**INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA**

Licenciatura em Engenharia Informática e de Computadores

Logo

Description automatically generated

Arquitetura de Computadores

**Relatório Trabalho 3**

Trabalho realizado por:

|  |  |
| --- | --- |
| Nome: André Monteiro | Nº 43842 |
| Nome: Constança Costa | Nº 50541 |
| Turma: LEIC24D  Docente: Rui Duarte |  |

2022 / 2023 Semestre Verão

13 de maio de 2023

**Índice**

1. **Introdução**
2. **Desenvolvimento do trabalho**
   1. Definição do mapa de endereços
   2. Caracterização da atividade dos barramentos
   3. Evolução da arquitetura
   4. Teste do sistema
3. **Conclusão**

# **Introdução**

Neste relatório, iremos analisar o projeto de módulos de memória e de portos paralelos de entrada e de saída num sistema com o processador P16. Os mapas de endereçamento são importantes para que o processador possa aceder corretamente aos dispositivos de memória e periféricos, permitindo que as informações sejam armazenadas e recuperadas de maneira eficiente.

# **Desenvolvimento do trabalho**

***2.1.*** **Definição do mapa de endereços**

a)

* #1 – ROM porque não tem sinal de escrita. Capacidade =2^13 \* 2 = 16 KB
* #2 & #3 – RAM pois tem sinal de escrita. Capacidade =2^12 \* 1 + 2^12 \* 1= 8KB

b)

#4 – porto de saída pois só escreve e não lê.

A sua dimensão é igual à capacidade, por isso 0001 bbbb bbbb bbbb, logo max = 0x1FFF e min = 0x1000, sendo assim max – min +1 = 0x1000 que são 4 KB.

Modo de acesso é word-wise.

c)

Uma imagem com gráfico

Descrição gerada automaticamente

d)

A afirmação é falsa pois a memória apresenta mais endereços do que aqueles a que temos “acesso”, subaproveitamento.

***2.2* Caracterização da atividade dos barramentos**

A screenshot of a computer

Description automatically generated

***2.3* Evolução da arquitetura**

a)

Chart

Description automatically generated

b)

CS ROM #1 = not A15 and not A14

CS RAM #2 & #3 = not A15 and A14

CS porto entrada = A15 and A14 and A13 and A12 and not A11

CS porto saida #4 = A15 and A14 and A13 and not A12

c)

desenho

***2.4* Teste do sistema**

Este foi o código produzido pelo grupo de forma a testar o sistema:

**.equ STACK\_SIZE, 64**

**.equ INPORT\_ADDRESS, 0xF000**

**.equ OUTPORT\_ADDRESS, 0xE000**

**.section .startup**

**b \_start**

**b .**

**\_start:**

**ldr sp, tos\_addr**

**mov r0, pc**

**add lr, r0, #4**

**ldr pc, main\_addr**

**b .**

**tos\_addr:**

**.word tos**

**main\_addr:**

**.word main**

**main:**

**ldr r0, in\_port\_addr**

**ldrb r1, [r0, #0]**

**mov r2, #0x01**

**and r0, r1, r2**

**bne change\_output\_port**

**b main**

**change\_output\_port:**

**lsr r1, r1, #1**

**mov r0, #0x40**

**and r2, r0, r1**

**bne zero**

**mov r1, #0x80**

**add r0, r0, r1**

**ldr r1, out\_port\_addr**

**str r0, [r1, #0]**

**b main**

**zero:**

**mov r1, #0**

**add r0, r0, r1**

**b main**

**in\_port\_addr:**

**.word INPORT\_ADDRESS**

**out\_port\_addr:**

**.word OUTPORT\_ADDRESS**

**.data**

**.section .bss**

**.section .stack**

**.space STACK\_SIZE**

**tos:**

**.word 0x8000 ; o stack começa no fim da RAM**

# Conclusão

Ao estudar os mecanismos de endereçamento, é possível entender como o processador P16 gerencia o acesso à memória e aos periféricos, tornando-se uma ferramenta fundamental para quem trabalha com sistemas embarcados e microcontroladores.