

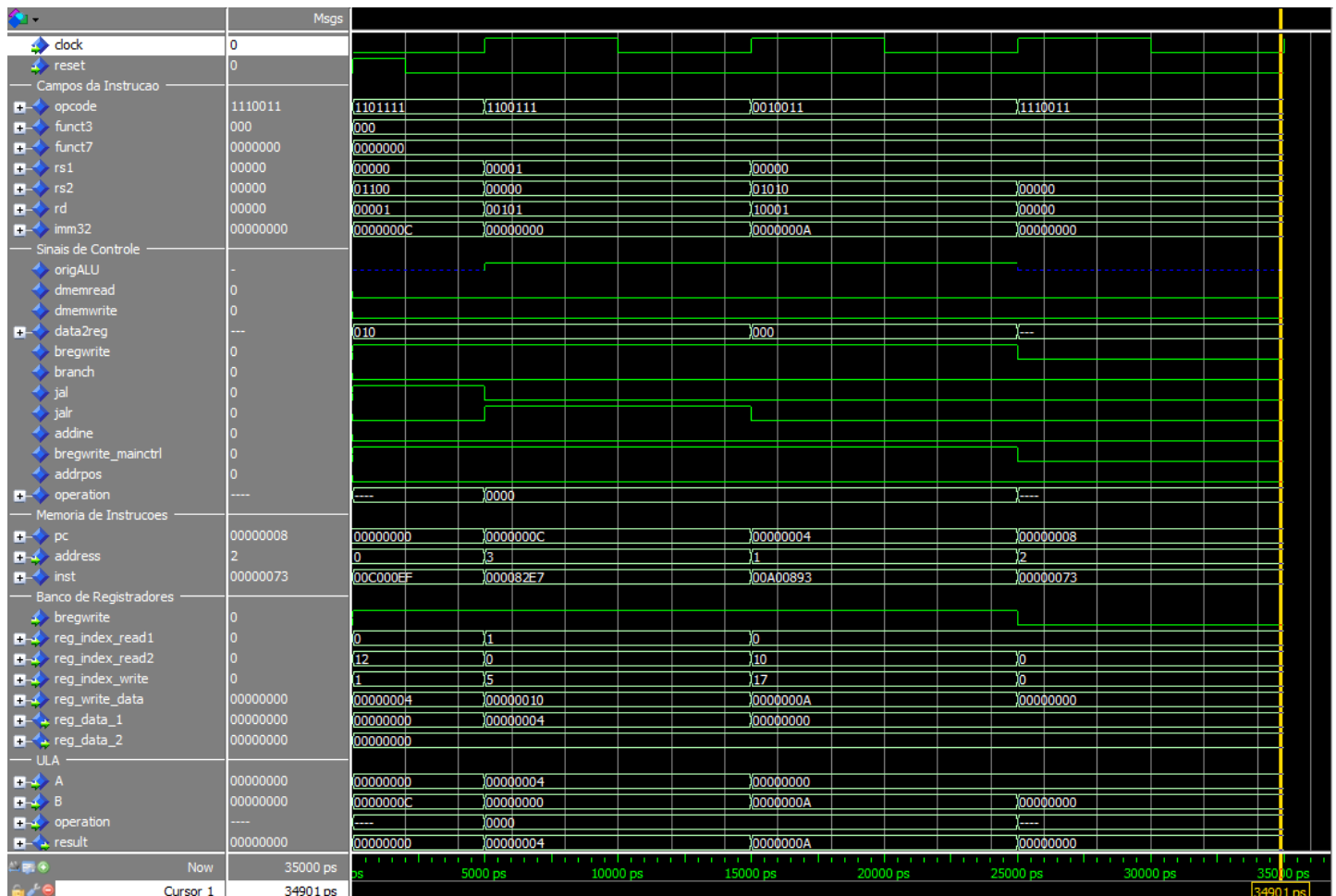
UNIVERSIDADE DE BRASÍLIA  
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO  
DISCIPLINA: ORGANIZAÇÃO E ARQUITETURA DE COMPUTADORES  
1º SEMESTRE 2020

TURMA C

RELATÓRIO DO TRABALHO 2 PARTE 2  
GIANLUCAS DOS SANTOS LOPES 18/0041991  
1) jalr

Para testar a instrução, foi utilizado o seguinte código:

```
.text
jal    ra, func
addi   a7, x0, 10
ecall  # terminate ecall
func:
jalr    t0, ra, 0
```



