



# CS305: Computer Architecture

World of Instructions-II (The MIPS language)

<https://www.cse.iitb.ac.in/~biswa/courses/CS305/main.html>

<https://www.cse.iitb.ac.in/~biswa/>

Trivia? How to store a 32-bit constant into a 32-bit register?

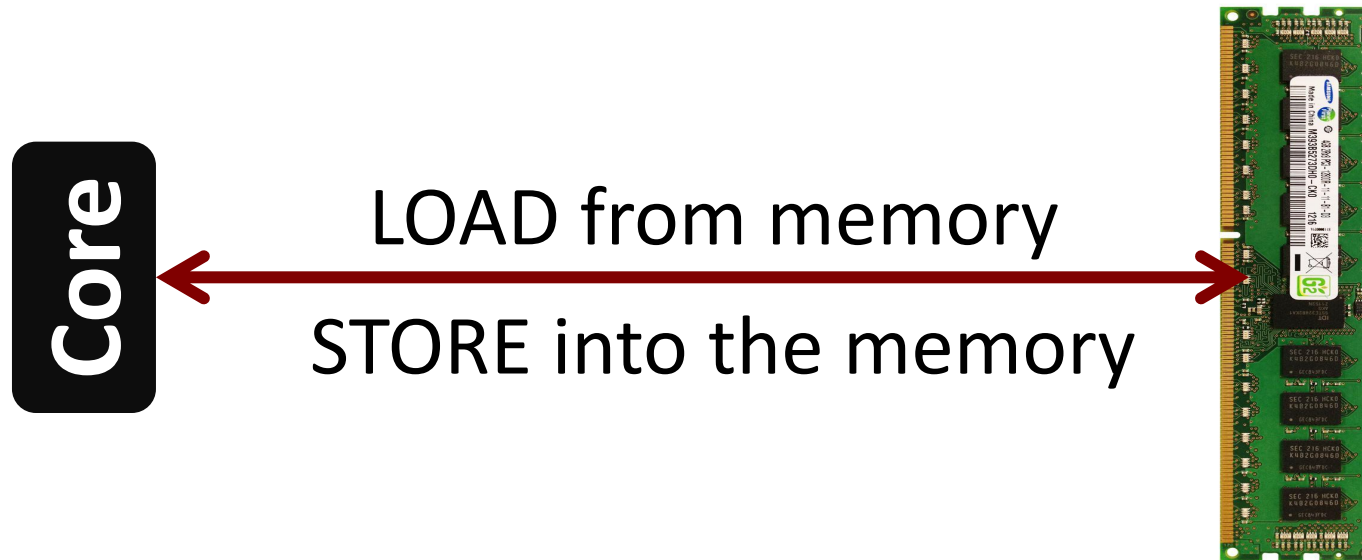
For example, **10101010 10101010 11110000 11110000**

lui \$t0, 0xAAAA #**1010101010101010**, lower bits all 0s.

ori \$t0, \$t0, 0xF0F0 #**1111000011110000**

lui: upper bits, ori/addi: lower bits

# Memory Instructions



```
lw $t0, 1($a0)    # $t0 = Memory[$a0 + 1]  
sw $t0, 1($a0)    # Memory[$a0 + 1] = $t0
```

Stored  
Program  
&  
Von  
Neumann



# Before Von Neumann (Pre 1944-45)

- Memory stores only data that is needed to perform an operation
- One program at a time
- 1944: Instructions: stored in memory
- Hence **stored program** as the binary is stored in memory

# Memory



4GB of Memory (DRAM)

