

OBJETIVOS

Los objetivos que el estudiante debe alcanzar al final de esta práctica se resumen en los siguientes puntos:

1. Conocer los bloques funcionales del computador básico: Memoria de Instrucciones, Unidad Central de Proceso y Memoria de Datos.
2. Conocer el conjunto de instrucciones que podemos utilizar para programar el computador básico.
3. Estudiar un programa sencillo y conocer la evolución del mismo ejecutando el programa entero, por instrucciones y por fases de instrucción.
4. Ser capaces de implementar un programa muy sencillo de suma de enteros.

MATERIAL

Para la realización de la práctica se dispondrá de un ordenador PC, con los sistemas operativos LINUX y/o Windows, y el simulador del Computador Básico. La práctica se debe realizar arrancando el ordenador en el sistema operativo LINUX con el objetivo de que el alumno se familiarice con su uso.

Para ello al arrancar el ordenador cuando aparezca la palabra LILO teclear linux y pulsar ENTER.

INICIO DE LA PRÁCTICA

- A) Arrancar la máquina en modo Linux.
- B) Abrir una sesión en Linux utilizando el login "*usuario*" y el password "*practicas*".
- C) Hacer click sobre el icono "Procesador Básico" del Escritorio para ejecutar el simulador.

I. BLOQUES FUNCIONALES DEL COMPUTADOR BÁSICO

Como podemos ver en la figura 1, el computador básico se divide en 3 bloques funcionales:

1. *Memoria de Instrucciones*: Donde se almacenan las instrucciones del programa que vamos a ejecutar.
2. *Unidad Central de Proceso*: En ella tenemos el "*Contador de Programa*", que nos indica en todo momento cuál es la siguiente instrucción a ejecutar. Así mismo, tenemos el "*Registro de Instrucciones*" que almacena la instrucción que estamos decodificando, así como la "*Unidad Aritmético-Lógica*", responsable de efectuar las operaciones aritmético lógicas que se requieran en las instrucciones.

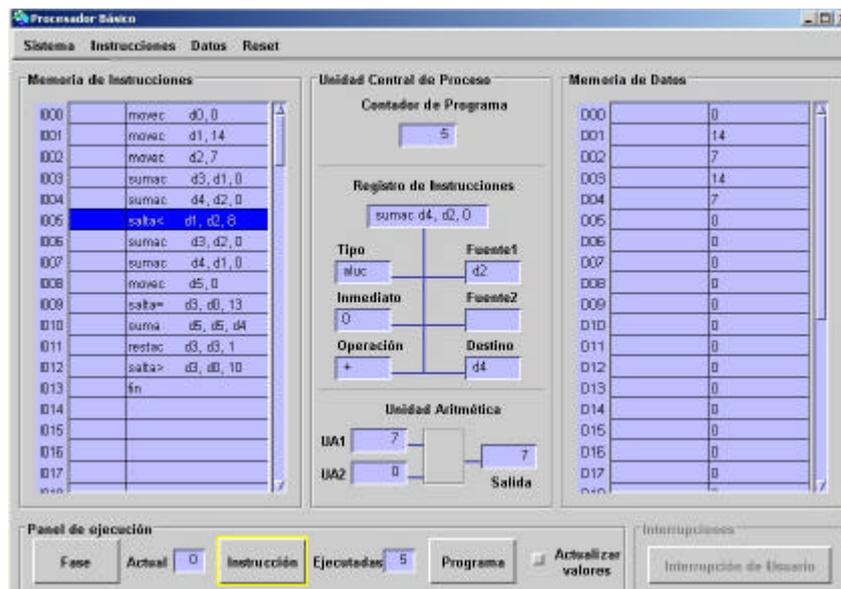


Figura 1

3. *Memoria de Datos*: Determina el área de memoria del computador de la cual la unidad central de proceso toma los datos para operar y donde deja los resultados.

Si nos fijamos en el panel inferior de la figura 1, además de "Memoria de Instrucciones", "Unidad Central de Proceso" y "Memoria de Datos" el programa presenta un panel de ejecución con 3 opciones: **Fase**, **Instrucción** y **Programa**. La opción "Programa" permite ejecutar el código cargado en la Memoria de Instrucciones de una vez. Por otro lado, la opción "Instrucción" permite hacerlo instrucción por instrucción, mientras que "Fase" permite visualizar las 4 fases que conforman la ejecución de una instrucción (adquisición de la instrucción e incremento del PC, lectura de los operandos, ejecución de la instrucción, escritura de resultados).