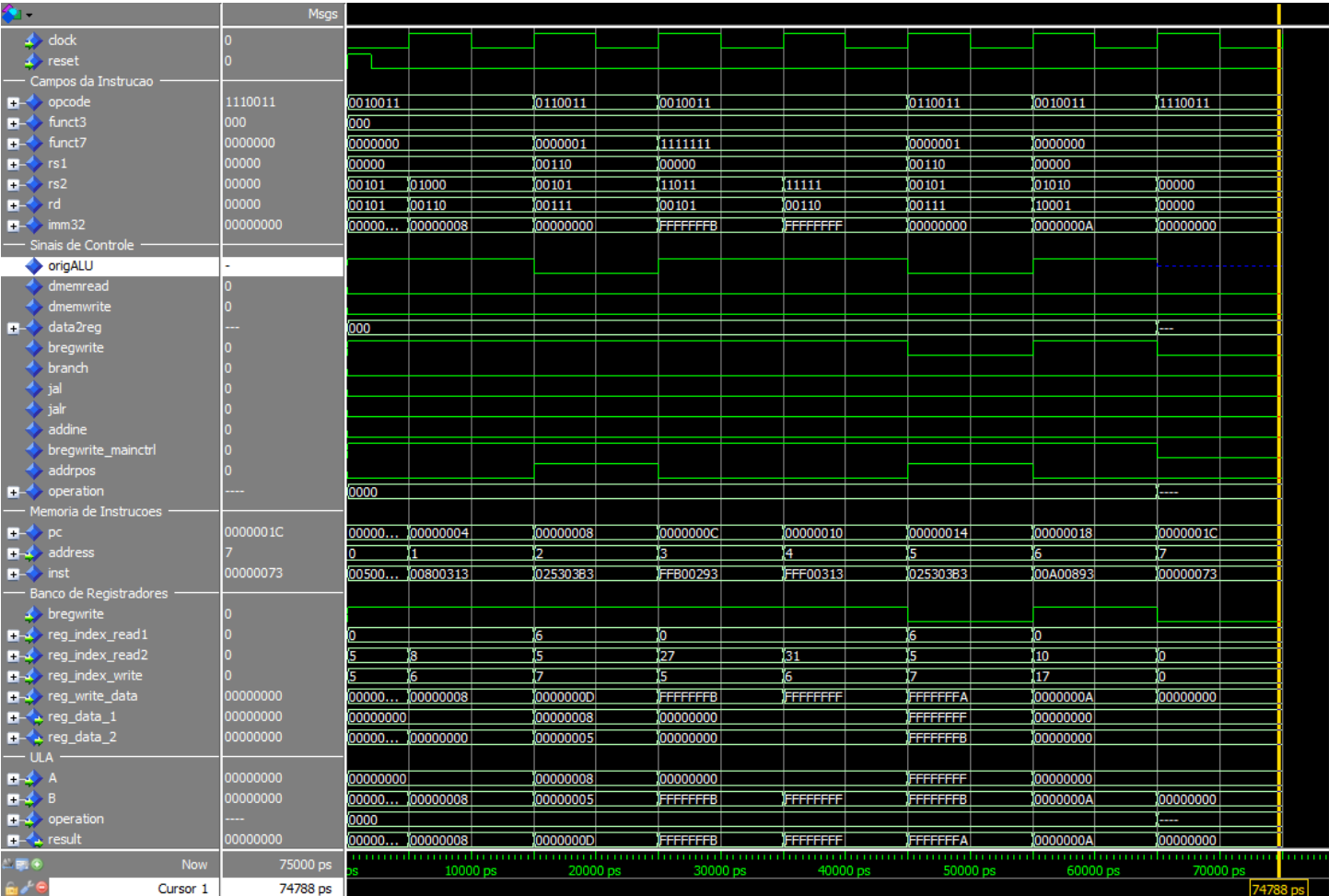
```
REG : 1 : 00000004
REG : 5 : 00000010
REG : 17 : 0000000A
syscall exit
```

É possível ver que o PC+imm foi escrito no registrador t0 (x5) e que ocorreu um salto para a instrução 4 (valor de ra).

