

Tutorial Petalinux

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

Resumo—A utilização de sistemas operacionais embarcados está cada vez mais popular pelo aumento da complexidade dos sistemas embarcados. A ferramenta Petalinux, criada pela Xilinx, tem o papel de criar sistemas operacionais embarcados, porém as informações sobre o uso dessa ferramenta são escassas. O objetivo deste trabalho é criar um tutorial em português para facilitar a criação de linux embarcado. Para isso, foi realizada uma investigação sobre a ferramenta, isto é, como ela funciona e relatado o passo a passo. A partir desse processo foram criados vídeos, scripts e um guia. Estes artefatos ajudarão os usuários a aprender como usar a ferramenta, principalmente usuários iniciantes, além de automatizar alguns processos.

Palavras chave:—Zedboard, FPGA, PetaLinux, Guia, Tutorial

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 necessárias 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 as 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.

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.

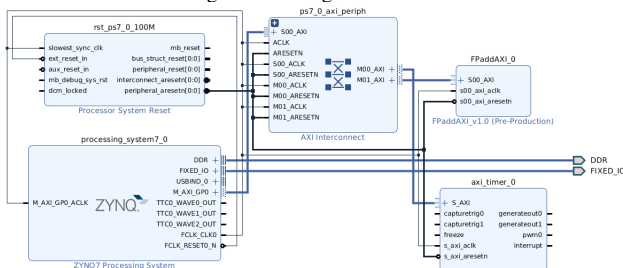
Tabela III
EQUIPAMENTO

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

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.

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 às 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

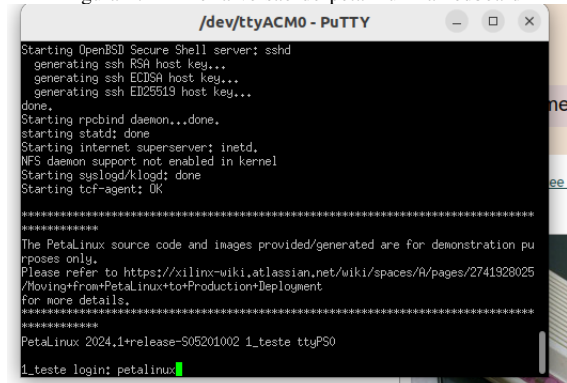


Tabela IV
CRONOGRAMA

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

A. Atualização de ferramentas

Dada as dificuldades encontradas, tomou-se a decisão de atualizar a versão do vivado/vitis. A versão escolhida foi a 2024.1, por ser a versão mais recente.

Com esta decisão, o processo de instalação de build se tornou muito mais estável e fácil de ser replicado.

B. Criação de scripts e vídeos

Percebeu-se que a instalação do petalinux e a criação de projetos não era trivial. Desta maneira, decidiu-se focar na criação de scripts e vídeos, que unidos ao guia, facilitassem o uso da ferramenta.

A gravação dos vídeos foi feita usando a ferramenta de captura do windows junto com a ferramenta *Microsoft Clipchamp* para a edição e os scripts são do tipo bash, isto é, `#!/bin/bash`.

Foi necessário remanejar as metas atuais para melhor atender as novas necessidades. Com as metas 2 a 5 sendo substituídas. Veja abaixo as novas metas e o novo cronograma que possui uma semana a mais.

- 1) Criar uma imagem de um linux embarcado simples utilizando a ferramenta do petalinux e verificar se funciona corretamente na FPGA.
- 2) Documentar todo o processo para a criação do tutorial em vídeo. Observação: deve ser feito paralelamente as outras metas.
- 3) Escrever o documento, processo que também deve ser feito de maneira paralela.
- 4) Criar scripts.
- 5) Criar o tutorial em vídeo.

Tabela V
CRONOGRAMA

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

III. RESULTADOS

Essa seção será dividida entre as metas.

