

ALU en detalle

Unidad aritmética-lógica en detalle

Introducción

En algún mientras se hablaba de la CPU de la computadora, escuchaste sobre la Unidad Aritmética-Lógica. En este documento veremos qué es realmente una ALU.
Discutiremos sobre una ALU de 4 bits. Supondremos que los registros asociados y el conjunto de instrucciones también es de 4 bits.

Semisumador

Comencemos un simple medio sumador. **Semisumador** añade dos entradas A y B . añade dos salidas, sum (S) y carry (C). La señal de transporte representa un desbordamiento en el siguiente dígito de una suma de varios dígitos. Las siguientes figuras ilustran un medio sumador simple construido a partir de compuertas lógicas

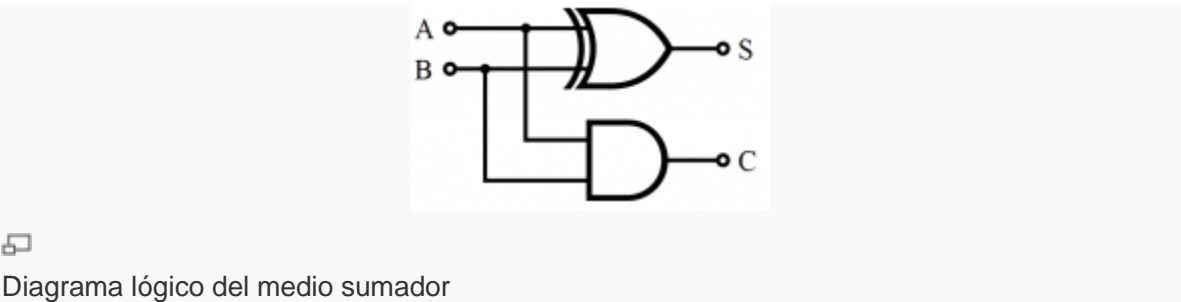
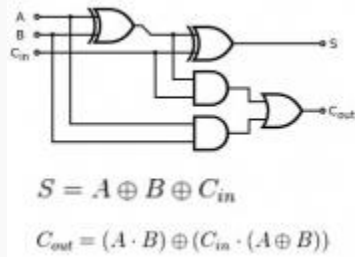


Diagrama lógico del medio sumador

| Entradas | | Salidas | |
|------------|-----------|----------|-----------|
| <i>UNA</i> | <i>si</i> | <i>S</i> | <i>do</i> |
| 0 0 | 0 0 | 0 0 | 0 0 |
| 1 | 0 0 | 1 | 0 0 |
| 0 0 | 1 | 1 | 0 0 |
| 1 | 1 | 0 0 | 1 |

Sumador completo

Full Adder (sumador completo) es una extensión de half adder (medio sumador) incluye también la entrada C_{in} . La tabla de verdad se puede implementar para formar el diagrama lógico como se muestra a continuación.



Sumador completo

| <i>A</i> | <i>B</i> | <i>Cin</i> | <i>Cout</i> | <i>S</i> |
|----------|----------|------------|-------------|----------|
| 0 0 | 0 0 | 0 0 | 0 0 | 0 0 |
| 1 | 0 0 | 0 0 | 0 0 | 1 |
| 0 0 | 1 | 0 0 | 0 0 | 1 |
| 1 | 1 | 0 0 | 1 | 0 0 |
| 0 0 | 0 0 | 1 | 0 0 | 1 |
| 1 | 0 0 | 1 | 1 | 0 0 |
| 0 0 | 1 | 1 | 1 | 0 0 |
| 1 | 1 | 1 | 1 | 1 |

La unidad básica: 1 bit ALU

Entonces, con los bloques de construcción anteriores, construimos una ALU simple que realice una operación aritmética (suma de 1 bit) y realice 3 operaciones lógicas, AND, NOR y XOR como se muestra a continuación. El multiplexor selecciona solo una operación a la vez. La operación seleccionada depende de las líneas de selección del multiplexor como se muestra en la tabla de verdad.

