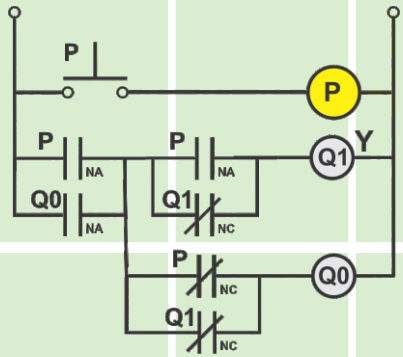
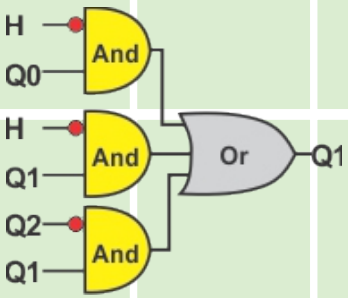
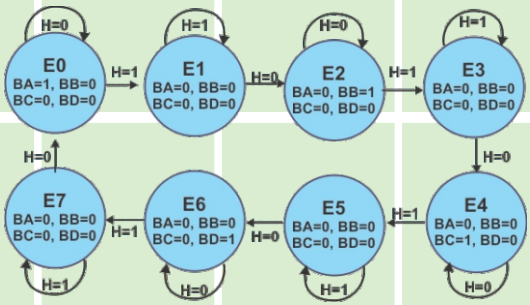
Diseño de Sistemas Secuenciales Asíncronos

Repertorio de problemas resueltos y propuestos







Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

J. A. Garza Garza, Platas Garza, Garza Camarena

Diseño de Sistemas Secuenciales Asíncronos

Repertorio de problemas propuestos y resueltos

Autores: ***Juan Angel Garza Garza, Miguel Angel Platas Garza, Jesús Daniel Garza Camarena.***

Universidad Autónoma de Nuevo León

Santos Guzmán López
RECTOR

Juan Paura García
SECRETARIO GENERAL

José Javier Villareal Tostado
SECRETARIO DE EXTENSIÓN Y CULTURA

Antonio Ramos Revillas
DIRECTOR DE EDITORIAL UNIVERSITARIA

Arnulfo Treviño Cubero
DIRECTOR DE LA FACULTAD DE INGENIERIA MECANICA Y ELECTRICA

ISBN: **978-607-27-2100-5**

Primeraedición, Agosto 2023

©*Universidad Autónoma de Nuevo León*

Reservados todos los derechos. Ninguna parte de este libro puede ser reproducida, archivada o transmitida, en cualquier sistema -electrónico, mecánico, de foto reproducción, de almacenamiento en memoria o cualquier otro-, sin hacerse acreedor a las sanciones establecidas en las leyes, salvo con el permiso expreso del titular del copyright. Las características tipográficas, de composición, diseño, formato, corrección son propiedad del editor.

Email: *juan.garzagza@uanl.edu.mx*

Página web: https://jagarza.fime.uanl.mx/

Agradecimientos

En primer lugar, a mi familia, mi esposa *Marina*, mis hijos *Juan Ángel, Eduardo Nicolás y Jesús Daniel Garza Camarena*, por ceder de su tiempo y espacio que por cariño les corresponde y me han permitido consolidar mis proyectos.

Al Dr. Arnulfo Treviño Cubero de la Facultad de Ingeniería Mecánica y Eléctrica de la Universidad Autónoma de Nuevo León, un líder humano, visionario e incluyente, que siempre ha impulsado y apoyado fuertemente los proyectos realizados por los académicos de nuestra institución.

Muy especialmente a mi maestro M.C. Guadalupe Ignacio Cantú Garza, quien por el aporte de sus conocimientos ha sido el impulsor de este proyecto.

Es justo agradecer al Dr. Cesar Elizondo González la aportación del método de la Variable Lógica en donde propone el teorema fundamental secuencial, que es un recurso didáctico en la solución de algunos problemas de diseño secuencial asíncrono que se muestran en la parte final de esta obra.

También una dedicatoria especial para los maestros M. C. Guadalupe Evaristo Cedillo Garza, Dr. Fernando Banda Muñoz, M. C. Francisco Javier Esparza Ramírez, Dr. Cesar Guerra Torres, Dr. Juan Antonio Aguilar Garib, por sus contribuciones y sugerencias a este trabajo.

Índice

Prólogo **5**
Introducción **6**
Método de Diseño de un Sistema Secuencial Asíncrono: **23**
Problema 1.1, Botón caso 1 ..... **27**
Ocho Formas Estándar **39**
Problema 1.2, Botón caso 2 ..... **75**
Problema 1.3, Botón caso 3 ..... **80**
Problema 1.4, Botón caso 4 ..... **85**
Problema 1.5, Botón caso 5 ..... **89**
Problemas propuestos **92**
Problema 2, Control de una bomba para el llenado de un tanque **111**
Problema 3, Detector de nivel de un tanque **123**
Problema 4, Sistema alternativo de dos bombas no simultaneas **129**
Problema 5, Sistema alternativo de tres bombas no simultaneas **138**
Problema 6, Sistema alternativo de 2 de 3 bombas **153**
Problema 7, Llenado de un tanque cuatro bombas 3 de 4 ..... **162**
Problema 8, Llenado de un tanque cuatro bombas 1 de 4 ..... **171**
Problema 9, llenado de un tanque cuatro bombas 2 de 4 ..... **179**
Problema 10, Detector de tres niveles en un tanque. **187**
Problema 11, Dos bombas simultáneas caso 1 ..... **195**
Problema 12, Dos bombas simultáneas caso 2 ..... **202**
Problema 13, Dos bombas simultáneas caso 3. ..... **210**
Problema 14, Sistema de Arranque y Paro **222**
Problema 15, Arranque con secuencia AB y paro con secuencia BA **228**
Problema 16, Barreras de Seguridad de una Vía de Ferrocarril **235**
Problema 17, Detector de entrada y salida de estacionamiento **241**
Problema 18, Dos tolvas de Ensacado **250**
Problema 19, tres tolvas de ensacado **253**
Problema 20, Concurso de 2 participantes **259**
Problema 21, Concurso de 3 participantes **265**
Problema 22, Laberinto (caso1) **274**
Problema 23, Laberinto (caso 2) **281**
Problema 24, Detector de giro **287**
Problema 25, Medidor de longitud **295**
Problema 26, Pulsos Completos. **303**
Problema 27, impulsos sucesivos **324**
Problema 28, Secuencia de puerta dos botones **330**

Variable Lógica.....340
Problema 29, Botón caso 1.1.....341
Problema 30, Control de una bomba para el llenado de un tanque 350
Problema 31, Detector de Nivel de un tanque 354
Problema 32, Sistema alternativo de dos bombas no simultaneas 356
Problema 33, Detector de tres niveles de un tanque 360
Epílogo.....363

Prólogo

En el ámbito de la ingeniería, el diseño de sistemas secuenciales asíncronos representa un desafío fundamental que requiere habilidades analíticas, creatividad y un profundo entendimiento de los conceptos electrónicos. Es un campo donde la sincronización y coordinación de eventos juegan un papel crucial en el funcionamiento correcto de dispositivos y sistemas complejos. Este libro, titulado "Diseño de Sistemas Secuenciales Asíncronos", surge como una guía esencial para los estudiantes de ingeniería que buscan dominar las técnicas y métodos alternativos para abordar este tipo de sistemas de manera efectiva y eficiente.

La naturaleza asincrónica de estos sistemas implica que los eventos no están vinculados a un reloj común, lo que plantea desafíos únicos en comparación con los sistemas síncronos. Este libro se proporciona a los estudiantes las herramientas necesarias para abordar estos desafíos con confianza y creatividad. A través de una combinación de teoría sólida, casos de estudio diversos y ejercicios prácticos, los lectores serán guiados en un viaje educativo que les permitirá comprender y resolver problemas relacionados con el diseño de sistemas secuenciales asíncronos.

Una característica distintiva de este libro es su enfoque en métodos alternativos. En lugar de limitarse a una sola perspectiva o técnica, se presentan múltiples enfoques para abordar cada problema. Esto permite a los estudiantes explorar diversas estrategias y seleccionar la que mejor se adapte a cada situación particular. Cada problema presenta casos variados en los que se aplican diferentes recursos y enfoques, brindando a los lectores una visión amplia y equilibrada de las posibles soluciones.

Además de la teoría y los enfoques conceptuales, esta obra se adentra en la aplicación práctica del diseño de sistemas secuenciales asíncronos. La simulación es presentada como una herramienta esencial para validar y refinar los diseños antes de la implementación física. Pero no nos detenemos ahí. También exploramos la construcción de prototipos y brindamos las bases para la programación de PLCs (Controladores Lógicos Programables), lo que enriquece la experiencia de aprendizaje al vincular los conceptos teóricos con situaciones del mundo real.

Esperamos que "Diseño de Sistemas Secuenciales Asíncronos" sea una herramienta valiosa para los estudiantes, profesores e ingenieros que deseen fortalecer sus habilidades en este fascinante y desafiante campo, este libro les guiará paso a paso hacia el dominio de estos conceptos vitales en el mundo de la ingeniería.

¡Bienvenidos a un viaje de aprendizaje emocionante y enriquecedor!

Juan Angel Garza Garza

Introducción

El propósito de este libro es proporcionar a los estudiantes de ingeniería métodos alternativos para la solución en el diseño de sistemas secuenciales asíncronos, El libro presenta más de 33 problemas de diseño, con soluciones detalladas y verificación mediante simulación.

El método utilizado para resolver los problemas está basado en el libro *Fundamentals of Logic Design* del autor Charles H. Roth. (Roth, C. H., Jr. (1995). Fundamentals of Logic Design 4th ed. Pws Pub Co.)

En una segunda parte de este libro se aplica en algunos problemas el método de la Variable Lógica desarrollado por el Dr. Cesar Elizondo González (Aplicación de Variable Lógica a la Solución de Problemas Combinacionales o Secuenciales Asíncronos ISBN: 970-32-2137-8) mostrando en los casos presentados ventajas en el diseño con respecto al método anterior.

En cada uno de los problemas se presentan casos diversos en donde se aplican los diferentes recursos de los métodos mencionados para la solución de los problemas.

Para efectos de aprendizaje se propone obtener el diseño del problema 1.1 para los casos de las asignaciones 6, 7 y 8.

Es importante tener conocimientos básicos de Álgebra Booleana en sus formas de representación y técnicas de minimización para comprender la aplicación de los métodos de diseño propuestos.

En la fase preliminar del diseño, se recomienda verificar los resultados obtenidos en la simplificación de las ecuaciones. Para esto, se propone el uso del programa LogicAid desarrollado por Charles H., Jr. Roth.

Se menciona la posibilidad de construir prototipos de los diseños utilizando PLC (Controlador Lógico Programable), PLD (Dispositivo Lógico Programable) o circuitos digitales de función fija (TTL o CMOS), buscando soluciones económicamente viables.

Para el proceso de diseño y construcción del prototipo, se sugiere el uso del lenguaje de descripción de Hardware ABEL-HDL o VHDL a través del programa IspLEVER Classic de la compañía Lattice Semiconductor. Se pueden obtener licencias académicas gratuitas para este programa

Es recomendable realizar pruebas de simulación para verificar el funcionamiento esperado antes de construir el prototipo. Se mencionan dos opciones de simulación: el comando Test_vectors del programa IspLEVER Classic, que permite obtener un diagrama de tiempos, y el programa PROTEUS, una herramienta CAE que permite la simulación de circuitos electrónicos y la elaboración de Circuitos Impresos.

La simulación ya sea por medio del Test_vectors o con el programa PROTEUS las cuales nos permiten comprobar que en la parte correspondiente al diseño no se tiene ninguna falla antes de construir un prototipo.

En esta obra se incluyen códigos de respuesta rápida **QR** (del inglés Quick Response code, "código de respuesta rápida")**,** con el propósito de tener fácil acceso, tanto a la página del autor, como a los cinco videos en donde se explica en detalle el procedimiento de la solución del problema 1.1.

Detección y corrección de fallas

Propuesta del Método de Análisis y Detección de Fallas en el Proceso de Diseño de Sistemas Secuenciales Asíncronos: Detección y Corrección de Fallas

Durante el proceso de diseño de un sistema y la construcción de un prototipo, pueden surgir diversos errores que afecten el correcto funcionamiento del sistema.

Estos errores pueden clasificarse en tres tipos principales:

1.- Errores en el diseño.

2.- Errores en la implementación del prototipo.

3.- Componentes defectuosos.

Para abordar esta problemática, se propone un método de análisis y detección de fallas que permita identificar y corregir los errores de manera eficiente y efectiva.:

A continuación, se muestra el diagrama de flujo recomendado para el análisis y la detección y corrección de fallas.

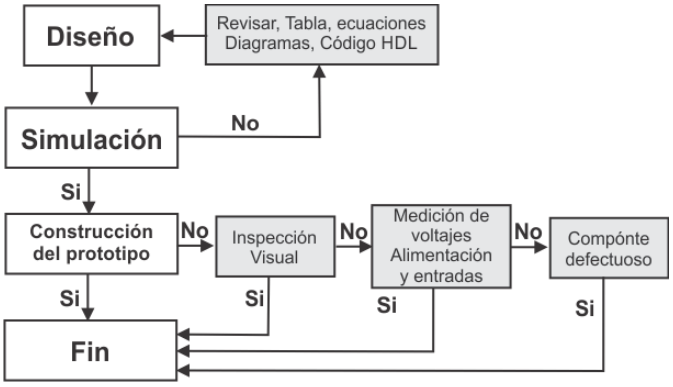


Fig. 1. Gráfica del Método de detección y corrección de fallas

1.- En la etapa de diseño, se recomienda verificar los resultados mediante la utilización de un procedimiento de simulación para poder compararlos con los valores obtenidos.

Esta simulación puede llevarse a cabo utilizando el comando "Test_Vectors" (vectores de prueba del lenguaje ABEL-HDL), que generará una representación gráfica como se muestra en la siguiente figura:

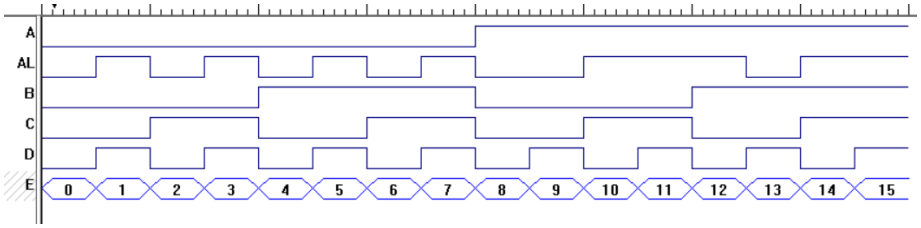


Fig. 2. Imagen gráfica de simulación por medio del Test_Vectors

La simulación mediante el uso de vectores de prueba proporciona una visualización clara y comprensible de cómo se comportará el sistema en diferentes situaciones. Esto permite identificar posibles desviaciones o discrepancias entre los resultados esperados y los obtenidos durante la simulación.

Al utilizar el comando Test_Vectors, se pueden generar diferentes conjuntos de datos de entrada para evaluar el comportamiento del sistema en condiciones variadas. Esto ayuda a detectar posibles problemas o errores en el diseño, así como a validar el funcionamiento correcto del sistema.

Es importante destacar que la simulación con vectores de prueba no reemplaza la necesidad de realizar pruebas exhaustivas y validaciones adicionales durante el proceso de diseño y construcción del sistema. Sin embargo, constituye una herramienta invaluable para la detección temprana de posibles fallas y la optimización del diseño antes de proceder a la implementación del prototipo.

La inclusión de simulaciones utilizando el comando Test_Vectors en el proceso de diseño de sistemas secuenciales asíncronos contribuirá a mejorar la precisión y confiabilidad del sistema, permitiendo realizar ajustes y correcciones necesarias antes de pasar a la etapa de implementación del prototipo.

En el caso de que los resultados obtenidos durante la simulación no coincidan con los valores esperados, se debe revisar detalladamente cada uno de los pasos del diseño. Es importante identificar las posibles causas de las discrepancias y tomar las medidas necesarias para corregir cualquier error o desviación encontrada.

A continuación, se sugieren algunos pasos a seguir para revisar el diseño en caso de discrepancias en los resultados de la simulación:

Verificar las especificaciones del diseño: Volver a revisar las especificaciones del diseño y asegurarse de que se hayan tenido en cuenta todos los requisitos y restricciones. Es posible que se haya pasado por alto algún detalle importante durante la etapa de diseño inicial.

Revisar cuidadosamente cada uno de los pasos del método propuesto para el diseño de que se haya realizado correctamente según las especificaciones establecidas. Es posible que se hayan cometido errores en el proceso o que se haya omitido algún evento esencial.

Analizar los vectores de prueba: Examinar los vectores de prueba utilizados en la simulación y verificar si son adecuados para evaluar todos los posibles casos de funcionamiento del sistema. Asegurarse de que los vectores de prueba cubran una amplia gama de situaciones y condiciones.

Realizar pruebas adicionales: En caso de discrepancias persistentes, realizar pruebas adicionales utilizando diferentes métodos o herramientas de simulación. Esto puede ayudar a identificar si el problema está relacionado con la configuración de la herramienta de simulación utilizada o si existe un error más profundo en el diseño.

En resumen, cuando los resultados de la simulación no coinciden con los valores esperados, es necesario realizar una revisión minuciosa del diseño, considerando cada paso y componente involucrado. La identificación y corrección de errores durante esta etapa contribuirá a garantizar la integridad y confiabilidad del sistema secuencial asíncrono diseñado.

Otra forma de realizar la simulación y comparar los resultados es utilizando el programa PROTEUS. En este caso, se requiere utilizar el archivo JED (Joint Electron Device Engineering Council JEDEC), que es un estándar establecido para el intercambio de información entre programas y dispositivos programables. Este archivo JED se genera utilizando el código ABEL-HDL y el programa IspLEVER Classic.

El archivo JED permite realizar una simulación en PROTEUS, donde se pueden obtener los valores de salida al proporcionar los valores de entrada correspondientes. A continuación, se muestra en la figura 3 una representación visual de la simulación:

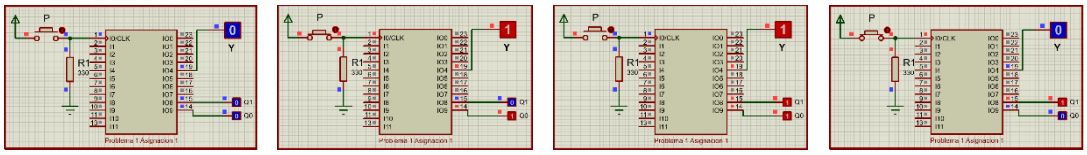


Fig. 3. Imágenes de la Simulación por medio del programa PROTEUS

En estas imágenes de la simulación en PROTEUS, se pueden observar los resultados obtenidos para cada valor de entrada proporcionado. Esto facilita la comparación de los resultados esperados con los resultados simulados, permitiendo identificar posibles discrepancias y evaluar el correcto funcionamiento del sistema.

El uso de PROTEUS y el archivo JED en la simulación brinda una representación visual clara y detallada del comportamiento del sistema en diferentes escenarios. Esto ayuda a detectar posibles problemas o desviaciones en el diseño, así como a realizar ajustes y correcciones necesarias.

La inclusión de simulaciones en PROTEUS utilizando el archivo JED en el proceso de diseño de sistemas secuenciales asíncronos contribuirá a mejorar la precisión y confiabilidad del sistema, permitiendo realizar ajustes y correcciones necesarias antes de proceder a la etapa de implementación del prototipo

2.- En la etapa de construcción de un prototipo, si no cumple con lo esperado, se recomienda primero una inspección visual, para asegurar que todo esté conectado correctamente y si aún persiste la falla se recomienda la medición de los voltajes esperados tanto de alimentación del sistema de preferencia bajo carga, como en cada una de las entradas y salidas, que cumplan con lo recomendado por el fabricante en sus hojas de datos, por lo general un uno lógico se considera de 1.8 v a 5 v y un cero lógico de 0 v a 0.8 v., se recomienda medir directamente a la terminal del dispositivo

3.- En caso de continúe la falla hay que considerar que algún o algunos componentes puedan estar dañados o mal programados.

"El enfoque del ingeniero se basa en la evidencia, no en la creencia"

J. A. Garza G.

El Ingeniero no cree, comprueba, mide, verifica, confirma.

Diseño de Sistemas Secuenciales Asíncronos

Comenzaremos por establecer una propuesta de definición clara de lo que se busca lograr con el diseño en el campo de la ingeniería.

Diseño en ingeniería:

"Es la creación y desarrollo de un producto, proceso o sistema económicamente viable para satisfacer necesidades definidas por un cliente o proceso"

Andrew McLaren, Approaches to the Teaching of Design, Engineering Subject Centre, The Higher Education Academy, University of Sheffiled UK, 2008, ISBN 978-1-904804-802**Sistema:**

Por otro lado, es importante definir lo que se entiende por Sistema, que se considera como un **Conjunto de cosas o partes coordinadas según una ley, o que, ordenadamente relacionadas entre sí, contribuyen a determinado objeto o función.**

Diccionario General de la Lengua Española Vox

Sistema Secuencial: Es un bloque digital en donde los valores de salida no dependen únicamente de las combinaciones de entrada, sino también de la salida misma.

Asíncrono: eventos que no están coordinados en el tiempo. (Modo Nivel)

Circuitos Secuenciales Asíncronos

Los circuitos secuenciales pueden clasificarse en dos grandes grupos:
1.-Circuitos secuenciales sincrónicos o síncronos (modo Clk).

2.- Circuitos secuenciales asincrónicos o asíncronos (modo nivel).

Los primeros caen en la categoría de *circuitos modo reloj*. En tales circuitos un cambio de estado ocurre sólo en respuesta a un pulso de reloj. Así, un cambio de estado se +realiza ***simultáneamente*** en todos los *Flip Flop's* que constituyen el circuito, debido a que todos ellos están sincronizados por un pulso de reloj común. Las entradas cambian entre pulsos y las salidas se leen inmediatamente después de dicho pulso. Cuando las entradas cambian, el circuito no pasa inmediatamente al estado siguiente, sino después de recibir el consecuente pulso de reloj.

Aun cuando los circuitos secuenciales síncronos son muy útiles, hay tres casos en los que no pueden utilizarse:

Cuando las entradas no pueden sincronizarse con el pulso de reloj.

Cuando el circuito es tan grande que no se puede asegurar que el pulso de reloj llegue simultáneamente a todos los flip flop's que lo forman.

Cuando se desea que el circuito opere tan rápido como sea posible, sin tener que esperar a un pulso de reloj después de que cambian las entradas.

En los casos anteriores, es necesario utilizar circuitos secuenciales asíncronos, los cuales caen en la categoría de *circuitos modo nivel*. En ellos, la operación no está sincronizada por un pulso de reloj y cuando alguna entrada cambia, el estado del circuito puede cambiar casi inmediatamente sin tener que esperar a un pulso de reloj. Evidentemente, si el circuito está formado por varios flip flop's, no hay garantía de que todos cambien simultáneamente. Así que el diseño de los circuitos asíncronos presenta más dificultades que el de los circuitos síncronos, debido al problema de sincronía involucrado, resultante de retardos desiguales a través de distintas trayectorias en el circuito.

Los circuitos secuenciales asíncronos pueden estar construidos alrededor de ***flip flop's R-S***, los cuales no requieren de un pulso de sincronía llamado reloj para pasar de un estado a otro. Pero también pueden estar formados exclusivamente por compuertas, sin usar flip flop's u otros elementos de memoria, incluyendo solamente trayectorias de retroalimentación asociadas a ***retardos*** usados para distinguir entre un estado presente (*a la salida del retardo*) y un estado siguiente (*a la entrada del retardo*).

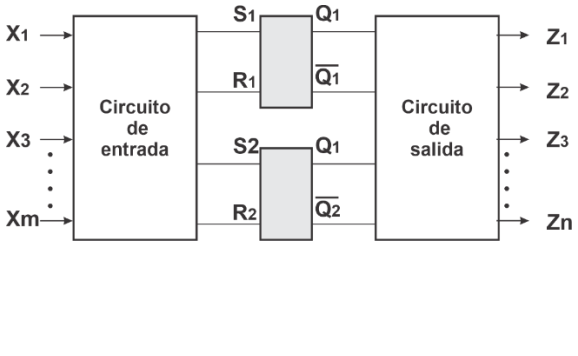


FIg. 4. Circuito asíncrono con flip flop's R-S

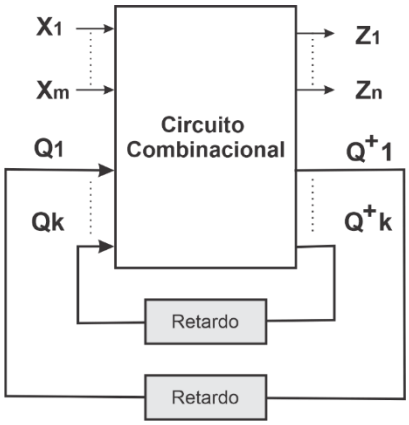


Fig. 5. Circuito asíncrono sin flip flop's

Cada combinación de valores de las entradas al circuito ***x's*** y de las salidas de memoria ***Q's*** (Las salidas de los flip flop's R-S, cuando el circuito los incluye o las salidas en los retardos de las ramas de retroalimentación, cuando el circuito no tiene flip flop's), se llama ***estado total*** del circuito: $x_1,\ldots x_m\ Q_1,\ldots Q_k$. El estado de los flip flop's o de las salidas de los retardos ***Q's*** exclusivamente, se llama ***estado interno*** del circuito: $Q_1,\ldots Q_k$. Cuando el ***estado interno siguiente*** en un circuito, es el mismo que el ***estado interno presente*** ($Q_1^+\ldots Q_k^+ = Q_1\ldots Q_k$), el ***estado total*** correspondiente es ***estable***. Si, por el contrario, los estados internos siguiente y presente son diferentes ($Q_1^+\ldots Q_k^+ \neq Q_1\ldots Q_k$), el ***estado total*** correspondiente es ***inestable***.

Cuando en un circuito, un cambio en alguna entrada hace que éste pase de un estado estable a una serie de estados inestables sin alcanzar ningún estado estable hasta que haya un nuevo cambio en la entrada, se dice que el circuito ha caído en una condición llamada ***ciclo***.

Cuando dos o más variables de un estado interno (flip flop's, en su caso) deben cambiar en respuesta a un solo cambio de entrada, el resultado es una ***condición de carrera*** entre estas variables o entre los flip flop's. Si el estado estable resultante que se alcanza es el mismo sin importar en qué orden cambiaron las variables del estado interno o los flip flop's en su caso, la carrera es ***no-crítica***. Por el contrario, si el estado resultante alcanzado es diferente de acuerdo con que variable cambia primero, entonces la carrera es ***crítica***.

El diseño de los circuitos secuenciales asíncronos debe ser tal que evite tanto las condiciones de ***ciclos*** como las de ***carreras.*** De lo contrario, el comportamiento de estos circuitos sería impredecible y, en consecuencia, no serían capaces de proporcionar respuestas satisfactorias y coherentes a los problemas planteados

Diseño de Sistemas Secuenciales Asíncronos Modo Nivel

Objetivo. - Este libro tiene el propósito de aplicar un método sistemático para diseñar sistemas secuenciales asíncronos en ejemplos, en donde su implementación puede realizarse en un Controlador Lógico Programable (PLC) por medio del diagrama escalera o implementarlo en un Dispositivo Lógico Programable (PLD) utilizando las ecuaciones obtenidas en un Lenguaje de Descripción de Hardware o basados en el diagrama esquemático utilizando Captura Esquemática.

Definición:

Sistema Secuencial Asíncrono: es aquel sistema en donde los valores de salidas (Z1....Zn) no dependen únicamente de las combinaciones de entrada (X1…Xn) sino también de la salida (Y1…Yp) y los cambios de estado dependen de un cambio en la entrada (X1…Xn) sin incluir una señal adicional de sincronía, tal como se muestra en la figura 6.

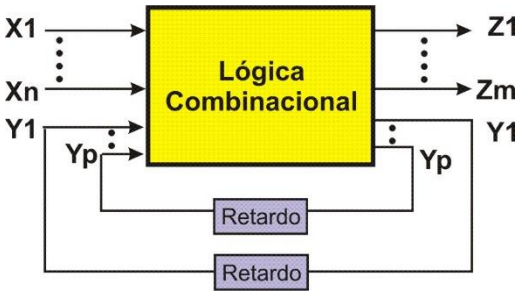


Fig. 6. Descripción a Bloques de un sistema secuencial asíncrono

Otra forma de describir estos sistemas es que son circuitos lógicos combinacionales con retroalimentación.

Los recursos con los que podemos representar el comportamiento de un sistema secuencial asíncrono son:

- Diagrama de Tiempos.
- Diagrama de Transición con la descripción del problema.

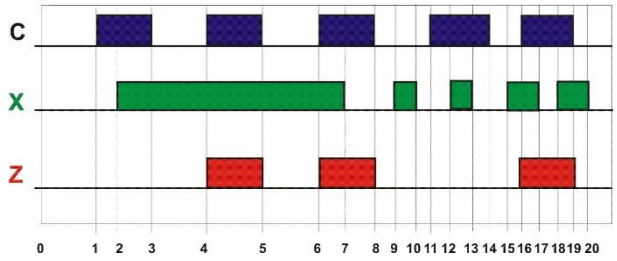


Fig. 7. Diagrama de Tiempos

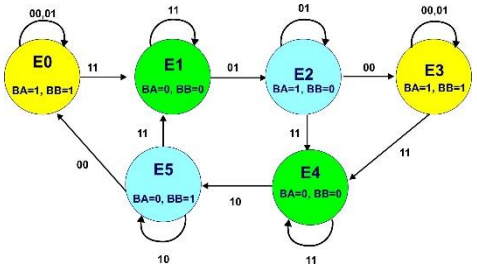


Fig. 8. Diagrama de transición

Características de los sistemas secuenciales asíncronos:

- No necesita del reloj como señal de control.
- La salida cambia cuando una variable de entrada cambia (por eventos).
- Solo está permitido un cambio de variable de entrada por evento.
- Se implementan a partir de operadores And, Or y Not, utilizando retroalimentación o a partir de FF RS básico (sin reloj).
- No requieren de FF's estándar (JK, T o D).
- Más rápidos en su respuesta por no depender de una señal adicional de sincronía (Clk), sólo de un cambio de nivel.
- Directamente adaptable a la implementación de PLC por medio de un circuito escalera.

Definición de Estados

Estado o evento: Una condición o situación dada, durante la cual satisface una condición, realiza una actividad o está esperando un evento.

Estado Total: Descrito por la combinación de los valores de las variables de entrada (X, Y) y los valores de las variables que determinan el estado presente (Q1, Q2).

Estado Interno: Descrito por los valores de los Estados Presentes (Q1, Q2).

Estado Estable: El Estado interno próximo o siguiente es igual al Estado interno presente $Q1^{+}\ Q2^{+} = Q1\ Q2$.

Estado Inestable: El Estado interno próximo o siguiente es distinto al Estado interno presente $Q1^{+}\ Q2^{+} \neq Q1\ Q2$.

Modelos de sistemas secuenciales y sus representaciones.

Se puede clasificar en dos modelos que a continuación se describen:

Máquina de **Moore,** en donde la salida solo depende del estado presente o de los valores de Q.

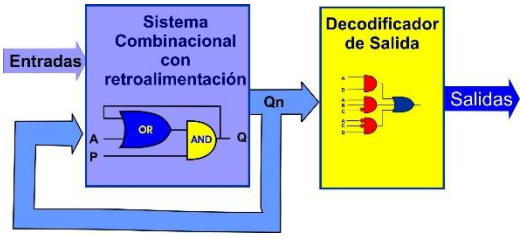


Fig. 9. Modelo de la máquina de Moore

Máquina de **Mealy,** en el que la salida depende tanto de la entrada como del estado presente.

