Masked Dual-Rail Pre-Charge Logic: DPA-Resistance without Routing Constraints *

Thomas Popp[1] and Stefan Mangard[1]

Institute for Applied Information Processing and Communications (IAIK)
TU Graz, Inffeldgasse 16a, A-8010 Graz, Austria
{Thomas.Popp, Stefan.Mangard}@iaik.at

Abstract. During the last years, several logic styles that counteract side-channel attacks have been proposed. They all have in common that their level of resistance heavily depends on implementation constraints that are costly to satisfy. For example, the capacitive load of complementary wires in an integrated circuit may need to be balanced. This article describes a novel side-channel analysis resistant logic style called MDPL that completely avoids such constraints. It is a masked and dual-rail pre-charge logic style and can be implemented using common CMOS standard cell libraries. This makes MDPL perfectly suitable for semi-custom designs.

Keywords: Side-Channel Analysis, DPA, Hardware Countermeasures, MDPL, Masking Logic, Dual-Rail Pre-Charge Logic

1 Introduction

During the last years, many logic styles that counteract side-channel analysis (SCA) attacks have been proposed. The big advantage of counteracting SCA attacks at the logic level is that this approach treats the problem right where it arises. If the basic building blocks, *i.e.* the logic cells, are resistant against SCA attacks, a designer can build a digital circuit with an arbitrary functionality and it will also be resistant against SCA attacks. Having SCA-resistant cells means that hardware as well as software designers do not need to care about SCA attacks any more. This greatly simplifies the design flow of a cryptographic device. Only the designers of the logic cells themselves need to be aware of SCA attacks.

An asynchronous logic style that makes devices more resistant against SCA attacks has for example been presented in [13]. However, it has been shown in [5] that this logic style has some weaknesses.

So far, the most promising logic styles to make devices resistant against SCA attacks are dual-rail pre-charge (DRP) logic styles that consume an equal amount of power for every transition of a node in a circuit. The most relevant logic styles

* This work has been supported by the European Commission under the Sixth Framework Programme (Project SCARD, Contract Number IST-2002-507270).

of this kind are SABL [18, 19], WDDL [20], and Dual-Spacer DRP [16]. In these DRP logic styles, the signals are represented by two complementary wires. The constant power consumption is achieved by guaranteeing that in every clock cycle one of these two wires is charged and discharged again. Which one of the two wires performs this charge and discharge operation depends on the logical value that the wires represent.

Obviously, a constant power consumption can only be achieved, if the complementary wires have the same capacitive load. Otherwise, the amount of energy needed per clock cycle would depend on which of the two nodes is switched and therefore would be correlated to the logical value. Unfortunately, the requirement to balance the capacitive load of two wires is hard to fulfill in a semi-custom design flow.

In a semi-custom design flow, so-called EDA tools place and route a digital circuit automatically. There exist only sub-optimal mechanisms to tailor the place and route operation such that the capacitive load of two wires is equal. Such a partial solution is for example parallel routing as introduced by Tiri and Verbauwhede [21]. However, integrating such mechanisms becomes more and more difficult for deep submicron process technologies where the transistor sizes and wiring widths continuously shrink. Hence, the capacitance of a wire more and more depends on the state of adjacent wires rather than on the length of the wire and its capacitance to VDD or GND. Therefore, it is very hard to guarantee a certain resistance against SCA attacks, if a DRP circuit is placed and routed automatically. Placing and routing a circuit manually, *i.e.* doing a full-custom design, significantly increases the design costs.

In this article, we present an approach that can easily be integrated in existing semi-custom design flows, *i.e.* there are no constraints for the place and route operation. The basic idea of our approach is to use masked cells to randomize the power consumption of cryptographic devices.

So far, masking has mainly been used at the software level. In [3] for example, Chari et al. analyze a secret sharing scheme where each intermediate bit of the original calculation is probabilistically split into k shares and every subset of $k-1$ shares is statistically independent of the original bit. A similar approach is described by Goubin and Patarin in [6] to mask the S-boxes of DES.

Masking on the gate level was considered for the first time in a patent [11] of Messerges et al. in 2001. However, the masked gates described in [11] are extremely big because they are built based on multiplexors. A different approach has been pursued later on by Gammel et al. in [7]. This patent shows how to mask complex circuits such as crypto cores, arithmetic-logic units, and complete micro controller systems. Other masked logic styles have been presented in [10] and in [22] where masked cells are built from standard CMOS cells.

The problem with all the mentioned masked logic styles is that glitches occur in these circuits. As shown in [9] and in [17], glitches in masked CMOS circuits reduce the SCA resistance significantly. Therefore, glitches must be considered when introducing an SCA countermeasure based on masking. In [17], a masked logic style called random switching logic (RSL) is presented. It avoids glitches

in the circuit. Yet, RSL needs a careful timing of enable signals. Furthermore, a new standard cell library must be compiled where all combinational gates have enable inputs.

In the current article, we present the so-called masked dual-rail pre-charge logic (MDPL) that uses masking at the gate level and that avoids glitches in the circuit by using a dual-rail pre-charge approach. There are no constraints for the place and route process. All MDPL cells can be built from standard CMOS cells that are commonly available in standard cell libraries.

