TECHNIQUES TO ENABLE THE USE OF BLOCK RAMS ON FPGAS WITH DYNAMIC AND DIFFERENTIAL LOGIC

Rajesh Velegalati, Jens-Peter Kaps

ECE Department, George Mason University
4400 University Drive, Fairfax, VA 22030, USA

ABSTRACT

Block RAMs (BRAMs) are commonly used by implementations of cryptographic algorithms on Field Programmable Gate Arrays (FPGAs). Unfortunately, any hardware implementation of a cryptographic function is susceptible to differential power analysis (DPA) attacks unless it is protected. Dynamic and Differential Logic (DDL), a constant power consumption logic style, is the most popular and successful defense method against DPA attacks. The required Measurements to Disclosure (MTD) of the key has been shown to be larger than the life period of the secret key in most systems. DDL implementations on FPGAs proposed till date incur a large area overhead. In this paper we show that BRAMs can be used within a DDL design without compromising its security. We propose and analyze several implementation techniques for using BRAMs in DDL designs. Our results show that such DDL implementations increase the MTDs by a factor 4 over unprotected designs which use BRAMs and by a factor 2.5 over DDL implementations which do not use BRAMs.

Index Terms— Cryptography, Differential Power Analysis, Block RAMs, SDDL, Xilinx FPGA.

1. INTRODUCTION

An FPGA consists of programmable Look-Up Tables (LUTs), flip-flops, Block RAMs (BRAMs), etc. and a network of programmable interconnects. In Xilinx Spartan 3 FPGAs two LUTs and two flip-flops are combined to one slice, four slices to one Configurable Logic Block (CLB). Four LUTs within each CLB can be configured as so called Distributed RAM of 16-bit per LUT. Each CLB has a dedicated switch box which provides programmable connections between slices and the FPGA wide routing network. Special interconnects are provided between a BRAM and its adjacent CLBs. BRAMs consist of static RAM cells and contain 18,432 bits of total RAM. Each has two completely independent access ports called Port A and Port B. Each port is synchronous with its own clock, clock enable and write enable [1]. BRAMs can be used as data storage, FIFOs, large (LUTs), data width converters, circular buffers, shift registers, wide logic functions and other basic functional primitives of cryptographic algorithms.

However, data integrity sensitive applications are vulnerable to passive, non-invasive attacks such as Differential Power Analysis (DPA) attacks. These attacks are powerful, easy to mount and almost guarantee success every time. Therefore, many countermeasures against DPA attacks were proposed. One such countermeasure is Dynamic and Differential Logic (DDL) introduced by Tiri [2]. DDL tries to "hide" the secret data by achieving constant power consumption per clock cycle using the following two basic principles.

Constant switching activity: During each clock cycle either a gate output in the direct path switches or the corresponding gate output in the complementary path.

Constant load capacitance: The capacitive loads driven by the gates in the direct path are equal to the loads driven by the gates in the complementary path.

DDL achieves these goals by duplication of the original circuit into direct and complementary parts. During the *precharge phase* all gate outputs are set to logic '0', during *evaluation phase* either the direct output evaluates to logic '1' or the complementary output.

Separated DDL for FPGAs (SDDL for FPGAs) is a variant of DDL tailored for lightweight implementations on FPGAs [3]. It allows FPGA CAD tools to have maximum flexibility to optimize a given design for the target FPGA. Such an optimized design allows logic packing in LUTs and also allows the use of FPGA intrinsic features such as BRAMs. A variant of the SDDL logic style called iWDDL is described in [4]. It extracts negative logic and isolates the resulting inverters between flip-flops. The authors analyze the usage of Distributed RAMs with iWDDL. In [5], the authors propose BCDL, a variant of Wave DDL [2] (WDDL), which uses all positive logic. It directly applies precharge as one of the inputs to the Look-up Tables (LUTs) and as a part of the address for BRAMs, which is similar to our design in Fig. 3. However this method leads to memory usage increase by a factor of 4. In this paper we introduce a method that halves that increase.