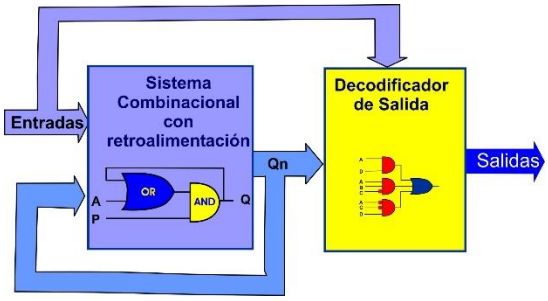


Fig. 10. Modelo de la máquina de Mealy

Diagrama de Transición

Es una forma muy explícita de especificar el comportamiento de un sistema secuencial, este diagrama se describe por medio de eventos o estados, transiciones, entradas y salidas.

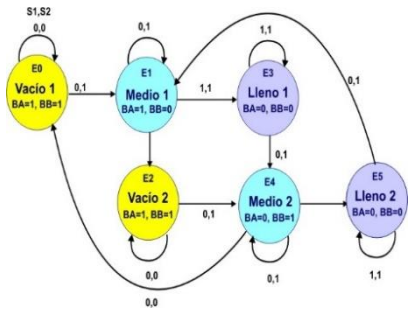


Fig. 11. Diagrama de Transición

Un diagrama de transición se compone de:

Estados o Eventos: Una condición o situación en la vida de un objeto, durante la cual satisface una condición, realiza una actividad o está esperando un evento.

Figura 12. Estado o evento

Transición en el mismo estado: Una entrada X cuyo estado próximo es el mismo que el anterior.

Figura 13. Transición al mismo estado

Transición entre dos estados: Una relación entre estados que indica que un objeto que está en el primer estado realizará una acción especificada y, entrará en el segundo estado cuando un evento **Y** especificado ocurra y unas condiciones especificadas sean satisfechas.

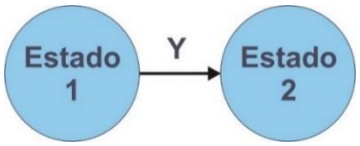


Figura 14. Transición entre dos estados

Entradas: Combinaciones que establecen un cambio de evento**.**

Salidas: Valores combinacionales que determinan un evento.

Y

En una tabla de flujo (o transiciones) se tiene la misma información que en un diagrama de transiciones, pero está organizada de forma tabular. Si en la construcción de ella se establecen las siguientes condiciones:

a). - Solo debe de haber un estado estable por fila.

b). - Solo cambia una variable de entrada a la vez.

Ésta tabla recibe el nombre de **Tabla de Flujo Primitiva**.

En una tabla de flujo primitiva se pueden considerar eventos no descritos en el diagrama de tiempo o en el diagrama de transición de tal suerte que se asegura la consideración de todos los eventos posibles en el funcionamiento u operación de un diseño.

Tabla de flujo primitiva: Es aquella que tiene únicamente un estado estable por renglón, con salidas especificadas sólo para estados estables. (1)

1 pag. 248 de principios de Diseño Lógico Digital Norman Balabanian, Bradley Carlson

Método de Diseño de un Sistema Secuencial Asíncrono:

A continuación, se muestran 14 pasos sugeridos para el desarrollo y diseño de sistemas secuenciales asíncronos:

1.- Especificar el Sistema.

2.- Tabla de Flujo Primitiva.

3.- Eliminación de estados redundantes.

4.- Mezcla de Filas.

5.- Expandir tabla de salidas.

6.- Tabla de estados internos.

7.- Asignación de Valores a los Estados.

8.- Tabla de Estados totales.

9.- Completar Tabla de Salidas.

10.- Obtención de las ecuaciones mínimas.

11.- Elaborar el Archivo en formato ABEL-HDL.

12.- Simulación.

13.- Representación Gráfica de los resultados.

14.- Implementación o construcción del prototipo.

1.- Especificar el Sistema.

Por medio de un Diagrama de Tiempos (gráfica) o un Diagrama de transición (eventos) se puede describir el comportamiento esperado del sistema secuencial.

2.- Tabla de Flujo Primitiva.

Tabla que tiene únicamente un estado estable por renglón, con salidas especificadas sólo para estados estables, en donde son incluidos todos los posibles eventos del sistema.

3.- Eliminación de estados redundantes o equivalentes.

Se pueden reducir el número de estados o eventos si estos son equivalentes, para ser considerados como equivalentes se debe de cumplir con las siguientes condiciones:

a) Son estados estables en la misma columna (misma combinación de entradas).

b) Tienen la misma salida.

c) Sus estados siguientes son equivalentes.

4.- Mezcla de Filas.

Mediante un análisis de cada una de las hileras se pueden mezclar entre ellas siempre y cuando no exista algún conflicto, entendiéndose por conflicto la ocupación simultánea de una columna por dos estados diferentes. La salida no se considera como un factor de conflicto en la mezcla de filas. Esto es, dos filas con salidas diferentes pueden mezclarse.

En este proceso la tabla de flujo primitiva se transforma en una tabla de estados totales.

5.- Expandir tabla de salidas.

Si para una hilera se tiene la posibilidad de mezclar filas, pero los estados estables tienen salidas diferentes es conveniente expandir las salidas.

En el caso de no ser necesario la expansión de salidas el sistema adopta el modelo de la máquina de Moore y en el caso de ser necesario el sistema adopta el modelo de la máquina de Mealy.

6.- Tabla de estados internos.

Convertir la tabla de estados totales obtenida al mezclar las filas en una tabla de estados internos.

7.- Asignación de Valores a los Estados.

Para cumplir con las transiciones descritas se asignas estados de modo que entre ellos solo cambie un valor

8.- Tabla de Estados Totales.

En esta tabla se sustituyen los estados internos por el valor de la asignación propuesta en el paso anterior**.**

9.- Completar Tabla de Salidas.

En algunos casos las salidas no están completamente definidas y esto puede generar valores transitorios no convenientes para el sistema por tal razón es necesario asignar un valor a la salida de modo que no se presente el transitorio.

10.- Obtención de las ecuaciones por medio de minimización.

a) Manipulación algebraica

b) Mapas de Karnaugh

c) Software LogicAid recurso para la obtención de las ecuaciones mínimas.

11.- Elaborar el Archivo en formato ABEL-HDL.

Por medio de esta herramienta nos permite describir el comportamiento del sistema por medio de ecuaciones, tabla de verdad, tabla de estados o la descripción del comportamiento y posteriormente efectuar la simulación para comprobar el si el funcionamiento obtenido es el deseado del sistema.

12.- Simulación.

Por lo general la forma de presentarse es por medio de un diagrama de tiempo en donde se incluyen las entradas y salidas, así como los valores de los estados (Q's).

O utilizar programas que nos permitan comprobar los resultados ya sea a partir de un circuito o en un PLD como PROTEUS.

13.- Representación gráfica.

a) Diagrama esquemático.

b) Diagrama escalera.

14.- Implementación

a) PLD.

b) PLC.

c) Dispositivos TTL o CMOS.

Problema 1.1, Botón caso 1

Diseñe, efectué la simulación y construya un prototipo de un sistema secuencial asíncrono con un botón de entrada P y una salida Y.

Se establecen condiciones iniciales en las que P=0 y Y=0 en el instante T0. A partir de estas condiciones, se especifican los siguientes eventos y cambios en las salidas:

1.- Al oprimir por primera vez el botón P, la salida Y debe cambiar de cero a uno. Por lo tanto, se espera que en el instante t1, se cumpla P=1 y Y=1.

2..- Al soltar el botón (en el instante t2), la salida Y debe mantenerse en uno. Por lo tanto, en este momento, se espera que P=0 y Y=1.

3.- Al oprimir el botón por segunda vez, la señal de salida Y debe cambiar a cero. Por lo tanto, en el instante t3, se espera que P=1 y Y=0.

4.- Al soltar nuevamente el botón (P=0), la salida Y debe permanecer en cero. En este momento, se regresa a las condiciones iniciales, es decir, P=0 y Y=0, como se indica en el instante T0.

La gráfica adjunta representa visualmente estos cambios esperados en las salidas a lo largo del tiempo.

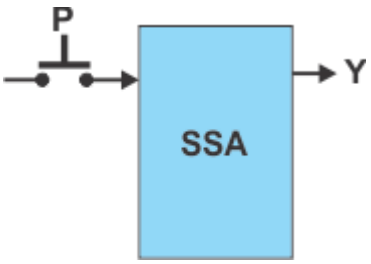


Fig. 15 Diagrama de bloques

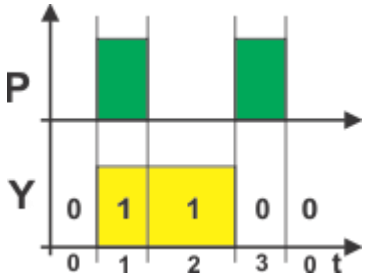


Fig. 16 Diagrama de tiempos

1.- Especificar el Sistema.

Para especificar el sistema secuencial asíncrono propuesto, se puede utilizar un diagrama de transición de estados.

Este diagrama mostrará el funcionamiento del sistema y los estados presentes (E0/0, E1/1, E2/1 y E3/0), así como los estados siguientes en función de la entrada.

A continuación, se describirá el diagrama de transición de estados correspondiente a este sistema:

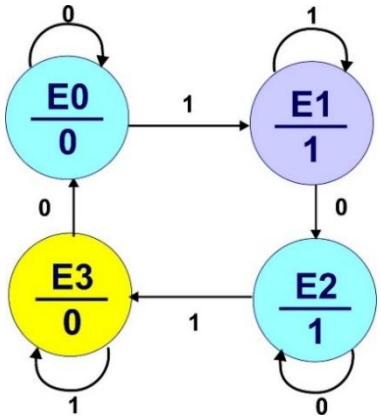
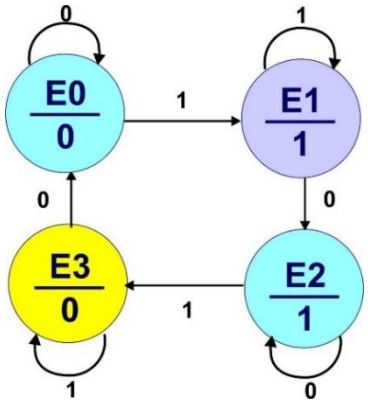


Fig. 17 Diagrama de Transición

2.- Tabla de Flujo Primitiva.

Para construir la tabla de flujo primitiva nos basamos en el diagrama de transición por ejemplo en el estado E0 sucede que con entrada cero (P=0) es estable lo que como se indica en la hilera 0 como **E0** además la salida Y para esa Hilera es cero, y con entrada uno (P=1) pasaría a un estado **E1** transitorio y que es estable en la hilera 1 que se indica como **E1** y con salida Y igual a uno como se indica en la figura abajo mostrada.



	P		Y
	0	1	
0	**E0**	E1	0
1		**E1**	1

Fig. 18 Diagrama de Transición e inicio de la tabla de flujo primitiva

Estando en el estado **E1** estable (hilera 1, columna P=1), al soltar el botón debe cambiar a un estado **E2** transitorio, que es estable **E2** en la hilera 2 y columna P=0, con salida Y=1.

	P		Y
	0	1	
0	**E0**	E1	0
1	E2	**E1**	1
2	**E2**		1

Estando en el estado **E2** estable (hilera 2, columna P=0), al oprimir el botón debe cambiar a un estado **E3** transitorio, que es estable **E3** en la hilera 3 y columna P=1 con salida Y=0.

	P		Y
	0	**1**	
0	**E0**	E1	**0**
1	E2	**E1**	**1**
2	**E2**	E3	**1**
3		**E3**	**0**

Estando en el estado **E3** estable (hilera3, columna P=1), al soltar el botón pasará a un estado transitorio **E0** y regresará al estado estable **E0**, con lo que ya tenemos definidas todas las posibles transiciones que se muestran en la siguiente tabla de flujo primitiva:

	P		Y
	0	**1**	
0	**E0**	**E1**	**0**
1	**E2**	**E1**	**1**
2	**E2**	**E3**	**1**
3	**E0**	**E3**	**0**

Fig. 19 tabla de flujo primitiva completa

3.- Eliminación de estados redundantes.

No necesariamente los estados establecidos en la tabla de flujo primitiva son indispensables, puede haber estados redundantes. Por ello es necesario identificarlo (si los hay) y eliminarlos para obtener un sistema óptimo.

Ha y que considerar que un estado es redundante solo si existe uno equivalente.

Dos estados son equivalentes si:

a) Son estados estables en la misma columna (misma combinación de entradas).

b) Tienen la misma salida.

c) Sus estados siguientes son equivalentes.

	P		Y
	0	1	
0	**E0**	E1	0
1	E2	**E1**	1
2	**E2**	E3	1
3	E0	**E3**	0

En la tabla anterior observamos que para el valor de entrada P=0 **E0** y **E2** son estables, pero no tienen la misma salida, y para la entrada P=1 **E1** y **E3** son estables, pero no tienen la misma salida, por lo que se concluye que para este ejemplo no hay reducción de Estados.

4.- Mezcla de Filas.

Una vez eliminados los estados redundantes, las filas o hileras pueden mezclarse para reducir la tabla.

Dos filas o más se pueden mezclar siempre y cuando no hay ningún conflicto sobre qué estado debe ocupar cada columna.

Entendiéndose por conflicto la ocupación simultánea de una columna por dos estados diferentes.

La salida no se considera como un factor de conflicto en la mezcla de filas. Esto es, dos filas con salidas diferentes pueden mezclarse.

Con el propósito de tener una visualización completa sobre las posibilidades de mezcla de las filas, se construye un diagrama de mezcla.

El diagrama de mezcla consiste en asignar un punto por cada fila y se unen esos puntos por líneas cuando estos pueden mezclarse. En este problema no es posible la mezcla de filas.

5.- Expandir Tabla de Salidas

No es necesario este paso, ya que no se realizó mezcla de filas.

6.-Tabla de Estados Internos.

Para obtener la tabla de estados internos sustituimos los estados estables de la primera hilera por la letra “a”, así mismo la segunda hilera por la letra b y así hasta la última hilera por d.

	P		Y
	0	1	
a	**E0**	E1	0
b	E2	**E1**	1
c	**E2**	E3	1
d	E0	**E3**	0

Sustituyendo los nombres de E0 $\rightarrow$ a, E1 $\rightarrow$ b, E2 $\rightarrow$ c y E3 $\rightarrow$d obtenemos la tabla de estados internos:

	P		Y
	0	1	
a	**a**	b	0
b	c	**b**	1
c	**c**	d	1
d	a	**d**	0

Fig. 20 tabla de estados internos

7.- Asignación de Valores a los Estados.

Para la asignación de valores a los estados se analiza individualmente cada una de las columnas, partiendo de los estados no estables hacia el valor de la columna base. En este caso tenemos que:

La Columna 0) (b $\rightarrow$ c), (d $\rightarrow$ a), como se muestra en la siguiente tabla:

	P			Y
		0	1	
a		a	b	0
b	→	c	b	1
c		c	d	1
d	→	a	d	0

La Columna 1) (a $\rightarrow$ b), (c $\rightarrow$ d), como se muestra en la siguiente tabla:

	P			Y
	0		1	
a	a	→	b	0
b	c		b	1
c	c	→	d	1
d	a		d	0

Para la asignación de valores diferentes a las cuatro variables a, b, c y d se proporcionada a cada una de ella un valor diferente de dos dígitos binarios (con las posibles combinaciones de dos variables binarias 00, 01, 10 y 11 cubrimos la posible asignación).

Para este problema es requisito en la asignación que **b** sea contigua con **c** (de modo que, entre b y c, solo cambie uno de los dos valores asignados) además **b** también debe de ser contigua con a, de la misma forma a sea contigua con **d** y además con b.

Una forma de visualizar fácilmente las diferentes posibilidades de asignación y cumplir con las condiciones (b → c), (d → a), (a → b), (c → d), es usar la estructura de un mapa de Karnaugh, en este caso de dos variables de modo que entre cuadros contiguos solo exista un cambio, *para este problema se pueden tener* ***ocho diferentes posibles asignaciones que cumplen con el requisito antes mencionado como se muestra a continuación***:

Por lo que el problema tiene ocho posibles resultados todos válido.

Asignación 1

Q1 \ Q2	0	1
0	**a**	**d**
1	**b**	**c**

		Q1	Q0
a	=	**0**	**0**
b	=	**0**	**1**
c	=	**1**	**1**
d	=	**1**	**0**

Asignación 2

Q1 \ Q2	0	1
0	**d**	**c**
1	**a**	**b**

		Q1	Q0
a	=	**0**	**1**
b	=	**1**	**1**
c	=	**1**	**0**
d	=	**0**	**0**

Asignación 3

Q1 \ Q2	0	1
0	**c**	**b**
1	**d**	**a**

		Q1	Q0
a	=	**1**	**1**
b	=	**1**	**0**
c	=	**0**	**0**
d	=	**0**	**1**

Asignación 4

Q1 \ Q2	0	1
0	**b**	**a**
1	**c**	**d**

		Q1	Q0
a	=	**1**	**0**
b	=	**0**	**0**
c	=	**0**	**1**
d	=	**1**	**1**

Asignación 5

Q1 \ Q2	0	1
0	**a**	**b**
1	**d**	**c**

		Q1	Q0
a	=	**0**	**0**
b	=	**1**	**0**
c	=	**1**	**1**
d	=	**0**	**1**

Asignación 6

Q1 \ Q2	0	1
0	**d**	**a**
1	**c**	**b**

		Q1	Q0
a	=	**1**	**0**
b	=	**1**	**1**
c	=	**0**	**1**
d	=	**0**	**0**

Asignación 7

Q1 \ Q2	0	1
0	**c**	**d**
1	**b**	**a**

		Q1	Q0
a	=	**1**	**1**
b	=	**0**	**1**
c	=	**0**	**0**
d	=	**1**	**0**

Asignación 8

Q1 \ Q2	0	1
0	**b**	**c**
1	**a**	**d**

		Q1	Q0
a	=	**0**	**1**
b	=	**0**	**0**
c	=	**1**	**0**
d	=	**1**	**1**

8.- Tabla de Estados totales.

Sustituyendo los valores de la asignación 1, obtenemos la siguiente tabla de estados totales**:**

Q1Q0	P		Y
	0	1	
00	00	01	0
01	11	01	1
11	11	10	1
10	00	10	0

De la tabla de estados totales trasladaremos los valores a una tabla de Verdad:

Q1Q0	P		Y
	0	1	
00	00	01	0
01	11	01	1
11	11	10	1
10	00	10	0

Tabla de Estados Totales

m	P	Q1	Q0	Q1+	Q0+	Y
0	0	0	0	**0**	**0**	**0**
1	0	0	1	**1**	**1**	**1**
2	0	1	0	**0**	**0**	**0**
3	0	1	1	**1**	**1**	**1**
4	1	0	0	**0**	**1**	**0**
5	1	0	1	**0**	**1**	**1**
6	1	1	0	**1**	**0**	**0**
7	1	1	1	**1**	**0**	**1**

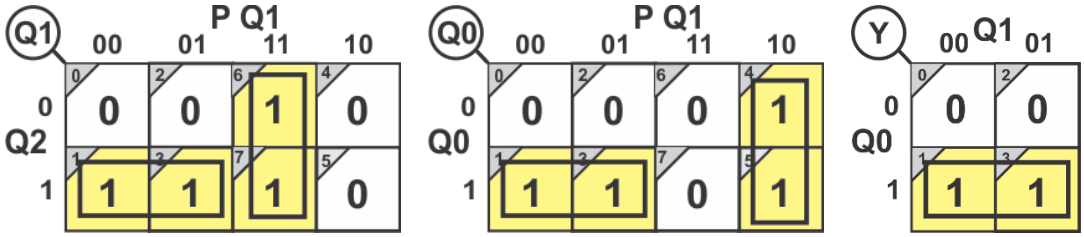
Tabla de Verdad

9.- Completar Tabla de Salidas.

No es necesario completar la tabla de salidas.

10.- Obtención de las ecuaciones mínimas.

Partiendo de la tabla de Verdad podemos obtener las ecuaciones mínimas utilizando como recurso los mapas de karnaugh:



Q1= P Q1 + **P'** Q0 Q0 = P **Q1'** + **P'** Q0 Y = Q0

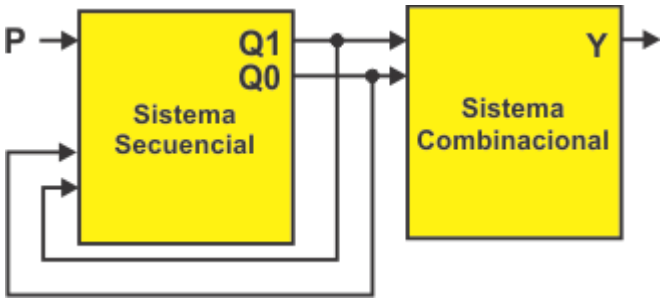


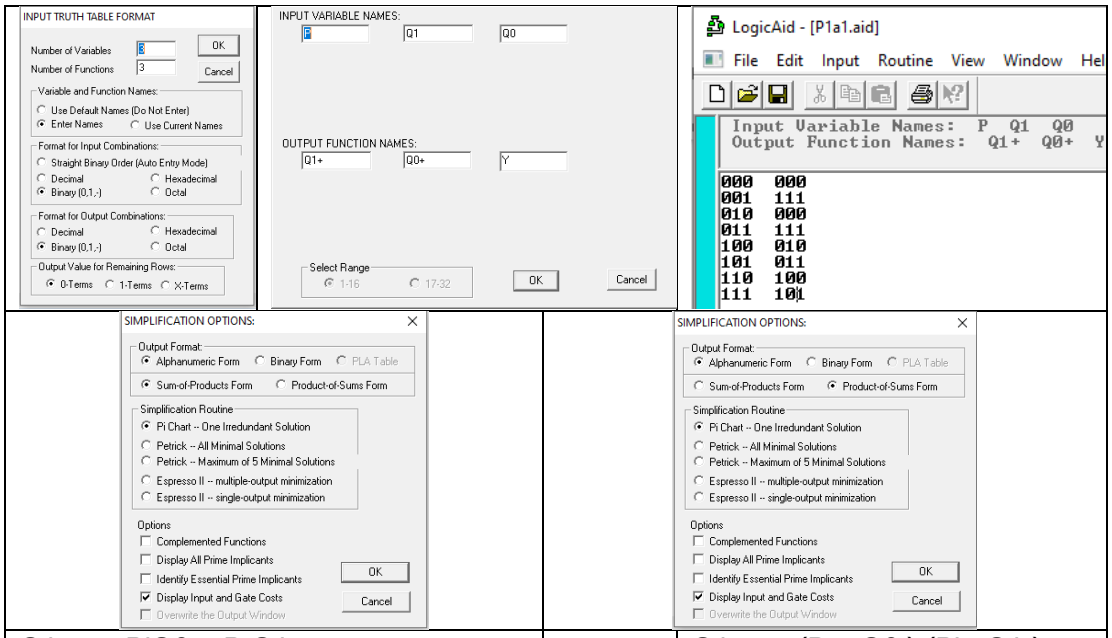
Fig. 21 Bloques del sistema

***Fig. 18 Diagrama del modelo máquina de More indicando los módulos secuencial y combinacional*.**

Para comprobar los resultados podemos hacer uso del programa de aplicación (Software) LogicAid y obtener dos resultados que se pueden implementar en las formas And/Or y Or/And

Ecuaciones mínimas con el uso de LogicAid

Partiendo de los valores de la tabla de verdad se pueden obtener dos resultados mínimos equivalentes el primero en la forma And/Or o llamada SOP (Suma de productos) el segundo Or/And (Productos de Suma) como se muestra a continuación.



Q1+ = P'Q0 + P Q1 Input Cost = 6 Gate Cost = 3 Q0+ = P'Q0 + P Q1' Input Cost = 6 Gate Cost = 3 Y = Q0 Input Cost = 0 Gate Cost = 0		Q1+ = (P + Q0) (P'+ Q1) Input Cost = 6 Gate Cost = 3 Q0+ = (P + Q0) (P'+ Q1') Input Cost = 6 Gate Cost = 3 Y = (Q0) Input Cost = 0 Gate Cost = 0

And/Or		**Or/And**
Q1= **P'** Q0 + P Q1		Q1= (P + Q0) (**P'**+ Q1)
Q0= **P'** Q0 + P **Q1'**		Q0= (P + Q0) (**P'**+ **Q1'**)
Y = Q0		Y = Q0

Ocho Formas Estándar

Existen ocho formas reconocidas como estándar para presentar un circuito y las primeras cuatro pertenecen a la Suma de Productos (SOP) y las cuatro restantes al Producto de Sumas (POS)

Las formas estándar para expresar algebraicamente una función booleana son:

1.- Suma de Productos SOP (And/OR) partiendo de los Minitérminos.

2.-Productos de Sumas POS (Or/And) partiendo de los Maxitérminos. Para obtener las ocho formas estándar es necesario aplicar el Teorema de D´ Morgan

	SOP			POS
1	And/Or		5	Or/And
2	Nor/Or		6	Nand/And
3	Or/Nand		7	And/Nor
4	Nand/Nand		8	Nor/Nor

El nombre de la forma estándar proviene del orden en que se realizan las operaciones, por ejemplo: en la siguiente figura en el primer nivel del lado izquierdo se realiza la operación And y en el segundo nivel la operación Or por lo que esta forma seria la **And/Or**.

Si aplicamos este concepto tendremos entonces ocho posibles soluciones o circuitos equivalentes en la implementación para cada una de las asignaciones.

En la siguiente figura se muestran los dos ciclos correspondientes a las ocho formas estandar, los primeros cuatro pertenecientes a a la suma de productos SOP, y el segundo ciclo a los productos de suma POS.

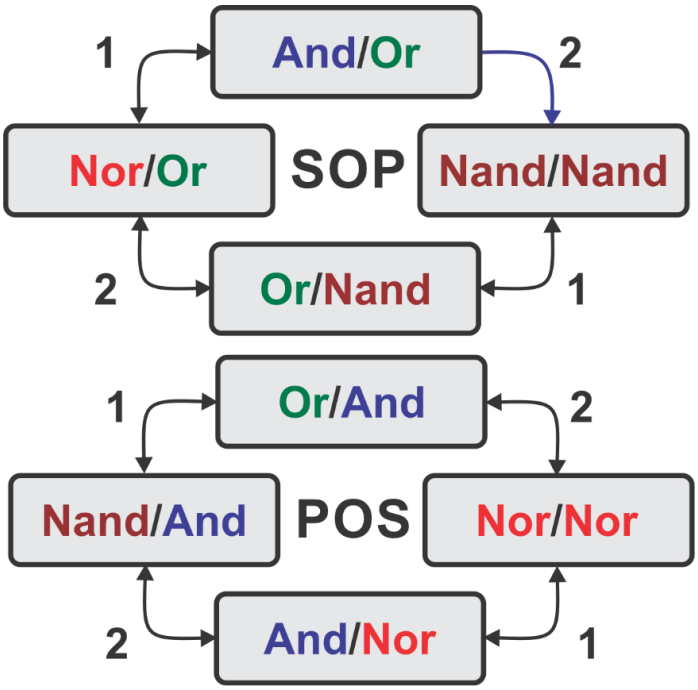
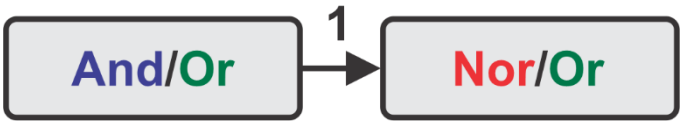
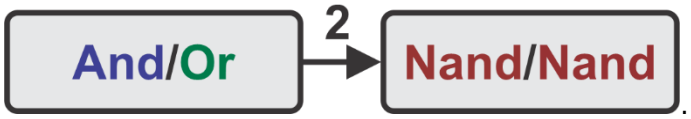


Fig. 21 Ciclos de las ocho formas estándar

Partiendo de la forma And/Or, si aplicamos el teorema de D'Morgan solo a la operación And (1 primer nivel) se obtiene la forma Nor/Or.



Partiendo de la misma forma And/Or, Ahora si aplicamos el teorema de D'Morgan solo a la operación Or (2 segundo nivel) se obtiene la forma Nand/Nand



.

Forma And/Or
Q1= (P + Q0) (**P'**+ Q1), Q0=(P + Q0) (**P'**+ **Q1'**), Y = Q0

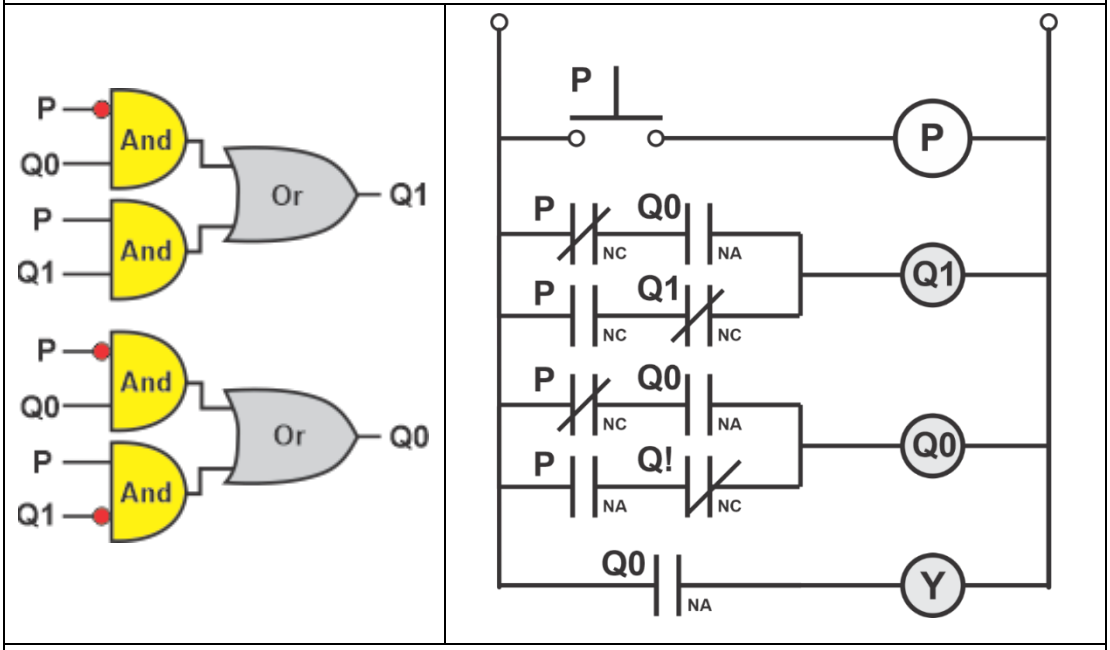



Fig. 22 Ecuación, representación gráfica y diagrama escalera de la forma And/Or

A continuación, se muestran los dos ciclos de las 8 formas estándar

Partiendo de la forma And/Or del ciclo SOP para obtener la forma Nand/Nand se aplica el teorema de D'Morgan solo a la operación Or cambiándola por **Nand** con las entradas negadas y al negar las entradas por consecuencia se cambiaría la operación And del primer nivel por Nand, como se muestra en la siguiente figura.

