

ESTRUCTURA DE COMPUTADORES. UNIDAD 2.- LA UNIDAD ARITMÉTICO LÓGICA (ALU)

2.1. Demostrar por medio de una tabla de verdad, que la expresión lógica $C_n \oplus C_{n-1}$ genera una señal denominada bandera de desbordamiento (*overflow*, V), para identificar errores que se generan al sumar números enteros con signo en codificación complemento a 2.

2.2. Además del resultado, la ALU genera un conjunto de bits que pueden ser utilizados por el sistema o por los usuarios para el control de las operaciones aritmético-lógicas desarrolladas. Entre estos los más conocidos son el bit o bandera de signo (N, N = '1' si el resultado es negativo), el bit o bandera de cero (Z, Z = '1' si el resultado es cero), el bit o bandera de acarreo (C, C = '1' si hay acarreo en la operación de suma entre los bits más significativos de ambos operandos), y el bit o bandera de overflow o rebosamiento (V) (V = '1' si se supera la capacidad de representación del sistema). Utilizando números binarios de 8 bits con signo y representados en complemento a 2, realice las operaciones señaladas con dos operandos en decimal y compruebe en cada caso, el valor de estos cuatro bits, N, Z, C y V, señale en cada caso su significado.

- a) $46 + 67$ b) $112 - 89$ c) $75 + 95$ d) $-34 - 97$

2.3. Utilizando números binarios de 8 bits con signo y representados en complemento a 2, realice en el orden señalado por los paréntesis las operaciones indicadas. Para cada resultado parcial, compruebe si se produce un desbordamiento aritmético y calcule también la validez del resultado final. Analice los resultados obtenidos en relación con el valor de la bandera V del problema 2.1 y observe que, resultados parciales incorrectos, no suponen necesariamente que el resultado final lo sea.

- a) $(((((32 + 100) + 70) + 24) - 62) - 50)$. b) $(((((43 - 12) + 34) + 75) - 47)$ c) $(((((15 - 77) - 43) - 38) + 32)$

2.4. Multiplicar las parejas de números con signo en complemento a 2 siguientes:

- a) $010111_2 \times 110110_2$ b) $110011_2 \times 101100_2$ c) $0011111_2 \times 0011111_2$
d) $56_{10} \times (-67)_{10}$ e) $-43_{10} \times (-113)_{10}$ f) $A4_{16} \times B8_{16}$.

Nota: Aplicando la propiedad de extensión de signo del complemento a 2, utilizar en todos los casos 8 bits para cada multiplicando.

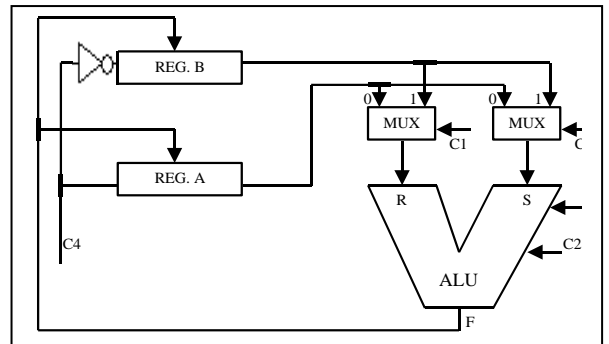
2.5. Una ALU trabaja con números enteros con signo de 4 bits, en notación complemento a 2. Se pide, escribir una tabla donde se muestre el resultado de la división de cada uno de estos número por -2 $\{R = A/(-2)\}$.

2.6. Considerar la ALU y los registros que se muestran en la figura. Responder a las siguientes cuestiones escribiendo la(s) palabra(s) de control adecuada(s). Cada palabra de control debe especificarse de acuerdo con el formato $C_4C_3C_2C_1C_0$. Por ejemplo, la operación "A+B→A" se escribiría 10001.

a) Sugerir dos métodos para llevar el registro A al valor cero.

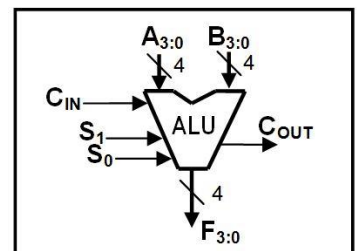
b) Sugerir una secuencia de control que intercambie los contenidos de los registros A y B. La interpretación de los distintos puntos de control se resume en la tabla adjunta.

C_1C_0	→R	→S	C_3C_2	F	C_4	ACCION
00	A	A	00	R + S	0	F → B
01	A	B	01	R - S	1	F → A
10	B	A	10	R AND S		
11	B	B	11	R XOR S		



2.7. Utilizando la ALU de la figura, indique las operaciones a realizar en la ALU para que las salidas representen el módulo del resultado de la diferencia (X - Y), en donde X e Y son números positivos de 4 bits en complemento a 2.

