

Universidade de Brasília

Departamento de Engenharia Elétrica



Trabalho 2 OAC – Parte 2

Autor:

Matheus Augusto Silva Pinho 18/0024906

Brasília, 15 de Dezembro de 2020

JALR

Código usado para teste

Para que a testagem do programa em VHDL fosse possível, foi adicionado ao “imem.txt” os códigos correspondentes à instruções em assembly. Para que o teste do “jalr” acontecesse utilizou-se:

```
00a38367 ← jalr t1, t2,10
00128293 ← addi t0, t0,1 : função apenas para complementar o teste
00a00893 ← Começa o exit
00000073
```

Resultado da Simulação



Output do programa

```
output_file.txt
REG : 6 : 00000004
REG : 17 : 0000000A
syscall exit
```

Conforme podemos observar, o registrador 6 (rd) receberá 4 (PC + 4) e o registrador 17 (PC) receberá 10 (rs1 + imediato).

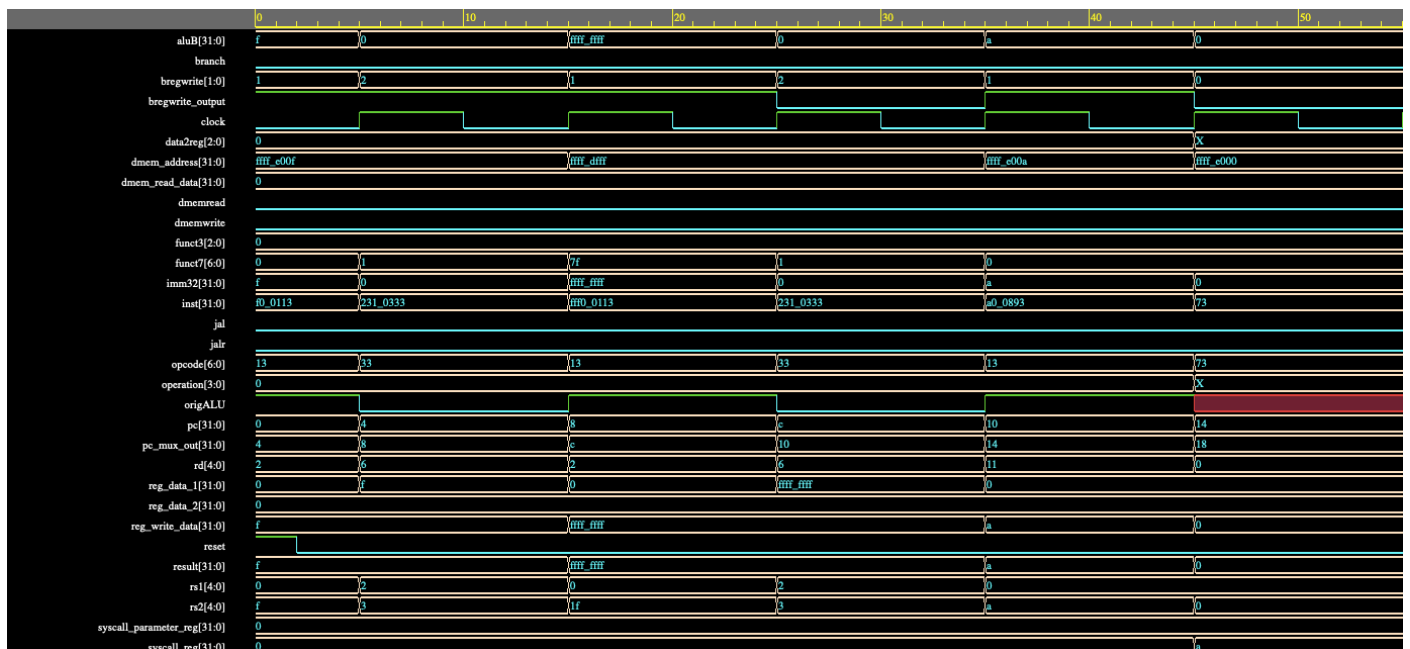
ADDRPOS

Código usado para teste

Para que a testagem do programa em VHDL fosse possível, foi adicionado ao “imem.txt” os códigos correspondentes à instruções em *assembly*. Para que o teste do “addrpos” acontecesse utilizou-se:

```
00f00113 ← addi x6,x0,15
02310333 ← addrpos x7, x0, x6
fff00113 ← addi x6,x0,-1
02310333 ← addrpos x7, x0, x6
00a00893
00000073
```

