

Plataforma de Integração de Componentes para Sistemas Embarcados em Aplicações Espaciais

Carlos Natham Domingos
Universidade do Vale do Itajaí, Brasil
carlosnatham@edu.univali.br

Douglas Almeida dos Santos
Universidade do Vale do Itajaí, Brasil
Université de Montpellier, França
douglasas@edu.univali.br

Wesley Grignani
Universidade do Vale do Itajaí, Brasil
wesley.grignani@edu.univali.br

Douglas Rossi de Melo
Universidade do Vale do Itajaí, Brasil
drm@univali.br

ABSTRACT

Artificial satellite systems launched into space are designed for various activities, such as communication, space observation, and remote sensing. These satellites have consisted of a growing number of modules to perform a wide variety of tasks. Due to the limitations of onboard processors, there is an increasing need for structures that support the functions performed by the satellites. One approach is to use specific purpose processors to accelerate these functions. However, integrating accelerators with processors is a complex activity. Considering this, we propose the development of a platform that seeks to facilitate the creation of Systems-on-Chip with hardware and software components for use in space applications.

KEYWORDS

Sistemas Integrados, Aceleradores, RISC-V, Aplicações Espaciais

1 INTRODUÇÃO

A utilização de satélites artificiais tem aumentado cada vez mais no decorrer deste século, já que esses podem desempenhar muitas funções que auxiliam no dia a dia das pessoas, sendo em áreas da meteorologia, comunicação, navegação ou observação. Módulos de serviço e de carga útil são responsáveis pelo funcionamento desses satélites e pelos acoplamentos dos subsistemas relacionados à missão que será realizada [1].

Os módulos e subsistemas presentes em um satélite são desenvolvidos para realizar tarefas específicas relacionadas a sua missão, com diversos periféricos integrados, como processadores e memórias, com tamanho reduzido e menor custo de energia possível [2]. Com aplicações cada vez mais específicas, a demanda de velocidade da transmissão de dados cresce também. Dessa forma, componentes dedicados para executar tarefas específicas são utilizados com o objetivo de diminuir a carga operacional do processador principal, com maior desempenho e eficiência energética. Tais componentes são conhecidos como coprocessadores ou aceleradores em hardware [3].

Nesse contexto, percebe-se que a maioria dos projetos desenvolvidos para sistemas integrados focaram no provimento de processadores e aceleradores de maneira isolada. Dessa forma, neste trabalho foi desenvolvida uma plataforma que permite a fácil integração de módulos previamente descritos para elaboração de sistemas integrados com foco em aplicações espaciais.

2 FUNDAMENTAÇÃO TEÓRICA

Os sistemas embarcados são a classe de computadores que têm apresentado maior crescimento, presentes na maioria dos telefones, câmeras e outros dispositivos de uso comum [4]. Isso se deve pelas vantagens que o uso desses sistemas pode trazer, desde o tamanho reduzido até a eficiência energética. Tais vantagens são atreladas à principal característica do sistema, que é a funcionalidade, pois são construídos para realizarem procedimentos específicos [5].

A construção de um sistema embarcado depende de alguns fatores a serem considerados no projeto, já que a execução realiza somente as funções necessárias. Os processadores de uso específico, conhecidos como coprocessadores ou aceleradores, executam apenas uma aplicação específica e são empregados na construção de sistemas embarcados complexos [6].

Os sistemas integrados, ou SoCs (Systems-on-Chip), são projetos que contêm processadores, memórias e componentes de entrada/saída. Todos esses componentes estão presentes em um único pedaço de silício, e cada componente é conhecido como núcleo ou IP (Intellectual Property), projetados para serem reutilizáveis em diversos projetos. Esses blocos são interconectados para funcionarem de maneira conjunta no chip [7].

As interconexões dos blocos do sistema integrado são realizadas por meio de barramentos. Os barramentos realizam o compartilhamento das conexões físicas de endereços, dados e controle de sinais entre cada bloco, organizados geralmente de forma hierárquica [8].

Atualmente, o barramento de maior adoção na academia e na indústria é o AMBA (Advanced Microcontroller Bus Architecture), desenvolvido pela empresa ARM (Advanced RISC Machine), o qual possui uma arquitetura que emprega diferentes níveis de hierarquia dos protocolos [9].

3 TRABALHOS RELACIONADOS

Existem alguns projetos que fornecem ferramentas ou plataformas para a construção de sistemas integrados, com destaque para os trabalhos dos processadores LEON e NOEL-V com o GRLIB da Gaisler, o PULPino e o Rocket Chip.

