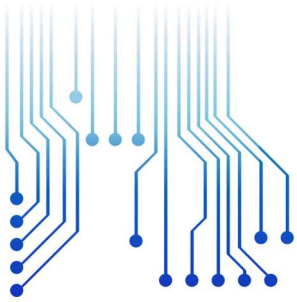
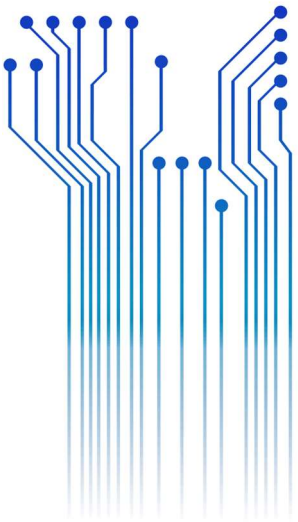


CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

RUBENS FERNANDES ROUX ABRANTES

RELATÓRIO DE ESTÁGIO SUPERVISIONADO
LABORATÓRIO EMBEDDED



Campina Grande
2018

RUBENS FERNANDES ROUX ABRANTES

RELATÓRIO DE ESTÁGIO SUPERVISIONADO
LABORATÓRIO EMBEDDED

Relatório de estágio supervisionado submetido à Coordenação do Curso de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Área de Concentração: Microeletrônica

Orientador:
Professor Gutemberg Gonçalves dos Santos Júnior, D.Sc.

Campina Grande
2018

RUBENS FERNANDES ROUX ABRANTES

RELATÓRIO DE ESTÁGIO SUPERVISIONADO
LABORATÓRIO EMBEDDED

Relatório de estágio supervisionado submetido à Coordenação do Curso de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Área de Concentração: Microeletrônica

Aprovado em / /

Professor Marcos Ricardo Alcântara Morais, D.Sc.
Universidade Federal de Campina Grande
Avaliador

Professor Gutemberg Gonçalves dos Santos Júnior, D.Sc.
Universidade Federal de Campina Grande
Orientador, UFCG

Dedico este trabalho a todos que
me apoiaram ao longo do curso.

AGRADECIMENTOS

Agradeço a minha mãe, Maria Sabina, por ter se esforçado tanto para me proporcionar uma boa educação, por ter me alimentado com saúde, força e coragem, as quais que foram essenciais para superação de todas as adversidades ao longo desta caminhada.

Agradeço também ao meu padrasto, Luciano Alberto, que me indicou o curso e ajudou a formar o caráter que tenho hoje.

Agradeço também a minha namorada, Emilly Melo, que me apoia desde o início do curso.

Ao meu orientador Prof. Gutemberg Gonçalves, por todas as oportunidades e orientações concedidas ao longo desses últimos anos.

A todos os membros do projeto de excelência em microeletrônica.

Agradeço também a toda minha família, que com todo carinho e apoio, não mediu esforços para eu chegar a esta etapa da minha vida.

Agradeço também ao *StackOverFlow* pois sempre que tive dúvidas ele foi capaz de saná-las sem qualquer dificuldade.

Enfim, agradeço a todos que de alguma forma, passaram pela minha vida e contribuíram para a construção de quem sou hoje.

“The Universe is under no obligation to make sense to you”

Neil deGrasse Tyson.

RESUMO

Este trabalho apresenta o relatório das atividades realizadas pelo aluno Rubens Fernandes Roux Abrantes durante o Estágio Supervisionado no Laboratório EMBEDDED localizado no Departamento de Engenharia Elétrica (DEE) da Universidade Federal de Campina Grande (UFCG), sob a orientação do Professor Gutemberg Gonçalves dos Santos Júnior e a supervisão do Professor Marcos Ricardo Alcântara Morais. O estágio consistiu em realizar testes em um *Field Programmable Gate Array* (FPGA) de periféricos do PULPino com a adição do módulo do AES previamente desenvolvidos no laboratório.

Palavras-chave: FPGA, PULPino, AES, testes.

ABSTRACT

This work presents a report of the activities developed by the student Rubens Fernandes Roux Abrantes during the Supervised Internship at the EMBEDDED Lab located in the Department of Electrical Engineering (DEE) of the Federal University of Campina Grande (UFCG), under the guidance of Professor Gutemberg Gonçalves dos Santos Júnior and supervision of Professor Marcos Ricardo Alcântara Morais. The internship consisted in performing PULPino peripherals' tests with the addition of the AES module previously developed in the laboratory on a Field Programmable Gate Array (FPGA).

