

Auxiliar 9

Profesor: Pablo Guerrero.
Auxiliar: Ian Yon
Jueves 6 de agosto de 2015



Problema 1: Problema 2 Control 2 - Primavera 2010

Queremos agregar una instrucción de multiplicación de enteros sin signo **MUL** a M32. Esta utiliza el mismo formato que la instrucción **ADD**:

MUL reg_s, val, reg_d

El operando val puede ser un registro o un valor inmediato de 13 bits. Para implementar esta instrucción se usa un multiplicador en serie, que tiene el siguiente diagrama de tiempo:

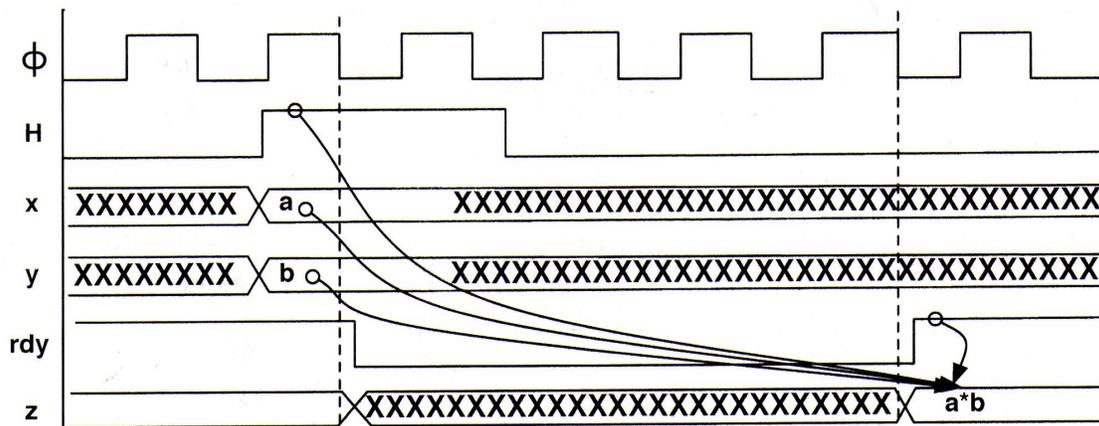


Ilustración 1: Diagrama de tiempo del multiplicador en serie a utilizar

Este circuito entrega un valor de 64 bits, mientras que los buses de M32 son sólo de 32 bits. Cuando el resultado cabe en 32 bits, el valor se almacena en reg_d . Cuando el resultado no cabe en 32 bits, se debe truncar a 32 bits y señalar que hay un overflow.

1. Modifique la arquitectura física de M32 para que la unidad de control sea capaz de realizar **MUL** por medio del circuito dado. haga un dibujo con la nueva arquitectura mostrando los circuitos que agregue o modifique (buses, señales de control, líneas de entrada a la unidad de control, etc).
2. Indique el número de estados que se necesita para poder ejecutar **MUL** y luego, para cada uno de éstos, especifique las señales de control emitidas por la unidad de control. Señale también, cómo se calcula el estado para el próximo ciclo.

Problema 2: Co-procesador para M32

La ALU de M-32 carece de operaciones con números de punto flotante. Para aumentar el poder de cómputo de este procesador, sin modificar su diseño, se provee un co-procesador, que básicamente es una ALU, con operaciones especiales con números en punto flotante, y se trata como un dispositivo de E/S.

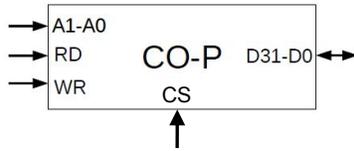
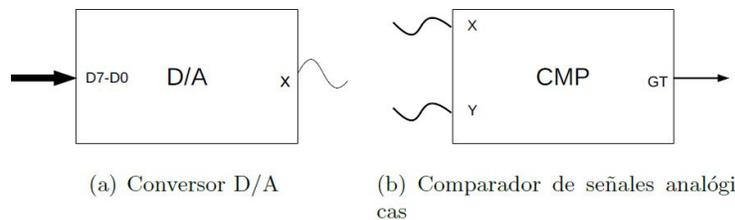


Ilustración 2: Co-procesador para M32

Este co-procesador provee operación de suma, resta, multiplicación y división en punto flotante, y funciona de la siguiente manera. Por D31-D0 se escribe el primer y segundo operando en instantes diferentes. Luego en A1-A0 se pone un código de operación y se pone en 1 la entrada CS, con lo que en D31-D0 aparece el resultado de la operación.

1. Diseñe una interfaz de memoria para este co-procesador. Este debe tener entrada en 0xFFFFC00 y el resultado de las operaciones debe estar entre 0xFFFFF00 , 0xFFFFF04, 0xFFFFF08 y 0xFFFFF0C, respectivamente para suma, resta, multiplicación y división.
2. Haga la función suma(float a, float b) que use el co-procesador para calcular la suma de dos números en punto flotante.

Problema 3



Las figuras (a) y (b) muestran un convertor digital/análogo (D/A) y un comparador (CMP). El convertor digital/análogo recibe como entrada un número entero binario en D7-D0 (entre 0 y 255) y entrega en X una señal analógica consistente en un voltaje entre 0 y 12V proporcional al valor de la entrada. El comparador recibe dos valores analógicos X e Y y los compara entregando un valor binario 1 que indica que $X > Y$ o 0 en caso contrario.

1. Diseñe e implemente una interfaz de entrada/salida para ambas componentes. Considere un microprocesador con bus de datos de 8 bits y un bus de direcciones de 16 bits. Haga que al escribir en la dirección 0xFFFF se establezca la entrada del convertor D/A (preocúpese que esta entrada se mantenga constante hasta que se reescriba un nuevo valor) y al leer en la misma dirección se lea la salida GT de CMP. La señales X e Y salen del sistema para ser conectadas por el usuario.
2. Explique cómo un usuario puede usar estas componentes para convertir una señal analógica entre 0 y 12V en un valor binario entre 0 y 255. Escriba en C la rutina `convertirAD()` que hace esta conversión (busque la simplicidad en esta rutina y no la eficiencia).