

# PRÁTICA DE ELETRÔNICA DIGITAL 1 (FGA0071)

Aula 03 - Vivado ML Edition

Prof. Marcelino Andrade

Semestre 2023.2  
Universidade de Brasília  
UnB



# Sumário

- 1 Instalação do Ambiente
- 2 Criação do Projeto Vivado
- 3 Criação do Código Fonte
- 4 Simulando o Programa
- 5 Analise, Síntese e Implementação
- 6 Constrs, Bitstream e Programando
- 7 Concluindo



# Página de Download e Manual de Instalação

🏠 / Adaptive Computing Support / Downloads

## Downloads

Licensing Help

NIC Software & Drivers

Vivado (HW Developer)

Vitis (SW Developer)

Vitis Embedded Platforms

Power Design Manager

Alveo Packages

PetaLinux

Device Models

Documentation Navigator

## Version

- 2023.1
- 2022.2
- 2022.1
- [Vivado Archive](#)
- [ISE Archive](#)
- [CAE Vendor Libraries Archive](#)

### Vivado ML Edition - 2023.1 Full Product Installation

#### Important Information

Vivado™ ML 2023.1 is now available for download:

- Average QoR Improvement of 8% for Versal™ Adaptive SoCs and 13% for UltraScale+ FPGAs using Intelligent Design Runs
- Extending multithreading support for bitstream generation for Versal devices
- Enhancements in Report QoR Assessment (RQA)
- Power Design Manager (PDM) now a part of Unified Installer – Separate from Vivado tool
- Added support for Versal HBM devices in Power Design Manager (PDM)

We **strongly recommend** to use the web installers as it reduces download time and saves significant disk space.

Please see [Installer Information](#) for details.

#### Note:

- Download verification is only supported with Google Chrome and Microsoft Edge web browsers.
- Vivado ML 2021.1 and later versions require upgrading your license server tools to the Flex 11.17.2.0 versions.

#### Download Includes

Download Type

Last Updated

Answers

Documentation

Support Forums

Vivado ML Edition

Full Product Installation

May 17, 2023

[2023.x - Vivado Known Issues](#)

[Release Notes](#)  
[OS Support Update](#)  
[What's New in Vivado](#)

[Installation and Licensing](#)

Escolha a versão do sistema Windows ou Linux e sigam o material do professor Guilherme Palermo, da Faculdade UnB-Gama, para a instalação. Clique aqui!

Figura 1. Página de Download



# Criação do Projeto

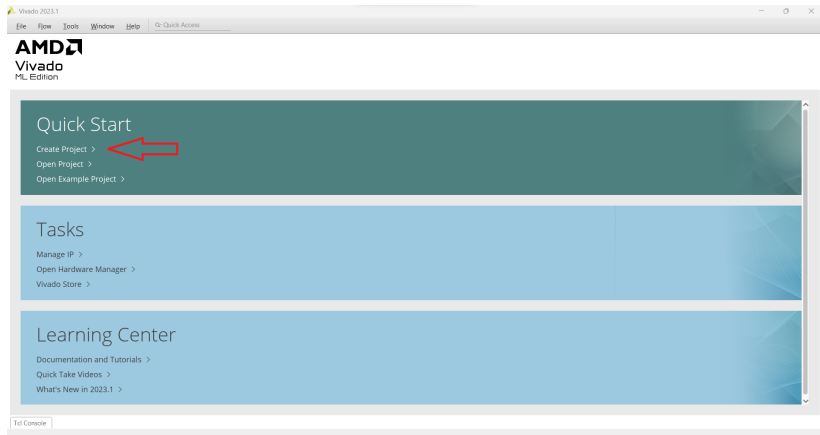


Figura 2. Tela Inicial de Criação de Projeto



# Passos de Criação ...

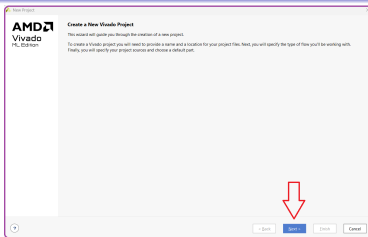


Figura 3. Iniciando a Criação

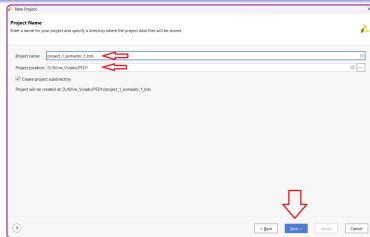


Figura 4. Escolhendo Nome e Local

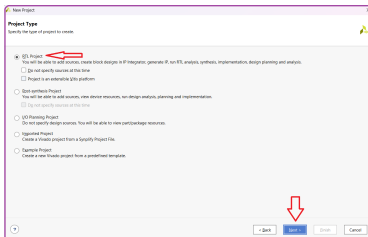


Figura 5. Escolhendo o Tipo

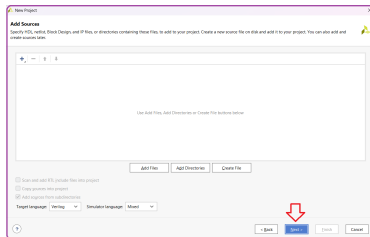


Figura 6. O Código Fonte - Fazer depois!



