INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

No. 119 **Serie:** VERDE **Fecha:** Mayo 2007

Implementación de Sistemas Digitales Básicos mediante Neuronas de Tipo McCullough-Pitts

Antonio Hernández Zavala[1]
Cornelio Yáñez Márquez[2]
Oscar Camacho Nieto[2]
Octavio López Leyva[1]

RESUMEN

En este trabajo se introduce un método para modelar circuitos lógicos combinacionales, haciendo uso de las funciones lógicas básicas como AND, OR y NOT, mismas que son implementadas con bloques de neuronas simples de tipo McCullough-Pitts. Al concatenar estos bloques básicos, se logra realizar la función lógica requerida. Finalmente, como ejemplo, se presentan un circuito medio sumador, un sumador completo y un comparador binario de cuatro bits.

Esta publicación es un producto de los proyectos de investigación desarrollados en el Centro de Investigación en Computación del IPN.

Palabras Clave: Operación Lógica, Entrada Inhibitoria, Entrada Excitatoria, Umbral, Neurona McCullough-Pitts NMCP.

Agradecimientos: Los autores agradecen el apoyo económico recibido para la realización de este trabajo, a las siguientes instituciones: CIC-IPN, SIP-IPN y CONACYT.

[1]Estudiante del Doctorado en Ciencias de la Computación del CIC-IPN Antonio_z@hotmail.com
[2]Profesor-Investigador del CIC-IPN cyanez@cic.ipn.mx, oscarc@cic.ipn.mx

Instituto Politécnico Nacional
Centro de Investigación en Computación
Av. Juan de Dios Bátiz casi esq.
Miguel Othón de Mendizabal Ote.
México, 07738, D.F.
ISBN 10: 970-36-0407-2
ISBN 13: 978-970-36-0407-4
Impreso en México

ADVERTENCIA

Este reporte contiene información desarrollada por el Centro de Investigación en Computación del Instituto Politécnico Nacional a partir de datos y documentos con derechos de propiedad y por lo tanto su uso queda restringido a las aplicaciones que explícitamente se convenga.

La aplicación no convenida exime al Centro su responsabilidad técnica y da lugar a las consecuencias legales que para efecto se determinen.

Información adicional sobre este reporte podrá obtenerse acudiendo a la Unidad de Publicaciones y Reportes Técnicos del Centro de Investigación en Computación.
Av. Juan de Dios Batiz s/n en la unidad profesional Adolfo López Mateos.
Teléfono 5729-6000 ext. 56571

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1. Introducción

En este trabajo se presenta, de una manera simple, el funcionamiento de neuronas de tipo McCulloch-Pitts [1], dichas neuronas, fueron el primer tipo de neurona artificial propuesto, por tal motivo, su funcionamiento es muy simple y a partir de los lineamientos establecidos para la operación de las mismas, se pueden generan bloques que realicen las funciones lógicas básicas como lo son AND, OR y NOT. Aplicando cierta heurística y con el propósito de demostrar que redes neuronales sencillas son capaces de representar adecuadamente las leyes lógicas fundamentales, en el presente trabajo, son conectadas entre si mediante sinapsis excitatorias e inhibitorias y posteriormente es asignando un valor umbral para la activación de la unidad de salida.

Para el desarrollo de la propuesta, primeramente se dará un breve resumen del funcionamiento de las neuronas McCulloch-Pitts (MCP), en donde se abordará el esquema físico y matemático que siguen estas neuronas para su operación. Posteriormente, a manera de ejemplo, se muestran neuronas MCP que sigan el comportamiento de las operaciones lógicas básicas mencionadas. Además, haciendo uso de estas operaciones lógicas básicas, y siguiendo prácticamente la misma metodología de diseño que en circuitos lógicos combinacionales, pasando por la creación de una tabla de verdad, de donde se puedan obtener una serie de ecuaciones que describan el comportamiento del circuito, para finalmente ser implementadas, realizando un medio sumador y un sumador completo.

Por último, se presenta un comparador binario de cuatro bits utilizando neuronas del tipo McCullough-Pitts (MCP), en donde, se sigue la misma metodología que para los sumadores, este tipo de comparador provee de tres salidas una para cada caso A>B, A<B, A=B.

