UNIVERSIDAD AUTONOMA de AGUASCALIENTES

LÓGICA DÍGITAL

"P R O Y E C T O final"

Profesor: Armando Álvarez Fernández

equipo

\- I n t e g r a n t e s:

Ángel David Ortiz Quiroz	ID: 261481	
Erick Iván Ramírez Reyes	ID: 260806	
Ximena Rivera Delgadillo	ID: 261261	
Diego Emanuel Saucedo Ortega	ID: 261230	(Representante)
Carlos Daniel Torres Macias	ID: 244543	

Fecha de entrega: 10 de junio de 2022.

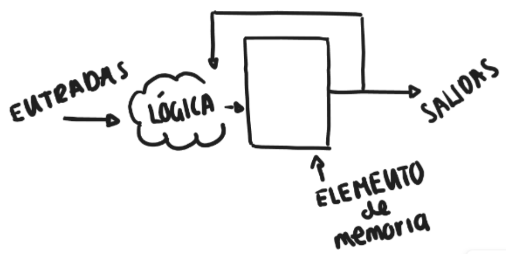
MAQUINAS de ESTADO

¿Qué es una máquina de estado?

Primero que nada, un ‘estado’ es un conjunto identificable y único de valores medidos en diversos puntos de un sistema digital. Un ‘diagrama de estado’ por otro lado ilustra la forma y funcionamiento de la máquina de estado y usualmente se dibuja como un diagrama de burbujas y flechas.

Y Las máquinas de estado son aquellas que pueden realizar tareas y que utilizan estados como su núcleo También son conocidas como máquinas de estado finitas, lo que significa que sabemos todos los posibles estados de ella. Los puntos clave de la máquina de estado son el concepto del tiempo y la historia. El estado de la máquina es evaluado periódicamente. Cada vez que es evaluada, un nuevo estado es elegido (el que podría ser el mismo estado nuevamente) y el resultado es presentado.

Una máquina de estado genérica tiene la siguiente estructura:



*El **elemento de memoria** contiene el nuevo estado conocido como el **estado variable**. Cuando la máquina de estado cuenta con los servicios, el estado variable es actualizado con el valor de la próxima etapa. La **nueva etapa** es una función de ambos; el estado actual y algunos inputs. La nube de la lógica es un sistema que decide cual será el próximo estado, o la próxima **lógica de estado**.*

¿Qué es una ramificación?

Ramificación El cambio del estado presente al estado siguiente. Estado siguiente: es el estado hacia el cual la máquina de estado realiza la siguiente transición, determinada por las entradas presentes cuando el dispositivo es secuenciado por un clock.

Las máquinas de estado pueden ser:

SÍNCRONAS: Necesitan de la intervención de un pulso de reloj. Si la entrada participa también en la salida se denomina Máquina de estado de Mealy, y si no participa se denomina de Moore.

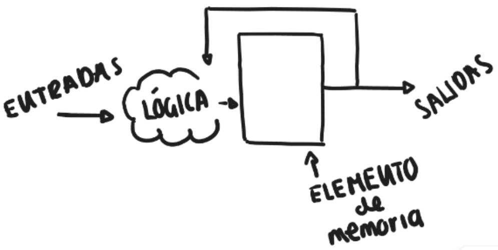
ASÍNCRONAS: No necesitan de la intervención de un pulso de reloj. Estos circuitos evolucionan cuando cambian las entradas.

MAQUINAS de ESTADO SÍNCRONAS o SINCRONIZADAS POR RELOJ

Máquina de estado Moore

Edward Moore escribió un ensayo en 1956 (*Gedanken-experiments on Sequential Machines*) y por lo tanto el estilo de la máquina lleva su nombre. Él dice que la salida depende solo del estado, y el próximo estado es dependiente del estado actual (o salida), y la entrada.

Entonces la máquina de estado de Moore en resumen es una máquina de estado que determina sus salidas solamente dependiendo de los estados presentes de la máquina.



Con nuestro diagrama previo

Podemos notar que no importa cuál será el estado de la entrada, la salida solo depende el estado actual contenido dentro del elemento de la memoria.

