

CRONOS. Generador de cronogramas

La aplicación, que corresponde a la simulación del funcionamiento de un esquema de computador sencillo, contempla los tres programas siguientes:

- CRONOS
- GENCRON
- MODIFICA

PROGRAMA CRONOS

CRONOS es un programa didáctico que genera cronogramas de un repertorio de instrucciones determinado, de una arquitectura de computador simple; permitiendo a su vez observar una simulación de la ejecución de instrucciones que se defina.

La aplicación CRONOS se complementa con los programas: GENCRON y MODIFICA. Mediante GENCRON, definimos cronogramas. Con la aplicación MODIFICA, se puede modificar el cronograma asociado a una instrucción; el nuevo cronograma ha debido definirse previamente con la aplicación GENCRON.

Menú de edición

Cuando se ejecuta el programa Cronos, la primera pantalla en visualizarse es la correspondiente al menú de edición. En él se presentan las posibles opciones de operación con el fichero de instrucciones.

El fichero de instrucciones no es más que un fichero ASCII, que contiene aquellas instrucciones que queremos se definan sus cronogramas.

En el menú de edición las posibles opciones son:

- *Editar fichero de instrucciones.*
Esta opción llama al editor de Cronos; pudiéndose editar cualquier fichero de instrucciones definido.
Sólo se puede elegir esta opción cuando se haya dado nombre al fichero mediante las opciones: *Escribir en . . .* ó *Nuevo*.
- *Escribir en . . .*
Esta opción permite introducir el nombre del fichero de instrucciones que va a ser utilizado. La extensión de este fichero debe ser **.CRO**.
- *Salvar.*
Esta opción graba en el directorio activo el fichero de instrucciones creado.
- *Nuevo.*
Esta opción es semejante a la de *Escribir en . . .*, salvo que toma como nombre de fichero de instrucciones el nombre **NONAME.CRO**.
- *Directorio.*
Esta opción cambia el directorio activo en el cual se leerán y escribirán los ficheros de instrucciones.
- *Salir al menú principal.*
Para salir al menú principal. No podrá ser elegida a menos que se tenga cargado un fichero de instrucciones exento de errores.

En la edición se puede obtener ayuda de edición pulsando la tecla “F1”. Para obtener ayuda sobre las instrucciones hay que pulsar “CTRL-F1”.

Se finaliza el proceso de edición pulsando la tecla “Esc”. Esto provoca el análisis del contenido del fichero de instrucciones. Si dicho análisis finaliza correctamente, se entra directamente en el menú de edición, con el fichero cargado dispuesto para entrar en el menú principal. Si por el contrario se detecta algún error, aparecerá en pantalla un mensaje indicando el tipo de error y la línea donde se produce; volviendo a entrar en el editor, para que el error sea subsanado.

Todos los ficheros editados con las instrucciones tienen que terminar con la directiva *END.* .

En cualquier momento es posible abortar una operación, pulsando la tecla “Esc”.

INSTRUCCIONES

Las instrucciones que se contemplan en este simulador son las siguientes:

| | | | |
|-----|------|------|-----|
| ADC | BZ | DIV | NOT |
| AND | CALL | ENDL | OR |
| BC | CLC | INC | RET |
| BNC | CLE | LD | SHR |
| BNS | CLS | LOOP | ST |
| BNZ | CLZ | MOVE | SUB |
| BS | DEC | MULT | XOR |

ADC

Esta instrucción produce la suma teniendo en cuenta el bit de acarreo de los operandos. Los modos de direccionamiento válidos son:

```

ADC .X .Y
ADC .X .Y++
ADC .X AAA
ADC .X [AAA]
ADC .X (.Y) AAA
ADC .X [(Y) AAA]
ADC .X (.Y++) AAA
ADC .X [(Y++) AAA]

```

Donde X e Y son registros numerados del 0 al 9 y AAA es un número decimal.

AND

Esta instrucción produce la operación lógica AND sobre los operandos. Los modos de direccionamiento válidos son:

```

AND .X .Y
AND .X .Y++
AND .X AAA
AND .X [AAA]
AND .X (.Y) AAA
AND .X [(Y) AAA]
AND .X (.Y++) AAA
AND .X [(Y++) AAA]

```

Donde X e Y son registros numerados del 0 al 9 y AAA es un número decimal.

BC

Esta instrucción produce una bifurcación al lugar indicado por el operando sólo si el bit de acarreo es 1. Es obligatorio colocar la directiva {C0} o {C1} al final. Los modos de direccionamiento son:

```
BC .X
BC .X++
BC AAA
BC [AAA]
BC (.X) AAA
BC [(X) AAA]
BC (.X++) AAA
BC [(X++) AAA]
```

Siendo X un registro (0..9) y AAA es un número decimal.

BNC

Esta instrucción produce una bifurcación al lugar indicado por el operando sólo si el bit de acarreo es 0. Es obligatorio colocar la directiva {C0} o {C1} al final. Los modos de direccionamiento son:

```
BNC .X
BNC .X++
BNC AAA
BNC [AAA]
BNC (.X) AAA
BNC [(X) AAA]
BNC (.X++) AAA
BNC [(X++) AAA]
```

Siendo X un registro (0..9) y AAA es un número decimal.

BNS

Esta instrucción produce una bifurcación al lugar indicado por el operando sólo si el bit de signo es 0. Es obligatorio colocar la directiva {S0} o {S1} al final. Los modos de direccionamiento son:

```
BNS .X
BNS .X++
BNS AAA
BNS [AAA]
BNS (.X) AAA
BNS [(X) AAA]
BNS (.X++) AAA
BNS [(X++) AAA]
```

Siendo X un registro (0..9) y AAA es un número decimal.

BNZ

Esta instrucción produce una bifurcación al lugar indicado por el operando sólo si el bit de cero es 0. Es obligatorio colocar la directiva {Z0} o {Z1} al final. Los modos de direccionamiento son:

```
BNZ .X
BNZ .X++
BNZ AAA
BNZ [AAA]
BNZ (.X) AAA
BNZ [(X) AAA]
BNZ (.X++) AAA
BNZ [(X++) AAA]
```

Siendo X un registro (0..9) y AAA es un número decimal.

BS

Esta instrucción produce una bifurcación al lugar indicado por el operando sólo si el bit de signo es 1. Es obligatorio colocar la directiva {S0} o {S1} al final. Los modos de direccionamiento son:

```
BS .X
BS .X++
BS AAA
BS [AAA]
BS (.X) AAA
BS [(X) AAA]
BS (.X++) AAA
BS [(X++) AAA]
```

Siendo X un registro (0..9) y AAA es un número decimal.

BZ

Esta instrucción produce una bifurcación al lugar indicado por el operando sólo si el bit de cero es 1. Es obligatorio colocar la directiva {Z0} o {Z1} al final. Los modos de direccionamiento son:

```
BZ .X
BZ .X++
BZ AAA
BZ [AAA]
BZ (.X) AAA
BZ [(X) AAA]
BZ (.X++) AAA
BZ [(X++) AAA]
```

Siendo X un registro (0..9) y AAA es un número decimal.

CALL

Esta instrucción llama a una subrutina. Es necesario haberla definido previamente y colocar su nombre al final de la instrucción. Modos de direccionamiento:

```
CALL .X      {NOM_SUB}
CALL .X++    {NOM_SUB}
CALL AAA     {NOM_SUB}
CALL [AAA]   {NOM_SUB}
CALL (.X) AAA {NOM_SUB}
CALL [(.X) AAA] {NOM_SUB}
CALL (.X++) AAA {NOM_SUB}
CALL [(.X++) AAA] {NOM_SUB}
```

Donde X es un registro (0..9) y AAA es un número decimal.

