



EXAMEN PARCIAL 2

ALU



14 DE ABRIL DE 2021
SERGIO ALEXANDER SANCHEZ GARCIA
Unicaribe

Estado del arte

En computación, la unidad aritmética lógica o unidad aritmético-lógica, también conocida como ALU (siglas en inglés de arithmetic logic unit), es un circuito digital que realiza operaciones aritméticas (suma, resta) y operaciones lógicas (SI, Y, O, NO) entre los valores de los argumentos (uno o dos)

Por mucho, los circuitos electrónicos más complejos son los que están contruidos dentro de los chips de microprocesadores modernos. Por lo tanto, estos procesadores tienen dentro de ellos un ALU muy complejo y potente. De hecho, un microprocesador moderno (y los mainframes) puede tener múltiples núcleos, cada núcleo con múltiples unidades de ejecución, cada una de ellas con múltiples ALU.

Muchos otros circuitos pueden contener en el interior una unidad aritméticológica: unidades de procesamiento gráfico como las que están en las GPU modernas, FPU como el viejo coprocesador matemático 80387, y procesadores digitales de señales como los que se encuentran en tarjetas de sonido, lectoras de CD y los televisores de alta definición. Todos estos tienen en su interior varias ALU potentes y complejas.

Historia: Propuesta de Von Neumann

El matemático John von Neumann propuso el concepto de la ALU en 1945, cuando escribió un informe sobre las fundaciones para un nuevo computador llamado EDVAC (Electronic Discrete Variable Automatic Computer) (Computador Automático Variable Discreto Electrónico). Más adelante, en 1946, trabajó con sus colegas diseñando un computador para el Princeton Institute of Advanced Studies (IAS) (Instituto de Princeton de Estudios Avanzados). El IAS computer se convirtió en el prototipo para muchos computadores posteriores. En esta propuesta, von Neumann esbozó lo que él creyó sería necesario en su máquina, incluyendo una ALU.

Von Neumann explicó que una ALU es un requisito fundamental para una computadora porque necesita efectuar operaciones matemáticas básicas: adición, sustracción, multiplicación, y división.¹ Por lo tanto, creyó que era "razonable que una computadora debería contener los órganos especializados para estas operaciones".

Sistemas numéricos

Una ALU debe procesar números usando el mismo formato que el resto del circuito digital. Para los procesadores modernos, este formato casi siempre es la representación del número binario de complemento a dos. Las primeras computadoras usaron una amplia variedad de sistemas de numeración, incluyendo complemento a uno, formato signo-magnitud, e incluso verdaderos sistemas decimales, con diez tubos por dígito.

Las ALU para cada uno de estos sistemas numéricos mostraban diferentes diseños, y esto influyó la preferencia actual por el complemento a dos, debido a que ésta es la representación más simple, para el circuito electrónico de la ALU, para calcular adiciones, sustracciones, etc.

Introducción práctica

La ALU se compone básicamente de: Circuito Operacional, Registros de Entradas, Registro Acumulador y un Registro de Estados, conjunto de registros que hacen posible la realización de cada una de las operaciones.

La mayoría de las acciones de la computadora son realizadas por la ALU. La ALU toma datos de los registros del procesador. Estos datos son procesados y los resultados de esta operación se almacenan en los registros de salida de la ALU. Otros mecanismos mueven datos entre estos registros y la memoria.

Una unidad de control controla a la ALU, al ajustar los circuitos que le señala a la ALU qué operaciones realizar.

Detalle

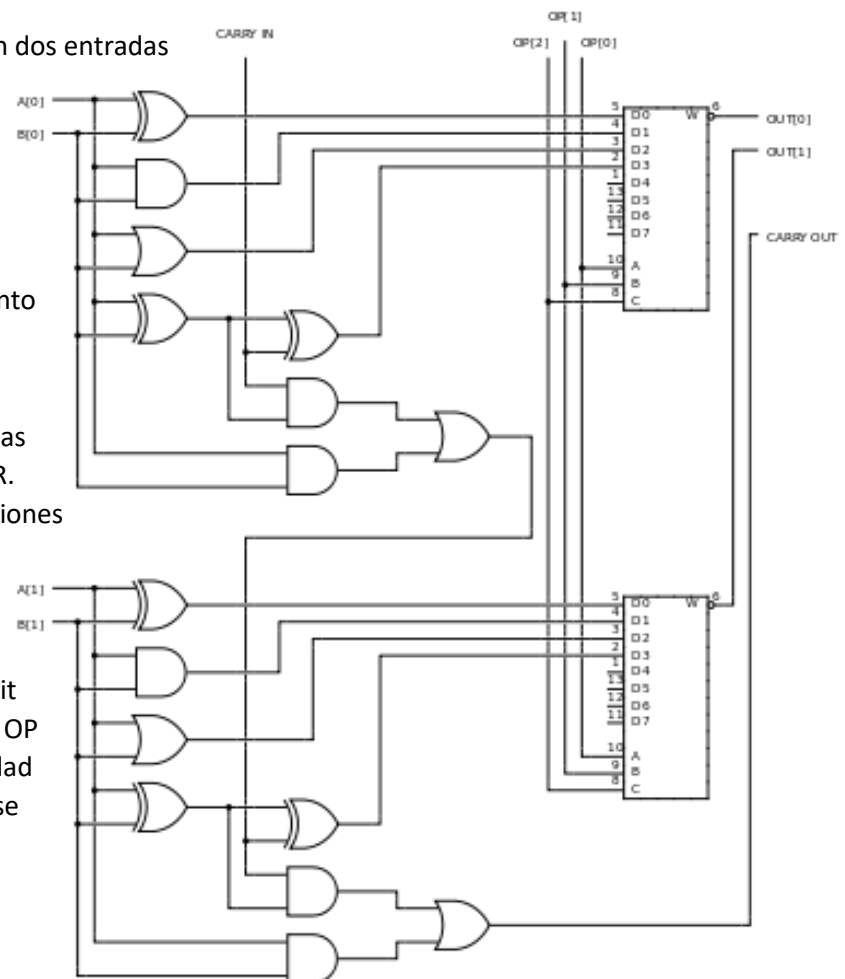
En la imagen se detalla una ALU de 2 bits con dos entradas (operandos) llamadas A y B: A[0] y B[0] corresponden al bit menos significativo y A[1] y B[1] corresponden al bit más significativo.

