

Projeto Overview

O projeto final da disciplina deve ser um tutorial com alguma relação aos objetivos de aprendizagem da matéria:

- Formular soluções que satisfazem requisitos de hardware e software de projetos com FPGA-SoC (System-on-a-chip)
- Integrar em um protótipo solução para um sistema embarcado com requisitos de processamento e/ou tempo real via FPGA-SoC
- Interfacear diferentes módulos em um sistema embarcado (processadores, firmware e sistema operacional)

Alguns exemplos de áreas que podem ser atacadas:

1. Aceleração/ implementação de algum algoritmo em hardware
 - processamento de dados, FFT, compressão, criptografia, ...
 - HLD/ HLS/ OpenCL/ FPGA Amazon
2. Comparação de performance entre diferentes tecnologias
 - SoC vs GPU vs FPGA vs uC
3. Sistema operacional
 - Escalonador real time kernel linux, Android, RTOS embarcado

Tecnologias/ Ferramentas

A seguir uma lista de tecnologias que podem ser estudadas no tutorial:

- HDL (VHDL/Verilog)
- [Adicionar uma instrução customizada ao NIOS](#)
- Platform designer
 - Criar um sistema para controlar um dos robôs de robótica
- Criar um periférico para interfacear com o mundo externo (ler teclado/ motor/ fita de Led/ ...)

- [High Level Synthesis \(HLS\)](#)
- Criar um periférico que acelera uma função ([example](#))
- 👍 [OpenCL](#)
- Criar um hardware que acelera uma função [Terasic Manual](#) ([example](#))
- Linux
- [real time](#) / otimização energética / 👍 [boot time](#) / aplicações / 👍 [Android](#) / openCL

Hardwarees

Temos os seguintes kits de desenvolvimento disponível:

A seguir, eu tentei resumir os hardwarees disponíveis no Insper e as respectivas tecnologias que podem ser utilizados com ele

Kit	Empresa	Tecnologia	vhdl	HLS	OpenCL	Linux	OpenC
Arria 10 SoC	Intel	FPGA + ARM	x	x	x	x	x
DE10-Standard	Intel	FPGA + ARM	x	x	x	x	x
DE10-nano-soc	Intel	FPGA + ARM	x	x	x	x	x
Terasic SoC SoM	Intel	FPGA + ARM	x	x	x	x	x
DE5a-NET-DDR4	Intel	FPGA	x	x	x		x
ZedBoard	Xilinx	FPGA + ARM	x	x	x	x	x
instância F1	AWS	FPGA			x		
Jetson TK2	NVIDIA	ARM + GPU				x	x

Exemplos de temas/ coisas legais

!! : demanda uma dedicação maior

- Criando um SoftProcessor e API para controlar um Drone
- [OpenCV acelerado com OpenCL - ZedBoard](#)
- !! Criar uma aplicação com HLS/OpenCL que acelera uma função na FPGA
 - Processamento de imagem/ compressão de dados/ criptografia/ fft/ ...
- !! [Criar uma aplicação com OpenCL na AWS](#)

- [Embarcando ROS no SoC-FPGA](#) (primeiro passo para controlarmos os robôs de robótica com a FPGA)
- **!!** Usar o [LCD LT24](#) com o Linux (Comunicação ARM-FPGA)
- [Real Time kernel](#) é realmente tempo real? Estudo de latência...
- Otimizando o [boot time](#) do linux
- Executando [Android](#) na DE10-Standard
- Interface gráficas em sistemas embarcados (exe: criar um totem de pagamento)
- Device driver: Criar um driver no linux [para algum sensor de distância](#)
- **!!** Criar um periférico para controlar a fita de LED RGB e criar um driver para o Linux controlar
- BanchMark entre os diferentes kits de desenvolvimento
- Usando o [yocto](#) como alternativa ao buildroot para gerar o Linux

Rubrica

O tutorial deve ser de autoria do aluno e auto contido, publicado na wiki da disciplina. A rubrica é incremental, para tirar A precisa ter alcançado o B antes... Tutoriais em inglês são acrescidos de ½ conceito.

- A
- É um tutorial de um tema novo
- Possui um guia ao final do tutorial em como se aprofundar no tema
- Possui claro quais são os pontos críticos, e o que fazer em caso de erro
- B
- O tutorial é uma junção de outros tutoriais porém avança a onde os outros não foram
- O tutorial mescla teoria e prática de maneira aprofundada, mas sem travar o fluxo do mesmo
- C

- O tutorial é uma junção de outros tutoriais ou derivação de um exemplo já existente
- Tutorial é reproduzível (outra pessoa consegue seguir e chegar nos mesmos resultados)
- Possui um pouco de teoria, sem aprofundamento
- Possui referências externas