# Passos de Criação!

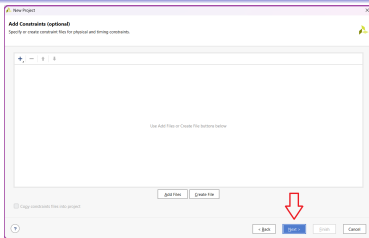


Figura 7. Relacionamento Placa e Código - Depois

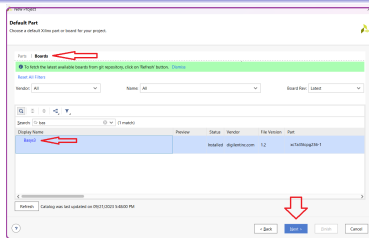


Figura 8. Escolha da Placa FPGA

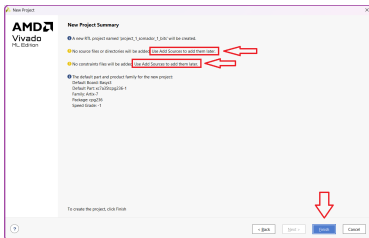


Figura 9. Resumo e Partes a Fazer

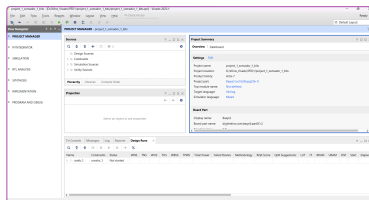


Figura 10. Projeto Criado



# Iniciando um Código Vazio

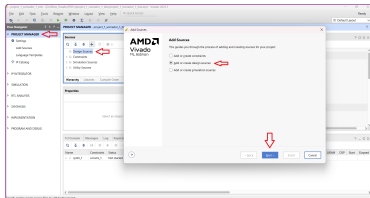


Figura 11. Design Sources

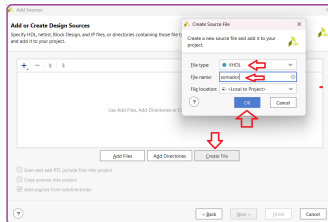


Figura 12. Escolha da Placa FPGA

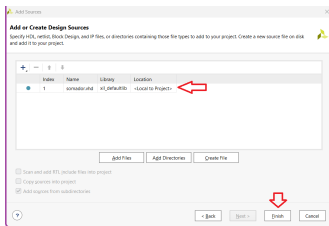


Figura 13. Confirmado o Arquivo VHDL

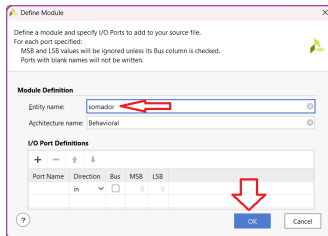


Figura 14. Definindo I/O - Opcional



# Carregando o Programa

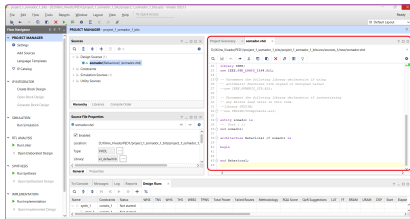


Figura 15. Arquivo VHDL Vazio

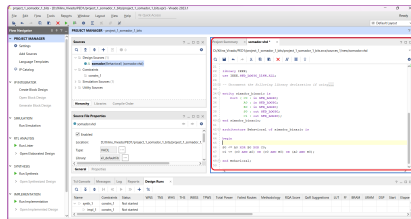


Figura 16. Arquivo VHDL - Somador Completo





# Simulação com Entradas Constantes

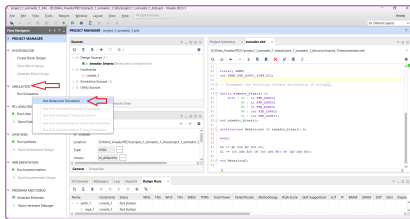


Figura 17. Iniciando Simulação

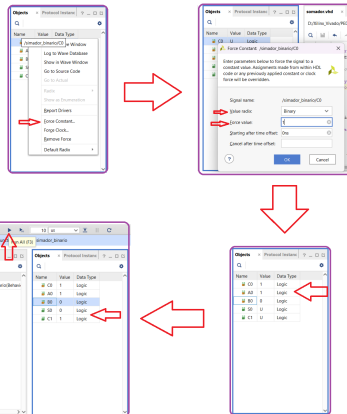


Figura 18. Carregando Entradas e Executando



# Desenvolvimento do Projeto

## 1 RTL Analysis (Análise RTL):

Etapa de avaliação de como registros e transferências de dados ocorrem em um circuito digital para garantir que o design atenda aos requisitos funcionais e temporais.

## 2 Synthesis (Síntese):

Tradução do projeto descrito em linguagem de hardware (VHDL ou Verilog) em elementos lógicos específicos que podem ser usados em um dispositivo programável, como um FPGA.

