* is suggested to be set to zero [15]. *MJ* ranges between 0.2 and 0.6 for real devices [17] and it is a fitting parameter that can be determined using the experimental capacitance data. *MJ* is 0.5 in our models.

4 Realization of Logic Functions with Switching Lattices

After modeling the proposed devices, we focus on testing their capability to realize logic functions by generating switching lattices using the proposed four-terminal

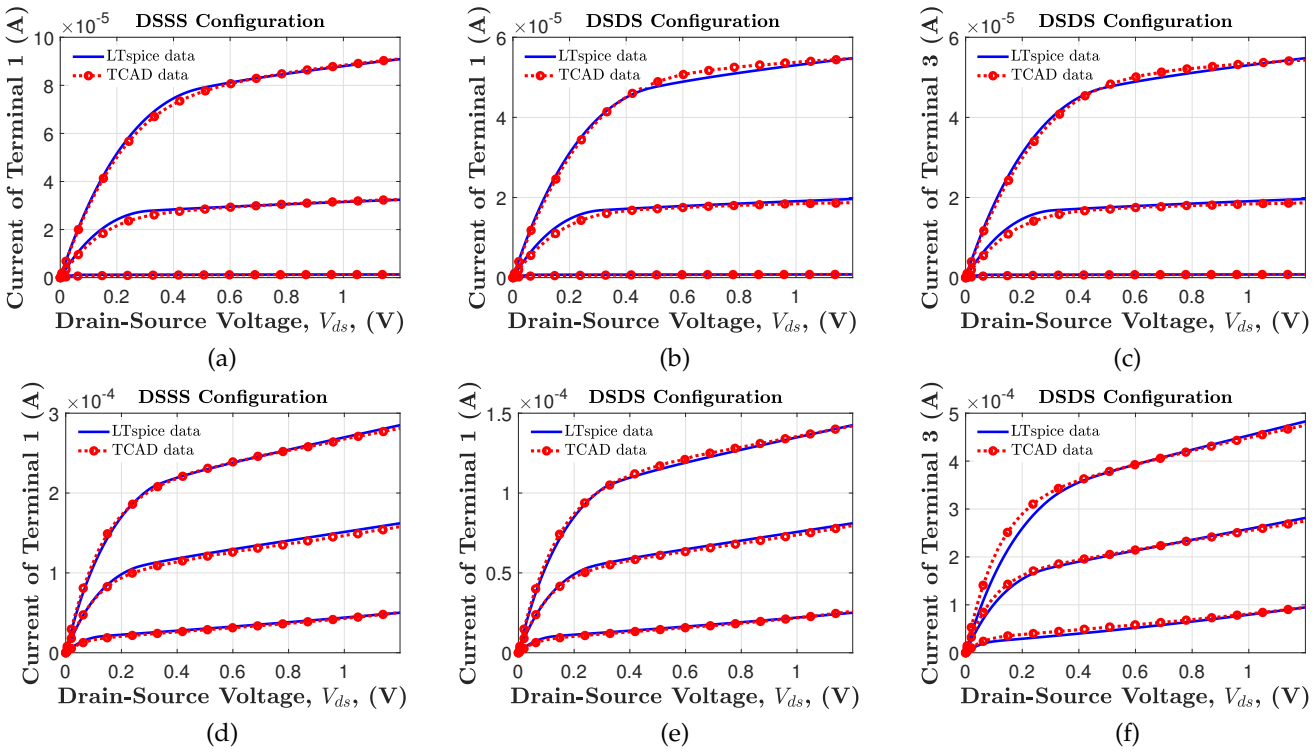

Fig. 8. TCAD and modeled LTspice data: (a) I1 current in square shaped for DSSS case; (b) I1 current in square shaped for DSDS case; (c) I3 current in square shaped for DSDS case; (d) I1 current in H shaped for DSSS case; (e) I1 current in H shaped for DSDS case; f) I3 current in H shaped for DSDS case.

```
.MODEL TYPE_A NMOS LEVEL  = 3
+ TOX    = 2.6E-09        NSUB   = 6.50E+17    GAMMA  = 0.3499
+ PHI    = 0.9162         VTO    = 0.29        DELTA  = 0.9744
+ UO     = 296            ETA    = 7.45E-04    THETA  = 0.0018
+ VMAX   = 1.04E+05       KAPPA  = 2.36        RS     = 0
+ RD     = 0              NFS    = 2.29E+13    TPG    = 1
+ XJ     = 8E-8           LD     = 0
+ CGDO   = 3.80E-10       CGSO   = 3.80E-10    CGBO=0
+ CJ     = 2.26E-03       MJ     = 0.5

.MODEL TYPE_B NMOS LEVEL  = 3
+ TOX    = 2.6E-09        NSUB   = 6.5E+17     GAMMA  = 0.3499
+ PHI    = 0.9162         VTO    = 0.29        DELTA  = 0.9797
+ UO     = 260            ETA    = 9.0E-04     THETA  = 0.2457
+ VMAX   = 5.83E+04       KAPPA  = 5.15        RS     = 0
+ RD     = 0              NFS    = 2.42E+13    TPG    = 1
+ XJ     = 8E-8           LD     = 0
+ CGDO   = 3.80E-10       CGSO   = 3.80E-10    CGBO=0
+ CJ     = 2.26E-03       MJ     = 0.5
```

Fig. 9. Level 3 parameters of the proposed model for the square shaped four-terminal switch.

