

**CSE 331 – Computer Organization**  
**HW4 REPORT**

**Abdullah Çelik**  
**171044002**

## Reports:

- Since increasing the size of the register in the data memory block increases the compile time, the size is kept small.
- In data memory txt and instruction memory txt, each block (each line) is kept 1 byte (8bit). This situation does not cause any problems in the execution of the lw or sw instruction. From the incoming address, 4 bytes are loaded or stored. In Instruction memory, since it is output as 4 bytes from the instruction of the pc, there is no problem here. Then pc 4 is increased and the next instruction is passed.

Instruction list will be as follows. Respectively pc, register and memory contents will be displayed in each instruction.

Address	Instruction	Instruction Opcode
0	lw \$s0, 0(\$sp)	100011_11101_10000_0000000000000000
4	lw \$s1, 4(\$sp)	100011_11101_10001_0000000000000100
8	sw \$s2, 8(\$sp)	101011_11101_10010_0000000000001000
12	sw \$s3, 12(\$sp)	101011_11101_10011_0000000000001100
16	j Next	000010_0000000000000000000000111
20	addn \$2, \$s0, \$s1	000000_10000_10001_00010_00000_100000
24	addn \$2, \$s0, \$s1	000000_10000_10001_00010_00000_100000
28	Next: j Next2	000010_00000000000000000000001010
32	addn \$2, \$s0, \$s1	000000_10000_10001_00010_00000_100000
36	addn \$2, \$s0, \$s1	000000_10000_10001_00010_00000_100000
40	Next2: jal Next3	000011_00000000000000000000001101
44	addn \$2, \$s0, \$s1	000000_10000_10001_00010_00000_100000
48	addn \$2, \$s0, \$s1	000000_10000_10001_00010_00000_100000
52	Next3: jal Next4	000011_00000000000000000000001000
56	addn \$2, \$s0, \$s1	000000_10000_10001_00010_00000_100000
60	addn \$2, \$s0, \$s1	000000_10000_10001_00010_00000_100000
64	Next4: jr \$s4	000000_10100_00000_00000_00000_001000
68	addn \$2, \$s0, \$s1	000000_10000_10001_00010_00000_100000
72	addn \$2, \$s0, \$s1	000000_10000_10001_00010_00000_100000
76	jr \$s5	000000_10101_00000_00000_00000_001000
80	addn \$2, \$s0, \$s1	000000_10000_10001_00010_00000_100000
84	addn \$2, \$s0, \$s1	000000_10000_10001_00010_00000_100000
88	beq \$s0, \$s0, Next5	000100_10000_10000_0000000000000010
92	False: addn \$2, \$s0, \$s1	000000_10000_10001_00010_00000_100000
96	addn \$2, \$s0, \$s1	000000_10000_10001_00010_00000_100000
100	Next5: beq \$s0, \$s1, False	000100_10000_10001_1111111111111101
104	bne \$s0, \$s1, Next6	000101_10000_10001_0000000000000010
108	addn \$2, \$s0, \$s1	000000_10000_10001_00010_00000_100000
112	addn \$2, \$s0, \$s1	000000_10000_10001_00010_00000_100000
116	Next6: bne \$s0, \$s0, False	000101_10000_10000_1111111111111001
120	addn \$2, \$s0, \$s1	000000_10000_10001_00010_00000_100000
124	addn \$3, \$s2, \$s3	000000_10010_10011_00011_00000_100000
128	subn \$4, \$s0, \$s1	000000_10000_10001_00100_00000_100010
132	subn \$5, \$s2, \$s3	000000_10010_10011_00101_00000_100010
136	xorn \$6, \$s0, \$s1	000000_10000_10001_00110_00000_100001
140	xorn \$7, \$s2, \$s3	000000_10010_10011_00111_00000_100001
144	andn \$23, \$s0, \$s1	000000_10000_10001_10111_00000_100100
148	andn \$24, \$s2, \$s3	000000_10010_10011_11000_00000_100100
152	orn \$25, \$s0, \$s1	000000_10000_10001_11001_00000_100101
156	orn \$26, \$s2, \$s3	000000_10010_10011_11010_00000_100101
160	ori \$27, \$s0, 127	001101_10000_11011_0000000001111111
164	ori \$28, \$s2, -91	001101_10010_11100_1111111110100101
168	lui \$29, 78	001111_00000_11101_0000000001001110
172	lui \$30, -102	001111_00000_11110_1111111110011010

