Análise de Alternativas de Integração do Processador LEON3 em Sistemas Embarcados

Lucas M. V. Pereira[1], Douglas R. Melo[1,2], Cesar A. Zeferino[1], Eduardo A. Bezerra[2]

[1]Laboratório de Sistemas Embarcados e Distribuídos
Universidade do Vale do Itajaí (UNIVALI) – Itajaí, SC – Brasil

[2]Laboratório de Comunicações e Sistemas Embarcados
Universidade Federal de Santa Catarina (UFSC) – Florianópolis, SC – Brasil

lucas.pereira@edu.univali.br, drm@univali.br,
zeferino@univali.br, eduardo.bezerra@ufsc.br

Abstract. *LEON3 processor consists of a softcore originally developed by the European Space Agency (ESA) for space applications. Currently it is part of an IP library known as GRLIB, supported by Cobham Gaisler. In addition to the IPs available in the library, it is possible to integrate custom components. The integration between the components in the system is performed through the AMBA 2.0 bus, developed by ARM, using the AHB protocol for the high performance cores and the APB protocol for peripherals. However, bus architecture does not meet the communication requirements of the future systems-on-chip, being the consensus in the literature the adoption of structures of greater scalability, such as networks-on-chip. This paper describes the studies based on LEON3 communication services, analyse the implementation of different architectures and the adaptations to integrate it to a network-on-chip.*

Resumo. *O processador LEON3 consiste de um softcore originalmente desenvolvido pela Agência Espacial Europeia (ESA) para uso em aplicações espaciais. Atualmente faz parte da biblioteca de IPs GRLIB mantida pela empresa Cobham Gaisler. Além dos IPs disponíveis nessa biblioteca, é possibilitada a incorporação de componentes customizados. A integração entre os componentes no sistema é realizada com o uso do barramento AMBA 2.0, desenvolvido pela ARM, utilizando o protocolo AHB para os núcleos de alto desempenho e o protocolo APB para periféricos. No entanto, o barramento não atende aos requisitos de comunicação dos futuros sistemas integrados, sendo consenso da literatura a adoção de estruturas de maior escalabilidade, como as redes-em-chip. Este artigo descreve os estudos feitos sobre a comunicação do processador LEON3, analisa a implementação de diferentes arquiteturas e as adaptações necessárias para a integração em uma rede-em-chip.*

1. Introdução

O termo “computador de bordo”, como o próprio nome sugere, refere-se a qualquer unidade automotiva ou aviônica que realiza o processamento de dados. Além disso, também se refere a computadores embarcados em satélites para suportar, por exemplo, o controle de altitude e órbita, comunicação, detecção de falhas, dentre outras funções [ESA 2014].

Um dos processadores usados em aplicações espaciais é o LEON, o qual foi desenvolvido pela Agência Espacial Europeia (ESA) em 1997. O projeto foi primariamente desenvolvido para aplicações espaciais europeias, além de disponibilizar o projeto de um processador aberto, portável, não proprietário e independente de ferramentas de síntese [Gaisler 2016].

Atualmente mantido pela Cobham Gaisler em sua terceira versão, o processador LEON3 faz parte da biblioteca de componentes denominada GRLIB, a qual utiliza o barramento AMBA 2.0 para a comunicação entre os núcleos. Dentre as possibilidades de configuração disponíveis pela biblioteca, nota-se o limite de quatro instâncias do LEON3, inviabilizando a escalabilidade acima deste número na implementação de um sistema integrado [Gaisler 2016].

A interconexão de núcleos usando um barramento é algo amplamente utilizado. No entanto, essa abordagem não oferece nem paralelismo em comunicação e nem a escalabilidade de desempenho necessárias para sistemas com dezenas de núcleos. Visando superar as limitações impostas pelos barramentos, são utilizadas redes chaveadas, denominadas de redes-em-chip (NoCs – Networks-on-Chip) para interligar os núcleos por meio de interfaces de rede, roteadores e enlaces ponto-a-ponto [Jantsch e Tenhunen 2003].

