



UNIVERSIDADE FEDERAL DE SANTA CATARINA - UFSC

EDUARDO BISCHOFF GRASEL - 22200355

JOÃO VITTOR BRAZ DE OLIVEIRA - 22215582

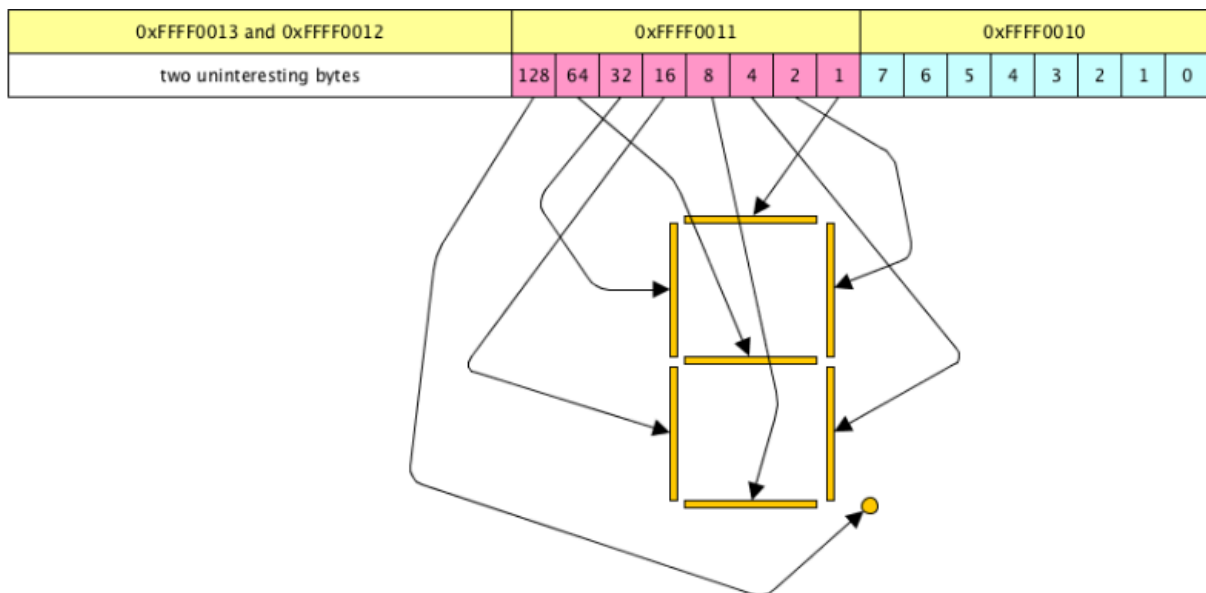
Relatório Organização de computadores

Objetivo do projeto

O projeto tem como objetivo aprofundar o conhecimento em relação a endereços de memória do Mips e o seu sistema de armazenamento, juntamente com a direta alteração destes endereços para utilização do Digital Lab Sim do Mars, o qual faz uma ponte entre a memória e o display.

Projeto 1

Iniciamos o projeto guardando o endereço 0xFFFF0010 que fará um ponte para o display do Digital Lab Sim em \$t0 utilizando um load immediate. Em seguida consideramos o valor de cada bit da memória nesse endereço para acender um led no display (1 = ligado, 0 = desligado), seguindo o padrão da imagem, a cor azul e rosa indicam que cada um desses endereços representa um valor diferente.



<https://wilkinsonj.people.cofc.edu/digilab.html>

Agora que já temos informações suficientes sobre o display, iniciaremos a mudança deste endereço de memória, utilizaremos primeiramente, seguindo o padrão representado pela imagem acima, o valor binário de cada parte do display em ordem, em seguida o transformamos em hexadecimal, por exemplo, para representar o número 1:

0 0 0 0, 0 1 1 0 = Valor do display binário
0 6 = Valor Hexadecimal (0x06)

Agora guardamos esse valor em \$s0 através do load immediate, seguidamente usamos o store byte passando \$s0 e o endereço presente em \$t0, e repetimos esse processo de li \$s0, hex seguido de sb \$s0, 0(\$t0) para os demais números do contador.

