Energy Optimization of Unrolled Block Ciphers using Combinational Checkpointing

Siva Nishok Dhanuskodi and Daniel Holcomb

sdhanusk@umass.edu, holcomb@engin.umass.edu

Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, USA

Abstract. Energy consumption of block ciphers is critical in resource constrained devices. Unrolling has been explored in literature as a technique to increase efficiency by eliminating energy spent in loop control elements such as registers and multiplexers. However these savings are minimal and are offset by the increase in glitching power that comes with unrolling. We propose an efficient latch-based glitch filter for unrolled designs that reduces energy per encryption by an order of magnitude over a straightforward implementation, and by 28-32% over the best existing glitch filtering schemes. We explore the optimal number of glitch filters that should be used in order to minimize total energy, and provide estimates of the area cost. Partially unrolled designs also benefit from using our scheme with energies competitive to fully serialized implementations. We demonstrate our approach on the SIMON-128 and AES-256 block ciphers.

1 Introduction

Energy efficiency is one of the main concerns in low-power implementations of block ciphers, regardless of the key strength of the algorithm being implemented. Minimization of energy is important because it allows computation on scavenged energy, long-running devices on small batteries, and so forth. Aside from energy efficiency, area cost and latency are secondary concerns. In the consumer electronics market, a focus on area is justified because area cost translates directly to monetary cost of a chip. Yet, there are many interesting energy-constrained scenarios in defense and healthcare where the monetary cost of silicon area is less critical. In these scenarios, one may wish to trade area against latency by unrolling repetitive cryptographic operations to perform a larger share of the computation in each cycle in order to complete operation sooner. This is especially important for block ciphers such as SIMON that require a large number of rounds. Yet, in order to be suitable for highly-constrained devices, unrolling must be done without significantly compromising the energy efficiency of the computation.

In this work, we address the problem of efficiency in unrolled block ciphers, by presenting a new technique of combinational checkpointing to minimize their energy. We term the approach as *checkpointing* because we are adding state-holding elements at intermediate stages of the combinational logic, and each set of these elements stores an intermediate snapshot of the entire state of the block cipher operation. The specific contributions we make are as follows:

– We present an efficient latch-based glitch filter design that reduces energy of unrolled block ciphers.
– We find the optimal spacing of glitch filters in deeply unrolled block cipher implementations.
– For the first time, we give a technique that allows partially and fully unrolled block ciphers to have an energy efficiency that is competitive with serialized implementations.

2 Background and Related Work

There are a variety of low power design techniques for integrated circuits including near-threshold and subthreshold operation [9] and adiabatic logic styles [2]. However, in this work we focus exclusively on microarchitectural techniques for reducing energy instead of exotic circuits. In the remainder of this section, we review considerations for implementing block ciphers, and existing techniques for mitigating the glitches that dominate their power consumption.

2.1 Implementing Cryptographic Block Ciphers

Block ciphers are cryptographic primitives used to encrypt and decrypt data, typically used as part of a larger encryption mode of operation. Block ciphers are almost always implemented as components of a larger overall system-on-chip design, and this prevents the block cipher from being freely optimized independently of the other SoC components. For example, the block cipher will have to use the same fabrication process and supply voltage as the other components, and typically will share a common clock frequency to avoid clock generation and clock domain crossing. Therefore, any attempt at optimizing block ciphers may be constrained by these chip-scale implementation decisions.

The block cipher algorithm iterates over a round function for a specified number of times using different subkeys. The rounds can be implemented through sequential reuse of a single combinational block for each round, or they can be unrolled. If a design is fully serialized (no unrolling), one round function is computed in each clock cycle, and the number of cycles needed to encrypt a block is the same as the number of rounds in the block cipher algorithm. Yet, small low-power SoCs will typically operate at slow clock frequencies, and therefore the clock period may far exceed the critical path delay of a block cipher round. The latency of the block-cipher is then being increased unnecessarily due to the serialization of the round function.

Unrolling a block cipher is the process of instantiating multiple rounds of the algorithm combinationally to be completed within each clock cycle. Unrolling allows the result to be computed in fewer cycles at the cost of increased area of the combinational circuit. Unrolling also saves some amount of register energy, as energy is not spent storing signals at the output of each round like the fully serialized case. The unrolling of block ciphers as an energy optimization technique has been explored in a number of recent works [13, 5].