As interfaces de rede são responsáveis por traduzir a comunicação entre os núcleos e a NoC. Um exemplo de interface de rede é a XIRU (eXtensible Interface for Routing Unit) [Melo, Wangham e Zeferino 2012] para aplicação na rede SoCIN (System-on-Chip Interconnection Network) [Zeferino e Susin 2003]. A interface XIRU provê serviços descritos nas camadas intermediárias do modelo OSI (Open Systems Interconnection) e é compatível com o padrão de barramento Avalon para integração dos processadores NIOS II, da fabricante de dispositivos lógicos programáveis Altera, de propriedade da Intel [Altera 2016].

Este trabalho visa o estudo e a elaboração de sistemas utilizando o LEON3 (e suas ferramentas de configuração) a fim de caracterizar os serviços de comunicação utilizados por esse processador, para a integração do mesmo a tecnologias disponíveis no laboratório. São disponibilizadas diferentes arquiteturas que possibilitam o uso de múltiplas instâncias do processador mantendo a compatibilidade com a GRLIB.

1.1 Sistemas Integrados

Um sistema integrado é um circuito que implementa a maioria ou todas as funções de um sistema computacional completo. A característica primária de um sistema integrado é a complexidade [Jerraya e Wolf 2005]. O sistema integrado agrupa diversos circuitos em um único e vem ganhando força principalmente no mercado de dispositivos móveis devido ao seu baixo consumo de energia [Bennett 2004].

Para descrever sistemas digitais, podem ser utilizadas linguagens de descrição de hardware (HDL – Hardware Description Language). Existem alguns padrões de HDLs que são amplamente usados na indústria, sendo esses o VHDL e o Verilog. O VHDL (Very High Speed Integrated Circuit Hardware Description Language) foi desenvolvido em 1980 pelo departamento de defesa dos Estados Unidos e padronizada pelo IEEE em 1987. Essa permite verificar o comportamento do sistema por meio de simulação e usar diversos tipos de abstração. Já o Verilog, criado em 1985 pela Gateway Design Automation e sendo um padrão IEEE desde 1995, suporta o projeto, verificação e

implementação de projetos analógicos, digitais e híbridos em vários níveis de abstração, tendo tanto objetivo como aplicações similares ao VHDL [Fregni e Saraiva 1995].

1.2 AMBA

Com o intuito de facilitar a comunicação entre processadores, controladores e outros periféricos, o AMBA (Advanced Microcontroller Bus Architecture) foi introduzido pela ARM em 1996. Na sua primeira versão, os barramentos AMBA se limitavam ao ASB (Advanced System Bus) e o APB (Advanced Peripheral Bus). Já na sua segunda versão, a ARM implementou o barramento AHB (AMBA High-Performace Bus) [ARM 1995], o mesmo usado pelo LEON3 e a GRLIB.

O barramento de alto desempenho AHB suporta múltiplos mestres e é adaptável para operações com alta largura de banda. Como ilustra a Figura 1, um sistema tipo AHB consiste de basicamente três componentes diferentes (mestres, escravos e árbitro) [ARM 2016].

