

Third

June 6, 2023

1 Load Word, Store Word

1.1 What do they do each?

lw: Load word instruction, loads a word (32 bits stored in the memory (4 bytes of data)) into a register,

sw: Store word on the other hand, stores a word into memory.

1.2 Which type of instructions they are?

They are **I format** instructions.

1.3 How can we use 'em in assembly code?

The syntax of using **lw** and **sw** in assembly code is like this:

```
(l/s)w $t1, 1200($s1)
```

lw: In load word instruction the first register (**\$t1** in this case) is the destination of what is in the address: 1200 + address stored in **\$s1**. That 1200 means: $4 * 300 = 1200$, go 300 steps **after** (because the number is positive, clearly **before** if it was negative) the address in register **\$s1** (because each word is 4 bytes long)

sw: In store word instruction the first register has the data to be stored. The addressing issue is dealt in the same way in both instructions.

1.4 Which signals of control unit are *don't care* for 'em?

lw: There is no *don't care* signal for **lw** instruction.

sw: **RegDst** and **MemToReg** signals are *don't care* for **sw** instruction.

1.5 Describe the data path and control unit state, meanwhile of the execution of one of these.

When **lw** or **sw** instructions are fetched from memory, they are meant to be decoded. Let's see a prototype of a fetched **lw** instruction.

```
lw $t1, 0($s1)
```

it will be:

let's assume: \$t1 -> 00000, \$s1 -> 00001

```
-----  
| 6-bit-opcode | 00001 | 00000 | 0000000000000000 |  
-----
```

So the needed signals:

- Decode phase: In this phase the needed operands are fetched and the instruction will be ready to be executed.
 - RegDest -> 0 to select Rt for the destination
 - ExtOp -> 1 to sign extend the 16-bit immediate value
 - RegWrite -> 1 'cause it needs the permission to write to register file
- Execution Phase: In this phase, ALU will add the extended value and the address in Rs and produce a memory address and will get the value in there
 - ALUSrc -> 1 to select the extended value to be one of the inputs of ALU
 - MemWrite -> 0 it should not write anything to memory
 - MemToReg -> 1 'cause it has to choose the output of memory not ALU
- Write back: The got value will be written in register file
 - MemRead -> 1 it should have the permission to read from memory
 - Branch -> 0 we do not branch in lw