

Computer Architecture

Instruction Set Architecture

Memory

4,294,967,295
 $(= 2^{32}-1)$



Slow

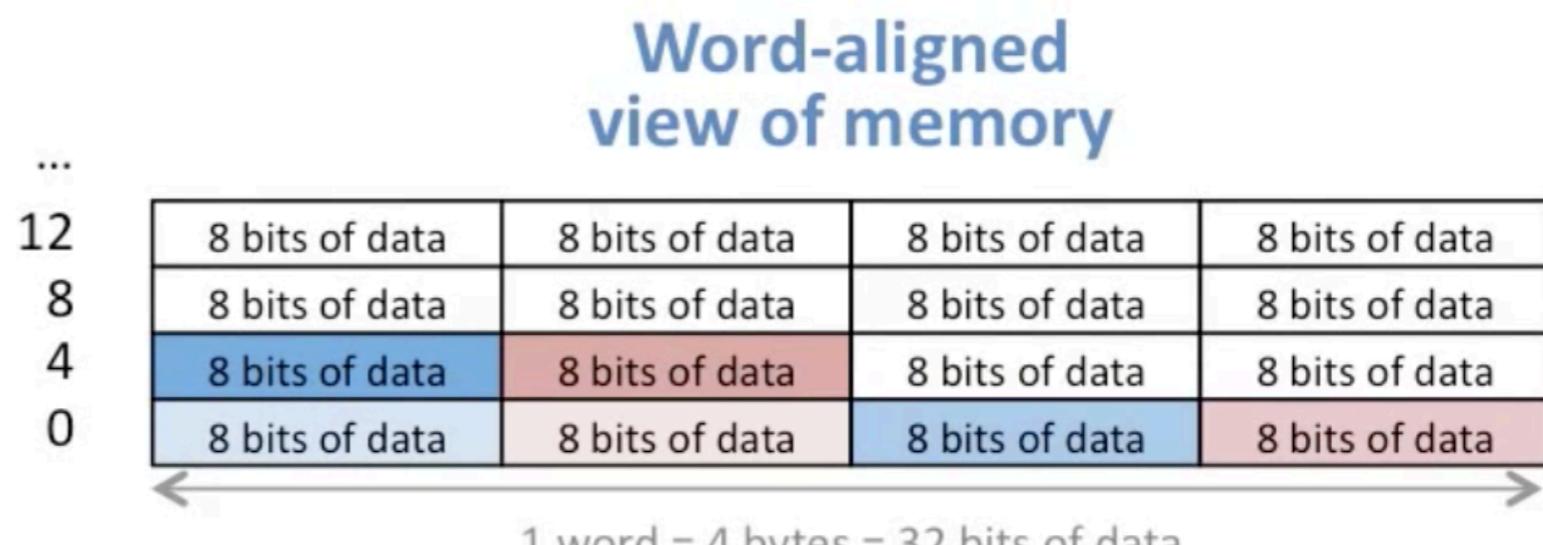
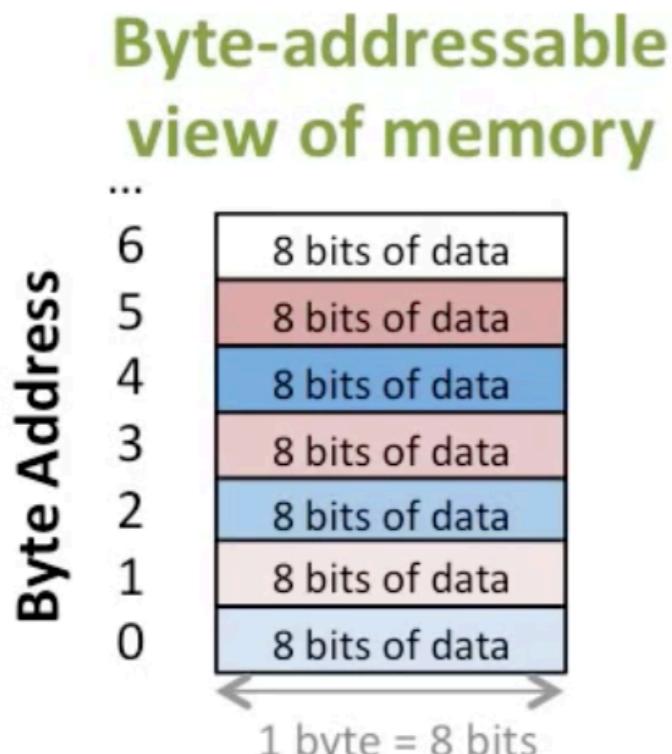
Register File



Fast

→ 4 Billion Memory Locations ←

→ 32 Registers ←



Registers hold 32 bits of data (1 word)
Addresses are 32 bits of data (1 word)

Most data in MIPS is handled in **words** not **bytes**

- A **word** is 32 bits or 4 bytes
- A **word** is the size of a register

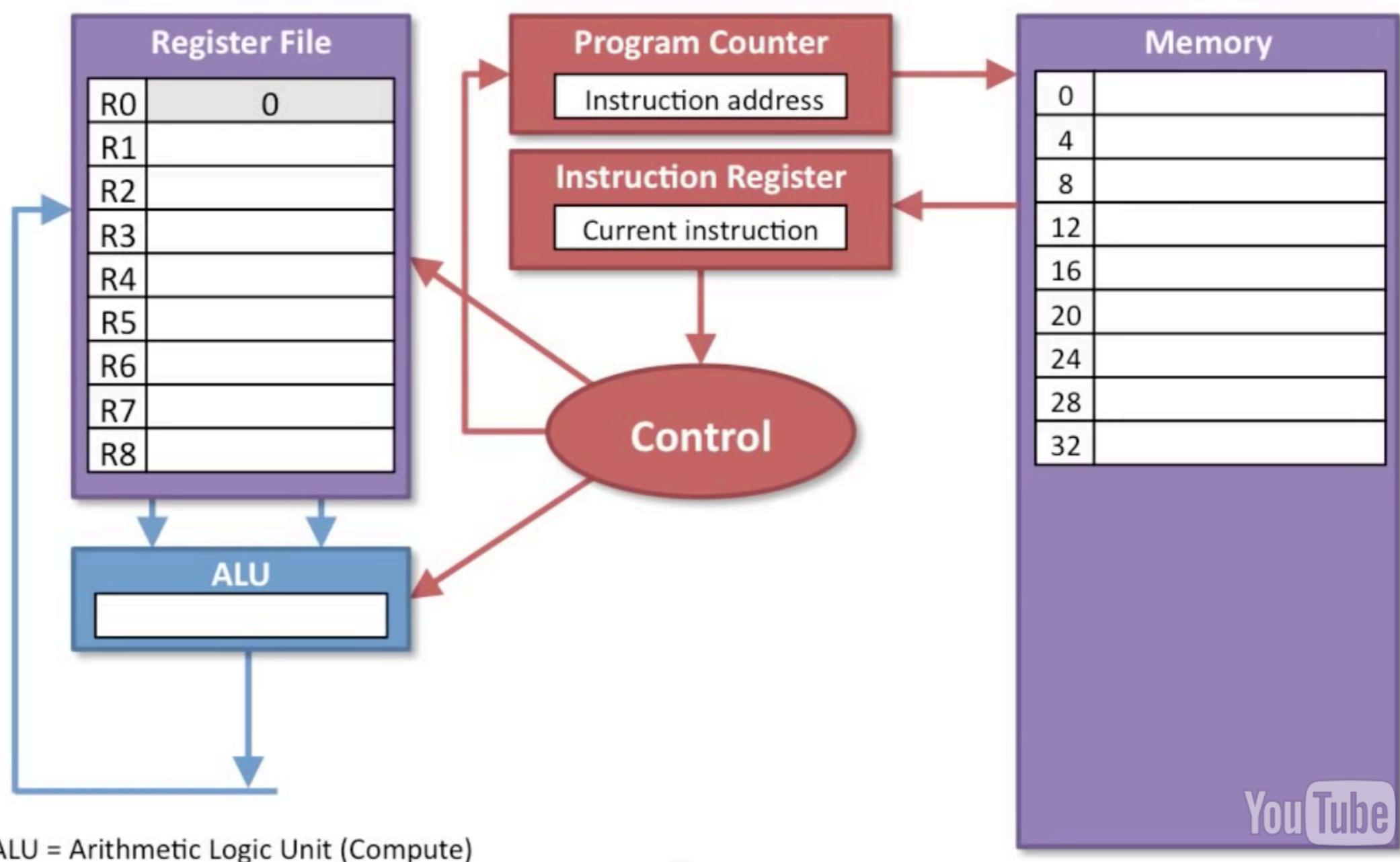
There are $2^{32} - 1$ (4, 294, 967, 296 - 1 = 4, 294, 967, 295) byte-addressable memory locations in a MIPS memory. How many word-addressable locations are there?

