



TÉCNICO LISBOA



Embedded Software Development Environment for the OpenRisc Processor (Bare-Metal)

Carlos Alexandre Antunes Rodrigues

Dissertação para a obtenção de Grau de Mestre em
Engenharia Electrotécnica e de Computadores

Júri

Presidente: Nome do Presidente

Orientador: Nome do Orientador

Vogais: Nome do Vogal 1

Nome do Vogal 2

Abril de 2014

Dedicated to someone special...

Agradecimientos

A few words about the university, financial support, research advisor, dissertation readers, faculty or other professors, lab mates, other friends and family...

Resumo

Inserir o resumo em Português aqui com o máximo de 250 palavras e acompanhado de 4 a 6 palavras-chave...

Palavras-chave: palavra-chave1, palavra-chave2,...

Abstract

Insert your abstract here with a maximum of 250 words, followed by 4 to 6 keywords...

Keywords: keyword1, keyword2,...

Conteúdo

Agradecimentos	v
Resumo	vii
Abstract	ix
Lista de Tabelas	xiii
Lista de Figuras	xv
Lista de Símbolos	xvii
Glossário	xix
1 Introdução	1
1.1 Contexto	1
1.2 Motivação	1
1.3 Objectivos	1
1.4 Desafios	1
2 Estado-da-arte	2
2.1 OpenRisc	2
2.1.1 Toolchain	2
2.1.2 OrpSoc	2
2.2 Ferramentas de simulação	2
2.2.1 Or1ksim	2
2.2.2 Icarus	3
2.2.3 Verilator	3
2.2.4 Placa FPGA	3
2.3 OpenOCD	3
2.4 Protocolos	3
2.4.1 Uart	3
2.4.2 GPIO	3
2.4.3 I2C	3
2.4.4 SPI	4
3 SPI	5
3.1 Mestre SPI	5

3.2	escravo SPI	5
4	I2C	6
4.1	Mestre I2C	6
4.2	escravo I2C	6
5	Interface FIFO	7
6	Bootrom	8
6.1	Testes de verificação	8
6.1.1	teste de leitura SPI	8
6.1.2	teste de escrita na memoria principal	8
7	Testes de funcionamento	9
7.1	adicionar novos testes	9
7.2	Testes desenvolvidos	9
7.2.1	SPI	9
7.2.2	I2C	9
7.2.3	Interface FIFO	10
7.2.4	Bootrom	10
8	Resultados	11
8.1	Análise da área utilizado pelo systema	11
8.2	Análise da frequência de trabalho do sistema	11
9	Conclusão	13
9.1	Achievements	13
9.2	Trabalho Futuro	13
A	Vector calculus	15
A.1	Vector identities	15
	Bibliografia	17

Lista de Tabelas

Lista de Figuras

Lista de Símbolos

Simbolos gregos

α	Angle of attack.
β	Angle of side-slip.
κ	Thermal conductivity coefficient.
μ	Molecular viscosity coefficient.
ρ	Density.

Simbolos romanos

C_D	Coefficient of drag.
C_L	Coefficient of lift.
C_M	Coefficient of moment.
p	Pressure.
\mathbf{u}	Velocity vector.
u, v, w	Velocity Cartesian components.

Subscritos

∞	Free-stream condition.
i, j, k	Computational indexes.
n	Normal component.
x, y, z	Cartesian components.
ref	Reference condition.

Sobrescritos

*	Adjoint.
T	Transpose.

Glossário

- CFD** Computational Fluid Dynamics is a branch of fluid mechanics that uses numerical methods and algorithms to solve problems that involve fluid flows.
- CSM** Computational Structural Mechanics is a branch of structure mechanics that uses numerical methods and algorithms to perform the analysis of structures and its components.
- MDO** Multi-Disciplinary Optimization is an engineering technique that uses optimization methods to solve design problems incorporating two or more disciplines.

Capítulo 1

Introdução

explicar o que é um sistema e o que se pretende com ele.

1.1 Contexto

não existe nenhum sistema desenvolvido com o que a startup pretende. e os que existem são demasiado caros, têm demasiadas coisas o que corresponde a um custo de energético elevado.

1.2 Motivação

desenvolver um sistema necessário para uma startup, o sistema vai ser desenvolvido a medida com o pretendido com a startup.