- lw \$s0, 0(\$sp)

Program counter (before instruction execution/after instruction execution):

[illegible]

Contents of registers (before/after):

0.	->	00000000000000000000000000000000
1.	->	00000000000000000000000000000000
2.	->	00000000000000000000000000000000
3.	->	00000000000000000000000000000000
4.	->	00000000000000000000000000000000
5.	->	00000000000000000000000000000000
6.	->	00000000000000000000000000000000
7.	->	00000000000000000000000000000000
8.	->	00000000000000000000000000000000
9.	->	00000000000000000000000000000000
10.	->	00000000000000000000000000000000
11.	->	00000000000000000000000000000000
12.	->	00000000000000000000000000000000
13.	->	00000000000000000000000000000000
14.	->	00000000000000000000000000000000
15.	->	00000000000000000000000000000000
16.	->	00000000000000000000000000000000
17.	->	00000000000000000000000000000000
18.	->	11110101010011010100111000100011
19.	->	11111110010010000110100010100100
20.	->	000000000000000000000000000001001100
21.	->	0000000000000000000000000000001011000
22.	->	000000000000000000000000000000000000
23.	->	000000000000000000000000000000000000
24.	->	000000000000000000000000000000000000
25.	->	000000000000000000000000000000000000
26.	->	000000000000000000000000000000000000
27.	->	000000000000000000000000000000000000
28.	->	000000000000000000000000000000000000
29.	->	000000000000000000000000000000000000
30.	->	000000000000000000000000000000000000
31.	->	000000000000000000000000000000000000

Contents of memory (before/after): No change in contents

- lw \$s1, 4(\$sp)

Program counter (before instruction execution/after instruction execution):

[illegible]

Contents of registers (before/after):

0. -> 00000000000000000000000000000000	0. -> 00000000000000000000000000000000
1. -> 00000000000000000000000000000000	1. -> 00000000000000000000000000000000
2. -> 00000000000000000000000000000000	2. -> 00000000000000000000000000000000
3. -> 00000000000000000000000000000000	3. -> 00000000000000000000000000000000
4. -> 00000000000000000000000000000000	4. -> 00000000000000000000000000000000
5. -> 00000000000000000000000000000000	5. -> 00000000000000000000000000000000
6. -> 00000000000000000000000000000000	6. -> 00000000000000000000000000000000
7. -> 00000000000000000000000000000000	7. -> 00000000000000000000000000000000
8. -> 00000000000000000000000000000000	8. -> 00000000000000000000000000000000
9. -> 00000000000000000000000000000000	9. -> 00000000000000000000000000000000
10. -> 00000000000000000000000000000000	10. -> 00000000000000000000000000000000
11. -> 00000000000000000000000000000000	11. -> 00000000000000000000000000000000
12. -> 00000000000000000000000000000000	12. -> 00000000000000000000000000000000
13. -> 00000000000000000000000000000000	13. -> 00000000000000000000000000000000
14. -> 00000000000000000000000000000000	14. -> 00000000000000000000000000000000
15. -> 00000000000000000000000000000000	15. -> 00000000000000000000000000000000
16. -> 00000000000000010011001110010101	16. -> 00000000000000010011001110010101
17. -> 00000000000000000000000000000000	17. -> 00000000000000010010100010101010
18. -> 11110101010011010100111000100011	18. -> 11110101010011010100111000100011
19. -> 1111110010010000110100010100100	19. -> 1111110010010000110100010100100
20. -> 0000000000000000000000000001001100	20. -> 0000000000000000000000000001001100
21. -> 0000000000000000000000000001011000	21. -> 0000000000000000000000000001011000
22. -> 0000000000000000000000000000000000	22. -> 0000000000000000000000000000000000
23. -> 0000000000000000000000000000000000	23. -> 0000000000000000000000000000000000
24. -> 0000000000000000000000000000000000	24. -> 0000000000000000000000000000000000
25. -> 0000000000000000000000000000000000	25. -> 0000000000000000000000000000000000
26. -> 0000000000000000000000000000000000	26. -> 0000000000000000000000000000000000
27. -> 0000000000000000000000000000000000	27. -> 0000000000000000000000000000000000
28. -> 0000000000000000000000000000000000	28. -> 0000000000000000000000000000000000
29. -> 0000000000000000000000000000000000	29. -> 0000000000000000000000000000000000
30. -> 0000000000000000000000000000000000	30. -> 0000000000000000000000000000000000
31. -> 0000000000000000000000000000000000	31. -> 0000000000000000000000000000000000

