Universidade Federal da Bahia
Escola Politécnica
Programa de Pós-Graduação em Engenharia Elétrica
Doutorado em Engenharia Elétrica

Nestor Dias Pereira Neto

Relatório de atividades - Projeto de pesquisa I

Orientador: Prof. Dr. Wagner Luiz Alves de Oliveira

Coorientador: Prof. Dr. Paulo César Machado de Abreu Farias

Salvador

Julho 2023

Sumário

1 Introdução

Nos últimos anos novas técnicas para construção de robôs t se destacado, em especial áreas como robótica móvel e robótica colaborativa. Uma das principais características dessas áreas é a exigência de um alto grau de percepção do ambiente que rodeia o robô, além de uma execução mais precisa em seus movimentos, isto devido ao fato de que as atividades desempenhadas por estes robôs estão exigindo um nível cada vez maior de interação com aquelas desempenhadas por seres humanos.

Este grau de precisão requerido no desenvolvimento de novos robôs exige sistemas robóticos cada vez mais complexos que precisam de processadores igualmente mais poderosos, consequentemente demandando um maior consumo de energia. Este aumento de consumo provoca uma verdadeira disputa entre poder de processamento e consumo, já na fase de levantamento dos requisitos de um novo projeto, o que pode vir a ser um problema principalmente em sistemas que fazem uso de baterias.

O FPGA é uma excelente alternativa para resolver este impasse, por oferecer um aumento do poder de processamento associado a um baixo consumo de energia. O potencial que os FPGAs possuem para melhorar o desempenho de sistemas computacionais já é bem conhecido há algum tempo. Myer-Baese (2014) descreve algumas vantagens dos FPGAs modernos para uso em processamento digitais de sinais, como as cadeias de *fast-carry* usadas para implementar MACs de alta velocidade e o paralelismo tipicamente encontrado em projetos implementados em FPGA.

Por essas características, FPGAs necessitam de frequências menores de trabalho para alcançar desempenho equivalente ou superior às soluções baseadas unicamente em processadores, diminuindo a dissipação térmica e, consequentemente, necessitando um consumo de energia consideravelmente menor. Todas estas características oferecidas pelo hardware configurável o tornam um recurso bastante interessante para o uso em projetos de robótica.

Entretanto, apesar de oferecer grandes vantagens, as facilidades de desenvolvimento encontradas em aplicações que fazem uso de softwares não estão disponíveis na mesma proporção no mundo do hardware configurável. As maiores dificuldades no desenvolvimento de soluções baseadas em FPGAs referem-se ao longo tempo de projeto e à necessidade de mão-de-obra extremamente especializada, o que aumenta consideravelmente o custo final de tais projetos. Desta forma, o uso de FPGAs em projetos de robótica acaba sendo desencorajado.

Atualmente o *framework* ROS está se consolidando como o padrão na criação de novas plataformas robóticas, tanto no desenvolvimento de manipuladores colaborativos

quanto na robótica móvel. O objetivo do ROS é facilitar a elaboração de novos robôs, através de um conjunto completo de ferramentas para desenvolvimento, como *drivers* para sensores e atuadores, bibliotecas e, principalmente, reuso de código. Agrupar inúmeros "blocos" de software usados em robótica, fornecer driver para componentes de hardwares específicos (sensores e atuadores), gerenciar troca de mensagens entre os nós que fazem parte do sistema, são as funções do ROS.

Estas características fazem com que o ROS seja reconhecido com um pseudo sistema operacional (PYO et al., 2017). Dessa maneira o ROS se tornou muito ágil no desenvolvimento de novas aplicações para robótica. Usando aplicações já desenvolvidas e testadas por outros desenvolvedores, pode-se criar novos sistemas completos apenas gerenciando estas aplicações na estrutura interna do ROS. Essa abordagem fez com que o número de pacotes para o ROS cresça a uma taxa muito rápida, desde o ano de seu lançamento, em 2007, até 2020, o ROS aumentou de 1 para 2647 pacotes (KOLAK et al., 2020).

Aproveitar as facilidades de desenvolvimento proporcionadas pelo ROS em conjunto com o alto poder de processamento e baixo consumo que FPGAs oferecem, seria um cenário ideal no desenvolvimento de novas aplicações com robôs. Para isso, precisamos estabelecer uma conexão com uma taxa de transferência de dados alta o suficiente para não influenciar de forma negativa no tempo de processamento e, adicionalmente, tornar relativamente fácil seu uso por desenvolvedores especializados em robótica, mas sem grande experiência em FPGA. Portanto, este trabalho tem como objetivo estabelecer uma comunicação de alto desempenho entre ROS e um FPGA para que se possa aproveitar o melhor das duas tecnologias em projetos de robótica.