2. Neuronas McCulloch-Pitts

2.1 Neuronas Biológicas.

A grandes rasgos, una neurona biológica, es una célula especializada capaz de propagar una señal. Tiene entadas o dendritas, un cuerpo celular y una estructura que propaga la salida conocido como axón, mismo que sirve para conectarse con las dendritas de otra neurona mediante una sinapsis. Cuando la neurona es activada, propaga el resultado hacia las otras neuronas a través de las sinapsis. En la figura No. 1, se muestra un esquema de neurona biológica, misma que inspiro a los autores de MCP para lograr representar un modelo matemático de esta.

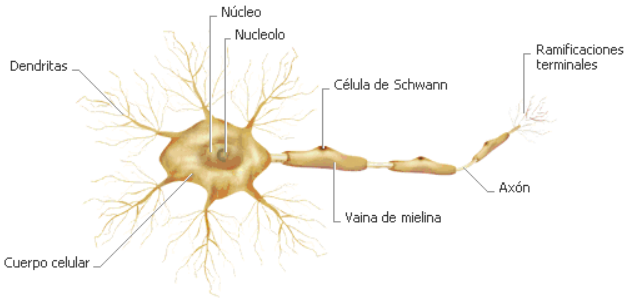

Figura 1. Neurona Biológica

2.2 Modelo de McCulloch-Pitts.

En el año de 1943 un par de ingenieros eléctricos, el neuropsicólogo Warren McCulloch, y el matemático Walter Pitts, publicaron un articulo titulado: "A logical calculus of the ideas immanent in nervous activity" en el *Bulletin of Mathematical Biophysics* 5:115-133. En este documento, se trata de explicar la forma en que el cerebro puede producir patrones de comportamiento complejos basado en celdas básicas interconectadas entre si. A estas unidades básicas se les asigno el nombre de neuronas y se representaron de una manera muy simple, a este modelo lo llamaron MCP, y al conjunto de varias MCP's se le da el nombre de red neuronal artificial, este modelo ha sido de gran importancia para la ciencia computacional.

En una red MCP existen dos tipos de entradas: Excitatorias (e_i) e Inhibitorias (i_j). Las primeras, son señales que se encargan de dar estímulos en la neurona de modo que al ser comparadas con un patrón o umbral θ, se produzca una salida (y). Por otro lado en las entradas inhibitorias, como su nombre lo indica, al ser recibida una señal de estas, la neurona se inhibe y no produce ningún resultado, es decir, su salida es cero. Al utilizar valores para cada tipo de entrada, 1 para un estimulo y 0 para que no pase nada, la neurona realiza un calculo con todas sus entradas y si el resultado es idéntico al umbral establecido entonces se genera una salida 1, en caso contrario la salida será 0.

Las neuronas MCP operan según lo siguiente:

1. Son dispositivos binarios (0,1)
2. Cada neurona tiene un patrón fijo o umbral θ.
3. La neurona recibe entradas excitatorias con los mismos pesos.
4. Las entradas inhibitorias tienen poder absoluto sobre las entradas excitatorias, es decir, garantizan una salida 0.
5. El proceso es síncrono, es decir, en cada ciclo las neuronas son simultáneamente actualizadas por los cambios en la entrada estableciendo una salida.
6. La estructura de la red de neuronas no cambia con el tiempo.

En resumen:

$$y = \begin{cases} 1 \; si \left(\left(\sum_{j=1}^{m} i_j = 0 \right) \wedge \left(\sum_{i=1}^{n} e_i \geq 0 \right) \right) \\ \\ 0 \qquad en \; otro \; caso \end{cases} \qquad (1)$$

Donde:

$$n, m \in Z^+, \; y, e_i, i_j \in \{0,1\}; \;\; i = 1, 2, ..., n; \; j = 1, 2, ..., m$$

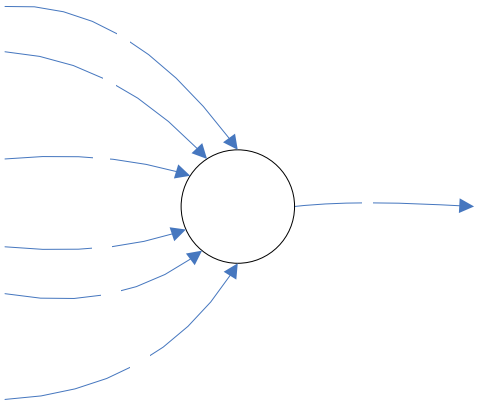

Figura2. Neurona McCullough-Pitts.

De acuerdo con la figura No. 2, pueden existir mas de una entrada excitatorias e inhibitorias, que al comparar la suma de estas con el umbral, se genera una salida *y*. Para una neurona con dos entradas, se tienen 4 posibles combinaciones en los datos de entrada, entonces para una neurona con dos entradas existen a lo más 16 MCP distintos para generar un 0 o 1 a la salida de acuerdo con los operadores lógicos.

3. Ejemplos de Aplicación.

3.1 Operaciones Lógicas Básicas.

Como se menciono anteriormente, es necesario contar con las operaciones lógicas básicas AND, OR y NOT, si se requiere construir circuitos lógicos. Entonces haciendo uso de la definición de neurona MCP de la sección anterior, se procede a construir los bloques mencionados.

3.1.1 Operación AND

Para el caso de la operación AND y de acuerdo a su tabla de verdad (ver Tabla No. 1), solamente existe un caso en el que la salida es igual a 1, y esto es cuando las dos entradas son iguales a uno es decir, la suma de estas es igual a dos, por lo que se opta por elegir $\theta = 2$ como umbral. Al observar la figura No. 3, se puede apreciar que en este caso, una entrada inhibitoria no es necesaria, por lo que A y B son excitatorias.

Tabla 1. Operación AND.

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

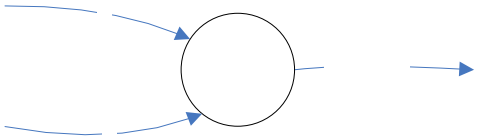

Figura 3. Neurona MCP para la operación AND.

3.1.2 Operación OR

Inversamente al caso anterior y de acuerdo con su tabla de verdad (ver Tabla No. 2), solamente existe un caso en el que la salida es igual a 0, y esto es cuando las dos entradas son iguales a cero, es decir, la suma de estas es igual a cero, por lo que se elige $\theta = 1$ como umbral, con esto se asegura que con solo una entrada que tenga valor 1, la salida sea 1. En este caso, una entrada inhibitoria tampoco es necesaria, por lo que A y B son igualmente excitatorias (ver figura No. 4).

Tabla 2. Operación OR.

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

A

B

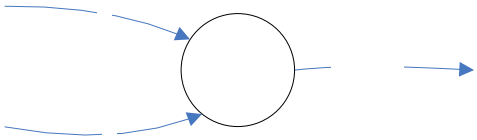

Figura 4. Neurona MCP para la operación OR.

3.1.3 Operación NOT

Como se muestra en la tabla No. 3, este caso es distinto, dado que lo que se pretende es obtener la inversa de la entrada, entonces se opta por colocar una entrada de tipo inhibitoria, que funcionara de la siguiente manera: se escoge un umbral $\theta = 0$, entonces, para el caso donde haya un 1 en la entrada, dado que esta es inhibitoria, generara un cero a la salida. Para el caso cuando la entrada es igual a cero, no hay inhibición, entonces se alcanza el umbral y la salida obtenida es igual a 1 (ver figura No. 5).

Tabla 3. Operación NOT.

A	NOT A
0	1
1	0

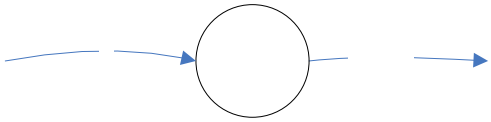

Figura 5. Neurona MCP para la operación NOT.

3.2 Medio Sumador.

En la tabla No. 4, se muestra al funcionamiento de un medio sumador, en donde A y B son las entradas y el resultado en la salida esta dado por:

$$S\ \overline{A}B + A\overline{B} \qquad (2)$$

Al simplificar se tiene

$$S = A \oplus B \qquad (3)$$

De lo que podrá notarse que es idéntico al operador Si y solo Si o XOR. Note en la figura No. 6, que existe un arco etiquetado con *i*, mismo que representa una entrada inhibitoria.

Tabla 4. Medio sumador.

