



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

¿Qué es?	3
Carry.....	4
Auxiliary Carry.....	4
Overflow	4
Parity.....	4
Zero.....	4
Composición	5
Circuito operacional.....	5
Registro de entradas	5
Registro acumulador	6
Registro de estados (flags)	6
Símbolo de la ALU	7
ALU 7	
Integrado.....	7
Tabla de funcionamiento	8
Activo en alto	9
Activo en bajo	9
Funcionamiento	10
Entradas y salidas.....	10
Codificación.....	10
Bibliografías	11

Historia

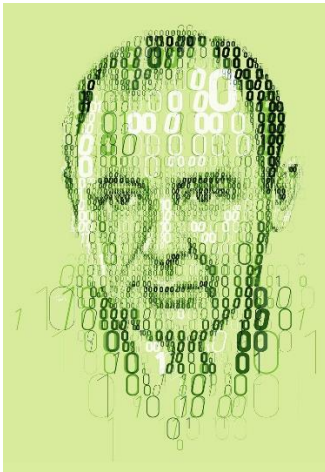


Figura 1: John Von Neumann

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.

¿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.
- ↪ Las banderas más comunes son: Carry, Auxiliary Carry, Borrow, Overflow, Parity, Zero.



Figura 2:
Operaciones aritméticas

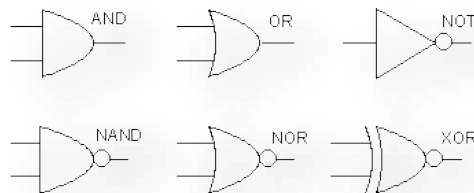


Figura 3:
Compuertas lógicas

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).

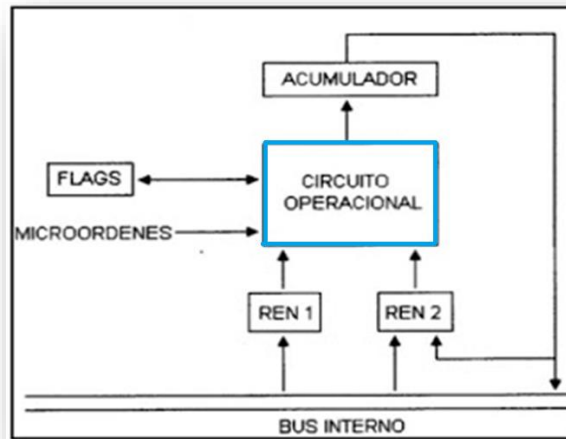


Figura 4: ALU internamente, refiriéndose al circuito operacional

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.

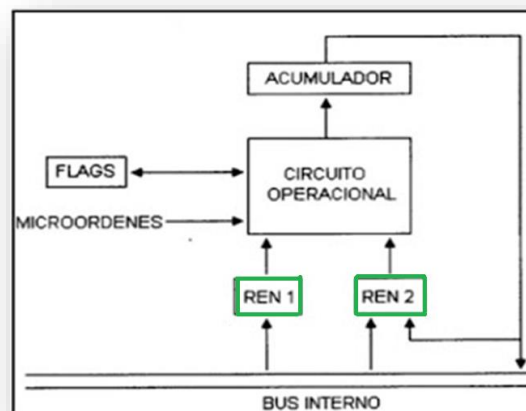


Figura 5: ALU internamente, refiriéndose al registro de entradas

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.

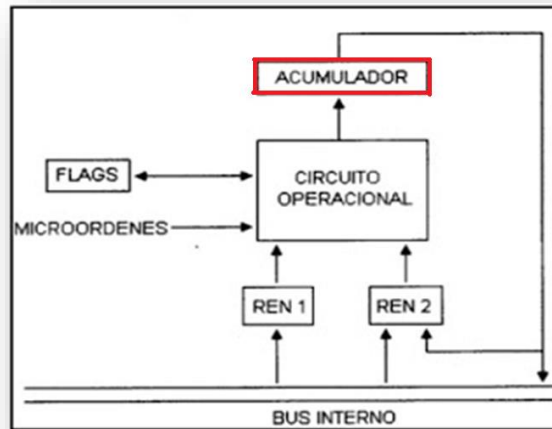


Figura 6: ALU internamente, refiriéndose al acumulador

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.