Contents of memory (before/after): No change in contents

- sw \$s2, 8(\$sp)

Program counter (before instruction execution/after instruction execution):

```
# pc= 000000000000000000000000000000001000
# pc= 000000000000000000000000000000001100
```

Contents of registers (before/after): No change in contents



Contents of memory (before/after):

00000000	00000000
00000001	00000001
00110011	00110011
10010101	10010101
00000000	00000000
00000001	00000001
00101000	00101000
10101010	10101010
00000000	11110101
00000000	01001101
00000000	01001110
00000000	00100011
00000000	00000000
00000000	00000000
00000000	00000000
00000000	00000000
00000000	00000000
00000000	00000000
00000000	00000000
00000000	00000000

- sw \$s3, 12(\$sp)

Program counter (before instruction execution/after instruction execution):

# pc= 000000000000000000000000000001100  
# pc= 000000000000000000000000000001000

Contents of registers (before/after): No change in contents

Contents of memory (before/after):

00000000	00000000
00000001	00000001
00110011	00110011
10010101	10010101
00000000	00000000
00000001	00000001
00101000	00101000
10101010	10101010
11110101	11110101
01001101	01001101
01001110	01001110
00100011	00100011
00000000	11111110
00000000	01001000
00000000	01101000
00000000	10100100
00000000	00000000
00000000	00000000
00000000	00000000
00000000	00000000

- j Next

Program counter (before instruction execution/after instruction execution):

```
# pc= 0000000000000000000000000000000010000
```

```
# pc= 0000000000000000000000000000000011100
```

Contents of registers (before/after): No change in contents

Contents of memory (before/after): No change in content

After the j Next instruction, the 20 and 24 addressed instructions are not executed and set to pc 28.

- j Next2

Program counter (before instruction execution/after instruction execution):

```
# pc= 0000000000000000000000000000000011100
```

```
# pc= 00000000000000000000000000000000101000
```

Contents of registers (before/after): No change in contents

Contents of memory (before/after): No change in contents

After the j Next2 instruction, the 32 and 36 addressed instructions are not executed and set to pc 40.

- jal Next3

Program counter (before instruction execution/after instruction execution):

```
# pc= 00000000000000000000000000000000101000
```

```
# pc= 00000000000000000000000000000000110100
```

Contents of registers (before/after):