- MIPS memory is 2^{32} (**4, 294, 967, 296**) in size
- There are **4 bytes** (32 bits) in one word
- Word-addressable memory locations is:
 $4, 294, 967, 296 / 4 = \underline{\textcolor{red}{1, 073, 741, 824}}$

In a MIPS processor, which register holds the instruction address?

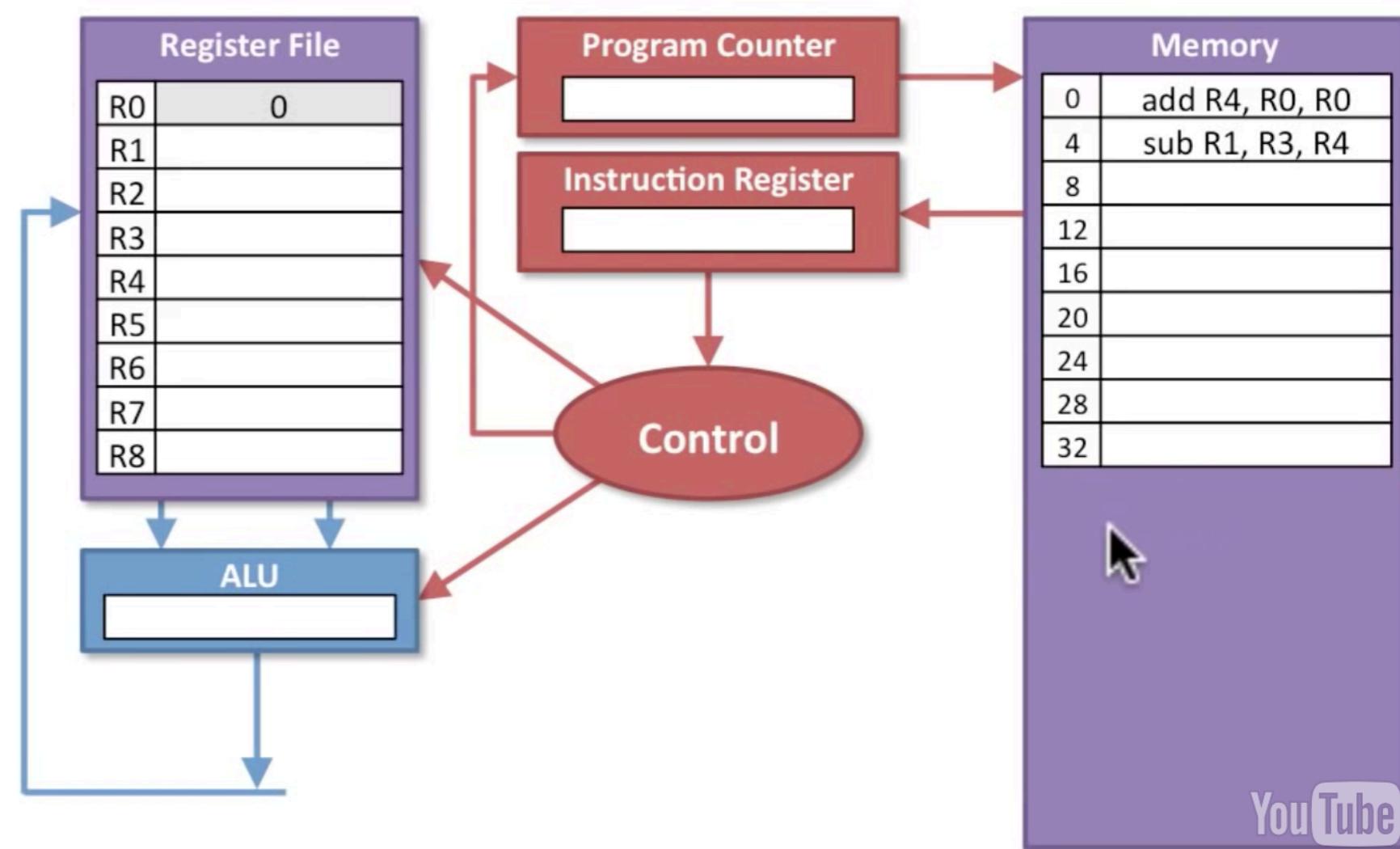
- A. Instruction Register
- B. Segment Register
- C. Program Counter
- D. Index Register
- E. None of the above