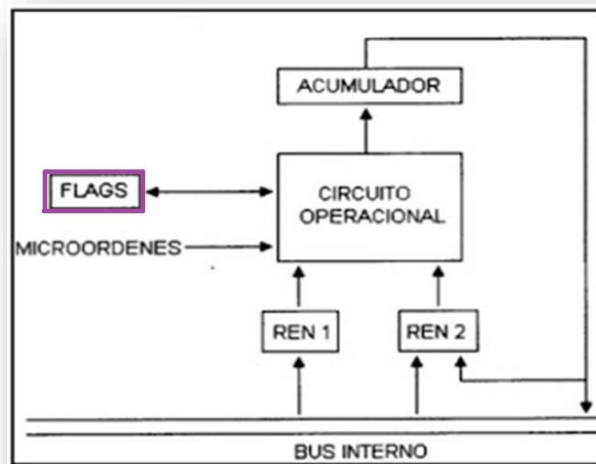


Figura 7: ALU internamente, refiriéndose a la bandera

Símbolo de la ALU

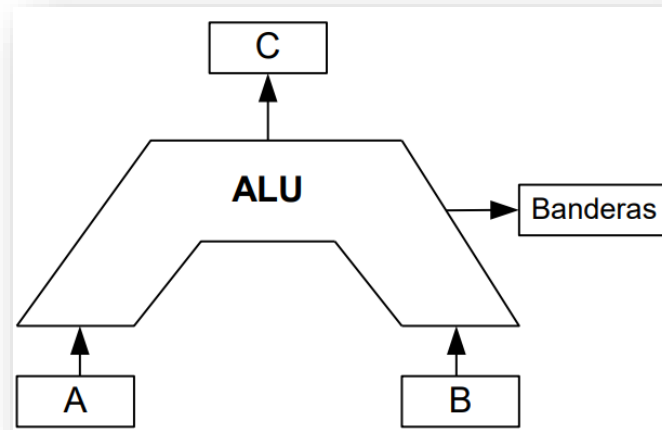


Figura 8: Diagrama de bloques de ALU

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

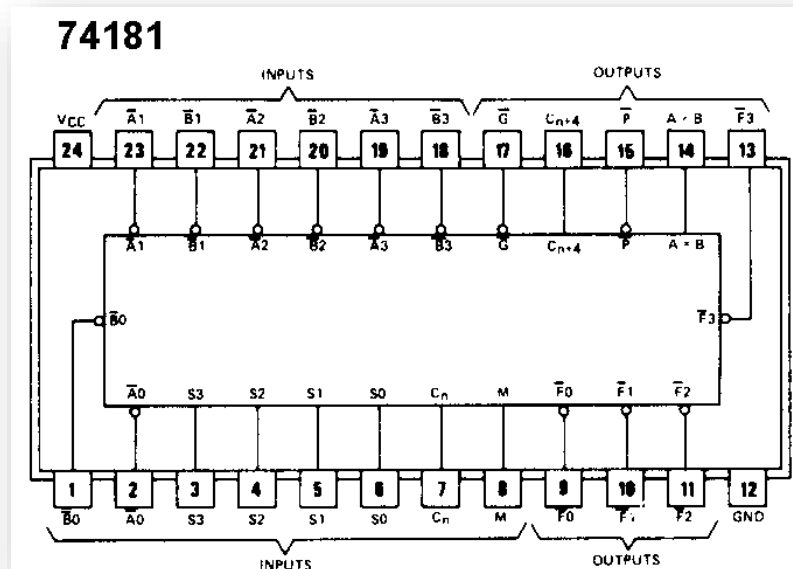


Figura 9: Esquema de ALU mostrando todos los pines

Tabla de funcionamiento

TABLE 1				
SELECTION				ACTIVE-LOW DATA
				M = L; ARITHMETIC OPERATIONS
S3	S2	S1	S0	
				M = H LOGIC FUNCTIONS
L	L	L	L	$F = \overline{A}$
L	L	L	H	$F = \overline{AB}$
L	L	H	L	$F = \overline{A} + B$
L	L	H	H	$F = 1$
L	H	L	L	$F = \overline{A + B}$
L	H	L	H	$F = \overline{B}$
L	H	H	L	$F = A \oplus B$
L	H	H	H	$F = A + \overline{B}$
H	L	L	L	$F = \overline{AB}$
H	L	L	H	$F = A \oplus B$
H	L	H	L	$F = B$
H	L	H	H	$F = A + B$
H	H	L	L	$F = 0$
H	H	L	H	$F = \overline{AB}$
H	H	H	L	$F = AB$
H	H	H	H	$F = A$
				Cn = L (no carry)
				$F = A \text{ MINUS } 1$
				$F = AB \text{ MINUS } 1$
				$F = \overline{AB} \text{ MINUS } 1$
				$F = \text{MINUS } 1 \text{ (2's COMP)}$
				$F = A \text{ PLUS } (A + \overline{B})$
				$F = AB \text{ PLUS } (A + \overline{B})$
				$F = A \text{ MINUS } B \text{ MINUS } 1$
				$F = A + \overline{B}$
				$F = A \text{ PLUS } (A + B)$
				$F = A \text{ PLUS } B$
				$F = \overline{AB} \text{ PLUS } (A + B)$
				$F = (A + B)$
				$F = A \text{ PLUS } A^\dagger$
				$F = AB \text{ PLUS } A$
				$F = \overline{AB} \text{ PLUS } A$
				$F = A$
				Cn = H (with carry)
				$F = A$
				$F = AB$
				$F = \overline{AB}$
				$F = \text{ZERO}$
				$F = A \text{ PLUS } (A + \overline{B}) \text{ PLUS } 1$
				$F = AB \text{ PLUS } (A + \overline{B}) \text{ PLUS } 1$
				$F = A \text{ MINUS } B$
				$F = (A + \overline{B}) \text{ PLUS } 1$
				$F = A \text{ PLUS } (A + B) \text{ PLUS } 1$
				$F = A \text{ PLUS } B \text{ PLUS } 1$
				$F = \overline{AB} \text{ PLUS } (A + B) \text{ PLUS } 1$
				$F = (A + B) \text{ PLUS } 1$
				$F = A \text{ PLUS } A \text{ PLUS } 1$
				$F = AB \text{ PLUS } A \text{ PLUS } 1$
				$F = \overline{AB} \text{ PLUS } A \text{ PLUS } 1$
				$F = A \text{ PLUS } 1$