Máquina de estado Mealy

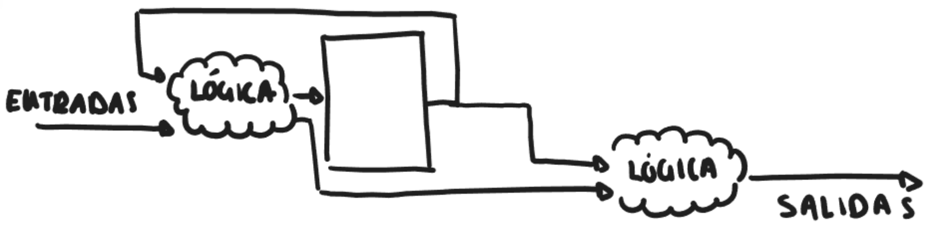
George Mealy escribió un ensayo un año antes que Moore, titulado "*A Method for Synthesizing Sequential Circuits*", en el cual entra en profundidad acerca de crear máquinas de estado desde funciones matemáticas, y describe esas salidas de máquinas de estado en términos de sus entradas.

En resumen, la Máquina de Mealy es una máquina de estado que determina sus salidas dependiendo de los estados presentes de la máquina y de las entradas.

Para diagramar la máquina Mealy, la salida está hecha para depender de ambos: el estado actual y la entrada. Aquí la nube de la lógica de la próxima etapa contiene la lógica de salida también:

Una forma de dibujar el diagrama de Mealy

También puede ser dibujado separando la nube en la lógica del próximo estado o lógica de salida.

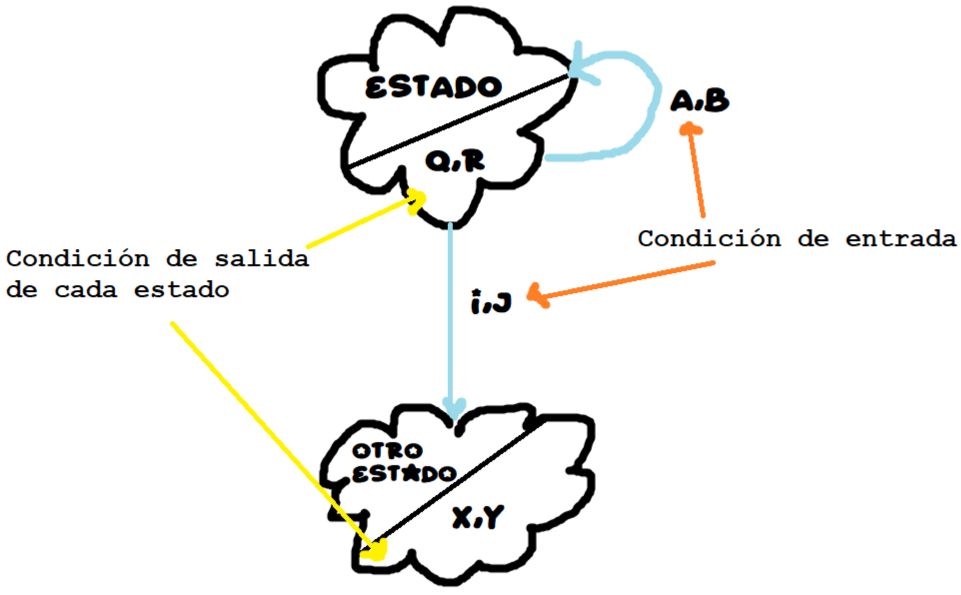


Normalmente, la máquina de estados de Moore emplea una mayor cantidad de estados internos y, por ello número de Flip-Flops. Mientras que la máquina de estados de Mealy suele tener un combinacional de salidas más complejo.

MÁQUINA DE MEALY	MÁQUINA DE MOORE
La salida depende del estado actual y la entrada	La salida depende únicamente del estado actual
Hay un menor número de estados	Tiene un número mayor o igual a la máquina de Mealy
Primero se realizan cambios en la entrada y después se efectúa un pulso de reloj	Se efectúa un pulso de reloj antes de realizar un cambio en la entrada
Las salidas se encuentran en las aristas	Las salidas se encuentran dentro del estado

DIAGRAMAS DE MÁQUINAS DE ESTADO

Diagrama de máquina de estado de Moore.