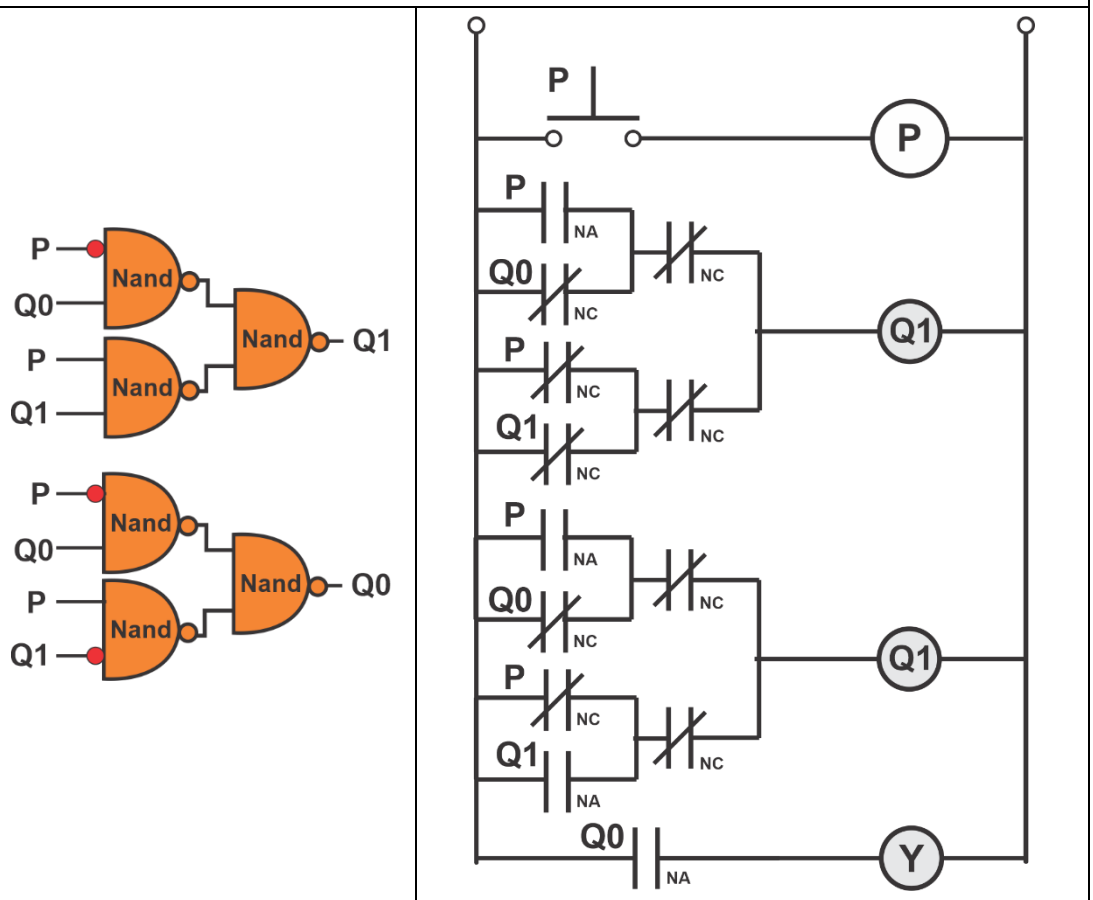
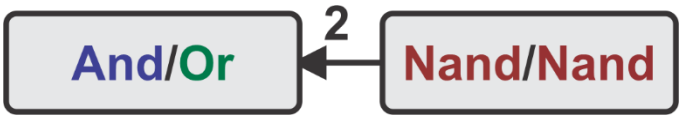
Forma Nand/Nand
Q1= **[** (P Q0**)' (P'** Q1**)']'**, Q0=**[(**P Q0**)' (P' Q1')']'**, Y = Q0


Fig. 23 Ecuación, representación gráfica y diagrama escalera de la forma Nand/Nand

Si a la forma **Nand/Nand** le aplicáramos el teorema de D'Morgan solo a la Nand del segundo nivel regresaríamos a la forma **And/Or**.



Partiendo de la forma And/Or y solo aplicar el teorema de D'Morgan a la operación And (1) ésta se cambia por Nor con entradas negadas y se obtiene la forma Nor/Or como se muestra a continuación:

Forma Nor/Or

Q1= (P + **Q0')'** + **(P'** + **Q1')'** , Q0= **(**P + **Q0')'** + (**P'** + Q1**)'** , Y = Q0

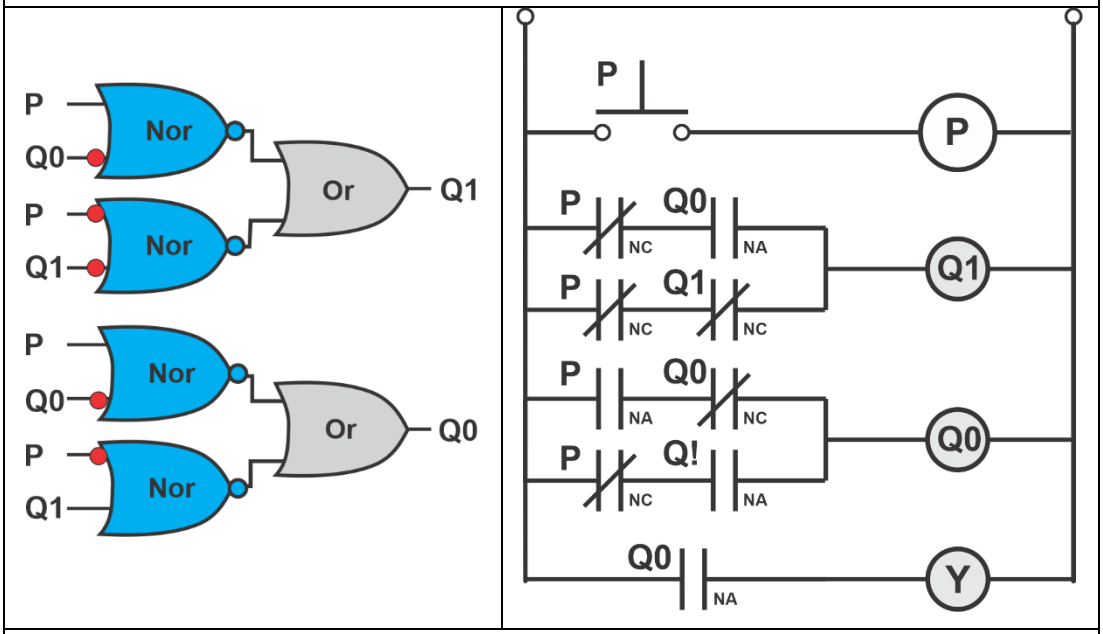


Fig. 24 Ecuación, representación gráfica y diagrama escalera de la forma Nor/Or

Para obtener la forma **Or/Nand** se puede partir de la forma **Nand/Nand** y aplicar el Teorema de D´Morgan a las Nand del primer nivel cambiando las Nand's por Or con las entradas negadas.

O también partiendo de la forma Nor/Or y aplicando el teorema D´Morgan solo a la Or se puede obtener la forma Or/Nand

Forma Or/Nand

Q1= **[** (P + **Q0'**) **(P'** + **Q1'**) **]'**, Q0=**[(**P + **Q0'**) (**P' +** Q1) **]'**, Y = Q0

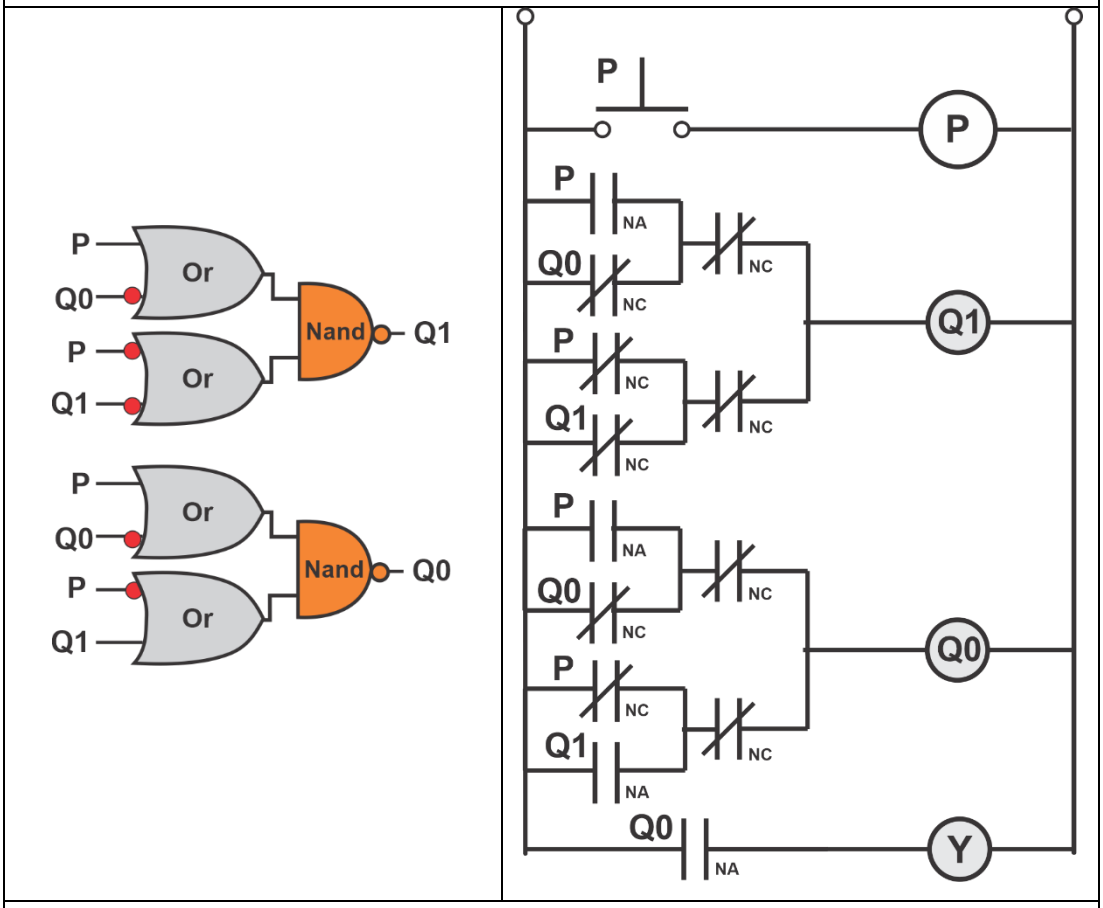


Fig. 25 Ecuación, representación gráfica y diagrama escalera de la forma Or/Nand

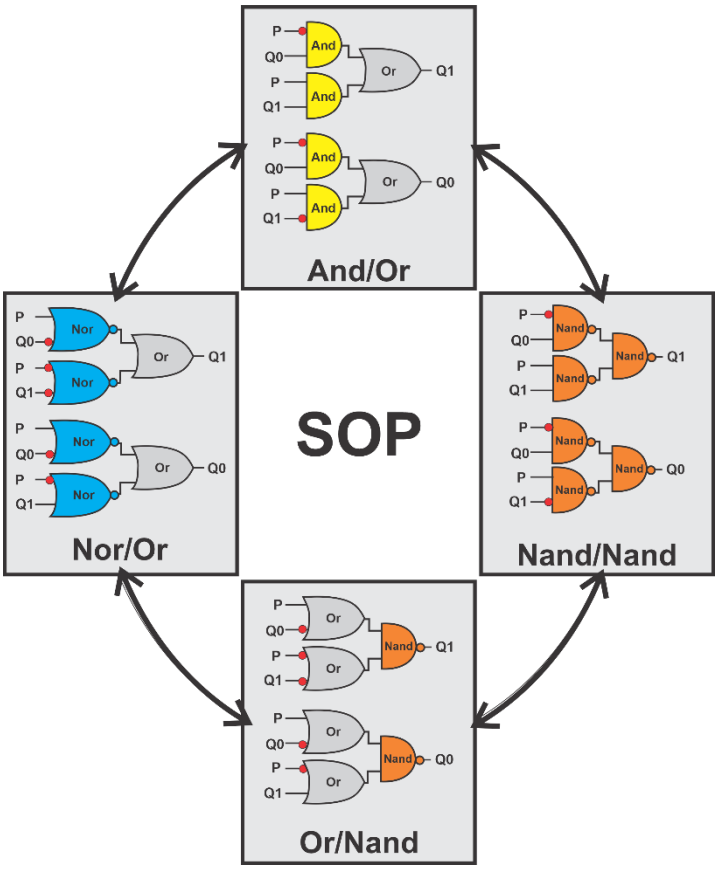


Fig. 26 representación gráfica de las primeras cuatro formas estándar

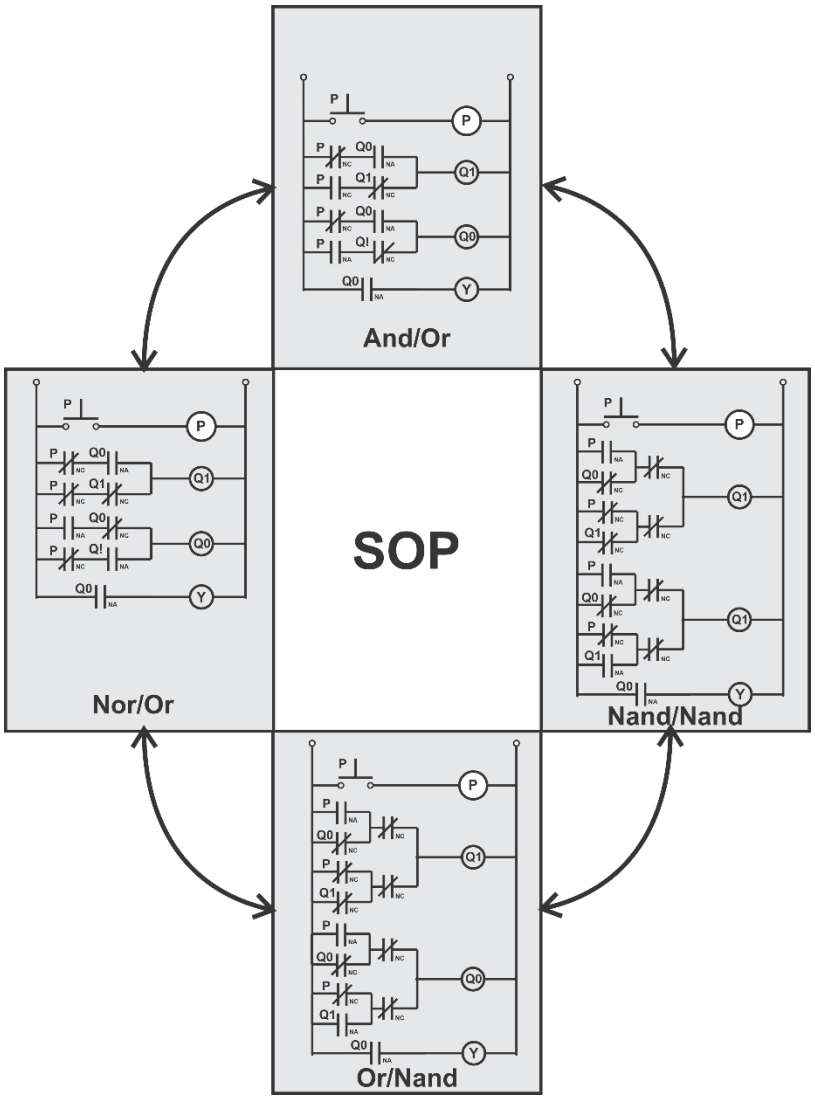


Fig. 27 Diagrama escalera de las primeras cuatro formas estándar

Las cuatro formas estándar pertenecientes a los productos de suma POS, se obtienen a partir de la forma Or/AND

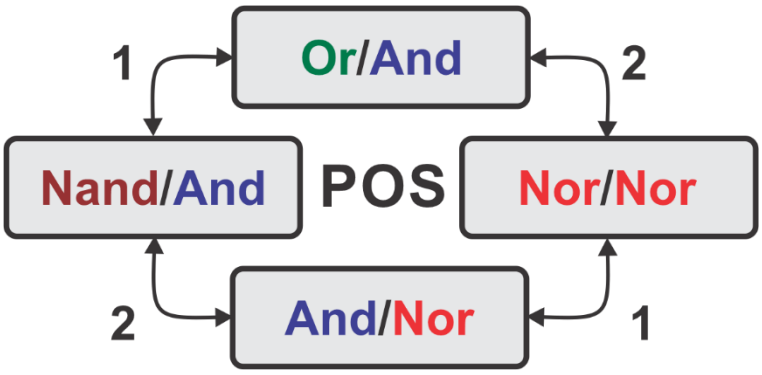


Fig. 28 Ciclo de las cuatro formas estándar correspondientes a POS

Forma Or/And	
Q1= (P + Q0) (**P'**+ Q1), Q0=(P + Q0) (**P'**+ **Q1'**), Y = Q0	

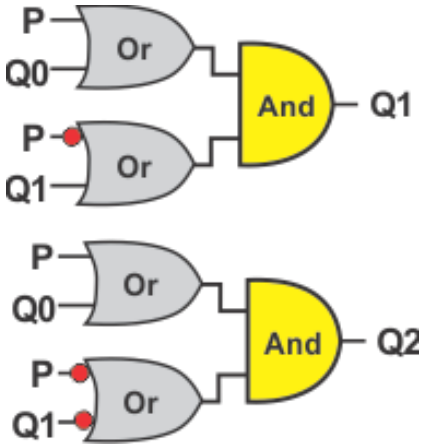
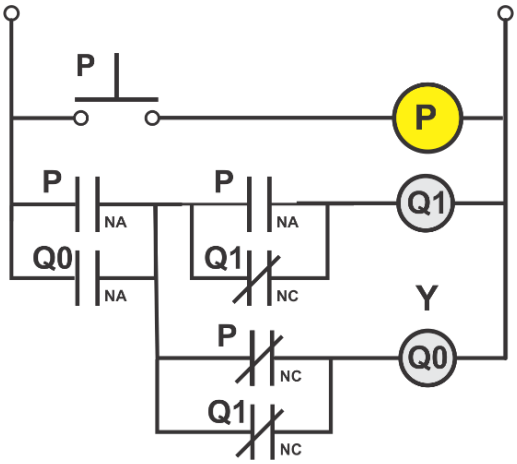

	

Diagrama esquemático Or/And	**Diagrama escalera Or/And**

Fig. 29 Ecuación, representación gráfica y diagrama escalera de la forma Or/And

Aplicando el teorema de D' Morgan solo a la operación And del segundo nivel se obtiene la forma Nor/Nor

Forma Nor/Nor
Q1= (P + Q0) (**P'**+ Q1), Q0=(P + Q0) (**P'**+ **Q1'**), Y = Q0

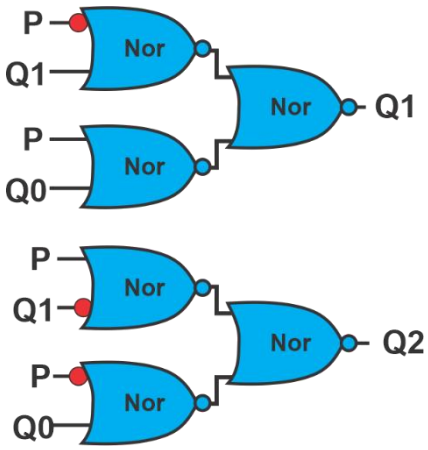



Fig. 29 Ecuación, representación gráfica y diagrama escalera de la forma Nor/Nor

Forma And/Nor
Q1= (P + Q0) (**P'**+ Q1), Q0= (P + Q0) (**P'**+ **Q1'**), Y = Q0

Forma Nand/And
Q1= (P + Q0) (**P'**+ Q1), Q0= (P + Q0) (**P'**+ **Q1'**), Y = Q0

11.- Archivo de código en formato ABEL-HDL

Por medio del código ABEL-HDL en el programa IspLEVER Classic se puede representar el comportamiento del sistema secuencial por medio de las ecuaciones con el comando **equations** o también por tabla de verdad utilizando el comando **Truth_table**, comprobando el funcionamiento correcto del sistema por medio de una imagen de la simulación del sistema (Test_Vectors) o en el programa proteus utilizando el archivo de formato JEDEC con extencion .JED, con el cual tambiense programa un PLD y construir un prototipo.

Código ABEL-HDL con ecuaciones		Código ABEL-HDL con Tabla de verdad
MODULE Puno "Asignación 1 equations		MODULE Puno " Asignación 1 Truth_table
"entrada P pin 1; "salida Y, Q1, Q0 pin 19, 14, 15 istype 'com';		"entrada P pin 1; "salida Y, Q1, Q0 pin 19, 14, 15 istype 'com';
Equations Q1=P&Q1#!P&Q0; Q0=P&!Q1#!P&Q0; Y=Q0;		**Truth_table** **([P,Q1,Q0]->[Q1,Q0,Y])** **[0,0,0]->[0,0,0];** **[0,0,1]->[1,1,1];** **[0,1,0]->[0,0,0];** **[0,1,1]->[1,1,1];** **[1,0,0]->[0,1,0];** **[1,0,1]->[0,1,1];** **[1,1,0]->[1,0,0];** **[1,1,1]->[1,0,1];**
test_vectors **(P->Y)** **0->.x.;** **1->.x.;** **0->.x.;** **1->.x.;** **0->.x.;** END		**test_vectors** **(P->Y)** **0->.x.;** **1->.x.;** **0->.x.;** **1->.x.;** **0->.x.;** END

Archivo de código en formato VHDL

```
LIBRARY ieee;
LIBRARY generics;
USE ieee.std_logic_1164.ALL;
USE ieee.numeric_std.ALL;
USE generics.components.ALL;
```

```
ENTITY testbench IS
END testbench;
```

```
ARCHITECTURE behavior OF testbench IS
```

```
COMPONENT boton
PORT(
p : IN std_logic;
q0 : OUT std_logic;
q1 : OUT std_logic;
y : OUT std_logic
);
```

```
END COMPONENT;
```

```
SIGNAL p :  std_logic;
SIGNAL q0 :  std_logic;
SIGNAL q1 :  std_logic;
SIGNAL y :  std_logic;
```

```
BEGIN
```

```
uut: boton PORT MAP(
p => p,
q0 => q0,
q1 => q1,
y => y
);
```

```
-- *** Test Bench - User Defined Section ***
   tb : PROCESS
   BEGIN
      wait; -- will wait forever
   END PROCESS;
-- *** End Test Bench - User Defined Section
***
```

```
END;
```

12.- Simulaciones.

a) Test_Vectors

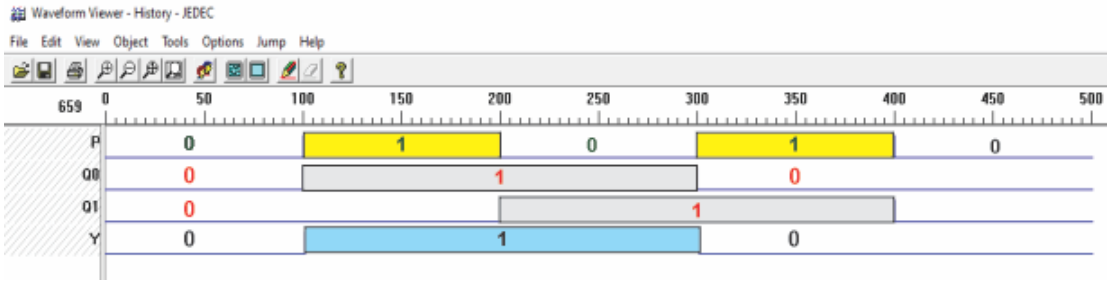


Fig. 29 Imagen de la simulación Test_ Vectors

En la figura 19 podemos observar que el valor de salida Y cumple con lo establecido en la propuesta inicial.

b) PROTEUS

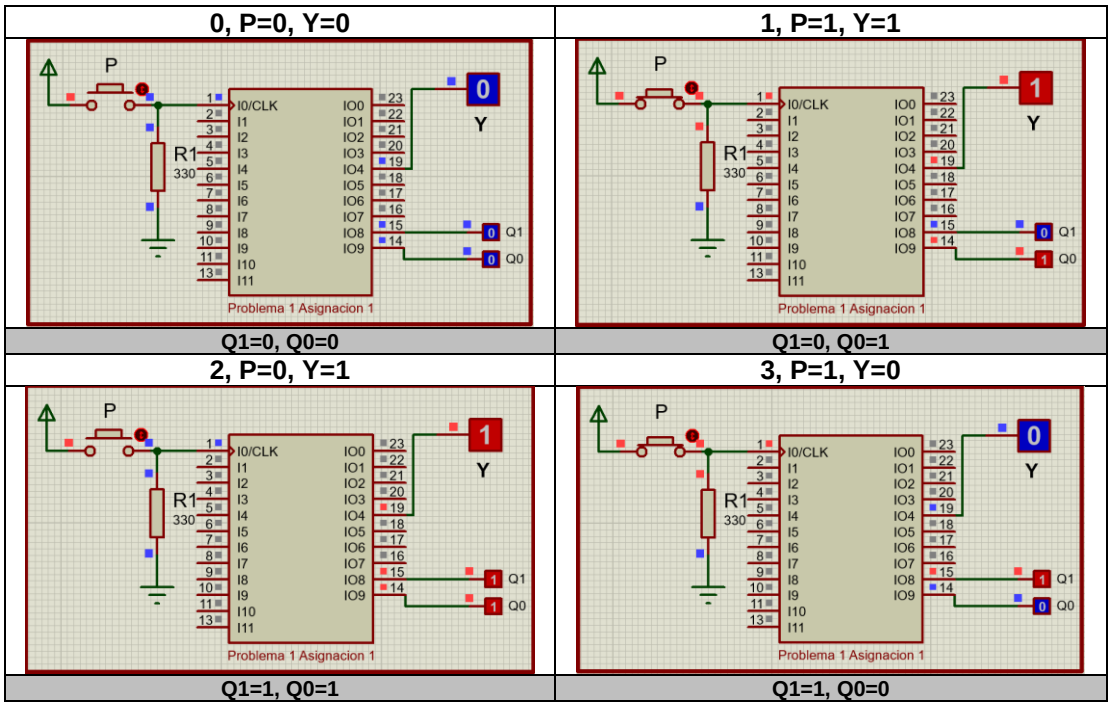
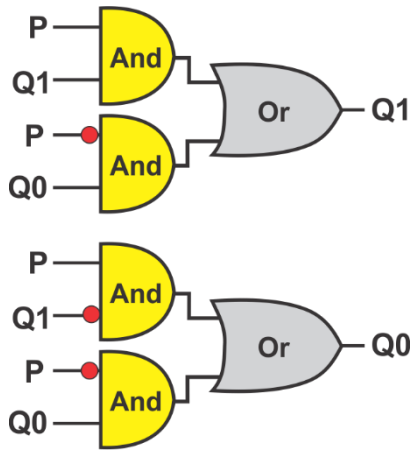
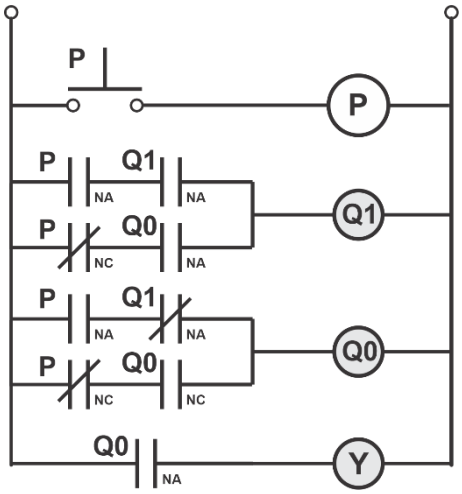
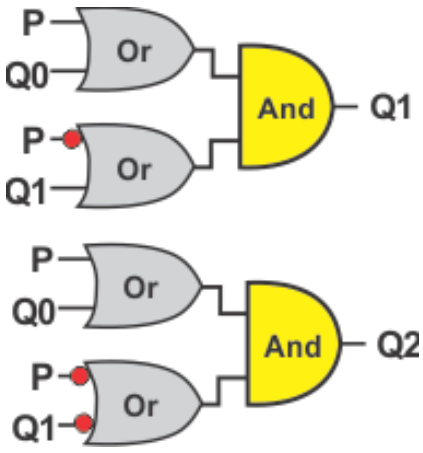
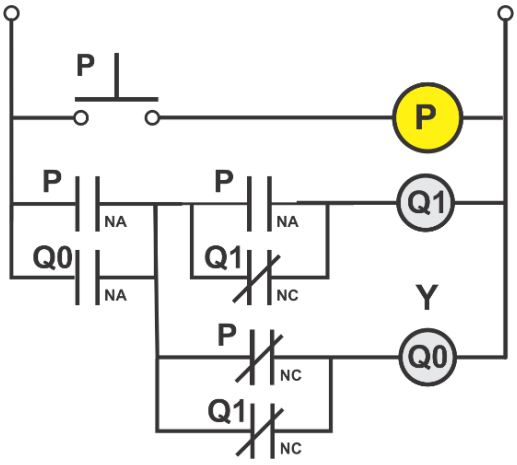


Fig. 30 Imágenes de la simulación en PROTEUS

En las figuras anteriores se muestra el funcionamiento del sistema, indicado los valores de entrada P y salida Y, así como los valores de las memorias Q1 y Q0 para cada uno de los cuatro tiempos de 0 a 3.

13.- Representación Gráfica de los resultados.

<table>
<tr><th colspan="2">Forma And/Or</th></tr>
<tr><td colspan="2">Q1= P Q1 + P' Q0, Q0= Q1'+ P' Q0 , Y = Q0</td></tr>
<tr><td> 
</td><td> 
</td></tr>
<tr><td>Diagrama esquemático And/Or</td><td>Diagrama escalera And/Or</td></tr>
<tr><th colspan="2">Forma Or/And</th></tr>
<tr><td colspan="2">Q1= (P + Q0) (P'+ Q1), Q0=(P + Q0) (P'+ Q1'), Y = Q0</td></tr>
<tr><td> 
</td><td> 
</td></tr>
<tr><td>Diagrama esquemático Or/And</td><td>Diagrama escalera Or/And</td></tr>
</table>

Asignación 2:

Consideremos la asignación 2 de valores a los estados

Asignación 2

Q1 \ Q2	0	1
0	**d**	**c**
1	**a**	**b**

		Q1	Q0
a	=	**0**	**1**
b	=	**1**	**1**
c	=	**1**	**0**
d	=	**0**	**0**

Tabla de estados Totales

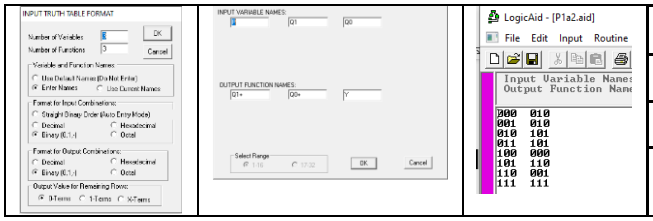
	P		Y
	0	**1**	
01	**01**	**11**	**0**
11	**10**	**11**	**1**
10	**10**	**00**	**1**
00	**01**	**00**	**0**

Tabla de verdad

m	P	Q1	Q0	Q1+	Q0+	Y
0	0	0	0	**0**	**1**	**0**
1	0	0	1	**0**	**1**	**0**
2	0	1	0	**1**	**0**	**1**
3	0	1	1	**1**	**0**	**1**
4	1	0	0	**0**	**0**	**0**
5	1	0	1	**1**	**1**	**0**
6	1	1	0	**0**	**0**	**1**
7	1	1	1	**1**	**1**	**1**

Ecuaciones mínimas:

Obtención de las ecuaciones mínimas con LogicAid.