A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

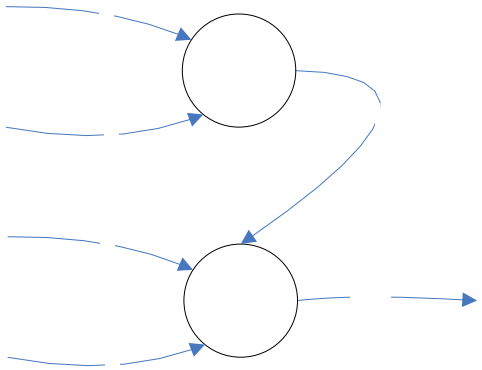

Figura 6. Diagrama MCP para un medio sumador.

3.3 Sumador Completo.

Para el caso de un sumador completo, en donde se incluyen los acarreos tanto de entrada como de salida, mismos que modifican el resultado mostrado anteriormente. A continuación se muestran la tabla de verdad (ver tabla No. 5) y el resultado correspondiente a las ecuaciones para ambas salidas después de simplificar para el acarreo de salida C_{out} y la suma S, donde se obtienen las ecuaciones (4) y (5).

Tabla 5. Sumador Completo.

A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$S = A \oplus (B \oplus C_{in}) \qquad (4)$$

$$C_{out} = BC_{in} + A(B \oplus C_{in}) \qquad (5)$$

En la figura No. 7, se muestra el diagrama correspondiente al sumador completo implementado con neuronas MCP, similar al caso anterior, los arcos etiquetados con *i* son entradas inhibitorias, mientras el resto son de tipo excitatorio.

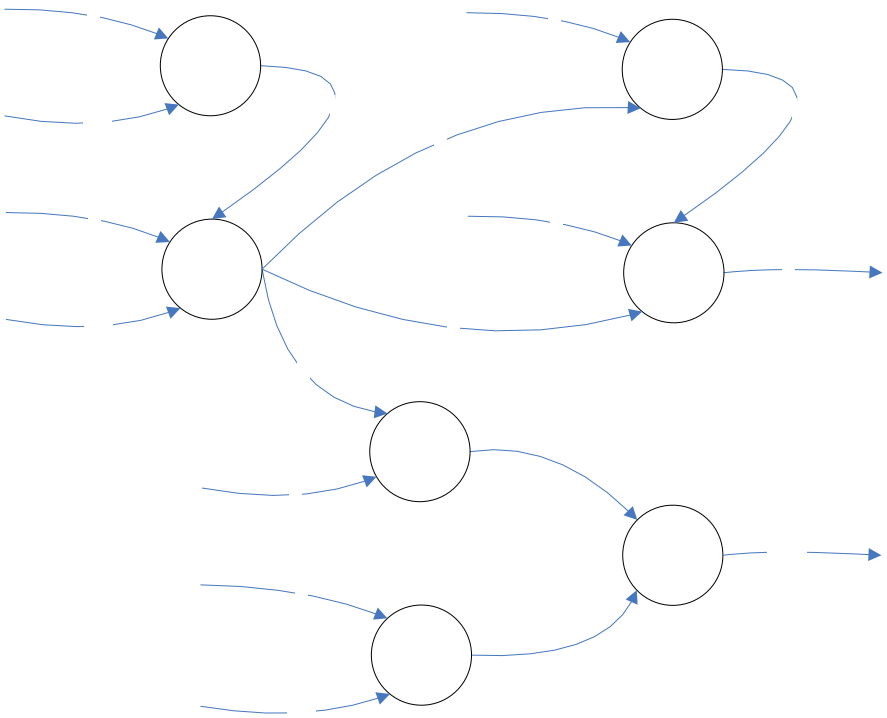

Figura 7. Sumador Completo.

B

4. Comparador Binario de Cuatro Bits.

0=2

En la tabla No. 6, puede apreciarse el funcionamiento del comparador binario. Los bits más significativos, son los que predominan de manera total sobre que es lo que pasará con el resultado, para el caso de que ambos valores sean diferentes. Note la propagación jerárquica según el peso de cada bit, las casillas indicadas con X significan que no importa el valor [3].

i

Cin

B

Tabla 6. Operación del comparador binario.

A_3,B_3	A_2,B_2	A_1,B_1	A_0,B_0	A=B	A<B	A>B
$A_3>B_3$	X	X	X	0	0	1
$A_3<B_3$	X	X	X	0	1	0
$A_3=B_3$	$A_2>B_2$	X	X	0	0	1
$A_3=B_3$	$A_2<B_2$	X	X	0	1	0
$A_3=B_3$	$A_2=B_2$	$A_1>B_1$	X	0	0	1
$A_3=B_3$	$A_2=B_2$	$A_1<B_1$	X	0	1	0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0>B_0$	0	0	1
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0<B_0$	0	1	0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	1	0	0

Para el caso en donde los valores sean iguales, es única la opción.