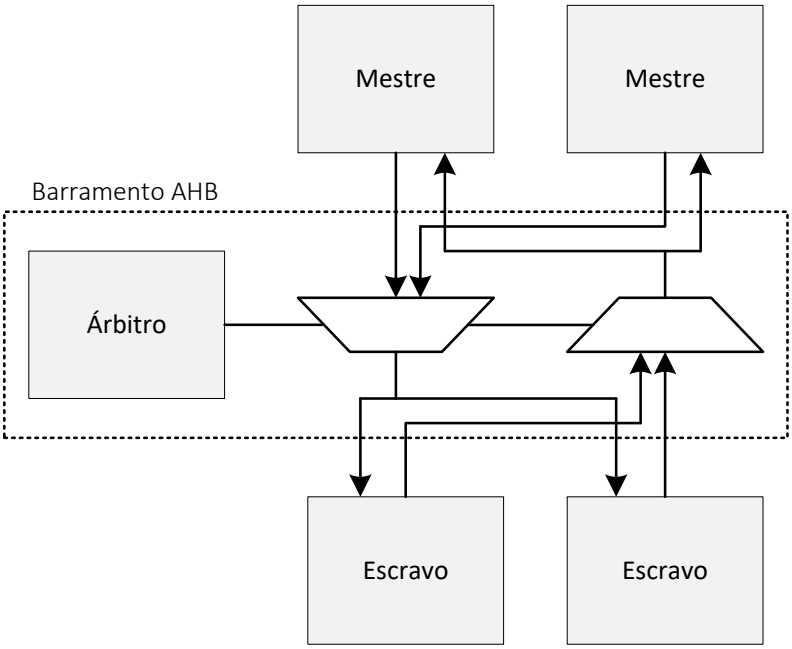

Figura 1. Interconexão do barramento AHB

Um sistema AHB pode possuir um ou mais mestres, os quais são núcleos capazes de iniciar operações de leitura e escrita. Para tal, um mestre provê ao barramento informações como endereço e informações de controle, além do dado em caso de operações de escrita.

Escravos AHB respondem às operações de leitura e escrita com um espaço de endereços reservado. Esses enviam um sinal para o mestre com o *status* da operação, se foi bem-sucedida, se ocorreu falha ou se está em espera. Os escravos mais comuns são memória em chip e interfaces de memória fora do chip.

O mecanismo de arbitragem é utilizado para assegurar que somente um mestre terá acesso ao barramento por vez. O árbitro observa as requisições provenientes de diferentes fontes para decidir qual o mestre com maior prioridade, garantindo assim o barramento ao mesmo.

1.3 Redes-em-Chip

A necessidade de redução de tempo de projeto fez com que diversos sistemas integrados fossem implementados utilizando blocos pré-projetados, também conhecidos como IPs (Intelectual Propriety). A integração de dezenas de IPs em um sistema com um único barramento acaba inviabilizando a implementação de sistemas mais complexos, surgindo como alternativa o emprego de NoCs [Jantsch e Tenhunen 2003].

Uma NoC é constituída de um conjunto de roteadores interligados por canais ponto-a-ponto. Cada roteador possui portas para se comunicar com roteadores vizinhos e com os componentes conectados ao mesmo. Um exemplo de NoC é a SoCIN, uma rede de baixo custo com desempenho escalável baseada em um roteador parametrizável que permite a síntese de redes com diferentes características de custo e desempenho [Zeferino e Susin 2003].

Além de roteadores e dos canais de comunicação, uma NoC necessita de uma interface de rede para adaptar seu protocolo de comunicação com o do núcleo conectado e, consequentemente, prover serviços de comunicação necessários ao sistema [Melo, Wangham e Zeferino 2014].

1.4 Processador LEON3

O LEON3 é um modelo VHDL parametrizável de um processador 32-bits baseado na arquitetura SPARC V8. Disponibilizado pela Cobham Gaisler, seu código é disponível sob licença GNU GLP, possibilitando uso gratuito para pesquisa e educação, mas também possui versões disponibilizadas sob licença comercial [Gaisler 2016]. A Figura 2 ilustra o diagrama de blocos correspondente ao LEON3.