And/Or	Or/And
Q1+=**P'**Q1+ PQ0	Q1+=(P+Q1)(**P'**+Q0)
Q0+=**P'Q1'** +PQ0	Q0+=(P+**Q1'**)(**P'**+Q0)
Y= Q1	

Fig. 31 proceso de minimización en LogiAid

Representación del comportamiento del sistema secuencial en el código ABEL-HDL, por medio de las ecuaciones con el comando **equations** y también por tabla de verdad utilizando el comando **Truth_table**,

Código ABEL-HDL con ecuaciones		Código ABEL-HDL con Tabla de verdad
MODULE Ados "Asignación 2 JAGG		MODULE Ados " Asignación JAGG
"entrada P pin 1; "salidas Y, Q1, Q0 pin 19, 14, 15 istype 'com';		"entrada P pin 1; "salidas Y, Q1, Q0 pin 19, 14, 15 istype 'com';
Q1=!P&Q1# P&Q0; Q1=!P&!Q1# P&Q0; Y = Q1;		Truth_table ([P,Q1,Q0]->[Q1,Q0,Y]) [0,0,0]->[0,1,0]; [0,0,1]->[0,1,0]; [0,1,0]->[1,0,1]; [0,1,1]->[1,0,1]; [1,0,0]->[0,0,0]; [1,0,1]->[1,1,0]; [1,1,0]->[0,0,1]; [1,1,1]->[1,1,1];
test_vectors **(P->Y)** **0->.x.;** **1->.x.;** **0->.x.;** **1->.x.;** **0->.x.;** END		**test_vectors** **(P->Y)** **0->.x.;** **1->.x.;** **0->.x.;** **1->.x.;** **0->.x.;** END

Simulaciones.

a) Test_Vectors

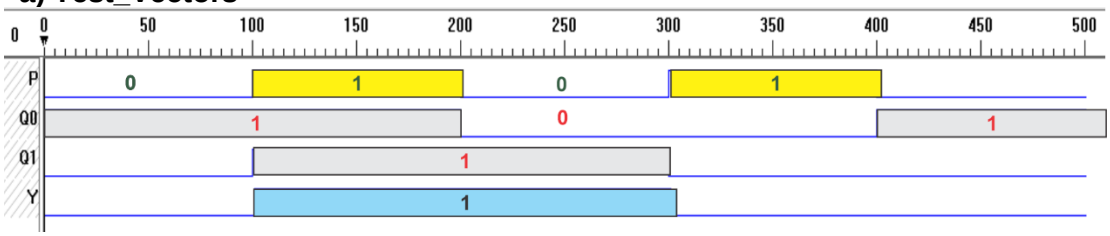
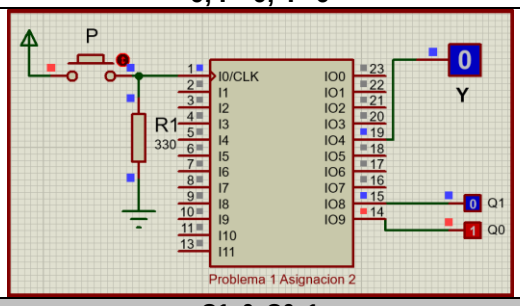
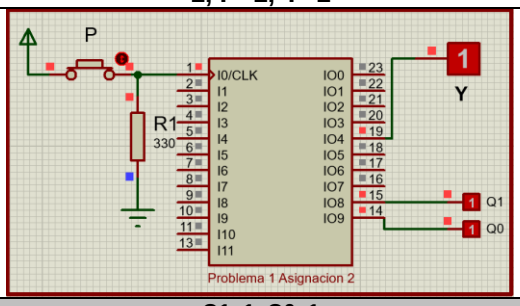
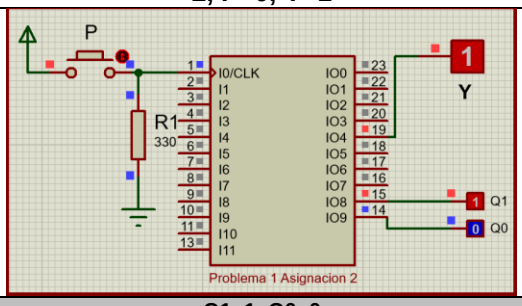
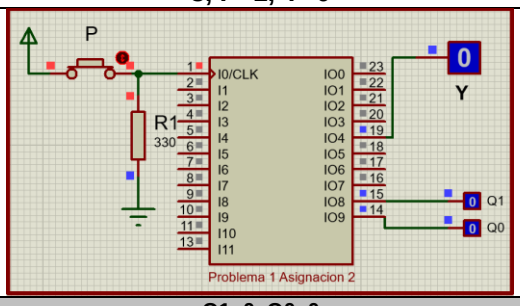


Fig. 32 Imagen de la simulación en TEST_VECTORS

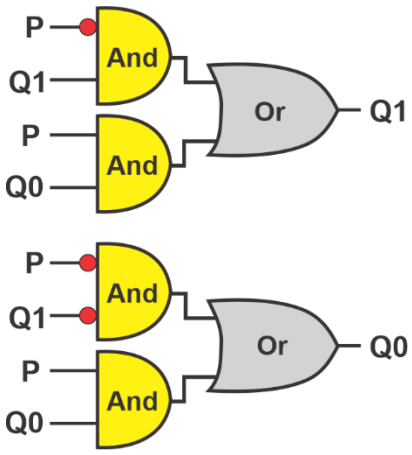
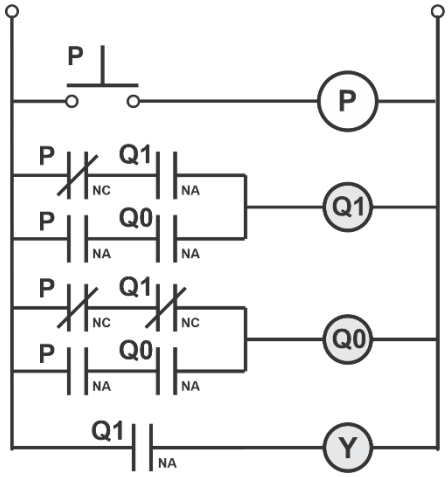
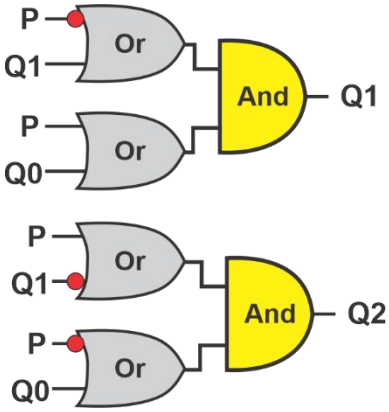
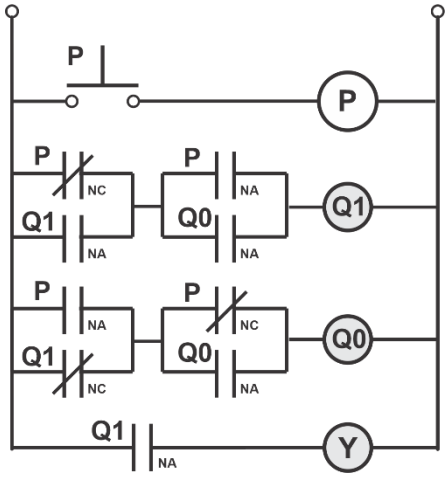
En la imagen anterior podemos observar que el valor de salida Y cumple con lo establecido en la propuesta inicial.

b) PROTEUS

0, P=0, Y=0	1, P=1, Y=1
	
Q1=0, Q0=1	**Q1=1, Q0=1**
2, P=0, Y=1	**3, P=1, Y=0**
	
Q1=1, Q0=0	**Q1=0, Q0=0**

En las figuras anteriores se muestra el funcionamiento del sistema, indicado los valores de entrada P y salida Y, así como los valores de las memorias Q1 y Q0 para cada uno de los cuatro tiempos de 0 a 3

Representación Gráfica de los resultados:

Diagrama esquemático And/Or	Diagrama escalera And/Or
	
Q1=**P'**Q1+ PQ0, Q0=**P'Q1'** +PQ0, Y= Q1	
Diagrama esquemático Or/And	**Diagrama escalera Or/And**
	
Q1=(**P'**+Q1) (P+Q0), Q0=(P+**Q1'**) (**P'**+Q0), Y= Q1	

Asignación 3:

Consideremos la asignación 3 de valores a los estados

Q1 \ Q2	0	1
0	c	b
1	d	a

	Q1	Q0
a	1	1
b	1	0
c	0	0
d	0	1

Tabla de estados Totales:

	P = 0	P = 1	Y
11	11	10	0
10	00	10	1
00	00	01	1
01	11	01	0

Tabla de verdad:

m	P	Q1	Q0	Q1+	Q0+	Y
0	0	0	0	**0**	**0**	**1**
1	0	0	1	**1**	**1**	**0**
2	0	1	0	**0**	**0**	**1**
3	0	1	1	**1**	**1**	**0**
4	1	0	0	**0**	**1**	**1**
5	1	0	1	**0**	**1**	**0**
6	1	1	0	**1**	**0**	**1**
7	1	1	1	**1**	**0**	**0**

Código ABEL-HDL:

```
MODULE Atres
"Caso 3 JAGG
"entrada
P pin 1;
"salida
Y, Q1, Q0 pin 19, 14, 15 istype 'com';
Truth_table
([P,Q1,Q0]->[Q1,Q0,Y])
[0,1,1]->[1,1,0];
[1,0,0]->[0,1,1];
[0,0,0]->[0,0,1];
[0,0,1]->[1,1,0];
[0,1,0]->[0,0,1];
[1,0,1]->[0,1,0];
[1,1,0]->[1,0,1];
[1,1,1]->[1,0,0];
test_vectors
([P,Q1,Q0]->[Y])
[0,1,1]->[.x.];
[1,1,0]->[.x.];
[0,1,0]->[.x.];
[1,0,0]->[.x.];
[0,0,0]->[.x.];
[1,0,1]->[.x.];
[0,0,1]->[.x.];
[1,1,1]->[.x.];
  END
```

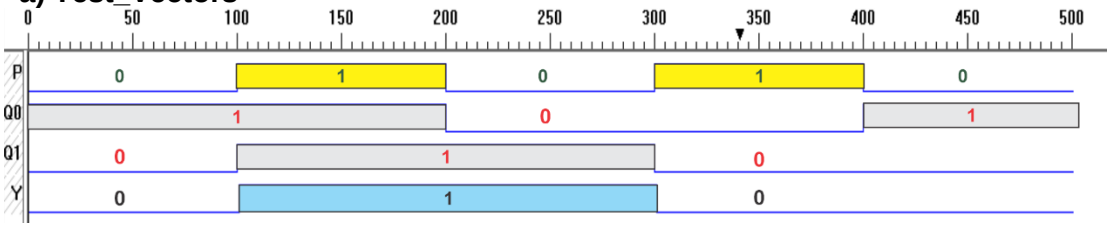
Ecuaciones mínimas:

Q1=(Q1& P#Q0&**!P**);

Q0=(**!Q1**&P#Q0&**!P**); Y= **!Q0**

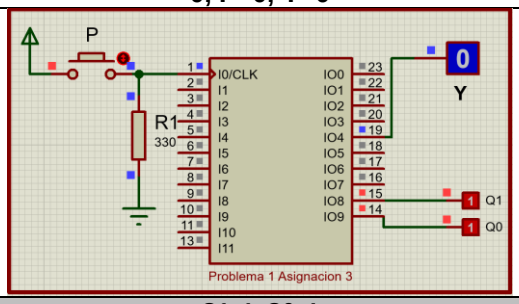
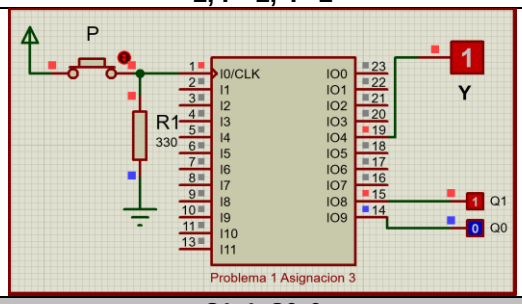
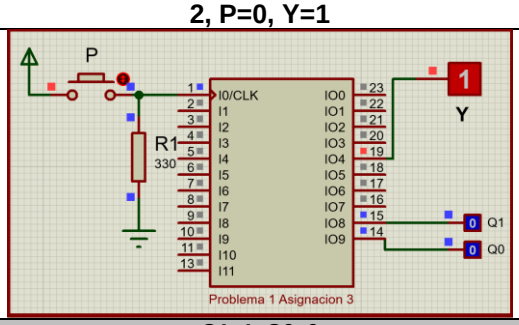
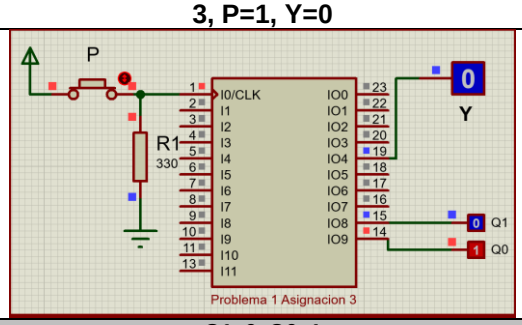
Simulaciones.

a) Test_Vectors



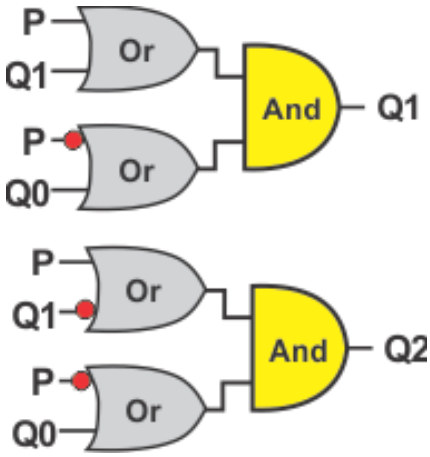
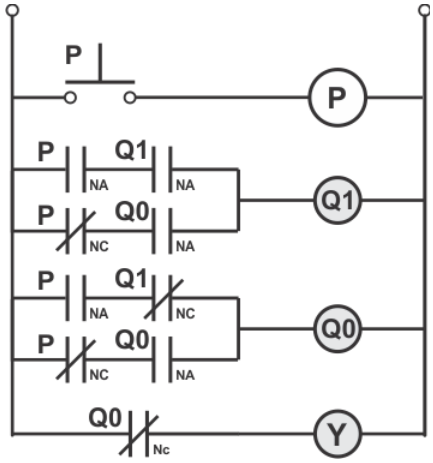
En la imagen podemos observar que cumple con lo establecido en la propuesta inicial

b) PROTEUS

0, P=0, Y=0	**1, P=1, Y=1**
	
Q1=1, Q0=1	**Q1=1, Q0=0**
2, P=0, Y=1	**3, P=1, Y=0**
	
Q1=1, Q0=0	**Q1=0, Q0=1**

En las figuras anteriores se muestra el funcionamiento del sistema, indicado los valores de entrada P y salida Y, así como los valores de las memorias Q1 y Q0 para cada uno de los cuatro tiempos de 0 a

Representación Gráfica de los resultados:

Diagrama esquemático Or/And	Diagrama escalera Or/And
	

Asignación 4:

Consideremos la asignación 4 de valores a los estados:

Q1 \ Q2	0	1
0	b	a
1	c	d

	Q1	Q0
a	1	0
b	0	0
c	0	1
d	1	1

Tabla de estados Totales:

	P		
	0	1	Y
10	10	00	0
00	01	00	1
01	01	11	1
11	10	11	0

Tabla de verdad:

M	P	Q1	Q0	Q1+	Q0+	Y
0	0	0	0	**0**	**1**	**1**
1	0	0	1	**0**	**1**	**1**
2	0	1	0	**1**	**0**	**0**
3	0	1	1	**1**	**0**	**0**
4	1	0	0	**0**	**0**	**1**
5	1	0	1	**1**	**1**	**1**
6	1	1	0	**0**	**0**	**0**
7	1	1	1	**1**	**1**	**0**

Código ABEL-HDL:

```
MODULE Acuatro
"Caso 4 JAGG
"entrada
P pin 1;
"salida
Y, Q1, Q0 pin 19, 14, 15 istype 'com';
Truth_table
([P,Q1,Q0]->[Q1,Q0,Y])
[0,0,0]->[0,1,1];
[0,0,1]->[0,1,1];
[0,1,0]->[1,0,0];
[0,1,1]->[1,0,0];
[1,0,0]->[0,0,1];
[1,0,1]->[1,1,1];
[1,1,0]->[0,0,0];
[1,1,1]->[1,1,0];
test_vectors
([P,Q1,Q0]->[Y])
[0,1,0]->[.x.];
[1,.x.,.x.]->[.x.];
[0,.x.,.x.]->[.x.];
[1,.x.,.x.]->[.x.];
[0,.x.,.x.]->[.x.];
END
```

Nota: En el Test_vectors aseguramos que el estado inicial sea E0 con valores de Q1=1 y Q0=0, tal como se seleccionó en la asignación

Ecuaciones mínimas:

Q1=(Q0 & P# Q1 & **!P**);

Q0=(Q0 & P#**!Q1**& **!P**); Y= **!Q1**

simulaciones:

a) Test_ Vectors

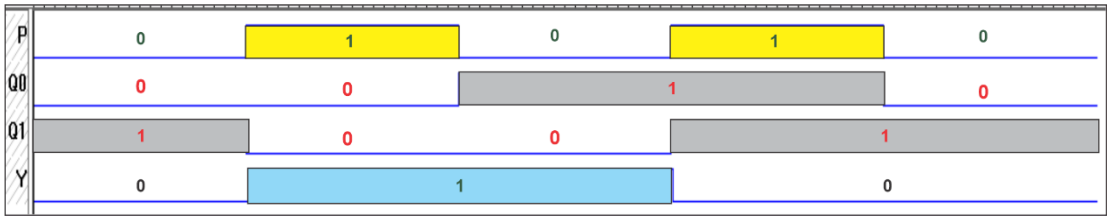
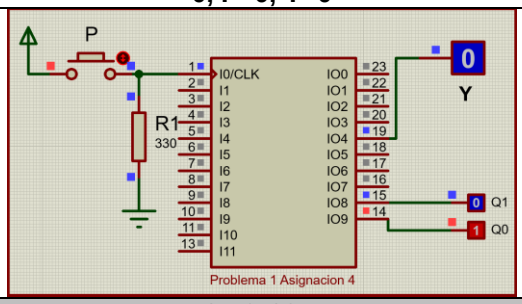
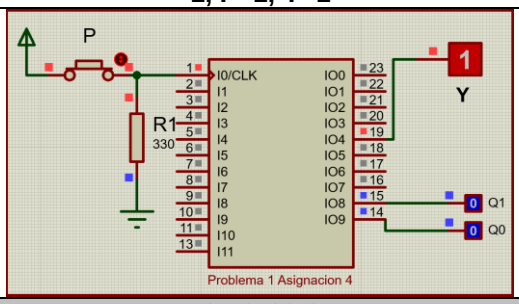
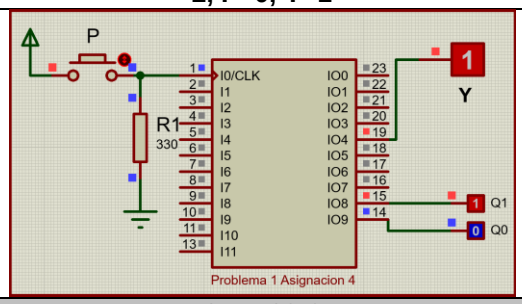
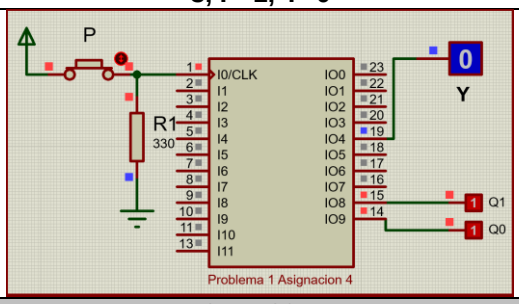
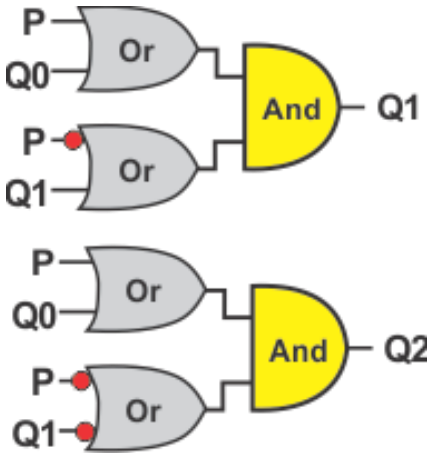
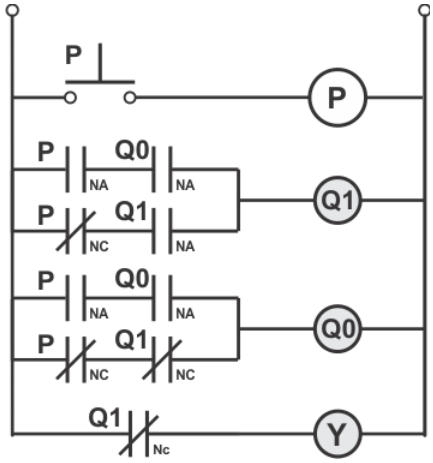


Fig. 41 Imagen de simulación con Test_Vectors

b) PROTEUS

0, P=0, Y=0	1, P=1, Y=1
	
Q1=0, Q0=1	**Q1=0, Q0=0**
2, P=0, Y=1	**3, P=1, Y=0**
	
Q1=1, Q0=0	**Q1=1, Q0=1**

Representación Gráfica de los resultados:

Diagrama esquemático Or/And	Diagrama escalera Or/And
	

Asignación 5:

Consideremos la asignación 5 de valores a los estados

Q1 \ Q2	0	1
0	a	b
1	d	c

	Q1	Q0
a	0	0
b	1	0
c	1	1
d	0	1

Tabla de estados Totales:

	P = 0	P = 1	Y
00	00	10	0
10	11	10	1
11	11	01	1
01	00	01	0

Tabla de verdad:

m	P	Q1	Q0	Q1+	Q0+	Y
0	0	0	0	**0**	**0**	**0**
1	0	0	1	**0**	**0**	**0**
2	0	1	0	**1**	**1**	**1**
3	0	1	1	**1**	**1**	**1**
4	1	0	0	**1**	**0**	**0**
5	1	0	1	**0**	**1**	**0**
6	1	1	0	**1**	**0**	**1**
7	1	1	1	**0**	**1**	**1**

Código ABEL-HDL:

```
MODULE Acinco
"Caso 5 JAGG
"entrada
P pin 1;
"salida
Y, Q1, Q0 pin 19, 14, 15 istype 'com';
Truth_table
([P,Q1,Q0]->[Q1,Q0,Y])
[0,0,0]->[0,0,0];
[0,0,1]->[0,0,0];
[0,1,0]->[1,1,1];
[0,1,1]->[1,1,1];
[1,0,0]->[1,0,0];
[1,0,1]->[0,1,0];
[1,1,0]->[1,0,1];
[1,1,1]->[0,1,1];
test_vectors
(P->Y)
0->.x.;
1->.x.;
0->.x.;
1->.x.;
0->.x.;
1->.x.;
0->.x.;
1->.x.;
END
```

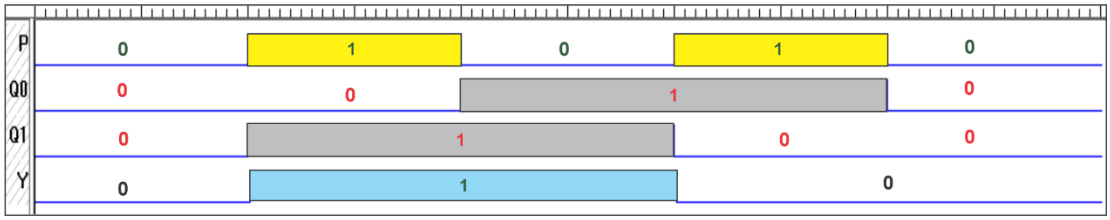
Ecuaciones Mínimas:

Q1=(**!Q0** & P # Q1 & **!P**);

Q0= (Q0 & P#Q1 &**!P**); Y=Q1;

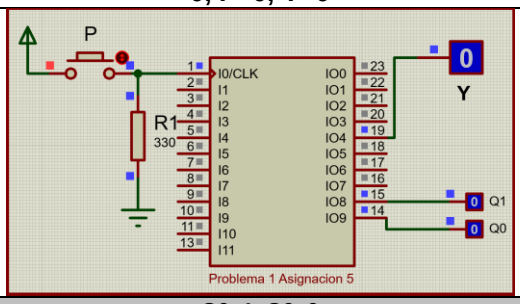
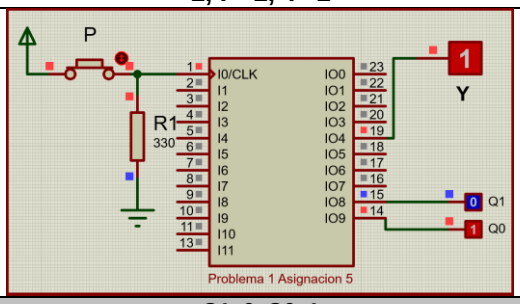
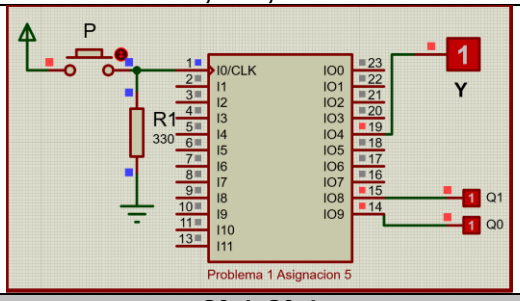
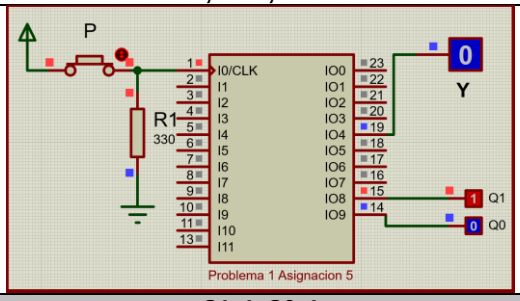
Simulaciones:

a) Test_Vectors

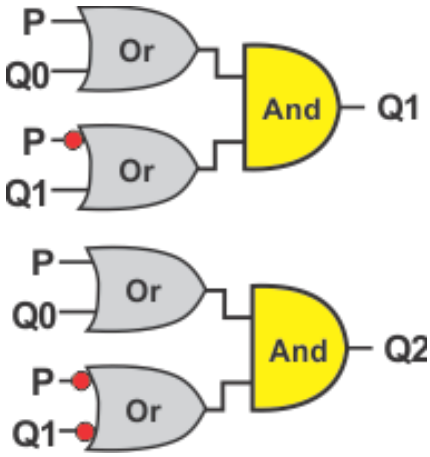
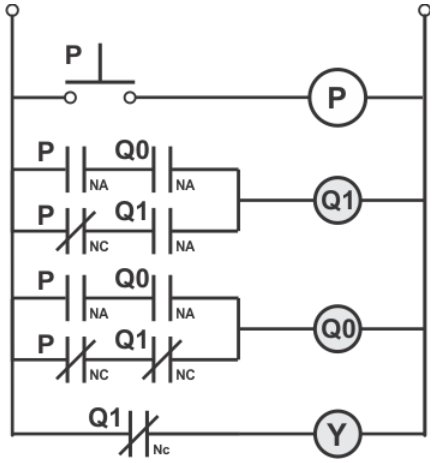


En la imagen podemos observar que el valor de salida Y cumple con lo establecido en la propuesta inicial

b) PROTEUS

0, P=0, Y=0	1, P=1, Y=1
	
Q0=1, Q0=0	**Q1=0, Q0=1**
2, P=0, Y=1	**3, P=1, Y=0**
	
Q0=1, Q0=1	**Q1=1, Q0=1**

Representación Gráfica de los resultados:

Diagrama esquemático Or/And	Diagrama escalera Or/And
	

Asignación 6:

Consideremos la asignación 6 de valores a los estados:

Q1 \ Q2	0	1
0	**d**	**a**
1	**c**	**b**

	Q1	Q0
a	**1**	**0**
b	**1**	**1**
c	**0**	**1**
d	**0**	**0**

Tabla de estados Totales:

	P = 0	P = 1	Y
10	**10**	**11**	**0**
11	**01**	**11**	**1**
01	**01**	**00**	**1**
00	**10**	**00**	**0**

Tabla de verdad:

m	P	Q1	Q0	Q1+	Q0+	Y
0	0	0	0	**1**	**0**	**0**
1	0	0	1	**0**	**1**	**1**
2	0	1	0	**1**	**0**	**0**
3	0	1	1	**0**	**1**	**1**
4	1	0	0	**0**	**0**	**0**
5	1	0	1	**0**	**0**	**1**
6	1	1	0	**1**	**1**	**0**
7	1	1	1	**1**	**1**	**1**

Código ABEL-HDL:

```
MODULE Aseis
"Caso 6 JAGG
"entrada
P pin 1;
"salida
Y, Q1, Q0 pin 19, 14, 15 istype 'com';
Truth_table
([P,Q1,Q0]->[Q1,Q0,Y])
[0,0,0]->[1,0,0];
[0,0,1]->[0,1,1];
[0,1,0]->[1,0,0];
[0,1,1]->[0,1,1];
[1,0,0]->[0,0,0];
[1,0,1]->[0,0,1];
[1,1,0]->[1,1,0];
[1,1,1]->[1,1,1];
test_vectors
([P,Q1,Q0]->[Y])
[0,1,0]->[.x.];
[1,.x.,.x.]->[.x.];
[0,.x.,.x.]->[.x.];
[1,.x.,.x.]->[.x.];
[0,.x.,.x.]->[.x.];
END
```

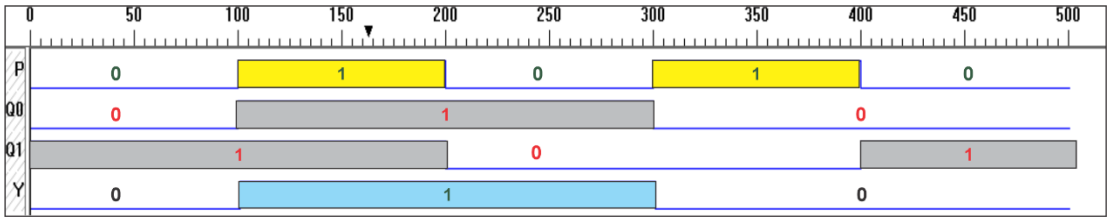
Nota: En el Test_vectors aseguramos que el estado inicial sea E0 con valores de Q1=1 y Q0=0, tal como se seleccionó en la asignación

Ecuaciones mínimas:

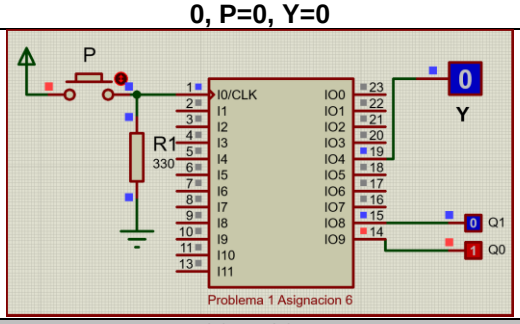
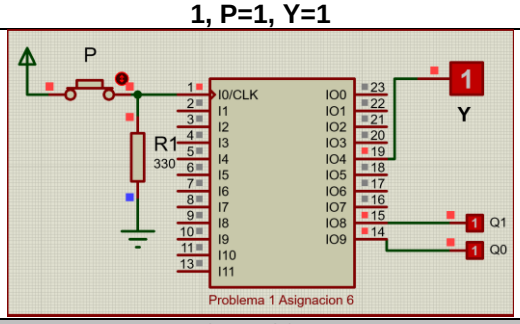
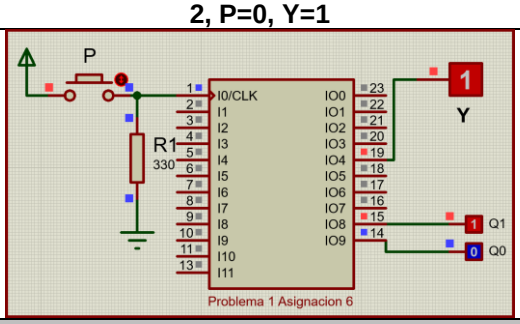
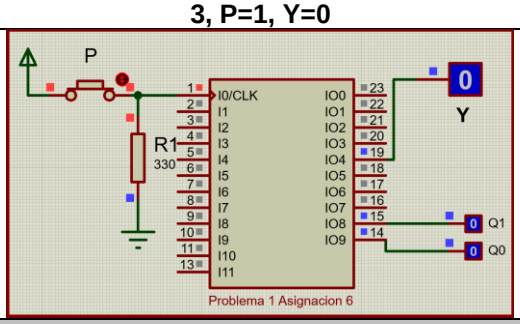
Q1= (Q1 & P #**!Q0** & **!P**);