En una máquina de estado de Moore, las salidas son mostradas dentro de la burbuja de estados, ya que la salida será la misma mientras se encuentre en el mismo estado.

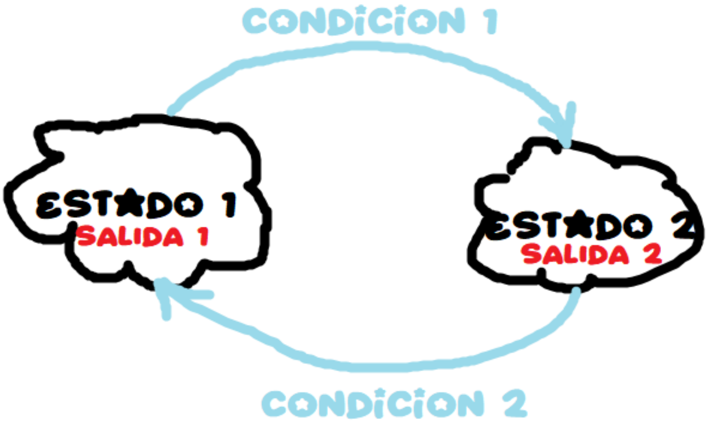
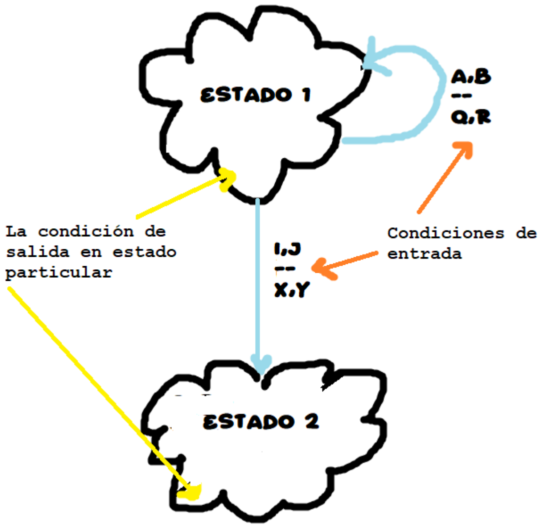
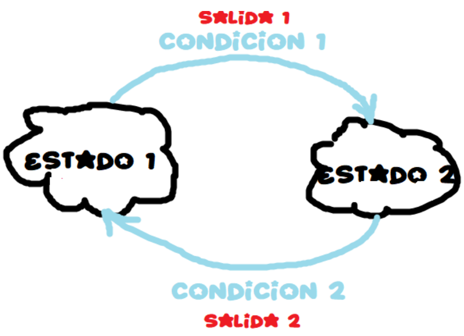


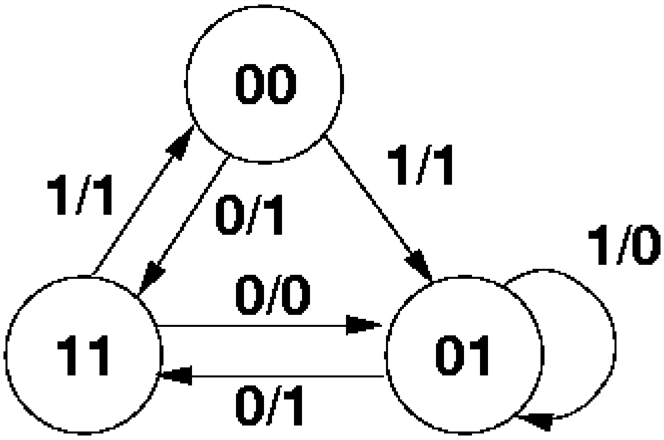
Diagrama de máquina de estado de Mealy.



La máquina de Mealy genera salidas con base en el estado actual y las entradas. De forma que es capaz de diseñar distintas señales de salida para un solo estado, dependiendo de las entradas presentes en el ciclo del reloj. Las salidas se muestran sobre las transiciones efectuadas en el momento, así como el estado a continuación.



