IET Computers & Digital Techniques

*Research Article*

IET Journals
The Institution of Engineering and Technology

# Mitigating information leakage during critical communication using S*FSM

ISSN 1751-8601
Received on 25th February 2018
Revised 11th January 2019
Accepted on 15th January 2019
E-First on 20th March 2019
doi: 10.1049/iet-cdt.2018.5186
www.ietdl.org

*Mike Borowczak*[1] ✉, *Ranga Vemuri*[2]

[1]*Department of Computer Science, College of Engineering and Applied Science, University of Wyoming, Wyoming, USA*
[2]*Department of Electrical Engineering and Computer Science, College of Engineering and Applied Science, University of Cincinnati, Ohio, USA*
✉ *E-mail: mike.borowczak@uwyo.edu*

**Abstract:** Security-centric components and systems, such as System-on-Chip early-boot communication protocols and ultra-specific lightweight devices, require a departure from minimalist design constructs. The need for built-in protection mechanisms, at all levels of design, is paramount to providing cost-effective, efficient, secure systems. In this work, Securely derived Finite State Machines (S*FSM) and power-aware S*FSM are proposed and studied. Overall results show that to provide an S*FSM, the typical FSM requires a 50% increase in the number of states and a 57% increase in the number of product terms needed to define the state transitions. These increases translate to a minimum encoding space increase of 70%, raising the average encoding length from 4.8 bits to 7.9 bits. When factoring in relaxed structural constraints for power and space mitigation, the respective increases of 53 and 67% raise the average number of bits needed to 7.3 and 7.9. Regarding power savings, current minimisation is possible for both FSMs and S*FSMs through the addition of encoding constraints with average current reductions of 30 and 70%, respectively. Overall, a power-constrained S*FSM consumes about 5% more power than insecure FSMs with binary encodings, though with a penalty of a 95% increase in layout area.

## 1 Introduction

While 30 years of electronic design automation have focused on circuit-level miniaturisation to enhance and achieve low power consumption and high-speed targets, this work proposes that security-centric components and systems, such as System-on-Chip (SoC) early-boot communication protocols and ultra-specific lightweight devices, require a departure from this minimalist mentality. The need for built-in protection mechanisms, at all levels of design, is paramount to providing cost-effective, efficient, secure systems. The focus of this work is the high-level design of secure sequential circuits by reducing the amount of information leakage from the underlying system. In particular, this work aims to secure a finite state machine (FSM) based communication and processing circuits (e.g. secure early-boot, shared-bus communication etc.). This work explores and mitigates two dominant sources of information leakage within FSMs that enable non-invasive, side-channel based attacks.

The typical approaches to FSM synthesis and encoding share one simple property – *minimise whenever possible*. The result of these approaches is minimally-sized FSMs with minimal-length encodings, which result in smaller-sized circuits, registers, and buses that ultimately form smaller circuit footprints that achieve low-power and high-frequency design targets. As a result of these efforts, compact FSMs have become easier to model and reconstruct, even without complete knowledge of the underlying design. This can be of benefit not only to designers but to attackers as well. Specifically, minimal encoding strategies and state reduction techniques allow the current state or transition between states of an FSM to be easily correlated to operational byproducts (e.g. circuit power consumption) that then enable state reconstruction.

This work proposes a two-fold method for the design of 'Secure' FSMs (S*FSMs) that mitigate the high-level information leakage typically exposed through optimised FSMs. The method for securing FSMs targets the underlying FSM structure as well as its encoding. Results demonstrate the effectiveness of security-driven approaches to FSM synthesis in eliminating the information leakage based on the relationship between common hardware-byproduct models and hardware implementations. These results are then supported by the characterisation and security analysis of fully synthesised FSM circuits.

## 2 Attack vectors and prior mitigation

In a world where access is power – authentication and control mechanisms are constantly under attack. From physical access to decoding media, the control/disruption of autonomous systems – attackers probe the weakest link in a network, device, subsystem, or subroutine. Since the authentication schemes used in many critical devices are mathematically and computationally difficult to compromise, attackers now routinely seek out secondary sources of information to extract information from systems. These secondary sources of information can directly attack an authentication scheme (e.g. direct key recovery) or can be used in conjunction with traditional sources of information (e.g. transmitted data) to form more intelligent, computationally easier attacks. A classic example of the latter are power side channel attacks which use energy information in the form of power or electromagnetic radiation to mount extremely powerful attacks on sub-circuit modules – generally of a cryptographic nature.

The protection of hardware devices from side-channel attacks is well documented at low levels of abstraction. A majority of existing low-level solutions focus on methods that reduce intra-cycle current variations through dual-routed logic cells [1–4]. These methods, albeit effective, come at a significant design cost and increased complexity in downstream design and implementation. The existing high-level masking and hiding alternatives tend to be restricted to specific algorithms and require detailed and specific knowledge to implement – making them less than desirable for automating the creation of secure hardware designs from arbitrary functions.

Rather than focus on the protection of a single class of cryptographic algorithms, the focus of this work is on a more general problem space: the realm of FSMs. Since their use in hardware devices is widespread and their significance within a larger device scope varies from minute to critical, the information gleaned from FSMs current state or transition can do everything from reducing the computational complexity of an attack to directly revealing sensitive information to allowing disruption of

its normal operation. Several specific direct uses of FSMs in security critical hardware include anti-lock brake systems, [5] pacemakers [6] and other medical devices [7]. Beyond these computational lightweight devices, FSMs are often used in the development of the communication protocols between reusable IP-blocks and core security and orchestration blocks in Systems-on-Chip [8, 9]. To alleviate this weakness, a high-level method described in [10] eliminates the relationship between FSMs and side channel models. This work applies this method to a subset of FSMs benchmarks to first validate the theoretical result as well as characterise the final design impact.

### 2.1 Attack fundamentals

Side channel attacks are hardware-based attack vectors that exploit the relationships between logical/functional operations of a device and their physical byproducts. In conjunction with some basic information-leakage models and standard device I/O these statically driven attacks can be used to determine information stored within the device.

Consider a target device $T$ that implements some set of functions $F$ with input/output IO and a set of side channels SC. Assume some sensitive information $S$ is processed by a functional subset $f_s$ of $F$, fundamentally, an attack requires, a relationship between at least one side channel and an I/O sensitive functional subset, and a secret and/or I/O dependent model of the side channel. More formally expressed:

i. $\exists(sc \subset \mathrm{SC}) \propto f_s(\mathrm{IO})$;

1. $M(s, IO) \propto sc \propto f_s(\mathrm{IO})$;

One of the most prevalent side channels attacks, due to its accessibility and low hardware cost, targets devices using information embedded within their data-dependent power consumption profile [11]. These attacks typically use Hamming-based models, including the Hamming weight (HW) and the Hamming distance (HD), found in (1) and (2), respectively. These models target a set of bits within a device (e.g. register, bus etc.), either by modelling the total number of bits 'on' or the number of bits that are switching [12]. When unguarded against, side channel attacks are extremely powerful as variations in a hamming model correlate to variations in the power consumed [13].

$$\mathrm{HW}(S[x\cdots 0]) = \sum_{i=0}^{x} S[i] \tag{1}$$

$$\mathrm{HD}(S_i[x\cdots 0], S_{i+1}[x\cdots 0]) = \sum_{b=0}^{x} S_i[b] \oplus S_{i+i}[b] \tag{2}$$

There are two basic requirements to perform side-channel attacks – the first requires a side channel that is related to the secure (sub)function while the second requires a modelled secure sub-function to be proportional to the side channel and intrinsically the underlying (sub)function. Most existing research removes the relationship between the side channel and the underlying data dependent functionality. From a statistics perspective, given a model $m$ and a power side channel $p$, each containing $j$ aligned data-points, the correlation, $C(m, p)$, is defined by the following equation:

$$C(m,p) = \frac{\sum\left((m_j - \overline{m}) \times (p_j - \overline{p})\right)}{\sqrt{\sum (m_j - \overline{m})^2 \times \sum (p_j - \overline{p})^2}} \tag{3}$$

### 2.2 Related research

Current research in side channel security focuses on preventing information leakage due to the physical realisation of design. In most cases, information leakage is mitigated using either low-level design techniques or with methods targeted at specific algorithms and implementations.

The most prevalent side channel solutions focus on low-level implementation methods to reduce intra-cycle current variations, typically through dual-logic cell styles [1–4]. These methods, while effective, come at a significant design cost and increased complexity in downstream design constrains and implementation. In particular, these styles suffer significant penalties either in the cell layout area, routing, delay, and overall power consumption [14]. The alternative to low-level cell manipulation is through high-level masking and hiding of information [15].