Cada bit de la ALU se procesa de manera idéntica, con la excepción del direccionamiento del bit del acarreo. El manejo de este bit es explicado más adelante.

Las entradas A y B van hacia las cuatro puertas de la derecha, de arriba abajo, XOR, AND, OR. Las tres primeras puertas realizan las operaciones XOR, AND, y OR sobre los datos A y B. La última puerta XOR es la puerta inicial de un sumador completo.

El paso final de las operaciones sobre cada bit es la multiplexación de los datos. La entrada OP de 3 bits, OP[0], OP[1] y OP[2] (desde la unidad de control) determina cuál de las funciones se van a realizar:

OP = 000 → XOR



OP = 001 → AND

OP = 010 → OR

OP = 011 → Adición

Claramente se ve que las otras cuatro entradas del multiplexor están libres para otras operaciones (sustracción, multiplicación, división, NOT A, NOT B, etc). Aunque OP[2] actualmente no es usada en este montaje (a pesar de estar incluida y conectada), ésta sería usada en el momento de realizar otras operaciones además de las 4 operaciones listadas arriba.

Los datos de acarreo de entrada y acarreo de salida, llamados flags (banderas), son típicamente conectados a algún tipo de registro de estado.

Operaciones simples

- La mayoría de las ALU pueden realizar las siguientes operaciones:
- Operaciones aritméticas de números enteros (adición, sustracción, y a veces multiplicación y división, aunque esto es más complejo)
- Operación lógica de bits (AND, NOT, OR, XOR, XNOR)
- Operación de desplazamiento de bits (Desplazan o rotan una palabra en un número específico de bits hacia la izquierda o la derecha, con o sin extensión de signo). Los desplazamientos pueden ser interpretados como multiplicaciones o divisiones por 2.

Operaciones complejas

Un ingeniero puede diseñar una ALU para calcular cualquier operación, sin importar lo compleja que sea; el problema es que cuanto más compleja sea la operación, tanto más costosa será la ALU, más espacio usará en el procesador, y más energía disipará, etc.

Por lo tanto, los ingenieros siempre calculan un compromiso, para proporcionar al procesador (u otros circuitos) una ALU suficientemente potente para calcular rápido, pero no de una complejidad de tal calibre que haga una ALU económicamente prohibitiva. Imagina que necesitas calcular, digamos, la raíz cuadrada de un número; el ingeniero digital examinará las opciones siguientes para implementar esta operación:

- Diseñar una ALU muy compleja que calcule la raíz cuadrada de cualquier número en un solo paso. Esto es llamado cálculo en un solo ciclo de reloj.
- Diseñar una ALU compleja que calcule la raíz cuadrada con varios pasos (como el algoritmo que algunos hemos aprendido en la escuela). Esto es llamado cálculo iterativo, y generalmente confía en el control de una unidad de control compleja con microcódigo incorporado.
- Diseñar una ALU simple en el procesador, y vender un procesador separado, especializado y costoso, que el cliente pueda instalar adicional al procesador, y que implementa una de las opciones de arriba. Esto es llamado coprocesador o unidad de coma flotante.

- Emular la existencia del coprocesador, es decir, siempre que un programa intente realizar el cálculo de la raíz cuadrada, hacer que el procesador compruebe si hay presente un coprocesador y usarlo si lo hay; si no hay uno, interrumpir el proceso del programa e invocar al sistema operativo para realizar el cálculo de la raíz cuadrada por medio de un cierto algoritmo de software. Esto es llamado emulación por software.
- Decir a los programadores que no existe el coprocesador y no hay emulación, así que tendrán que escribir sus propios algoritmos para calcular raíces cuadradas por software. Esto es realizado por bibliotecas de software.

