PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

ARQUITETURAS GALS PIPELINE PARA CRIPTOGRAFIA ROBUSTA A ATAQUES DPA E DEMA

RAFAEL IANKOWSKI SOARES

TESE APRESENTADA COMO REQUISITO PARCIAL À OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIA DA COMPUTAÇÃO NA PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL

ORIENTADOR: PROF. DR. NEY LAERT VILAR CALAZANS
CO-ORIENTADOR: PROF. DR. PHILIPPE MAURINE (LIRMM – UNIVERSITÉ MONTPELLIER 2)

PORTO ALEGRE
2010

Dados Internacionais de Catalogação na Publicação (CIP)

S676a Soares, Rafael Iankowski

Arquitetura GALS pipeline para criptografia robusta a ataques DPA e DEMA / Rafael Iankowski Soares. – Porto Alegre, 2010.

147 f.

Tese. (Doutorado) – Fac. de Informática, PUCRS.

Orientador: Prof. Dr. Ney Laert Vilar Calazans.

1. Informática. 2. Criptografia (Computação). 3. Pipeline.

4. Algoritmos (Computação). I. Calazans, Ney Laert Vilar.

Ficha Catalográfica elaborada pelo

Setor de Tratamento da Informação da BC-PUCRS

RESERVADO AO TERMO DE APRESENTAÇÃO

AGRADECIMENTOS

Agradeço a todos aqueles que de alguma forma contribuíram para este trabalho.

ARQUITETURAS GALS PIPELINE PARA CRIPTOGRAFIA ROBUSTA A ATAQUES DPA E DEMA

RESUMO

As últimas décadas presenciam uma necessidade crescente por sistemas computacionais que garantam o sigilo de informações, seja durante o processamento ou armazenamento destas. Hoje são comuns atividades como compras, transações bancárias, consulta a informações pessoais e reserva de passagens usando a Internet. O uso de redes abertas exige a transmissão protegida de dados confidenciais. O projeto de sistemas integrados em um único chip (em inglês, SoCs) que atendam a restrições de segurança requer protocolos especiais de comunicação e o emprego de criptografia, a ciência que se baseia na aritmética para ocultar informações. Em geral, SoCs que usam criptografia utilizam um texto relativamente curto, denominado *chave criptográfica*, cujo segredo condiciona a eficiência do processo de esconder informações. Em todo sistema criptográfico moderno, conhecer a chave criptográfica equivale a ser capaz de efetuar qualquer operação sobre o conjunto de informações de um dado usuário em um dado sistema. Algoritmos de criptografia são desenvolvidos para resistir à criptoanálise, a ciência de violar textos encriptados explorando vulnerabilidades do processo de ocultação de informação. Embora a maioria dos algoritmos atuais seja robusta a ataques baseados na matemática da criptografia empregada, uma nova classe de técnicas de criptoanálise pode ser usada contra suas implementações. Estes são os chamados *Ataques por Canais Escondidos* ou *Laterais* (do inglês, *Side Channel Attacks*, ou SCA), que permitem correlacionar informações sigilosas tal como uma chave criptográfica com propriedades físicas tais como tempo de processamento, consumo de potência e radiação eletromagnética de dispositivos eletrônicos. O fluxo tradicional de projeto que usa o paradigma síncrono e a tecnologia CMOS favorece a fuga de informações por canais escondidos. Várias propostas para imunizar sistemas criptográficos contra ataques SCA existem na literatura. Dentre as alternativas para a obtenção de sistemas criptográficos seguros, destacam-se paradigmas de projeto específicos tais como o Globalmente Assíncrono e Localmente Síncrono (GALS) e o completamente assíncrono. Esta tese propõe uma nova arquitetura GALS para melhorar a robustez de algoritmos criptográficos. Pressupõe-se o emprego de técnicas *pipeline* e de comunicação assíncrona entre estágios. A robustez é obtida através da combinação de replicação de hardware em estágios pipeline, comunicação assíncrona entre estes estágios e variação independente da freqüência de operação em cada estágio. Os resultados obtidos demonstram um aumento da robustez contra análises de consumo de potência e de radiação eletromagnética nas arquiteturas propostas. Além disso, as arquiteturas apresentam um aumento significativo da vazão de dados, ao custo de um aumento da latência de processamento e da área do circuito, este último provocado pela replicação de hardware. Comparado com o estado da arte em propostas de lógica assíncrona segura, o custo em área mostra-se inferior ou no pior caso compatível, demonstrando que a proposta é uma alternativa interessante de solução para neutralizar ataques SCA.

Palavras chave: criptografia, ataques criptográficos, GALS, pipeline.

GALS PIPELINE ARCHITECTURES FOR CRYPTOGRAPHY ROBUST TO DPA AND DEMA ATTACKS

ABSTRACT

The last decades have witnessed the growth of the need for secure computing systems for either stocking or processing sensitive information. Currently, the Internet is a primary medium for performing numerous activities such as shopping, banking, stocking personal information, ticket reservation among others. The use of open networks to keep and process such information requires computing systems that may securely deal with confidential information. The design of Systems on Chip (SoCs) that fulfill security requirements requires special communication protocols and the use of cryptography, the science based on arithmetic to hide information. In general, SoCs that use cryptography employ a relatively short text, named *cryptographic key*, whose secrecy determines the efficiency of the information hiding process. In any cryptosystem, knowing the cryptographic key enables any operation on any information belonging to a given user in a given system. The design of cryptographic algorithms deems to resist to cryptanalysis, the science of breaking encrypted information by exploiting the vulnerabilities of the information hiding process. Although most current cryptographic algorithms are robust to attacks based on the mathematics of cryptography, a new class of cryptanalysis techniques, called Side Channel Attacks (SCAs) allows correlating sensitive information such as cryptographic keys with the physical properties, such as processing time, power consumption and electromagnetic radiation, of the electronic devices supporting such applications. The traditional design flow that uses the synchronous paradigm and CMOS technology favors the leak of information through side channels. The literature abounds with proposals to make cryptosystems robust against SCA attacks. Among the alternatives available to obtain secure cryptographic systems stand out paradigms such as the Globally Asynchronous Locally Synchronous (GALS) and the use of fully asynchronous systems. This thesis proposes a new GALS architecture to enhance the robustness of cryptographic algorithms. It assumes the use of pipelining and asynchronous communication between each pair of neighbor stages. The approach achieves robustness through a combination of hardware replication into pipeline stages, asynchronous communication between such stages and independent variation of operating frequencies at each stage of the pipeline. The results show increased robustness against power consumption and electromagnetic radiation analysis. Moreover, the proposed and prototyped architectures display a significant data throughput improvement, at the cost of increased latency and area, the later caused by the hardware replication strategy. Compared to state-of-art asynchronous logic secure cryptography, the area costs achieved in this thesis are smaller than, or in the worst case compatible to the best proposals, proving that this is an interesting alternative against SCA attacks.

Keywords: Cryptography, cryptographic attacks, GALS, pipeline.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

ADLBL	*Asynchronous Directional Latch Based Logic*
AES	*Advanced Encryption Standard*
ANSSI	*Agence Nationale de la Securité des Systèmes d'Information*
ASIC	*Application Specific Integrated Circuit*
CEMA	*Correlation Electromagnetic Analysis*
CI	*Circuito Integrado*
CMOS	*Complementary Metal-Oxide Semiconductor*
CMVP	*Cryptographic Module Validation Program*
CAD	*Computer Aided Design*
CPA	*Correlation Power Analysis*
CRC	*Cyclic Redundancy Check*
CRT	*Chinese Remainder Theorem*
DCVSL	*Differential Cascode Voltage Switch Logic*
DDR	*Double Data Rate*
DEMA	*Differential Electromagnetic Analysis*
DES	*Data Encryption Standard*
DFA	*Differential Fault Attacks*
DFS	*Dynamic Frequency Scaling*
DPA	*Differential Power Analysis*
DPL	*Dual Rail Precharge Logic*
DR	*Dual Rail*
DSA	*Digital Signature Algorithm*
DVS	*Dynamic Voltage Scaling*
ECC	*Eliptic Curve Cryptography*
EMR	*Electromagnetic Radiation*
FFD	*Flip-Flop D*
FPGA	*Field Programmable Gate Array*
GALS	*Globally Asynchronous Locally Synchronous*
GF	*Galois Field*
HD	*Hamming Distance*
HO DPA	*High Order Differential Power Analysis*
HW	*Hamming Weight*
ITRS	*International Technology Roadmap of Semiconductors*
IP	*Intellectual Property*
IPA	*Inferential Power Analysis*
LIRMM	*Laboratoire d'Informatique Robotique et Microélectronique de Montpellier*
LR2A	*Leakage Resistant Reconfigurable Architecture*
LUT	*Look Up Table*
Mbps	*Megabits per second*
MD-5	*Message Digest-5*
MDPL	*Masked Dual Rail Pre-Charge Logic*
MTBF	*Mean Time Between Failures*
PDA	*Personal Digital Assistant*
NIST	*National Institute of Standards and Technology*
RCDDL	*Reduced Complementary Dynamic Differential Logic*
RDI	*Random Delay Insertion*
RDVFS	*Random Dynamic Voltage Frequency Scaling*
RNS	*Residue Number System*
RPA	*Refined Power Analysis*
RSA	*Rivest Shamir Adleman*
SABL	*Sense Amplifier Based Logic*
SBOX	*Substitution Box*
SCA	*Side Channel Attack*
SDDL	*Simple Dynamic Differential Logic*
SEMA	*Simple Electromagnetic Analysis*

SiP	*System in Package*
SNR	*Signal to Noise Ratio*
SPA	*Simple Power Analysis*
SoCs	*Systems on Chip*
SR	*Single Rail*
STTL	*Secure Triple Track Logic*
TA	*Timing Attacks*
UMM	*Unique Masked Method*
VLSI	*Very Large Scale Integration*
WDDL	*Wave Dynamic Differential Logic*
WDDL EE	*Wave Dynamic Differential Logic Early Evaluation*
ZPA	*Zero Power Analysis*

SUMÁRIO

1. INTRODUÇÃO

Nas últimas décadas a tecnologia CMOS (do inglês, *Complementary Metal-Oxide-Silicon*) desenvolveu-se vertiginosamente, impulsionada pela evolução da tecnologia submicrônica, que reduz consideravelmente as dimensões de transistores, componente elementar desta tecnologia. Esta evolução permite a concepção de circuitos integrados (CIs) com dezenas de bilhões de transistores, com dimensões na ordem de algumas dezenas de nm nas atuais tecnologias e previsão de dimensões de até 7,4 nm para o ano de 2024 segundo o International Technology Roadmap for Semiconductors (ITRS) [ITR09]. A chamada "*Lei de Moore*" determina a tendência de aumento da capacidade de integração de transistores em um CI [MOO65]. De acordo com a ITRS, no futuro, a integração da tecnologia CMOS com tecnologias não-CMOS tais como componentes biológicos, sensores e atuadores darão origem aos sistemas integrados em encapsulamentos (do inglês, *system in package* - SiP) os quais serão tendência em produtos nanométricos [ITR09].

O avanço na tecnologia de fabricação de circuitos integrados em alta escala de integração (do inglês, *Very Large Scale Integration* - VLSI) proporciona o advento de SoCs (do inglês, *Systems on Chip*) [MAR01]. O projeto e desenvolvimento de SoCs está baseado na técnica de reuso de componentes pré-projetados e pré-validados denominados de núcleos IP (do inglês, *Intellectual Property Core* - IP) [BER01], como forma de reduzir fortemente o tempo de concepção de um produto e garantindo o lançamento no mercado dentro de uma janela adequada de mercado. É possível encontrar SoCs hoje em produtos das mais diferentes áreas de aplicação tais como, por exemplo, sensoriamento remoto [HAN10], processamento paralelo [JAV10] e ainda, em aplicações que exijam segurança ao processamento de dados [MOR09] [NAO09] [CIL10] entre outras.

As últimas décadas presenciam a necessidade crescente por sistemas digitais que garantam o sigilo de informações, seja em seu processamento ou no armazenamento de dados. São comuns as atividades de compra pela Internet, transações bancárias, consultas a informações pessoais, sistemas de reserva de passagens entre outros, que exigem sistemas computacionais operando em redes de acesso como a Internet, o que requer a transmissão protegida de dados confidenciais. O projeto de SoCs que atendam as restrições de segurança exige protocolos especiais de comunicação e também o uso da criptografia, ciência que se baseia na aritmética para ocultar dados. Atualmente com a globalização das indústrias de semicondutores e a verticalização do processo de fabricação existe também a preocupação com a inserção de "*armadilhas*" no projeto de SoCs, principalmente em SoCs para fins militares, segundo Adee em [ADE08].

Na Europa, por exemplo, todos os produtos com restrições de segurança, antes de estarem disponíveis ao mercado, são antes avaliados por uma agência governamental responsável por regulamentar a qualidade dos produtos. Na França, por exemplo, a *Agence Nationale de la Securité des Systèmes d'Information* (ANSSI) emprega um plano denominado de Critérios Comuns para Avaliação da Segurança em Tecnologia da

Informação (*em francês, Critères Communs pour l'Évaluation de la Sécurité des Technologies de l'Information)* onde constam vários níveis de segurança os quais um produto é submetido para avaliar vulnerabilidades, possibilidade de clonagem e riscos provocados por usuários mal intencionados [CCR06]. Este plano de avaliação é desenvolvido em comum acordo com as demais agências européias e norte-americanas de segurança, segundo o Padrão Internacional ISO/IEC 15408:2005 [CCR06].

Nos Estados Unidos existe um programa governamental filiado ao NIST (do inglês, *National Institute of Standards and Technology*) responsável por validar e avaliar os níveis de segurança de algoritmos criptográficos. O Programa de Validação de Módulos Criptográficos (do inglês, *Cryptographic Module Validation Program - CMVP*) valida algoritmos usados em uma variedade de produtos tais como smart cards, dispositivos de armazenagem de dados e produtos com suporte a comércio eletrônico entre outros. A Figura 1.1 apresenta um gráfico do programa CMVP que demonstra o aumento da demanda por certificações e validações de módulos criptográficos nos EUA nos últimos anos [NIS09].

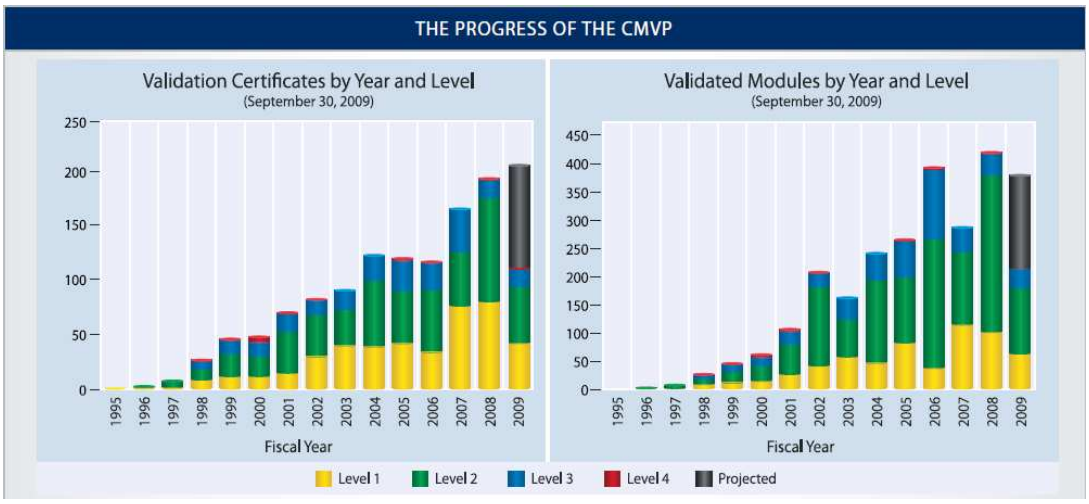


Figura 1.1 – Relatório de emissões de certificados de validação de módulos criptográficos realizados pelo NIST nos EUA [NIS09].

Embora a criptografia tenha sido continuamente desenvolvida para garantir que algoritmos sejam robustos às tentativas de violação de dados confidenciais [CAN09] [MOR09] [NAO09], novas técnicas demonstram que através de propriedades físicas dos sistemas digitais é possível revelar os dados secretos processados. Esta classe de técnicas conhecida como ataques a canais laterais ou escondidos (em inglês, *Side Channel Attacks* - SCA) explora a fuga de informações sensíveis a grandezas tais como o consumo de potência, a radiação eletromagnética, o tempo de processamento, etc. Estas fugas permitem descobrir informações secretas de um sistema, sobretudo aquelas protegidas de criptografia.

Estes ataques procuram estabelecer uma relação de dependência entre os dados processados e as grandezas físicas analisadas. As vulnerabilidades têm origem principalmente nas características de implementação da tecnologia de fabricação de circuitos e também no paradigma síncrono tradicionalmente adotado para a concepção de

sistemas digitais [KOC99] [MOO02]. O consumo de potência em circuitos pode ser modelado formalmente, e seu comportamento previsível pode ser explorado via SCA. O mesmo ocorre com a radiação eletromagnética (em inglês, electromagnetic radiation - EMR) do circuito, pois as variações de corrente ocasionadas durante o processamento geram variações proporcionais no campo eletromagnético radiado pelo circuito, deixando o sistema vulnerável também a este canal lateral. O processo de fabricação de CIs não garante que, por exemplo, os fios de um barramento de dados tenham o mesmo tempo de propagação. As ferramentas de síntese e as variabilidades do processo de fabricação do CI causam desalinhamentos destes fios o que os torna um sistema vulnerável, por exemplo, à análises de tempo de processamento [KOC96].

O projeto de sistemas criptográficos, ou seja, sistemas digitais que utilizem criptografia para ocultar informações confidenciais imunes a ataques por canais laterais (SCAs) é uma área de pesquisa relevante. Neste trabalho sistemas criptográficos são também referenciados como criptosistemas, termo comumente encontrado na literatura. O restante deste Capítulo introduz em maiores detalhes os fatores que influenciam na fuga de informações por canais laterais na Seção 1.1. A Seção 1.2 apresenta as principais técnicas de ataques a canais laterais. A Seção 1.3 introduz as alternativas de projeto de sistemas criptográficos para resistir a ataques por canais laterais. Em seguida são apresentados os objetivos do trabalho na Seção 1.4. A contribuição original é destacada na Seção 1.5 e as demais contribuições são listadas na Seção 1.6. Um resumo da organização dos capítulos que compõem a presente tese conclui este Capítulo na Seção 1.7.

1.1 PARADIGMA SÍNCRONO: PROBLEMAS E ALTERNATIVAS

O desenvolvimento de SoCs em sua quase totalidade parte do pressuposto da discretização de tempo. Este pressuposto submete todas as entradas do sistema à temporização de um único sinal de controle, gerado externamente e denominado relógio (em inglês, *clock*). O pressuposto da discretização de tempo pelo uso de um sinal de relógio global no projeto de sistemas digitais é a característica principal do estilo síncrono de projeto. O tempo transcorrido entre transições do sinal de relógio permite que os valores de entrada amostrados sejam usados na computação de novos resultados. Este tempo garante a estabilização dos valores nas entradas e saídas de elementos de armazenamento, bem como a ocorrência e o descarte adequado de valores transitórios no interior do circuito.

O uso intensivo deste estilo de projeto na concepção de CIs VLSI bem como o uso da tecnologia de fabricação CMOS não traz apenas benefícios. A adoção de um único relógio para controlar todo o sistema digital traz alguns problemas, tais como a distribuição global do relógio, o escorregamento e o consumo de potência neste sinal [CHA08]. Estes problemas eram desprezíveis até meados da década de 80 ou facilmente tratáveis, porém começam a se tornar difíceis de resolver com a acentuada miniaturização da tecnologia VLSI.

Segundo demonstram Ho et al. [HO01], à medida que a tecnologia VLSI evolui, o tempo de propagação de sinais em fios globais vai cada vez mais exceder o período de relógio. Desde a tecnologia 130 nm é necessário em média mais de um período de relógio para um sinal propagando-se em um fio global atravessar todo o comprimento de um chip. Isto contribui para aumentar a complexidade de projeto de sistemas síncronos.

A exigência de altas freqüências de operação para aumentar a velocidade de processamento eleva o consumo de potência, o que não é desejável, principalmente em produtos móveis tais como telefones celulares, games, PDAs, notebooks entre outros, caracterizando uma restrição do projeto de sistemas digitais com o paradigma síncrono. Além desta restrição, a sincronização das operações com o uso do relógio global facilita a correlação entre dados e efeitos físicos mensuráveis externamente, como apresentado por Kocher [KOC96] [KOC99].

A necessidade de métodos alternativos para projetar chips motiva o desenvolvimento de trabalhos tais como [CHA84] proposto por Chapiro. O Autor propõe um novo método de projeto denominado Globalmente Assíncrono e Localmente Síncrono (do inglês, *Globally Asynchronous Locally Synchronous* - GALS) que visa eliminar os relógios globais em sistemas digitais. Esta solução mantém o paradigma síncrono na concepção de módulos funcionais, denominados de ilhas síncronas e elimina o problema do uso de sinais globais por empregar interfaces de comunicação assíncrona entre módulos. Trabalhos tais como [GUR06] [TEE07] [VAN08] se apoiam nos métodos GALS de projeto para eliminar problemas causados por relógios globais.

1.2 FUGA DE INFORMAÇÕES POR CANAIS LATERAIS

Em geral, um sistema criptográfico utiliza uma palavra relativamente curta chamada de chave criptográfica cujo segredo condiciona sua eficiência. Em sistemas criptográficos modernos, conhecer a chave equivale a ser capaz de efetuar as operações no criptosistema. Em 1996, Kocher [KOC96] apresentou o primeiro ataque por canal lateral, denominado de ataque por análise de tempo (em inglês, *Timing Attacks* - TA). Este ataque analisa o tempo de execução de operações em um sistema criptográfico. TA explora ligeiras diferenças na quantidade de tempo necessária para efetuar o processamento de diferentes dados. O ataque mostrou-se capaz de quebrar o sigilo de informações em algoritmos tais como RSA (do inglês, *Rivest Shamir Adleman*) [RIV78] e *Diffie-Hellman* [DIF76].

Kocher et al. em 1999 [KOC99] provam ser possível relacionar o consumo de potência de um circuito digital com os dados processados. Estes Autores mostram que diferentes operações aritméticas possuem diferentes traços de consumo de potência, e denominaram este tipo de avaliação de Análise Simples de Potência (do inglês, *Simple Power Analysis - SPA*). Ainda, eles demonstram que com o uso de métodos estatísticos é possível estabelecer uma correlação entre o consumo de potência e os dados processados em um sistema criptográfico, sendo este tipo de avaliação denominada de Análise Diferencial de Potência (do inglês, *Differential Power Analysis - DPA*).

Mais tarde, em 2001 Gandolfi et al. [GAN01] apresentam experimentos comprovando a possibilidade de encontrar uma chave criptográfica através da análise da radiação eletromagnética de sistemas criptográficos usando os algoritmos DES (do inglês, *Data Encryption Standard*) ou RSA. Estas análises são denominadas de Análises Simples da Radiação Eletromagnética (do inglês, *Simple Electromagnetic Analysis* - SEMA) e Análises Diferenciais da Radiação Eletromagnética (do inglês, *Differential Electromagnetic Analysis* - DEMA).

As análises por consumo de potência, principalmente as análises DPA são as mais citadas na literatura, devido ao seu baixo custo de execução e eficiência. Estas análises exploram basicamente duas características dos circuitos: o comportamento da tecnologia CMOS quanto ao consumo de energia e o sincronismo das operações controladas pelo sinal de relógio. Com base nestes fatores, atacantes exploram a relação deste canal lateral com os dados processados pelo circuito. Maiores detalhes sobre esta análise são apresentados no Capítulo 2 do presente trabalho.

Os trabalhos citados nesta Seção deram início a uma importante área de pesquisa em segurança de projeto de sistemas digitais. As vulnerabilidades encontradas em sistemas criptográficos despertaram o interesse da comunidade acadêmica e da indústria de semicondutores, além de levantar questões de segurança, por exemplo, em entidades governamentais. Outras formas de ataques derivados destas primeiras propostas são encontradas na literatura. Exemplos são os ataques por correlação de modelos de potência [BRI04], análises de consumo de potência de segunda ordem [PRO09], e outras alternativas, revisadas no Capítulo 2 deste trabalho. O desenvolvimento de soluções que reduzam a fuga de informações a níveis que tornem impraticável a obtenção de dados sigilosos é então um tema pertinente de pesquisa.

1.3 ALTERNATIVAS PARA EVITAR A FUGA DE INFORMAÇÕES

O problema da fuga de informações por canais laterais tornou-se uma preocupação no projeto de sistemas digitais que operam com informações confidenciais tais como smart cards. Um smart card nada mais é que um circuito integrado embutido em um cartão plástico que permite o armazenamento, processamento e comunicação de dados de forma segura em uma rede de comunicação pública. Muitos trabalhos têm sido propostos desde então, visando evitar a fuga de informações ou neutralizar a ação de atacantes, pessoas mal intencionadas com alto conhecimento técnico.

Na literatura encontram-se basicamente relatos de três abordagens para reduzir as vulnerabilidades dos sistemas criptográficos a SCAs, sendo elas:

- Injeção de ruído
- Uniformização do consumo de potência
- Mascaramento de dados

A primeira abordagem visa construir circuitos de modo que o consumo de potência seja aleatório durante a execução do algoritmo de encriptação. Assim, a fuga de

informações é ocultada por ruídos que dificultam a ação dos atacantes. Isto pode ser feito em diferentes níveis de abstração do projeto, tais como no nível da arquitetura [GUR06] [STA04] e no nível de portas lógicas [ZAF08] [LU08].

A maioria das propostas de contramedidas que utilizam esta abordagem adiciona hardware extra e/ou atrasos pseudo-aleatórios ao processamento para dificultar as análises DPA [BEN03] [STA04] [ZAF08] [LU08] [KAM09]. Algumas propostas sugerem métodos específicos para implementar a abordagem. Os trabalhos [ZAF08] e [LU08] propõe a inserção de atrasos aleatórios em nível de circuito, visando aumentar a robustez de sistemas a análises DPA. Este método intuitivo pode ser superado por ataques especializados, tal como demonstrado em [NAG07], pois a encriptação completa de um dado é realizada com a mesma freqüência de relógio. Kamoun et al. [KAM09] propuseram a replicação das funções vulneráveis do algoritmo criptográfico AES, de modo a gerar ruído no consumo de potência e assim ocultar a fuga de informações. Standaert et al. [STA04] propuseram o uso de uma arquitetura pipeline para neutralizar ataques DPA. Os resultados revelam redução significativa na fuga de informações, embora existam relatos de ataques DPA bem sucedidos, mesmo que com uma pequena margem de certeza de hipóteses corretas de chave.

A segunda abordagem visa alterar as características de consumo de potência do circuito de modo a obter um consumo uniforme e independente dos dados processados durante a encriptação. A maior parte das soluções propostas opera no nível de portas lógicas, havendo a proposição de vários estilos lógicos específicos [CIL10] [KUL08] [RAM08] e fluxos de projetos [GUI08b] a fim de reduzir a fuga de informações sigilosas através de canais laterais. O uso do paradigma assíncrono de projeto é uma alternativa para a concepção de sistemas robustos a DPA conforme [BOU05] [KUL05] [CHE06] [TIR04], mas existem vulnerabilidades resultantes do processo de síntese de hardware, segundo estudos realizados por Razafindraibe et al. em [RAZ07].

A terceira abordagem visa mascarar os valores intermediários produzidos durante a execução do algoritmo de encriptação, ou seja, estes valores são alterados de modo que sejam independentes dos valores realmente produzidos. Embora as características de consumo de potência dos circuitos sejam as mesmas, a abordagem previne a ação dos ataques pela alteração dos valores intermediários apenas. Diversas propostas de sistemas criptográficos empregando mascaramento de dados são encontradas na literatura tais como [PRA04], [HUI07], [GHE08], [MES05] os quais serão discutidos em mais detalhes no Capítulo 3 deste trabalho.

Cabe salientar que em muitos trabalhos o termo *mascaramento* é utilizado para descrever a tentativa de ocultar a fuga de informações pela inserção de ruído ou atrasos ao processamento, como propõe a primeira abordagem. Este comportamento difere da terceira abordagem onde realmente existe a aplicação de uma máscara gerada aleatoriamente a fim de modificar os dados antes e/ou durante a execução do algoritmo de criptografia. Neste trabalho o termo *mascaramento* será utilizado apenas para os métodos descritos como esta terceira abordagem de contramedida.

1.4 OBJETIVOS

Esta tese propõe contramedidas para a concepção de sistemas criptográficos imunes a análises diferenciais de potência (DPA) e análises diferenciais de radiações eletromagnéticas (DEMA). Esta Seção lista os objetivos estratégicos e específicos deste trabalho.

Os objetivos estratégicos desta tese são:

- Contribuir para a pesquisa em Segurança a Sistemas Embarcados;
- Revisar o problema da fuga de informações através de canais escondidos;
- Dominar e avaliar os métodos empregados para resolver este problema;
- Dominar a metodologia não-síncrona de projeto adotada como base de projeto para a tese;
- Dominar o funcionamento do algoritmo criptográfico DES usado como estudo de caso neste trabalho;
- Propor soluções de imunização usando duas abordagens diferentes;
- Avaliar as soluções propostas.

Para alcançar os objetivos estratégicos são realizadas atividades para atingir os seguintes objetivos específicos:

- Definir uma infraestrutura para a prototipação de circuitos não-síncronos em FPGAs:
  - Propor uma biblioteca de hard macros (módulos implementados por componentes primitivos do FPGA permitindo restrições temporais ou de posicionamento);
- Definir uma infraestrutura GALS adequada para implementar contramedidas:
  - Definir um protocolo de comunicação assíncrono entre as ilhas síncronas;
  - Propor um subsistema de geração pseudo-aleatória de sinais de relógio;
  - Propor a replicação dos módulos funcionais do algoritmo DES de modo a obter uma arquitetura pipeline;
  - Encapsular os módulos funcionais em ilhas síncronas;
  - Validar os sistemas com e sem contramedidas.
- Propor diferentes configurações de pipelines para avaliar a robustez a ataques DPA e DEMA;
- Avaliar a infraestrutura proposta quanto aos custos em área, vazão e latência:
  - Construir um sistema preciso para medição de potência e de radiação eletromagnética.

1.5 ORIGINALIDADE DA TESE

A originalidade do trabalho consiste na proposta de contramedidas a análises DPA e DEMA, visando explorar a lacuna existente pela combinação dos métodos de inserção de atrasos, uso de arquiteturas pipeline e o método GALS de projeto associado à variação dinâmica de freqüência. Os métodos de inserção de ruído aplicam-se sobre o processamento completo de um algoritmo, sendo facilmente contornados por técnicas de ressincronização existentes. As arquiteturas pipelines encontradas na literatura são síncronas, o que facilita a ação de atacantes, devido ao determinismo do relógio. O

paradigma GALS permite a combinação de arquiteturas pipeline com técnicas de variação dinâmica de freqüência. Desta forma, potencializa os efeitos de aleatoriedade sobre padrões de consumo e radiação eletromagnética. Conseqüentemente, melhoram a robustez a estes tipos de análises, conforme demonstram os experimentos descritos e realizados aqui.

1.6 CONTRIBUIÇÕES DA TESE

O presente trabalho tem duas principais contribuições para a concepção de sistemas criptográficos robustos a ataques DPA e DEMA: (i) o desenvolvimento de um protótipo de uma biblioteca lógica assíncrona para prototipação em FPGAs; (ii) a proposta de arquiteturas GALS pipeline. Estas contribuições podem ser resumidas do seguinte modo:

Protótipo STTL:

1. Protótipo em FPGA da lógica assíncrona com codificação em três trilhas desenvolvida em standard cells por Razafindraibe et al. [RAZ07], denominada de lógica segura em três trilhas (em inglês, *Secure Triple Track Logic* - STTL).
2. Automatização do fluxo de projeto de circuitos STTL usando hard macros.
3. Infraestrutura para medição do consumo de potência e da radiação eletromagnética de circuitos prototipados em FPGA.
4. Avaliação da robustez do protótipo da lógica STTL.

Arquiteturas GALS Pipeline:

1. Diversos protótipos do algoritmo DES onde o laço do algoritmo é desenrolado em estágios de hardware de diferentes maneiras, validados em FPGA.
2. Infraestrutura GALS para o projeto de um pipeline com estágios síncronos e comunicação assíncrona entre estágios.
3. Infraestrutura para a implementação de estágios com domínios de freqüência diferentes e variação da freqüência para cada dado processado.
4. Avaliação da robustez das arquiteturas GALS pipeline propostas.

1.7 ORGANIZAÇÃO DO DOCUMENTO

O restante deste documento organiza-se da seguinte forma. O Capítulo 2 apresenta um estudo sobre o consumo de potência em circuitos com tecnologia CMOS e o fluxo de execução de um ataque DPA, mostrando as vulnerabilidades da tecnologia. O Capítulo 3 apresenta trabalhos relacionados aos temas abordados nesta tese. No Capítulo 4 discute-se um conjunto de experimentos relacionados ao uso de lógica assíncrona orientada a FPGAs como contramedida a SCAs. O Capítulo 5 detalha uma proposta geral de contramedida usando o paradigma GALS pipeline. O Capítulo 6 conclui a Tese, destacando os trabalhos desenvolvidos no contexto do doutorado, as contribuições deste e apontando um conjunto de atividades de continuidade para o trabalho.

2. DEFINIÇÕES BÁSICAS

Este Capítulo explora alguns conceitos básicos relacionados à criptografia e criptoanálise. Além disso, é apresentada uma classificação para os diversos tipos de criptoanálise, e um detalhamento sobre análises de consumo de potência, sua base teórica e sua dinâmica para revelar o conteúdo de uma mensagem. Estas análises servirão como ferramenta para avaliar as arquiteturas propostas no presente trabalho quanto à robustez a análises DPA e DEMA.

Utilizam-se aqui definições básicas de textos clássicos, tais como o livro de Schneier [SCH96]. Suponha que um *emissor* deseja enviar uma mensagem a um *receptor*. Suponha também que o *emissor* deseja enviar a mensagem de forma segura, tal que nenhum outro ente exceto o *receptor* possa ler a mensagem. O processo de disfarçar uma mensagem para conseguir este intento é denominado de *encriptação*. O processo de obter a mensagem original a partir da mensagem encriptada se denomina *decriptação*. *Criptografia* é a ciência responsável por manter mensagens seguras e ela é praticada por *criptógrafos*. Os *criptoanalistas* são os praticantes da *criptoanálise*, a arte e a ciência de violar textos encriptados. Diz-se que um *criptoanalista* realiza um *ataque criptográfico* quando este busca violar um texto encriptado. O ramo da matemática que inclui criptografia e criptoanálise é a *criptologia*, e seus praticantes são denominados *criptologistas*.

2.1 CRIPTOLOGIA

A criptologia, palavra derivada do grego *krypto* (oculto) e *logos* (estudos), tem como objetivo a concepção e análise de mecanismos que permitem assegurar a integridade, autenticidade e a confidencialidade de dados em comunicações. As Seções seguintes discutem sobre a criptografia e a criptoanálise.

2.1.1 CRIPTOGRAFIA

A criptografia é a ciência que se baseia no estudo de princípios e técnicas pelas quais mensagens podem ser transformadas de sua forma original para outra ilegível, de forma que possam ser conhecidas apenas pelos entes comunicantes, emissor e receptor. A operação de transformar os símbolos que compõem uma mensagem em uma sucessão distinta de símbolos aplicando cálculos de modo que o apenas o receptor das mensagens, de posse de informação privilegiada, possa decifrá-las, é a essência da encriptação. A criptografia é realizada geralmente com ajuda de uma *chave criptográfica*.

Uma *chave criptográfica* é um conjunto de símbolos que permite ao receptor da mensagem aplicar o processo de decriptação a um texto encriptado e assim obter a mensagem original. A chave é ou está intimamente relacionada ao conceito de *senha*, muito difundido entre usuários finais de aplicações que empregam criptografia como forma de obter segurança na comunicação de dados sensíveis. O objetivo principal de ataques criptográficos é descobrir chaves.

De acordo com a forma de disponibilização de chaves, os algoritmos de criptografia podem ser classificados em duas categorias: os algoritmos com *chave privada* e os algoritmos com *chave pública* [SCH96].

Os algoritmos com chave privada, também conhecidos como *algoritmos simétricos*, têm por princípio a utilização de uma mesma chave para as operações de encriptação e decriptação de mensagens. Esta última implica que as entidades que desejam se comunicar de maneira segura devem imperativamente trocar a chave e este é o principal inconveniente desta classe de algoritmos. Dentre os algoritmos de criptografia com chave privada, pode-se destacar o DES, o triplo DES (3-DES) e o AES. Este trabalho concentrará esforços nesta categoria de algoritmos, particularmente no algoritmo DES, por ter uma grande aplicação em smart cards [SEL09]. Embora seu sucessor AES opere com chaves de até 256 bits e também permita o emprego da abordagem proposta neste trabalho, a escolha pelo algoritmo DES foi tomada com base na experiência de estudos de caso anteriores realizados pelo grupo de pesquisa ao qual os Autores deste trabalho estão inseridos.

Os algoritmos com chave pública, também conhecidos como *algoritmos assimétricos*, têm por princípio a utilização de duas chaves: uma chave publicamente disponível para a encriptação e uma chave privada para a decriptação. Em um sistema de criptografia deste tipo, usuários escolhem uma chave aleatória que apenas eles devem conhecer (*a chave privada*). A partir desta chave e de um algoritmo eles geram a chave pública. Em seguida, os usuários disponibilizam a chave pública através de canais possivelmente não-seguros. Qualquer emissor pode assim criptografar mensagens com a chave pública que somente poderão ser decriptadas pelo receptor, de posse de sua chave privada. A comunicação bidirecional pressupõe então o uso de duas chaves públicas e duas chaves privadas, todas distintas. Este tipo de algoritmo foi estabelecido inicialmente por Diffie e Hellman [DIF76], a fim de resolver o problema ligado à transferência de chaves secretas. Dentre os algoritmos de criptografia com chave pública pode-se destacar o RSA, o El Gamal e o DSA (em inglês, *Digital Signature Algorithm*).

2.1.2 CRIPTOANÁLISE

A encriptação de mensagens via algoritmos de criptografia é necessária devido aos inúmeros meios disponíveis hoje para quebrar o sigilo de mensagens. Uma tentativa de criptoanálise é comumente chamada de *ataque criptográfico*. Entre os vários tipos de ataques existentes, é possível agrupá-los em duas grandes famílias, os *ataques lógicos* e os *ataques físicos*. Inicialmente, revisam-se ataques lógicos, que exploram as vulnerabilidades matemáticas dos algoritmos. Em seguida, revisam-se os ataques físicos, que exploram as vulnerabilidades físicas dos dispositivos eletrônicos que dão suporte à execução de algoritmos de criptografia, também referenciados neste trabalho como *sistemas criptográficos*.

ATAQUES LÓGICOS

Dentre os ataques que exploram as vulnerabilidades matemáticas é possível dividi-los em dois grupos, sendo eles *lineares* e *diferenciais*. Os *ataques lineares* propostos por Matsui em 1993 [MAT93] estudam as relações existentes entre os bits de uma mensagem, os bits da mensagem encriptada correspondente, ou *criptograma* correspondente, e da chave utilizada na criptografia. Estas relações são usadas para obter uma expressão linear capaz de predizer valores dos bits da chave quando muitas mensagens e os respectivos criptogramas são conhecidos. Aumentado o número de pares mensagem-criptograma disponível, é possível melhorar a precisão da aproximação. Os algoritmos de criptografia devem apresentar resistência a este tipo de ataque. Na literatura são encontradas outras propostas baseadas nos ataques lineares de Matsui, tais como [ROU03], [MAN06], [BAG07].

O ataque diferencial proposto por Bihan e Shamir [BIH90] baseia-se em ataque no qual o atacante dispõe de vários pares mensagem-criptograma escolhidos, cujas diferenças entre os respectivos criptogramas são analisadas. Entende-se por diferença a operação de ou-exclusivo (XOR) entre dois criptogramas. A criptoanálise é realizada sobre pares de criptogramas encriptados com a mesma chave e cujas mensagens correspondentes possuem certo valor particular de diferença. O efeito desta diferença é analisado através das ‘*n’* iterações do algoritmo resultando em parâmetros que permitem inferir possíveis valores da chave utilizada no processo de encriptação. Estes parâmetros são expressos analiticamente através de probabilidades e são usados como indicadores para tomada de decisão de qual chave foi utilizada para cifrar a mensagem criptoanalisada. O método fornece como resultado um conjunto de probabilidades associadas respectivamente a um conjunto de chaves. A decisão pela chave correta é feita escolhendo-se aquela cuja probabilidade é a de maior valor. Na literatura são encontradas diversas propostas baseadas nos ataques diferenciais de Bihan, tais como [JIE06] [DUN07] [JAK07].

ATAQUES FÍSICOS

Nos últimos anos vários tipos de ataques a sistemas criptográficos têm surgido com o propósito de revelar a chave criptográfica destes sistemas. Entretanto, ataques usados para alcançar este objetivo apresentam-se de várias maneiras se diferenciando significativamente em termos de custo, tempo, equipamentos necessários e conhecimento técnico. Em conseqüência, existem vários modos de classificá-los. Na literatura é comumente encontrada a classificação segundo dois critérios ortogonais. O primeiro destes divide os ataques com relação ao controle sobre o dispositivo, classificando-os como *passivos* ou *ativos.*

Ataques Passivos: o sistema criptográfico opera normalmente dentro de suas especificações. O ataque consiste apenas em observar as propriedades físicas do dispositivo tais como tempo de execução, consumo de potência e radiação eletromagnética.

Ataques Ativos: Estes ataques controlam as entradas e/ou o ambiente onde o sistema criptográfico está inserido. Deste modo, os ataques adulteram dados de entrada e/ou o ambiente de modo a explorar anormalidades produzidas nestes casos pelo sistema.

O segundo critério classifica os ataques segundo a interface do dispositivo explorada pelos atacantes. Os sistemas criptográficos têm diversas interfaces físicas e lógicas. Algumas destas podem ser acessadas facilmente enquanto outras exigem equipamentos especiais. Com base nestas interfaces os ataques físicos podem ser classificados em três grupos:

Ataques invasivos: nestes ataques não existem limites para se manipular o dispositivo eletrônico visando revelar a chave secreta do sistema. Tipicamente, inicia-se com a remoção do encapsulamento do dispositivo. A seguir, componentes são acessados diretamente por sondas especiais para observar sinais em um barramento (ataques passivos), por exemplo, ou provocar a alteração da funcionalidade do dispositivo em um ataque ativo. Estes ataques violam a integridade física do dispositivo, impossibilitando seu uso após o ataque. Além disso, exigem equipamentos especiais, o que torna os ataques extremamente caros. Exemplos deste ataques:

- *Ataque por sondagem* consiste em espionar a atividade elétrica do componente extraído do circuito, com a ajuda de uma sonda. Esta técnica permite recuperar os dados transitando dentro do circuito, mas permite também impor valores lógicos em certos pontos deste. Pode-se imaginar que com o controle do ambiente, o atacante pode estar medindo e deduzindo boa parte do segredo do circuito criptográfico.
- *Ataques por reconstrução de leiaute (engenharia reversa)*: consistem em estudar o componente eletrônico extraído, para determinar de maneira precisa a estrutura interna e assim deduzir seu funcionamento. Para isto é necessário extrair as informações sobre o local exato de todos os transistores e de todas as conexões, compondo a estrutura do sistema, a fim de reconstruir a totalidade do seu leiaute. Este ataque é muito raro, pois exige materiais sofisticados, pessoal altamente treinado e tem alto custo.

Ataques semi-invasivos: nestes ataques o dispositivo também tem seu encapsulamento removido. Entretanto, diferentemente dos ataques invasivos, não existe contato elétrico com o circuito do dispositivo, ou seja, o circuito permanece intacto. Tipicamente estes ataques não exigem equipamentos caros, porém os esforços para executá-los são relativamente altos, pois o processo de localizar a posição correta dos componentes para um ataque na superfície de um chip moderno exige tempo e conhecimentos especializados. É possível citar como exemplo os seguintes ataques:

- *Ataques por injeção de falhas*, introduzidos por Boneh et al. [BON97] em 1997 e Skorobogatov e Anderson em [SKO02], consistem em gerar intencionalmente falhas no criptosistema a fim de obter comportamentos anormais, pode-se então explorar

os mesmos para revelar informações secretas. No entanto, estes ataques precisam criar modelos de falhas, exigindo competências particulares e um conhecimento detalhado da estrutura interna do circuito. Sua eficiência representa hoje um forte perigo para a segurança de criptosistemas.

- *Ataques eletromagnéticos* consistem em revelar e analisar a radiação eletromagnética produzida por um circuito criptográfico. A maioria dos criptosistemas atuais é cadenciada por um sinal de relógio que sincroniza a operação de todo o circuito. Todos os sinais elétricos são decorrentes do movimento de cargas elétricas provocado por forças elétricas. Este movimento de cargas produz campos elétricos e magnéticos. Ciente de que estes sinais elétricos são fortemente dependentes dos dados manipulados, uma análise minuciosa da radiação eletromagnética pode permitir descobrir informações secretas contidas no dispositivo. Estes ataques podem ser classificados como semi-invasivos caso ocorra o processo de retirada do invólucro do circuito, mas normalmente são considerados não-invasivos.

Ataques não-invasivos: Neste ataques apenas as interfaces diretamente acessíveis são atacadas. O dispositivo não é permanentemente alterado, fato que não deixa evidências de que um ataque tenha sido realizado. A maioria destes ataques é realizada a custos reduzidos, se comparado a ataques invasivos. Isto os torna uma séria ameaça à segurança de sistemas criptográficos.

Em particular ataques passivos e não-invasivos têm recebido uma grande atenção durante os últimos anos. Estes ataques são freqüentemente referenciados como *ataques a canais laterais* (SCAs). Prover soluções arquiteturais para aumentar a robustez contra estes tipos de ataque é o alvo do presente trabalho. De fato, tais ataques consistem em explorar os canais laterais (tempo de cálculo, consumo de potência, radiação eletromagnética). Estes canais laterais correlacionam-se com o estado interno do circuito, sendo possível a partir deles extrair algumas informações secretas. Alguns exemplos deste tipo de ataque:

- *Ataques por análise de tempo*, propostos por Kocher em 1996 [KOC96] exploram a correlação entre os dados processados e o tempo de processamento durante as operações criptográficas. Algoritmos criptográficos têm tempos de cálculo dependentes dos dados e da chave secreta. Uma análise destes tempos pode permitir revelar o valor da chave secreta.
- *Ataques por análise do consumo de potência* consistem em analisar o consumo de potência de dados manipulados em um circuito a fim de extrair a chave criptográfica.

Este trabalho tem ênfase nos ataques por análise do consumo de potência e por análise da radiação eletromagnética como modo de avaliar a robustez das arquiteturas propostas.

Neste caso, as análises de radiação eletromagnéticas são não-invasivas, pois nenhum tipo de alteração do empacotamento do FPGA é realizado. Os ataques por análise de tempo são ignorados no processo de avaliação deste trabalho.

2.2 ATAQUES POR CONSUMO DE POTÊNCIA

Esta Seção detalha ataques por análise do consumo de potência. Em um primeiro instante revisam-se características do consumo de potência em circuitos CMOS. A seguir, detalham-se os ataques por análise diferenciais de potência.

2.2.1 CARACTERÍSTICAS DO CONSUMO DE POTÊNCIA EM CIRCUITOS CMOS

Para entender o fundamento de ataques por análise de consumo de potência, em particular os ataques DPA, é necessário revisar as características de consumo de potência da tecnologia CMOS. Como ilustrado na Figura 2.1, um circuito CMOS síncrono é tipicamente composto de *estágios*, formados por portas lógicas e delimitado por registradores, que por sua vez são compostos de transistores operando como chaves.

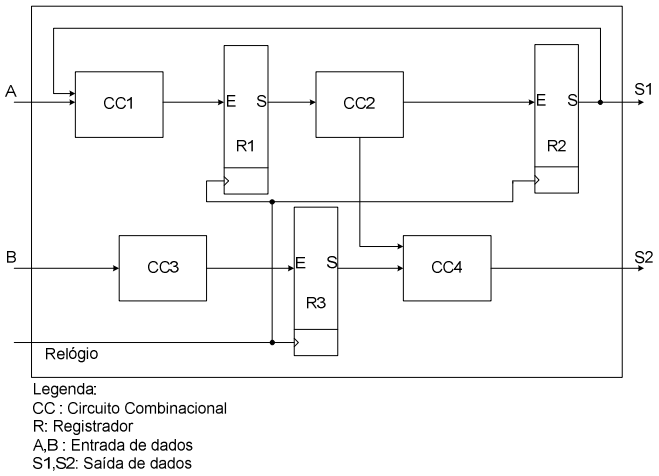


Figura 2.1 - Diagrama esquemático mostrando a estrutura de um estágio em um circuito CMOS síncrono.