[†]Each bit is shifted to the next more significant position.

Figura 10: Tabla de funcionamiento de la ALU

M = H
LOGIC
FUNCTIONS
$F = \overline{A}$
$F = \overline{AB}$
$F = \overline{A} + B$
$F = 1$
$F = \overline{A} + \overline{B}$
$F = \overline{B}$
$F = A \oplus B$
$F = A + \overline{B}$
$F = \overline{AB}$
$F = A \oplus B$
$F = B$
$F = A + B$
$F = 0$
$F = \overline{AB}$
$F = AB$
$F = A$

Figura 11:
Tabla de
Activo en alto

Activo 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 aritméticas 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

M = L; OPERACIONES ARITMÉTICAS	
$\overline{C_n} = H$ (sin acarreo)	$\overline{C_n} = L$ (con acarreo)
$F = A$	$F = A \text{ MÁS } 1$
$F = A + B$	$F = (A + B) \text{ MÁS } 1$
$F = A + \overline{B}$	$F = (A + \overline{B}) \text{ MÁS } 1$
$F = \text{MENOS } 1 \text{ (COMP. A2)}$	$F = \text{CERO}$
$F = A \text{ MÁS } \overline{AB}$	$F = A \text{ MÁS } \overline{AB} \text{ MÁS } 1$
$F = (A + B) \text{ MÁS } \overline{AB}$	$F = (A + B) \text{ MÁS } \overline{AB} \text{ MÁS } 1$
$F = A \text{ MENOS } B \text{ MENOS } 1$	$F = A \text{ MENOS } B$
$F = \overline{AB} \text{ MENOS } 1$	$F = \overline{AB}$
$F = A \text{ MÁS } AB$	$F = A \text{ MÁS } AB \text{ MÁS } 1$
$F = A \text{ MÁS } B$	$F = A \text{ MÁS } B \text{ MÁS } 1$
$F = (A + \overline{B}) \text{ MÁS } AB$	$F = (A + \overline{B}) \text{ MÁS } AB \text{ MÁS } 1$
$F = AB \text{ MENOS } 1$	$F = AB$
$F = A \text{ MÁS } A^1$	$F = A \text{ MÁS } A \text{ MÁS } 1$
$F = (A + B) \text{ MÁS } A$	$F = (A + B) \text{ MÁS } A \text{ MÁS } 1$
$F = (A + \overline{B}) \text{ MÁS } A$	$F = (A + \overline{B}) \text{ MÁS } A \text{ MÁS } 1$
$F = A \text{ MENOS } 1$	$F = A$

Figura 12: Tabla de Activo en bajo

Funcionamiento

- 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

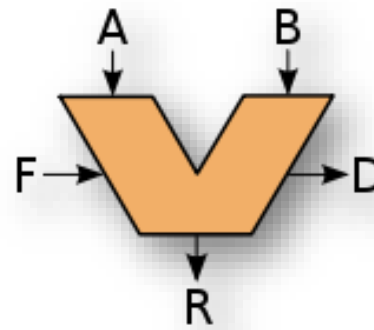


Figura 13:
Diagrama de ALU

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_logic_1164.all;
3.   Use ieee.std_logic_unsigned.all;
4.
5. Entity alur is port(
6.   Oper: in std_logic;
7.   X: in std_logic; --entrada a
8.   Y: in std_logic; --entrada b
9.   Bin: in std_logic; --acarreo
10.  Bout: out std_logic --acarreo salida
11.  S: out std_logic --salida
12. );
13. End alur;
14. Architecture arquí_alur of alur is
15. begin
16.   process
17.   Begin
18.     If oper = '0' then
19.       S <= x xor y xor bin;
20.       Bout <= (bin and (not(x xor y))) or (not(x) and y)
21.     End if;
22.   End process;
23. End arquí_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-funcionamiento-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>