



DEPARTAMENTO DE TECNOLOGÍA ELECTRÓNICA
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

El Computador Simple 2010 (CS2010)

*Enunciados de Prácticas de Laboratorio
Estructura de Computadores*

1. Introducción y objetivos

Los objetivos de esta práctica son:

- Ejercitar el uso del CS2010 y de su programación
- Comprobar el funcionamiento de la unidad de control
- Comprobar el funcionamiento de la unidad de datos
- Realizar y comprobar el funcionamiento de un programa sencillo para el CS2010 que realice operaciones de entrada-salida

Con estos objetivos se ha implementado¹ un sistema basado en el procesador CS2010 sobre la placa de prototipado Digilent Basys2 que se muestra en la figura 1. El sistema implementado incluye los siguientes componentes:

- El procesador
- La memoria de código (instrucciones o programa)
- La memoria de datos
- Dispositivos de entrada-salida mapeados en memoria
- Una unidad de depuración

¹ La implementación ha sido realizada por Jonathan Ruiz Páez como proyecto fin de carrera

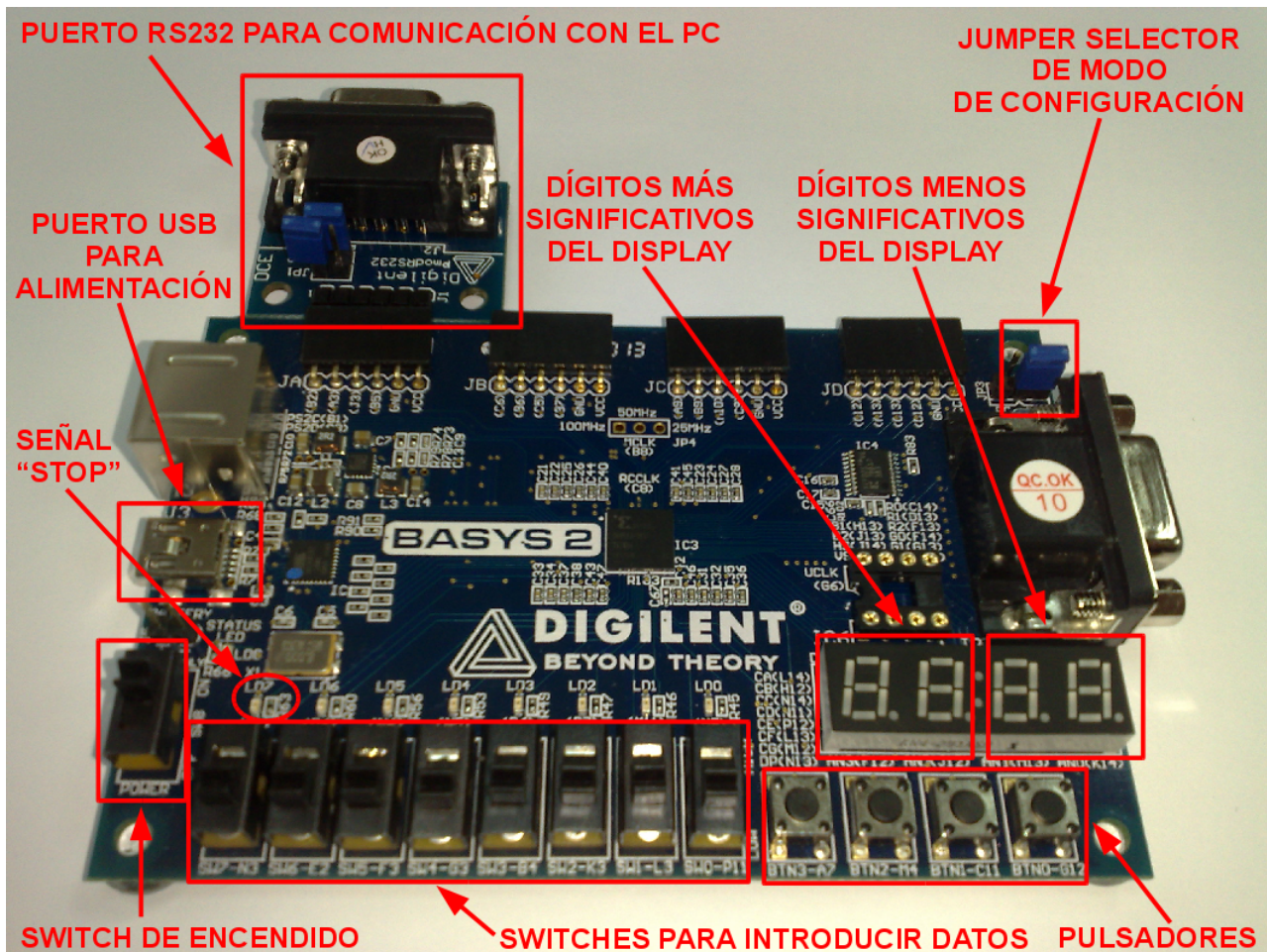


Figura 1. Placa de prototipado empleada en la práctica.