Say, a **word**: four bytes

# How to access instructions: Program Counter (PC)

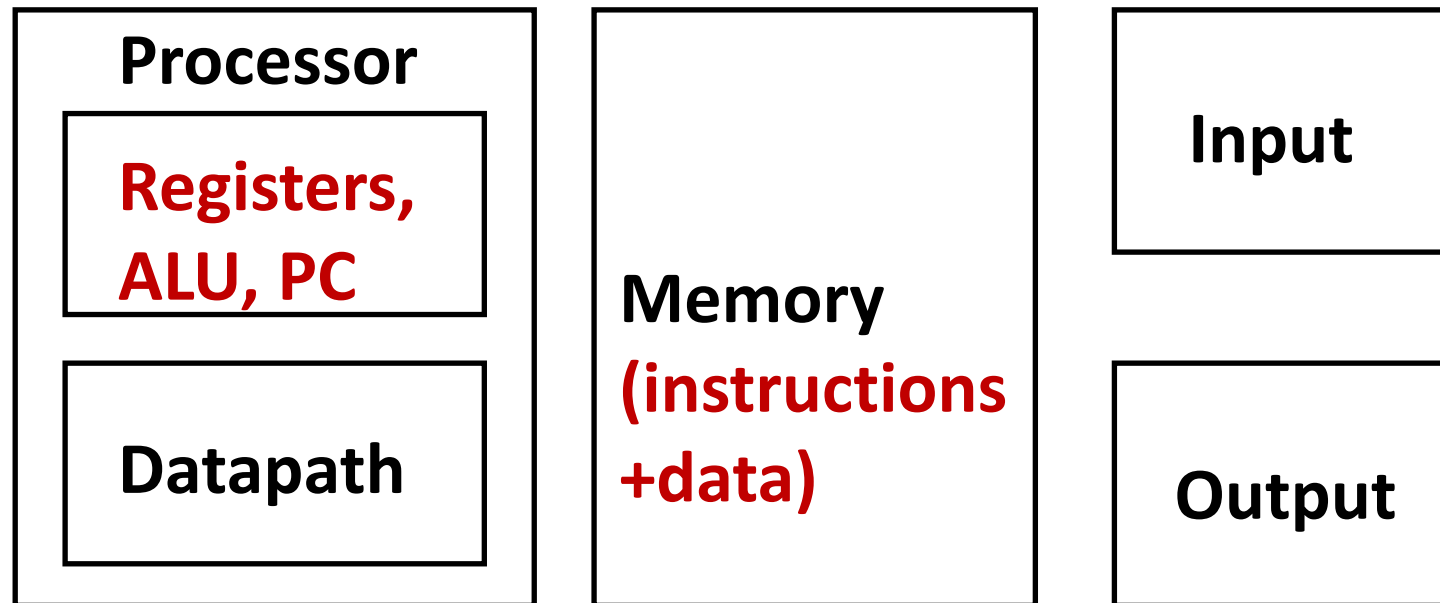
A register that stores the address of the instruction

32-bit processor: addresses are of width 32 bits (devil is in the details 😊 )

So the processor fetches PC, PC+4, PC+8, ..... in a **sequential order**

# 1946 onwards (Remember Lecture-0)

Since 1946 all computers have had 5 components





Example (Remember PC for the time being)

