UNIVERSIDAD DE COSTA RICA
Escuela de Ingeniería Eléctrica
IE0523 – Circuitos Digitales II

Tarea 9

Yeison Rodríguez Pacheco, B56074

08/05/19

Resumen

En esta tarea se utilizará una secuencia de pipelines para realizar un sumador por partes.

1. Contabilización del tiempo

Tabla 1: Contabilización del tiempo

Sesiones	Sesión 1
Búsqueda y estudio de información	10 min
Mejora del Makefile	0min
Programación del código	1h
Ejecución y pruebas	10 min
Reporte	40 min

2. Descripción arquitectónica del circuito

Este circuito consiste en un sumador por partes con etapa de pipeline. La razón de crear un circuito con estas características es que en ocasiones si los buses de datos de la suma son muy altos, puede ocurrir que la lógica combinacional tarde demasiado tiempo y por esto podrían ocurrir problemas en tiempos de setup y hold, por lo que se opta por realizar el calculo por partes.

En la figura 1 se presenta la arquitectura del circuito, notamos que existen dos lineas de pipeline y además que tanto el bus de datos de entrada como el de salida es de 4 bits **por lo que puede ocurrir overflow en la salida final**, pero así está el diseño especificado en el enunciado de la tarea por lo que no se modificará.

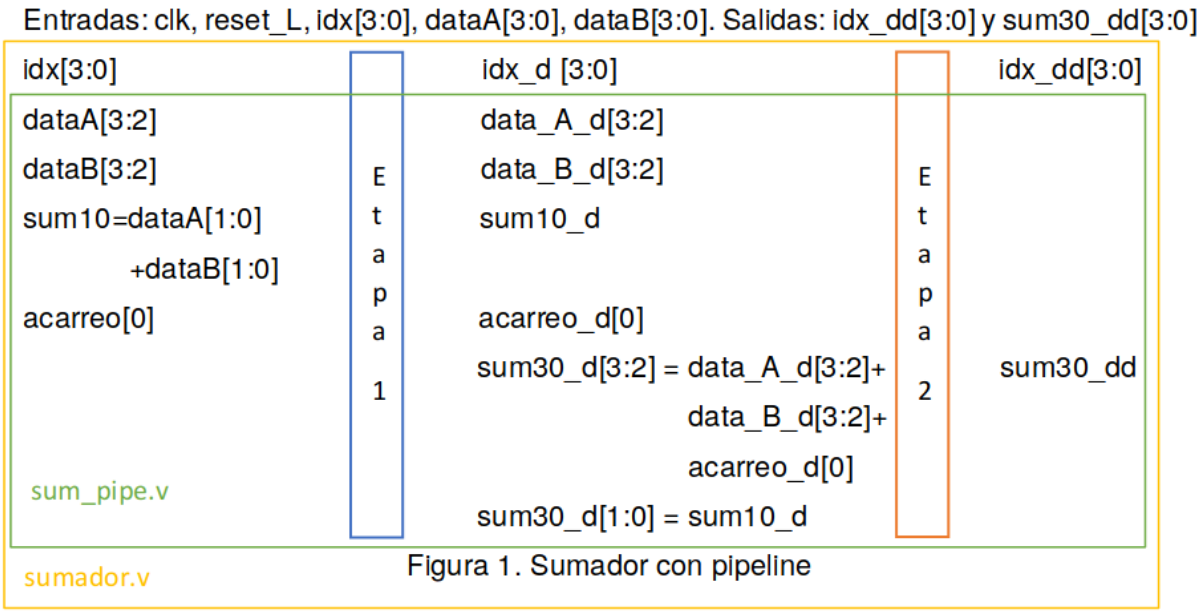

Figura 1: Arquitectura del circuito a diseñar

3. Plan de pruebas

Las pruebas consisten en probar el funcionamiento del circuito que calcula la suma de los números agregando variaciones a las entradas con el fin de verificar que la suma se hizo correctamente en el

tiempo establecido.

4. Instrucciones de utilización de la simulación

Para correr la prueba simplemente ejecutar el comando

```
make
```

Inmediatamente se presentarán las simulaciones en GTKwave y se podrá acceder a las señales del modelo estructural y conductual.