CLC

Coloca a cero el bit de acarreo. No lleva parámetros

CLE

Coloca a cero el bit de interrupción externa. No lleva parámetros

CLS

Coloca a cero el bit de signo. No lleva parámetros

CLZ

Coloca a cero el bit de cero. No lleva parámetros

DEC

Esta instrucción decrementa el contenido de un registro en una unidad. Sólo dispone de un modo de direccionamiento:

```
DEC .X
```

Donde X es un registro numerado del 0 al 9

DIV

Esta instrucción produce la división de los operandos. Los modos de direccionamiento válidos son:

```
DIV .X .Y
DIV .X .Y++
DIV .X AAA
DIV .X [AAA]
DIV .X (.Y) AAA
DIV .X [(.Y) AAA]
DIV .X (.Y++) AAA
DIV .X [(.Y++) AAA]
```

Donde X e Y son registros numerados del 0 al 9 y AAA es un número decimal.

ENDL

Esta instrucción debe cerrar un conjunto de instrucciones encabezado con un LOOP. Si no cerramos correctamente el análisis termina indicando el error.

INC

Esta instrucción incrementa el contenido de un registro en una unidad. Solo dispone de un modo de direccionamiento:

INC .X

Donde X es un registro numerado del 0 al 9.

LD

La instrucción LD se encarga de cargar el contenido de una dirección de memoria en un registro determinado. Los modos de direccionamiento correctos son:

LD .X AAA
LD .X [AAA]
LD .X (.Y) AAA
LD .X [(Y) AAA]
LD .X (Y++) AAA
LD .X [(Y++) AAA]

Donde X e Y son registros del 0 al 9 y AAA es un número decimal.

LOOP

Esta instrucción genera un bucle que se repetirá tantas veces como indique el direccionamiento. Los modos de direccionamiento correctos son:

LOOP .X
LOOP .X++
LOOP AAA
LOOP [AAA]
LOOP (.X) AAA
LOOP [(X) AAA]
LOOP (.X++) AAA
LOOP [(X++) AAA]

Donde X es un registro (0..9) y AAA es un número decimal.

MOVE

La instrucción MOVE se encarga de cargar el contenido de un registro en otro registro. Sólo dispone de un modo de direccionamiento que es el siguiente:

MOVE .X .Y

donde X e Y son registros numerados del 0 al 9.

MULT

Esta instrucción produce la multiplicación de los operandos. Los modos de direccionamiento válidos son:

MULT .X .Y
MULT .X .Y++
MULT .X AAA
MULT .X [AAA]
MULT .X (.Y) AAA
MULT .X [(Y) AAA]
MULT .X (Y++) AAA

MULT .X [(Y++) AAA]

Donde X e Y son registros numerados del 0 al 9 y AAA es un número decimal.

NOT

Esta instrucción produce la operación lógica NOT sobre el operando. Sólo dispone de un modo de direccionamiento:

NOT .X

Donde X es un registro numerado del 0 al 9.

OR

Esta instrucción produce la operación lógica OR sobre los operandos. Los modos de direccionamiento válidos son:

OR .X .Y
 OR .X .Y++
 OR .X AAA
 OR .X [AAA]
 OR .X (.Y) AAA
 OR .X [(Y) AAA]
 OR .X (.Y++) AAA
 OR .X [(Y++) AAA]

Donde X e Y son registros numerados del 0 al 9 y AAA es un número decimal.

RET

Este comando es necesario al finalizar un conjunto de instrucciones encabezado por DEF. Estas instrucciones comprenden una subrutina. Cuando se encuentra se produce un retorno al programa principal.

SHR

Esta instrucción produce un desplazamiento de un bit a la derecha en un registro. Solo dispone de un modo de direccionamiento:

SHR .X

Donde X es un registro numerado del 0 al 9.

ST

La instrucción ST carga el contenido de un registro en una posición de memoria determinada. Los direccionamientos válidos para esta instrucción son:

ST .X AAA
 ST .X [AAA]
 ST .X (.Y) AAA
 ST .X [(Y) AAA]
 ST .X (.Y++) AAA
 ST .X [(Y++) AAA]

Donde X e Y son registros del 0 al 9 y AAA es un número decimal.

