

PROBLEMAS

TEMA 6: Diseño del procesador

Problemas propuestos en examen

6.1 Para la UCP con bus interno de la Figura 6.1 escribir la secuencia de microoperaciones que se necesitan para sumar un número con el acumulador cuando el número es:

- a) Un operando inmediato.
- b) Un operando con direccionamiento directo.
- c) Un operando con direccionamiento indirecto.

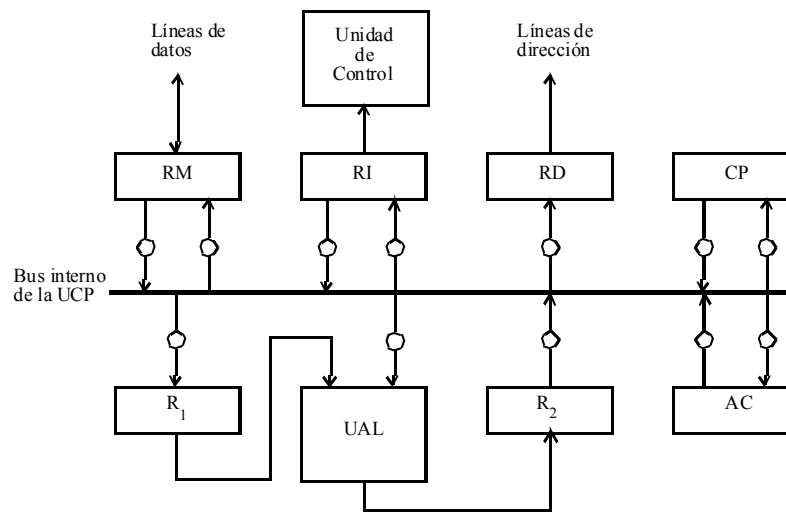


Figura 6.1

6.2 En la Figura 6.2 se muestra una UCP (Unidad de Control + Unidad de Procesamiento) y en la Tabla 6.2 se da el significado de cada una de las señales de control. La señal de condición s_0 se activa cuando el contenido del registro acumulador (AC) sea igual a cero. Obtener las secuencias de las microoperaciones y las señales de control que hay que activar para que esta UCP ejecute las siguientes instrucciones, suponiendo que éstas se encuentran ya almacenadas en el registro de instrucciones (RI) :

- a) Cargar acumulador con el contenido de una dirección de memoria.
- b) Almacenar acumulador en una dirección de memoria.
- c) Sumar al acumulador un operando con direccionamiento inmediato.
- d) Sumar al acumulador un operando con direccionamiento directo.
- e) Sumar al acumulador un operando con direccionamiento indirecto.
- f) AND con el acumulador de un operando en modo directo.
- g) Bifurcar, en modo de direccionamiento directo.
- h) Bifurcar si acumulador = 0, en modo de direccionamiento directo.

