

# Softcores

## Sistemas embarcados II

Prof. Renan Augusto Starke

Instituto Federal de Santa Catarina – IFSC  
Campus Florianópolis  
[renan.starke@ifsc.edu.br](mailto:renan.starke@ifsc.edu.br)

20 de julho de 2016



**INSTITUTO FEDERAL**  
**SANTA CATARINA**

Ministério da Educação  
Secretaria de Educação Profissional e Tecnológica  
**INSTITUTO FEDERAL DE SANTA CATARINA**

- ▶ Um sistema embarcado poder ser modelado, implementado e sintetizado através de uma linguagem de descrição de hardware
- ▶ No nosso caso: VHDL
- ▶ Construir um sistema embarcado com FPGA envolve:
  - análise dos requisitos do sistema
  - projeto de hardware
  - projeto de software

# Sistema com processador/microcontrolador

## Sistema básico:

- ▶ Processador
- ▶ Memória
- ▶ Entrada e saída
  - Pinos de propósito geral
  - Comunicação
  - ...
- ▶ Temporização
- ▶ Programação e depuração

# Sistema com Dispositivo Lógico Programável (PLD)

## Sistema básico:

- ▶ Componentes usuário
- ▶ Componentes licenciados (IPs)
- ▶ Máquinas de estado
- ▶ Entrada e saída
  - Pinos de propósito geral
  - Comunicação
  - ...
- ▶ Temporização
- ▶ Síntese e simulação

## Sistema básico:

- ▶ Processador (Softcore – IP)
  - Programação e depuração
- ▶ PLD
  - Componentes usuário
  - Componentes licenciados (IPs)
  - Temporização
  - Síntese e simulação
- ▶ Entrada e saída

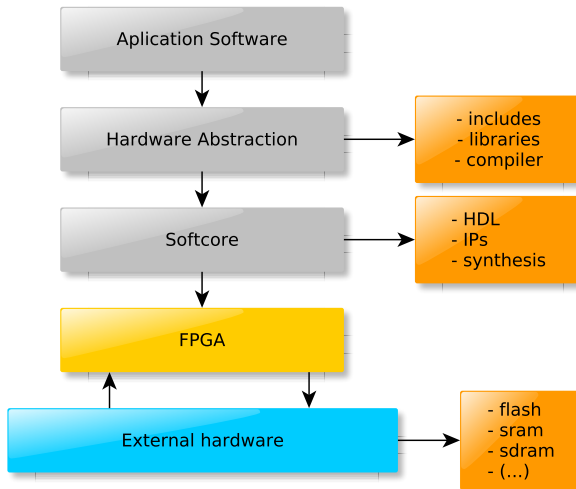
## Questões:

- ▶ Qual é o desempenho computacional que a aplicação requer?
- ▶ Quanto de banda ou *throughput* a aplicação requer?
- ▶ Quais são as interfaces necessárias?
- ▶ A aplicação necessita de *software multithreaded*?

## Baseando-se nestas questões:

- ▶ Que tipo de processador será necessário?
- ▶ Quais componentes e de que tipo.
- ▶ É necessário o uso de sistema operacional de tempo real?
- ▶ Lógica de aceleração para:
  - DMA?
  - Instruções especiais de DSP?

# Fluxo de desenvolvimento



- ▶ Componentes necessários para o sistema:
  - Núcleo
  - Memória interna de dados
  - Memória interna de instruções
  - Entrada e saída
  
- ▶ Outros componentes de hardware:
  - Conexão com a entrada e saída: botões, LEDs, ...
  - Hardware de comunicação (UART, Ethernet, ...)
  - Filtros de DSP
  - Multiplicadores
  - Unidades MAC
  - ...

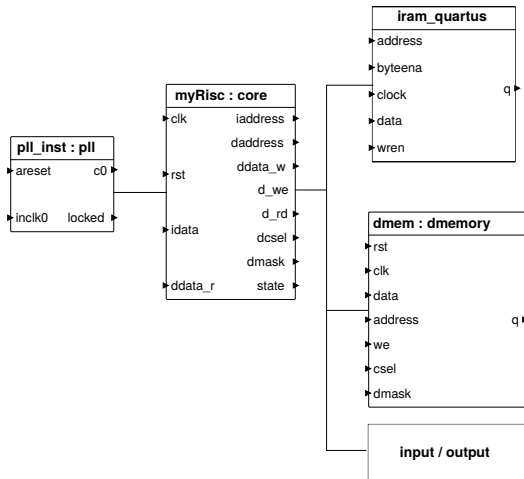


## RISC-V

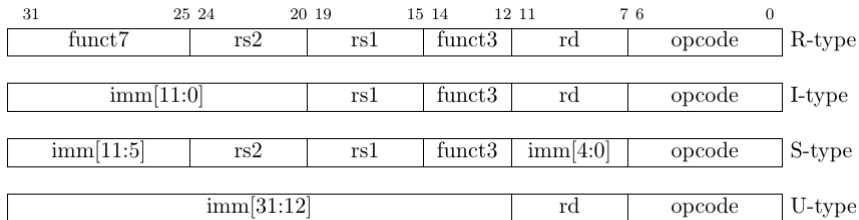
RISC-V é uma nova arquitetura de conjunto de instruções (ISA) que foi originalmente projetada para dar suporte à comunidade científica em pesquisas voltadas a arquiteturas de computadores e também para ensino.



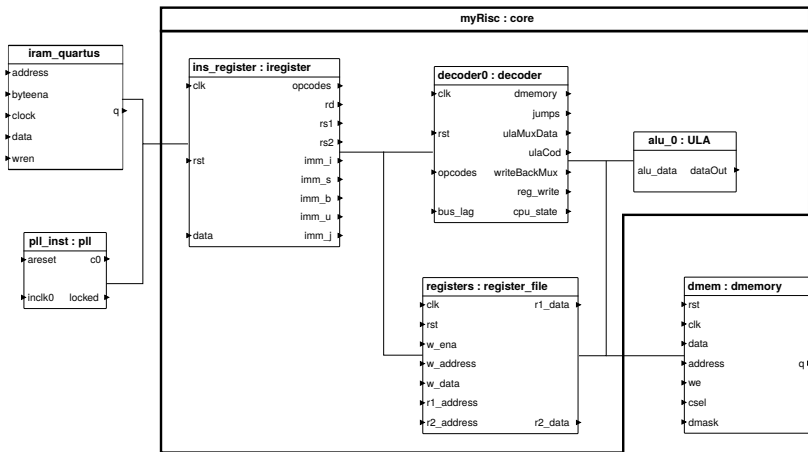
# Componentes básicos



# Componentes básicos



# Componentes básicos



- ▶ Clonar `https://github.com/xtarke/riscv-multicycle.git`
- ▶ Projeto de hardware:
  - Abrir o projeto no diretório `riscv-multicycle/sint/de10_lite`
  - Sintetizar o hardware e gravar configuração na FPGA
- ▶ Software:
  - Eclipse - File new Project - C/C++ - MakeFile with Existing Code
  - Browse - Selecionar dir `riscv-multicycle/tests/` - Toolchain (none)
  - Clicar c/ direito sobre o projeto - Show in Local Terminal here
  - Digitar Make
- ▶ Gravar Software:
  - Quartus: Tools - In System memory content editor
  - Edit - Import file from file
  - Selecionar arquivo `quartus.hex` no dir `riscv-multicycle/tests/`

- ▶ Faça a integração do conversor decimal para 7-segmentos
- ▶ Desenvolva um contador de 16-bits em software mostrando os valores nos displays de 7-segmentos.