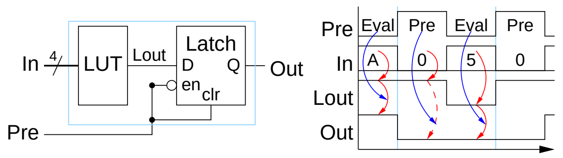


Fig. 1. LUT with Precharge

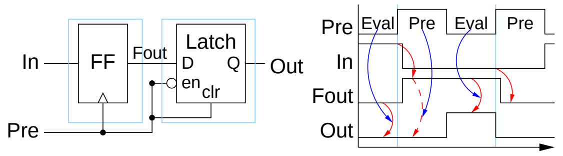


Fig. 2. Flip-Flop with Precharge

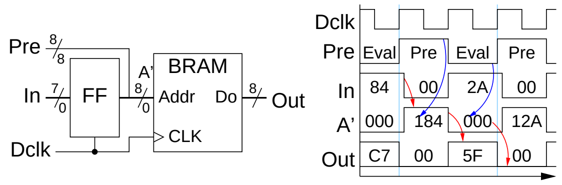


Fig. 3. Block RAM with Precharge Through Address

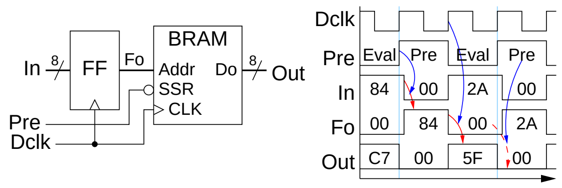


Fig. 4. Block RAM with Precharge Through Clear

The focus of this paper is to verify whether BRAMs in DDL implementations enhance or diminish their security. We propose several implementation techniques which facilitate the use of BRAMs in DDL implementations in Sect. 2. We test our designs on a circuit which uses basic functional primitives of the AES block cipher (Sect. 3) and analyze our results in Sect. 4.

2. IMPLEMENTATION OPTIONS

2.1. One Clock Cycle Operation

In this method, the precharge and evaluation phase share one clock cycle. The evaluation phase is defined by the "low" clock signal and the precharge phase by the "high" clock signal. This allows flip-flops to clock in new data at the rising edge of the clock as this is the transition from evaluation to precharge. The advantage of the one clock cycle operation is that the precharged circuit can be clocked with the same clock signal *clk* as external non-precharged circuits. The *Pre* input of the circuits in Fig. 1–Fig. 3 is connected to the *clk* signal.

In order to precharge the output of a LUT we use the technique proposed in [6]. The flip-flop following the LUT is configured as an asynchronously cleared latch. It forces the output of the slice to logic '0' during the precharge phase as shown in Fig. 1.

We precharge the output of a flip-flop in a similar manner and connect the flip-flop output to an asynchronously cleared latch (Fig. 2). Because the flip-flops within a slice share a common enable, clear, and clock input, we have to use a flip-flop of a different slice for the precharge latch. In earlier experiments [3] we discovered that connections within a CLB do not leak any exploitable information, however, connections between CLBs do. Therefore, we place the precharge latch in a flip-flop within the same CLB as shown in [7].

Unfortunately we cannot use the approach that we used for LUTs and flip-flops to precharge BRAMs If we use a precharge latch for a BRAM, wiring resources would be used to connect the non-precharged BRAM output to a CLB. This leads to information leakage. Nassar [5] uses a bit of the address to select a region of memory in which all data values are '0'. We use the MSB of the address for this as shown in Fig. 3. However, this doubles the memory usage within a BRAM which might not be feasible for memory demanding applications.

Therefore, we developed the precharge circuit shown in Fig 4 which does not double the memory usage. The outputs of the BRAMs are cleared to logic '0' by connecting the clock of the circuit to the synchronous set/reset (SSR) inputs of the BRAM. In both cases we have to operate BRAMs at twice the clock frequency (Dclk) compared to the rest of the circuit. One rising edge of Dclk resets the outputs of the BRAM to '0', the following releases the data stored at the address given in the previous Dclk cycle. This introduces a delay for the address by one Dclk cycle which we compensate for with a flip-flop. The flip-flop output is connected using leaky wiring resources to the BRAM, however, as can be seen in Fig. 4 this signal is precharged. Even though this precharge occurs during the evaluation phase this does not violate the goal of *constant switching activity*. The contents of the BRAMs are complemented using Eq. (1).

$$\mathrm{BRAM}(addr) = \overline{\mathrm{BRAM}\left(\overline{addr}\right)} \qquad (1)$$

2.2. Two Clock Cycle Operation

In this method, each phase, precharge and evaluate, takes a full clock cycle. Therefore, the precharged circuit has to run at twice the clock speed of external circuits. The precharge signal *Pre* reflects the phases and is low for one clock cycle and high for one clock cycle.

