

Laboratorio 3: Lógica combinacional y Aritmética I

Fiorella Delgado León Jonathan Guzmán Araya, Gerald Valverde McKenzie

Instituto Tecnológico de Costa Rica

Área Académica Ingeniería en Computadores

fio@gmail.com jonathana1196@gmail.com gerald@gmail.com

Abstract—

Index Terms—ALU, compuertas lógicas, flip-flops, microprocesadores

I. INTRODUCCIÓN

Los circuitos combinacionales son aquellos en los que las salidas solo dependen solamente de las entradas, y no de ningún tipo de sincronización con señales de reloj, esto hace que los sistemas combinacionales sean generalmente rápidos. En sistemas digitales complejos, como los microprocesadores, los circuitos de lógica combinacional desempeñan un papel fundamental. La arquitectura de un microprocesador es como la que se observa en la Figura 1.

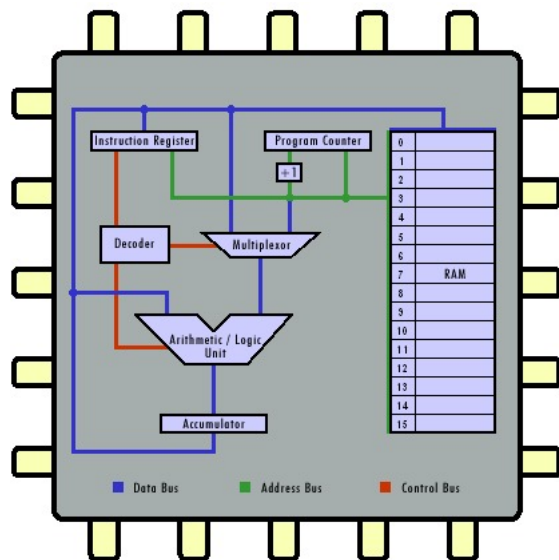


Fig. 1. Arquitectura básica de un microprocesador [1]

Una función esencial de muchas computadoras (microprocesadores) y calculadoras es la realización de operaciones lógicas y aritméticas. Estas operaciones se efectúan en la unidad aritmética-lógica de una computadora, donde se combinan compuertas lógicas y flip-flops de manera que puedan sumar, restar, multiplicar y dividir números binarios [2]. El símbolo de una ALU es como el de la Figura 2.

Además de los operadores lógicos, la ALU cuenta con una serie de registros para almacenar los datos y bits de información sobre los resultados, también llamados banderas [3], las banderas más comunes en una ALU son: Z, C, V, N.

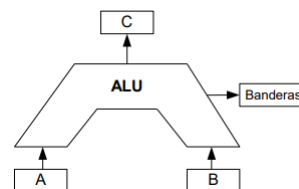


Fig. 2. Símbolo ALU [3]



Fig. 3. Compuerta NOT y su representación en álgebra de Boole [2]

- Zero flag: usada para determinar si dos valores son iguales, generalmente se usa una resta, y se revisa si el resultado es cero, si es así, esta bandera se enciende.
- Overflow flag: usada para determinar si el resultado de una operación (usualmente suma o multiplicación), tiene un tamaño superior al máximo representable, o inferior al mínimo representable en la ALU, es posible que existan como banderas separadas.
- Negative flag: usada para determinar si el resultado de una resta es menor a cero
- Carry/Borrow flag: Esta bandera indica cuando una operación resulta en un valor más largo del cual el acumulador puede representar o menor del cual el acumulador puede representar.

La ALU es simplemente un operador, es decir solo realiza operaciones, esta no toma decisiones, acepta datos binarios que están almacenados en la memoria y ejecuta operaciones con estos datos, de acuerdo a las instrucciones que vienen de la unidad de control.

Para realizar operaciones lógicas la ALU utiliza compuertas lógicas como:

- NOT: La Figura 3 muestra el símbolo para un circuito NOT, la operación NOT cambia de un nivel lógico al nivel lógico opuesto. Cuando la entrada está a nivel ALTO (1), la salida se pone a nivel BAJO (0). Cuando la entrada está a nivel BAJO, la salida se pone a nivel ALTO [4].
- AND: La Figura 4 muestra el símbolo lógico para una compuerta AND de dos entradas. La operación AND genera un nivel ALTO sólo cuando todas las entradas están a nivel ALTO, para el caso de dos entradas. Cuando

Q	Q'
0	1
1	0

TABLE I

TABLA DE VERDAD DE LA COMPUERTA NOT

una entrada está a nivel ALTO y la otra entrada está a nivel ALTO, la salida se pone a nivel ALTO. Cuando cualquiera de las entradas o todas ellas están a nivel BAJO, la salida se pone a nivel BAJO [4].