0. -> 00000000000000000000000000000000	0. -> 00000000000000000000000000000000
1. -> 00000000000000000000000000000000	1. -> 00000000000000000000000000000000
2. -> 00000000000000000000000000000000	2. -> 00000000000000000000000000000000
3. -> 00000000000000000000000000000000	3. -> 00000000000000000000000000000000
4. -> 00000000000000000000000000000000	4. -> 00000000000000000000000000000000
5. -> 00000000000000000000000000000000	5. -> 00000000000000000000000000000000
6. -> 00000000000000000000000000000000	6. -> 00000000000000000000000000000000
7. -> 00000000000000000000000000000000	7. -> 00000000000000000000000000000000
8. -> 00000000000000000000000000000000	8. -> 00000000000000000000000000000000
9. -> 00000000000000000000000000000000	9. -> 00000000000000000000000000000000
10. -> 00000000000000000000000000000000	10. -> 00000000000000000000000000000000
11. -> 00000000000000000000000000000000	11. -> 00000000000000000000000000000000
12. -> 00000000000000000000000000000000	12. -> 00000000000000000000000000000000
13. -> 00000000000000000000000000000000	13. -> 00000000000000000000000000000000
14. -> 00000000000000000000000000000000	14. -> 00000000000000000000000000000000
15. -> 00000000000000000000000000000000	15. -> 00000000000000000000000000000000
16. -> 00000000000000010011001110010101	16. -> 00000000000000010011001110010101
17. -> 00000000000000000000000000000000	17. -> 00000000000000010010100010101010
18. -> 11110101010011010100111000100011	18. -> 11110101010011010100111000100011
19. -> 1111110010010000110100010100100	19. -> 1111110010010000110100010100100
20. -> 0000000000000000000000000001001100	20. -> 0000000000000000000000000001001100
21. -> 0000000000000000000000000001011000	21. -> 0000000000000000000000000001011000
22. -> 0000000000000000000000000000000000	22. -> 0000000000000000000000000000000000
23. -> 0000000000000000000000000000000000	23. -> 0000000000000000000000000000000000
24. -> 0000000000000000000000000000000000	24. -> 0000000000000000000000000000000000
25. -> 0000000000000000000000000000000000	25. -> 0000000000000000000000000000000000
26. -> 0000000000000000000000000000000000	26. -> 0000000000000000000000000000000000
27. -> 0000000000000000000000000000000000	27. -> 0000000000000000000000000000000000
28. -> 0000000000000000000000000000000000	28. -> 0000000000000000000000000000000000
29. -> 0000000000000000000000000000000000	29. -> 0000000000000000000000000000000000
30. -> 0000000000000000000000000000000000	30. -> 0000000000000000000000000000000000
31. -> 0000000000000000000000000000000000	31. -> 0000000000000000000000000000000000

Contents of memory (before/after): No change in contents

After the jal Next3 instruction, the 44 and 48 addressed instructions are not executed and set to pc 48. 44 address is written to \$ra(\$31) register.

- jal Next4

Program counter (before instruction execution/after instruction execution):

```
# pc= 00000000000000000000000000000110100
# pc= 000000000000000000000000000001000000
```

Contents of registers (before/after):

0. -> 00000000000000000000000000000000	0. -> 00000000000000000000000000000000
1. -> 00000000000000000000000000000000	1. -> 00000000000000000000000000000000
2. -> 00000000000000000000000000000000	2. -> 00000000000000000000000000000000
3. -> 00000000000000000000000000000000	3. -> 00000000000000000000000000000000
4. -> 00000000000000000000000000000000	4. -> 00000000000000000000000000000000
5. -> 00000000000000000000000000000000	5. -> 00000000000000000000000000000000
6. -> 00000000000000000000000000000000	6. -> 00000000000000000000000000000000
7. -> 00000000000000000000000000000000	7. -> 00000000000000000000000000000000
8. -> 00000000000000000000000000000000	8. -> 00000000000000000000000000000000
9. -> 00000000000000000000000000000000	9. -> 00000000000000000000000000000000
10. -> 00000000000000000000000000000000	10. -> 00000000000000000000000000000000
11. -> 00000000000000000000000000000000	11. -> 00000000000000000000000000000000
12. -> 00000000000000000000000000000000	12. -> 00000000000000000000000000000000
13. -> 00000000000000000000000000000000	13. -> 00000000000000000000000000000000
14. -> 00000000000000000000000000000000	14. -> 00000000000000000000000000000000
15. -> 00000000000000000000000000000000	15. -> 00000000000000000000000000000000
16. -> 00000000000000010011001110010101	16. -> 00000000000000010011001110010101
17. -> 00000000000000010010100010101010	17. -> 00000000000000010010100010101010
18. -> 11110101010011010100111000100011	18. -> 11110101010011010100111000100011
19. -> 1111110010010000110100010100100	19. -> 1111110010010000110100010100100
20. -> 0000000000000000000000000001001100	20. -> 0000000000000000000000000001001100
21. -> 0000000000000000000000000001011000	21. -> 0000000000000000000000000001011000
22. -> 0000000000000000000000000000000000	22. -> 0000000000000000000000000000000000
23. -> 0000000000000000000000000000000000	23. -> 0000000000000000000000000000000000
24. -> 0000000000000000000000000000000000	24. -> 0000000000000000000000000000000000
25. -> 0000000000000000000000000000000000	25. -> 0000000000000000000000000000000000
26. -> 0000000000000000000000000000000000	26. -> 0000000000000000000000000000000000
27. -> 0000000000000000000000000000000000	27. -> 0000000000000000000000000000000000
28. -> 0000000000000000000000000000000000	28. -> 0000000000000000000000000000000000
29. -> 0000000000000000000000000000000000	29. -> 0000000000000000000000000000000000
30. -> 0000000000000000000000000000000000	30. -> 0000000000000000000000000000000000
31. -> 00000000000000000000000000000101100	31. -> 00000000000000000000000000000111000