The circuit for precharging the output of a LUT shown in Fig. 1 does not need to change for this method. Only the *Pre* input needs to be connected to the *Pre* signal. As we have now one full clock cycle for each phase, we can simplify the precharge circuit for flip-flops by just adding another flip-flop instead of a latch as shown in Fig. 5. The placement constraint

Table 1. Implementation Results of our Test Design

Design	Slices	FFs	4 input LUTs	BRAMs	Minimum Delay	MTD	MTD Improvement over SE
1. SE design w/o BRAM	82	24	155	0	8.088 ns	> 450	1
2. SE design with BRAM	23	16	27	1	5.710 ns	$> 7,000$	15
3. SDDL w/o BRAM	283	80	502	0	18.352 ns	$> 10,000$	22
4. SDDL with SSR 1 clock cycle	51	100	54	2	16.138 ns	$> 27,000$	60
5. SDDL with Address 1 clock cycle	51	100	54	2	16.138 ns	$> 27,000$	60
6. SDDL with SSR 2 clock cycles	51	100	54	2	8.069 ns	$> 27,000$	60
7. SDDL with Address 2 clock cycles	51	100	54	2	8.069 ns	$> 27,000$	60

we had for flip-flop precharge can now be relaxed as all signals are precharged.

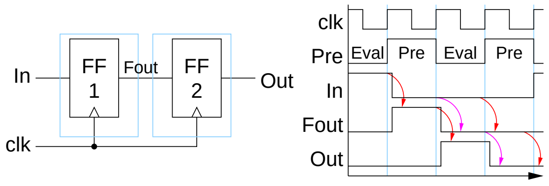


Fig. 5. Flip-Flop with Precharge Using Single Clock

For precharging the BRAM outputs in this method the circuits shown in Fig. 3 and Fig. 4 apply. However, the *Dclk* input must be connected to the *clk* signal and the *Pre* input to the *Pre* signal.

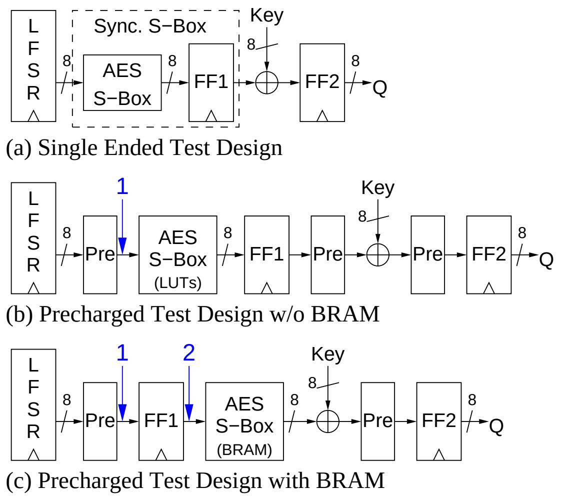
3. TEST SETUP AND ATTACK METHOD

The Test Design circuit consists of a synchronous (Sync.) S-Box whose input is connected to an 8-bit LFSR and output is XORed with an 8-bit Key. The result is stored in a register. The block diagram of this circuit is shown in Fig. 6(a). The Sync. S-Box can be implemented using look-up tables and a register as in Fig. 6(b), or a BRAM as in Fig. 6(c). The later option absorbs the register. The "Pre" blocks indicated in Fig. 6 are implemented using the different precharge options discussed in Sect. 2. We use the term Single Ended (SE) design to refer to the unprotected designs in this paper. In order to implement the SDDL versions of the SE designs we use the secure design flow described in [7]. It describes the step-by-step process of precharging, duplicating and complementing the logic. In our attacks we use Pearson's correlation to correlate instantaneous power consumption with hamming distance model [8]. The hamming distance equation for the attack on single ended implementations is shown in Eq. (2) and for SDDL implementations in Eq. (3).

