

Universidade Federal de Santa Maria - UFSM
Centro de Tecnologia - CT
Curso de Engenharia de Computação
ELC1054 - Projeto de Sistemas Digitais Integrados

Implementação ASIC para um Compressor JPEG



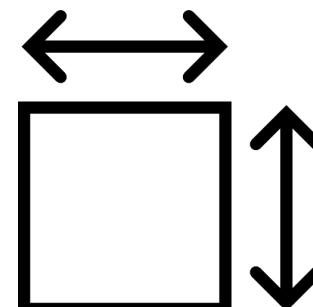
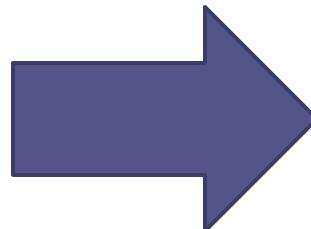
Luis Felipe de Deus – felipe.deus@ecomp.ufsm.br
Nathanael Luchetta – nathanael.luchetta@ecomp.ufsm.br
Tiago Knorst – tiago.knorst@ecomp.ufsm.br

Agenda

- Introdução
- Objetivo
- Metodologia
- Desenvolvimento
- Resultados
- Conclusões
- Trabalhos Futuros

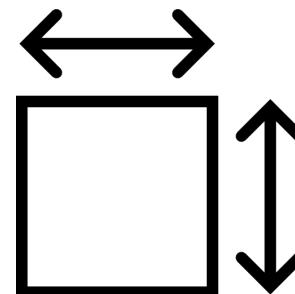
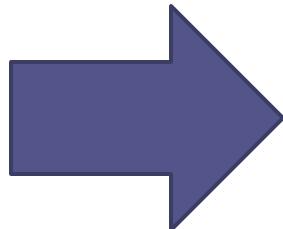
Introdução

- Sistemas Embarcados: Restrições !



Introdução

- Sistemas Embarcados: Restrições !



- Área
- Energia
- Tempo

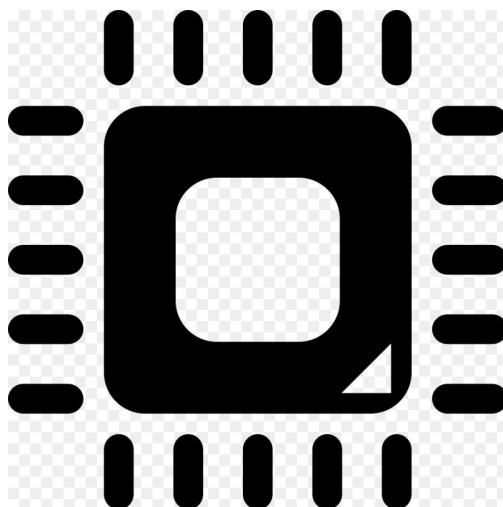
Objetivo

- Comprimir imagens do formato *bitmap* -> *JPEG*
- *.bmp (MB)* -> *.jpeg (KB)* com perdas baixas.



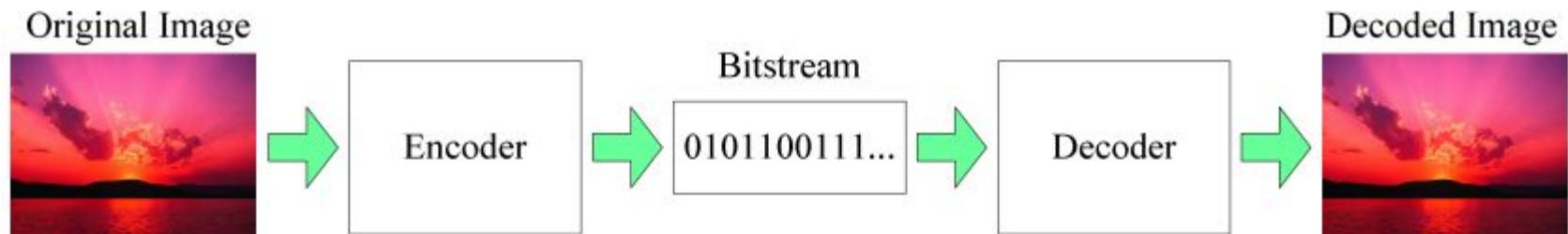
Objetivo

- Reduzir o uso de CPU e memória.