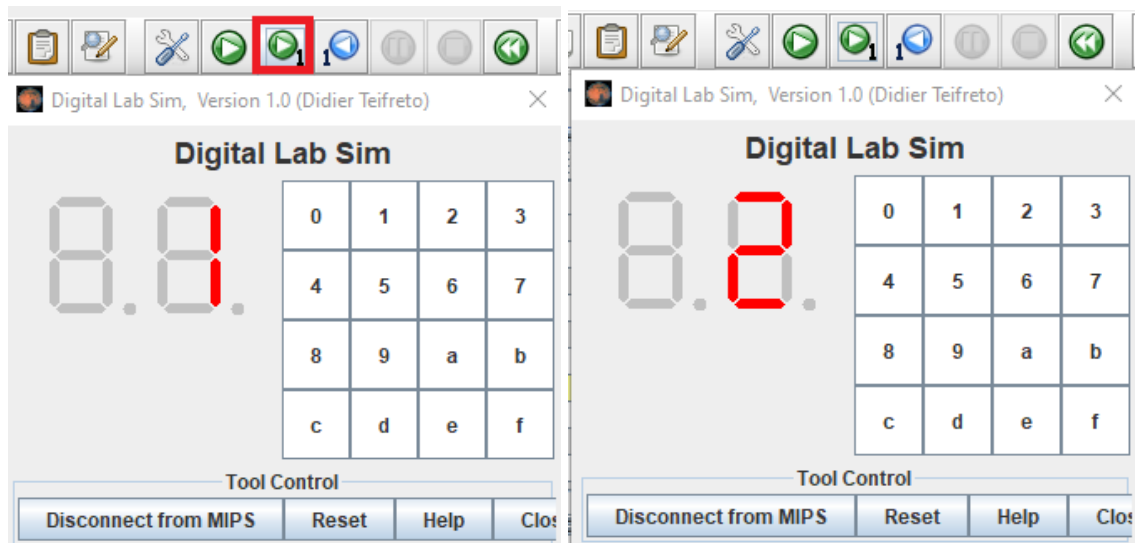
```

# 2
li $s0, 0x5b # para ligar o número 1, precisamos de 01011011, transformando em hex = 0x5b
sb $s0, 0($t0) # ativando os leds para mostrar o 2

# 3
li $s0, 0x4f # ...
sb $s0, 0($t0) # ativando os leds para mostrar o 3

```

O funcionamento se dará através do Digital Lab Sim, cada duas vezes que clicarmos em next com o programa compilado, ele atualizará o display.



Projeto 2

Inicialmente, já dentro do .text iniciamos o loop principal, que buscará cada linha e coluna do teclado alfanumérico. Para tal carregamos em \$t1 o valor decimal que faz referência a linha e o carregamos esse valor no Byte 0xFFFF0012, seguidamente damos um load byte em 0xFFFF0014 e guardamos em \$t2, com isso temos a base do programa, agora para finalizar a linha um do teclado alfanumérico salvamos os valores decimais relativos aos botões para compará-los com o devido valor nativo deles (que estão em \$t2), assim, caso seja pressionado seremos enviados para o bloco de código para mostrar o número no display, que nesse caso, para facilidade, foi usado o valor decimal para montar o display, simplesmente damos um load immediate no valor decimal, então guardamos esse valor através do store byte em 0xFFFF0010, então voltamos ao loop novamente.

```
loop:
li    $t1, 1
sb    $t1, 0xFFFF0012
lb    $t2, 0xFFFF0014

li    $t3, 17
beq   $t3, $t2, zero

li    $t3, 33
beq   $t3, $t2, one

li    $t3, 65
beq   $t3, $t2, two

li    $t3, -127
beq   $t3, $t2, three

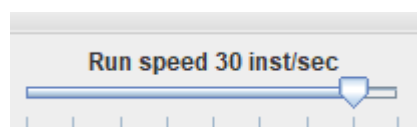
li    $t1, 2
sb    $t1, 0xFFFF0012
lb    $t2, 0xFFFF0014

one:
li    $t0, 6
sb    $t0, 0xFFFF0010
j     loop

two:
li    $t0, 91
sb    $t0, 0xFFFF0010
j     loop

three:
li    $t0, 79
sb    $t0, 0xFFFF0010
j     loop
```

Agora repetimos o processo, apenas mudando o valor carregado em \$t1 e armazenado nos bytes 0xFFFF0012, até que todas as teclas sejam devidamente mapeadas. A fim de rodar o programa, recomendamos diminuir a velocidade de leitura das instruções.



Digital Lab Sim

8.8.

0	1	2	3
4	5	6	7
8	9	a	b
c	d	e	f

Digital Lab Sim

8.8.

0	1	2	3
4	5	6	7
8	9	a	b
c	d	e	f

Digital Lab Sim

8.8.

0	1	2	3
4	5	6	7
8	9	a	b
c	d	e	f

Digital Lab Sim

8.8.

0	1	2	3
4	5	6	7
8	9	a	b
c	d	e	f

Conclusão

Ao fim do desafio foram absorvidos conhecimentos sobre alteração, leitura de bytes em memória e interação com o Digital Lab Sim. O maior obstáculo foi, sem dúvidas, a falta de informações sobre a ferramenta Digital Lab Sim online, então tivemos que nos guiar apenas pelo help do simulador, o que fornecia informações básicas sobre os bytes de endereçamento e nada mais. Embora o primeiro desafio tenha sido relativamente fácil, o segundo, talvez pela nossa falta de experiência, foi extremamente desafiador, isso é: A parte de encontrar a lógica correta para a leitura do teclado alfanumérico.