A potência consumida em tais circuitos depende da atividade de chaveamento de seus componentes. Esta atividade é promovida pela variação dos dados de entrada no circuito, bem como pela atividade na rede de distribuição de relógio. Inicialmente, assume-se que o circuito encontra-se em um *estado de equilíbrio*, onde todos os sinais de entrada têm valores constantes e estão estáveis, nenhum sinal interno encontra-se chaveando e os sinais de saída estão também em valores estáveis. Variações nos sinais de entrada provocam o chaveamento de sinais internos na primeira parte do estágio, o registrador de entrada. Caso haja então uma transição do sinal de relógio, as mudanças nas entradas propagam-se para a segunda parte do estágio (o circuito combinacional) e até a terceira parte do estágio (o registrador de saída). Uma transição posterior do sinal de relógio pode então propagar valores para fora do estágio, alterando os sinais da saída do estágio. A sequência de atividades de chaveamento descrita se tomada em nível de todo o circuito, com todos seus estágios considerados, é definida como uma transição de estado do

circuito. Deste modo, pode-se afirmar que o consumo de potência de um circuito depende das transições de estado e também do estado em que o mesmo se encontra [SZE02].

As fontes de dissipação de potência em um circuito CMOS podem ser divididas em duas classes de contribuições: componentes estáticos e componentes dinâmicos. No caso de uma porta CMOS, componentes estáticos correspondem à potência dissipada quando este se encontra em estado de equilíbrio, ou seja, quando não passa por transições nas suas entradas ou saídas. Idealmente, o consumo de potência estática de um circuito lógico CMOS deveria ser nulo na ausência de atividade. Contudo, transistores não são chaves perfeitas. Mesmo em estado de equilíbrio, eles apresentam uma corrente de fuga responsável pela dissipação estática de potência. Em tecnologias mais recentes esta componente de dissipação tem se tornado cada vez mais relevante [KIM03].

A potência dinâmica costumava ser e em alguns casos ainda é a principal fonte de dissipação nos circuitos CMOS. Esta potência é dissipada durante as transições de estado do circuito. É possível admitir que a potência estática seja insignificante diante da potência dinâmica enquanto o circuito está em plena atividade tal como, por exemplo, a execução de uma encriptação de dados. Por conseqüência, é possível exprimir a potência consumida por uma porta CMOS de acordo com a equação (1):

$$P = \alpha C_L Vdd^2 F \qquad (1)$$

Nesta Equação, α representa a taxa de atividade da porta, C_L representa a carga capacitiva desta, F é a freqüência de operação e Vdd corresponde à tensão de alimentação. Nesta Seção, a fim de limitar o escopo das análises, serão consideradas apenas as cargas/descargas capacitivas como as principais fontes de consumo de potência.

Em tecnologias de até 130nm é possível afirmar que a potência estática em um circuito CMOS é insignificante em relação à potência dinâmica. A partir das tecnologias de 65nm ou menores o consumo de potência estático pode tornar-se a maior componente no consumo de potência total de um circuito. Isto leva a crer que os efeitos causados pela encriptação de dados em criptosistemas correspondem à menor parte do consumo total de potência do sistema. Logo se espera que as fugas de informação por este canal lateral sejam atenuadas em tecnologias com dimensões inferiores a 65nm.

2.2.2 ANÁLISE DO TRAÇO DE CONSUMO DE CORRENTE

Os ataques por análise do consumo exploram as variações instantâneas de corrente de um circuito. Nesta Seção são apresentados alguns parâmetros do ambiente em que o circuito se encontra e parâmetros da tecnologia de concepção que influenciam nas variações do consumo de corrente das portas lógicas CMOS, seguindo estudo realizado por Razafindraibe et al. [RAZ04] [RAZ06].

O traço de consumo de corrente de uma porta lógica CMOS depende de parâmetros do ambiente no qual a porta está inserida e da tecnologia adotada, conforme

[AUV00] [NIK99]. Entre os parâmetros do ambiente inclui-se a rampa de entrada, definida como o tempo no qual um sinal transita do nível lógico ‘0’ para ‘1’ ou de ‘1’ para ‘0’, e a carga ou, em inglês, *fan-out*, definido como o número máximo de portas de entrada às quais uma porta de saída pode conectar-se. Entre os parâmetros da tecnologia de concepção estão a topologia do circuito e o dimensionamento dos transistores, entre outros. A fim de identificar o modo como alguns destes parâmetros influenciam os traços de corrente, Razafindraibe et al. em [RAZ04] [RAZ06] desenvolveram simulações elétricas sobre circuitos CMOS com diferentes complexidades e dimensões, usando uma tecnologia 130nm. O circuito inversor CMOS mostrado na Figura 2.2 foi usado como referência nestes estudos.

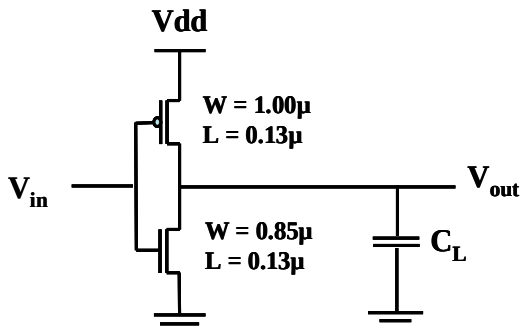


Figura 2.2 Inversor CMOS dimensionado em uma tecnologia 130 nm.

Primeiramente, analisa-se o impacto do parâmetro *carga* (C_L) sobre os traços de corrente. Para simular a ação deste parâmetro no consumo de corrente do circuito utiliza-se um capacitor com valor C_L que reproduz o comportamento elétrico de outras portas de entrada conectadas à saída do circuito. Neste caso, o parâmetro carga é diretamente proporcional à capacitância $C_{L.}$ Simulou-se o circuito inversor CMOS submetido às seguintes condições: rampa de entrada constante (τ_{in}=100ps) e a diferentes cargas (4fF à 160fF). Em um segundo instante, observou-se o efeito do parâmetro rampa de entrada sobre o traço de corrente, com o inversor submetido a uma carga constante (40fF) e a diferentes valores de rampa de entrada (10ps a 600ps).

Os resultados da simulação aparecem na Figura 2.3. V_{in} representa as possíveis transições de entrada [1-0], entre os instantes 0 e 1ns, e [0-1] entre 3ns e 4ns. As curvas V_{out} representam a tensão de saída do circuito para as diferentes cargas. *I(Vdd)* representa a corrente consumida durante as transições. A análise destes resultados permite concluir que as transições [0-1] e [1-0] produzem circulação de corrente com amplitudes significativas. A partir da Figura 2.3, percebe-se que as transições de entrada [1-0] produzem consumo de corrente significativamente superiores às transições [0-1].

O traço de corrente de um inversor depende significativamente da capacitância de carga C_L. A Figura 2.3 representa a evolução temporal das correntes de carga e descarga para diferentes valores de C_L. Contudo, a partir de certo valor de C_L, a amplitude máxima de corrente não depende mais do valor de C_L. Neste instante, o valor máximo de corrente é limitado pelas características de condução de corrente dos transistores. Em

conseqüência, os traços de corrente têm tendência a se espalhar no tempo de maneira proporcional ao valor da carga.

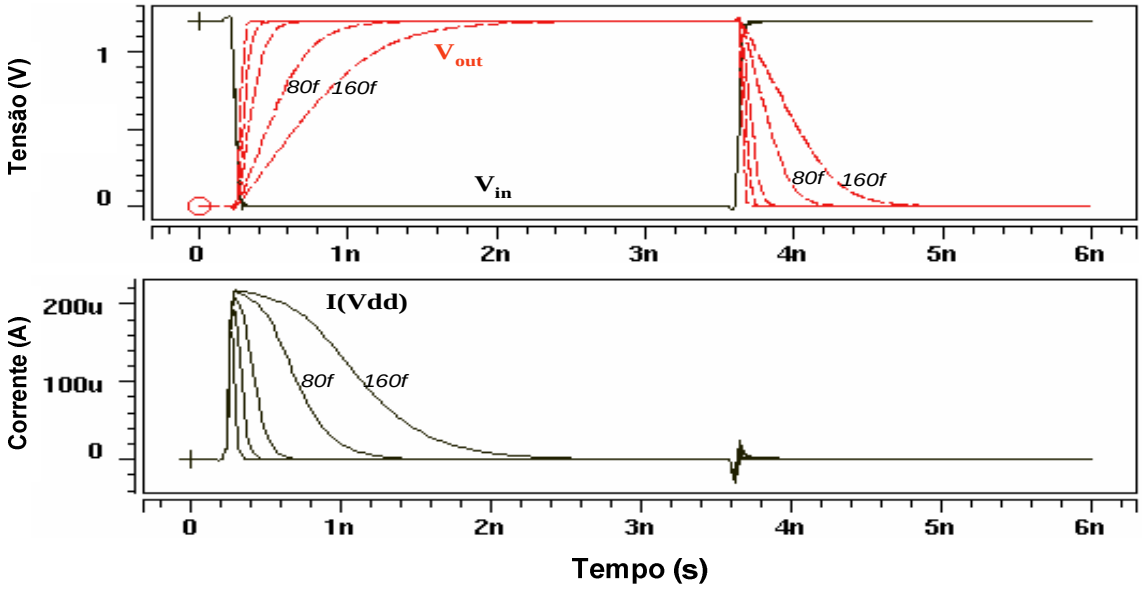


Figura 2.3 Avaliação do consumo de corrente devido à variação de valores de capacidade de carga C_L [RAZ06].

Os traços de corrente são muito sensíveis a variações de inclinação da rampa de entrada. Como mostra a Figura 2.4, a amplitude máxima de corrente é diretamente proporcional à inclinação da rampa de entrada. Por outro lado, uma diminuição da inclinação da rampa de entrada se traduz em um deslocamento temporal da resposta do inversor e por conseqüência, dos traços de corrente.

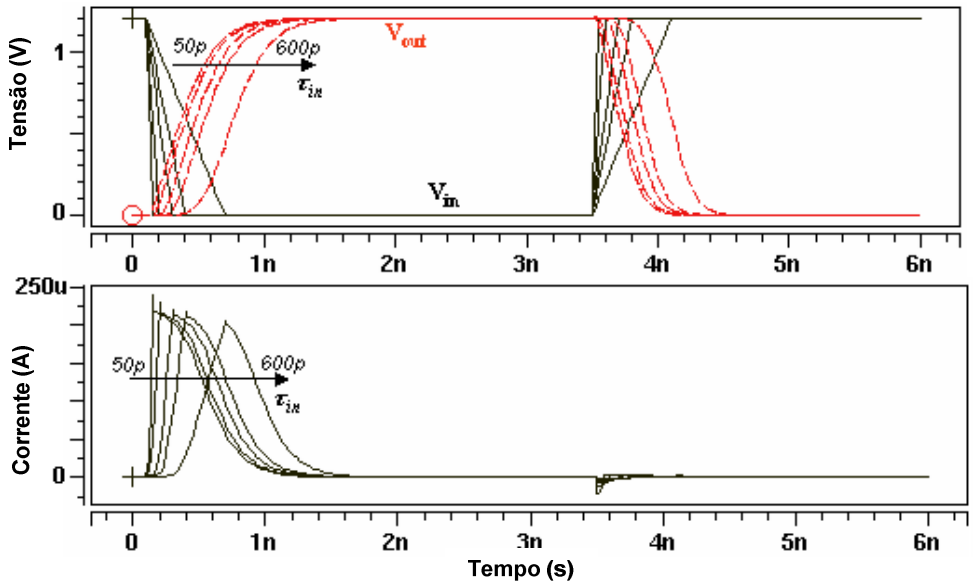


Figura 2.4 Avaliação da corrente para diferentes valores de rampa de entrada τ_{in} [RAZ06].

A fim de avaliar a influência de conexões série/paralelo de transistores em uma porta lógica sobre o traço de corrente, Razafindraibe et al. utilizaram uma porta lógica NAND de duas entradas (NAND-2) da Figura 2.5 como estudo de caso. Durante as simulações, a rampa de entrada (τ_{in}=100ps) e a carga (40f) são constantes. Os parâmetros que variam neste caso são apenas a ordem de chegada dos sinais e os valores de entrada.

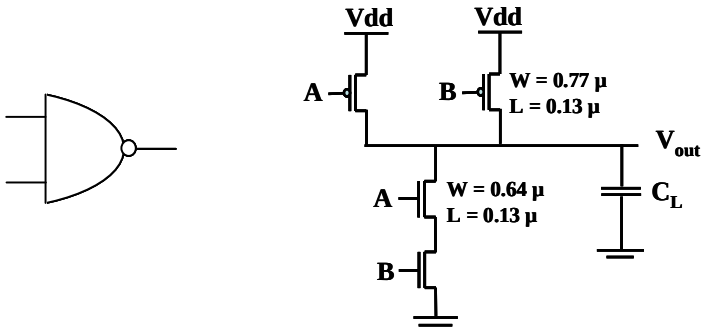


Figura 2.5 Esquemático de uma porta NAND de duas entradas.

Figura 2.6 apresenta o consumo de corrente em uma porta de NAND-2 em função das possíveis combinações de transições de entrada. A amplitude máxima de corrente é proporcional ao número de transistores em comutação. Por outro lado, assim como os atrasos de propagação da porta, a duração dos traços de corrente é muito sensível às combinações de transições de entrada.

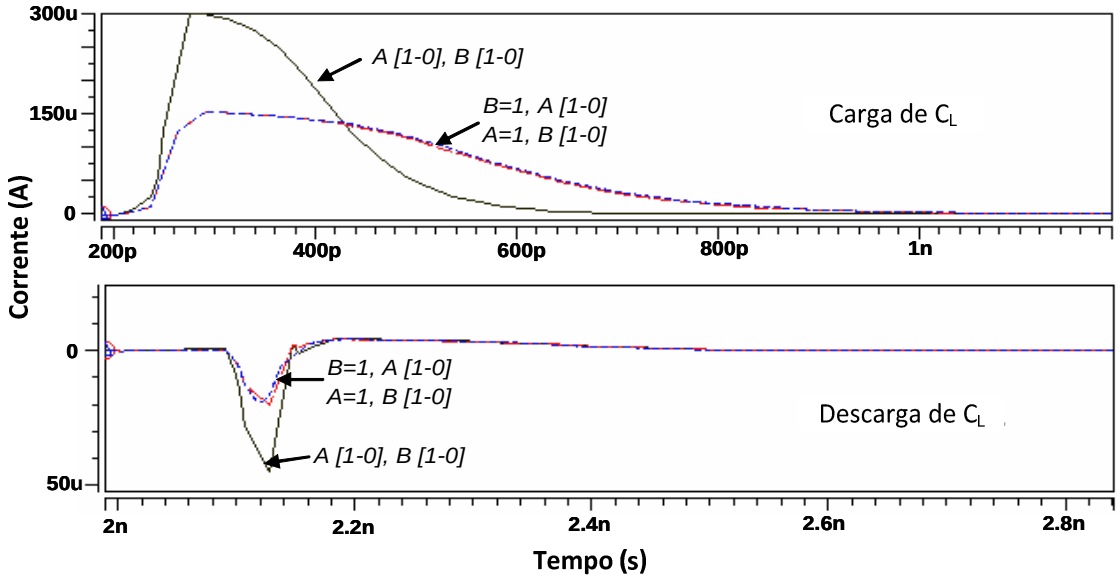


Figura 2.6 Avaliação da corrente segundo os vetores de entrada [RAZ06].

Como é possível observar na Figura 2.6 o consumo de potência dinâmica é maior quando ambas as entradas chaveiam simultaneamente do nível lógico 1 para 0. Neste caso, é possível relacionar o consumo de potência dinâmica de circuitos CMOS ao peso Hamming da porta de saída de dados do circuito. Por definição, peso Hamming (em inglês, *Hamming Weight* – HW) é o número de bits diferentes do nível lógico '0'. Logo, é

possível modelar o consumo de potência em circuitos CMOS segundo o peso Hamming das saídas do circuito em análise.

A fim de observar o impacto da atividade de um circuito sobre o traço de corrente, Razafindraibe et al. avaliam o consumo de corrente de um circuito composto por um grupo de portas NAND-2, conforme ilustra a Figura 2.7. As condições de controle, rampa de entrada (τ_{in}=100ps) e cargas capacitivas (C_L= 20fF) são mantidas constantes. Apenas a taxa de atividade (α) do circuito é variada, ou seja, a probabilidade de transição das portas a cada período de relógio. Neste estudo de caso, pressupõe-se que as simulações se desenvolvem durante um período de relógio e considera-se que no instante *t=0s*, todas as saídas são *'0'*. Nestas condições, a taxa de atividade é simplesmente o número de portas comutando dividido pelo número total de portas. O número de portas comutando representa seu peso Hamming.

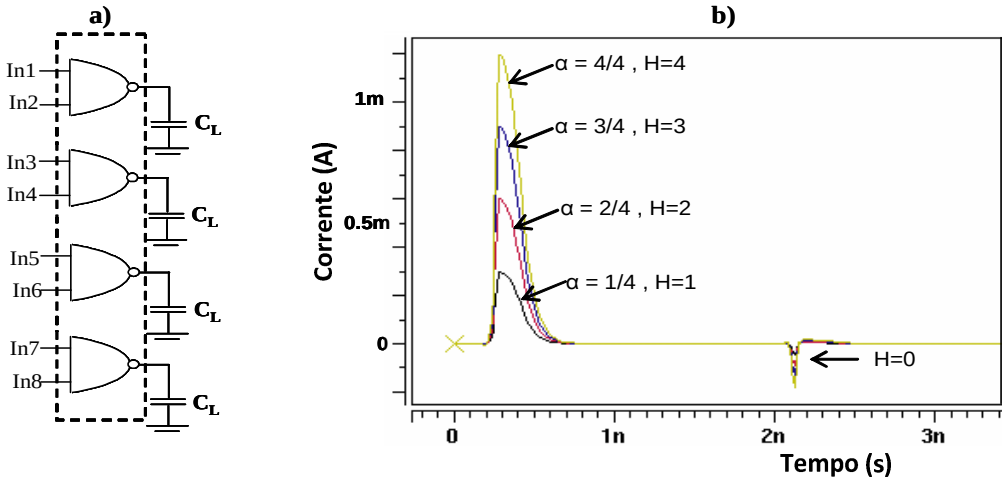


Figura 2.7 Descrição do experimento de observação do impacto da atividade de um circuito sobre o traço de corrente. a) Conjunto de portas NAND–2, b) avaliação da corrente em função da taxa de atividade α.

Como é de se esperar, quanto maior for a atividade do circuito maior será a amplitude de corrente, ou seja, a amplitude de corrente em um circuito é proporcional ao peso Hamming dos dados processados.

Através destes estudos de caso realizados por simulação, é possível notar que o traço de corrente de um circuito CMOS depende dos valores de cargas capacitivas, valores de rampa de entrada, do tipo de transições efetuadas, da atividade do circuito ou ainda do peso Hamming dos dados tratados. Conseqüentemente, uma análise detalhada dos traços de corrente pode permitir a descoberta dos dados manipulados. No caso da Figura 2.7 (b), uma simples observação das formas de onda de corrente permite concluir sobre os dados calculados. No domínio da criptoanálise, isto é conhecido como análise de potência simples ou SPA. Do ponto de vista de segurança é claramente estabelecido que o fundamento dos ataques por análise de consumo de potência é a dependência entre os dados manipulados e o traço de corrente. A mesma análise pode ser feita para circuitos

prototipados em FPGA, que por sua vez são fundamentalmente circuitos CMOS programáveis. Logo, estão sujeitos aos mesmos problemas de fuga de informação.

2.2.3 ATAQUES POR ANÁLISE DE POTÊNCIA SIMPLES

Introduzidos por Kocher et al. em [KOC99], os ataques por análise do consumo de potência exploram essencialmente duas dependências: a dependência de dados e a dependência de operações. Kocher et al. observaram que os traços de consumo de potência diferem para diferentes operações e para diferentes dados. Os ataques que exploram estas dependências realizando uma interpretação direta de traços de potência medidos durante operações criptográficas são referenciados como ataques SPAs. Estes ataques exigem um conhecimento detalhado sobre o modo de implementação do algoritmo criptográfico executado no dispositivo atacado, além de exigirem conhecimento do algoritmo criptográfico. SPAs são úteis quando apenas um traço ou poucos traços de potência estão disponíveis para um conjunto de dados de entrada. O ataque explora diferenças dentro de um traço causadas por dependências da chave.

Kocher et al. realizaram experimentos com o algoritmo DES para demonstrar a efetividade desta análise. Detalhes sobre este algoritmo estão disponíveis no Anexo B deste trabalho. A Figura 2.8 representa o traço de corrente correspondente a um cálculo criptográfico com o DES: a permutação inicial, as 16 rodadas e a permutação final são claramente identificáveis. Neste caso, os ataques SPAs possuem precisão para identificar as diferentes operações e suas instâncias de ocorrência.

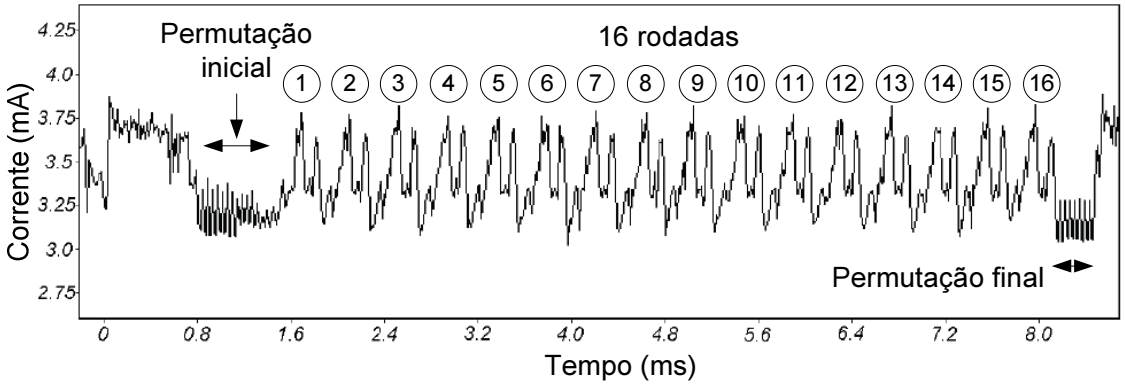


Figura 2.8 Traço de corrente correspondente ao processamento de um dado no algoritmo criptográfico DES.

Na análise de Kocher et al. [KOC99a], uma observação detalhada sobre as rodadas 2 e 3 permitiu identificar os traços de corrente dos blocos de geração de subchaves do algoritmo DES e concluir sobre diferentes operações tais como permutações e deslocamento à esquerda as quais a chave é submetida. Além disso, a cada operação de deslocamento, um teste sobre o valor de um bit da chave é efetuado. De acordo com o valor do bit (‘0’ ou ‘1’), o traço de corrente de um salto condicional difere ligeiramente. Por conseqüência, o atacante é capaz de concluir sobre o valor de certo número de bits da chave secreta. Nota-se que as implementações do DES em software executando sobre um processador síncrono sem contramedidas são vulneráveis a ataques SPA. Claro, o

sucesso destes ataques exige um conhecimento detalhado do algoritmo de criptografia e a maneira através da qual este é implementado.

Atualmente, com o surgimento de novas técnicas de contramedida, o ataque SPA não representa mais uma ameaça séria a criptosistemas. Por outro lado, é possível verificar que este ataque serve como uma etapa preliminar em análises diferenciais de potência.

2.2.4 ATAQUES POR ANÁLISE DIFERENCIAL DE POTÊNCIA

O ataque por análise diferencial de potência (DPA) é o mais popular ataque por consumo de potência. Isto se deve ao fato de não exigir conhecimento detalhado sobre o dispositivo atacado. Além disso, ele pode revelar a chave secreta de um criptosistema mesmo na presença de perturbações elétricas causadas durante o processo de medição dos traços de potência. Ao contrário de SPA, DPA exige um grande número de traços de potência para a análise.

A principal vantagem de DPA em relação a SPA é não precisar de conhecimento detalhado sobre o modo de implementação do algoritmo no dispositivo criptográfico. O conhecimento do algoritmo executado pelo criptosistema é suficiente para realizar a análise. Como tais algoritmos são tipicamente de domínio público, nota-se como ataques DPA podem ser efetivos na prática. Além disso, o modo como os traços são analisados difere em cada um dos ataques. Em SPA, o consumo de potência é analisado ao longo do eixo do tempo, ou seja, o atacante busca encontrar padrões de consumo em um único traço. Já em DPA, a forma do traço ao longo do eixo do tempo não é tão importante. DPA analisa como o consumo de potência em instantes fixos de tempo depende dos dados processados. Logo, é possível dizer que ataques DPA exploram as dependências de dados do consumo de potência dos criptosistemas. Por estas razões, estes se tornam mais eficientes e ameaçadores à segurança de criptosistemas que ataques SPA.

Ao contrário de SPA, ataques DPA empregam uma estratégia genérica que é usada em todos os ataques. Esta estratégia consiste de 5 etapas, descritas a seguir.

PASSO 1: ESCOLHER UM RESULTADO INTERMEDIÁRIO ALVO

A primeira etapa de um ataque DPA é escolher um resultado intermediário do algoritmo criptográfico que será alvo do ataque. Este resultado precisa ser uma função $f(d,k)$, onde d é um dado conhecido e k é uma parte da chave criptográfica secreta. Resultados intermediários que satisfazem esta condição podem ser usados para revelar k. Na maioria dos ataques, d é uma mensagem de entrada ou um criptograma de saída.

PASSO 2: MEDIR E COLETAR TRAÇOS

A segunda etapa de um ataque DPA é medir o consumo de potência do criptosistema enquanto este encripta ou decripta um conjunto de dados distintos D usando a mesma chave criptográfica. Para cada encriptação ou decriptação executada, o atacante

precisa conhecer o valor *d* correspondente ao dado que está envolvido no cálculo do resultado intermediário escolhido no Passo 1. Estes valores de dados conhecidos são definidos pelo vetor ***d*** = (d_1, ..., d_D)', onde d_i denota o valor do dado na $i^{ésima}$ execução de encriptação ou decriptação.

Durante cada uma destas execuções o atacante armazena o traço de consumo de potência correspondente. Estes traços são definidos como $t_i' = (t_{i,1},\ldots,t_{i,T})$, onde *T* denota o número de amostras de consumo de potência medido em cada traço. O atacante mede um traço para cada dado d_i contido no conjunto *D* de dados. Portanto os traços são armazenados em uma matriz M_T de tamanho D x T. É importante para os ataques DPA que os traços medidos sejam corretamente alinhados. Isto significa que os valores de consumo de potência de cada coluna t_j da matriz M_T devem corresponder à execução das mesmas operações realizadas durante a encriptação ou decriptação. Para obter o consumo de potência alinhado, o osciloscópio usado na medição deve ser disparado de modo que os traços de consumo de potência correspondam exatamente à mesma seqüência de operações durante cada encriptação ou decriptação executada.

PASSO 3: CALCULAR VALORES INTERMEDIÁRIOS HIPOTÉTICOS

O próximo passo do ataque é calcular o *valor intermediário hipotético* para todas as possibilidades de valores de *k,* de acordo com a função *f*(*d*,*k*). Estes valores são definidos pelo vetor **k** = (k_1,...,k_K), onde *K* denota o número total de possibilidades de *k*. No contexto de ataques DPA, é comum denominar este vetor como as hipóteses de chave. Dado o vetor de dados **d** e as hipóteses de chave **k,** um atacante pode facilmente calcular todos os valores intermediários hipotéticos possíveis para *f*(*d*,*k*). Estes cálculos mostrados genericamente pela Equação 2 resultam na matriz M_V de tamanho *D* x *K*. A primeira parte da Figura 2.9 ilustra esta etapa de cálculos.

$$v_{i,j} = f\left(d_i, k_j\right) \quad i = 1,\ldots, \text{D e } j = 1,\ldots, \text{K} \qquad (2)$$

Cada coluna *j* de M_V contém os resultados intermediários calculados com base na hipótese de chave k_j. É claro que uma coluna de M_V contém os valores intermediários reais calculados pelo criptosistema durante as execuções de encriptação ou decriptação realizada no Passo 2. Lembrando, o vetor **k** contém todas as escolhas possíveis para *k*. Portanto, o valor de chave *k* da função *f(d,k)* usado pelo criptosistema no Passo 2 é um elemento do vetor **k**. Define-se o índice deste elemento como *ck*. Portanto, k_{ck} é a chave realmente usada pelo criptosistema. O objetivo do ataque DPA é encontrar qual coluna de M_V contém os mesmos valores produzidos por *f(d,k)* durante a encriptação ou decriptação do vetor *D*.

PASSO 4: APLICAR MODELO DE CONSUMO AO DISPOSITIVO ATACADO

A próxima etapa do ataque DPA é aplicar um modelo de consumo de potência ao dispositivo atacado visando simular valores de consumo de potência a partir dos resultados hipotéticos obtidos no Passo 3, conforme mostrado na Figura 2.9.

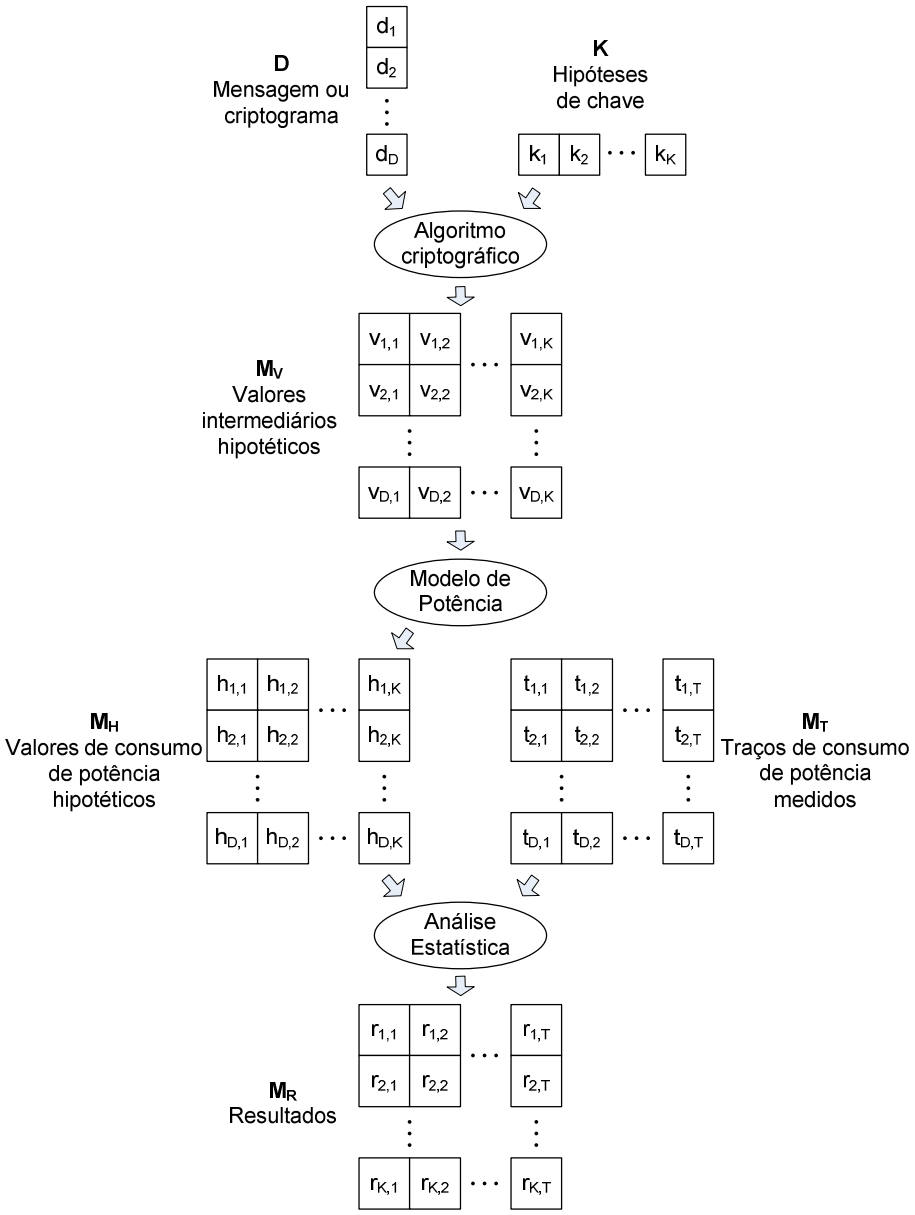


Figura 2.9 Diagrama em blocos ilustrando os passos 3 a 5 de um ataque DPA.

Como discutido anteriormente em nível de portas lógicas, o consumo de potência de um circuito desenvolvido com tecnologia CMOS é proporcional ao peso Hamming dos resultados obtidos em sua porta de saída de dados. Nesta etapa do ataque DPA, o atacante relaciona o consumo de potência do criptosistema com os pesos Hamming calculados a partir dos resultados hipotéticos intermediários $v_{i,j}$ obtidos no Passo 3. Entretanto, este modelo de consumo de potência não é muito adequado para descrever o consumo de um circuito CMOS, pois o consumo em circuitos depende também das transições de sinais internos e não apenas do resultado obtido. Este modelo é geralmente usado quando o atacante não conhece os dados consecutivamente aplicados à função escolhida no Passo 1.

Outro modelo de consumo de potência é a distância Hamming (HD). Neste caso, a idéia básica é contar o número de transições [0-1] e [1-0] que ocorrem em um circuito

digital durante certo intervalo de tempo. Este número de transições é usado para descrever o consumo de potência neste intervalo de tempo. Definindo de outro modo, a distância Hamming entre dois resultados intermediários correspondente a $HD = (r_0 \oplus r_1)$, onde r_0 e r_1 representam seus respectivos pesos Hamming. Em geral, o modelo HD pode ser usado para simular o consumo de potência de uma parte do criptosistema, tal como a função escolhida no Passo 1. Para isso, o atacante deve conhecer os valores dos dados processados consecutivamente no criptosistema atacado.

Isto não implica que simulações baseadas no modelo HW são inúteis. O consumo de potência em criptosistemas é apenas aproximado proporcionalmente ao número de transições que ocorrem no dispositivo. O modelo HD assume que transições [0-1] e [1-0] apresentam o mesmo consumo de potência. Na prática, estas transições apresentam consumos diferentes como mostrado pelos experimentos de Razafindraibe et al. discutidos na Seção anterior. Neste caso, é possível assumir que em média, o consumo de potência é maior quando são obtidos resultados com valores de HW maiores em relação a valores HW menores.

Estes modelos são os mais utilizados para modelar o consumo de potência em ataques DPA. Porém, outros modelos são também propostos para dispositivos com características específicas. Tais modelos podem ser derivados dos modelos HD. O modelo HD, por exemplo, assume que todos os *n* bits de um dispositivo contribuem igualmente para o consumo de potência, assumindo que as cargas dos *n* bits são iguais. Mas se um atacante sabe que alguns bits consomem mais que outros, isto pode ser considerado no modelo, de modo a estender o modelo HD e torná-lo mais específico para um dado dispositivo.

Outra possibilidade de modelar o consumo de potência com base no modelo HD é introduzir pesos diferentes para diferentes transições ocorridas em um dispositivo. Por exemplo, a transição [0-1] pode ter um peso equivalente ao dobro da transição [1-0], já que experimentos como os de Razafindríbe et al. mostram que estas transições produzem consumos diferentes.

Kocher et al. em [KOC99], simulam o consumo de potência do dispositivo atacado usando um modelo binário, ou seja, os resultados intermediários hipotéticos $v_{i,j}$ produzem coeficientes binários $h_{i,j} \in \{0,1\} \forall i, j$ para a matriz M_H, conforme mostrado na Figura 2.9. Analisando apenas um bit de cada resultado intermediário hipotético $v_{i,j}$, se o bit corresponde ao valor lógico ‘1’ apresenta um consumo maior, logo $h_{i,j} = 1$. Caso contrário, se o bit corresponde ao valor lógico ‘0’ apresenta um consumo menor, e assim o coeficiente correspondente é $h_{i,j} = 0$.

Portanto, usando um destes modelos, o consumo de potência do criptosistema para cada valor intermediário hipotético $v_{i,j}$ é simulado de forma a obter-se um valor de consumo de potência hipotético $h_{i,j}$. A qualidade da simulação depende fortemente do conhecimento do atacante sobre o criptosistema analisado. A melhor simulação é aquela que mais se aproxima das características do consumo de potência do criptosistema atacado.

PASSO 5: AVALIAR HIPÓTESES DE SUBCHAVES

Depois de calcular os valores intermediários hipóteses M_V e obter os respectivos valores de consumo de potência M_H a partir de um dado modelo de potência, a última etapa do ataque tem como objetivo avaliar as hipóteses de chave. Neste Passo, cada coluna h_j de M_H é comparada com cada coluna de t_j da matriz M_T. Isto significa que o atacante compara os valores de consumo de potência hipotéticos de cada hipótese de chave com os traços coletados. O resultado desta comparação é a matriz M_R de tamanho K x T, onde cada elemento $r_{i,j}$ contém o resultado da comparação entre as colunas h_j e t_j. A comparação é feita com base no método da diferença das médias, conforme proposto por Kocher et al. em [KOC99]. Outros métodos para avaliação das chaves são também possíveis tal como o uso de coeficientes de correlação proposto por Brier et al. [BRI04].

Os traços de potência correspondem ao consumo de potência do dispositivo enquanto este executa um algoritmo criptográfico usando diferentes dados de entrada. O resultado intermediário escolhido no Passo 1 é uma parte deste algoritmo. Portanto, o dispositivo precisa calcular o valor intermediário v_{ck} durante as diferentes execuções do algoritmo. Conseqüentemente, os traços coletados dependem destes valores intermediários em um mesmo instante de tempo. Este instante no traço de potência é referenciado como *ct*, ou seja, a coluna t_{ct} contém os valores de consumo de potência que dependem dos valores intermediários v_{ck}.

Os valores de consumo de potência hipotéticos h_{ck} são simulados pelo atacante com base nos valores v_{ck}. Portanto, as colunas h_{ck} e t_{ct} são fortemente relacionadas. Estas duas colunas conduzem a um valor alto em M_R, ou seja, o maior valor da matriz M_R é o valor $r_{ck,ct}$. Todos os outros valores são menores porque as colunas de M_H e M_T não são fortemente relacionadas. Um atacante pode revelar o índice da chave correta ck e o instante de tempo ct por simplesmente observar o valor mais alto na matriz M_R. Os índices deste valor são então o resultado do ataque DPA.

Kocher et al. propuseram o método da diferença das médias para avaliar as hipóteses de chaves. Este método estabelece uma relação entre as colunas das matrizes M_H e M_T com base na seguinte observação. Analisam a seqüência de zeros e uns dos coeficientes $h_{i,j}$ calculados no Passo 4. Para verificar se uma hipótese de chave K_i está correta ou não, o atacante divide a matriz M_T em duas matrizes M_{T0} e M_{T1} de acordo com os valores de h_i. A primeira matriz M_{T0} contém as linhas da matriz M_T cujos coeficientes $h_{i,j}$ são zeros. A segunda matriz M_{T1} contém todas as linhas restantes de M_T. A seguir, a média das linhas deve ser calculada, para da uma das matrizes. O vetor m'_{0i} denota a média das linhas da matriz M_{T0} e m'_{1i} denota a média das linhas da matriz M_{T1}. As hipóteses de chave k_i são corretas se ocorrer uma diferença significativa entre m'_{0i} e m'_{1i} no mesmo instante de tempo.

A diferença entre m'_{0i} e m'_{1i} indica que existe uma correlação entre h_{ck} e as colunas de M_T. Esta diferença ocorre exatamente no instante em que o valor intermediário que corresponde a h_{ck} é processado. Em todos os outros instantes a diferença entre os vetores é aproximadamente zero. No caso de uma hipótese não correta, a diferença entre m'_{0i} e

m'_{1i} é próxima de zero em todos os instantes de tempo. O resultado de um ataque DPA baseado no método da diferença é uma matriz M_R onde cada linha de M_R corresponde às diferenças entre as médias dos vetores m'_{0i} e m'_{1i} de uma hipótese de chave.

É importante destacar a possibilidade de todos os valores contidos em M_R serem aproximadamente os mesmos. Neste caso, o atacante não coletou uma quantidade suficiente de traços para estimar a relação entre as colunas de M_H e M_T. Quanto maior a quantidade de traços, mais elementos estão nas colunas, conseqüentemente mais preciso será o ataque. Isto também implica que quanto mais medições e coletas de traços sejam feitas, as menores relações entre as colunas podem ser determinadas.

2.3 REVISÃO DE PROPOSTAS DE ATAQUES DPA E VARIANTES

Os ataques DPA revisados e discutidos até então tem a propriedade de explorar apenas um valor intermediário do algoritmo alvo para revelar a chave criptográfica, pois apenas os dados de saída da primeira rodada ou os dados de entrada da última rodada são considerados durante as análises. Estes ataques são também referenciados na literatura como *ataques de primeira ordem*. Se vários valores intermediários são considerados para formular as hipóteses de chave, então os ataques são referenciados como *ataques de alta ordem* (do inglês, *High Order DPA - HO DPA*).

Prouff et al. em [PRO09] propuseram uma versão especializada do ataque DPA, chamada de DPA de segunda ordem (do inglês, *Second Order DPA*, ou SO-DPA). Esta proposta é especializada em atacar criptosistemas protegidos por métodos de mascaramento de dados. Esta análise combina as fugas de informações causadas pelo processamento e pelo mascaramento de dados, relacionando-as a fim de encontrar a chave secreta conforme proposto também em [MES00] [WAD04] [PEE05] e [OSW06].

Bevan e Knudsen [BEV03] propõem melhorias no ataque proposto por Kocher [KOC99]. Os Autores apresentam justificativas para a ocorrência de picos em hipóteses incorretas de subchaves e propõem utilizar estas informações de modo a tornar os ataques mais eficientes, ou seja, reduzir o número de traços necessários para revelar a chave secreta. Esta eficiência é obtida através de *ataques multi-bits*, onde a função seleção proposta por Kocher não é aplicada apenas a um bit por ataque e sim a combinações de dois ou mais bits. Deste modo, os consumos de potência são somados tornando os ataques mais rápidos e eficientes conforme demonstram seus resultados.

Brier et al. em [BRI04] propuseram outra especialização do ataque DPA denominada Análise por Correlação de Potência (do inglês, *Correlation Power Analysis* - CPA). CPA é uma análise DPA que emprega um modelo linear de consumo de potência aplicado sobre os dados manipulados. Esta abordagem basicamente visa reduzir o problema de *picos fantasmas* durante os ataques DPA. Classicamente utiliza-se o modelo peso Hamming para correlacionar dados manipulados e o consumo de potência conforme revisado na Seção 2.2.1. Neste caso, utiliza-se o modelo de potência distância Hamming, cujo cálculo é realizado segundo a variação do número de bits com valor ‘1’ entre uma mensagem *m* e uma dada mensagem de referência *mr* estimada previamente segundo

Brier et al. em [BRI04]. Logo, o consumo de potência pode ser resumido pela equação $W = aH(m \oplus mr) + b$, onde '*a*' é um ganho escalar entre a distância HD '*H*' e a potência consumida '*W*'. Já *b* inclui todas as variações de consumo decorrentes dos elementos que compõem o restante do circuito e que são independentes dos dados manipulados tais como variações de tensão (offsets), atrasos intrínsecos aos componentes e ruído. Com base nesta Equação são obtidos fatores lineares de correlação que relacionam as variâncias dos termos considerados com a potência consumida medida. Estes fatores de correlação são capazes de rejeitar falsas hipóteses de subchaves segundo o modelo de potência HD adotado como mostrado em [BRI04]. Estas análises são utilizadas para avaliar a robustez das arquiteturas propostas neste trabalho.

Fahn e Pearson em [FAH99] propuseram a análise de potência por inferência (do inglês, *Inferential Power Analysis* - IPA). Esta análise se desenvolve basicamente em duas etapas. Inicialmente realizam-se operações estatísticas tais como diferenciação entre traços de consumo e médias entre outras, aplicadas sobre uma grande quantidade de traços de consumo a fim de aprender detalhes de implementação, o que conduz à localização e identificação dos bits da chave criptográfica. A etapa seguinte extrai a chave por inferência, a partir de poucos traços de potência, conforme [FAH99].

Chari et al. em [CHA02] introduziram os ataques denominados em inglês *Template Attacks*. Estes ataques probabilísticos assumem um modelo Gaussiano de ruído para definir e registrar os templates de traços DPAs relativos a um conjunto pré-definido de operações. As informações são aprendidas e registradas segundo parâmetros tais como a média e uma matriz de covariância as quais são otimizadas para cada operação, de acordo com o princípio da probabilidade máxima. A seguir, o ataque propriamente dito ocorre quando um traço relativo a uma operação tem suas propriedades estatísticas comparadas ao repositório de templates através de regras Bayesianas. Estas regras classificam o traço de acordo com as probabilidades e atribuem ao traço a operação correspondente. Pressupõe-se que o repositório de templates e os traços relativos ao canal lateral analisado sejam obtidos através do mesmo dispositivo.

Goubin em [GOU03] apresenta uma análise de potência refinada (do inglês, *Refined Power Analysis* - RPA) que permite revelar a chave criptográfica de algoritmos que se baseiam na estrutura algébrica de curvas elípticas tal como o algoritmo ECC (do inglês, *Eliptic Curve Cryptography*) mesmo na presença de algumas contramedidas. Outra técnica proposta por Akishita e Takagi em [AKI03] explora uma característica especial das curvas elípticas onde o consumo é nulo. Este método é uma extensão de RPA proposto por Goubin e é definido pelos Autores como análise de potência nula (do inglês, Zero-value Point Attacks - ZPA). Mesmo em pontos da curva usados para a encriptação de dados onde não exista coordenadas com valor nulo, o método proposto explora registradores laterais que podem obter o valor nulo e sobrepor algumas possíveis contramedidas. Logo, estes tipos de análises não são uma ameaça à segurança de algoritmos tais como DES e AES.

Nagashima et al. em [NAG07] propuseram um método para realizar ataques DPA em criptosistemas que usam inserção de atrasos aleatoriamente como contramedida. O

método proposto realiza uma ressincronização de formas de onda de diferentes fases como pré-processamento ao ataque DPA. Os Autores usam funções matemáticas normalmente usadas em tratamento de sinais, tais como a Transformada Discreta de Fourier para definir parâmetros que identificam a diferença de fase entre curvas de potência. Embora seja uma interessante ferramenta de criptoanálise, o método limita-se apenas a analisar o deslocamento de tempo que um traço de consumo de potência sofre durante uma encriptação completa.

2.3.1 DISCUSSÃO SOBRE OS ATAQUES

Uma revisão dos avanços em ataques DPA é realizada e resumida na Tabela 2.1. Depois que Kocher et al. propuseram os ataques TA, SPA e DPA, estes ataques sofreram algumas modificações ou receberam etapa de pré-processamento, porém sua essência permanece a mesma.

Tabela 2.1 Revisão de ataques DPAs e variantes.

	Ataque proposto	**Método**	**Proposta desta Tese**
Kocher et al. [KOC99]	SPA e DPA	SPA – simples observação do consumo de potência DPA – análise diferencial e estatística do consumo de potência	Não vulnerável a estes ataques, até onde se testou
Prouff et al. [PRO09]	SO-DPA	DPA na presença de mascaramento de dados	Não usa mascaramento de dados. Não vulnerável a este ataque
Brier et al. [BRI04]	CPA	DPA com uso de modelo de potência	Não vulnerável a estes ataques, até onde se testou
Fahn e Pearson [FAH99]	IPA	Modelos estatísticos para identificar fugas de informação	Não testado. Acredita-se na dificuldade em detectar padrões de consumo
Chari et al.[CHA02]	Templates	Ferramentas estatísticas para identificar fuga de informação	Não testado. Acredita-se na dificuldade em detectar padrões de consumo
Goubin [GOU03]	RPA	Propriedades de curvas elípticas	Não vulnerável
Akishita e Takagi [AKI03]	ZPA	Propriedade de curvas elípticas	Não vulnerável
Nagashima et al. [NAG07]	Phase-match DPA	Transformada de Fourrier (tratamento de sinais)	Não testado. Acredita-se que seja efetivo apenas para variações de fase simples em sinais

As criptoanálises que se baseiam em métodos probabilísticos tais como IPA e ataques por templates apresentam resultados apenas em ensaios com criptosistemas sem contramedidas. Embora usem ferramentas e modelos estatísticos eficientes, na prática a aleatoriedade inserida pelas arquiteturas propostas neste trabalho representa um desafio a este tipo de criptoanálise.

Os métodos RPA, ZPA são criptoanálises propostas para algoritmos que se baseiam em propriedades algébricas de curvas elípticas que por sua vez fogem o escopo do presente trabalho. Já SO-DPA é proposta como um método de ataque DPA capaz de correlacionar mensagens, mensagens mascaradas e criptogramas para encontrar a chave secreta do sistema. Como o trabalho proposto não usa este tipo de contramedida, esta abordagem não é adequada para realizar avaliação da robustez, tais como RPA e ZPA.

Os ataques propostos por Nagashima et al. [NAG07] e Clavier et al. [CLA00] embora sejam especializados em criptoanalisar sistemas protegidos pela abordagem de inserção de aleatoriedade para ocultar a fuga de informações, os estudos de casos apresentados mostram ataques bem sucedidos apenas à inserções de aleatoriedade simples tal como instruções *dummies* (nenhuma operação realizada) e deslocamento temporal do processo de encriptação integral. No presente trabalho a execução do algoritmo usando a arquitetura GALS com domínios de freqüência aleatórios representa um cenário desafiador para estes tipos de ataques. Acredita-se que estas abordagens juntamente com métodos de ressincronização de curvas sejam o caminho para quebrar o sigilo de criptosistemas que empreguem aleatoriedade do modo como é proposto aqui.

