



Proyecto #2

Divisor (ALU) y Registros de Memoria

Generalidades

- Fecha de entrega: Viernes, 5 de Noviembre, 2010 @ 4:00pm.
- Modalidad: Grupal (2 personas)

Descripción

Diseñar e implementar el elemento faltante de la Unidad Aritmética-Lógica: un divisor de 6 bits. Además se deberán diseñar e implementar registros de memoria de trabajo y un subsistema de memoria.

Requerimientos

Divisor

El divisor debe permitir la división entre números de 6 bits en complemento a dos (debe poder trabajar con números negativos).

Las líneas de entrada serán:

- Dividendo
- Divisor
- Línea Habilitadora
- Clock (Opcional, solamente si se implementa utilizando técnicas secuenciales)

Las líneas de salida serán:

- Cociente: 6 bits
- Residuo: 6 bits

Las siguientes banderas de salida deben ser incluidas:

- ZF (Zero Flag): Indica si hubo división entre cero (resultado indefinido).
- SF (Sign Flag): Indica si el resultado de la división (cociente) es un número negativo.



Registro de Almacenamiento de 6 bits

Se deberá implementar un registro de trabajo *universal* de 6 bits (dos modos de funcionamiento: Parallel-In/Serial-Out y Serial-In/Parallel-Out).

Las líneas de entrada serán:

- Señal de control lectura/escritura (R/W): R/W=0 para lectura y R/W=1 para escritura.
- Modo de operación: Modo=0 para Parallel-In/Serial-Out y Modo=1 para Serial-In/Parallel-Out.
- Línea habilitadora
- Clock
- Reset: R=1 “resetea” los FlipFlops asignando un valor de 0 a la salida Q.
- Datos: De 6 bits, indica el dato a almacenar en caso de que R/W=1. Si R/W=0 se debe ignorar. Si Modo=0 se utilizan los 6 bits y si Modo=1 el LSB se utiliza como bit único para escritura serial.

Las líneas de salida serán:

- Datos: De 6 bits, indica el dato que se leyó en caso de que R/W=0. Si R/W=1 deben salir ceros. Si Modo=0 el LSB se utiliza como bit único para lectura serial y si Modo=1 se utilizan los 6 bits.

Registro de Trabajo de 6 bits

Se deberá implementar un registro de trabajo de tipo Parallel-In/Parallel-Out de 6 bits con funciones de inversión de los datos y corrimientos bidireccionales.

Las líneas de entrada serán:

- Señal de control lectura/escritura (R/W): R/W=0 para lectura y R/W=1 para escritura.
- Línea habilitadora
- Clock
- Reset: R=1 “resetea” los FlipFlops asignando un valor de 0 a la salida Q.
- Datos: De 6 bits, indica el dato a almacenar en caso de que R/W=1. Si R/W=0 se debe ignorar.
- Función: De 2 bits, si Función=0 no se aplica ninguna operación. Si Función=1 se aplica una inversión de los datos almacenados. Si Función=2 se aplica un corrimiento lógico a la izquierda. Y si Función=3 se aplica un corrimiento aritmético a la derecha. Cada operación es no-volatil, esto quiere decir que si se aplica una operación esta debe alterar el valor de los FlipFlops.

Las líneas de salida serán:



- Datos: De 6 bits, indica el dato que se leyó en caso de que $R/W=0$. Si $R/W=1$ deben salir ceros.

Subsistema de Memoria

Se debe crear una pastilla de memoria RAM para palabras de 6 bits capaz de almacenar un total de 16 palabras.

Las líneas de entrada serán:

- Señal de control lectura/escritura (R/W): $R/W=0$ para lectura y $R/W=1$ para escritura.
- Línea habilitadora
- Clock
- Reset: $R=1$ “resetea” los FlipFlops asignando un valor de 0 a la salida Q.
- Datos: De 6 bits, indica el dato a almacenar en caso de que $R/W=1$. Si $R/W=0$ se debe ignorar.
- Dirección: De 4 bits, indica la dirección de memoria que se quiere leer (en caso de que $R/W=0$) o a la que se quiere escribir (en caso de que $R/W=1$).

Las líneas de salida serán:

- Datos: De 6 bits, indica el dato que se leyó en caso de que $R/W=0$. Si $R/W=1$ deben salir ceros.

Además, y de acuerdo al modelo seguido para el desarrollo de la arquitectura (Von Neumman), es necesario implementar los registros utilizados para acceder la memoria (ya que los otros componentes no pueden accederla directamente). Estos registros son el MAR (Memory Address Register) y el MDR (Memory Data Register).

Como se puede apreciar en la Ilustración 1 ambos registros son la interfaz de la memoria RAM con el resto de la arquitectura. La RAM se conecta al bus de comunicación mediante estos dos registros.