```
.MODEL TYPE_A NMOS LEVEL  = 3
+ TOX    = 2.6E-9         NSUB   = 6.5E17      GAMMA  = 0.3499
+ PHI    = 0.9162         VTO    = 0.19        DELTA  = 2.0950
+ UO     = 349            ETA    = 4.39E-4     THETA  = 0.1829
+ VMAX   = 1.74E+5        KAPPA  = 0.1237      RS=0
+ RD     = 0              NFS    = 3.5E+12     TPG    = 1
+ XJ     = 8.0E-8         LD     = 0
+ CGDO   = 3.80E-10       CGSO   = 3.80E-10    CGBO=0
+ CJ     = 2.26E-3        MJ     = 0.5

.MODEL TYPE_B NMOS LEVEL  = 3
+ TOX    = 2.6E-9         NSUB   = 6.5E17      GAMMA  = 0.3499
+ PHI    = 0.9162         VTO    = 0.19        DELTA  = 5.8415
+ UO     = 199            ETA    = 1.6E-3      THETA  = 0.3
+ VMAX   = 9.96E+4        KAPPA  = 0.9747      RS = 0
+ RD     = 0              NFS    = 4.7E+12     TPG    = 1
+ XJ     = 8.0E-8         LD     = 0
+ CGDO   = 3.80E-10       CGSO   = 3.80E-10    CGBO=0
+ CJ     = 2.26E-3        MJ     = 0.5

.MODEL TYPE_C NMOS LEVEL  = 3
+ TOX    = 2.6E-9         NSUB   = 6.5E17      GAMMA  = 0.3499
+ PHI    = 0.9162         VTO    = 0.17        DELTA  = 8.1861
+ UO     = 364            ETA    = 8.46E-4     THETA  = 0.7460
+ VMAX   = 2.04E+5        KAPPA  = 0.07647     RS=0
+ RD     = 0              NFS    = 4.3E+12     TPG    = 1
+ XJ     = 8.0E-8         LD     = 0
+ CGDO   = 3.80E-10       CGSO   = 3.80E-10    CGBO=0
+ CJ     = 2.26E-3        MJ     = 0.5
```

Fig. 10. Level 3 parameters of the proposed model for the H shaped four-terminal switch.

switches. Since the body (bulk) terminal is always grounded, one of the six terminals is neglected, so the model has five terminals, *i.e.*, four D/S terminals and one gate (control) terminal. The behavior of devices is like n-type switches and the supply voltage is set to 1.2V considering the device characteristics and 65nm process.