EJEMPLOS DE MÁQUINAS DE ESTADOS

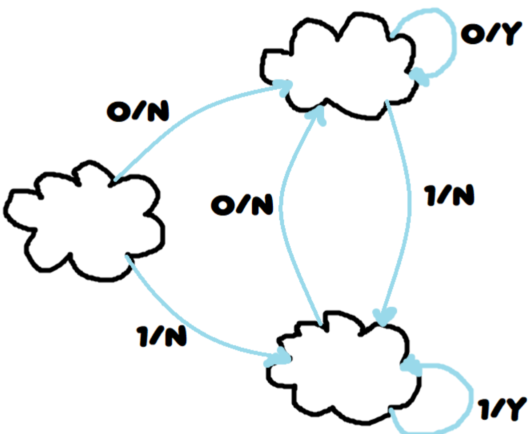


Podemos suponer que este ejemplo de máquina Mealy contiene dos Flip-Flops. La notación utilizada en los vértices tiene una forma como 1/0, en donde la primera palabra binaria (1) nos indica la entrada dada a la máquina y la segunda palabra binaria (0) nos indica la salida producida al llevarse a cabo la transición de un estado al siguiente. Esta máquina puede contener tres estados q1q0: 00, 01 y 11.

Si la máquina se encuentra en el estado q1q0=00, entonces de acuerdo con la notación en el vértice, 1/1, si se le aplica a la máquina una entrada de 1 entonces en el siguiente "pulso de reloj" transicionará al estado q1q0=01 produciendo una salida de 1. Si está en ese estado de q1q0=00 y se le aplica a la máquina una entrada de 0, entonces en el siguiente "pulso de reloj" la máquina transicionará al estado q1q0=11 produciendo una salida de 1.

Máquina de Mealy (COMPARADOR DE SECUENCIA)

Por ejemplo, queremos contar las apariciones de “00” y “11” de una cadena binaria ingresada, de forma que enviemos a una salida “Y” si se encuentra una aparición o a una salida “N” en caso de que no suceda. Utilizamos tres estados.



Suponiendo que se ingresa “10010011”, el proceso sería el siguiente:

El primer digito siempre ira a la salida “N” pues no hay otro digito para comparar si corresponde a alguna aparición esperada (“00” o “11”). Así el siguiente estado será el encargado de determinar la salida.

Si se trata del dígito “0” y el siguiente digito también, se tendrá salida a “Y”, y se dirigirá al mismo estado pues hay la posibilidad de que el siguiente digito lo sea de nuevo. Por lo contrario, si el siguiente digito es “1”, se tendría una secuencia no válida, por ello se envía a la salida “N”.

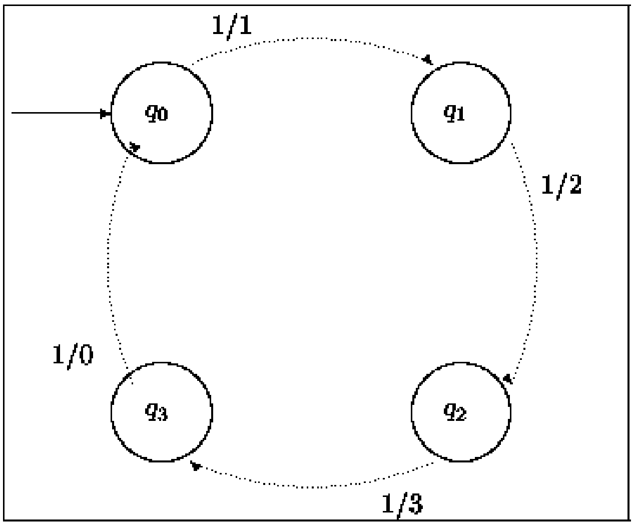
En el caso de 1, se mueve al estado para comparar 1, si en efecto el siguiente digito es “1” entonces se envía a la salida “Y” y se repite el estado, de caso contrario, se envía a la salida “N” y se pasa al estado que compara los “0”.

De forma que se mantiene el autómata en los estados de comparación de 1 y 0.

Otro ejemplo de una máquina tipo Mealy

Residuos módulo 4: Si $n \in \mathbb{N}$ entonces $n^{-1} = 1^{(n)}$ es la representación unaria de n. Presentaremos una máquina que calcula el residuo módulo 4, de una cadena de 1's, cuando se ve a esa cadena como la representación unaria de un número no-negativo.