PCX: lw

PCY: add

PCZ: lui

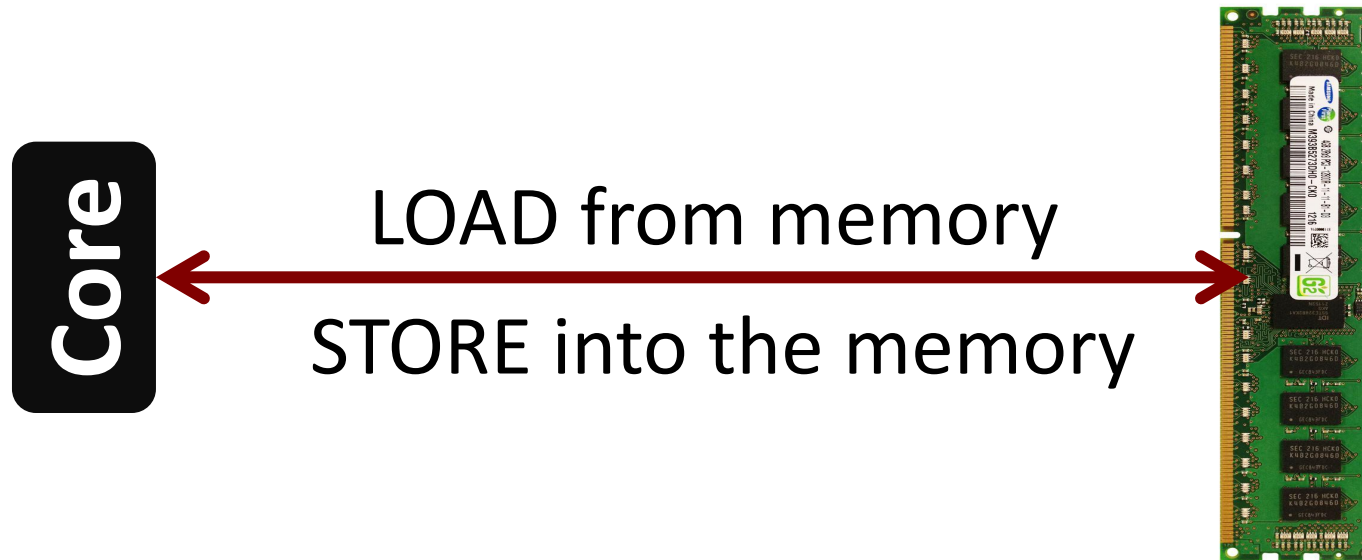
$PCZ = PCY + 4$  and  $PCY = PCX + 4$



# Why Memory? Why Not Registers?

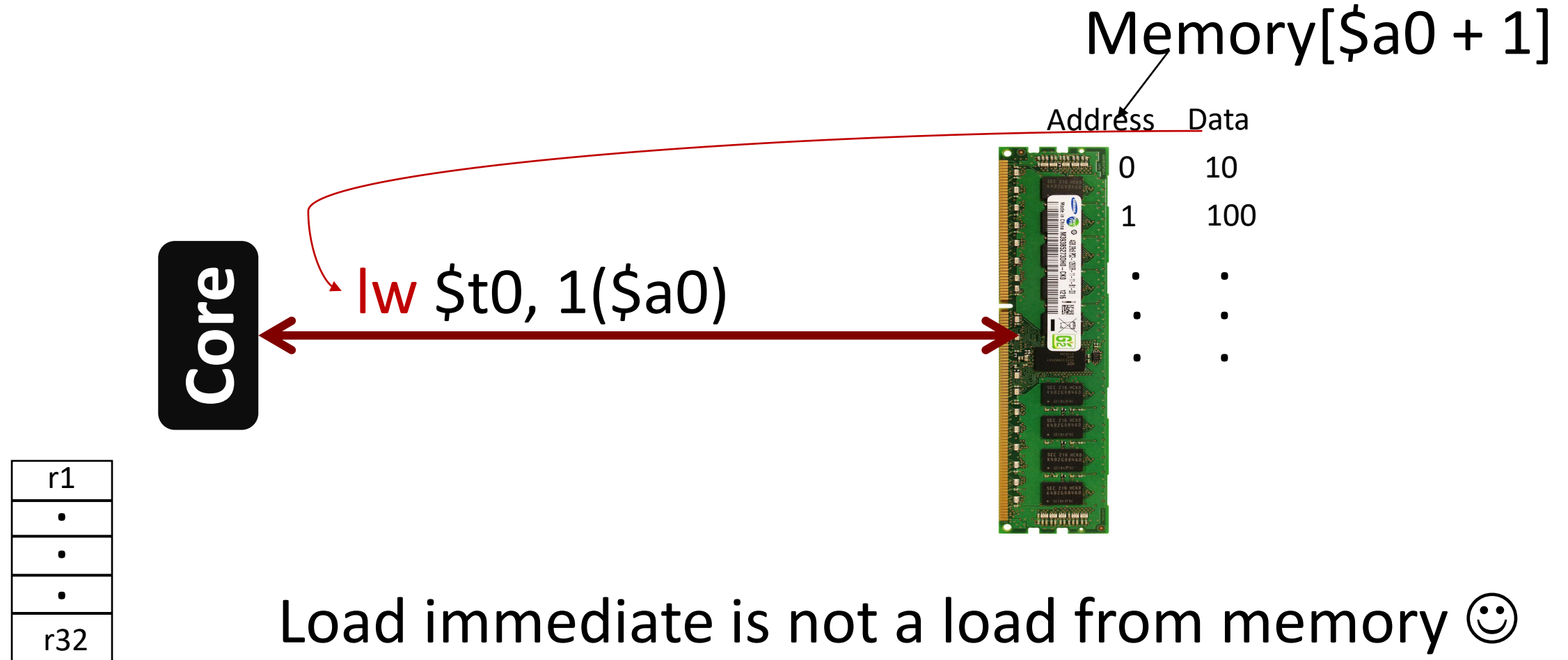
- Registers are limited. More #registers, higher access time.
- How? we will see sooner than later.
- Let's focus on the data part now. How to access data for our instructions?