O LEON é um processador SPARC 32 bits que está contido na biblioteca GRLIB, voltado para a implementação em FPGAs (Field Programmable Gate Array). Na plataforma, os núcleos são conectados por meio de um barramento AMBA-2.0, utilizando os padrões AHB e APB [10].

XIV Computer on the Beach

30 de Março a 01 de Abril de 2023, Florianópolis, SC, Brasil

Além do processador LEON, a Gaisler também disponibiliza o processador NOEL-V, um modelo VHDL baseado na arquitetura RISC-V. Além do barramento AMBA-2.0 AHB/APB, o NOEL-V oferece suporte ao AMBA-4.0 AXI [11].

O PULPino é um sistema integrado voltado para o uso com os processadores RI5CY e Ibex. É descrito na linguagem Verilog, voltado principalmente para simulação em RTL (Register Transfer Level) e ASIC (Application Specific Integrated Circuit), embora também exista uma versão FPGA. O PULPino utiliza o barramento AMBA-4.0, na qual suporta os padrões AXI e APB [12].

O Rocket Chip é um gerador de sistemas que usa a linguagem Chisel, capaz de produzir muitas instâncias de um projeto a partir de uma única fonte de código de alto nível. Utiliza o barramento AMBA-3.0, com suporte aos padrões AXI e APB [13].

A Tabela 1 apresenta um comparativo dos trabalhos relacionados com o trabalho desenvolvido. A proposta de plataforma deste trabalho é baseada no uso do processador HARV (HArdened Risc-V) com o barramento AMBA AXI4-Lite para as interconexões. A linguagem utilizada para a descrição de hardware foi o VHDL, e o suporte ao software oferecido na plataforma é para o uso do FreeRTOS ou aplicação direta (*baremetal*). Para a construção do sistema foi utilizada a linguagem Python.

Tabela 1: Comparativo com os trabalhos relacionados

Plataforma	Processador	Descrição de hardware	Suporte a software	Construção de sistema
LEON/ GRLIB	LEON3FT LEON5FT (SPARC)	VHDL	Linux RTEMS VxWorks	TCL
NOEL-V/ GRLIB	NOEL-V (RISC-V)	VHDL	Linux RTEMS VxWorks	TCL
PULPino	RI5CY (RISC-V)	Verilog	FreeRTOS	Python
Rocket Chip	Rocket Core (RISC-V)	Chisel	Linux	Scala
Este trabalho	HARV (RISC-V)	VHDL	FreeRTOS	Python

4 SOLUÇÃO PROPOSTA

A Figura 1 apresenta uma visão geral da plataforma implementada. A ferramenta consiste em uma plataforma na qual é possível realizar configurações para a prototipação de sistemas integrados em diferentes cenários. Essas configurações também podem ser utilizadas para a verificação por simulação e testadas por prototipação.

Dentre as opções de configurações, a primeira permite que sejam configuradas características do processador, como, por exemplo, a aplicação de técnicas de tolerância a falhas. Na segunda opção, configuração do barramento, o usuário pode definir algumas opções no barramento AMBA AXI4-Lite utilizado. A próxima opção consiste na configuração dos periféricos a serem integrados no sistema. A etapa de configuração do software permite a escolha de uma aplicação direta ou de um sistema operacional. Por fim, a ferramenta possui a opção de incluir um acelerador externo e a opção de gerar o sistema integrado.