A partir de la tabla No. 6, y para cada salida 1, se obtienen las siguientes ecuaciones en suma de productos para cada caso:

$$A = B,\ A_3B_3 + A_2B_2 + A_1B_1 + A_0B_0 + \overline{A_3}\,\overline{B_3} + \overline{A_2}\,\overline{B_2} + \overline{A_1}\,\overline{B_1} + \overline{A_0}\,\overline{B_0} \qquad (6)$$

$$A > B,\ A_3\overline{B_3} + A_2\overline{B_2} + A_1\overline{B_1} + A_0\overline{B_0} \qquad (7)$$

$$A < B,\ \overline{A_3}B_3 + \overline{A_2}B_2 + \overline{A_1}B_1 + \overline{A_0}B_0 \qquad (8)$$

Una vez llegando a este punto, se hace la analogía con las neuronas, este circuito es relativamente simple y la idea es comprobar lo que en aquel entonces mencionaron los autores acerca de sus neuronas, "*Este tipo de neuronas, pueden utilizarse para representar cualquier función lógica*".

Para el primer caso (6), esta claro que la salida se activará solamente cuando todos los valores son idénticos. Para el segundo (7) y tercer casos (8), las ecuaciones aparentemente son correctas, pero no es así, dado que en mas de un caso las salidas A<B y A>B pueden estar activas simultáneamente, debido a que un valor contenga bits distintos, tanto mayores como menores; entonces, se divide el problema en dos casos, para cada uno se tienen ecuaciones, diagramas y una forma de implementarse a modo de algoritmo, para que no se pierda la numeración de las aristas.

Para que cada bit tenga un peso dominante sobre alguno menor, entonces se utiliza una característica de la neurona que es el tener *entradas inhibitorias*, con lo cual, se puede propagar la inhibición a los bits menos significativos. La metodología es la siguiente: cuando una pareja de bits A_i,B_i, se encuentran en el caso de que $A_i>B_i$, entonces esto significa que los demás bits no importan, dado que los bits de mayor peso son los que indican la preferencia,

por lo tanto, al detectar la pareja i de bits, los siguientes i-1, serán menores o mayores, cosa que ya no es importante saber, por lo que, se decide inhibir las demás salidas, dando preferencia a los bits mas significativos, como se muestra en la figura No. 8, que será explicada mas adelante.

4.1 Caso 1: Los Valores de A y B Son Iguales.

La figura No. 8 corresponde al caso en que los dos valores son iguales. Note que cada entrada tiene dos caminos distintos, uno cuando tiene un valor 1 y el otro cuando tiene un valor 0, las aristas etiquetadas con i_x son inhibitorias, mientras que las etiquetadas con e_x son entradas excitatorias.

Supónganse un par de bits A_3, B_3 iguales a 1. Al seguir ambos la ruta i_1 e i_2, llegan a neuronas con un valor $\theta = 0$ (NOT), entonces su valor se convierte en 0 para ambos, por lo que la suma de estos no iguala el umbral $\theta = 2$ (AND) de la siguiente neurona y por lo tanto no ocasiona su activación, teniendo esta a la salida 0 y mermando la continuación del camino.

Si seguimos la ruta e_1, e_2, los valores se conservan y al ser sumados igualan el umbral $\theta = 2$ de la siguiente neurona con lo que se activa la salida e_{16} =1, misma que al ser sumada con el valor de e_{17} =0, da un total de 1 con lo que alcanza el umbral $\theta = 1$ (OR) logrando la activación de la señal e_{25}.

Al seguir el mismo proceso para cada par de bits restantes (A_2, B_2), (A_1, B_1), (A_0, B_0), y en caso de tener un valor 1 en las señales e_{26}, e_{27} y e_{28}, se alcanza el umbral $\theta = 4$ (AND de 4 entradas) de la ultima neurona, con lo que se tiene un valor 1 en la salida A=B, lo que indica entonces que los valores de A y B son iguales.

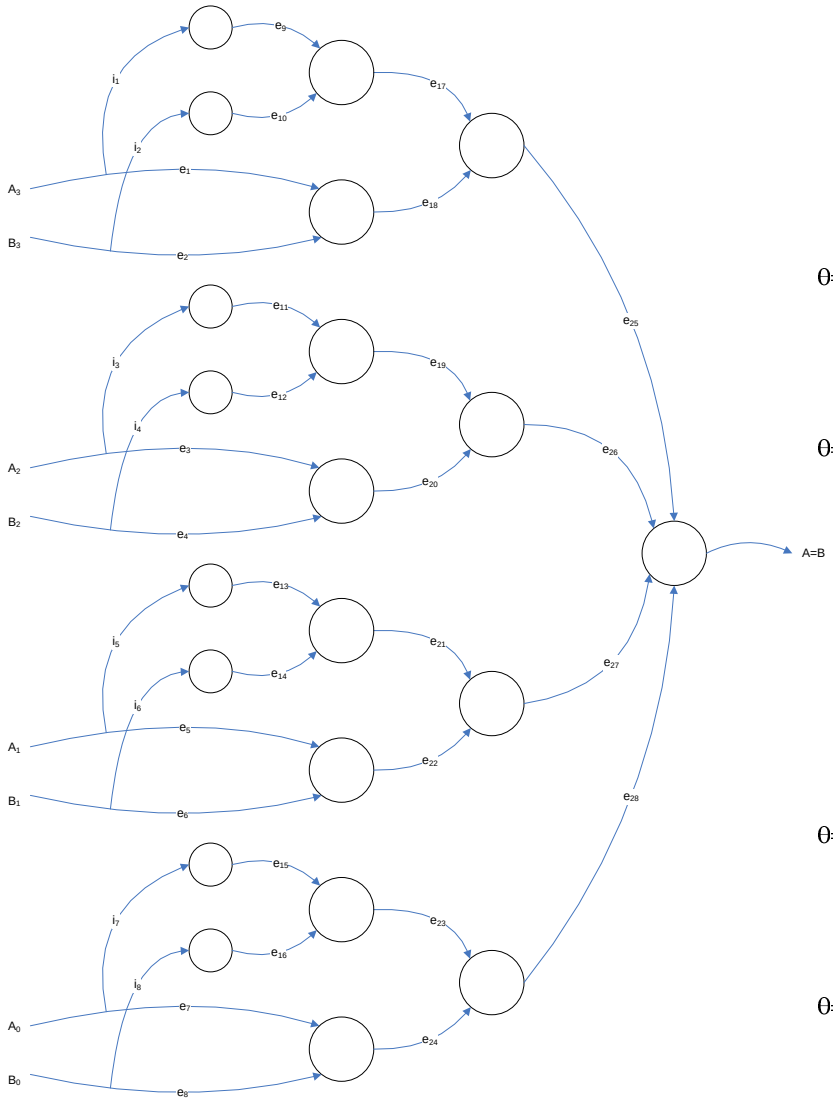

Figura 8. Salida A=B con neuronas MCP.

Con notación matemática de acuerdo con la figura anterior, sean A_x y B_x dos números binarios de 4 bits, con x en [0,3] y sean α, β, ω vectores de dimensión 4. Para cada A_x, B_x, existen dos caminos, uno excitatorio e_j y un inhibitorio i_i, con $1 \le j \le 8$ y $1 \le i \le 8$.

Para toda i_i invertirla realizando la operación

$$e_k = \bar{i}_i, \qquad \text{con} \qquad 1 \le i \le 8$$

Para poder realizar con incrementos de 2 en cada j y k, teniendo a $1 \le m \le 4$; $1 \le j \le 8$; $9 \le k \le 16$.

$$\alpha_m = e_j + e_{j+1} \qquad (9)$$

$$\beta_m = e_k + e_{k+1} \qquad (10)$$

Ahora para obtener el resultado por bit

$$\omega_m = \alpha_m + \beta_m \qquad m = 1, 2, 3, 4; \qquad (11)$$

Entonces para obtener el resultado global y dado que todos los bits deben ser iguales,

Si $\sum_{m=1}^{4} \omega = 4$, entonces A es igual que B.

4.2 Caso 2: Cuando los Valores no Son Iguales.

Para los casos A < B y A > B, es decir cuando no son iguales, en la figura No. 9 se muestra el diagrama de neuronas MCP que realizan el calculo para A y B, del caso anterior, puede notarse que cuando los valores sean distintos, la salida A=B no será activada.