II. CONJUNTO DE INSTRUCCIONES

Las instrucciones que soporta el computador básico son las siguientes:

Instrucciones de transformación

suma d0, d1, d2	(d0 = d1 + d2)
resta d0, d1, d2	(d0 = d1 - d2)
sumac d0, d1, num	(d0 = d1 + num)
restac d0, d1, num	(d0 = d1 - num)
multiplica d0, d1, d2	(d0 = d1 * d2)
divide d0, d1, d2	(d0 = d1 / d2)
modulo d0, d1, d2	(d0 = d1 mod d2)

donde la operación modulo calcula el resto de la división entera de dos números

Instrucciones de transferencia

movec d0, num	(d0 = num)
---------------	------------

Instrucciones de control de flujo

salta 5	(sig. instr. = i5)
salta= d0, d1, 5	(si d0 == d1 → sig. instr. = i5)
salta> d0, d1, 5	(si d0 > d1 → sig. instr. = i5)
salta< d0, d1, 5	(si d0 < d1 → sig. instr. = i5)

Instrucciones especiales

fin	(Detiene la Ejecución)
-----	------------------------

III. ANÁLISIS DE UN PROGRAMA SENCILLO

En este apartado vamos a cargar un programa ya desarrollado en el código de instrucciones del Computador Básico y lo ejecutaremos utilizando las diferentes opciones disponibles. Para ello debes crear con un editor de texto ([ver apartado IV](#)) el siguiente listado:

“primero.asm”:

```

movec d0, 0
movec d1, 14
movec d2, 7
sumac d3, d1, 0
sumac d4, d2, 0
salta< d1, d2, 8
sumac d3, d2, 0
sumac d4, d1, 0
movec d5, 0
salta= d3, d0, 13
suma d5, d5, d4
restac d3, d3, 1
salta> d3, d0, 10
fin

```

Después seleccionar en el menú **Sistema** la opción **Básico**:

- Seleccionar la opción **Cargar** del menú **Instrucciones**. Aparecerá una ventana indicando primeramente el filtro a aplicar para la búsqueda del archivo (por defecto "*.asm"). Pulsamos "Aceptar" y seguidamente aparecerán todos los ficheros del directorio cuya extensión sea ".asm". Seleccionamos el fichero "*primero.asm*". El programa se cargará en la memoria de instrucciones, tal cual aparece en la figura 1.
- Selecciona el botón "*Programa*", el cual permite ejecutar el programa "primero.asm" en su totalidad. Podrás apreciar cómo el procesador va ejecutando las instrucciones secuencialmente, a la vez que va tomando datos de la Memoria de Datos y dejando los resultados en la misma.

Pregunta: Viendo el resultado que genera el programa sobre la memoria de datos y suponiendo que las direcciones de memoria "D01" y "D02" guardan las entradas del programa, ¿En qué dirección de memoria está el resultado? ¿Podrías anticipar qué hace exactamente el programa?

- C) Restauramos el programa y la memoria de datos a su punto de partida seleccionando consecutivamente las opciones **Reset** del Menú **Datos** y **Reset** del Menú **Procesador**.
- D) Ahora procederemos a ejecutar de nuevo el programa, esta vez usando para ello el botón "Instrucción", que nos permite parar tras la ejecución de cada instrucción. Observa cómo se van actualizando los resultados parciales del programa en la memoria de datos, así como lo que ocurre cuando se ejecuta una instrucción de salto.

Pregunta: ¿Cuántas veces pasa el programa sobre la instrucción de salto "I12"? ¿Qué operación se efectúa tras la comprobación del salto "I12"? ¿Podrías decir ahora qué algoritmo implementa el programa?

- E) Restauramos de nuevo el procesador y la memoria de datos.
- F) Vamos ejecutando el programa mediante el botón "Fase", que nos permite subdividir la ejecución de una instrucción en sus 4 fases principales: adquisición de la instrucción e incremento del PC, lectura de los operandos, ejecución de la instrucción, escritura de resultados.
- G) Parar cuando en el registro de instrucción aparezca la instrucción: "**sumac d3, d1, 0**".