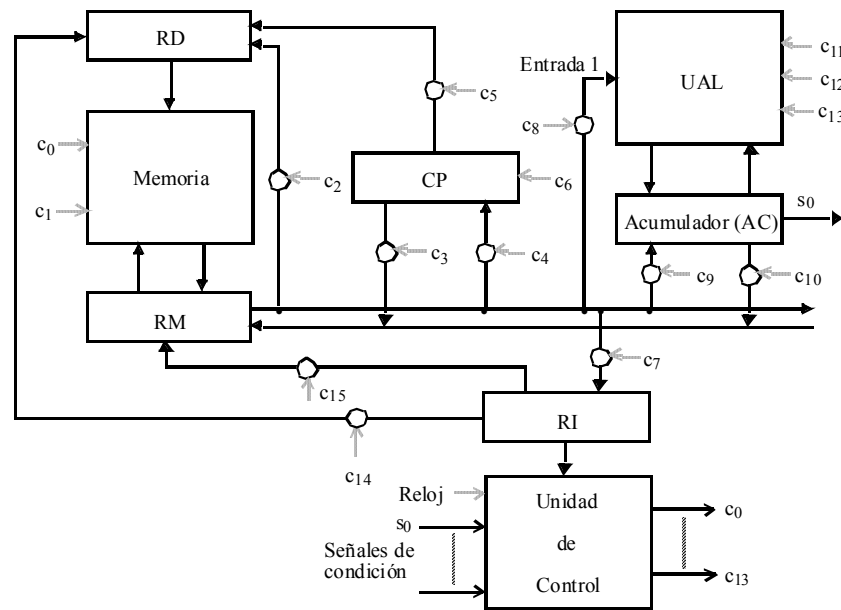


Figura 6.2: Caminos de datos y puntos de control

Señal de control	Microoperación controlada
c ₀	Leer de la memoria
c ₁	Escribir en la memoria
c ₂	Transferir el contenido de RM a RD ($RD \leftarrow (RM)$)
c ₃	Transferir el contenido de CP a RM ($RM \leftarrow (CP)$)
c ₄	Transferir el contenido de RM a CP ($CP \leftarrow (RM)$)
c ₅	Transferir el contenido de CP a RD ($RD \leftarrow (CP)$)
c ₆	Incrementar en 1 el contenido del CP ($CP \leftarrow (CP) + 1$)
c ₇	Transferir el contenido de RM a RI ($RI \leftarrow (RM)$)
c ₈	Transferir el contenido de RM a la entrada 1 de la UAL
c ₉	Transferir el contenido de RM a AC ($AC \leftarrow (RM)$)
c ₁₀	Transferir el contenido de AC a RM ($RM \leftarrow (AC)$)
c ₁₁	$AC \leftarrow (AC) + \text{Entrada 1}$
c ₁₂	$AC \leftarrow (AC) \text{ AND } \text{Entrada 1}$
c ₁₃	Complementar el contenido de AC ($AC \leftarrow (\overline{AC})$)
c ₁₄	Transferir el campo de dirección de RI a RD ($RD \leftarrow (RI(\text{dirección}))$)
c ₁₅	Transferir el campo de datos de RI a RM ($RM \leftarrow (RI(\text{datos}))$)

Tabla 6.2: Señales de control de la CPU de la figura 6.2

6.3 Indique si las siguientes afirmaciones son ciertas :

- I. Para la fase de ejecución del ciclo de instrucción existen tantas secuencias de microinstrucciones como códigos de operación existen en el repertorio de instrucciones.
- II. Cada fase del ciclo de instrucción se puede descomponer en una secuencia de operaciones elementales llamadas instrucciones máquina

6.4 En un procesador con instrucciones de cero direcciones (procesador con pila), ¿qué secuencia de instrucciones calcula la expresión $X = Y^2(X+Z)$?

- A) Push[X]; Push[Z]; Add; Push[Y]; Push[Y]; Mult; Mult; Pop[X];
- B) Push[Y]; Push[Y]; Push[X]; Push[Z]; Add; Mult; Mult; Pop[X];
- C) Push[Y]; Push[Y]; Mult; Push[X]; Push[Z]; Add; Mult; Pop[X];
- D) Todas las anteriores

6.5 El formato de instrucción de un procesador de una dirección tiene 6 bits para el código de operación y 10 para la dirección del operando. Suponiendo que una instrucción de bifurcación, con direccionamiento relativo al contador de programa, almacenada en la posición 530 (en decimal), origina un salto a la posición 620 (en decimal). Si el código de operación de la instrucción de salto es 110011, ¿cuál es la codificación en binario de dicha instrucción?

6.6 Considérese un procesador con instrucciones de cero direcciones (procesador con pila). Se tiene la siguiente secuencia de instrucciones:

Push M[C]; Push M[D]; Add; Push M[C]; Push M[D]; Add; Mult; Pop M[A]

¿Que expresión calcula?

6.7 Empleando un procesador de una dirección (procesador con acumulador) con un registro R1, indicar qué operación calcula la secuencia de instrucciones: Load A; Add B; Add C; Mult D; Add A; Store R1, Mult R1, Div A, Store C.

6.8 En un procesador con instrucciones de cero direcciones, indicar cuantos accesos a memoria principal se necesitan para completar la secuencia $Y = Y(X+Z)$. Para el cálculo de accesos a memoria se supondrá que el ancho de palabra es suficiente para que cada instrucción ocupe una única palabra de memoria. También se supondrá que la pila es cableada.

6.9 Empleando un procesador de una dirección (procesador con acumulador) con un registro R1, indicar qué operación calcula la secuencia de instrucciones: Load A; Add B; Mult D; Add A; Store R1, Mult R1, Mult R1; Mult A; Div B; Add A; Store C.

6.10. Cierta procesador tiene 32 registros, utiliza operandos inmediatos de 16 bits y tiene 142 instrucciones en su repertorio de instrucciones. En un determinado programa el 20% de las instrucciones tienen un registro de entrada y un registro de salida, el 30% tienen dos registros de entrada y uno de salida, el 25% tiene un registro de entrada, un registro de salida y un operando inmediato y el 25% restante tiene un operando inmediato y un registro de salida

a) ¿Cuántos bits se necesitan para cada uno de los cuatro tipos de instrucciones? Supóngase que se impone que todas las instrucciones tengan una longitud múltiplo de 8 bits.

b) ¿Cuánta memoria menos ocupa el programa cuando se utiliza una codificación del repertorio de instrucciones de longitud variable que cuando se utiliza una codificación de longitud fija?

Nota: Un repertorio de instrucciones es de longitud fija cuando todas las instrucciones tienen la misma longitud y es de longitud variable si la longitud de las instrucciones no es igual para todas las instrucciones del repertorio.

6.11 En un procesador con instrucciones de cero direcciones, determinar cuantos accesos a memoria se necesitan para completar la secuencia $Y = (X+Y)+Z$.

6.12 Empleando un procesador de una dirección (procesador con acumulador) con un banco de registro Ri, indicar qué operación calcula la secuencia de instrucciones: Load X; Add Y; Add Z; Mult X, Store R1, Mult R1, Div X, Store X.