

Unidad aritmético lógica



Este artículo o sección necesita **referencias** que aparezcan en una **publicación acreditada**, como revistas especializadas, monografías, prensa diaria o páginas de Internet fidedignas.

Puedes añadirlas así o avisar al autor principal del artículo ^[1] en su página de discusión pegando: {{subst:Aviso referencias|Unidad aritmético lógica}} ~~~~

En computación, la **unidad aritmético lógica**, también conocida como **ALU** (siglas en inglés de *arithmetic logic unit*), es un circuito digital que calcula operaciones aritméticas (como suma, resta, multiplicación, etc.) y operaciones lógicas (sí, y, o, no), entre dos números.

Muchos tipos de circuitos electrónicos necesitan realizar algún tipo de operación aritmética, así que incluso el circuito dentro de un reloj digital tendrá una ALU minúscula que se mantiene sumando 1 al tiempo actual, y se mantiene comprobando si debe activar el pitido del temporizador, etc.

Por mucho, los más complejos circuitos electrónicos son los que están contruidos dentro de los chips de microprocesadores

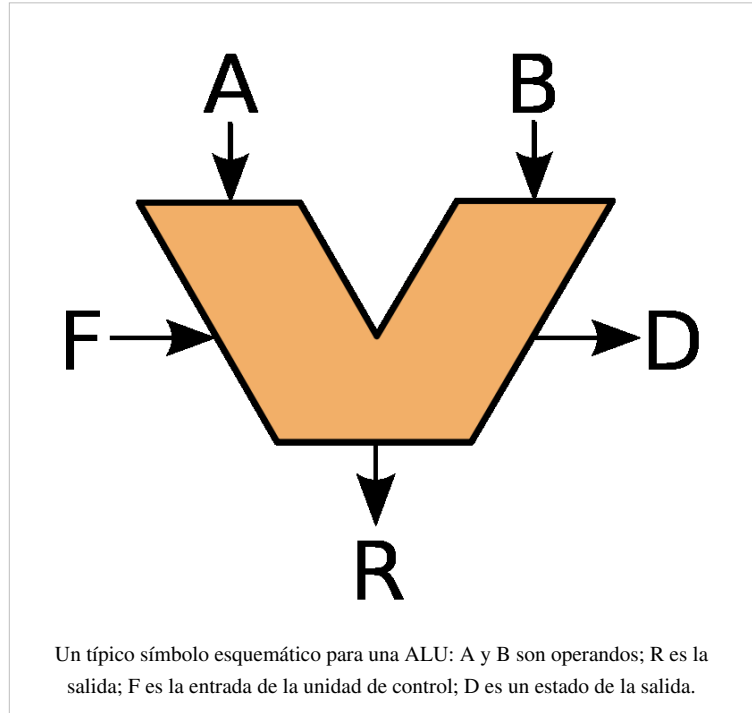
modernos como el Intel Core i7 o el Phenom II. Por lo tanto, estos procesadores tienen dentro de ellos un ALU muy complejo y potente. De hecho, un microprocesador moderno (y los mainframes) pueden tener múltiples núcleos, cada núcleo con múltiples unidades de ejecución, cada una de ellas con múltiples ALU.

Muchos otros circuitos pueden contener en el interior una unidad aritmético lógica: unidades de procesamiento gráfico como las que están en las GPU NVIDIA y AMD, FPU como el viejo coprocesador matemático 80387, y procesadores digitales de señales como los que se encuentran en tarjetas de sonido Sound Blaster, lectoras de CD y los televisores de alta definición. Todos éstos 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 la 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 está garantizado que tendrá que efectuar operaciones matemáticas básicas, incluyendo adición, sustracción, multiplicación, y división.^[2] Por lo tanto, creyó que era "razonable que una computadora debería contener los órganos especializados para estas operaciones".^[2]



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.

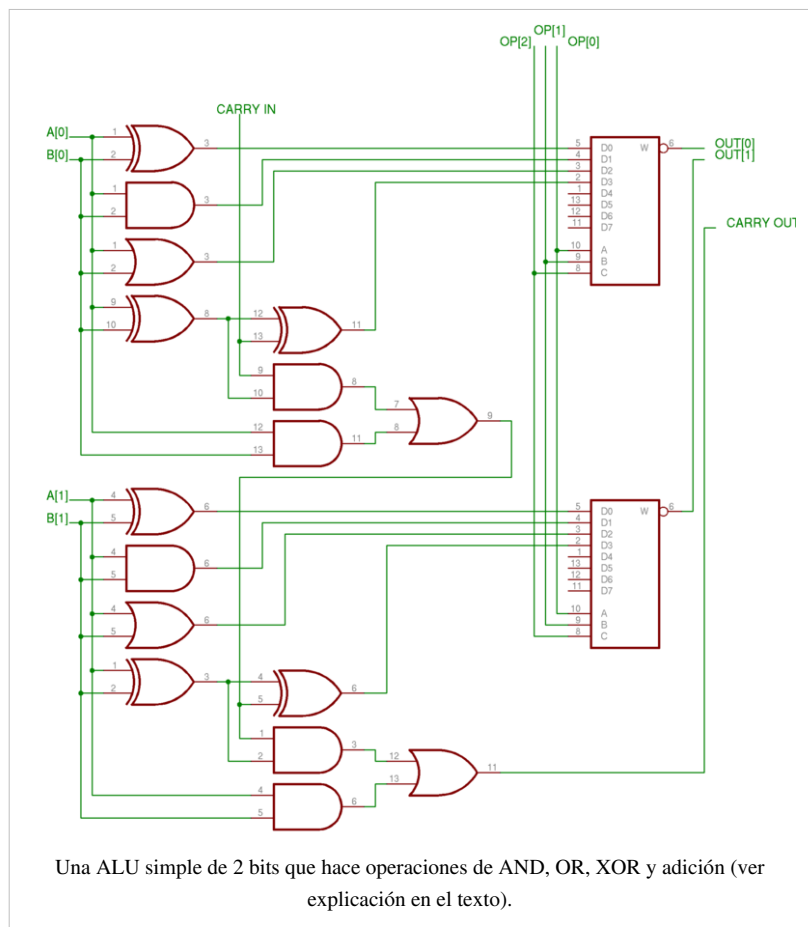
Las ALU para cada uno de estos sistemas numéricos mostraban diferentes diseños, y esto influyó la preferencia actual por el complemento a dos, debido a que ésta es la representación más simple, para el circuito electrónico de la ALU, para calcular adiciones y sustracciones, etc.

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.^[3]

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 izquierda, de arriba a abajo, AND, OR, y XOR. 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 cual 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)
- Operaciones lógicas de bits (AND, NOT, OR, XOR, XNOR)
- Operaciones 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:*

1. 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**.
2. Diseñar una ALU compleja que calcule la raíz cuadrada con varios pasos (como el algoritmo que aprendimos en la escuela). Esto es llamado **cálculo interactivo**, y generalmente confía en el control de una unidad de control compleja con microcódigo incorporado.
3. 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 **coprocador** o unidad de coma flotante.
4. Emular la existencia del coprocador, es decir, siempre que un programa intente realizar el cálculo de la raíz cuadrada, hacer que el procesador compruebe si hay presente un coprocador 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**.
5. Decir a los programadores que no existe el coprocador 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.^[3]

ALU vs. FPU

Una unidad de coma flotante, *Floating Point Unit* (FPU), también realiza operaciones aritméticas entre dos valores, pero lo hace para números en representación de coma flotante, que es mucho más complicada que la representación de complemento a dos usada comúnmente en una ALU. Para hacer estos cálculos, una FPU tiene incorporados varios circuitos complejos, no incluyendo algunas ALU internas.

Generalmente los ingenieros llaman ALU al circuito que realiza operaciones aritméticas en formatos de número entero (como complemento a dos y BCD), mientras que los circuitos que calculan en formatos más complejos como coma flotante, números complejos, etc., reciben generalmente un nombre más ilustre.

Véase también

- Arquitectura de CPU
- CPU
- Microprocesador
- Unidad de control
- Unidad de punto flotante
- Bus interface unit
- Unidad de gestión de memoria
- Unidad de ejecución
- Unidad de proceso
- Registro (hardware)
- Microcódigo
- Barrel shifter
- Serie 7400 (chip # 74181)
- Sumador
- Multiplicador
- Circuito digital
- División (electrónica)

Referencias

[1] http://en.wikipedia.org/wiki/Unidad_aritm%C3%A9tico_l%C3%B3gica?action=history

[2] Stallings page 19

[3] Stallings page 290-291

- Stallings, William (2006). *Computer Organization & Architecture: Designing for Performance 7th ed* (<http://williamstallings.com/COA/COA7e.html>). Pearson Prentice Hall. ISBN 0-13-185644-8.

Enlaces externos

- A Simulator of Complex ALU in MATLAB (<http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=12762&objectType=FILE>)

Fuentes y contribuyentes del artículo

Unidad aritmético lógica *Fuente:* <http://es.wikipedia.org/w/index.php?oldid=42961131> *Contribuyentes:* ALE!, Adrián Hermida, Bentzia, Biasoli, Camilo, Centeno, Diegusjaimes, Dodo, FrancoGG, GermanX, Goofys, Imapix, JMPerez, JorgeGG, Justy, KanTagoff, Lucien leGrey, Lungo, Magister Mathematicae, ManuelGR, Matdroses, Moriel, Museo8bits, Nihilo, Olea, Piero71, Porao, Rbuj, RoyFocker, Superzerocool, Taichi, Tigerfenix, TorQue Astur, Truor, Varano, Vladimirdlc, 117 ediciones anónimas

Fuentes de imagen, Licencias y contribuyentes

Imagen:Question book.svg *Fuente:* http://es.wikipedia.org/w/index.php?title=Archivo:Question_book.svg *Licencia:* GNU Free Documentation License *Contribuyentes:* El Muñeco Shakes It Up, Baby, Javierme, Loyna, Remember the dot, Victormoz, Wouterhagens, 5 ediciones anónimas

Archivo:ALU symbol.svg *Fuente:* http://es.wikipedia.org/w/index.php?title=Archivo:ALU_symbol.svg *Licencia:* GNU Free Documentation License *Contribuyentes:* Cburnett, Eadthem, Herbythyme, 4 ediciones anónimas

Archivo:2-bit ALU.png *Fuente:* http://es.wikipedia.org/w/index.php?title=Archivo:2-bit_ALU.png *Licencia:* GNU Free Documentation License *Contribuyentes:* Cburnett, German, Rod9h, WikipediaMaster, 10 ediciones anónimas

Licencia

Creative Commons Attribution-Share Alike 3.0 Unported
<http://creativecommons.org/licenses/by-sa/3.0/>