1.3 Objectivos

dizer qual é o objectivo do projecto. explicar que é ter um sistema com um processador, spi, ROM, flash, i2c, fifos de comunicação, uart, GPIO

Imagem com um sistema que é pretendido,

explicar as interfaces de entrada e saída de dados, e os modelos dentro do sistema.

1.4 Desafios

desenvolver um sistema com todos os modelos a funcionar correctamente.

Capítulo 2

Estado-da-arte

2.1 OpenRisc

é uma comunidade opensource que desenvolve hardware e ferramentas que ajudam a na criação de novos sistemas o OrpSoc

cores têm código escrito verilog e VDH sendo o verilog maioritário

2.1.1 Toolchain

é a toolchain da GNU onde foi adicionado o processador desenvolvida pela comunidade.

foi desenvolvida pela comunidade falar sobre ferramentas

tem a seleção do de onde o código vai correr.

por uma imagem a explicar a ideia de ser preciso compilar com o mboard

2.1.2 OrpSoc

explicar a ferramenta que foi desenvolvida para que serve e as vantagens

mudou de nome (FuseSoc) para se tornar uma ferramenta independente da comunidade.

por uma imagem de como o OrpSoc está construído.

2.2 Ferramentas de simulação

vantagem de ter um sistema de simulação.

2.2.1 Or1ksim

dizer que é um simulador desenvolvido em C que pretende simular o processador.

vantagens e desvantagens

2.2.2 Icarus

explicar como funciona o icarus que precisa de uma testbench em verilog, dá para ver as ondas e tem a vantagem de ter o estado de alta impedancia.

Mas demora muito tempo.

Por uma imagem das ondas com alta impedancia com um tempo igual no verilog

2.2.3 Verilator

explica como funciona o verilator, converte os codigos em verilog em codigo C ou systemC conforme o que a pessoa pretender.

pode ser a testbench escrita em C ou systemC, que se pode ver as ondas e têm a desvantagem que não conseguem simular o estado de alta impedancia.

muito rapido comparado com o icarus

Por uma imagem das ondas com alta impedancia com um tempo igual no icarus

2.2.4 Placa FPGA

explicar o que é como funciona

como funciona

2.3 OpenOCD

explicar para que serve e como é ligado. e as portas para cada ferramenta.

Por uma imagem sobre como é ligado o openocd

2.4 Protocolos

2.4.1 Uart

falar da uart como funciona o protocolo

2.4.2 GPIO

falar um pouco do GPIO como funciona

2.4.3 I2C

Falar do I2C como funciona os pinos que tem para que servem

falar e imagem do start, stop bit e dos dados

2.4.4 SPI

falar do SPI. como funcionar quais são os pinos que tem. para que serve cada um

Capítulo 3

SPI

Explicar o protocolo spi. como funciona e as linhas que tem. explicar em que casos é mais usado por uma imagem os um master e slaves para mostrar como são as ligações

3.1 Mestre SPI

estado do core. dizer que o master spi só conseguia enviar e nao recebia qualquer tipo de informação por um diagrama de blocos de como circula a informação dentro do meu modelo de SPI. fazer uma descrição do funcionamento com uma descrição do fluxo de dados.

por a tabela de sinais da interface de SPI (sinais de entrada e saida)

tabela de registos (endereços do registos).

explicar os 2 modos de funcionamento de escrita e de leitura que tem 2 modos.

3.2 escravo SPI

dizer que foi desenhado desde o inicio.

por um diagrama de blocos de como circula a informação dentro do meu modelo de SPI. fazer uma descrição do funcionamento com uma descrição do fluxo de dados.

por a tabela de sinais da interface de SPI (sinais de entrada e saida)

tabela de registos (endereços do registos).

explicar os 2 modos de funcionamento de escrita e de leitura que tem 2 modos.

Capítulo 4

I2C

Explicar o protocolo i2c. como funciona e as linhas que tem. explicar em que casos é mais usado por uma imagem os um master e slaves para mostrar como são as ligações

4.1 Mestre I2C

dizer como estava o master a I2C que não conseguia enviar ao receber dados.

por um diagrama de blocos de como circula a informação dentro do meu modelo de SPI. fazer uma descrição do funcionamento com uma descrição do fluxo de dados.

por a tabela de sinais da interface de i2c (sinais de entrada e saída)

tabela de registos (endereços do registos).

explicar os 2 modos de funcionamento de escrita e de leitura que tem 2 modos.