## 3 Implementation (Implementação):

Conversão do projeto sintetizado em um layout físico para um dispositivo específico, envolvendo alocação de recursos, roteamento de conexões e otimizações de desempenho e consumo de energia.

## 4 Bitstream Generation (Geração de Bitstream):

Criação de um arquivo binário (bitstream) que contém as informações necessárias para programar um dispositivo programável com o design implementado, permitindo que ele execute a função desejada.



## Síntese (Synthesis):

- **O que é:** A síntese é o processo de tradução de um projeto de hardware descrito em uma linguagem de descrição de hardware de alto nível, como VHDL ou Verilog, em um conjunto de elementos lógicos específicos que podem ser implementados em um dispositivo lógico programável (FPGA ou ASIC).
- **Objetivo:** A principal finalidade da síntese é criar uma representação eficiente em termos de hardware a partir de uma descrição de alto nível do circuito. Isso envolve mapear funções lógicas e comportamentos descritos no código em elementos de hardware concretos, como portas lógicas, flip-flops e outros blocos lógicos disponíveis no dispositivo alvo.
- **Nível de Abstração:** A síntese opera em um nível mais alto de abstração, concentrando-se na funcionalidade do circuito e na lógica subjacente, sem se preocupar com detalhes de implementação específicos.
- **Exemplo:** Se você projetou uma calculadora em VHDL, a síntese traduzirá a lógica de como a calculadora realiza as operações matemáticas em elementos lógicos do FPGA.



# Implementação (Implementation):

- **O que é:** A implementação refere-se à etapa em que o design sintetizado é convertido em um layout físico que pode ser implementado em um dispositivo específico.
- **Objetivo:** O objetivo principal é tornar o projeto lógico em algo que possa ser fisicamente criado em um chip de silício ou programado em um dispositivo programável, como um FPGA. Isso inclui a alocação de recursos, a configuração das conexões entre os componentes lógicos e otimizações para atender a requisitos de desempenho e consumo de energia.
- **Nível de Abstração:** A implementação lida com detalhes físicos, como a colocação e a rotação de transistores em um chip de silício ou a definição de como os elementos lógicos do FPGA estão conectados entre si.
- **Exemplo:** Se a síntese criou uma descrição lógica de uma unidade de processamento, na implementação, a ênfase está na organização física de uma unidade de processamento, incluindo o roteamento de fios e a alocação de espaço para memória, registradores e etc.



# Tudo junto!

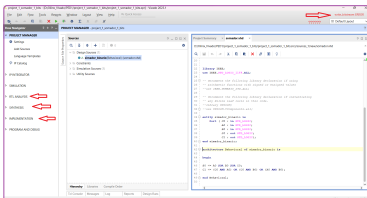


Figura 19. Campos de Ação

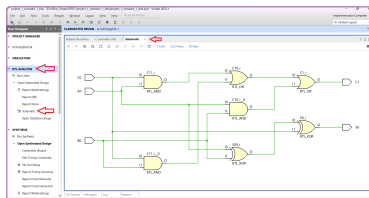


Figura 20. RTL Analysis

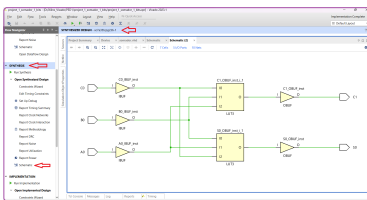


Figura 21. Synthesis

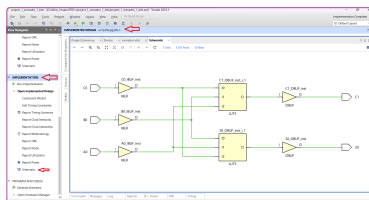


Figura 22. Implementation



# Processo de Gravação

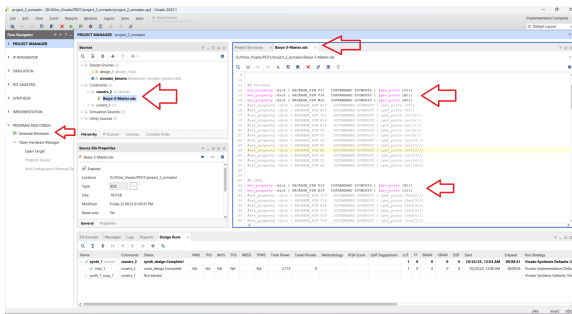


Figura 23. Gerando Bitstream

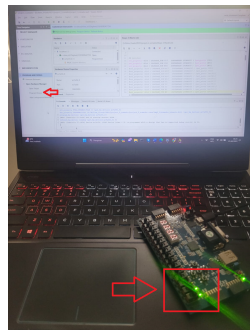
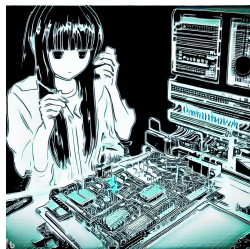


Figura 24. Placa Gravada

# Obrigado pela Atenção!



Prof. Marcelino Andrade  
*andrade@unb.br*