Other approaches include special asynchronous designs methods which again pose issues in widespread design automation [16]. Similarly, approaches that are specific to an algorithm, typically deal with masking specific computations – methods not generically or automatically adaptable to any logic circuit [17, 18]. These techniques not only trap designers into looking for patterns rather than solving the underlying problem, but they are both impractical for widespread, generic use and are prey to higher-order side channel attacks.

Low-level techniques used towards the mitigation of power based side channel attacks focus on removing information from the physical side channel. The general principle consists of removing the inter-cycle variation from within a (sub)circuit due to data-dependent inputs. A majority of secure CMOS logic styles follow two conditions outlined by Tiri *et al.* [19]:

i. Each logic cell must go through exactly one output transition during each cycle (regardless of input); and
ii. The total switched capacitance must always be constant.

When both conditions are satisfied, the result is a power side channel with no cycle-to-cycle variation $(\forall j, p_j = \overline{p})$. The correlation between the model and the modified side channel, as derived in (4)–(6), is indeterminate. In real-world implementations, the second condition of perfectly matching the total switched capacitance is unfeasible, especially post-layout, and leads to slight variations in data-dependent power consumption

$$C(m,p) = \frac{\sum\left((m_j - \overline{m}) \times (\overline{p} - \overline{p})\right)}{\sqrt{\sum (m_j - \overline{m})^2 \times \sum (\overline{p} - \overline{p})^2}} \tag{4}$$

$$= \frac{\sum\left((m_j - \overline{m}) \times (0)\right)}{\sqrt{\sum (m_j - \overline{m})^2 \times (0)}} \tag{5}$$

$$= \frac{0}{\sqrt{0}} \;\; Indeterminate\,. \tag{6}$$

To implement a high-level solution, a similar strategy is employed within the side channel model space. The objective is to minimise and eliminate leakage from and information theory perspective. Specifically, the objective is that for each side channel model all target data points are constant, or equal to the average of the entire set $(\forall j, m_j = \overline{m})$. As such, correlation between a variable side channel and constant, static, model will be indeterminate.

### 2.3 Minimal FSMs and information leakage

FSMs represent a computational model typically used heavily in the design of circuit level sequential hardware and computing programs found in everything from locking mechanisms and vending machines to communication protocols in critical infrastructure and warfare systems. While the security requirements and implications of a vending machine are hardly comparable to that of a UAV to a UAV communication protocol, the motivation for S*FSMs leverages a rather traditional FSM for the sake of clarity. Imagine the classical computer architecture problem of designing a branch predictor. Fig. 1 shows a branch predicting FSM implementation that utilised a 2-bit saturating counter with each state of the counter assigned a minimal binary encoding.

The implementation is a generic four-state Moore machine: if the previous branch was taken ($T = 1$), the branch-predictor moves

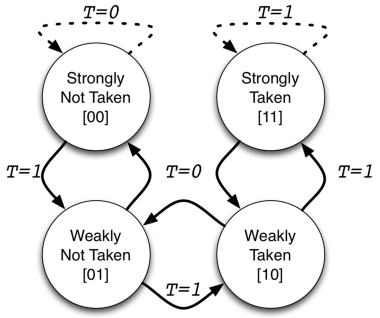


**Fig. 1** *FSM for a two-stage branch predictor*

**Table 1** Annotated transition table for the FSM in Fig. 1

| $S_{current}$ | $S_{next}$ | $T$ | HD | HW[$S_{current}$] |
|---|---|---|---|---|
| ST | ST | 1 | 0 | 2 |
| ST | WT | 0 | 1 | — |
| WT | ST | 1 | 1 | 1 |
| WT | WNT | 0 | 2 | — |
| WNT | WT | 1 | 2 | 1 |
| WNT | SNT | 0 | 1 | — |
| SNT | WNT | 1 | 1 | 0 |
| SNT | SNT | 0 | 0 | — |

towards (or remains in) the Strongly Taken (*ST*) state, otherwise when the previous branch is not taken ($T = 0$) the predictor moves towards the Strongly Not Taken (*SNT*) state. The two intermediate states, Weakly Taken and Weakly Not Taken (*WT* and *WNT*), aide in providing an increased prediction history or memory, greatly increasing its accuracy [20, 21].

The minimally encoded two-stage branch predictor shown in Fig. 1 is summarised in the extended transition table (Table 1) which shows both the HW and HD ranging from 0 to 2. The variability of the HW and HD is unavoidable, regardless of the minimal binary encoding applied. A secure FSM (SFSM) strategy is required. Notice that if the HW is known, the current state is known exactly in two instances (ST, SNT), in the other case, when HW = 1, the search space of potential states is reduced from four state to two (WT, WNT). Capturing this information overtime could enable complete reconstruction of the FSM through observation of HW/HD alone.

### 2.4 Contributions

A significant amount of work has attempted to remove the variability of underlying physical circuits. While possible in simulation, this is non-trivial when dealing with mass-produced devices since even perfectly lay out and routed designs are subject to intra-die process variations [22–24].

This work mitigates the variability of the side channel models themselves. This approach follows the traditional top-down architecture dogma that design choices at high-levels are more efficient and broadly realised than those implemented at the low level of implementation [25]. While greater efficiency in no way implies greater security, the high-level methods presented in this work act as a first pass solution that any designer can implement to gain increased security while balancing overhead that is, on average, better than low-level methods. Increased security, in a top-down approach, can be achieved by introducing security mechanisms at each design stage until the designs security targets are met.

Within the minimalist spectrum, FSM synthesis and encoding has been explored extensively. The current research focus in FSM synthesis and encoding lies in low-power applications [26–28] as well as some use as protection methods against fault injection-based attacks [29–31]. To address these power dominating requirements this work also introduces power as a competing design objective by relaxing and constraining specific security parameters.

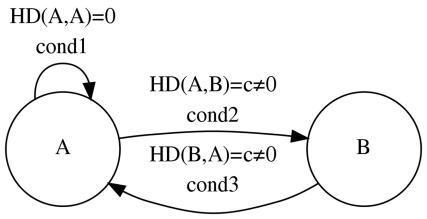


**Fig. 2** *No encoding of these two states can satisfy the HD requirement*

The major contribution of this work includes: (i) the elimination of variation within the side channel models of FSMs through a high-level method to protect any FSM based devices against power side channel attack, and (ii) an approach to satisfy the competing design objective found in the design and synthesis of low-power SFSMs.

## 3 S*FSMs

To define S*FSMs, it is helpful to begin with the basic FSMs which can be explicitly defined as the quintuple $(\Sigma, S, s_0, \delta, F)$, where

- $\Sigma$ is the finite, non-empty, set of symbols.
- $S$ is the finite, non-empty, state space.
- $s_0 \in S$ is the initial state.
- $\delta: S \times \Sigma \rightarrow S$ is the state-transition function.
- $F$ is the set of final states.

An S*FSM is defined as one which eliminates the relationship between side channel models and the internal states and transitions. Two high-level conditions are required and sufficient to secure an FSM against traditional HW and HD side channel models. Specifically, an S*FSM must first eliminate the relationship between the current state and HW models. Secondly, an S*FSM must eliminate the relationship between transitions and HD models. The elimination of the relationship is accomplished by imposing the following restrictions on the state encoding within the S*FSM. First, as shown in (7), all states within an S*FSM must have the same constant HW. Second, as shown in (8), the transitions between connected states ($s$ and $s'$) must have the same HD. This second property has the effect of creating uniform HD between all interconnected states within an S*FSM

$$\forall s \in S: \mathrm{HW}(s) = c_1 \tag{7}$$

$$\forall s, s' \in S \,|\, \exists \alpha \in \Sigma: s' = \delta(s, \alpha) \rightarrow \mathrm{HD}(s, s') = c_2 \tag{8}$$

The first condition, which requires that each state within an S*FSMs has a constant HW, can be readily achieved, though not optimally with existing methods. The second condition, however, which requires that each state transition has a constant HD requires structural modification as well as non-trivial state assignments. Thus, to create S*FSMs, we propose a two-part FSM hardening process which includes (i) structural modifications to physical topology and (ii) the use of intelligent encoding schemes.