Keywords: FPGA, PULPino, AES, tests.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Este relatório apresenta as atividades realizadas pelo aluno Rubens Fernandes Roux Abrantes durante o Estágio Supervisionado no Laboratório Embedded, sob a orientação do Professor Gutemberg Gonçalves dos Santos Júnior e a supervisão do Professor Marcos Ricardo Alcântara Morais. O estágio foi prestado entre o período de 14 de junho até 02 de agosto de 2018, com uma carga horária de 30 horas semanais, somando 210 horas totais.

System on chip (SoC) é um circuito integrado que abrange todos os componentes de um computador ou outros sistemas eletrônicos. Esses componentes incluem uma unidade central de processamento (CPU), memória, portas de entrada e saída, e armazenamento secundário - todos em um único substrato. SoCs são comuns no mercado de computação móvel devido ao baixo consumo de energia [5]. Estes são comumente aplicados na área de sistemas embarcados.

Existem três tipos distintos de SoCs: constituídos em torno de um microcontrolador, construídos em torno de um microprocessador; e especializados projetados para aplicações específicas que não se encaixam nas duas categorias anteriores. Neste estágio foi utilizado o PULPino que é um SoC projetado em torno de um microprocessador, que foi adaptado para a integração do AES, um *design* previamente elaborado em laboratório.

Este trabalho está organizado da seguinte forma: a seguir nesta seção, será comentado os objetivos deste trabalho e a metodologia aplicada; na seção 2 encontra-se uma fundamentação teórica acerca do tema; na seção 3 os procedimentos para realização dos testes, como foram realizados e os resultados obtidos com eles; e, finalmente nas seções 4 e 5, as considerações finais e referências utilizadas.

1.1 OBJETIVOS

1.1.1 OBJETIVO GERAL

Realizar testes em FPGA de periféricos do *System-on-Chip* (SoC) PULPino com adição em conjunto com o AES, previamente desenvolvido em laboratório.

1.1.2 OBJETIVOS ESPECÍFICOS

- Revisão bibliográfica acerca de implementações em FPGA do PULPino;
- Testes dos periféricos do PULPino;
- Testes do SoC com o AES;
- Testes dos periféricos em conjunto com o AES.

1.2 METODOLOGIA

A metodologia empregada neste trabalho envolveu a realização de pesquisa e atualização bibliográfica sobre o tema proposto.

Inicialmente foram realizados testes com os periféricos do PULPino, em seguida com o AES e por fim com os periféricos atuando em conjunto com o AES.

Os testes foram realizados na placa Altera DE1 da Terasic, programada a partir do *software* Intel® Quartus® II. Também foi utilizado o *software* Synplify Premier® da Synopsys® devido a possibilidade de otimização do uso de LUTs.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 FPGA

Uma FPGA é um circuito integrado que pode ser customizado para uma aplicação específica. Diferente de CPUs, FPGAs são "programáveis de campo", i.e., elas podem ser configuradas pelo usuário após a manufatura.

FPGAs contém blocos lógicos programáveis que podem ser ligados em diferentes configurações. Estes blocos criam um vetor físico de portas lógicas, a partir de *Look Up Tables* (LUTs), que permite que eles performem operações distintas. Como as portas lógicas são customizáveis, FPGAs podem ser otimizadas para qualquer operação computacional, com potencial de realizá-las de forma mais ágil do que uma CPU.

Estes processadores são tipicamente programados utilizando uma linguagem de descrição de hardware (HDL), *Verilog* ou *SystemVerilog*. Os comandos utilizados configuram as portas lógicas e como elas se conectam.

Como FPGAs são projetadas para serem programadas em aplicações específicas, elas não são adequadas para computadores pessoais, entretanto elas possuem uma extensa variedade de aplicações. Estas incluem telecomunicações, data centers, computação científica e processamento de áudio/vídeo. Além disso, são também utilizadas em servidores, computadores de *high-end*, dispositivos eletrônicos como TVs, rádios, e equipamentos médicos.