Figure 3.1: Máquina de Mealy para el cálculo de residuos módulo 4 en representación unaria.



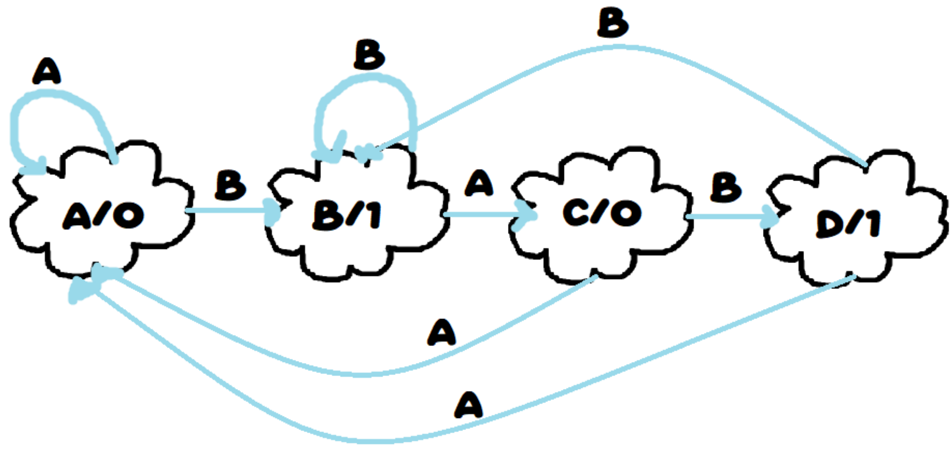
tran	1
q_0	q_1
q_1	q_2
q_2	q_3
q_3	q_0

res	1
q_0	1
q_1	2
q_2	3
q_3	0

(a) (b)

Máquina de MOORE (COMPARADOR DE SECUENCIA)

Un ejemplo similar para comparador de secuencia con Moore, pero incrementando la dificultad a una secuencia de tres caracteres “bab”.



Si el primer carácter es una “a”, inmediatamente se sabe que no corresponde a la secuencia por lo que se queda en el mismo estado hasta que el carácter que sigue no sea otro más que “b”. Si “b” sucede, hay dos posibilidades: que el siguiente carácter sea una “a” nos daría la pauta que vamos en camino a completar la secuencia y tendría paso el siguiente estado. En caso de que se tenga una “b” sería una secuencia incorrecta, pero se puede evitar regresar al primer estado pues se considera “repite” el inicio de la secuencia.

Pasando al siguiente estado, si obtenemos una “a” la secuencia podrá avanzar al estado siguiente, de caso contrario, habría una repetición de “a” y se regresaría al estado que se repite hasta que deja de ser “a”.

El último estado es muy restrictivo, si se encuentra con una “b” tendria una secuencia correcta y se podría enviar “1 lógico” a la salida, enseguida regresa al segundo estado pues la secuencia correcta termina en “b” y comienza en “b”, entonces podría existir la posibilidad de que tenga dos concatenaciones de la secuencia.

Si se tiene una “a” en el último estado, se estropea la secuencia y no hay otra salida más que volver al primer estado y evaluar que no se tenga una “a” de nuevo.

¿QUÉ USOS TIENEN LAS MÁQUINAS DE ESTADO?

Uso de máquina de Mealy.

Una máquina de Mealy puede implementarse en una máquina expendedora, la cual vende golosinas de $4 pesos cada una, que recibe monedas de $1, $2, $5 y $10 pesos. Supongamos que la máquina funciona bajo los siguientes supuestos:

- el costo de las golosinas puede cubrirse con cualquier combinación de monedas aceptables,
- la máquina sólo da cambio en monedas de $1 peso, las cuales están almacenadas en una alcancía. Si no puede dar cambio, es decir, si el contenido de la alcancía no es suficiente, regresa la moneda insertada, y
- sólo se puede insertar monedas en orden inverso a su denominación.