We simulate logic functions using the square and H shaped four-terminal switches as a pull-down network of a pseudo NMOS logic and footed dynamic logic [19]. Fig. 11 shows their circuit structures. In this figure, the inputs to the switching lattices are actually the literals of the logic function. Although the pseudo NMOS logic implementation given in Fig. 11(a) is a simple and straightforward solution, we note that the difference between the best-case and the worst-case equivalent pull-down resistances can be very large in these designs. Since the pull-up network is a single PMOS transistor, which is always on, the PMOS transistor needs to be weaker than the worst-case pull-down configuration. Therefore, the low to high propagation delay ends up being significantly worse than the high to low propagation delay. If the pull-up transistor is too strong, the output low voltage level will be too high.

To tackle this propagation delay problem, a dynamic logic implementation is a viable option. In this case, the pull-up transistor is pulsed with a clock ϕ as shown in Fig. 11(b). The output is evaluated only when the PMOS transistor is off. Therefore, the output voltage during the precharge is not

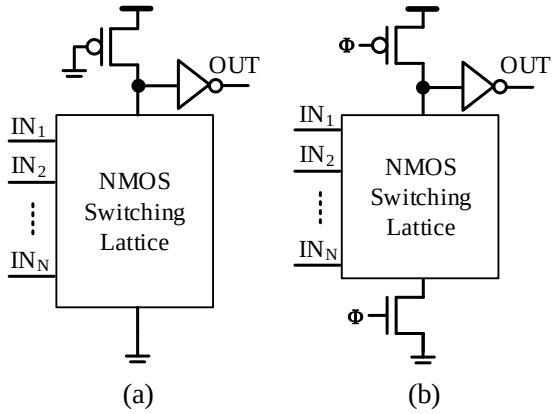

Fig. 11. Circuit structures for the realization of logic functions using switching lattices: (a) pseudo NMOS; (b) footed dynamic.

a concern. The PMOS transistor in a dynamic logic circuit can be significantly larger than that in the pseudo NMOS circuit. Therefore, the low to high propagation delay during the precharge can be small.

Note also that while an implementation using pseudo NMOS has a large static power consumption, the use of footed dynamic logic eliminates this problem.

5 Experimental Results

In this section, we present the design results of logic functions using CMOS two-terminal switches and also, the square and H shaped four-terminal switches. For the two-terminal realizations, we used the Cadence GENUS tool with the TSMC 65nm digital library, *i.e.*, tcbn65lpbwp12ttc with the track number 12 and standard threshold voltage value, to synthesize a logic function and the Cadence INNOVUS tool to place and route the design. Throughout the place and route process, the area, delay, and power consumption of the design were obtained. For the four-terminal realizations, initially, we used JANUS [13] to find the realization of a logic function using a switching lattice. Then, we used the Cadence VIRTUOSO tool with the TSMC 65nm library to design the switching lattice including the square and H shaped four-terminal switches. During the generation of layouts of switching lattices, we did the placement manually by simply placing the four-terminal switches to form the lattice. However, the routing can be rather complicated depending on the function to be implemented [20]. In our case, we could fit the whole metal routing on the lattice without wasting any peripheral area for the metal routing. We could also draw all internal connections of the lattice by the automated routing tool of Cadence using three metal layers. Throughout the place and route process, we obtained the area of the design. Moreover, based on the four-terminal device, *i.e.*, square or H shaped, and the design architecture, *i.e.*, pseudo NMOS or footed dynamic logic, we generated the SPICE netlist for the switching lattice and synthesized it using the sub-circuit of the square or H shaped four-terminal switch. Finally, we run LTspice simulations for 10,000 random inputs and as a result of these simulations, we obtained the worst-case delay and average power consumption. Although the delay and power consumption results of two-terminal realizations and those of lattice realizations using the proposed square and H shaped four-terminal switches are obtained under different design environments, they are comparable since all these designs are implemented under the 65nm CMOS technology.