Section 2 of this article explains the functionality and implementation details of MDPL cells. Furthermore, the overall architecture of MDPL circuits is introduced. In Section 3, experimental results of MDPL cells and circuits are presented. Conclusions are formulated in Section 4.

2 Masked Dual-Rail Pre-Charge Logic

Currently the most promising DPA-resistant logic styles require the balancing of complementary wires. In the following sections, the masked dual-rail pre-charge logic (MDPL) is introduced, which completely avoids this constraint. Furthermore, MDPL can be implemented using a commonly available standard CMOS cell library. Hence, MDPL can be used easily in semi-custom design flows.

2.1 Masking CMOS Logic

Currently, the most widely used logic style to implement digital integrated circuits is CMOS [23]. A main characteristic of CMOS logic is that it requires primarily dynamic power while its static power consumption is almost zero. The dynamic power consumption is caused by transitions of logic signals that occur in the CMOS circuit. The type and the probability of signal transitions depend on the logical function of a circuit and on the processed data. As a result, the power consumption of a CMOS circuit depends on the data that is being processed and hence, DPA attacks as described in [8] are possible.

In a digital CMOS circuit, there are essentially four transitions that can occur at a node of the circuit at a given moment of time. These include the two degenerated events where the node's value stays the same. Table 1 lists all transitions that can occur at a node n storing the data value d_{t-1} at the time $t-1$ and the value d_t at the time t. The energy that is dissipated in order to perform the respective transition is denoted by $E_{00} \dots E_{11}$. Each of these transitions occurs with a certain probability, denoted by $p_{00} \dots p_{11}$.

In a DPA attack on a cryptographic device, several power traces or EM traces of the device are recorded while it operates on different input data. The traces are then split into two sets according to the value d_t at a certain time t. d_t is calculated based on the input data and a key hypothesis. Subsequently, the attacker determines the difference of the means (DM) of the two sets of traces. We refer to these means as $M_{d_t=0}$ and $M_{d_t=1}$.

Table 1. Transitions of the value d of a node in a CMOS circuit

d_{t-1}	d_t	Energy	Probability
0	0	E_{00}	p_{00}
0	1	E_{01}	p_{01}
1	0	E_{10}	p_{10}
1	1	E_{11}	p_{11}

Of course, at the time t not only the value of node n performs a transition. Several other nodes also switch their value at this moment of time. However, the energy dissipation that is caused by these other nodes can be modeled as gaussian noise (see for example [12]). Therefore, the expected value of the difference of the means, $\mathcal{E}(DM_{d_t})$, can be calculated as shown in Equation 1.

$$\mathcal{E}(DM_{d_t}) = \mathcal{E}(M_{d_t=1}) - \mathcal{E}(M_{d_t=0}) = \frac{p_{11}E_{11} + p_{01}E_{01}}{p_{11} + p_{01}} - \frac{p_{00}E_{00} + p_{10}E_{10}}{p_{00} + p_{10}} \quad (1)$$

In case of standard CMOS logic ($E_{00} \approx E_{11} \ll E_{10} \neq E_{01}$), this difference is different from zero and can therefore be detected by an attacker. At all other moments of time, except for t, the partitioning of the traces according to d_t is meaningless. Consequently, the expected value for the difference of the means at these moments of time is zero. Furthermore, if a wrong key value is used to calculate d_t, the partitioning of the traces is again meaningless and also leads to an expected value of zero. As a result, the attacker in general gets a significant peak for a single key hypothesis, which is then the correct key.

The straightforward method to prevent an attacker from seeing such a peak is to use cells with the property that $E_{00} = E_{01} = E_{10} = E_{11}$. This is in fact the motivation for using dual-rail pre-charge logic styles such as SABL [18] or WDDL [20]. DRP logic styles have the property that transitions need the same amount of energy, if all pairs of complementary wires are perfectly balanced, *i.e.* have the same capacitive load. However, as already discussed, this requirement is very hard or even impossible to guarantee. This is the motivation for MDPL. MDPL is based on a completely different approach to prevent DPA attacks.

Resistance against DPA attacks at the gate level cannot only be achieved by consuming the same amount of energy for all transitions, but also by randomizing the logic signals in the circuit. This is the basic idea of MDPL. Instead of representing a logical value d by two complementary signals $\mathbf{d}$ and $\overline{\mathbf{d}}$, we represent d by $d_m = d \oplus m$, where $\oplus$ denotes the addition modulo 2 and m is a mask value. The mask is randomly generated and updated in every clock cycle.

Table 2 shows all transitions that can occur at a node n storing the masked value d_m when the time moves from $t-1$ to t. It also shows the required energy and the probability of each transition. The probability in line 1 is for example calculated as follows: $\frac{1}{4}p_{00} = p(m_{t-1} = 0) \cdot p(m_t = 0) \cdot p(d_{t-1} = 0) \cdot p(d_t = 0)$.