Objetivo

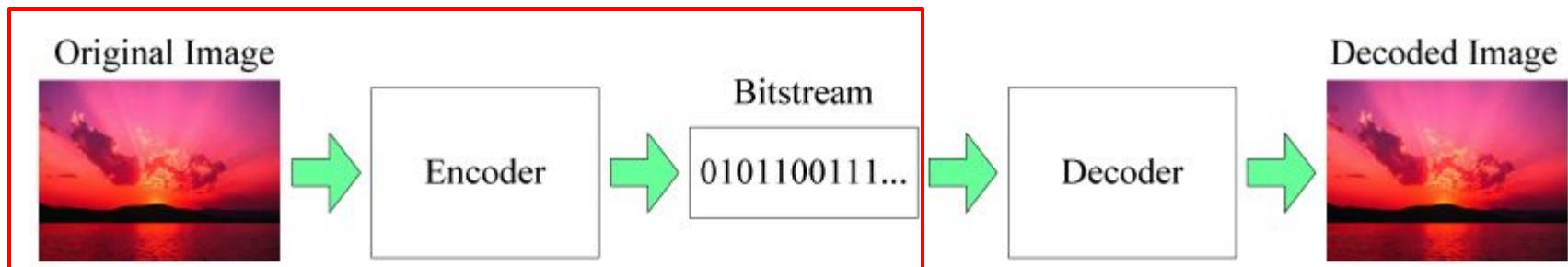
- Fluxo de codificação simplificado



Fonte: WEI, W. An Introduction to Image Compression, 2013.

Objetivo

- Fluxo de codificação simplificado



Fonte: WEI, W. An Introduction to Image Compression, 2013.

Metodologia

- Linguagens:
 - VHDL – VHSIC *Hardware Description*;
 - C;
 - Matlab.
- Ferramentas:
 - ISE Design Suite – Xilinx;
 - ModelSim – Mentor Graphics;
 - Encounter RTL – Cadence Design Systems;
 - NCLaunch - Cadence Design Systems;
 - Encounter Digital Implementation - Cadence Design Systems;
 - Matlab.

Metodologia

- *Datasets:*
 - Comparação com implementação de Krepa (2009);
- Métrica de desempenho:
 - Comparação com ARM Cortex A9;
 - Comparação com ASIC de Pastuszak (2005) ;

Metodologia

- Vetores de teste: imagem Lena.bmp

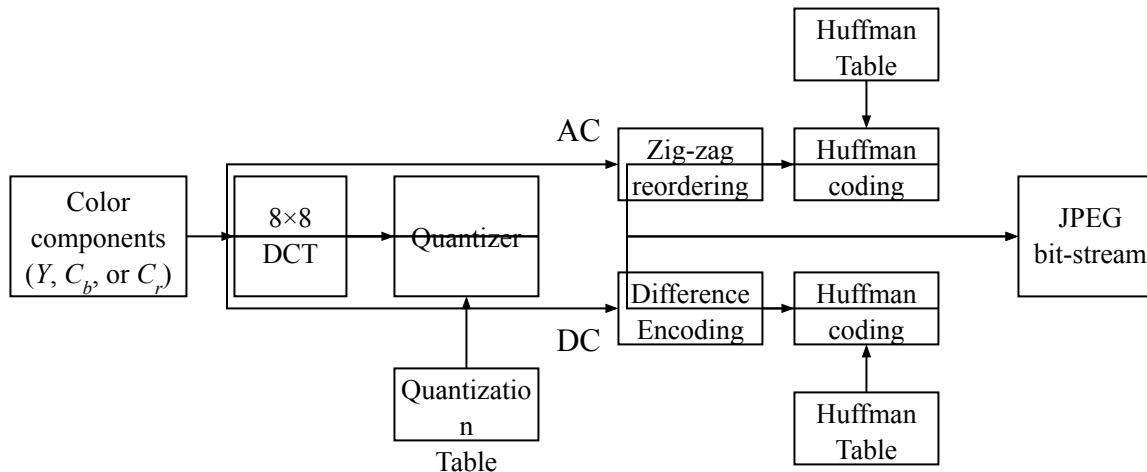
Tamanhos:

- 8x8;
- 64x64;
- 512x512.



