PROGRAMADOR UNIVERSITARIO
LICENCIATURA EN INFORMÁTICA
INGENIERÍA EN INFORMÁTICA
Facultad de Ciencias Exactas y Tecnología
Universidad Nacional de Tucumán

ARQUITECTURA Y ORGANIZACIÓN DE COMPUTADORAS
Proyecto (primera parte)
Diseño de la MIC-1

Introducción

En la primera parte del proyecto vieron el camino de datos de nuestro procesador MIC-1. En esta parte, vamos a completar la microarquitectura, agregándole la unidad de control. También se vera como funcionan las operaciones de memoria, aunque no serán implementadas aquí.

Operaciones de memoria

La arquitectura MIC-1 trabaja con una memoria que tiene dos puertos de datos: uno es de 32-bit y el otro de 8-bit. Internamente, la misma contienen celdas de 8-bits (1 Byte) y se direcciona a través de un solo puerto de 32-bits. Esto significa que, por ejemplo, cuando se direcciona la dirección x00AB4200, el puerto de 8-bit tendrá el Byte guardado en x00AB4200, mientras el puerto de 32-bits tendrá el mismo Byte y los siguientes 3 Bytes, osea, los 4 Bytes que se encuentran en x00AB4200, x00AB4201, x00AB4202 y x00AB4203.

Los dos puertos están manejados por diferentes registros:

- El de 32-bits, nuestro tamaño de palabra, se controla a través de los registros MAR (Memory Address Register) y MDR (Memory Data Register). MAR se usa para direccionar mientras MDR se usa para los datos, que, serán leídos de memoria y o escritos a ella.
- El de 8-bits, 1 Byte, se controla a través de los registros PC (Program Counter) y MBR (Memory Buffer Register). PC se usa para direccionar y es el registro que controla la ejecución de las instrucciones, en otras palabras, PC contiene la dirección de la próxima instrucción a ejecutarse. A través de este puerto solo se lee datos de memoria y se los escribe a MBR. Como el ancho de datos es de 1 Byte, solo se escribe el primer Byte de MBR (el resto permanecen en 0).

Para hacer una operación con memoria, MAR o PC deben primero ser cargados antes de iniciar la operación, sea lectura o escritura. La diferencia entre estos dos registros de direcciones es que MAR contienen direcciones para leer palabras, mientras PC contienen direcciones para leer Bytes. En otras palabras, valores x00000000, x00000001, x00000002,

etc. en MAR significan palabras consecutivas, mientras los mismos en PC significan bytes consecutivos. Por esta razón, para dirreccionar correctamente la memoria, MAR y PC están conectados de manera diferente al bus de direcciones.

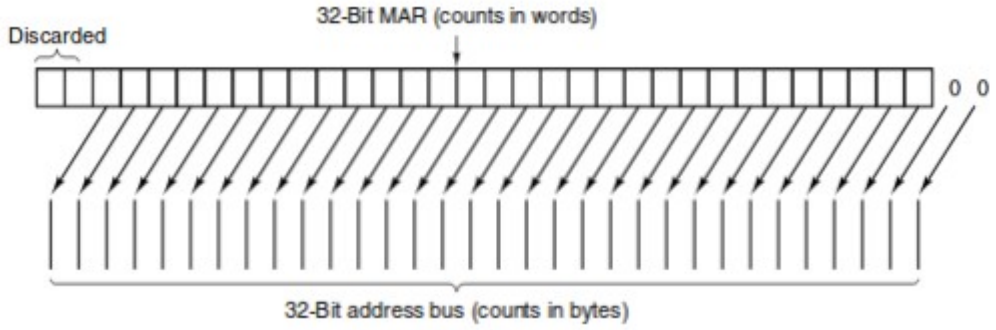

Fig. 1: Conexión del registro MAR al bus de direcciones.

Usando Fig. 1 como referencia, PC se conecta directamente al bus de direcciones ya que cada valor en PC representa 1 Byte de datos; a diferencia de MAR, que se conecta al bus de forma desplazada, osea, el primer bit de MAR se conecta al tercer bit del bus de direcciones. De esta forma el MAR cuenta de a 4 bytes (1 palabra).