2.2 Glitches and Glitch Filtering

The limiting factor in energy minimization of block ciphers is switching energy. This is especially true in unrolled block ciphers because combinational logic glitches at the input of each round diffuse through the round to cause more glitches at the output of the round. Leakage power is negligible relative to switching power for typical clock periods and technologies used in low power designs [13]. Fundamentally, glitches occur because of mismatched arrival times of gate inputs. This causes the gate output to switch once when the first input arrives, and then switch again when the next input arrives. These two switching events then propagate to many other nodes and cause more switching events in a cascading fashion.

Several techniques to filter glitches have been proposed in literature. Pipelining [8, 19] stops glitches because they cannot propagate through a register, as a register can change its output value only once per clock cycle upon arrival of the clock transition. Gate-freezing [7] stalls the computation in a gate by using an NMOS footer transistor to filter 1-to-0 transitions. The stalled gate is allowed to compute only when its inputs have reached their final state. The scheme has a limitation in that it allows 0-to-1 transitions to pass through a stalled gate. Retiming [15] by moving or adding flip-flops in the datapath to high activity nodes that have a large fanout can reduce glitches and save power. Yet another approach is delay balancing to equalize input arrival times at a gate and reduce the number of output switching events [14, 12, 11].

An AND gate based glitch filtering scheme (Round Gating) was proposed in [4]. The output signals of each round in this scheme are gated by AND gates that wait on an enable signal. The enable signal is derived from a delayed clock such that it goes high to propagate the round outputs through the AND gates only after they have stopped glitching and become stable. A drawback of this scheme is that the enable signals must be reset low between the end of one computation and the start of the next in order to stop propagation of the glitches in the next operation. When the enable signals go low, waves of 0s propagate forward from the glitch filters and propagate through the circuit to charge and discharge the nodes in the round functions similar to a normal computation of the round function. Effectively, resetting the glitch filters is thus causing a second, unnecessary, power-wasting computation to occur. State-retaining barriers [16] provide a mechanism for preventing this power-wasting computation.

3 Methodology

Combinational checkpointing is a microarchitectural technique to increase energy efficiency in a combinational circuit by filtering glitches. In this section, we describe the application of latch-based checkpoints in a block cipher and the methodology used to evaluate the approach.

3.1 Proposed use of Checkpoints for Glitch Filtering

We propose a new standard-cell compatible glitch filtering mechanism as shown in Fig. 1. The topology is similar to that of round gating using AND gates [4],

except that the glitch filtering element consists of a positive latch implemented using a multiplexer (MUX) at the output of the round function. The purpose of the filter is to make sure that any glitching activity from its input is not propagated to its output.

The operation of the filter is as follows. The MUX holds on to its previous output value when the enable (select) signal is low, and becomes transparent when enable is high. This causes the latch to be transparent only during the enable pulse. The enable pulse is generated at the rising edge of the clock as the AND of the clock signal and a delayed inverted version of clock. The enable pulse is propagated to the glitch filters combinationally with timing controlled by adding a delay element per round function. If the propagation delay of the delay element (t_d) is greater than the critical delay of a round function (t_r), then round output r_i stabilizes before the rising edge of signal en_i, so the latches only become transparent after the glitching has stopped. Therefore, when this timing condition ($t_d > t_r$) is satisfied, glitches generated in round i do not propagate through the glitch filters to round $i + 1$. Because the latch stays open for the duration of the enable pulse, the circuit will function correctly as long as the round outputs stabilize before the falling edge of en_i, but the circuit will not filter any glitches that arrive when the latch is open, and the glitch filter will not have the intended effect.

The timing waveform for a single round is shown in Fig. 2. When the enable signal pulses at the first glitch filter, the stable outputs of round $i - 1$ propagate through round i and cause a total of 122 transitions on the 128 round output signals. The round outputs wait for the enable signal to arrive at the second glitch filter, and upon its arrival, only 60 transitions occur on the inputs of round $i + 1$; these 60 transitions are single transitions on 60 of the 128 signals, which is close to the expected number of bits that would differ between two uncorrelated 128-bit signals. In this case, the filter has prevented all the spurious glitches from propagating across rounds.

3.2 Evaluation Methodology

We use the SIMON and AES block ciphers to study the effectiveness of our glitch filtering scheme. SIMON is a lightweight Feistel cipher suitable for resource constrained systems, and we use SIMON-128 [6], which has a 128-bit key, 128-bit block size, and requires 68 rounds for each encryption. Being a very simple design, the RTL for our SIMON implementation is written by us and validated for correctness against a software implementation of the same. AES refers to three standardized variants [17] of the Rijndael cipher, based on a substitution-permutation network. Relative to SIMON, AES is a more complicated design, and we specifically use the most complicated variant, AES-256; which has 128-bit block size, a 256-bit key, and requires 14 rounds per encryption. The RTL for our AES implementation is publicly available from OpenCores.org [10], and we validate its correctness against an online AES software implementation. To give an idea of the relative scales of the two ciphers, the round and key functions of fully unrolled SIMON require around 30,000 gates, whereas the round and key functions of fully unrolled AES are more than 8 times larger, requiring around 250,000 gates.