Las opciones superiores van de la más rápida y más costosa a la más lenta y económica. Por lo tanto, mientras que incluso la computadora más simple puede calcular la fórmula más complicada, las computadoras más simples generalmente tomarán un tiempo largo porque varios de los pasos para calcular la fórmula implicarán las opciones #3, #4 y #5 de arriba.

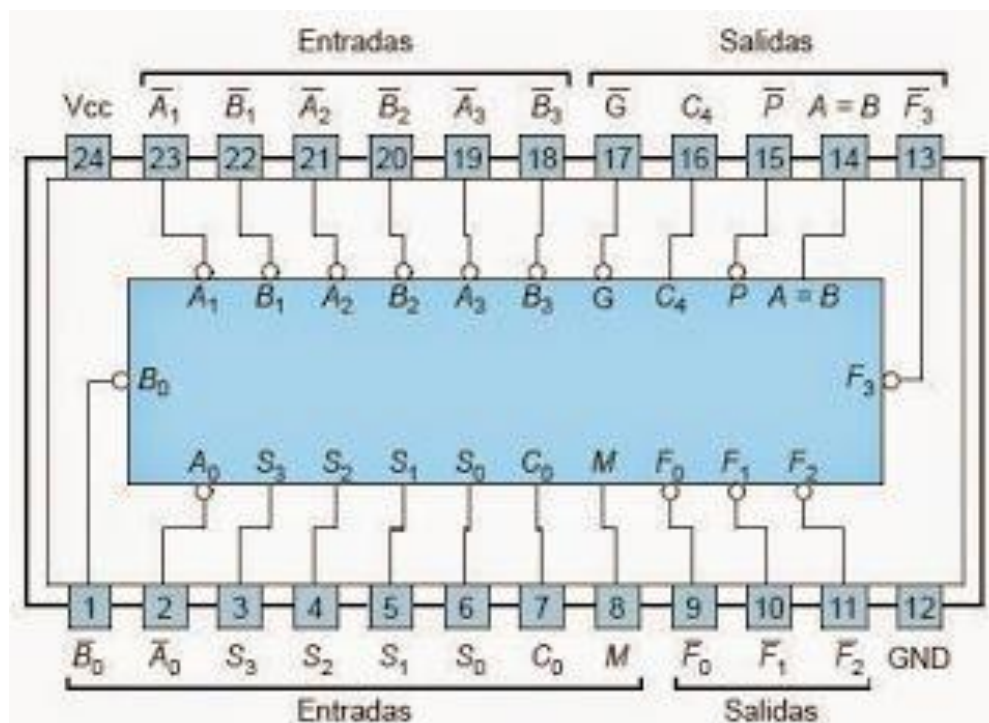
Los procesadores complejos como el Pentium IV y el AMD Athlon 64 implementan la opción #1 para las operaciones más complejas y la más lenta #2 para las operaciones extremadamente complejas. Eso es posible por la capacidad de construir ALU muy complejas en estos procesadores.

Entradas y salidas

Las entradas a la ALU son los datos en los que se harán las operaciones (llamados operandos) y un código desde la unidad de control indicando qué operación realizar. Su salida es el resultado del cómputo de la operación.

En muchos diseños la ALU también toma o genera como entradas o salidas un conjunto de códigos de condición desde o hacia un registro de estado. Estos códigos son usados para indicar casos como acarreo entrante o saliente, overflow, división por cero, etc.