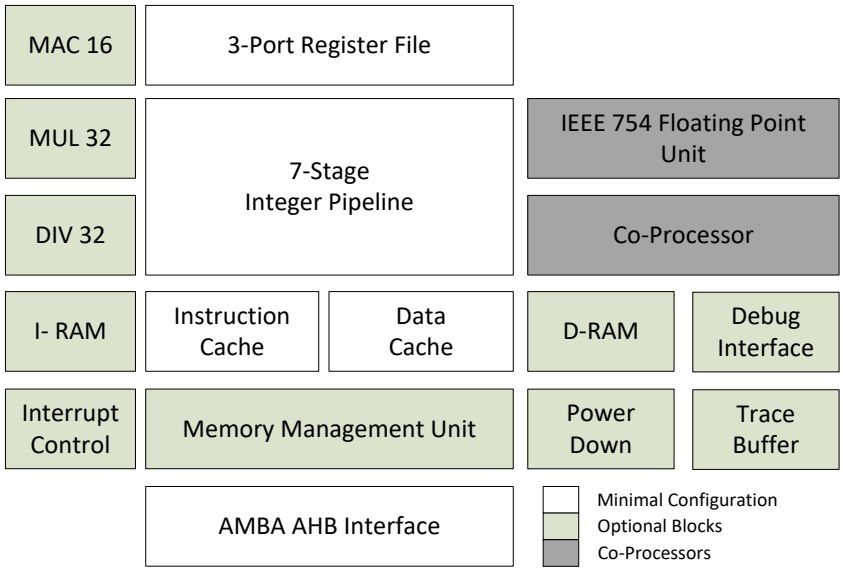

Figura 2. Diagrama de blocos do LEON3

A sua terceira versão faz parte da biblioteca de IPs GRLIB. Essa biblioteca é composta de uma vasta opção de núcleos desenhados para desenvolvimento de sistemas em chip. A GRLIB é independente de fabricantes e de tecnologias de silício, com suporte a diferentes ferramentas de CAD e diferentes tecnologias. Um único método *plug & play* é usado para configurar e integrar núcleos sem ter a necessidade de alterar nenhum recurso global.

Na GRLIB, as informações *plug & play* consistem de três itens: (*i*) um identificador único de núcleo; (*ii*) mapa de memória; e (*iii*) o vetor de interrupção utilizado. Essas informações são registradas como constantes para o decodificador AHB e qualquer mestre do barramento possui acesso de leitura. A Figura 3 apresenta a palavra de configuração de cada componente, incluindo o ID do fornecedor, ID do componente e informação de roteamento de interrupção.

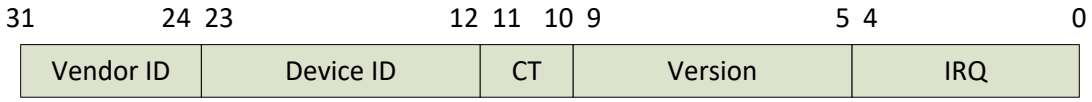

Figura 3. Palavra de configuração

A GRLIB funciona de forma independente da tecnologia alvo, podendo ser facilmente implementada tanto em ASIC quanto em FPGA. A portabilidade é garantida por componentes virtuais para a tecnologia desejada. Na arquitetura do componente, declarações VHDL são utilizadas para instanciar as respectivas macros da biblioteca de tecnologias.

2. Desenvolvimento

Partindo do estudo sobre o processador LEON3 e suas ferramentas, foram realizadas simulações e verificações para obter as características estruturais e de funcionamento da GRLIB. Foram realizadas sínteses visando o dispositivo Cyclone II EP2C35F672C6N e prototipação no kit de desenvolvimento DE2. Posteriormente, foi realizada a expansão da biblioteca de componentes para a inserção de IPs externos. O desenvolvimento se deu a partir da arquitetura do sistema padrão (disponibilizada pela GRLIB), com posteriores adaptações para a integração da interface de rede conectada ao barramento e também conectada diretamente ao processador LEON3.

2.1 Arquitetura de sistema padrão

Para a análise dos sinais de comunicação utilizados pelo LEON3, foi elaborada uma arquitetura padrão fornecida pela GRLIB, conforme ilustra a Figura 4. O sistema é composto de um barramento AHB que interliga uma instância do processador LEON3, um controlador de memória e uma ponte AHB/APB. A ponte AHB é conectada a um componente UART por meio do barramento de periféricos APB. Nessa figura, são abstraídos os blocos de JTAG e Debug Support, necessários para a gravação do sistema no FPGA e depuração.