Contents of memory (before/after): No change in contents

After the jal Next4 instruction, the 56 and 60 addressed instructions are not executed and set to pc 64. 56 address is written to \$ra(\$31) register.

- jr \$s4

Program counter (before instruction execution/after instruction execution):

```
# pc= 000000000000000000000000000001000000
# pc= 000000000000000000000000000001001100
```

Contents of registers (before/after): No change in contents

Contents of memory (before/after): No change in contents

After the jr \$s4 instruction, the 68 and 72 addressed instructions are not executed and set to pc 76.

- jr \$s5

Program counter (before instruction execution/after instruction execution):

```
# pc= 000000000000000000000000000001001100
# pc= 000000000000000000000000000001011000
```



Contents of memory (before/after): No change in contents

```
beq $s0, $s0, Next5
```

```
# pc= 000000000000000000000000000000001011000
```

```
# pc= 000000000000000000000000000000000000000000000000000
```

Contents of memory (before/after): No change in contents

```
beq $s0, $s1, False
```

[illegible]

```
# pc= 000000000000000000000000000000001101000
```

Contents of memory (before/after): No change in contents

```
bne $s0, $s1, Next6
```

```
# pc= 000000000000000000000000000000001101000
```

```
# pc= 000000000000000000000000000000001110100
```

Contents of memory (before/after): No change in contents

```
bne $s0, $s0, False
```

```
# pc= 000000000000000000000000000000001110100
```

```
# pc= 000000000000000000000000000000001111000
```

Contents of registers(before/after): No change in contents



Contents of registers (before/after):

0. -> 00000000000000000000000000000000	0. -> 00000000000000000000000000000000
1. -> 00000000000000000000000000000000	1. -> 00000000000000000000000000000000
2. -> 00000000000000000000000000000011	2. -> 00000000000000000000000000000011
3. -> 00000000000000000000000000000000	3. -> 00000000000000000000000000000010
4. -> 00000000000000000000000000000000	4. -> 00000000000000000000000000000000
5. -> 00000000000000000000000000000000	5. -> 00000000000000000000000000000000
6. -> 00000000000000000000000000000000	6. -> 00000000000000000000000000000000
7. -> 00000000000000000000000000000000	7. -> 00000000000000000000000000000000
8. -> 00000000000000000000000000000000	8. -> 00000000000000000000000000000000
9. -> 00000000000000000000000000000000	9. -> 00000000000000000000000000000000
10. -> 00000000000000000000000000000000	10. -> 00000000000000000000000000000000
11. -> 00000000000000000000000000000000	11. -> 00000000000000000000000000000000
12. -> 00000000000000000000000000000000	12. -> 00000000000000000000000000000000
13. -> 00000000000000000000000000000000	13. -> 00000000000000000000000000000000
14. -> 00000000000000000000000000000000	14. -> 00000000000000000000000000000000
15. -> 00000000000000000000000000000000	15. -> 00000000000000000000000000000000
16. -> 00000000000000100101110000111111	16. -> 00000000000000100101110000111111
17. -> 00000000000000010010100010101010	17. -> 00000000000000010010100010101010
18. -> 11110101010011010100111000100011	18. -> 11110011100101011011011011000111
19. -> 1111110010010000110100010100100	19. -> 1111110010010000110100010100100
20. -> 000000000000000000000000001001100	20. -> 000000000000000000000000001001100
21. -> 000000000000000000000000001011000	21. -> 000000000000000000000000001011000
22. -> 000000000000000000000000000000000	22. -> 000000000000000000000000000000000
23. -> 000000000000000000000000000000000	23. -> 000000000000000000000000000000000
24. -> 000000000000000000000000000000000	24. -> 000000000000000000000000000000000
25. -> 000000000000000000000000000000000	25. -> 000000000000000000000000000000000
26. -> 000000000000000000000000000000000	26. -> 000000000000000000000000000000000
27. -> 000000000000000000000000000000000	27. -> 000000000000000000000000000000000
28. -> 000000000000000000000000000000000	28. -> 000000000000000000000000000000000
29. -> 000000000000000000000000000000000	29. -> 000000000000000000000000000000000
30. -> 000000000000000000000000000000000	30. -> 000000000000000000000000000000000
31. -> 000000000000000000000000000000111000	31. -> 000000000000000000000000000000111000