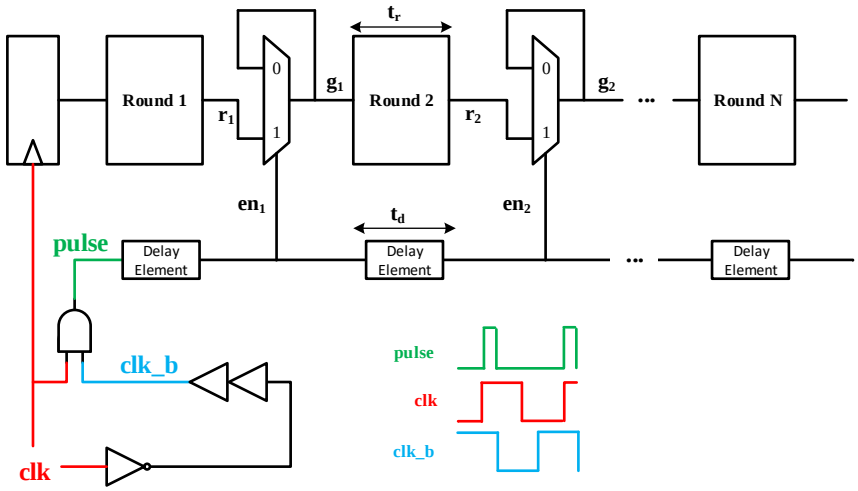


Fig. 1: Schematic of latch-based checkpoints for glitch filtering.

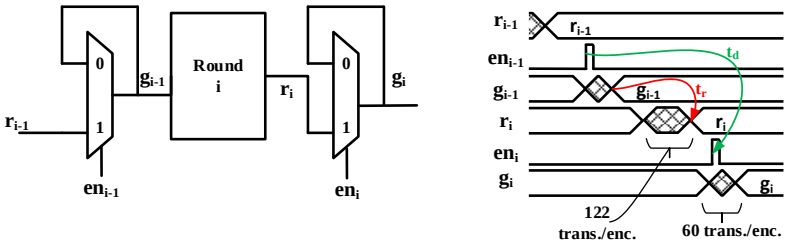


Fig. 2: Timing diagram of glitch filter operation, annotated with the number of switching events happening at each point in the circuit for SIMON-128.

All of the measurements we present in this work are from simulation. Specifically, we simulate designs with 45nm NCSU PDK [1] implemented using CMOS logic style. Synopsys Design Compiler and HSIM are used for synthesis and circuit simulation, respectively. We rely on circuit simulation rather than power simulations using characterized libraries to ensure that we accurately capture glitch propagation effects. Given the time consuming nature of circuit simulation on large designs, which takes several days per encryption for the unrolled AES design, we simulate only two encryptions per design, using inputs that are chosen at random. The first encryption initializes the circuit state, and the second encryption is used for measuring metrics described below. The accuracy of our results should not be compromised by the small number of encryptions simulated because a block cipher's behavior is fairly independent of the input value used. In support of this claim, the energy consumption of partially unrolled (17 rounds) SIMON for 100 random input vectors is shown in Fig. 3. The variation in energy consumption is small ($\sigma = 0.032pJ/bit$) for the chosen input vectors.

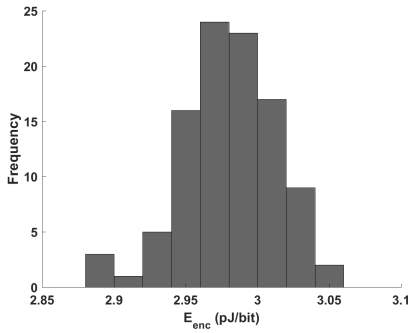


Fig. 3: SIMON-128 energy per encryption histogram for 100 random inputs.

Metrics such as toggle rate and energy consumption are measured during the circuit simulation and used to compare our scheme's performance with others. **Toggle rate** is measured as the average number of signal transitions at round outputs per encryption. For example, in SIMON-128 a round output has 128 signals. We compute the total number of signal transitions in all 128 signals that occur during an encryption operation, and divide by bit-width (128) to get the toggle rate. We present energy numbers using a metric of **energy-per-encrypted bit** denoted as $\boldsymbol{E_{enc}}$, which is the total energy consumed to perform an encryption operation divided by the number of encrypted bits generated during the operation. When considering individual rounds of the block cipher, we use as a metric the contribution of that round to the overall E_{enc}. In our experiments, clock frequencies are chosen such that idle time is minimal, and are above 10MHz in all cases.