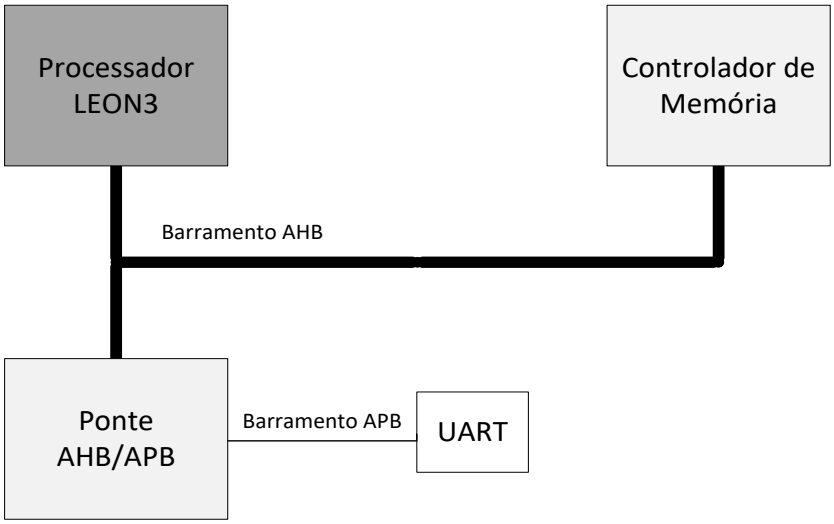

Figura 4. Arquitetura de sistema padrão

Simulação

Para a simulação da arquitetura padrão, foi feito uso da ferramenta ModelSim, presente no ambiente Quartus II [Altera 2016], onde pôde-se analisar os sinais de comunicação utilizados não só pelo processador, mas também pelos outros componentes. O arquivo de *Testbench* utilizado para a simulação é fornecido pela biblioteca GRLIB.

Foi criada uma rotina em linguagem C para um cenário de validação da comunicação entre o processador e um escravo, cujo código pode ser observado no Quadro 1. Este cenário foi adicionado à simulação, sendo carregado em um arquivo de memória no formado SREC (S-record) que é executado após o *boot* do processador.

```
int main(){
    // Define o endereço de memória a ser usado
    volatile int *ptr = 0x400019C0;

    // Armazena o dado no endereço definido
    *ptr = 0xCAFECAFE;

    return 0;
}
```

Quadro 1. Programa de teste para simulação

Na Figura 5, pode-se observar a operação de escrita na memória efetuada pelo processador. A transação ocorre pelo endereçamento da área de memória (`0x400019C0`) onde acontece a escrita, e no ciclo de relógio seguinte, na via de dados de escrita (*hwdata*), com a palavra `0xCAFECAFE`.

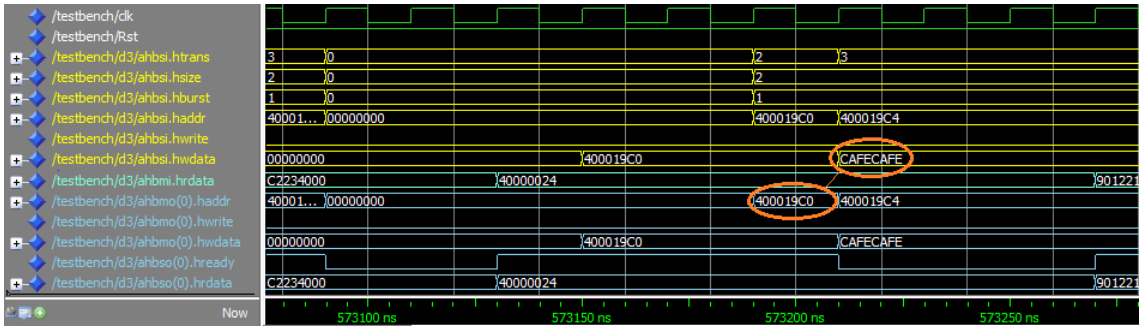

Figura 5. Simulação de transferência simples

Verificação

Para verificar os resultados obtidos na simulação, foi feita a prototipação do sistema em FPGA. Acrescido à necessidade de depurar a execução, foram adicionados ao sistema o link JTAG e a Debug Suport Unit, os quais são responsáveis por colocar o processador em modo de depuração para execução do software de teste (Quadro 2). Essa etapa é feita com o uso da ferramenta de depuração GRMON [Gaisler 2016] que fornece comunicação com o sistema, permitindo escrita e leitura em todos os registradores e memória, execução de aplicações e depuração.

