INSTITUTO SUPERIOR TÉCNICO

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

ORGANIZAÇÃO DE COMPUTADORES

LEIC

Conjunto de Exercícios III

Arquitetura Lógica

Versão 1.4

2023/2024

Computer Arithmetic

Exercise 1 *

The table below contains two base-ten values for register `$s1`.

a.	-1_{ten}
b.	1024_{ten}

Assume that register `$s0 = 0x7FFFFFFF`. For each value in the table, will there be overflow if the instruction `add $s0,$s0,$s1` is executed?

Exercise 2 †

Answer the following questions assuming that (i) the instruction `add $s0,$s0,$s1` is executed, (ii) register `$s0 = 0x70000000`, and (iii) register `$s1` is initialized with the binary value:

$$0010\ 0100\ 1001\ 0010\ 0100\ 1001\ 0010\ 0100_{two}$$

(a) What are the initial values of `$s0` and `$s1` in base ten?

(b) Will there be overflow?

(c) What is the result in hexadecimal notation?

(d) What is the result in base ten?

Exercise 3

What are the results of the following operations, given the initial values of `$t0` and `$t1` indicated and the output notation requested?

(a) $t0 = 000000F0$_{16}$, $t1 = 177$_{10}$
```
and $t0,$t0,$t1
```
$t0 = _____$_{10}$

(b) $t0 = 000000A5$_{16}$, $t1 = -9$_{10}$
```
ori $t0,$t0,74
```
$t0 = _____$_{16}$

(c) $t0 = A5A55A5A$_{16}$, $t1 = 0000FFFF$_{16}$
```
xor $t0,$t0,$t1
```
$t0 = _____$_{16}$

(d) $t0 = 00005678$_{16}$
```
srl $t0,$t0,3
andi $t0,$t0,15
```
$t0 = _____$_{16}$ = _____$_{10}$

*Exercise 2.9.1 from [1].
†Exercise 2.9.2 from [1].

(e) $t0 = 00007234_{16} , $t1 = $07E0_{16}$
```
and $t0,$t0,$t1
srl $t1,$t0,5
```
$t0 = _____16 = _____10

(f) $t0 = $000000A5_{16}$, $t1 = 65536_{10}
```
andi $t0,$t1,1
```
$t0 = _____10

Exercise 4 ‡

Assuming that `$t0 = 0xAAAAAAAA` and `$t1 = 0x12345678`, find the value of `$t2` for each of the following instruction sequences:

Sequence 1	Sequence 2
`sll $t2, $t0, 4`	`srl $t2, $t0, 3`
`or $t2, $t2, $t1`	`andi $t2, $t2, 0x7FEF`

Exercise 5 §

Consider the following code fragment:

```
sll    $t2, $t0, 1
andi   $t2, $t2, -1
```

Determine the value of register `$t2` after the execution of these instructions. Compute this value for each of the following initial conditions relative to register `$t0`:

(a) `$t0 = 0x0000A5A5`

(b) `$t0 = 0xA5A5FFFF`

Instruction Set

Exercise 6 ¶

Consider the following hypothetical MIPS instruction:

```
rpt  $t2, loop   # R[rs]=R[rs]-1, if(R[rs]!=0) PC=PC+4+BranchAddr
```

(a) Why is this instruction not included in the MIPS instruction set?

(b) If this instruction were to be implemented in the MIPS instruction set, what would be the most appropriate instruction format?

(c) Find the shortest sequence of MIPS instructions that performs the same operation.

‡From exercises 2.13.1—2.13.6 in [1].
§From exercises 2.13.1—2.13.6 in [1].
¶Exercises 2.17.1—2.17.3 from [1].

Exercise 7 [‖]

Study the following MIPS code fragment:

```
LOOP:  addi  $s2, $s2, 2
       addi  $t1, $t1, -1
       bne   $t1, $0, LOOP
DONE:
```

(a) What is the value in register `$s2` assuming that `$s2` is initially zero and that register `$t1` is initialized to the value 10?

(b) Write an equivalent C code routine. Assume that the registers `$s2` and `$t1` are integers `A` and `i`, respectively.

(c) By initializing register `$t1` with the value `N`, how many MIPS instructions would be executed?

Exercise 8

Consider a non-pipelined processor [**] with a *Load-Store* architecture and an instruction execution cycle composed of 5 sequential phases, each taking one clock cycle:

Fetch → Decode → eXecute → Memory → Write-Back

All instructions step through the M phase but only the *Load* and *Store* instructions access the memory in this phase. Memory accesses are implemented using a single bus.

After a given set of benchmark programs has been executed, the occurrence rates of some classes of instructions were obtained:

Class of instruction	Ocurrence
ALU	40%
Load	28%
Store	10%
Jump	22%

The inclusion in this architecture of an extra addressing mode – *register-memory* – is being considered. This extension allows the replacement of the following code sequence:

```
lw  $t2, 0($t7)          by      add $t3, $t1, 0($t7)
add $t3, $t1, $t2
```

The introduction of this new addressing mode will reduce the amount of *Loads* required by this set of programs to 60% of the original.

(a) There are two design options to implement this new addressing mode:

A - Keep the number of processing phases unchanged, computing the effective address and accessing the memory operand in phase D, causing an increment of the clock period by 16%;

[‖] Exercises 2.17.4—2.17.6 from [1].

[**] Instruction cycles do not overlap. The processor only starts the execution cycle of the next instruction after the end of the current instruction cycle.

B - Keep the clock period unchanged and add a new stage in the instruction execution cycle, between stages D and X, to compute the effective address and implement the access to the memory operand: (D $\rightarrow$ A $\rightarrow$ X).

Determine the option that leads to the best performance.

(b) What is the reduction in the number of memory accesses that is achieved with this new architecture?

$\mathrm{Reduction} = (\mathrm{mem\_accesses\_new} - \mathrm{mem\_accesses\_orig})/\mathrm{mem\_accesses\_orig}$

References

[1] David Patterson and John Hennessy. *Computer Organization and Design: The Hardware/Software Interface*. Morgan Kaufmann, 4^{th} edition, 2011.