- MAR: Es un registro de memoria simple cuyo trabajo es almacenar la dirección de memoria que se quiere acceder, ya sea para escritura o para lectura. Es del tipo Parallel-In/Parallel-Out y debe contar con las siguientes líneas:
 - Entrada:
 1. Habilitadora
 2. R/W
 3. Palabra de 4 bits que indica la dirección de memoria que se desea acceder
 4. Clock
 5. Reset



– Salida:

1. Palabra de 4 bit con la dirección de memoria que se desea acceder

La salida del MAR debe conectarse al circuito decodificador (decoder de la memoria RAM) que habilitará únicamente la celda de memoria asignada a la dirección almacenada en el MAR.

- MDR: Es un registro de memoria simple (también del tipo Parallel-In/Parallel-Out) cuyo trabajo es almacenar el valor que se debe escribir a memoria o el valor leído de memoria. debe contar con las siguientes líneas:

– Entrada:

1. Habilitadora
2. R/W
3. Palabra de 6 bits que indica el valor a leer/escribir
4. Clock
5. Reset

– Salida:

1. Palabra de 6 bit con el valor a leer/escribir

El MDR funciona como un registro temporal de los datos que entran y salen de la memoria RAM.

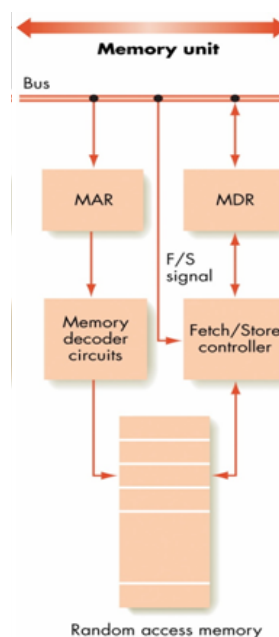


Ilustración 1. Estructura del subsistema de memoria según Von Neumann.



En resumen, las operaciones que debe realizar el subsistema de memoria son únicamente dos: leer y escribir.

- Leer:

1. Se escribe en el MAR la dirección de la celda de memoria (de 0 a 15) que se quiere leer. Para esto se debe habilitar el registro MAR y ponerlo en modo escritura.
2. Se coloca la RAM en modo lectura. El circuito decodificador de la RAM recibe la dirección de la celda a leer mediante las salidas del MAR, este circuito habilita únicamente la celda correspondiente (que debe estar en modo lectura).
3. El registro MDR se coloca en modo escritura y se habilita. Como la memoria RAM está conectado a las entradas del MDR, la salida de la memoria RAM será escrita en el MDR.

- Escribir:

1. Se escribe en el MAR la dirección de la celda de memoria que se quiere escribir. Para esto se habilita el registro MAR y se pone en modo escritura.
2. Se escribe en el MDR el valor que se desea escribir en la memoria RAM. Para esto se habilita el registro y se coloca en modo escritura.
3. Se coloca la RAM en modo escritura. El circuito decodificador transforma la señal de la dirección que recibe del MAR en la línea habilitadora de la celda que se desea escribir. La celda de la RAM que se debe modificar lee el valor del MDR y lo guarda.



Estructura de Archivos

Los circuitos deben entregarse en una carpeta organizada como se muestra a continuación:

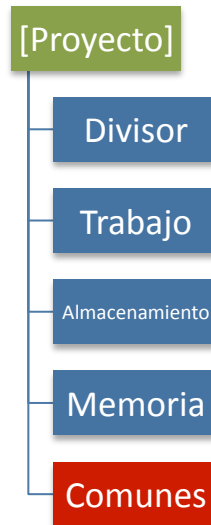


Ilustración 2. Estructura de archivos de entregable.

La carpeta *raíz* (indicada con el nombre *[Proyecto]*) debe ser reemplazada con el nombre de los dos integrantes del equipo siguiendo el formato: *Nombre1Apellido1-Nombre2Apellido2*. Luego deberán existir 4 carpetas dentro de la carpeta *raíz*. Cada una contendrá uno de los 4 circuitos solicitados en el proyecto. En caso de que existan circuitos que necesiten ser reutilizados (e.g. FlipFlops) estos deberán ser incluidos en la carpeta llamada *Comunes*.

Archivo de Prueba

Cada carpeta deberá contener un archivo llamado *TEST*. Este archivo contendrá la *caja negra* del circuito diseñado con las entradas y salidas configuradas para poder realizar pruebas.

Entregable

- Circuitos digitales en forma digital.
- Reporte en forma digital siguiendo el formato dado el primer día de clases.
- La sección de contenido del reporte deberá dividirse en las siguientes secciones:
 - Análisis de resultados: Incluir un resumen de cada uno de los circuitos implementados y si fueron concluidos con éxito o no y por qué.
 - Análisis de la solución: Describir cada circuito implementado, cómo se implementó y cualquier otro detalle de su funcionamiento (e.g. códigos de control).
 - Se debe indicar, para cada circuito, cuál es el tiempo de propagación para cada una de sus operaciones. Por ejemplo, para el registro de almacenamiento se debe indicar cuántos ciclos de reloj le toma al circuito realizar una escritura completa.



- El reporte debe incluir un anexo con un **plan de pruebas**. Donde se muestren las pruebas planeadas y realizadas para asegurar el correcto funcionamiento de los circuitos diseñados.

Criterios de Evaluación

- Circuitos Digitales – 60 pts.
- Documentación – 40 pts.