SUB

Esta instrucción produce la resta de los operandos. Los modos de direccionamiento válidos son:

```
SUB .X .Y
SUB .X .Y++
SUB .X AAA
SUB .X [AAA]
SUB .X (.Y) AAA
SUB .X [(Y) AAA]
SUB .X (.Y++) AAA
SUB .X [(Y++) AAA]
```

Donde X e Y son registros numerados del 0 al 9 y AAA es un número decimal.

XOR

Esta instrucción produce la operación lógica XOR sobre los operandos. Los modos de direccionamiento válidos son:

```
XOR .X .Y
XOR .X .Y++
XOR .X AAA
XOR .X [AAA]
XOR .X (.Y) AAA
XOR .X [(Y) AAA]
XOR .X (.Y++) AAA
XOR .X [(Y++) AAA]
```

Donde X e Y son registros numerados del 0 al 9 y AAA es un número decimal.

DIRECTIVAS

Como en cualquier programación en ensamblador, las directivas son indicaciones al traductor. En este caso, las directivas se añaden al final de la instrucción, modificando en cierta forma sus cronogramas. Podemos diferenciar los siguientes tipos de directivas:

- *Directivas de bifurcación.* Van asociadas a las instrucciones de bifurcación (**BC**, **BNC**, **BS**, **BNS**, **BZ** y **BNZ**). Indican el valor que se supone tendrá el biestable de estado implicado cuando se ejecute esa instrucción, definiendo de esa forma el cronograma correcto correspondiente.

{C0} {C1} {S0} {S1} {Z0} {Z1}

- *Directivas de trap o de interrupción.* Indican que durante la ejecución de una instrucción se produce una interrupción externa o una interna (trap). Las interrupciones implican bifurcación; en este caso, la bifurcación es meramente simbólica, produciéndose simplemente a nivel de cronograma; se continúa con el cronograma de la instrucción que está definida a continuación en el fichero de instrucciones.

{E} {D} {P}

- *Directivas de programa.* Estas directivas no tienen repercusión alguna en los cronogramas y simplemente dan orientaciones al programa en el análisis del fichero de instrucciones.

DEF Definición de subrutina. Es necesario colocar el nombre de la subrutina detrás de éste:

DEF *NOM SUB*

INSTRUCCIÓN

INSTRUCCION

...

RET Debe terminar siempre con esta instrucción

END. Final del fichero de instrucciones

{CDR} Cronogramas con el periodo de decodificación complejo. Puede aparecer en cualquier línea del fichero de instrucciones, siempre que en dicha línea no aparezca ninguna otra instrucción.

CONSIDERACIONES DE SINTAXIS EN LA EDICIÓN.

Al definir el fichero de instrucciones hay que tener en cuenta las siguientes consideraciones.

- En cada línea debe aparecer obligatoriamente una sola instrucción.
- Las instrucciones se pueden definir tanto en mayúsculas como en minúsculas.
- No pueden aparecer líneas en blanco.
- Se pueden incluir comentarios; debiendo comenzar por “;” y debiendo seguir siempre en la línea a una instrucción.

MENÚ PRINCIPAL.

Una vez salimos del menú de edición, vamos al menú principal. Las opciones disponibles en este menú son:

- *Menú de edición.* Para volver al menú de edición.
- *Flujo de información.* Para visualizar el flujo de información en el sistema computador simple planteado.
- *Cronograma.* Para visualizar el cronograma en la pantalla de las instrucciones definidas en el archivo de instrucciones.
- *Imprimir.* Se imprime el cronograma por la impresora.
- *Salir del programa.* Termina la ejecución del programa. Antes de salir, se pide confirmación y, en caso de haberse modificado el fichero de instrucciones sin que se hayan salvado los cambios, se pregunta si se quieren almacenar dichos cambios.

PROGRAMA GENCRON

GENCRON es un programa de definición de cronogramas correspondientes a las instrucciones del sencillo sistema computador propuesto.