Los dispositivos de entrada-salida se han mapeado en el espacio de direccionamiento de la memoria de datos tal y como muestra la tabla siguiente:

| Dirección | Sentido | Dispositivo | Descripción |
|-----------|---------|--|---|
| \$80 | entrada | Switches | Cada uno de los 8 bits de esta posición de memoria indica si el switch correspondiente está activo. Así, el bit 7 valdrá uno si y sólo si sw7 está activo, el bit 6 valdrá uno si y sólo si sw6 está activo, etc. |
| \$81 | entrada | Pulsadores | Los 4 bits menos significativos de esta posición de memoria indican si los botones correspondientes están pulsados. Así, el bit 3 valdrá uno si y sólo si btn3 está pulsado, el bit 2 valdrá uno si y sólo si btn2 está pulsado, etc. Los 4 bits más significativos de la posición siempre valen 0. |
| \$82 | salida | dígitos menos significativos del display | El contenido de esta posición de memoria se muestra en hexadecimal en la derecha del display |
| \$83 | salida | dígitos más significativos del display | El contenido de esta posición de memoria se muestra en hexadecimal en la izquierda del display |

Tabla 1. Dispositivos mapeados en memoria.

La unidad de depuración es un componente que se comunica con el PC a través del puerto RS232 y

permite a éste leer y escribir la memoria de código y datos. Además, la unidad de depuración hace posible depurar los programas descargados mediante:

- Ejecución de ciclos de reloj individuales.
- Ejecución de instrucciones individuales.
- Inspección del contenido de la memoria y los registros.
- Inspección del estado de las líneas de control.
- Habilitación/inhibición del reloj del procesador.

La interacción con la unidad de depuración se realiza a través del software descrito en los apéndices.

2. Estudio teórico

1. Lea atentamente los apéndices y aprenda el formato de los programas en ensamblador del CS2010.
2. Escriba una subrutina del CS2010 que introduzca en el registro R7 el producto de R5 y R6 asumiendo notación sin signo.
3. Escriba un programa que, usando la subrutina anterior, calcule el producto de 22 por 10 y almacene el resultado en la posición de memoria 4.
4. Escriba una tabla que indique qué señales se activan y qué transferencias entre registros se realizan en cada ciclo de reloj durante la ejecución de las dos primeras instrucciones del programa del apartado anterior.
5. Usando la subrutina anterior, escriba un programa que implemente el siguiente algoritmo:

```
HACER
  MIENTRAS EL BOTÓN 0 (BTN0) NO ESTÉ PULSADO
    NO HACER NADA
  FIN MIENTRAS
R5<-NÚMERO CODIFICADO EN LOS SWITCHES
MIENTRAS EL BOTÓN 3 (BTN3) NO ESTÉ PULSADO
  NO HACER NADA
FIN MIENTRAS
R6<-NÚMERO CODIFICADO EN LOS SWITCHES
R7<-R5*R6
MUESTRA R7 EN EL DISPLAY
POR SIEMPRE
```

6. Compruebe que la sintaxis de los programas escritos es correcta usando el software disponible en la página de la asignatura.

NOTA IMPORTANTE: Es imprescindible traer a la sesión de laboratorio los ficheros de los programas ya escritos en soporte electrónico y con la sintaxis ya chequeada.

3. Estudio experimental

1. Cambie el jumper de selección de modo de configuración de la placa (JP3) para que conecte el pin central con el etiquetado con "ROM".
2. Conecte la placa al puerto RS232 del PC (vea la figura 1 para ver como conectar la extensión RS232 a la placa de prototipado).
3. Alimente la placa de prototipado a través del puerto USB del PC.
4. Cargue en la memoria de código el programa que realiza la multiplicación de 22 por 10 tal y como se describe en los apéndices.
5. Ejecute ciclo a ciclo las dos primeras instrucciones comprobando que en cada ciclo se activan las señales de control que esperaba y se realizan las transferencias entre registros que esperaba.
6. Ejecute instrucción a instrucción las dos líneas siguientes y compruebe que tienen el efecto esperado.
7. Habilite el reloj para ejecutar las instrucciones restantes y vuelva a deshabilitarlo para comprobar el resultado final.
8. Cargue en la memoria de código el programa que implementa el algoritmo del apartado 5 del estudio teórico y haga que se ejecute de forma autónoma habilitando el reloj del CS2010. Compruebe que el programa funciona correctamente multiplicando distintos valores.