2.4 ATAQUES POR INDUÇÃO A FALHAS

Outra técnica importante de criptoanálise é o ataque por indução a falhas em criptosistemas. Um atacante pode induzir falhas durante a computação do criptosistema e explorar o resultado defeituoso para extrair informação sobre a chave secreta. As falhas podem ser caracterizadas como permanentes, se o dano causado ao dispositivo é definitivo, tal como forçar uma posição de memória para um valor constante, ou podem ser transientes, se o dispositivo sofrer distúrbios durante seu funcionamento em operações específicas. Como exemplo de modos de inserção de falhas é possível citar o bombardeio radioativo, criar anormalidades na freqüência do relógio ou na tensão de alimentação do dispositivo, a geração de *glitchs* no barramento de dados, o aumento da temperatura, a emissão de luz sobre o silício e a imersão do circuito em campos eletromagnéticos. Este método pode ser classificado como semi-invasivo ou não-invasivo dependendo de como a inserção de falhas no dispositivo é realizada.

Biham e Shamir [BIH90] propuseram análises diferenciais de falhas (do inglês, Differential Fault Analysis - DFAs) destinadas a algoritmos de encriptação simétricos. Os Autores usam o modelo de falha transiente aplicado aos registradores de uma implementação do algoritmo. O método proposto é aplicado durante as operações de encriptação/decriptação do criptosistema, causando a inversão do valor de um bit com uma pequena probabilidade de ocorrência, comportando-se como uma falha transiente.

As análises DFAs apresentam-se como um método eficiente e de baixo custo para atacar criptosistemas. Estes métodos empregam, por exemplo, a variação da tensão de alimentação ou da freqüência do relógio do circuito. O estudo e avaliação da robustez segundo este método não faz parte do escopo do presente trabalho.

3. ESTADO DA ARTE

Na literatura encontram-se várias propostas para contramedir ataques por análise do consumo de potência a criptosistemas. Estas propostas podem ser agrupadas em três grupos de acordo com a abordagem usada para imunização. A primeira abordagem consiste em introduzir ruído nas medidas de consumo de potência. A idéia básica aqui é reduzir a relação sinal-ruído, de modo a impossibilitar ou inviabilizar na prática tentativas de correlacionar o consumo do criptosistema com os dados processados. Uma segunda abordagem consiste em mascarar os dados processados, e assim inviabilizar a realização de análises de correlação. Finalmente, a terceira abordagem concentra esforços para obter um consumo de potência equilibrado ou uniforme para qualquer seqüência de valores de dados de entrada.

Na literatura encontram-se também trabalhos específicos a fim de imunizar criptosistemas a ataques por SEMA/DEMA [GEB05] e indução a falhas [MAI08] [BHA09]. Como prevenção a ataques por indução a falhas não é o alvo deste trabalho, a revisão destes temas é apresentada de forma mais sucinta, apenas visando apresentar aos leitores outra área relevante de pesquisa. Já os ataques SEMA/DEMA por serem mais recentes possuem poucos trabalhos propondo métodos de prevenção. Além disso, como a radiação eletromagnética está diretamente relacionada à variação de corrente no circuito, os métodos propostos para imunizar circuitos a ataques por consumo de potência também imunizam ou no mínimo reduzem as fugas de informações através das radiações eletromagnéticas [LOM09]. No restante deste Capítulo apresenta-se uma revisão de propostas subdivididas de acordo com o tipo de abordagem usada para imunizar criptosistemas contra ataques DPA.

3.1 MÉTODOS POR MASCARAMENTO DE DADOS

O objetivo de toda contramedida é tornar a fuga de informação por canais secundários de um sistema criptográfico independente dos dados processados. O mascaramento atinge este objetivo através da aleatorização dos valores intermediários processados pelo sistema. Uma vantagem desta abordagem é poder ser implementada no nível do algoritmo sem mudar as características de consumo do dispositivo. Ou seja, mesmo o dispositivo apresentando consumo de potência dependente de dados, o método consegue descorrelacionar o consumo dos dados manipulados. Apesar disso, na literatura são encontradas propostas de mascaramento aplicadas em nível de porta lógicas tal como sugerido em [POP06] e [GOL07]. A seguir são revisadas diferentes propostas para mascaramento de dados.

Pramstaller et al. em [PRA04] propõem um método para mascarar dados no algoritmo AES e o comparam com outras duas propostas. O principal desafio encontrado em ambas as propostas é evitar a fuga de informações na operação SubBytes realizada por SBOXes paralelos, que são funções booleanas não-lineares alvo dos ataques SCA, maiores detalhes no Anexo B. Cada SBOX realiza operações sobre uma porção de 8 bits de dados de entrada, em cada uma das rodadas do AES. SubBytes realiza

transformações não-lineares em campos de Galois (do inglês, *Galois Field* - GF). As transformações consistem em um inverso multiplicativo[1] em GF(2^8) e uma transformação afim[2]. O primeiro trabalho revisado, proposto por Akkar e Giraud [AKK01] realiza uma operação de soma de 128 bits entre dado e máscara (um número gerado aleatoriamente), operação denominada *máscara aditiva*. A máscara aditiva é removida antes da inversão de bytes e substituída por uma máscara multiplicativa (operação de multiplicação envolvendo o dado e a uma nova máscara) também de 128 bits. Após a inversão de byte, a máscara multiplicativa é removida e a máscara aditiva é re-introduzida. Este método introduz custo em área significativo, pois é necessário gerar duas máscaras aleatórias por operação de encriptação/decriptação. Este método é vulnerável a ataques conhecidos como Valor Zero. Tais ataques exploram um caso específico onde os valores parciais do dado e da chave são iguais [GOL02]. Na proposta de Trichina et al. [TRI03], a máscara aditiva é reusada, excluindo o uso da máscara multiplicativa. Isto reduz consideravelmente o custo de cálculo. Entretanto, uma nova máscara é exigida para cada execução da rodada para garantir que a máscara não seja zero. Por outro lado, isto não soluciona o problema do valor zero. A proposta de Pramstaller et al. [PRA04] é denominada de IAIK pelos Autores . Este método não remove a máscara aditiva antes da inversão de byte. A saída deste passo é portanto (Dado + Máscara)$^{-1}$, o multiplicativo inverso do dado mascarado. O método IAIK obtém o resultado esperado (Dado^{-1} + Máscara) por calcular termos de correção em paralelo. Aritmética sobre GFs é usada extensamente nesta abordagem e a inversão de byte em GF(2^8) é primeiro mapeada para GF(2^4) e novamente para GF(2^2). Neste campo, a inversão pode ser computada efetivamente. A abordagem IAIK é a única imune a ataques por Valor Zero. Um criptosistema foi desenvolvido sob a forma de um ASIC em tecnologia CMOS de 0,25 µm empregando o algoritmo AES com a abordagem IAIK. O custo para se obter uma maior resistência a ataques DPA foi uma redução entre 40 e 50% no desempenho devido à inserção dos métodos de segurança.

Mesquita et al. em [MES06] propõem o projeto de um criptosistema como uma arquitetura reconfigurável que mascara o consumo de potência. Os Autores exploram propriedades das funções modulares de cálculo do algoritmo RSA. Este algoritmo divide seu processamento em operações de multiplicação e exponenciação. Os Autores propõem o uso da multiplicação modular proposta por Montgomery, que permite o uso do sistema numérico residual (em inglês, *Residue Number System* - RNS) baseado no teorema chinês do resto (em inglês, *Chinese Remainder Theorem* - CRT). Deste modo, o módulo de multiplicação é capaz de realizar cálculos em diferentes bases numéricas. Isto produz um mascaramento dos dados processados e conseqüentemente o descorrelacionamento com o consumo de potência. A arquitetura proposta, denominada de Arquitetura Reconfigurável Resistente a Fuga (do inglês, *Leak Resistant Reconfigurable Architecture* – LR^{2}A) permite que em tempo de execução o módulo de multiplicação tenha sua base alterada aleatoriamente. A arquitetura sofre penalidades em área, mas apresenta um bom desempenho com relação a criptosistemas existentes.

---

[1] Na matemática, o inverso multiplicativo de um número x é denotado por $1/x$ ou x^{-1}. Este número quando multiplicado por x produz a identidade multiplicativa, ou seja, 1.

[2] Em geometria, uma transformação afim entre dois espaços vetoriais consiste de uma transformação linear Ax seguida por uma translação $+b$, ou seja, $x \rightarrow Ax + b$.

Popp e Mangard, em [POP06] propõem um novo estilo lógico para a concepção de criptosistemas resistentes a ataques DPA, aplicando mascaramento de dados em nível de portas lógicas. O estilo, denominado pelos Autores de Lógica Pré-Carregada com Trilha Dupla Mascarada (em inglês, *Masked Dual Rail Pre-charge Logic* - MDPL), permite ao criptosistema um consumo de corrente aleatório, tendo como principal benefício a não necessidade de balancear o circuito complementado. A idéia do mascaramento é tornar aleatório cada resultado intermediário de um circuito ou algoritmo aplicando a equação $(dm = d \oplus m)$, onde respectivamente *dm* é o dado mascarado, *d* o dado propriamente dito e *m* uma máscara aleatória. Os experimentos realizados mostram elevados custos de potência e área comparados a circuitos projetados em lógica CMOS tradicional. Motivados por estes resultados os Autores adicionam técnicas de redução de potência aos circuitos MDPL. Durante os períodos em que os circuitos não desempenham operações consideradas críticas a ataques DPA, estas técnicas reduzem em torno de 4 vezes o consumo de potência em relação a um circuito CMOS tradicional equivalente.

Ghosh et al. em [GHO07] motivam-se pelo fato de que circuitos de mascaramento usados para evitar fuga de informações através de SBOXes do AES apresentam vulnerabilidades devido à ocorrência de *glitchs*. Os Autores propõem um método de mascaramento para circuitos multiplicadores de SBOXes AES. Neste método, as entradas dos multiplicadores são sincronizadas por componentes seqüenciais ou portas lógicas *AND* controladas, visando evitar a fuga de informações. Este trabalho destaca que a principal desvantagem das arquiteturas de mascaramento anteriormente propostas é a comutação desbalanceada dos circuitos. A propagação dos sinais de entrada através de caminhos com diferentes comprimentos faz com que as portas lógicas "*XOR*" realizem operações bit a bit com diferentes temporizações. Isto provoca *glitchs* internos, responsáveis por fuga de informações. A arquitetura proposta possui dois propósitos: reduzir os *glitchs* internos do circuito e reduzir o atraso no pior caso. Os Autores propõem duas abordagens de implementação. A primeira é uma estrutura *pipeline* de dois estágios baseada na aplicação de elementos seqüenciais para sincronizar sinais diferentes. Na segunda abordagem, os Autores introduzem elementos combinacionais extras para sincronizar as entradas, de modo a terem mesmo tempo de propagação. Em ambas as abordagens, os Autores organizaram as portas *XOR* internas em uma estrutura balanceada, para garantir que as entradas de cada porta comutem uma vez a cada ciclo de relógio. Simulações SPICE confirmam que as modificações evitam ataques DPA.

Huiping et al. [HUI07] propõem outro método para mascaramento de dados no algoritmo DES. Os Autores introduzem no algoritmo operações lógicas e aritméticas (*XOR* e *soma*) realizadas entre dados e números gerados aleatoriamente, para produzir o mascaramento de dados em duas partes do algoritmo. Estas operações são realizadas na função *f* da rodada, no momento da entrada da subchave e antes da execução das SBOXes, conforme mostra a Figura 3.1. Para aumentar o desempenho do algoritmo implementa-se um *pipeline* com a replicação das rodadas do algoritmo. Os Autores propõem 4 configurações diferentes de *pipelines* com o DES, de modo a analisar o desempenho do processamento. A partir dos resultados obtidos concluem que o *pipeline* com 16 estágios (uma rodada executada em cada estágio) obtém o melhor desempenho,

porém o maior custo em área. O circuito é implementado com tecnologia CMOS 0,25µm e possibilita operações em freqüências de até 100 MHz, atendendo a exigências de projetos de smart cards.

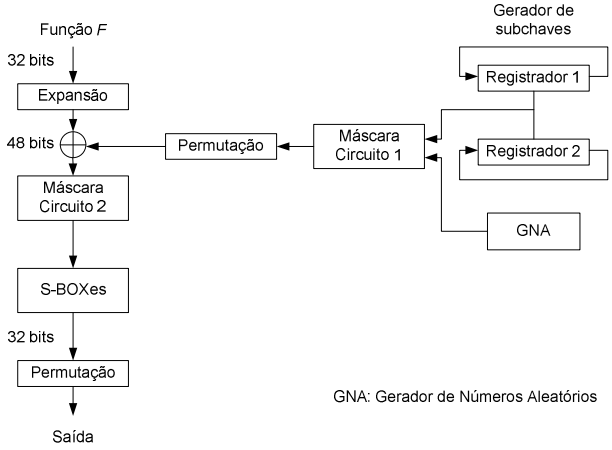


Figura 3.1 - Diagrama em blocos da função *F* da rodada do algoritmo DES. Huiping et al. propõem a inserção de circuitos para mascarar dados antes do processamento de partes vulneráveis da função (XOR e SBOXes). Um gerador de números aleatórios é inserido em cada rodada para alimentar os circuitos propostos.

Ordu e Örs em [ORD07] são os primeiros a apresentar uma implementação em FPGA do algoritmo AES usando métodos para mascarar dados. Os Autores propõem dois métodos diferentes, mascaramento aditivo de IAIK [OSW05] e mascaramento multiplicativo de Akkar [AKK01]. Os autores realizaram três implementações do algoritmo AES: (i) sem contramedidas; (ii) AES com método de Akkar e; (iii) AES com método de IAIK. Comparações de custos em área e desempenho foram realizados no dispositivo FPGA Virtex-E 1000 da Xilinx. Os resultados indicam que a implementação utilizando o método IAIK apresentou menor custo em área e menor período de relógio em relação ao método proposto por Akkar.

Haijun et al. em [HAI07] analisam várias contramedidas a ataques DPA e concluem que o método de mascaramento único (em inglês, *Unique Masking Method* - UMM) não é eficiente para proteger criptosistemas contra ataques DPA de alta ordem. Os Autores apresentam um novo método de proteção baseado em circuitos projetados com lógica diferencial dinâmica (em inglês, *Simple Dynamic Differential Logic* - SDDL) proposta em [TIR04], também conhecida na literatura como lógica de trilha dupla com dois estágios de funcionamento, pré-carga e avaliação (em inglês, *Dual Rail Precharge Logic* - DPL). Maiores detalhes sobre tais lógicas encontram-se na Seção 3.3. Em um estudo de caso usando o algoritmo DES, o método propõe implementar em lógica SDDL as duas primeiras e duas últimas rodadas do algoritmo e manter o mascaramento UMM nas rodadas restantes, 3 a 14, visando aumentar a resistência a ataques DPA. Os resultados demonstram que o método proposto aumenta a robustez de UMM a ataques DPA de alta ordem ao custo de um aumento de 29,3% no consumo de potência em relação à implementação do algoritmo sem contramedidas.

Golic em [GOL07] propôs um novo método para mascaramento de dados em hardware no nível de portas lógicas. Operações de mascaramento realizadas tanto em software quanto em hardware no nível de palavras não são eficientes para imunizar sistemas a ataques DPA, conforme demonstra o Autor. No nível de bit, as operações podem ser desbalanceadas, causando *glitches* durante o processamento e sendo dependente dos dados de entrada. O método proposto é classificado pelo Autor como *mascaramento lógico*,e pode ser aplicado a algoritmos criptográficos em geral e a técnicas de conversão de operações lógicas e aritméticas de mascaramento. O método proposto é baseado no uso de multiplexadores (MUXs), destinados a balancear em nível de bit operações de mascaramento. Mostra-se a possibilidade de implementação do método usando apenas portas lógicas NAND. Como vantagens, o método apresenta uma redução em área e latência em relação a técnicas de mascaramento similares tal como [GOL04]. Por outro lado, o método mostra-se eficiente apenas a ataques DPA de primeira ordem, sendo vulneráveis a ataques de alta ordem (HO DPA).

Ghellar e Lubaszewski em [GHE08] propõem uma nova implementação do algoritmo criptográfico AES para resistir a ataques DPA. O método proposto visa mascarar os dados com base em propriedades da estrutura de campos de Galois, mais precisamente em GF(2). Resumidamente, campos de Galois possuem elementos representados por dois algarismos (0 e 1), permitem apenas duas operações binárias (soma e multiplicação) e, para o caso específico $GF(2^8)$, contém 256 elementos. Uma propriedade desta estrutura é o fato de que campos GF(n) com mesmo número de elementos são isomórficos entre si. Deste modo, o mascaramento de dados é realizado através da definição de uma função de mapeamento para converter representações existentes em módulos da rodada do AES. Como vantagem do método, destaca-se um aumento pelo fator de 240 na complexidade dos ataques. Em contrapartida, o hardware adicional aumenta 295% em área e reduz a freqüência de operação em 60% em relação ao algoritmo original.

Goodwin e Wilson em [GOO08] apresentam uma modificação simples na implementação do algoritmo AES sem alterar suas características de funcionamento como forma de mascarar dados e evitar ataques DPA. No AES, a chave criptográfica é expandida inicialmente e a cada nova rodada de execução do algoritmo utiliza-se o valor da chave da rodada anterior. No sistema proposto pelos Autores um módulo de controle gera chaves a cada rodada do AES. O método proposto foi implementado em FPGA e os resultados mostram um aumento da robustez a um custo baixo em área em relação ao algoritmo sem contramedida. Por outro lado, a complexidade do módulo de geração empregado reduz significativamente a freqüência máxima de operação do relógio e por conseqüência causa uma redução de 30% na vazão do criptosistema.

3.1.1 COMPARAÇÃO ENTRE PROPOSTAS

Ao analisar as propostas aqui discutidas cujo resumo encontra-se na Tabela 3.1, é possível concluir que o uso de mascaramento em nível algorítmico causa um aumento considerável na latência de processamento, sofrendo também penalidades quanto ao consumo de potência e área. O uso desta contramedida em nível de circuito também eleva

os custos em área e o tempo de execução. A título de exceção é possível citar [POP06]. Apesar da lógica proposta impor um aumento considerável no consumo de potência, os Autores conseguem adaptar estratégias que reduzem substancialmente o consumo de potência do circuito. O método proposto nesta tese usa inserção de aleatoriedade e ruído no processamento para ocultar a fuga de informações, diferenciando-se dos demais trabalhos que visam modificar os dados de entrada durante o processamento de operações do algoritmo vulneráveis a ataques. Conforme revisado, a presente tese apresenta estudos de caso com o algoritmo DES como os trabalhos de [HUI07] e [HAI07] e protótipos em FPGA conforme [MES06], [ORD07] e [GOO08]. Os custos da segurança obtida pelos métodos propostos e pela tese aqui proposta se resumem a um aumento de área e latência. Porém, a tese proposta possui a vantagem do aumento da vazão de dados obtida pela implementação pipeline.

Tabela 3.1 Resumo de propostas que usam mascaramento para descorrelacionar o consumo de potência dos dados processados.

	Algoritmo	**Método**	**Custos**	**Tecnologia**
Pramstaller et al. [PRA04]	AES	Operações (+ e mapeamento GF)	Latência	CMOS 0,25µm
Mesquita et al. [MES06]	RSA	Reconfiguração modular de bases numéricas	Alto custo em área	FPGA
Popp e Mangard [POP06]	AES	Lógica geradora de ruído	Potência	CMOS
Ghosh et al. [GHO07]	AES	Circuitos balanceados	-	CMOS 0,65µm
Huiping et al. [HUI07]	DES	Operações (+ e x)	Área	CMOS 0,25µm
Ordu e Örs [ORD07]	AES	Operações (+ e mapeamento GF)	Latência	FPGA
Haijun et al. [HAI07]	DES	Usa SDDL e UMM	Área	CMOS 0,25µm
Golic [GOL07]	AES	Operações (+, -, xor)	Vulnerável a DPA HO	-
Ghellar e Lubaszewski [GHE08]	AES	Mapeamento de campos GF(2)	Área e Latência	CMOS
Goodwin e Wilson [GOO08]	AES	Alteração do algoritmo	Vazão	FPGA
Método proposto nesta Tese	DES	Aleatoriedade de execução, execução sobreposta (pipeline)	Área e latência	FPGA

3.2 MÉTODO POR INJEÇÃO DE RUÍDO

Este método tem como objetivo construir criptosistemas com consumo de potência aleatório. Isto significa que a cada ciclo de relógio uma quantidade aleatória de potência é consumida pelo sistema. Obter um consumo de potência idealmente aleatório não é possível na prática. Entretanto, existem várias propostas de como se aproximar do ideal. Na literatura são encontradas propostas que afetam o tempo de execução do algoritmo criptográfico e propostas que tornam o consumo de potência aleatório, ou seja, inserem uma atividade de chaveamento aleatória no circuito com consumo de potência dominante sobre o consumo correspondente à execução do algoritmo.

Em engenharia elétrica, a qualidade de um sinal medido pode ser definida como a razão existente entre o sinal pretendido e interferências sobre este sinal. A esta razão dá-se o nome de *relação sinal-ruído* (em inglês, *signal to noise ratio* - SNR). Esta relação é definida como a razão entre as potências de um sinal contendo algum tipo de informação (P_{signal}) e do ruído (P_{noise}), conforme indicado na Equação 1.

$$SNR = \frac{\mathrm{P}signal}{\mathrm{P}noise} \qquad (1)$$

Em termos de segurança, o chaveamento aleatório produz um ruído que reduz a relação sinal-ruído, aumentando a complexidade das análises de correlação. Em um caso ideal, o SNR deve ser *zero*, obtido teoricamente caso a potência do ruído inserido tenda a um valor infinito. Na prática, a potência de ruído pode ser elevada de modo a atender aos requisitos de segurança, sem elevar excessivamente o consumo de potência do dispositivo.

A seguir apresentam-se algumas propostas que empregam aleatoriedade para contramedir ataques DPA.

Benini et al. em [BEN03] propuseram uma combinação de técnicas de redução de potência e de controle da ativação do sinal de relógio, visando introduzir aleatoriedade significativa no traço de potência sem aumentar e, em alguns casos até reduzindo, o consumo de potência de sistemas. Deste modo, a técnica introduz ruído ao processamento de operações criptográficas. A maior contribuição dos Autores é não aumentar o consumo de potência com o uso da técnica. Esta abordagem reduz o consumo de potência se comparada com a implementação inicial. A idéia básica é oferecer um conjunto de primitivas de hardware (macros) que podem ser instanciadas pelos projetistas em muitos algoritmos. Adicionalmente, permite-se ao projetista reduzir o consumo de potência de modo controlado em tempo de projeto. Isto é uma característica desejável, pois encriptação e decriptação são freqüentemente exigidas em ambientes com restrições de potência.

Ciet et al. em [CIE03] propuseram o desenvolvimento de um criptosistema tomando como base o algoritmo RSA. Para imunizar o criptosistema contra ataques DPA, os Autores empregam o algoritmo de multiplicação de Montgomery, que permite cálculos em diferentes bases numéricas usando como recurso números RNS. A arquitetura proposta

replica o módulo de multiplicação de 16 bits de largura, de modo a evitar perda de desempenho. Diferentes estratégias de controle são aplicadas, de modo a se ter operações de 512 bits. O processamento paralelo e o uso de diferentes bases numéricas para realizar a multiplicação contribuem para descorrelacionar o consumo de potência dos dados processados. A arquitetura é validada em FPGA e os resultados mostram eficiência promissora, com tempos de processamento inferiores a 150 ms para uma chave de 1024 bits e com área adicional competitiva com outras implementações com contramedidas.

Standaert et al. em [STA04] investigaram a vulnerabilidade da implementação em hardware do algoritmo AES usando *pipelines*. Os Autores propuseram inicialmente o uso de pipeline no processamento de uma rodada do algoritmo, alterando o tempo de execução de 1 ciclo de relógio para 5 ciclos de relógios. Após, propuseram um ataque CPA considerando os registradores internos adicionados às rodadas, bem como a atividade de chaveamento dos circuitos e o modelo de potência baseado na distância de Hamming. O ataque realiza uma predição dos dados no conjunto de registradores a cada ciclo de relógio, estabelecendo uma matriz de predição para uma dada função de seleção escolhida. Resultados teóricos (via simulação) e práticos demonstram ser possível revelar a chave secreta. Assim, os Autores propõem uma arquitetura com pipeline interno à rodada do algoritmo e a replicação das rodadas deste para neutralizar os ataques. Resultados teóricos mostram que a chave correta é revelada, porém com uma probabilidade muito baixa em relação a outras hipóteses de chaves. O método usado utiliza o paradigma síncrono de projeto, facilitando ao atacante prever através de modelos adequados os instantes onde ocorrem os chaveamentos de dados no circuito. Além disso, não foram mostrados resultados práticos de ataques. A proposta sofre ainda um alto custo em área e latência, esta última compensada parcialmente pela utilização de pipeline.

Bucci et al. em [BUC05] propuseram uma nova contramedida a ataques DPA baseada na inserção de elementos de atraso no caminho de dados de criptosistemas. Inicialmente, cada dispositivo é composto por um flip-flop tipo D (FFD), uma cadeia com n elementos de atraso em série (sendo n o número de elementos) e um multiplexador (MUX). Um circuito de controle deve selecionar pseudo-aleatoriamente entre a saída direta do FFD ou a saída do FFD através da cadeia de atraso. O dispositivo proposto pode ainda ser estendido para 2^m possibilidades de cadeias de atraso (sendo m o número de cadeias), de modo a aumentar a aleatoriedade, por outro lado aumentando o dispêndio de hardware. Este dispositivo é inserido em cada bit que compõe o caminho de dados do criptosistema, visando aleatorizar o consumo de potência e assim reduzir a correlação entre dados processados e potência consumida. Estudos de caso foram realizados com um SBOX do algoritmo AES, usando tecnologia CMOS 0,18µm. Os resultados foram obtidos por simulação usando a ferramenta Nanosim da Synopsys. Como vantagens do método destacam-se o uso de um fluxo de projeto padrão e a não interferência na árvore de relógio do circuito. Como desvantagem destaca-se a excessiva área adicional ao circuito. Além disso, a inserção de atrasos aumenta a latência de processamento no criptosistema.

Yang et al. em [YAN05] apresentam uma nova abordagem contra ataques DPA, baseada no chaveamento aleatório de freqüência e tensão de alimentação (em inglês,

Dynamic Voltage and Frequency Switching - DVFS). Os Autores reutilizam uma estrutura proposta originalmente para redução do consumo de energia em sistemas intrachip para descorrelacionar o consumo de potência dos dados processados. Propõem-se três estratégias de controle diferentes. Estas são classificadas pelos Autores de acordo com seu nível de imunidade, sendo elas ingênua (*naive*), melhorada (*improved*) e avançada (*advanced*). As duas primeiras aumentam a robustez, porém ainda apresentam vulnerabilidades. A terceira revelou-se uma estratégia eficiente para bloquear ataques SPA e DPA. Os resultados mostram que os traços de potência apresentam níveis de aleatoriedade suficiente para evitar a fuga de informações. Além disso, o consumo médio de energia é reduzido em 27% na implementação do algoritmo DES. Por outro lado, o criptosistema sofre um aumento de 16% em seu tempo de execução em relação à implementação original.

Gürkaynak et al. em [GUR06] apresentam desafios e experiências de projeto com o paradigma GALS, tomando como estudo de caso o algoritmo criptográfico AES. Os Autores propuseram o uso do paradigma GALS para aumentar a robustez de um criptosistema a ataques por consumo de potência. O paradigma GALS oferece a projetistas recursos adicionais para implementar contramedidas a DPA. Para demonstrar isto, os Autores desenvolveram o criptosistema Acácia. Acácia, cuja estrutura é mostrada na Figura 3.2, implementa o algoritmo AES o qual executa sucessivamente uma rodada composta por 4 blocos funcionais: *AddroundKey*, *SubBytes*, *ShiftRows* e *MixColumns*. Acácia é subdividida em um modulo chamado *Goliath*, composto por um caminho de dados de 128 bits e um gerador aleatório de chaves criptográficas. Dois outros módulos menores denominados *David* completam o criptosistema. Estes módulos são equipados com caminho de dados de 32 bits e realizam as operações *SubBytes* e *Mixcolumns* da rodada do AES. Acácia é equipada com várias camadas de contramedidas a ataques DPA. São elas: (i) operações com dados falsos; (ii) *David* contém dois operadores *SubBytes* de 8 bits. Para executar a operação *MixColumns* são necessárias 4 operações *SubBytes*. *David* pode escalonar aleatoriamente estas quatro operações, de forma que em um dado ciclo de relógio, todos, apenas um, ou nenhum dos operadores *SubBytes* podem processar dados reais, enquanto os demais processam dados falsos. (iii) para cada rodada do AES são necessárias quatro operações *MixColumns*, sendo que estas podem ser executadas em qualquer ordem nos módulos *David*. (iv) todos os três caminhos de dados são implementados em módulos GALS com geradores de relógios próprios. (v) os módulos GALS usam geradores de relógio que podem pausar o relógio a cada ciclo. Com a combinação de todas as contramedidas listadas os Autores esperam oferecer um desafio maior a atacantes que usam técnicas de análise de consumo de potência. O particionamento proposto aumenta significantemente a latência do sistema, o que se agrava ainda com o processamento de dados falsos para aleatorizar o consumo. A interface assíncrona empregada requer o pausamento do relógio para troca de dados o que pode tornar o sistema vulnerável aos ataques SPA, visto que o relógio é o principal responsável pelo consumo de um circuito.

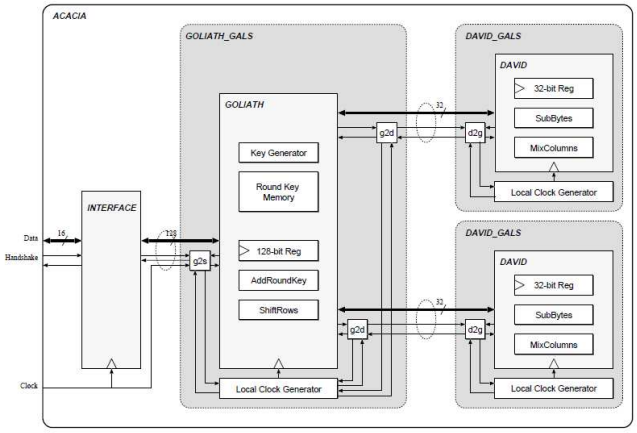


Figura 3.2 Estrutura GALS do sistema Acácia, proposto em [GUR06]. Acácia é composto pelo modulo Goliath que executa operações com 128 bits e por dois módulos David com operações de 32 bits.

Baddam e Zwolinski em [BAD07] apresentam uma análise e discussão sobre o uso de variação aleatória de freqüência e tensão (em inglês, *Random Dynamic Voltage and Frequency Scaling* - RDVFS) como contramedida a ataques DPA. Os Autores mostram que a partir da análise do traçado de corrente é possível medir a freqüência de operação e/ou o par tensão-freqüência durante o processo de criptografia. Os Autores propõem um método que emprega apenas chaveamento aleatório de tensão e mantém constante a freqüência de operação do circuito. Os resultados comprovam a eficiência do método contra ataques DPA e destacam como vantagem do método o fato deste não exigir mudanças na lógica e/ou fluxo de projeto, bem como não implicar em custos adicionais significativos em área, desempenho ou consumo de potência. A principal restrição da proposta é não permitir ao atacante ter acesso à conexão entre o gerador de números aleatórios e o controlador de voltagem. Caso isso aconteça, o sistema perde sua aleatoriedade e se torna vulnerável a ataques DPA. Outra restrição do método é que a taxa de mudança de voltagem seja menor que o tempo para processar o número de entradas mínimo, de modo a evitar um ataque DPA. Se a taxa é muito próxima ao número mínimo, o atacante pode implementar um ataque bem sucedido antes da aleatoriedade ser introduzida.

Zafar e Har em [ZAF08] propuseram um método para variação aleatória de freqüências em tempo de execução como forma de ocultar a fuga de informações em um criptosistema contendo o algoritmo AES. O módulo gerador do sinal de relógio com freqüência aleatória proposto seleciona uma nova freqüência de relógio a cada nova mensagem de entrada no criptosistema. Com isso, a aleatoriedade é inserida de modo a descorrelacionar os dados processados e o consumo de potência medido. O módulo gerador de relógio é composto pelo oscilador proposto em [ZAF05] sendo facilmente implementado em FPGA.

Lu et al. em [LU08] investigam o uso da técnica de inserção de atrasos aleatórios (do inglês, *Random Delay Insertion* - RDI) em projetos de criptosistemas destinados a FPGAs. Os Autores provam teórica e praticamente que a técnica é efetiva contra ataques DPA e propõem parâmetros que podem ser utilizados para otimizar a segurança do

projeto em termos de área, desempenho e potência consumida. RDI foi inicialmente aplicada a criptosistemas microprocessados por Clavier et al. em [CLA00] para reduzir a correlação entre um modelo de potência previamente definido e o consumo de potência verdadeiro de um dispositivo. Para tanto, uma cadeia de elementos de atrasos programáveis é adicionada ao caminho de dados, a fim de aleatorizar as curvas de potência na dimensão do tempo. Este método é vulnerável quando aplicado em sistemas microprocessados, segundo os Autores. Logo, Lu et al. propuseram a implementação em FPGA com o ajuste de parâmetros tais como variação do atraso inserido, seu desvio padrão e um fator de multiplicação *k*. Os resultados obtidos mostram que RDI em FPGA é mais eficaz que a versão original do algoritmo implementada em FPGA. A proposta sofre uma penalização em área de cerca de 100%, custo relativamente baixo comparado a outros métodos.

Kamoun et al. em [KAM09] propuseram um gerador de ruído como forma de descorrelacionar o consumo de potência dos dados processados em um projeto orientado a FPGA. Os Autores usam o algoritmo AES como estudo de caso para validação da técnica. O gerador proposto nada mais é que a replicação das duas funções mais vulneráveis do AES, *AddRoundKey* e *SubBytes*. Quando um dado é processado pelo criptosistema, este é executado concorrentemente por estas operações, porém com chaves diferentes. Um deles utiliza a chave secreta e o gerador de ruído usa uma chave aleatória. O método proposto apresenta baixo custo em área comparado a um método de mascaramento [CAN08], embora estes módulos replicados representem a maior área do algoritmo. Por outro lado, o método determina restrições nas etapas de posicionamento e roteamento para garantir que o processamento seja realmente paralelo.

3.2.1 COMPARAÇÃO ENTRE PROPOSTAS

Conforme a revisão das propostas resumidas na Tabela 3.2 é possível afirmar que esta abordagem depende da capacidade de reduzir a taxa SNR referente ao consumo de potência do criptosistema. Quanto maior for a aleatoriedade inserida no sistema, ou seja, o ruído e as variações no tempo de execução de operações, melhor será a eficiência do método. A necessidade de inserção de hardware extra para aumentar a aleatoriedade do consumo exige um aumento em área e conseqüentemente um aumento no consumo de potência do sistema. Observa-se ainda que esta técnica pode ser desenvolvida tanto em hardware como em software. Em [YAN05] os Autores usam uma estrutura de hardware para efetivamente atuar na variação de freqüência e tensão, porém o gerenciamento é executado em software em um processador de propósito geral. A Tese aqui proposta também implementa um pipeline tal como Standaert et al. propuseram, mas a diferença neste caso está no fato de os estágios do pipeline comunicarem-se assincronamente. Em relação à proposta de Ciet et al., a Tese aqui proposta diferencia-se por replicar o bloco de encriptação completo do algoritmo ao invés de apenas uma operação. Em relação aos demais trabalhos, a Tese diferencia-se por inserir aleatoriedade no processamento de grupos de rodadas e com variação da freqüência do sinal do relógio em cada estágio a cada novo dado. Tal como implementado nos estudos de caso, é possível processar o algoritmo DES em grupos de oito rodadas (pipeline 2 estágios), quatro rodadas (pipeline 4

estágios), duas rodadas (pipeline 8 estágios) e apenas 1 rodada em cada estágio em uma arquitetura pipeline com 16 estágios.

Tabela 3.2 Resumo de propostas que visam descorrelacionar dados aleatorizando o consumo de potência.

	Algoritmo	Método	Custos	Tecnologia
Benini et al. [BEN03]	RSA	Ativação do relógio	Baixo ou redução de potência	CMOS 0,18µm
Ciet et al. [CIE03]	RSA	Replicação e processamento paralelo	Área e desempenho compatíveis	FPGA
Standaert et al. [STA04]	AES	Pipeline em dois níveis	Área e latência	FPGA
Bucci et al. [BUC05]	AES	Atrasos no caminho de dados	Alto custo em área	CMOS 0,18µm
Yang et al. [YAN05]	DES	DVFS	Redução de potência e desempenho	CMOS
Gürkaynak et al. [GUR06]	AES	Dados falsos e chaveamento de módulos	Alto custo em área	CMOS 0,25µm
Baddam e Zwolinski [BAD07]	AES	DVS	Baixo em potência	CMOS
Zafar e Har [ZAF08]	AES	Variação da freqüência do relógio	Baixo em área	FPGA
Lu et al. [LU08]	AES	Atrasos aleatórios no caminho de dados	Redução no desempenho	FPGA
Kamoun et al. [KAM09]	AES	Replicação de módulos	Baixo em área e freqüência	FPGA
Método proposto	DES	Aleatoriedade de execução, execução sobreposta (pipeline)	Área e latência	FPGA

3.3 MÉTODO POR UNIFORMIZAÇÃO DO CONSUMO DE POTÊNCIA

Em oposição ao método revisado anteriormente, este método tem como objetivo construir criptosistemas com consumo de potência constante, ou ainda, independente dos dados processados. Isto significa que a cada ciclo de relógio uma quantidade constante de potência é consumida pelo sistema. No caso ideal, apenas contramedidas que tornam o consumo de potência em um criptosistema exatamente igual para todas operações e todos valores de dados oferecem perfeita proteção aos ataques DPA. Na prática existem duas estratégias para atingir este objetivo, a primeira delas emprega estilos lógicos específicos no projeto do dispositivo criptográfico. A segunda abordagem utiliza um filtro

para remover as dependências de consumo de potência existentes nas operações e dados manipulados. Esta segunda abordagem é pouco usada, a julgar pela quantidade de literatura disponível. Por outro lado, encontra-se uma grande quantidade de propostas que utiliza a primeira abordagem.

A seguir são apresentadas algumas estratégias adotadas no projeto de circuitos com consumo de potência independente de dados.

Tiri et al. em [TIR02] propuseram um novo estilo de lógica CMOS que opera com consumo de potência independente de valores lógicos e da seqüência de dados. O estilo denominado em inglês, *Sense Amplifier Based Logic* ou SABL, apresenta um consumo independente de dados seguindo dois princípios: (i) ter apenas um evento de chaveamento por ciclo de relógio, independentemente da seqüência de dados de entrada e; (ii) ter uma carga capacitiva constante durante este evento de chaveamento. SABL apresenta uma estrutura dinâmica e diferencial similar à lógica DCVSL (do inglês, *Differential Cascode Voltage Switch Logic*) [RAB03]. Entende-se por lógica dinâmica circuitos lógicos que alternam sucessivamente entre etapas de pré-carga (saídas forçadas para um valor lógico determinado) e avaliação (saída computada para o valor lógico correto para os valores de entrada). Já a lógica diferencial é caracterizada por representar um bit de informação com dois fios que representam o valor de um bit quando em polaridades inversas ou a ausência de informação quando ambos os fios estão em ‘0’. Ou seja, esta é uma codificação denominada trilha dupla (do inglês, *Dual Rail* - DR). Experimentos realizados com uma SBOX do algoritmo *Kasumi* demonstram uma variação de energia normalizada 116 vezes menor quando comparada a implementações CMOS típicas. Por outro lado, a solução apresenta um custo de área e potência duas vezes maior.

Tiri e Verbauwhede em [TIR04] apresentam um novo método de projeto de *standard cells* que visa construir portas lógicas com consumo de potência uniforme para fluxos de projeto ASIC e FPGA. Embora SABL [TIR02] tenha sido projetado para este fim, o fluxo de projeto de CIs com esta lógica requer a caracterização da nova biblioteca. O método proposto pelos Autores em [TIR04] evita o projeto de uma nova biblioteca completa, permitindo construir portas complexas a partir de bibliotecas já existentes, seguindo o comportamento de SABL. A lógica diferencial dinâmica simples (em inglês, *Simple Dynamic Differential Logic* - SDDL) é complementar à lógica convencional. Além disso, adiciona um circuito de pré-carga às saídas tradicional e complementar de modo a forçá-las ao nível lógico ‘0’. Um projeto alternativo desta lógica é a implementação do circuito de pré-carga na entrada do circuito e a introdução de registradores na saída. Esta alternativa é chamada de lógica diferencial dinâmica em onda (em inglês, *Wave Dynamic Differential Logic* - WDDL). Experimentos revelam que esta é eficiente na redução da variação do consumo de potência tanto em ASICs quanto em FPGAs, porém apresenta custos significativos em área, desempenho e consumo de potência.

Vahedi et al. em [VAH06] propuseram um circuito para uniformizar a corrente consumida em criptosistemas microprocessados. O circuito proposto controla dinamicamente o consumo de potência de duas maneiras: (i) injetando corrente; e (ii)

regulando a tensão sobre o criptosistema. A corrente de alimentação é monitorada por um sensor que repassa as variações de corrente a um circuito responsável por manter a corrente consumida constante durante a encriptação de dados. Quando as variações de corrente excedem os limites de atuação do circuito de controle de corrente, o regulador de tensão é acionado, de modo a redimensionar a tensão sobre o criptosistema fazendo com que o controle de corrente volte a atuar novamente sobre o criptosistema. Como desvantagem do método, o circuito regulador de tensão deve limitar-se a operar com variações de tensão definidas pelo fabricante do microprocessador. Além disso, a redução da tensão de alimentação ocasiona atrasos no processamento de operações, e por conseqüência pode impor limites para o sistema atender aplicações que tenham restrições de tempo de processamento.

Razafindraibe et al. em [RAZ07] realizaram uma avaliação detalhada da robustez da lógica em trilha dupla (DR) a ataques DPA, e mostraram que a faixa de operação da lógica considerada efetivamente robusta é surpreendentemente pequena. DR não reduz suficientemente a correlação entre dados e o tempo de computação para caracterizar-se como uma contramedida robusta a ataques DPA. Motivados por esta análise, os Autores propõem o uso de uma lógica alternativa à DR, chamada de lógica segura em três trilhas (em inglês, *Secure Triple Track Logic* - STTL). Esta lógica utiliza uma terceira trilha para a validação de dados. O circuito de validação é projetado com portas de baixa corrente, cujos atrasos de propagação são maiores que os do restante do circuito. Esta característica garante que a validação seja gerada após a estabilização dos dados na saída do circuito e independentemente dos dados processados. Outra característica de STTL é que o processamento do dado ocorre somente após a chegada de todos os sinais de validação envolvidos em seu processamento. Isto garante a STTL uma tolerância ao desequilíbrio do tempo de propagação entre fios complementares na lógica DR, um efeito naturalmente introduzido durante a etapa de posicionamento e roteamento do circuito. A lógica de validação redundante garante que os dados sejam processados independentemente de tempo e do consumo de potência. Esta lógica apresenta custos em área semelhantes aos da lógica DR. Por outro lado, apresenta custos elevados em latência devido ao circuito de validação empregado adicionalmente.

Guilley et al. em [GUI08] propuseram uma investigação da contramedida WDDL proposta por Tiri e Verbauwhede. Estes Autores apresentam uma metodologia de CAD para desenvolver WDDL em FPGA e a seguir realizam uma avaliação de robustez da lógica, usando como estudo de caso o algoritmo DES. Os Autores discutem um método para redução de área e, além disso, propõem uma avaliação de algumas ferramentas de síntese de hardware. Neste caso, os Autores propõem uma heurística para obter SBOXs menores em relação às geradas automaticamente pelo CAD de ASICs.

Em outro trabalho, **Guilley et al.** em [GUI08b] propuseram um estudo do impacto causado pelas etapas de posicionamento e roteamento em FPGA usando duas das principais ferramentas de CAD disponíveis no mercado, os ambientes de síntese física da Altera e da Xilinx. Segundo os Autores, as lógicas DPL propostas preocupam-se em manter a quantidade de chaveamento de transistores constante para qualquer dado processado como forma de manter o consumo de potência independente dos dados

processados. Porém, os Autores destacam que o tempo de propagação dos sinais entre as trilhas duplas também tem importância. Assim, realizaram experimentos de maneira a estabelecer restrições de posicionamento que atenuem os desequilíbrios de tempo de propagação dos sinais. As ferramentas da Altera apresentaram melhores resultados que as ferramentas da Xilinx. Embora o método seja interessante e apresente resultados relevantes, a tarefa de estabelecer restrições de roteamento em fios no fluxo de projeto de FPGAs é complexa, devido ao não-determinismo das ferramentas de síntese. Esta é a principal desvantagem do método.

Kulikowski et al. em [KUL08] propuseram o uso de projetos com codificação em trilha dupla com consumo de potência independente de dados, tolerante ao desequilíbrio de interconexões entre portas lógicas e à variabilidade do processo de fabricação de circuitos integrados. O projeto, denominado pelos Autores em inglês Asynchronous Directional Latch Based Logic (ADLBL) permite um consumo independente de dados pela adição de um protocolo de descarga capacitiva direcional em trilha dupla, baseado no projeto de um *latch* que permite a descarga completa de ambas as trilhas. Os resultados obtidos por simulação com a implementação de um submódulo do AES demonstram que mesmo na presença de desequilíbrio de capacitâncias no circuito, a lógica proposta resiste a ataques CPA, mostrando-se mais robusta que WDDL [TIR04]. Apenas experimentos teóricos foram realizados, não sendo mostradas avaliações práticas de robustez da lógica proposta. Os custos em área e potência são potencialmente elevados, mas os Autores não apresentam avaliação de área. Além disso, a técnica requer o projeto de uma biblioteca específica.

Muresan e Gregori propuseram em [MUR08] o uso de um circuito de proteção contra ataques DPA para criptosistemas tais como *smart cards*. O circuito é baseado na técnica de regulação de corrente (em inglês, *current flattening technique*) inicialmente introduzida em [MUR04] e posteriormente usada em [MUR05] e [MES05]. Este circuito pode ser integrado ao mesmo chip ou ao mesmo encapsulamento do criptosistema. O objetivo desta técnica é manter constante a corrente necessária para suprir o criptosistema, mascarando assim a dependência entre dados processados e a corrente consumida. Como a corrente I_s consumida pelo criptosistema varia a cada dado processado, o circuito de proteção monitora Is e varia dinamicamente seu consumo de corrente *If* de modo que a corrente total consumida pela fonte $Idd = I_s + I_f$ seja constante. A principal vantagem desta proposta é a simplicidade da integração ao criptosistema existente, não sendo necessário o uso de biblioteca dedicada, reprojeto do criptosistema ou modificações em software. Por outro lado, a técnica eleva o consumo de potência, visando uniformizá-lo.

Vahedi et al. em [VAH08] propuseram modificações no trabalho proposto em [VAH06]. Neste último adiciona-se um método de chaveamento de freqüências para remediar as penalidades com atrasos no processamento. Durante a operação de encriptação no processador, se a corrente não alcança a faixa de alimentação pré-definida para o processador, então um bloco de chaveamento de freqüência é ativado para encontrar a freqüência de operação mínima que satisfaça as exigências de tempo de

processamento. Este bloco aumenta a faixa de voltagens de alimentação do processador e traz dois benefícios: (i) garante a funcionalidade do processador e; (ii) reduz os custos com consumo de potência. Este novo método de projeto introduz um circuito de uniformização de corrente mais eficiente quanto ao consumo de potência. A proposta apresenta um circuito com baixos custos de potência e área, sendo útil para aplicações tais como *smart cards* e dispositivos de comunicação móvel.

Rammohan et al. em [RAM08] apresentaram o estilo lógico denominado Lógica Diferencial e Dinâmica Complementar Reduzida (em inglês, *Reduced Complementary Dynamic and Differential Logic* - RCDDL) para conceber sistemas imunes a ataques DPA. Este estilo garante um consumo de potência uniforme para dados de entrada (mensagem e chaves criptográficas). Em oposição aos estilos DPL existentes, que complementam todas as portas lógicas para gerar a saída diferencial, o estilo RCDDL propõe o reuso de portas lógicas, de modo a reduzir o número de portas na geração da lógica complementar para obter a saída diferencial. Esta é a primeira proposta de reuso de portas lógicas para conceber uma lógica complementar e conseqüentemente reduzir o custo em área. Duas são as exigências para garantir segurança a lógicas DPL: (i) exatamente uma transição de saída a cada ciclo de relógio; (ii) a capacitância total (de carga e de descarga) deve permanecer constante a cada ciclo de relógio. Resultados experimentais sobre circuitos tais como o algoritmo criptográfico DES e circuitos sintéticos mostram que o uso de RCDDL produz um aumento significativo de resistência a ataques, apresentando uma redução de 42% na variação de corrente máxima em relação à lógica WDDL e 11 vezes quando comparada à lógica CMOS ordinária. Os resultados também mostram uma melhora no consumo médio de potência e área, mas com uma penalidade na latência, quando comparado à lógica WDDL.

Moradi et al. [MOR09a] propuseram um estudo com a lógica de recuperação de energia [KHA08], destinada à concepção de circuitos de baixo consumo de energia. Esta lógica apresenta características tais como mecanismo de *pipeline*, consumo reduzido de energia dependente de dados e reduzida radiação de energia, o que é útil em áreas de aplicações tais como segurança de circuitos criptográficos embarcados. Os Autores examinam a robustez de portas lógicas utilizando a lógica proposta bem como os custos de implementação associados. Os resultados demonstram que a lógica proposta requer menos área em relação a lógicas destinadas à segurança tais como SABL e demais DPLs. Por outro lado, os estudos mostram que a segurança da lógica é limitada pela freqüência de operação, apresentando melhores resultados em baixas freqüências.