When a DPA attack is performed on the masked circuit, the power traces are split according to the value of d_t at time t. However, in the circuit now the masked value d_{m_t} is actually processed. In order to calculate $\mathcal{E}(M_{d_t=0})$, the odd

Table 2. Transitions of the value d_m of a masked node

Line no.	d_{t-1}	m_{t-1}	$d_{m_{t-1}}$	d_t	m_t	d_{m_t}	Energy	Probability
1	0	0	0	0	0	0	E_{00}	$\frac{1}{4}p_{00}$
2	0	0	0	1	1	0	E_{00}	$\frac{1}{4}p_{01}$
3	1	1	0	0	0	0	E_{00}	$\frac{1}{4}p_{10}$
4	1	1	0	1	1	0	E_{00}	$\frac{1}{4}p_{11}$
5	0	0	0	0	1	1	E_{01}	$\frac{1}{4}p_{00}$
6	0	0	0	1	0	1	E_{01}	$\frac{1}{4}p_{01}$
7	1	1	0	0	1	1	E_{01}	$\frac{1}{4}p_{10}$
8	1	1	0	1	0	1	E_{01}	$\frac{1}{4}p_{11}$
9	0	1	1	0	0	0	E_{10}	$\frac{1}{4}p_{00}$
10	0	1	1	1	1	0	E_{10}	$\frac{1}{4}p_{01}$
11	1	0	1	0	0	0	E_{10}	$\frac{1}{4}p_{10}$
12	1	0	1	1	1	0	E_{10}	$\frac{1}{4}p_{11}$
13	0	1	1	0	1	1	E_{11}	$\frac{1}{4}p_{00}$
14	0	1	1	1	0	1	E_{11}	$\frac{1}{4}p_{01}$
15	1	0	1	0	1	1	E_{11}	$\frac{1}{4}p_{10}$
16	1	0	1	1	0	1	E_{11}	$\frac{1}{4}p_{11}$

lines are used while $\mathcal{E}(M_{d_t=1})$ is calculated based on the even lines. As shown in Equation 2, the two expected values are equal. Therefore, the expected value of the difference of the means $\mathcal{E}(DM_{d_t})$ is zero and DPA attacks are not possible any more.

$$\mathcal{E}(M_{d_t=0}) = \mathcal{E}(M_{d_t=1}) = \frac{1}{4}\left(E_{00} + E_{01} + E_{10} + E_{11}\right) \tag{2}$$

Note that it is necessary to prevent glitches in a masked CMOS circuit in order to be resistant against DPA attacks. The fact that glitches lead to a leakage of side-channel information in masked CMOS circuits has been shown in [9] and in [17]. In CMOS circuits, the value of a node may switch several times before it reaches the correct value. The reason is that the input signals of the cell driving a node in general arrive at different moments of time. Glitches typically account for a significant amount of the dynamic power consumption of a CMOS circuit [15] and depend on the data that is processed. MDPL completely avoids glitches in the masked circuit as it is explained in the next section.

2.2 MDPL Cells

MDPL is a masked logic style that prevents glitches by using the DRP principle. Hence, for each signal d_m also the complementary signal $\overline{d_m}$ is present in the circuit. Every signal in an MDPL circuit is masked with the same mask m. The actual data value d of a node n in the circuit results from the signal value d_m that is physically present at the node and the mask m: $d = d_m \oplus m$. In the following, we show the implementation of an MDPL AND gate. All other combinational MDPL gates are based on this gate.

An MDPL AND gate takes six dual-rail inputs (a_m, $\overline{a_m}$, b_m, $\overline{b_m}$, m, $\overline{m}$) and produces two output values (q_m, $\overline{q_m}$). The truth table of an MDPL AND is shown in Table 3. The outputs of the MDPL AND gate are calculated according to the following equations: $q_m = ((a_m \oplus m) \wedge (b_m \oplus m)) \oplus m$ and $\overline{q_m} = ((\overline{a_m} \oplus \overline{m}) \wedge (\overline{b_m} \oplus \overline{m})) \oplus \overline{m}$

Table 3. Truth table of an MDPL AND gate

Line no.	a_m	b_m	m	q_m	$\overline{a_m}$	$\overline{b_m}$	$\overline{m}$	$\overline{q_m}$
1	0	0	0	0	1	1	1	1
2	0	0	1	0	1	1	0	1
3	0	1	0	0	1	0	1	1
4	0	1	1	1	1	0	0	0
5	1	0	0	0	0	1	1	1
6	1	0	1	1	0	1	0	0
7	1	1	0	1	0	0	1	0
8	1	1	1	1	0	0	0	0

Table 4. Truth table of an MDPL OR gate

Line no.	a_m	b_m	m	q_m	$\overline{a_m}$	$\overline{b_m}$	$\overline{m}$	$\overline{q_m}$
1	0	0	0	0	1	1	1	1
2	0	0	1	0	1	1	0	1
3	0	1	0	1	1	0	1	0
4	0	1	1	0	1	0	0	1
5	1	0	0	1	0	1	1	0
6	1	0	1	0	0	1	0	1
7	1	1	0	1	0	0	1	0
8	1	1	1	1	0	0	0	0

In Table 3, it can be seen that q_m and $\overline{q_m}$ can be calculated by the so-called majority (MAJ) function. The output of this function is 1, if more inputs are 1 than 0. Otherwise, the output is 0: $q_m = MAJ(a_m, b_m, m)$ and $\overline{q_m} = MAJ(\overline{a_m}, \overline{b_m}, \overline{m})$. A majority gate is a commonly used gate and it is available in a typical CMOS standard cell library. The schematic of a CMOS majority gate is shown in Figure 1.

