INSTITUTO POLITÉCNICO NACIONAL

Escuela Superior de Cómputo

Unidad de Aprendizaje

Diseño de sistemas digitales

Presentación de

“ALU”

Profesor:

López Leyva Luis Octavio

Grupo:

2CV8

Alumnos:

Cortez Enríquez Jovanny Wilver

Luciano Espina Melisa

Fecha de Entrega:

04/Mayo/2018

Contenido

[Historia 3](#_Toc515923579)

[¿Qué es? 3](#_Toc515923580)

[Carry 4](#_Toc515923581)

[Auxiliary Carry 4](#_Toc515923582)

[Overflow 4](#_Toc515923583)

[Parity 4](#_Toc515923584)

[Zero 4](#_Toc515923585)

[Composición 5](#_Toc515923586)

[Circuito operacional 5](#_Toc515923587)

[Registro de entradas 5](#_Toc515923588)

[Registro acumulador 6](#_Toc515923589)

[Registro de estados (flags) 6](#_Toc515923590)

[Símbolo de la ALU 7](#_Toc515923591)

[ALU 7](#_Toc515923592)

[Integrado 7](#_Toc515923593)

[Tabla de funcionamiento 8](#_Toc515923594)

[Activo en alto 9](#_Toc515923595)

[Activo en bajo 9](#_Toc515923596)

[Funcionamiento 10](#_Toc515923597)

[Entradas y salidas 10](#_Toc515923598)

[Codificación 10](#_Toc515923599)

[Bibliografías 11](#_Toc515923600)

# Resultado de imagen para von neumannHistoria

El matemático John von Neumann propuso el concepto de la ALU en 1945

Explicó que una ALU es un requisito fundamental para una computadora porque necesita efectuar operaciones matemáticas básicas: adición, sustracción, multiplicación, y división.

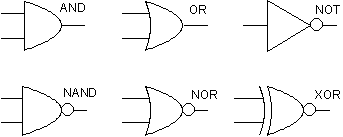
*Figura 1: John Von Neumann*

# ¿Qué es?



* Unidad aritmética lógica
* Hace operaciones lógicas (AND, OR, NOT, XOR, etc)
* Hace operaciones aritméticas (Suma, resta, multiplicación, división)
* Además de los operadores lógicos y aritméticos, la ALU cuenta con una serie de registros para almacenar los datos, y bits de información sobre los resultados, también llamados banderas.

*Figura 2: Operaciones aritméticas*

* Las banderas más comunes son: Carry, Auxiliary Carry, Borrow, Overflow, Parity, Zero.

*Figura 3: Compuertas lógias*

## Carry

* Indica acarreo o préstamo en el bit más significativo, después de una operación aritmética
* Cf= 0; No hubo acarreo
* Cf= 1; hubo acarreo

## Auxiliary Carry

* Si vale 1: indica que hubo préstamo de nibble
* Si vale 0: No hubo préstamo

## Overflow

* Se activa después de que una operación aritmética de suma o resta ha ocurrido un sobreflujo
* No hubo sobre flujo (Of = 0)
* Hubo sobre flujo (Of = 1)

## Parity

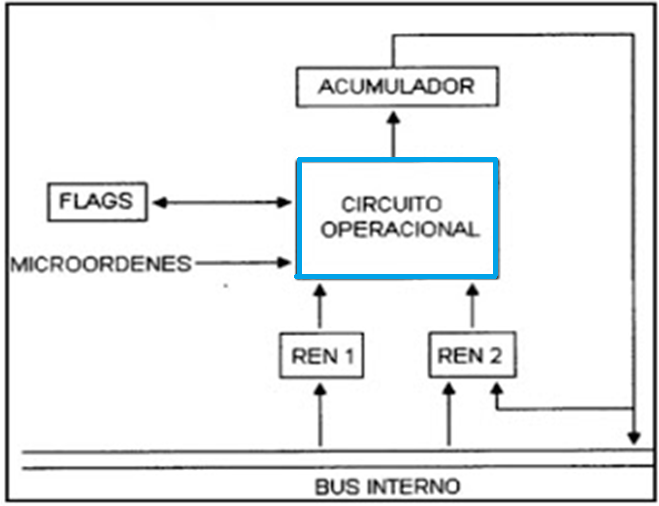
* Se refiere a la paridad del resultado de una operación aritmética o lógica
* Si vale 1 = paridad par
* Si vale 0 = paridad impar