- **Como estabelecer a comunicação entre o ROS e um sistema de processamento auxiliar embarcado em um FPGA?**

Este problema é o que este trabalho busca resolver, possibilitando assim, o uso de aceleração por hardware através do FPGA, para inclusão de tal dispositivo no desenvolvimento de novos projetos de robótica. Projetistas especializados em robótica poderão aproveitar dos benefícios do uso do hardware dedicado em seus projetos, enquanto profissionais que trabalham com descrição de hardware poderão desenvolver novas soluções para problemas de robótica de forma modularizada.

1.1 Objetivos

1.1.1 Objetivo Geral

Desenvolver uma solução para estabelecer comunicação entre *Field-Programmable Gate Array - FPGA*, configurado como um co-processador de vídeo.

1.1.2 Objetivos Específicos

- Estudar teoria dos assuntos relevantes ao projeto: Verilog HDL, Embedded Linx, Cyclone V, TCP/IP Stack, ROS;
- Estudar conceitos de programação de redes usando sockets em liguagem C++ e detalhes dos protocolos da rede TCP/IP usada para comunicação interna dos nós e serviços ROS;
- Implementar distribuição Embedded Linux para processador ARM embarcado no SoC Cyclone V da Intel;
- Estabelecer comunicação entre o ROS e o Cyclone V, através da tecnologia Gigabit Ethernet;
- Desenvolver aplicação em Verilog para testar comunicação;
- Avaliar o desempenho da rede entre o computador e o protótipo após a inclusão do FPGA ao sistema.

1.2 Organização

No Capítulo 1 é apresentada a introdução do texto, com uma breve contextualização do tema, além do problema ao qual o trabalho se propõe a resolver. Neste capítulo também são apresentados a justificativa e os objetivos gerais e específicos. Na sequência o texto é dividido em três partes: Referencial teórico, Desenvolvimento e Resultados.

No Referencial teórico temos dois capítulos onde são apresentadas as tecnologias utilizadas no trabalho: no Capítulo **??** é discorrido sobre o *System on Chip* - SoC utilizado para o desenvolvimento do trabalho e no Capítulo **??** é falado sobre o *Robot Operating System* - ROS.

Na segunda parte deste trabalho, chamada de Desenvolvimento, são explicadas as etapas do desenvolvimento. A arquitetura do sistema é explicada no Capítulo **??**. No Capítulo **??** é explicado o Cliente, enquanto o Servidor é descrito no Capítulo **??**

Na última parte deste documento são discutidos os Resultados alcançados, que são apresentados no Capítulo **??**. Já no Capítulo **??** é apresentada a conclusão desta pesquisa e no Capítulo **??** são apresentadas algumas sugestões de trabalhos futuros.

Referências

ALTERA. *Cyclone V Hard Processor System: Technical Reference Manual*. [S.l.], 2018. Nenhuma citação no texto.

ALTERA-OPENSOURCE. *linux-socfpga*. 2022. Disponível em: <https://github.com/altera-opensource/linux-socfpga>. Nenhuma citação no texto.

ARM. *ARM ® Cortex ® -A9 MPCore Technical Reference Manual*. r4p1. [S.l.], 2016. Nenhuma citação no texto.

ARM. *arm Glossary FPGA*. 2022. Disponível em: <https://www.arm.com/glossary/fpga>. Acesso em: 5 março 2022. Nenhuma citação no texto.

ARM, D. *Cortex A9*. 2022. Disponível em: <https://developer.arm.com/Processors/Cortex-A9#Technical-Specifications>. Acesso em: 23 junho 2022. Nenhuma citação no texto.

FERGUSON, M. *Rosserial*. 2018. Rosserial. Disponível em: <http://wiki.ros.org/rosserial>. Acesso em: 20 julho 2021. Nenhuma citação no texto.

FLYNN, M. J.; LUK, W. *Computer System Designs: System-on-Chip*. 1. ed. New Jersey: Wile, 2011. Nenhuma citação no texto.

INTEL. *FPGAs Cyclone® V e FPGAs SoC*. 2022. Disponível em: <https://www.intel.com.br/content/www/br/pt/products/details/fpga/cyclone/v.html>. Acesso em: 23 março 2022. Nenhuma citação no texto.

JOSEPH, L. *Mastering ROS for Robotics Programming*. 1. ed. Birmingham: Packt Publishing Ltd, 2015. Nenhuma citação no texto.

KERRISK, M. *Linux Programmer's Manual - mem, kmem, port - system memory, kernel memory and system ports*. [S.l.], 2021. Disponível em: <https://man7.org/linux/man-pages/man4/mem.4.html>. Acesso em: 20 julho 2022. Nenhuma citação no texto.

KOLAK, S. et al. It takes a village to build a robot: An empirical study of the ros ecosystem. *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, p. 430–440, 2020. Disponível em: <https://ieeexplore.ieee.org/document/9240632/references#references>. Citado na página 3.

MAAN, L.; BAKER, L. B. *Intel to buy Altera for $16.7 billion in its biggest deal ever*. 2015. Disponível em: <https://www.reuters.com/article/us-altera-m-a-intel-idUSKBN0OH2E020150601>. Acesso em: 23 março 2022. Nenhuma citação no texto.