Desenvolvimento

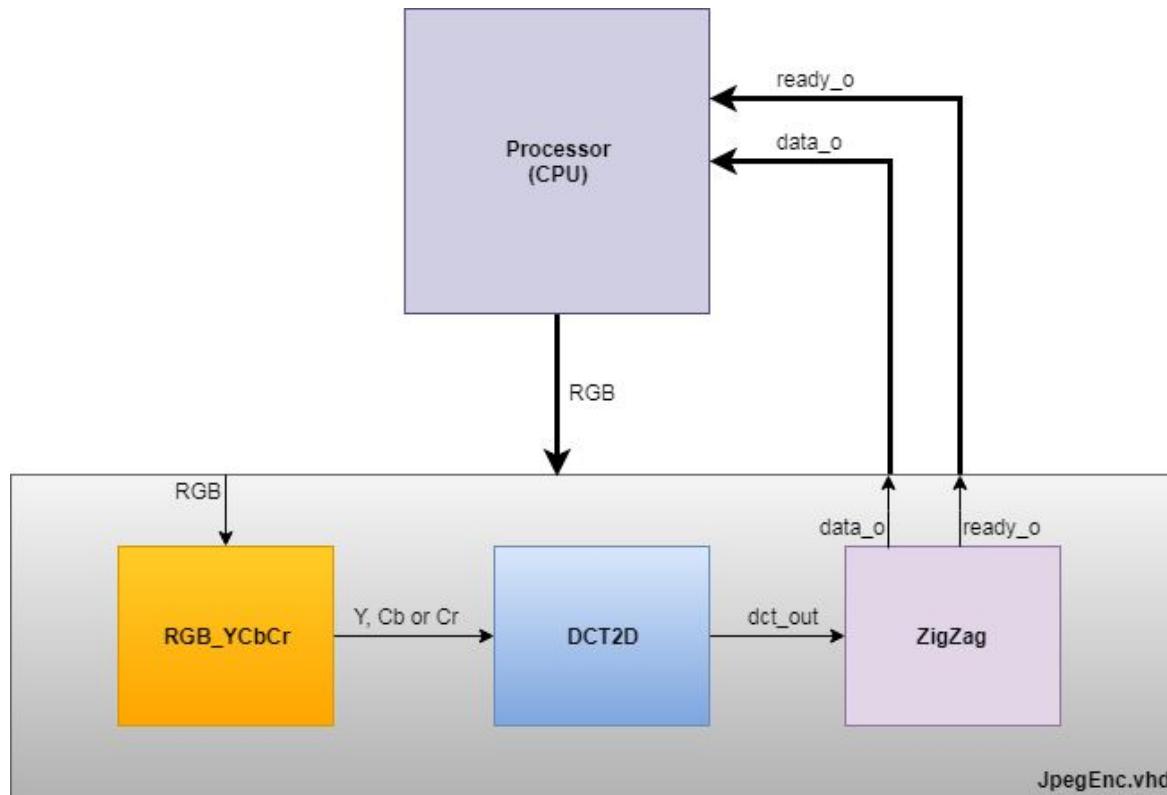
- Fluxo de compressão JPEG



Fonte: HUANG, J. The JPEG Standard. 2008.