2.2 PULPINO

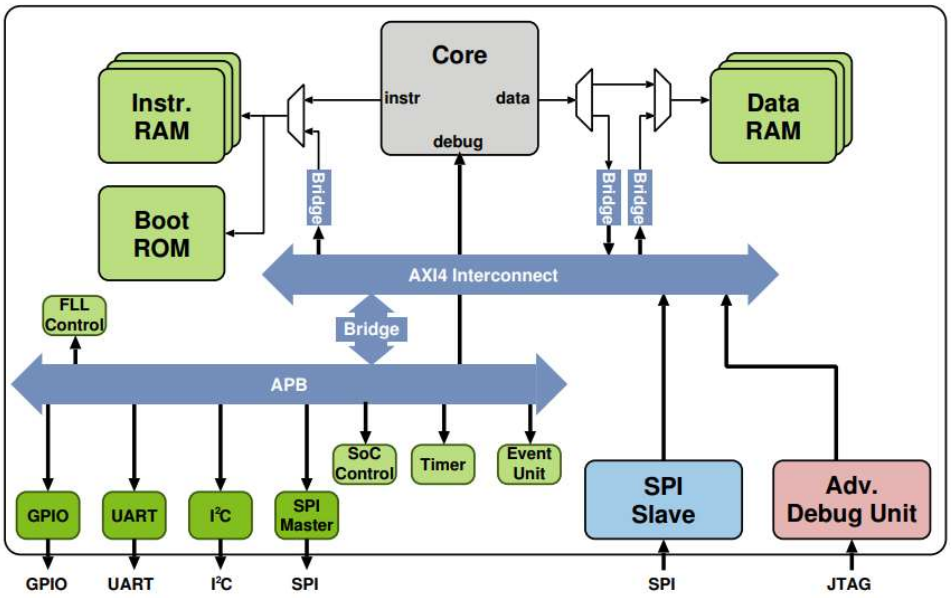
PULPino é o primeiro lançamento *open source* da plataforma de processamento *Parallel Ultra Low Power* (PULP), criada pela Universidade de Bologna e a ETH Zürich em 2013 com objetivo de explorar novas e eficientes arquiteturas para processamento de baixo consumo de energia.

Este é um SoC de um núcleo e reutiliza a maioria dos componentes do PULP e portas separadas para as RAMs de dados e instruções. Inclui uma ROM de boot que

contém informações de inicialização que pode carregar um programa via SPI por um dispositivo externo de flash [1].

O SoC utiliza a interface AXI4 como principal interconexão com uma ponte para a interface APB para os periféricos *GPIO*, *UART*, *I2C*, *SPI Master*, *SoC Control*, *Timer* e *Event Unit*. Ambas as interfaces utilizam um bus com canais de dados de 32 *bits* [1].

Figura 1 – PULPino.



Fonte: PULPino: *Datasheet*.

O PULPino é voltado principalmente para simulação RTL e ASICs, embora haja também uma versão para FPGA. As versões para FPGA não são otimizadas em termos de desempenho, pois são utilizadas como uma plataforma de emulação, em vez de uma plataforma *standalone* [1].

2.2.1 GPIO

O periférico de entrada e saída e propósito geral (GPIO) do PULPino possui nove registradores de 32 *bits*. Destacam-se os registradores *PADDIR*, *PADIN* e *PADOUT*. O primeiro controla a direção dos dados de cada um dos pinos de GPIO, o segundo é utilizado para os pinos de entrada e o último para os de saída.

2.2.2 UART

O módulo de *Universal Asynchronous Receiver-Transmitter* (UART) utilizado é compatível com 16750. Contém todos os sinais típicos de um UART além de adicionais definidos pelo 16750. A Tabela 1 apresenta estes sinais.

Tabela 1 – Sinais externos de UART.

Sinal	**Direção**	**Descrição**
uart_tx	saída	Transmite dados
uart_rx	entrada	Recebe dados
uart_rts	saída	Solicitar envio
uart_cts	entrada	Limpar para enviar
uart_dtr	saída	Terminal de dados pronto
uart_dsr	entrada	Grupo de dados pronto

Fonte: PULPino: *Datasheet*.

2.2.3 TIMER

O módulo do *timer* possui dois temporizadores e cada um deles possui três registradores: *TIMER*, *CTRL* e *CMP*. O primeiro armazena o valor do temporizador e quando chega em 0xFFFFFFFF uma interrupção acontece. O segundo controla tanto quando é inicializado quanto um *prescaler* que determina o intervalo no qual o temporizador é incrementado. O último determina um limite, menor que 0xFFFFFFFF, para comparar o valor do módulo para que ocorra uma interrupção.