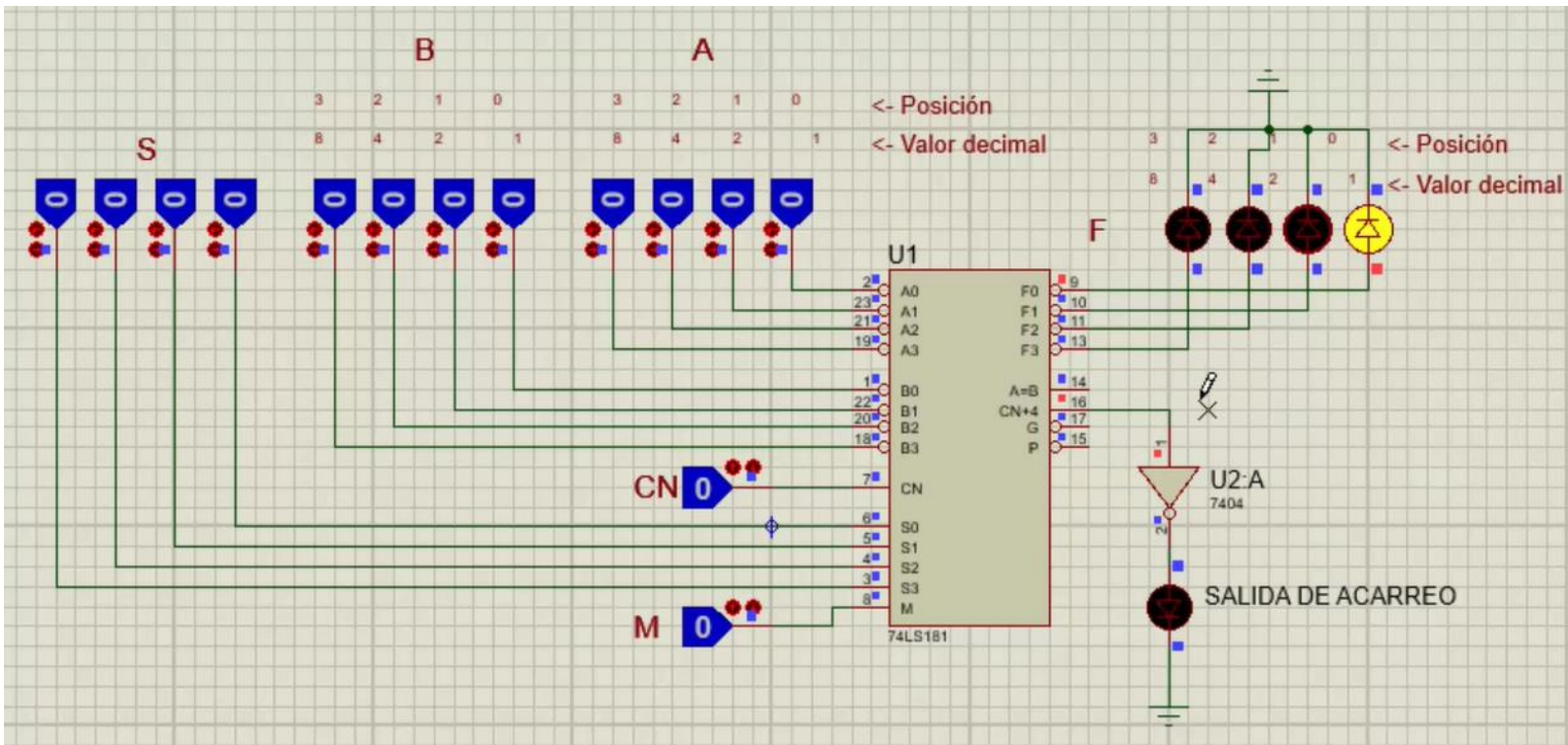
Pinout



Componentes

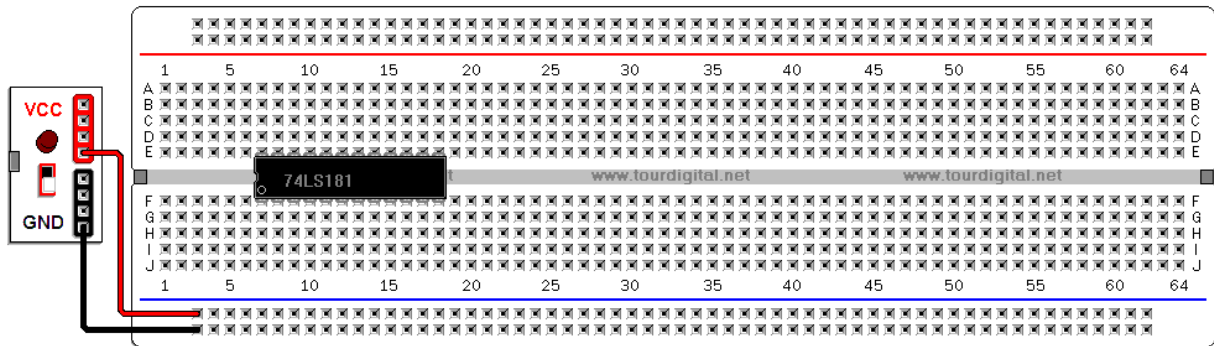
La ALU se compone básicamente de: Circuito Operacional, Registros de Entradas, Registro Acumulador y un Registro de Estados, conjunto de registros que hacen posible la realización de cada una de las operaciones. La mayoría de las acciones de la computadora son realizadas por la ALU. La ALU toma datos de los registros del procesador. Estos datos son procesados y los resultados de esta operación se almacena en los registros de salida de la ALU. Otros mecanismos mueven datos entre estos registros y la memoria. Una unidad de control controla a la ALU, al ajustar los circuitos que le señala a la ALU qué operaciones realizar.