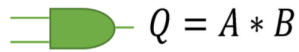


Fig. 4. Compuerta AND y su representación en álgebra de Boole [2]

A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

TABLE II

TABLA DE VERDAD DE LA COMPUERTA AND

- OR: La Figura 5 muestra el símbolo lógico para una compuerta OR de dos entradas. La operación OR genera un nivel ALTO cuando una o más entradas están a nivel ALTO, para el caso de dos entradas. Cuando una de las entradas está a nivel ALTO o ambas entradas están a nivel ALTO, la salida es un nivel ALTO. Cuando ambas entradas están a nivel BAJO, la salida será un nivel BAJO [4].



Fig. 5. Compuerta OR y su representación en álgebra de Boole [2]

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

TABLE III

TABLA DE VERDAD DE LA COMPUERTA OR

- XOR: La Figura 6 muestra el símbolo lógico para una compuerta XOR de dos entradas. En una compuerta OR-exclusiva, la salida X es un nivel ALTO si la entrada A está a nivel BAJO y la entrada B está a nivel ALTO; o si la entrada A está a nivel ALTO y la entrada B está a nivel BAJO; X es un nivel BAJO si tanto A como B están a nivel ALTO o BAJO [4].



Fig. 6. Compuerta XOR y su representación en álgebra de Boole [2]

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

TABLE IV

TABLA DE VERDAD DE LA COMPUERTA XOR

Uno de los problemas más desafiantes en el diseño de circuitos es el tiempo, hacer que un circuito funcione rápido. Una salida necesita tiempo para cambiar en respuesta a un cambio en la entrada, como se observa en la Figura 7 [5].

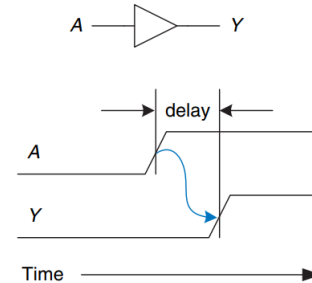


Fig. 7. Tiempo de respuesta de una señal [5]

En la lógica combinacional existen dos tipos de tiempos, tiempo de propagación y tiempo de contaminación. El tiempo de propagación t_{pd} es el tiempo máximo desde que la entrada cambia hasta que el o las salidas llegan a su valor final y el tiempo de contaminación t_{cd} es el tiempo mínimo que dura la entrada en cambiar hasta que alguna salida empiece a cambiar su valor [5].

Además de los tiempos de propagación y contaminación en la electrónica de circuitos existe algo llamado la ruta crítica, que es la ruta más larga que tiene una entrada, hablando de circuitos combinacionales, para llegar a la salida, por lo tanto, esta ruta también sería la más lenta. Ahora para describir a un circuito hay dos factores muy importantes: latencia y tasa de transferencia. La latencia funcionaría como la frecuencia del circuito, ya que es el tiempo que se necesita para que el cambio en la entrada haga un cambio en la salida; esta puede ser expresada en tiempo o, para circuitos sincrónicos, un cierto número de ciclos de reloj. Y la tasa de transferencia a la velocidad con la que la información puede ser procesada. Por ejemplo, en la Figura 8 se tienen tres rutas distintas desde una entrada hasta la salida, siendo la ruta azul la más lenta de estas.