2.2.4 I²C

I²C é um protocolo de sinalização *open-drain*, i.e., o driver de saída do bloco será ligado e desligado, sempre com um valor baixo quando ativado. Valores lógicos altos são obtidos usando um resistor *pull-up* nas linhas de SDA e SCL. A Tabela 2 apresenta os sinais de I²C.

Tabela 2 – Sinais de I²C.

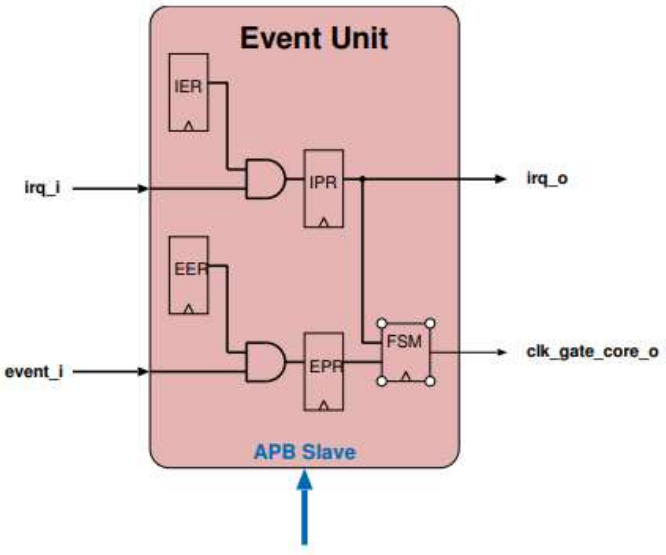
Sinal	Direção	Descrição
scl_pad_i	entrada	Entrada SCL
scl_pad_o	saída	Saída SCL (sempre 0)
scl_padoen_o	saída	Direção do bloco SCL
sda_pad_i	entrada	Entrada SDA
sda_pad_o	saída	Saída SDA (sempre 0)
sda_padoen_o	saída	Direção do bloco SDA
interrupt_o	saída	Evento/Interrupção

Fonte: PULPino: *Datasheet*.

2.2.5 EVENT UNIT

O PULPino tem uma unidade que suporta até 32 interrupções e eventos. As linhas de eventos e interrupções são armazenadas em buffer separadamente, como apresentado na Figura 2.

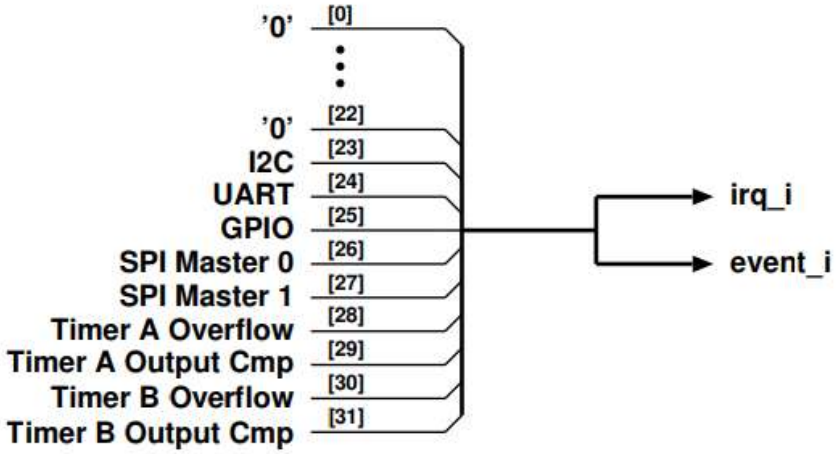
Figura 2 – Unidade de eventos.



Fonte: PULPino: *Datasheet*.

As interrupções e eventos são organizadas como apresentado na Figura 3. Vale destacar que *irq_i* e *event_i* possuem o mesmo valor.

Figura 3 – *Bits* de eventos.



Fonte: Adaptado de PULPino: *Datasheet*.

