

Tutorial Petalinux

1st Ivan Diniz Dobbin
Universidade de Brasília (UnB)
Brasília, Brasil
ivandinizdobbin2@gmail.com

Resumo—This document is a model and instructions for L^AT_EX. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. *CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.

Index Terms—component, formatting, style, styling, insert

I. INTRODUÇÃO

Um sistema embarcado é um sistema computacional construído com o objetivo de desenvolver um propósito específico. Estes sistemas possuem mais restrições que um computador normal, como memória e processamento reduzidos [1] e [2].

A utilização de sistemas operacionais embarcados Linux tem se tornado cada vez mais popular. Com a justificativa principal sendo o aumento da complexidade dos equipamentos, seguindo a lei de Moore [3]. Assim, de acordo com [3], existem vantagens ao se utilizar esses sistemas operacionais: como *scheduler*, estrutura para wifi, bluetooth, portabilidade para uma grande quantidade de arquitetura de processadores, *open source*, uma comunidade extremamente ativa, etc. Contudo, esta opção pode não se adequar aos requisitos do sistema, visto que consome mais recursos do que sistemas operacionais de tempo real (RTOS) como VxWorks e QNX [3].

A Xilinx (adquirida pela AMD [4]) é responsável por desenvolver plataformas de processamento altamente adaptáveis e flexíveis, as quais permitem rápida inovação em uma variedade de tecnologias. Ela desenvolveu um Kit de Desenvolvimento de Software (SDK) Linux embarcado voltado para projetos de FPGA e projetos de SoC (system-on-a-chip) baseados em FPGA, chamado de PetaLinux [5]. Essa ferramenta pode ser usada na geração de diferentes versões de linux embarcado para FPGAs. Sendo uma de suas vantagens a possibilidade de transformar os IP cores criados, por indivíduos e grupos, em drivers de dispositivo dentro do sistema operacional gerado.

Existem outras ferramentas para a criação de sistemas operacionais embarcados como o Yocto Project [6] e o Ubuntu Core [7], porém este trabalho focará apenas no PetaLinux.

De acordo com o [5], o PetaLinux é uma ferramenta de desenvolvimento que permite o seu usuário realizar todas as etapas necessários para construir, desenvolver, testar e publicar sistemas linux embarcados. Porém, a sua utilização se torna difícil pela escassez de tutoriais em português. A maioria dos tutoriais, como os vídeos de [8] e [9], assim como o manual de referência [5], se encontram em inglês.

Desta maneira, o objetivo deste trabalho é criar um tutorial em português que facilite a criação de um linux embarcado,

com os IP cores desenvolvidos, usando o PetaLinux. Assim, este trabalho será dividido nas seguintes metas:

- 1) Criar uma imagem de um linux embarcado simples utilizando a ferramenta do PetaLinux e verificar se funciona corretamente na FPGA.
- 2) Editar a imagem do sistema operacional, adicionando device drivers (sem conectar com os ips cores ainda), e verificar que funciona corretamente na FPGA.
- 3) Transformar um ip core mais simples em um driver de dispositivo para ser utilizado dentro da FPGA.
- 4) Pegar o ip core que define a rede neural (4x4x3) para o problema de classificação de padrões e transformá-lo em um driver de dispositivo.
- 5) Generalizar o processo de adição de IP Cores como drivers de dispositivo, se possível.
- 6) Documentar todo o processo para a criação do tutorial. Observação: deve ser feito paralelamente aos outras metas.
- 7) Escrever o documento, processo que também deve ser feito de maneira paralela.

Existem 9 semanas entre o dia 09/05/2024 e o dia 11/07/2024. Assim, a Tabela I mostra como se planeja desenvolver este trabalho.

Tabela I
CRONOGRAMA

Meta/Semana	1	2	3	4	5	6	7	8	9
Meta 1									
Meta 2									
Meta 3									
Meta 4									
Meta 5									
Meta 6									
Meta 7									

II. DESENVOLVIMENTO

Esta seção tem como objetivo esclarecer as ferramentas utilizadas, as condições nas quais foram usadas, o arquivo de hardware xsa, o processo de instalação, o processo de documentação, as metas concluídas e o cronograma.