Entonces siguiendo un camino para los bits A_3, B_3, puede notarse que cada uno de estos tiene dos caminos distintos, vamos a probar para el caso donde A_3=1, B_3=0, es inmediato que A>B, dado que estos son los bits mas significativos y como se mostró en la tabla 1, son los predominantes y por lo tanto los demás no deben alterar el resultado, entonces al seguir el camino e_1, i_2, en la figura 2, causa la negación del valor de B_3, y se tiene entonces $e_3 = \overline{B_3}$. Posteriormente, al realizar la suma $e_1 + e_3$ se iguala el valor de $\theta = 2$, con lo que se activa la salida, misma que toma dos caminos uno excitatorio hacia la salida e_{17}, mismo que causara la activación de la salida A>B. Podrá notar entonces que los valores de A_3 y B_3 siguiendo el camino i_1, e_2 no iguala el valor $\theta = 2$ de la neurona teniendo 0 a su salida.

Como se menciono anteriormente, dado que estos son los bits mas significativos, deben ser decisivos, siguiendo el segundo camino, el inhibitorio i_9 es propagado hacia las neuronas encargadas de proporcionar un resultado A<B, estableciendo su valor a 0 e impidiendo que se active esta salida.

Veamos ahora el caso donde A_3 = 0, B_3 = 1. Es obvio que $B_3 > A_3$ y por ende todo el numero, entonces se sigue un camino similar al caso anterior con la diferencia que se tiene $e_4 = \overline{A_3}$ que será sumado con e_2 y al igualar el valor de activación $\theta = 2$, el camino e_{17} genera de manera inmediata una salida A<B, y al igual que el caso anterior una señal inhibitoria i_{10} es propagada hacia las neuronas que generan A>B de los bits menos significativos.

Lo mismo sucede para el resto de bits, ya que siguen caminos idénticos a los descritos para A_3 y B_3, la diferencia esta a la salida, dado que las señales inhibitorias se propagan únicamente hacia los bits de menor peso.

Matemáticamente puede expresarse de la siguiente manera:

Sean A_x y B_x dos números binarios de 4 bits, con x en [0,3] y sean α, β vectores de dimensión 4. Para cada A_x, B_x, existen dos caminos, uno excitatorio y un inhibitorio.
Para toda i_i con $1 \le i \le 8$, realizar las operaciones

$$e_j = \overline{i_o}, \qquad \text{con} \qquad o = \{2,4,6,8\},\ j = \{3,7,11,15\}$$
$$e_m = \overline{i_p}, \qquad \text{con} \qquad p = \{1,3,5,7\},\ m = \{2,6,10,14\}$$

Entonces podemos obtener los valores para

$$\alpha_i = e_k + e_j \quad \text{con} \quad k = \{1,5,9,13\},\ j = \{3,7,11,15\}$$

$$\beta_i = e_l + e_m \quad \text{con} \quad l = \{4,8,12,16\},\ i = \{3,2,1,0\},\ m = \{2,6,10,14\}$$

De donde el resultado final se puede obtener con:

$$\text{A>B} \quad \text{si} \quad \sum_{i=0}^{3} \alpha_i > 1 \qquad (12)$$

$$\text{A<B} \quad \text{si} \quad \sum_{j=0}^{3} \beta_j > 1 \qquad (13)$$

O de una manera más eficiente

$$Si \quad \alpha_i = 1 \quad entonces \quad \beta = 0$$
$$Si \quad \beta_i = 1 \quad entonces \quad \alpha = 0 \quad \text{Para} \quad i = \{3,2,1,0\};$$

De lo anterior puede verse que comenzando por los bits más significativos si se obtiene un valor 1 para cualquiera de los casos se inhibe por completo el otro vector, sucesivamente, dado que de cumplirse el primer caso, termina la ejecución, en caso contrario significa que los bits son iguales y compara los siguientes menos significativos.