O cenário de teste consiste de três passos: (*i*) escrita em memória; (*ii*) leitura no mesmo endereço de memória; (*iii*) retorno do dado pela UART, localizada no barramento APB e conectada ao console da GRMON.

```
#include <stdio.h>
int main(){
    // Define o endereço de memória a ser usado
    volatile int *ptr = 0x400019C0;

    // Variável para leitura
    int var;

    // Armazena o dado no endereço definido
    *ptr = 0xCAFECAFE;
    var = *ptr;

    // Impressão via UART
    printf("Valor lido: 0x%X", var);

    return 0;
}
```

Quadro 2. Programa de teste para verificação

Para obter o resultado da comunicação, foi utilizada a ferramenta SignalTap II [Altera 2016], a qual é responsável por capturar sinais provenientes do FPGA. Definido um gatilho para a ferramenta, no caso o endereço a ser escrito e o dado do mesmo, pode-se observar o envio da informação para o endereço determinado, com posterior leitura. A Figura 6 apresenta os sinais obtidos do FPGA, os quais são similares aos observados na simulação.

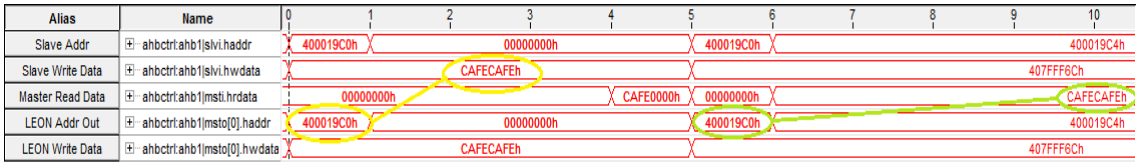

Figura 6. Verificação de transferência de escrita e leitura

Para a efetiva utilização de múltiplas instâncias do processador LEON3 em sistemas integrados baseados em NoCs, faz-se necessário o uso de interfaces de rede que realizam a adaptação do padrão de comunicação adotado pelo processador (AHB no caso do LEON3) ao padrão da rede. A seguir são descritas as duas alternativas de arquitetura de sistema analisadas.

2.2 Arquitetura com interface de rede conectada ao barramento

A primeira arquitetura de sistema consiste no uso de uma interface de rede conectada como um periférico ao barramento. Esta centraliza todas as comunicações externas e permite com que o barramento do sistema opere de forma independente da rede, permitindo a utilização de outros periféricos, barramentos e até mesmo uma rede-em-chip. A Figura 7 ilustra essa arquitetura.