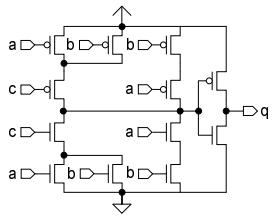

Fig. 1. Schematic of a CMOS majority gate

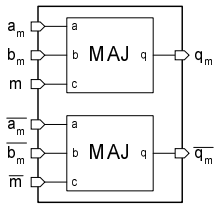

Fig. 2. Schematic of an MDPL AND gate

In an MDPL circuit, all signals are pre-charged to 0 before the next evaluation phase occurs. A so-called pre-charge wave is started from the MDPL D-flip-flops, similar to WDDL [20]. First, the outputs of the MDPL D-flip-flops are switched to 0. This causes the combinational MDPL cells directly connected to the outputs of the D-flip-flops to pre-charge. Then, the combinational gates in

the next logic level are switched into the pre-charge phase and so on. Note that also the mask signals are pre-charged. In Table 3, it can be seen that the pre-charge wave propagates correctly through the MDPL AND gate (see line 1 and line 8, respectively). The output signals of the MDPL AND gate are pre-charged if all inputs are pre-charged. All combinational MDPL gates are implemented in that way. Therefore, in the pre-charge phase, the pre-charge wave can propagate through the whole combinational MDPL circuitry and all signals are pre-charged correctly.

A majority gate in a pre-charge circuit switches its output at most once per pre-charge phase and at most once per evaluation phase, *i.e.* there occur no glitches. In a pre-charge circuit, all signals perform monotonic transitions in the evaluation phase ($0 \rightarrow 1$ only) and in the pre-charge phase ($1 \rightarrow 0$ only), respectively. Furthermore, the majority function is a so-called monotonic increasing (positive) function. Monotonic transitions at the inputs of such a gate lead to an identically oriented transition at its output. Hence, a majority gate performs at most one ($0 \rightarrow 1$) during the evaluation phase and at most one ($1 \rightarrow 0$) during the pre-charge phase. Since an MDPL AND gate is built from majority gates, an MDPL AND gate will produce no glitches.

Figure 2 shows the schematic of an MDPL AND gate. As already discussed, it is constructed using two CMOS majority gates. An MDPL NAND can be built by swapping (=inverting) the complimentary wires of the output signal. An MDPL OR can be generated by swapping the complementary masking wires (see Table 4) and an MDPL NOR is built by swapping the wires of the mask and that of the output signal. Note that the swapping of complementary wires does not invalidate the considerations made for the MDPL AND gate concerning propagation of the pre-charge wave and glitches. Therefore, also the MDPL NAND, OR and NOR gates propagate the pre-charge wave correctly and produce no glitches.

Figure 3 shows the schematic of an MDPL XOR gate that is built using three MDPL NAND gates. Note that the connections for the mask signals have been omitted for the sake of clarity. Using two 3-input XOR's as an MDPL XOR (like the MAJ gate is used in the MDPL AND) would also lead to a functionally correct gate. However, it would not be free of glitches since an XOR is not a monotonic function. The use of the MDPL NAND gates in order to implement the MDPL XOR gate prevents glitches. Furthermore, also the pre-charge wave is propagated correctly. An MDPL XNOR is realized by swapping the complementary wires of the output signal.

The implementation of an MDPL D-flip-flop is shown in Figure 4. The MDPL XOR gate at the input is used to switch the mask m of the current clock cycle to the new mask m_n of the next clock cycle. Hence, the CMOS D-flip-flop stores a value at the positive clock edge that is already masked with the mask of the next clock cycle. Note that the fixed input signals of the MDPL XOR gate still allow the gate to pre-charge correctly if all other inputs are 0. The MDPL D-flip-flop must be supplied with the special signals $m \oplus m_n$ and $\overline{m \oplus m_n}$ in order to switch masks.