Contents of memory (before/after): No change in contents



- subn \$4, \$s0, \$s1

Program counter (before instruction execution/after instruction execution):

```
# pc= 00000000000000000000000000000000000000000000000000000
```

```
# pc= 0000000000000000000000000000000010000100
```

Contents of registers (before/after):

[illegible]

Contents of memory (before/after): No change in contents



- subn \$5, \$s2, \$s3

Program counter (before instruction execution/after instruction execution):

[illegible]

Contents of registers (before/after):

[illegible]

Contents of memory (before/after): No change in contents

- xorn \$6, \$s0, \$s1

Program counter (before instruction execution/after instruction execution):

[illegible]

### Contents of registers (before/after):

[illegible]

Contents of memory (before/after): No change in contents

- xorn \$7, \$s2, \$s3

Program counter (before instruction execution/after instruction execution):

```
# pc= 0000000000000000000000000000000000000000000000000
```

```
# pc= 000000000000000000000000010010000
```

Contents of registers (before/after):

[illegible]

Contents of memory (before/after): No change in contents



- andn \$23, \$s0, \$s1

Program counter (before instruction execution/after instruction execution):

```
# pc= 0000000000000000000000000000000010010000
```

```
# pc= 0000000000000000000000000000000000000000000000000000000
```

### Contents of registers (before/after):

[illegible]

Contents of memory (before/after): No change in contents



- andn \$24, \$s2, \$s3

Program counter (before instruction execution/after instruction execution):

```
# pc= 000000000000000000000000010010100
```

[illegible]

### Contents of registers (before/after):

0.	->	00000000000000000000000000000000
1.	->	00000000000000000000000000000000
2.	->	00000000000000000000000000000011
3.	->	00000000000000000000000000000010
4.	->	00000000000000000000000000000011
5.	->	00000000000000000000000000000010
6.	->	00000000000000000000000000000011
7.	->	00000000000000000000000000000011
8.	->	00000000000000000000000000000000
9.	->	00000000000000000000000000000000
10.	->	00000000000000000000000000000000
11.	->	00000000000000000000000000000000
12.	->	00000000000000000000000000000000
13.	->	00000000000000000000000000000000
14.	->	00000000000000000000000000000000
15.	->	00000000000000000000000000000000
16.	->	000000000000000000000000100000101010
17.	->	00000000000000001001010001010101010
18.	->	00001011000001010010011010000111
19.	->	11111110010010000110100010100100
20.	->	000000000000000000000000001001100
21.	->	000000000000000000000000001011000
22.	->	000000000000000000000000000000000
23.	->	000000000000000000000000000000011
24.	->	000000000000000000000000000000000
25.	->	000000000000000000000000000000000
26.	->	000000000000000000000000000000000
27.	->	000000000000000000000000000000000
28.	->	000000000000000000000000000000000
29.	->	000000000000000000000000000000000
30.	->	000000000000000000000000000000000
31.	->	0000000000000000000000000000111000

Contents of memory (before/after): No change in contents

- orn \$25, \$s0, \$s1

Program counter (before instruction execution/after instruction execution):

```
# pc= 0000000000000000000000000000000000000000000000000
```

```
# pc= 0000000000000000000000000000000010011100
```

