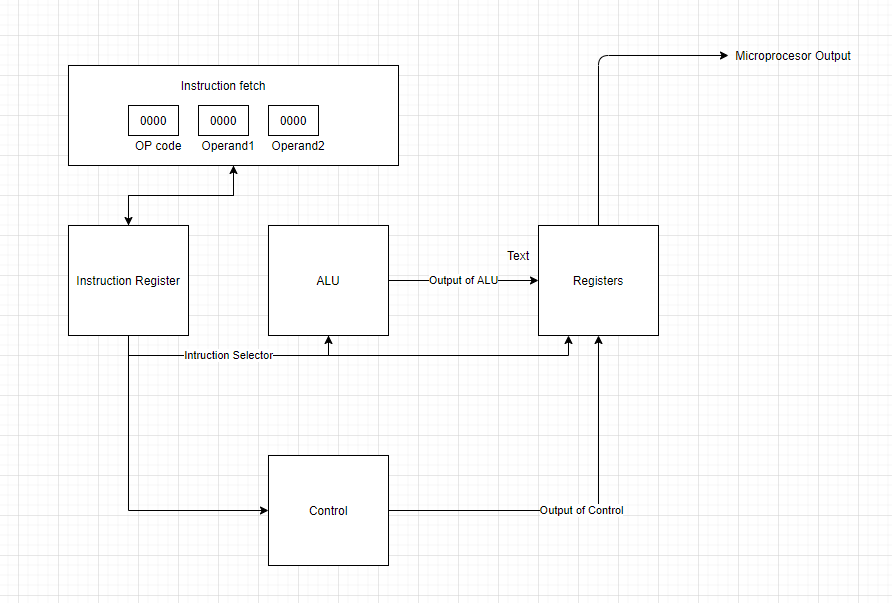
**David Corzo – 20190432**

**1) Construya un diagrama de bloques con los componentes básicos de un microprocesador.**



**2) ¿Qué es y para qué se usa el algebra de Boole?**

El algebra de Boole se utiliza para representar expresiones de una tabla de verdad y para luego poder ser implementadas en forma de un circuito. El algebra de Boole permite también identificar posibles simplificaciones al circuito que no se notarían tan fácilmente si no se implementara una expresión booleana algebraica.

**3) Comente sobre los objetivos que se buscan alcanzar con la minimización de lógica combinacional al momento de diseñar un circuito.**

Los objetivos son:

* Lograr mitigar el calor generado ahorrándole a la circuitería tener que hacer operaciones no necesarias.
* Dar más velocidad al circuito por que si se utiliza la implementación que requiere mínimas operaciones se tarda menos en dar un resultado.
* Disminuir la complejidad del circuito para ser implementado fácilmente.

**4) ¿Cómo funciona un transistor?**

Un transistor consiste en una base, colector y salida, si hay electrones o corriente en la base la salida tira un 1 lógico, de lo contrario tira 0 lógico. Funciona por medio de moleculas N-Type y P-Type una tiene un electrón más y otra le falta un electrón. Al agrupar el circuito correctamente podemos manipular por medio de 'precedencia' qué señal tirará en la salida.

**5) ¿Cuáles son los pasos a seguir para el diseño de un circuito digital?**

Los pasos para seguir son básicamente los siguientes:

1. Identificar las entradas (cantidad y etiquetarlas respectivamente)
2. Identificar las salidas (cantidad y etiquetarlas respectivamente).
3. Plantear la tabla de verdad. Dos métodos.
   1. Se deben de tomar las entradas correspondientes, en otras palabras, las que tienen salida como 1 lógico. Si hay muchas salidas se toma cada una por separado y se construye una expresión booleana algebraica.
   2. Se deben de tomar las entradas e ingresarlas a un mapa de Karnaugh, tras hacer esto se identifican los rectángulos verticales y horizontales y se construye una expresión booleana algebraica.
4. Al tener la expresión booleana algebraica es importante ver y verificar que esté en la forma más simplificada, pues si no lo está el circuito tendrá que hacer mas operaciones de las que absolutamente debe y puede empezar a calentar.
5. Cuando ya se aseguró que este en su forma más simplificada lo puede implementar con la expresión booleana, la suma se hace con compuertas OR, multiplicación con AND, expresiones XOR con la puerta XOR, etcétera.
6. Finalmente debe de chequear que se haya conectado bien todo y que el circuito esté tirando en las salidas lo que se planteó en la tabla de verdad.

**6) ¿Cómo funciona un flip flop?**

Un flip flop consiste de una entrada de reloj, una entrada de un bit que representa el dato que queremos guardar llamado usualmente D, un write enable (en algunos) llamado WE, una salida Q y ~Q. la idea es que las salidas Q representan el bit que deseamos almacenar, muy similar a un Gate Latch un flip flop tiene la diferencia que permite ser sincronizado con el pulso de un reloj. Entonces cuando el flip flip sensa el 'clockazo' y el write enable está activo, deja pasar lo que este en la entrada D a ser almacenado en Q. Así se mantiene un estado, el reloj sirve para mitigar el efecto Dalton que ocurre con la circuitería que los electrones tardan en viajar y para eso se necesita un flip flop para que todo esté sincronizado al pulso del reloj y que le de tiempo a la circuitería de mandar señales a sus respectivos destinos.

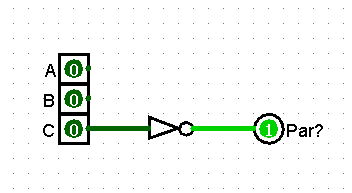
**7) Diseñe un circuito que solicite un número al usuario (3 bits) y que indique si el número es par o impar.**

**Tabla: (\*Según internet cero es par)**

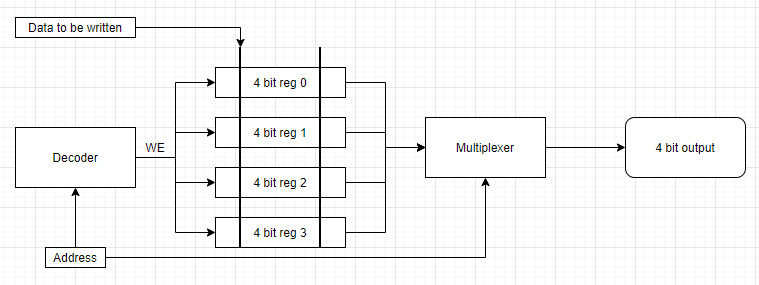
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **A** | **B** | **C** | **Out (par=1)** |
| **0** | **0** | **0** | **0** | **1** |
| **1** | **0** | **0** | **1** | **0** |
| **2** | **0** | **1** | **0** | **1** |
| **3** | **0** | **1** | **1** | **0** |
| **4** | **1** | **0** | **0** | **1** |
| **5** | **1** | **0** | **1** | **0** |
| **6** | **1** | **1** | **0** | **1** |
| **7** | **1** | **1** | **1** | **0** |

**Simplificación:**

**Implementación:**

****

**8) Muestre el diagrama de bloques de cómo se vería un Register File de 4 registros de 4 bits.**

****

**9) Investigue sobre el circuito “Carry-lookahead adder”, y compare sus ventajas o desventajas respecto al “Full-Adder”**

La ventaja de un Carry-lookahead adder es el componente 'Carry lookahead logic' que permite que el carry-lookahead adder no tenga que esperar a recibir el carry out de cada sumador en cascada sino determinar cuál va a ser el carry out antes o al mismo tiempo que termina de sumarse, esto permite que se tenga un sumador más rápido en ejecución. Sin embargo el problema con este tipo de sumador es que involucra más lógica para implementarse. El full adder debe de hacer operaciones en cascada, lo que implica que cada sumador debe de esperar el resultado del anterior por que debe de recibir el carry, esto lo hace más lento que el carry-lookahead adder. La ventaja es que uno tiene más rápida ejecución con el costo de tener más lógica y el otro tiene menos lógica y es más lento.

**10)Desarrolle un tema de los vistos en clase y que no se haya tocado en las preguntas anteriores. (“Pitch tecnológico”)**

La RAM, consiste en 8 registros alambrados en paralelo. Cada registro almacena 1 solo bit y este arreglo es lo que permite que todo un byte se escriba en paralelo, dando esa característica de rapidez a la RAM que ningún otro componente de memoria tiene de la misma manera. Además de esto curiosamente puede ser que los bytes alocados no estén consecutivos en cada registro puesto a que se opera como una matriz de bits que juntos hacen un byte en lugar de filas y de bytes.