### 3.1 Structural

A two-state FSM can be used to motivate the need for structural modifications of S*FSMs. Consider the two unique states, $A$ and $B$, as shown in Fig. 2. When in state $A$, the transitions to $A$ and $B$ result in two unique HDs – regardless of the strategy used to encode the states [10]. The self-loop transition always results in a $\mathrm{HD}(A, A) = 0$ while the other transition results in $\mathrm{HD}(A, B) \neq 0$. For example, when using one hot encoding $\{A = 01, B = 10\}$ the HD = 2, while using a minimal binary encoding $\{A = 0, B = 1\}$ the HD = 1. To eliminate the relationship between HD and an FSM, an S*FSM cannot contain any self-loops.

The need for this structural modification in S*FSM is further demonstrated when considering the need for a constant HD between all state transitions. Consider first a single-node FSM with

```
1: procedure LOOPREMOVE(V, E)
2:     for each v ∈ V do
3:         if ∃e : e(v, v, t) ∈ E then          ▷ Self-loop w/cond. t
4:             E ← E − e
5:             v′ ← v                            ▷ Create new node
6:             V ← V ∪ {v′}
7:             E ← E ∪ {{v, v′, c}, {v′, v, c}}
8:             for each u ∈ V|u ≠ v, e(v, u, c) ∈ E do
9:                 E ← E ∪ e(v′, u, c)
10:            end for
11:        end if
12:    end for
13: end procedure
```

**Fig. 3** *Algorithm 1: FSM loop unroll*

one self-looping transition. Albeit not useful, regardless of its encoding, it will always have a constant HW as well as an HD of zero. It satisfies both conditions needed for an SFSM. On the other hand, the two-node FSM in 2 can never satisfy the second condition. Regardless of encoding selected for states A and B, the HD between two unique states can never be zero, while the HD between any node and itself is always zero.

The only solution capable of eliminating this conflict is the unrolling of self-loops – essentially the opposite of state collapsing, which is often used to reduce FSM complexity [32]. Thus, to satisfy the second condition, *a multi-state, side-channel hardened, FSM cannot contain any self-loops*. Algorithm 1 (see Fig. 3) demonstrates a straightforward, yet effective method of removing self-loops for all FSMs. To achieve this goal, it takes each node, checks for a self-loop, and upon finding one, remove it from the transition list. A new state is added with two corresponding edges, each with the same transition condition(s). Finally, all out-going edges are replicated, maintaining the original functionality of the FSM. Without loss of generality, we assume that self-loops should not be present in any hardened S*FSM. Applying this algorithm to the FSM in Fig. 1, the resulting restructured FSM can be seen in Fig. 4.

### 3.2 Encoding

Assuming structural modification has been applied to a design, there is already an existing encoding strategy that satisfies both conditions for S*FSMs. One-hot encodings have a constant HW, specifically HW = 1. Trivially, the HD between two states is also constant, HD = 2. Unfortunately, their ease of use is quickly overshadowed by the amount of overhead required to encode a complete FSM. An $n-state$ FSM requires an $n-bit$ one-hot encoding, where each possible state is encoded using one exclusive bit. This linear increase in encoding length limits one-hot to all but small FSMs. While other common encoding strategies exist (e.g. Grey, minimal binary), they fail one or both of the needed S*FSM conditions in most practical examples.

To quantify the maximum number of states ($S_{\max}$) or a maximum number of transitions ($T_{\max}$) that can be expressed given specific encoding lengths ($n$), our primary secure encoding constraint must be taken into account. That constraint requires that every S*FSM state must have a constant HW ($c$). Simply stated, the total number of states ($s$) that can be represented by an $n$-bit encoding where $c$-bits must be on in every encoding, is defined as $\binom{n}{c}$. Stated another way, inequality of (9) must be satisfied. Several other key relationships bound the number of states and transitions that can be achieved given specific encoding-lengths ($n$) and choices of 'on-bits' ($c$). Of particular interest, shown in (11), are the maximum number possible states ($S_{\max}$) that could be represented given an encoding length of $n$. Additionally, as shown in (12), the selection of $c$ with respect to $n$ should be based on the maximum number of transitions out of any states within the FSM ($T_{\max}$). For example, given a $c = 1$ in one-hot encodings, the number of bits required to in a secure encoding must be $n \geq s$. Optimally, a one-hot encoding ($c = 1$) uses exactly $n = s$ bits to encode an $s$-state FSM, this

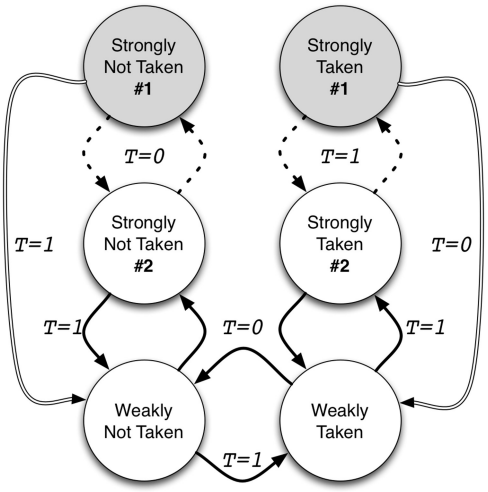


**Fig. 4** *Original two-stage branch predictor after application of the structural modification algorithm in Algorithm 1 (Fig. 3)*

enables that the maximum number of transitions from any one-hot encoded state is $n-1$

$$\binom{n}{c} \geq s \tag{9}$$

$$c \leq \left\lfloor \frac{n}{2} \right\rfloor \tag{10}$$

$$\binom{n}{\left\lfloor \frac{n}{2} \right\rfloor} = S_{\mathrm{MAX}} \tag{11}$$

$$c \times (n-c) \geq T_{\mathrm{MAX}} \tag{12}$$

While a one-hot encoding defined by $n = s$ and $c = 1$ readily satisfies both SFSM and S*FSM constraints, a more optimal secure encoding length in typical S*FSMs is choosing $c = \lfloor n/2 \rfloor$. This secure encoding length (S*Opt) maximises the allowable state space and the number of internal transitions while minimising the encoding length. In particular, S*Opt encodings allow for at most $\lfloor (n^2/4) \rfloor$ connections from any given state ($T_{\mathrm{MAX}}$). In cases, where the S*Opt does not satisfy the $T_{\mathrm{MAX}}$ requirement an increase in $n$ is required (while a reduction of $c$ alone would be counter-productive).

The main challenge in deriving a minimal secure encoding is minimising the bits required while maximising the number of transitions that can be made from a single state. To formally define the problem, assume machine with $s$ states. Clearly, to satisfy the first condition of S*FSMs, all state encodings should have the same number of bits 'on'. Thus, given an $n$-bit encoding ($n \leq s$), each of the $s$ states should have $c$ bits on and $n-c$ bits off. Selection of $n$ and $c$ must satisfy (9) to insure all the states can be represented uniquely. Furthermore, to minimise power consumption $c$ should satisfy (10). Finally, to satisfy the second security constraint while minimising the HD between states (e.g. min bound = 2), (12) must be satisfied.

## 4 Power-constrained S*FSMs

Often, low power designs rely on heavy minimisation and optimisation procedures while many secure designs use low-level duplication mechanisms to thwart attacks. This section addresses the trade-offs between low-power and SFSM by proposing a flexible, secure, encoding strategy which, in conjunction with security-based structural modifications, can provide low-power implementations that reduce information leakage. The secure encoding strategy includes methods that relax and tighten the original constraints to provide varying levels of protection that approach traditional low-power encoding methods.

**Table 2** Variability of HW/HD, given an FSM type original, S*FSMs

| FSM.Enc | HW | HD |
|---|---|---|
| O.HDR | Var | $c_2$ and 0 |
| O.HDR + OH | $c_1$ | $c_2$ and 0 |
| S*.CHD = 4 | $c_1$ | 4 |
| S*.CHD = 2 | $c_1$ | 2 |
| S*.CHD = 2, CHW = 1 | 1 | 2 |

While secure state encodings are a new design parameter, extensive work has already been done in the area of low-power/power aware state encodings [33–38]. In particular, most solutions focus on the minimisation of switching activity (by reducing HD between connected states) using a variety of techniques ranging from genetic local search, to integer linear programming, to SAT-based algorithms. This work builds upon some of these methods, specifically those that target peak current minimisation ($N_{\text{peak}}$) as their objective [35, 36].

This section extends the set of high-level solutions for reducing data-dependent current variations during FSM state transitions introduced in the previous section. This work provides methods to relax the security constraints, thereby enabling S*FSM encodings that approach existing low-power solutions.