3.3.1 COMPARAÇÃO ENTRE PROPOSTAS

A Tabela 3.3 resume as características dos trabalhos revisados na Seção 3.3. O tipo de contramedida revisado aqui visa eliminar a fuga de informação ao tornar o consumo de potência constante e independente dos dados processados pelo sistema criptográfico. Com este objetivo, os Autores buscam propor novas estruturas lógicas para conceber circuitos com consumo uniforme, evitando tanto quanto possível as variações de cargas capacitivas no circuito.

Tabela 3.3 Resumo de propostas que visam descorrelacionar dados através da uniformização do consumo de potência.

	Algoritmo	Método	Custos	Tecnologia
Tiri et al. [TIR02]	Kasumi	Lógica diferencial e dinâmica (caracterizada)	Alto custo em potência	CMOS 0,18µm
Tiri e Verbauwhede [TIR04]	Kasumi, DES, AES	Lógica diferencial e dinâmica	Alto custo em potência	CMOS 0,18µm e FPGA
Vahedi et al. [VAH06]	DES	Circuito regulador de voltagem	Custo em área	CMOS
Razafindraibe et al. [RAZ07]	DES	Lógica em três trilhas (validação)	Alto custo em área	CMOS 0,35µm e FPGA
Guilley et al. [GUI08]	DES	Balanced WDDL	Área	FPGA
Guilley e al. [GUI08b]	DES	Comparação de ferramentas de síntese para fluxo de projeto com lógicas balanceadas	Area	FPGA
Kulikowski et al. [KUL08]	AES	Latch e protocolo tolerante a desequilíbrios capacitivos	Custo em área	CMOS
Muresan et al. [MUR08]	DES	Circuito regulador de corrente	Alto custo em potência	CMOS
Vahedi et al. [VAH08]	DES	Circuito regulador de voltagem e freqüência	Custo em área	CMOS
Rammohan et al. [RAM08]	DES	DPL com reuso de portas	Médio	CMOS
Moradi et al. [MOR09]	Não identificado	Lógica para redução de consumo	Freqüência de operação	CMOS
Método proposto nesta Tese	DES	Lógica de validação com tempo constante (STTL em FPGA)	Latência e área	FPGA

A vantagem deste método é atuar diretamente na origem do problema, buscando alternativas para contornar os problemas inerentes das tecnologias atualmente disponíveis. Por outro lado, os custos para se obter uma lógica com tais características são altos em termos de área, consumo e latência, principalmente devido as portas passarem a ter estruturas diferenciais (uso de lógica complementar) e comportamento dinâmico (ter obrigatoriamente o mesmo número de chaveamentos para cada dado processado), a fim de obter cargas capacitivas constantes. Além disso, alterações no fluxo de projeto podem ser necessárias para permitir a concepção do circuito com uma nova biblioteca lógica. O protótipo da lógica STTL proposto neste trabalho visa eliminar as fugas de informações através da uniformização do consumo de potência. Ao contrário das demais lógicas, STTL utiliza três trilhas para codificar um bit de informação. A lógica mostra-se robusta a ataques por consumo de potência e por radiação eletromagnética. Por outro lado, STTL sofre uma penalidade na latência da computação. Quanto ao custo em área, STTL mostra-se similar às demais lógicas em trilha dupla.

3.4 MÉTODOS PARA CONTRAMEDIR ATAQUES POR INDUÇÃO A FALHAS

Nesta Seção revisam-se alguns trabalhos que têm por objetivo imunizar criptosistemas contra a indução maliciosa de falhas. De um modo geral, os trabalhos encontrados na literatura utilizam técnicas de detecção e correção de erros para evitar que criptogramas defeituosos sejam produzidos e conseqüentemente usados em ataques DFA. Embora este tipo de ataque esteja fora do escopo deste trabalho, uma pequena revisão da literatura é apresentada em caráter exploratório.

Yen e Wu em [YEN06] propuseram vários esquemas de detecção de erros implementados em hardware com base na verificação de redundância cíclica (*n+1,n*) (do inglês, *Cyclic Redundancy Check* - CRC). O esquema proposto facilmente prevê a paridade do resultado de uma operação. Os Autores usam o algoritmo AES como estudo de caso devido a sua estrutura orientada a bytes, o que facilita a previsão da paridade de operações internas, a partir de combinações lineares das paridades de entrada. Como vantagens do método proposto, destaca-se sua escalabilidade, que permite sua aplicação em arquiteturas com caminho de dados de 8, 32 e 128 bits. Além disso, o cálculo de paridade proposto é simétrico, ou seja, aplicável tanto a operações de encriptação como de decriptação. O processo de geração de paridade em ambas operações é muito similar, o que traz benefícios na implementação da abordagem em hardware.

Regazzoni et al. em [REG07] propuseram um estudo sobre o impacto na robustez de um criptosistema equipado com uma contramedida específica para um dado canal lateral, porém submetido a ataques por outro canal lateral. Neste trabalho os Autores utilizam um circuito de detecção de erros, mais precisamente um detector de paridade, a fim de imunizar contra ataques por indução a falhas uma SBOX do algoritmo Kasumi, implementada em hardware com tecnologia CMOS 0,18 µm. Em simulação, a SBOX é então submetida a ataques DPA visando avaliar sua robustez. Durante os experimentos, os Autores realizaram ataques em duas versões do circuito, uma com e outra sem contramedida. Os resultados obtidos revelaram um aumento de 35% no número de chaves criptográficas descobertas em relação ao circuito sem a contramedida a DFA.

Embora a contramedida e o circuito adotados sejam pouco complexos, ou seja, apenas um submódulo do algoritmo Kasumi equipado com um detector de um bit de paridade, a questão levantada pelos Autores é relevante. O estudo revela que os esforços em pesquisa direcionados a encontrar uma solução a um tipo de ataque pode causar vulnerabilidades a outro método de ataque.

Maistri e Leveugle em [MAI08] propuseram o uso da técnica denominada em inglês *double data rate* (DDR) como base para implementar uma contramedida a ataques DFA. Trabalhos anteriores revelaram que o uso de redundância temporal como método de detecção de falhas tem um impacto negativo na latência de um criptosistema. Ou seja, a encriptação/decriptação de um dado deve ser realizada duas vezes e por conseqüência ocasiona a redução da vazão no criptosistema. Com o uso de DDR, os Autores contornam este problema, obtendo uma vazão compatível com implementações típicas do algoritmo. Em relação à robustez, o método mostra-se compatível com métodos anteriormente propostos. Por outro lado, a área e a freqüência de operação sofrem penalizações neste método.

Bhasin et al. em [BHA09] apresentam um estudo provando que os estilos lógicos em trilha dupla dedicados a evitar ataques por consumo de potência são naturalmente imunes à maioria dos ataques por indução a falhas. Os Autores apresentam um estudo de caso com o estilo lógico WDDL sem avaliação antecipada (do inglês, *WDDL without early evaluation* ou WDDL EE). Este estilo lógico remove a fase de avaliação antecipada de modo a evitar vulnerabilidades provocadas pelos tempos de propagação diferentes dos sinais diferenciais de entrada [BHA09]. Estas lógicas garantem que em caso de problemas com os sinais de entrada presos a um valor nulo ou a chegada de um valor não especificado as saídas terão sempre valores nulos (os espaçadores da codificação DR), o que evita ataques DFA. O fluxo de projeto apresentado é dedicado a FPGAs Altera, mas pode ser adaptado para a FPGAs Xilinx segundo os Autores. A proposta dos Autores apresenta penalidades em área e desempenho.

3.4.1 CONSIDERAÇÕES SOBRE O MÉTODO

Este método é discutido aqui para complementar a revisão da literatura sobre os métodos de contramedidas para circuitos criptográficos. O objetivo básico dos métodos é impedir que a ocorrência de falhas no processamento, seja qual for sua origem, propague um erro ao meio exterior do circuito. Isto pode ser usado como informação privilegiada para a descoberta da chave secreta em um circuito criptográfico. A inclusão de métodos de detecção de erros ocorre geralmente em nível de hardware, a um custo de área e principalmente gerando aumento da latência. Isto se deve principalmente ao uso de redundância temporal ou previsão de cálculo usado para evitar as falhas induzidas.

A Tabela 3.4 apresenta um resumo dos trabalhos revisados sobre métodos para evitar que análises por indução a falhas revelem os dados secretos de um criptosistema.

Tabela 3.4 Comparação de características de alguns trabalhos que propõem métodos para contramedir fuga de informações por indução a falhas.

	Algoritmo	**Método**	**Custos**	**Tecnologia**
Yen e Wu [YEN06]	AES	Detecção de Erros usando CRC	Área	-
Regazzoni et al. [REG07]	AES e Kasumi	Detector de paridade	Vulnerabilidade DPA	CMOS 0,18µm
Maistri e Leveugle [MAI08]	AES	Redundância temporal acelerada por DDR	Área e baixa freqüência de operação	CMOS
Bhasin et al. [RAM09]	AES	DPLs sem fase de avaliação	Área e fluxo complexo	FPGA

3.5 CONCLUSÕES

A presente Seção apresentou uma revisão de propostas para evitar a fuga de informações através de canais laterais em sistemas criptográficos. Como revisado no Capítulo 2, o problema da fuga de informações através de canais laterais como consumo de potência ocorre devido ao uso da tecnologia CMOS de concepção de circuitos integrados. Após o problema ser definido nos experimentos realizados por Kocher, muitos trabalhos foram propostos visando encontrar uma solução a estes tipos de vulnerabilidades. Na literatura nota-se claramente que existem 3 métodos principais para evitar a correlação de dados com o consumo de potência de um dispositivo eletrônico. São estes: mascaramento, inserção de aleatoriedade e uniformização do consumo de potência, conforme revisado neste Capítulo. Dois destes métodos, mascaramento e inserção de aleatoriedade propõem soluções para impedir a correlação de dados com a fuga de informação no canal lateral analisado. Nestes casos, as características de consumo da tecnologia CMOS são mantidas, porém os métodos provocam desordenamento do consumo, a fim de dificultar a tarefa dos atacantes. Já o método de uniformização do consumo é aplicado diretamente na tecnologia de concepção visando obter circuitos com consumo de potência constante para qualquer dado processado. Logicamente, a segurança proposta pelos métodos apresenta custos em área, potência e desempenho. De acordo com a revisão realizada, nenhum método aplicado isoladamente é capaz de garantir a segurança completa de um criptosistema, todos apresentam algum tipo de restrição. Acredita-se que a combinação de métodos eleva o nível de segurança nestes sistemas.

4. PROTOTIPAÇÃO DE LÓGICA NÃO-SÍNCRONA ROBUSTA A DPA E DEMA

Os estudos e avaliações de lógicas em trilha dupla (DPLs) propostas para imunizar sistemas criptográficos a ataques DPA revelaram vulnerabilidades conforme indicado por Razafindraibe et al. em [RAZ07]. Com base nestas avaliações os Autores propuseram novas diretivas para conceber a lógica STTL visando reforçar a imunidade de DPLs ao introduzir uma lógica de validação associada a um sinal de validade redundante conforme discutido anteriormente. Uma biblioteca lógica é proposta em tecnologia CMOS e validada com um estudo de caso para prova de conceitos. Os resultados obtidos por simulação mostram um processamento de dados quase independente do consumo de potência e do tempo de propagação motivando o uso da lógica STTL.

Para melhor avaliar a robustez da lógica proposta torna-se imprescindível submetê-la a experimentos práticos de avaliação do consumo de potência. O tempo de projeto e os custos para desenvolver estudos de caso usando um circuito integrado dedicado são elevados. Deste modo a prototipação usando dispositivos FPGA mostra-se atrativa para esta nova etapa de validação da lógica. Este foi basicamente o objetivo do estágio sanduíche realizado no LIRMM - (Université Montpellier II - França) durante um ano e uma das contribuições originais desta tese. Este Capítulo apresenta as atividades e os resultados obtidos com o desenvolvimento de protótipos da biblioteca STTL em FPGA bem como a avaliação de sua robustez às análises de consumo de potência e de radiação eletromagnéticas.

Como as propostas de contramedidas desenvolvidas neste trabalho baseiam-se no paradigma não-síncrono de projeto, a Seção 4.1 apresenta uma revisão específica de pressupostos empregados para o desenvolvimento de circuitos não-síncronos. A Seção 4.2 apresenta os pressupostos da lógica STTL. A proposta de prototipação em FPGA de uma biblioteca de portas lógicas STTL é mostrada na Seção 4.3. Um estudo de caso com a implementação do algoritmo DES usando a lógica STTL é apresentado na Seção 4.4. Nas Seções 4.5 e 4.6 são abordados os sistemas de medição e aquisição de dados usados para as análises. As avaliações de robustez a ataques por consumo de potência e por radiação eletromagnética estão discutidas nas Seções 4.7 e 4.8.

4.1 REVISÃO SOBRE PROJETO DE CIRCUITOS NÃO-SÍNCRONOS

Com o objetivo de superar as limitações do projeto síncrono, diversos grupos de pesquisa estão retomando o interesse no desenvolvimento de circuitos não-síncronos. Para facilitar a distinção entre os estilos de projeto, o termo *não-síncrono* é utilizado neste trabalho para englobar paradigmas assíncronos (*clockless*), o paradigma GALS e outros paradigmas tal como a dessincronização [COR06].

Circuitos não-síncronos são circuitos que assumem sinais binários, mas não assumem o pressuposto de discretização do tempo, isto é, em circuitos assíncronos o tempo é tratado como uma variável contínua [SPA02]. Este tipo de circuito pode eliminar os problemas de escorregamento e de dissipação de potência do sinal de relógio

[MAH98]. Entretanto, o desenvolvimento de sistemas não-síncronos esbarra na falta de ferramentas adequadas para a automatização do processo de desenvolvimento [SOT02] [RAF10].

Motivados pelas limitações do estilo síncrono de projeto, pela falta de ferramentas para dar suporte a circuitos totalmente assíncronos, e pela grande popularidade do estilo síncrono de projeto, alguns trabalhos de pesquisa propõem soluções intermediárias entre o projeto síncrono e o projeto assíncrono. O objetivo principal é manter as ferramentas do projeto síncrono e eliminar ou reduzir o uso de sincronização através do sinal de relógio. Entre essas propostas pode-se destacar o uso de sistemas GALS e o uso da dessincronização (do inglês, *desynchronization*) [COR06].

A dessincronização é uma técnica de geração de circuitos assíncronos a partir do estilo síncrono [COR06]. A única alteração do estilo síncrono é a substituição da etapa de geração da árvore de relógio por uma etapa de inserção de circuitos de handshake, um para o controle de cada etapa do circuito síncrono. O período do sinal de relógio é substituído por um elemento de atraso que deve possuir atraso maior que o atraso de propagação de pior caso da lógica combinacional a qual está associado.

A segunda proposta que pode potencialmente preencher a lacuna entre sistemas síncronos e assíncronos é a decomposição de um sistema síncrono em diversos módulos que não trabalham globalmente sincronizados, ou seja, onde cada módulo possua um domínio de relógio distinto. Este estilo de projeto é conhecido como Globalmente Assíncrono e Localmente Síncrono [CHA84]. Em sistemas GALS, cada módulo trabalha sincronamente [TEE07] [KRS07], mas a interação entre módulos utiliza uma interface de comunicação assíncrona, responsável por efetuar a transferência de informações entre os módulos síncronos [PON07].

Estilos de projeto não-síncronos mantêm o pressuposto de discretização dos níveis de tensão, mas adotam outros pressupostos quanto ao tempo ou nenhum pressuposto neste sentido. Dessa forma, o tempo tem de ser tratado como uma variável contínua, o que torna tais circuitos mais sensíveis a fenômenos temporais que ocorrem em circuitos digitais. A seguir serão descritos alguns destes fenômenos temporais, protocolos de comunicação, codificação de dados extraídos na sua maioria do trabalho de Sparsø e Furber [SPA02].

4.1.1 FENÔMENOS TEMPORAIS

Transitórios (em inglês, *hazards*) são exemplos de fenômenos temporais que podem afetar o funcionamento de circuitos não-síncronos. Em sistemas síncronos, valores transitórios podem ocorrer sem problemas, desde que no instante de amostragem todos os valores estejam estáveis. Em sistemas assíncronos, um valor transitório pode levar o circuito a um estado inválido ou indesejado [CAL98].

A existência de transitórios acontece devido à ocorrência de atrasos diferenciados de componentes ativos e fios ao longo do circuito. Entretanto, existem algumas técnicas

que podem garantir o desenvolvimento de circuitos combinacionais com comportamento livre de transitórios [NOW05] [SPA02].

Outro fenômeno temporal que pode prejudicar o funcionamento de circuitos digitais é a metaestabilidade. Este fenômeno é mais difícil de tratar quando técnicas não-síncronas de projeto são adotadas. A metaestabilidade é um fenômeno que pode ocorrer em dispositivos de armazenamento [WES94]. Quando um dado a ser registrado em um elemento de armazenamento altera o seu valor simultaneamente ou muito próximo à transição do sinal que habilita a amostragem, a saída pode apresentar problemas. Isto provavelmente ocorre se não forem respeitadas as restrições de tempos de setup e hold [WES94] dos elementos de memória. Algumas falhas possíveis são: a saída apresentar um valor indeterminado entre '*0*' e '*1*', ou haver uma demora arbitrariamente longa para que a transição de valor surja na saída. Quando este fenômeno ocorre, ele pode produzir uma falha de sincronização, isto é, o valor de saída do elemento de armazenamento pode ser interpretado como distinto do valor correto por circuitos subseqüentes.

4.1.2 PROTOCOLOS DE COMUNICAÇÃO

O protocolo de comunicação assíncrono mais simples e mais comum é conhecido como handshake. Este protocolo utiliza dois sinais, geralmente denominados de *request* (Req) e *acknowledge* (Ack), para controlar um processo de transmissão de dados ou de sincronização. O canal de dados é opcional, uma vez que o protocolo pode ser utilizado apenas para sincronização entre os módulos, sem troca de dados. No caso de troca de dados, um canal do tipo *push channel*, ou seja, um canal onde a fonte de dados é o mestre do protocolo de handshake. Uma forma alternativa desse protocolo é a utilização de *pull channels*, onde o destino dos dados atua como mestre.

De acordo com o número de sinalizações envolvidas no protocolo handshake têm-se duas versões deste: protocolo de quatro fases e protocolo de duas fases.

4.1.3 CODIFICAÇÃO DE DADOS

A codificação de dados em um circuito digital pode ser feita de diversas formas. A codificação mais comumente empregada é conhecida como trilha única (em inglês, *single rail* - SR). Em circuitos não-síncronos, a utilização do protocolo de *handshake* associado a dados codificados em trilha única determina o estilo de projeto denominado *bundled data*. Nesse estilo, o sinal de requisição de comunicação (Req) é responsável também por sinalizar a validade dos dados transmitidos. Para o funcionamento do estilo *bundled data* é necessário que o atraso do sinal de requisição seja maior que o atraso de todos os sinais de dados. Dito de outra forma, a máquina de estados responsável por produzir o sinal de requisição deve ser projetada de forma que a geração do sinal de requisição só aconteça após os dados estabilizarem na entrada do receptor.

Para eliminar a restrição de temporização imposta à geração do sinal de requisição no estilo *bundled data*, projetistas de circuitos assíncronos exploram possibilidades de codificação capazes de transportar o dado juntamente com a sua informação de validade.

Tais tipos de codificação permitem o desenvolvimento de protocolos de comunicação onde não é necessária a verificação de atendimento de restrições temporais na troca de dados. Tais codificações são conhecidas como insensíveis a atrasos (do inglês, *delay insensitive* – DI).

Em uma codificação DI o sinal de requisição é embutido como parte inseparável dos dados, mas o sinal de reconhecimento (Ack) ainda é necessário. Assim, o receptor deve ser capaz de verificar a validade dos dados, bem como sua presença ou ausência no canal de comunicação. Códigos especiais são reservados por esquemas DI para indicar a ausência de dados, os chamados *códigos neutros*. Outro nome para códigos neutros é *espaçadores*. Existem várias formas de codificação DI, mas as duas formas mais utilizadas são o código trilha dupla (DR) e o código m-de-N [MAR06].

4.1.4 IMPLEMENTAÇÃO DE COMPONENTES ASSÍNCRONOS

A implementação de dispositivos e circuitos assíncronos pode ser feita tanto sobre informação codificada usando trilha única, através de circuitos combinacionais semelhantes aos utilizados em lógica síncrona, quanto sobre informação expressa via codificações DI. A seguir exploram-se essas formas de implementação. Antes, contudo será apresentado o C-element de Muller, um componente básico em diversos estilos de projeto assíncronos, e também empregado em vários pontos neste trabalho.

O C-ELEMENT

Trata-se de um componente de importância em muitos estilos de projeto assíncrono. Ele também é chamado de C-element de Muller [BER92] [MAR06]. O C-element funciona como um sincronizador de eventos, produzindo um evento na sua saída quando todas as suas entradas recebem eventos específicos. A Tabela 4.1 é uma especificação de comportamento de um C-element de 2 entradas, onde a e b são os sinais de entrada e Z_i é o sinal de saída. O C-element gera 0 na sua saída quando ambas as entradas são 0, 1 na saída quando ambas as entradas são 1 e mantém o valor anterior para qualquer outra configuração de entradas. Logo, trata-se de um circuito seqüencial que pode ser implementado de diversas formas.

Tabela 4.1 Tabela verdade de um C-element de Muller com 2 entradas.

a	b	Z_i
0	0	0
0	1	Z_{i-1}
1	0	Z_{i-1}
1	1	1

Para FPGAs, a maneira mais simples de implementar um C-Element é apresentada na Figura 4.1 (a). O circuito possui duas entradas e sua saída é realimentada. Assim, um C-Element de duas entradas pode ser implementado em uma LUT (do inglês, Look Up Table) de 4 entradas (o que representa apenas metade das LUTs

disponíveis em um *slice* nas famílias Virtex II e Spartan 3, por exemplo). C-elements são na realidade uma família de componentes. Outros tipos de C-element podem ser obtidos variando o número de entradas, a polaridade de entradas ou acrescentando sinais de controle (como *reset* e/ou *set*). O sinal de realimentação de um C-element constitui uma derivação que deve ser isócrona assimétrica, ou seja, o fio de realimentação deve possuir atraso menor que o fio de saída [BER92].

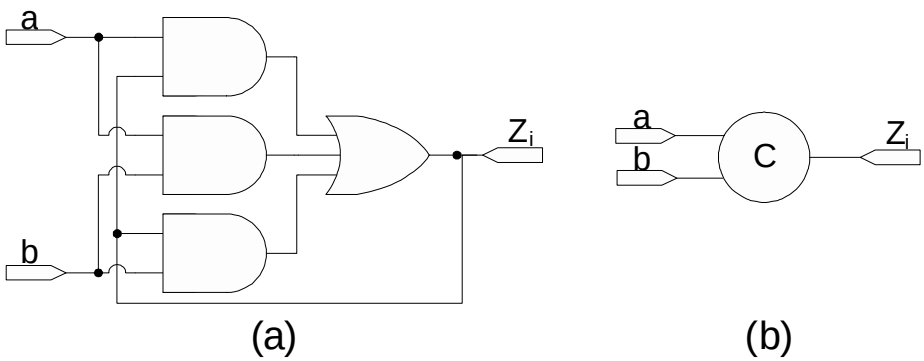


Figura 4.1 (a) Uma forma de implementação de um C-Element de Muller de 2 entradas usando portas lógicas. (b) Símbologia comumente usada para representar um C-Element.

PORTAS DELAY INSENSITIVE MINTERM SYNTHESIS (DIMS)

Existem diversas técnicas de implementação de blocos funcionais para codificação trilha dupla [BRE05] [DAV92] [SPA02]. Um exemplo de técnica empregada para construir blocos funcionais é *Delay Insensitive Minterm Synthesis* (DIMS). Essa técnica utiliza um conjunto de C-elements para gerar todos os mintermos das variáveis de entrada. Uma porta OR é usada para somar os mintermos que levam a saída ao estado de *set* (‘1’) e outra para somar os mintermos que levam a saída ao estado de *reset* (‘0’). A Figura 4.2 mostra um exemplo de porta XOR DIMS de duas entradas, que pode ser usada para a construção de blocos funcionais.

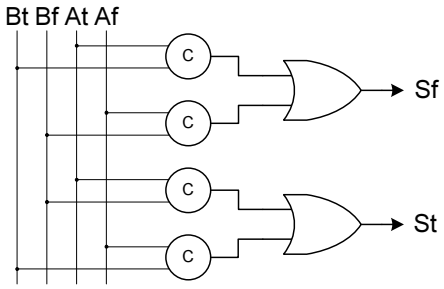


Figura 4.2 Exemplo de uma porta XOR implementada com a técnica DIMS.

4.2 FUNDAMENTOS DA LÓGICA STTL

Para uma melhor compreensão da influência do deslocamento temporal sobre o traço de corrente de uma célula em trilha dupla, a Figura 4.3 apresenta uma abstração deste efeito. A topologia de uma célula em trilha dupla pode ter seu comportamento abstraído como mostrado na Figura 4.3 a). Nesta Figura, AT_T e AT_F representam respectivamente os tempos de chegada dos fios de entrada de um inversor trilha dupla

teórico, τ representa a duração da transição dos sinais e Δt o deslocamento temporal entre os inícios das variações nas duas entradas. Como ilustrado na Figura 4.3 b), o traço diferencial de corrente é a diferença entre os traços de corrente de INV1 e INV2. Assume-se, sem perda de generalidade, tratar-se de inversores.

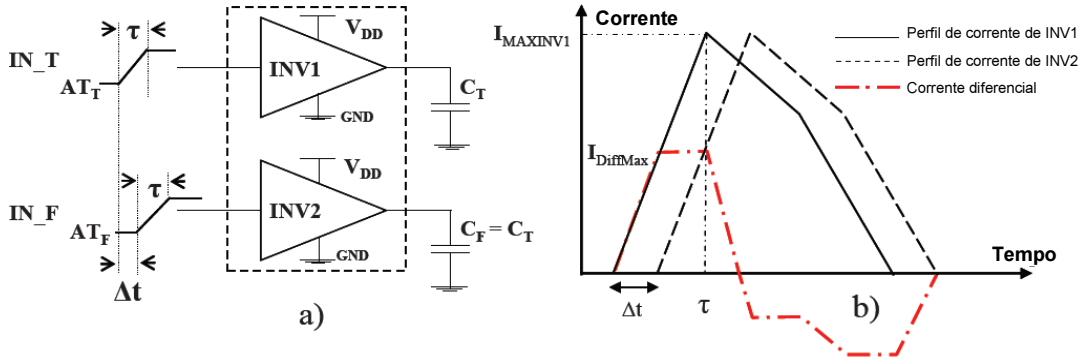


Figura 4.3 Impacto do deslocamento temporal sobre a corrente diferencial [RAZ07].

Como se pode constatar, um deslocamento temporal Δt nas entradas resulta em um deslocamento de mesma ordem nos traços de corrente. Por conseqüência, durante este tempo Δt, o traço diferencial de corrente é igual ao traço de corrente do INV1, cuja amplitude máxima $I_{MAXINV1}$ pode ser atingida se Δt = τ.

Neste contexto, o objetivo em propor STTL é garantir que este deslocamento temporal Δt não seja perceptível no nível de corrente em células trilha dupla. A Figura 4.4a) mostra a idéia principal por trás de STTL usando uma célula produtora A e uma célula consumidora B. É possível imaginar as células em trilha dupla usando um sinal de validade W, responsável por habilitar que B use a saída de A ao final de um tempo $t \geq \Delta t_i$. Este tempo é ajustável e deve garantir que os sinais de entrada da célula B estejam em um estado válido e estável. Analisando a Figura 4.4 b), nota-se que se o deslocamento temporal entre as entradas S_F e S_T de B não excede Δt_i, isto garante que não existirá nenhum impacto nos traços de corrente do módulo B. Em outras palavras, espera-se com isto que os traços de corrente do módulo B sejam quase superpostos e por conseqüência insensíveis a qualquer deslocamento temporal existente entre os sinais S_F e S_T, desde que este deslocamento não exceda Δt_i.

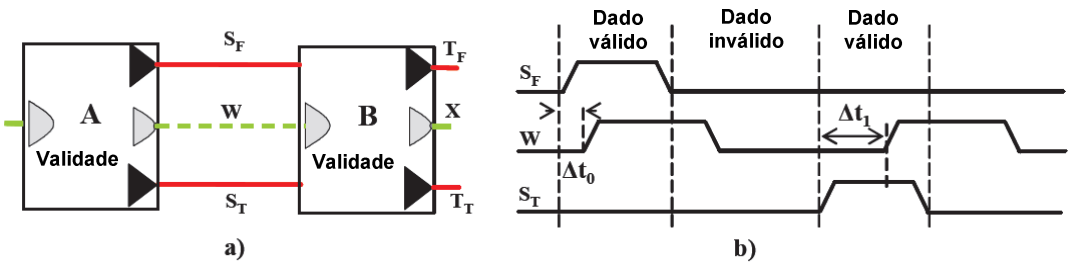


Figura 4.4 a) Estrutura tolerante ao deslocamento temporal. b) Modo de funcionamento da estrutura tolerante a deslocamento temporais [RAZ07].

Com base nesta premissa, os Autores de [RAZ07] propuseram a lógica STTL. De um modo geral, esta lógica tem características bem próximas da lógica em trilha dupla (DPL), porém com uma codificação de dados e um modelo de funcionamento particulares.

Diferentemente das lógicas DPL existentes, STTL utiliza três fios ao invés de dois para codificar um bit de informação. Dois fios servem para a codificação de dados enquanto que o terceiro fio serve para identificar o estado de validade dos dados. A Figura 4.5 apresenta a codificação de dados adotada por STTL. Como se pode constatar, não se trata de uma codificação em três trilhas (1 de 3), a informação contida neste terceiro fio é redundante e corresponde à validade dos dados. A conseqüência direta é que em um ciclo de processamento, este terceiro fio comuta a VDD na fase de avaliação e depois retorna a 0 (GND) na fase de pré-carga. Tendo uma atividade completamente independente dos dados tratados, este terceiro fio não apresenta nenhuma incidência particular sobre os traços hipóteses DPA.

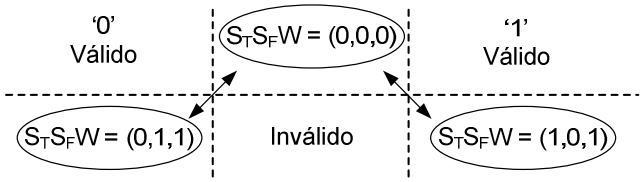


Figura 4.5 Codificação de dados utilizada pela lógica STTL [RAZ07].

A Figura 4.6 ilustra o funcionamento da lógica STTL através de um circuito exemplo.

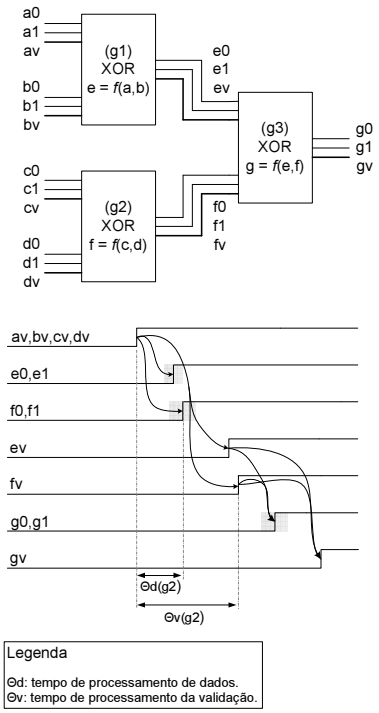


Figura 4.6 Operação básica das portas STTL. Θd representa o tempo de processamento de dados pela porta. Este tempo pode sofrer variações como indicado pelo retângulo cinza, os quais não afetam os sinais de validação, conseqüentemente mantendo ΘV constante.

Depois de ativados os sinais aV, bV, cV e dV (assumindo-se que ocorram ao mesmo tempo), e0, e1, f0, f1 estabilizam primeiro. A seguir, a validação ocorre em eV e fV, que habilitam o chaveamento de g0 ou g1, seguido por gV, assumindo-se (como especificado pela lógica) que os sinais de validade tenham atraso de propagação maior. Deste modo, o chaveamento das portas STTL é controlado pela trilha de validade, caracterizada por um chaveamento mais lento em relação às trilhas de dados. Em outras palavras, as trilhas de validade representadas pelas flechas pontilhadas na Figura 4.6 operam como o principal suporte de temporização dos blocos lógicos, dando seqüência aos eventos independentemente dos dados processados.

Observa-se que durante a seqüência de chaveamento, os tempos de ativação de e0 (f0, g0) e de e1 (f1, g1) podem variar devido aos possíveis desequilíbrios de carga de saída. Isto é representado pelos hachurados cinza na Figura 4.6. Entretanto, estes desequilíbrios de tempo não afetam a operação das portas seguintes, pois são ativadas pelos sinais de validação. Esta característica evita os efeitos de acúmulo de desequilíbrios ao longo do caminho de dados. Isto garante um consumo de potência e tempo de processamento quase independente de dados no circuito.

4.3 PROTOTIPAÇÃO DA LÓGICA STTL EM FPGA

Esta Seção apresenta a proposta de uma biblioteca de portas lógicas STTL baseadas nos pressupostos de projeto de circuitos não-síncronos orientados a dispositivos reconfiguráveis. O protótipo da biblioteca lógica é seguir as premissas definidas pelos Autores em [RAZ07], de modo que seja possível avaliar sua robustez através de análises reais de consumo de potência e de radiação eletromagnéticas.

A maior parte dos circuitos implementados em FPGAs baseia-se no uso de ferramentas de síntese automática. Entretanto, pode ser difícil satisfazer requisitos de temporização rígidos (como os exigidos em módulos assíncronos). Também, obter um percentual alto de uso de um dispositivo FPGA pode não ser factível via síntese automática apenas. Para tentar superar estas restrições, a maioria das ferramentas de síntese automática dá suporte à especificação de uma ampla variedade de restrições, indo desde especificações de planta baixa e freqüência de operação mínima até o controle de atrasos em fios específicos. Ainda assim, restrições são apenas guias para ferramentas de alto nível e um processo de síntese pode chegar ao seu final violando uma ou mais das restrições impostas. Um dos recursos oferecidos em FPGAs da Xilinx para se ter um controle do processo de síntese são as chamadas *hard macros*. *Hard macros* são células definíveis em nível de leiaute do FPGA, através de uma ferramenta (*FPGA Editor*) que dá acesso a fios, LUTs e outros elementos específicos de um dispositivo FPGA específico de uma família específica. A liberdade de usar os recursos do FPGA não é absoluta, mas suficientemente detalhada para permitir criar portas STTL que respeitam restrições de temporização impostas pela lógica.

Como FPGAs são em essência matrizes bidimensionais de componentes idênticos, uma vez projetadas, hard macros podem ser utilizadas como uma célula de biblioteca, instanciadas em descrições de alto nível (como VHDL ou Verilog). Elas são manipuladas

como um módulo de leiaute rígido por ferramentas de síntese lógica e física. Partes críticas em funcionalidade e/ou temporização podem assim ser projetadas a mão. Hard macros provêem controle sobre projetos, obviamente ao custo de complexidade adicional.

Hard macros não são novidade em projeto de FPGAs. Elas foram usadas, por exemplo, por Martín-Langerwerf et al. em [MAR02] para reduzir o número de FPGAs e o tempo de execução para aplicações de tratamento de vídeo. Além disto, hard macros já foram usadas para habilitar o uso de reconfiguração dinâmica e parcial de FPGAs, como descrito, por exemplo, em [HUE04] e [MOL06]. Aqui se emprega hard macros para implementar primitivas assíncronas da lógica STTL, viabilizando o uso desta lógica em FPGAs de forma compacta. Estudos realizados por Pontes et al. [PON07] mostram que FPGAs da Xilinx possuem recursos que habilitam a implementação da biblioteca pretendida.

Quatro versões de portas lógicas AND são desenvolvidas com base nos pressupostos DIMS de concepção assíncronos de circuitos. Como exemplo, a Figura 4.7 apresenta quatro diferentes versões implementadas da porta lógica AND de duas entradas. Inicialmente, desenvolveram-se duas versões em trilha dupla (DR), em (a) uma versão DR DIMS conforme definida na literatura e em (b) uma versão DR DIMS balanceada, ou seja, as portas de saída Z_1 e Z_0 tendo a mesma profundidade lógica. A seguir, desenvolveram-se duas versões de protótipos da lógica STTL. A versão (c) mostra uma lógica baseada nos pressupostos DIMS, porém adaptada às premissas definidas em [RAZ07]. A seguir, uma versão empregando menos recursos (d) é definida, apresentando o mesmo comportamento de (c), mas com vantagens em termos de área e tempo de cálculo.

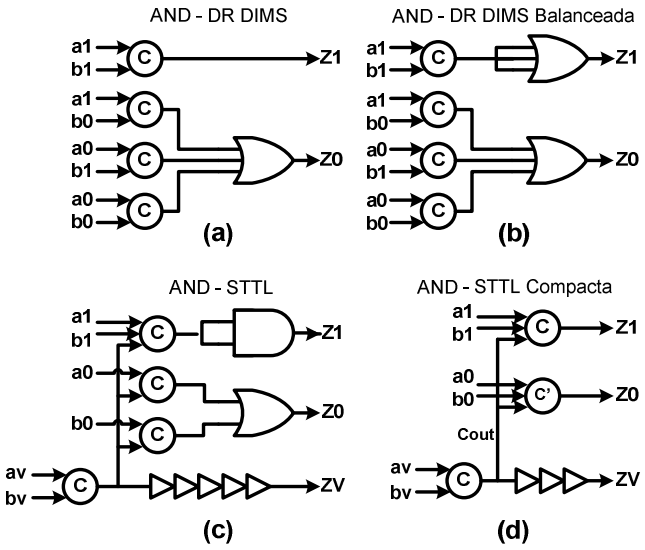


Figura 4.7 Proposta de estruturas lógica e física de uma porta lógica AND assíncrona de duas entradas. Em (a) a versão básica DR DIMS (DR), em (b) uma versão DR segura (DR2), em (c) o primeiro protótipo de STTL (STTL) e em (d) uma versão compacta de STTL (STTL2). A letra C dentro dos círculos representa um C-element e o símbolo C' é um C-element especial de 3 entradas, cujo comportamento de saída é expresso pela equação booleana Z0 = Cout.Z0 + (Z0+Cout).(a0+b0).

Como a Figura mostra, a lógica é composta por C-element de Muller [MAR06], responsáveis por evitar que transições espúrias ocorridas nas entradas sejam propagadas através de portas lógicas tradicionais OR, AND, NAND. Para implementar estas portas em lógica STTL, é necessário que os caminhos de dados da lógica-verdadeira (Z_1) e lógica-falsa (Z_0) tenham a mesma profundidade lógica. Esta característica é importante para se obter consumo de potência e tempo de propagação quase independentes dos dados em cada célula.

Um C-element de três entradas é usado com o objetivo de otimizar o protótipo da lógica STTL apresentado na Figura 4.7 (c). A função lógica de C' apresenta o mesmo comportamento das portas envolvidas na geração da saída Z_0 na Figura 4.7 (c). Isto permite reduzir o número de LUTs necessárias para implementar a lógica e conseqüentemente reduzir o atraso necessário para o sinal de validade. A Figura 4.8 apresenta a implementação da porta AND STTL da Figura 4.7 (d) sobre um bloco lógico configurável (do inglês, *Configurable Block Logic* - CLB) de um FPGA Xilinx da família Spartan3.

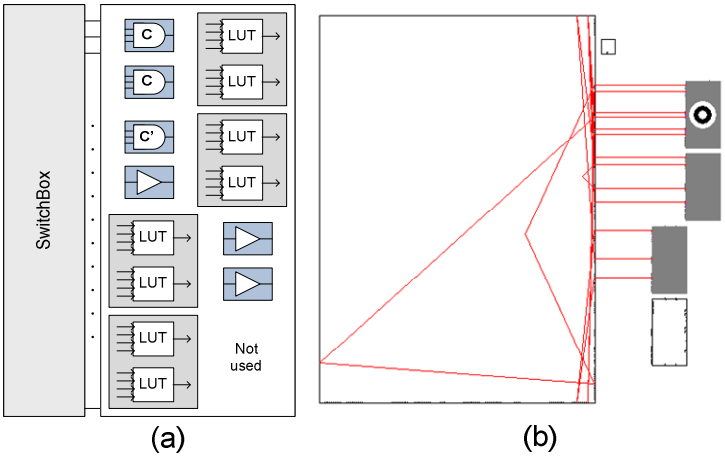


Figura 4.8 (a) Abstração de um CLB Xilinx da família Spartan3 composto por 4 slices e um switch box. Cada slice contém 2 LUTs responsáveis por implementar uma função lógica. (b) Leiaute da hard macro que implementa a porta AND indicada na Figura 4.7 (d).

A lógica de validade em portas STTL é implementada a partir das entradas A_V e B_V conforme discutido anteriormente para circuitos CMOS. Assim, uma lógica independente com tempo de propagação maior em relação aos sinais de saída Z_1 e Z_0 é projetada a partir de elementos de atraso como mostrado na Figura 4.7. Este atraso é obtido pelo uso de LUTs em cascata implementando cada uma a função identidade, de modo a obter um atraso constante para todos os dados de entrada.

O projeto de uma porta lógica AND STTL de duas entradas tal como da Figura 4.7 (c) pode ser realizado usando 11 LUTs (ou 6 slices). Destas 11 LUTs, 6 são usadas para a definição da lógica propriamente dita e 5 empregadas na lógica de validação. Com esta proposta de porta lógica é possível notar que a concepção de uma porta STTL sobre o FPGA apresenta alto custo de área. Em um momento inicial, o objetivo principal desta proposta foi validar os conceitos de STTL e não necessariamente encontrar um meio ótimo de desenvolver a lógica sobre FPGAs. Porém, uma otimização se faz necessária de

modo a torná-la competitiva em relação a outras propostas disponíveis na literatura. Assim, a segunda versão (Figura 4.7 (d)) utiliza 6 LUTs, gerando um ganho de quase 50% em termos de área.

Seguindo este método de concepção, outras portas lógicas STTL foram implementadas, incluindo: NAND, OR, NOR, XOR e XNOR. No caso das portas lógicas XOR e XNOR não é possível usar a otimização dada por C'. Para isso, seria necessário construir um C-element com mais de 4 sinais de entrada. Como as LUTs dos dispositivos Spartan3 possuem apenas 4 sinais de entrada, isto exigiria que o C-element fosse implementado em duas ou mais LUTs. Isto tornaria complexa a tarefa de obedecer às exigências temporais do C-element, e ainda, não reduziria o número total de LUTs para sua implementação. Portanto estas portas lógicas são implementadas apenas com a versão inicial, onde são necessários 5 elementos de atraso tal como mostrado na Figura 4.7 (c). Já um inversor tanto em lógica de trilha dupla como em STTL é definido simplesmente como a inversão dos fios verdadeiro e falso da codificação trilha dupla, não sendo necessário o uso de componentes ativos para sua implementação.

4.3.1 FLUXO DE PROJETO E VALIDAÇÃO DA LÓGICA STTL

Antes de avaliar a robustez do processo de prototipação da lógica STTL, é necessário inicialmente validar o funcionamento das portas lógicas desenvolvidas e avaliar através de simulação se os tempos de processamento são independentes dos dados de entrada. Para isso o submódulo SBOX1 do algoritmo DES associado a uma função XOR de entrada foi escolhido como estudo de caso. O motivo para tal escolha é o simples fato deste submódulo ter sido usado anteriormente pelo grupo de pesquisa, reduzindo assim o tempo de sua validação. Outros algoritmos mais recentes ou submódulos destes, tal como o AES e sua SBOX, também poderiam ser usados neste estudo de caso.

O submódulo do DES nada mais é que uma função combinacional com seis bits da mensagem de entrada, 6 bits da subchave criptográfica e 4 bits de saída para a mensagem cifrada, tal como mostrado na Figura 4.9. Maiores detalhes podem ser obtidos no Anexo B deste trabalho.

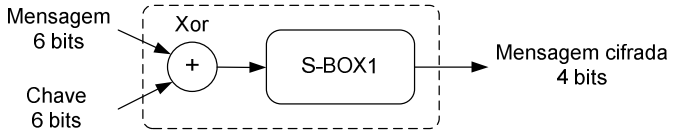


Figura 4.9 Submódulo de algoritmo DES.

Inicialmente é necessário mapear esta função para portas lógicas tradicionais de duas entradas, já que a biblioteca de protótipos da lógica STTL possui apenas portas de duas entradas. Descrições de hardware com este tipo de restrição podem ser obtidas a partir de uma descrição do submódulo DES em linguagem de descrição de hardware como VHDL ou Verilog e de um sintetizador de hardware. O sintetizador deve ser parametrizado, de modo a gerar uma descrição do circuito desejado em portas lógicas de

duas entradas, descrição esta denominada de *netlist* pelas ferramentas de síntese de hardware [XIL10].

Com a geração do netlist do submódulo DES verificou-se que o circuito combinacional é composto por 175 portas lógicas de duas entradas. Este número significativo de portas lógicas para um estudo de caso exigiu uma ferramenta para automatizar a transcrição da descrição em trilha única para três trilhas como exigido pela lógica STTL. Descrições de hardware de circuitos mais complexos tais como os algoritmos DES e AES exigem um número significativamente maior de portas lógicas, o que pode se tornar inviável de ser realizado manualmente. Deste modo, foi necessário propor uma nova etapa ao fluxo de projeto para automatizar o desenvolvimento de circuitos com a lógica STTL. Esta nova etapa pode ser vista no fluxo de síntese de hardware da Xilinx, como mostra a Figura 4.10.

O fluxo de projeto tem como entrada a descrição VHDL ou Verilog do circuito codificado em trilha única. A etapa inicial de síntese de hardware deve ser executada para gerar o netlist. O sintetizador XST da Xilinx (do inglês, *Xilinx Synthesis Technology* - XST) gera arquivos netlists em formato NGC, um formato específico da Xilinx que contém informações do projeto lógico e também restrições de projeto [XIL10], sendo complexo para ser usado como entrada para a etapa de tradução. Deste modo é necessário usar um sintetizador tal como o *Encounter RTL Compiler* da Cadence [CAD05] que gera netlist em formato Verilog, o que simplifica o processo de tradução na etapa seguinte do fluxo.

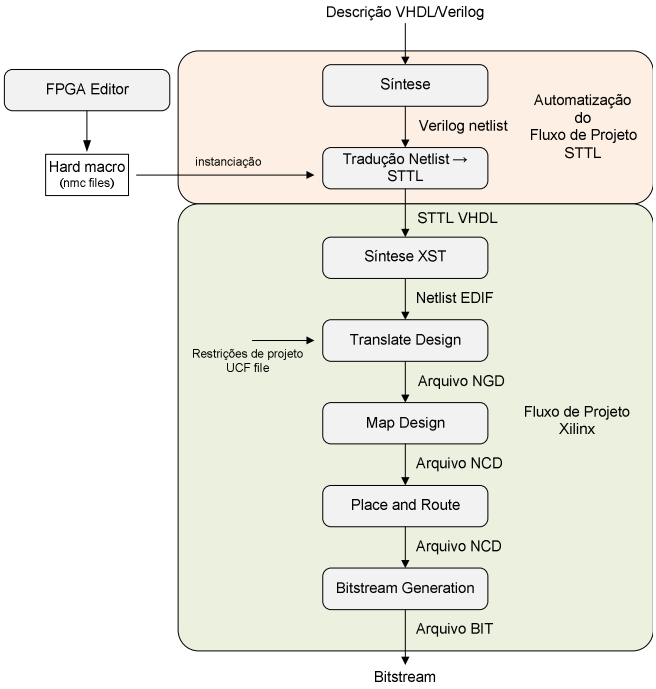


Figura 4.10 Adição de uma nova etapa no fluxo de projeto da Xilinx para automatizar o projeto de circuitos usando lógica STTL.

A etapa de tradução do netlist em trilha única para uma descrição com portas lógicas STTL foi desenvolvida em linguagem C, sendo uma das contribuições deste

trabalho. Esta etapa gera como saída uma descrição do circuito em três trilhas, instanciando as hard macros contidas na biblioteca STTL. Ao final desta etapa se obtém um arquivo com a descrição VHDL do circuito STTL que é usado como entrada para o fluxo de projeto da Xilinx.

Como um grande número de hard macros é instanciado no projeto, é ainda necessário realizar restrições de posicionamento dos componentes envolvidos no projeto, como indicado na Figura 4.10 na etapa "*Translate Design*". Isto permite que a ferramenta de síntese física tenha capacidade de realizar o posicionamento e roteamento de fios a fim de interconectar todos componentes do projeto satisfazendo restrições temporais de projeto. Assim, quanto mais recursos de um dispositivo prototipação forem utilizados, mais complexa é a tarefa de encontrar restrições de posicionamento que satisfaçam as exigências do projeto. Maiores detalhes sobre estas etapas de síntese de hardware podem ser obtidos em [XIL10].