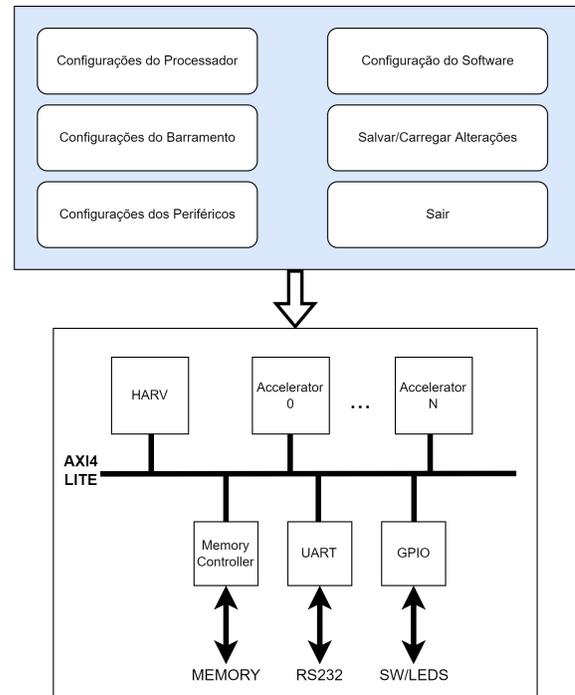


Figura 1: Visão geral da plataforma

4.1 Materiais

A etapa em alto nível do sistema foi programada utilizando a linguagem Python, com a biblioteca PyQt para a construção da interface. Já o desenvolvimento e a integração dos componentes em hardware utilizaram a linguagem de descrição VHDL. Para gerar esses componentes, o software Xilinx Vivado é utilizado para fornecer serviços de síntese lógica e física de hardware, usando os arquivos gerados pelos comandos em Python. Como dispositivo alvo da síntese é utilizado o FPGA Zynq-7000, presente no kit de desenvolvimento Zedboard.

Como processador principal é utilizado o HARV, desenvolvido por [14]. Atualmente, a plataforma integra o acelerador de imagens hiperespectrais, desenvolvido por [15]. A arquitetura AMBA AXI4-Lite foi utilizada para interconectar os periféricos. Para a configuração de software, foi escolhido o FreeRTOS como sistema operacional, além da execução *baremetal*. Todos esses componentes foram selecionados por terem sido desenvolvidos ou adaptados em projetos relacionados a este trabalho.

4.2 Elaboração do sistema integrado

Durante o procedimento de configuração do usuário, cada informação digitada ou marcada é salva em um arquivo de configuração. A interface gráfica desenvolvida apresenta as configurações possíveis somente para os componentes de hardware e software que foram utilizados neste trabalho, sendo necessária a implementação de novas interfaces para utilizar diferentes componentes. Após o usuário modificar todas as opções que desejar e clicar na opção de gerar VHDL, é iniciado o procedimento para gerar o sistema integrado, conforme ilustrado na Figura 2.

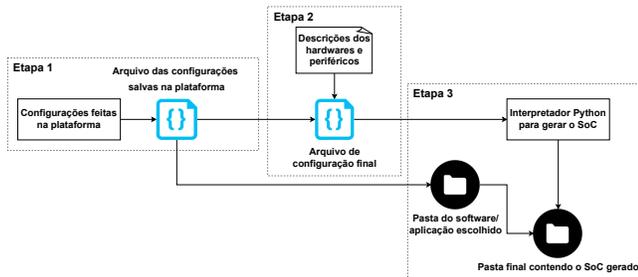


Figura 2: Diagrama de construção do sistema integrado

A primeira etapa é a geração do arquivo final de configuração, a partir das configurações selecionadas na plataforma. Após este procedimento, o arquivo gerado é utilizado por um gerador VHDL que irá identificar todas as informações e irá gerar a entidade topo do sistema integrado. Todo esse procedimento é realizado por scripts feitos na linguagem Python.

Após ser gerado o arquivo topo do sistema integrado, é criada uma pasta do projeto no caminho escolhido pelo usuário, com os diretórios necessários para a síntese e execução. Caso tenha sido marcada a opção para habilitar a aplicação/software, também é gerado um diretório com a aplicação selecionada.

Com a pasta do sistema integrado gerada é possível realizar a síntese ou simulação usando o software Xilinx Vivado. Para isso, foram adicionados dois botões na tela inicial que permitem a síntese e simulação, rodando os comandos necessários via terminal.

A Figura 3 apresenta a interface da plataforma, com as opções correspondentes à criação do sistema integrado, bem como a síntese e simulação.

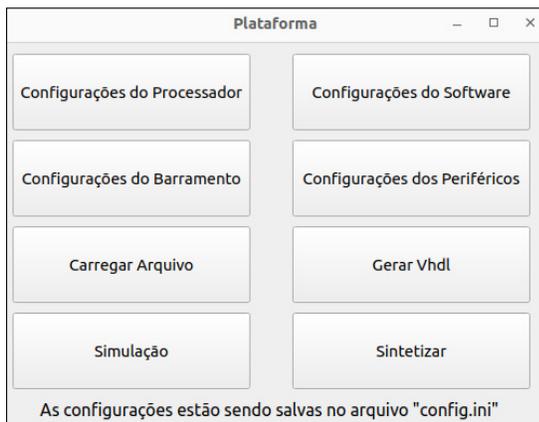


Figura 3: Interface da plataforma

5 RESULTADOS

Para o sistema avaliado neste trabalho, foi utilizado o processador HARV (com tolerância a falhas), o barramento AMBA (com o *timeout* desabilitado e a fonte de *reset* via periféricos), uma aplicação *baremetal*, além de todos os periféricos disponíveis na plataforma habilitados e endereçados.