Codifiquemos el funcionamiento de la máquina con los conjuntos siguientes:

- *Monedas a insertarse*:

$$
\begin{array}{rcl}
m_0 & : & \text{ninguna moneda se inserta,} \\
m_1 & : & \text{moneda de un peso,} \\
m_2 & : & \text{moneda de dos pesos,} \\
m_5 & : & \text{moneda de cinco pesos,} \\
m_{10} & : & \text{moneda de diez pesos.}
\end{array}
$$

- *Respuestas de la máquina*:

$$
\begin{array}{rcl}
s_0 & : & \text{continúa sin más,} \\
s_1 & : & \text{entrega una golosina,} \\
s_2 & : & \text{da un peso de cambio,} \\
s_3 & : & \text{devuelve la moneda insertada.}
\end{array}
$$

- *Estados de la máquina*:

$$
\begin{array}{rcl}
q_0 & : & \text{estado inicial,} \\
\forall i \in [0,5] : a_i & : & \text{resta por devolver } i \text{ pesos,} \\
\forall j \in [1,2] : b_i & : & \text{falta por pagar } j \text{ pesos cuando se inició el pago con \$2,} \\
\forall k \in [1,3] : c_i & : & \text{falta por pagar } k \text{ pesos cuando se inició el pago con \$1.}
\end{array}
$$

- *Depósito en la alcancía*:

$$
\begin{array}{rcl}
\forall i \in [1,6] : p_i & : & \text{NO alcanza a haber } i \text{ pesos,} \\
p_7 & : & \text{al menos hay \$6 pesos.}
\end{array}
$$

Table 3.1: Transiciones y repuestas de la máquina expendedora.

$\forall j \leq 6 :$ $tran(q_0, m_{10}p_j) = q_0$ $res(q_0, m_{10}p_j) = s_3$	Si se inserta una moneda de $10 pesos y no hay cambio suficiente, se devuelve la moneda y se reinicia el proceso,

$\begin{aligned} tran(q_0, m_{10}p_7) &= a_5 \\ res(q_0, m_{10}p_7) &= s_2 \end{aligned}$	Ya que lo hay, procédase a dar cambio,
$\forall k = 5,4,3,2,1:$ $\begin{aligned} tran(a_k, m_0P) &= a_{k-1} \\ res(a_k, m_0P) &= s_2 \end{aligned}$	Para $P=p_j$, cualquiera que sea j, continúese devolviendo un peso hasta completar el cambio. Obsérvese que aquí, en principio, puede haber combinaciones (a_k,p_j) contradictorias. Sin embargo, la interpretación que se está construyendo excluye que aparezcan esas inconsistencias.
$\begin{aligned} tran(a_0, m_0P) &= q_0 \\ res(a_0, m_0P) &= s_1 \end{aligned}$	Al terminar de dar el cambio, se entrega la golosina y se reinicia el proceso.

Table 3.2: Transiciones y repuestas de la máquina expendedora (contador).