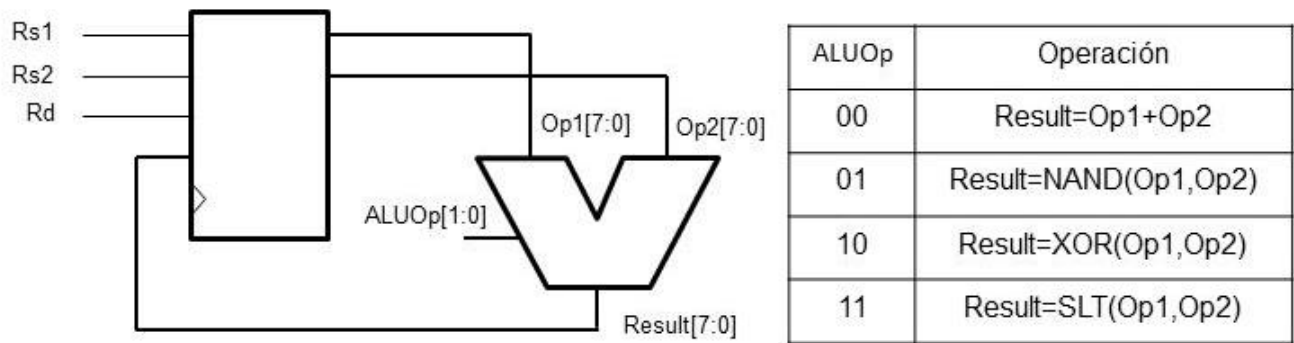
$S_1 S_0$	Operación	$S_1 S_0$	Operación
0 0	$F = \text{AND}(A, B)$	0 1	$F = /A + C_{IN}$
1 0	$F = A + B + C_{IN}$	1 1	$F = A + /B + C_{IN}$



2.8. Diseñar en VHDL un circuito *Barrel Shift*, para números de 32 bits y un máximo de 31 desplazamientos, tanto a la izquierda como a la derecha.

2.9. Se tiene la arquitectura de la figura con un banco de dos registros (R0 y R1) de 8 bits y una ALU de 8 bits que realiza cuatro operaciones (ver tabla). La señal de control ALUOp, de 2 bits, sirve para elegir la operación a realizar. Las señales de control Rs1, Rs2 y Rd sirven para indicar cuáles son los 2 registros utilizados en la ALU (Rs1 para Op1 y Rs2 para Op2) y cuál es el registro escrito (Rd para Result). En todas ellas, '0' indica el registro R0 y '1' indica el registro R1. La operación SLT pone la salida a 1 (número entero 1) si $Op1 < Op2$ y a 0 (número entero 0) en caso contrario, considerando que los operandos tienen signo y están en complemento a 2.

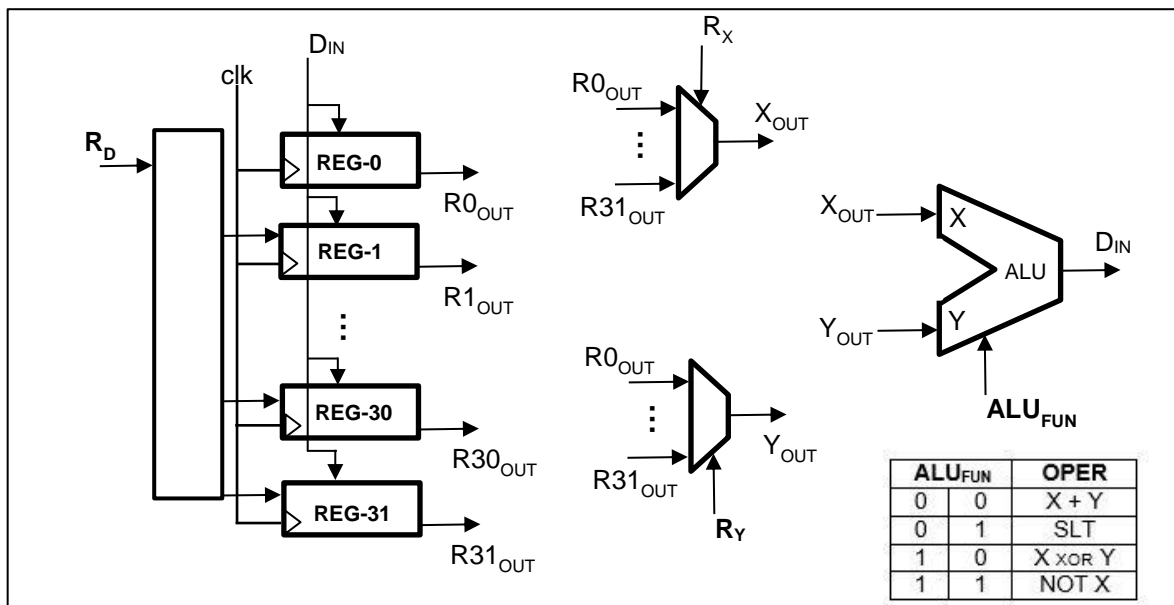
ESTRUCTURA DE COMPUTADORES. UNIDAD 2.- LA UNIDAD ARITMÉTICO LÓGICA (ALU)



La palabra de control es de 5 bits, (ALUOp, Rs1, Rs2, Rd). Por ejemplo, si la palabra de control es "00011" se realizará la operación $R1 \leq R0 + R1$;