O script de tradução e o fluxo de projeto completo foram validados com o circuito da SBOX1. Simulações explorando todas as possíveis 64 entradas, a análise e validação dos dados de saída comprovaram o funcionamento correto do circuito. Embora o script seja útil aos propósitos do projeto, este ainda limita-se a traduzir apenas descrições de portas lógicas, não sendo capaz de traduzir estruturas mais complexas tal como flip-flops, latches, multiplexadores, entre outros componentes comuns em projeto de circuitos digitais.

Este fluxo de projeto permite que a etapa de tradução seja alterada para implementar circuitos em trilha dupla, tal como o estilo DIMS apresentado anteriormente. Assim, é possível realizar outros estudos de caso, visando comparar a versão STTL com a implementação DIMS, por exemplo.

De um modo geral, o fluxo de projeto baseado em hard macros apresenta uma restrição. As hard macros quando desenvolvidas são dependentes da tecnologia do dispositivo escolhido para sua prototipação. Por exemplo, uma hard macro construída para o dispositivo XC3S200 da família Spartan3 não pode ser reusado em um projeto sobre o dispositivo da família Spartan3E. Para contornar isto, a Ferramenta FPGA Editor permite a geração de scripts a fim de automatizar a construção de hard macros para dispositivos diferentes. Este script registra todos os passos usados para criar a hard macro. Ao final, esta mesma hard macro pode ser construída sobre outros dispositivos simplesmente por executar o script especificando o dispositivo ao qual se deseja projetar. Neste trabalho scripts foram criados para construir as hard macros usadas nos estudos de caso, sendo esta outra contribuição deste trabalho.

4.3.2 AVALIAÇÃO DO TEMPO DE CÁLCULO E DE ÁREA DE PROTÓTIPOS STTL

A nova etapa de tradução do fluxo de projeto permite a obtenção de quatro versões diferentes do submódulo tomado como estudo de caso: Uma implementação para cada versão descrita na Figura 4.7, além da versão original, projetada em trilha única. Com a

realização da síntese completa de todas as versões é possível avaliar e comparar os tempos de cálculo e os custos em área da prototipação.

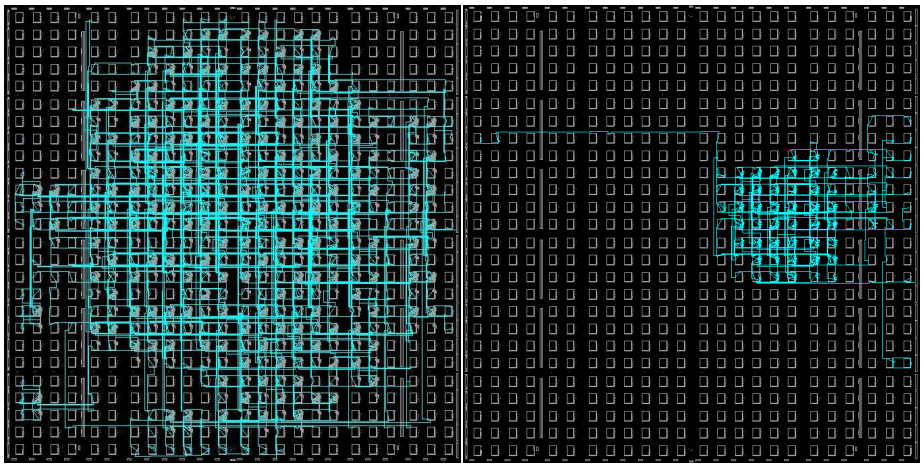
Para avaliar a dependência entre dados e o tempo de cálculo foram realizadas simulações pós-síntese com os circuitos, visando medir o tempo de cálculo para cada uma das combinações possíveis de dados de entrada. Para isso, fixa-se uma das entradas (chave) em um valor aleatório e aplica-se um vetor à entrada de dados com as 64 combinações de valores possíveis, registrando o tempo de cálculo para cada dado processado. O experimento foi realizado nas 5 versões do submódulo, todos prototipados no dispositivo Spartan XC3S200 da Xilinx e simulados com a ferramenta ModelSim da Mentor [MEN10].

Os resultados apresentados na Tabela 4.2 demonstram que o tempo de cálculo do submódulo STTL é rigorosamente constante para todos os dados de entrada, como se esperava. Por outro lado, o tempo de cálculo de STTL é 5 vezes maior que o tempo de cálculo obtido na versão trilha única. A lógica de validação independente implementada pelo atraso segundo a Figura 4.7 explica este resultado. Reduções de 48% em termos de área e 20% em termos de tempo de cálculo são obtidos com a otimização proposta. Os resultados também mostram que o tempo de cálculo reduz-se em torno de 44% ao se usar um balanceamento em lógica DR, como visto em DR2.

Tabela 4.2 Tempo de cálculo e área ocupada para implementação da SBOX1 em 3 versões diferentes, SR, DR (DIMS) e STTL.

	SR	**DR**	**DR2**	**STTL**	**STTL2**
Min (ns)	15,6	48,1	55,9	103,0	81,7
Max (ns)	26,6	58,5	61,7	103,0	81,7
Média (ns)	22,2	53,5	58,9	103,0	81,7
Diferença (ns)	10,9	10,4	5,8	0	0
Área (slices)	175	490	490	966	501
Área (%)	9	25%	25%	50%	26%

Os resultados obtidos pós-síntese revelam a área necessária para implementar cada uma das versões do submódulo, como apresentado na Tabela 4.2. O custo em termos de área para se obter tempo de cálculo independente de dados é na ordem de 5 vezes em relação à implementação tradicional. As plantas-baixas dos submódulos implementados em STTL e trilha simples são mostradas na Figura 4.11.



(a) (b)

Figura 4.11 Plantas-baixas obtidas com o processo de síntese da SBOX1 do algoritmo DES a) usando lógica STTL e b) usando lógica tradicional.

4.4 ALGORITMO DES STTL

Após a validação da biblioteca STTL para prototipação em FPGAs, com a implementação do submódulo SBOX1, a etapa seguinte é projetar o algoritmo DES com a biblioteca. Este estudo de caso permite avaliar a robustez da lógica STTL através da execução de um algoritmo difundido. O algoritmo DES é composto basicamente por uma função *F* (*Feistel Function*), funções de permutação e expansão de dados. A função *F* é puramente combinacional, sendo factível sua implementação em lógica STTL. As funções de permutação e expansão resumem-se a permutações de fios no projeto de hardware, não representando grande complexidade para o projeto. Uma lógica seqüencial de controle é utilizada na geração de subchaves criptográficas e no controle da realimentação de dados parcialmente cifrados durante as 16 rodadas de execução do algoritmo. Mais detalhes do algoritmo DES serão apresentados no Capítulo 5.

A implementação parcial do DES em STTL descrita na Seção 4.3 serve como ponto de partida para criar uma versão completa do algoritmo [DEH10]. Esta implementação utiliza uma abordagem híbrida, do ponto de vista de sincronização, devido à complexidade de lidar com grandes quantidades de hard macros ao longo do fluxo de projeto de FPGAs. Nesta abordagem, registradores e a lógica de controle do laço de execução do DES são módulos síncronos, enquanto que a implementação das SBOXes e outras partes da lógica combinacional (operações XOR) são realizadas com hard macros STTL. Além disto, o controle seqüencial de geração de subchaves mantém-se síncrono, porém adaptado a interface em três trilhas de STTL. A Tabela 4.3 apresenta os resultados de área e latência para implementar esta versão do algoritmo DES usando a biblioteca STTL2 e a comparação com uma versão completamente síncrona do DES. Dada a natureza híbrida da implementação STTL pode-se esperar que uma implementação completamente assíncrona conduza a áreas maiores, devido à expansão dos módulos que empregam lógica trilha tripla, e possivelmente uma maior robustez a SCA.

Nota-se claramente o alto custo em área da implementação STTL2. Embora apresente um aumento de aproximadamente 20 vezes em relação à implementação síncrona, STTL2 é compatível em termos de área com outras DPLs. As simulações usadas na validação do algoritmo mostram que o tempo de processamento é constante e independente de dados, diferentemente da implementação regular, onde ocorre uma variação de até 2 ns no tempo de execução quando submetido ao processamento da mesma seqüência de dados.

Tabela 4.3 Avaliação de área e latência para as implementações convencional e STTL2 do algoritmo DES.

	DES Regular	**DES STTL2**
Min (ns)	5	140
Max (ns)	7	140
Diferença (ns)	2	0
Área (slices)	267	5130

4.5 SISTEMA DE MEDIÇÃO DE TRAÇOS DE CONSUMO DE POTÊNCIA E DE RADIAÇÃO ELETROMAGNÉTICA

Com a validação da biblioteca de portas lógicas STTL e a verificação por simulação da premissa tempo de cálculo independente de dados, a etapa seguinte é avaliar a robustez da lógica proposta, através das análises de potência DPA. Como visto anteriormente, no Capítulo 2, a primeira fase das análises é a medição e armazenamento dos traços de consumo de potência de cada dado processado. Logo, para realizar estes experimentos é necessário um sistema de medição preciso, capaz de medir consumo de potência em circuitos prototipados em uma plataforma FPGA.

O sistema proposto nesta Seção também foi utilizado para análises de radiação eletromagnética produzida pelo circuito, usando SEMA, DEMA e CEMA. Para isto, é necessário apenas substituir a sonda eletromagnética de corrente por uma sonda eletromagnética adequada às características de freqüência das ondas emitidas pelo circuito. Como estas ondas possuem pequena intensidade, utiliza-se também um amplificador de sinal.

Por definição, *potência ou potência instantânea* (P) é o produto entre os valores instantâneos da *tensão elétrica* (V) e da *corrente elétrica* (I), $P = V(t) \cdot I(t)$ [TOR02]. Neste caso, considerando que a tensão de alimentação de um dispositivo FPGA é constante, a potência consumida será diretamente proporcional às variações de corrente elétrica produzidas pelo chaveamento do circuito durante o processamento dos dados de entrada. Na literatura encontram-se basicamente dois modos de medição de potência: *(i)* usando um resistor em série com a fonte de alimentação, onde a variação de tensão sobre o resistor permite encontrar a corrente elétrica no dispositivo segundo a Lei de Ohm $I = \frac{V}{R}$

[TOR02]; e *(ii)* usando uma sonda eletromagnética para medição de corrente. Experimentos com ambos os modos de medição foram realizados. As medições de potência usando a sonda eletromagnética apresentaram menor ruídos e, portanto foi adotada nestes experimentos.

Para realizar a primeira fase do ataque DPA define-se o sistema de medição de potência como ilustra a Figura 4.12. Neste caso, um osciloscópio preciso é o equipamento mais indicado para medir o consumo de um circuito. Uma sonda eletromagnética de medição de corrente conectada a um canal de entrada do osciloscópio serve como sensor de corrente do sistema. A plataforma de prototipação contém o protótipo do sistema alvo da medição e um dispositivo de comunicação para entrada e saída de dados, neste caso uma porta de comunicação serial. Um hospedeiro deve gerar os dados necessários a serem medidos e armazenar seus respectivos traços de consumo de potência.

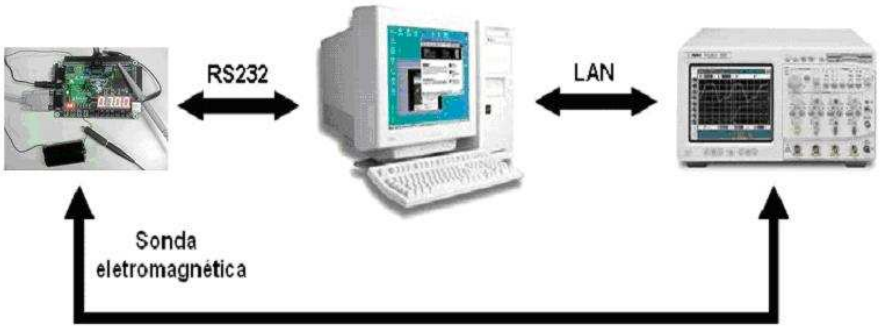


Figura 4.12 Sistema de medição e armazenamento de traços de potência para avaliação da robustez da lógica STTL.

Além destes equipamentos, é necessário um sistema embarcado para servir de interface entre o circuito em avaliação e o restante do sistema. Este sistema deve ser capaz de indicar ao osciloscópio o momento exato do final do processamento para o circuito em avaliação. Deste modo, o hospedeiro armazena o traço de consumo referente apenas ao período de tempo em que o dado é processado, permitindo medições precisas. Além disso, a plataforma de prototipação deve sofrer uma modificação no circuito de alimentação do núcleo FPGA, para que seja possível usar a sonda de corrente. A alteração consiste em substituir o regulador de tensão destinado à alimentação do núcleo FPGA por uma bateria recarregável externa, que passa a alimentar exclusivamente o FPGA na plataforma. O consumo de potência causado pelos componentes presentes na plataforma é desprezado nestes experimentos.

Os FPGAs possuem outros pinos de alimentação destinados aos circuitos de entrada e saídas de dados (E/S). Estes são organizados em grupos de pinos e possuem tensões de alimentação ajustáveis [XIL09]. Os experimentos realizados neste trabalho levam em conta apenas o consumo do núcleo de lógica programável do dispositivo onde se encontra o protótipo do circuito em avaliação. Entretanto, o chaveamento de tensão nos pinos de E/S pode gerar ruído na alimentação do núcleo do FPGA. Portanto, é aconselhável restringir ao máximo o uso de pinos de E/S de dados no sistema embarcado

projetado, ou manter estes inoperantes durante o período de medição do traço de potência.

A Figura 4.13 apresenta um diagrama em blocos do sistema embarcado proposto e desenvolvido. Este é composto por um controlador de interface RS-232, responsável por receber dados do hospedeiro através de um canal de comunicação serial. Três registradores são disponibilizados para armazenar respectivamente o dado de entrada, a chave criptográfica e o dado cifrado de saída. O sistema também possui um decodificador de números binários para o mostrador de sete segmentos, para exibição dos dados de entrada e saída do circuito avaliado, e possui obviamente o circuito alvo de medição, neste caso o submódulo SBOX1. Além disso, um módulo de controle gerencia o envio de dados ao circuito e a sinalização de disparo (*trigger*) ao osciloscópio, para que este realize a medição de corrente no momento exato de processamento dos dados. Um gerador de relógio é utilizado para reduzir a freqüência de operação do sistema. Como mostra a Tabela 4.2, a versão STTL opera em circuitos com freqüência de relógio inferiores a 10 MHz. A plataforma Digilent Spartan3 Starter Board usada dispõe de um sinal de relógio com freqüência de 50 MHz [XIL05], logo a divisão da freqüência deste sinal é necessária.

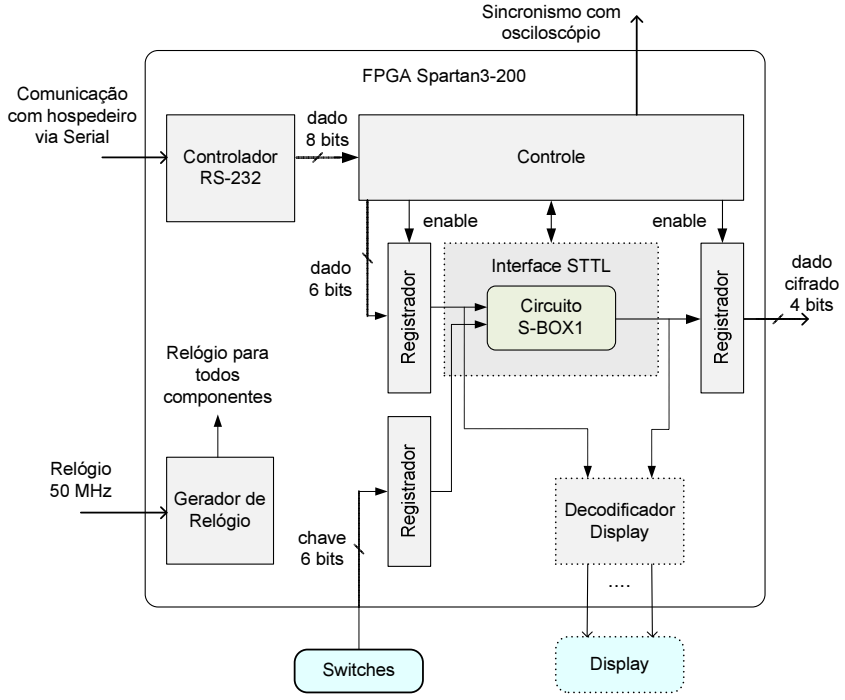


Figura 4.13 Sistema embarcado para controle de medição de potência sobre o circuito alvo. Os módulos Interface STTL, Decodificador do Mostrador 7 Segmentos são usados apenas para validação do sistema. O mesmo sistema sem estes módulos é usado para realizar as medições de consumo.

O circuito Decodificador do Mostrador é usado apenas em um instante inicial, para fins de validação do sistema. A versão do sistema sem este recurso é utilizada no processo de medição. Isto visa reduzir ruídos causados pelo chaveamento de pinos de E/S do FPGA durante o processo de medição de consumo, e assim melhorar a precisão do sistema.

Este mesmo sistema embarcado deve servir para a avaliação das 3 versões do submódulo a ser avaliado, (i) lógica trilha simples (SR), (ii) trilha dupla (DR) e (III) STTL. O sistema embarcado é projetado com lógica tradicional, porém deve dar suporte à avaliação das versões STTL e DR. Logo, é necessário uma interface para codificar cada bit de informação para uma codificação em três trilhas ou duas trilhas. Esta interface é composta por portas lógicas AND, OR e XOR, responsáveis por gerar as codificações necessárias e um controle para as fases de pré-carga e avaliação das lógicas em análise. A Figura 4.14 mostra uma simulação usada para validar o sistema embarcado projetado para medir os traços de potência de uma SBOX1 STTL.

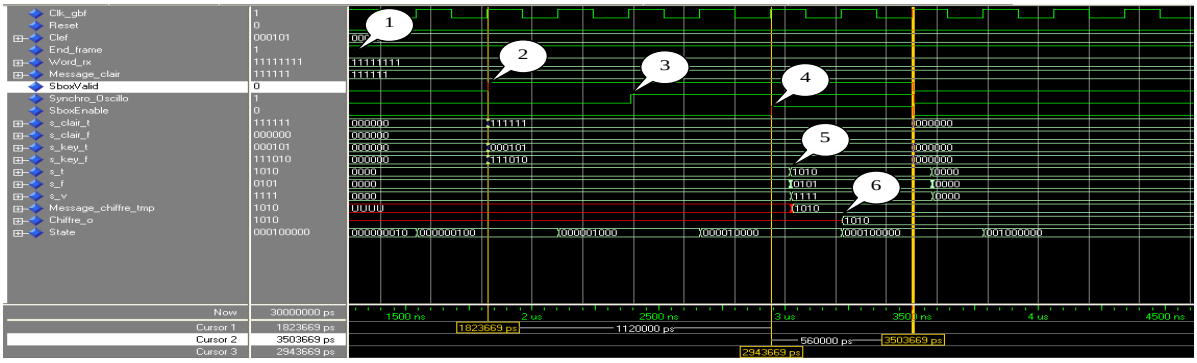


Figura 4.14Resultado de uma simulação, mostrando a ordem de eventos durante o processo de medição e coleta de dados. Em (1) end_frame indica que um dado chegou via interface serial e está disponível. Em (2), SboxValid indica a validade dos dados codificados para STTL. Em (3), synchro_oscillo sinaliza ao osciloscópio o início do processo de medição. Em (4), SboxEnable habilita o cálculo realizado pela SBOX1. Em (5), o sinal de validade 's_v' indica o fim do cálculo e a estabilidade dos sinais na saída da SBOX1. Em (6), os dados estão disponíveis na saída do FPGA.

O período entre os eventos (4) e (5) representa o tempo de processamento da SBOX1 que deve ser analisado posteriormente pelos ataques DPA. No instante 3, quando o sinal *synchro_oscillo* é ativado, um ruído intenso é produzido, fato que prejudica a análise de potência. Neste caso, o experimento tem como objetivo avaliar a eficiência da lógica proposta contra os ataques. O ruído produzido pela ativação deste sinal interfere nesta avaliação, de modo a ocultar informações do consumo de potência úteis aos ataques e conseqüentemente atrapalhando a avaliação da robustez da lógica. Portanto é necessário atrasar o início do processamento do circuito para que ocorra a estabilização do consumo do FPGA, obtendo-se assim uma melhor qualidade do sinal medido.

O hospedeiro executa um programa que gera e envia dados ao sistema embarcado pela porta serial de comunicação e armazena os dados de consumo medidos pelo osciloscópio. Este programa, inicialmente desenvolvido em linguagem C, realiza 3 operações: *(i)* lê dados a partir de um arquivo com a seqüência de dados a serem enviados e medidos pelo sistema, *(ii)* aguarda o término do processamento e da medição do traço de consumo de potência ou radiação eletromagnética pelo osciloscópio e *(iii)* armazena os traços medidos pelo osciloscópio. A sincronização entre hospedeiro, sistema embarcado e osciloscópio ocorre através de uma função disponibilizada pelo fabricante do osciloscópio que é ativada pelo sinal de gatilho enviado pelo sistema embarcado, o sinal synchro_oscillo. Ao finalizar a medição, a função é desbloqueada para realizar a leitura da

memória do osciloscópio através da rede local de comunicação (LAN). Os dados são armazenados em um arquivo em formato ASCII cujo nome deriva do dado, chave e valor de saída do submódulo. Após a medição e armazenamento dos traços de consumo ou radiação eletromagnética é encerrada a primeira etapa da análise DPA.

4.6 SISTEMA DE MEDIÇÃO ADAPTADO AO ALGORITMO DES

O sistema utilizado para medir os traços de consumo de potência e de radiações eletromagnéticas do algoritmo DES STTL passa por algumas adaptações em relação ao usado anteriormente. Neste novo sistema são enviados dados e mensagens de 64 bits ao invés de dados de apenas 8 bits. Ainda, realiza-se a verificação do resultado cifrado antes do armazenamento do traço medido. O sistema embarcado empregado para dar suporte à avaliação do algoritmo DES STTL é capaz de receber e enviar dados pela porta serial. Usa-se também uma interface para adaptar os dados codificados em trilha única para três trilhas e vice-versa.

Para medir a radiação eletromagnética emitida pela implementação do algoritmo DES STTL utilizam-se os seguintes elementos: (1) uma plataforma Digilent Spartan 3E-1600 Development Board com um dispositivo Xilinx Spartan-3E XC3S1600E, (2) uma sonda eletromagnética de 500 μm, (3) um amplificador com baixo nível de ruído (1 GHz de largura de banda e 63 dB) para amplificar o sinal obtido pela sonda e aumentar a precisão da medição, (4) um osciloscópio Agilent Infinium DS80000B (4 GHz – 40 GSa/s), e (5) um PC com suporte a scripts MATLAB para controle completo do sistema de medição.

O sistema de controle executado no hospedeiro é totalmente reescrito para o ambiente MATLAB. Como este ambiente é propício para tratamento de sinais e as análises DPA, DEMA, CEMA, CPA foram todas desenvolvidas no MATLAB, optou-se por unificar o desenvolvimento dos scripts de aquisição de dados para este mesmo ambiente. Estes scripts foram desenvolvidos originalmente no LIRMM, conforme descrito por Lomné et al. [LOM09] e Dehbaoui et al. [DEH09]. Estes foram também usados no DPA Contest 2009 [DPA09].

4.7 AVALIAÇÃO DA LÓGICA QUANTO A ROBUSTEZ AS ANÁLISES DPA E DEMA

Para desenvolver as análises DPA e DEMA é preciso coletar inicialmente os respectivos traços para todos os possíveis dados de entrada, como se discutiu anteriormente. Neste caso, como o circuito estudo de caso possui apenas 6 bits de entrada (SBOX1 do DES), é possível realizar análises exaustivas, ou seja, analisar os traços de processamento de todos os possíveis dados de entrada e todas as possíveis sequências de envio de dados. Porém, para o algoritmo DES, que usa efetivamente como entrada 56 bits de dados e 48 bits de chave, não é facial viabilizar a realização de análises exaustivas. Neste caso, define-se uma seqüencia de dados de entrada de 64 bits, sem repetições e gerados aleatoriamente. De acordo com a edição do *DPA Contest* em 2009 [DPA09], análises realizadas com menos de 200 traços são suficientes para revelar a chave secreta de um algoritmo criptográfico sem prevenções a ataques DPA.

Para que os traços coletados tenham uma influência de ruídos reduzida, o traço referente a cada dado é obtido a partir da média de 50 medições sucessivas do processamento deste dado. Isto aumenta a quantidade de informação e reduz a quantidade de ruído no sinal adquirido, e por conseqüência aumenta a relação sinal ruído (SNR). Uma vez finalizada a coleta de traços, aplica-se o fluxo de análise descrito na Figura 4.15.

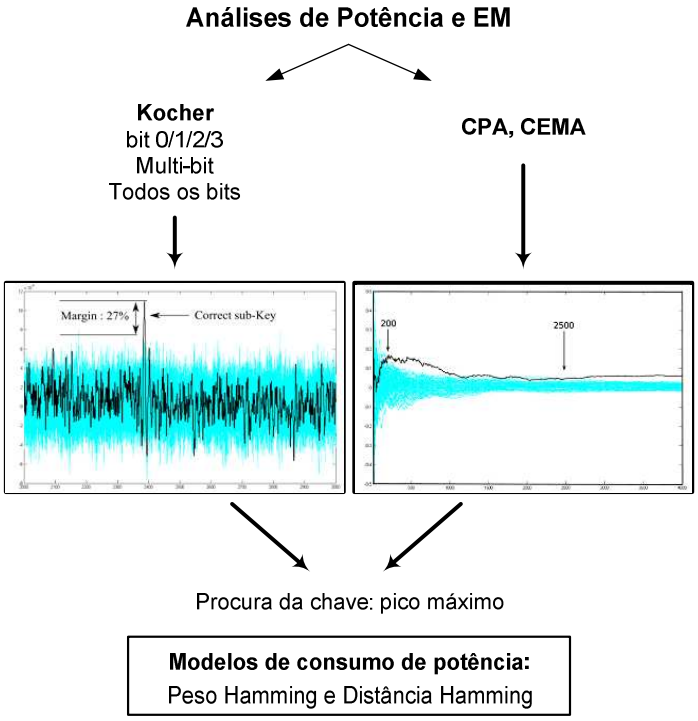


Figura 4.15 Uma visão geral do fluxo de análise empregado para validar a robustez da lógica STTL.

Conforme apresentado no fluxo, as análises diferenciais de potência propostas empregam diferentes funções de seleção. Além da função de seleção proposta por Kocher [KOC99], que analisa cada bit de saída da SBOX1 individualmente, utilizam-se também análises multibits [BEV03]. Neste caso, os traços são selecionados de acordo com o valor de 2 ou mais bits de saída. No caso de análises de 2 bits, os traços de potência resultantes de processamentos cujos bits analisados apresentam os valores ‘11’ e ‘00’ são recolhidos e divididos respectivamente em dois grupos V1 e V0, e descartam-se todos os demais traços. Análises considerando os 4 bits de saída da SBOX1 são também executadas da mesma forma.

Uma variante destas análises é considerar o valor HD do resultado da saída da SBOX1. Por exemplo, no caso da análise ao bit ‘0’ de saída da SBOX1, cada traço resultante de uma encriptação é selecionado de acordo com o valor HD do dado produzido. Este método pode ser aplicado também em análises multibits, sendo mais eficiente que as funções seleção propostas por Kocher, pois se aproxima do consumo de potência real de um circuito CMOS.

Outra variante de função seleção é o uso de HW. Este método é usado especificamente para análises simultâneas aos 4 bits de saída da SBOX1. Neste caso, a função seleção define que traços correspondentes a valores de saída com HW maiores que 2 pertencem a um grupo e os demais traços pertencem ao outro grupo.

As análises CPA e CEMA utilizam também HW e HD como função de seleção. Estas análises utilizam um fator de correlação linear, o qual é definido a partir do modelo de distância Hamming de consumo de potência [BRI04]. Na prática o coeficiente é calculado conforme a Equação 2.

$$\rho_{WH(R)} = \frac{N\sum W_i H_{i,R} - \sum W_i \sum H_{i,R}}{\sqrt{N\sum W_i^2 - \left(\sum W_i\right)^2}\sqrt{N\sum H_{i,R}^2 - \left(\sum H_{i,R}\right)^2}} \qquad (2)$$

Nesta equação, *N* é o número de traços utilizados, W_i os traços com *i* pontos e $H_{i,R} = H(M_i \oplus R)$ a distância Hamming a partir do estado anterior R. Este fator maximiza predições corretas de hipóteses de chaves e minimiza predições erradas, gerando traços hipóteses CPA ou CEMA com maiores probabilidades de acerto [BRI04].

Como ilustram a Figura 4.16 e a Figura 4.17, todas as análises realizadas apresentam 64 traços hipóteses representados em duas dimensões, uma quantidade versus tempo.

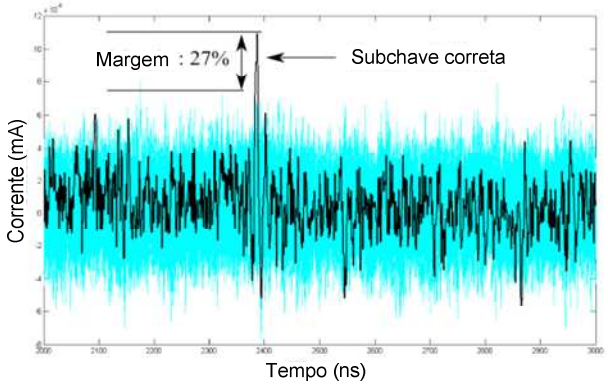


Figura 4.16 Traços hipóteses de análises DPA obtidos para a SBOX1 implementada em trilha única. O traço hipótese correto corresponde à subchave 10 com margem de 27% sobre a segunda hipótese de subchave.

Estes traços representam a probabilidade associada a cada hipótese de chave e são resultantes de uma diferença de valores de corrente, de campo eletromagnético ou uma medida de correlação. Em geral, a chave secreta corresponde teoricamente ao traço com maior amplitude. Porém, uma margem deve ser considerada na prática para garantir um maior nível de confiabilidade para se concluir uma análise. A margem é definida como a diferença mínima entre a amplitude da assinatura diferencial obtida para a chave correta e a amplitude obtida para a segunda maior amplitude. Uma análise é considerada bem sucedida se o resultado possuir margem maior que 10%.

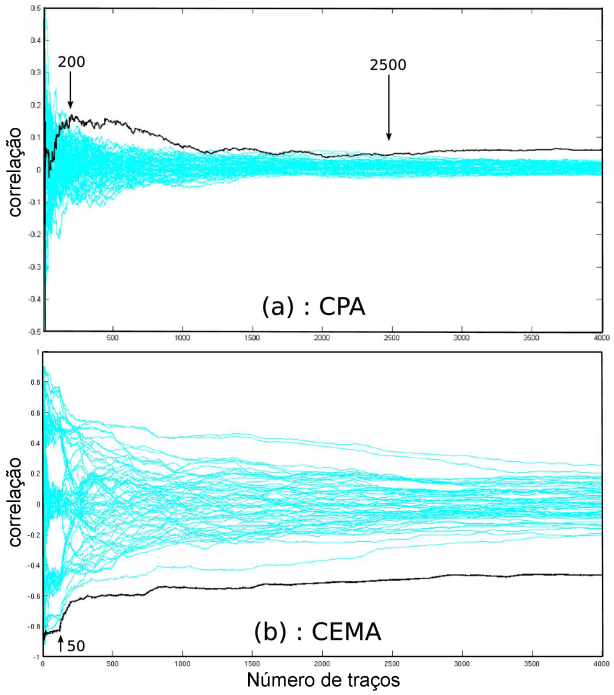


Figura 4.17 Traços hipótese em análises CPA e CEMA obtidos para a SBOX1 implementada em trilha única.

A Figura 4.18 apresenta uma avaliação da robustez de STTL contra DPA e DEMA partindo da medição do desvio padrão do consumo de corrente durante a computação do SBOX1 implementado nas seguintes versões: (i) trilha única (SR), (ii) trilha dupla (DR2) (Figura 4.7 (b)) e (iii) STTL2 (Figura 4.7 (d)). Nesta Figura é possível observar que o desvio padrão de ambas lógicas balanceadas é aproximadamente três vezes menor em relação à lógica em trilha única. Este fato valida a efetividade de trilha dupla e STTL do ponto de vista da amplitude.

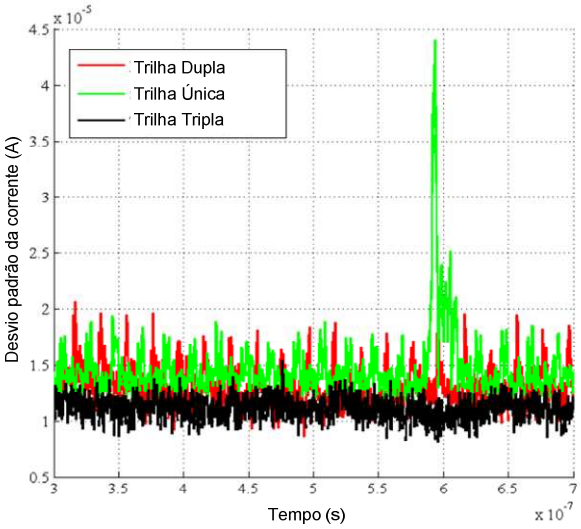


Figura 4.18 Desvio padrão da corrente consumida medido durante o processamento da SBOX1.

Todas as análises descritas no fluxo da Figura 4.15 são aplicadas na avaliação da SBOX1. Realizam-se as análises usando uma seqüência de entrada de 4033 dados. Esta seqüência contém os 64 possíveis dados de entrada com repetições, de modo a garantir que se exercitam todas as possíveis transições de um valor qualquer de dado de entrada para qualquer outro valor. O valor 4033 garante que entre o primeiro e o último dado ocorrem 4032 transições. É fácil calcular que para um vetor de n bits realizar todas as possíveis transições de qualquer valor para qualquer outro são necessárias $(2^n-1)*n$ transições. Assim é possível desta forma obter todos os traços de potência e de radiação eletromagnética para a SBOX1 em análise. Para cada valor de subchave, a maioria das análises DPA e DEMA é bem sucedida. Entretanto, as margens obtidas para análises DPA variam entre 10% e 30% enquanto para análises DEMA variam entre 16% e 52%.

Isto pode ser notado na Figura 4.16, onde se mostram as assinaturas diferenciais de potência obtidas para a subchave 10, enquanto a Figura 4.17 representa a evolução dos coeficientes de correlação com respeito ao número de traços usados para desenvolver o CPA e o CEMA. Nesta última, 200 e 50 traços são respectivamente suficientes para revelar a subchave secreta usando CPA e CEMA, mesmo se a convergência estatística não é completamente alcançada.

Em seguida realizam-se os mesmos experimentos de avaliação sobre a SBOX1 implementada com lógica DR DIMS e STTL. Porém nestes casos não é possível usar o mesmo vetor de entrada contendo todas as possíveis transições de dados, pois entre cada dado processado é obrigatório o uso do espaçador. Logo, as análises limitam-se a processar apenas HW. A Tabela 4.4 relata o percentual de hipóteses de subchaves corretas reveladas após as análises.

Tabela 4.4 Percentual de subchaves corretas obtidas com a avaliação.

Lógica analisada	**Hipóteses corretas**
Submódulo SR	70%
Submódulo DR Figura 4.7 (a)	90%
Submódulo DR Figura 4.7 (b)	3%
Submódulo STTL Figura 4.7 (c)	5%
Submódulo STTL Figura 4.7 (d)	1,5%

Com esses resultados é possível notar que STTL mostra-se mais robusta aos ataques DPA/CPA em relação às demais lógicas analisadas. Entretanto, este aumento em robustez é obtido ao custo de área adicional. Por outro lado pode ser uma vantagem, pois não é necessário o uso de roteamento específico como aplicado em [TIR06] [ORD08]. Certamente outras lógicas DPL, tal como WDDL, são mais robustas que a lógica usada como comparação. Porém, no escopo deste trabalho é inviável avaliar todas as opções de lógica disponíveis, devido ao tempo necessário para projeto e validação das mesmas, sem

mencionar o tempo para medição e execução do fluxo de análises. Logo, apenas a lógica DR foi avaliada aqui.

Realizou-se também um experimento visando avaliar a robustez de STTL a ataques EM. Durante este experimento, substituiu-se a sonda de corrente por uma sonda eletromagnética posicionada sobre o FPGA. Coletou-se traços EM relativos às versões SR, DR e STTL e executaram-se as mesmas análises sobre todas as versões. Os resultados obtidos com as análises são apresentados na Tabela 4.5.

Tabela 4.5 Percentual de subchaves corretas obtidas para os experimentos.

Lógica analisada	**Hipóteses corretas**
Submódulo SR	99%
Submódulo DR Figura 4.7 (b)	31%
Submódulo STTL Figura 4.7 (d)	1,5%

A partir destes resultados é possível concluir que a lógica em trilha dupla e STTL mostram-se mais resistentes a análises EM em relação à lógica trilha simples. Além disso, nota-se que STTL é mais resistente que lógica trilha dupla. Considera-se que o comportamento de tempo de processamento quase independente de dados de STTL explica o aumento de resistência a ataques EM. De fato, o balanço simultâneo do chaveamento de corrente e do tempo de processamento permite balancear o campo eletromagnético, o qual é proporcional à variação de corrente no tempo (di/dt), radiado em todo o circuito.

Entretanto, o balanceamento em nível de bloco não garante que todos os pontos do circuito radiam o mesmo campo eletromagnético, uma vez que o posicionamento do circuito e o roteamento das linhas de alimentação deste não possuem restrições. Isto explica a fuga de informação remanescente em DR e STTL em relação aos ataques DEMA e CEMA. Assim, devem-se empreender esforços no posicionamento do circuito. Isto implica distribuir melhor a atividade do circuito, e rotear as linhas de alimentação e terra, que são as principais fontes de radiação eletromagnética [ORD08].

4.8 AVALIAÇÃO DA ROBUSTEZ DO ALGORITMO DES STTL

Para a avaliação do algoritmo DES em lógica STTL realizou-se a coleta de 400.000 traços diferentes de radiação eletromagnética. O fluxo de análises empregado é o mesmo realizado anteriormente. Porém neste caso realizam-se ataques às oito SBOXes do algoritmo, visando revelar a chave criptográfica completa. Após as análises DEMA, é possível resumir os resultados das análises conforme mostra a Tabela 4.6.

Com aproximadamente 10.000 traços é possível revelar a chave secreta usada pelo algoritmo DES implementado em lógica regular. As análises realizadas na implementação do algoritmo com lógica STTL exigiram um número significativamente maior de traços para que os ataques obtivessem sucesso. Embora seja necessário um número maior de traços,

a lógica STTL ainda mostrou-se vulnerável a ataques DEMA. Um dos motivos que conduzem a tal vulnerabilidade é a implementação parcialmente síncrona do algoritmo DES. O controle síncrono do algoritmo revela os instantes em que ocorre o término da execução de cada rodada. Além disso, a versão STTL usa os mesmo registradores para armazenar temporariamente os dados parcialmente cifrados, fato que contribui para a fuga de informações. Apenas uma subchave não foi revelada. Uma justificativa para isso pode ser a posição da sonda, que não favoreceu a captação de ondas eletromagnéticas provenientes desta parte do circuito durante a fase de medições, e por conseqüência não propiciou bons resultados para as análises sobre a SBOX1.

Tabela 4.6 Número de traços para revelar as subchaves do algoritmo DES.

	DES Regular (número de traços)	**DES STTL (número de traços)**
Subchave 1	4320	não revelada
Subchave 2	2360	68.600
Subchave 3	7480	229.440
Subchave 4	10380	72.100
Subchave 5	10720	107.000
Subchave 6	4640	96.000
Subchave 7	3280	238.780
Subchave 8	2920	190.260

4.9 CONCLUSÕES

Este Capítulo introduziu uma avaliação experimental da robustez de STTL a ataques DPA e DEMA. Esta avaliação realizou-se em FPGAs, usando hard macros e síntese de hardware com roteamento e posicionamento padrão, sem restrições de área. Os resultados permitem as seguintes conclusões:

(a) STTL é definitivamente mais robusta contra DPA/CPA em relação à lógica em trilha única e ligeiramente mais robusta que lógica em trilha dupla;

(b) O projeto de DR e STTL em FPGA ocupa praticamente a mesma área;

(c) STTL, enquanto mais resistente que trilha única e DR não é totalmente robusto aos ataques por consumo de potência e por radiações eletromagnéticas.

Os últimos resultados sugerem que mais esforços devem ser feitos para balancear espacialmente, em amplitude e no tempo, o fluxo de chaveamento de corrente dentro do chip.

5. ARQUITETURAS GALS PARA CONTRAMEDIR ATAQUES DPA E DEMA

Este Capítulo apresenta a principal contribuição deste trabalho, a proposta de arquiteturas pipeline GALS como contramedida a análises por consumo de potência e radiação eletromagnética. As arquiteturas propostas usam também circuitos não-síncronos em sua infraestrutura. Porém, diferentemente da biblioteca STTL, estas empregam inserção de aleatoriedade no processamento como método de prevenção a ataques. Como já discutido no Capítulo 3, apenas uma proposta de contramedida usando metodologia GALS de projeto é encontrada na literatura. Já arquiteturas pipeline são encontradas na literatura com diversos propósitos, dentre estes a prevenção a ataques DPA tal como proposto em [STA04]. Este Capítulo propõe pela primeira vez a combinação de arquiteturas pipeline implementadas com metodologia GALS de projeto usando relógios com freqüências de operação aleatoriamente selecionadas como forma de contramedir a ação de análises por consumo de potência e radiação eletromagnética. Uma vantagem adicional significativa da arquitetura proposta é o aumento da vazão de dados, proporcionado pelo estilo pipeline de implementação.

5.1 ALGORITMO DES: CARACTERÍSTICAS

Este trabalho emprega uma técnica de projeto de hardware muito comum em processadores, o pipeline de execução de instruções [HEN05]. As instruções são executadas por processadores em várias etapas. O processador apresenta uma infraestrutura com hardware dedicado a cada uma das etapas, de modo que seja possível haver diferentes instruções sendo executadas em etapas distintas simultaneamente, a fim de aumentar o desempenho do processamento global do sistema. Este mesmo princípio de execução pode ser aplicado a sistemas criptográficos em hardware. Neste caso, o algoritmo criptográfico deve permitir implementações em modo pipeline, fato que pode restringir a abordagem proposta a um grupo de algoritmos. Dentre os algoritmos passíveis de se beneficiar de implementações pipeline, possível citar DES, AES, SHA-1 e MD-5. Além disso, DES e AES são os algoritmos mais usados em *smart cards* e aparecem comumente na literatura como alvo de avaliações de robustez a ataques por consumo de potência e radiação eletromagnética.

A escolha do algoritmo DES justifica-se por ter este sido alvo de estudos de Kocher em [KOC96] e [KOC99] e por utilizar menos área do que seu sucessor AES, quando implementado em hardware. A Figura 5.1 mostra a estrutura geral do DES. Este algoritmo opera sobre blocos de entrada de 64 bits. Cada entrada passa através de 16 rodadas de modificação. Cada rodada usa uma subchave distinta de 48 bits, duas entradas de 32 bits em operações que incluem permutações (P), expansões de dados (E), deslocamentos, operações lógicas ou exclusivo (XORs ‘⊕’) e caixas de substituição (SBOXes). Depois de executar 16 rodadas, o algoritmo produz um resultado que sofre uma permutação inversa e se torna o dado cifrado de 64 bits.

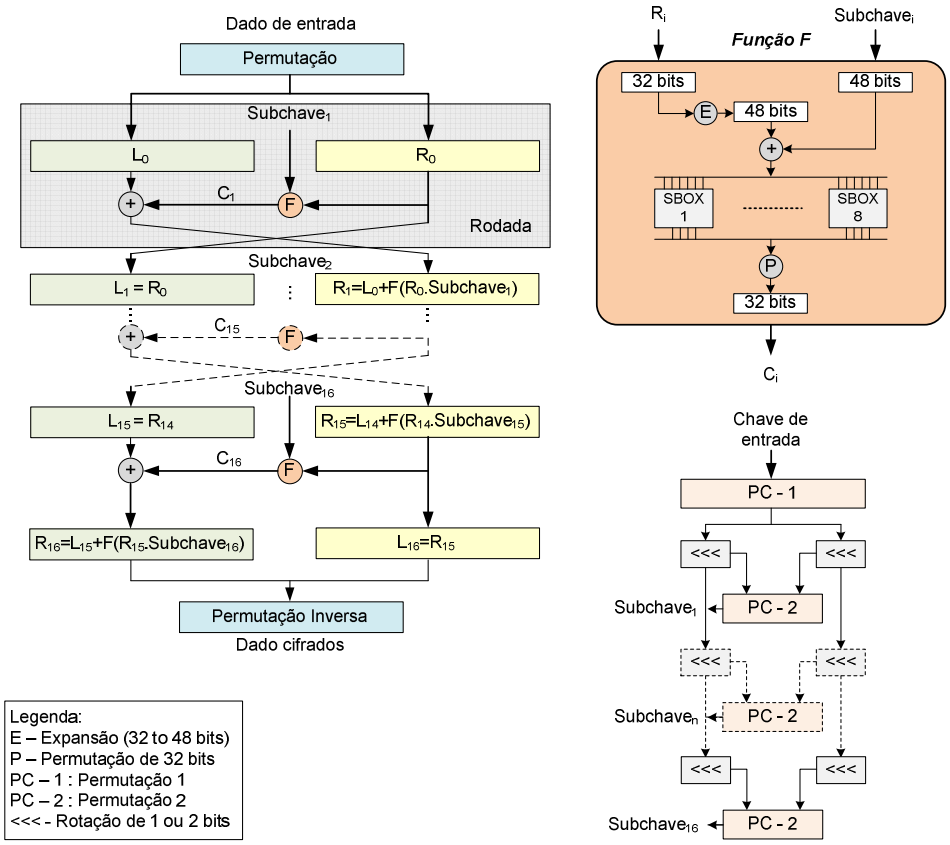


Figura 5.1 Estrutura geral do algoritmo DES.

O algoritmo DES esquematizado na Figura 5.1 pode ser implementado em hardware ou software. Uma implementação convencional do DES cria um único bloco de encriptação do algoritmo (seja este em software ou hardware) e executa 16 iterações sobre este bloco com parametrização adequada. Entretanto, o uso de múltiplos blocos de encriptação permite implementar o algoritmo em modo pipeline. Assim, é possível ter arquiteturas pipeline a partir de 2 até 16 estágios para executar o algoritmo completo. Uma vantagem deste tipo de implementação é auferir um aumento significativo da vazão de dados cifrados. Por outro lado, como desvantagem, esta abordagem apresenta claramente um custo adicional em área.

5.2 INFRAESTRUTURA GALS

Este trabalho propõe a implementação de um algoritmo DES pipeline GALS como mostrado na Figura 5.2. A implementação é baseada na replicação do bloco de encriptação do algoritmo. Cada estágio do pipeline é envolvido por uma interface assíncrona e gerenciado por um módulo de controle. Este controle é definido como uma máquina de estados finita que gerencia a comunicação ponto a ponto com seus estágios vizinhos através de um protocolo de comunicação *handshake* (Req - Ack) de 2 fases.

Um subsistema externo ao pipeline é responsável por gerar um sinal de relógio com freqüência pseudo-aleatória, o qual supre os estágios envoltos pela interface assíncrona, também chamados de ilhas síncronas por Chapiro [CHA84]. Este subsistema gera uma

nova freqüência de relógio sempre que um estágio termina o processamento de um dado. Neste contexto, a estrutura do subsistema de relógio pode ser construída de diversas maneiras. Osciladores externos a cristal são uma opção e osciladores em anel internos ao circuito são outra possibilidade.

A Figura 5.2 mostra ainda o esquema da interface assíncrona. O uso de sincronizadores simples do tipo 2-flip-flops permite a comunicação entre módulos com diferentes domínios de relógio. A principal suposição de tais sincronizadores é que o tempo reservado para a resolução de metaestabilidade permite um tempo médio entre falhas (do inglês, *Mean Time Between Failures* - MTBF) satisfatório [RAB03]. Outras propostas de circuitos para sincronização em sistemas que envolvem diferentes domínios de freqüência estão disponíveis na literatura, tipicamente mais eficientes e mais complexos de desenvolver que o esquema básico 2-flip-flops. O leitor interessado pode para tanto consultar, por exemplo, Dobkin e Ginosar [DOB09].

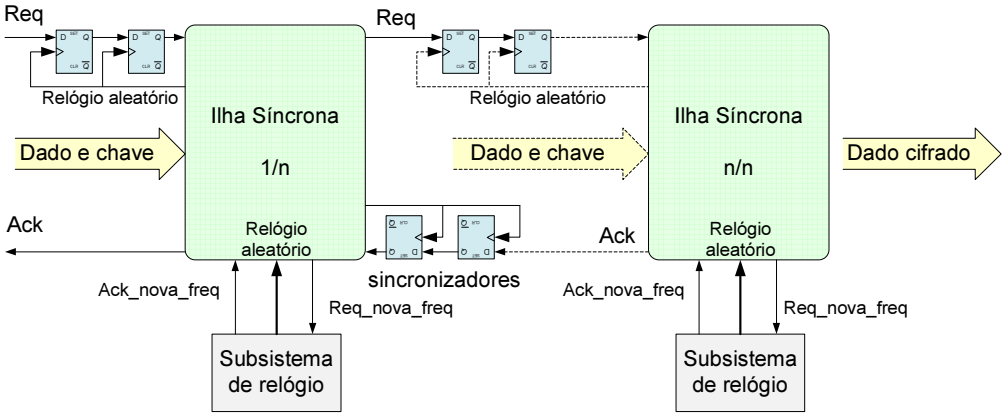


Figura 5.2 Infraestrutura do algoritmo DES implementado em modo pipeline usando um método GALS de projeto. A proposta inclui interfaces assíncronas do tipo 2-flip-flops usando protocolo de comunicação em 2 fases e geradores de relógio independentes que se operam sob comando de cada estágio pipeline.