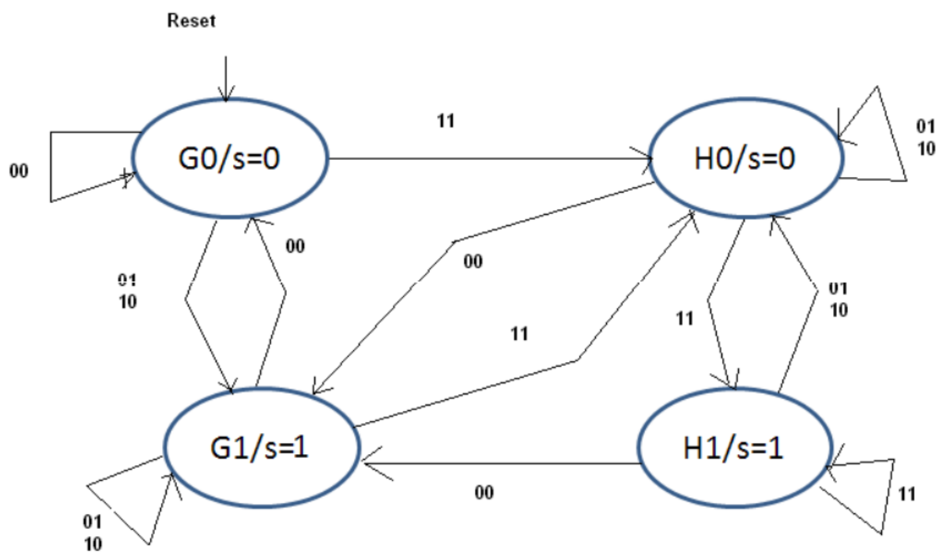
$\begin{aligned} tran(q_0, m_5p_1) &= q_0 \\ res(q_0, m_5p_1) &= s_3 \end{aligned}$	Si se inserta una moneda de $5 pesos y no hay cambio, se devuelve la moneda y se reinicia el proceso,
$\begin{aligned} tran(q_0, m_5P) &= a_0 \\ res(q_0, m_5P) &= s_2 \end{aligned}$	Si hay monedas en la alcancía, i.e. $P \neq p_1$, entonces se da el peso de cambio,
$\begin{aligned} tran(q_0, m_2P) &= b_2 \\ res(q_0, m_2P) &= s_0 \end{aligned}$	Se insertan $2 pesos y se espera a completar el importe de $4 pesos,
$\begin{aligned} tran(b_2, m_2P) &= q_0 \\ res(b_2, m_2P) &= s_1 \end{aligned}$	Habiéndose completado el costo de la golosina, se lo entrega y se reinicia el proceso,
$\begin{aligned} tran(b_2, m_1P) &= c_1 \\ res(b_2, m_1P) &= s_0 \end{aligned}$	Se inserta un peso más y hay que esperar a que llegue el último,
$\begin{aligned} tran(b_2, MP) &= b_2 \\ res(b_2, MP) &= s_3 \end{aligned}$	Si llega una moneda con denominación mayor M=m_5,m_{10} entonces se la devuelve y se continúa la espera,
$\begin{aligned} tran(q_0, m_1P) &= c_3 \\ res(q_0, m_1P) &= s_0 \end{aligned}$	Si se inicia el pago con una moneda de un peso hay que esperar los otros tres pesos,
$\forall k = 3,2,1:$ $\begin{aligned} tran(c_k, m_1P) &= c_{k-1} \\ res(c_k, m_1P) &= s_0 \end{aligned}$	Se continúa el pago, recibiendo un peso a la vez. Aquí c_0=a_0. Si se recibe monedas de mayor denominación, se devuelve éstas.
	Cualquier otra posibilidad (Estado,Entrada) es inconsistente e inalcanzable en la máquina.

$\forall j \leq 6:$ $tran(q_0, m_{10}p_j) = q_0$ $res(q_0, m_{10}p_j) = s_3$	Si se inserta una moneda de $10 pesos y no hay cambio suficiente, se devuelve la moneda y se reinicia el proceso,
$tran(q_0, m_{10}p_7) = a_5$ $res(q_0, m_{10}p_7) = s_2$	Ya que lo hay, procédase a dar cambio,
$\forall k = 5,4,3,2,1:$ $tran(a_k, m_0P) = a_{k-1}$ $res(a_k, m_0P) = s_2$	Para P=p_j, cualquiera que sea j, continúese devolviendo un peso hasta completar el cambio. Obsérvese que aquí, en principio, puede haber combinaciones (a_k,p_j) contradictorias. Sin embargo, la interpretación que se está construyendo excluye que aparezcan esas inconsistencias.
$tran(a_0, m_0P) = q_0$ $res(a_0, m_0P) = s_1$	Al terminar de dar el cambio, se entrega la golosina y se reinicia el proceso.
$tran(q_0, m_5p_1) = q_0$ $res(q_0, m_5p_1) = s_3$	Si se inserta una moneda de $5 pesos y no hay cambio, se devuelve la moneda y se reinicia el proceso,
$tran(q_0, m_5P) = a_0$ $res(q_0, m_5P) = s_2$	Si hay monedas en la alcancía, i.e. $P \neq p_1$, entonces se da el peso de cambio,
$tran(q_0, m_2P) = b_2$ $res(q_0, m_2P) = s_0$	Se insertan $2 pesos y se espera a completar el importe de $4 pesos,
$tran(b_2, m_2P) = q_0$ $res(b_2, m_2P) = s_1$	Habiéndose completado el costo de la golosina, se lo entrega y se reinicia el proceso,
$tran(b_2, m_1P) = c_1$ $res(b_2, m_1P) = s_0$	Se inserta un peso más y hay que esperar a que llegue el último,
$tran(b_2, MP) = b_2$ $res(b_2, MP) = s_3$	Si llega una moneda con denominación mayor M=m_5,m_{10} entonces se la devuelve y se continúa la espera,
$tran(q_0, m_1P) = c_3$ $res(q_0, m_1P) = s_0$	Si se inicia el pago con una moneda de un peso hay que esperar los otros tres pesos,
$\forall k = 3,2,1:$ $tran(c_k, m_1P) = c_{k-1}$ $res(c_k, m_1P) = s_0$	Se continúa el pago, recibiendo un peso a la vez. Aquí c_0=a_0. Si se recibe monedas de mayor denominación, se devuelve éstas.
	Cualquier otra posibilidad (Estado,Entrada) es inconsistente e inalcanzable en la máquina.