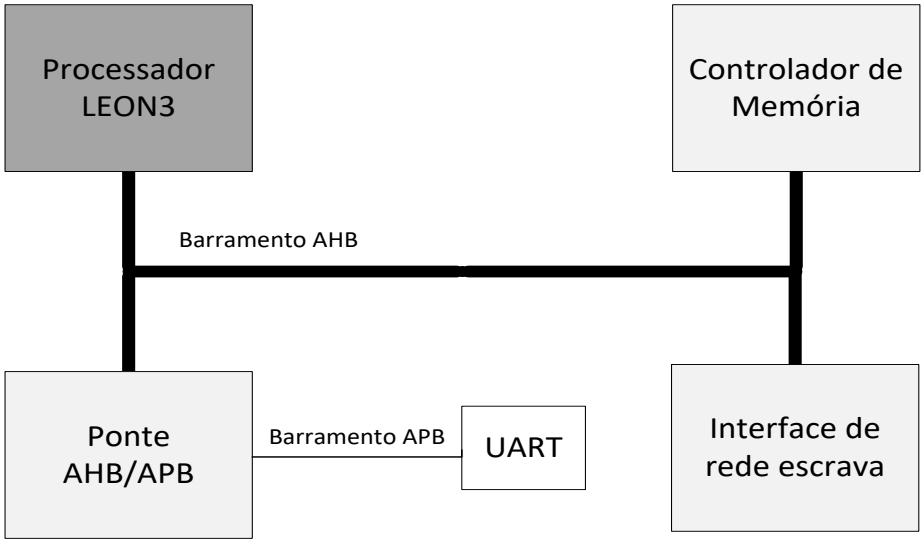

Figura 7. Arquitetura com interface de rede conectada ao barramento

Para a elaboração dessa arquitetura, foi utilizado o trabalho de Syed (2010) como referência. Nesse trabalho, é realizada a implementação de quatro instâncias do sistema LEON3 padrão em uma NoC de dimensões 2x2.

2.3 Arquitetura com interface de rede conectada ao processador

A segunda arquitetura de sistema avaliada emprega uma interface de rede para cada núcleo presente no sistema. Dessa forma, domínios de barramento independentes são definidos entre as interfaces de rede e os núcleos e não há a necessidade de estruturas de arbitragem, pois não há concorrência pelo meio. A Figura 8 apresenta essa arquitetura de sistema.

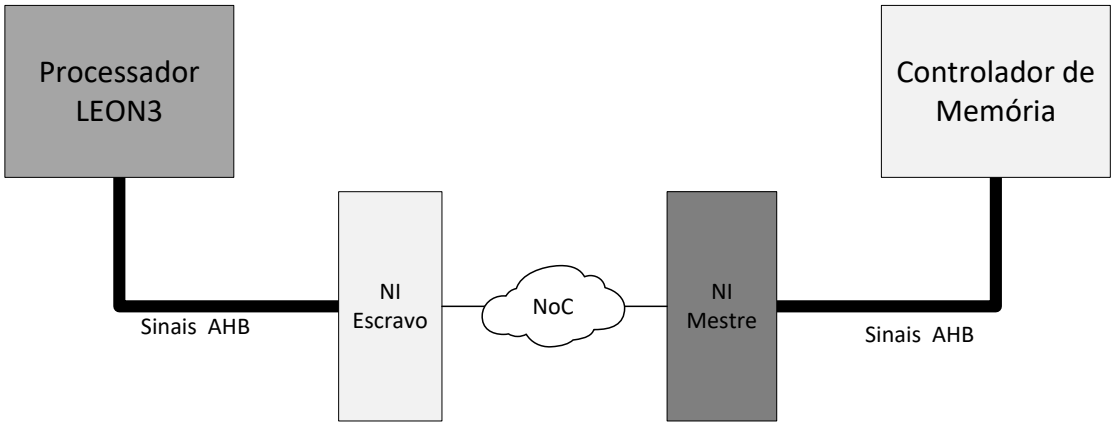

Figura 8. Arquitetura com interface de rede conectada ao processador

O trabalho de Gaya et al. (2017) foi utilizado para a elaboração dessa arquitetura. Nele foram feitas adaptações na interface de rede XIRU de forma a suportar os sinais e serviços do protocolo AMBA-AHB, admitindo assim que núcleos baseados nesse padrão sejam integrados à rede SoCIN.

3. Resultados

A fim de comparar os custos, as alternativas de arquitetura foram sintetizadas visando o dispositivo FPGA Cyclone II EP2C35F672C6N da Altera.

Na Tabela 1, são apresentados os custos de silício expressos pelo número de LUTs (Lookup Tables) e de FFs (Flip-Flops) consumidos em cada solução. Como esperado, a arquitetura padrão utilizada como referência é a de menor custo, enquanto a solução utilizando o barramento apresentou um sobrecusto de 21% em LUTs e 23% em FFs, e a conexão direta entre processador e interface com 34% em LUTs e 61% em FFs.