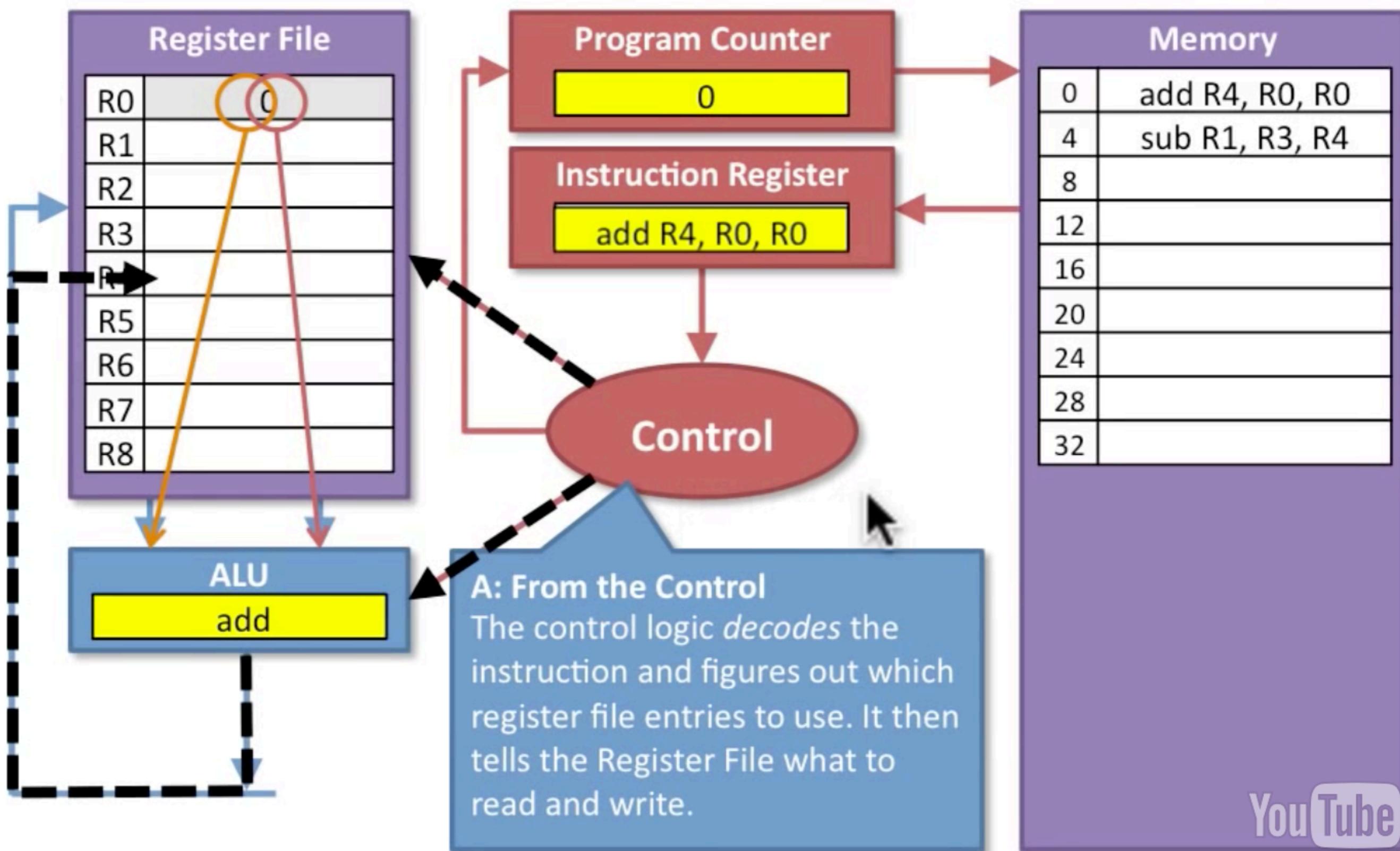
1. Program Counter holds the instruction address.
2. Instructions are *fetched* from memory into the Instruction Register.
3. Control logic *decodes* the instruction and tells the ALU and Register File what to do.
4. ALU *executes* the instruction and results flow back to the Register File.
5. The Control logic *updates* the Program Counter for the next instruction.



Which does the Register File get the information on which registers to read and write from?