To provide designers with flexibility and low-power options in their designs, we provide two methods, which, when used in conjunction mimic existing low-power solutions. The first method increases flexibility, at a potentially significant cost to security – involves reverting to FSMs that contain self-loops. In certain scenarios, when many states have self-loops, detecting a self-loop does not provide as much information *if* other transitions are not identifiable. To maintain the transition masking – the original HD constraint must be relaxed slightly (HDRelaxed, HDR) as shown in (13). This modified HDR constraint still forces all non-looping transitions to be a constant HD but is achievable in non-restructured FSMs.

The second method further restricts the original HW and HD/HDR constraints to minimise the peak power consumed to satisfy low-power design objectives. Following previously established work [35, 36] this translates to minimising the maximum number of $0 \rightarrow 1$ and $1 \rightarrow 0$ transitions ($N_{\text{peak}}$). Equations (14) and (15), formally define $N_{\text{peak}}$ with unique and identical technology dependent weighting factors $w_1, w_2$, respectively. While previous works were also concerned with switching activity [36] (16), our current discussion relaxes this constraint in favour of our core security objectives. The minimisation of $N_{\text{peak}}$ is accomplished by forcing additional restrictions on HD/HDR – mainly minimising $c_2$ in (13), respectively,

$$\text{HDR} = \text{HD}(s_i, s_j) = c_2 \quad \forall i, j \,|\, i \neq j \tag{13}$$

$$N_{\text{peak}} = \max\{w_1 \cdot N_{0\rightarrow 1}, w_2 \cdot N_{1\rightarrow 0}\} \tag{14}$$

$$N_{\text{peak}} = \max\{N_{0\rightarrow 1}, N_{1\rightarrow 0}\} \tag{15}$$

$$\text{SW}_{\text{tot}} = \sum_{s_i \rightarrow s_j} \text{HD}(s_i, s_j) \cdot p_{i,j} \tag{16}$$

If the primary HW and HD constraints are maintained, then assuming two states, $s_1, s_2$, each $0 \rightarrow 1$ or $1 \rightarrow 0$ in $s_1$ forces a respective $1 \rightarrow 0$ or $0 \rightarrow 1$ transition in $s_2$. Thus $N_{\text{peak\_secure}} = (c_2/2)$ as shown in (17). When guaranteeing either the HD or HDR constraint, $N_{\text{peak\_secure}}$ is equal to half of the HD between any two connected states. Minimisation of the HD/HDR constraint constant $c_2$ will also minimise $N_{\text{peak\_secure}}$. For example, a one-hot encoding will have a $c_2 = 2$ and an $N_{\text{peak\_secure}} = 1$ – the lower bound of $N_{\text{peak}}$. While not formally expressed, similar constraint restrictions can be placed on the HW constant $c_1$ to mitigate static power consumption

$$N_{\text{peak\_secure}} = N_{0\rightarrow 1} = N_{1\rightarrow 0} = \frac{c_2}{2} \tag{17}$$

Both of these methods result in new families of encoding strategies, of which two subsets will be explored here – original FSMs with HDR and S*FSMs with power-aware constraint modifications (CHD and CHW) as shown in Table 2. While it is possible to further constrain the original FSM using a CHW that discussion is beyond the scope of this work.

In summary, to prevent any information leakage from the HW and HD models, an S*FSM is required. While the structural requirement can be relaxed (assuming the HDR constraint is used) an FSM with self-loops will always have variability in the HD model. Further, it possible to restrict the HW/HD constants (CHW/CHD) to minimise the peak current ($N_{\text{peak}}$) – in the secure case – minimisation of $N_{\text{secure\_peak}}$ is directly proportional to minimisation of the HD constraint constant $c_2$.

## 5 Implementation and analysis

The experimental setup for this work consisted of two main phases: a characterisation phase and a security analysis phase. Preliminary characterisation results were done using over a 150 BenGen FSM benchmark [39], followed by more robust real-world FSMs from the MCNC Benchmark Suite [40] and supplemented by our own independently created a benchmark suite. The second phase, dealing with security analysis, while relatively straight forward at low levels due to use of correlation as a metric – became increasing complex and uninformative – due to the nature of the data being used for correlation. This motivates a different mechanism to compare and the security of our solutions – specifically one grounded in mutual information (MI) theory.

### 5.1 Characterisation

The characterisation of the S*FSM was broken down into two classes – for each benchmark, there is a standard un-modified version (FSM) as well as a structurally secure, loop-unrolled version (S*FSM). A first pass comparison between the two classes using the state and transitions requirements for both FSMs essentially reveals a predicable state space and transition increase relative to the frequency of self-loops times the original number of states.

Next, to characterise the bit-encoding requirement the theorem prover generator was tuned to ignore all security constructs and satisfy a single constraint – each state must have a unique encoding. The theorem prover, without knowledge of the constraint, or a theoretical lower bound, started at a predetermined upper-bound reducing the number of bits until it could no longer satisfy the single constraint. As expected, the theorem prover returned a minimal binary encoding for both the FSM and S*FSM. Using the same solver, with all security constraints in place (constant HW and HD) the minimal secure encoding was derived for the SFSM.

To move beyond theoretical requirements, each FSM and SFSM was paired with an associated encoding. These pairings were then converted and synthesised using a standard cell library in 90 nm technology to calculate area requirements as well as run low-level Nanosim power simulations for the next phase of the experiment.

### 5.2 Security analysis phase

To evaluate security, the relationship between the power side channel and hamming models had to be analysed, quantified and ranked. In previous, low-level work many authors claimed that reduced variation in the power signal should reduce correlation – since variation still existed in the signal correlation metric was still defined – though questionable in its true meaning. Using correlation as a security metric involves making some strong assumptions (e.g. linearity between side channel and model, non-sparse data, non-uniform vectors, correlation and causation). Considering these parameters, a more robust approach using MI theory was used. Consider two datasets $A$ and $B$ – MI quantifies

**Table 3** Correlation between HW/HD model and state/transitions for three encoding strategies on standard and S*FSMs

| | No. of cycles | Standard FSM | | S*FSM | | |
|---|---|---|---|---|---|---|
| | | BE | OH | BE | OH | S*O |
| state, HW | 10 | 0.71 | — | 0.45 | — | — |
| | 500 | 0.62 | — | 0.59 | — | — |
| | 5000 | 0.63 | — | 0.59 | — | — |
| | 50,000 | 0.63 | — | 0.60 | — | — |
| tran, HD | 10 | 0.61 | 0.61 | 0.01 | — | — |
| | 500 | 0.17 | 0.17 | 0.30 | — | — |
| | 5000 | 0.17 | 0.17 | 0.32 | — | — |
| | 50,000 | 0.18 | 0.18 | 0.32 | — | — |

how much entropy can be removed from a set A by knowing set B (or from set A if B is known).

$$E(A) = H(A) = -\sum_{a \in A} p(a)\log_2 p(a) \tag{18}$$

$$\begin{aligned} I(A;B) &\equiv I[p(A,B)] \\ &\equiv \sum_{a,b} p(a,b)\log_2\left(\frac{p(a,b)}{\sum_b p(a,b)\sum_a p(a,b)}\right) \end{aligned} \tag{19}$$

$$\begin{aligned} I(A,B)|_{\forall_{b \in B} b = \bar{b}} &= \sum_{a,b} p(a,b)\log_2\left(\frac{p(a,b)}{\sum_b p(a,b)\sum_a p(a,b)}\right) \\ &= \sum_{a,b} p(a,b)\log_2(1) = 0 \end{aligned} \tag{20}$$

### 5.3 Characterisation using theoretical simulation

This section experimentally validates the two-part method proposed for S*FSMs by demonstrating the variability of two Hamming models on two different FSMs. The first is the unsecured FSM, while the second is the unrolled structurally modified FSM. Each FSM was encoded with a minimal encoding (BE) derived by the satisfiability solver as well as secure encoding methods when they are feasible – one hot (OH) and/or S*Opt (S*O).

The experiment targets the worst-case variability of the side channel models: if the model itself shows no variability, it is rendered useless. To measure the variability of both the HW and HD models, a simple event-driven FSM simulator was constructed. The high-level simulator applied random transition vectors (ranging in size from 10 to 50,000 transitions) to a given FSM. During each transition, the current state and transition path are recorded along with computed HW and HD values. While a simplification concerning a real hardware system, the goal is to determine whether a relationship between the two recorded events (state and transition) and the models (HW and HD) exists.