Diagrama

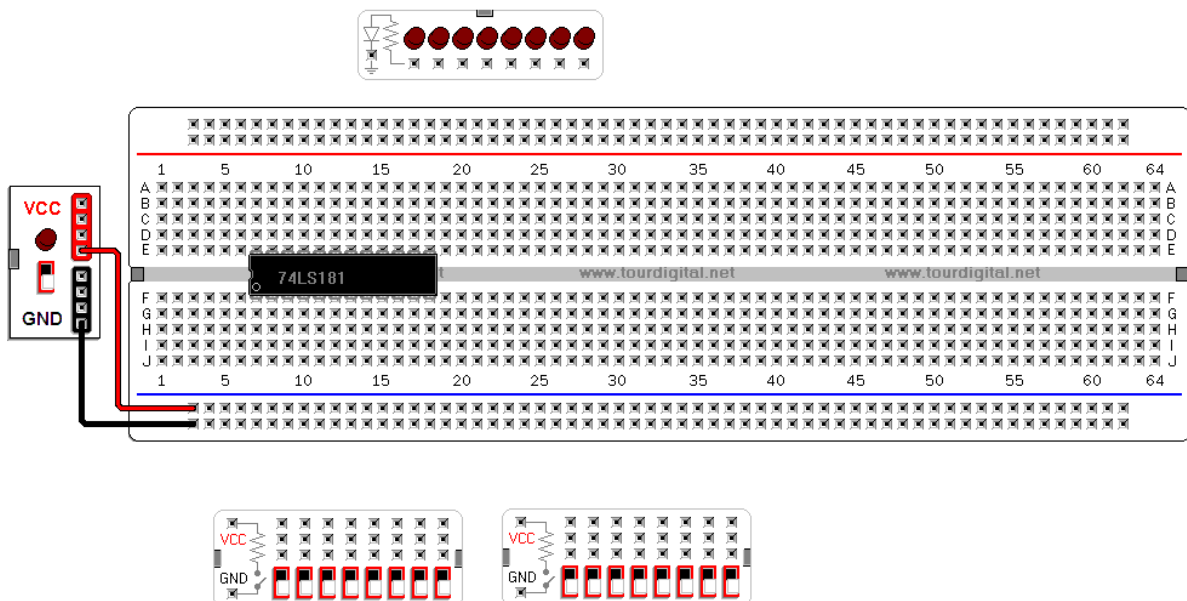


Proceso de diseño:

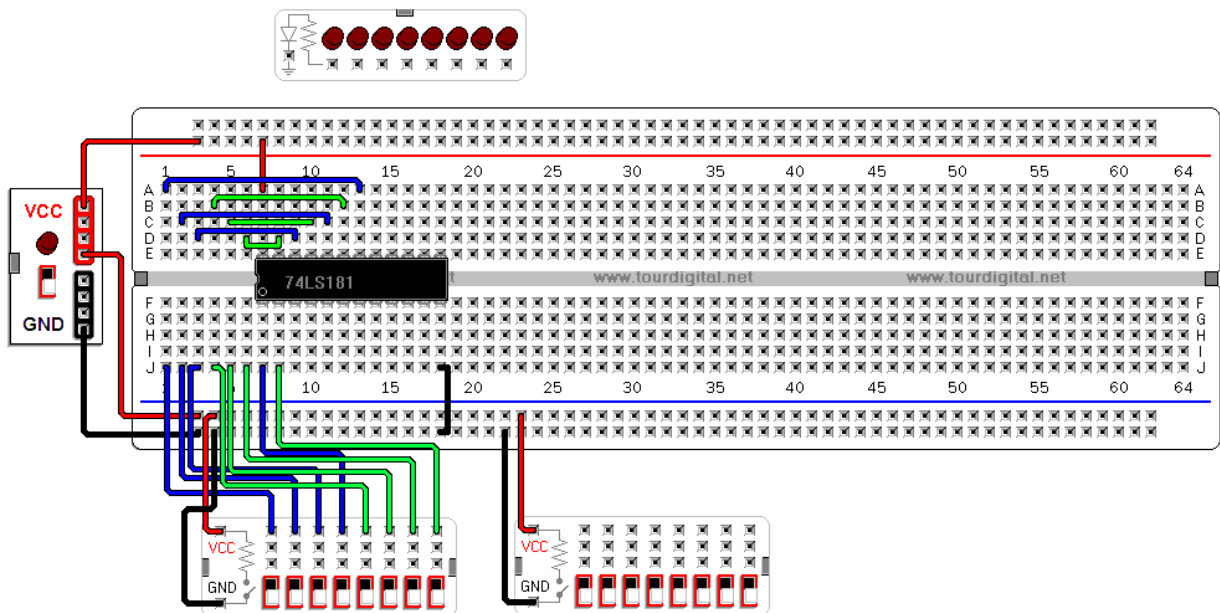
Primero tenemos nuestro protoboard, entonces procedemos a colocar el chip del ALU 72LS181 de 4 bits.