- A. Program Counter
- B. Instruction Register
- C. Control Logic
- D. ALU

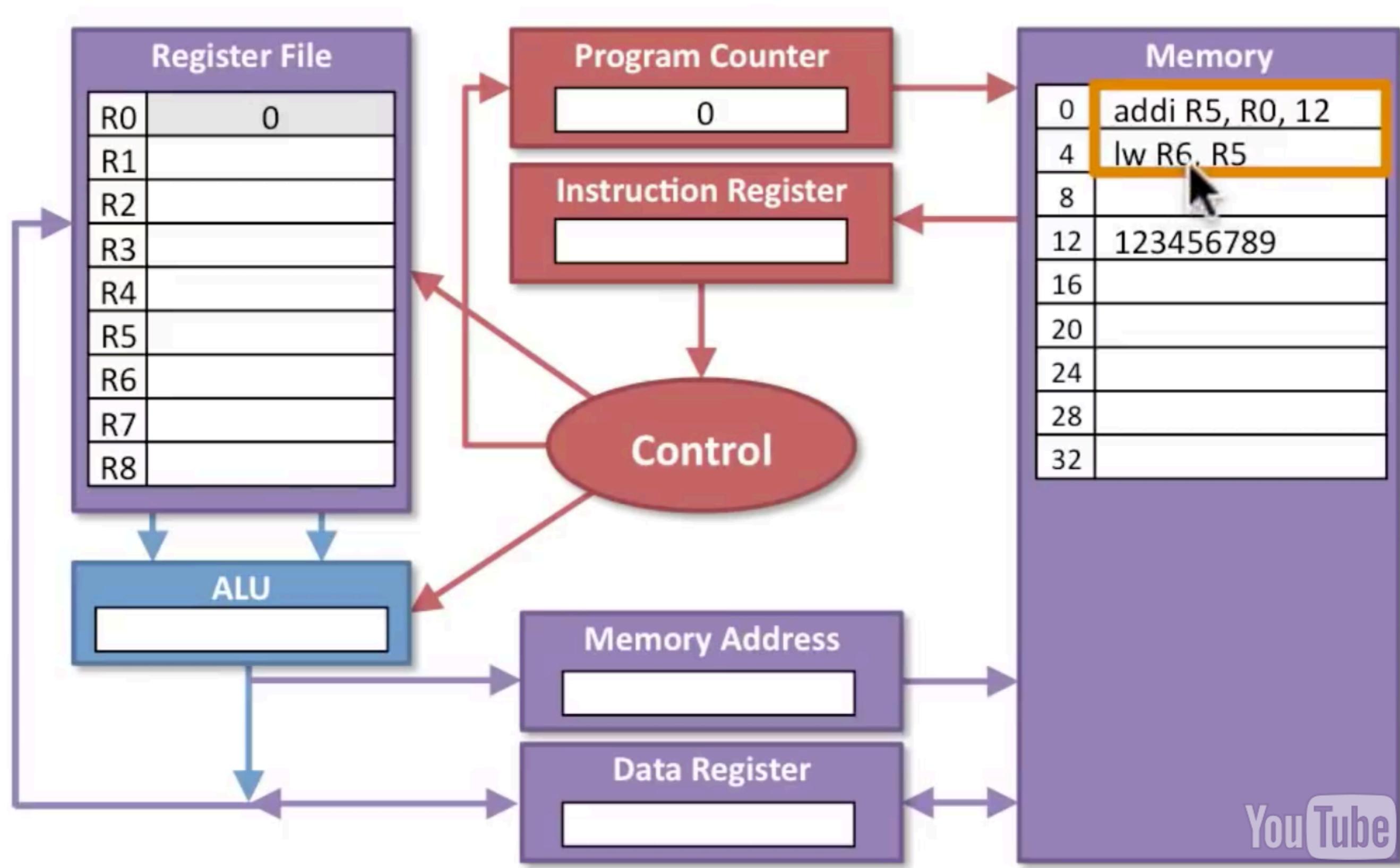


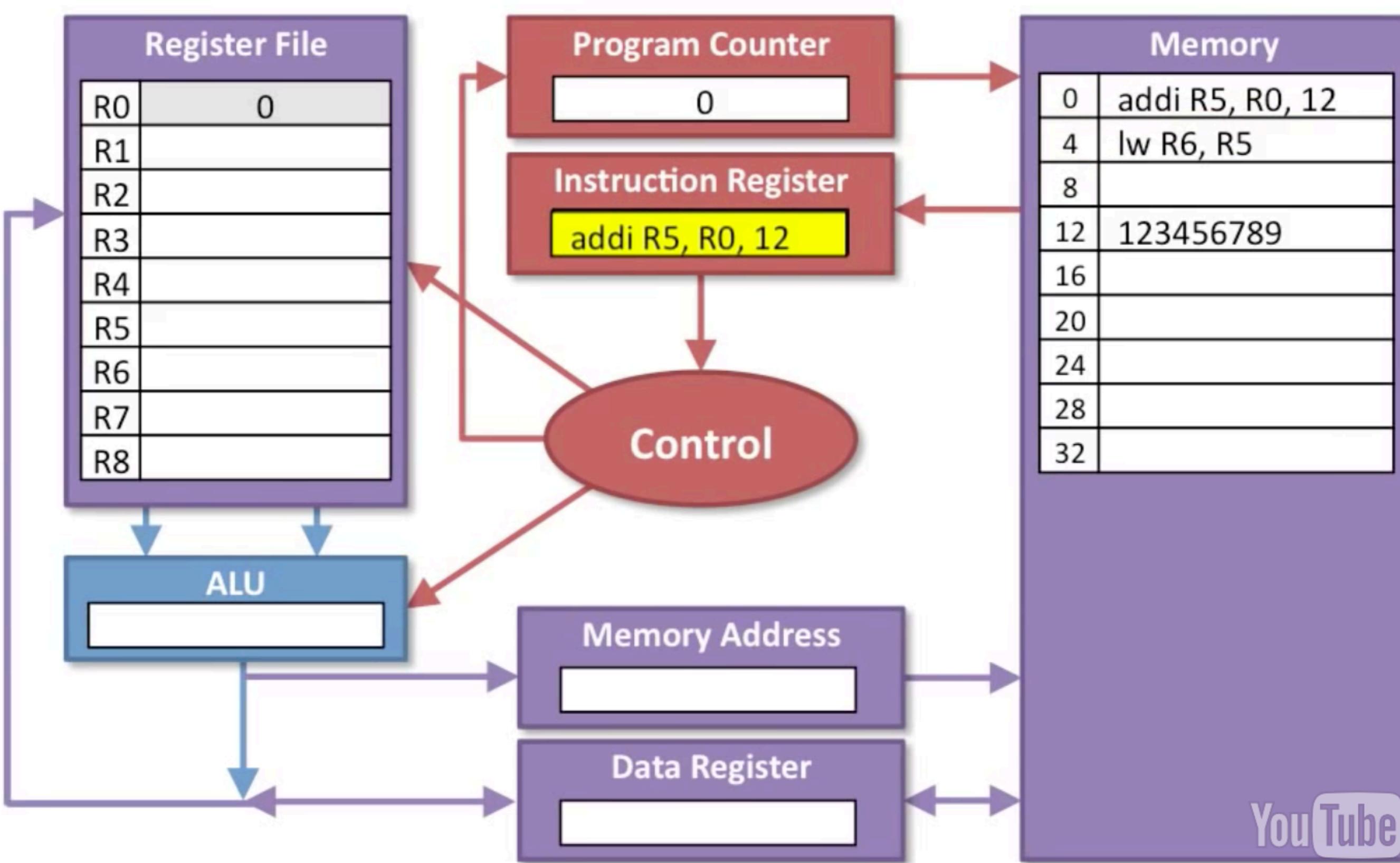


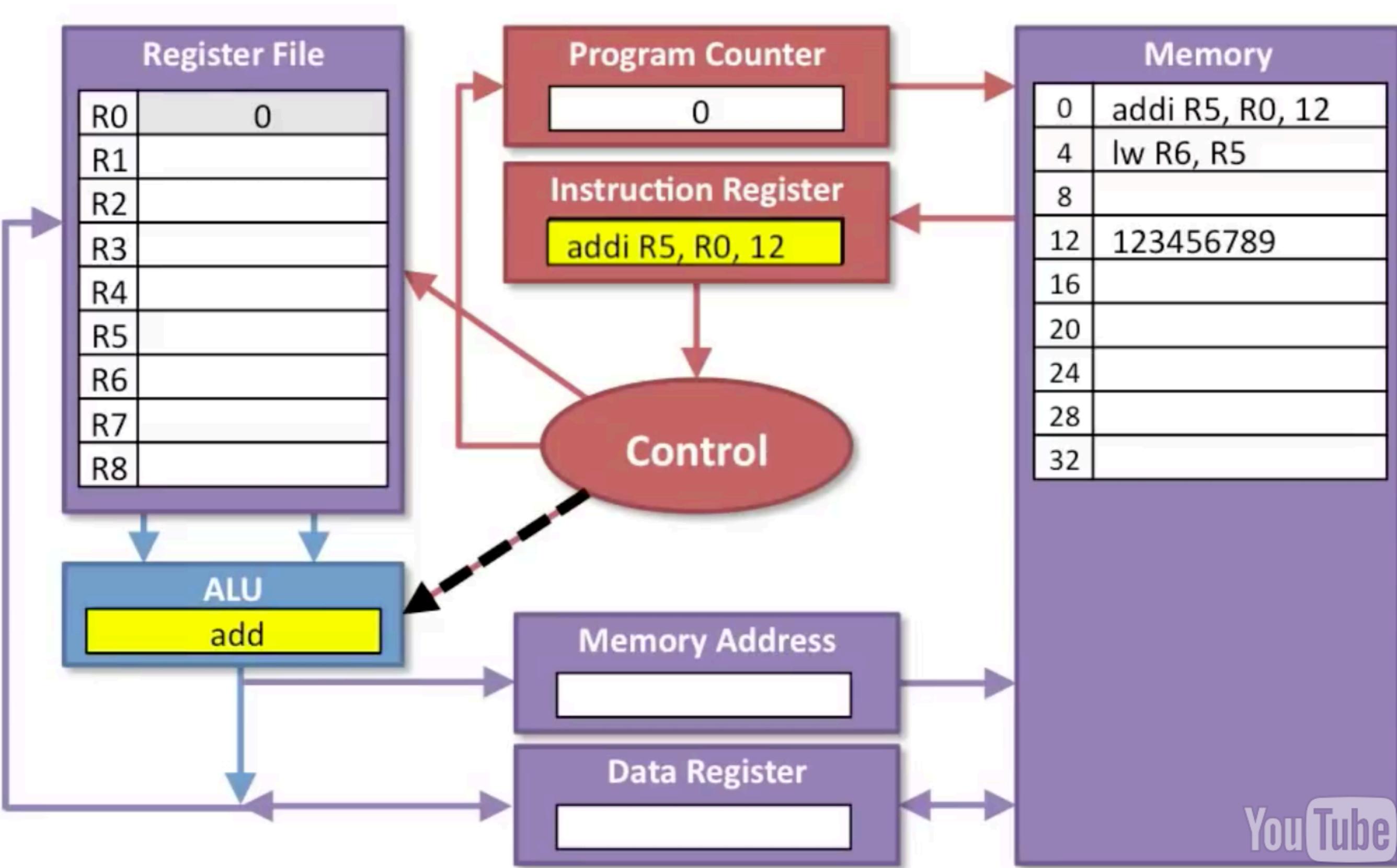
**Suppose that the memory contains data and instructions as shown on the right. What is value of R6 after the 2 instructions are executed?
Assume that the value of the PC is initially 0.**