MBR

El registro MBR es un registro especial que contiene información sobre la instrucción a ejecutarse y puede ser leído de dos formas al bus B. Una forma es como un valor no signado, adonde los 24-bits superiores del mismo contienen cero, y la otra forma es como un valor signado, adonde se duplica el bit de signo (bit 7) a los 24-bits superiores. Este proceso se llama sign extension (extensión del signo).

Flujo de datos

Las 26 señales de control de nuestro camino de datos pueden ser divididas en 3 grupos.

- 9 señales para escribir datos del bus C a los registros.
- 9 señales para escribir datos de los registros al bus B.

- 6 para controlar el ALU y 2 para el shifter.

Hay dos grupos mas de señales que no se ven y no serán implementadas en este proyecto, pero se mencionaran.

- 2 señales para indicarle a la memoria si es una operación de lectura/escritura a través de MAR/ MDR.
- 1 señal para indicarle a la memoria is es una operación de lectura a través de PC/MBR.

Los valores de estas señales gobiernan como se va a comportar el camino de datos durante un ciclo del reloj. También, para esta arquitectura, el comienzo de nuestro ciclo esta definido como el flanco descendente. Por esta razón en un ciclo típico, los datos de los registros en nuestro camino de datos son propagados al bus B, ALU y finalmente al bus C en la primera mitad de ciclo, ya que al llegar al flanco ascendente del ciclo, los datos del bus C son escritos a los registros apropiados. Esto se ve ejemplificado en Fig. 2.

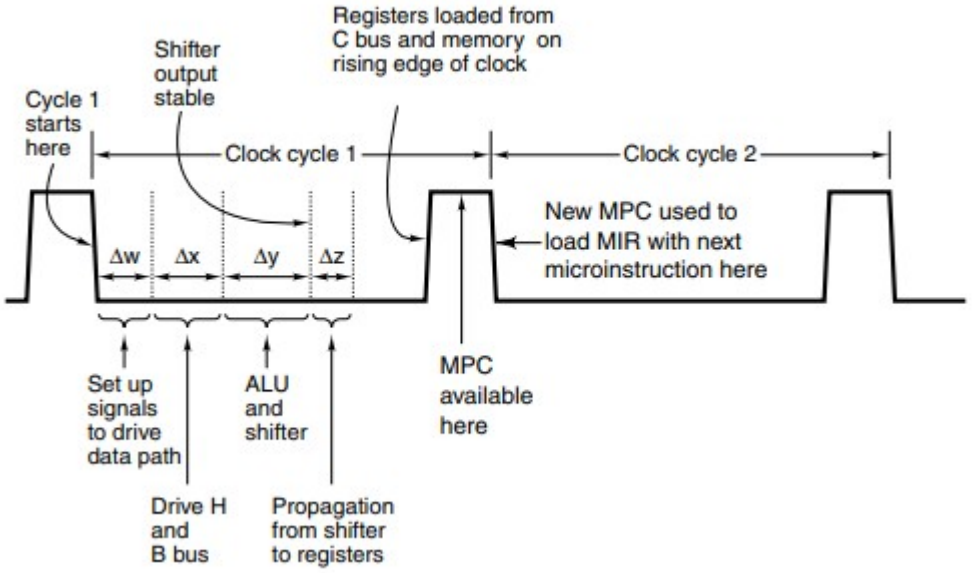

Fig. 2: Diagrama de tiempo de un ciclo en el camino de datos.

Con respecto a operaciones de memoria, se las inicia en la segunda mitad del ciclo, ya que MAR/PC recién contienen un valor valido después del flanco ascendente. También, cuando los datos de memoria están listos, se desperdicia un ciclo para escribirlos a MDR/MBR antes de poder ser usados.