Table 3 summarises the simulation-based maximum theoretical correlation ($Cr_{Mtheory}$) for the standard FSM as well as the structurally secure S*FSM for three unique encoding schemes including BE, OH and S*O. The $Cr_{Mtheory}$ values reported here are average over 100 unique simulation runs at given cycle lengths. The unique set of 100 input vectors is shared across each row to eliminate random bias.

The results highlight first and foremost that a proper encoding strategy is key: FSMs, regardless of structure, are insecure when encoded using typical binary encoding schemes. Secondly, the only way to eliminate the variability of both the HD and HW models is through a combined use of a secure encoding scheme (S*O or OH) and structural modification. To quantify the relative security FSMs will be fully implemented and the MI between side channel and models computed.

### 5.4 Physical realisation of S*FSM

The practicality and strength of S*FSMs are tested and verified with gate level, power-accurate implementations and realisation of logic circuits to insure that the theoretical justifications hold. While the ultimate test is with physically realised designs this is both impractical and expensive at the current time. S*FSM validation is accomplished using an analogue electronic circuit simulator to show the relationship between the theoretical maximum correlation ($r_{Mtheory}$) results presented earlier and the worst-case correlation to a simulated power side channel ($r_{Spower}$). For this discussion, $r_{Spower}$ is computed by correlating the power side channel against two different data sources: an 'Oracle' ($r_{Spower}[O]$) and an attacker best-case model ($r_{Spower}[M]$). The final security metric for any FSM consist of the MI between $r_{Spower}$ and $r_{Spower}[M]$ or more simply MI($P, M$) for all models $M$ (HW and HD).

- $r_{Spower}[O]$ is computed by capturing the actual state or transition of the FSM under attack (worst-case scenario).
- $r_{Spower}[M]$ is computed by using the Oracle state information to perfectly capture the side channel models.

### 5.5 Implemented flow

The objective of this work is to quantify, using real-world FSM benchmarks, the effectiveness and cost of implementing S*FSMs in hardware. Our experimental platform uses a collection of over 150 FSM benchmarks, generated, and acquired from the authors of BenGen [39] and MCNC [40], ranging in size from 4 to 60 states, with total transition ranging from 8 to 216. The implemented workflow begins with original benchmarks, covering the extreme ranges of both the state and transition space. These benchmarks are then converted to structurally secured versions with near S*Opt encodings. Both sets of FSM, the original BenGen benchmarks as well as the S*FSMs are converted to Verilog where they can be synthesised at the gate-level using DC compiler. The resulting gate-level netlist is converter to Spice to gain cycle-accurate power information using Nanosim. The Nanosim data in conjunction with, worst case, FSM Oracle data is used to measure the correlation between the Hamming models and the current FSM state or transition between states.

Fig. 5 depicts the low-level FSM synthesis flow used to generate the correlation data $r_{Spower}$. The bold solid path represents the complete S*FSM path, while the relaxed dashed lines represent naive FSMs (either in structure and/or encoding). Finally, the compressed dashed-lines represent FSM specific information used in conjunction with Oracle data to determine attack best-case correlations.

The current process begins with a high-level structural description of an FSM. (a) Structurally secure implementations follow the right branch (b) at which point the flow can rejoin the naive, left branch (c) which only enables standard encoding options or continue to secure encoding styles (d). The information from either (c) or (d) is then passed to an existing high-level to low-level compiler (e). The resulting low-level circuit is simulated using a low-level simulator (f) which uses stimulus data (generated during the theoretical analysis) to create power-accurate traces. The stimulus generator (h) also computes needed oracle and best-case attack models which are all used to determine the maximum $r_{Spower}$.

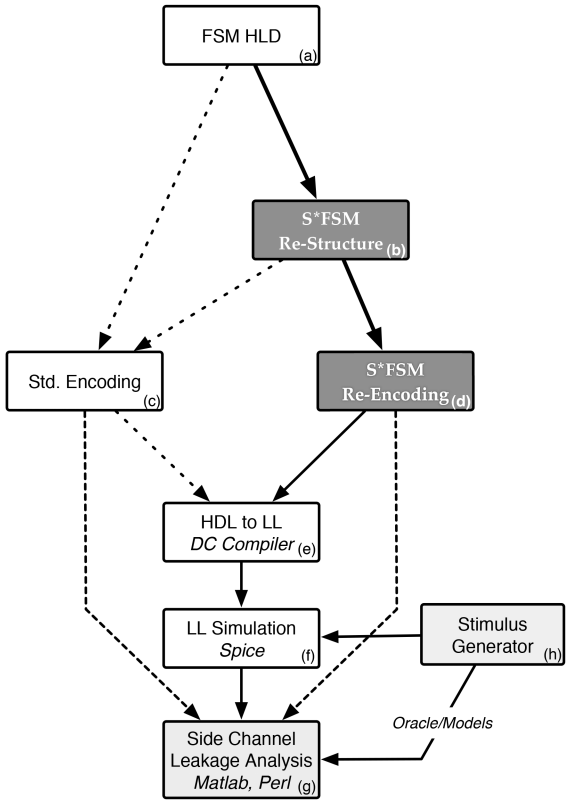


**Fig. 5** *FSM flow to test and verify theoretical results using gate-level realisation of FSMs and S*FSMs using multiple encodings*

## 6 Results

### 6.1 S*FSMs

Results from our experiment fall into two general classes. The first set deals with the characterisation results which contain the theoretical costs based entirely on the number of FSM states and transitions as well as the implementation cost defined by the synthesised hardware layout of the original and secured FSMs assuming using the Synopsys 90 nm standard cell library. The second set of results focuses on the security, regarding shared information, afforded by the particular FSM + encoding strategy selected.

### 6.2 Characterisation results

The characterisation results are broken down into the following four sections – the first includes the physical topological changes that occur due to FSM restructuring though similar information is consolidated and easily synthesised within the second sections which details the state space and transition requirements of the original and restructured FSM. The third part of the characterisation reports the effects of encoding strategy on the bitlength requirements while the final section on the layout space takes into consideration state space and encoding requirements in synthesised circuits.

*6.2.1 State-space requirements:* While only the average state space increase is shown in Table 4, the change in S*FSMs state space for the BenGen benchmarks ranged between 14 and 100%, the maximum possible when each state contains a self-loop. Next, the average transition counts for BenGen, as shown in Table 5, increase drastically for most non-trivial FSMs and range from a 25–100% increase. Though the overall average increase is 77%, more complex FSMs had an average increase closer to 93%. While the BENGEN and our custom FSMs benchmarks are purely synthetic – the MCNC benchmark suite represents a more realistic picture of typical FSMs and therefore the true impact on the state-space requirements. As such a large majority of the future results are dominated by the MCNC benchmarks.

**Table 4** Average number of states in BenGen and MCNC

| | BenGen ($n = 7$) | MCNC ($n = 10$) |
|---|---|---|
| FSM | 36.3 | 31.8 |
| S*FSM | 68.7 | 42.6 |
| increase, % | 71.6 | 58 |

**Table 5** Average number of transitions in BenGen and MCNC

| | BenGen ($n = 7$) | MCNC ($n = 10$) |
|---|---|---|
| FSM | 119 | 103.2 |
| S*FSM | 231 | 161.3 |
| increase, % | 77 | 68 |

**Table 6** Average encoding length for BenGen and MCNC

| | BenGen ($n = 7$) | MCNC ($n = 10$) |
|---|---|---|
| binary encode FSM | 4.6 | 4.8 |
| binary encode S*FSM | 5.3 | 5.6 |
| S*opt encode S*FSM | 7.9 | 8.1 |
| overall increase, % | 79 | 73 |

**Table 7** Average layout area ($nm^2$) in 90 nm SAED

| | BenGen ($n = 7$) | MCNC ($n = 5$) |
|---|---|---|
| FSM | 14.6K | 2.9K |
| S*FSM | 25.2K | 5.4K |
| overall increase, % | 96% | 85% |

*6.2.2 Bitlength encoding requirements:* To demonstrate the effectiveness of the S*Opt encoding, the number of bits needed for the secure encoding, along with the increase over the original FSMs binary encoding is presented in Table 6. Note that on the average the increase for the BenGen benchmarks is around 79% though for all but the smallest FSM the average is a 67% increase. As with the state space and transition requirements – the MCNC benchmarks provide a more accurate baseline for bit length encoding impact in real FSMs. The difference between benchmark suites is less noticeable simply due to the logarithmic nature of bit encoding requirements.