Os sincronizadores introduzem uma penalidade de tempo que aumenta a latência do sistema. Com a ativação do sinal de requisição de processamento (Req), dois ciclos de relógio do estágio receptor podem ser necessários para que este sinal se propague até o próximo estágio e dê início ao processamento dos dados naquele. Um sinal de reconhecimento (Ack) é enviado ao requisitante a fim de informá-lo que os dados foram recebidos com sucesso.

Nesta arquitetura, cada subsistema de relógio modifica sua freqüência de saída a cada nova requisição de processamento. Isto pressupõe que o tempo de processamento em cada estágio é alterado para cada novo dado de entrada. A latência inserida pelos sincronizadores pode ser vista como uma desvantagem em termos de desempenho. Porém este atraso é variável devido aos sinais de relógio com freqüências distintas. Estes fatores aumentam a aleatoriedade do processamento, dificultando a ação das análises de consumo de potência e de radiações eletromagnéticas.

O subsistema de relógio desempenha um papel importante na abordagem proposta, pois uma nova freqüência deve ser gerada a cada novo dado processado. Isto tende a aumentar a segurança, pois oculta a fuga de informações do criptosistema. Trabalhos anteriores propõem alcançar este mesmo objetivo [GUR06] [LU08] [STA04] [ZAF08]. Porém, até onde o autor pode avaliar, o presente trabalho é o primeiro a combinar uma implementação GALS baseada em freqüências de relógio aleatórias e uma estrutura pipeline. O modo como o subsistema de relógio é implementado não é um foco fundamental desta arquitetura. Contudo, uma proposta de implementação é discutida na Seção 5.3.1.

5.3 PROVA DE CONCEITO

O método proposto requer a replicação de hardware para contribuir para a neutralização de análises DPA e DEMA. De fato, o processamento paralelo das rodadas produz um ambiente ruidoso que dificulta a tarefa de SCAs. Por outro lado, a replicação de hardware em um pipeline naturalmente aumenta a vazão de dados do criptosistema. Esta Seção disponibiliza uma comparação do método proposto com alguns trabalhos anteriores em termos de área, vazão e robustez. Os resultados apresentados sobre robustez mostram evidências de que este método melhora a segurança do criptosistema.

5.3.1 IMPLEMENTAÇÃO EM FPGA

A Figura 5.3 apresenta a arquitetura projetada para conduzir os experimentos de avaliação de robustez e comparações em termos de área e vazão. Esta arquitetura contém uma ilha síncrona responsável pela comunicação de dados com um hospedeiro através de uma porta serial. Este mesmo módulo também tem como função manter o pipeline GALS sempre preenchido ao máximo com dados a criptografar, independente da taxa de comunicação com o hospedeiro. Esta ilha opera com um sinal de relógio único e convencional, com freqüência de 50 MHz, conforme disponibilizado em diversas plataformas de prototipação.

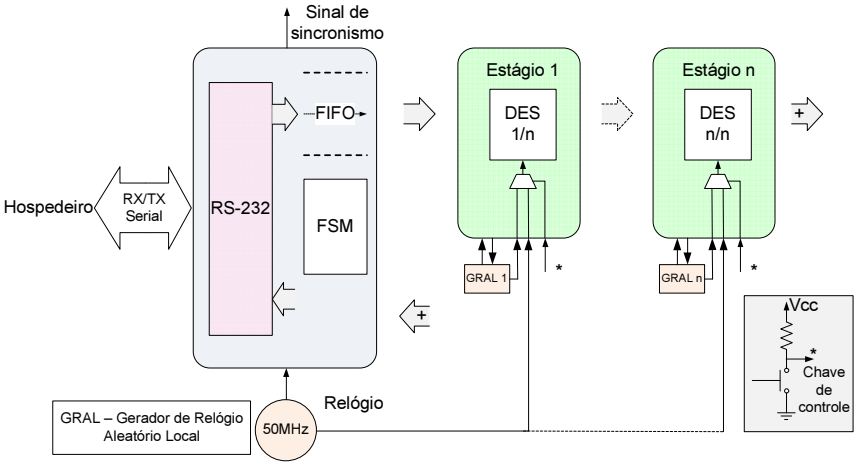


Figura 5.3 Visão geral da arquitetura usada para avaliar a robustez da arquitetura pipeline GALS.

Como mencionado antes, a fim de manter o pipeline da arquitetura repleto de dados, a ilha principal contém um dispositivo de armazenamento FIFO (do inglês, *First In First Out*). Este dispositivo permite que a arquitetura proposta tenha sua robustez a ataques DPA/DEMA avaliada de duas maneiras. A primeira destas é sem uso da FIFO. Aqui, cada dado de entrada é processado consecutivamente em todos os estágios do pipeline, onde cada estágio opera com um sinal de relógio diferente escolhido aleatoriamente pelo seu respectivo GRAL como mostrado na Figura 5.3. Neste caso, apenas um dado de entrada permanece no pipeline até que este seja completamente encriptado e transmitido ao hospedeiro para que o próximo dado seja processado. Isto permite avaliar o efeito produzido pelos relógios com diferentes freqüências escolhidas aleatoriamente de forma a aumentar a robustez da arquitetura. Na segunda maneira, com uso da FIFO, além das execuções com diferentes freqüências avalia-se a arquitetura processando diferentes dados simultaneamente. A FIFO permite que dados sejam enviados ao pipeline tão logo quanto o primeiro estágio esteja disponível para atender a uma nova requisição de processamento. O processamento paralelo dos estágios aumenta o ruído no consumo de potência e conseqüentemente na radiação eletromagnética produzidos pela arquitetura.

A Figura 5.3 também mostra o sinal (saída localizada na parte superior do controlador da comunicação serial) usado para sincronizar o osciloscópio com o processamento dos dados.

A arquitetura genérica apresentada na Figura 5.3 foi prototipada em 4 versões diferentes, com 2, 4, 8 e 16 estágios, respectivamente denominadas PIPE-2, PIPE-4, PIPE-8 e PIPE-16 GALS. Na implementação PIPE-2 GALS o bloco de encriptação do algoritmo é replicado duas vezes e cada um destes executa 8 rodadas do algoritmo. Raciocínio similar define as demais implementações.

Cada estágio de cada uma das implementações emprega um subsistema gerador de relógio local que produz um sinal de relógio com uma freqüência escolhida aleatoriamente entre n possíveis a cada dado processado. O gerador de relógio pode ser composto de *n* osciladores em anel compostos por um número ímpar definido de elementos de atraso operando cada um como um inversor. Um multiplexador livre de transitórios [MAH09] é usado para chavear entre os sinais de relógios. Esta restrição se faz necessária, pois a ocorrência de um transitório no sinal de relógio pode conduzir o sistema a estados incorretos. A Figura 5.4 apresenta o circuito do multiplexador empregado, para n=4.

Um módulo com a função de gerar números aleatórios usados para a escolha de um sinal de relógio deve compor o subsistema de relógio. Neste caso emprega-se um registrador LFSR (do inglês, *Linear Feedback Shift Register* – LFSR). Embora seu comportamento determinístico permita apenas a geração de dados pseudo-aleatórios [HEA09], este módulo serve à validação do método proposto. A Figura 5.5 apresenta uma simulação da operação de chaveamento livre de transitórios deste subsistema.

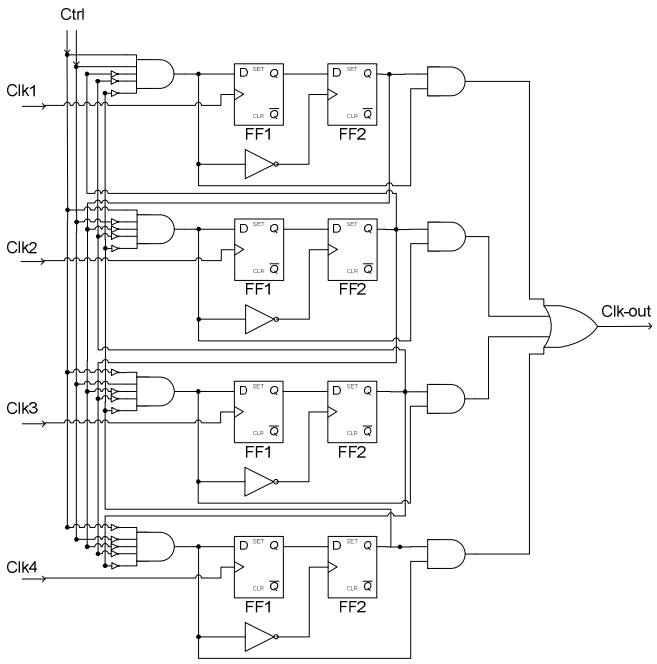


Figura 5.4 Circuito de um multiplexador 4:1 livre de transitórios.

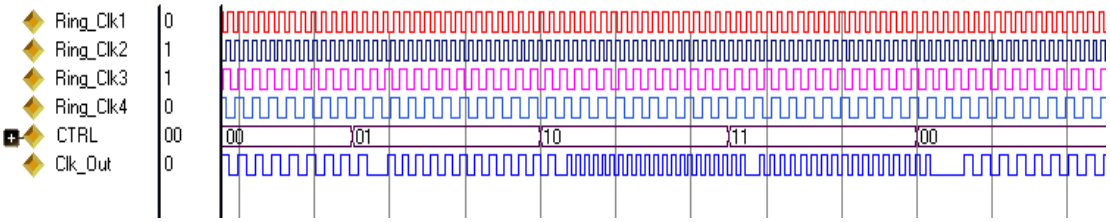


Figura 5.5 Simulação mostrando um chaveamento de sinais de relógio livre de transitórios.

5.3.2 AVALIAÇÃO DE ÁREA

Observando a Tabela 5.1, nota-se que o método proposto exige menos área que implementações assíncronas na maioria dos casos, mas sofre de um custo adicional em área de até 25 vezes no pior caso quando comparada a implementação regular do DES. Esta avaliação baseou-se no dispositivo FPGA VirtexII-XC2V4000 da Xilinx [XIL07]. Assim, a abordagem GALS pipeline permite negociar área a fim de obter robustez. Além disso, é virtualmente impossível implementar STTL em FPGAs usando um fluxo de projeto síncrono padrão. Os resultados na Tabela 5.1 originados do uso de um fluxo específico baseado no leiaute de hard macros para FPGAs Xilinx, conforme detalhado no Capítulo 4.

WDDL é outra lógica assíncrona na qual Guilley et al. [GUI08] relatam alto custo em área. Estes Autores mencionam uma otimização de 23% em área para uma implementação do triplo DES, partindo de uma implementação inicial de 3.019 slices para 2.321 slices.

Tabela 5.1 Comparação em termos de área entre as diversas implementações do DES.

Arquitetura	Slices	Flip-flops	Aumento relativo ao DES regular
Regular DES	267	125	1
GALS PIPE-2	935	1037	3.5
GALS PIPE-4	1830	2033	6.85
GALS PIPE-8	3605	4006	13.5
GALS PIPE-16	6614	7267	24.77
STTL	5130	Não disponível	19.21

5.3.3 AVALIAÇÃO DE LATÊNCIA E VAZÃO DE DADOS

A Tabela 5.2 apresenta os resultados de latência e vazão de dados para as seguintes implementações do algoritmo DES: (i) regular, (ii) GALS PIPE, (iii) STTL e a comparação destas arquiteturas com a abordagem proposta por Gürkaynak et al. disponível na literatura.

Tabela 5.2 Comparações em termos de latência e vazão de dados das implementações do algoritmo DES. As medidas de vazão são expressas em bits por segundo (Mbps).

		Vazão			
Arquitetura	**Latência (ciclos)**	**f=7.2Mhz (Mbps)**	**f=21Mhz (Mbps)**	**f=100Mhz (Mbps)**	**f=*Mhz (Mbps)**
Regular DES	17	201,9	61,1	290,9	-
GALS PIPE-2	24	21,8	64	304,7	-
GALS PIPE-4	40	27	79	376,4	-
GALS PIPE-8	61	32,9	96	457	-
GALS PIPE-16	109	35,4	103,4	492,3	-
STTL	NA	NA	NA	NA	14,3
Gürkaynak et al. [GUR06]	ND	ND	ND	ND	256*

NA – Não aplicável

ND – Não disponível

* Autores não apresentam detalhes sobre as condições de medição, mas mencionam o uso de três domínios de relógio, um de 190MHz e dois outros de 250MHz.

Nestes experimentos, os osciladores em anel dos subsistemas de relógios são implementados a partir de cadeias de LUTs. Embora limitado, este é um modo simples

para implementar uma prova de conceito da operação das arquiteturas propostas. A freqüência mínima é 7,2 MHz e a máxima 21 MHz. Apesar das baixas freqüências usadas nos experimentos, a freqüência máxima estimada pela ferramenta de síntese para as arquiteturas GALS PIPE é em torno de 100 MHz. Observa-se que na terceira coluna avalia-se o pior caso de vazão das arquiteturas e na quinta coluna é apresentado o melhor caso de vazão atingida. É importante notar que as freqüências operacionais dos subsistemas de relógio são escolhidas de tal modo que os estágios operem com freqüências distintas. Por isso, para estimar os limites de vazão, comparações são baseadas na execução de todos os estágios das arquiteturas na mesma freqüência, conforme a Tabela 5.2.

Observa-se que a replicação do hardware não apresenta efeito significativo na freqüência máxima de operação das arquiteturas pipeline. Em relação à implementação regular do DES, as versões GALS PIPE aumentam a vazão proporcionalmente ao número de estágios, como esperado. Já em relação à STTL, mostram uma vazão muito superior. O problema com STTL é que a propagação do sinal de validade de informação apresenta alto custo em latência devido às cadeias de atrasos. A magnitude da vazão das arquiteturas GALS PIPE são compatíveis com abordagens similares disponíveis na literatura, tal com Gürkaynak et al. [GUR06].

5.3.4 AVALIAÇÃO DA ALEATORIEDADE

O método proposto aqui se baseia na inserção de aleatoriedade no processamento criptográfico, de forma a ocultar a fuga de informações. Como revisado na literatura, o simples fato de inserir atrasos no processamento não garante completamente o sucesso da contramedida empregada. Portanto, a forma como esta contramedida é inserida é importante para garantir a eficiência do método.

O método insere aleatoriedade em partes do algoritmo implementado em pipeline. Para tanto, cada estágio do pipeline possui seu domínio de freqüência e controle da inserção de aleatoriedade. Deste modo, quanto maior for o número de estágios do pipeline, maior será a aleatoriedade inserida no sistema. Os ataques DPA relacionam a amplitude da potência consumida pelo circuito e o período de tempo na qual os dados são computados. Portanto, a aleatoriedade inserida para ocultar a fuga de informações pode ser uma variação na amplitude da potência bem como deslocamentos temporais de cálculos executados pelo circuito. O método proposto insere ruídos através do processamento paralelo dos estágios do pipeline e através do chaveamento de sinais de relógio com diferentes freqüências. Para estimar de maneira simplificada a aleatoriedade inserida no domínio do tempo, define-se aqui que a aleatoriedade inserida é equivalente a variação do instante de tempo em que o sinal de sincronismo fica ativo para a medição de consumo e radiação eletromagnética do circuito.

No modo de operação sem FIFO, onde se analisa apenas a influência do sinal de relógio, o número de ciclos de relógio para se processar um dado é constante. Por conseqüência, o número de ciclos de relógio no qual o sinal de sincronismo fica ativo é constante também. Logo, o período do sinal de sincronismo varia apenas com os períodos

dos sinais de relógio dos estágios do pipeline. Com o objetivo de avaliar de forma aproximada a aleatoriedade inserida nas arquiteturas propostas é realizada uma medição e avaliação estatística sobre o sinal de sincronismo. A Tabela 5.3 apresenta os resultados.

Tabela 5.3 Avaliação da aleatoriedade das arquiteturas propostas.

Parâmetros	**PIPE-2**	**PIPE-4**	**PIPE-8**
Média (μ)	935,12 ns	990,97 ns	1,8207 μs
Mediana (Md)	919,86 ns	959,73 ns	1,8595 μs
Desvio Padrão (σ)	184,41 ns	134,80 ns	206,62 ns
Moda	779,7 ns	939,80 μs	1,8595 μs
μ ± 1σ	62,1%	70%	65,8%
μ ± 2σ	100%	95,7%	93,4%
μ ± 3σ	100%	100%	100%
Período Mínimo	619,5 ns	759,6 ns	1,2599 μs
Período Máximo	1,32 μs	1,32 μs	2,1393 μs
Diferença Período (Δ)	700,5 ns	560,4 ns	879,4 ns
Número de amostras	50000	50000	50000

Inicialmente, observa-se os valores das médias aritméticas de tempo para encriptar um dado. Nota-se que a média μ aumenta proporcionalmente com o número de estágios usados na implementação da arquitetura. Este aumento é decorrente do número de sincronizadores usados e do protocolo de comunicação usado entre cada estágio. Isto é confirmado pelos períodos mínimo e máximo para cada uma das arquiteturas e também pelos valores de latência medidos e apresentados na Tabela 5.2. Os valores referentes ao parâmetro Moda indicam os tempos que mais ocorreram durante as avaliações. Estes valores são resultados dos geradores LFSRs usados para inserir aleatoriedade no processo de escolha de sinais de relógios. As probabilidades das escolhas de cada um dos 4 sinais disponíveis não são igualmente distribuídas. Logo, existem sinais de relógio em cada um dos estágios que possuem maiores ocorrências em relação aos demais. Isto é demonstrado pelo parâmetro Moda em cada uma das arquiteturas. O desvio padrão σ indica o desvio médio do período de tempo em relação à média μ. Quanto maior for o desvio padrão nas arquiteturas propostas, melhor será a aleatoriedade inserida no processamento, conseqüentemente maior será a complexidade das tarefas de ataques para correlacionar o consumo com os dados processados. Esta Tabela revela que o desvio padrão na arquitetura PIPE-4 é o menor em relação aos demais. Isto revela que as freqüências usadas nos subsistemas ou as funções pseudo-aleatórias usadas devem ser otimizadas de modo a obter-se um maior desvio padrão.

5.4 SISTEMA DE MEDIÇÃO

O sistema utilizado para medir os traços de consumo de potência e de radiações eletromagnéticas é fundamentalmente o mesmo usado na avaliação de STTL. Porém, neste sistema é necessário enviar dados de 64 bits ao invés de dados de 8 bits, e verificar se o resultado cifrado está correto antes de armazenar o traço medido.

Para avaliar as arquiteturas propostas utilizaram-se os seguintes elementos: (1) uma plataforma Digilent Spartan-3 Board com um dispositivo Xilinx Spartan-3 XC3S1000, (2) uma sonda eletromagnética de 500 μm, (3) um amplificador com baixo nível de ruído (1 GHz de largura de banda e 63 dB) para amplificar o sinal obtido pela sonda e aumentar a precisão da medição, (4) uma mesa cartesiana XYZ para posicionamento automático da sonda eletromagnética, (5) um osciloscópio Agilent Infinium DS80000B (4 GHz – 40 GSa/s), (6) uma gaiola de Faraday para eliminar interferências eletromagnéticas e (7) um PC (Xeon 3.8 GHz quad-core; 12GB de memória RAM, Linux RedHat 5) com suporte a scripts MATLAB para controle do sistema de medição. A Figura 5.6 mostra imagens do sistema de medição empregado.

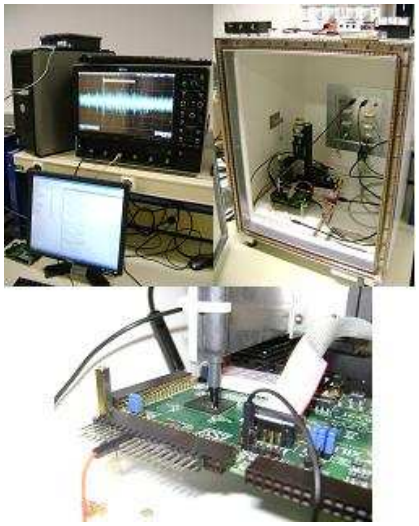


Figura 5.6 Sistema de medição usado para a avaliação das arquiteturas propostas.

Esta nova versão do sistema de medição adiciona a mesa cartesiana e a gaiola de Faraday ao sistema descrito no Capítulo 4. Além disso, o sistema de controle executado no hospedeiro também foi reescrito para uso no ambiente MATLAB. Estes scripts foram todos desenvolvidos conforme descrevem Lomné et al. [LOM09] e Dehbaoui et al. [DEH09] e usados no DPA Contest 2009 [DPA09].

O uso da sonda eletromagnética permite que sejam explorados vários pontos de medição sobre a superfície do chip a atacar. Alguns fatores, tais como restrições de planta baixa no projeto da arquitetura e disposição dos pinos de E/S do chip influenciam na escolha dos pontos de medição. Os pinos de entrada do sinal de relógio e saída do sinal de sincronização com o osciloscópio produzem ruídos que interferem significativamente na medição de traços. Havendo a necessidade de automatização da procura pelos melhores pontos de medição, desenvolveu-se um script para controlar a mesa cartesiana e realizar

uma varredura sobre a superfície do chip, visando selecionar os melhores pontos de medição isto é, os pontos de medição onde se descobre evidências das rodadas executadas. Os melhores pontos são escolhidos (geralmente entre 1 e 3 pontos). A seguir, suas coordenadas cartesianas são registradas para que se realize o processo de aquisição de traços.

5.5 AVALIAÇÃO DA ROBUSTEZ A ATAQUES DPA E DEMA

Na avaliação das arquiteturas propostas apenas um ponto de medição foi usado para a coleta de traços referentes às radiações eletromagnéticas geradas pelo circuito. A decisão sobre por este ponto é justificada pelo uso de uma simples técnica para posicionamento da sonda eletromagnética sobre o FPGA: (i) o FPGA é dividido em 64 pequenos quadrados e 5 traços de radiação eletromagnética são adquiridos em cada ponto; (ii) uma inspeção visual de todos os 320 traços define o melhor ponto como aquele que melhor lembrar a forma de onda da Figura 2.8. Uma alternativa a este abordagem para definir pontos atacados seria empregar uma técnica tal como *Weighted Global Magnitude Squared Incoherence* (WGMSI) sugerida por Dehbaoui et al. [DEH10].

Conforme já discutido anteriormente, menos de 200 traços são suficientes para se revelar uma subchave criptográfica de um algoritmo sem métodos de prevenção a ataques SCAs [DPA09]. Nos experimentos descritos aqui, processa-se uma seqüência de 100 mil dados distintos aleatoriamente gerados com uso de uma única chave criptográfica.

5.5.1 ETAPA DE MEDIÇÃO E COLETA DE TRAÇOS

A Figura 5.7 e a Figura 5.8 mostram, respectivamente, os traços de radiação eletromagnética e de consumo de potência obtidos pelo sistema de medição para arquiteturas DES PIPE-2. Nas versões com FIFO (*ii* e *iv*), são enviados 3 dados distintos para serem processados, sendo eles *dado 1*, *dado 2* e *dado 3*. Porém, os ataques por consumo de potência e/ou radiação eletromagnética são realizados apenas sobre o *dado 2*. Os *dados 1 e 3* são gerados aleatoriamente pelo hospedeiro e armazenados na FIFO a fim de manter ambos os estágios da arquitetura PIPE-2 processando durante o período em que se realiza a medição e coleta do traço referente a *dado 2*.

Durante a realização das medições de consumo de potência, nota-se claramente a interferência provocada por uma fonte de ruído não identificada externa ao circuito. Mesmo com o uso da gaiola de Faraday e uso de uma fonte de alimentação estabilizada não foi possível eliminá-la. Na Figura 5.8 (iii) e (iv) é possível identificar claramente esta interferência. Como estas arquiteturas utilizam sinais de relógios com diferentes freqüências não é possível realizar a média dos traços adquiridos para reduzir os efeitos do ruído sobre os traços. Neste caso, o tempo de processamento de cada dado varia de acordo com as freqüências selecionadas e a realização da média ocultaria a fuga de informações contidas nos traços.

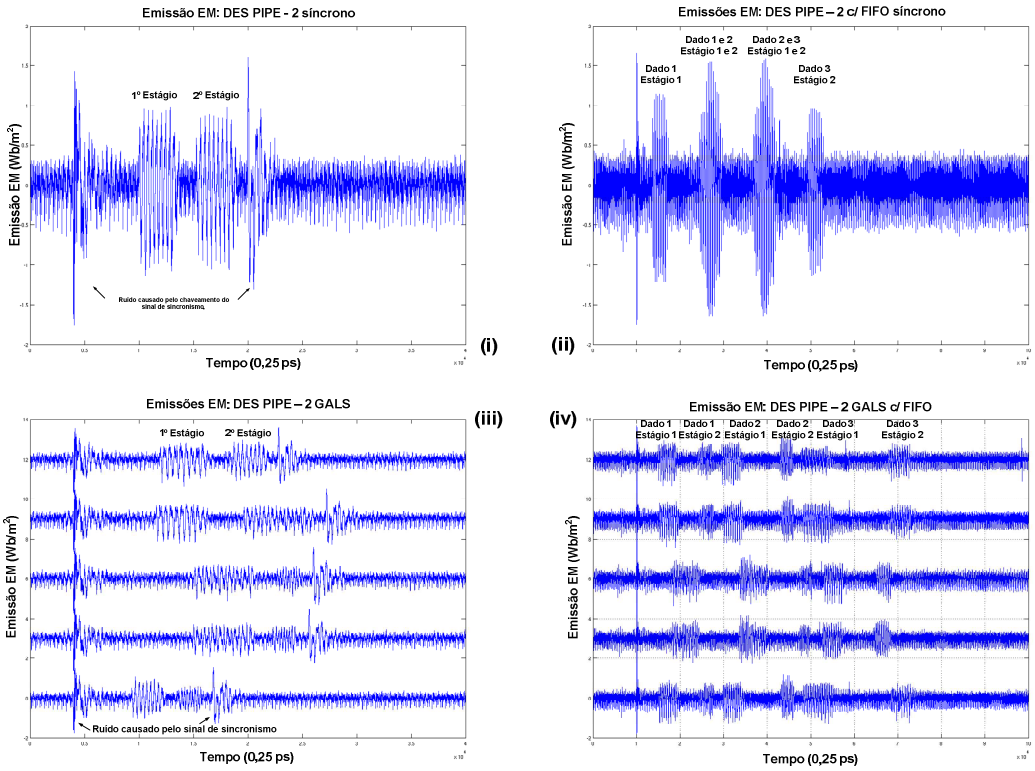


Figura 5.7 Medidas de radiação eletromagnética emitida pelas arquiteturas DES PIPE-2: (i) síncrona, (ii) síncrona com FIFO, (iii) GALS e (iv) GALS com FIFO.

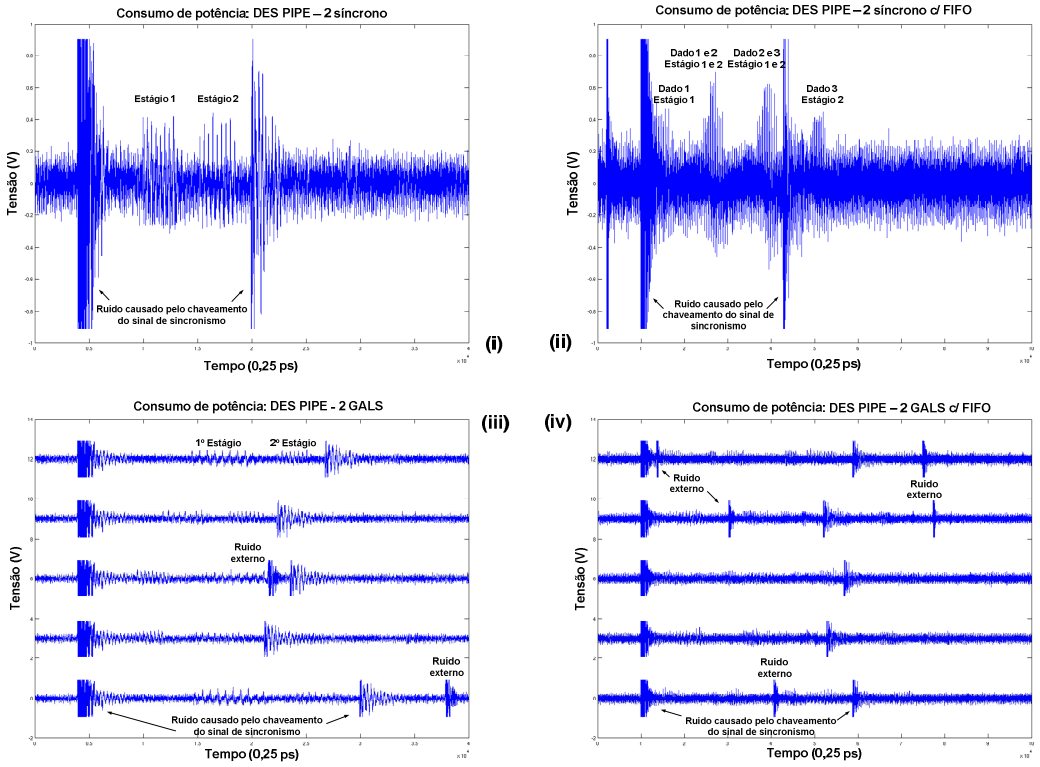


Figura 5.8 Consumo de potência das arquiteturas DES PIPE-2: (i) síncrona, (ii) síncrona com FIFO, (iii) GALS e (iv) GALS com FIFO.

A Figura 5.9 apresenta as medidas de radiação eletromagnética obtidas para as arquiteturas DES PIPE-4. Em (i) é possível identificar as rodadas sendo processadas nos 4 estágios do pipeline. Cada estágio executa 4 rodadas do algoritmo. Estes grupos de 4 rodadas são identificados na Figura 5.9 (i). Porém nas versões (ii) e (iii) as rodadas não são claramente visíveis. Em (ii) são enviados 5 dados distintos para manter o pipeline completo no instante da medição. Neste caso, as medições são realizadas apenas sobre o *dado 3*. Estas arquiteturas foram prototipadas sobre um FPGA Spartan-3 XC3S1000. Neste dispositivo as medições de consumo de potência obtiveram traços com uma qualidade ruim do sinal, não sendo possível identificar as rodadas do mesmo com a prototipação da arquitetura em modo síncrono. A hipótese provável para isso não foi identificada, pois a plataforma é a mesma em ambos os casos, apenas o dispositivo é alterado. Por este motivo, apenas medições de traços de radiação eletromagnética foram realizadas.

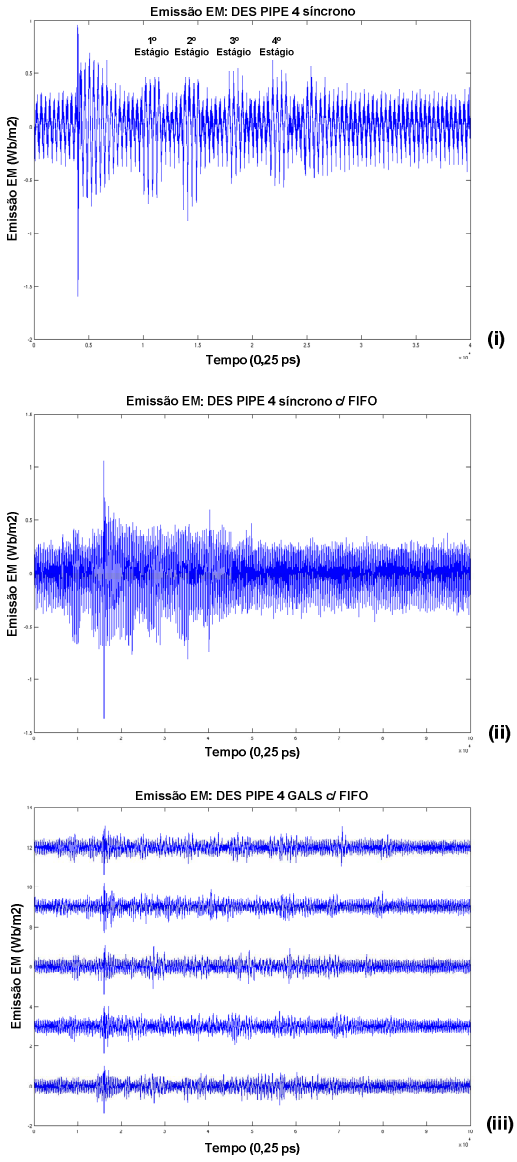


Figura 5.9 Radiações eletromagnéticas das arquiteturas DES PIPE-4: (i) síncrona, (ii) síncrona com FIFO e (iii) GALS com FIFO.

A Figura 5.10 apresenta traços de radiação eletromagnética das arquiteturas PIPE-8. A Figura 5.10 *(i)* contém o traço para uma arquitetura DES PIPE-8 síncrona. Em *(ii)* mostra-se um traço da arquitetura DES PIPE-8 com FIFO e em *(iii)* um traço referente à radiação eletromagnética da arquitetura GALS DES PIPE-8. Como se pode perceber em *(ii)* e *(iii)*, não é possível identificar as rodadas do algoritmo DES em execução.

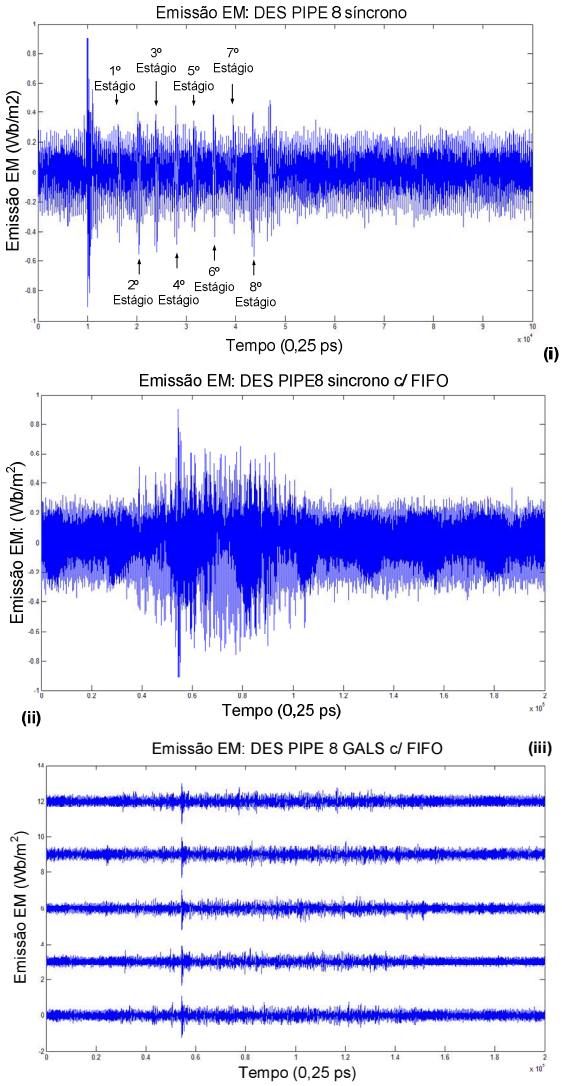


Figura 5.10 Medidas de radiação eletromagnética das arquiteturas DES PIPE-8: (i) síncrona, (ii) síncrona com FIFO e (iii) GALS com FIFO.

Os demais traços de consumo de potência e radiação eletromagnética medidos estão disponíveis no Apêndice A deste volume.

5.5.2 RESULTADO DAS ANÁLISES

Para avaliar a robustez das arquiteturas propostas são utilizados quatro tipos de ataques: DPA, DEMA, CPA e CEMA. Inicialmente, 100 mil traços foram coletados para executar os ataques. A Tabela 5.4 apresenta os resultados relativos às análises DEMA e DPA. Estes resultados representam o número mínimo de traços necessários para

descobrir as subchaves referentes às SBOXes 1 a 8, as quais foram alvos dos ataques. A partir destes resultados nota-se que nenhuma das versões GALS propostas permitiram revelar suas subchaves, como esperado. Todas as versões síncronas tiveram suas subchaves descobertas provando usa vulnerabilidade. O uso exclusivo do processamento paralelo das rodadas como contramedida não apresenta melhoras significativas na resistência a estes ataques e, em alguns casos, os ataques foram até mais eficazes em versões com processamento paralelo, tal como PIPE-2 e PIPE-8.

As análises DEMA aplicadas sobre a arquitetura PIPE-2 apresentaram um comportamento atípico para a arquitetura síncrona. Um provável motivo para este comportamento é posicionamento da sonda eletromagnética durante o processo de coleta de traços. A posição escolhida provavelmente capta radiações eletromagnéticas com baixa intensidade produzidas pelo circuito SBOX8, o que reduz a quantidade de informação e dificulta as análises. Uma solução alternativa ao problema do posicionamento da sonda é o uso de WGMSI [DEH10]. Já os resultados obtidos pelas análises DPA demonstram um comportamento esperado, ou seja, um menor número de traços para se obter a chave criptográfica em relação à versão síncrona com processamento paralelo.

Tabela 5.4 Número de traços necessários para revelar a chave criptográfica utilizando análises DEMA e DPA.

	PIPE-2		**PIPE-4**		**PIPE-8**	
Arquiteturas	**DEMA (#)**	**DPA (#)**	**DEMA (#)**	**DPA (#)**	**DEMA (#)**	**DPA (#)**
Síncrona	49336	21614	16365	x	88540	ne
Síncrona com FIFO	10269	53485	29416	ne	37686	ne
GALS	-	-	ne	ne	ne	ne
GALS com FIFO	-	-	-	ne	-	ne

(-) : não descoberta
(ne): não executada
(#): número de traços para encontrar a chave completa
(x): apenas a subchave da SBOX8 não foi revelada.

Observando estes resultados e comparando-os com os obtidos na avaliação do DES STTL na Tabela 4.6 nota-se que com 100 mil traços 3 subchaves são reveladas na arquitetura DES STTL e evidências estatísticas demostram que outras subchaves ficam próximas de serem reveladas. Deste modo um aumento para 400 mil traços permitiu revelar as demais subchaves, à exceção da subchave 1. Entretanto, para as arquiteturas GALS PIPE os resultados dos ataques não apresentam indicação, mesmo com 100 mil traços, de convergir para a subchave correta.

Para justificar estes resultados, duas razões explicam o fato de não ocorrer uma convergência durante as análises executadas sobre as arquiteturas GALS PIPE. A primeira delas, o aumento da aleatoriedade no processamento, sendo efetivo em

aleatorizar o momento da encriptação alvo e o tempo de duração da mesma. A outra justificativa é dada a execução superposta dos estágios que de certa forma contribuem para dificultar as análises.

A execução destas análises sobre 100 mil traços exigiu até dois dias para serem concluídas. Na maioria dos casos, revelar a subchave exigiu apenas uma pequena fração do número total de traços. Da mesma forma, a execução das análises sobre a arquitetura STTL usando 400 mil traços exigiu um tempo de processamento em torno de uma semana para revelar quase a totalidade das subchaves.

A Tabela 5.5 apresenta os resultados referentes às análises CPA e CEMA. Nota-se que estas análises também encontram as subchaves secretas do algoritmo, de um modo geral com um menor número de traços. Apenas em um caso, sobre a arquitetura PIPE-2 com FIFO, a análise CEMA exigiu um maior número de traços em relação à DEMA. Em dois outros casos, existe um equilíbrio, tal como nas arquiteturas PIPE-4 sem FIFO e PIPE-2 com FIFO. Com base nesta amostra de resultados é possível afirmar que em geral ataques por correlação são mais eficientes que os demais.

Tabela 5.5 Número de traços necessários para revelar a chave criptográfica utilizando análises CEMA e CPA.

	PIPE-2		**PIPE-4**		**PIPE-8**	
Arquiteturas	**CEMA (#)**	**CPA (#)**	**CEMA (#)**	**CPA (#)**	**CEMA (#)**	**CPA (#)**
Síncrona	20025	7849	16285	x	71056	ne
Síncrona com FIFO	90007	53707	16058	ne	ne	ne
GALS	-	-	ne	ne	ne	ne
GALS com FIFO	-	-	-	ne	-	ne

(-) : não descoberta
(ne): não executada
(#): número de traços para encontrar a chave completa
(x): apenas a subchave da SBOX8 não foi revelada.

EFICIÊNCIA DOS ATAQUES

A Figura 5.11 apresenta o número mínimo necessário de traços para revelar cada uma das subchaves do algoritmo DES.

Com estes resultados é possível concluir que, de um modo geral, o ataque DPA exige um maior número de traços em relação aos demais ataques. A análise por correlação aplicada sobre os traços de consumo de potência (CPA) é ligeiramente mais eficiente que DPA considerando os resultados obtidos para cada uma das arquiteturas. De um modo geral, as análises sobre traços de radiações eletromagnéticas mostram-se mais eficientes em relação ao consumo de potência. Em alguns casos o número de traços é praticamente o mesmo, em raras ocasiões exige um número maior de traços. O mesmo comportamento é encontrado entre as análises DEMA e CEMA, ou seja, as análises por

correlação demonstram de um modo geral, um melhor comportamento em relação a análises diferenciais. Com estes resultados conclui-se que ataques por correlação exigem um número menor de traços para revelar a chave secreta de um algoritmo ao custo de um maior tempo de processamento para executar as análises.

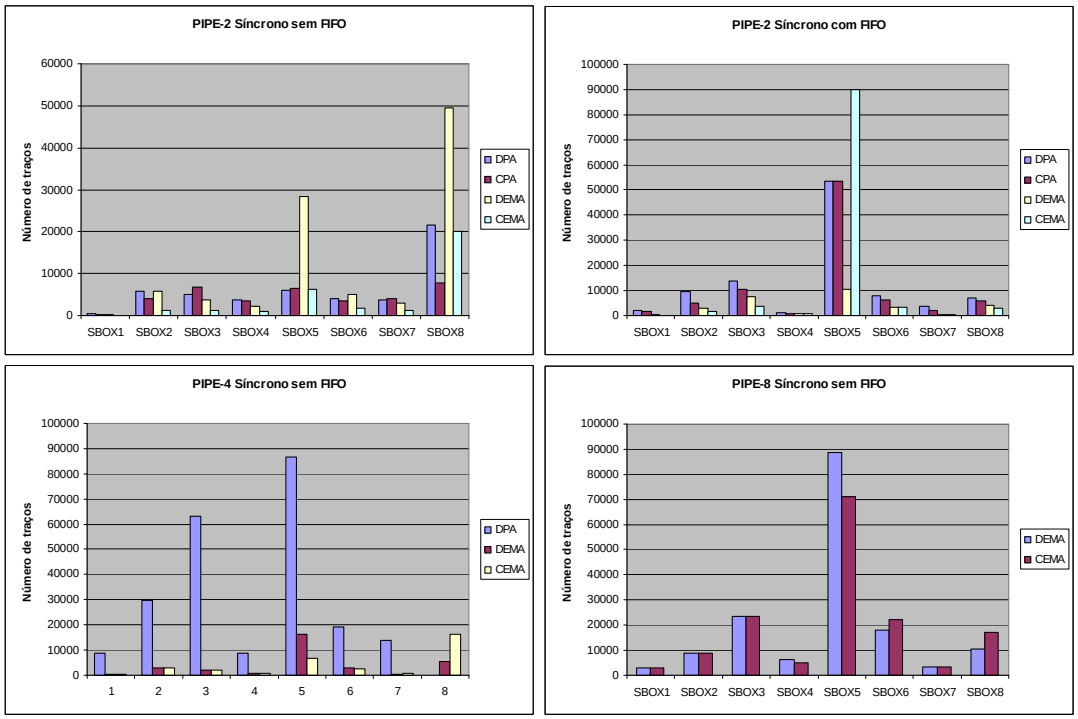


Figura 5.11 Gráficos comparando o número de traços necessários para revelar as subchaves de cada SBOX do algoritmo DES.

AVALIAÇÃO DA ROBUSTEZ VERSUS PROCESSAMENTO PARALELO

Os gráficos apresentados na Figura 5.12 demonstram a robustez obtida com o ruído inserido pelo processamento paralelo dos estágios operando em modo síncrono. Os resultados não demonstram uma vantagem significativa do processamento paralelo para neutralizar os ataques, confirmando as conclusões obtidas por Standaert et al. em [STA04]. Embora não contribuam significativamente no aumento da robustez quando usados em arquiteturas síncronas, esta contribuição potencializa a robustez a ataques quando acrescida à aleatoriedade inserida pelas arquiteturas GALS PIPE.

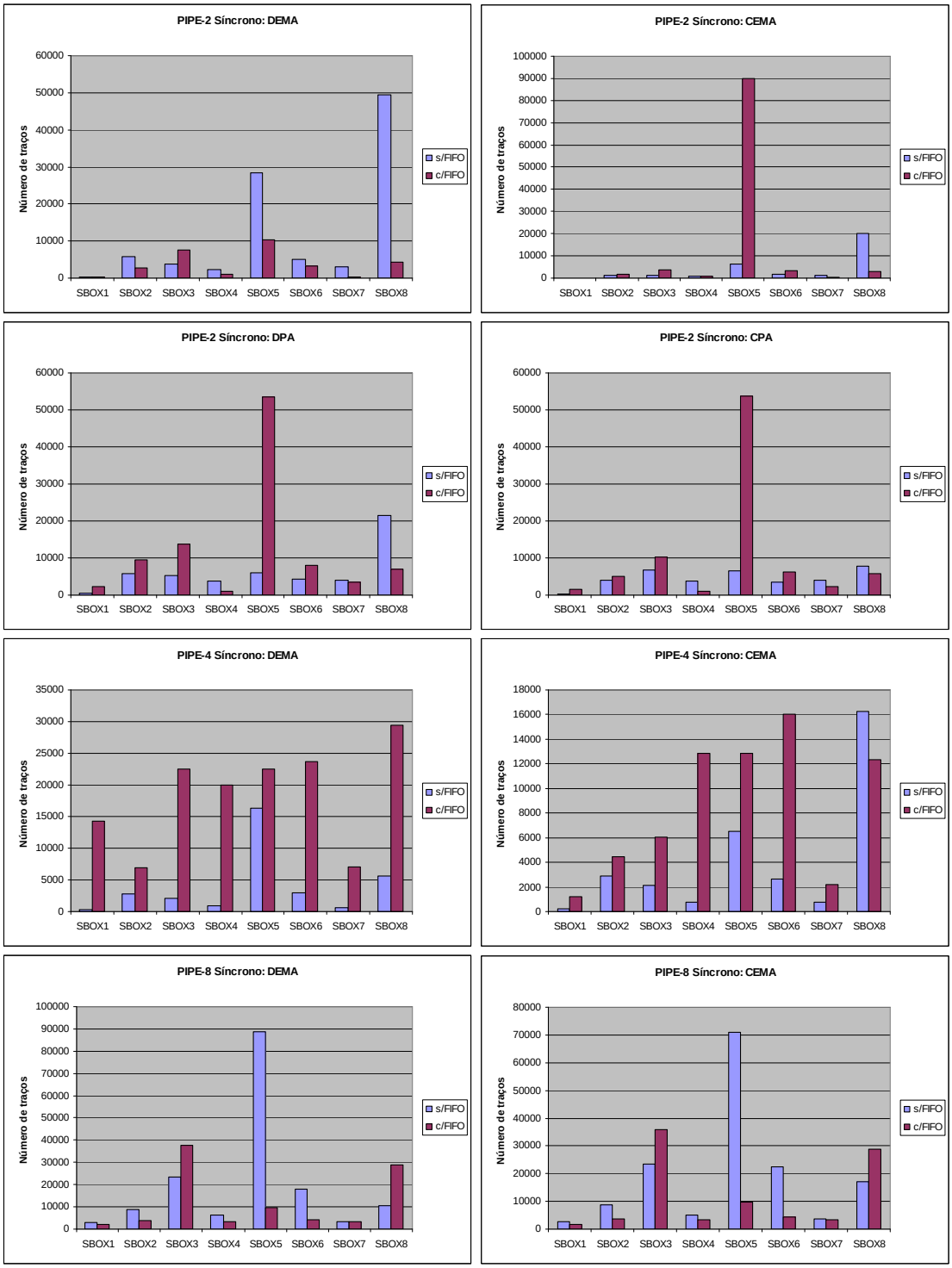


Figura 5.12 Comparação da efetividade do processamento paralelo para neutralizar ataques DPA e DEMA.

AVALIAÇÃO DA ROBUSTEZ VERSUS REPLICAÇÃO DO HARDWARE

Os gráficos da Figura 5.13 apresentam uma comparação dos resultados obtidos com as análises DEMA e CEMA realizadas sobre as arquiteturas PIPE propostas. Nota-se, a partir destes gráficos, que na maioria dos casos, (59% das 32 análises efetuadas), a

arquitetura com maior número de estágios exigiu um número maior de traços para revelar a chave secreta do algoritmo e a arquitetura com menos estágios exigiu menos traços na maior parte das análises. Com estas informações é possível concluir que apenas a replicação do bloco de encriptação do algoritmo em uma arquitetura síncrona já aumenta a robustez a ataques por análise de radiação eletromagnética.

