

**SCHOOL OF ELECTRICAL AND ELECTRONIC ENGINEERING**

*College of Engineering and Built Environment*

**Bachelor of Engineering in Computers and Communications**

**Programme Code: (DT 021A) 2014/18**

**Year 4**

**Computer Architecture**

**Ray Lynch**

**Student Name(s): Aisling Lee**

**Class Group: DT021A/4**

**Student Number C12358536**

**Assignment title MIPS**

**Date issued 05/10/17**

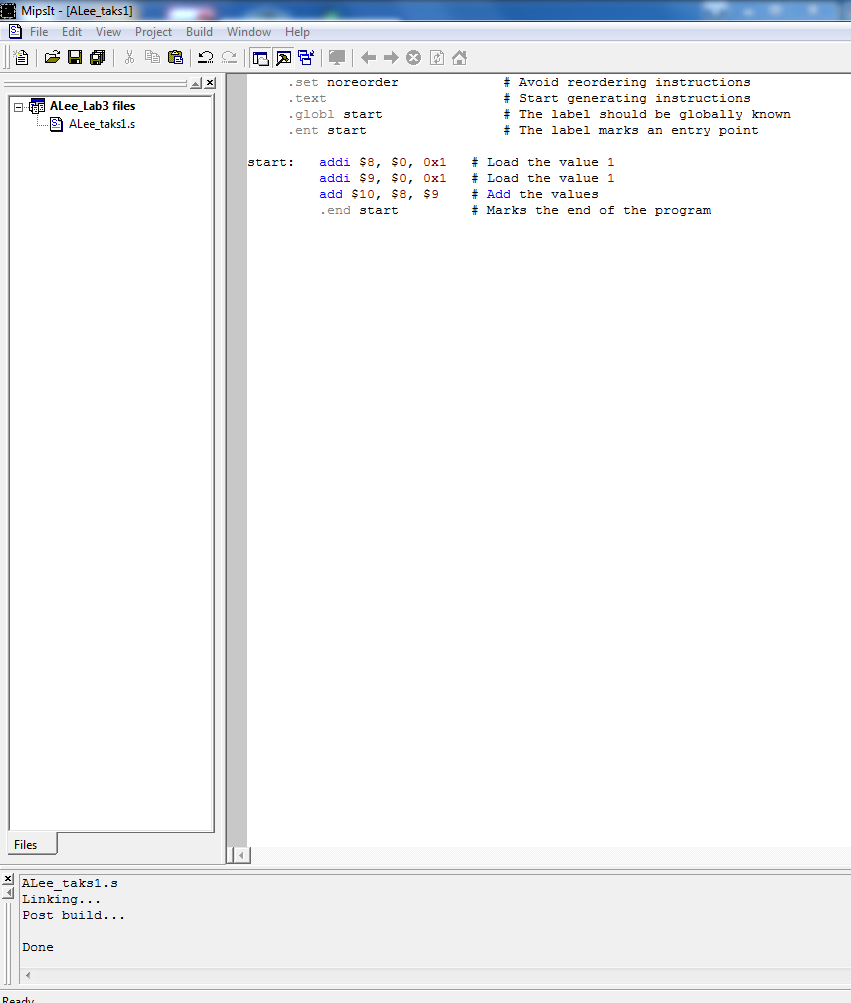
**Date due for return**

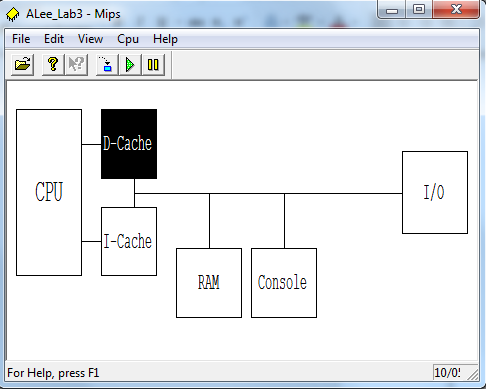
**Date returned**

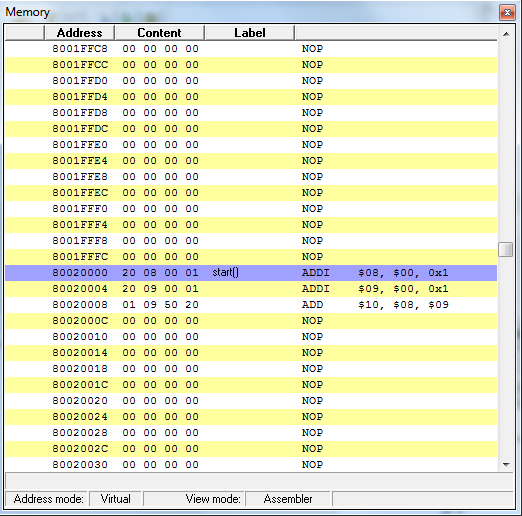
Lab 3

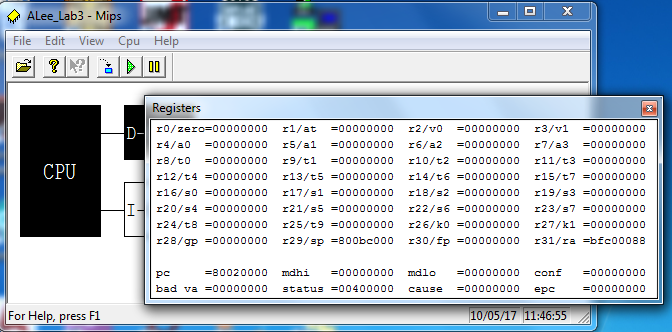
## Introduction:

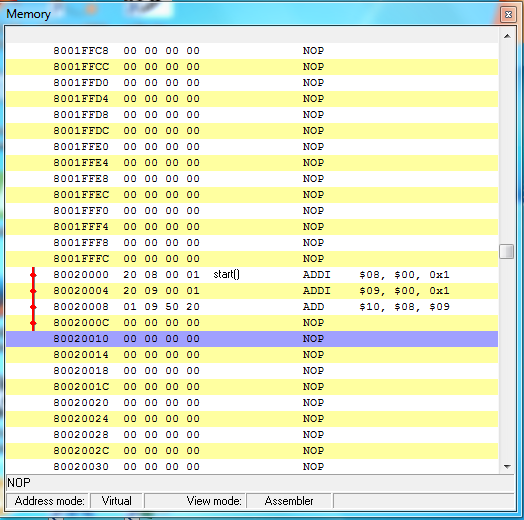
# Results:

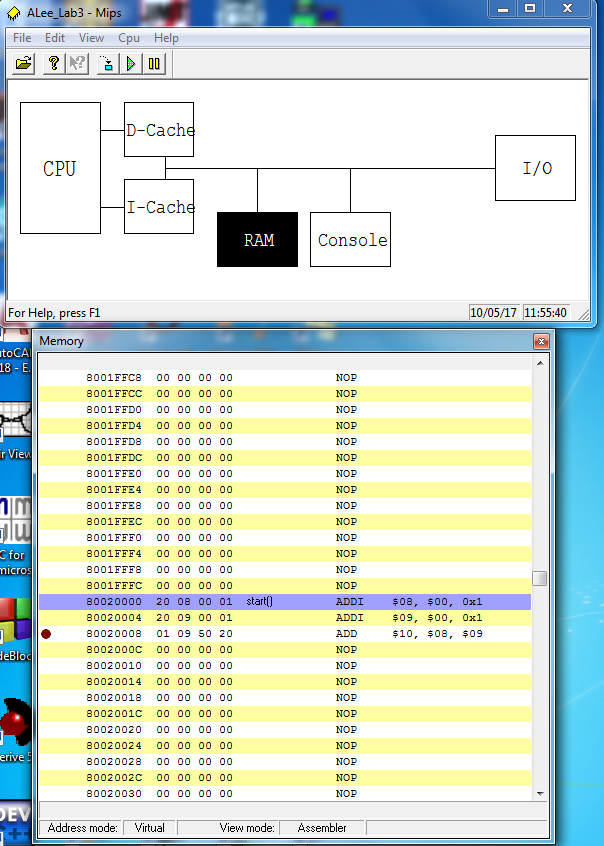


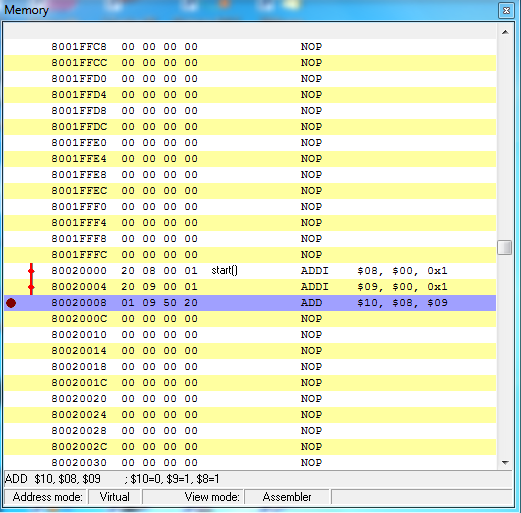


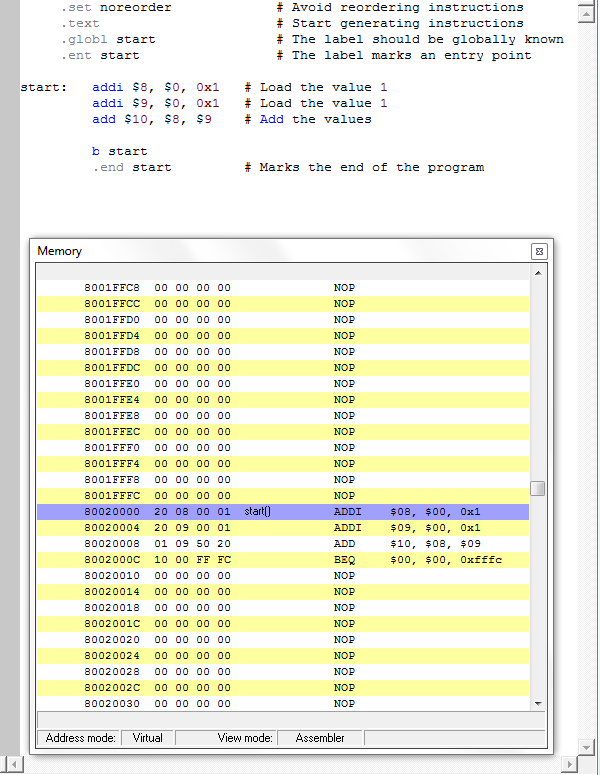


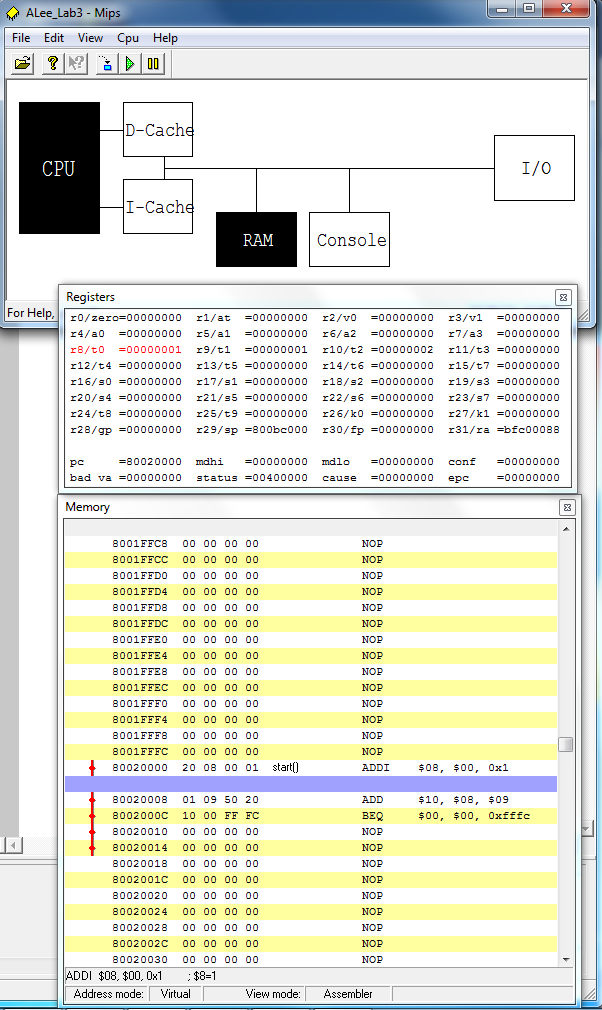












The simulator disassembler can show memory contents as integers, floating point numbers, ASCII text, or as instructions.

What is it that really is stored in the memory? Binary

Write down the memory contents at the address 0x80020000 as

an integer: 537395201

as a floating point number: 1.15196e-019

an ASCII sequence ---

as an instruction. $08, $00, 0x1

http://www.rapidtables.com/convert/number/ascii-hex-bin-dec-converter.htm

How is the conversion of the binary number to each of these formats done?

|  |  |
| --- | --- |
| Sign | Magnitude |

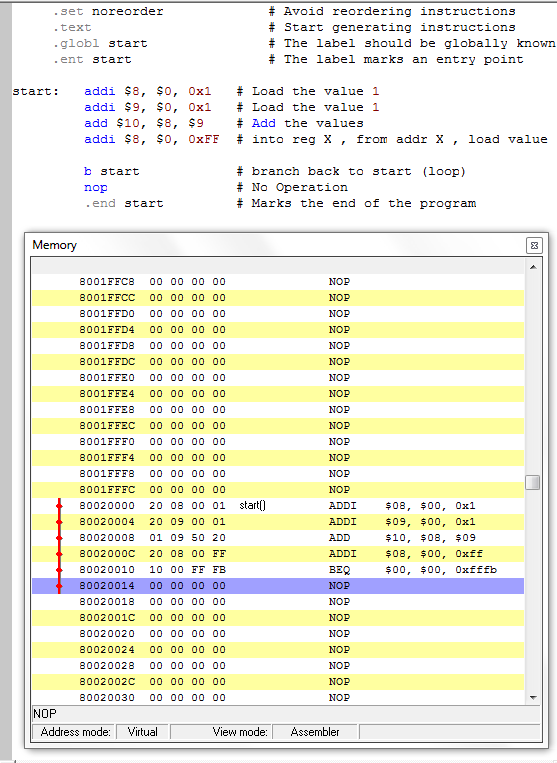
|  |  |  |
| --- | --- | --- |
| Sign | Exponent | Mantissa |

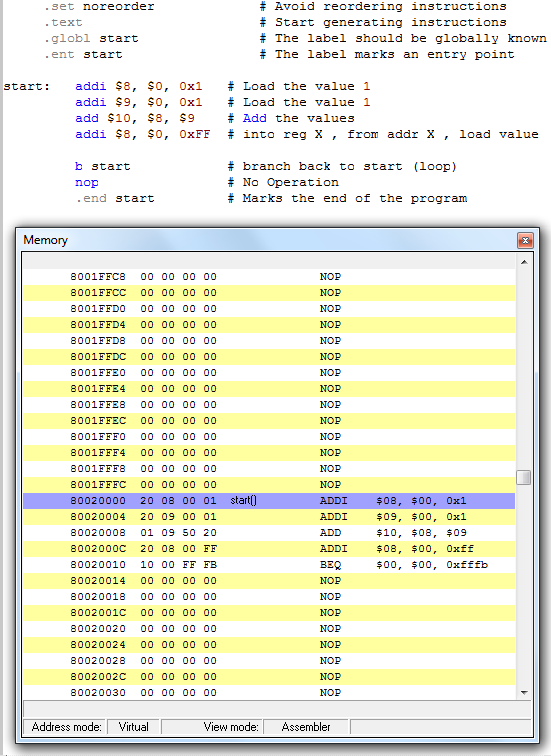
How can the computer know that the stored bit pattern is an instruction?