- Indique la palabra de control para conseguir que R0 quede a 0
- Indique la palabra de control para conseguir que R1 reciba NOT(R0)
- Si inicialmente $R0=0x08$ y $R1=0xFF$, indique a qué queda el registro R1 tras realizar la siguiente secuencia de instrucciones: 1. "00010" y 2. "11101"
- Si inicialmente $R0=0xFE$ y $R1=0x05$, indique a qué queda el registro R1 tras realizar la siguiente secuencia de instrucciones: 1. "11011" y 2. "10011"

2.10. En el circuito de la figura se muestra la arquitectura de un cierto sistema digital en el que se distinguen elementos combinacionales ya conocidos como multiplexores, decodificadores y una unidad aritmético lógica (ALU), y elementos secuenciales como un conjunto de 32 registros, en el que como es habitual, el registro R0 es de sólo lectura y su valor es siempre 0. Considere que todos los registros son de 32 bits y salvo R0 con valores desconocidos. En la figura se identifican cuatro señales de control con distinto tamaño en bits, R_D , R_X , R_Y y ALU_{FUN} .



La operación SLT de la ALU, como se ha visto en clase, pone un '1' en el registro destino si el valor de la entrada X es menor que el de la entrada Y.

A la vista del esquema facilitado, se pide:

- Señale, justificando necesariamente la respuesta, el tamaño en bits y la función de la señal de control " R_D ".
- Señale, justificando necesariamente la respuesta, el tamaño en bits y la función de la señal de control " R_X ".
- Describa un algoritmo como desee y asócielo a la palabra o palabras de control correspondientes para su ejecución para conseguir que $R_5 \leq 0xFFFFFFFF$.
- Diseñe un algoritmo y escriba la palabra o palabras de control necesarias para su ejecución para calcular la operación $R_4 \leq \text{Complemento a 2 de } R_3$.

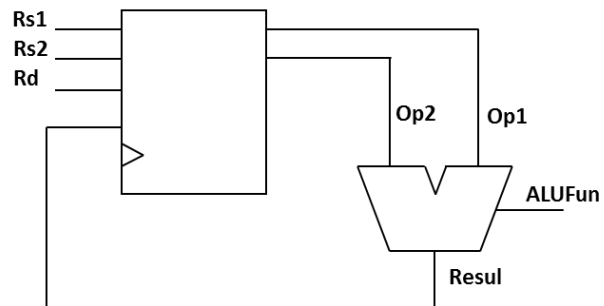
NOTA: Para considerar válida una palabra de control, debe señalar la secuencia de bits que corresponda en el orden de las variables R_D , R_X , R_Y , ALU_{FUN} . Utilice una coma para separar cada señal.

ESTRUCTURA DE COMPUTADORES. UNIDAD 2.- LA UNIDAD ARITMÉTICO LÓGICA (ALU)

2.11. En el circuito de la figura se muestra la arquitectura básica de un sistema microprocesador, con un banco de 8 registros de 16 bits y una ALU capaz de realizar 8 operaciones distintas. Los 8 registros se denominan R0 a R7. El registro R0 es constante e igual a 0, el registro R1 también es constante e igual a 1, y el registro R2 también es constante e igual a 2. El resto de registros, R3 a R7, son variables y de propósito general.

Los bits de control Rs1 y Rs2 indican, respectivamente, qué registros irán a Op1 y Op2 de la ALU, mientras que los bits de control Rd indican qué registro recibirá el valor de la salida de la ALU, Resul. Por su parte, la operación realizada en la ALU se decide mediante los bits de control ALUFunc. En el caso de los desplazamientos, Op1 es el operando desplazado y Op2 indica en cuántas posiciones hay que desplazarlo.

ALUFunc	Operación
000	Resul = Op1 + Op2
001	Resul = Op1 - Op2
010	Resul = Op1 NOR Op2
011	Resul = Op1 XOR Op2
100	Resul = Op1 AND Op2
101	Resul = Op1 SLL Op2
110	Resul = Op1 SRL Op2
111	Resul = Op1 SRA Op2



Se pide:

- Dadas las instrucciones en código máquina de la tabla adjunta, se pide, traducir este código a la "Instrucción ensamblador" correspondiente y calcular el valor de los registros destinos del código dado. El estado inicial de los registros es desconocido, pero las instrucciones indicadas se ejecutan en el orden señalado.
- Sin tener en cuenta los valores de los registros de las operaciones anteriores, ahora se quiere realizar la operación $R4 = 7 \cdot R3$. Al no existir dicha operación en este micro, se realizará a través de una serie de instrucciones. Se pide el código ensamblador así como el código máquina que realizan la operación solicitada. Se valorará, por este orden, utilizar el mínimo número de instrucciones posible y no modificar el contenido de los registros no implicados en la operación (o el mínimo número de registros).

ALUFunc, Rd, Rs1, Rs2
001, 011, 000, 001
011, 100, 001, 010
101, 101, 001, 010
110, 110, 011, 010
111, 111, 011, 010