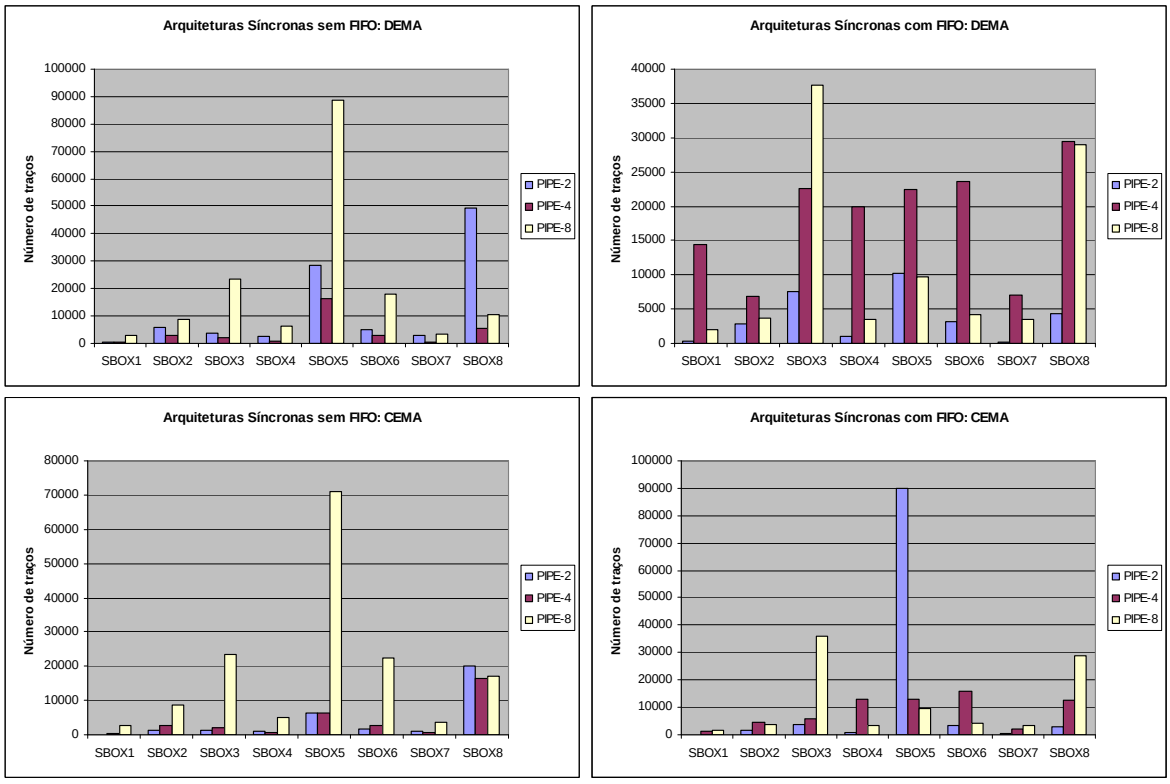


Figura 5.13 Comparação de análises DEMA realizadas sobre arquiteturas DES PIPE com diferentes números de estágios.

Os resultados até aqui comprovam mais uma vez que circuitos síncronos são vulneráveis a ataques SCA. A eficiência destas análises depende da precisão do sistema de medição usado. Ou seja, quanto maior for a razão SNR dos traços de consumo de potência e de radiação eletromagnética medidos, mais eficientes serão as análises. Considera-se uma análise eficiente aquela que conseguir revelar a chave criptográfica de um criptosistema com o menor número possível de traços.

As análises sobre as arquiteturas GALS propostas não obtiveram êxito. Nenhuma das oito subchaves referentes às SBOXes atacadas foi revelada, mesmo usando 100 mil traços obtidos de dados distintos. Isto comprova que as arquiteturas propostas são robustas a este tipo de ataques. As Figuras a seguir demonstram os resultados finais obtidos para cada uma das arquiteturas propostas e submetidas à avaliação de robustez.

A Figura 5.14 mostra os resultados das análises DPA e DEMA sobre a SBOX3 em todas as arquiteturas DES PIPE GALS implementadas. Cada gráfico mostra os 64 traços hipóteses correspondentes às subchaves possíveis. Os traços pretos correspondem à hipótese correta da subchave, os traços vermelhos correspondem à hipótese incorreta da

subchave, já os traços azuis correspondem aos demais traços hipóteses. Quando o traço hipótese de maior pico é um traço preto, o ataque é bem sucedido, ou seja, a subchave correta foi encontrada. Por outro lado, se o traço com maior pico não corresponde à subchave correta, este traço é representado em vermelho correspondendo a uma hipótese incorreta, ou seja, um ataque mal sucedido.

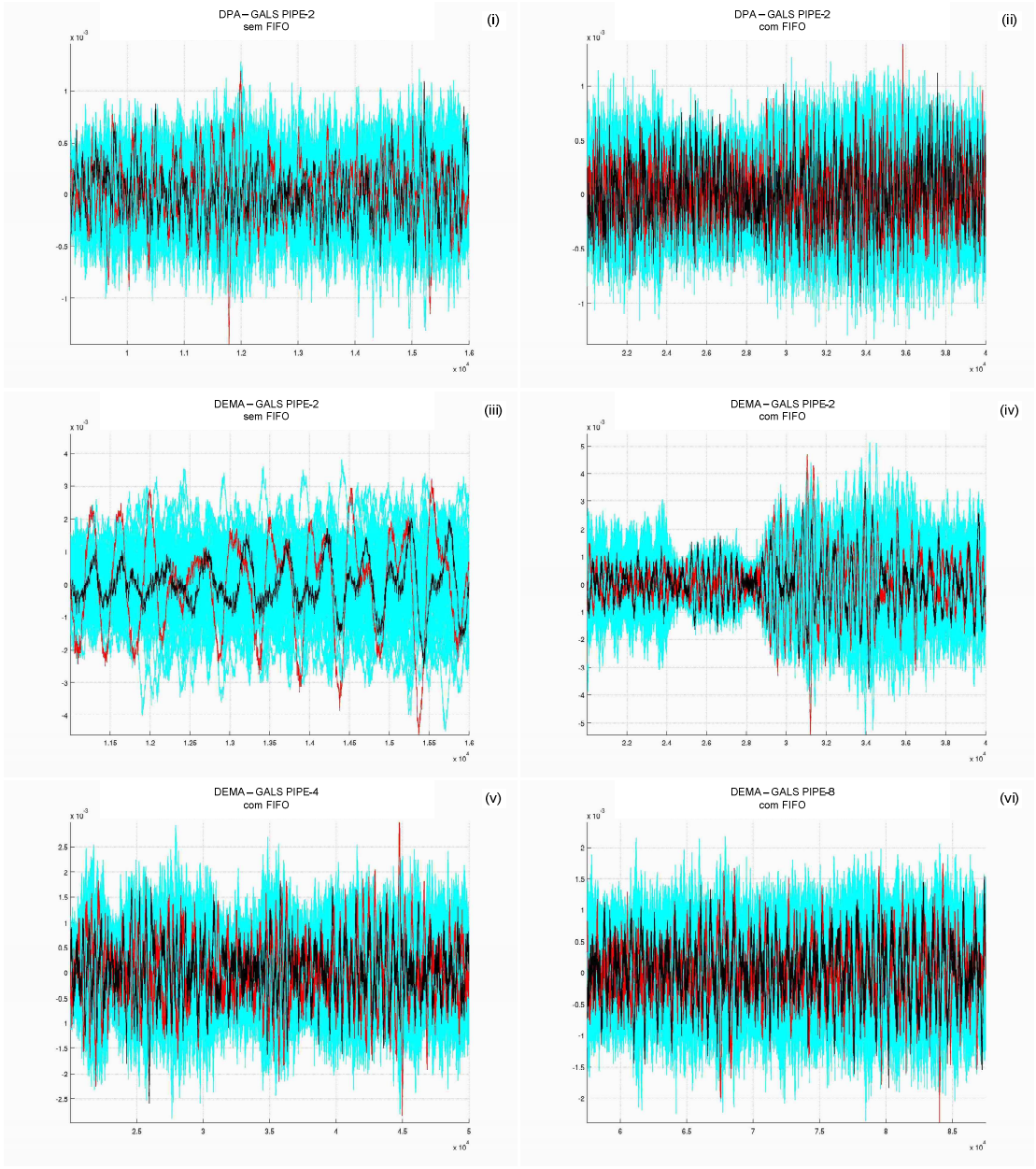


Figura 5.14 Resultados das análises DPA para as arquiteturas PIPE-2: (i) GALS e (ii) GALS com FIFO. Resultados das análises DEMA para as mesmas arquiteturas são mostrados em (iii) e (iv). As análises DEMA desenvolvidas sobre as arquiteturas DES PIPE-4 e DES PIPE-8 versões com FIFO são apresentadas em (v) e (vi). Os traços hipóteses pretos correspondem à subchave correta, os traços vermelhos correspondem à subchave incorreta e os demais traços azuis completam as 64 hipóteses possíveis de subchave para a SBOX3 do algoritmo DES.

Na Figura 5.14 mostra-se em (i) e (ii) os resultados das análises DPA sobre as arquiteturas GALS PIPE-2 nas versões sem e com FIFO. Os resultados das análises DEMA sobre as mesmas arquiteturas aparecem respectivamente em (iii) e (iv). As arquiteturas PIPE-4 e PIPE-8 com FIFO foram submetidas apenas a análises DEMA, devido ao longo tempo de medição e a restrições de espaço em disco necessário (na ordem de Terabytes). Os resultados aparecem em (v) e (vi). Em todas as análises, nota-se que não foi possível encontrar a subchave secreta, mesmo usando 100 mil traços nas análises. Observa-se também certo equilíbrio entre os traços hipóteses. Em todos os casos não existe um traço hipótese que tenha uma amplitude destacada entre os demais. Logo, as margens de traços hipóteses são pequenas, o que reduz a eficiência dos ataques.

Este mesmo efeito ocorre nas demais SBOXes do algoritmo submetidas aos ataques. A aleatoriedade inserida no processamento do algoritmo pela arquitetura proposta demonstra um aumento da robustez a ataques DPA e DEMA como necessário em sistemas criptográficos robustos. Embora os resultados tenham comprovado o comportamento desejado das arquiteturas, outras análises específicas para este tipo de contramedida devem ser realizadas sobre as arquiteturas propostas, tais como as análises propostas por Nagashima et al. [NAG07] discutida anteriormente. Assim, poder-se-ia afirmar que mesmo submetido a análises específicas o método proposto mostra-se robusto.

5.6 CONCLUSÃO

Este Capítulo apresentou uma nova proposta de arquitetura pipeline GALS para o algoritmo criptográfico DES, com o objetivo de neutralizar a ação de análises de consumo de potência e de radiações eletromagnéticas. Pela primeira vez obtém-se robustez combinando o uso de replicação de blocos de encriptação operando em modo pipeline e comunicação assíncrona entre os blocos. Cada bloco de encriptação é suprido com sinais de relógio com freqüências distintas sorteadas aleatoriamente antes de processar um dado para aumentar a aleatoriedade do processamento do circuito visando neutralizar ataques DPA e DEMA.

O compromisso entre robustez e área é a principal preocupação do método. Comparado a uma implementação regular do DES (não-pipeline), a arquitetura proposta realmente apresenta alto custo em área. Entretanto, comparada ao estado da arte de lógicas assíncronas propostas para conceber sistemas criptográficos seguros tal como STTL, a maioria das arquiteturas GALS pipeline exige menos área. Além disso, os resultados obtidos com as análises DPA, DEMA, CPA e CEMA demonstram excelente robustez. Quando as mesmas análises são aplicadas a arquiteturas síncronas nas mesmas condições, praticamente todos os estudos de casos obtiveram sucesso na tarefa de descobrir a chave secreta.

As avaliações de robustez realizadas confirmam também as previsões de Standaert et al. em [STA04], onde os Autores afirmam que apenas o uso da implementação pipeline não é suficiente para resistir a ataques SCA. Porém, o uso de arquiteturas pipelines onde

cada estágio tem seu próprio domínio de freqüência e ainda, o sinal de relógio em cada estágio tem sua freqüência alterada dinamicamente mostra-se robusta a ataques.

A partir destes experimentos nota-se também que as análises por correlação de potência são mais eficientes. De um modo geral, todos os ataques por correlação revelaram as chaves secretas com um número menor de traços em relação às análises propostas por Kocher [KOC96] [KOC99]. Porém, nem mesmo o uso de um modelo de potência correlacionado às medições reais geradas, tal como o usado nas análises CEMA e CPA, foi suficiente para neutralizar a contramedida proposta.

A robustez da abordagem proposta está diretamente associada à aleatoriedade inserida no processamento associado ao método GALS de projeto. O aumento do número de estágios da arquitetura paralelizando o processamento do algoritmo DES permite que se tenha uma maior aleatoriedade no sistema e conseqüentemente se torne mais difícil a identificação de informações que escapem pelos canais colaterais do sistema.

6. CONCLUSÃO E TRABALHOS FUTUROS

Este Capítulo apresenta uma relação das contribuições do trabalho, na Seção 6.1. Em seguida, mostra-se um conjunto de conclusões resultantes do trabalho na Seção 6.2. Finalmente a Seção 6.3 apresenta um conjunto de sugestões de trabalhos futuros.

6.1 CONTRIBUIÇÕES DO TRABALHO

Dentre as contribuições deste trabalho citam-se aqui as mais importantes:

- **Revisão do problema da fuga de informações por canais laterais:** a primeira contribuição deste trabalho consiste na revisão do problema da fuga de informações através de características físicas dos circuitos, tais como o tempo de processamento, o consumo de potência (análises DPA) e a radiação eletromagnética (análises DEMA), conforme apresenta o Capítulo 2.
- **Revisão do estado da arte**: a segunda contribuição deste trabalho é a elaboração do estado da arte de propostas de contramedidas às análises DPA. No que diz respeito ao alvo desta pesquisa, apresenta-se uma revisão de propostas subdivididas em três grupos, cada um destes contendo uma abordagem para neutralizar análises DPA. O primeiro grupo emprega mascaramento de dados, o segundo inserção de aleatoriedade no sistema, tal como proposto nesta tese e o terceiro grupo emprega a uniformização do consumo de potência durante o processamento de dados. O Capítulo 3 descreve esta contribuição.
- **Biblioteca STTL:** a terceira contribuição do trabalho consiste na proposta, elaboração e validação de uma biblioteca de células para a prototipação eficiente da lógica STTL em FPGAs. A lógica STTL foi inicialmente proposta e validada através do uso de uma biblioteca dedicada, usando standard cells CMOS voltada para implementação em ASICs. Esta biblioteca apresenta as mesmas premissas de funcionamento daquela proposta por Razafindraibe et al. [RAZ07]. Os protótipos são baseados em estilo de projeto assíncrono insensível a atrasos, usando hard macros para obedecer a restrições de tempo. O Capítulo 4 deste trabalho detalha esta biblioteca de prototipação.
- **Implementação de script para automatizar fluxo de projeto de circuitos STTL:** a quarta contribuição deste trabalho é a implementação de um script para converter descrições verilog de portas lógicas (SR) contidas em netlists para instâncias de hard macros STTL descritas em VHDL a fim de automatizar o fluxo de projeto STTL.
- **Implementações do algoritmo DES em pipeline GALS:** a quinta contribuição deste trabalho é a proposta da primeira implementação do algoritmo DES em modo pipeline usando o método GALS de projeto. As ilhas síncronas de encriptação usam um sinal de relógio com freqüências que mudar de forma pseudo-aleatória, o que aumenta a robustez do sistema criptográfico a análises de consumo de potência e de radiação eletromagnética. O Capítulo 5 deste trabalho apresenta e discute esta contribuição.

- **Infraestrutura de avaliação dos sistemas**: a sexta contribuição do trabalho surgiu da necessidade de se desenvolver uma infraestrutura que permita a prototipação de sistemas criptográficos, tanto usando a lógica STTL quanto usando a abordagem pipeline GALS, oferecendo recursos que permitam a comunicação com um hospedeiro e com um osciloscópio para medição dos traços necessários.
- **Sistema de medição de traços de consumo:** um sistema para medição do consumo de potência e a radiação eletromagnética produzidos por um sistema prototipado em FPGA foi proposto neste trabalho.

Publicações: o desenvolvimento deste trabalho resultou até o momento em um conjunto de publicações. Dois trabalhos referem-se ao início do doutorado, envolvendo o uso de hard macros em interfaces assíncronas em FPGAs e implementação de redes intrachip com método GALS de projeto, respectivamente. Os quatro trabalhos seguintes dizem respeito às atividades desenvolvidas durante o estágio sanduíche, realizado no LIRMM (em Montpellier, França). O último trabalho é a primeira publicação de resultados iniciais de avaliação da robustez da abordagem GALS pipeline. A publicação mais extensa de resultados é trabalho em andamento.

1. Pontes, J.; Soares, R.; Carvalho, E.; Moraes, F.; Calazans, N. "**SCAFFI: An Intrachip FPGA Asynchronous Interface based on Hard Macros**". In: 25th IEEE International Conference on Computer Design (ICCD'07), May 2007, pp. 541-546.
2. Pontes, J.; Moreira, M.; Soares, R.; Calazans, N. "**Hermes-GLP: A GALS Network on Chip Router with Power Control Techniques**". In: IEEE Computer Society Annual Symposium on VLSI Design (ISVLSI 2008), Apr 2008, pp. 347-352.
3. Soares, R.; Lomné, V.; Calazans, N.; Maurine, P.; Torres, L.; Robert, M. "**Evaluating the Robustness of Secure Triple Track Logic through Prototyping**". In: 21st Symposium on Integrated Circuits and Systems Design (SBCCI'08), Sep 2008, pp. 193-198.
4. Lomné,V.; Ordas, T.; Maurine, P.; Torres, L.; Robert, M.; Soares, R.; Calazans, N. "**Triple Rail Logic Robustness against DPA**". In: International Conference on Reconfigurable Computing and FPGA (RECONFIG'08), Dec 2008, pp. 415-420.
5. Lomné, V.; Soares, R.; Ordas, T.; Calazans, N.; Maurine, P.; Torres, L.; Robert, M. "**Evaluation on FPGA of Triple Rail Logic Robustness against DPA and DEMA**". In: Design, Automation and Test in Europe Conference and Exposition (DATE'09), Mar 2009, pp. 634-639.
6. Lomné, V.; Dehbaoui, A.; Ordas, T.; Maurine, P.; Torres, L.; Robert, M.; Soares, R.; Calazans, N.; Moraes, F. "**Secure Triple Track Logic Robustness Against Differential Power Analysis Attacks**". *Journal of Integrated Circuits and Systems*, vol. 4-1, Mar 2009, pp. 20-28.

7. Soares, R.; Calazans, N.; Lomné, V.; Dehbaoui, A.; Maurine, P.; Torres, L. "**A GALS Pipeline DES Architecture to Increase Robustness Against DPA and DEMA Attacks**". In: 23rd Symposium on Integrated Circuits and Systems Design (SBCCI'10), Sep 2010, pp. 115-120.

6.2 CONCLUSÕES

Com a realização deste trabalho foi possível verificar que a tecnologia CMOS usada no projeto de circuitos digitais tem características de consumo de potência dependentes dos dados computados pelo circuito. Além disso, o paradigma síncrono de projeto de circuitos, utilizado pela grande maioria dos equipamentos eletrônicos, usa um sinal global de relógio para sincronizar todas as operações executadas por um sistema. Estas duas características são as principais responsáveis pela fuga de informações relevantes que tornam sistemas computacionais vulneráveis a análises baseadas no consumo de potência [KOC99] e na radiação eletromagnética [GAN01].

A revisão da literatura sobre o problema da fuga de informações através de canais laterais revelou, de um modo geral, que existem três possíveis abordagens para conceber sistemas seguros evitando a correlação de dados sigilosos através da fuga de informações. Duas abordagens, mascaramento de dados e inserção de ruído e aleatoriedade, não alteram nem as características da lógica à qual são projetados os circuitos nem o fluxo de projeto. Elas criam meios para ocultar a fuga de informações existente nos circuitos. Já a abordagem por uniformização do consumo de potência visa propor alterações na estrutura lógica e no fluxo de projeto a fim de obter circuitos onde o consumo de potência, o tempo de propagação dos sinais e a radiação eletromagnética emitida pelo circuito sejam constantes e independentes dos dados computados.

Teoricamente, a tarefa de projetar circuitos com consumo de potência uniforme parece ser uma forma simples e eficaz de eliminar a fuga de informações. Porém na prática esta tarefa é complexa e exige um grande esforço de projeto, tal como a construção de bibliotecas lógicas personalizadas e a definição de fluxos de projeto com diversas restrições para obter caminhos balanceados para que o circuito tenha um consumo de potência o mais próximo possível do caso ideal. Apesar destes esforços, um sistema de medição robusto capaz de medir e armazenar algumas centenas de milhares de traços, adicionado a um longo tempo de processamento possibilita revelar a chave secreta na maioria das situações.

Com a revisão da literatura foi possível verificar que a abordagem por uniformização do consumo de potência implica custos elevados em termos de área, latência e potência dissipada. Como exemplo desta, o uso de lógica assíncrona com codificação em trilha dupla representa um aumento significativo da área. Experimentos realizados com a lógica STTL [SOA08] demonstraram um aumento significativo da robustez a custos similares ao de lógicas em trilha dupla. Portanto, abordagens alternativas acabam sendo soluções de menor custo em termos de área e latência. Por outro lado, sabe-se que estas abordagens não eliminam a fuga de informações, apenas a ocultam de tal modo que se tenha esforço

computacional elevado e algoritmos complexos para revelar os dados secretos, sendo ainda possível obter análises com sucesso.

O trabalho proposto foi desenvolvido visando obter arquiteturas robustas a ataques por análise do consumo de potência e da radiação eletromagnética através da inserção de aleatoriedade no consumo de potência do circuito. O trabalho combina o uso do modo de implementação pipeline tal como proposto em [STA04] e [HUI07], porém com comunicação assíncrona entre os estágios, conforme a metodologia GALS de projeto. Além disso, os estágios operam em diferentes domínios de freqüência com variação da freqüência do sinal do relógio para cada dado processado, similar aos trabalhos propostos em [BUC05] [GUR06] [LU08] e [ZAF08].

Implementou-se o algoritmo DES em oito versões usando arquitetura pipeline: PIPE-2, PIPE-4, PIPE-8 e PIPE-16. Cada uma destas pode operar em modo síncrono e modo GALS. Estas arquiteturas, prototipadas em FPGA, foram submetidas a ataques DPA, CPA, DEMA e CEMA, com exceção da arquitetura PIPE-16 que não foi possível prototipar na plataforma usada nos experimentos. As avaliações de robustez realizadas confirmam os resultados obtidos por Standaert et al. em [STA04], onde os Autores afirmam que apenas o uso de implementação pipeline não é suficiente para resistir a ataques DPA. O processamento paralelo dos estágios demonstrou um pequeno aumento da robustez, porém todas as análises obtiveram sucesso.

Um dado interessante obtido a partir da avaliação dos resultados foi que, na maioria dos casos (59% das 32 análises efetuadas) a arquitetura com maior número de estágios exigiu um número maior de traços para revelar a chave secreta do algoritmo, significando nesse caso um aumento da robustez. Todas as arquiteturas síncronas tiveram as chaves criptográficas reveladas pelas análises. Já as arquiteturas GALS, com diferentes domínios de relógios e variação de freqüência não tiveram nenhuma subchave revelada pelas análises.

Com a avaliação dos resultados das análises pode-se concluir que os ataques por correlação (CPA e CEMA) são mais eficientes em relação aos demais. Uma redução no número de curvas foi constatada nas avaliações efetuadas. Esta eficiência é obtida ao custo de um aumento no tempo de processamento, em geral cinco vezes maior que ataques DPA ou DEMA. Em relação ao canal lateral analisado, as análises de radiação eletromagnética se mostraram ligeiramente melhores em relação a ataques por consumo de potência. Isto se deve também ao ruído existente no sistema de medição usado durante os experimentos, que prejudica as análises por consumo de potência.

A robustez obtida através da tese proposta está diretamente associada à aleatoriedade inserida ao processamento. O aumento do número de estágios da arquitetura particiona o processamento das rodadas do algoritmo DES em estágios, permitindo que se tenha uma maior aleatoriedade no sistema e conseqüentemente torne mais difícil a tarefa de identificar informações por canais laterais do sistema. Os resultados das análises comprovaram que o aumento do número de estágios, o processamento paralelo das rodadas e a variação da freqüência do sinal do relógio em cada estágio

contribuem para dificultar os ataques, ou até mesmo inviabilizá-los. Acredita-se que usando novas técnicas que empreguem um pré-processamento de traços a fim de ressincronizá-las haja uma possibilidade de contra-atacar a implementação proposta. Deste modo, pode ser possível obter resultados que comprovem os limites da imunidade obtida com esta tese proposta.

6.3 TRABALHOS FUTUROS

Esta Seção apresenta um conjunto de sugestões para trabalhos futuros.

A primeira sugestão de trabalhos é a avaliação da robustez usando alguma técnica de pré-processamento sobre os traços coletados tais como filtros de sinais como proposto por Nagashima et al. em [NAG07]. Deste modo, a avaliação da robustez das arquiteturas propostas em relação às análises por consumo de potência e radiação eletromagnética torna-se mais real.

A segunda sugestão é propor uma técnica que permita a ressincronização dos traços de consumo de potência e radiação eletromagnética, de modo que as análises possam ser executadas com maior precisão.

O processo de validação e prova de conceito das arquiteturas propostas neste trabalho foram realizados sobre dispositivos FPGAs conforme descrito nos diversos experimentos. Estes dispositivos compostos de várias estruturas programáveis tais como blocos lógicos onde as LUTs são configuradas e canais de interconexão de sinais por onde são roteados os fios de comunicação. Embora seja um dispositivo confiável para a prova de conceito destes experimentos, estas estruturas programáveis consomem energia adicional e são fontes potenciais de geração de ruído. Portanto, sugere-se aqui o projeto de um circuito de aplicação específica (ASIC) para as arquiteturas aqui propostas visando avaliar sua robustez em um chip dedicado a este propósito.

As arquiteturas propostas apresentam uma vulnerabilidade decorrente do modo pipeline de implementação. As análises propostas por Kocher realizam ataques na primeira e na última rodada do algoritmo. Logo, quando apenas um dado é processado no pipeline, o método aqui proposto comporta-se tal como o método proposto por Lu et al. [LU08]. A terceira sugestão é a proposta de um método que evite esta vulnerabilidade ocorrida na entrada e na saída do pipeline. Um dado gerado aleatoriamente deve ser processado paralelamente com a primeira rodada do algoritmo, bem como, com a última rodada processada no último estágio.

Uma sugestão interessante é o projeto de um bloco de encriptação genérico, ou seja, um bloco que permite a execução de $n \leq 16$ rodadas do algoritmo. Assim, um pipeline pode ter a opção de executar rodadas em diferentes estágios. Isto além de aumentar a aleatoriedade no processamento também dificultaria as análises de radiação eletromagnética, pois a execução dar-se-ia em diferentes regiões do chip em instantes diferentes. Outra opção é o projeto de uma rede de interconexão blocos de encriptação genéricos, o que expande o processamento aleatório em duas dimensões do chip.

Obviamente, estes métodos aumentam ainda mais a penalidade em área e potência dissipada, mas a segurança obtida pode justificar tal método.

A automatização do projeto também é importante para o uso de um método de contramedida. Uma sugestão neste sentido é o desenvolvimento de uma ferramenta de geração automática de pipeline DES. Esta ferramenta pode ser capaz de gerar arquiteturas pipeline de diversos tamanhos, com 2 a 16 blocos de encriptação do DES. Cada estágio pode ter seu hardware facilmente replicado. Porém, um cuidado necessário é levar em conta no projeto do circuito a geração de subchaves, pois como mostrado no Capítulo 5, cada estágio possui associado uma rotação de quantidades diferentes de bits.

Otimizações na arquitetura podem ser realizadas de modo a reduzir os custos em área de sua implementação. O subsistema de relógio também pode ser otimizado de modo a oferecer um maior número de freqüências de sinais de relógio a menores custos de implementação visando obter uma melhor aleatoriedade e conseqüentemente uma maior robustez a ataques.

REFERÊNCIAS BIBLIOGRÁFICAS

[ADE08] Adee, S. “The Hunter for the Kill Switch”. *IEEE Spectrum*, vol. 45-5, Jan 2008, pp. 34-39.

[AKK01] Akkar, M.; Giraud, C. “An Implementation of DES and AES Secure Against some Attacks”. In: Third International Workshop on Cryptographic Hardware and Embedded Systems (CHES’01), May 2001, pp. 309-318.

[AKI03] Akishita, T.; Takagi T. “Zero Value Point Attacks on Elliptic Curve Cryptosystem”. In: 6th International Conference on Information Security (ISC’03), Oct 2003, pp. 218-233.

[AUV00] Auvergne, D.; Daga, J.; Rezzoug, M. “Signal Transition Time Effect on CMOS Delay Evaluation”. *IEEE Transaction on Circuits and Systems*, vol. 47-9, Sep 2000, pp. 1362-1369.

[BAD07] Baddam, K.; Zwolinski, M. “Evaluation of Dynamic Voltage and Frequency Scaling as a Differential Power Analysis Countermeasure”. In: 20th International Conference on VLSI Design (VLSID’07), Jan 2007, pp. 854-862.

[BAG07] Baignères, T.; Stern, J.; Vaudenay, S. “Linear Cryptanalysis of Non Binary Ciphers”. In: 14th International Conference on Selected Areas in Cryptography (SAC’07), Aug 2007, pp. 184-211.

[BEN03] Benini, L.; Macii, A.; Macii, E.; Omerbegovic, E.; Pro, F.; Poncino, M. “Energy-Aware Design Techniques for Differential Power Analysis Protection”. In: 40th Design Automation Conference (DAC '03), Jun 2003, pp. 36-41.

[BER01] Bergamaschi, R.; Bhattacharya, S. ; Wagner, R. ; Fellenz, C.; Muhlada, M.; White, F.; Daveau, J.; Lee, W. “Automating the Design of SoC using Cores”. *IEEE Design & Test of Computers*, vol. 18-5, Sep 2001, pp. 32-45.

[BER92] van Berkel, K. “Beware the Isochronic Fork”. Integration, The VLSI Journal, vol. 13-2, Jun 1992, pp. 103-128.

[BEV03] Bevan, R.; Knudsen, E. “Ways to Enhance Differential Power Analysis”. In: International Conference on Information Security and Cryptology (ICISC’03), Jan 2003, pp. 327-342.

[BHA09] Bhasin, S.; Danger, J.; Flament, F.; Graba, T.; Guilley, S.; Mathieu, Y.; Nassar, M.; Sauvage, L.; Selmane, N. “Combined SCA and DFA Countermeasures Integrable in a FPGA Design Flow”. In: International Conference on Reconfigurable Computing and FPGAs (Reconfig’09), Dec 2009, pp. 213-218.

[BIH90] Biham, E.; Shamir, A. “Differential Cryptanalysis of DES-like Cryptosystems”. In: 10th Annual International Cryptology Conference (CRYPTO’90), Aug 1990, pp. 2-21.

[BON97] Boneh, D.; DeMillo, R.; Lipton, R. “On the Importance of Checking Cryptographic Protocols for Faults”. In: International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT’97), May 1997, pp. 37-51.

[BOU05] Bouesse, G.; Renaudin, M.; Dumont, S.; Germain, F. “DPA on Quasi Delay Insensitive Asynchronous Circuits: Formalization and Improvement”. In: Design, Automation and Test in Europe Conference and Exposition (DATE’05), Mar 2005, pp. 424-429.

[BUC05] Bucci, M.; Luzzi, R.; Guglielmo, M.; Trifiletti, A. “A Countermeasure against Differential Power Analysis based on Random Delay Insertion”. In: IEEE International Symposium on Circuits and Systems (ISCAS’05), May 2005, pp. 3547-3550.

[BRE05] Brej, C. “Early Output Logic and Anti-Tokens” Tese de Doutorado, School of Computer Science, University of Manchester, Sep 2005.

[BRI04] Brier, E.; Clavier, C.; Olivier, F. “Correlation Power Analysis with a Leakage Model”. In: 6th International Workshop on Cryptographic Hardware and Embedded Systems (CHES’04), Aug 2004, pp. 16-29.

[CAD05] Cadence Design Systems, Inc. “Encounter User Guide”. User Guide, May 2005, vol. 4.1.5, 738p.

[CAL98] Calazans, N. L. V. “Projeto Lógico Automatizado de Circuitos Lógicos Seqüênciais”. Imprinta, 1998, 342p.

[CAN08] Canright, D.; Batina, L. “A Very Compact “Perfectly Masked” S-box for AES”. In: 6th International Conference on Applied Cryptography and Network Security (ACNS’08), Jun 2008, pp. 446-459.

[CAN09] Cannière, C.; Dunkelman, O.; Knezevic, M. “KATAN and KTANTAN – A Family of Small and Efficient Hardware-Oriented Block Ciphers”. In: 11th International Workshop on Cryptographic Hardware and Embedded Systems (CHES’09), Aug 2009, pp. 272-288.

[CCR06] The Common Criteria Recognition Arrangement. “Common Criteria for Information Technology Security Evaluation: Part I: Introduction and general model”. CCMB-2006-09-001, Version 3.1, Revision 1, Sep 2006, 86 p. Capturado em http://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R1.pdf.

[CHA02] Chari, S.; Rao, J.; Rohatgi, P. “Template Attacks”. In: 4th International Workshop on Cryptographic Hardware and Embedded Systems (CHES’02), Aug 2002, pp. 13-28.

[CHA08] Chang, C.; Huang, S.; Ho, Y.; Lin, J.; Wang, H.; Lu, Y. “Type-Matching Clock Tree for Zero Skew Clock Gating”. In: 45th Design Automation Conference (DAC’08), Jun 2008, pp. 714-719.

[CHA84] Chapiro, D. “Globally Asynchronous Locally Synchronous”. Tese de Doutorado, Stanford University, Oct 1984, 134 p.

[CHE06] Chen, Z.; Zhou, Y. “Dual-Rail Random Switching Logic: A Countermeasure to Reduce Side Channel Leakage”. In: 8th International Workshop on Cryptographic Hardware and Embedded Systems (CHES’06), Oct 2006, pp. 242-254.

[CIE03] Ciet, M.; Neve, M.; Peeters, E.; Quisquater, J.-J. “Parallel FPGA Implementation of RSA with Residue Number Systems - Can Side-Channel Threats be Avoided?”. In: 46th IEEE International Midwest Symposium on Circuits and Systems (MWSCAS'03), Dec 2003, pp. 27-30.

[CIL10] Cilio, W.; Linder, M.; Porter, C.; Di, J.; Smith, S.; Thompson, D. “Side-Channel Attacks Mitigation Using Dual-Spacer Dual-Rail Delay-Insensitive Logic (D3L)”. In: IEEE SoutheastCon 2010, Mar 2010, pp. 471-474.

[COR06] Cortadella, J.; Kondratyev, A.; Lavagno, L.; Sotiriou, C. “Desynchronization: Synthesis of Asynchronous Circuits from Synchronous Specifications”. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25-10, Oct 2006, pp. 1904-1921.

[CLA00] Clavier, C.; Coron, J.; Dabbus, N. “Differential Power Analysis in the Presence of Hardware Countermeasures”. In: 2nd International Workshop on Cryptographic Hardware and Embedded Systems (CHES’00), Aug 2000, pp. 13-25.

[DAV92] David, I.; Ginosar, R.; Yoeli, M. “An Efficient Implementation of Boolean Function as Self-Timed Circuits”. *IEEE Transaction on Computers*, vol. 41-1, Jan 1992, pp. 2-11.

[DEH09] Dehbaoui, A.; Lomne, V.; Maurine, P.; Torres, L. “Magnitude Squared Incoherence EM Analysis for Integrated Cryptographic Module Localisation”. *IET Electronic Letters*, vol. 45-15, Jul 2009, pp. 778-780.

[DEH10] Dehbaoui, A.; Ordas, T.; Lomné, V.; Maurine, P.; Torres, L.; Robert, M. “Incoherence Analysis and its Application to Time Domain EM Analysis of Secure Circuits”. In: Asia-Pacific Symposium on Electromagnetic Compatibility (APEMC’10), Apr 2010, pp. 1039-1042.

[DIF76] Diffie, W.; Hellman, M. “New Directions in Cryptography”. *IEEE Transactions on Information Theory*, vol. 22-6, Nov 1976, pp. 644-654.

[DOB09] Dobkin, R.; Ginosar, R. “Two-phase Synchronous with Sub-cycle Latency”. *Integration, the VLSI Journal*, vol. 42-3, Jun 2009, pp. 364-375.

[DPA09] DPA Contest. Capturado em: http://www.dpacontest.org/index.php, December 2009.

[DUN07] Dunkelman, O.; Keller, N. “A New Criterion for Nonlinearity of Block Ciphers”. *IEEE Transactions on Information Theory*, vol. 53-11, Nov 2007, pp. 3944-3957.

[FAH99] Fahn, P.; Pearson, P. “IPA: A New Class of Power Attacks”. In: 1st International Workshop on Cryptographic Hardware and Embedded Systems (CHES’99), Aug 1999, pp. 173-186.

[GAN01] Gandolfi, K.; Moutel, C.; Olivier, F. “Electromagnetic Analysis: Concrete Results”. In: Third International Workshop on Cryptographic Hardware and Embedded Systems (CHES’01), May 2001, pp. 251-261.

[GEB05] Gebotys, C.; Tiu, C.; Chen, X. “A Countermeasure for EM Attacks of a Wireless PDA”. In: International Conference on Information Technology: Coding and Computing (ITCC’05), Apr 2005, pp. 544-549.

[GHE08] Ghellar, F.; Lubaszewski, M. “A Novel AES Cryptographic Core Highly Resistant to Differential Power Analysis”. In: 21st Symposium on Integrated Circuits and Systems Design (SBCCI’08), Sep 2008, pp. 140-145.

[GHO07] Ghosh, S.; Alam, M.; Kumar, K.; Mukhopadhyay, D.; Chowdhury, D. “Preventing the Side-Channel Leakage of Masked AES S-box”. In: International Conference on Advanced Computing and Communications (ADCOM’07), Dec 2007, pp. 18-21.

[GOL02] Golic, J.; Tymen, C. “Multiplicative Masking and Power Analysis of AES”. In: 4th International Workshop on Cryptographic Hardware and Embedded Systems (CHES’02), Aug 2002, pp. 198-212.

[GOL04] Golic, J.; Menicocci, R. “Universal Masking on Logic Gate Level”. *IET Electronics Letters*, vol. 40-9, Apr 2004, pp. 526-527.

[GOL07] Golic, J. “Techniques for RandomMasking in Hardware”. *IEEE Transactions on Circuits and Systems*, vol. 54-2, Feb 2007, pp. 291-300.

[GOO08] Goodwin, J.; Wilson, P. “Advanced Encryption Standard (AES) Implementation with Increased DPA Resistance and Low Overhead”. In: International Symposium on Circuits and Systems (ISCAS’08), May 2008, pp. 3286-3289.

[GOU03] Goubin, L. “A Refined Power-Analysis Attack on Elliptic Curve Cryptosystems”. In: 6th International Workshop on Theory and Practice in Public Key Cryptography (PKC’03), Jan 2003, pp. 199-211.

[GUI08] Guilley, S.; Sauvage, L.; Danger, J.; Graba, T.; Mathieu, Y. “Evaluation of Power-Constant Dual-Rail Logic as a Protection of Cryptographic Applications in FPGAs”. In: 2nd International Conference on Secure System Integration and Reality Improvement (SSIRI’08), Jul 2008, pp. 19-23.

[GUI08a] Guilley, S.; Sauvage, L.; Danger, J.; Hoogvorst, P. “Area Optimization of Cryptographic Co-Processors Implemented in Dual-rail with Precharge Positive Logic”. In: International Conference on Field Programmable Logic and Applications (FPL’08), Sep 2008, pp. 161-166.

[GUI08b] Guilley, S.; Chaudhuri, S.; Sauvage, L. ; Graba, T. ; Danger, J. ; Hoogvorst, P. ; Vong, V. ; Nassar, M. “Place and Route Impact on the Security of DPL Design in FPGAs”. In: IEEE International Workshop on Hardware-Oriented Security and Trust (HOST’08), Jun 2008, pp. 26-32.

[GUR06] Gürkaynak, F.; Oetiker, S.; Kaeslin, H.; Felber, N.; Fichtner, W. “Design Challenges for a Differential-Power-Analysis Aware GALS-based AES Crypto ASIC”. *Electronic Notes in Theoretical Computer Science*, vol. 146-2, Jan 2006, pp. 133-149.

[HAI07] Haijun, L.; Tao, Z.; Huiling, L.; Qiang, S. “A New Method Against High-Order Differential Power Analysis”. In: International Conference on Wireless Communications, Networking and Mobile Computing (WiCom’07), Sep 2007, pp. 2188-2191.

[HAN10] Han, X.; Cao, X.; Lloyd, E.; Shen, C. “Fault-Tolerante Relay Node Placement in Heterogeneous Wireless Sensor Networks”. *IEEE Transactions on Mobile Computing*, vol. 9-5, May 2010, pp. 643-656.

[HEA09] Healy, D. “Understanding Linear Feedback Shift Registers – The Easy Way”. Capturado em: http://www.yikes.com/~ptolemy/lfsr_web/index.htm, April 2009.

[HEN05] Hennessy, J.; Patterson, D. “Organização e Projeto de Computadores: a interface hardware/software”. Elsevier, 3º edição, Rio de Janeiro, 2005, 484p.

[HUE04] Hüebner, M.; Becker, T.; Becker, J. “Real-Time LUT-Based Network Topologies for Dynamic and Partial FPGA Self-Reconfiguration”. In: 17th Symposium on Integrated Circuits and Systems (SBCCI’04), Sep 2004, pp. 28-32.

[HO01] Ho, R.; May, K.; Horowitz, M. “The Future of Wires”. *Proceedings of the IEEE,* vol. 89-4, Apr 2001, pp. 490-504.

[HUI07] Huiping, J.; Rui, X.; Sheng, B. “Advanced DES Algorithm against Differential Power Analysis and its Hardware Implementation”. In: 1st International Symposium on Data, Privancy and E-Commerce, Nov 2007, pp. 316-320.

[ITR05] ITRS. “International Technology Roadmap for Semiconductors, 2005 Edition”. Capturado em: http://www.itrs.net/Links/2005ITRS/Home2005.htm, Dezembro 2009.

[ITR09] ITRS. “International Technology Roadmap for Semiconductors, 2009 Edition”. Capturado em: http://www.itrs.net/Links/2009ITRS/Home2009.htm, Janeiro 2009.

[JAK07] Jakimoski, G.; Subbalakshmi, K. “Discrete Lyapunov Exponent and Differential Cryptanalysis”. *IEEE Transactions on Circuits and Systems – II Express Briefs*, vol. 54-6, Jun 2007, pp. 499-501.

[JAV10] Javaid, H.; Janapsatya, A.; Haque, M.; Parameswaran, S. “Rapid Runtime Estimation Methods for Pipelined MPSoCs”. In: Design, Automation and Test in Europe Conference and Exhibition (DATE’10), Mar 2010, pp. 363-368.

[JIE06] Jie, C.; Yongzhuang, W.; Yupu, H. “A New Method for Impossible Differential Cryptanalysis of 7-round Advanced Encryption Standard”. In: International Conference on Communications, Circuits and Systems (ICCCAS’06), Jun 2006, pp. 1577-1579.

[KAM09] Kamoun, N.; Bossuet, L.; Ghazel, A. “Correlated Power Noise Generator as a Low Cost DPA Countermeasures to Secure Hardware AES Cipher”. In: International Conference on Signals, Circuits and Systems (ICCS’09), Oct 2009, pp. 1-6.

[KHA08] Khatir, M.; Moradi, A.; Ejlali, A.; Shalmani, M.; Salmasizadeh, M. “A Secure and Low-Energy Logic Style Using Charge Recovery Approach”. In: International Symposium on Low Power Electronics and Design (ISLPED’08), Aug 2008, pp. 259-264.

[KIM03] Kim, N. S.; Austin, T.; Blaauw, D.; Mudge, T.; Flautner, K.; Hu, J. S.; Irwin, M. J.; Kandemir, M.; Narayanan, V. “Leakage Current: Moore’s Law Meets Static Power”. *IEEE Computer*, vol. 36-12, Dec 2003, pp. 68-75.

[KOC96] Kocher, P. “Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and others Systems”. In: 16th International Cryptology Conference on Advances in Cryptology (CRYPTO’96), Aug 1996, pp. 104-113.

[KOC99] Kocher, P.; Jaffe, J.; Jun, B. “Differential Power Analysis”. In: 19th International Cryptology Conference on Advances in Cryptology (CRYPTO’99), Aug 1999, pp. 388-397.

[KOC99a] Kocher, P.; Jaffe, J.; Jun, B. “Introduction to Differential Power Analysis and Related Attacks”. Technical Report, Cryptography Reserch, 1999, 5p.

[KRS07] Krstic, M.; Grass, E.; Gurkaynak, F.; Vivet, P. “Globally Asynchronous, Locally Synchronous Circuits: Overview and Outlook”. *IEEE Design & Test of Computers*, vol. 24-5, Sep-Oct 2007, pp. 430-441.

[KUL05] Kulikowski, K.; Su, M.; Smirnov, A.; Taubin, A.; Karpovsky, M.; MacDonald, D. “Delay Insensitive Encoding and Power Analysis: A Balancing Act”. In: 11th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC’05), Mar 2005, pp. 116-125.

[KUL08] Kulikowski, K.; Venkataraman, V.; Wang, Z.; Taubin, A.; Karpovsky, K. “Asynchronous Balanced Gates Tolerant to Interconnect Variability”. In: International Symposium on Circuits and Systems (ISCAS’08), May 2008, pp. 3190-3193.

[LEU07] Leung, L.; Tsui, C. “Energy-Aware Synthesis of Network-on-Chip Implemented with Voltage Islands”. In: 44th Design Automation Conference (DAC '07), Jun 2007, pp. 128-131.

[LOM09] Lomne, V.; Maurine, P.; Torres, L.; Robert, M. Soares, R.; Calazans, N. “Evaluation on FPGA of Triple Track Logic Robustness against DPA and DEMA”. In: Design, Automation and Test in Europe Conference and Exhibition (DATE’09), Apr 2009, pp. 634-639.

[LU08] Lu, Y.; O’Neill, M.; McCanny, J. “FPGA Implementation and Analisys of Random Delay Insertion Countermeasure against DPA”. In: International Conference on Field-Programmable Technology (FTP’08), Dec 2008, pp. 201-208.

[MAH98] Maheshwari, N.; Sapatnekar, S. “Timing Analysis and Optimization of Sequential Circuits”. Springer, 1998, 208p.

[MAH09] Mahmud, R. “Techniques to Make Clock Switching Glitch Free”. Capturado em: http://www.eetimes.com/news/design/showArticle.jhtml?articleID=16501239, May 2009.

[MAI08] Maistri, P.; Leveugle, R. “Double-Data-Rate Computation as a Countermeasure against Fault Analysis”. *IEEE Transactions on Computers*, vol. 57-11, Nov 2008, pp. 1528-1539.

[MAN06] Mansoori, S.; Bizaki, H. “Linear Cryptanalysis on Second Round Simplified AES”. In: 8th International Conference on Advanced Communication Technology (ICACT’06), Feb 2006, pp. 1210-1214.

[MAR01] Martin, G.; Chang, H. “System-on-Chip Design”. In: 4th International Conference on ASIC (ASICON’01), Tutorial T.2, Oct 2001, pp. 12-17.

[MAR02] Martín-Langerwerf, J.; Reuter, C.; Kropp, H.; Pirsch, P. “Benefits of Macro-based Multi-FPGA Partitioning for Video Processing Applications”. In: 13th Workshop on Rapid Systems Prototyping (RSP’02), Jul 2002, pp. 60-65.

[MAR06] Martin, A.; Nystrom, M. “Asynchronous Techniques for System-on-Chip Design”. *Proceedings of the IEEE*, vol. 94-6, Jun 2006, pp. 1089-1120.

[MAT93] Matsui, M. “Linear Cryptoanalysis Method for DES Cypher”. In: Workshop on the Theory and Application of Cryptographic Techniques Advances in Cryptology (EUROCRYPT’93), May 1993, pp. 386-397.

[MEN10] Mentor Graphics Corporation. “ModelSim SE User’s Manual”. User Manual, v6.6b, 2010, 1544p.

[MES00] Messerges, T. “Using Second Order Power Analysis to Attack DPA Resistant Software”. In: 2nd International Workshop on Cryptographic Hardware and Embedded Systems (CHES’00), Aug 2000, pp. 238-251.

[MES05] Mesquita, D.; Techer, J.-D.; Torres, L.; Sassatelli, G.; Cambon, G.; Robert, M.; Moraes, F. “Current Mask Generation: A Transistor Level Security Against DPA Attacks”. In: 18th Symposium on Integrated Circuits and Systems Design (SBCCI’05), Sep 2005, pp. 115-120.

[MES06] Mesquita, D.; Badrignans, B.; Torres, L.; Sassatelli, G.; Robert, M.; Moraes, F.G. “A Leak Resistant SoC to Counteract Side Channel Attacks”. In: International Symposium on System-on-Chip (SoC’06), Nov 2006, pp. 1-4.

[MOL06] Möller, L.; Soares, R.; Carvalho, E.; Grehs, I.; Calazans, N.; Moraes, F. “Infrastructure for Dynamic Reconfigurable Systems: Choices and Trade-offs”. In: 19th Symposium on Integrated Circuits and Systems Design (SBCCI’06), Sep 2006, pp. 44-49.

