

EJERCICIO Nº 3

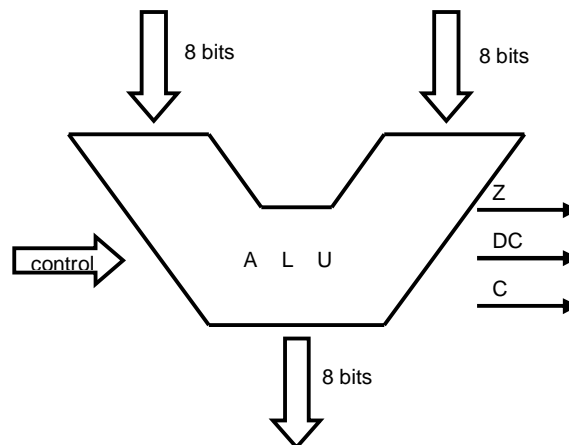
En este ejercicio se va a realizar el módulo del proyecto final con mayor carga combinacional: la ALU. La mayor parte de los módulos se realizarán con las facilidades de compilación que proporciona el programa Microwind, con lo que la dificultad del ejercicio radicará en la integración de todos los elementos en un trazado *lo más pequeño y compacto posible*. Aún así, se valorará muy positivamente la implementación de cualquiera de los módulos sin emplear el compilador; en este caso se da total libertad para escoger el tipo de lógica que se considere adecuada, siempre que se razone.

Observaciones:

- Para el cálculo de la potencia, utilice una frecuencia que permita a los datos recorrer el camino crítico.
- Para el cálculo de áreas, en el programa Microwind, haga click en File → Properties y obtendrá la superficie en  $\mu\text{m}^2$ . Siempre trabaje con  $\mu\text{m}$  y  $\mu\text{m}^2$ , los datos en lambdas no se considerarán válidos.
- Para el cálculo del retardo en el caso peor, justifique cómo ha seleccionado este caso.
- No se olvide de seleccionar la tecnología de  $0.12\mu\text{m}$  y de polarizar el sustrato p y el pozo n.

Este ejercicio se realizará íntegramente en los grupos de trabajo del proyecto final.

La estructura de la ALU es la siguiente:



Las operaciones y el control de la unidad, están descritos en la tabla que se encuentra al final de este enunciado, extraída del proyecto final.

**1(a)** Implemente y simule las siguientes operaciones de 8 bits, utilice el compilador del programa si lo considera necesario:

- Suma
- AND
- Complemento
- Decremento
- Incremento
- OR
- Resta
- Bit Clear
- Bit Set

**1(b)** Caracterice cada una de las operaciones que ha implementado, proporcionando el área, el retardo para el peor caso y el consumo de potencia. Tenga en cuenta que estas medidas se han de realizar con las salidas de las operaciones cargadas a 5fF.  
Nota: para la medida del retardo en el peor caso considere todas las transiciones posibles de los datos de entrada.

**2(a)** Diseñe y realice “a mano” el trazado de una puerta XOR de dos entradas. Trate de minimizar tanto el área como el retardo de la operación (considere que tiene una carga a su salida de 5 fF). Puede realizarlo utilizando cualquier diseño disponible en la literatura o visto en clase.

**2(b)** Repita el apartado 1(b) para la operación que ha diseñado en el apartado 2(a).

**2(c)** Calcule la métrica  $A \times t_p^2$ , el producto del área del diseño por el cuadrado del retardo en el peor de los casos. Cuanto más baja sea esta métrica, mejor se considerará el diseño.

**3(a)** Realice el trazado de la ALU completa, para ello siga las siguientes recomendaciones:

- Trate de conseguir un plano equilibrado, ni muy alargado, ni muy estirado.
- Trate de conseguir un trazado lo más compacto posible, minimizando al máximo los espacios vacíos.
- Planifique previamente cómo se van a colocar los distintos módulos dentro del plano global, teniendo en cuenta sus tamaños y formas.
- Trate de hacer las interconexiones lo más cortas posibles.
- En una versión simplificada que se va a implementar, la información de control que llega a la ALU son los siete bits más significativos del código de operación. ¿Se le ocurre alguna manera de mejorar este diseño?