$$P_{est.} = \mathrm{HD}(lfsr_{(i-1)}, \mathrm{SBOX}^{-1}(k_{guess} \oplus Q_i)) \qquad (2)$$

$$P_{est.} = \mathrm{HD}(0x00, \mathrm{SBOX}^{-1}(k_{guess} \oplus Q_i)) \qquad (3)$$

Attack point 1, indicated in Fig. 6 by arrow 1, is the precharged output of the LFSR. At attack point 2 data arrives half a clock cycle later than at point 1 during single cycle clock operation.



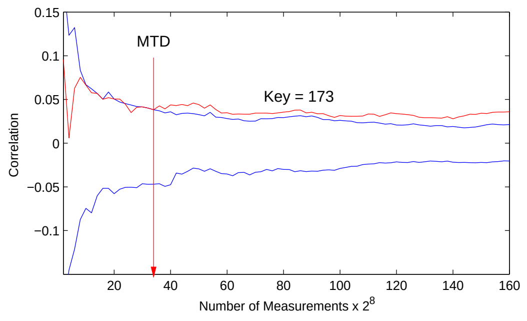
L F S R
Sync. S−Box
AES S−Box
FF1
Key
FF2
Q
8
(a) Single Ended Test Design
1
Pre
AES S−Box (LUTs)
FF1
Pre
Pre
FF2
(b) Precharged Test Design w/o BRAM
2
AES S−Box (BRAM)
(c) Precharged Test Design with BRAM

Fig. 6. Block Diagrams of Test Design

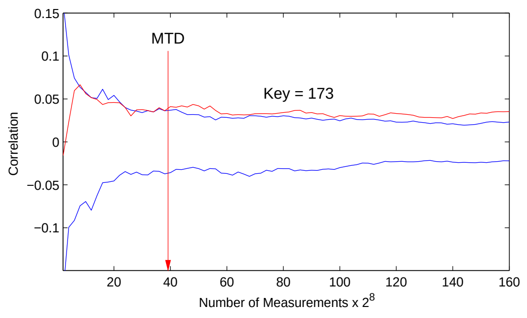
In case of two clock cycle operation the data arrives at point 2 one clock cycle later compared to point 1. Hence, in order to attack the design at point 2 appropriate adjustments must be made while correlating the data with the power models.

4. RESULTS AND CONCLUSIONS

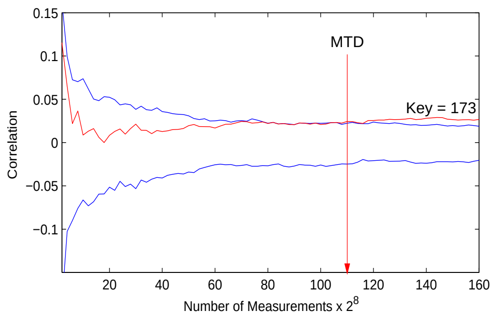
Table 1 compares the results of SE designs with several SDDL implementations with regards to area consumption, speed and Measurements to Disclosure (MTD) of the key. The results for MTD were obtained from the plots shown in Fig. 7. By comparing the MTD of the key between between the SE Designs 1 and 2, it is clear that BRAMs provide some inherent resistance against DPA attacks. All SDDL implementations of Design 2, namely Designs 4–7, have a 4 times higher MTD compared to the unprotected Design 2. Furthermore, they are more secure than Design 3 which is also an SDDL implementation but does not use BRAMs. We note that there is no difference in security between using address or SSR to force the BRAM output to '0' during precharge as can be seen from Fig. 7(c) and Fig. 7(f). Hence, we can conclude that lowering memory usage by using SSR does not impact security in a negative way. At attack point 2 the output of the flip-flop is precharged during the evaluation phase and the data evaluates