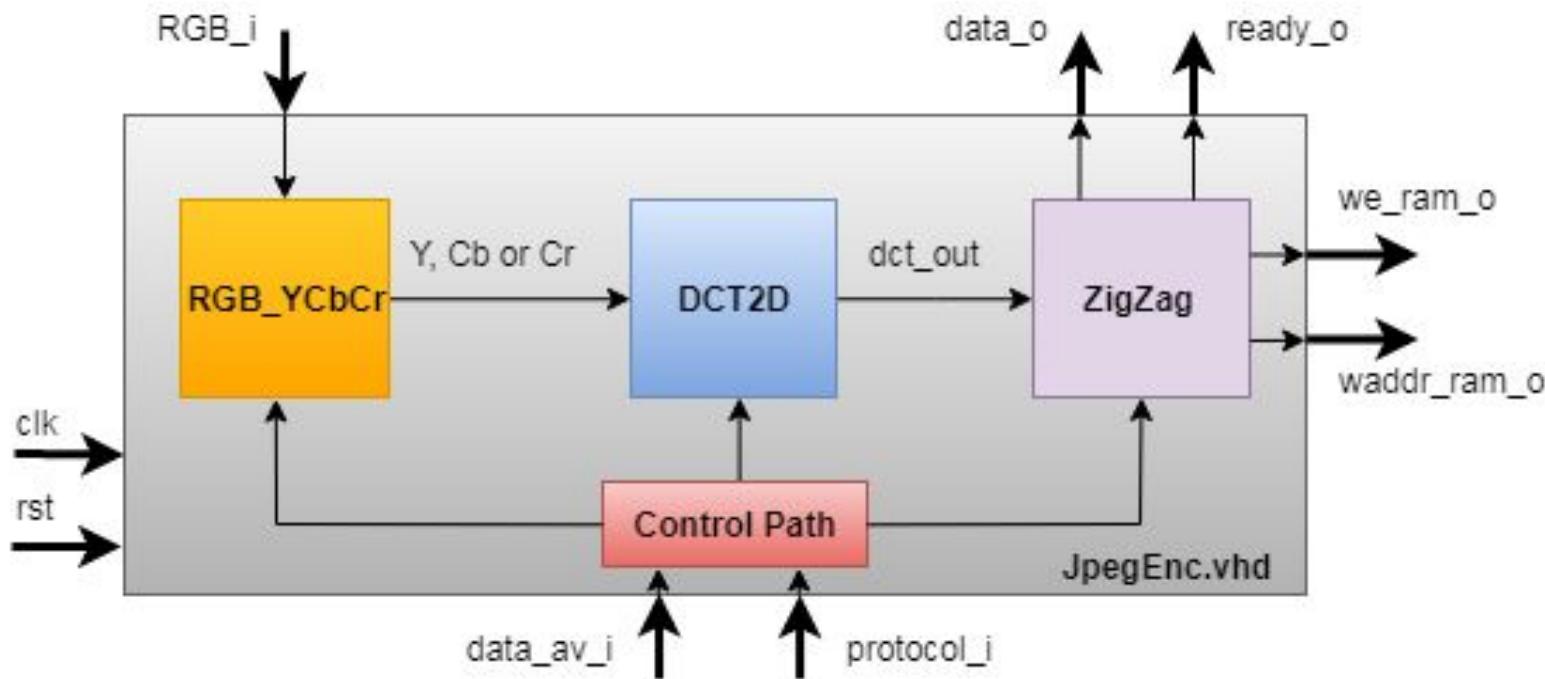
Desenvolvimento

- Diagrama de blocos



Desenvolvimento

- Fluxo de dados do projeto



Desenvolvimento

- *Control Path*
 - Pipeline;
 - Sequencial.
- *Data Path*
 - RGB_YCbCr ;
 - $DCT2D$;
 - *Scanner ZigZag*.