Unidad de Control

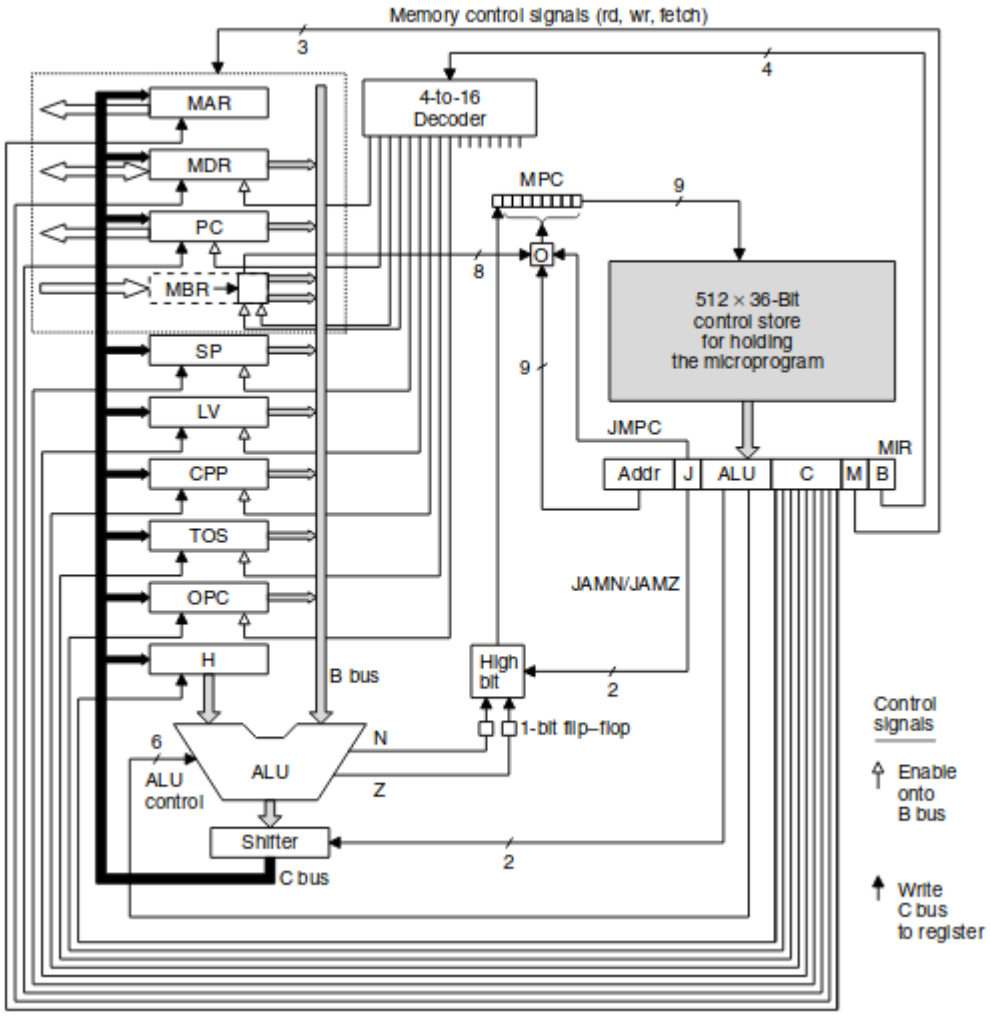

Fig. 3: La microarquitectura completa del MIC-1.

La unidad de control de nuestra MIC-1 es del tipo Microprogrammable (Microprogram). Esto significa que contiene esencialmente un ROM adonde sus direcciones son microinstrucciones definidas para resolver la totalidad de la instrucción cargada en MBR. Específicamente, nuestro ROM es de 512x36-bits.

Los datos saliendo del ROM son guardados en un registro especial de 36-bits, llamado el MIR (MicroInstruction Register). El mismo, se escribe con el flanco descendente del reloj y contiene las señales de control que pueden ser divididas en 2 grupos:

- El estado del camino de datos.
- La dirección de la próxima microinstruccion a ser ejecutada.