Pregunta: ¿Dónde apunta en este momento el contador de programa? ¿Por qué?

- H) Sigue ejecutando las 4 fases de la instrucción "sumac d3, d1, 0".

Pregunta: ¿Qué entradas aparecen en la ALU? ¿Cuál es el resultado? ¿Qué se almacena en memoria de datos?

IV. IMPLEMENTACIÓN DEL PRIMER PROGRAMA

En este apartado aprenderemos a crear un fichero de texto con un editor y abrir este fichero en el simulador. Así mismo, daremos un repaso a las diferentes alternativas que tenéis en cuanto al almacenamiento de este fichero en algún lugar para vuestro uso particular después de las prácticas (disquete, cuenta en anubis.uji.es, etc).

- A) Utilizando un editor de textos (p.e., **Text Editor** del botón "KDE", carpeta "Aplicaciones") crea un fichero con el siguiente contenido:

```
movc d0, 1
movc d1, 2
suma d2, d0, d1
fin
```

- B) Guardamos el fichero con nombre "suma.asm", utilizando la opción "Save As" del menú "File".
- C) Volvemos al programa simulador y limpiamos la memoria de datos y el procesador.
- D) Cargamos el fichero "suma.asm" en el simulador. Lo ejecutamos y comprobamos que el resultado generado es la suma de 1 y 2.
- E) **Ahora vamos a guardar el fichero "suma.asm" en el disquete.** Para ello hacemos click en el icono de disquetera que se encuentra en el escritorio (Floppy). Aparecerá un explorador con el contenido del disquete.
- F) Hacemos click en el icono que da acceso al explorador de nuestros ficheros en el disco. El icono se encuentra en la barra inferior de aplicaciones (Icono con una casa y un archivador). Nos aparecerán los ficheros y directorios de nuestro disco, y por lo tanto visualizaremos el fichero "suma.asm".
- G) Hacemos click sobre el fichero "suma.asm" con el botón derecho del ratón y seleccionamos la opción "Copy".
- H) Vamos ahora a la ventana del explorador en el disquete y seleccionamos la opción "Paste". El resultado será la copia en el disquete del fichero que previamente teníamos guardado en el disco duro.
- I) Finalmente sólo nos queda decirle al ordenador que queremos sacar el disquete. Para ello haremos click con el botón derecho del ratón sobre el icono "Floppy" del escritorio, y seleccionaremos la opción "UnMount". En el momento en que la luz verde del icono "Floppy" se apague ya podremos sacar el disquete.
- J) La otra opción que tenemos para guardar nuestros datos es **copiar este fichero a nuestra cuenta en "anubis"**. El primer paso será abrir un explorador de ficheros del disco duro como en el punto "F".
- K) Ahora le diremos que cargue los ficheros remotos de nuestra cuenta en "anubis". Para ello escribiremos en la barra de dirección del explorador el texto **ftp://usuario@anubis.uji.es**, siendo

"usuario" vuestro login en esa máquina (mismo que con el correo electrónico). Tras introducir el password aparecerán todos los ficheros en vuestra cuenta y podréis copiar y pegar ficheros entre "anubis" y vuestro disco duro, siguiendo el procedimiento "Copy-Paste" descrito en el caso anterior.

Hasta aquí nos hemos familiarizado con el entorno (Linux), hemos sido capaces de crear un programa en código ensamblador del computador básico y ejecutarlo, así como llevarnos nuestro trabajo en un disquete y/o a nuestra cuenta en "anubis". Ahora nos podemos centrar en la programación.

V. PROBLEMA PROPUESTO

Implementa en un fichero llamado "divide.asm" la división de dos números enteros utilizando restas. Cárgalo en el simulador y verifica su funcionamiento.