Resultados

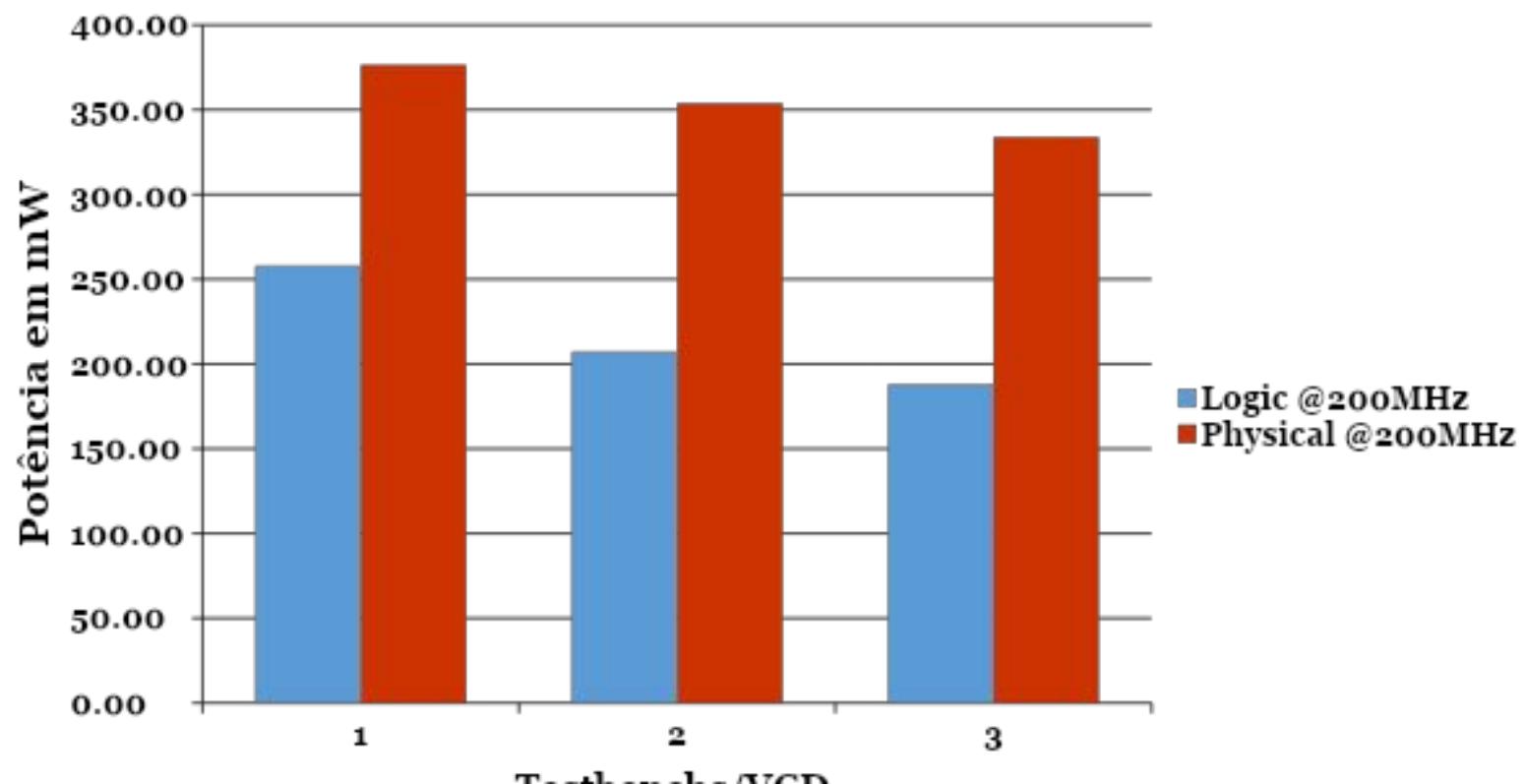
- Análise de tempo: 200MHz

Resultados de performance	
Vetor	Tempo (μs)
Lena: 8x8	3,75
Lena: 64x64	47,2
Lena: 512x512	2760