Task 10

What is the new value of the program counter? Why?

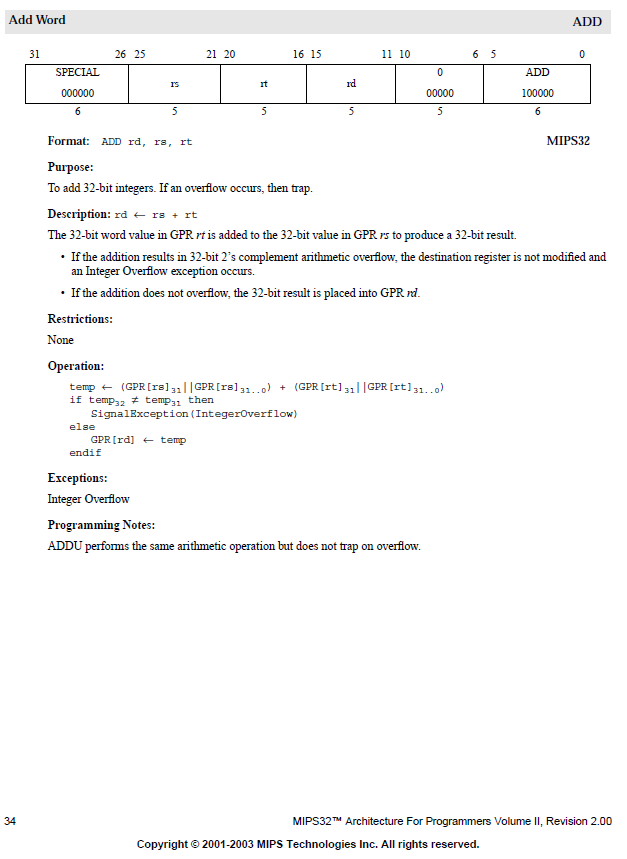
What is the new value of register $8? Why?





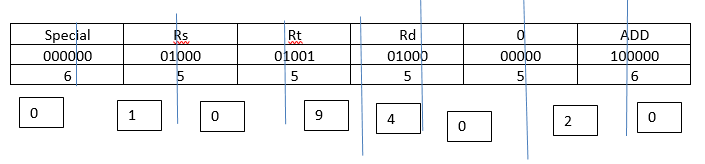
Task 11

**Defn:**

**Immediate Addressing**. An **immediate** operand has a constant value or an expression. When an instruction with two operands uses **immediate addressing**, the first operand may be a **register** or memory location, and the second operand is an **immediate** constant. The first operand defines the length of the data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Special | Rs | Rt | Rd | 0 | ADD |
| Conv to Binary | 8 | 9 | 8 |  |  |
| 000000 | 01000 | 01001 | 01000 | 00000 | 100000 |
| 6 | 5 | 5 | 5 | 5 | 6 |

|  |  |  |  |
| --- | --- | --- | --- |
| add | $8 | $8 | $9 |
| Manipulate | $8 | $9 | $8 |
| ADD | Rd | Rs | rt |



|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Special | | Rs | | Rt | | Rd | | 0 | | ADD | |
|  | 6 | | 5 | | 5 | | 5 | | 5 | | 6 | |
| Binary | 000000 | | 01000 | | 01001 | | 01000 | | 00000 | | 100000 | |
| Convert to Hex  (4 bin = hex) | 0000 | 0001 | | 0000 | | 1001 | 0100 | 0000 | | 0010 | | 0000 |
| 0 | 1 | | 0 | | 9 | 4 | 0 | | 2 | | 0 |

add $8, $8, $9 -> ADD rd, rs, rt => ADD $8, $9, $8 => 01 09 40 20



Set register $8 to 1 and $9 to 2 and execute the instruction.

What does register $8 contain after the execution? Why?

This addressing mode is called *register addressing*