TABLE 4
Details of the logic functions.

Instance	Function Details			JANUS [13]	
	inputs	products	degree	size	switches
5xp1_1	7	11	5	4x6	24
apex4_17	9	12	8	7x7	49
apex4_18	9	14	8	7x8	56
b12_02	8	7	5	4x4	16
ex5_07	8	10	4	3x8	24
ex5_10	6	7	3	3x6	18
ex5_12	8	9	3	3x5	15
ex5_21	8	10	3	3x7	21
misex1_02	7	5	5	5x4	20
mp2d_03	10	5	8	4x6	24
mp2d_04	10	6	9	7x3	21
sao2_01	10	20	10	12x7	84

As an experiment set, we used 12 logic functions taken from [21]. Table 4 presents the number of inputs, the number of products, and the maximum number of literals in the products of the logic function, called degree. This table also shows the details on the solutions of JANUS [13] in terms of lattice size and number of four-terminal switches. In the selection of these logic functions, the aim is to consider functions with a different number of inputs, products, and degree and with a different switching lattice realization. Also, logic functions realized using lattices with the same number of switches but different row and column values, *e.g.*, 3×7 and 7×3, are chosen to observe their behavior.

Table 5 presents the results of realizations using two-terminal and four-terminal switches. In this table, area, delay, and power denote the area in μm^2, worst-case delay in ns, and average power consumption in μW.

Observe from Table 5 that the realization of logic functions using switching lattices leads to nearly $2\times$ area improvement when compared to the conventional CMOS designs. As explained thoroughly in Section 3, switching lattices with the H shaped four-terminal switches proposed in this article offer better delay results with a slightly larger area when compared to those with square shaped four-terminal switches. Also, the delay results of switching lattices using footed dynamic logic are better than those using the pseudo NMOS logic. This is mainly because when a lattice has a large number of rows, the pull-down network has too many switches in series in the worst-case. Therefore, the worst-case pull-down resistance of the lattice is very large. Moreover, the long pull-down paths have more parasitic capacitance due to the increased number of nodes on the paths. Since the pull-up transistor must be weaker than the worst-case pull-down network, the low to high propagation delay is very dramatic for large lattices. Moreover, when the pull-up transistor has to charge inner node parasitic capacitances as well as the output node, the low to high propagation delay will be even worse. To overcome this problem, dynamic logic is used. Its low to high propagation delays are superior to those of the pseudo NMOS implementations since the precharge transistors can be larger than pseudo NMOS transistor. The pull-down resistance of the lattice is the same in dynamic and pseudo NMOS implementations. Simulation test benches used to

TABLE 5
Summary of results of designs implemented using two-terminal and four-terminal switches.

Instance	Designs with Two-Terminal Switches			Designs with Four-Terminal Switches											
	CMOS - Cadence			Pseudo NMOS						Footed Dynamic					
				Square shaped			H shaped			Square shaped			H shaped		
	area	delay	power	area	delay	power	area	delay	power	area	delay	power	area	delay	power
5xp1_1	55.35	0.28	0.98	16.23	2.83	9.25	19.93	0.50	29.15	17.41	1.86	0.52	21.21	0.41	0.34
apex4_17	81.32	0.38	1.20	27.58	7.00	5.54	34.90	1.05	19.35	29.05	3.88	0.78	36.51	0.74	0.50
apex4_18	99.01	0.38	1.47	31.12	6.79	5.64	39.39	1.13	20.24	32.71	4.00	0.89	41.12	0.78	0.56
b12_02	31.58	0.25	0.77	11.45	2.59	8.63	14.04	0.48	27.67	12.39	1.84	0.41	15.06	0.40	0.28
ex5_07	42.46	0.32	0.80	17.64	2.14	13.12	21.30	0.59	32.21	19.02	1.33	0.56	22.76	0.34	0.36
ex5_10	20.53	0.23	0.37	13.62	1.32	13.10	16.44	0.50	33.16	14.76	1.07	0.49	17.65	0.30	0.33
ex5_12	24.20	0.20	0.43	11.62	2.23	11.60	14.01	0.53	31.17	12.63	1.35	0.41	15.09	0.32	0.28
ex5_21	38.51	0.30	0.71	15.63	2.31	13.27	18.87	0.52	30.58	16.89	1.15	0.48	20.20	0.32	0.32
misex1_02	22.25	0.22	0.65	13.28	2.70	7.92	16.50	0.53	26.01	14.28	1.94	0.48	17.59	0.46	0.32
mp2d_03	29.91	0.31	0.52	16.23	4.02	6.52	19.93	0.55	27.16	17.41	2.95	0.49	21.21	0.53	0.33
mp2d_04	27.06	0.34	0.43	13.41	4.99	5.09	16.92	0.69	17.23	14.39	3.55	0.44	18.02	0.62	0.31
sao2_01	94.23	0.43	1.22	42.51	10.83	3.05	54.92	1.81	12.11	44.25	7.24	1.01	56.88	1.42	0.64
Average	47.18	0.30	0.80	19.19	4.15	8.56	23.93	0.74	25.50	20.43	2.68	0.58	25.28	0.55	0.38