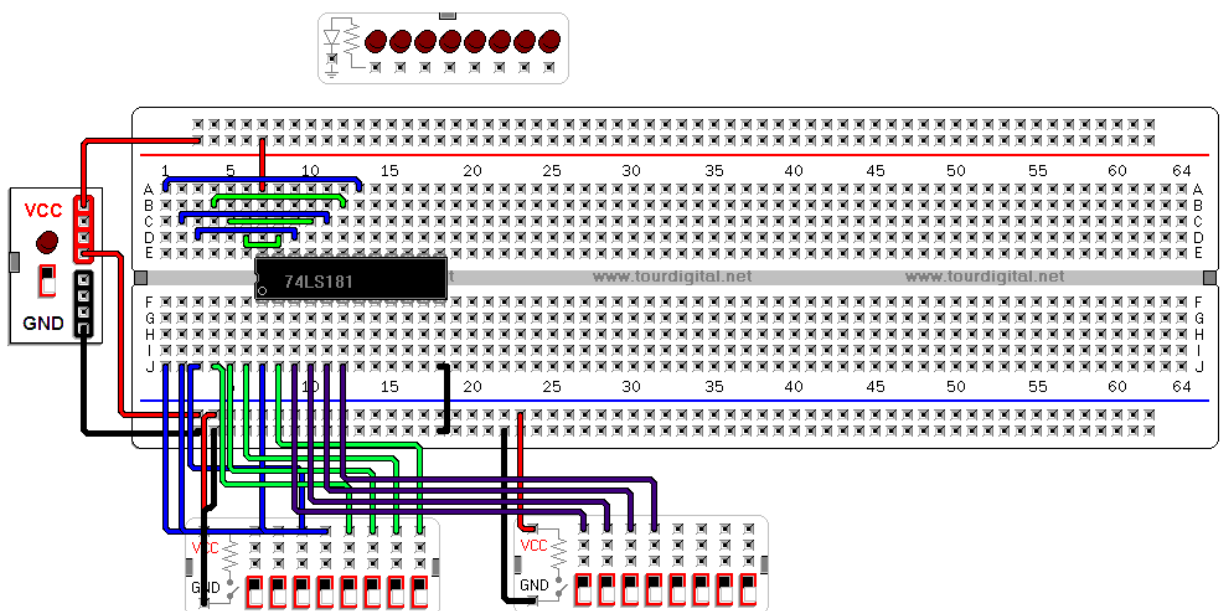
Colocamos los interruptores A y B con los que vamos a insertar nuestras entradas, y unos leds para demostrar las salidas (F).



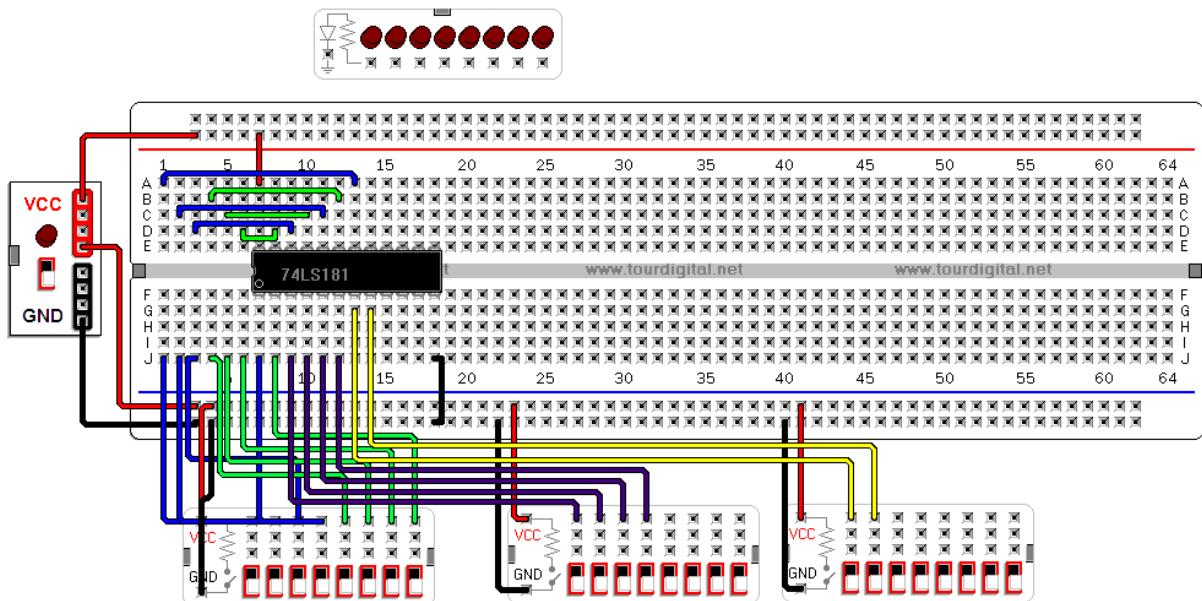
Conectamos los cables de energía positiva y negativa, para alimentar los interruptores, chip, y leds. Una vez que ya tengamos eso, podemos colocar nuestras entradas del primer interruptor “A”, siguiendo el orden del pinout del chip ALU los primeros 4 del primer tablero de interruptor debe ir con las entradas A0 - A3. La segunda mitad debe ir con las entradas B0 – B3



