

## CONTENIDO DE LA CARPETA

**SISC-vN-plus.cct** Es el computador que os damos funcionando correctamente sobre el que tenéis que hacer los cambios para que pueda ejecutar las nuevas instrucciones. Tiene algunas modificaciones respecto al visto en clase para así facilitaros el trabajo. Se comentan más adelante.

**R\_Q+\_ori.txt** y **R\_O\_ori.txt** Ficheros de texto con el contenido de las dos ROMs de la unidad de control del SISC-vN-plus.cct tal como os lo entregamos. Estos ficheros os serán útiles para modificarlos con los cambios que requiera vuestro diseño para crear las nuevas ROMs que deberán sustituir a las originales.

**PruLMnew-Byte-1.hex** y **PruLMnew-Byte-0.hex** Ficheros del programa de prueba (que contiene dos de las nuevas instrucciones) ensamblado a lenguaje máquina para ser cargados en los dos módulos de memoria. Este programa en lenguaje ensamblador es el siguiente:

```
MOVI    R1, 0xF2
MOVI    R2, 0x03
MOVI    R7, 0xF0
ST      0(R7), R2
ADD     R3, R1, (R7)
IN      R0, PRINT-STATUS
BZ      R0, -2
OUT     PRINT-DATA, R3      ; Tiene que mostrar 0xFFFF5
CMPLT   R3, R1, (R7)
IN      R0, PRINT-STATUS
BZ      R0, -2
OUT     PRINT-DATA, R3      ; Tiene que mostrar 0x0001
BNZ     R0, -1              ; Se queda indefinidamente ejecutando
                                ; esta instrucción.
                                ; Para volver a ejecutar el código
                                ; hacer reset en el simulador.
```

**LibPrac6.clf** Librería que debéis tener abierta para salvar las dos ROMs que creéis para sustituir a las originales de la unidad de control del SISC-vN-plus. No contiene ningún dispositivo.

## COMENTARIOS QUE OS PUEDEN AYUDAR

Recordad que se puede añadir nuevo hardware tanto en la UP como en la UC y para simplificar el esquema lógico resultante, si es necesario, se puede usar la conexión por nombre de señales binarias y de buses. Si el nuevo hardware se obtiene recortando y pegando de algunos de los bloques con que está construido el computador debéis asegurarnos que las señales/buses de entrada/salida del bloque pegado no tengan nombre, que puede no estar visible, de lo contrario se crearía una conexión por nombre, no visible, que seguramente no deseáis).

**SISC-vN-plus.cct** Es el Computador SISC Von Neumann preparado para que os sea cómodo añadir nuevas instrucciones. El computador tiene cargado en memoria el programa KEY\_MEM\_PRINT que usasteis para probar las entradas salidas en la práctica 5. Se le entra por el teclado el número de datos que se entrarán por teclado a continuación, el programa los

guarda en memoria y luego los escribe en la impresora en el mismo orden que entraron. Esto os lo damos así simplemente para que veáis que el circuito funciona correctamente.

En el laboratorio, sobre este circuito, debéis:

- Añadir el hardware necesario (en la UP y/o en la UC, según hayáis decidido en vuestro diseño) para ejecutar las nuevas instrucciones
- Cambiar las dos ROMs de la unidad de control (RQ+ori y ROori) por las que vosotros creéis, que se podrían llamar RQ+new y RQnew.
- Cargar en memoria el programa de prueba que usa las nuevas instrucciones, que os damos ensamblado y separado en dos ficheros .hex para ser cargados en los dos módulos de memoria (PruLMoriginal-Byte-0.hex y PruLMoriginal-Byte-1.hex)
- Y probar que todo funciona correctamente.

Para facilitaros la labor hemos adaptado el computador que os entregamos de la siguiente manera:

- El reloj tiene un Tc de 2.000 ut. El tiempo de ciclo original (1.400 ut) puede ser insuficiente según qué cambios se realicen en el computador para poder ejecutar nuevas instrucciones. El nuevo tiempo de ciclo tiene que ser más que suficiente para implementar las nuevas instrucciones.
- Además del circuito generador del Clk, se ha dispuesto un binary switch que podéis usar para generar el Clk a mano, quitando el nombre Clk al la salida del generador de reloj y poniéndolo a la salida del binary switch.
- Se ha dejado más espacio entre dispositivos para que podáis añadir bloques en la Unidad de Proceso.
- El bloque SISC CONTROL UNIT tiene tres nuevos bits en la palabra de control: New3, N3w2, New1 y New0. Estos bits los genera directamente la ROM\_OUT de la unidad de control. Estos 4 nuevos bits de la ROM\_OUT son más que suficientes para implementar las nuevas instrucciones. Pueden usarse como nuevos bits de la palabra de control o para controlar multiplexores en la propia unidad de control.
- El circuito interno de la unidad de control, UC, (bloque SISC CONTROL UNIT) también se ha espaciado un poco por si tenéis que añadir nuevos bloques.
- El circuito tiene displays binarios (para las señales) y hexadecimales (para los buses). Están conectados por nombre. Si deseáis visualizar otra señal o bus en un visor podéis cambiar el nombre de la señal/bus del visor. Hacerlo con la simulación parada (y no cambiéis el nombre del visor ADDR-IO, que es de 8 bits, por otro bus de 16 bits).
- En la unidad de control la ROM\_Q+ se llama RQ+ori y su contenido es como el de la ROM\_Q+ original de teoría. El contenido de esta ROM se da en el fichero QQ+ori.txt de esta carpeta. La ROM\_OUT se llama ROori y es igual que la original de teoría pero con cuatro señales nuevas a su salida (New3,..., New0). Estas señales salen formando parte de la palabra de control por si necesitáis usarlas en la UP, pero también pueden usarse en el propio circuito de la UC. En este caso, para conectarlas por nombre debéis darle nombre a la señal (al cable rojo). Se puede usar el mismo nombre del conector pero hay que nombrar el cable.