### Contents of registers (before/after):

0.	->	00000000000000000000000000000000
1.	->	00000000000000000000000000000000
2.	->	00000000000000000000000000000011
3.	->	00000000000000000000000000000010
4.	->	00000000000000000000000000000011
5.	->	00000000000000000000000000000010
6.	->	00000000000000000000000000000011
7.	->	00000000000000000000000000000011
8.	->	00000000000000000000000000000000
9.	->	00000000000000000000000000000000
10.	->	00000000000000000000000000000000
11.	->	00000000000000000000000000000000
12.	->	00000000000000000000000000000000
13.	->	00000000000000000000000000000000
14.	->	00000000000000000000000000000000
15.	->	00000000000000000000000000000000
16.	->	00000000000000000000000000000000
17.	->	00000000000000000000000000000000
18.	->	00001010000000000000000000000000
19.	->	1111110010010000110100010100100
20.	->	00000000000000000000000000000000
21.	->	00000000000000000000000000000000
22.	->	00000000000000000000000000000000
23.	->	00000000000000000000000000000000
24.	->	00000000000000000000000000000000
25.	->	00000000000000000000000000000000
26.	->	00000000000000000000000000000000
27.	->	00000000000000000000000000000000
28.	->	00000000000000000000000000000000
29.	->	00000000000000000000000000000000
30.	->	00000000000000000000000000000000
31.	->	00000000000000000000000000000000

Contents of memory (before/after): No change in contents

- orn \$26, \$s2, \$s3

Program counter (before instruction execution/after instruction execution):

[illegible]

### Contents of registers (before/after):

0.	->	00000000000000000000000000000000
1.	->	00000000000000000000000000000000
2.	->	00000000000000000000000000000011
3.	->	00000000000000000000000000000010
4.	->	00000000000000000000000000000011
5.	->	00000000000000000000000000000010
6.	->	00000000000000000000000000000011
7.	->	00000000000000000000000000000011
8.	->	00000000000000000000000000000000
9.	->	00000000000000000000000000000000
10.	->	00000000000000000000000000000000
11.	->	00000000000000000000000000000000
12.	->	00000000000000000000000000000000
13.	->	00000000000000000000000000000000
14.	->	00000000000000000000000000000000
15.	->	00000000000000000000000000000000
16.	->	00000000000000010010100010101010
17.	->	00000000000000010010100010101010
18.	->	0000101000000000010000010000100
19.	->	1111110010010000110100010100100
20.	->	0000000000000000000000000001001100
21.	->	00000000000000000000000000001011000
22.	->	00000000000000000000000000000000
23.	->	000000000000000000000000000000011
24.	->	000000000000000000000000000000011
25.	->	000000000000000000000000000000011
26.	->	00000000000000000000000000000000
27.	->	00000000000000000000000000000000
28.	->	00000000000000000000000000000000
29.	->	00000000000000000000000000000000
30.	->	00000000000000000000000000000000
31.	->	0000000000000000000000000000111000

Contents of memory (before/after): No change in contents



- ori \$27, \$s0, 127

Program counter (before instruction execution/after instruction execution):

```
# pc= 0000000000000000000000000000000010100000
```

```
# pc= 0000000000000000000000000000000010100100
```

### Contents of registers (before/after):

[illegible]

Contents of memory (before/after): No change in contents



```
- ori $28, $s2, -91
```

Program counter (before instruction execution/after instruction execution):

```
# pc= 0000000000000000000000000010100100
```

```
# pc= 000000000000000000000000010101000
```