4 Results

In this section we present results showing energy benefits of using checkpointing. We first demonstrate that glitch filtering using checkpointing leads to a reduction in toggle rate, which translates to energy savings. Further, we vary the number of checkpoints to explore the trade off between checkpointing overhead and glitching energy saved. Finally, we evaluate the effectiveness of checkpointing in partially unrolled designs, and also estimate area penalty incurred by checkpointing.

4.1 Comparison of Average Switching Rates

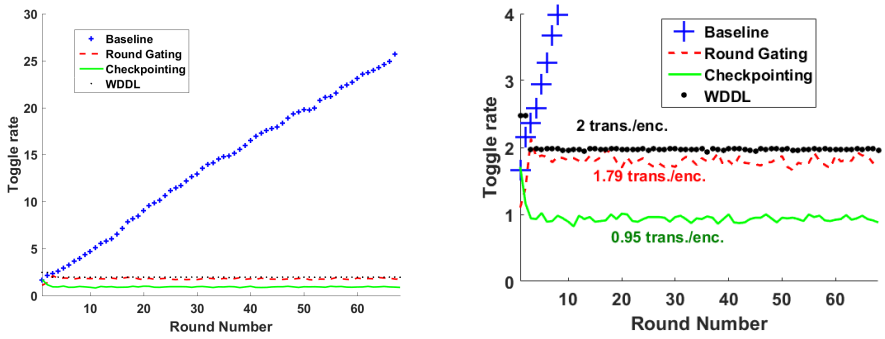
We first study the effectiveness of the proposed glitch filter by counting switching events on a fully unrolled implementation of SIMON-128. Fig. 4 shows a comparison of signal toggle rates (signal transitions/encryption) for the outputs of all the 68 round and key functions. In the ideal case of no glitching activity, at the round outputs, one can expect 0.5 transitions per signal for each encryption, as round outputs are uncorrelated across encryptions.

When no glitch filtering is used (baseline design), the switching activity is observed to increase linearly with logic depth (number of rounds). This increase in switching occurs because the logic of the block cipher tends not to mask transitions as they propagate, and because the diffusion property of block ciphers tends to propagate each transition out to many nodes. Our finding of linear increase is consistent with observations made in previous works [3]. For each encryption in the baseline design, the average switching across all rounds is 14.16 transitions per signal, and in the later rounds it is 2x larger than this average.

We analyze the effectiveness of checkpointing and two other techniques that mitigate switching. Compared to baseline, the Round Gating scheme [4] achieves a much lower average switching of 1.79 transitions per signal. Also, the switching activity stays fairly constant across rounds because glitches are never propagated across round boundaries. However as noted in Sec. 2, resetting the AND gates every clock cycle leads to unnecessary switching activity. Our checkpointing scheme has no such resetting and is therefore able to reduce switching to 0.95 transitions per signal, a 47% reduction relative to Round Gating. For comparison purposes, we implement SIMON-128 also using WDDL logic style [18]. WDDL is a dual-rail precharge based logic that is glitch free by design. To mitigate power side channel leakages, every signal pair in WDDL always has exactly 2 transitions per encryption; specifically, among the true and complementary representations of each signal, it is always the case that exactly one representation goes through a 1-0 transition during precharge and a subsequent 0-1 transition during evaluation.

4.2 Energy Savings from Checkpointing in Fully Unrolled Designs

The significant reduction in average switching rates implies that glitch filtering can reduce the overall energy used for encryption. In this section we study the energy savings achieved by using checkpointing to filter glitches in fully unrolled implementations of SIMON-128 and AES-256.



(a) Toggle rate (b) Zoomed in

Fig. 4: Comparison of the average toggle rate of the output signals of each round of SIMON-128 for four different implementation styles.