VI. ASIGNACIÓN DE ETIQUETAS EN LOS REGISTROS DE INSTRUCCIONES Y DATOS

Una vez cargado un programa en la memoria de instrucciones se pueden realizar modificaciones del mismo con el simple uso del editor de instrucciones. Para ello basta con:

- A) Pulsar el menú **Instrucciones** y escoger Editar.
- B) Aparece una nueva ventana con la lista de instrucciones. Ahora se pueden realizar modificaciones.
- C) Al acabar pulsar Aceptar y a continuación se puede salvar las modificaciones.
- D) Escoger en el menú de Instrucciones Guardar e indicar el nombre del fichero. Pulsar Aceptar.

También con el editor de datos se pueden asignar etiquetas a los registros de datos de forma que podemos utilizar en nuestro programa las etiquetas asignadas en lugar de los nombres de los registros de datos.

Para realizar esto basta con

- A) Acceder al menú Datos y escoger Editar.
- B) Aparece una nueva ventana en la cual podemos escribir en la columna de etiquetas el nombre que queremos asignar a un determinado registro de datos. Por ejemplo poner en d0 la etiqueta dividendo.
- C) Al acabar pulsar Aceptar.



Una vez asignadas las etiquetas podemos modificar el programa cargado en la memoria de instrucciones, tal y como se indicó antes, sustituyendo los nombres de los registros por los nombres de las etiquetas. Después se puede ejecutar el programa.

Comprueba el uso de asignación de etiquetas en el registro de instrucciones y datos sustituyendo en el programa "divide.asm" los registros de datos usados (por ejemplo d0, d1, d2) usados para el dividendo, divisor y cociente por una etiqueta. Después sustituye dichos registros de datos en el programa por las etiquetas y comprueba el funcionamiento.

VII. EJERCICIOS

EJERCICIO 1.- El siguiente programa eleva un número a una potencia, encontrándose el número almacenado en d0 y el exponente en d1. El resultado se proporciona en d3.

```
PROGRAMA:      i0      movec    d0, 5
                i1      movec    d1, 3
                i2      movec    d3, 1
                i3      movec    d4, 0
                i4      sumac    d2, d1, 0
                i5      multiplica d3, d3, d0
                i6      restac   d2, d2, 1
                i7      salta>   d2, d4, 5
                i8      fin
```

1. ¿Cuál es el bucle que se repite en el programa?
2. ¿Qué variable se va decrementando para controlar el bucle?
3. ¿Cuántas veces se ejecuta la instrucción i7?

EJERCICIO 2.- El siguiente programa calcula el máximo común divisor de dos números que se encuentran en d0 y d1.

```
PROGRAMA:      i0      movec    d0, 15
                i1      movec    d1, 6
                i2      movec    d3, 0
                i3      sumac    d2, d0, 0
                i4      salta>   d1, d0, 6
                i5      sumac    d2, d1, 0
                i6      modulo   d4, d0, d2
                i7      modulo   d5, d1, d2
                i8      restac   d2, d2, 1
                i9      salta=   d4, d3, 11
                i10     salta    6
                i11     salta=   d5, d3, 13
                i12     salta    6
                i13     sumac    d6, d2, 1
                i14     fin
```

1. ¿Qué se realiza entre las instrucciones i3-i5?
2. ¿Qué condición suponen las instrucciones i9-i12?
3. ¿Qué algoritmo utiliza el programa para calcular el máximo común divisor?
4. ¿Dónde se almacena el resultado?

VIII. PROGRAMAS PROPUESTOS

1.- Diseña un algoritmo y desarrolla un programa que determine si un número es par. Utiliza para ello las siguientes direcciones de memoria:

- d0 para almacenar el número.
- d1 para almacenar el resultado, d1 será cero si d0 es par y uno si es impar.

2.- Diseña un algoritmo y desarrolla un programa que calcule el factorial de un número. Utiliza para ello las siguientes direcciones de memoria:

- d0 para almacenar el número del que se quiere obtener el factorial.
- d1 para almacenar el resultado.

3.- Diseña un algoritmo y desarrolla un programa que obtenga el mayor de cuatro números. Utiliza las cuatro primeras posiciones de memoria (d0-d3) para almacenar los cuatro números a comparar y d4 para el resultado.

4.- Diseña un algoritmo y desarrolla un programa que indique si un número es primo. Para ello, almacena el número a comprobar en la dirección de memoria d0 y guarda el resultado en d1 (el resultado será 0 si el número es primo y 1 si no lo es).