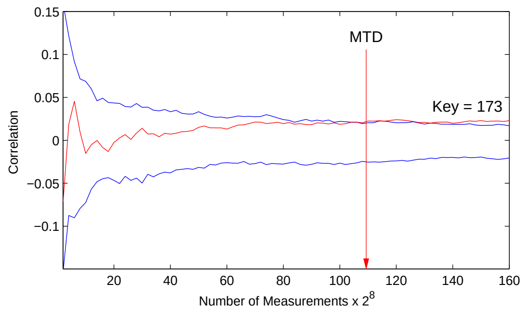
(a) SE Design with BRAM



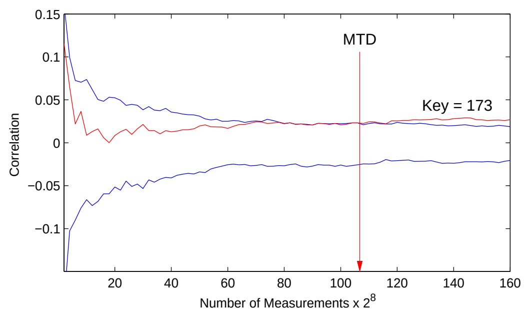
(b) SDDL Design w/o BRAM



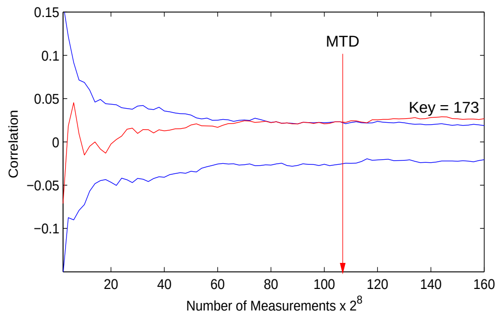
(c) SDDL Design with SSR, Attack Point 1, 1 clock cycle



(d) SDDL Design with SSR, Attack Point 1, 2 clock cycles



(e) SDDL Design with SSR, Attack Point 2, 1 clock cycle



(f) SDDL Design with Address, Attack Point 1, 1 clock cycle

Fig. 7. Measurements to Disclosure (MTD) for Implementation Options of Test Design

during precharge phase. At attack point 1 the data follows the phases normally. Fig. 7(c) and Fig. 7(e) show that this has no impact on security as the MTDs at both points are equivalent.

In this paper, we proposed and analyzed different implementation techniques for using BRAMs in DDL designs. Our results indicate that DDL implementations with BRAMs increase the MTDs by a factor 4 over unprotected designs which use BRAMs and a factor 2.5 over DDL implementations which do not use BRAMs.

5. REFERENCES

[1] Xilinx, Inc., *Using Block RAM in Spartan-3 Generation FPGAs*, xapp463 (v2.0) edition, Mar 2005.

[2] K. Tiri and I. Verbauwhede, "A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation," in *DATE'04*. Feb 2004, pp. 246–251, IEEE Computer Society.

[3] R. Velegalati and J.-P. Kaps, "DPA resistance for light-weight implementations of cryptographic algorithms on FPGAs," in *FPL 2009*, Martin Daněk, Jiří Kadlec, and Brent Nelson, Eds. Aug 2009, pp. 385–390, IEEE.

[4] R. P. McEvoy, C. C. Murphy, W. P. Marnane, and M. Tunstall, "Isolated WDDL: A hiding countermeasure for differential power analysis on FPGAs," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 2, no. 1, pp. 1–23, Mar 2009.

[5] M. Nassar, S. Bhasin, J.-L. Danger, G. Duc, and S. Guilley, "BCDL: A high speed balanced DPL for FPGA with global precharge and no early evaluation," in *DATE 2010*. Mar 2010, pp. 849–854, IEEE.

[6] P. Yu and P. Schaumont, "Secure FPGA circuits using

controlled placement and routing," in *CODES+ISSS '07*, New York, NY, USA, 2007, pp. 45–50, ACM.

[7] J.-P. Kaps and R. Velegalati, "DPA resistant AES on FPGA using partial DDL," in *FCCM 2010*. May 2010, pp. 273–280, IEEE.

[8] Eric Brier, Christophe Clavier, and Francis Olivier, "Correlation power analysis with a leakage model," in *CHES 2004*. Aug 2004, vol. 3156 of *LNCS*, pp. 135–152, Springer.