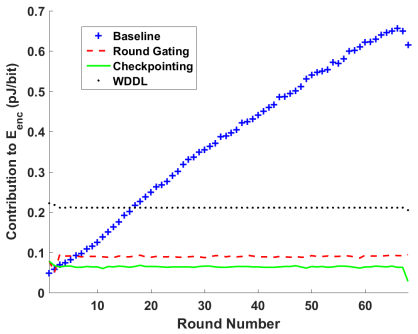


Fig. 5: Contribution of each round to the overall energy per encrypted bit in four different implementation styles of fully unrolled SIMON-128

SIMON-128 The energy use of each round in the fully unrolled SIMON-128 implementation is plotted in Fig. 5. The energy trends are similar to the toggle rate trends shown in Fig. 4. Tab. 1 lists the breakdown of energy per encryption (E_{enc}) for baseline (no glitch filter) designs and three glitch filtering schemes. A fully-unrolled implementation with checkpointing (4.46pJ/bit) performs much better than fully unrolled baseline (25.91pJ/bit) by saving glitching energy. Checkpointing is also competitive in energy with a baseline design that is not unrolled (1-unrolled, 3.78pJ/bit) while offering single cycle latency, compared to 68 cycles in the 1-unrolled baseline. In comparison to Round Gating [4], checkpointing consumes 27.9% lower E_{enc}. The savings comes from a 47% reduction in toggle rate which leads to a 44.6% reduction in data and key computation energy specifically, while the costs of other components are similar across the two schemes. Note that WDDL and Round Gating schemes have similar toggle rates,

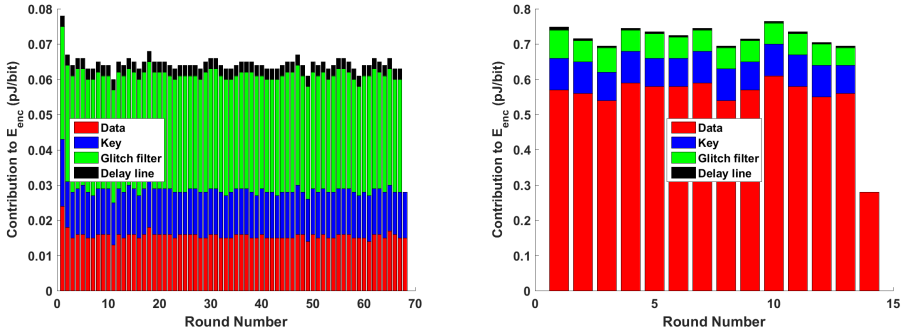
Table 1: Breakdown of E_{enc} (pJ/bit) in fully unrolled SIMON-128. Glitch filters are added after every round in Round Gating and our work.

E_{enc}	Baseline		Glitch filtering scheme (68-unrolled)		
breakdown	**1-unrolled**	**68-unrolled**	**Round Gating**	**Checkpointing**	**WDDL**
Data	0.58	16.37	1.90	1.07	6.95
Key	0.40	9.42	1.62	0.88	7.42
Glitch Filter	–	–	2.36	2.20	–
Delay line	–	–	0.18	0.19	–
Other	2.80	0.12	0.12	0.12	0.45
Total	**3.78**	**25.91**	**6.19**	**4.46**	**14.82**

yet WDDL consumes 2.4x more energy because it uses only positive gates, and therefore requires approximately 3x more gates to implement the same function.

Fig. 6a shows the breakdown of energy consumption per encryption for our scheme. As can be seen in the figure, the switching energy does not increase across rounds, because each round similarly starts its computation from a single switching event. However, the glitch filters themselves consume about 50% of the total energy relative to the extremely simple combinational round function of SIMON. Hence, there is a possibility that using fewer glitch filters might reduce E_{enc} further, if the glitches do not increase significantly. We explore this in Sec. 4.3. It can also be noted that the simple delay line that propagates the enable is not costly in energy, as it is a single inverter chain relative to a 128-bit wide computation path. The delay line does not require any tuning if care is taken by adding some margin (buffers) to ensure $t_d > t_r$ even in the presence of process variation (Fig. 2).

AES-256 We also study the effectiveness of our scheme for the larger design, the fully unrolled implementation of AES-256. The energy breakdown per encryption in Fig. 6b shows that glitches are filtered effectively as there is no significant increase in switching energy with logic depth (round number). The energy cost of glitch filtering is small compared to that of actual computation. Note that the last round in AES is simpler, and therefore consumes less energy. The energy breakdown summary is tabulated in Tab. 2. Our scheme consumes an E_{enc} of 9.77 pJ/bit, which is 7.5x lower than fully unrolled baseline and 32.6% lower than Round Gating. These savings directly come from a lower switching activity. Unlike the extremely simple round functions of SIMON, AES round and key functions constitute about 90% of the total energy. As a result, in comparison to Round Gating our scheme saves more energy in AES-256 (32.6%) than in SIMON-128 (27.9%). It is important to note that our checkpointing scheme has similar energy efficiency as a fully serialized implementation (no unrolling, consumes 9.69 pJ/bit) while achieving single cycle latency. This is because the fully serialized implementation incurs a penalty of 2.19pJ/bit for loop control and multiplexing, which is larger than leakage/glitch filter costs associated with checkpointing. Because AES-256 uses alternating key functions, we also implemented a 2-unrolled baseline design (not in table) that has smaller loop control penalty, but in that case glitches cause the total energy to increase to 13.7 pJ/bit for an encryption operation.