Os resultados de implementação do sistema integrado completo foram coletados pela ferramenta de síntese, apresentando um consumo de 6247 Look-Up Tables, 3164 Flip-Flops e 130 blocos de memória. Além disso, o sistema gerado possui uma frequência máxima de operação de 61 MHz e uma potência dissipada estimada de 139 mW. Foi observado que a descrição em VHDL gerada pela plataforma equivale a uma construção manual do sistema integrado, tanto na síntese como nas simulações.

6 CONSIDERAÇÕES FINAIS

Neste trabalho, foi apresentada uma plataforma para geração de sistemas integrados de maneira prática e simples. O desenvolvimento desta plataforma foi motivado pela falta de soluções de integração entre os processadores e aceleradores, que são geralmente desenvolvidos de maneira isolada. Como trabalhos futuros, pretende-se aumentar o suporte ao número de processadores e aceleradores para a construção de sistemas mais complexos e heterogêneos.

AGRADECIMENTOS

Este trabalho foi financiado em parte pelo Programa de Bolsas Universitárias de Santa Catarina (UNIEDU), pela Fundação de Amparo à Pesquisa e Inovação de Santa Catarina (FAPESC-2021TR001907) e pela Region d'Occitanie e École Doctorale I2C da Universidade de Montpellier (20007368/ALDOCT-000932).

REFERÊNCIAS

- [1] INPE. Perguntas Frequentes, Principais produtos e Serviços do Inpe, 2021. URL <http://www.inpe.br/faq/index.php?pai=4>.
- [2] Andre Carlos Ponce Leon Ferreira Carvalho and Ana Carolina Lorena. *Introdução à Computação - Hardware, Software*. LTC, São Paulo, 2017.
- [3] Yakun Sophia Shao and David Brooks. *Research Infrastructures for Hardware Accelerators*. Morgan & Claypool, 2015.
- [4] Jörg Henkel and Nikil Dutt. *Dependable Embedded Systems*. Springer Cham, Cham, Suíça, 1st edition, 2021.
- [5] Parineeth M. Reddy. Embedded systems. *Resonance*, 7:20–30, 12 2002.
- [6] Frank Vahid and Tony Givargis. *Embedded System Design: A Unified Hardware/Software Introduction*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 2001.
- [7] D. Greaves. *Modern System-on-Chip Design on Arm*. ARM Education Media, 2021.
- [8] Michael Flynn and Wayne Luk. *Computer System Design: System on a Chip*. 06 2011.
- [9] ARM. AMBA 4, 2022. URL <https://www.arm.com/architecture/system-architectures/amba/amba-4>.
- [10] Gaisler Cobham. *GRLIB IP Library User's Manual*. CAES, Goteborg, 2022.
- [11] Gaisler. NOEL-V Processor, 2021. URL <https://www.gaisler.com/index.php/products/processors/noel-v>.
- [12] Andreas Traber and Michael Gautschi. PULPino: Datasheet, 2017. URL <https://github.com/pulp-platform/pulpino/blob/master/doc/datasheet/datasheet.pdf>.
- [13] Krste ASANOVIĆ, Rimas Avizienis, Jonathan Bachrach, Scott Beamer, David Biancolin, Christopher Celio, Henry Cook, Daniel Dabbelt, John Hauser, Adam Izraelevitz, Sagar Karandikar, Ben Keller, Donggyu Kim, John Koenig, Yunsup Lee, Eric Love, Martin Maas, Albert Magyar, Howard Mao, Miquel Moreto, Albert Ou, David A. Patterson, Brian Richards, Colin Schmidt, Stephen Twigg, Huy Vo, and Andrew Waterman. The rocket chip generator. Technical Report UCB/EECS-2016-17, EECS Department, University of California, Berkeley, Apr 2016. URL <http://www2.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-17.html>.
- [14] Douglas Almeida Santos, Lucas Matana Luza, Cesar Albenes Zeferino, Luigi Dilillo, and Douglas Rossi Melo. A Low-Cost Fault-Tolerant RISC-V Processor for Space Systems. In *2020 15th Design Technology of Integrated Systems in Nanoscale Era (DTIS)*, pages 1–5, 2020.
- [15] Wesley Grignani, Gabriela Wisbecki, Felipe Viel, and Douglas Rossi Melo. A high-level synthesis compressor of hyperspectral images based on ccsds 123.0-b-2. In *IAA Joint 5th Latin American CubeSat Workshop and 3rd Latin American Symposium on Small Satellites*, 2022.