NOTA: Si el software indica que no puede realizarse la conexión a pesar de haber conectado correctamente la extensión RS232, es posible que la placa no esté aun configurada para implementar el CS2010. En tal caso pulse sobre el icono "configurar BASYS2 para CS2010" y, cuando la grabación termine, apague y vuelva a encender la placa.

Apéndice 1: Formato de los programas en ensamblador del CS2010

El código binario que ejecuta el procesador se denomina "código máquina". Escribir directamente en código máquina no resulta práctico. En lugar de eso escribiremos los programas en un lenguaje más legible para los humanos denominado "lenguaje ensamblador". Los programas en ensamblador pueden ser traducidos al código binario ejecutable por la máquina mediante una herramienta denominada también "ensamblador". A esta acción se la denomina "ensamblar". Un programa en ensamblador del CS2010 puede contener 3 tipos de líneas:

- Línea en blanco: contiene únicamente caracteres de espacio, tabuladores y/o comentarios. Estas líneas son ignoradas por el ensamblador.
- Línea de instrucción de programa: describe una instrucción del CS2010 mediante los nemónicos vistos en teoría. Si una instrucción requiere un operando "numérico", este puede especificarse de cuatro formas:
 - Usando un numeral decimal, en cuyo caso puede ir precedido por un signo.
 - Usando un numeral binario, en cuyo caso debe ir precedido por el prefijo "0B" y tener

exáctamente 8 bits.

- Usando un numeral hexadecimal, en cuyo caso debe ir precedido por el prefijo "0x" o el prefijo "\$" y tener uno o dos dígitos hexadecimales.
- Usando una etiqueta: una etiqueta es una cadena que tiene asociado un valor de 8 bits. Deben empezar por una letra y pueden contener un número arbitrario de letras y dígitos. Más adelante veremos la forma de asociar valores a etiquetas.

Son numerales válidos "1234", "+2", "-1", "0b11000110", "\$4F", "0x32", "\$3". No son numerales válidos "12F3", "+0b10101010", "0b000000000000001", "0b10", "0x023".

- Línea de directiva de ensamblado: le indica al ensamblador que realice determinadas acciones. En el CS2010 son de dos tipos:
 - Directiva EQU: permite asociar un valor a una etiqueta. Su sintaxis es:

```
.EQU <etiqueta> = <numeral>
```

- Tras encontrar una línea como ésta, el ensamblador cambiará cualquier ocurrencia de la etiqueta que encuentre en las líneas de instrucción por el valor asociado. Nótese que estas líneas no se corresponderán con ninguna instrucción máquina del programa ensamblado.
- Directiva OP CODE: permite insertar instrucciones indicando explícitamente su código máquina. Su sintaxis es:

```
.OPCODE <byte_más_significativo> , <byte_menos_significativo>
```

donde <byte_más_significativo> debe ser un numeral y <byte_menos_significativo> puede ser un numeral o una etiqueta.

Existe una segunda forma de asignar valores a etiquetas: tanto a las líneas de instrucciones de programa como a las líneas con la directiva OP CODE puede añadirse un prefijo en la forma:

```
<etiqueta> :
```

Esto hará que el ensamblador asocie a la etiqueta la dirección de la posición de memoria que ocupará la instrucción máquina correspondiente a esta línea. Esto es útil para hacer más claros los programas que usan instrucciones de salto.

Otra forma de hacer más legibles los programas es añadir comentarios al final de las líneas. Los comentarios comienzan por el carácter ';'. Todo el texto situado entre ese carácter y el final de la línea es ignorado por el ensamblador.

Por último hay que hacer notar que el ensamblador no distingue entre mayúsculas y minúsculas.