*6.2.3 Layout space requirements:* To validate the S*FSM approach, the original and secure designs were automatically synthesised without any optimisations using the Synopsys 90 nm standard cell library. The resulting layout areas are summarised in Table 7. Note that while the range varied from a 50% increase to 160% (for the smallest circuit due to overhead) the average increase, even without optimisations, is 4% smaller than 1-to-1 low-level duplication methods. Extending the analysis to include a larger subset ($n = 25$) of the original benchmarks shows an average total area increase closer to 75%. As with the previous metric – the MCNC benchmarks provide a more accurate baseline for the layout area in real FSMs. Overall increases in the area, when including small FSM outliers, tend towards a 100% layout increase – without these the average tends closer to a 65% increase.

### 6.3 Low-power S*FSMS

The characterisation of low-power S*FSMs focuses on: bit encoding, layout area, and now power. The results then focus on security aspects of power constrained solutions. A subset of the common MCNC [40] benchmarks is used for alignment with existing FSM restructuring and encoding solutions.

To characterise solutions, three different aspects are considered: the bit encoding requirement, the physical layout requirements and finally the power requirements. State space elements are not considered as the two classes of FSMs – baseline and restructured – remain unmodified from previous discussions.

**Table 8** Bits required to encode the original benchmark using a binary encoding (OBE), the S*FSM using a binary encoding (SBE) and the minimum number needed using our S*Opt method (S*Opt)

| Benchmark | OBE | SBE | SBE % Inc. OBE | S*Opt | S*Opt % Inc. OBE |
|---|---|---|---|---|---|
| bbara | 4 | 5 | 25% | 8 | 100% |
| bbsse | 4 | 5 | 25% | 8 | 100% |
| dk16 | 5 | 5 | 0% | 8 | 60% |
| dk512 | 4 | 4 | 0% | 6 | 50% |
| ex1 | 5 | 6 | 20% | 9 | 80% |
| modulo12 | 4 | 5 | 25% | 7 | 75% |
| opus | 4 | 5 | 25% | 7 | 75% |
| planet | 6 | 6 | 0% | 8 | 33% |
| sand | 5 | 6 | 20% | 8 | 60% |
| scf | 7 | 7 | 0% | 9 | 29% |
| sse | 4 | 5 | 25% | 8 | 100% |
| styr | 5 | 6 | 20% | 9 | 80% |
| *avg* | *4.75* | *5.4* | *15%* | *7.9* | *70%* |

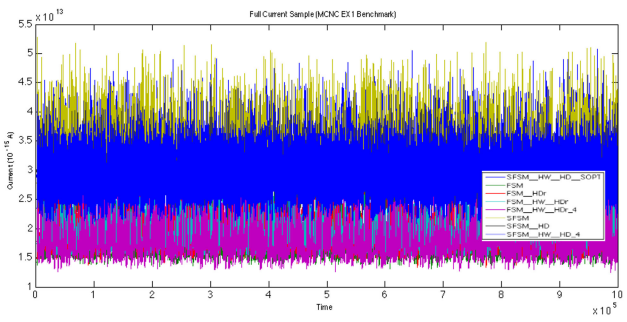


**Fig. 6** *Current ($10^{-15}$ A) required by eight FSMs over a 1K inputs*

*6.3.1 Bit encoding requirements:* The bits required for a binary encoding of the original (O.Binary) and structurally modified FSMs (S.Binary) as well as the bits required for a secure optimal encoding (SS.Opt) were theoretically derived and experimentally validated, while the optimal secure encodings were generated using an SMT solver. Using the minimum binary encoding for the original FSM (O.Binary) as a baseline we found that a binary encoding for the restructured S*FSM (S.Binary) requires a 15% increase in the number of encoding bits. Similarly, a bitlength-optimal, secure encoding of the S*FSM (SS.Opt) requires a 70% increase over the original FSM or a 47% increase over the binary-encoded version of the restructured S*FSM.

Three power-constrained versions of S*FSMs were compared to evaluate the burden on the encoding requirement. The baseline, in which CHD = 4, also happens to line up with S.Opt for this subset of MCNC benchmarks. While further constraining the HD (CHD = 2) more than doubles the required bitlength and requires on average a 112% increase in the number of bits, it does reduce $N_{peak}$ to it's minimum – 1. Further constraining the encoding by imposing a constrained HW (CHW = 1), forces the S*FSM to be encoded using a one-hot encoding that significantly impacts the encoding bitlength by increasing it almost 400%. We avoid this implementation in future discussions due to the limited practicality in anything but simple controllers.

Using the previous O.Binary and SS.Opt as lower and upper bounds, respectively, we applied the relaxed HD (O.RHD) and relaxed HD with HW (O.RHD + HW) constraints on the original FSM. Since the O.RHD allows for variable HW, it is safely considered the weakest of the available modifications though it still requires a 30% average increase in encoding length over O.Binary (see Table 8). When adding the constant HW requirement, the average increase is nearly 55% over O.Binary. In other words, the O.RHD + HW encoding is a 9% decrease over SS.Opt at the cost of exposing any self-loop transitions.

*6.3.2 Layout requirements:* The area requirements for both baseline MCNC benchmarks and restructured MCNC benchmarks show the same linear increase in the required layout area as additional constraints are added. Generally, the greatest increase in area is due to the addition of the HD constraint – within the baseline FSM (15% increase), this corresponds to additional encoding bits, whereas the SFSM sees a greater increase (20%) due to both additional unrolled states as well as encoding bits. The average maximum increases from baseline to fully constrained (e.g. FSM/SFSM to HW + HD/HDr = 4) are 32 and 27%, respectively.

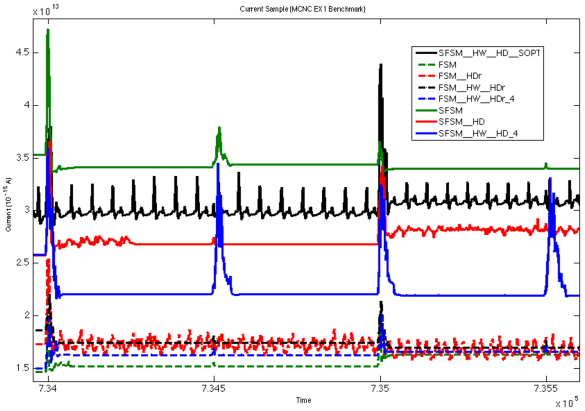


**Fig. 7** *Current ($10^{-15}$ A) required by eight FSMs over four input stimulus*

*6.3.3 Power requirements:* The two classes of FSMs – base FSM and restructured S*FSMs each under four unique encodings all have unique power profiles. Before exploring the aggregated results, all eight current traces for a single 1000 input experimental run of the MCNC EX1 benchmark are shown in Fig. 6. The figure shows the global picture: S*FSMs, in general, have higher current profiles than standard FSMs. Additionally, it is possible to reduce the current draw of both S*FSMs as well as FSMs using special encoding techniques (that reduce the $N_{peak}$).

Focusing on a single benchmark, and observing eight FSM + encoding choices. Fig. 7 depicts the current drawn per round. While some circuits are noisier than others (FSM HW + HDr and S*FSM HD) the average current profiles are well delineated. The S*FSM consume the most power, while the four baseline FSMs consume the least. The constrained S*FSM can reduce the power overhead substantially requiring an overhead of only 50%, while the other secure methods require closer to 100% overhead.