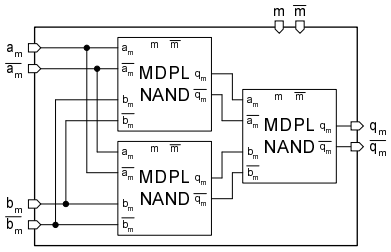

Fig. 3. Schematic of an MDPL XOR gate

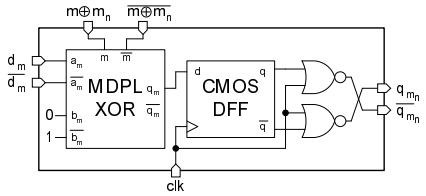

Fig. 4. Schematic of an MDPL D-flip-flop

The two CMOS NOR gates at the output of the MDPL D-flip-flop are required to start the pre-charge wave when the clock signal clk is 1. During $clk = 0$, the circuit is in the evaluation phase and the MDPL D-flip-flop provides the differentially encoded data value at its output. In the MDPL D-flip-flop, there is a timing constraint that must be satisfied: When the positive edge of the clock signal clk arrives at the cell, the CMOS NOR gates must switch their outputs to the pre-charge level 0 before the CMOS D-flip-flop stores the new input value at its output. Otherwise, there may be glitches introduced in the circuit. However, this timing constraint is satisfied because NOR gates are faster than D-flip-flops. Additionally, the CMOS D-flip-flop and the CMOS NOR gates are leaf cells of the clock tree that is build during the design process. Therefore, the skew between the clock signals connected to the CMOS D-flip-flop and the NOR gates is minimal.

There is also an implementation of the MDPL D-flip-flop possible that does not have such a timing constraint. Yet, it is much bigger and requires a doubling of the clock frequency in order to keep the same data rate. Details can be found in Appendix A. The fixed inputs of the MDPL XOR gate that is used in the MDPL D-flip-flop allows an optimization of the flip-flop. This optimization is presented in Appendix B.

Table 5 summarizes the basic MDPL cells and their respective implementation with standard CMOS cells. The table also shows the area complexity of the MDPL cells when the $0.35\mu m$ standard cell library C35B3 of austriamicrosystems [1] is used for the implementation. The area requirements of the MDPL cells are furthermore compared to the area requirements of their standard CMOS counterparts. Note that for the MDPL D-flip-flop, the optimized implementation as introduced in Appendix B is considered. The size of an MDPL circuit compared to the size of a standard CMOS implementation depends on the used cell types.

2.3 MDPL Circuits

The main features of MDPL circuits are that they can be based on a typical standard cell library and that there are no routing constraints concerning the

Table 5. MDPL cells and their CMOS implementations

MDPL cell	CMOS implementation of MDPL cell	Area (gate equivalents) of MDPL cell	Area (gate equivalents) of std. CMOS cell	Ratio $\frac{MDPL}{CMOS}$
Inverter	Wire swapping	0	0.67	0
Buffer	2×Buffer	2	1	2
AND, OR (2-in)	2×MAJ (3-in)	4	1.67	2.4
NAND, NOR (2-in)	2×MAJ (3-in)	4	1	4
XOR (2-in)	6×MAJ (3-in)	12	2.33	5.1
XNOR (2-in)	6×MAJ (3-in)	12	2	6
D-Flip-Flop	2×AND, 2×OR (both 2-in) 2×MAJ (3-in), 1×D-FF	17.67	5	3.5

balancing of complementary wires. Note that the dual-rail pre-charge property of MDPL is exclusively used to prevent glitches in the circuit. Hence, complementary wires are present, but they do not need to be balanced.

The design flow that is used to implement an MDPL circuit is almost the same as that one used when implementing a standard CMOS circuit. The only additional tool that is required is a converter that translates the synthesized CMOS netlist into an MDPL netlist. The MDPL cells in general cannot be used for synthesis since its masked and dual-rail attributes cannot be handled by typical state-of-the-art synthesizers. Furthermore, this would require the compilation of an MDPL synthesis cell library, which causes additional effort.

The converter replaces all CMOS cells by their MDPL versions as indicated in Table 5. It also adds the complementary wires, swaps a complementary wire pair if an inverter is present in the CMOS netlist and adds the mask nets. Dedicated single-rail nets like the clock net must be kept single-rail. We have built such a converter and have successfully translated and implemented an AES module. Details of this implementation can be found in Section 3.

The basic architecture of an MDPL circuit is shown in Figure 5. The combinational MDPL gates are supplied with the mask signals m and $\overline{m}$, the MDPL D-flip-flops must be supplied with the mask signals $m \oplus m_n$ and $\overline{m \oplus m_n}$. Note that also the mask signals are pre-charged in an MDPL circuit.

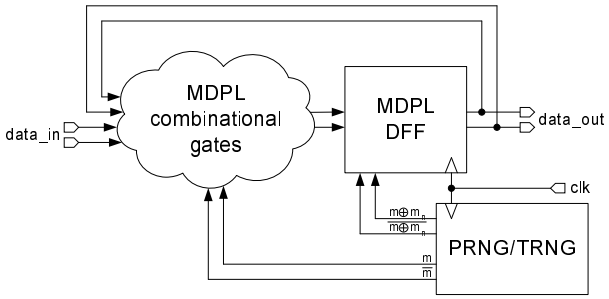

Fig. 5. Architecture of an MDPL circuit

The masks of the MDPL circuit are generated by a pseudo random-number generator (PRNG) that is seeded by a true random-number generator (TRNG) [2]. An MDPL circuit requires only one new masking bit per clock cycle.

In an MDPL circuit, all masks are dual-rail pre-charge signals. Therefore, it is not possible to perform SPA attacks an the masks. The complementary mask networks have approximately the same load. Hence, it is not possible to determine the masks based on one measurement. If the masks were not encoded in this way, SPA attacks would be possible. The mask networks are very big and hence, it would be possible for an attacker to determine whether a mask is switched or not.