Q0 = (Q1 & P #Q0 & **!P**); Y = Q0

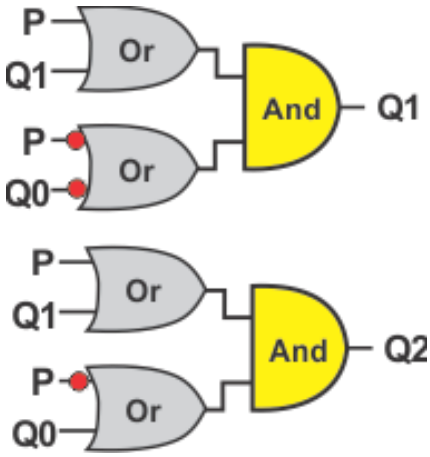
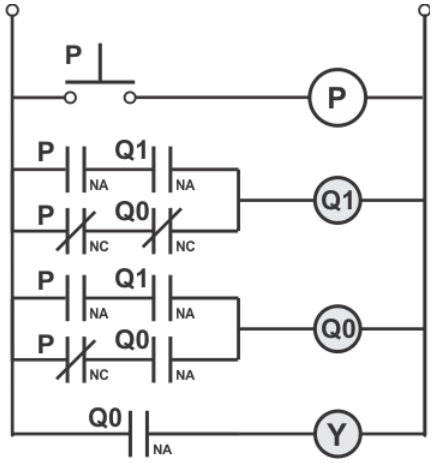
Imagen de la simulación:



b) PROTEUS

0, P=0, Y=0	1, P=1, Y=1
	
Q0=1, Q0=1	**Q1=1, Q0=1**
2, P=0, Y=1	**3, P=1, Y=0**
	
Q0=1, Q0=1	**Q1=0, Q0=0**

Representación Gráfica de los resultados:

Diagrama esquemático Or/And	Diagrama escalera Or/And
	

Asignación 7:

Consideremos la asignación 7 de valores a los estados

Q1 \ Q2	0	1
0	c	d
1	b	a

	Q1	Q0
a	1	1
b	0	1
c	0	0
d	1	0

Tabla de estados:

	P = 0	P = 1	Y
11	11	01	0
01	00	01	1
00	00	10	1
10	11	10	0

Tabla de verdad:

m	P	Q1	Q0	Q1+	Q0+	Y
0	0	0	0	**0**	**0**	**1**
1	0	0	1	**0**	**0**	**1**
2	0	1	0	**1**	**1**	**0**
3	0	1	1	**1**	**1**	**0**
4	1	0	0	**1**	**0**	**1**
5	1	0	1	**0**	**1**	**1**
6	1	1	0	**1**	**0**	**0**
7	1	1	1	**0**	**1**	**0**

Código ABEL-HDL:

```
MODULE Asiete
"Caso 7 JAGG
"entrada
P pin 1;
"salida
Y, Q1, Q0 pin 19, 14, 15 istype 'com';
Truth_table
([P,Q1,Q0]->[Q1,Q0,Y])
[0,0,0]->[0,0,1];
[0,0,1]->[0,0,1];
[0,1,0]->[1,1,0];
[0,1,1]->[1,1,0];
[1,0,0]->[1,0,1];
[1,0,1]->[0,1,1];
[1,1,0]->[1,0,0];
[1,1,1]->[0,1,0];
test_vectors
(P->Y)
0->.x.;
1->.x.;
0->.x.;
1->.x.;
0->.x.;
1->.x.;
0->.x.;
1->.x.;
0->.x.;
END
```

Ecuaciones mínimas

Q1=(**!Q0** & P #Q1 &**!P**);

Q0 = (Q0 & P #Q1& **!P**); Y= **!Q1**

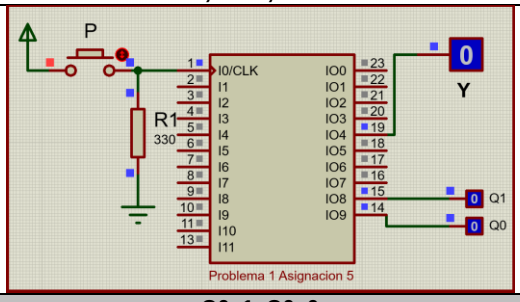
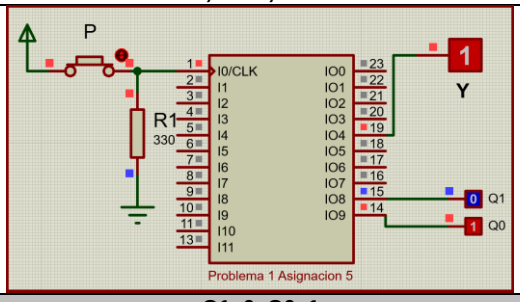
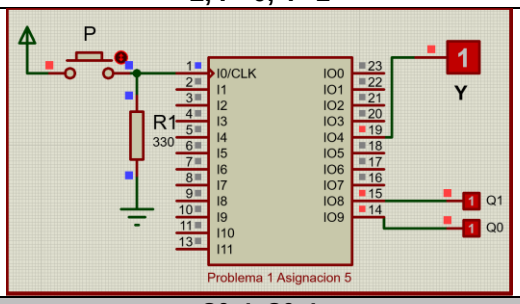
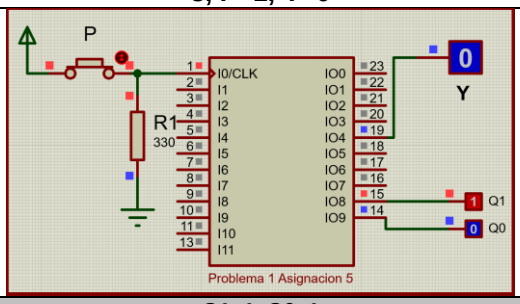
Simulaciones:

Test_vectors

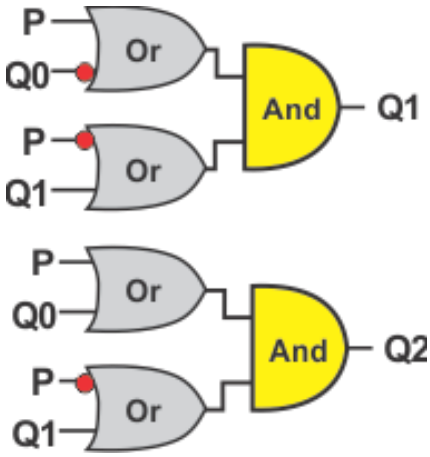
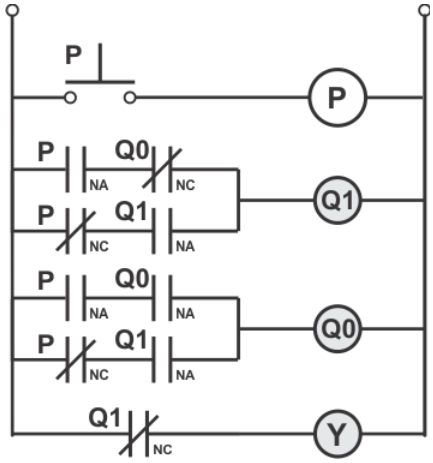
Fig. 37 Imagen de simulación con Test_Vectors

En la imagen hay que considerar que las condiciones iniciales se toman con Q1=1 y Q0=0 (t=200) y a que partir de ahí podemos observar que cumple con lo establecido en la propuesta inicial

b) PROTEUS

0, P=0, Y=0	1, P=1, Y=1
	
Q0=1, Q0=0	**Q1=0, Q0=1**
2, P=0, Y=1	**3, P=1, Y=0**
	
Q0=1, Q0=1	**Q1=1, Q0=1**

Representación Gráfica de los resultados:

Diagrama esquemático And/Or	Diagrama escalera And/Or
	

Asignación 8

Consideremos la asignación 8 de valores a los estados

Q1 \ Q2	0	1
0	**b**	**c**
1	**a**	**d**

	Q1	Q0
a	**0**	**1**
b	**0**	**0**
c	**1**	**0**
d	**1**	**1**

Tabla de estados:

	P		Y
	0	**1**	
01	**01**	**00**	**0**
00	**10**	**00**	**1**
10	**10**	**11**	**1**
11	**01**	**11**	**0**

Tabla de verdad:

m	P	Q1	Q0	Q1+	Q0+	Y
0	0	0	0	**1**	**0**	**1**
1	0	0	1	**0**	**1**	**0**
2	0	1	0	**1**	**0**	**1**
3	0	1	1	**0**	**1**	**0**
4	1	0	0	**0**	**0**	**1**
5	1	0	1	**0**	**0**	**0**
6	1	1	0	**1**	**1**	**1**
7	1	1	1	**1**	**1**	**0**

Código ABEL-HDL:

```
MODULE Aocho
"Caso 8 JAGG
"entrada
P pin 1;
"salida
Y, Q1, Q0 pin 19, 14, 15 istype 'com';
Truth_table
([P,Q1,Q0]->[Q1,Q0,Y])
[0,0,0]->[1,0,1];
[0,0,1]->[0,1,0];
[0,1,0]->[1,0,1];
[0,1,1]->[0,1,0];
[1,0,0]->[0,0,1];
[1,0,1]->[0,0,0];
[1,1,0]->[1,1,1];
[1,1,1]->[1,1,0];
test_vectors
(P->Y)
0->.x.;
1->.x.;
0->.x.;
1->.x.;
0->.x.;
1->.x.;
0->.x.;
1->.x.;
0->.x.;
END
```

Ecuaciones mínimas:

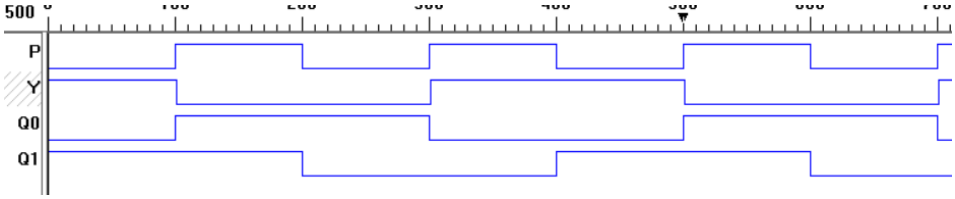
$$Q1=(Q1\&P\#\mathbf{!Q0}\&\mathbf{!P});$$

$$Q0= (Q1 \& P \#Q0 \&\mathbf{!P});$$

$$Y= \mathbf{!Q0};$$

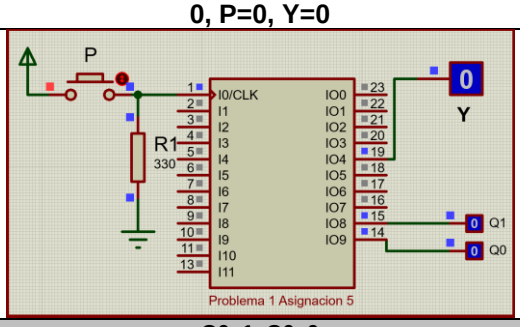
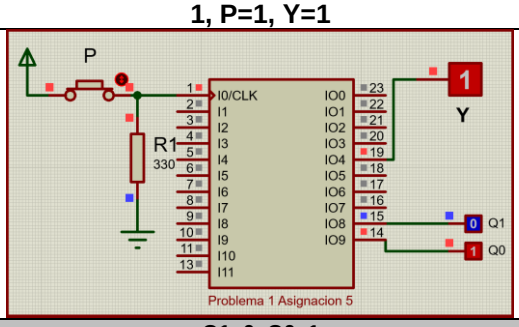
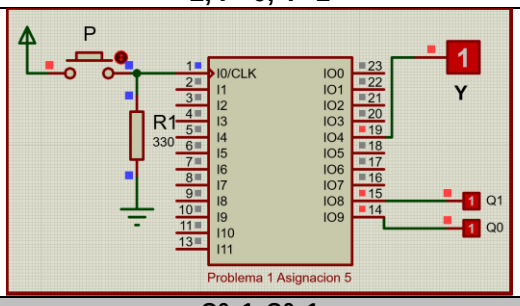
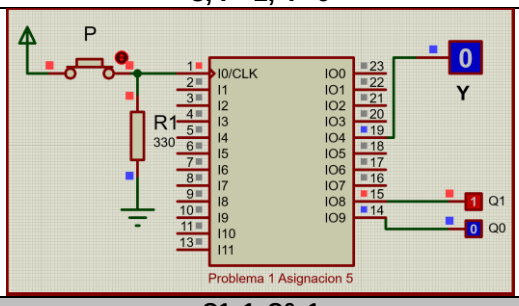
Simulaciones:

a) Test_Vectors

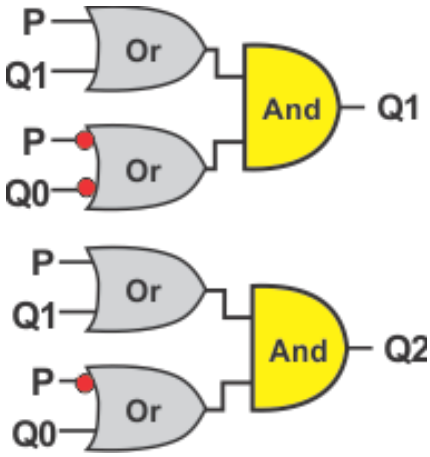
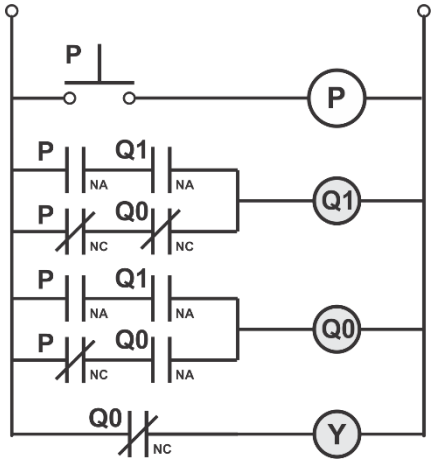


En la imagen podemos observar que cumple con lo establecido en la propuesta inicial.

b) PROTEUS

0, P=0, Y=0	1, P=1, Y=1
	
Q0=1, Q0=0	**Q1=0, Q0=1**
2, P=0, Y=1	**3, P=1, Y=0**
	
Q0=1, Q0=1	**Q1=1, Q0=1**

Representación Gráfica de los resultados:

Diagrama esquemático And/Or	Diagrama escalera And/Or
	

Problema 1.2, Botón caso 2

Diseñe un sistema secuencial asíncrono que contenga un botón de entrada **P** y una salida **Y**, partiendo de condiciones iniciales P=0, Y=0 (T0), al oprimir por primera vez P la salida Y deberá de cambiar de un valor inicial cero a uno, P=1, Y=1 (t1), al soltar el botón (t2) la salida deberá de mantenerse en uno, P=0, Y=1 (t2), al oprimirlo por segunda vez la señal de salida se mantiene P=1, Y=1 (t3), y al soltarlo la salida deberá regresar a condiciones iniciales P=0, Y=0 (t0), como lo indica la siguiente gráfica:

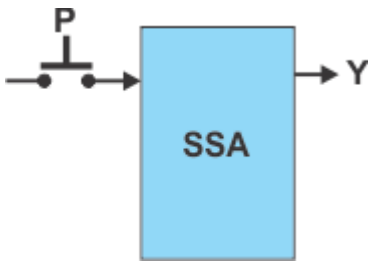


Diagrama de bloques

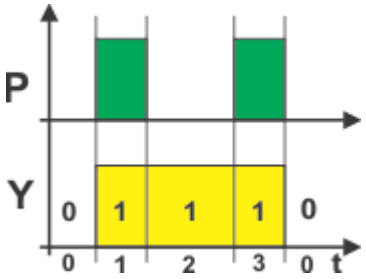


Diagrama de tiempos

1.- Especificar el sistema

Diagrama de transición

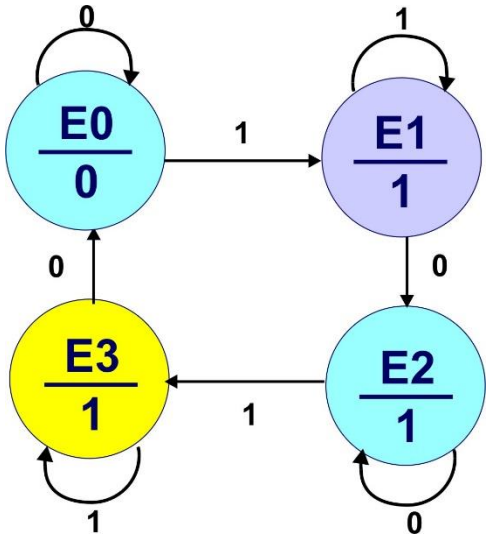
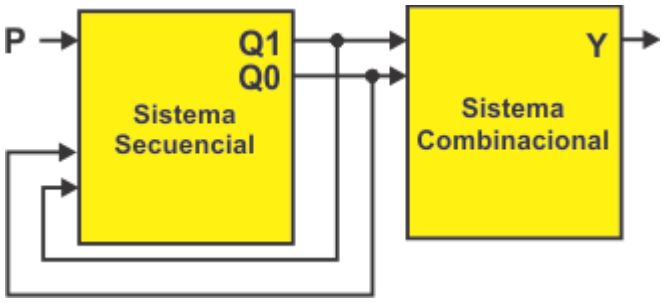


Diagrama de transición

2.- Tabla de flujo primitiva

	P		Y
	0	1	
a	**E0**	E1	0
b	E2	**E1**	1
c	**E2**	E3	1
d	E0	**E3**	1

Podemos observar en la tabla anterior, que la parte del sistema secuencial no cambia y el único cambio es la salida Y, de modo que tomando los valores anteriores de Q1 y Q0 obtenemos el valor de la parte combinacional de la salida Y:

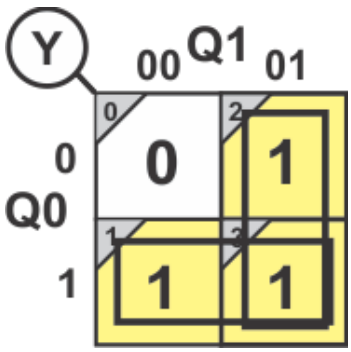


Por lo anterior no es aplicar los pasos del 4 al 9.

10.- Obtención de las ecuaciones por medio de minimización.

La salida Y está en función de Q1 y Q2 de modo que de la tabla anterior tenemos que:

m	P	Q1	Q0	Q1+	Q0+	Y
0	0	0	0	**0**	**0**	**0**
1	0	0	1	**1**	**1**	**1**
2	0	1	0	**0**	**0**	**1**
3	0	1	1	**1**	**1**	**1**
4	1	0	0	**0**	**1**	**0**
5	1	0	1	**0**	**1**	**1**
6	1	1	0	**1**	**0**	**1**
7	1	1	1	**1**	**0**	**1**

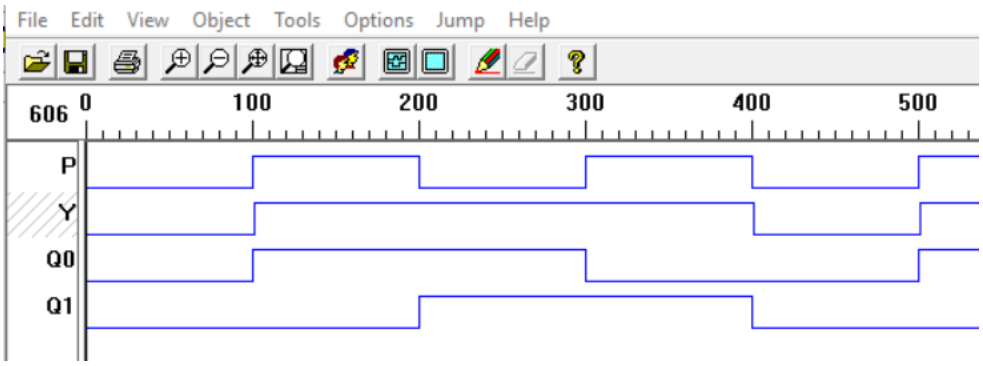


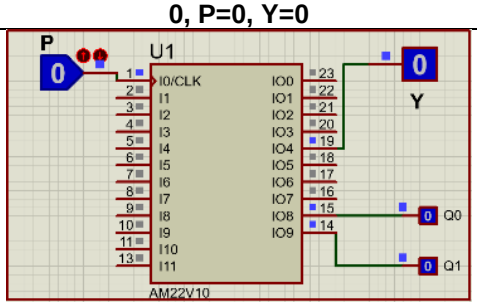
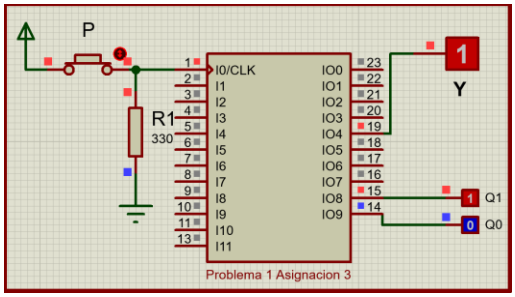
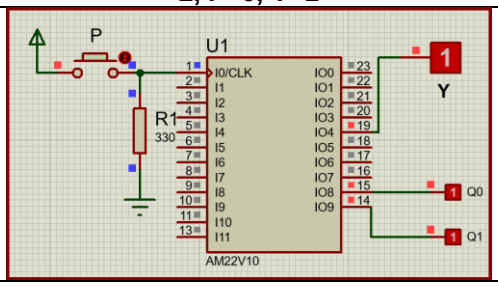
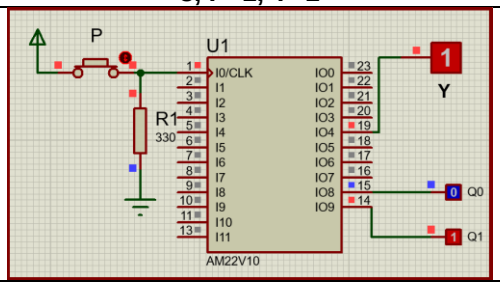
$Y = Q1 + Q0$

11.- Archivo en formato ABEL-HDL

```
MODULE botón
“Problema 2 JAGG
"entrada
P pin 1;
"salida
Y, Q1, Q0 pin 19..17 istype 'com';
equations
Q1=P&Q1#!P&Q0;
Q0=P&!Q1#!P&Q0;
Y=Q1#Q0;
test_vectors
(P->Y)
0->.x.;
1->.x.;
0->.x.;
1->.x.;
0->.x.;
1->.x.;
0->.x.;
1->.x.;
0->.x.;
END
```

12.- Simulaciones



0, P=0, Y=0	1, P=1, Y=1
	
2, P=0, Y=1	**3, P=1, Y=1**
	

En las figuras anteriores se muestra el funcionamiento del sistema, indicado los valores de salida para cada uno de los cuatro tiempos de 0 a 3.

En las figuras anteriores se muestra el funcionamiento del sistema, indicado los valores de salida para cada uno de los cuatro tiempos de 0 a 3.

13.- Representación.

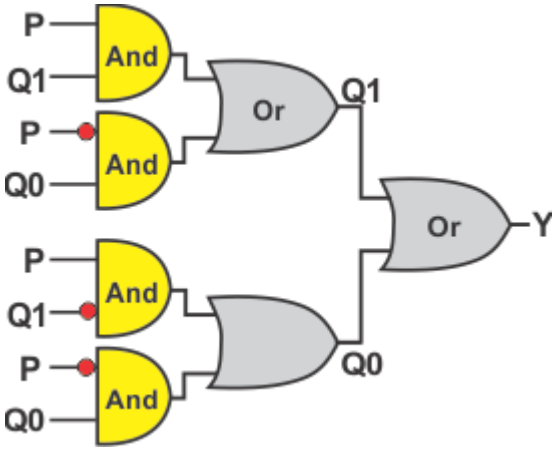


Diagrama esquemático

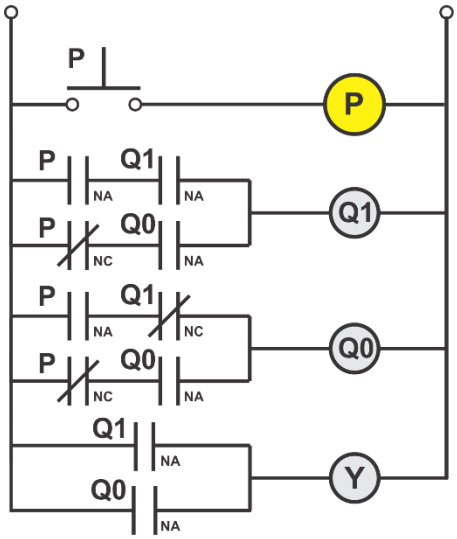


Diagrama escalera

Problema 1.3, Botón caso 3

Diseñe un sistema secuencial asíncrono que contenga un botón de entrada **P** y una salida **Y**, partiendo de condiciones iniciales P=0, Y=0 (T0), al oprimir por primera vez P la salida Y deberá de mantenerse en un valor de cero, P=1, Y=0 (t1), al soltar el botón (t2) la salida deberá de cambiar a uno, P=0, Y=1 (t2), al oprimirlo por segunda vez la señal de salida se mantiene P=1, Y=1 (t3), y al soltarlo la salida deberá regresar a condiciones iniciales P=0, Y=0 (t0), como lo indica la siguiente gráfica:

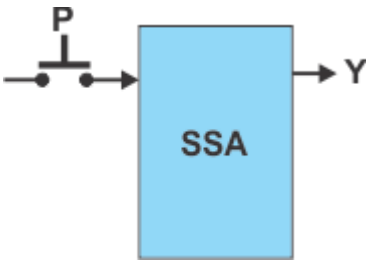


Diagrama de bloques

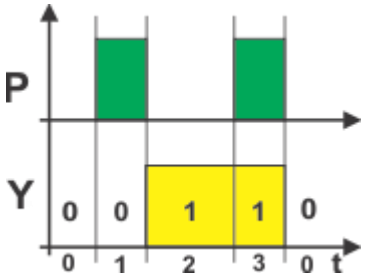


Diagrama de tiempos

1.-Especificar el sistema

Diagrama de transición

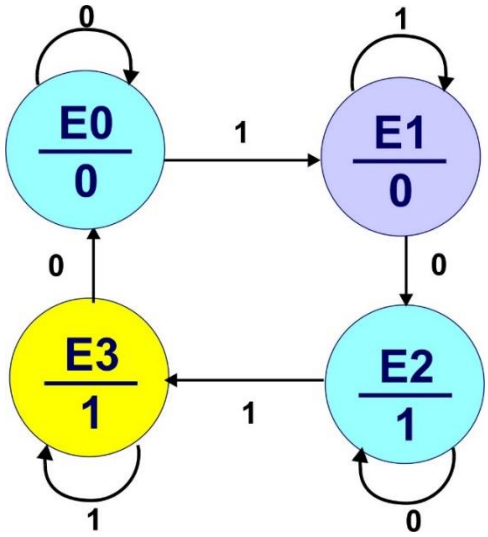
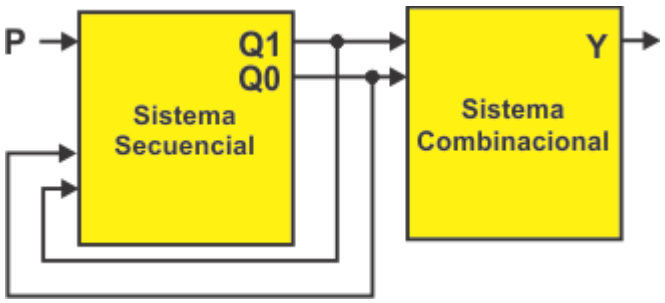


Diagrama de Transición

2.- Tabla de flujo primitiva

	P		Y
	0	1	
a	**E0**	E1	0
b	E2	**E1**	0
c	**E2**	E3	1
d	E0	**E3**	1

Podemos observar en la tabla anterior, que la parte del sistema secuencial no cambia y el único cambio es la salida Y, de modo que tomando los valores anteriores de Q1 y Q0 obtenemos el valor de la parte combinacional de la salida Y:

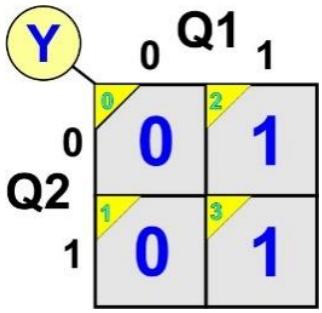


Por lo anterior no es aplicar los pasos del 3 al 9.

10.- Obtención de las ecuaciones mínimas.

La salida Y está en función de Q1 y Q2 de modo que de la tabla tenemos que:

m	P	Q1	Q2	Q1+	Q2+	Y
0	0	0	0	**0**	**0**	**0**
1	0	0	1	**1**	**1**	**0**
2	0	1	0	**0**	**0**	**1**
3	0	1	1	**1**	**1**	**1**
4	1	0	0	**0**	**1**	**0**
5	1	0	1	**0**	**1**	**0**
6	1	1	0	**1**	**0**	**1**
7	1	1	1	**1**	**0**	**1**

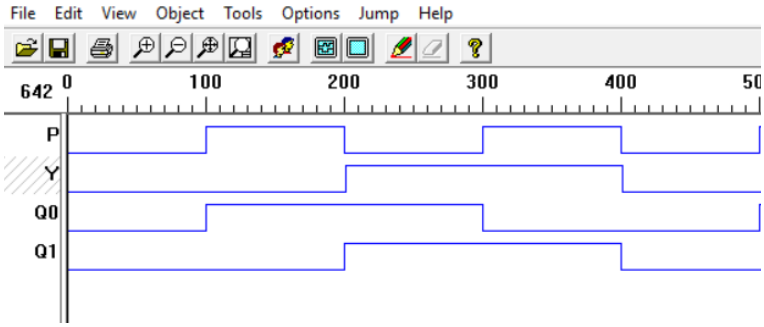


Y = Q1

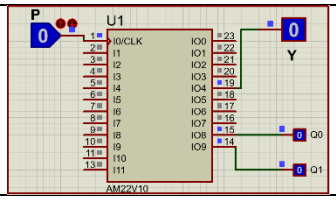
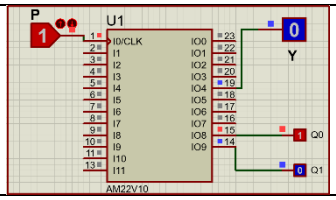
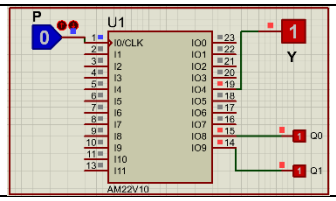
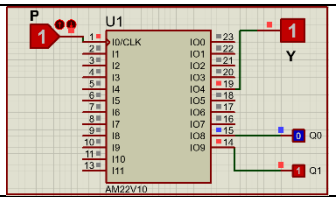
11.- Archivo en formato ABEL-HDL

```
MODULE botón
“Problema 3 JAGG
"entrada
P pin 1;
"salida
Y, Q1, Q0 pin 19, 14,15 istype 'com';
equations
Q1=P&Q1#!P&Q0;
Q0=P&!Q1#!P&Q0;
Y=Q1;
test_vectors
(P->Y)
0->.x.;
1->.x.;
0->.x.;
1->.x.;
0->.x.;
1->.x.;
0->.x.;
1->.x.;
0->.x.;
END
```

12.- Simulaciones en donde se muestra el valor de salida para cada uno de los cuatro tiempos de 0 a 3



Simulación en PROTEUS

0	1
	
2	3
	

3.- Representación Gráfica.