4.2 escravo I2C

processo de construção do escravo

por um diagrama de blocos de como circula a informação dentro do meu modelo de SPI. fazer uma descrição do funcionamento com uma descrição do fluxo de dados.

por a tabela de sinais da interface de i2c (sinais de entrada e saída)

tabela de registos (endereços do registos).

explicar os 2 modos de funcionamento de escrita e de leitura que tem 2 modos.

Capítulo 5

Interface FIFO

porque que é necessario.

Explicar o protocolo a interface. como funciona e as linhas que tem.

por uma imagem para mostrar como são as ligações

por um diagrama de blocos de como circula a informação dentro do meu modelo de interface. fazer uma descrição do funcionamento com uma descrição do fluxo de dados.

por a tabela de sinais da interface de (sinais de entrada e saída) e para para os 2 lados.

tabela de registos (endereços do registos).

Capítulo 6

Bootrom

explicar o que é uma bootrom

o que se tem de fazer para correr a partir da bootrom.

explicar o que se pretende com a bootrom.

por um fluxograma do fluxo da bootrom

6.1 Testes de verificação

para que servem estes testes?

6.1.1 teste de leitura SPI

explicar como é feito o teste de leitura

como mostra o erro tanto com o GPIO como na uart.

6.1.2 teste de escrita na memoria principal

explicar como é feito o teste de escrita na memoria principal

como mostra o erro tanto com o GPIO como na uart.

Capítulo 7

Testes de funcionamento

explicar para que quero os testes.

como foi implementado no Orpsoc.

adicionar uma imagem explicativa como foi implementado.

falar sobre as flags do teste

como posso ver os resultados dos teste

7.1 adicionar novos testes

fazer novos testes.

7.2 Testes desenvolvidos

para mostrar que o sistema funciona corretamente foi desenvolvido estes testes

7.2.1 SPI

explicar como é testado

simulador verilator

Board VGA

7.2.2 I2C

explicar como é testado

simulador verilator

Board VGA

7.2.3 Interface FIFO

explicar como é testado

simulador verilator

Board VGA

7.2.4 Bootrom

explicar como é testado

simulador verilator

Board VGA

Capítulo 8

Resultados

8.1 Analise da área utilizado pelo systema

8.2 Analise da frequencia de trabalho do sistema

Capítulo 9

Conclusão

Insert your chapter material here...

9.1 Achievements

The major achievements of the present work...

9.2 Trabalho Futuro

dese

Apêndice A

Vector calculus

In case an appendix is deemed necessary, the document cannot exceed a total of 100 pages...

Some definitions and vector identities are listed in the section below.

A.1 Vector identities

$$\nabla \times (\nabla \phi) = 0 \tag{A.1}$$

$$\nabla \cdot (\nabla \times \mathbf{u}) = 0 \tag{A.2}$$

Bibliografia

- Satish Balay, Kris Buschelman, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Barry F. Smith, and Hong Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 2.3.0, Argonne National Laboratory, 2004.
- Antony Jameson, Niles A. Pierce, and Luigi Martinelli. Optimum aerodynamic design using the Navier–Stokes equations. In *Theoretical and Computational Fluid Dynamics*, volume 10, pages 213–237. Springer-Verlag GmbH, January 1998.
- A. C. Marta, C. A. Mader, J. R. R. A. Martins, E. van der Weide, and J. J. Alonso. A methodology for the development of discrete adjoint solvers using automatic differentiation tools. *International Journal of Computational Fluid Dynamics*, 21(9–10):307–327, October 2007.
- Andre C. Marta, Sriram Shankaran, D. Graham Holmes, and Alexander Stein. Development of adjoint solvers for engineering gradient-based turbomachinery design applications. In *Proceedings of the ASME Turbo Expo 2009: Power for Land, Sea and Air*, number GT2009-59297, June 2009.
- Joaquim R. R. A. Martins, Juan J. Alonso, and James J. Reuther. High-fidelity aerostructural design optimization of a supersonic business jet. *Journal of Aircraft*, 41(3):523–530, May 2004.
- Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. Springer, 1999.