## Zero

* Para operaciones lógicas y aritméticas, la bandera de cero se pone en el estado alto (1), si el resultado es cero.
* En caso contrario la bandera se mantiene en estado bajo (0)
* El resultado no fue cero (Zf = 0)
* El resultado fue cero (Zf = 1)

# Composición

## Circuito operacional

Contiene los circuitos necesarios para la realización de las operaciones con los datos procedentes de los registros de entrada (REN). Este circuito tiene unas entradas de órdenes para seleccionar la clase de operación que debe realizar en cada momento (suma, resta, etc).

*Figura 4: ALU internamente, refiriéndose al circuito operacional*

# Registro de entradas

En ellos se almacenan los datos u operandos que intervienen en una instrucción antes de la realización de la operación por parte del circuito operacional. También se emplean para el almacenamiento de resultados intermedios o finales de las operaciones respectivas.

Imagen que contiene captura de pantalla

Descripción generada con confianza muy alta

*Figura 5: ALU internamente, refiriéndose al registro de entradas*

# Registro acumulador

Almacena los resultados de las operaciones llevadas a cabo por el circuito operacional. Está conectado con los registros de entrada para realimentación en el caso de operaciones encadenadas. Asimismo, tiene una conexión directa al bus de datos para el envío de los resultados a la memoria central o a la unidad de control.

Imagen que contiene captura de pantalla

Descripción generada con confianza muy alta

*Figura 6: ALU internamente, refiriéndose al acumulador*

# Registro de estados (flags)

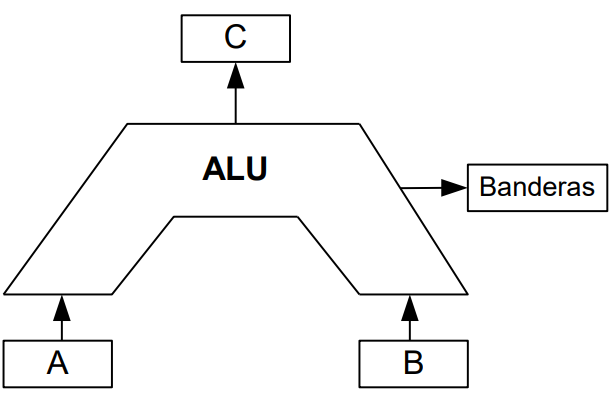
Son registros de memoria en los que se deja constancia algunas condiciones que se dieron en la última operación realizada y que habrán de ser tenidas en cuenta en operaciones posteriores. Por ejemplo, en el caso de hacer una resta, tiene que quedar constancia si el resultado fue cero, positivo o negativo.

Imagen que contiene captura de pantalla

Descripción generada con confianza muy alta

*Figura 7: ALU internamente, refiriéndose a la bandera*

# Símbolo de la ALU



*Figura 8: Diagrama de bloques de ALU*

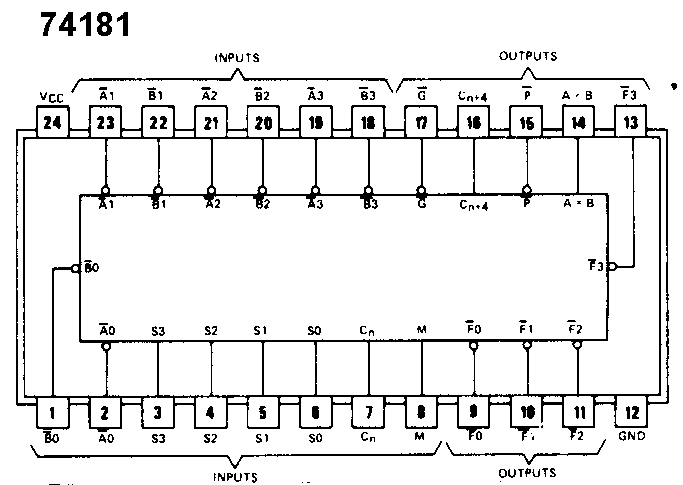
*Se observa que es una “V” representado el nombre de Von Neumann*