Resultados

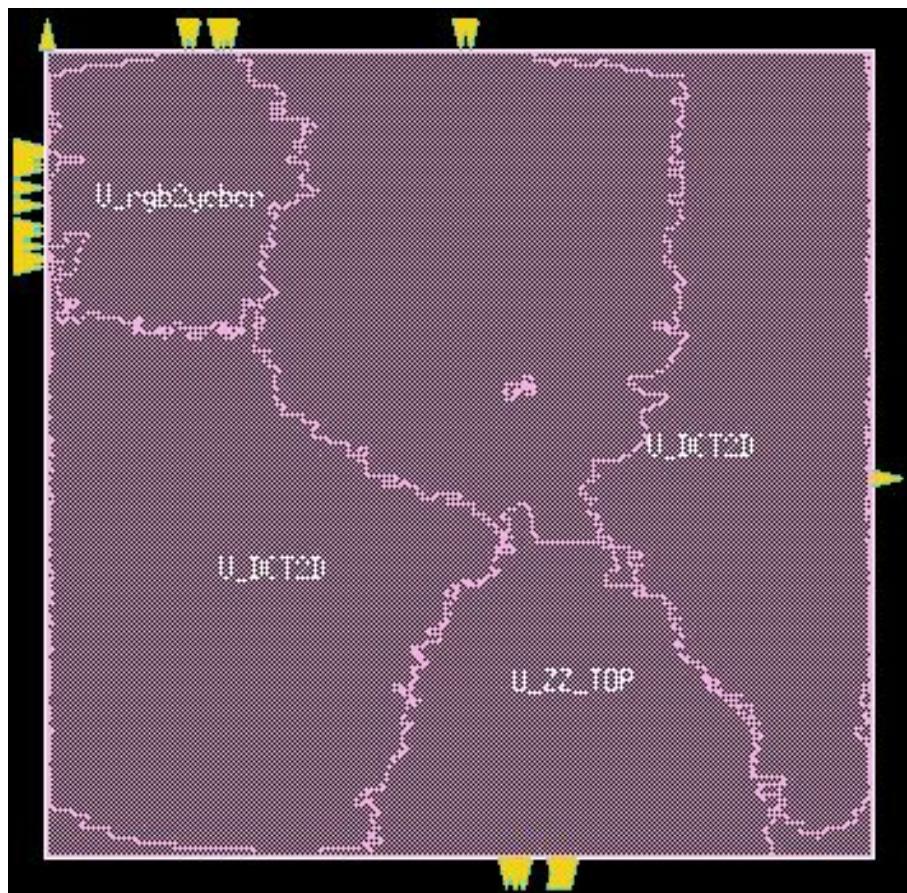
- Análise de potência sobre vetores de teste