A. Meta 1:

No início do desenvolvimento foi possível concluir a meta 1, como mostrado na Fig. 2. Contudo, o processo não podia ser replicado sem dificuldades. Assim, as metas 2, 3, 4 e 5 visaram resolver esta questão.

B. Meta 2:

Durante todo o processo de desenvolvimento, o guia foi escrito. Estando concluído e disponível para leitura no repositório <https://github.com/darmsDD/Guia-Petalinux/tree/main>. Este guia começa mostrando como instalar uma máquina virtual, segue com a instalação do vitis/vivado e depois a instalação do petalinux. Também é detalhado o passo a passo para a criação de projetos e seu envio para a fpga.

C. Meta 3:

A escrita deste documento foi realizada durante todo o cronograma. O documento relata o processo de criação do guia, scripts e vídeo. Estando concluído e disponível para leitura no repositório <https://github.com/darmsDD/Guia-Petalinux/tree/main>.

D. Meta 4:

Com a redução de erros pela troca de ferramenta, foi possível automatizar alguns processos. Assim, foram criados diversos scripts, com o intuito de reduzir o trabalho do usuário. Os scripts podem ser acessados no repositório <https://github.com/darmsDD/Guia-Petalinux/tree/main>. Lista de scripts e suas funções:

- `vivado_installation.sh`: instala o vivado/vitis.
- `bin_alias_creation.sh`: cria o alia *vivado*, permitindo sua execução pelo terminal.
- `petalinux_installation.sh`: instala o petalinux.
- `create_petalinux_macro.sh`: gera a macro PETALINUX, permitindo a execução de suas funções.
- `create_peta_project.sh`: Cria um projeto petalinux.
- `build_peta_project.sh`: Cria a imagem do petalinux que será inserida na fpga.
- `send_to_fpga.sh`: envia a imagem para a fpga

E. Meta 5:

Com o desenvolvimento da meta 1 e a troca de ferramentas, foi possível criar vários vídeos demonstrando como instalar e utilizar o petalinux. Os vídeos estão disponíveis em https://youtube.com/playlist?list=PLfWPlvC30l0he-1B10K9eKANaNRX8mprM&si=cwa-_l5YKKlfev68. Lista de vídeos e suas funções:

- Instalação do Petalinux: mostra como instalar o petalinux.
- Criando um projeto petalinux: mostra como criar um projeto petalinux.
- Fazendo a build e testando no QEMU: mostra como criar a imagem de um projeto petalinux e testar no simulador QEMU.
- Enviando por jtag o petalinux: mostra como enviar a imagem do petalinux criada para dentro da zedboard via jtag.
- Formatando o cartão sd: mostra como formatar um cartão sd, que será usado para receber uma imagem petalinux.
- Copiando a imagem para o cartão sd: mostra como copiar a imagem para dentro do cartão SD.
- Mostrando o petalinux dentro da zedboard: mostra a imagem do petalinux dentro da zedboard usando putty.

IV. CONCLUSÃO

O petalinux é uma ferramenta que ainda tem poucos recursos disponíveis, mesmo em inglês. O objetivo deste trabalho era criar um tutorial em português que facilitasse a utilização desta ferramenta.

Pode-se afirmar que o objetivo foi concluído com a criação deste documento, do guia, dos scripts e dos vídeos. Todos estes artefatos facilitam o uso da ferramenta, principalmente para usuários sem nenhuma experiência.

É importante ressaltar algumas dificuldades deste trabalho. Sendo as 2 principais: a falta de compatibilidade entre versões, que gerou erros inesperados e de difícil replicação e o sistema operacional da máquina do autor corromper 4 vezes durante a escrita deste documento, o que criou problemas na recuperação de arquivos e gasto de tempo em reinstalações.

Por fim, espera-se que os próximos trabalhos avancem no desenvolvimento de *device drivers* para a utilização de IP-cores gerados, criem o script para formatação e uso do cartão SD e refinem os scripts já criados.

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 acquire 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_sFDEcF993v2QBj, 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.