Resultado



Output do programa

```

output_file.txt
REG : 2 : 0000000F
REG : 6 : 0000000F
REG : 2 : FFFFFFFF
REG : 17 : 0000000A
syscall exit

```

Pode – se perceber que o resultado recebido foi correspondente ao esperando, tendo em vista que primeiramente, quando o registrador 2 (soma rs1 + rs2) tinha o valor “15”, o rd (reg 6) o recebeu, entretanto ao ter um valor negativo, rd fica sem valor.

GETBFW

Código usado para teste

Para que a testagem do programa em VHDL fosse possível, foi adicionado ao “imem.txt” os códigos correspondentes à instruções em *assembly*. Para que o teste do “getbfw” acontecesse utilizou-se:

```

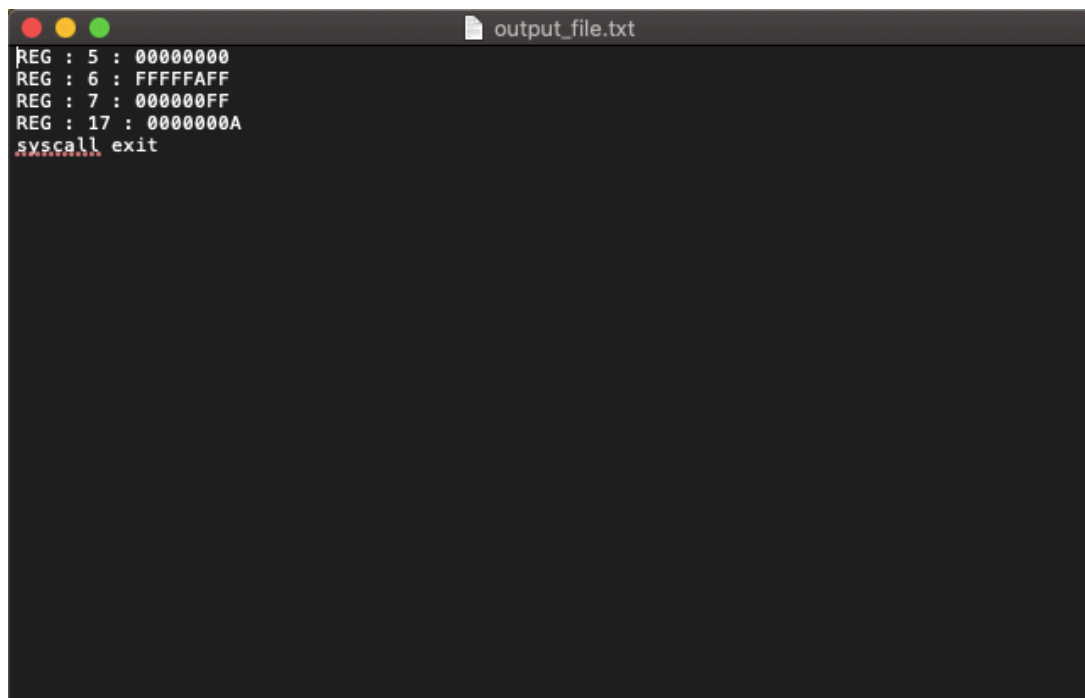
00000293 ← addi x5, x0, 0
1d600313 ← addi x6, x0, 0xfffffaff
005303ff ← getbfw x7, x6, x5
00a00893 ← Começa o exit ...
00000073

```

Resultado



Output do programa

A screenshot of a debugger window titled 'output_file.txt'. The window has a dark background and shows the following text: 'REG : 5 : 00000000', 'REG : 6 : FFFFFFFF', 'REG : 7 : 000000FF', and 'REG : 17 : 0000000A'. Below these, the instruction 'syscall exit' is displayed. The 'syscall' part of the instruction is highlighted with a red dotted underline.

```
REG : 5 : 00000000
REG : 6 : FFFFFFFF
REG : 7 : 000000FF
REG : 17 : 0000000A
syscall exit
```

Conforme esperado, o registrador 7, teve como resultado o byte menos significativo (0xff) da word inserida.