[MOO65] Moore, G. “Cramming More Components Onto Integrated Circuits”. *Electronics*, vol. 38-8, Apr 1965, pp. 114-117.

[MOO02] Moore, S.; Anderson, R.; Cunningham, P.; Mullins, R.; Taylor, G. “Improving Smart Card Security using Self-Timed Circuits”. In: 8th International Symposium on Asynchronous Circuits and Systems (ASYNC’02), Apr 2002, pp. 211-218.

[MOR09] Morris, B.; Rogaway, P.; Stegers, T. “How to Encipher Messages on a Small Domain Deterministic Encryption and the Thorp Shuffle”. In: 29th Annual International Cryptographic Conference (CRYPTO’09), Aug 2009, pp. 286-302.

[MOR09a] Moradi, A.; Khatir, M.; Salmasizadeh, M.; Shalmani, M. “Charge Recovery Logic as a Side Channel Attack Countermeasure”. In: 10th International Symposium on Quality of Electronic Design (SQED), Mar 2009, pp. 686-691.

[MUR04] Muresan, R.; Gebotys, C. “Current Flattening in Software and Hardware for Security Applications”. In: International Conference on Hardware/Software Codesign and System Synthesis (CODES/ISSS’04), Sep 2004, pp. 218-223.

[MUR05] Muresan, R.; Vahedi, H.; Yang, Z.; Gregori, S. “Power Smart System On Chip Architecture for Embedded Cryptosystems”. In: International Conference on Hardware/Software Codesign and System Synthesis (CODES/ISSS’05), Sep 2005, pp. 184-189.

[MUR08] Muresan, R.; Gregori, S. “Protection Circuit against Differential Power Analysis Attacks for Smart Cards”. *IEEE Transaction on Computers*, vol. 57-11, Nov 2008, pp. 1540-1549.

[NAG07] Nagashima, S.; Homma, N.; Imai, Y.; Aoki, T.; Satoh, A. “DPA Using Phase-Based Waveform Matching against Random-Delay Countermeasure”. In: International Symposium on Circuits ans Systems (ISCAS’07), May 2007, pp. 1807-1810.

[NAO09] Naor, M.; Segev, G. “Public-Key Cryptosystems Resilient to Key Leakage”. In: 29th Annual International Cryptographic Conference (CRYPTO’09), Aug 2009, pp. 18-35.

[NIK99] Nikolaidis, S.; Chatzigeorgiou, A. “Analytical Estimation of Propagation Delay and Short-Circuit Power Dissipation in CMOS Gates”. *International Journal of Circuit Theory and Applications*, vol. 27-4, Sep 1999, pp. 375-392.

[NIS09] National Institute of Standards and Technology. “Computer Security Division, 2009 Annual Report”. Annual Report, 2009, 64 p.

[OGR07] Ogras, U.; Marculescu, R.; Choudhary, P.; Marculescu, D. “Voltage-Frequency Island Partitioning for GALS-based Network-on-Chip”. In: 44th Design Automation Conference (DAC '07), Jun 2007, pp. 110-115.

[ORD07] Ordu, L.; Örs, B. “Power Analysis Resistant Hardware Implementations of AES”. In: 14th IEEE International Conference on Eletronics, Circuits and Systems (ICECS’07), Dec 2007, pp. 1408-1411.

[ORD08] Ordas, T.; Lisart, M.; Sicard, E.; Maurine, P.; Torres, L. “Near-Field Mapping System to Scan in Time Domain the Magnetic Emissions of Integrated Circuits”. In: 18th International Workshop on Power and Timing Modeling Optimization and Simulation (PATMOS’08), Sep 2008, pp. 229-236.

[OSW05] Oswald, E.; Mangard, S.; Pramstaller, N.; Rijmen, V. “A Side-Channel Analysis Resistant Description of the AES S-box”. In: 12th International Workshop on Fast Software Encryption (FSE’05), Feb 2005, pp. 413-423.

[OSW06] Oswald, E.; Mangard, S.; Herbst, C.; Tillich, S. “Pratical Second Order DPA Attacks for Masked Smart Card Implementions of Block Ciphers”. In: Topics in Cryptology – Cryptographers Track at the RSA Conference (CT-RSA’06), Feb 2006, pp. 192-207.

[PEE05] Peeters, E.; Standaert, F.; Donckers, N.; Quisquater, J. “Improving High-Order Side Channel Attacks with FPGA Experiments”. In: 7th International Workshop on Cryptographic Hardware and Embedded Systems (CHES’05), Aug-Sep 2005, pp. 309-321.

[PON07] Pontes, J.; Soares, R.; Carvalho, E.; Moraes, F.; Calazans, N. “SCAFFI: An Intrachip FPGA Asynchronous Interface based on Hard Macros”. In: 25th IEEE International Conference on Computer Design (ICCD’07), May 2007, pp. 541-546.

[POP06] Popp, T.; Mangard, S. “Implementation Aspects of the DPA-Resistant Logic Style MDPL”. In: IEEE International Symposium on Circuits and Systems (ISCAS’06), May 2006, pp. 2913-2916.

[PRA04] Pramstaller, N.; Gürkaynak, F.; Haene, S.; Kaeslin, H.; Felber, H.; Fichtner, W. “Towards an AES Crypto-chip Resistent to Differential Power Analysis”. In: 30th European Solid-State Circuits Conference (ESSCIRC’04), Sep 2004, pp. 307-310.

[PRO09] Prouff, E.; Rivain, M.; Bévan, R. “Statistical Analysis of Second Order Differential Power Analysis”. *IEEE Transactions on Computers*, vol. 58-6, Jun 2009, pp. 799-811.

[RAB03] Rabaey, J. “Digital Integrated Circuits: A Design Perspective”. Upper Saddle River, 2nd Edition, Pearson Education, 2003, 761p.

[RAF10] Rafiev, A.; Murphy, J.; Yakovlev, A. “Secure Design Flow for Asynchronous Multi-Valued Logic Circuits”. In: 40th IEEE International Symposium on Multiple-Valued Logic (ISMVL’10), May 2010, pp. 264-269.

[RAM08] Rammohan, S.; Sundaresan, V.; Vemuri, R. “Reduced Complementary Dynamic and Differential Logic: A CMOS Logic Style for DPA-Resistant Secure IC Design”. In: 21st International Conference on VLSI Design (VLSID’08), Jan 2008, pp. 699-705.

[RAZ04] Razafindraibe, A.; Maurine, P.; Robert, M.; Bouesse, F.; Folco, B.; Renaudin, M. “Secured Structures for Secured Asynchronous QDI Circuits”. In: 19th International Conference on Design of Circuits and Integrated Systems (DCIS’04), Nov 2004, pp. 20-26.

[RAZ06] Razafindraibe, A.; Robert, M.; Maurine, P. “Formal Evaluation of the Robustness of Dual-Rail Logic against DPA Attacks”. In: 16th International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS’06), Sep 2006, pp. 634-644.

[RAZ07] Razafindraibe, A.; Robert, M.; Maurine, P. “Improvement of Dual Rail Logic as a Countermeasure against DPA”. In: IFIP International Conference on Very Large Scale Integration (VLSI-SoC’07), Oct 2007, pp. 270-275.

[REG07] Regazzoni, F.; Eisenbarth, T.; Großschädl, J.; Breveglieri, L.; Ienne, P.; Koren, I.; Paar, C. “Power Attacks Resistance of Cryptographic S-boxes with Added Error Detection Circuits”. In: 22nd IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT VLSI’07), Sep 2007, pp. 508-516.

[RIV78] Rivest, R.; Shamir, A.; Adleman, L. “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”. *Communications of the ACM*, vol. 21-2, Feb 1978, pp. 120-126.

[ROU03] Rouvroy, G.; Standaert, F.; Quisquater, J.; Legat, J. “Efficient Uses of FPGAs for Implementations of DES and Its Experimental Linear Cryptanalysis”. *IEEE Transactions on Computers*, vol. 52-4, Apr 2003, pp. 473-482.

[SCH96] Schneier, B. “Applied Cryptography – Protocols, Algorithms and Source Code in C”. John Wiley & Sons, 1996, 2nd Edition, 784 p.

[SEL09] Selimis, G.; Fournaris, A.; Kostopoulos, G.; Koufopavlou, O. “Software and Hardware Issues in Smart Card Technology”. *IEEE Communications Surveys & Tutorials*, vol. 11-3, Third Quarter 2009, pp. 143-152.

[SKO02] Skorobogatov, S.; Anderson, R. “Optical Fault Induction Attacks”. In: 4th International Workshop on Cryptographic Hardware and Embedded Systems (CHES’02), Aug 2002, pp. 2-12.

[SOA08] Soares, R.; Lomné, V.; Calazans, N.; Maurine, P.; Torres, L.; Robert, M. “Evaluating the Robustness of Secure Triple Track Logic through Prototyping”. In: 21st Symposium on Integrated Circuits and Systems Design (SBCCI’08), Sep 2008, pp. 193-198.

[SOT02] Sotiriou, C. “Implementing Asynchronous Circuits using a Conventional EDA Tool-Flow”. In: 39th Design Automation Conference (DAC'02), Jun 2002, pp. 415-418.

[SPA02] Sparsø, J.; Furber, S. “Principles of Asynchronous Circuits Design – A System Perspective”. Kluwer Academic Publishers, 2002, 360p.

[STA04] Standaert, F.; Örs, S.; Preneel, B. “Power Analysis of an FPGA Implementation of Rijndael: Is Pipelining a DPA Countermeasure?” In: 6th International Workshop on Cryptographic Hardware and Embedded Systems (CHES’04), Aug 2004, pp. 30-44.

[SZE02] Sze, S. “Semiconductors Devices: Physics and Technology”. John Wiley & Sons, 2002, 2nd Edition, 568p.

[TEE07] Teehan, P.; Greenstreet, M.; Lemieux, G. “A Survey and Taxonomy of GALS Design Styles”. *IEEE Design & Test of Computers*, vol. 24-5, Sep-Oct 2007, pp. 418-428.

[TIR02] 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”. In: 28th European Solid-State Circuits Conference (ESSCIRC’02), Sep 2002, pp. 403-406.

[TIR04] Tiri, K.; Verbauwhede, I. “A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation”. In: Design, Automation and Test in Europe Conference and Exhibition (DATE’04), Feb 2004, pp. 246-251.

[TIR06] Tiri, K.; Verbauwhede, I. “A Digital Design Flow for Secure Integrated Circuits”. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol. 25-7, Jul 2006, pp. 1197-1208.

[TOR02] Torres, G. “Fundamentos de Eletrônica”. Axcel Books, 2002, 244p.

[TRI03] Trichina, E.; De Seta, D.; Germani, L. “Simplified Adaptative Multiplicative Masking for AES and its Secure Implementation”. In: 5th International Workshop on Cryptographic Hardware and Embedded Systems (CHES’03), Sep 2003, pp. 187-197.

[VAH06] Vahedi, H.; Muresan, R.; Gregori, S. “On-chip Current Flattening Circuit with Dynamic Voltage Scaling”. In: IEEE International Symposium on Circuits and Systems (ISCAS’06), May 2006, pp. 4277-4280.

[VAH08] Vahedi, H.; Gregori, S.; Muresan, R. “Improved Current Flattening Circuit using Current Injection, Voltage Regulation, and Frequency Switching”. In: Canadian Conference on Electrical and Computer Engineering (CCECE’08), May 2008, pp. 2061-2066.

[VAN08] Vangal, S.; Howard, J.; Ruhl, G.; Dighe, S.; Wilson, H.; Tschanz, J.; Finan, D.; Singh, A.; Jacob, T.; Jain, S.; Erraguntla, V.; Roberts, C.; Hoskote, Y.; Borkar, Nitin.; Borkar, S. “An 80-Tile Sub-100-W TeraFLOPS Processor in 65-nm CMOS”. *IEEE Journal of Solid State Circuits*, vol. 43-1, Jan 2008, pp. 29-41.

[WAD04] Waddle, J.; Wagner, D. “Toward Efficient Second Order Power Analysis”. In: 6th International Workshop on Cryptographic Hardware and Embedded Systems (CHES’04), Aug 2004, pp. 1-15.

[WES94] Weste, N.; Eshraghian, K. “Principles of CMOS VLSI Design”. Addison-Wesley, 2nd edition, 1994, 735p.

[XIL05] Xilinx, Inc. “Spartan-3 Starter Kit Board User Guide”. User Guide (UG130), v1.1, May 2005, 64p.

[XIL07] Xilinx, Inc. “Virtex-II Platform FPGAs: Complete Data Sheet”. Data Sheet (DS031), v3.5, Nov 2007, 318p.

[XIL09] Xilinx, Inc. “Spartan-3 FPGA Family Data Sheet”. Product Specification (DS099), Dec 2009, 217p.

[XIL10] Xilinx, Inc. “Synthesis and Simulation Design Guide”. Xilinx User Guide (UG626), v12.1, Apr 2010, 180p.

[YAN05] Yang, S.; Wolf, W.; Vijaykrishnan, N.; Serpanos, D.N.; Xie, Y. “Power Attack Resistant Cryptosystem Design: A Dynamic Voltage and Frequency Switching Approach”. In: Design, Automation and Test in Europe Conference and Exhibition (DATE’05), Mar 2005, pp. 64-69.

[YEN06] Yen, C.; Wu, B. “Simple Error Detection Methods for Hardware Implementation of Advanced Encryption Standard”. *IEEE Transactions on Computers*, vol. 55-6, Jun 2006, pp. 720-731.

[ZAF05] Zafar, Y.; Ahmad, M. “Adaptative On-chip Oscillator for FPGA based Synchronous Designs”. In: International Conference on Emerging Technologies (ICET’05), Sep 2005, pp. 295-300.

[ZAF08] Zafar, Y.; Har, D. “A Novel Countermeasure Enhancing Side Channel Immunity in FPGAs”. In: International Conference on Advanced in Electronics and Micro-electronics (ENICS’08), Sep-Oct 2008, pp. 132-137.

APÊNDICE A – TRAÇOS RESULTANTES DO PROCESSO DE MEDIÇÃO DAS ARQUITETURAS GALS

Neste Apêndice são apresentados os traços de potência consumida, tracos de radiação eletromagnética das arquiteturas propostas medidos durante as análises realizadas. O Apêndice também apresenta os traços hipóteses resultantes das análises DPA, DEMA e realizadas.

A.1 TRAÇOS MEDIDOS

Nesta Seção são apresentados os traços de consumo de potência e de radição eletromagnética obtidos com o sistema de medição do laboratório LIRMM.

A.1.1 CONSUMO DE POTÊNCIA: ARQUITETURAS PIPE-2

Traços de consumo de potência obtidos para as arquiteturas DES pipeline GALS com 2 estágios. Estes resultados foram obtidos usando a plataforma da Digilent Spartan-3 Board [XIL05] e prototipados no dispositivo XC3S200 [XIL09].

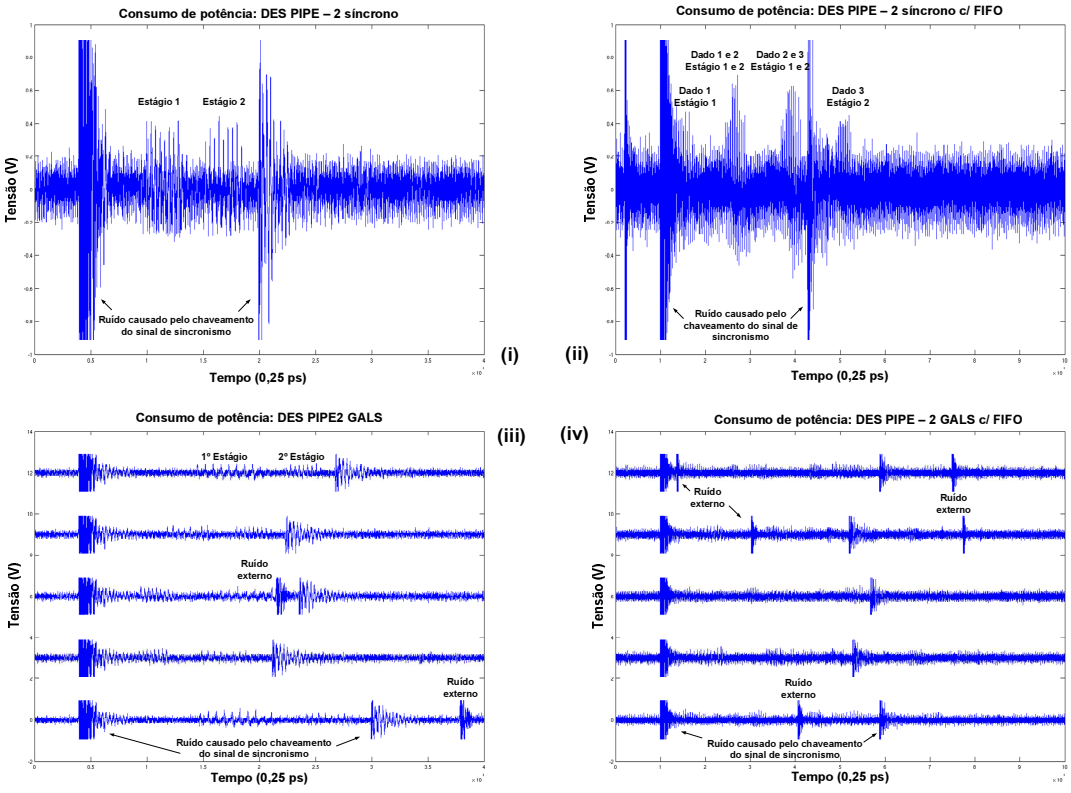


Figura 1 - Traços de consumo de potência das arquiteturas DES pipeline: (i) síncrono sem FIFO, (ii) síncrono com FIFO, (iii) GALS sem FIFO e (iv) GALS com FIFO.

A.1.2 RADIAÇÃO ELETROMAGNÉTICA: ARQUITETURA PIPE-2

Traços da radiação eletromagnética obtidos sobre as arquiteturas DES pipeline GALS com 2 estágios. Estes resultados foram obtidos usando a plataforma da Digilent Spartan-3 Board [XIL05] e prototipados no dispositivo XC3S200 [XIL09].

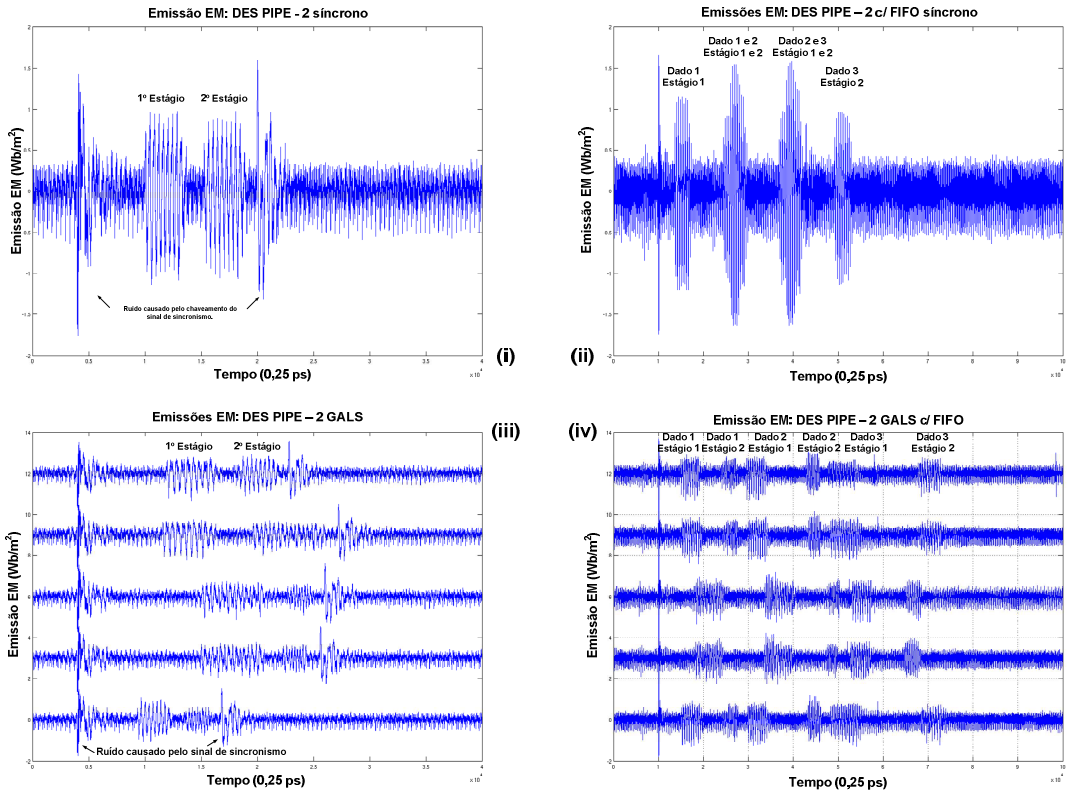


Figura 2 - Traços de radiação eletromagnética das arquiteturas DES pipeline: (i) síncrono sem FIFO, (ii) síncrono com FIFO, (iii) GALS sem FIFO e (iv) GALS com FIFO.

A.1.3 CONSUMO DE POTÊNCIA: ARQUITETURA PIPE-4

Traço de consumo de potência medido sobre a arquitetura DES pipeline GALS com 4 estágios. Estes resultados foram obtidos usando a plataforma da Digilent Spartan-3 Board [XIL05] e prototipados no dispositivo XC3S1000 [XIL09].

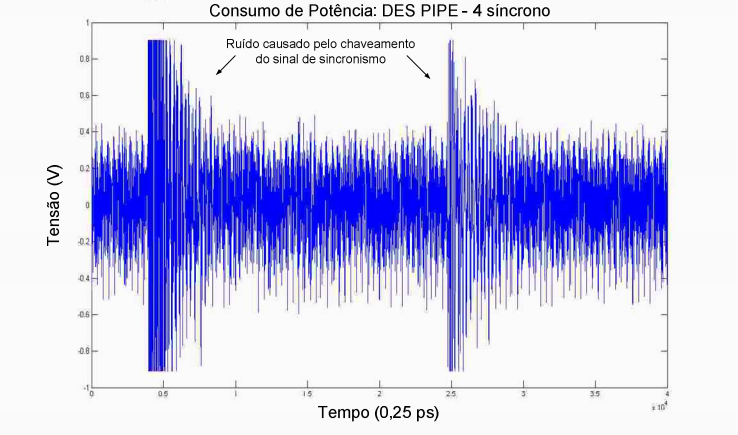


Figura 3 - Traço de consumo de potência obtido para a arquitetura DES pipeline com 4 estágios.

A.1.4 RADIAÇÃO ELETROMAGNÉTICA: ARQUITETURA PIPE-4

Traços de radiação eletromagnética das arquiteturas DES pipeline com 4 estágios. Cada estágio processa 4 rodadas do algoritmo. Os experimentos são todos realizados em um dispositivo XC3S1000 [XIL09].

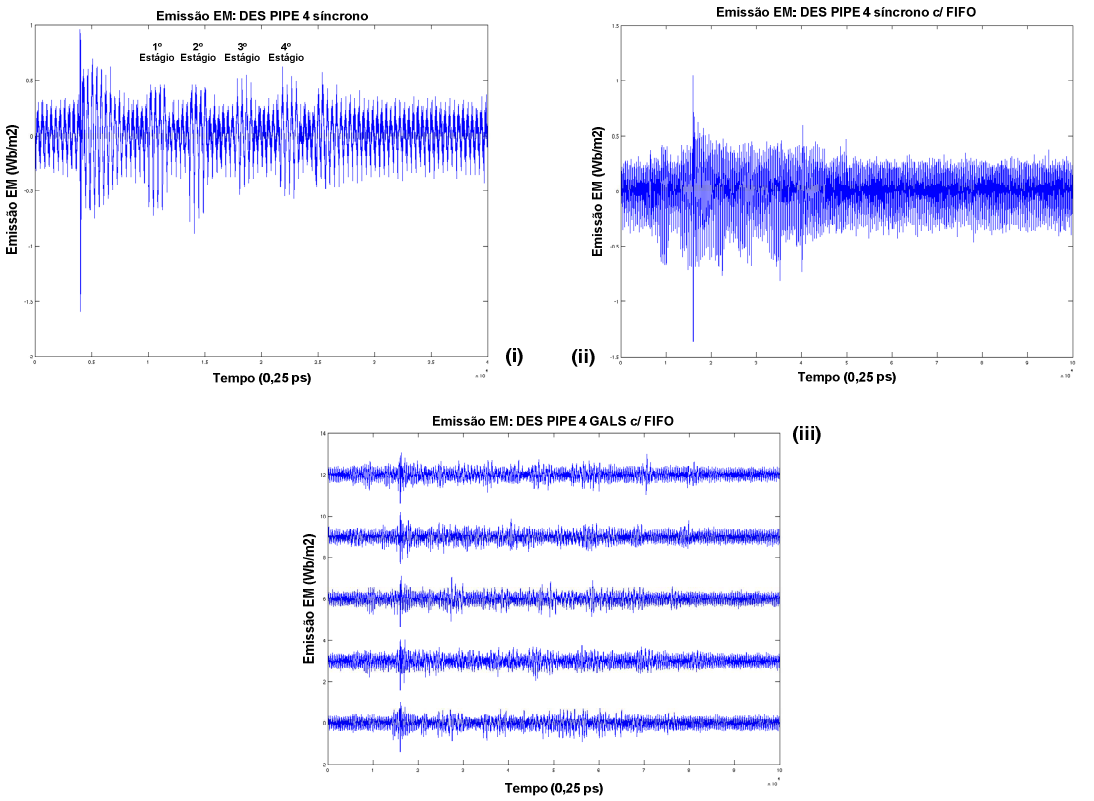


Figura 4 - Traços de radiação eletromagnética das arquiteturas DES pipelines com 4 estágios: (i) síncrono sem FIFO, (ii) síncrono com FIFO, e (iii) GALS com FIFO.

A.1.5 RADIAÇÃO ELETROMAGNÉTICA: ARQUITETURAS PIPE-8

Traços de radiação eletromagnética das arquiteturas DES pipeline com 8 estágios. Nestas arquiteturas apenas 2 rodadas do algoritmo são executadas em cada estágio, como visto na versão síncrona. Estes traços foram obtidos usando a plataforma da Digilent Spartan-3 Board [XIL05] e prototipados no dispositivo XC3S1000 [XIL09].

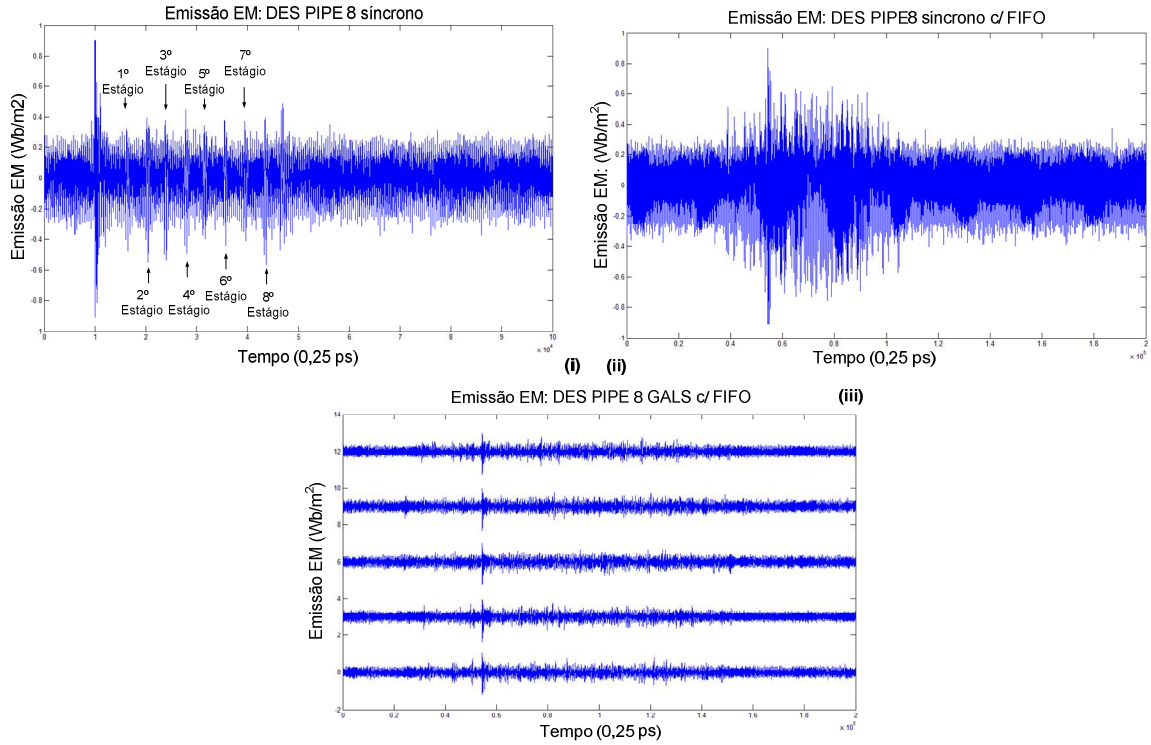


Figura 5 - Traços de radiação eletromagnética das arquiteturas DES pipelines com 8 estágios: (i) síncrono sem FIFO, (ii) síncrono com FIFO, e (iii) GALS com FIFO.

A.2 TRAÇOS HIPÓTESES DE SUBCHAVES

A.2.1 ARQUITETURAS PIPE-2: TRAÇOS RESULTANTES DAS ANÁLISES DEMA

Traços hipóteses resultantes das análises DEMA. Cada gráfico apresenta 64 hipóteses de traços de subchaves, onde os traços em preto correspondem à hipótese correta da subchave, os traços vermelhos à hipótese errada dada como correta pelas análises e os traços azuis são os demais traços. A Figura abaixo apresenta os resultados de ataques DEMA sobre a SBOX3 das diferentes versões da arquitetura PIPE-2: (i) síncrono sem FIFO, (ii) síncrono com FIFO, (iii) GALS sem FIFO e (iv) GALS com FIFO.

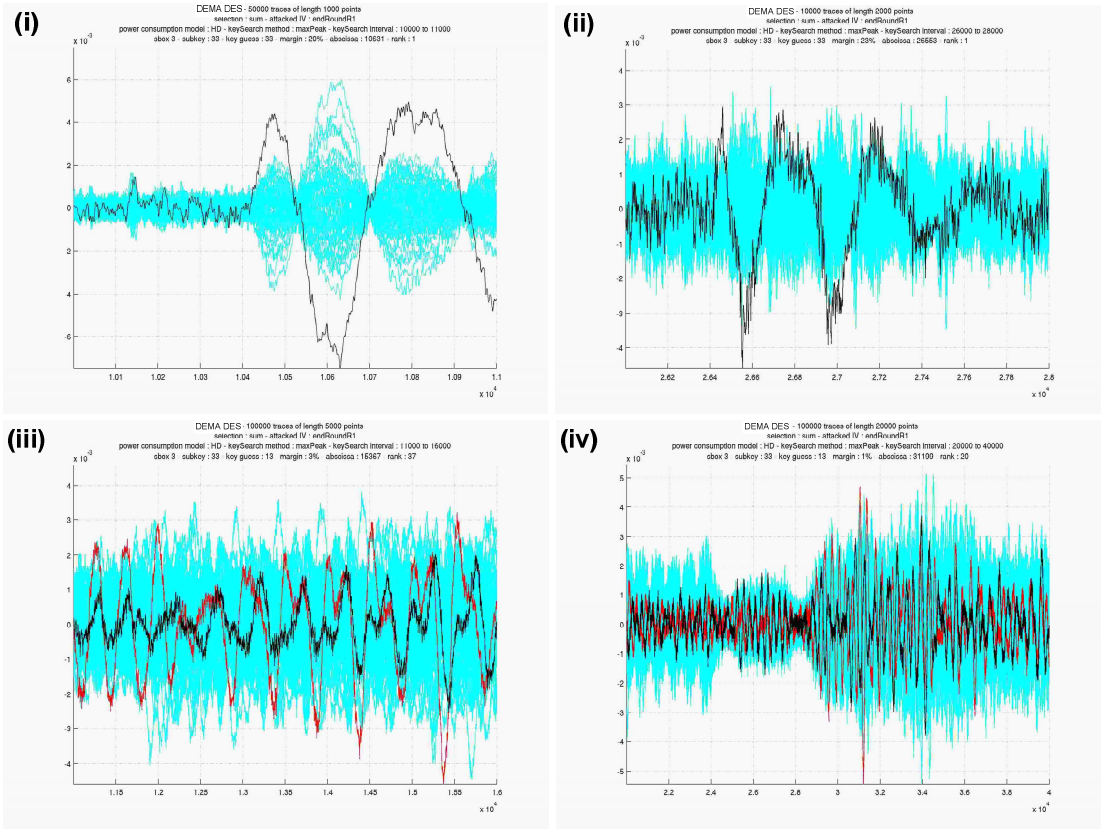


Figura 6 - Resultados das análises DEMA para a arquitetura DES PIPE-2: (i) síncrono sem FIFO, (ii) síncrono com FIFO, (iii) GALS sem FIFO e (iv) GALS com FIFO.

A.2.2 ARQUITETURAS PIPE-2: TRAÇOS RESULTANTES DAS ANÁLISES DPA

Nesta Seção são apresentados os traços resultantes das análises DPA. Os resultados mostram que as versões síncronas são vulneráveis aos ataques DPA. Já os ataques sobre as arquiteturas propostas se mostraram robustas a estes ataques, não havendo nenhum caso de sucesso dos ataques. Nesta Seção apenas os resultados de ataques realizados sobre a SBOX3 são apresentados.

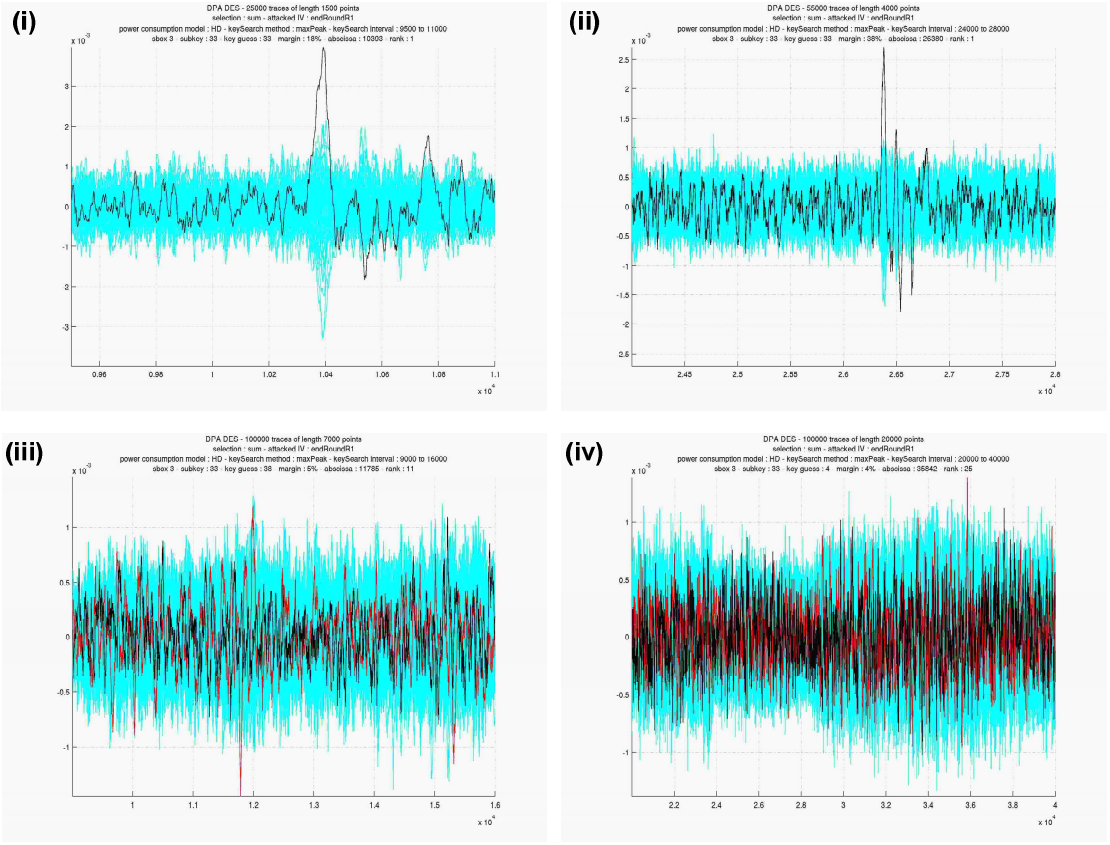


Figura 7 - Resultados das análises DPA para a arquitetura DES PIPE-2: (i) síncrono sem FIFO, (ii) síncrono com FIFO, (iii) GALS sem FIFO e (iv) GALS com FIFO.

A.2.3 ARQUITETURAS PIPE-4: TRAÇOS RESULTANTES DAS ANÁLISES DEMA

Nesta Seção são apresentados os traços hipóteses resultantes das análises DEMA sobre a SBOX3 das arquiteturas DES pipeline com 4 estágios. Os resultados demonstram que as versões síncronas são vulneráveis aos ataques DPA. Já os ataques sobre as arquiteturas GALS propostas se mostram robustas a estes ataques, não havendo nenhum caso de sucesso dos ataques.

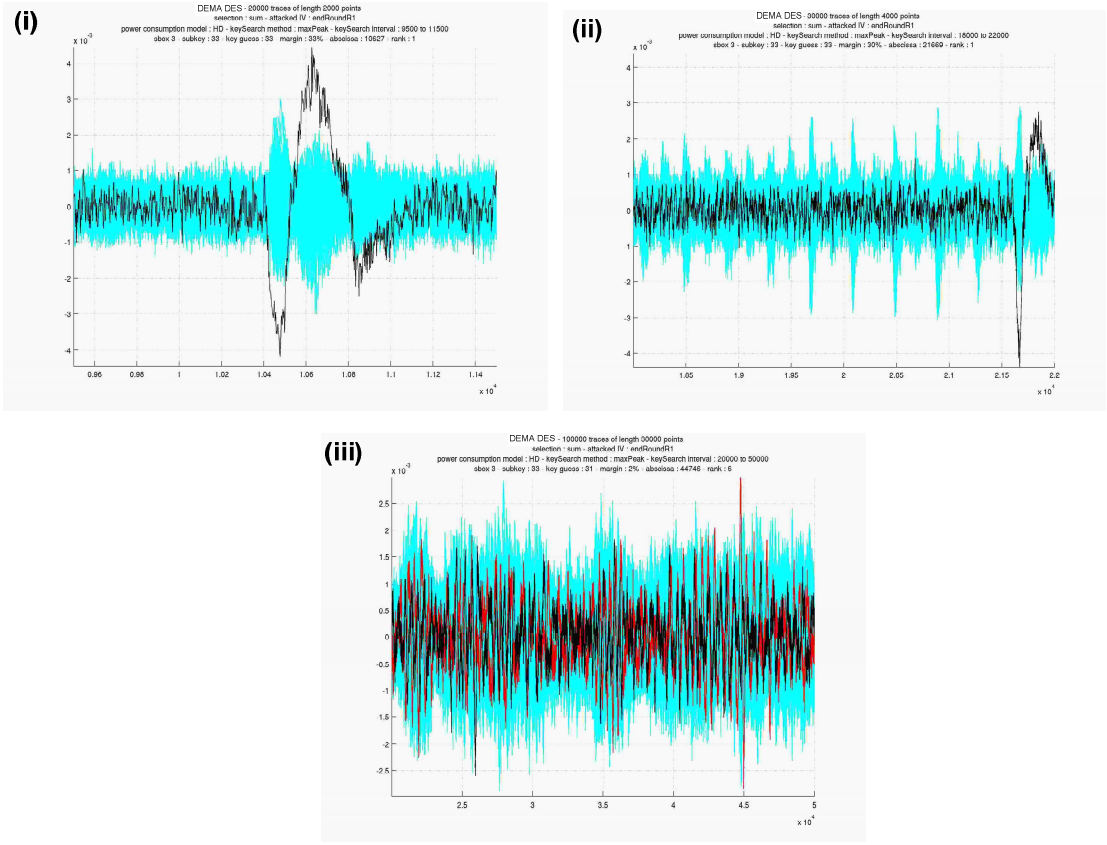


Figura 8 - Traços hipóteses obtidos com a realização dos ataques DEMA sobre as arquiteturas DES pipeline com 4 estágios: (i) síncrona sem FIFO, (ii) síncrona com FIFO e (iii) GALS com FIFO.

A.2.4 ARQUITETURAS PIPE-8: TRAÇOS RESULTANTES DAS ANÁLISES DEMA

Nesta Seção são apresentados os traços resultantes das análises DEMA sobre as arquiteturas DES GALS pipeline com 8 estágios. Os resultados mostram que as versões síncronas são vulneráveis aos ataques DPA. Já os ataques sobre as arquiteturas propostas se mostraram robustas a estes ataques, não havendo nenhum caso de sucesso dos ataques.

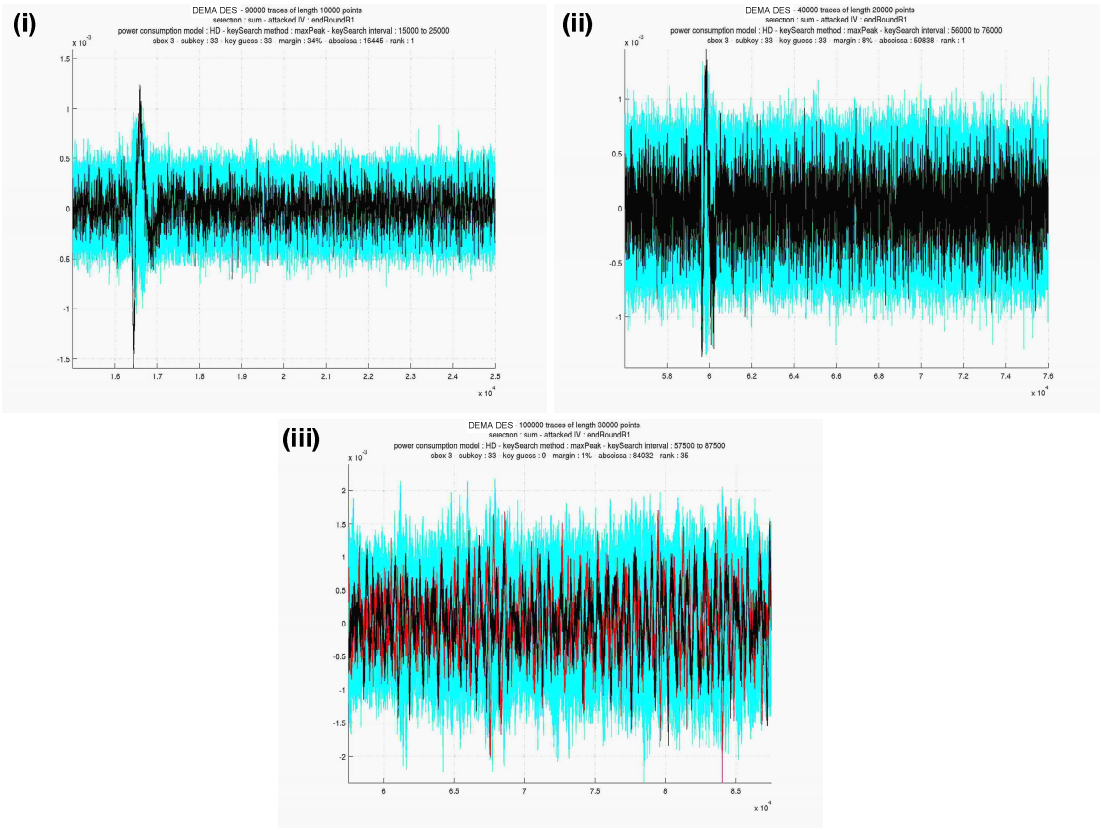


Figura 9 - Traços hipóteses obtidos com a realização dos ataques DEMA sobre as arquiteturas DES pipeline com 8 estágios: (i) síncrona sem FIFO, (ii) síncrona com FIFO e (iii) GALS com FIFO.

ANEXO B – TABELAS DO ALGORITMO DES

Tabelas de permutação, substituição e função do algoritmo DES. As tabelas devem ser lidas da esquerda para a direita e de cima para baixo. Por exemplo, a especificação da Permutação Inicial diz que o bit 58 de uma palavra de 64 bits deve ser movido para a posição 1, o bit 50 para a posição 2, o bit 42 para a posição 3 e assim por diante. Note que os vetores de 64 bits são numerados de 1 a 64 e não de 0 a 63 como se faz em muitos projetos de hardware.

Permutação Inicial

PI-1	58	50	42	34	26	18	10	2
	60	52	44	36	28	20	12	4
	62	54	46	38	30	22	14	6
	64	56	48	40	32	24	16	8
	57	49	41	33	25	17	9	1
	59	51	43	35	27	19	11	3
	61	53	45	37	29	21	13	5
	63	55	47	39	31	23	15	7

Permutação Inicial Inversa

PI-2	40	8	48	16	56	24	64	32
	39	7	47	15	55	23	63	31
	38	6	46	14	54	22	62	30
	37	5	45	13	53	21	61	29
	36	4	44	12	52	20	60	28
	35	3	43	11	51	19	59	27
	34	2	42	10	50	18	58	26
	33	1	41	9	49	17	57	25

Função de Expansão – E (Permutação de Expansão)

E	32	1	2	3	4	5
	4	5	6	7	8	9
	8	9	10	11	12	13
	12	13	14	15	16	17
	16	17	18	19	20	21
	20	21	22	23	24	25
	24	25	26	27	28	29
	28	29	30	31	32	1

Tabela de Substituição SBOX1

SBOX1	14	4	13	1	2	15	11	8
	3	10	6	12	5	9	0	7
	0	15	7	4	14	2	13	1
	10	6	12	11	9	5	3	8
	4	1	14	8	13	6	2	11
	15	12	9	7	3	10	5	0
	15	12	8	2	4	9	1	7
	5	11	3	14	10	0	6	13

Tabela de Substituição SBOX2

SBOX2	15	1	8	14	6	11	3	4
	9	7	2	13	12	0	5	10
	3	13	4	7	15	5	8	14

	12	0	1	10	6	9	11	5
	0	14	7	11	10	4	13	1
	5	8	12	6	9	3	2	15
	13	8	10	1	3	15	4	2
	11	6	7	12	0	5	14	9

Tabela de Substituição SBOX3

SBOX3	10	0	9	14	6	3	15	5
	1	13	12	7	11	4	2	8
	13	7	0	9	3	4	6	10
	2	8	5	14	12	11	15	1
	13	6	4	9	8	15	3	0
	11	1	2	12	5	10	14	7
	1	10	13	0	6	9	8	7
	4	15	14	3	11	5	2	12

Tabela de Substituição SBOX4

SBOX4	7	13	14	3	0	6	9	10
	1	2	8	5	11	12	4	15
	13	8	11	5	6	15	0	3
	4	7	2	12	1	10	14	9
	10	6	9	0	12	11	7	13
	15	1	3	14	5	2	8	4
	3	15	0	6	10	1	13	8
	9	4	5	11	12	7	2	14

Tabela de Substituição SBOX5

SBOX5	2	12	4	1	7	10	11	6
	8	5	3	15	13	0	14	9
	14	11	2	12	4	7	13	1
	5	0	15	10	3	9	8	6
	4	2	1	11	10	13	7	8
	15	9	12	5	6	3	0	14
	11	8	12	7	1	14	2	13
	6	15	0	9	10	4	5	3

Tabela de Substituição SBOX6

SBOX6	12	1	10	15	9	2	6	8
	0	13	3	4	14	7	5	11
	10	15	4	2	7	12	9	5
	6	1	13	14	0	11	3	8
	9	14	15	5	2	8	12	3
	7	0	4	10	1	13	11	6
	4	3	2	12	9	5	15	10
	11	14	1	7	6	0	8	13

Tabela de Substituição SBOX7

SBOX7	4	11	2	14	15	0	8	13
	3	12	9	7	5	10	6	1
	13	0	11	7	4	9	1	10
	14	3	5	12	2	15	8	6
	1	4	11	13	12	3	7	14
	10	15	6	8	0	5	9	2
	6	11	13	8	1	4	10	7
	9	5	0	15	14	2	3	12

Tabela de Substituição SBOX8

SBOX8	13	2	8	4	6	15	11	1
	10	9	3	14	5	0	12	7
	1	15	13	8	10	3	7	4
	12	5	6	11	0	14	9	2
	7	11	4	1	9	12	14	2
	0	6	10	13	15	3	5	8
	2	1	14	7	4	10	8	13
	15	12	9	0	3	5	6	11

Permutação P-Box

P	16	7	20	21	29	12	28	17
	1	15	23	26	5	18	31	10
	2	8	24	14	32	27	3	9
	19	13	30	6	22	11	4	25

Permutação da Chave - CP-1

CP-1	57	49	41	33	25	17	9	1	58	50	42	34	26	18
	10	2	59	51	43	35	27	19	11	3	60	52	44	36
	63	55	47	39	31	23	15	7	62	54	46	38	30	22
	14	6	61	53	45	37	29	21	13	5	28	20	12	4

Permutação de Compressão (da Chave) - CP-2

CP-2	14	17	11	24	1	5	3	28	15	6	21	10
	23	19	12	4	26	8	16	7	27	20	13	2
	41	52	31	37	47	55	30	40	51	45	33	48
	44	49	39	56	34	53	46	42	50	36	29	3211