Evolução da potência em função do VCD

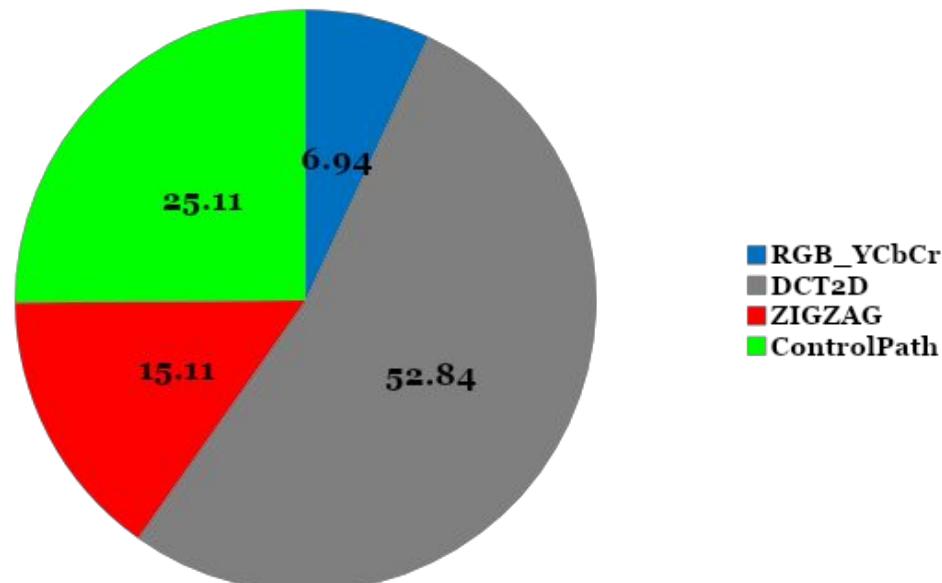


Resultados

- Distribuição de área (*die*) no *chip* JpegEnc: Área = 0,988 mm²



Distribuição de Área no *Chip* JpegEnc



Comparação: ARM Cortex A9

- Apple: iPhone 4s
- Samsung: Galaxy S2



Comparação

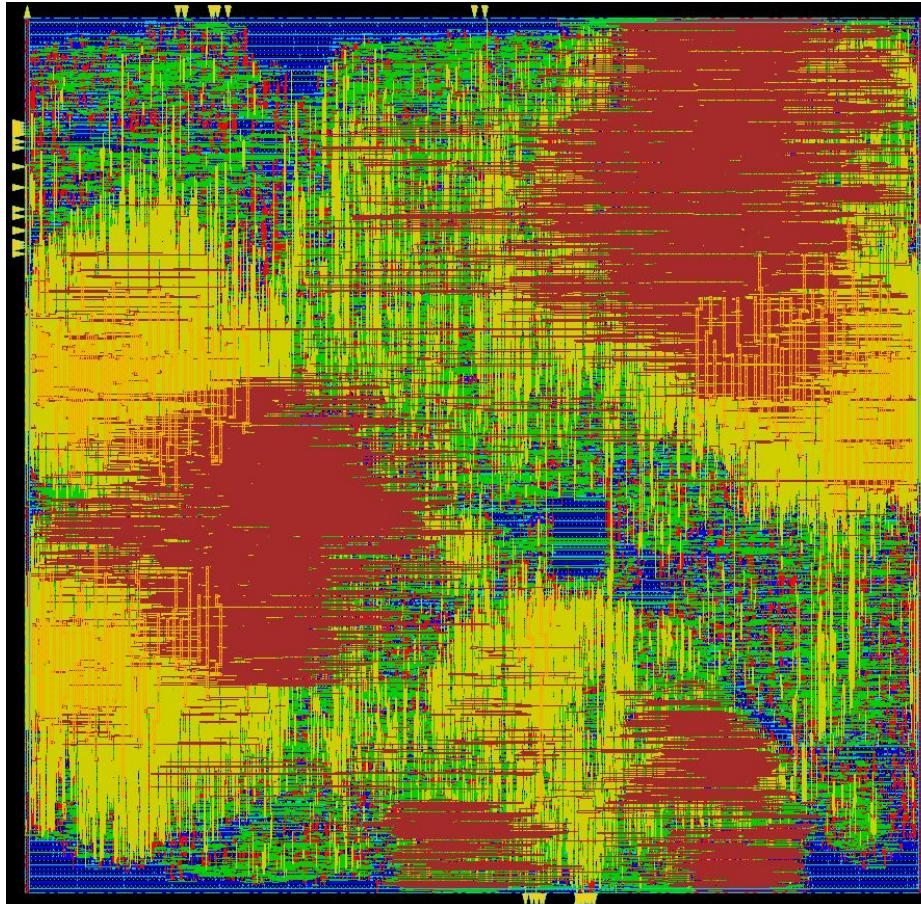
 vetor de teste: Lena 512x512			
Dispositivo	JpegEnc (180nm)	ASIC (Pastuszak)	ARM Cortex A9 (45 nm)
Área (mm²)	0,988	-	16,092
Potência (W)	0,334	-	7,323
Freq. (MHz)	200	200	1000
Tempo (ms)	2,76	10,01	580,56

Custo x Benefício em relação ARM Cortex A9

Dispositivo	Área (mm²)	Potência (W)
ARM Cortex A9 (45 nm)	16,092	7,32316
JpegEnc (180 nm)	0,988	0,33404
Acréscimo JpegEnc sobre Cortex A9	6,14%	4,56%

Resultados

- *Layout* físico do *chip* JpegEnc: Área= 0,988 mm²



Fonte: Encouter Digital Implementation - Cadence

Conclusões

- Trabalhar em cima de um objetivo concreto é de suma importância;
- Decisões iniciais de projeto são muito importantes;
- O *chip* JpegEnc é economicamente viável;
- Alto Custo x Benefício ao processador ARM Cortex A9.

Trabalhos Futuros

- Implementação dos blocos: Quantizador e codificação Huffman;
- Controle *pipeline* para maximizar o desempenho;
- Implementação Encoder e Decoder JPEG no mesmo *chip*.

Obrigado pela Atenção !

Perguntas?



Luis Felipe de Deus – felipe.deus@ecomp.ufsm.br

Nathanael Luchetta – nathanael.luchetta@ecomp.ufsm.br

Tiago Knorst – tiago.knorst@ecomp.ufsm.br