A Tabela II mostra as ferramentas utilizadas e a Tab. III mostra o equipamento.

Quando possível, foram utilizadas versões atualizadas das ferramentas. Sendo o Vivado/Vitis a exceção, pois esta é a mesma versão que se encontra nos laboratórios. Reduzindo assim, alguns erros de compatibilidade entre versões.

Tabela II
FERRAMENTAS

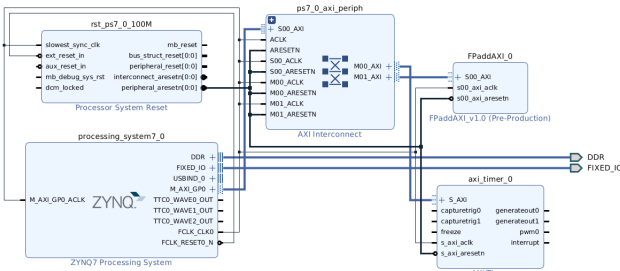
Nome	Versão	Justificativa
Petalinux	2024.1	Versão mais atual.
Ubuntu	Ubuntu 22.04.4 LTS	Versão 22.04 mais atual, apesar de não ter suporte.
Virtual Box	7.0.18	Versão mais atual
Vivado/Vitis	2019.2	Mesma versão do laboratório.

Tabela III
EQUIPAMENTO

Nome	ZedBoard Zynq Evaluation and Development Kit [10]
Versão do Arquivo	1.4
Peça	xc7z020clg484-1 [10]

Para a montagem inicial, decidiu-se utilizar um ip-core criado em sala de aula, FPaddAXI (disponível em <https://github.com/darmsDD/Guia-Petalinux>). A função deste IP é realizar soma em ponto flutuante com 27 bits em vez de 32. Observe a Fig 1, onde é mostrado o design dos blocos.

Figura 1. Design dos blocos



Pode-se notar que foi necessário a adição do IP AXI Interconnect para permitir a conexão do ZYNQ com outros IPs.

ZYNQ é um *all programmable system-on-chip* (APSoC). É um sistema que possui os componentes de um computador (processadores ARM Cortex-A9 de núcleo duplo) e tecnologia FPGA [11]. Permitindo diferentes abordagens para o desenvolvimento de um projeto.

Utilizando o ZYNQ, foi criado um arquivo.xsa. Este é o arquivo utilizado para a configuração do hardware no processo de criação da *build* do Petalinux.

O processo de instalação do Petalinux e criação da *build* foi realizado seguindo as instruções de [5] e [9]. A partir desse manual e vídeo, foi possível criar um guia simplificado em português, disponível em <https://github.com/darmsDD/Guia-Petalinux>. Mostrando o desenvolvimento das metas 6 e 7.

A primeira *build* funcional foi criada em uma máquina virtual contendo Ubuntu, dentro de um Windows 11. Houveram diversos problemas, com a *build* falhando mesmo com todos os pacotes mencionados instalados. A causa do problema pode ser a utilização de versões muito distantes entre o Petalinux e o Vivado/Vitis, além de usar uma versão do Ubuntu sem

suporte oficial. Porém, esta primeira *build* foi perdida, pois o Windows 11 se corrompeu.

A segunda *build* funcional foi criada em um sistema Ubuntu, na mesma versão da máquina virtual, instalado diretamente no SSD da máquina. Por utilizarem a mesma versão de sistema operacional, a solução dos problemas da *build* 1 puderam ser reutilizados. Facilitando o desenvolvimento, além de mostrar o funcionamento da ferramenta em 2 sistemas diferentes (na máquina virtual e diretamente no ssd). É importante notar que apenas a segunda *build* foi testada na zedboard.

Em relação as metas, afirma-se que a meta 1 foi concluída com sucesso (Fig. 2), e as metas 6 e 7 tiveram um bom andamento com a construção do manual. Porém, as metas 2 e 3 tiveram baixo progresso, pois apesar do IP FPaddAXI estar disponível no kernel, isto é, aparece na árvore de dispositivos, não foi possível descobrir como utilizá-lo até o momento. Existem 4 semanas entre o dia 11/06/2024 e o dia 11/07/2024. Assim, a Tabela IV mostra como se planeja desenvolver este trabalho.