(a) SIMON (b) AES

Fig. 6: Energy/encryption breakdown in fully unrolled implementations using checkpointing after every round.

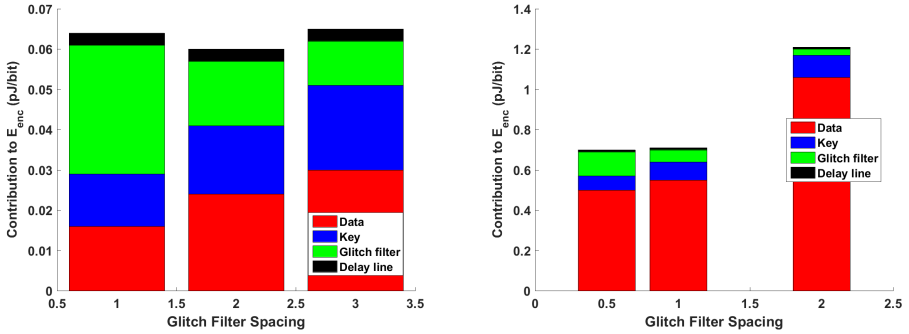
Table 2: Breakdown of E_{enc} (pJ/bit) in fully enrolled AES-256. Glitch filters applied after every round.

	No unrolling	**Baseline**	**Round Gating**	**Checkpointing**
Data	5.95	65.07	11.77	7.70
Key	1.55	8.64	1.82	1.13
Glitch Filter	–	–	0.82	0.82
Delay line	–	–	0.03	0.07
Other	2.19	–	–	–
Total	**9.69**	**73.76**	**14.50**	**9.77**

4.3 Optimal Placement of Checkpoints for Glitch Filtering

In this section we explore the optimal number of glitch filters to use in our scheme so as to minimize the total energy consumption. Energy optimal glitch filtering requires finding the right trade-off between the cost of glitch filtering and the energy saved by filtering glitches. If too many filters are used, then the cost of the filters themselves will dominate; but if too few filters are used, then the cost of the glitches will dominate. Fig. 7 shows how each round contributes to the energy per encrypted bit when different number of rounds are implemented between the checkpoints. When checkpoints are added after every round (spacing=1) in fully unrolled SIMON-128 (Fig. 7a), more energy is spent in glitch filtering than is spent in actual computation. However, if spacing is increased to 2 where checkpointing is done every other round, the average energy per round is decreased. Increasing the spacing beyond 2 further reduces the cost of glitch filtering but the glitches increase the key and data energy by a larger amount and the total energy increases. Therefore a spacing of 2 rounds between checkpoints is optimal for SIMON-128.

The energy breakdown of E_{enc} for each round of the fully unrolled SIMON-128 with optimal glitch filter placement is shown in Fig. 8. The even rounds have more glitching, and only the even rounds spend energy on checkpointing. At the



(a) SIMON (b) AES

Fig. 7: Energy efficiency varies with the spacing between checkpoints in fully unrolled designs. Performing more computation between checkpoints reduces checkpointing energy, but allows more data switching to occur

optimal spacing of 2, the design consumes 4.18pJ/bit per encryption which is 6.3% lower than the 4.46pJ/bit when checkpointing is applied after every round (Tab. 1). In addition, the area will be reduced because of the fewer checkpoints. Any block cipher implementation will have some optimal tradeoff of checkpointing energy versus glitching, but the specifics are of course design and technology dependent.

Fig. 7b shows that in AES, the much larger round function justifies adding glitch filtering after every round. Given the small energy cost of the checkpoints relative to round function, one might consider adding glitch filters at half round boundaries. Doing this reduces glitches but increases the cost of glitch filter such that the total energy consumption becomes comparable to glitch filter spacing of 1. Therefore, checkpoint spacing of 1 is optimal in AES-256 as it requires fewer glitch filters for the same energy efficiency as half-round checkpointing.

