Práctica 2: Computador Básico, Introducción

Objetivos:
1) Introducir el funcionamiento del Computador Básico: Memoria de Instrucciones, Unidad Central de Proceso y Memoria de Datos.
2) Introducir el Conjunto de Instrucciones que podemos utilizar para programar el Computador Básico.
3) Cargar un programa sencillo y ver la evolución del mismo ejecutando el programa entero, por instrucciones, y por fases de instrucción.
4) Implementar un programa muy sencillo de suma de enteros.

I. Bloques Funcionales del Computador Básico

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

Como podemos ver en la figura 1 el procesador 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.
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.

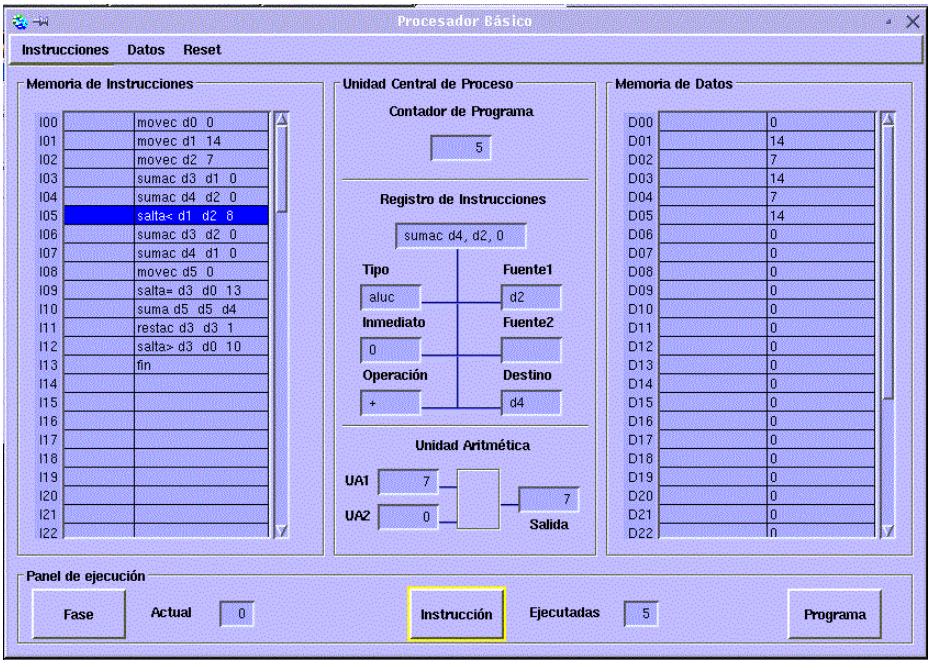

Figura 1

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, búsqueda de operandos, ejecución de la instrucción, escritura de resultados).

II. Conjunto de Instrucciones

Las instrucciones que soporta el Procesador 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)

Instrucciones de transferencia
movec d0, num (d0 = num)

Instrucciones de control de flujo
salta i5 (sigInstr = i5)
salta= d0,d1,i5 (si d0==d1 = sigInstr = i5)
salta> d0, d1, i5 (si d0>d1 = sigInstr = i5)
salta< d0, d1, i5 (di d0<d1 = sigInstr = 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.

A) 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.
B) 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 (Decodificación e Incremento del PC, Obtención de los operandos, Ejecución de la Instrucción, Almacenamiento de los resultados). 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?

G) Sigue ejecutando las 4 fases de la instrucción "sumac d3, d1, 0". ¿ 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 anubis, etc):

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

```
movec d0, 1
movec 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 la password os aparecerán todos los ficheros en vuestra cuenta y podréis copiar y pegar ficheros entre "anubis" y vuestro disco duro, siguiendo el procedimiento "Copy-Paste" explicado para el caso del disquete.

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

Problema Propuesto

- Implementa en un fichero llamando "divide.asm" la división de dos números enteros utilizando restas, tal cual se ha explicado en la clase de teoría. Cárgalo en el simulador y verifica su funcionamiento.