3 Experimental Results

In this section, we present the results of some practical investigations of MDPL circuits. We compare how the DPA resistance of the dual-rail pre-charge logic style WDDL [20] and of MDPL depend on the balancing of complementary wires. Furthermore, we show the results of an AES module implemented in MDPL. For our circuit implementations we have used the $0.35\mu m$ standard cell library C35B3 of austriamicrosystems [1].

First, we have studied the effects of unbalanced routing on the dual-rail pre-charge logic style WDDL and compared them to the behavior of MDPL under the same condition. As expected, the results show that MDPL is completely independent to unbalanced routing while it has a significant effect on WDDL.

Figure 6 shows a comparison of the difference-of-mean-energies (DME) of CMOS, WDDL and MDPL implementations of a NAND gate. The energy consumption of the NAND gates were simulated by Spice simulations (*spectre* from Cadence) for all possible input values and input transitions. These energies were then split into two groups according to the respective output value of the gate. In a last step, the difference of the means of the energies in both groups were calculated. The DME value represents the height of the DPA peak if the output signal of such a NAND gate is attacked.

In case of the CMOS NAND gate, the DME value and thus the hight of the DPA peak raises linearly with the load at the output of the gate. For the differential NAND gates implemented in WDDL and MDPL, the effect of an unbalanced differential output load is also shown in Figure 6. An unbalance of 5% means that the output q has a 5% lower capacitance than the output $\overline{q}$ of the cell. The figure clearly shows that the DPA resistance of WDDL depends significantly on the degree of balancing. MDPL is completely immune to an unbalanced differential output load – the three lines lie upon each other. The difference between the MDPL NAND 0% and the WDDL NAND 0% is caused by charge sharing effects and small imbalances in the MDPL NAND gate which is more complex than the WDDL NAND gate (more internal nodes).

In a second experiment, we have simulated the power traces of an AES module [4] while it encrypts different data blocks. The power traces were then used in a DPA attack that targeted the intermediate result of the first SubBytes oper-

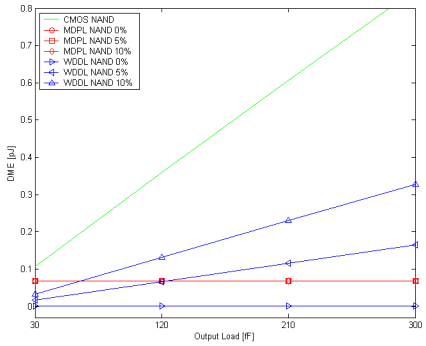

Fig. 6. Comparison of the DME of CMOS, WDDL and MDPL implementations of a NAND gate

ation after the initial AddRoundKey operation [14]. More precisely, one output bit of an 8-bit S-box was used as the selection bit in a standard DPA attack [8]. For the power simulation, the gate-level netlist of the AES module without parasitics was used. The power simulation was done for the 256 different input values of the data byte that corresponds to the key byte under attack.

In Figure 7, the result of the DPA attack on the AES module implemented in CMOS is shown. The correct key (43d) is clearly identifiable. Figure 8 shows the result of the DPA attack on the same AES implemented using MDPL. In this case, the correct key cannot be disclosed by the attack.

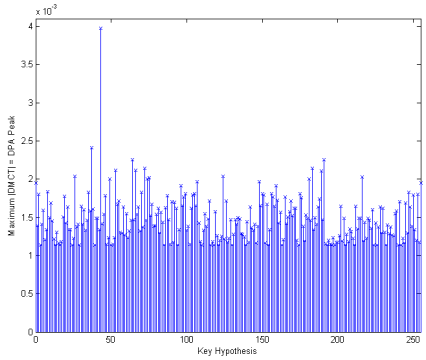

Fig. 7. DPA peaks for all key hypothesis for a DPA attack on an AES implemented in CMOS

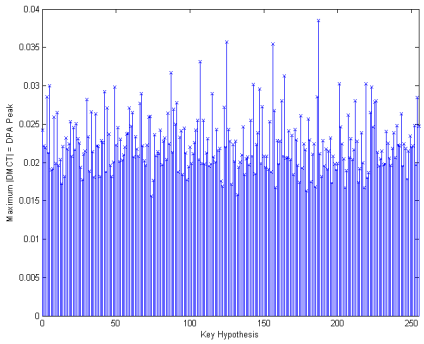

Fig. 8. DPA peaks for all key hypothesis for a DPA attack on an AES implemented in MDPL

In Table 6, a comparison of the main properties of the CMOS and the MDPL implementation of the AES module is shown. The increase in the area is signif-

icant, yet the DPA resistance of the MDPL circuit does not depend on the balancing of complementary wires. Speed is halved because not only the logic signals but also the pre-charge wave needs to propagate through the MDPL circuit.