obtain values in Table 5 allow the input changes only during the precharge time. All intermediate nodes discharge during precharge phase, improving the discharge speed. Considering that while pseudo NMOS implementation of switching lattices consume static power, the footed dynamic NMOS implementations do not, their power consumption values are much smaller than those of the pseudo NMOS implementations.

In pseudo NMOS structure, we observed that realization of logic functions using the proposed H shaped four-terminal switches have less delay contrary to the cost of higher power consumption. Since the off current of the H shaped device is higher than that of the square shaped device, pull-down resistance of the lattice is lower. Therefore, average delay of the H shaped device is approximately six times lower and its power consumption is three times higher with respect to those in the square shaped device. The use of footed dynamic logic significantly improves the power consumption. It is reduced from $25.50\mu W$ to $0.38\mu W$ for the H shaped device and from $8.56\mu W$ to $0.58\mu W$ for the square shaped device. Moreover, the average delay for logic circuits are also improved in the footed dynamic logic structures and it is approximately five times lower for the H shaped device when compared with the square shaped device.

The proposed implementations with four-terminal switches reduce the area by almost half and they provide comparable delay and power performances, especially using the footed dynamic logic structure when compared to the conventional CMOS implementations. To summarize, implementing logic functions with the proposed four-terminal switches has advantages over the conventional CMOS implementations.

6 Conclusions

In this article, we present the implementations of switching lattices using the proposed square and H shaped four-terminal devices and the realizations of logic functions on these lattices using the pseudo NMOS and footed dynamic logic structures under the TCAD and LTspice environments. We show that the implementations using four-terminal switches occupy significantly smaller area than those of designs using two-terminal switches. Also, the delay and power consumption of designs using four-terminal switches can be reduced significantly using the footed dynamic logic structure, leading to prominent designs competitive to those using two-terminal switches.

Our improvement in terms of area over the conventional CMOS circuits is expected to get better if smaller CMOS technologies are used. We believe that we can significantly improve our area gain with under 30nm technologies. The reason is that the smaller CMOS technology the larger the interconnection problem which favors the switching lattices having dense, regular, and metal-connection-free structures.

Acknowledgments

This work is part of a project that has received funding from the European Union's H2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 691178, and supported by the TUBITAK-Career project #113E760 and TUBITAK-2501 project #218E068.

References

[1] S. Safaltin, O. Gencer, M. C. Morgul, L. Aksoy, S. Gurmen, C. A. Moritz, and M. Altun, "Realization of four-terminal switching lattices: Technology development and circuit modeling," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2019, pp. 504–509.

[2] Y. Huang, X. Duan, Y. Cui, L. J. Lauhon, K.-H. Kim, and C. M. Lieber, "Logic gates and computation from assembled nanowire building blocks," *Science*, vol. 294, no. 5545, pp. 1313–1317, Nov. 2001.