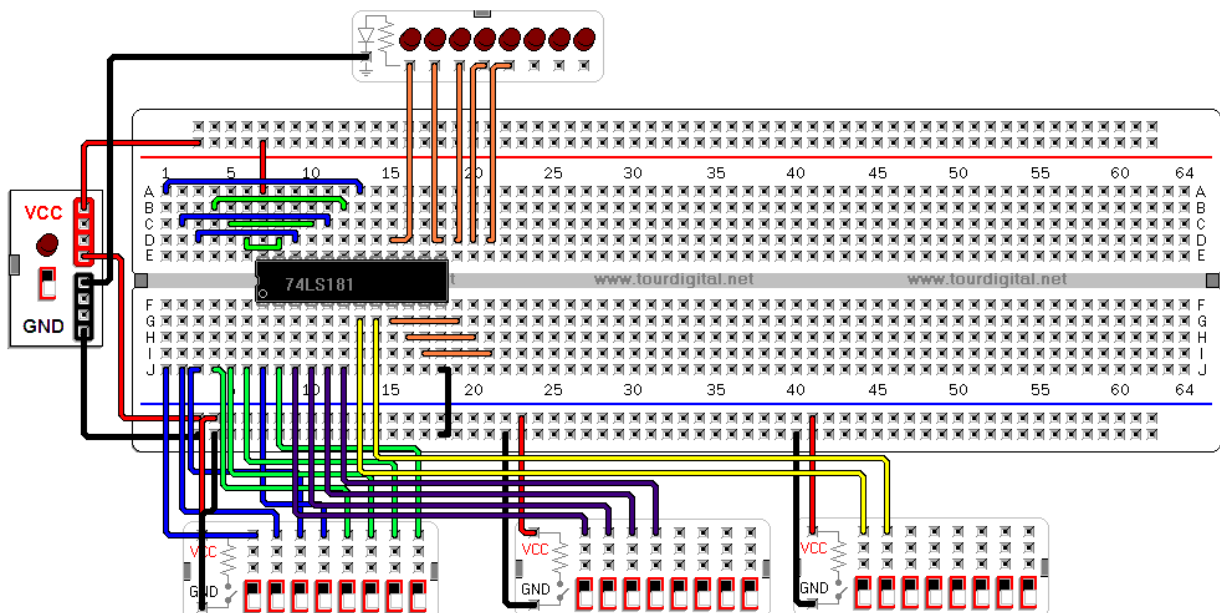
Y con los del segundo tablero de interruptores, colocamos los primeros 4 interruptores con las entradas S3 – S0 para los datos.



Ya tenemos las entradas, ahora colocamos un tercer tablero de interruptores para poder manipular las entradas CN y M para seleccionar el tipo de operación a realizar con M para lógica o aritmética y CN para cuando hay o no acarreo de entrada.



Finalmente colocamos las salidas, con el tablero de leds conectando con el primero con la salida C4 de acuerdo con el pinout, para la salida de acarreo. Los siguientes 4 conectados con las salidas F3 – F0. Y con esto tenemos nuestra ALU hecha con las entradas A,B, los datos con S, salidas C4, F y el control de operación con CN y M.



Descripción (funcionamiento)

Entrada A:

Posiciones 0 = 1, 1 = 2, 2 = 4, 3 = 8

Entrada B:

Posiciones: 4 = 1, 5 = 2, 6 = 4, 7 = 8

El ALU presentado funciona así:

Se ingresa el primer dígito en binario en la sección A, por ejemplo, si queremos ingresar 6, se levanta el interruptor 1 y 2 porque el valor de 1 es 2, y el valor de 2 es 4, por lo que $2 + 4 = 6$. Lo mismo sucede con la sección B.

En la sección S ingresamos la operación según la tabla de funciones siguiente:

SELECTION				ACTIVE-HIGH DATA	
				M = H LOGIC FUNCTIONS	M = L; ARITHMETIC OPERATIONS
S3	S2	S1	S0		<div> $\overline{C}_n = H$ (no carry) </div> <div> $\overline{C}_n = L$ (with carry) </div>
L	L	L	L	$F = \overline{A}$	$F = A$
L	L	L	H	$F = \overline{A + B}$	$F = A + B$
L	L	H	L	$F = \overline{AB}$	$F = A + \overline{B}$
L	L	H	H	$F = 0$	$F = \text{MINUS 1 (2's COMPL)}$
L	H	L	L	$F = \overline{AB}$	$F = A \text{ PLUS } \overline{AB}$
L	H	L	H	$F = \overline{B}$	$F = (A + B) \text{ PLUS } \overline{AB}$
L	H	H	L	$F = A \oplus B$	$F = A \text{ MINUS } B \text{ MINUS } 1$
L	H	H	H	$F = \overline{AB}$	$F = \overline{AB} \text{ MINUS } 1$
H	L	L	L	$F = \overline{A + B}$	$F = A \text{ PLUS } AB$
H	L	L	H	$F = \overline{A \oplus B}$	$F = A \text{ PLUS } B$
H	L	H	L	$F = B$	$F = (A + \overline{B}) \text{ PLUS } AB$
H	L	H	H	$F = AB$	$F = AB \text{ MINUS } 1$
H	H	L	L	$F = 1$	$F = A \text{ PLUS } A$
H	H	L	H	$F = A + \overline{B}$	$F = (A + B) \text{ PLUS } A$
H	H	H	L	$F = A + B$	$F = (A + \overline{B}) \text{ PLUS } A$
H	H	H	H	$F = A$	$F = A \text{ MINUS } 1$
					$F = A$