Table 6. Comparison of AES implementations in CMOS and MDPL

	CMOS	MDPL	Ratio $\frac{MDPL}{CMOS}$
Area (gate equivalents)	3628	16465	4.54
Speed (MHz; worst-case speed corner)	16.91	9.82	0.58

4 Conclusions

We presented the DPA-resistant logic style MDPL which has two major advantages: it can be implemented using commonly available standard cells and, most importantly, its security does not rely on balanced complementary wires. Experimental results show that MDPL is effective against DPA attacks and that an unbalanced load of complementary wires does not affect the DPA resistance of the MDPL cells. The trade-off is in increased area requirements and power consumption and in a reduced circuit speed.

References

1. austriamicrosystems. `http://www.austriamicrosystems.com`.
2. Holger Bock, Marco Bucci, and Raimondo Luzzi. An Offset-Compensated Oscillator-Based Random Bit Source for Security Applications. In Marc Joye and Jean-Jacques Quisquater, editors, *Cryptographic Hardware and Embedded Systems – CHES 2004, Sixth International Workshop, Boston, USA, August 11-13, 2004, Proceedings*, volume 3156 of *Lecture Notes in Computer Science*, pages 268–281. Springer, 2004.
3. Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards Sound Approaches to Counteract Power-Analysis Attacks. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 398–412. Springer, 1999.
4. Martin Feldhofer, Sandra Dominikus, and Johannes Wolkerstorfer. Strong Authentication for RFID Systems using the AES Algorithm. In Marc Joye and Jean-Jacques Quisquater, editors, *Cryptographic Hardware and Embedded Systems – CHES 2004, Sixth International Workshop, Boston, USA, August 11-13, 2004, Proceedings*, volume 3156 of *Lecture Notes in Computer Science*, pages 357–370. Springer, 2004.
5. Jacques J. A. Fournier, Simon Moore, Huiyun Li, Robert D. Mullins, and George S. Taylor. Security Evaluation of Asynchronous Circuits. In Colin D. Walter, Çetin Kaya Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2003, 5th International Workshop, Cologne, Germany, September*

8-10, 2003, Proceedings, volume 2779 of *Lecture Notes in Computer Science*, pages 137–151. Springer, 2003.

6. Louis Goubin and Jacques Patarin. DES and Differential Power Analysis – The Duplication Method. In Çetin Kaya Koç and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems, First International Workshop, CHES'99, Worcester, MA, USA, August 12-13, 1999, Proceedings*, volume 1717 of *Lecture Notes in Computer Science*, pages 158–172. Springer, 1999.
7. Franz Klug, Oliver Kniffler, and Berndt Gammel. Rechenwerk, Verfahren zum Ausführen einer Operation mit einem verschlüsselten Operanden, Carry-Select-Addierer und Kryptographieprozessor. German Patent DE 10201449 C1, January 2002.
8. Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. In Michael Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999.
9. Stefan Mangard, Thomas Popp, and Berndt M. Gammel. Side-Channel Leakage of Masked CMOS Gates. In Alfred Menezes, editor, *Topics in Cryptology - CT-RSA 2005, The Cryptographers' Track at the RSA Conference 2005, San Francisco, CA, USA, February 2005, Proceedings*, volume 3376 of *Lecture Notes in Computer Science*, pages 351–365. Springer, 2005.
10. Renato Menicocci and Johan Pascal. Elaborazione Crittografica di Dati Digitali Mascherati. Italian Patent IT MI0020031375A, July 2003.
11. Thomas S. Messerges, Ezzy A. Dabbish, and Larry Puhl. Method and Apparatus for Preventing Information Leakage Attacks on a Microelectronic Assembly. US Patent 6,295,606, September 2001. Available online at `http://www.uspto.gov/`.
12. Thomas S. Messerges, Ezzy A. Dabbish, and Robert H. Sloan. Examining Smart-Card Security under the Threat of Power Analysis Attacks. *IEEE Transactions on Computers*, 51(5):541–552, January 2002.
13. Simon Moore, Ross J. Anderson, Paul Cunningham, Robert D. Mullins, and George S. Taylor. Improving smart card security using self-timed circuits. In *Proceedings of the Eighth International Symposium on Asynchronous Circuits and Systems (ASYNC 02)*, pages 211–218. IEEE Computer Society, 2002.
14. National Institute of Standards and Technology (NIST). FIPS-197: Advanced Encryption Standard, November 2001. Available online at `http://www.itl.nist.gov/fipspubs/`.
15. Jan M. Rabaey. *Digital Integrated Circuits*. Prentice Hall, 1996. ISBN 013-1786091.
16. Danil Sokolov, Julian Murphy, Alex Bystrov, and Alex Yakovlev. Improving the Security of Dual-Rail Circuits. In Marc Joye and Jean-Jacques Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, volume 3156 of *Lecture Notes in Computer Science*, pages 282–297. Springer, 2004.
17. Daisuke Suzuki, Minoru Saeki, and Tetsuya Ichikawa. Random Switching Logic: A Countermeasure against DPA based on Transition Probability. Cryptology ePrint Archive (`http://eprint.iacr.org/`), Report 2004/346, 2004.
18. Kris Tiri, Moonmoon Akmal, and Ingrid Verbauwhede. A Dynamic and Differential CMOS Logic with Signal Independent Power Consumption to Withstand Differential Power Analysis on Smart Cards. In *28th European Solid-State Circuits Conference (ESSCIRC 2002)*, 2002.