[3] G. Snider, "Computing with hysteretic resistor crossbars," *Applied Physics A: Materials Science & Processing*, vol. 80, no. 6, pp. 1165–1172, Mar. 2005.

[4] G. Snider, P. Kuekes, T. Hogg, and R. S. Williams, "Nanoelectronic architectures," *Applied Physics A*, vol. 80, no. 6, pp. 1183–1195, Mar. 2005.

[5] H. Yan, H. S. Choe, S. Nam, Y. Hu, S. Das, J. F. Klemic, J. C. Ellenbogen, and C. M. Lieber, "Programmable nanowire circuits for nanoprocessors," *Nature*, vol. 470, no. 7333, pp. 240–244, Feb. 2011.

[6] M. Altun and M. Riedel, "Logic synthesis for switching lattices," *IEEE Transactions on Computers*, vol. 61, no. 11, pp. 1588–1600, Nov 2012.

[7] G. Gange, H. Søndergaard, and P. J. Stuckey, "Synthesizing optimal switching lattices," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 20, no. 1, pp. 1–14, Nov 2014.

[8] M. C. Morgul and M. Altun, "Synthesis and optimization of switching nanoarrays," in *International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS)*, 2015, pp. 161—-164.

[9] A. Bernasconi, V. Ciriani, L. Frontini, V. Liberali, G. Trucco, and T. Villa, "Logic synthesis for switching lattices by decomposition with p-circuits," in *Euromicro Conference on Digital System Design (DSD)*, Aug 2016, pp. 423–430.
[10] A. Bernasconi, V. Ciriani, L. Frontini, and G. Trucco, "Synthesis on switching lattices of dimension-reducible boolean functions," in *IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, Sep. 2016, pp. 1–6.
[11] A. Bernasconi, V. Ciriani, L. Frontini, and G. Trucco, "Composition of switching lattices for regular and for decomposed functions," *Elsevier Journal on Microprocessors and Microsystems*, vol. 60, pp. 207–218, 2018.
[12] M. C. Morgül and M. Altun, "Optimal and heuristic algorithms to synthesize lattices of four-terminal switches," *Integration*, vol. 64, pp. 60–70, Jan. 2019.
[13] L. Aksoy and M. Altun, "A satisfiability-based approximate algorithm for logic synthesis using switching lattices," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, Mar. 2019, pp. 1637–1642.
[14] A. Bernasconi, F. Luccio, L. Pagli, and D. Rucci, *Literal Selection in Switching Lattice Design*. Advanced Boolean Techniques, Springer International Publishing, 2020, pp. 159–175.
[15] D. P. Foty, *MOSFET Modeling with SPICE: Principles and Practice*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1997. ISBN 0-13-227935-5
[16] N. Akkan, M. Altun, and H. Sedef, "Modeling and parameter extraction of ofet compact models using metaheuristics-based approach," *IEEE Access*, vol. 7, pp. 180 438–180 450, Dec. 2019.
[17] N. Arora, *Mosfet Modeling for Vlsi Simulation: Theory And Practice (International Series on Advances in Solid State Electronics)*. USA: World Scientific Publishing Co., Inc., 2006. ISBN 981256862X
[18] J. Meyer, "Mos models and circuit simulation," *RCA Review*, vol. 32, pp. 42–63, 1971.
[19] N. Weste and D. Harris, *Integrated circuit design: International version: A circuits and systems perspective*. Pearson Education, 2010.
[20] A. Bernasconi, A. Boffa, F. Luccio, and L. Pagli, "Two combinatorial problems on the layout of switching lattices," in *IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, 2018, pp. 137–142.
[21] S. Yang, *Logic Synthesis and Optimization Benchmarks User Guide: Version 3.0*. Microelectronics Center of North Carolina (MCNC), 1991.