El estado del camino de datos es gobernado por las 29 señales que se definieron en la sección previa. Estas 29 señales pueden ser reducidas a 24 señales. Como ya sabemos, mientras se pueden escribir datos del bus C en simultaneo a varios registros, es conveniente solo tener un registro conectado al bus B a la vez. Por esta razón, las 9 señales para controlar el bus B se pueden reducir a 4, si usamos un decoficicador para seleccionar cada una de ellas. En consecuencia, el estado del camino de datos son los primeros 24-bits del MIR.

La dirección de la próxima microinstruccion es gobernada por dos campos en el MIR. Uno de 9-bits que contienen una dirección potencial de la próxima microinstruccion y otro de 3 bits que define como la próxima microinstruccion es seleccionada.

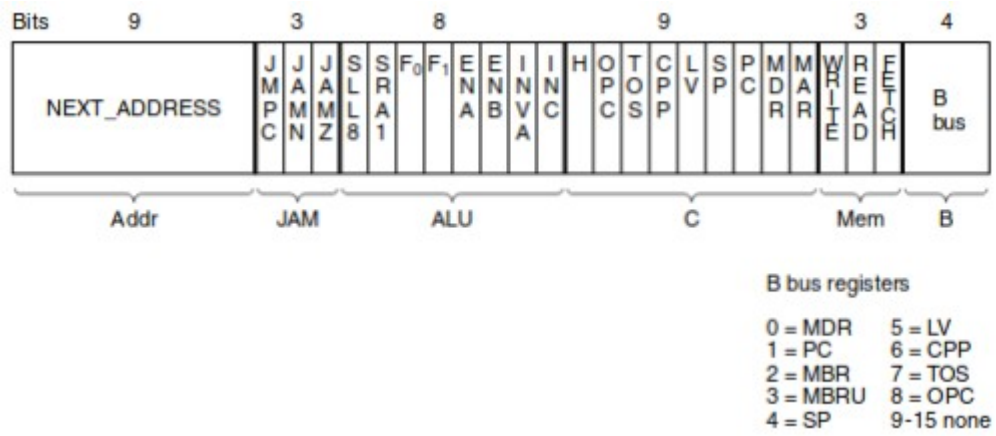

Fig. 4: MIR.

De esta forma se conforman los 36-bits de nuestro MIR y se lo divide en los siguientes campos que pueden ser vistos en Fig. 4.

- Addr – 9-bits – Contienen una dirección potencial de la próxima microinstruccion.
- JAM – 3-bits – Define como se selecciona la próxima microinstruccion.
- ALU – 8-bits – Señales para el ALU y el shifter.
- C – 9-bits – Señales para el bus C.
- Mem – 3-bits – Señales para operaciones de memoria.
- B – 4-bits – Señales para el bus B.

La dirección de la microinstruccion a ser ejecutada esta guardada en el registro MPC (MicroProgram Counter). El libro propone varias maneras de resolver MPC; en este proyecto MPC sera implementado como un registro formado de D-Latch que se escribe en la parte positiva del ciclo.

El valor del MPC puede estar compuesto por el valor de MBR, Addr del MIR, N, y Z, dependiendo de los valores de JAM del MIR. Las dos posibilidades son las siguientes:

- Si JMPC es 1, las funciones booleanas son

$$MPC[7-0]=MBR[7-0]+NEXT\_ADDRESS[7-0]$$

y

$$MPC[8]=NEXT\_ADDRESS[8]$$

- Si JMPC es 0, las funciones booleanas son

$$MPC[7-0]=NEXT\_ADDRESS[7-0]$$

y

$$MPC[8]=(JAMZ \cdot Z)+(JAMN \cdot N)+(NEXT\_ADDRESS[8])$$

Registros N y Z son registros de 1-bit que se escriben en el flanco ascendente. Estos dos registros se conocen como registros de estado. N es 1 cuando el resultado del ALU es negativo, mientras Z es 1 cuando el resultado del ALU es cero. N y Z son de suma importancia cuando estamos buscando hacer saltos con las instrucciones IFEQ offset, IFLT offset y IF_ICMPEQ offset.