**Práctica 4 – Programación en ensamblador**

Jaime Arana

Manuel Ferrero

3ºA

**Inspección del código máquina:**

TIPO-R

****

Instrucción: **addu rd, rs, rt** (suma)

Dicha instrucción tiene la siguiente distribución de bits:

**000000 + rs(5 bits) + rt (5 bits) + rd(5 bits) + 00000 + 100001**

Por lo tanto, si se sustituye por sus valores en decimal de los registros:

**000000 + 29 + 0 + 30 + 00000 + 100001**

Si pasamos estos a código máquina, que es la instrucción en hexadecimal, obtenemos:

**03A0F021**

TIPO-I



Instrucción: **rd, desp16(rs)** (store word)

Dicha instrucción tiene la siguiente distribución de bits:

**101011 + rs(5 bits) + rd(5 bits) + desplazamiento16**

Por lo tanto, si se sustituye por sus valores en decimal de los registros:

**101011 + 29 + 30 + 12**

Si pasamos estos a código máquina, que es la instrucción en hexadecimal, obtenemos:

**AFBE000C**

TIPO-J



Instrucción: **j Dir28** (Salto incondicional)

Dicha instrucción tiene la siguiente distribución de bits:

**000010 + constante << 2**

Por lo tanto, si se sustituye por sus valores en decimal de los registros:

**000010 + 0xD0001D4 << 2**

Si pasamos estos a código máquina, que es la instrucción en hexadecimal, obtenemos:

**0B400075**

**Ensamblador en línea:**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Función para generar retardos:**

Para implantar la función que genera los retardos, creamos un retardo de 1ms e iteramos hasta llegar al valor del retardo solicitado.

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente