

Erick Enrique Castro Espinosa

Preguntas

1. Operaciones logicas: por lo menos NOT y AND o NOT y OR, ya que son conjuntos logicos completos, pero si tuvieras NOT, OR y AND seria mas eficiente ya que no tendrias que hacer las equivalencias logicas y entre mas operaciones logicas tenga el procesador pues seria mas rapido pero sin descuidar el espacio que necesitan, pero las basicas serian NOT y AND u OR.

Operaciones aritmeticas: la suma por lo menos y si la resta esta seria mejor, podriamos llegar a considerar como operación basica la multiplicacion y la division pero ya son mas complejas, pero la suma y podria ser que la resta son operaciones basicas.

2. Porque el acarreo de la suma pasa por 2 compuertas por cada ALU de 1 bit hasta llegar a la salida. Entonces eso es ineficiente ya que si hubieramos echo una ALU mas grande el acarreo pasaria por $2n$ compuertas, con n el numero de ALU's de 1 bit. Un sumador de tipo acarreo anticipado, ya que asi podemos saber si hay acarreo al final con una formula.

3. Bajo este diseño de ALU si se realiza trabajo inutil ya que todo el tiempo se calculan todas las operaciones pero dependiendo cual queramos usar gracias a los multiplexores solo dejamos salir un resultado de todas las operaciones que realiza la ALU, aunque como se hace simultaneamente no se pierde tiempo ya que todas se hacen a la ves, pero cuesta mas esta ALU ya que los circuitos correspondientes para cada operación estan utilizando energia todo el tiempo y como solo sale el resultado de una operación entonces la energia que utilizan las demas operaciones se gastan en vano.

4. Con el diseño de la ALU de 8 bits final se le pueden agregar 2 operaciones mas, y pues dependiendo que operaciones se modifica la ALU final o las de 1 bit, depende si las operaciones nuevas se pueden realizar con las que ya se tienen o son nuevas.