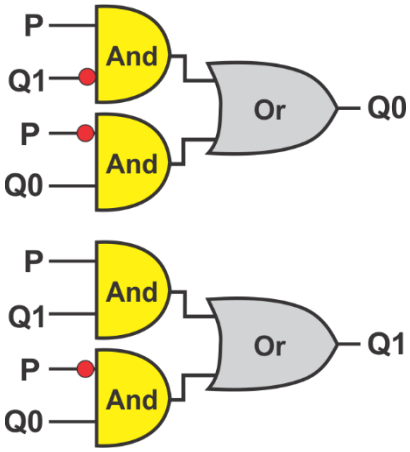


Diagrama esquemático

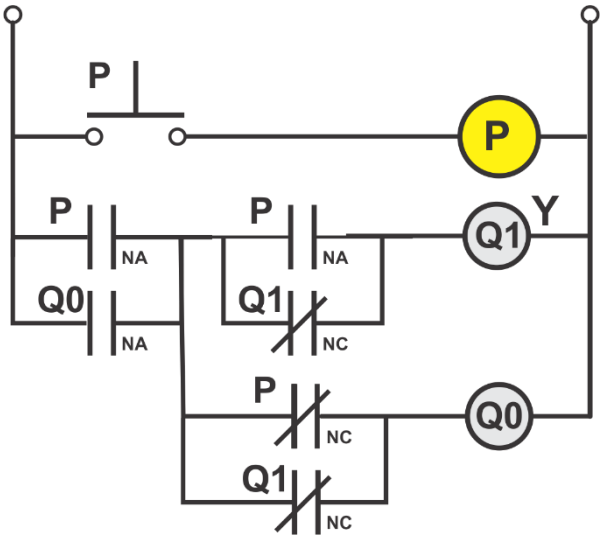


Diagrama escalera

Problema 1.4, Botón caso 4

Diseñe un sistema secuencial asíncrono que contenga un botón de entrada **P** y una salida **Y**, partiendo de condiciones iniciales P=0, Y=0 (T0), al oprimir por primera vez P la salida Y deberá de mantenerse en un valor de cero, P=1, Y=0 (t1), al soltar el botón (t2) la salida deberá de cambiar a uno, P=0, Y=1 (t2), al oprimirlo por segunda vez la señal de salida cambia P=1, Y=0 (t3), y al soltarlo la salida deberá regresar a condiciones iniciales P=0, Y=0 (t0), como lo indica la siguiente gráfica:

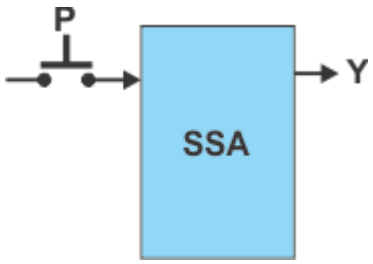


Diagrama de bloques

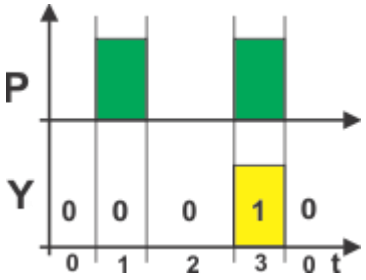


Diagrama de tiempos

1.- Especificar el sistema

Diagrama de transición

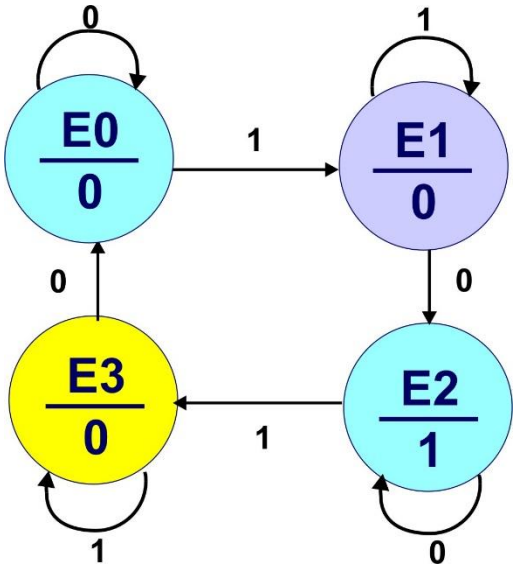
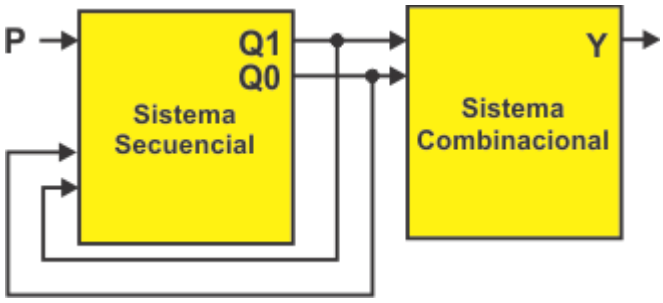


Diagrama de transición

2.- Tabla de flujo primitiva

	P		Y
	0	1	
a	E0	E1	0
b	E2	E1	0
c	E2	E3	0
d	E0	E3	1

Podemos observar en la tabla anterior que la parte del sistema secuencial no cambia y el único cambio es la salida Y de modo que no tomando los valores anteriores de Q1 y Q0 obtenemos el valor de la parte combinacional de la salida Y:

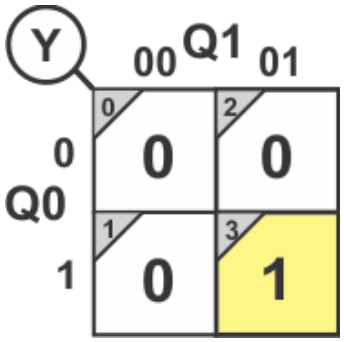


Por lo anterior no es aplicar los pasos del 3 al 9.

10.- Obtención de la ecuación mínima paya la salida Y.

La salida Y está en función de Q1 y Q2 de modo que de la tabla anterior tenemos que:

m	P	Q1	Q0	Q1+	Q0+	Y
0	0	0	0	**0**	**0**	**0**
1	0	0	1	**1**	**1**	**0**
2	0	1	0	**0**	**0**	**0**
3	0	1	1	**1**	**1**	**1**
4	1	0	0	**0**	**1**	**0**
5	1	0	1	**0**	**1**	**0**
6	1	1	0	**1**	**0**	**0**
7	1	1	1	**1**	**0**	**1**



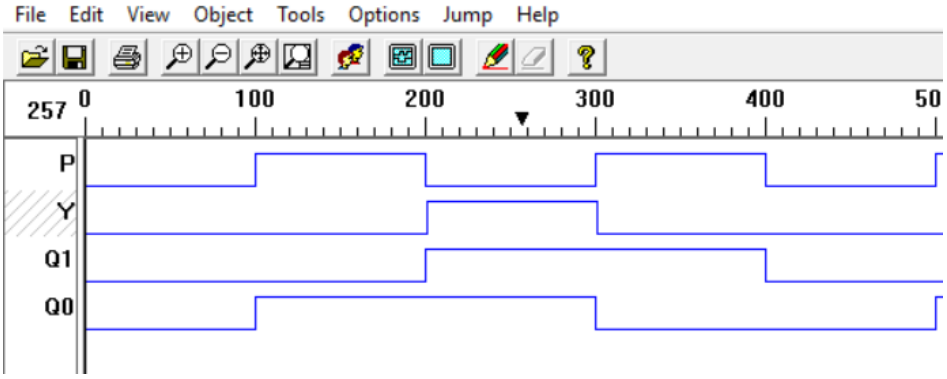
$$Y = Q1\ Q0$$

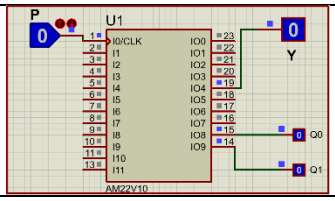
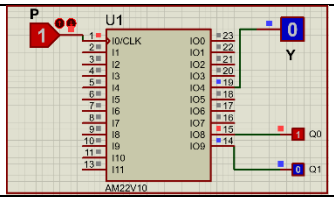
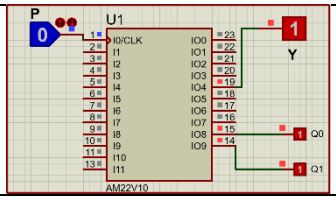
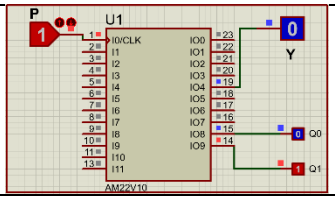
11.- Archivo en formato ABEL-HDL

```
MODULE boton
"Problema 4 JAGG
"entrada
P pin 1;
"salida
Y, Q1, Q0 pin 19..17 istype 'com';
equations
Q1=P&Q1#!P&Q0;
Q2=P&!Q1#!P&Q0;
Y=Q1&Q0;
```

```
test_vectors
(P->Y)
0->.x.;
1->.x.;
0->.x.;
1->.x.;
0->.x.;
1->.x.;
0->.x.;
1->.x.;
0->.x.;
END
```

12.- Simulaciones



0	1
	
2	**3**
	

13.- Representación gráfica

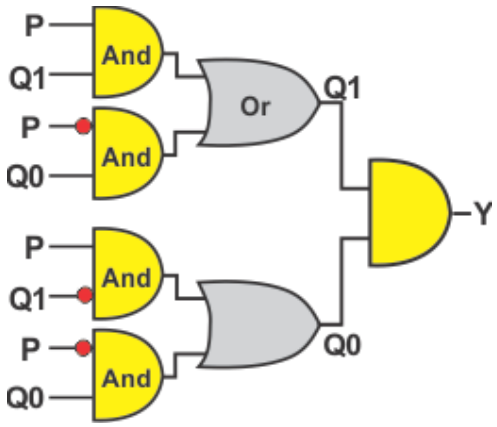


Diagrama esquemático

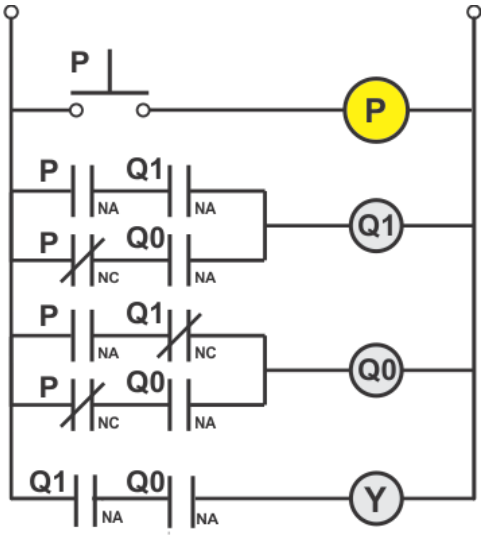


Diagrama escalera

Problema 1.5, Botón caso 5

Caso 5: Diseñe un sistema secuencial asíncrono que contenga un botón de entrada **P** y una salida **Y**, partiendo de condiciones iniciales P=0, Y=1 (T0), al oprimir por primera vez P la salida Y deberá de mantenerse en un valor de uno, P=1, Y=1 (t1), al soltar el botón (t2) la salida deberá de cambiar a cero, P=0, Y=0 (t2), al oprimirlo por segunda vez la señal de salida se mantiene P=1, Y=0 (t3), y al soltarlo la salida deberá regresar a condiciones iniciales P=0, Y=1 (t0), como lo indica la siguiente gráfica:

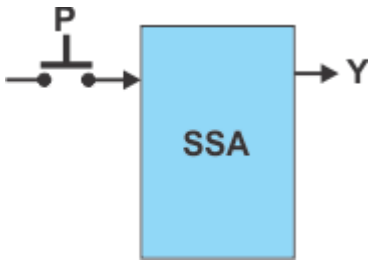


Diagrama de bloques

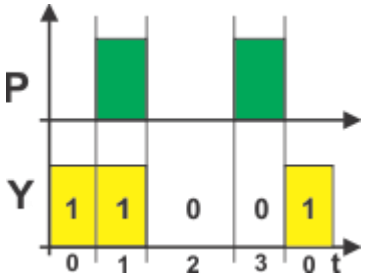
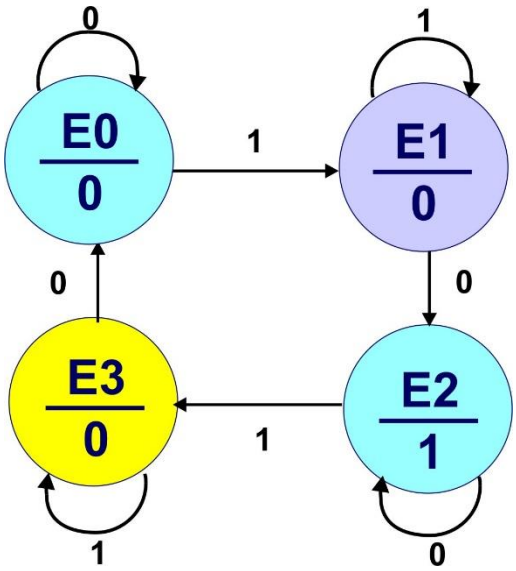


Diagrama de tiempos

1.- Especificar el sistema

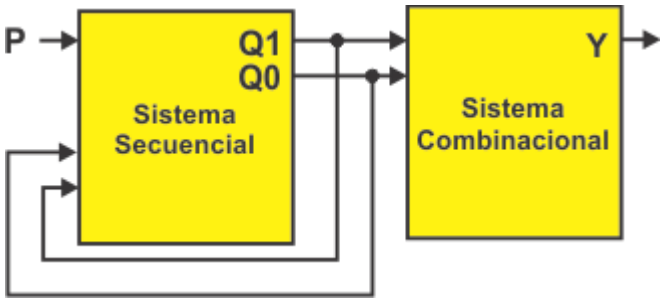
Diagrama de transición



2.- Tabla de flujo primitiva

	P		Y
	0	1	
a	E0	E1	0
b	E2	E1	0
c	E2	E3	0
d	E0	E3	1

Podemos observar en la tabla anterior que la parte del sistema secuencial no cambia y el único cambio es la salida Y de modo que no tomando los valores anteriores de Q1 y Q0 obtenemos el valor de la parte combinacional de la salida Y:

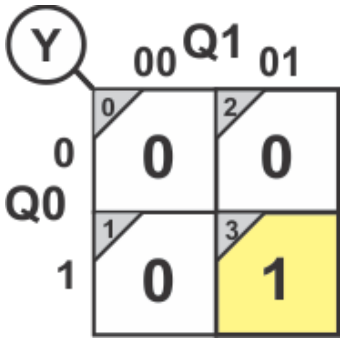


Por lo anterior no es aplicar los pasos del 3 al 9.

10.- Obtención de la ecuación mínima paya la salida Y.

La salida Y está en función de Q1 y Q2 de modo que de la tabla anterior tenemos que:

m	P	Q1	Q0	Q1+	Q0+	Y
0	0	0	0	**0**	**0**	**0**
1	0	0	1	**1**	**1**	**0**
2	0	1	0	**0**	**0**	**0**
3	0	1	1	**1**	**1**	**1**
4	1	0	0	**0**	**1**	**0**
5	1	0	1	**0**	**1**	**0**
6	1	1	0	**1**	**0**	**0**
7	1	1	1	**1**	**0**	**1**



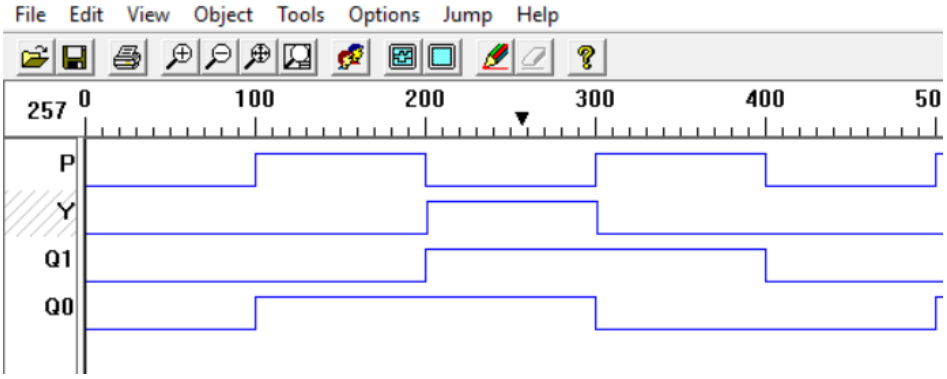
$$Y = Q1\ Q0$$

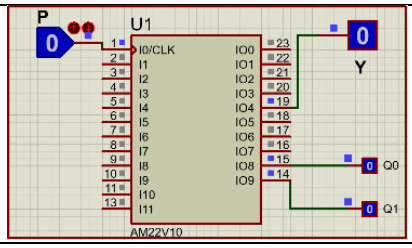
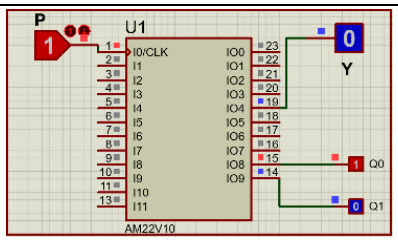
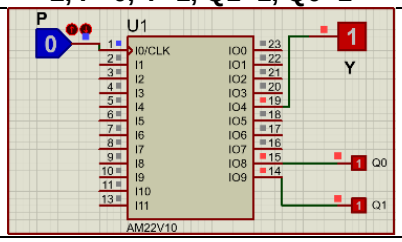
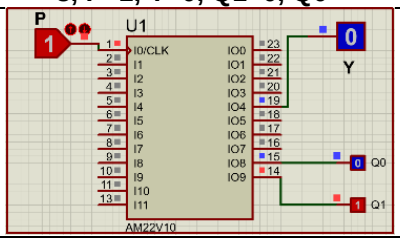
11.- Archivo en formato ABEL-HDL

```
MODULE boton
"Problema 4 JAGG
"entrada
P pin 1;
"salida
Y, Q1, Q0 pin 19..17 istype 'com';
equations
Q1=P&Q1#!P&Q0;
Q2=P&!Q1#!P&Q0;
Y=Q1&Q0;
```

```
test_vectors
(P->Y)
0->.x.;
1->.x.;
0->.x.;
1->.x.;
0->.x.;
1->.x.;
0->.x.;
1->.x.;
0->.x.;
END
```

12.- Simulaciones



0, P=0, Y=0, Q1=0, Q0=0	1, P=1, Y=0, Q1=1, Q0=0
	
2, P=0, Y=1, Q1=1, Q0=1	**3, P=1, Y=0, Q1=0, Q0=**
	

Problemas propuestos

Caso 6: Diseñe un sistema secuencial asíncrono que contenga un botón de entrada **P** y una salida **Y**, partiendo de condiciones iniciales P=0, Y=1 (T0), al oprimir por primera vez P la salida Y deberá de mantenerse en un valor de uno, P=1, Y=1 (t1), al soltar el botón (t2) la salida deberá de mantenerse, P=0, Y=1 (t2), al oprimirlo por segunda vez la señal de salida cambia P=1, Y=0 (t3), y al soltarlo la salida deberá regresar a condiciones iniciales P=0, Y=1 (t0), como lo indica la siguiente gráfica:

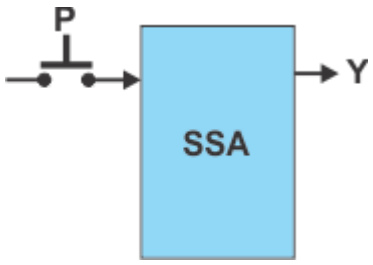


Diagrama de bloques

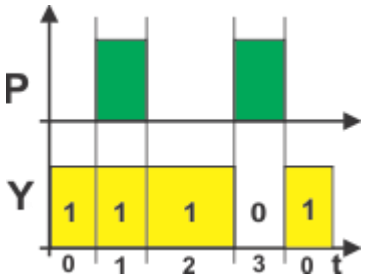


Diagrama de tiempos

1.- Especificar el sistema

Dibuje el Diagrama de transición

2.- Tabla de flujo primitiva

	P		Y
	0	1	
E0			
E1			
E2			
E3			

3.- Eliminación de estados redundantes o equivalentes.

Nota:

4.- Mezcla de Filas.

Nota:

5.- Expandir tabla de salidas.

Nota:

6.- Tabla de estados internos.

	P		Y
	0	1	
a			
b			
c			
d			

7.- Asignación de Valores a los Estados.

Columna							
0		→				→	
1		→				→	

Q2		
	0	1
0		
Q1		
1		

	Q1	Q0
a		
b		
c		
d		

8.- Tabla de Estados totales.

Sustituyendo los valores de la asignación 1, obtenemos la siguiente tabla de estados totales**:**

Q1Q0	P		Y
	0	1	

9.- Completar Tabla de Salidas.

10.- Obtención de las ecuaciones mínimas. (sugerencia usar LogicAid)

11.- Elaborar el Archivo en formato ABEL-HDL.

12.- Simulación.

13.- Representación gráfica.

a) Diagrama esquemático.

b) Diagrama escalera.

14.- Implementación

a) PLD

b) PLC

Caso 7: Diseñe un sistema secuencial asíncrono que contenga un botón de entrada **P** y una salida **Y**, partiendo de condiciones iniciales P=0, Y=1 (T0), al oprimir por primera vez P la salida Y deberá de cambiar a ceo, P=1, Y=0 (t1), al soltar el botón (t2) la salida deberá de mantenerse, P=0, Y=0 (t2), al oprimirlo por segunda vez la señal de salida se mantiene P=1, Y=0 (t3), y al soltarlo la salida deberá regresar a condiciones iniciales P=0, Y=1 (t0), como lo indica la siguiente gráfica:

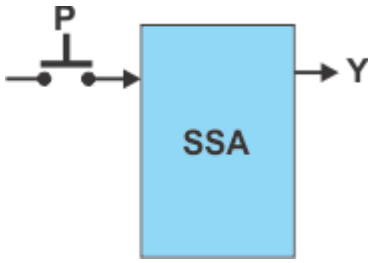


Diagrama de bloques

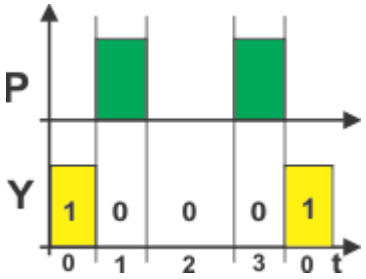


Diagrama de tiempos

1.- Especificar el sistema

Dibuje el Diagrama de transición

2.- Tabla de flujo primitiva

	P		Y
	0	1	
E0			
E1			
E2			
E3			

3.- Eliminación de estados redundantes o equivalentes.

Nota:

4.- Mezcla de Filas.

Nota:

5.- Expandir tabla de salidas.

Nota:

6.- Tabla de estados internos.

	P		Y
	0	1	
a			
b			
c			
d			

7.- Asignación de Valores a los Estados.

Columna							
0		→				→	
1		→				→	

Q2		
	0	1
0		
Q1		
1		

	Q1	Q0
a		
b		
c		
d		

8.- Tabla de Estados totales.

Sustituyendo los valores de la asignación 1, obtenemos la siguiente tabla de estados totales**:**

Q1Q0	P		Y
	0	1	

9.- Completar Tabla de Salidas.

10.- Obtención de las ecuaciones mínimas. (sugerencia usar LogicAid)

11.- Elaborar el Archivo en formato ABEL-HDL.

12.- Simulación.

13.- Representación gráfica.

a) Diagrama esquemático.

b) Diagrama escalera.

14.- Implementación

a) PLD

b) PLC

Caso 8 :Diseñe un sistema secuencial asíncrono que contenga un botón de entrada **P** y una salida **Y**, partiendo de condiciones iniciales P=0, Y=1 (T0), al oprimir por primera vez P la salida Y deberá de mantenerse, P=1, Y=1 (t1), al soltar el botón (t2) la salida deberá de cambiar, P=0, Y=0 (t2), al oprimirlo por segunda vez la señal de salida cambia P=1, Y=1 (t3), y al soltarlo la salida deberá regresar a condiciones iniciales P=0, Y=1 (t0), como lo indica la siguiente gráfica:

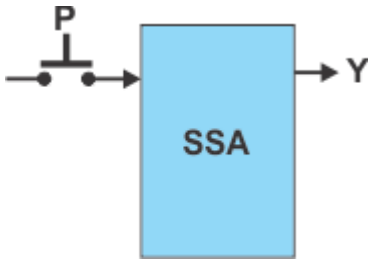


Diagrama de bloques

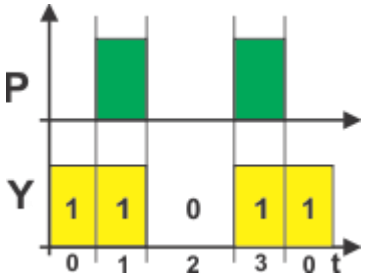


Diagrama de tiempos

1.- Especificar el sistema

Dibuje el Diagrama de transición

2.- Tabla de flujo primitiva

	P		Y
	0	1	
E0			
E1			
E2			
E3			

3.- Eliminación de estados redundantes o equivalentes.

Nota:

4.- Mezcla de Filas.

Nota:

5.- Expandir tabla de salidas.

Nota:

6.- Tabla de estados internos.

	P		Y
	0	1	
a			
b			
c			
d			

7.- Asignación de Valores a los Estados.

Columna							
0		→				→	
1		→				→	

Q2		
	0	1
0		
Q1		
1		

	Q1	Q0
a		
b		
c		
d		

8.- Tabla de Estados totales.

Sustituyendo los valores de la asignación 1, obtenemos la siguiente tabla de estados totales**:**

Q1Q0	P		Y
	0	1	

9.- Completar Tabla de Salidas.

10.- Obtención de las ecuaciones mínimas. (sugerencia usar LogicAid)

11.- Elaborar el Archivo en formato ABEL-HDL.

12 simulación.

13.- Representación gráfica.

a) Diagrama esquemático.

b) Diagrama escalera.

14.- Implementación

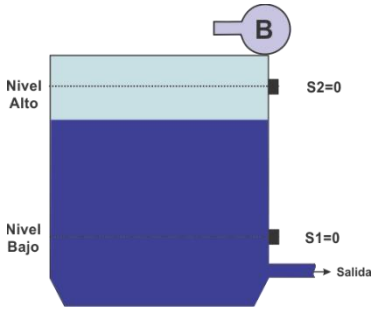
a) PLD

b) PLC

Problema 2, Control de una bomba para el llenado de un tanque

Diseñe, efectúe la simulación y construya un prototipo de un sistema secuencial asíncrono para control de llenado de un tanque por medio de una Bomba (B), el tanque cuenta con dos sensores de nivel S2 detecta nivel alto y S1 detecta Nivel bajo.

El sistema deberá de trabajar con la siguiente lógica:



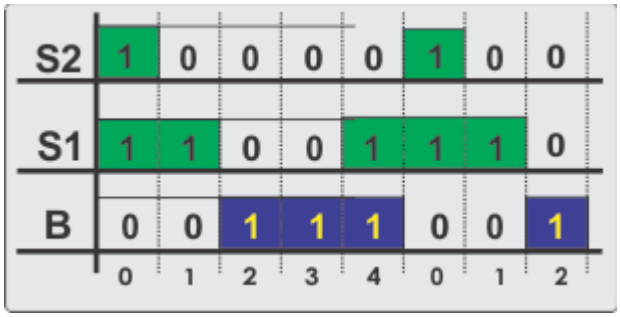
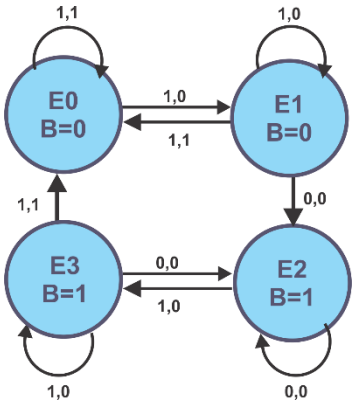
Tanque con sensores de nivel.

a)Partiendo que el tanque se encuentre lleno (S2=1, S1=1), la bomba B permanecerá apagada (B=0) y encenderá (B=1), hasta cuando el nivel del tanque esté vacío (S2=0, S1=0).

b)La bomba permanecerá encendida hasta que el nivel suba a tanque lleno (S2=1, S1=1).

1.- Especificar el Sistema.

Existen varios recursos para describir el funcionamiento del sistema secuencial, entre los que podemos mencionar son:

Diagrama de tiempos	Diagrama de Transición
	

El diagrama de tiempos es una gráfica en donde por medio de niveles (Alto=1, Bajo=0) se representa el comportamiento del sistema con respecto al tiempo.

El diagrama de transición es una representación de un sistema secuencial por medio de eventos (estados) y transiciones, en donde se muestra cada uno de los estados y su posible transición dependiendo del valor de una señal de entrada.

2.- Tabla de flujo primitiva

En la tabla se está considerando que el valor de entrada 01 no se puede presentar ya que implicaría que solo el sensor S2 detectara nivel lo cual no es posible dentro de las condiciones normales de funcionamiento**.**

1.-Considerando que en condiciones normales no se puede presentar la combinación S1=0 y S2=1, se le asigna el valor de X (Don´t Care) a la columna correspondiente.

2.-Partiendo de que el tanque se encuentra lleno (S1=1, S2=1), se crea el evento cero $E_{0,}$ que es estable (**$\underline{E_0}$**) en la fila 1 columna 11 y con la salida de la bomba B=0.

S_1S_2	00	01	11	10	B
0		X	$\underline{E_0}$		0
1		X			
2		X			
3		X			

No es posible pasar de lleno a vacío se descarta (X) la celda de la fila 1 columna 1

S_1S_2	00	01	11	10	B
0	X	X	$\underline{E_0}$		0
1		X			
2		X			
3		X			

Si el nivel llegara a medio (S1=1 y S2=0) entonces se efectuará una transición del E_0 al E_1 en donde éste es estable (**$\underline{E_1}$**) en la fila 2 columna 10 y la bomba apagada B=0.

S_1S_2	00	01	11	10	**B**
0	**X**	**X**	**$\underline{E_0}$**	E_1	**0**
1		**X**		**$\underline{E_1}$**	**0**
2		**X**			
3		**X**			

Estando en **$\underline{E_1}$** puede suceder dos acciones:

1.- Que se vuelva a llenar (S1=1, S2=1), en donde en ese caso regresaría a la fila 1 columna 11, con el mismo valor de salida.

S_1S_2	00	01	11	10	**B**
0	**X**	**X**	**$\underline{E_0}$**	E_1	**0**
1		**X**	E_0	**$\underline{E_1}$**	**0**
2		**X**			
3		**X**			

2.-Que se vacié (S1=0, S2=0), en donde en ese caso se requiere de un nuevo evento E_3, que es estable (**$\underline{E_3}$**) en la fila 3 columna 00 y cambiando el valor de la salida con la bomba encendida B=1.

S1S2	00	01	11	10	**B**
0	**X**	**X**	**$\underline{E_0}$**	E_1	**0**
1	E_2	**X**	E_0	**$\underline{E_1}$**	**0**
2	**$\underline{E_2}$**	**X**			**1**
3		**X**			

También es importante considerar que no es posible pasar de tanque vacío (S1=0, S2=0) a lleno (S1=1, S2=1), sin antes parar por el nivel medio por lo que se considera X. (fila 3 columna 11).

Estando en **E_2** estable se puede presentar que el tanque llegue a nivel medio (S1=1, S2=1), que daría lugar a un evento E3, que sería estable en la fila 4 columna 11 con la salida B=1.

S_1S_2	00	01	11	10	B
0	X	X	**E_0**	E_1	0
1	E_2	X	E_0	**E_1**	0
2	**E_2**	X		E_3	1
3		X		**E_3**	1

Estando en E_3 se pueden presentar dos eventos:

1.Que se regrese a nivel del tanque vacío (S1=0, S2=0), en ese caso se regresará a E_2

2.Que el nivel del tanque se llene (S1=1, S2=1), que en este caso se regresará a E_0, cambiando el valor de la salida B=0.

S_1S_2	00	01	11	10	B
0	X	X	**E_0**	E_1	0
1	E_2	X	E_0	**E_1**	0
2	**E_2**	X	X	E_3	1
3	E_2	X	E_0	**E_3**	1

Una vez completado el ciclo estando definidas todas las posibles transiciones declaradas en la tabla de flujo primitiva se procederá a la reducción de estados.

3.- Eliminación de estados redundantes.

Un estado es redundante si existe uno equivalente. Dos estados son equivalentes si:

d) Son estados estables en la misma columna (misma combinación de entradas).

e)Tienen la misma salida y sus estados siguientes son equivalentes

Para este ejemplo no hay reducción de estados

4.- Mezcla de Filas

Mezclando los renglones 0 con 1 y 2 con 3, de modo que no es necesaria la expansión de las salidas puesto que tienen el mismo valor.

S_1S_2	00	01	11	10	B
0,1	E_2	X	**E_0**	**E_1**	**0**
2, 3	**E_2**	X	E_0	**E_3**	**1**

5.-Expansión de Tabla de salidas.

No es necesario expandir tabla de salidas.

6.-Tabla de Estados Internos.