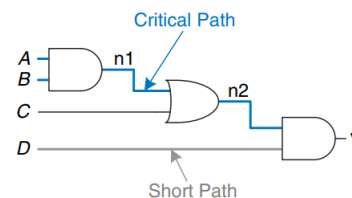


Fig. 8. Ruta crítica [5]

Así el tiempo de propagación de un circuito combinacional es la suma de los tiempos de propagación de cada elemento

que lo compone como se observa en la Ecuación 1 y el tiempo de contaminación es la suma de los tiempos de contaminación de la más corta del circuito combinacional que se puede observar en la Ecuación 2.

$$t_{pd} = 2t_{pdAND} + t_{pdOR} \quad (1)$$

$$t_{pd} = 2t_{pdAND} + t_{pdOR} \quad (2)$$

En circuitos más complejos como un procesador pipeline como el de la Figura 9, la velocidad de ciclos del reloj decrece haciendo retrasos en la ruta crítica, para esto se utiliza la técnica de Pipelining, el cual consiste en dividir el circuito en varias partes con un registro al final de cada una, de esta manera se divide la ruta crítica en pequeñas rutas, permitiendo que la velocidad del reloj aumente y consecuentemente también lo haga la tasa de transferencia. La segmentación consiste en descomponer la ejecución de cada instrucción en varias etapas para poder empezar a procesar una instrucción diferente en cada una de ellas y trabajar con varias a la vez. Las etapas de segmentación pueden ser: IF (búsqueda), ID (decodificación), EX (ejecución), MEM (memoria), WB (escritura), en una arquitectura relativamente sencilla.

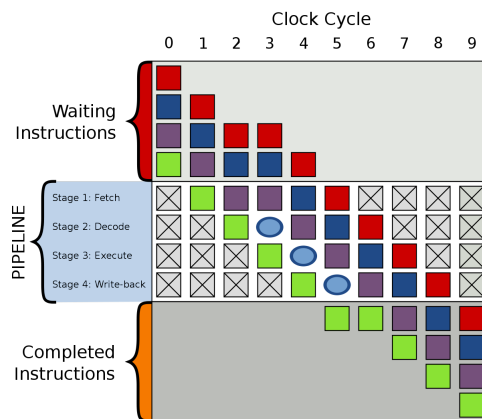


Fig. 9. Procesador Pipeline [6]

La frecuencia máxima de operación de un circuito está determinada por la ruta crítica del mismo, para esto se toma la ruta crítica (en un procesador se realiza tomando la instrucción que tarda más en ejecutarse por completo sin afectar a las otras) y se calcula el tiempo que tarda en ciclos de reloj T_c , como se observa en la Ecuación 3, esta frecuencia se mide en Hz .

$$f_c = \frac{1}{T_c} \quad (3)$$

II. CONCLUSIONES

III. BIBLIOGRAFÍA

REFERENCES

- [1] A. J. F. González. (2017, Jul.) Arquitectura de microprocesadores desde el principio. beBee. [Online]. Available: <https://www.bee.com/producer/@http-alfredo-j-feijoo-webnode-es-contacto/arquitectura-de-microprocesadores-desde-el-principio>

- [2] G. M. Tocci, Ronald J. Neal S. Widmer, *Sistemas digitales: principios y aplicaciones*, L. M. C. Castillo, Ed. Pearson Educación, 2007.
- [3] D. A. G. García. Procesadores. Departamento de Mecatrónica. [Online]. Available: <http://homepage.cem.itesm.mx/garcia.andres/PDF201411/ArquitecturaComputacional.pdf>
- [4] T. L. Floyd, *Fundamentos de Sistemas Digitales*, M. Martin-Romo, Ed. Pearson Educación, 2006.
- [5] D. M. H. Sarah L. Harris, *Digital Design and Computer Architecture*. Morgan Kauffmann, 2010.
- [6] Arquitectura en pipeline (informática). Wikipedia. [Online]. Available: [https://es.wikipedia.org/wiki/Arquitectura_en_pipeline_\(inform\unhbox\voidb@x\bggroup\let\unhbox\voidb@x\setbox\@tempboxa\hbox{a\global\mathchardef\accent@spacefactor\spacefactor}\let\begin\group\end\group\relax\let\ignorespaces\relax\accent19a\egroup\spacefactor\accent@spacefactor\ignorespaces\)](https://es.wikipedia.org/wiki/Arquitectura_en_pipeline_(inform\unhbox\voidb@x\bggroup\let\unhbox\voidb@x\setbox\@tempboxa\hbox{a\global\mathchardef\accent@spacefactor\spacefactor}\let\begin\group\end\group\relax\let\ignorespaces\relax\accent19a\egroup\spacefactor\accent@spacefactor\ignorespaces))