

Pipelining

Gabriel Fernando Araya Mora B80525

Escuela de Ingeniería Eléctrica, Universidad de Costa Rica

Circuitos Digitales II (IE-0523)

1. Tiempo de ejecución de la tarea:

- **Buscar información:** Siempre antes de empezar cualquier trabajo es importante tomarse un tiempo prudente para buscar información e investigar acerca del tema. Esto es especialmente cierto cuando se trata de programar. Para la búsqueda de información, se revisó la documentación, y la teoría detrás del pipelining y se combinó con lo que ya se sabía de sumadores. La búsqueda de información me llevó una hora.
- **Usando la información:** Encontrados varios ejemplos en la web de cómo implementar un sumador con pipelining, se comparó para saber cual se ajustaba más al dado por el profesor en el enunciado de la tarea. Hecho esto entonces se modifica el código conductual encontrado para que cumpla con todos los requisitos para la tarea. La elaboración del código para esta tarea me tomó al rededor de 2 horas.
- **Elaboración del reporte:** La elaboración del reporte me tomó aproximadamente una hora.
- **Total:** De lo anterior se saca que el tiempo total para la realización de la tarea es de 4h.

2. Descripción Arquitectónica del diseño

Para esta última tarea, se pretende realizar un sumador con pipeline que reciba en la entrada dos datos de 4 bits, además en la primera etapa suma los dos bits menos significativos de cada dato y aparte tome en cuenta si existe un acarreo o no, en el siguiente ciclo se suman los dos bits más significativos de cada dato junto al acarreo anteriormente calculado, este resultado serán los dos bits más significativos del resultado final y los dos bits menos significativos de la suma de la primera etapa serán los dos bits menos significativos del resultado final.

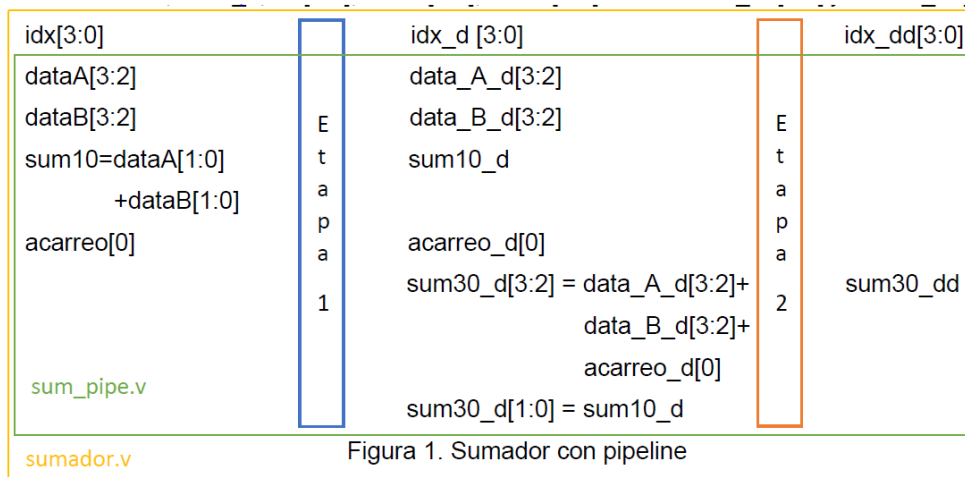


Figura 1: Descripción arquitectónica del circuito.

3. Plan de Pruebas

Al implementar el módulo del sumador se le da un estímulo de 4 bits, y se espera que en la primera fase tener un resultado de la suma de los dos bits menos significativos y también de este obtener el acarreo de esta suma, seguido a esto para la segunda etapa se espera sumar el acarreo que se obtuvo como resultado en la etapa anterior junto a los dos bits más significativos de los datos de entrada, con esto se obtiene la salida final del sumador.

Es de esperar que después de dos ciclos de reloj se obtenga la salida del sumador, esto debido a las dos etapas de flip-flops.

4. Instrucciones de uso de la aplicación

Para la utilización de la tarea, basta con estar dentro de la carpeta de código fuente, abrir una terminal en ella y escribir la palabra **make**, esto abrirá el gtkwave.

```
all:

yosys -s sumador.ys
sed -i "s/sumador/sumador_synth/g" sumador_synth.v
sed -i "s/sum_30_dd/sum_30_dd_synth/g" sumador_synth.v
sed -i "s/idx_dd/idx_dd_synth/g" sumador_synth.v
sed -i "s/sum_pipe/sum_pipe_synth/g" sumador_synth.v
sed -i "s/identificador/identificador_synth/g" sumador_synth.v

iverilog banco.v
./a.out

rm a.out

gtkwave tarea9.vcd
```

5. Resultados



Figura 2: Resultados obtenidos para el sumador.

6. Análisis de los resultados

Analizando los resultados obtenidos en la figura #3. Como se puede ver en la figura anterior cuando se tiene en la entrada una señal de *reset_L*, esto ya que el reset apaga toda la circuitería. Cuando la señal de *reset_L* se pone en alto, se ve que comienzan a llegar los datos *data_A* y *data_B* a la circuitería del sumador, un ciclo después de la primera llegada de datos, se observa donde se genera la suma de los dos bits menos significativos de los datos de entrada, además se puede notar que para el cuarto flanco es donde tenemos el resultado dado como *sum_10* y además que el acarreo de esta primera está dado por la señal acarreo el cual es el bit más significativo de esta suma.

Para la segunda etapa, se observa donde los valores *data_A_d* y *data_B_d* corresponden a los dos bits más significativos, así como el *acarreo_d* representa el acarreo anteriormente obtenido en la primera suma. Además de que el valor de *sum_10_d* es el valor de los dos bits menos significativos de la primera suma. Por último se ve reflejado en la señal *sum_30_dd*, la cual representa el valor final del sumador un ciclo del reloj después de la segunda etapa, este valor final del sumador corresponde en sus dos bits menos significativos a el valor de *sum_10_d* y en los dos bits más significativos corresponde a la suma de *data_A_d*, *data_B_d* y el *acarreo_d*, con esto se ve reflejado un correcto comportamiento del sumador tanto conductual como el módulo sintetizado donde su salida está dada por *sum_30_dd_synth*.

También se nota el correcto comportamiento de los identificadores y su correcta propagación de la misma forma que el sumador, donde se observa que cuando el *reset_L* se activa, comienza a entrar una señal por *idx*, la cual al pasar por las dos etapas de flip-flops de la misma forma que lo hacen los datos por el sumador, va a reflejar en la salida *idx_dd* el resultado dos ciclos de reloj después, de la misma forma que con el sumador se puede comprobar su correcto funcionamiento estructural y conductual donde la salida del módulo sintetizado está dada por *idx_dd_synth*.