Sustituyendo E0 y E1 por **a** y E2 y E3 por **b**, dado que en ambos casos son estables en el mismo renglón obtenemos lo siguiente:

S_1S_2	00	01	11	10	B
a	E_2	X	**E_0**	**E_1**	**0**
b	**E_2**	X	E_0	**E_3**	**1**

S_1S_2	00	01	11	10	B
a	b	X	**a**	a	**0**
b	**b**	X	a	b	**1**

7.- Asignación de Valores a los Estados.

S_1S_2		00	01		11	10	B
a	→	b	x		**a**	a	**0**
b		**b**	x	→	a	b	**1**

Como a→b y b→a, Asignamos los valores a = 0 y b = 1:

8.- Tabla de estados totales.

Q	S_1S_2				B
	00	01	11	10	00
0	1	X	0	0	0
1	1	X	0	1	1

9.- Completar tabla de Salidas.

No es necesario completar la tabla de salidas.

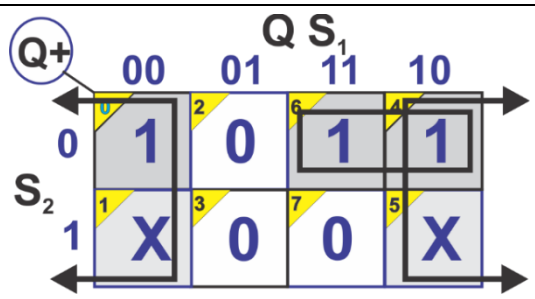
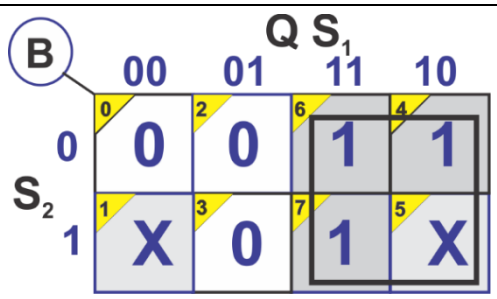
10.- Obtención de las ecuaciones mínimas.

Pasaremos la tabla de estados a tabla de verdad

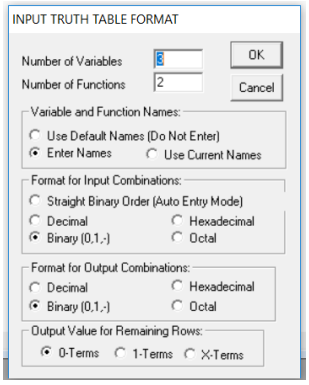
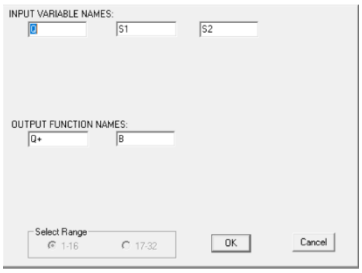
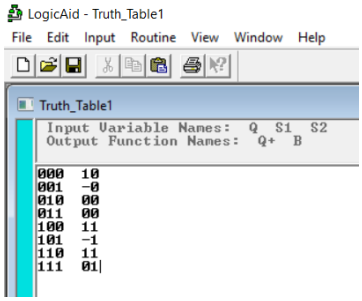
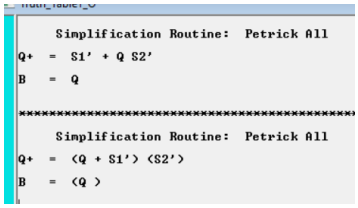
m	Q	S_1	S_2	Q+	M
0	0	0	0	**1**	**0**
1	0	0	1	**X**	**0**
2	0	1	0	**0**	**0**
3	0	1	1	**0**	**0**
4	1	0	0	**1**	**1**
5	1	0	1	**X**	**1**
6	1	1	0	**1**	**1**
7	1	1	1	**0**	**1**

Partiendo de la tabla de verdad se pueden obtener las ecuaciones mínimas, por medio del uso de mapas de Karnaugh (Kmap), o también se puede utilizar el procedimiento de manipulación algebraica o podemos utilizar un programa de aplicación (software) como el LogicAid.

A continuación, presentamos los mapas de Karnaugh con los resultados correspondientes para cada una de las salidas, para las formas And/Or y Or/And.

		
And/Or	Q= **S1'**+ Q **S2'**	B=Q
Or/And	Q = (Q + **S1'**) **S2'**	B=Q

En el software LogicAid podemos comprobar los resultados

En esta sección definimos el número de variables de entrada (3) y salidas (2), además asignaremos los nombres de las variables	
Asignación de variables de entrada y salida	
Captura de la tabla de verdad, para el caso del Don´t care se utiliza el guion -	
Resultados en: Or/And (POS) y And/Or (SOP)	

11.- Elaborar el Archivo en formato ABEL-HDL

El costo de implementación de ambos resultados es el mismo por lo que elaboraremos el archivo ABEL-HDL con la ecuación And/Or. ("Q+ = S1' + Q S2')

Archivo en formato ABEL-HDL

```
MODULE bomba
"Entradas
"21 abril 2019
S2, S1 pin 1,2;
"Salidas
Q pin 14 istype 'com';
equations
"Q+  =  S1' + Q S2'
Q=!S1 # Q &!S2;
```

```
test_vectors
([S2,S1]->[Q])
[1,1]->[.x.];
[0,1]->[.x.];
[1,1]->[.x.];
[0,1]->[.x.];
[0,0]->[.x.];
[0,1]->[.x.];
[0,0]->[.x.];
[0,1]->[.x.];
[1,1]->[.x.];
END
```

12.- Simulación. (test_vectors)

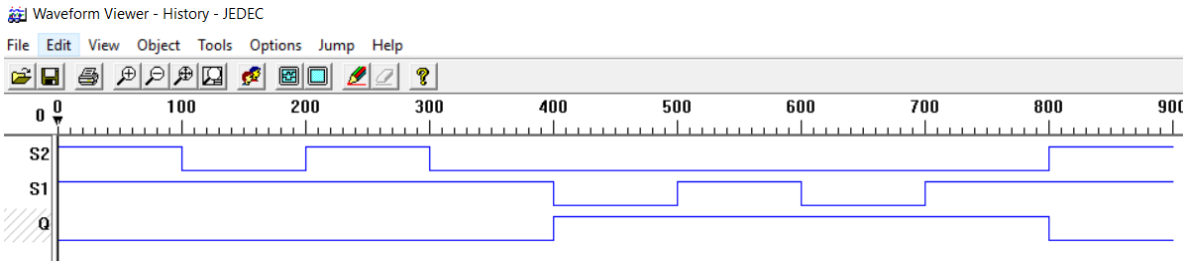
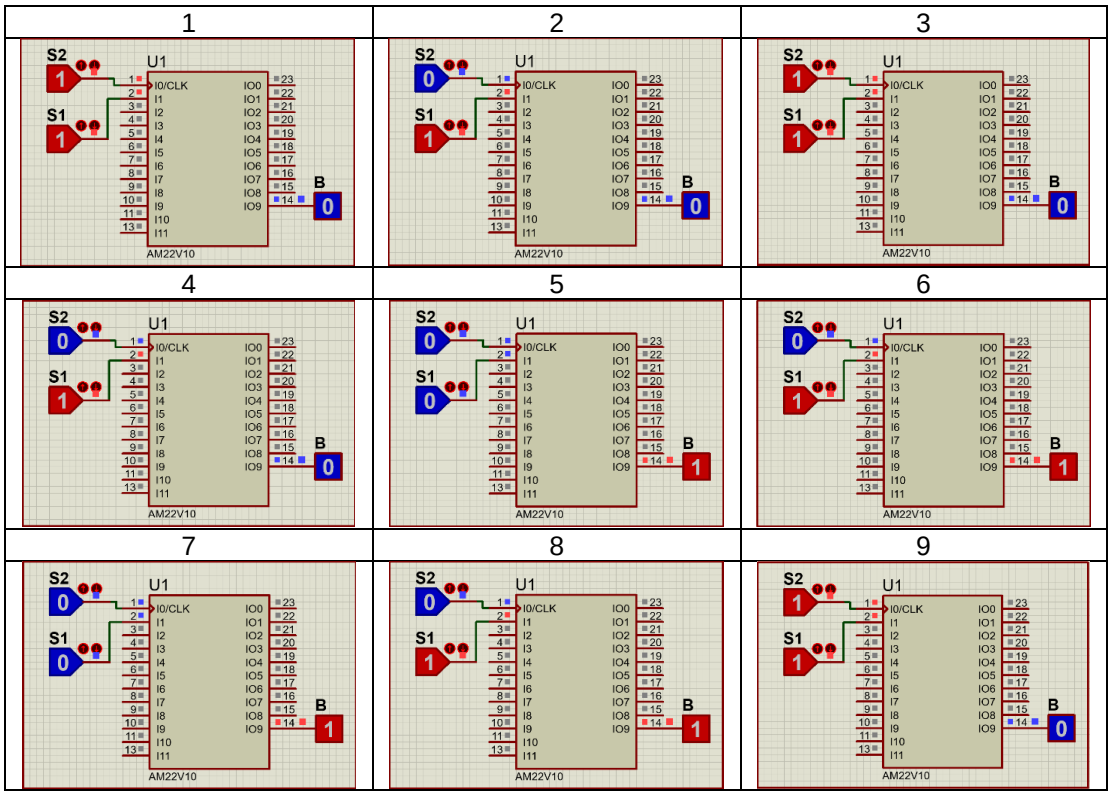


Diagrama de tiempos que demuestra el funcionamiento correcto del sistema secuencial asíncrono

Simulación. PROTEUS



En el tiempo 1 (T1) partimos con tanque lleno (S_2=1, S_1=1) y la bomba apagada (B=0), en T2 el nivel pasa a nivel medio (S_2=0, S_1=1) y aun la bomba apagada, en T3 simulamos un oleaje de nuevo el nivel lleno y en T4 regresamos a nivel medio, en T5 se muestra por primera vez el nivel bajo (S_2=0, S_1=0) y el estado de la bomba cambia a B=1 encendiéndola, en T6 el nivel sube a medio (S_2=0, S_1=1) y la boba permanece encendida, en T7 simulamos un oleaje del nivel regresando a bajo (S_2=0, S_1=0) con la bomba encendida, en T8 de nuevo regresa a niel medio (S_2=0, S_1=1) manteniendo la bomba encendida B=1 y por último en T9 pasa a tanque lleno (S_2=1, S_1=1) apagando la bomba B=0. Demostrando por medio de la simulación el correcto funcionamiento del sistema propuesto.

13.- Representación grafica

Diagrama esquemático

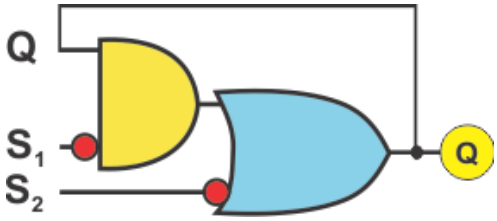
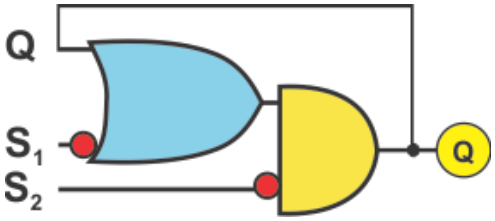
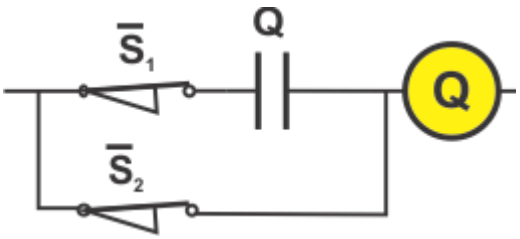
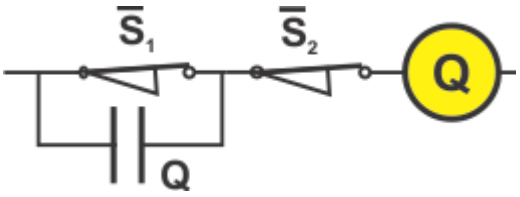
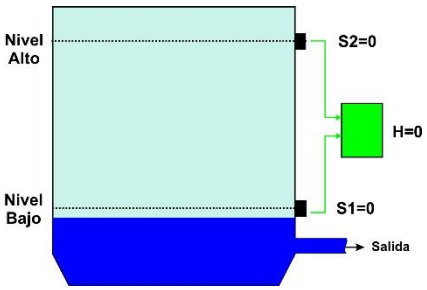
Forma And/Or	Forma Or/And
	

Diagrama escalera

Forma And/Or	Forma Or/And
	

Problema 3, Detector de nivel de un tanque.

A continuación, mostramos un ejemplo de un sistema secuencial asíncrono para detectar nivel de un tanque que cuenta con dos sensores llamados S1 (Nivel Bajo), S2 (Nivel Alto), que contenga una salida H de modo que: H=0 cuando el nivel va de S1 hacia S2 (subida) hasta que llegue a S2 y H=1 cuando el nivel va de S2 hacia S1. (bajada) hasta que llegue a S1 como lo indica el siguiente diagrama de tiempos.



Tanque con sensores de nivel.

1.- Especificar el Sistema.

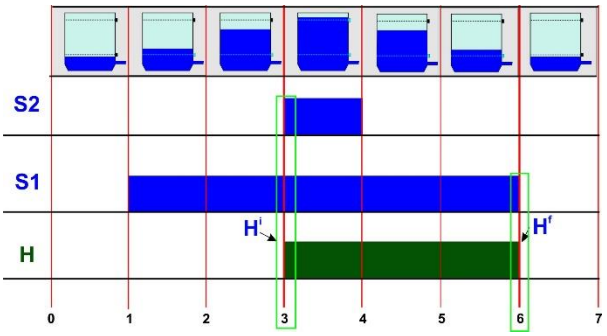


Diagrama de tiempos

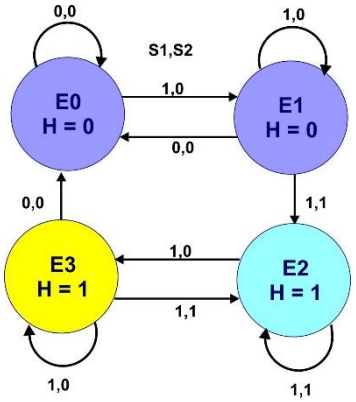


Diagrama de Transición

En el diagrama de transición podemos observar que estando en el estado E1 si el nivel baja hasta S1 (0,0) el sistema regresa al estado E0 y en el estado E3 si el nivel sube a S2 (1,1) el sistema regresa a E2 estas transiciones no están contempladas en el diagrama de tiempos.

2.- Tabla de flujo primitiva

En la tabla se está considerando que el valor de entrada 01 no se puede presentar ya que implicaría que solo el sensor S2 detectara nivel lo cual no es posible dentro de las condiciones normales de funcionamiento**.**

S_1S_2	00	01	11	10	H
0	**E_0**	-	-	E_1	0
1	E_0	-	E_2	**E_1**	0
2	-	-	**E_2**	E_3	1
3	E_0	-	E_2	**E_3**	1

3.- Reducción de Estados.

Para este ejemplo no hay reducción de estados

4.- Mezcla de Filas

Mezclando los renglones 0 con 1 y 2 con 3 además expandiendo la salida obtenemos:

	S_1S_2				**H**
	00	**01**	**11**	**10**	**00**
0, 1	**E_0**	-	E_2	**E_1**	**0**
2, 3	E_0	-	**E_2**	**E_3**	**1**

5.-Expansión de Tabla de salidas.

No es necesario expandir tabla de salidas.

6.- Estados Internos.

Sustituyendo E0 y E1 por a y E2 y E3 por b, dado que en ambos casos son estables en el mismo renglón obtenemos lo siguiente:

	S_1S_2				H
	00	01	11	10	00
a	**E_0**	-	E_2	**E_1**	0
b	E_0	-	**E_2**	**E_3**	1

	S_1S_2				H
	00	01	11	10	00
a	**a**	-	b	**a**	0
b	a	-	**b**	**b**	1

7.- Asignación de Valores a los Estados.

Como a→b y b→a, Asignamos los valores a = 0 y b = 1 :

8.- Tabla de Estados Totales.

Q	S_1S_2				H
	00	01	11	10	00
0	0	-	**1**	0	**0**
1	0	-	**1**	**1**	**1**

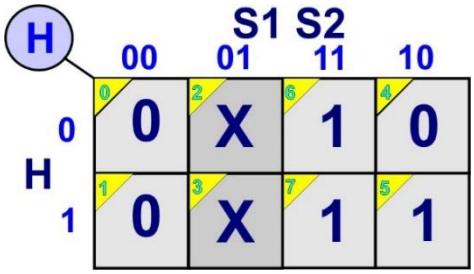
9.- Completar tabla de Salidas.

No es necesario completar la tabla de salidas.

10.- Obtención de las ecuaciones por medio de minimización.

Podemos observar en la tabla anterior que el valor de **Q^+ = H**,

Para obtener la ecuación mínima podemos hacer uso del mapa de karnaugh



Ecuaciones:

$H = S_2 + S_1H$ (1) (agrupando unos)

$H = (S_2 + H)\, S_1$ (2) (agrupando ceros)

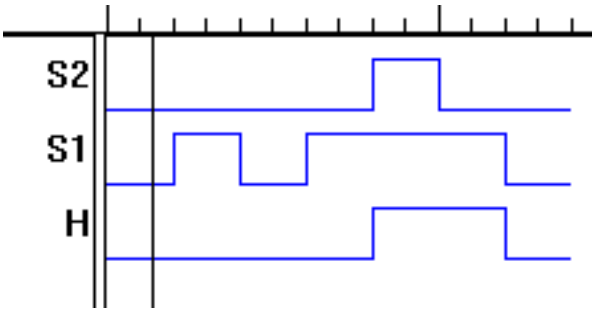
11.- Elaborar el Archivo en formato ABEL-HDL

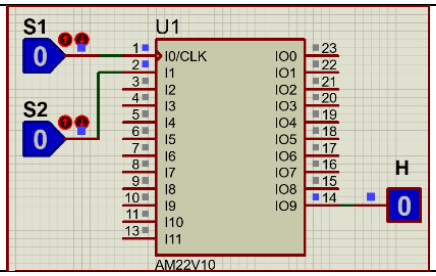
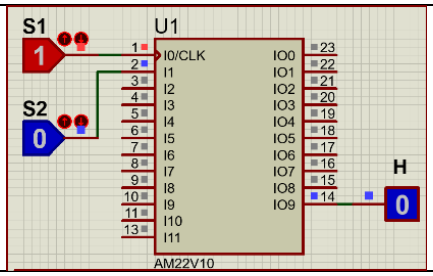
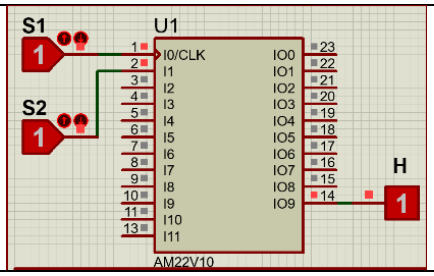
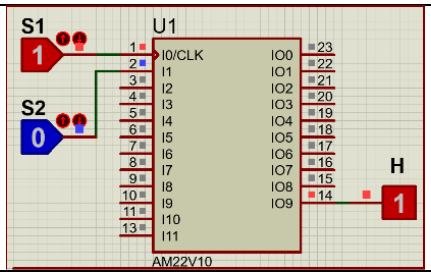
El costo de implementación de ambos resultados es el mismo por lo que elaboraremos el archivo ABEL-HDL con la ecuación (2)

Archivo en formato ABEL-HDL

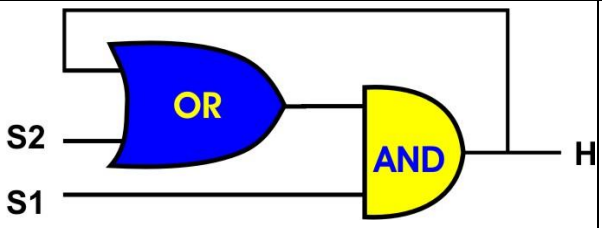
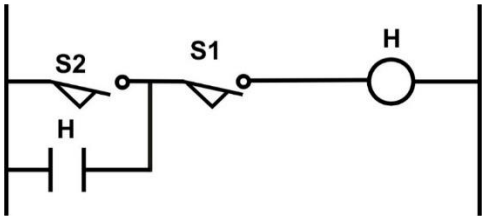
```
MODULE sensor
"Entradas
S1, S2 pin 1,2;
"Salida
H pin 19 istype 'com';
equations
H =(S2 # H)& S1;
test_vectors
([S1,S2]->[H])
[0,0]->[.x.];
[1,0]->[.x.];
[0,0]->[.x.];
[1,0]->[.x.];
[1,1]->[.x.];
[1,0]->[.x.];
[0,0]->[.x.];
END
```

12.- Simulación.



0	1
	
2	**3**
	

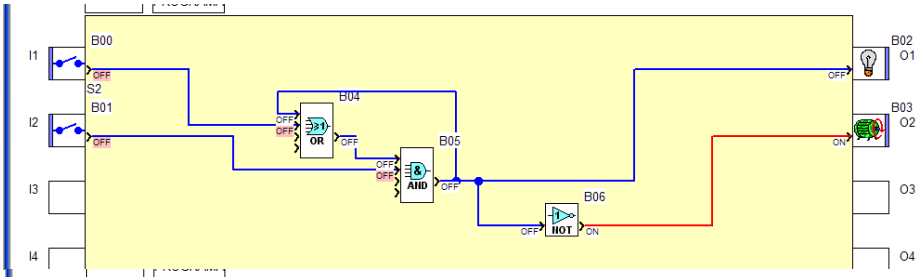
13.- Representación.

a) Diagrama esquemático	b) Diagrama escalera
	

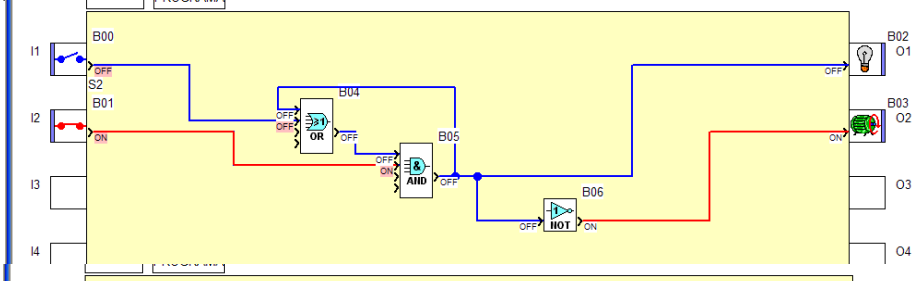
14.- Implementación.

Implementación y simulación en un PLD Marca Crouzet partiendo del diagrama esquemático, Se incluyó una salida para el motor de la bomba que es igual al complemento de H de modo que cuando H=0 el motor está trabajando y cuando H=1 el motor está apagado.

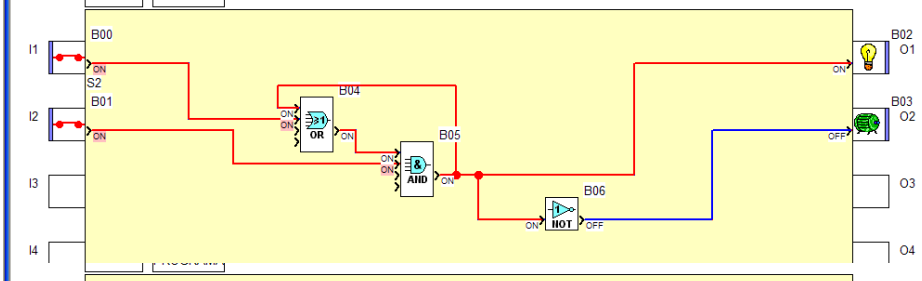
Evento 1 S1=0 y S2=0 Motor encendido y H=0



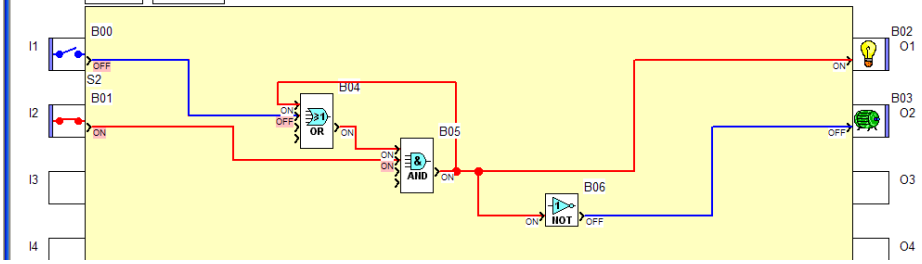
Evento 2 S1=1 y S2=0 Motor encendido y H=0



Evento 3 S1=1 y S2=1 Motor apagado y H=1



Evento 4 S1=1 y S2=0 Motor apagado y H=1



Problema 4, Sistema alternativo de dos bombas no simultaneas

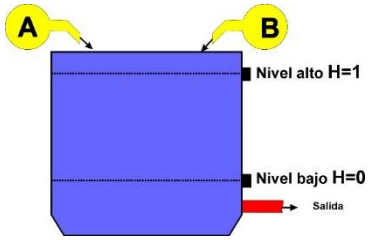
Diseñe un Sistema Secuencial que controle el llenado de un Tanque con las siguientes características:

a) El sistema consta de dos bombas llamadas **"A"** y **"B**"

b) Un sensor de nivel **"H"**, que indica con H=1 Tanque lleno y H=0 Tanque vacío (descrito en el ejercicio anterior).

c) Partiendo de que el Tanque se encuentra vacío (H=0), el llenado deberá iniciarse encendiendo la bomba "A" hasta llenar el Tanque (H=1), y entonces se apaga.

d) Si de nuevo se vacía el Tanque (H=0), el llenado deberá hacerse encendiendo ahora la bomba "B", hasta llenar el Tanque (H=1), y entonces se apaga.



e) Si se vacía de nuevo el Tanque, el llenado deberá hacerse con la bomba "A" y así sucesivamente con la finalidad de que las bombas se alternen en su funcionamiento.

1.- Especificar el Sistema.

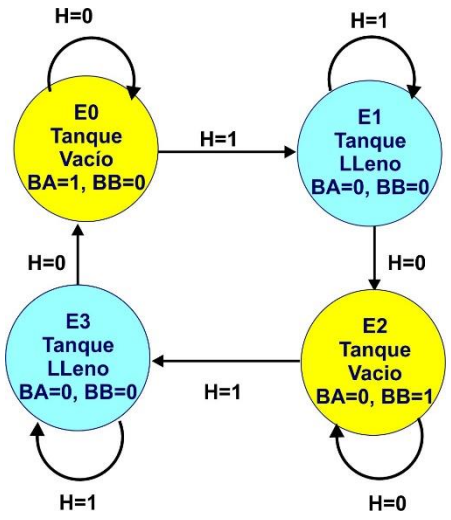


Diagrama de transición

2.- Tabla de Flujo Primitiva.

	Entrada H		Salidas	
	0	**1**	**A**	**B**
0	**E_0**	E1	**1**	**0**
1	E_2	**E_1**	**0**	**0**
2	**E_2**	E_3	**0**	**1**
3	E_0	**E_3**	**0**	**0**

3.- Reducción de Estados.

No hay reducción de estados

4.- Mezcla de Filas.

No es posible mezclar filas

	Entrada H		Salidas		
	0	**1**	**A**	**B**	
0	**E_0**	E1	**1**	**0**	a
1	E_2	**E_1**	**0**	**0**	b
2	**E_2**	E_3	**0**	**1**	c
3	E_0	**E_3**	**0**	**0**	d

5.-Expandir Tabla de Salidas.

No es necesario expandir tabla de salidas.

6.- Tabla de Estados Internos.

Sustituyendo los nombres de los estados por las variables propuestas obtenemos:

	Entrada H		Salidas	
	0	**1**	**A**	**B**
a	**a**	b	**1**	**0**
b	c	**b**	**0**	**0**
c	**c**	d	**0**	**1**
d	a	**d**	**0**	**0**

7.- Asignación de Valores a los Estados.

	Entrada H				Salidas	
	0		1		A	B
a		**a**	→	b	1	0
b	→	c		**b**	0	0
c		**c**	→	d	0	1
d	→	a		**d**	0	0

Requisitos: columna 0) (b → c), (d → a)

Columna 1), (a → b), (c → d),

Q1 \ Q2	0	1
0	a	d
1	b	c

Asignación

a =00, b = 01, c = 11, d= 10

8.- Tabla de Estados Totales.

Sustituyendo las variables por los valores asignados obtenemos:

Q1 Q2	Entrada H		Salidas	
	0	1	A	B
00	00	01	1	0
01	11	01	0	0
11	11	10	0	1
10	00	10	0	0

9.- Completar Tabla de Salidas.

No es necesario completar tabla de salidas.

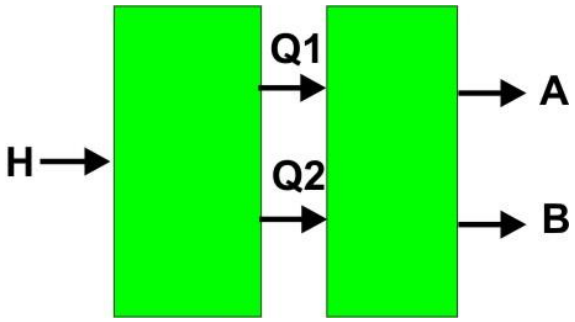
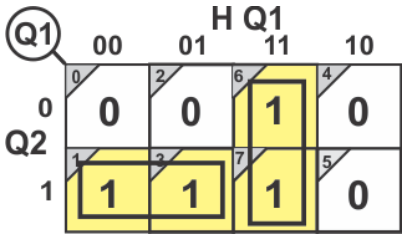


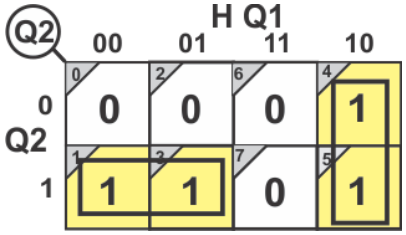
Diagrama de Bloques de la máquina de Moore

10.- Ecuaciones mínimas.

m	H	Q1	Q2	Q1+	Q2+
0	0	0	0	0	0
1	0	0	1	1	1
2	0	1	0	0	0
3	0	1	1	1	1
4	1	0	0	0	1
5	1	0	1	0	1
6	1	1	0	1	0
7	1	1	1	1	0



Q1=**H'**Q2 + H Q1



Q1=**H'**Q2 + H **Q1'**

c) Tabla de salidas obteniendo los Minitérminos de las funciones A y B.

m	Q1	Q2	A	B
0	0	0	1	0
1	0	1	0	0
2	1	0	0	0
3	1	1	0	1

A = **Q1' Q2'**

B = Q1 Q2

Ecuaciones obtenidas por medio de LogicAid

And/Or

Q1 = **H'**Q2 + H Q1

Q2 = **H'**Q2 + H **Q1'**

A = **Q1' Q2'**

B = Q1 Q2

Or/And

$$Q1 = (H + Q2\ )\ (H' + Q1\ )$$

$$Q2 = (H + Q2\ )\ (H' + Q1')$$

$$A = \mathbf{Q1'\ Q2'}$$

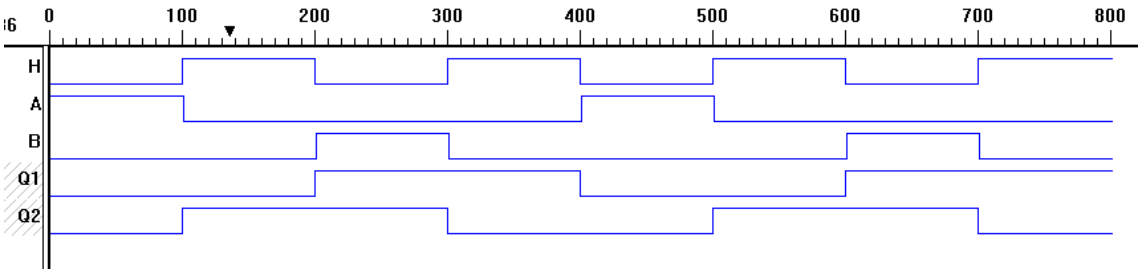
$$B = Q1\ Q2$$

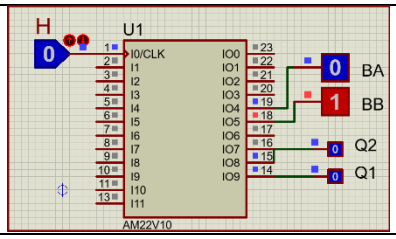
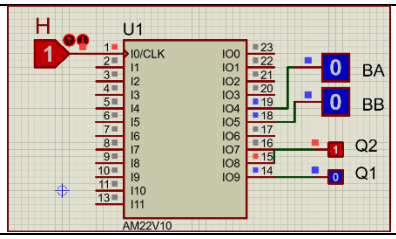
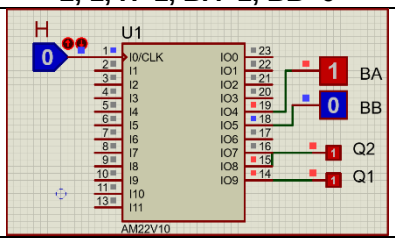
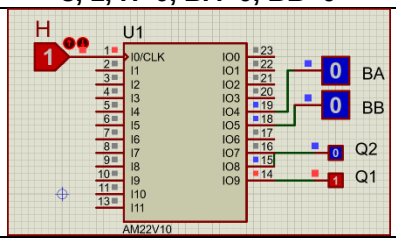
11.- Elaborar el Archivo en formato ABEL-HDL

```
MODULE dbayb
"Entrada
H pin 1;
"Salidas
Q1,Q2,A,B pin 19..16 istype 'com';
equations
Q1=!H&Q2#H&Q1;
Q2=!H&Q2#H&!Q1;
A=!Q1&!Q2;
B=Q1&Q2;
```

```
test_vectors
(H->[A,B])
0->[.x.,.x.];
1->[.x.,.x.];
0->[.x.,.x.];
1->[.x.,.x.];
0->[.x.,.x.];
1->[.x.,.x.];
0->[.x.,.x.];
1->[.x.,.x.];
END
```

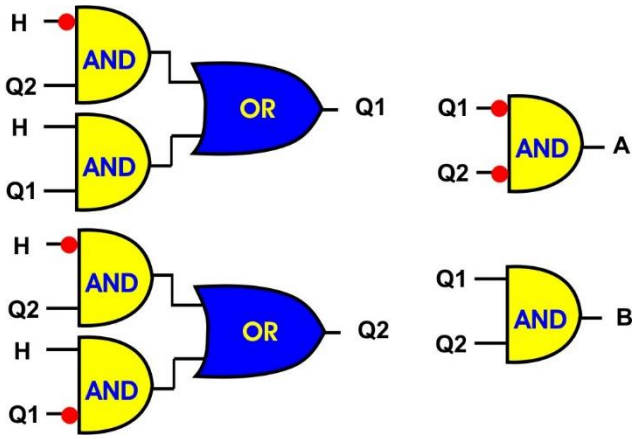
12.- Simulaciones Test Vectors y PROTEUS .