A continuación se muestra un ejemplo de programa en ensamblador del CS2010:

```

;esto es un comentario y será ignorado por el ensamblador

LDI R0,15; esto mete en R0 el número 15
;los espacios en blanco y tabuladores son ignorados
        LDI  R1,$0F          ;esto mete en R1 el número 15
LDI  R2,-1;esto mete en R2 el número 255
;esto es otro comentario seguido de líneas en blanco

        .EQU menos1=0b11111111 ; esto no genera código máquina
.equ uno=1
        LDI R3,0
bucle:   SUBI R0, uno
BRCS termina
        st (r3), r0 ; no importa si usamos minúsculas
        ADDI R3, UNO

jmp bucle
        TERMINA:   STOP

;al ejecutarse esto se rellenarán las 14 primerasposiciones de memoria
;con los enteros entre el 14 y el 1 ordenados de forma descendente

```

Apéndice 2: Software de programación y depuración

Para interactuar con la unidad de depuración usaremos el software instalado en los PC del laboratorio. Dicho software está disponible en la página web de la asignatura. Para lanzarlo simplemente pulsamos sobre el icono correspondiente en el escritorio. Al hacerlo nos aparecerá la siguiente ventana:

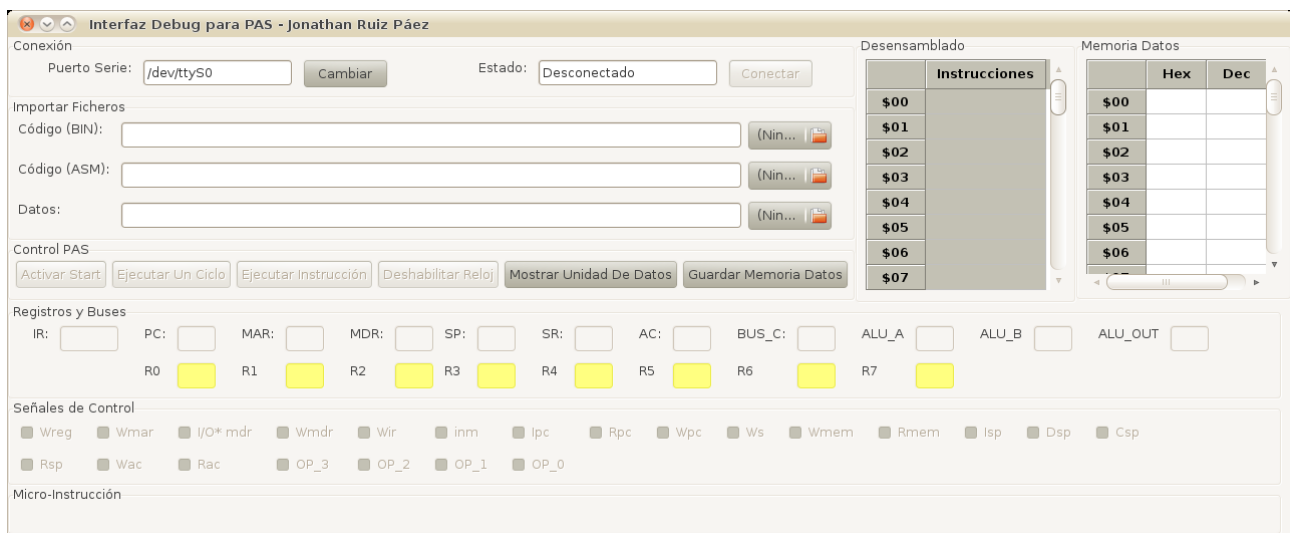


Figura 2: Software de interfaz con la unidad de depuración del CS2010.

La parte superior de la ventana contiene una serie de controles para interactuar con la unidad de depuración del CS2010 que se describen a continuación:

- Campo para la ruta del puerto serie. Aparece relleno por defecto con la correspondiente al puerto serie de los ordenadores del laboratorio, por lo que en principio no debe cambiarse.
- Campo de fichero de código binario (BIN): este campo debe contener la ruta del fichero binario

cuyo contenido se escribirá en la memoria de código cuando se inicialice el procesador. Al rellenarse se mostrará en la parte superior derecha de la ventana el contenido de la memoria desensamblado. Normalmente no tendremos que preocuparnos de rellenar este campo ya que usaremos un fichero en formato ensamblador para especificar el programa que deberá escribirse.

- Campo de fichero de código ensamblador (ASM): este campo debe contener la ruta de un fichero de código ensamblador correspondiente al programa que será escrito en la memoria de código. Al rellenar este campo la herramienta ensamblará el fichero informando de los posibles errores y escribirá la ruta del fichero binario resultante en el campo comentado anteriormente. Rellenar este campo no es obligatorio, pero es la forma habitual de trabajar con la herramienta.
- Campo de fichero de datos: si deseamos inicializar la memoria de datos con el contenido de un fichero binario, podemos indicar su ruta en este campo.
- Botón de establecimiento de conexión e inicialización: al pulsarlo el software establece una conexión a través del puerto RS232 con la unidad de depuración, ordena inicializar el CS2010 y escribe las memorias de código y datos con el contenido de los ficheros especificados. Únicamente tras esto podremos interactuar con la unidad de depuración.
- Botón “Mostrar Unidad De Datos”: al pulsar este botón aparecerá la ventana siguiente:

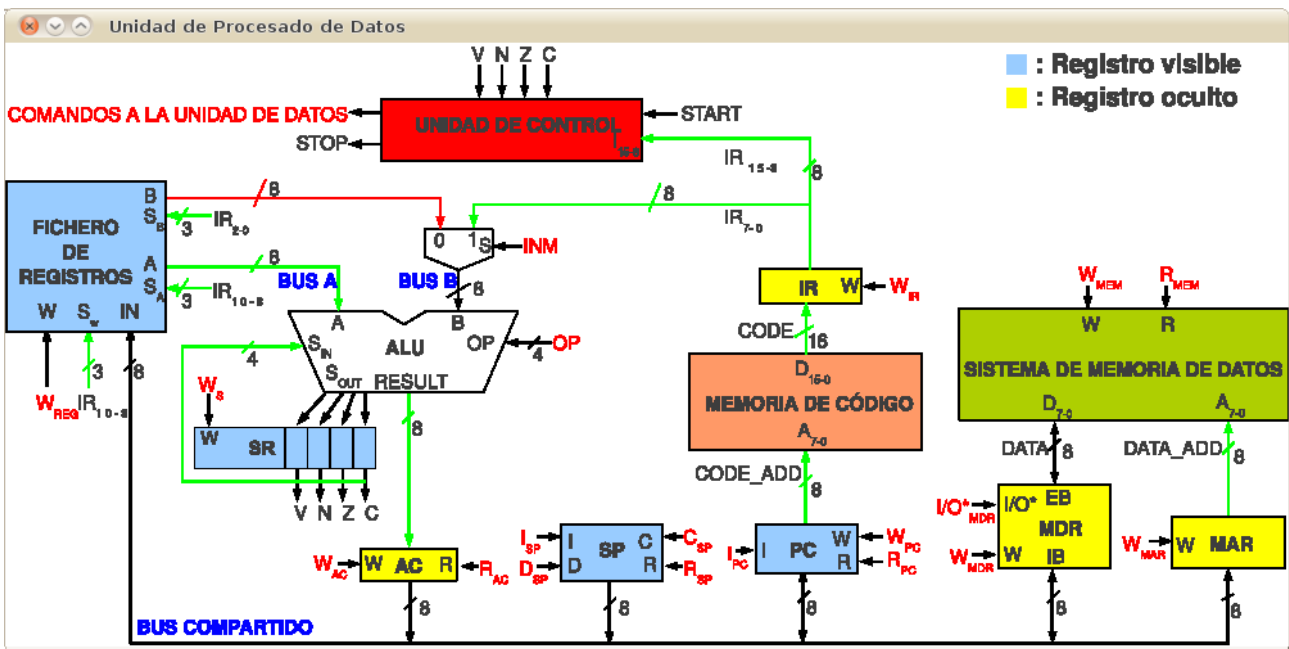


Figura 3: Unidad de datos del CS2010.

Durante las ejecuciones ciclo a ciclo se resaltarán en esta ventana las señales que se vayan activando.

- Botón “Habilitar Reloj”/“Deshabilitar Reloj”: Si el reloj del CS2010 está detenido, este control permite ponerlo en marcha para que se ejecute el programa de forma autónoma hasta encontrar una instrucción de STOP. Si el reloj está ya en marcha permite detenerlo para inspeccionar el estado del procesador y la memoria.
- Botón “Activar Start”: Le indica a la unidad de depuración que genere un pulso alto en la señal de inicio del CS2010.
- Botón “Ejecutar Un Ciclo”: Le indica a la unidad de depuración que habilite el reloj durante un

solo ciclo.

- Botón “Ejecutar Una Instrucción”: Le indica a la unidad de depuración que habilite el reloj hasta que se ejecute la instrucción en curso o hasta que se llegue al estado de espera.

Cuando el reloj del CS2010 está detenido se muestra el estado del procesador incluyendo:

- Contenido de la memoria de datos.
- Contenido de los registros visibles.
- Contenido de los registros ocultos.
- Estado de las líneas de control.
- Valor de los buses de entrada y salida de la ALU y del bus común.

Además, la instrucción en curso aparecerá resaltada. El contenido de las posiciones memoria de datos puede ser editado manualmente. También es posible guardar el contenido de la memoria de datos en un fichero binario pulsando sobre el botón “Guardar Memoria Datos”.