Since the structure and underlying characteristics of the MCNC benchmarks are varied and represent a functional cross-section of FSMs the aggregate, normalised power requirements are of interest to designers interested in fine-tuning power (security and power are explored in the following section). Fig. 8 shows the aggregate normalised power for all of the MCNC benchmarks evaluated. The trends show the following – an FSM constrained in both HW and HD (or HDr) consumes less power, on average than one that is not

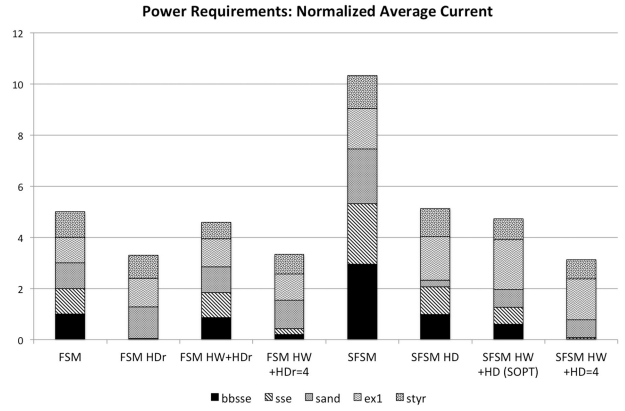


**Fig. 8** *Normalised power requirements of the MCNC benchmarks in the presence of structural and encoding constraints*

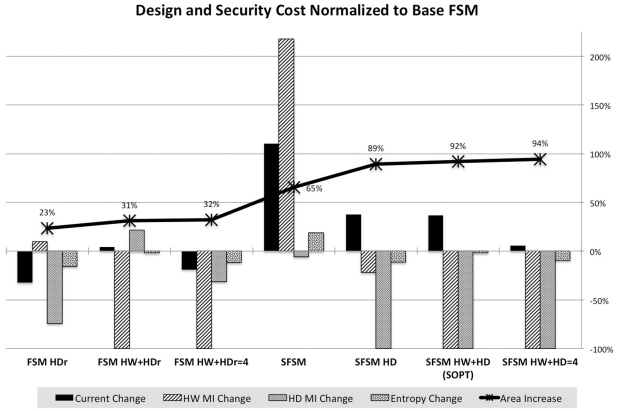


**Fig. 9** *MCNC overall design and security with costs normalised concerning FSMs with binary encodings, ranked in order of area penalty*

**Table 9** Number of bits needed for the S*Opt encoding, the S*Opt encoding when the HD is constrained to 4 when the HW constraint is removed, and the original insecure encoding

| | Bits needed | | | |
|---|---|---|---|---|
| | S*Opt | HW, HD = 4 | HD = 4 | Original |
| avg | 7.9 | 7.9 | 7.1 | 4.8 |
| S*Opt % diff | — | 0% | −12% | −67% |

**Table 10** Bits needed to encode the original, unsecured, FSMs using various HDr constraints to minimise power consumption

| | Bits needed in structurally relaxed (original) FSM | | | | |
|---|---|---|---|---|---|
| | OBE | HW constrained | | Unconstrained HW | |
| | | HDR = 4 | HDR = 2 | HDR = 4 | HDR = 2 |
| avg | 4.8 | 7.3 | 7.9 | 6.6 | 6.5 |
| OBE % increase | — | 53% | 67% | 39% | 37% |

HDR = 4 ($N_{peak} = 2$) and HDR = 2 ($N_{peak} = 1$) require compromises on the number of bits needed to encode while maintaining some level of side channel protection.

**Table 11** Ratio of protected versus unprotected ASIC circuits

| Countermeasure | Area | CLK Freq. | Power |
|---|---|---|---|
| WDDL [2] | 3 | >2 | 3.5 |
| MDLP [41] | 4–5 | 2 | 17 |
| SDMLp [4] | 1.4 | 1.5 | 1.07 |
| CLK-JITTER [42] | 1.25 | 1.6 | 1.18 |
| S*FSM – SOPT | 1.92 | — | <1.5 |
| S*FSM – HW + HD = 4 | 1.94 | — | <1.05 |

constrained – obviously the penalty in the layout area. To minimise current consumption, setting a low HD/HDr value is crucial as it directly impacts $N_{peak}(N_{peak} = (HD/HDR/2))$. Implementing an HD/HDr constraint without the HW is only beneficial in the baseline FSMs (Tables 9 and 10).

### *6.4 Security results*

Two factors are examined concerning information security: MI between the power side channel and the HW model, and the MI between the power side channel and the HD model. While not explicitly shown here the amount of information within the current consumption was computed to determine the maximum amount of information overlap between the models and the current. The security results are succinctly shown in Fig. 9.

First, it is important to realise that the amount of information contained within the baseline FSM power side channel fluctuates. In most cases, the heavily constrained encoding, which minimises power, has less information than a typical binary encoding. The next two encoding strategies – relaxed HD (HDR) and the inclusion of HW without constraint are generally on par with the information contained in the baseline FSM. While similar to the standard FSM regarding entropy, the entropy of the constrained S*FSMs encodings is almost always less than the binary-encoded SFSM.

Looking at the MI between the power side channel and the HW models is significantly more interesting. First, in the unsecured FSMs, the MI generally increases when only the HDR constraint is added. As expected, the HW constrained encodings have no MI. The S*FSMs fair slightly better, but still expose information if only the HD constraint is applied.

While it might be expected that the MI between the power side channel and HD fair the same as the HW, this is simply not the case. First, note that only in instances where the base model has few/no self-loops can the MI be reduced to near 0. Second, the greatest reduction in MI occurs when only the reduced HD (HDr) constraint is applied. When further constraints are applied this exacerbates the bi-modal distribution of the HD model thereby increasing the MI. Finally, as expected, the restructured S*FSM leaks no information through the HD constrained encodings. The only parameter to consider here is, therefore, power and layout area costs.

### *6.5 S*FSM versus state-of-the-art*

The current state-of-the-art for high-level synthesisable side channel protection using automated design automation techniques is summarised in Table 11. S*FSMs provide a mechanism to enhance security at a minimal cost to power and clock speed with any underlying standard cell logic and without impacting sensitive clocking networks.

## 7 Conclusion

The overall results show that to provide a secure solution, the typical MCNC FSM requires a 50% increase in the number of states and a 57% increase in the number of product terms needed to define the state transitions. These increases translate to a minimum encoding space increase of 70% raising the average number of bits needed to encode the MCNC benchmarks from 4.8 to 7.9. When factoring in a relaxed structural constraints, and corresponding HDR constraint, we found respective increases of 53 and 67% raising the average number of bits needed to 7.3 and 7.9. The focus of this work was on reducing the power requirements while providing security. Regarding power savings, the current minimisation was possible for both FSMs and SFSMs through the addition of HD/HDr constraints with on average current reductions of 30 and 70%, respectively. Overall, a constrained S*FSM can consume as little as 5% more power than a nominal FSM with binary encodings while requiring a 95% increase in layout area.

A designer with additional layout real-estate can easily trade it for both increased security while mitigating the overall power consumption. If a designer cannot afford to increase area by more

than a third – FSM restructuring is out of the question and the base FSMs with alternative encodings are of interest. In order to increase security they can only guarantee the mitigation of the HW leakage for a nominal current usage penalty. Reduction in the HD leakage is dependent on its bi-modal behaviour – decreasing the HDr constraint (to HDr = 4 or HDr = 2) mitigates the effect but slightly increases the required area (32% increase). If a designer has much greater flexibility in the layout area, then the S*FSMs are recommended for complete side-channel security. A nominal SFSM with binary encoding, while requiring the least overhead of S*FSMs is never recommended. In fact, its use increases the vulnerability of the system. At a 90% increase in layout area a designer can guarantee an FSM free of HD information leakage at 30% current penalty. A 92% layout area increase guarantees an FSM free of both HW and HD information leakage through still requiring a 30% current overhead penalty. Finally, a 94% layout increase still maintains complete HW and HD information secrecy while reducing the current penalty to under 5%.

## 8 References


[1] Tiri, K., Akmal, M., Verbauwhede, I.: 'A dynamic and differential CMOS logic with signal independent power consumption to withstand differential power analysis on smart cards'. Proc. the 28th European Solid-State Circuits Conf. (ESSCIRC 2002), Florence, Italy, 2002, pp. 403–406

[2] Tiri, K., Verbauwhede, I.: 'A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation'. Proc. Design, Automation and Test in Europe Conference and Exhibition, Paris, France, 2004, pp. 246–251

[3] Sundaresan, V., Rammohan, S., Vemuri, R.: 'Power invariant secure IC design methodology using reduced complementary dynamic and differential logic'. IFIP Int. Conf. Very Large Scale Integration (VLSI – SoC 2007), Atlanta, GA, USA, 2007, pp. 1–6

[4] Ramakrishnan, L.N., Chakkaravarthy, M., Manchanda, A.S., *et al.*: 'SDMLp: on the use of complementary pass transistor logic for design of DPA resistant circuits'. 2012 IEEE Int. Symp. Hardware-Oriented Security and Trust (HOST), San Francisco, CA, USA, 2012

[5] Alagar, V., Periyasamy, K.: 'Extended finite state machine', in '*Specification of software systems*' (Texts in Computer Science), (Springer, London, 2011), pp. 105–128

[6] Jiang, Z., Pajic, M., Moarref, S., *et al.*: 'Modeling and verification of a dual chamber implantable pacemaker', in '*Tools and algorithms for the construction and analysis of systems*', (Springer, Berlin, Germany, 2012), pp. 188–203