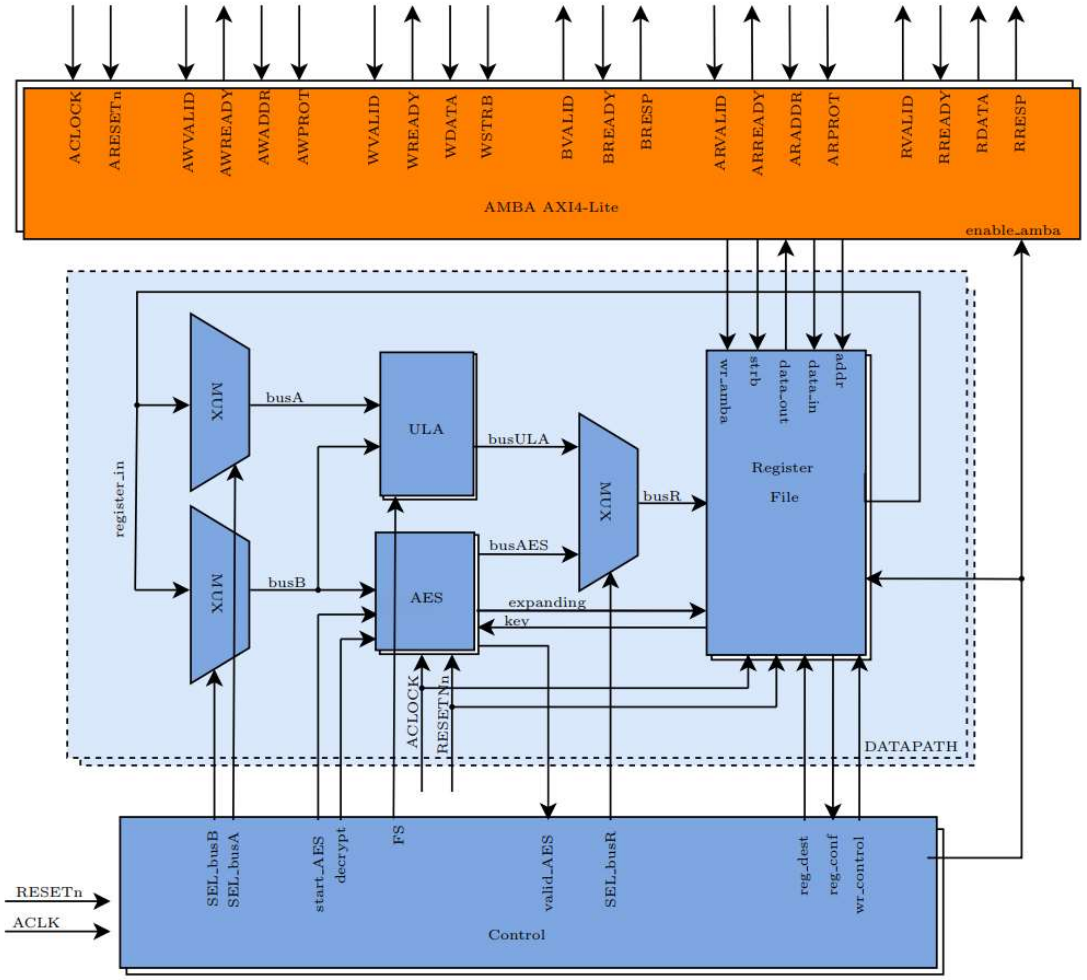
Este módulo possui dez registradores. Destaca-se os quatro de interrupções: *IER*, que habilita as instruções por bits; *IPR*, registrador de leitura e escrita de interrupções por linha; *ISP*, utilizado para estabelecer a interrupção no registrador *IPR*; e *ICP*, ao colocar um bit, a interrupção correspondente em *IPR* será zerada. Os quatro registradores de eventos: *EER*, *EPR*, *ESP* e *ECP* funcionam de maneira similar aos de interrupção.

2.3 AES

O Padrão de Encriptação Avançado (AES) é um algoritmo de criptografia destinado a compor sistemas de cifragem e decifragem simétrica, i.e., mesma chave para encriptar e decriptar. É uma cifra de bloco, ou seja, opera em blocos de tamanho fixo de 128 *bits* ou 16 *bytes*. Pode operar com chaves de 128, 192 ou 256 bits. Foi desenvolvido pelo governo dos Estados Unidos e anunciado pelo Instituto Nacional de Padrões e Tecnologia dos Estados Unidos (NIST) como U.S. FIPS PUB (FIPS 197).