Figura 2. Primeira versão do petalinux na zedboard

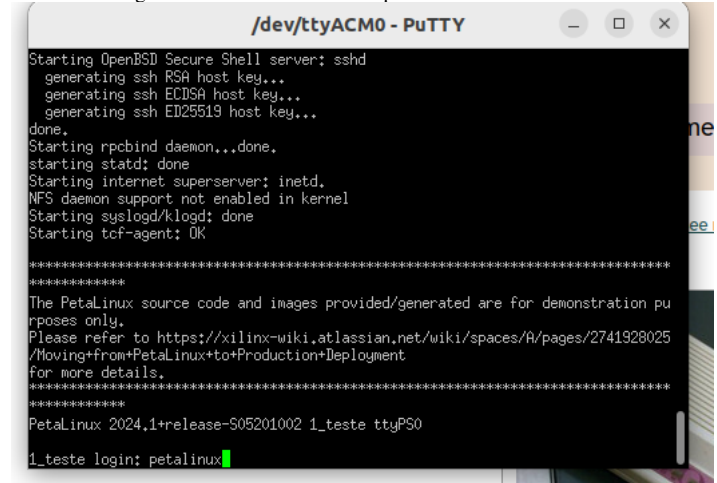


Tabela IV
CRONOGRAMA

Meta/Semana	1	2	3	4
Meta 2				
Meta 3				
Meta 4				
Meta 5				
Meta 6				
Meta 7				

III. RESULTADOS

IV. CONCLUSAO

REFERÊNCIAS

- [1] R. Hu, *Embedded systems architecture and software design*, english edition ed. [Seattle, WA]: Kindle Direct Publishing, 2022, e-book (354 p.) Edição Kindle.
- [2] E. White, *Making embedded systems*. California: O'Reilly, 2012, e-book (506 p.) Edição Kindle.

- [3] F. V. Chris Simmonds, *Mastering Embedded Linux Programming Third Edition*, 3rd ed. Birmingham: Packt Publishing Ltd, 2021, pdf (758 p.).
- [4] “AMD adquire a Xilinx criando a líder de alto desempenho e computação adaptável da indústria,” Advanced Micro Devices, Inc, 2024, acessado em 9 de maio de 2024. [Online]. Available: <https://www.amd.com/pt/corporate/xilinx-acquisition.html>
- [5] *PetaLinux Tools Documentation*, 2023rd ed., AMD, out. 2018, disponível em <https://docs.amd.com/r/en-US/ug1144-petalinux-tools-reference-guide>.
- [6] “The Yocto Project it’s not an embedded linux distribution, it creates a custom one for you.” The Linux Foundation, 2024, acessado em 9 de maio de 2024. [Online]. Available: <https://www.yoctoproject.org/>
- [7] “Ubuntu Core the embedded linux os for devices,” Canonical Ltd, 2024, acessado em 9 de maio de 2024. [Online]. Available: <https://ubuntu.com/core>
- [8] Leonardo, “Embedded Linux Introduction #01,” <https://youtu.be/92-uLpWIRaI?si=dB37s9NWesijq9Qa>, Out. 2015, acessado em 9 de maio de 2024.
- [9] P. Salmony, “Embedded Linux + FPGA/SoC (Zynq Part 5) - Phil’s Lab,” https://youtu.be/OfozFBfvWeY?si=J_sFDEcF993v2QBi, Mar. 2023, acessado em 9 de maio de 2024.
- [10] “Zedboard,” <https://www.avnet.com/wps/portal/us/products/avnet-boards/avnet-board-families/zedboard/>, AVNET, 2024, acessado em 11 de junho de 2024.
- [11] D. Horn, “What is zynq?” <https://digilent.com/blog/what-is-zynq/#:~:text=Zynq%20is%20an%20innovative%20and,a%20faster%20time%20to%20market.,> AVNET, 2023, acessado em 11 de junho de 2024.