

Práctica 2

Sumador de 4 bits y Sumador SIMD

.....

Nombre y Apellidos	
Nombre y Apellidos	

Número de grupo de laboratorio	
--------------------------------	--

Observación: en el enunciado de la práctica, los apartados etiquetados como Trabajo deben realizarse aunque no hay que entregarlos.

Preguntas

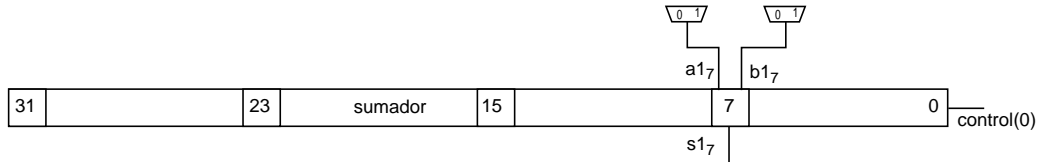
Utilice la descripción estructural del sumador de 4 bits con propagación serie del acarreo descrito en la práctica. **Referencias:** “Esquema de un circuito sumador de 4 bits” en la página 54, “Esquema de puertas lógicas de un sumador de 1 bit.” en la página 20.

- 1 Entregue una traza de la ventana de tiempo donde observen consecutivamente 6 casos. En los 2 primeros el retardo debe ser máximo. Seguidamente debe observarse dos casos donde el retardo sea mínimo. Por último, debe observarse otros dos casos donde el retardo no sea ni máximo ni mínimo. Muestre también las sentencias concurrentes utilizadas para generar los frentes de onda de las señales de entrada del sumador (“Programas de prueba en VHDL (“testbench”)” en la página 72).

Las siguientes preguntas son relativas al sumador SIMD. **Referencia:** “Sumador SIMD” en la página 77. Las 2 primeras preguntas se centran en la operación de suma (página 79).

- 2 Muestre un esquema, con puertas lógicas (de 2 entradas), multiplexores (de 2 entradas de datos) y el sumador como caja negra, que permita realizar la operación suma SIMD de datos de 8 bytes. El número de puertas y multiplexores utilizado debe ser el mínimo

imprescindible.



- 3** Implemente el esquema de la respuesta a la pregunta anterior en VHDL. Muestre los conjuntos de sentencias de asignación concurrente y modifique los ficheros `simd_ent.dvw` y `simd_sal.dvw`. Compruebe el diseño mediante los ficheros `sumaSIMD.tsv` y `suma32.tsv`. Para especificar un multiplexor utilice sentencias de asignación de señal con selección. **Referencia:** “Selección en la asignación de una señal” en la página 93.

Modificación de las entradas (simd_ent.dwv)

Modificación de la salida (simd_sal.dvv)

En las siguientes preguntas se completa el diseño de la alu añadiendo la operación de resta (página 80).

- 4 Proponga una forma de modificar los vectores de bits de entrada para que se genere un acarreo de entrada en la suma de los bits con ponderación 8, 16 y 24 respectivamente. Especifique las expresiones algebraicas teniendo en cuenta que debe mantenerse la funcionalidad de sumas SIMD. En la salida del sumador, indique la información que suministran los bits con ponderación 7, 15 y 23. En otra pregunta posterior se solicita la corrección de los mismos bits en la salida del sumador.

Expresiones

$$a1_{8 \times i + 7} =$$

$$b1_{8 \times i + 7} =$$

donde $0 \leq i < 3$ indica el número de byte

- 5 Los bits cuyo valor debe corregirse en la salida del sumador son 7, 15 y 23. Proponga la forma de corregir el bit más significativo de cada byte. La propuesta debe mantener la funcionalidad de efectuar sumas SIMD y operaciones en 32 bits. Especifique las expresiones algebraicas.

Corrección cuando la operación es SIMD

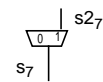
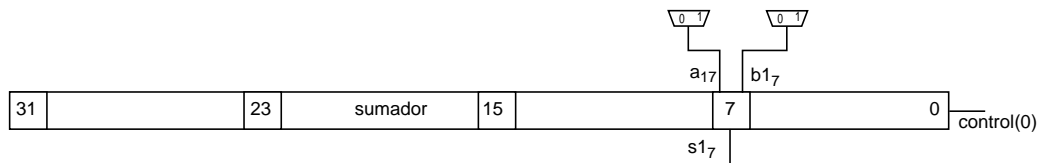
$$s_{2 \times i + 7} =$$

donde $0 \leq i < 3$ indica el número de byte

Mantener la funcionalidad de suma algebraica de 32 bits

$$s_{8 \times i + 7} =$$

- 6 Muestre un esquema, con puertas lógicas (de 2 entradas), multiplexores (de 2 entradas de datos) y el sumador como caja negra, que permita realizar las operaciones de suma y resta SIMD de datos de 8 bytes. El número de puertas y multiplexores utilizados debe ser el mínimo imprescindible.



- 7 Implemente el esquema de la respuesta a la pregunta anterior en VHDL. Muestre los conjuntos de sentencias de asignación concurrente y modifique de nuevo los ficheros `simd_ent.dvw` y `simd_sal.dvw` donde ha implementado la suma SIMD. Compruebe el diseño mediante los ficheros `restaSIMD.tsv` y `resta32.tsv`.

Modificación de las entradas (simd_ent.dvw)

Modificación de la salida (simd_sal.dvw)