# Memory Instructions



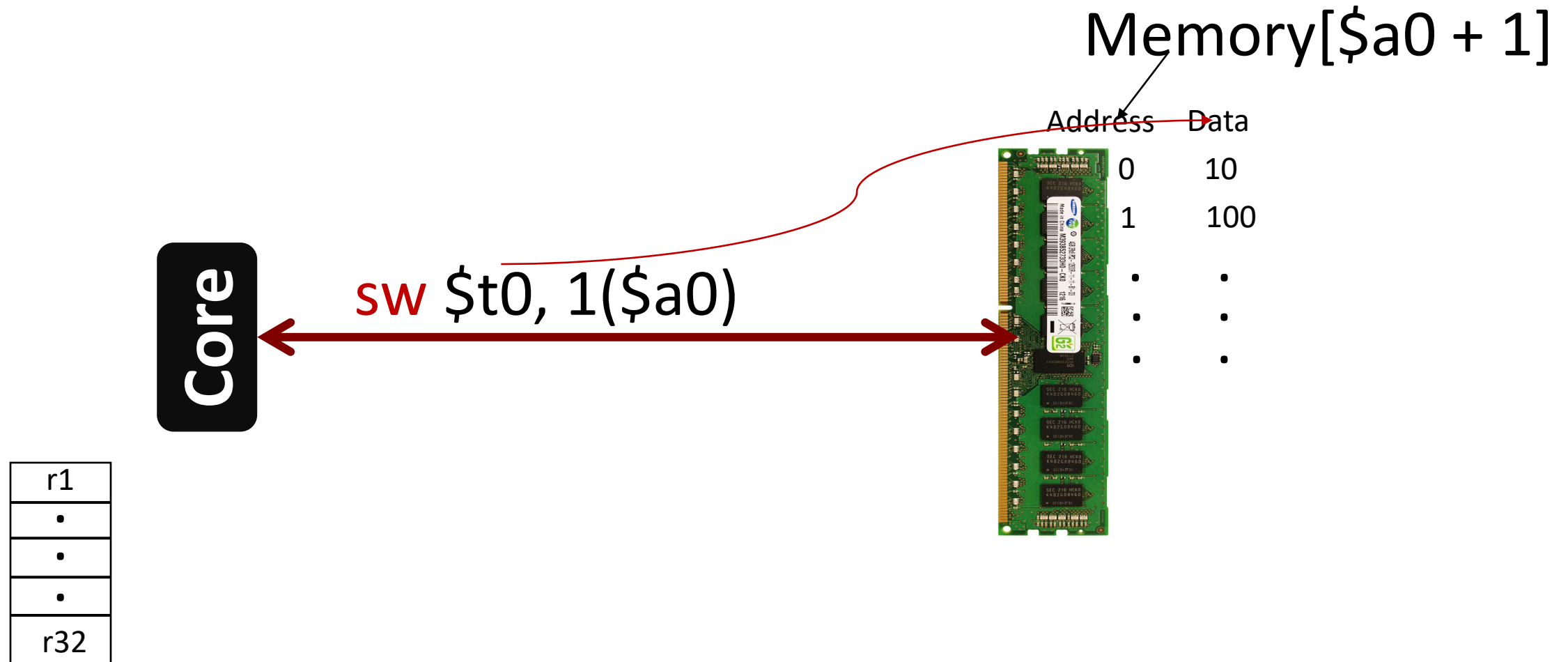
```
lw $t0, 1($a0)    # $t0 = Memory[$a0 + 1]  
sw $t0, 1($a0)    # Memory[$a0 + 1] = $t0
```

# LOAD From the Memory (data-transfer insts)



Load immediate is not a load from memory 😊

# STORE



Both instructions and data from memory

$g = h + A[8];$

**PCX:** lw \$t0, 8(\$3) # A[8]

**PCY:** add \$s1, \$s2, \$t0 #  $g = h + t0$

$PCY = PCX + 4$



Merci