En la tabla 2 se dejará una lista detallada de comandos del makefile, muchos de estos solo se programaron para la creación del laboratorio, y no para las pruebas en sí.

Tabla 2: Comandos Makefile

Comandos del Makefile para pruebas (usuario)	
make	Compila y ejecuta la prueba estandar, también abre el GTKwave con las simulaciones
make gtk	Ejecuta nuevamente el gtkWave, por si necesita abrirlo de nuevo sin compilar
make prueba1	Ejecuta el archivo prueba2
make pdf	Abre el pdf del informe
Comandos de Makefile para facilitar generación de código (diseñador)	
make generar_archivos	Crea los archivos conductual, probador y banco de prueba según variables de los nombres deseados especificadas en el Makefile (Comando parametrizado)
make generar_archivo_yosys	Genera el archivo con los comandos de Yosys con respecto a los nombres elegidos por el diseñador en las variables del Makefile (Comando parametrizado)
make llenarBancoPrueba	Este comando hace los include e instancias necesarias para que el banco de pruebas funcione correctamente. (Comando parametrizado)
make ejecutar_yosys	Este comando ejecuta el archivo .ys y además se encarga de cambiar el nombre del módulo generado por la síntesis, y hace automaticamente el include del archivo que contiene la librería de compuertas a utilizar (Comando parametrizado)

5. Discusión y Análisis de resultados

5.1. Sumador

En la figura 2 vemos la respuesta estructural y conductual del sumador con pipeline. Vemos que en un inicio cuando se levanta reset el index es 1 y los datos de entrada son 0 y 1, por lo que la suma es 1 y se presenta hasta dos ciclos después (por el pipeline). También vemos que el index es 1 en el comienzo, y el index_dd es también de 1 después de dos ciclos, por lo que el circuito cumple con lo especificado.

En las siguientes iteraciones se puede ver que tanto el conductual como el estructural realizan las sumas correctamente.

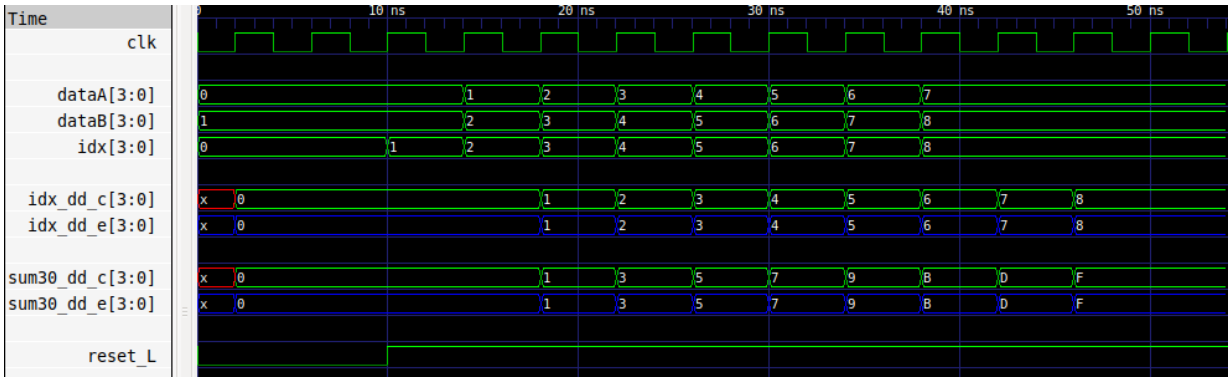

Figura 2: Respuesta conductual y estructural del circuito sumador con pipeline. Hecho en GTKwave

Es importante recalcar que **si se ingresan datos a la entrada cuya suma sea mayor a 16, la suma de resultado no va a ser la correcta ya que el bus de salida es de 4 bits, pero así estaba en el diseño del enunciado de la tarea, por lo que se respetó de esta forma.**

6. Conclusión

- Fue posible demostrar que la implementación de los pipeline es sencilla y útil para mejorar la eficiencia de los circuitos y evitar posibles problemas de setup y hold.