4.4 Checkpointing in Partially Unrolled Designs

Partially unrolled designs, which implement some number of rounds combinationally, offer a tradeoff between area and latency of encryption. Aside from this tradeoff, partial unrolling may also be desirable due to design constraints (area, clock period) which do not allow for a fully unrolled implementation. Since the optimal spacing of checkpoints is a low number (every round for AES-256, and every second round for SIMON-128), it is beneficial to use checkpointing even for partially unrolled designs. Tab. 3 shows the energy per encryption numbers for different partially unrolled implementations of SIMON-128. Glitching in the baseline design increases with the degree of unrolling and so do the energy savings offered by checkpointing, up to 84% in the fully unrolled case. Checkpointing allows for a deeper unrolling with minimal energy penalty resulting in lower latency. In comparison to the most efficient baseline implementation (4-unrolled,

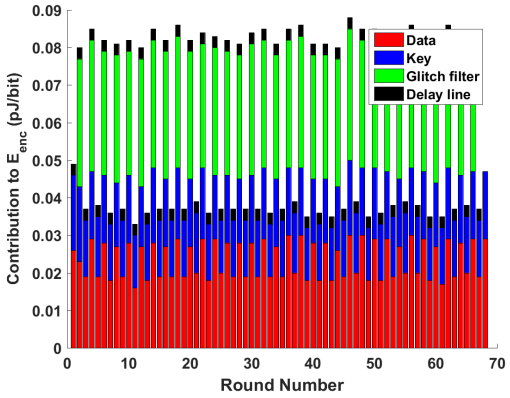


Fig. 8: Energy breakdown of E_{enc} for each round in fully unrolled SIMON-128 in the optimal configuration of checkpointing every second round.

2.89pJ/bit), checkpointing enables 34-unrolled design (3.41pJ/bit) to be competitive in energy at a much lower latency. Fully unrolling helps save loop control energy but incurs leakage cost, leading to less efficient design (4.18pJ/bit).

Also, 1-unrolled baseline consumes more energy than the 2-unrolled and 4-unrolled baselines because the SIMON key expansion function requires storing key_{i-2} in additional registers to compute key_i if no unrolling were done [6]. The frequencies in Tab. 3 are chosen conservatively to account for process variations, but the design could be optimized for performance.

Table 3: SIMON-128 E_{enc} (pJ/bit) comparison between optimal checkpointing and the baseline design for various degrees of unrolling.

	Unrollings	**1**	**2**	**4**	**17**	**34**	**68**
Baseline	E_{enc} (pJ/bit)	3.78	2.95	2.89	6.15	12.43	25.91
	$I_{leak}(\mu A)$	132.58	133.51	169.8	417.4	753.3	1419.8
	Frequency (MHz)	1667	833	417	98	49	25
Checkpointing	E_{enc} (pJ/bit)	–	–	2.92	2.99	3.41	4.18
	$I_{leak}(\mu A)$	–	–	170.4	556.6	1080.2	2016.6
	Frequency (MHz)	–	–	185	73	37	19
Either	Latency (cycles)	68	34	17	4	2	1

4.5 Area Cost of Checkpointing

Using our glitch filtering scheme does incur some area penalty as tabulated in Tab. 4. In terms of number of gate equivalents, we incur a small 3.7% penalty if checkpoints are added after every round in AES-256. In the case of a lightweight block cipher like SIMON-128 that has a very small round function and larger

number of rounds, the penalty is more pronounced. For SIMON, we incur a 44% overhead if checkpoints are placed at the energy-optimal spacing of every second round. It is worth noting that using checkpoints after every round in SIMON-128 would incur a much higher 80% area penalty in addition to not being energy optimal.

With regard to timing, we introduce a small timing penalty because of the introduction of the glitch filters in the critical path and some timing margin to make sure the delay element is sufficiently long so that the enable pulse to a glitch filter arrives after the corresponding round output stabilizes. Though we report conservative frequency numbers in Tab. 3 to account for process variations, we do not have any requirements to double the clock period as in other schemes such as WDDL or Round Gating.

Table 4: Area penalty of proposed glitch filtering scheme in units of gate equivalents. Even in absolute terms, the area cost of checkpointing is significantly higher in SIMON-128 than in AES-256 because the larger number of rounds requires a larger number of checkpoints, even though the checkpoints are only applied at every second round.