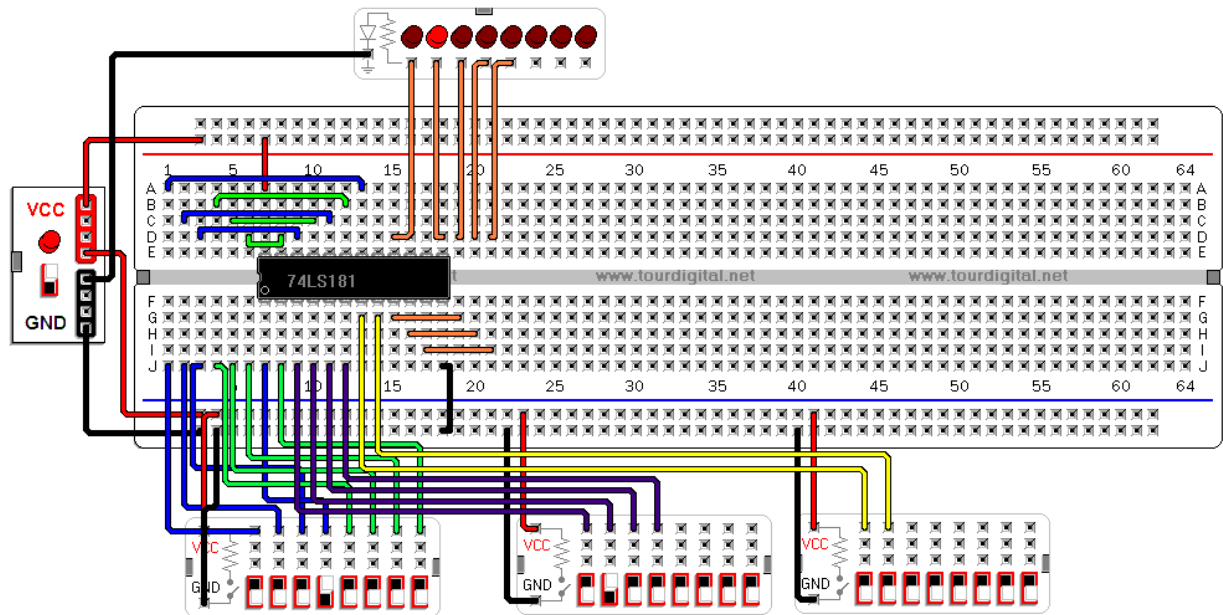
Seleccionamos M si queremos que la operación sea lógica o si queremos que sea aritmética, y seleccionamos CN si hay o no acarreo de entrada.

En la sección F se verá reflejado el resultado de la operación de los valores A y B.

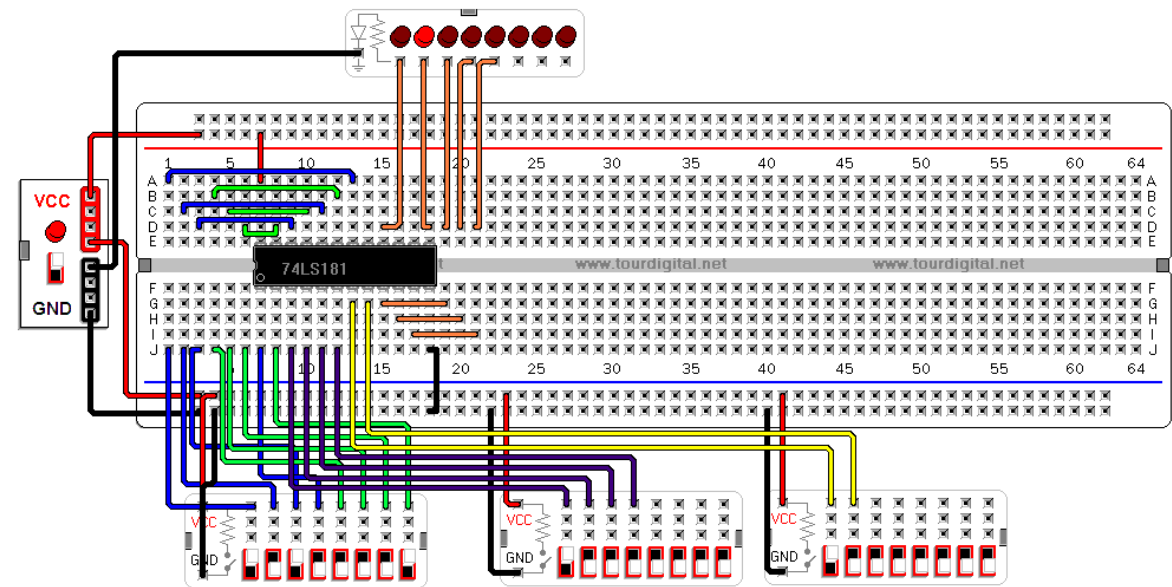
El led del acarreo o la sección C4 se prenderá en caso de que la operación no pueda representarse con 4 bits, o sea si es mayor de 15. Tiene el efecto contrario si está apagado.

Pruebas

| L L L H | $F = \overline{A+B}$ | $F = A+B$ | $F = (A+B) \text{ PLUS } 1$ |



| H L H H | $F = AB$ | $F = AB \text{ MINUS } 1$ | $F = AB$ |



Conclusión

A partir del chip 74LS181 se pudo diseñar una Unidad Aritmética (ALU) de cuatro bits de entrada la cual hace uso diferentes funciones booleanas y estas se dividieron en bloques para crear las diferentes operaciones que simularía el procesador de una computadora así mismo se pudieron realizar diversas funciones lógicas tanto como booleanas para las entradas de cuatro bits.