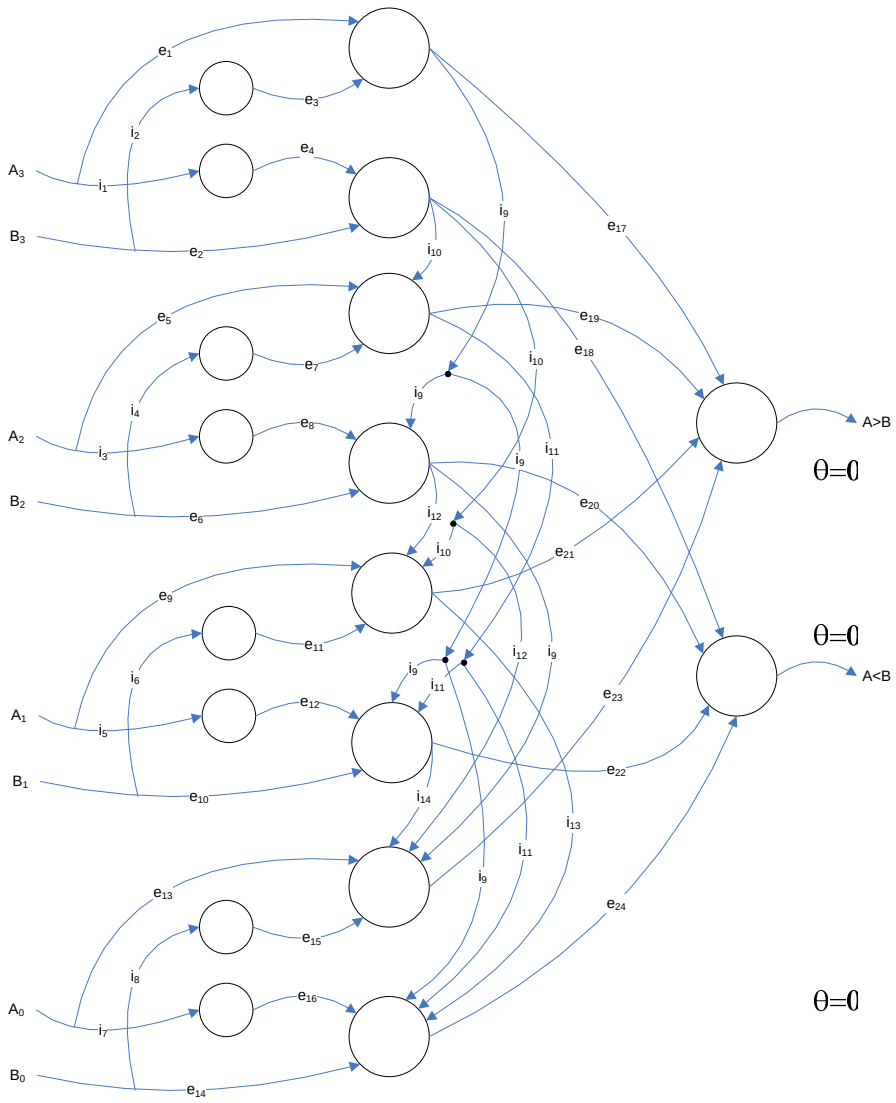

Figura 9. Salidas para A<B y para A>B.

θ=2
θ=0
θ=0
θ=2
θ=2
θ=0
θ=0
θ=2
θ=0
θ=0
θ=2

5. Conclusiones

Derivado de la disquisición experimental realizada en el presente trabajo, se concluye que la creación de circuitos digitales utilizando bloques básicos de neuronas AND, OR y NOT de tipo McCullouch – Pitts, es relativamente simple y que además permite la concatenación de dichos bloques básicos de neuronas; con el mismo orden de ideas, podemos concluir que aun cuando el diagrama resultante para cada circuito es muy complejo, esta complejidad varia según la aplicación, nos permite observar la efectividad y capacidad de este tipo de neuronas para adaptarse a distintos problemas, como lo sugieren en [1]. Además, la experiencia adquirida en el presente este trabajo, nos permitirá diseñar y construir circuitos digitales mas complejos como sea necesario, usando la concatenación de bloques básicos de neuronas.

6. Bibliografía

[1] McCulloch, W. S., Pitts W. (1943): A logical calculus of the ideas immanent in nervous activity, **Bulletin of Mathematical Biophysics.** (5), 115-133.

[2] Tocci, R. J. (1993): **Sistemas Digitales, Principios y aplicaciones**. (5ª ed.). Prentice-Hall.

[3] Motorola (2000): **Fast and LS TTL Data.** DL121/D. Rev. 6.