2) addrpos

Para testar a instrução, foi utilizado o seguinte código:

```
.text
    li t0, 5
    li t1, 8
    addrpos t2,t1,t0
    li t0, -5
    li t1, -1
    addrpos t2,t1,t0
    addi    a7, x0, 10
    ecall # terminate ecall
```



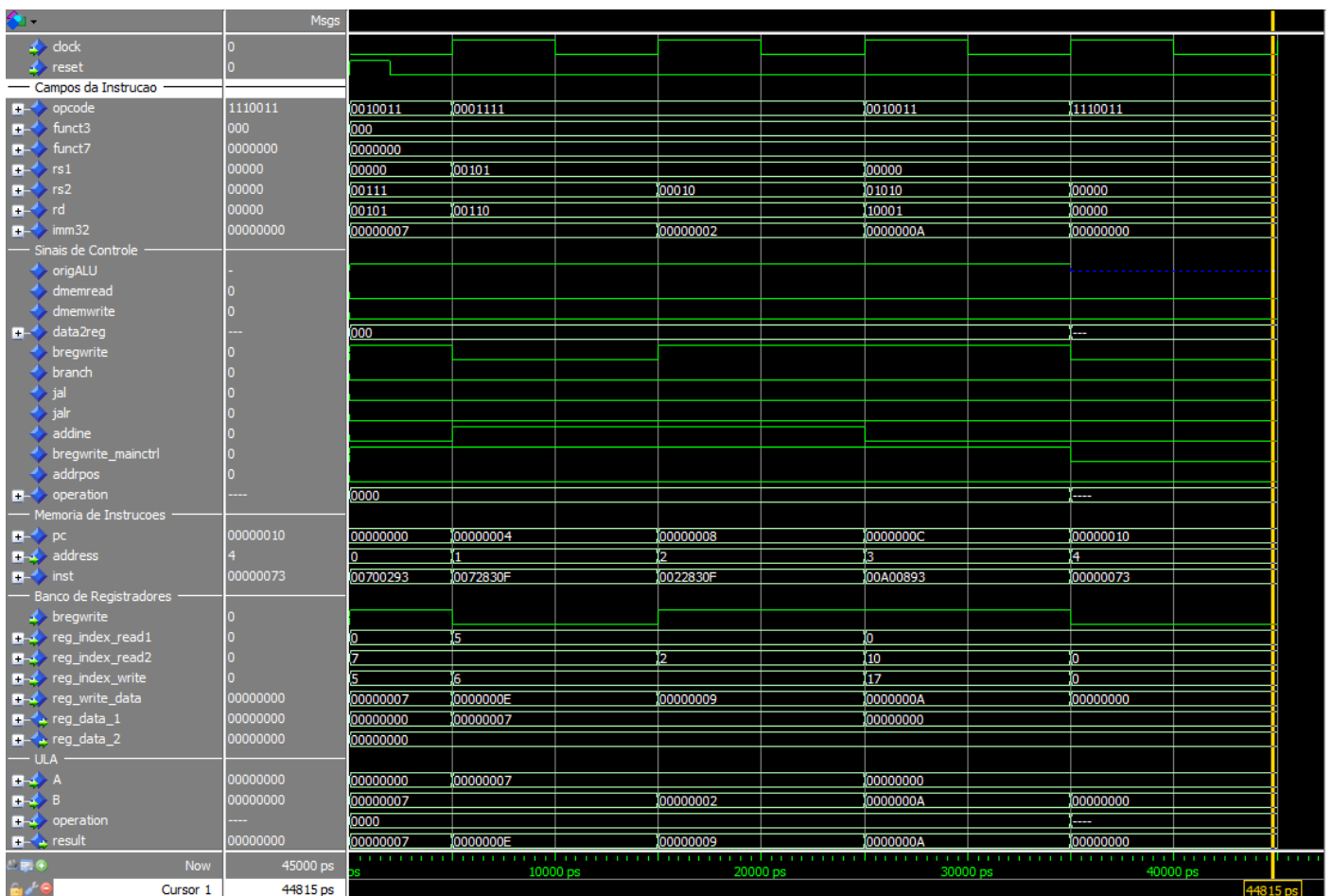
```
REG : 5 : 00000005
REG : 6 : 00000008
REG : 7 : 0000000D
REG : 5 : FFFFFFFB
REG : 6 : FFFFFFFF
REG : 17 : 0000000A
syscall exit
```

É possível ver que o valor do resultado só foi escrito no registrador de destino quando o mesmo era positivo.

### 3) addine

Para testar a instrução, foi utilizado o seguinte código:

```
.text
    addi    t0, x0, 7
    addine  t1, t0, 7
    addine  t1, t0, 2
    addi    a7, x0, 10
    ecall # terminate ecall
```