[7] Jiang, Z., Pajic, M., Mangharam, R.: 'Cyber–physical modeling of implantable cardiac medical devices', *Proc. IEEE*, 2012, **100**, (1), pp. 122–137

[8] Hwang, Y.-T., Lin, S.-C.: 'Automatic protocol translation and template based interface synthesis for IP reuse in SoC'. Proc. the 2004 IEEE Asia-Pacific Conf. Circuits and Systems, Tainan, Taiwan, 2004, vol. 1, pp. 565–568

[9] Zitouni, A., Badrouchi, S., Tourki, R.: 'Communication architecture synthesis for multi-bus SoC', *J. Comput. Sci.*, 2006, **2**, (1), pp. 63–71

[10] Borowczak, M., Vemuri, R.: 'S*FSM: an paradigm shift for attack resistant FSM designs and encodings'. Int. Conf. Cyber Security Redefining and Integrating Security Engineering (RISE 2012), Washington, DC, USA, 2012, pp. 651–655

[11] Kocher, P., Jaffe, J., Jun, B.: 'Differential power analysis', in Wiener, M., (Ed.): '*Advances in cryptology âÂŤ CRYPTOâÂŹ 99*', (LNCS, 1666), (Springer, Berlin/Heidelberg, 1999), pp. 789–789

[12] Verbauwhede, I.: '*Secure integrated circuits and systems, integrated circuits and systems*' (Springer London, Limited, 2010)

[13] Mangard, S., Oswald, E., Popp, T.: '*Power analysis attacks: revealing the secrets of smart cards*' (Advances in Information Security) (Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007)

[14] Schaumont, P., Tiri, K.: 'Masking and dual-rail logic don't add up', in Paillier, P., Verbauwhede, I., (Eds.): '*Cryptographic hardware and embedded systems – CHES 2007*', (LNCS, 4727), (Springer, Berlin/Heidelberg, 2007), pp. 95–106

[15] Macé, F., Standaert, F.-X., Quisquater, J.-J.: 'Information theoretic evaluation of side-channel resistant logic styles', in Paillier, P., Verbauwhede, I., (Eds.): '*Cryptographic hardware and embedded systems - CHES' 2007*', (LNCS, 4727), (Springer, Berlin/Heidelberg, 2007), pp. 427–442

[16] Kulikowski, K., Smirnov, A., Taubin, A.: 'Automated design of cryptographic devices resistant to multiple side-channel attacks'. Workshop on Cryptographic Hardware and Embedded Systems, Yokohama, Japan, 2006, pp. 339–413

[17] Golić, J., Tymen, C.: 'Multiplicative masking and power analysis of AES', in Kaliski, B., Koetin, C.P., (Eds.): '*Cryptographic hardware and embedded systems – CHES 2002*', (LNCS, 2523), (Springer, Berlin/Heidelberg, 2003), pp. 31–47

[18] Oswald, E., Mangard, S., Pramstaller, N., *et al.*: '*A side-channel analysis resistant description of the AES S-Box*' (Fast Software Encryption, Springer, Berlin, Germany, 2005), pp. 413–423

[19] Tiri, K., Hwang, D., Hodjat, A., *et al.*: 'Prototype IC with WDDL and differential routing – DPA resistance assessment'. In Cryptographic Hardware and Embedded Systems – CHES, Edinburgh, UK, 2005, pp. 354–365

[20] Hennessy, J., Patterson, D., Arpaci-Dusseau, A.: 'Computer architecture: a quantitative approach', in '*The Morgan Kaufmann series in computer architecture and design*', vol. 1 (Elsevier, New York, NY, USA, 2011), pp. C.27–C.28

[21] Yeh, T.-Y., Patt, Y.N.: 'Two-level adaptive training branch prediction'. Proc. the 24th Annual Int. Symp. Microarchitecture, MICRO 24, New York, NY, USA, 1991, pp. 51–61

[22] Dietrich, M., Haase, J.: '*Process variations and probabilistic integrated circuit design*' (Springer, New York, NY, USA, 2011)

[23] Li, T., Zhang, W., Yu, Z.: 'Full-chip leakage analysis in nanoscale technologies: mechanisms, variation sources, and verification'. 45th ACM/ IEEE Design Automation Conf. (DAC 2008), Anaheim, CA, USA, 2008

[24] Shen, R., Tan, S.X. D., Yu, H.: '*Statistical performance analysis and modeling techniques for nanometer VLSI designs*' (Springer, New York, NY, USA, 2012)

[25] Johansson, J., Forskitt, J.: '*System designs into silicon*' (Taylor & Francis, Abingdon, UK, 1993)

[26] Cao, C., Oelmann, B.C.: 'The analysis of power-related characteristics of FSM benchmarks'. 50th Midwest Symp. S.M. Circuits and Systems (MWSCAS 2007), Montreal, QC, Canada, 2007

[27] Koegst, M., Franke, G., Feske, K.D.A.C.W.E.-V., *et al.*: 'State assignment for FSM low power design'. Design Automation Conf., 1996, with EURO-VHDL '96 and Exhibition, Proc. EURO-DAC '96, European, Geneva, Switzerland, 1996

[28] Pasha, M.A., Derrien, S., Sentieys, O.C.: 'Ultra low-power FSM for control oriented applications'. IEEE Int. Symp. Circuits and Systems (ISCAS), Taipei, Taiwan, 2009

[29] Akdemir, K.D., Sunar, B.C.D.T.I.: 'Generic approach for hardening state machines against strong adversaries', *IET Comput. Digital Techn.*, 2010

[30] Moradmand, H., Payandeh, A.A.T.F.C.A.I.C.O.: 'Secure finite state integer arithmetic codes'. 2011 Int. Conf. Advanced Technologies for Communications (ATC), Da Nang, Vietnam, 2011

[31] Wang, Z., Karpovsky, M.O.-L.T.S.I.I.T.I.: ' Robust FSMs for cryptographic devices resilient to strong fault injection attacks'. 2010 IEEE 16th Int. Line Testing Symp. (IOLTS), Corfu, Greece, 2010

[32] Grune, D., Jacobs, C.: '*Parsing techniques: a practical guide*' Monographs in computer science. (Springer, New York, NY, USA, 2008)

[33] Cao, C., Oelmann, B.: 'Mixed synchronous/asynchronous state memory for low power FSM design'. Euromicro Symp. Digital System Design (DSD 2004), Rennes, France, 2004, pp. 363–370

[34] Gao, F., Hayes, J.: 'ILP-based optimization of sequential circuits for low power'. Proc. of the 2003 Int. Symp. on Low Power Electronics and Design (ISLPED '03), Seoul, Republic of Korea, 2003, pp. 140–145

[35] Huang, S.-H., Chang, C.-M., Nieh, Y.-T.: 'State re-encoding for peak current minimization'. IEEE/ACM Int. Conf. on Computer-Aided Design (ICCAD '06), San Jose, CA, USA, 2006, pp. 33–38

[36] Lee, Y., Kim, T.: 'State encoding algorithm for peak current minimisation', *IET Comput. Digital Techn.*, 2011, **5**, (2), pp. 113–122

[37] Olson, E., Kang, S.: 'State assignment for low-power FSM synthesis using genetic local search'. Proc. the IEEE 1994 Custom Integrated Circuits Conf., San Diego, CA, USA, 1994, pp. 140–143

[38] Yuan, L., Qu, G., Villa, T., *et al.*: 'FSM re-engineering and its application in low power state encoding'. Proc. the Asia and South Pacific Design Automation Conf. (ASP-DAC 2005), Shanghai, China, 2005, vol. 1, pp. 254–259

[39] Jozwiak, L., Gawlowski, D., Slusarczyk, A.: 'An effective solution of benchmarking problem: FSM benchmark generator and its application to analysis of state assignment methods'. Euromicro Symp. Digital System Design (DSD 2004), Rennes, France, 2004, pp. 160–167

[40] Yang, S.: 'Logic synthesis and optimization benchmarks user guide version 3.0'. 1991

[41] Popp, T., Mangard, S.: 'Masked dual-rail pre-charge logic: DPA-resistance without routing constraints'. Proc. of the 7th Int. Conf. Cryptographic Hardware and Embedded Systems, CHES'05, Springer-Verlag, Berlin, Heidelberg, 2005, pp. 172–186

[42] Bayrak, A.G., Velickovic, N., Regazzoni, F., *et al.*: 'An EDA-friendly protection scheme against side-channel attacks'. Proc. of the Conf. on Design, Automation and Test in Europe, San Jose, CA, USA, EDA Consortium, 2013, pp. 410–415