	Baseline	**Checkpointing**	**Area overhead**
SIMON-128	56,488	81,321	44.0%
AES-256	342,805	355,630	3.7%

5 Conclusion

In this paper, we have presented an efficient latch-based checkpointing mechanism to reduce the energy per encryption of unrolled block cipher implementations. We demonstrated significant energy savings (28-32%) compared to the best existing scheme for glitch filtering in unrolled block ciphers. Our scheme performs well on block ciphers with simple round functions as in SIMON, and complex round functions as in AES. We also showed that optimal use of glitch filters can lead to further energy savings, resulting in energy consumption that is competitive to a fully serialized implementation while maintaining the latency advantages of an unrolled design. Further, partially unrolled implementations can also greatly benefit from our scheme in scenarios where design constraints limit the degree of unrolling. This technique has applications in improving the efficiency of different unrolled block cipher implementations.

References

1. Ncsu free pdk 45. `http://www.eda.ncsu.edu/wiki/FreePDK45:Contents`.
2. Athas, W. C., Svensson, L. J., Koller, J. G., Tzartzanis, N., and Chou, E. Y.-C. Low-power digital systems based on adiabatic-switching principles. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems 2*, 4 (1994), 398–407.

3. Banik, S., Bogdanov, A., and Regazzoni, F. Exploring energy efficiency of lightweight block ciphers. In *International Conference on Selected Areas in Cryptography* (2015), Springer, pp. 178–194.
4. Banik, S., Bogdanov, A., Regazzoni, F., Isobe, T., Hiwatari, H., and Akishita, T. Round gating for low energy block ciphers. In *2016 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)* (May 2016), pp. 55–60.
5. Batina, L., Das, A., Ege, B., Kavun, E. B., Mentens, N., Paar, C., Verbauwhede, I., and Yalçın, T. Dietary recommendations for lightweight block ciphers: Power, energy and area analysis of recently developed architectures. In *Radio Frequency Identification*. Springer, 2013, pp. 103–112.
6. Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., and Wingers, L. The simon and speck families of lightweight block ciphers. Cryptology ePrint Archive, Report 2013/404, 2013. `http://eprint.iacr.org/2013/404`.
7. Benini, L., Micheli, G. D., Macii, A., Macii, E., Poncino, M., and Scarsi, R. Glitch power minimization by selective gate freezing. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems 8*, 3 (June 2000), 287–298.
8. Boemo, E., Oliver, J. P., and Caffarena, G. Tracking the pipelining-power rule along the fpga technical literature. In *Proceedings of the 10th FPGAworld Conference* (New York, NY, USA, 2013), FPGAworld '13, ACM, pp. 9:1–9:5.
9. Hanson, S., Zhai, B., Bernstein, K., Blaauw, D., Bryant, A., Chang, L., Das, K. K., Haensch, W., Nowak, E. J., and Sylvester, D. M. Ultralow-voltage, minimum-energy cmos. *IBM Journal of Research and Development 50*, 4.5 (2006), 469–490.
10. Hsing, H. Tiny aes project. `opencores.org/project,tiny_aes`.
11. Huda, S., and Anderson, J. Towards pvt-tolerant glitch-free operation in fpgas. In *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays* (New York, NY, USA, 2016), FPGA '16, ACM, pp. 90–99.
12. Karthik, H. S., and Naik, B. M. K. Glitch elimination and optimization of dynamic power dissipation in combinational circuits. In *Advances in Electronics, Computers and Communications (ICAECC), 2014 International Conference on* (Oct 2014), pp. 1–6.
13. Kerckhof, S., Durvaux, F., Hocquet, C., Bol, D., and Standaert, F.-X. *Towards Green Cryptography: A Comparison of Lightweight Ciphers from the Energy Viewpoint*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 390–407.
14. Lamoureux, J., Lemieux, G. G. F., and Wilton, S. J. E. Glitchless: Dynamic power minimization in fpgas through edge alignment and glitch filtering. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems 16*, 11 (Nov 2008), 1521–1534.
15. Monteiro, J., Devadas, S., and Ghosh, A. Retiming sequential circuits for low power. In *Computer-Aided Design, 1993. ICCAD-93. Digest of Technical Papers., 1993 IEEE/ACM International Conference on* (Nov 1993), pp. 398–402.
16. Musoll, E., and Cortadella, J. Low-power array multipliers with transition-retaining barriers. In *Power and Timing Modeling, Optimization and Simulation (PATMOS)* (Oct. 1995), pp. 227–238.
17. Pub, N. F. 197: Advanced encryption standard (aes). *Federal Information Processing Standards Publication 197* (2001), 441–0311.
18. Tiri, K., and Verbauwhede, I. A digital design flow for secure integrated circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 25*, 7 (July 2006), 1197–1208.
19. Wilton, S. J. E., Ang, S.-S., and Luk, W. *The Impact of Pipelining on Energy per Operation in Field-Programmable Gate Arrays*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 719–728.