Uso de máquina de Moore.

Debido a que la salida es tomada directamente del estado, resulta ser demasiado útil para realizar circuitos de conteo, pues requiere una cantidad menor de circuitos combinacionales. La máquina toma las salidas directamente del registro por tanto no necesita ser sintetizada.

Sumador Serial con Moore:



Observemos que corresponde a 4 estados, 2 salidas con 2 estados (0,1) para poder realizar las combinaciones de sumas de 2 dígitos por medio de cambios de estados. Las transiciones están dadas por los elementos ingresados en el sumador.

Conclusión.

El desarrollo de la lógica es fundamenta no sólo para entender este tema, sino para todo lo que es nuestra carrera, pues tanto en el hardware como en el software, es indispensable, pues al final de cuentas, todo esto son circuitos y funciones lógicas.

En el presente documento se estudió las maquinas finitas de estado sincrónicas, estudiando sus dos variaciones, tanto la de Mealy y la de Moore, teniendo cada una distintas funciones, aplicaciones e implementaciones, analizando su funcionamiento y en base a esto, la investigación y el razonamiento inductivo; discernir en que es mejor emplear cada una de las máquinas de acuerdo a la situación.

Además, esto nos sirve como introducción o un primer acercamiento a materias que prontamente tendremos en nuestro paso por la universidad, como lo es Autómatas, que son los primeros ejemplos que tenemos de estos.

Referencias:

- Materan, D. (2018). USO DE LAS MAQUINAS DE ESTADO FINITOS Y MAQUINAS DE ESTADO ALGORITMICAS EN LOS LENGUAJES DE DESCRIPCION DE HARDWARE Autora.
- *apuntesDSD10_mealy_moore*. (s. f.). Recuperado 8 de mayo de 2022, de https://bloganalisis1.files.wordpress.com/2011/01/apuntesdsd10_mealy_moore.pdf
- *1. Máquinas de Moore y Mealy*. (2021, 15 mayo). [Vídeo]. YouTube. https://www.youtube.com/watch?v=kcaY76wGygs&t=7s
- Jorge Alejandro Gutiérrez Orozco. (2008). Máquinas de Estados Finitos Breve Introducción. Junio, 2022, de Escuela Superior de Computo Sitio web: https://www.comunidad.escom.ipn.mx/genaro/Papers/Veranos_McIntosh_files/alejandroFinal2008.pdf
- paguayo. (2019, June 18). Máquinas de estado - MCI Capacitación. MCI Capacitación. https://cursos.mcielectronics.cl/2019/06/18/maquinas-de-estado/
- Casasnovas, I., Centro, C., De, U., De Automatización, D., & Robótica, Y. (n.d.). https://www.profesores.frc.utn.edu.ar/electronica/tecnicasdigitalesi/pub/file/AportesDelCudar/Maquinas%20de%20Estado%20MC%20V5.pdf
- Tema 7. Análisis de Circuitos Secuenciales. (n.d.). https://personales.unican.es/manzanom/Planantiguo/EDigitalI/Tema_VII.pdf
- *5.2 - Circuitos de Moore y de Mealy*. (2019, 4 septiembre). [Vídeo]. YouTube. https://www.youtube.com/watch?v=TNxK2kvaKOk&t=366s