MAHTANI, A. et al. *Effective Robotics Programming with ROS*. 3. ed. Birmingham: Packt Publishing Ltd, 2016. Nenhuma citação no texto.

MARTINEZ, A.; FERNáNDEZ, E. *Learning ROS for Robotics Programming*. 1. ed. [S.l.]: Packt Publishing, 2013. Nenhuma citação no texto.

MYER-BAESE, U. *Digital Signal Processing with Field Programmable Gate Arrays*. 4. ed. Nova York: Springer, 2014. Citado na página 2.

NETO, N. P. *Biblioteca projeto interfacesocket*. 2021. Disponível em: <https://github.com/NestorDP/libinterfacesocket>. Nenhuma citação no texto.

NETO, N. P. *interface_socketr*. 2021. Disponível em: <https://github.com/NestorDP/interface_socket>. Nenhuma citação no texto.

NETO, N. P. *interface_socket_server*. 2021. Disponível em: <https://github.com/NestorDP/interface_socket_server>. Nenhuma citação no texto.

OKANE, J. M. *A Gentle Introduction to ROS*. 1. ed. Columbia: University of South Carolina, 2016. Nenhuma citação no texto.

PYO, Y. et al. *ROS Robot Programming*. Nova York: ROBOTIS, 2017. Citado na página 3.

ROBIN, S. *rsyocto*. 2022. Disponível em: <https://github.com/robseb/rsyocto>. Nenhuma citação no texto.

ROCKETBOARDS.ORG. *Embedded Linux Beginners Guide*. 2015. Disponível em: <https://rocketboards.org/foswiki/Documentation/EmbeddedLinuxBeginnerSGuide>. Acesso em: 5 outubro 2021. Nenhuma citação no texto.

ROS. *ROS - Robot Operating System*. 2011. Open Source Robotics Fundation. Disponível em: <https://www.ros.org/>. Acesso em: 22 outubro 2021. Nenhuma citação no texto.

ROS. *TCPROS*. 2013. Open Source Robotics Fundation. Disponível em: <http://wiki.ros.org/ROS/TCPROS>. Acesso em: 22 outubro 2019. Nenhuma citação no texto.

ROS. *UDPROS*. 2013. Open Source Robotics Fundation. Disponível em: <http://wiki.ros.org/ROS/UDPROS>. Acesso em: 22 outubro 2019. Nenhuma citação no texto.

ROS. *Metapackages*. 2014. Disponível em: <http://wiki.ros.org/Metapackages>. Acesso em: 21 setembro 2021. Nenhuma citação no texto.

ROS. *catkin*. 2017. Open Source Robotics Fundation. Disponível em: <http://wiki.ros.org/catkin>. Acesso em: 22 outubro 2019. Nenhuma citação no texto.

ROS. *srv*. 2017. Open Source Robotics Fundation. Disponível em: <http://wiki.ros.org/srv>. Acesso em: 22 outubro 2019. Nenhuma citação no texto.

ROS. *Introduction*. 2018. Open Source Robotics Fundation. Disponível em: <http://wiki.ros.org/ROS/Introduction>. Acesso em: 20 julho 2021. Nenhuma citação no texto.

ROS. *Master*. 2018. Open Source Robotics Fundation. Disponível em: <http://wiki.ros.org/Master>. Acesso em: 22 outubro 2019. Nenhuma citação no texto.

ROS. *Nodes*. 2018. Open Source Robotics Fundation. Disponível em: <http://wiki.ros.org/Nodes>. Acesso em: 22 outubro 2018. Nenhuma citação no texto.

ROS. *catkin/package.xml*. 2019. Disponível em: <http://wiki.ros.org/catkin/package.xml>. Acesso em: 21 setembro 2021. Nenhuma citação no texto.

ROS. *msg*. 2019. Open Source Robotics Fundation. Disponível em: <http://wiki.ros.org/msg>. Acesso em: 22 outubro 2019. Nenhuma citação no texto.

ROS. *Pckages*. 2019. Open Source Robotics Fundation. Disponível em: <http://wiki.ros.org/Packages>. Acesso em: 20 julho 2021. Nenhuma citação no texto.

TERASIC. *DE10-Nano User Manual*. 2.2. ed. [S.l.], 2020. Rev. B2/C Hardware. Nenhuma citação no texto.

XILINX, A. *Field Programmable Gate Array (FPGA)*. 2022. Disponível em: <https://www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html>. Acesso em: 5 março 2022. Nenhuma citação no texto.

YAMASHINA, K. et al. Proposal of ros-compliant fpga component for low-power robotic systems: case study on image processing application. *2nd International Workshop on FPGAs for Software Programmers (FSP 2015)*, p. 62–67, 2015. Disponível em: <https://arxiv.org/pdf/1508.07123.pdf>. Nenhuma citação no texto.