**3(b)** Caracterice la ALU. Establezca el área total, el área ocupada (porcentual aproximada), el área vacía (porcentual aproximada), el camino crítico, el consumo de potencia y  $A \times t_p^2$ , explicando cómo ha realizado los cálculos. Considere que la ALU está cargada en sus salidas con 50fF.

Para realizar la entrega genere un fichero comprimido que incluya la ficha de resultados en formato excel y todos los trazados que se piden y envíelo a [marisa@die.upm.es](mailto:marisa@die.upm.es) y a [pituero@die.upm.es](mailto:pituero@die.upm.es) con el asunto ME\_ejercicio3\_grupo (tú número de grupo). Nombra el fichero comprimido únicamente con grupo\_(tu número de grupo) y la extensión correspondiente (zip, rar, etc.).

Nemónico, operandos	Descripción	Código operación	Reg. Estado
<b>Operaciones a nivel de byte en el RF</b>			
ADDWF f, d	Suma W y f	00 0111 dfff ffff	C,DC,Z
ANDWF f, d	AND W con f	00 0101 dfff ffff	Z
CLRF f	Pone a 0 f	00 0001 lfff ffff	Z
CLRW -	Pone a 0 W	00 0001 0xxx xxxx	Z
COMF f, d	Complemento de f*	00 1001 dfff ffff	Z
DECF f, d	Decrementa f	00 0011 dfff ffff	Z
DECFSZ f, d	Decrementa f, salta si 0*	00 1011 dfff ffff	
INCF f, d	Incrementa f	00 1010 dfff ffff	Z
INCFSZ f, d	Incrementa f, salta si 0*	00 1111 dfff ffff	
IORWF f, d	OR W con f	00 0100 dfff ffff	Z
MOVF f, d	Mueve f	00 1000 dfff ffff	Z
MOVWF f	Mueve W to f	00 0000 lfff ffff	
NOP -	No Operación	00 0000 0xx0 0000	
RLF f, d	Rota Izq. f a través de Carry*	00 1101 dfff ffff	C
RRF f, d	Rota Drcha. f a través de Carry*	00 1100 dfff ffff	C
SUBWF f, d	Resta W de f	00 0010 dfff ffff	C,DC,Z
SWAPF f, d	Intercambia nibbles en f	00 1110 dfff ffff	
XORWF f, d	OR Exclusivo W con f*	00 0110 dfff ffff	Z
<b>Operaciones a nivel de bit en el RF</b>			
BCF f, b	Bit Clear f	01 00bb bfff ffff	
BSF f, b	Bit Set f	01 01bb bfff ffff	
BTFSC f, b	Bit Test f, Skip if Clear*	01 10bb bfff ffff	
BTFSS f, b	Bit Test f, Skip if Set*	01 11bb bfff ffff	
<b>Operaciones de control y literales</b>			
ADDLW k	Suma literal y W	11 111x kkkk kkkk	C,DC,Z
ANDLW k	AND literal y W	11 1001 kkkk kkkk	Z
CALL k	Llama subrutina**	10 0kkk kkkk kkkk	
CLRWDAT -	Clear Watchdog Timer*	00 0000 0110 0100	TO,PD
GOTO k	Ir a dirección**	10 1kkk kkkk kkkk	
IORLW k	OR literal con W*	11 1000 kkkk kkkk	Z
MOVLW k	Mueve literal a W	11 00xx kkkk kkkk	
RETFIE -	Retorno de interrupción**	00 0000 0000 1001	
RETLW k	Retorno con literal en W**	11 01xx kkkk kkkk	
RETURN -	Retorno de Subrutina**	00 0000 0000 1000	
SLEEP -	Pasa a modo standby	00 0000 0110 0011	TO,PD
SUBLW k	Resta W de un literal	11 110x kkkk kkkk	C,DC,Z
XORLW k	OR Exclusivo de literal con W*	11 1010 kkkk kkkk	Z

Tabla 1. Conjunto de instrucciones. \*No se van a implementar. \*\*Instrucciones que requieren dos ciclos de reloj