Este módulo possui seis modos de operação e utiliza a interface AXI4-Lite e permite chaves de encriptação de 128 bits. São vinte registradores, sendo os quatro primeiros para a chave, seguido por quatro do bloco, quatro do vetor de inicialização, quatro do resultado, dois do contador, o registrador de comando e por fim o de status. A Figura 4 apresenta a arquitetura do IP.

Figura 4 – Arquitetura do IP AES.

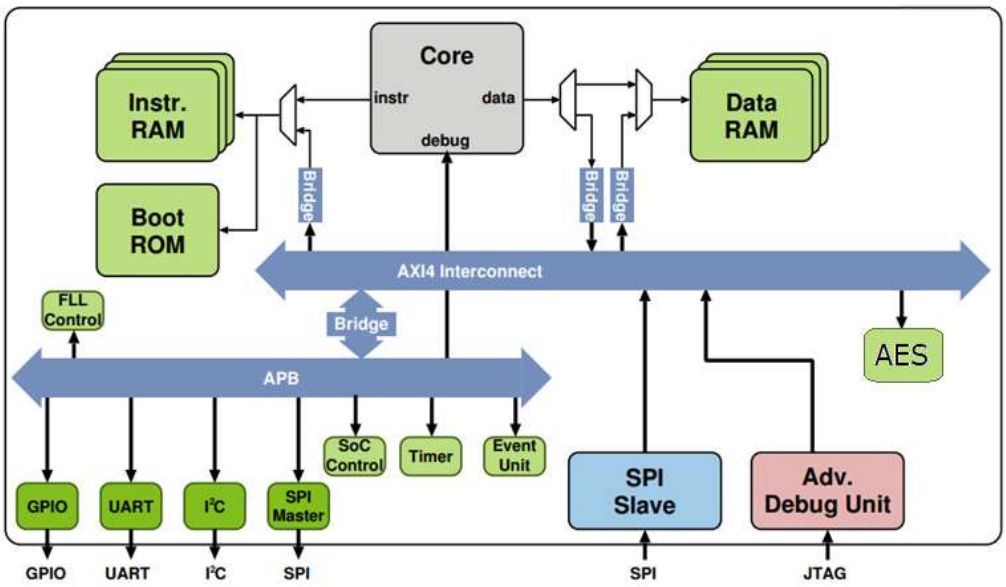


Fonte: O próprio autor.

3 TESTES REALIZADOS

Para possibilidade de realização dos testes, o AES foi introduzido à arquitetura do PULPino, apresentada na Figura 5.

Figura 5 – PULPino com acréscimo do AES.