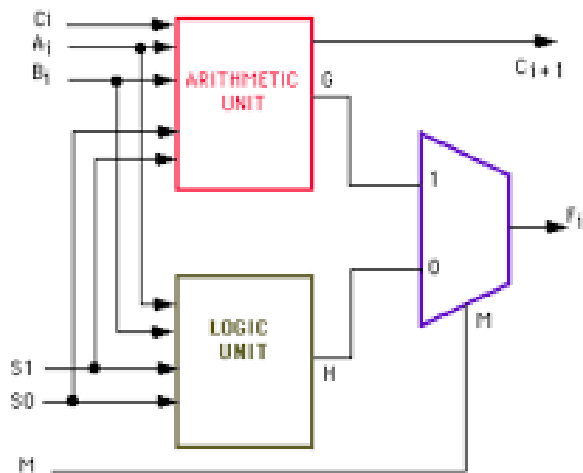


Figure 2: Block diagram of a bit-slice ALU

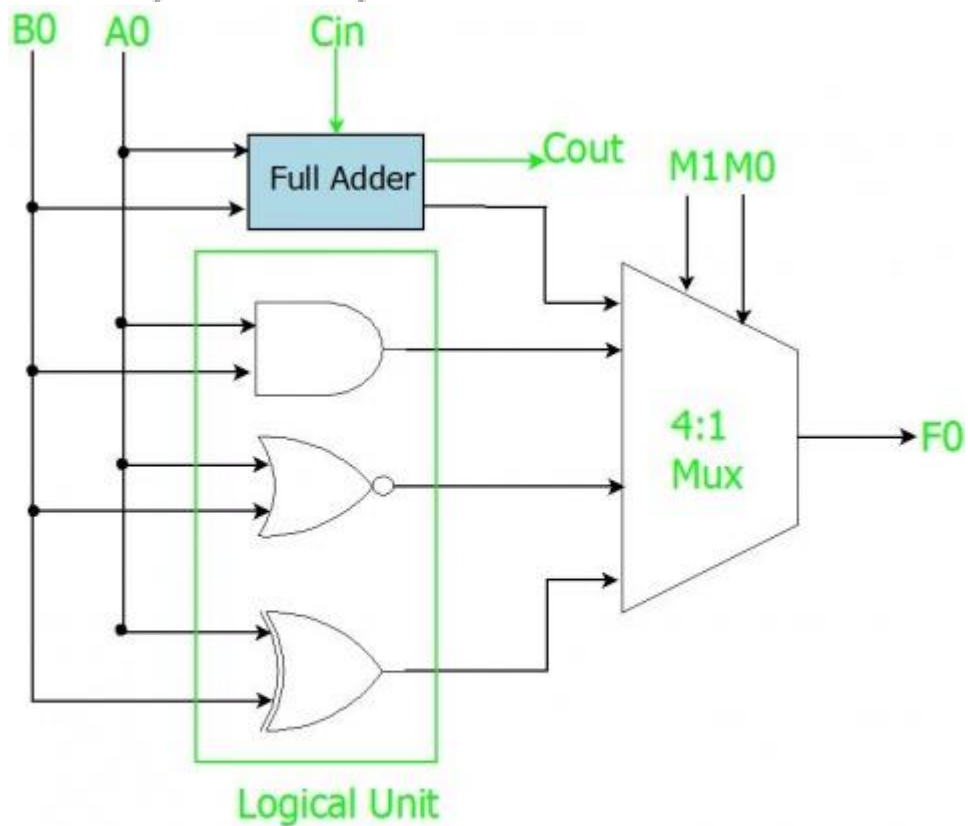


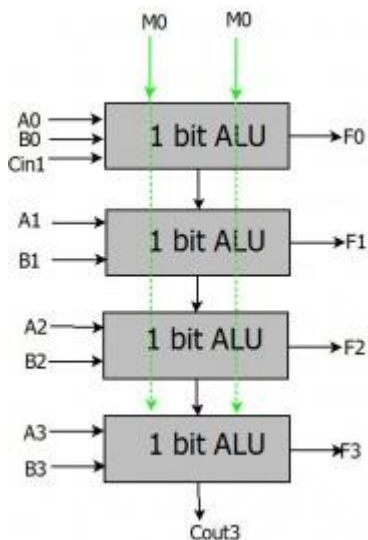
Figure: 1 bit ALU

| Entradas | | Salidas |
|----------|------|-----------|
| $M1$ | $M0$ | Operación |

| | | |
|-----|-----|------|
| 0 0 | 0 0 | SUMA |
| 1 | 0 0 | Y |
| 0 0 | 1 | O |
| 1 | 1 | XOR |

4 BIT ALU

Ahora podemos tomar la ALU de 1 bit como bloque y construir una ALU de 4 bits, que realice todas las funciones de la ALU de 1 bit en las entradas de 4 bits. Por lo tanto, un solo bloque de construcción se puede construir y utilizar de forma recursiva. Las entradas A y B son de cuatro bits y la salida también es de 4 bits. La siguiente figura lo ilustra:



Conclusiones importantes

Aquí hay algunas conclusiones importantes:

- Las líneas de selección **M0** y **M1** seleccionan la función que realiza ALU. Estas líneas de selección combinadas con los argumentos de entrada y las funciones deseadas pueden formarse un **conjunto de instrucciones**.
- Estas instrucciones pueden usarse para crear programas significativos. Dado que se requiere que estén fácilmente disponibles, se pueden almacenar en la unidad **ROM**.
- Los argumentos de entrada **A** y **B** a menudo se almacenan en **registros internos**. Estos junto con otros registros de propósito especial forman los **registros** del microcontrolador.
- Las memorias ROM son más lentas en velocidad, por lo tanto, a menudo se usa una **RAM** intermedia de alta velocidad.

- Todos los tiempos críticos, la decodificación de las instrucciones a menudo se agrupan en una ***unidad de control y tiempos*** separada '
- Si un Microcontrolador se construyera solo a partir de ALU, RAM, ROM, no habría ninguna interfaz externa. Por lo tanto, tenemos puertos IO de ***entrada / salida*** .
- Se incorporan características adicionales como ' ***interrupciones, protocolos de comunicación, EEPROM, temporizadores / contadores, interfaces de depuración, etc.*** para completar un controlador.

En la discusión anterior, podríamos haber omitido detalles intrincados involucrados en un ALU, diseño de CPU. Pero el objetivo era entender ALU / CPU a un nivel más profundo.