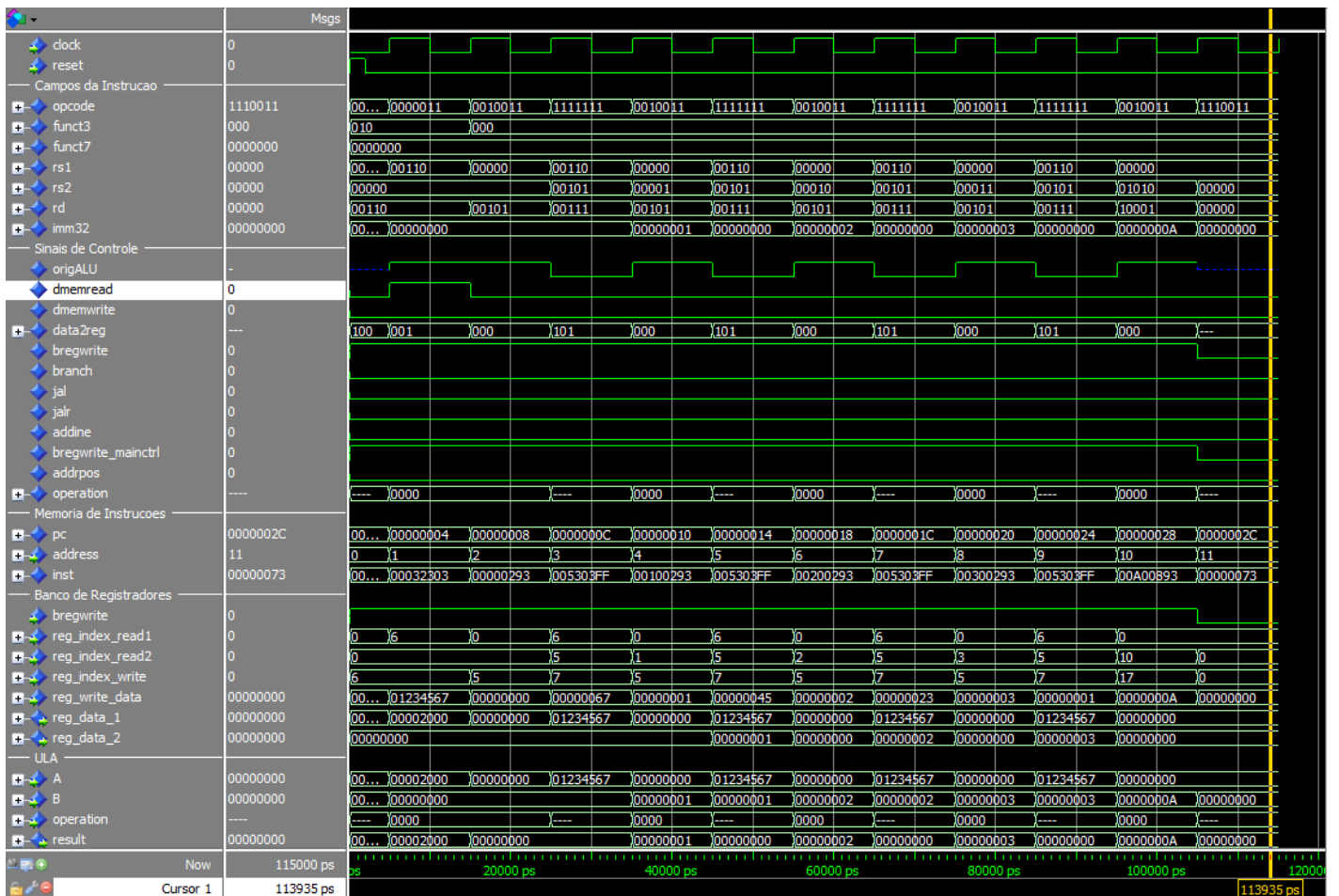
```
REG : 5 : 00000007
REG : 6 : 00000009
REG : 17 : 0000000A
syscall exit
```

Percebe-se que o resultado da soma só foi escrito no registrador t1 (x6) quando o imediato era diferente do valor contido no registrador t0 (x5).

#### 4) getbfw

Para testar a instrução, foi utilizado o seguinte código:

```
.data
    n: .word 0x01234567
.text
    lw      t1,n
    addi    t0,x0,0
    #getbfw t2, t1, t0
    addi    t0,x0,1
    #getbfw t2, t1, t0
    addi    t0,x0,2
    #getbfw t2, t1, t0
    addi    t0,x0,3
    #getbfw t2, t1, t0
    addi    a7, x0, 10
    ecall # terminate ecall
```



```
REG : 6 : 00002000  
REG : 6 : 01234567  
REG : 5 : 00000000  
REG : 7 : 00000067  
REG : 5 : 00000001  
REG : 7 : 00000045  
REG : 5 : 00000002  
REG : 7 : 00000023  
REG : 5 : 00000003  
REG : 7 : 00000001  
REG : 17 : 0000000A  
syscall exit
```

Mais uma vez é possível ver que os bytes são selecionados corretamente de acordo com o valor de rs2.