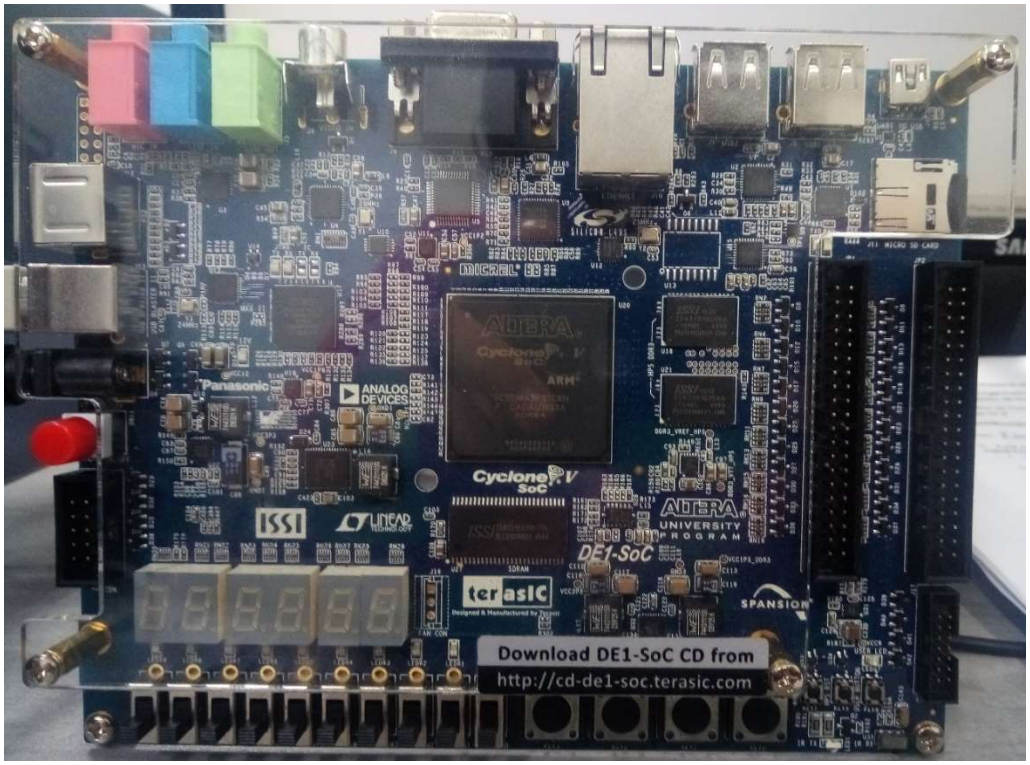


Fonte: Modificado de PULPino: *Datasheet*.

Todos os testes seguiram, em ordem, os seguintes procedimentos:

1. O código em C++ foi criado para programar o processador do PULPino.
2. Foi gerada uma ROM a partir do programa em C++ para programar a memória de instruções, procedimento deve ser substituído por uma leitura de memória externa para programar o SoC depois de inserido na FPGA.
3. O *software* Synplify Premier® foi utilizado para otimizar a quantidade de LUTs de forma que o SoC com o AES pudesse ser implementado na placa escolhida.
4. A placa Altera DE1, apresentada na Figura 6, foi programada com o *software* Intel® Quartus® II.
5. Por fim os resultados foram validados, seja por meio do Signal Tap II ou por componentes da placa (LEDs, chaves e displays).

Figura 6 – Placa de desenvolvimento Altera DE1.



Fonte: O próprio autor.

3.1 GPIO

O primeiro teste a ser realizado foi com o periférico de GPIO devido a possibilidade de utilização dos componentes da placa para validar os seguintes. Este tinha o simples objetivo de acender um LED com uma chave. Os procedimentos anteriormente apresentados foram seguidos e observou-se que os pinos de entrada e saída não são distintos, portanto um total de 32 pinos podem ser utilizados, seja para entrada ou saída. Ao fim deste, o funcionamento do módulo seguiu o padrão esperado e foi possível a programação do processador de forma que o LED dependesse da posição da chave.

Também foi realizado um teste para confirmar o funcionamento de todos os pinos do SoC. Este utilizou os displays de sete segmentos e os LEDs disponíveis na placa para apresentar um valor arbitrário introduzido no código em C++.

Por fim, concluiu-se que para utilizar o GPIO, deve-se indicar que os pinos serão utilizados com este propósito, indicar a direção (entrada ou saída) de cada um que for usado na aplicação no registrador *PADDIR*. No caso de direção de entrada, os dados serão armazenados, e é possível realizar a leitura dos mesmos, no registrador correspondente, *PADIN*, e os valores indicados na saída são apresentados no registrador *PADOUT*.

3.2 TIMER

Para testar o timer foram realizados três experimentos para cada um dos temporizadores (timerA e timerB): O primeiro consistiu em iniciar o temporizador e confirmar que está utilizando todos os 32 bits antes de reiniciar automaticamente, o segundo foi utilizado o registrador de comparação para limitar o valor máximo do timer, e o último modificou os valores do registrador de configuração para alterar a velocidade do contador.

No primeiro experimento, observou-se que ambos os temporizadores atuam de forma esperada. O segundo destacou que o temporizador reinicia ao chegar no valor de comparação. No último, não foram obtidos resultados esperados logo que independente do valor aplicado no registrador de controle para alterar o *prescale* o temporizador continuava atuando no mesmo *clock*.

Para utilizar o timer, deve-se inicializá-lo no registrador de controle, *CTRL*, caso necessário indicar um valor limite para a contagem no registrador *CMP*, e para obter o valor do contador, é realizada uma leitura do registrador *TIMER*.

3.3 INTERRUPÇÕES DO TIMER

As interrupções funcionam de forma diferente dos periféricos apresentados anteriormente, para testá-las é necessária a utilização de algum desses. Para este teste, o timer foi escolhido e com ele foram testadas todas as interrupções de cada um dos temporizadores (timerA e timerB).

Todos os testes de interrupção do timer utilizaram um pino de GPIO conectado a um LED para indicar caso ela seja chamada e outro para indicar que estava saindo desta. O primeiro utilizou a interrupção de *overflow* dos temporizadores e o segundo a de comparação. Verificou-se que as duas estavam funcionando da forma esperada.