# ALU

* El circuito ALU es simplemente un operador, es decir, sólo realiza operaciones.
* La ALU no toma decisiones.
* Las entradas deben contener tanto la magnitud como el signo que corresponda a la operación
* La ALU requiere de un mecanismo de control que le permita saber el tipo de operación a realizar

# Integrado

* Unidad lógica aritmética (ALU) de 4 bits
* 16 funciones aritméticas incluyendo suma, resta, comparar, doblar
* 16 funciones lógicas incluyendo and, or, not, nand, nor, xor, comparar
* Operandos activos en nivel alto o bajo
* Entrada y salidas de carry. Múltiples unidades pueden ser combinadas
* Voltaje de alimentación: 4.75 V a 5.25 V
* Encapsulado: DIP ancho de 24 pines



*Figura 9: Esquema de ALU mostrando todos los pines*

# Resultado de imagen para Diagrama 74ls181Tabla de funcionamiento

*Figura 10: Tabla de funcionamiento de la ALU*

## Resultado de imagen para Diagrama 74ls181Activo en alto

Para activar las funciones lógicas, la entrada de control M (PIN 8) debe estar en alto. Ver figura 11

## Activo en bajo

* Para las operaciones artiméticas debe estar en activo bajo
* Para operaciones sin acarreo “Cn” debe estar en alto
* Para hacer operaciones de acarreo “Cn” debe estar en bajo, ya que la entrada está negada

Ver figura 12

Imagen que contiene texto

Descripción generada con confianza alta

*Figura 11: Tabla de Activo en alto*

*Figura 12: Tabla de Activo en bajo*

# https://upload.wikimedia.org/wikipedia/commons/thumb/8/82/ALU_symbol.svg/175px-ALU_symbol.svg.pngFuncionamiento

* Se tienen dos valores de entrada A y B son operandos de entrada
* R: Es el resultado dado por A y B
* F: Entrada de operación de control, indica qué operación se va a hacer
* D: Salida

*Figura 13: Diagrama de ALU*

# 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.

# Codificación

1. Library ieee;
2. Use ieee.std\_logi\_1164.all;
3. Use ieee.std\_logic\_unsigned.all;
4. Entity alur ir port(
5. Oper: in std\_logc;
6. X: in std\_logic; --entrada a
7. Y: in std\_logic; --entrada b
8. Bin: in std\_logic; --acarreo
9. Bout: out std\_logic –acarreo salida
10. S: out std\_logic --salida
11. );
12. End alur;
13. Architecture arqui\_alur of alur is
14. begin
15. process
16. Begin
17. If oper = ‘0’ then
18. S <= x xor y xor bin;
19. Bout <= (bin and (not(x xor y))) or (not(x) and y)
20. End if;
21. End process;
22. End arqui:alur;

# Bibliografías

* [1] Pedro Gutierrez, ¿Cómo funciona la computación actual? Funcionamiento de la arquitectura de von neuman| Genbetadev.com [Online]Available: <https://www.genbetadev.com/actualidad/como-funciona-la-computacion-actual-funcionaiento-de-la-arquitectura-de-von-neumann>
* [2] Diccionario de tecnología, Definición de UAL| Alegsa.com.ar [Online] Available: <http://www.alegsa.com.ar/Dic/ual.php>
* [3] Diccionario tecnología, Definición de ALU| Alegsa.com.ar [Online] Available: <http://www.alegsa.com.ar/Dic/alu.php>