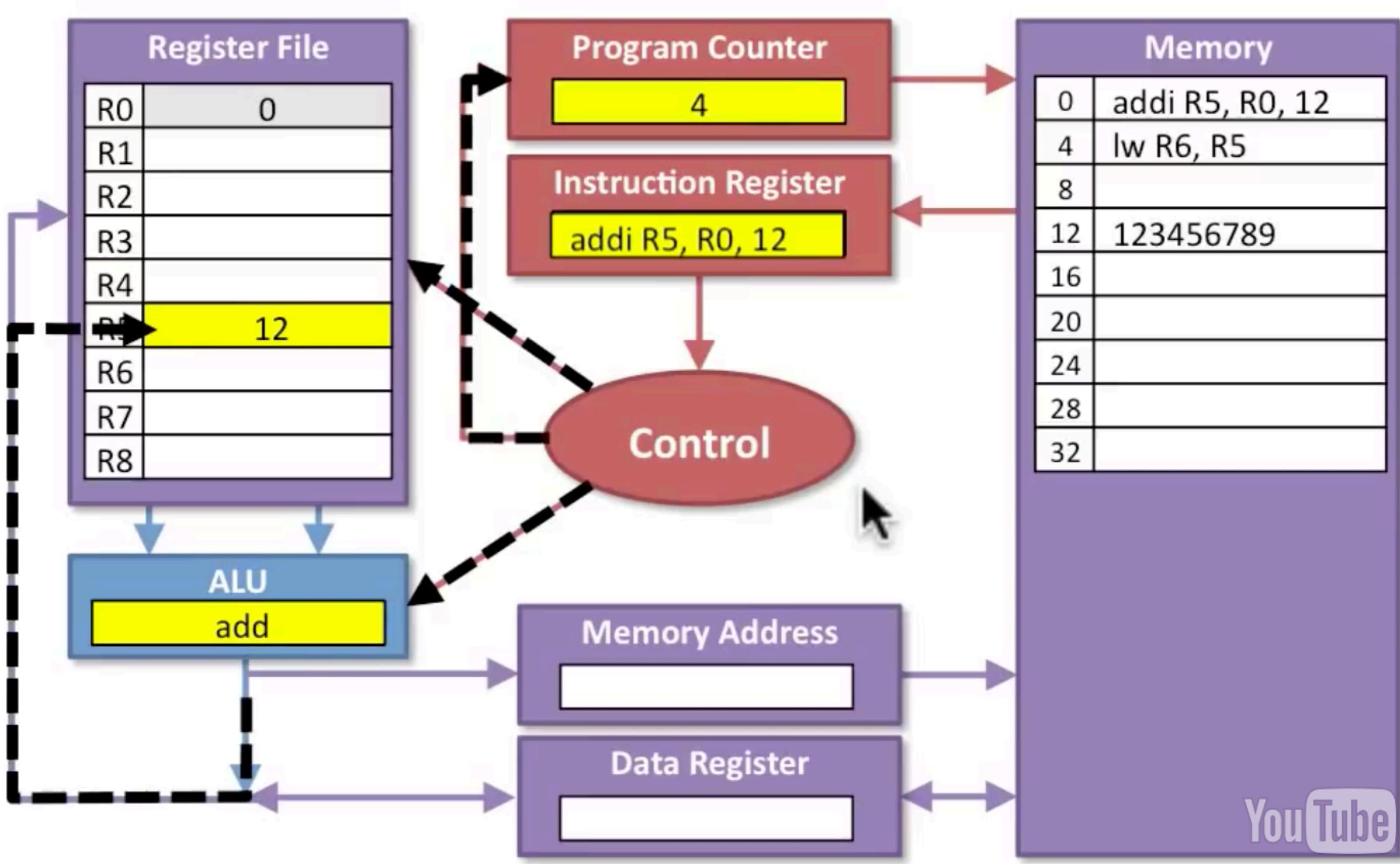
Memory	
0	addi R5, R0, 12
4	Iw R6, 4(R5)
8	
12	123456789
16	999999999
20	
24	
28	
32	