Para utilizar qualquer interrupção, é necessária a indicação que aquela deve ser chamada com assistência do registrador *IER* e quando esta for requisitada, deve-se informar ao processador, por meio do registrador *ICP*, que a respectiva interrupção foi tratada e indicá-lo a voltar ao processo principal.

3.4 AES

Os testes realizados com o AES utilizaram o *Signal Tap II* para obter os dados determinados pelo SoC em cada um dos modos de operação e compará-los com os resultados de simulações em *software*.

Para realizá-los foram inseridos os dados da chave de encriptação, em seguida o bloco para ser encriptado, posteriormente foi indicado o vetor de inicialização e por fim o registrador de controle, indicando qual o modo de operação e para começar o funcionamento do IP. Estes mesmos procedimentos foram realizados para a decriptação e todos os resultados obtidos foram verificados pelas simulações em *software*.

3.5 AES COM INTERRUPÇÕES

Neste teste foi necessária a alteração do funcionamento do processador do PULPino, de forma a incluir uma nova interrupção no mesmo. Esta etapa consistiu em entender como as interrupções dos periféricos são declaradas e replicá-las para que o sinal de interrupção do AES seja interpretado pelo processador de maneira similar.

Após declarar a interrupção do AES, a mesma foi testada de forma análoga ao teste com a interrupção do timer. Primeiro foi indicado ao processador, no registrador *IER*, para atentar-se a uma interrupção do AES, e um pino de GPIO foi conectado a um LED para indicar caso esta esteja ocorrendo, e outro para confirmar a saída desta operação, informando ao processador, pelo registrador *ICP*, que a interrupção foi tratada.

3.6 AES COM DADOS DO TIMER

Já ciente do funcionamento adequado do AES e do timer, foi realizada um teste simples para confirmar a possibilidade de utilização simultânea de periféricos do PULPino com o AES. Para isso, foi utilizado um pino de GPIO de entrada ligado a uma chave para determinar um momento arbitrário em que o processador deve adquirir o valor do timer, e encriptá-lo com o AES. Para adquirir este valor, foi obtido o resultado da

encriptação com auxílio do *Signal Tap II* e este foi decriptado em simulação em *software* para compará-lo com o valor do timer, apresentado nos *displays* de sete segmentos da placa. Foi verificado que os valores obtidos pós decriptação é idêntico ao do temporizador, confirmando a possiblidade de funcionamento simultâneo do AES com os periféricos do PULPino.

4 CONSIDERAÇÕES FINAIS

Este estágio proporcionou um maior conhecimento de SoCs e do funcionamento de processadores, abortando desde a programação e adaptação do funcionamento do processador, até a alteração do SoC para adicionar o IP do AES.

As elaborações realizadas permitiram conhecer melhor as diversas ferramentas oferecidas pelo *Synplify Premier*®, e também entender o funcionamento do *Signal Tap II*.

Com os resultados obtidos, foi possível entender o funcionamento de periféricos do processador do PULPino e abriu a possibilidade de investigação do motivo de não funcionamento do *prescaler* do Timer. Ainda se confirmou a possibilidade de introdução de IPs ao SoC.

A realização de testes com os periféricos que não foram abordados assim como a criação de um SoC com outros IPs são sugestões para trabalhos futuros. Assim como a implementação de uma leitura de um dispositivo de flash externo para preencher a memória de instruções, permitindo a programação do processador sem a necessidade de reprogramar a FPGA.

5 REFERÊNCIAS BIBLIOGRÁFICAS

[1] Gautschi, M., Traber, A. PULPino: Datasheet, Junho 2017. Disponível em: <https://pulp-platform.org//wp-content/uploads/2017/08/datasheet.pdf>

[2] Federal Information “Advanced Encryption Standard (AES)”, Processing Standards Publication 197, Novembro 2001. Disponível em: <https://csrc.nist.gov/csrc/media/publications/fips/197/final/documents/fips-197.pdf>.

[3] MANO, M.M. “Computer System Architecture Third Edition”, Pearson, 1992.

[4] “AMBA® AXI™ and ACE™ Protocol Specification”, 2011. Disponível em: <http://www.gstitt.ece.ufl.edu/courses/fall15/eel4720_5721/labs/refs/AXI4_specification.pdf>

[5] Bennett, P. "The why, where and what of low-power SoC design.", EE Times, Dezembro, 2004.