19. Kris Tiri and Ingrid Verbauwhede. Securing Encryption Algorithms against DPA at the Logic Level: Next Generation Smart Card Technology. In Colin D. Walter, Çetin Kaya Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2003, 5th International Workshop, Cologne, Germany, September 8-10, 2003, Proceedings*, volume 2779 of *Lecture Notes in Computer Science*, pages 137–151. Springer, 2003.
20. Kris Tiri and Ingrid Verbauwhede. A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation. In *2004 Design, Automation and Test in Europe Conference and Exposition (DATE 2004), 16-20 February 2004, Paris, France*, pages 246–251. IEEE Computer Society, 2004.
21. Kris Tiri and Ingrid Verbauwhede. Place and Route for Secure Standard Cell Design. In *CARDIS 2004 - Sixth Smart Card Research and Advanced Application IFIP Conference, 23-26 August 2004, Toulouse, France*, 2004.
22. Elena Trichina. Combinational Logic Design for AES SubByte Transformation on Masked Data. Cryptology ePrint Archive (`http://eprint.iacr.org/`), Report 2003/236, 2003.
23. Neil H. E. Weste and Kamran Eshraghian. *Principles of CMOS VLSI Design - A Systems Perspective*. Addison-Wesley, 2nd edition, 1993. ISBN 0-201-53376-6.

A An MDPL D-Flip-Flop Without Timing Constraints

As mentioned in Section 2.2, the MDPL D-flip-flop can be designed in a way that it has no timing constraints that must be satisfied. In the original MDPL D-flip-flop implementation shown in Figure 4, the CMOS NOR gates must switch the outputs to pre-charge level before the CMOS D-flip-flop stores a new value at the positive clock edge. Otherwise, the MDPL D-flip-flop may emit glitches which potentially cause side-channel leakage.

In general, CMOS NOR gates switch significantly faster than CMOS D-flip-flops. The clock signal connected to the CMOS NOR gates and the CMOS D-flip-flop arrive nearly at the same time since these cells are leaf nodes of the clock tree. Therefore, the skew should be minimal. Consequently, the timing constraint should be satisfied if the clock tree is created correctly.

If it is necessary to avoid such a timing constraint at all, it is possible to use an MDPL D-flip-flop implementation as shown in Figure 9. This version uses four CMOS D-flip-flops. One column of the flip-flops stores the pre-charge signal 0, hence these two flip-flops need to be reset in the beginning. The other column stores the differential encoded data value. Note that for a correct power-on reset of these two flip-flops, the above one needs to be reset to 0 while the below one needs to be preset to 1.

The sequence of the pre-charge and evaluation phases with respect to the clock signal when using the MDPL D-flip-flop without timing constraints is shown in Figure 10. A disadvantage is that the clock rate must be doubled in order to keep the data rate of the circuit constant. This increases the power consumption significantly.

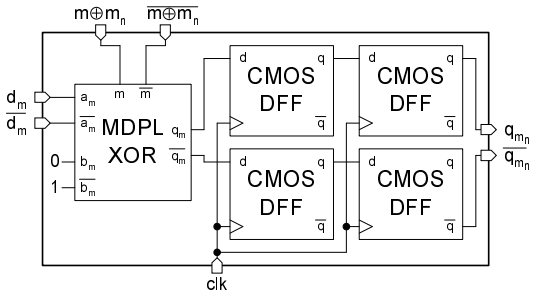

Fig. 9. Schematic of an MDPL D-flip-flop with no timing constraints

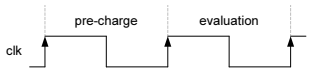

Fig. 10. Sequence of pre-charge and evaluation phases in an MDPL circuit using MDPL D-flip-flops without timing constraints

B The MDPL D-Flip-Flop With an Optimized MDPL XOR

The schematic of the basic MDPL D-flip-flop introduced in Figure 4 can be optimized. The reason is that two inputs of the used MDPL XOR are fixed to 0 and 1, respectively. Those inputs are connected to two MDPL NAND gates inside the MDPL XOR. Each of these two MDPL NAND gates can be reduced to a CMOS AND and a CMOS OR gate as shown in Figure 11. This reduces the area requirements of the MDPL D-flip-flop (implemented using the $0.35\mu m$ standard cell library C35B3 of austriamicrosystems [1]) from 19 gate equivalents to 17.67 gate equivalents. AND and OR functions are both monotonic increasing functions, and so glitches cannot occur in the cell.

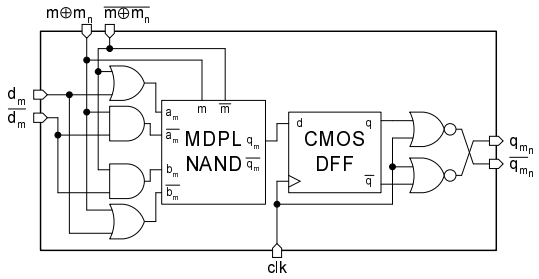

Fig. 11. Schematic of an MDPL D-flip-flop with optimized MDPL XOR