Contents of registers (before/after):

```

0. -> 00000000000000000000000000000000
1. -> 00000000000000000000000000000000
2. -> 00000000000000000000000000000011
3. -> 00000000000000000000000000000010
4. -> 00000000000000000000000000000011
5. -> 00000000000000000000000000000010
6. -> 00000000000000000000000000000011
7. -> 00000000000000000000000000000011
8. -> 00000000000000000000000000000000
9. -> 00000000000000000000000000000000
10. -> 00000000000000000000000000000000
11. -> 00000000000000000000000000000000
12. -> 00000000000000000000000000000000
13. -> 00000000000000000000000000000000
14. -> 00000000000000000000000000000000
15. -> 00000000000000000000000000000000
16. -> 00000000000000010010100010101010
17. -> 00000000000000010010100010101010
18. -> 11111110010010000110100010100100
19. -> 11111110010010000110100010100100
20. -> 000000000000000000000000001001100
21. -> 000000000000000000000000001011000
22. -> 00000000000000000000000000000000
23. -> 00000000000000000000000000000011
24. -> 00000000000000000000000000000011
25. -> 00000000000000000000000000000011
26. -> 00000000000000000000000000000010
27. -> 00000000000000010010100011111111
28. -> 00000000000000000000000000000000
29. -> 00000000000000000000000000000000
30. -> 00000000000000000000000000000000
31. -> 00000000000000000000000000111000

```

Contents of memory (before/after): No change in contents

- lui \$29,78

Program counter (before instruction execution/after instruction execution):

```
# pc= 0000000000000000000000000000000010101000
```

```
# pc= 00000000000000000000000000000000000000000000000000000
```

Contents of registers (before/after):

```
0. -> 00000000000000000000000000000000
1. -> 00000000000000000000000000000000
2. -> 00000000000000000000000000000000
3. -> 00000000000000000000000000000001
4. -> 00000000000000000000000000000010
5. -> 00000000000000000000000000000011
6. -> 00000000000000000000000000000010
7. -> 00000000000000000000000000000011
8. -> 00000000000000000000000000000000
9. -> 00000000000000000000000000000000
10.-> 00000000000000000000000000000000
11.-> 00000000000000000000000000000000
12.-> 00000000000000000000000000000000
13.-> 00000000000000000000000000000000
14.-> 00000000000000000000000000000000
15.-> 00000000000000000000000000000000
16.-> 00000000000000010010100010101010
17.-> 00000000000000010010100010101010
18.-> 11111110010010000110100010100100
19.-> 11111110010010000110100010100100
20.-> 00000000000000000000000001001100
21.-> 000000000000000000000000001011000
22.-> 00000000000000000000000000000000
23.-> 00000000000000000000000000000001
24.-> 00000000000000000000000000000001
25.-> 00000000000000000000000000000001
26.-> 00000000000000000000000000000010
27.-> 00000000000000010010100011111111
28.-> 1111111001001000111111110100101
29.-> 00000000000000000000000000000000
30.-> 00000000000000000000000000000000
31.-> 000000000000000000000000000111000
```

Contents of memory (before/after): No change in contents

- lui \$30, -102

Program counter (before instruction execution/after instruction execution):

```
# pc= 00000000000000000000000000000000
```

```
# pc= 0000000000000000000000000000000010110000
```

### Contents of registers (before/after):

0.	->	000000000000000000000000000000
1.	->	000000000000000000000000000000
2.	->	0000000000000000000000000000011
3.	->	0000000000000000000000000000010
4.	->	0000000000000000000000000000011
5.	->	0000000000000000000000000000010
6.	->	0000000000000000000000000000011
7.	->	0000000000000000000000000000011
8.	->	0000000000000000000000000000000
9.	->	0000000000000000000000000000000
10.	->	0000000000000000000000000000000
11.	->	0000000000000000000000000000000
12.	->	0000000000000000000000000000000
13.	->	0000000000000000000000000000000
14.	->	0000000000000000000000000000000
15.	->	0000000000000000000000000000000
16.	->	00000000000000010010100010101010
17.	->	00000000000000010010100010101010
18.	->	11111110010010000110100010100100
19.	->	11111110010010000110100010100100
20.	->	00000000000000000000000001001100
21.	->	00000000000000000000000001011000
22.	->	0000000000000000000000000000000
23.	->	00000000000000000000000000000011
24.	->	00000000000000000000000000000011
25.	->	00000000000000000000000000000011
26.	->	00000000000000000000000000000010
27.	->	00000000000000010010100011111111
28.	->	1111111001001000111111110100101
29.	->	0000000001001110000000000000000
30.	->	0000000000000000000000000000000
31.	->	0000000000000000000000000111000

Contents of memory (before/after): No change in contents