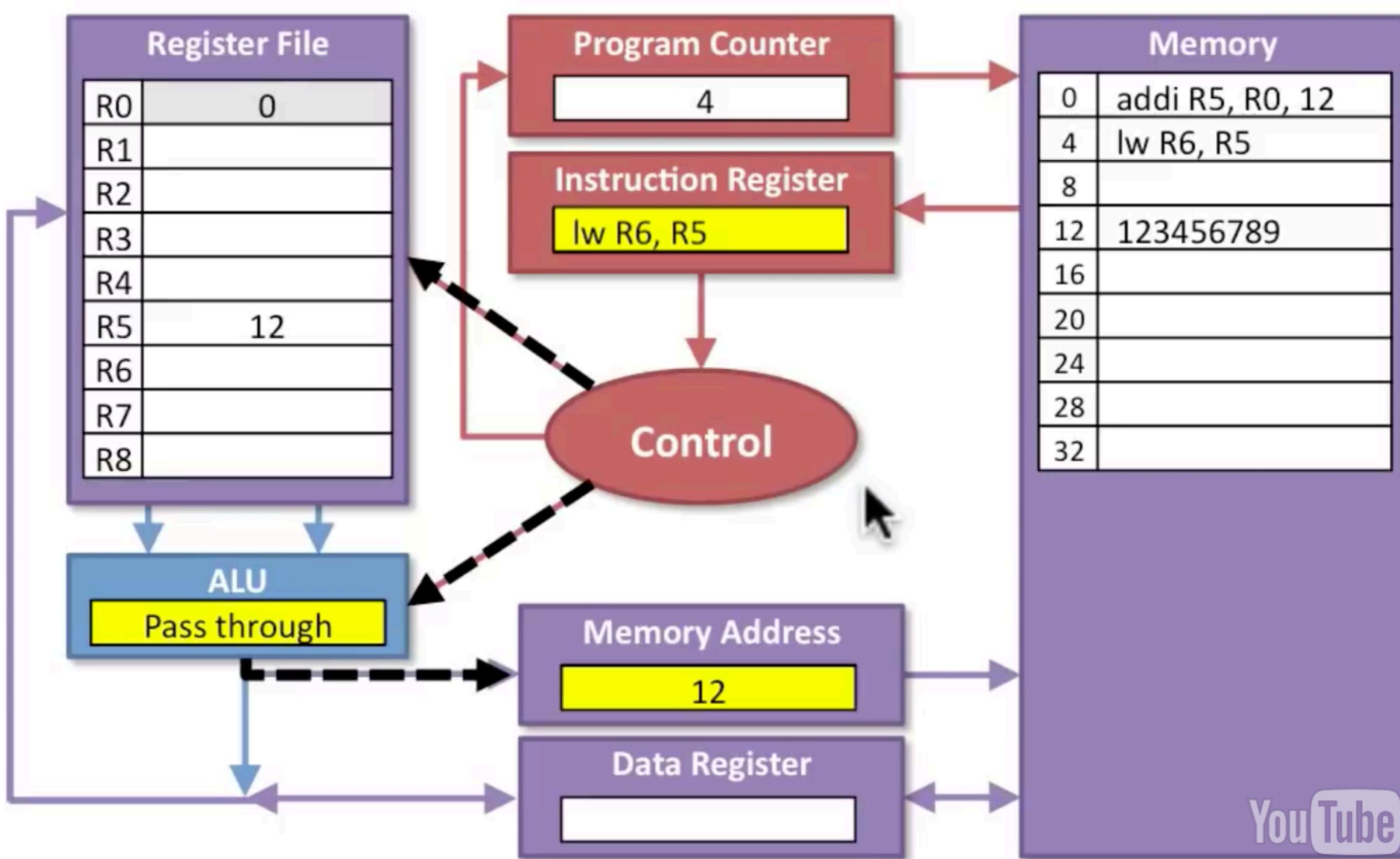


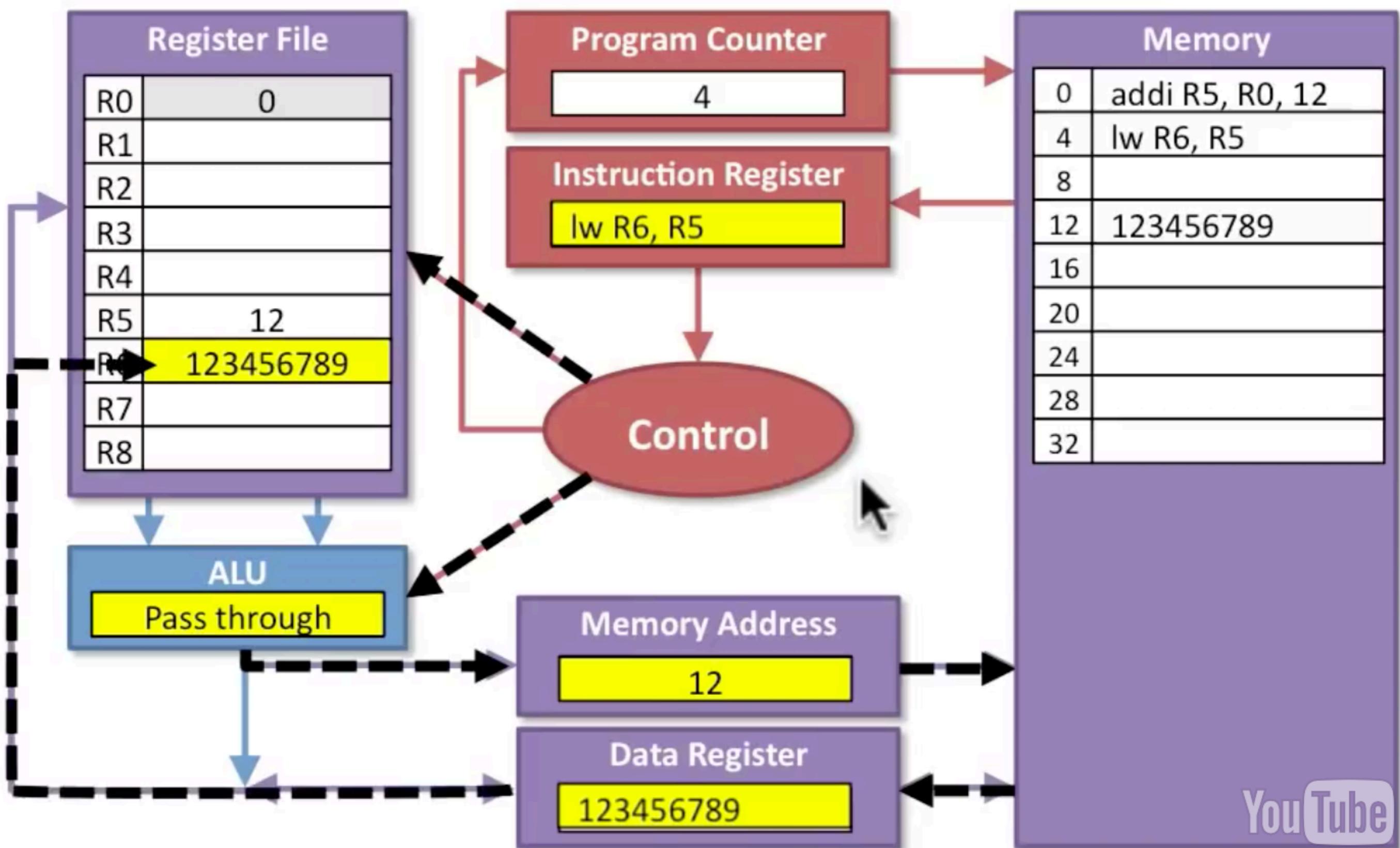





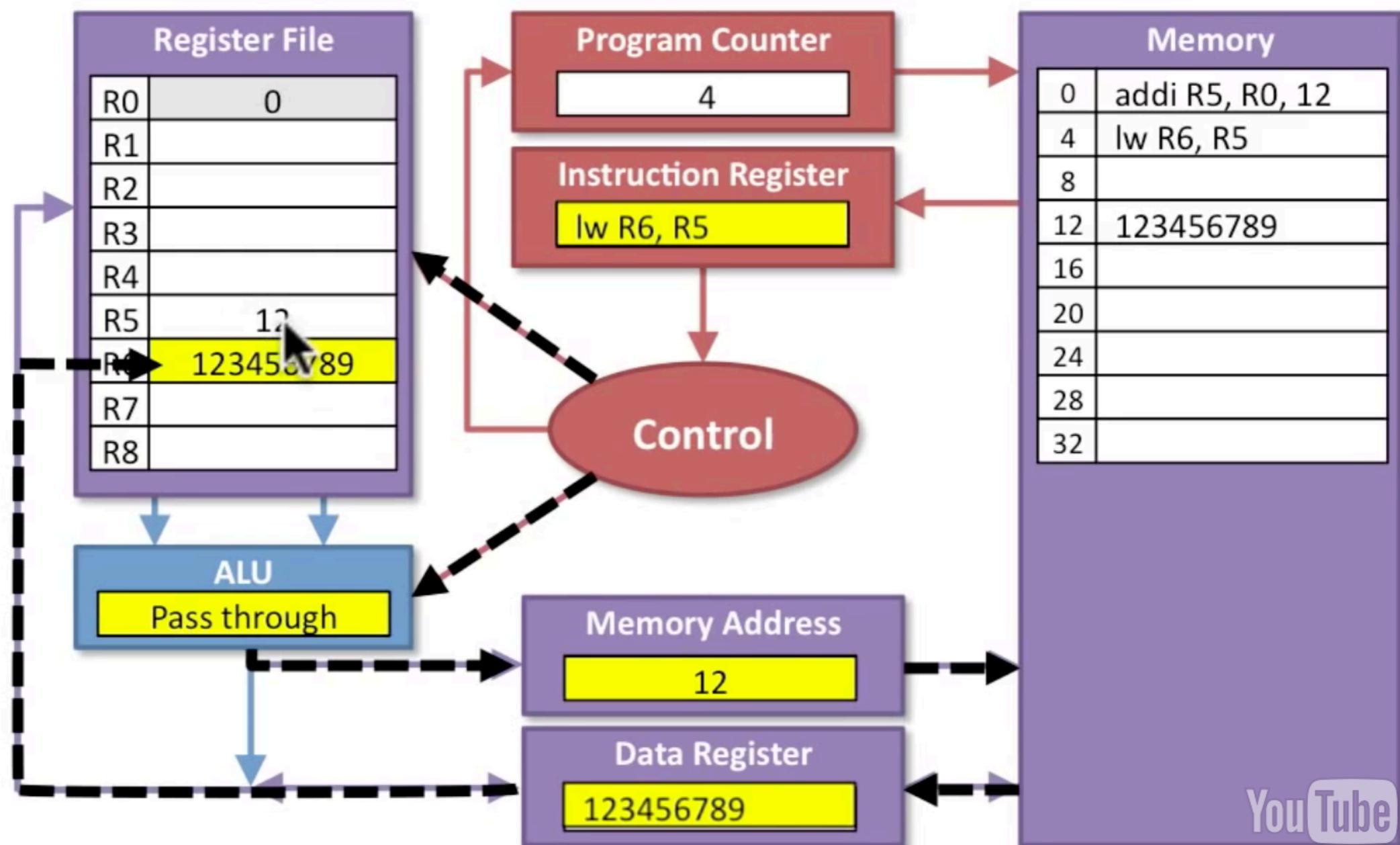