Arquitetura	LUTs	FFs	Frequência de operação	Potência dissipada
Padrão	4.745	1.812	62,2 MHz	258,7 mW
Interface de rede conectada ao barramento	5.784	2.231	64,1 MHz	260,3 mW
Interface de rede conectada ao processador	6.387	2.925	62,1 MHz	262,9 mW

Tabela 1. Tabela comparativa de custo

Também são apresentadas a frequência máxima de operação e a potência dissipada nas três arquiteturas. A frequência foi obtida por meio do relatório de síntese do Quartus II, já a potência estática foi obtida por meio da ferramenta PowerPlay da Altera. Como pode ser observado, não ocorreram mudanças significativas na frequência de operação e na potência estática das soluções.

4. Conclusões

Este artigo apresentou uma análise comparativa de três arquiteturas de sistemas para a integração do processador LEON3 em um sistema embarcado. Além da arquitetura padrão disponibilizada pela ferramenta GRLIB, foram analisadas abordagens

dependentes de barramento e conexão direta entre interface e processador, sendo que essas oferecem escalabilidade por permitirem a integração de múltiplas instâncias do processador por meio de uma NoC.

Como pôde ser observada, a solução dependente do barramento apresentou um sobrecusto de 21% em LUTs e 23% em FFs quando comparada com a arquitetura padrão, enquanto a independente resultou em um valor 34% maior em LUTs e 61% maior em FFs na mesma comparação. O sobrecusto elevado se dá devido ao número reduzido de núcleos utilizados nas implementações, o que tende a ser atenuado em sistemas reais e de maior complexidade. Com relação à frequência de operação e potência dissipada, não ocorreu mudança significativa nos diferentes cenários.

Como contribuição, foi disponibilizada uma plataforma que permite a integração de novos componentes que utilizam o barramento AHB, mantendo a compatibilidade com as devidas ferramentas mencionadas anteriormente, sem a necessidade de utilizar o sistema de barramento nativo do processador. Como trabalhos futuros, é proposta a elaboração de sistemas complexos, com dezenas de núcleos, de forma a verificar a viabilidade das soluções oferecidas na plataforma desenvolvida.

Referências

Altera (2016) “Measure Adventure”, https://www.altera.com, Dezembro.

ARM (1995) “The Architecture for the Digital World”, http://www.arm.com, Dezembro.

Bennet, P. (2004) “The why, where and what of low-power SoC design”, http://www.eetimes.com/document.asp?doc_id=1276973, Dezembro.

ESA (2004) “On-board computers”, http://www.esa.int/Our_Activities/Space_Engineering_Technology/Onboard_Computer_and_Data_Handling/Onboard_Computers, Dezembro.

Fregni, E. e Saraiva, A. M. (1995) “Engenharia do Projeto Lógico Digitais: conceitos e prática”, Editado por Edgard Blücher, São Paulo.

Gaisler (2016), http://www.gaisler.com, Dezembro.

Gaya, F. L., Zeferino, C. A., Melo, D. R., Bezerra, E. A.. (2017) “AMBA-AHB Network Interface for Core Interconnection in a Network-on-Chip”, Iberchip, 13th, Bariloche.

Jantsch, A. e Tenhunen, H.. (2003) “Networks on chip”, Editado por Kluwer, Boston.

Jerraya, A. A. e Wolf, W. (2005) “Multiprocessor systems-on-chip: the Morgan Kaufmann series in systems on silicon”, Editado por Elsevier, São Francisco.

Melo, D. R., Wangham, M. S. e Zeferino, C. A. (2014) “XIRU: Interface de Rede Extensível para Integração de Núcleos a uma Rede-em-Chip”, Revista de Informática Teórica e Aplicada: RITA, v. 21, p. 10-31.

Syed, J. R. (2010) “LEON3 NoC System Generator”, Tese (Master Degree in System on Chip Design), KTH Royal Institute of Technology.

Zeferino, C. A. e Susin A. A. (2003) “SoCIN: a parametric and scalable Network-on-Chip”, Symposium on Integrated Circuits and Systems (SBCCI), 16th, São Paulo, p. 169-174.