Nihat Akkan received his BSc degree in electrical and electronics engineering from Hacettepe University, Ankara, Turkey, in 2011. He received his MSc degree in electronics engineering from Yıldız Technical University (YTU), Istanbul, Turkey, in 2015. He is currently a PhD student and a research assistant in the faculty of electrical and electronics engineering at YTU. His main research areas are compact modeling of organic and inorganic semiconductor devices, solid-state electronics, and circuit and system designs.

Şerzat Safaltın received his BSc degree in metallurgy and materials science engineering from Istanbul Technical University (ITU), Istanbul, Turkey, in 2015. He received his MSc degree in extractive metallurgical technologies engineering from ITU, in 2017. He is currently a PhD student and a research assistant in the metallurgical and materials engineering department at ITU. His research interests include modeling & simulation, 2D materials and metallic alloy nanoparticle production, conductive inks, and characterization methods.

Levent Aksoy received his M.S. and Ph.D. degrees in electronics and communication engineering and electronics engineering from Istanbul Technical University (ITU), Istanbul, Turkey, in 2003 and 2009, respectively. He worked as a post-doctoral researcher in Algorithms for Optimization and Simulation (ALGOS) group of the Instituto de Engenharia de Sistemas e Computadores (INESC-ID), Lisbon. Then, he joined Dialog Semiconductor as a digital design engineer and worked mainly on the design and verification of asynchronous logic circuits. Currently, he is a research fellow in the Emerging Circuits and Computation (ECC) Group in ITU. His research interests include CAD for VLSI circuits with emphasis on solving EDA problems using SAT models and optimization techniques.

İsmail Çevik received his BSc degree in electrical engineering from Boğaziçi University, Istanbul, Turkey, in 2003, his MSc degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 2004 and PhD degree in electrical engineering from University of Idaho, Moscow, ID, USA, in 2014. He is currently working as an assistant professor in the electronics and communication engineering department at Istanbul Technical University, Istanbul, Turkey. His research interest includes high-speed analog and mixed-signal integrated circuit (IC) design, CMOS image sensors, ultra-low power IC design, micro solar cell structures, energy efficient DC-DC converters, and high-speed data converters.

Herman Sedef received B.Sc. 1984, M.Sc. 1987 and Ph.D. 1994 in electronics and communication engineering, from Yıldız Technical University (YTU), Istanbul, Turkey. He was working as a research assistant in between 1986 and 1994. From 1994 to 2000, he was an assistant professor in the department of circuit and systems in YTU. From 2000 to 2007, he was an associate professor in the department of circuit and systems in YTU. From 2007, he has been working as a professor in YTU. His research interests are active filter synthesis, filter design and realization of transfer functions.

Csaba Andras Moritz received his Ph.D. degree in computer systems from the Royal Institute of Technology, Stockholm, Sweden, in 1998. From 1997 to 2000, he was a Research Scientist with the Laboratory for Computer Science, The Massachusetts Institute of Technology (MIT), Cambridge, MA, USA. He has consulted for several technology companies in Scandinavia and held industrial positions ranging from CEO to CTO and to founder. His most recent company, BlueRISC Inc., develops security microprocessors, hardware-assisted security, and system assurance solutions for antitamper and cyber defense. He is currently a Professor in the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA, USA. His research interests include nanoelectronics and nanoscale systems, computer architecture, and security.

Mustafa Altun received his BSc and MSc degrees in electronics engineering at Istanbul Technical University in 2004 and 2007, respectively. He received his PhD degree in electrical engineering with a PhD minor in mathematics at the University of Minnesota in 2012. Since 2018, he has served as an associate professor of electrical engineering at Istanbul Technical University. Dr. Altun runs the Emerging Circuits and Computation (ECC) Group in the same university. Dr. Altun has been served as a principal investigator/researcher of various projects including EU H2020 RISE, National Science Foundation of USA (NSF), TUBITAK Career, and TUBITAK University-Industry Collaboration projects. He is an author of more than 50 peer reviewed papers and a book chapter, and the recipient of the TUBITAK Success, TUBITAK Career, and Werner von Siemens Excellence awards.