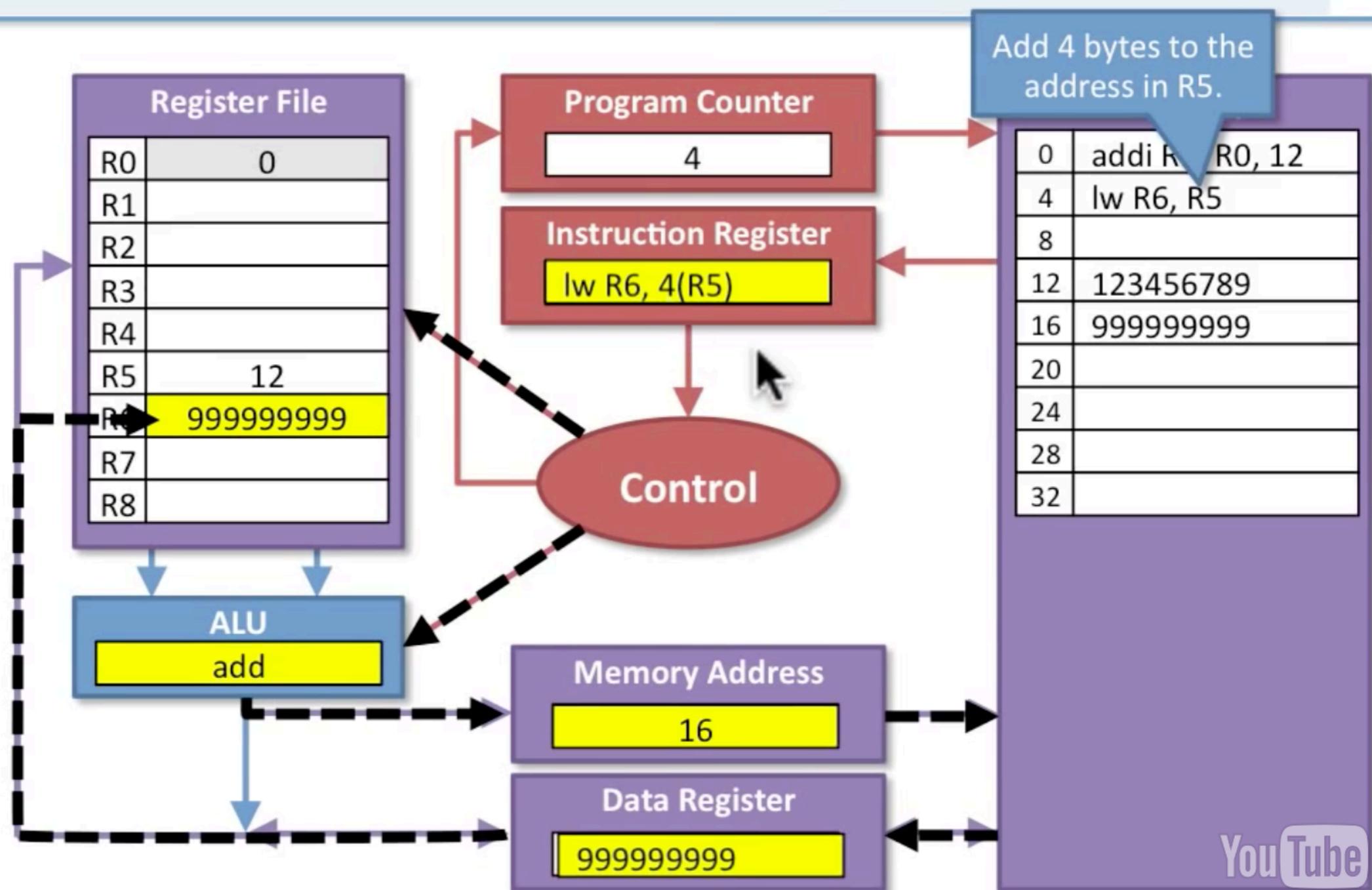


We used the address from the register file to load data from memory.