La primera pantalla que se visualiza al ejecutar Gencron es la plantilla de cronogramas; en ella aparecen todas las señales de control existentes en la arquitectura de computador propuesta inicializadas a cero. Con el cursor nos movemos por toda la plantilla, con el fin de ir definiendo el valor de las distintas señales de control en cada periodo. Con la barra espaciadora se van obteniendo alternativamente los distintos valores posibles para cada una de las señales de control.

Podemos definir el periodo de decodificación con la pulsación de la tecla “P”. Existen dos tipos de periodos de decodificación: el simple y el complejo. En el simple la unidad de control únicamente realiza la función de decodificación; y, en el complejo, además de la decodificación, realiza el cálculo de la dirección con direccionamiento relativo. Con la tecla “C”, activamos y desactivamos alternativamente el periodo de decodificación complejo.

En el programa existe la opción de borrar todas las señales desde la posición del cursor hasta el final de la plantilla de cronograma; esto se realiza pulsando la tecla “V”. Esta operación es necesario realizarla cuando queremos que el archivo que definamos con esta aplicación, sustituya posteriormente al asociado a una de las instrucciones del repertorio de instrucciones existentes. Para definir el cronograma de una instrucción hay que tener en cuenta además que:

- El número máximo de periodos asociados a una instrucción es de once.
- Sólo se permite un periodo de decodificación.
- Si la instrucción implica registros de propósito general, debemos seleccionar .1 y .2, si son dos, y en ese orden, y .1, si sólo es necesario definir un registro.
- Se puede almacenar el archivo del cronograma definido con Gencron, pulsando la tecla “F2”. El nombre del fichero no debe tener extensión ya que la aplicación lo va a salvar con la extensión .CAR.

Si al entrar en la aplicación Gencron queremos trabajar con un archivo .CAR existente, se debe pulsar la tecla “L” desde la plantilla de cronogramas.

Para terminar la ejecución del programa Gencron, se debe pulsar la tecla “Q”.

PROGRAMA MODIFICA

Mediante la aplicación MODIFICA podemos modificar el cronograma de una instrucción con otro que previamente ha sido definido con la aplicación Gencron.

Al comenzar su ejecución, se pide el nombre del fichero creado anteriormente con la aplicación Gencron. Si no queremos continuar, podemos teclear EXIT para salir del programa.

Una vez introducido el nombre del fichero .CAR, el programa pide el nombre de la instrucción que se va a modificar (ADC, AND, BC-1, BNC-1, BNS-1, BNZ-1, BS-1, BZ-1, BC-0, BNC-0, BNS-0, BNZ-0, BS-0, BZ-0, CALL, CLC, CLE, CLS, CLZ, DEC, DEF, DIV, ENDL, INC, LD, LOOP, MOVE, MULT, NOT, OR, RET, SHR, ST, SUB y XOR. Se puede teclear EXIT para salir del programa.

Cuando se ha introducido el nombre de la instrucción, el programa muestra todos los tipos de direccionamiento que puede tener la instrucción seleccionada. El usuario debe

desplazarse con las teclas de cursor hasta posicionarse sobre el modo de direccionamiento deseado. En ese momento se debe pulsar ENTER. A partir de ahí, la aplicación Modifica analiza el cronograma introducido y, en caso de ser correcto, sustituirá al anterior de la instrucción seleccionada; en caso de ser incorrecto, se abortará la operación, finalizando la ejecución sin modificar el cronograma.

Cada instrucción tiene un número de periodo en el cual la unidad de control comienza a atender las posibles interrupciones internas (traps) que se puedan producir durante su ejecución. Si la instrucción se cambia, el trap se debe atender en otro periodo. El programa Modifica también ofrece la posibilidad de modificar el número de periodo a partir del cual la Unidad de Control atienda el trap. Si no se quiere modificar, hay que teclear “N” cuando interroge tal circunstancia; si se quiere modificar, se pulsa “S” y posteriormente el número de periodo que el usuario crea conveniente.