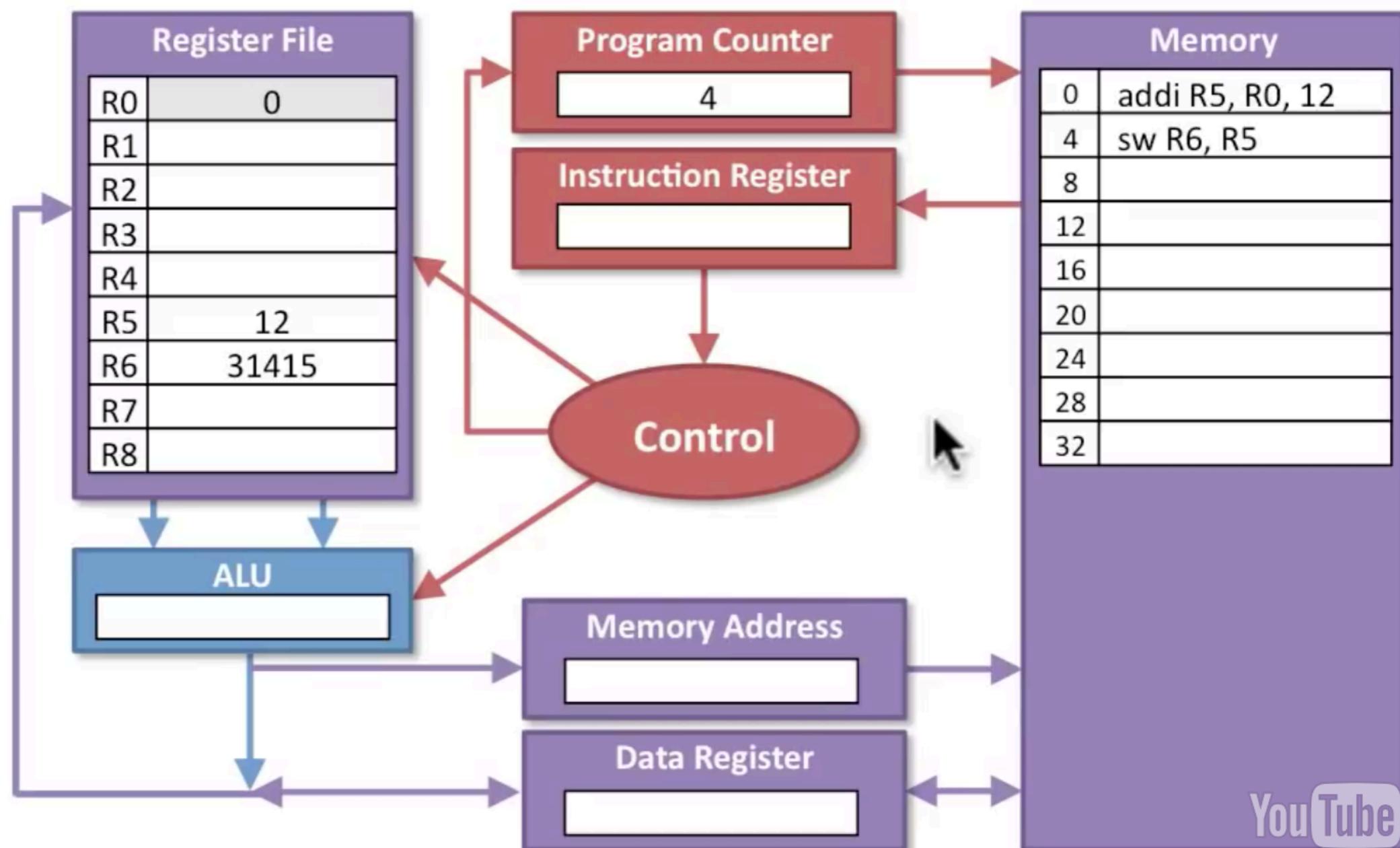
An **offset** can be added to addresses as part of the **lw/sw** instructions.

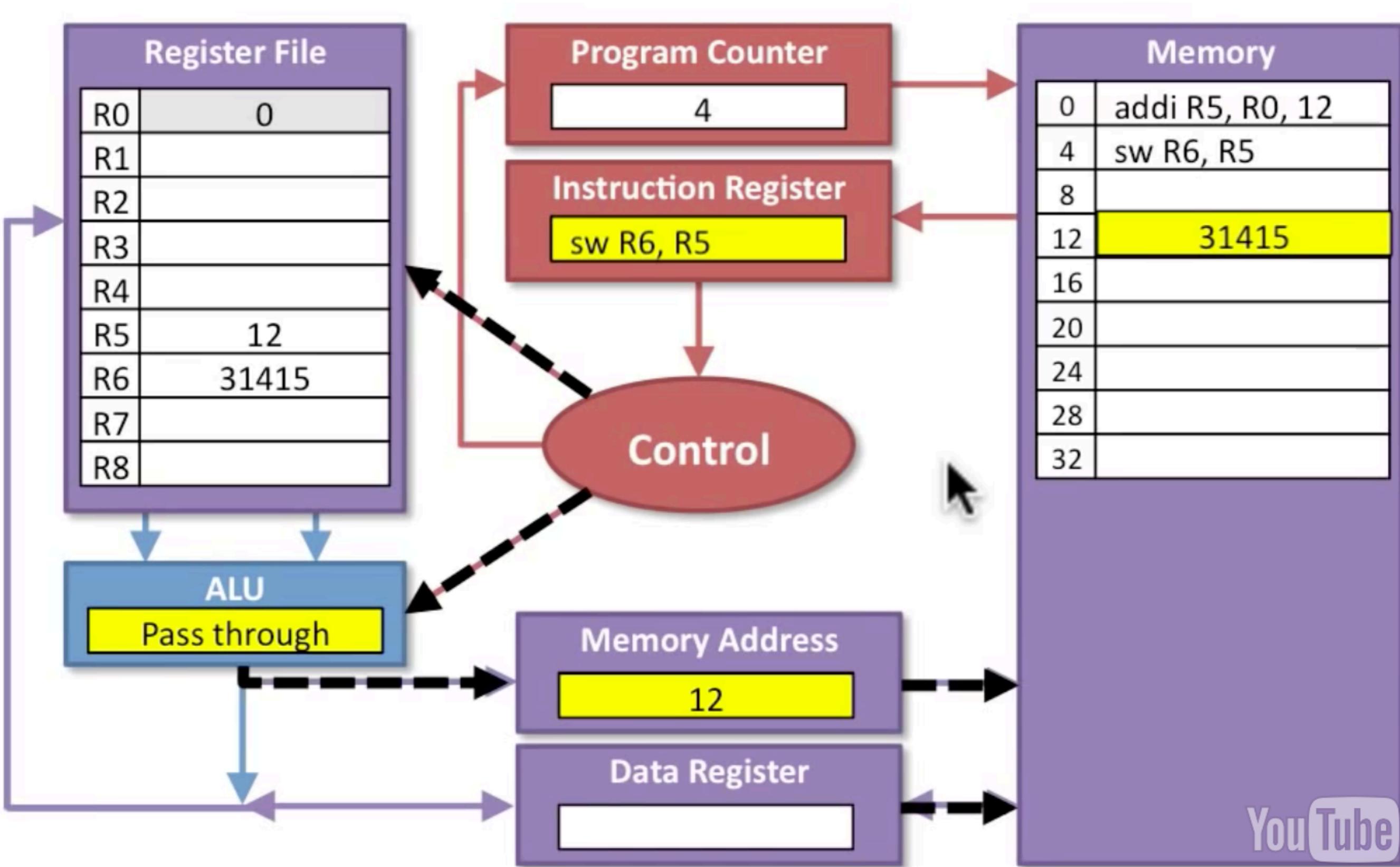
lw R6, 4(R5)



Consider the following contents of the memory, register file and PC. What is the content at memory location 12 after the instructions are executed?

For stores we need:
the address
(from ALU)
the data
(from register)





Using MIPS conditional and unconditional jump instructions, convert the following high-level code to MIPS assembly.

```
if (i == j)  
    h = i + j;
```

Use the following registers for each of the variables:

```
else  
    h = 7;
```

R3 for h

...

R1 for i

R2 for j

```
if (i == j)  
    h = i + j;
```

```
else  
    h = 7;
```

...

bne R1, R2 DoElse
add R3, R1, R2
j skipElse
DoElse:
addi R3, R0, 7
skipElse:

...

```
if (i == j)  
    h = i + j;
```

```
else  
    h = 7;
```

...

be R1, R2 DolF
addi R3, R0, 7
j skipElse
DolF:
add R3, R1, R2
skipElse:

...