0, H=0, BA=0, BB=1	1, H=0, BA=0, BB=0
	
2, 1, H=1, BA=1, BB=0	**3, 1, H=0, BA=0, BB=0**
	

13.- Representación.

a) diagrama esquemático



b) Diagrama escalera

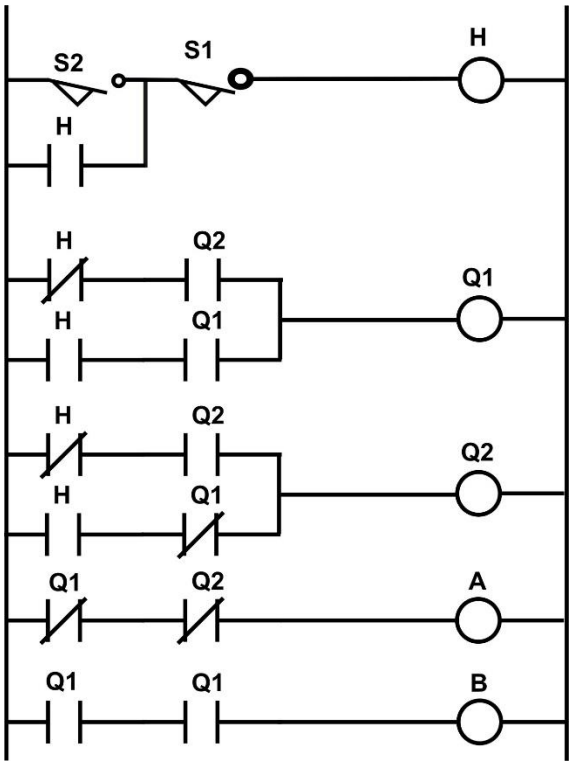


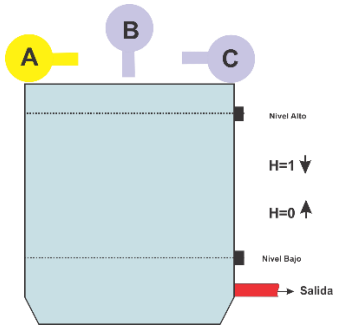
Diagrama escalera incluyendo sensor de nivel

14 implementación

Problema 5, Sistema alternativo de tres bombas no simultaneas

Diseñe, efectúe la simulación y construya un prototipo de un sistema secuencial asíncrono para control de llenado de un tanque por medio de tres Bombas llamadas A, B y C, el tanque cuenta con un sensor de nivel H, en donde H=0 detecta nivel bajo y H=1 detecta Nivel alto.

El sistema deberá de trabajar con la siguiente lógica:



Tanque con sensores de nivel.

d)Partiendo que el tanque se encuentre vacío (H=0), el llenado deberá de iniciarse con la bomba A, y apagarse cuando se detecte nivel alto (H=1).

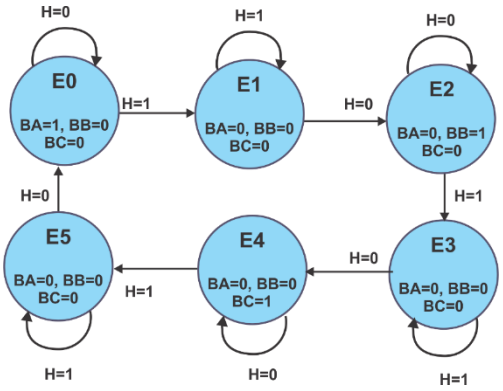
e)Si de nuevo el sensor indica nivel bajo (H=0), ahora trabajara solo la bomba B y apagarse cuando se detecte nivel alto (H=1).

f)Si de nuevo el sensor indica nivel bajo (H=0), ahora trabajara solo la bomba C y apagarse cuando se detecte nivel alto (H=1).

g)Si de nuevo el sensor indica nivel bajo (H=0), ahora trabajara de nuevo solo la bomba A y apagarse cuando se detecte nivel alto (H=1). Y así sucesivamente con la finalidad de que las bombas se alternen en su funcionamiento.

1.- Especificar el Sistema.

Diagrama de Transición



2.- Tabla de flujo primitiva

H	0	1	BA	BB	BC
0	**E0**	E1	1	0	0
1	E2	**E1**	0	0	0
2	**E2**	E3	0	1	0
3	E4	**E3**	0	0	0
4	**E4**	E5	0	0	1
5	E0	**E5**	0	0	0

3.- Reducción de Estados.

Un estado es redundante si existe uno equivalente. Dos estados son equivalentes si:

f) Son estados estables en la misma columna (misma combinación de entradas).

g)Tienen la misma salida y sus estados siguientes son equivalentes

Para este ejemplo no hay reducción de estados

4.- Mezcla de Filas

H	0	1	BA	BB	BC
0	**E0**	E1	1	0	0
1	E2	**E1**	0	0	0
2	**E2**	E3	0	1	0
3	E4	**E3**	0	0	0
4	**E4**	E5	0	0	1
5	E0	**E5**	0	0	0

No hay mezcla de filas

5.-Expansión de Tabla de salidas.

No es necesario expandir tabla de salidas.

6.- Estados Internos.

Sustituyendo E0 y E1 por **a** y E2 y E3 por **b**, dado que en ambos casos son estables en el mismo renglón obtenemos lo siguiente:

H	0	1	BA	BB	BC
0	**E0**	E1	1	0	0
1	E2	**E1**	0	0	0
2	**E2**	E3	0	1	0
3	E4	**E3**	0	0	0
4	**E4**	E5	0	0	1
5	E0	**E5**	0	0	0

H	0	1	BA	BB	BC
a	**a**	b	1	0	0
b	c	**b**	0	0	0
c	**c**	d	0	1	0
d	e	**d**	0	0	0
e	**e**	f	0	0	1
f	a	**f**	0	0	0

7.- Asignación de Valores a los Estados. (1)

Como a→b y b→c, c→d y d→e, e→f y f→a,

Q2 Q1

Q0	00	01	11	10
0	a	d	e	f
1	b	c		

Para que se cumpla la asignación debe de haber un solo cambio entre los valores de las variables, en donde se propone la siguiente asignación:

	Q2	Q1	Q0
a	0	0	0
b	0	0	1
c	0	1	1
d	0	1	0
e	1	1	0
f	1	0	0

8.- Tabla de Estados Totales.

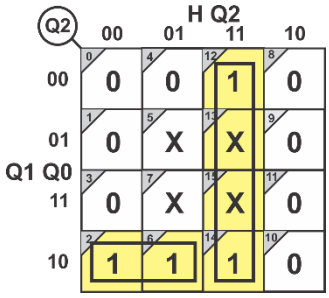
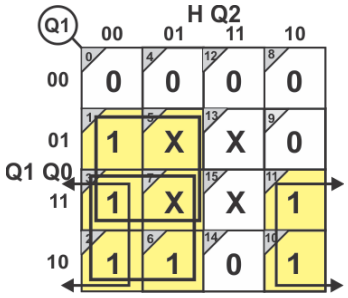
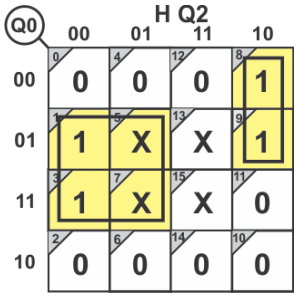
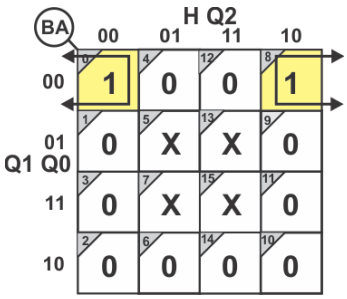
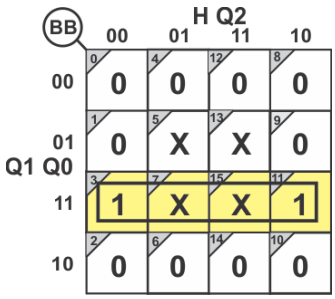
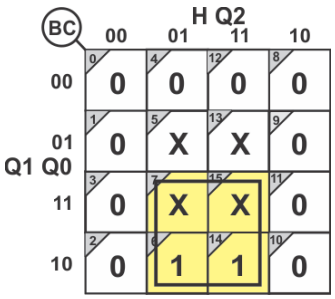
H	0	1	BA	BB	BC
000	000	001	1	0	0
001	011	001	0	0	0
011	011	010	0	1	0
010	110	010	0	0	0
110	110	100	0	0	1
100	000	100	0	0	0

9.- Completar tabla de Salidas. No es necesario completar la tabla de salidas.

10.- Obtención de las ecuaciones mínimas. Tabla de verdad

m	H	Q2	Q1	Q0	Q2+	Q1+	Q0+	BA	BB	BC
0	0	0	0	0	0	0	0	1	0	0
1	0	0	0	1	0	1	1	0	0	0
2	0	0	1	0	1	1	0	0	0	0
3	0	0	1	1	0	1	1	0	1	0
4	0	1	0	0	0	0	0	0	0	0
5	0	1	0	1	X	X	X	X	X	X
6	0	1	1	0	1	1	0	0	0	1
7	0	1	1	1	X	X	X	X	X	X
8	1	0	0	0	0	0	1	1	0	0
9	1	0	0	1	0	0	1	0	0	0
10	1	0	1	0	0	1	0	0	0	0
11	1	0	1	1	0	1	0	0	1	0
12	1	1	0	0	1	0	0	0	0	0
13	1	1	0	1	X	X	X	X	X	X
14	1	1	1	0	1	0	0	0	0	1
15	1	1	1	1	X	X	X	X	X	X

10.- Ecuaciones Mínimas

		
Q2=H' Q1 **Q0' +** H Q2	Q1=**H'**Q0+**H'**Q1+**Q2'**Q1	Q0=**H'**Q0+H**Q2'Q1'**
		
BA= **Q2' Q1´Q0'**	BB= Q1 Q0	BC=Q2 Q1

Ecuaciones mínimas (LogicAid)

And/Or.

Q2= **H'**Q1 **Q0'** + H Q2

Q1= **H'**Q0 + **H'**Q1 + **Q2'**Q1

Q0= **H'**Q0 + H **Q2'Q1'**

BA= **Q2' Q1´Q0'**

BB= Q1 Q0

BC=Q2 Q1

Or/And

Q2+=(**Q0'**) (H + Q1) (**H'**+ Q2)

Q1+=(Q1+Q0)(**H'**+Q1)(**H'**+ Q2')

Q0+=(H + Q0) (**H'**+ **Q1**') (**Q2'**)

11.- Archivo en formato ABEL-HDL

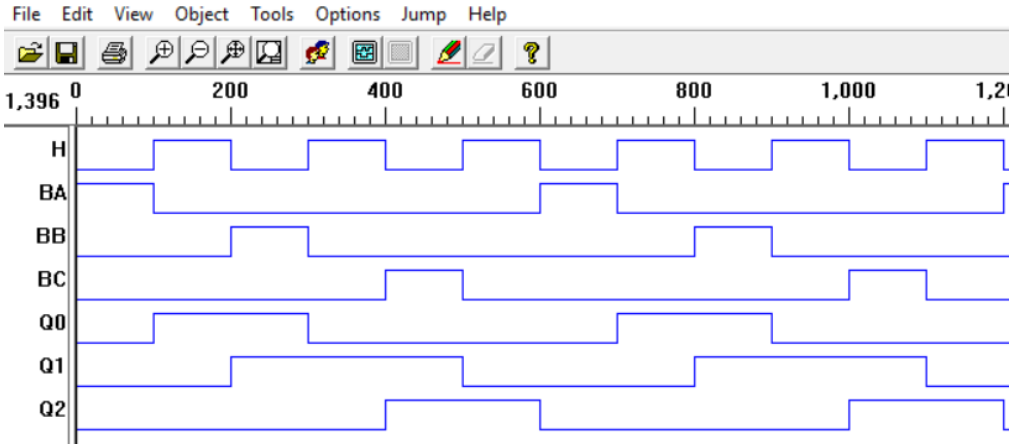
El costo de implementación de ambos resultados es el mismo por lo que elaboraremos el archivo ABEL-HDL con la ecuación And/Or. ("Q+ = **S1'** + Q **S2'**)

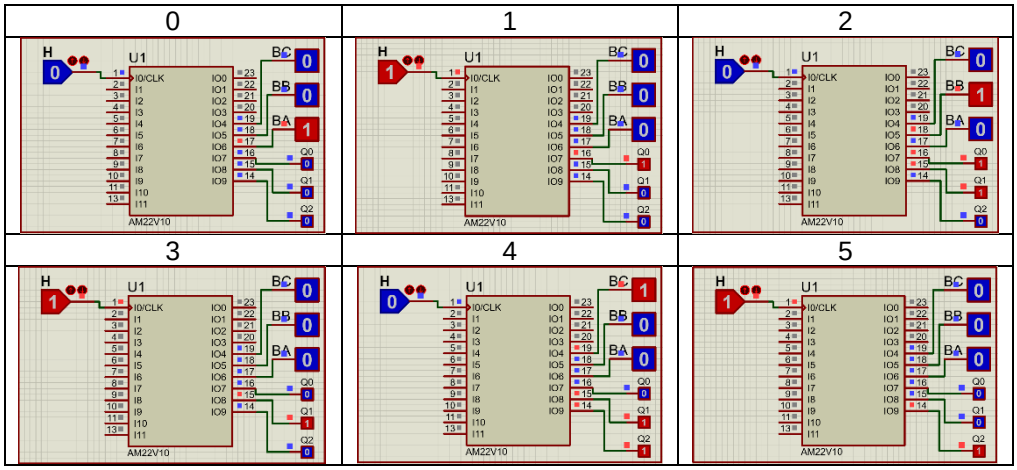
Archivo en formato ABEL-HDL

```
MODULE Pocho
"Sistema alternativo de tres bombas no simultaneas
"Entradas
"7 DE Mayo 2019
H pin 1;
"Salidas
Q2,Q1,Q0,BA,BB,BC pin 14..19 istype 'com';
equations
 Q2= !H&Q1&!Q0#H&Q2;
Q1= !H&Q0#!H&Q1#!Q2&Q1;
 Q0= !H&Q0#H&!Q2&!Q1;
BA=!Q2&!Q1&!Q0;
BB = Q1&Q0;
BC=Q2&Q1;
Test_vectors
(H->[Q2,Q1,Q0,BA,BB,BC])
0->[.x.,.x.,.x.,.x.,.x.,.x.];
1->[.x.,.x.,.x.,.x.,.x.,.x.];
0->[.x.,.x.,.x.,.x.,.x.,.x.];
1->[.x.,.x.,.x.,.x.,.x.,.x.];
0->[.x.,.x.,.x.,.x.,.x.,.x.];
1->[.x.,.x.,.x.,.x.,.x.,.x.];
0->[.x.,.x.,.x.,.x.,.x.,.x.];
1->[.x.,.x.,.x.,.x.,.x.,.x.];
0->[.x.,.x.,.x.,.x.,.x.,.x.];
1->[.x.,.x.,.x.,.x.,.x.,.x.];
0->[.x.,.x.,.x.,.x.,.x.,.x.];
1->[.x.,.x.,.x.,.x.,.x.,.x.];
0->[.x.,.x.,.x.,.x.,.x.,.x.];
1->[.x.,.x.,.x.,.x.,.x.,.x.];
0->[.x.,.x.,.x.,.x.,.x.,.x.];
1->[.x.,.x.,.x.,.x.,.x.,.x.];

END
```

12.- Simulación. (test_vectors)

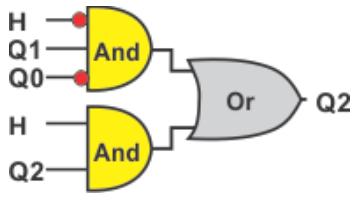
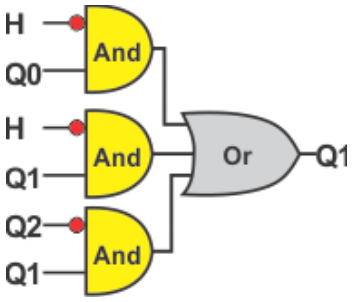
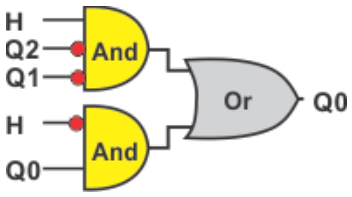
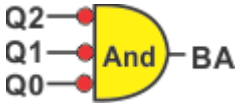
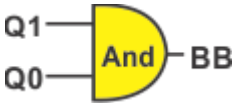
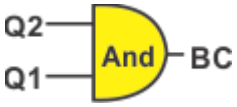




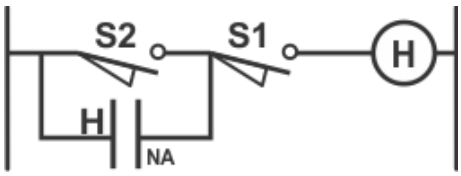
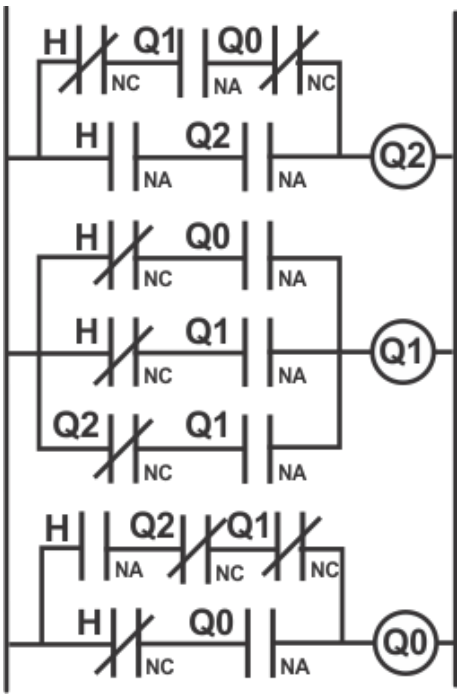
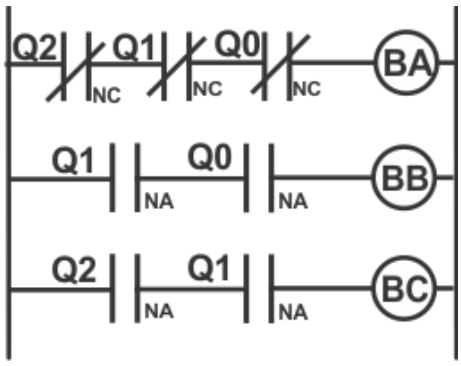
Simulación en PROTEUS mostrando los valores de salidas para cada uno de los seis tiempos del ciclo

13 Representación Gráfica

Diagrama esquemático

		
Q2=H' Q1 **Q0' +** H Q2	Q1=**H'**Q0+**H'**Q1+**Q2'**Q1	Q0=**H'**Q0+H**Q2'Q1'**
		
BA= **Q2' Q1´Q0'**	BB= Q1 Q0	BC=Q2 Q1

Diagramas escalera

	
Sensor de Nivel	
	
Sistema Combinacional	**Sistema Secuencial**

7.- Propondremos una asignación diferente de Valores a los Estados (2).

Como a→b y b→c, c→d y d→e, e→f y f→a

Q0 \ Q2 Q1	00	01	11	10
0	a	b		f
1		c	d	e

Para que se cumpla la asignación debe de haber un solo cambio entre los valores de las variables, en donde se propone la siguiente asignación:

	Q2	Q1	Q0
a	0	0	0
b	0	1	0
c	0	1	1
d	1	1	1
e	1	0	1
f	1	0	0

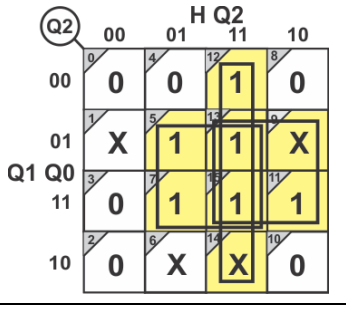
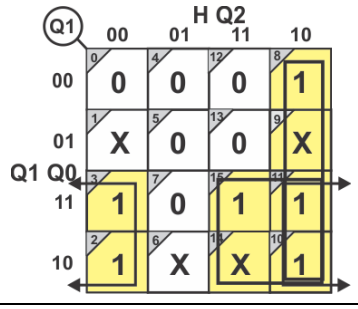
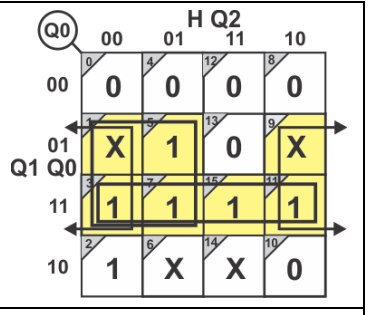
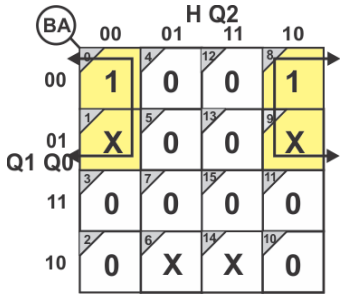
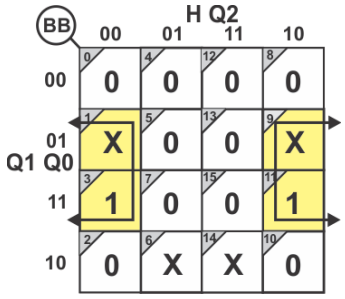
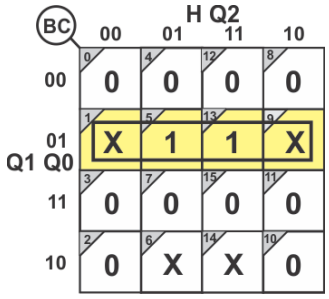
8.- Tabla de Estados Totales.

H	0	1	BA	BB	BC
000	000	010	1	0	0
010	011	010	0	0	0
011	011	111	0	1	0
111	101	111	0	0	0
101	101	100	0	0	1
100	000	100	0	0	0

Tabla de verdad

m	H	Q2	Q1	Q0	Q2+	Q1+	Q0+	BA	BB	BC
0	0	0	0	0	**0**	**0**	**0**	**1**	**0**	**0**
1	0	0	0	1	**X**	**X**	**X**	**X**	**X**	**X**
2	0	0	1	0	**0**	**1**	**1**	**0**	**0**	**0**
3	0	0	1	1	**0**	**1**	**1**	**0**	**1**	**0**
4	0	1	0	0	**0**	**0**	**0**	**0**	**0**	**0**
5	0	1	0	1	**1**	**0**	**1**	**0**	**0**	**1**
6	0	1	1	0	**X**	**X**	**X**	**X**	**X**	**X**
7	0	1	1	1	**1**	**0**	**1**	**0**	**0**	**0**
8	1	0	0	0	**0**	**1**	**0**	**1**	**0**	**0**
9	1	0	0	1	**X**	**X**	**X**	**X**	**X**	**X**
10	1	0	1	0	**0**	**1**	**0**	**0**	**0**	**0**
11	1	0	1	1	**1**	**1**	**1**	**0**	**1**	**0**
12	1	1	0	0	**1**	**0**	**0**	**0**	**0**	**0**
13	1	1	0	1	**1**	**0**	**0**	**0**	**0**	**1**
14	1	1	1	0	**X**	**X**	**X**	**X**	**X**	**X**
15	1	1	1	1	**1**	**1**	**1**	**0**	**0**	**0**

10.- Ecuaciones Mínimas

		
Q2=HQ0+ Q2+Q2Q0	Q1 =H **Q2'**+HQ1+**Q2'**Q1	Q0= **H'**Q1+**H'**Q0+Q1Q0
		
BA= **Q2´Q1'**	BB= **Q2'** Q0	BC=**Q1'** Q0

Ecuaciones mínimas (LogicAid)

And/Or

Q2 = H Q0 + H Q2 + Q2 Q0

Q1 = H **Q2'** + H Q1 + **Q2'**Q1

Q0 = **H'**Q1 + **H'**Q0 + Q1 Q0

BA = **Q2'Q1'**

BB = **Q2'**Q0

BC = **Q1'**Q0

Or/And

Q2=(H+Q2)(H+Q0)(Q2 + Q0)

Q1= (H+Q1)(H+**Q2'**)(**Q2'**+ Q1)

Q0=(**H'**+Q0)(**H'**+Q1)(Q1+ Q0)

BA = (**Q1') (Q2'**)

BB = (Q0) (**Q2'**)

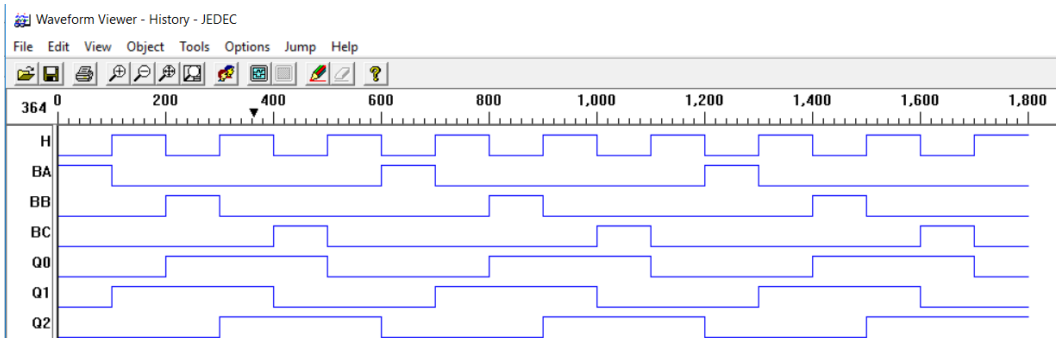
BC = (Q0) (Q1')

Código ABEL-HDL

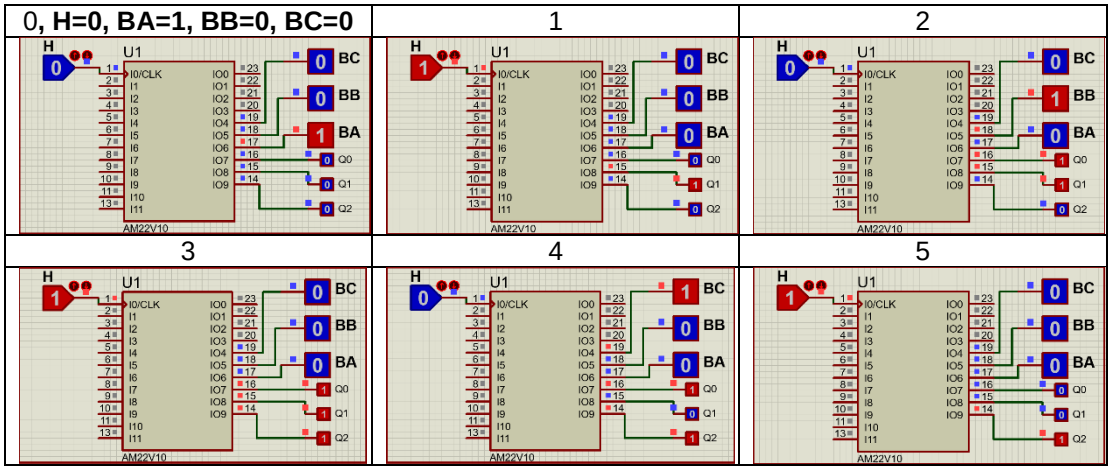
```
MODULE TRESB
"Entradas
"21 abril 2019
H pin 1;
"Salidas
Q2,Q1,Q0,BA,BB,BC pin 14..19 istype 'com';
equations
"Q2+=H Q0 + H Q2 + Q2 Q0
Q2=H&Q0#H&Q2#Q0&Q2;
"Q1+=H Q2' + H Q1 + Q2'Q1
Q1=H&!Q2#H&Q1#!Q2&Q1;
"Q0+= H'Q1 + H'Q0 + Q1 Q0
Q0=!H&Q1 #!H&Q0#Q1&Q0;
"BA = Q2'Q1'
BA=!Q2&!Q1;
"BB = Q2'Q0
BB =!Q2&Q0 ;
"BC = Q1'Q0
BC=!Q1&Q0;

Test_vectors
(H->[Q2,Q1,Q0,BA,BB,BC])
0->[.x.,.x.,.x.,.x.,.x.,.x.];
1->[.x.,.x.,.x.,.x.,.x.,.x.];
0->[.x.,.x.,.x.,.x.,.x.,.x.];
1->[.x.,.x.,.x.,.x.,.x.,.x.];
0->[.x.,.x.,.x.,.x.,.x.,.x.];
1->[.x.,.x.,.x.,.x.,.x.,.x.];
0->[.x.,.x.,.x.,.x.,.x.,.x.];
1->[.x.,.x.,.x.,.x.,.x.,.x.];
0->[.x.,.x.,.x.,.x.,.x.,.x.];
1->[.x.,.x.,.x.,.x.,.x.,.x.];
0->[.x.,.x.,.x.,.x.,.x.,.x.];
1->[.x.,.x.,.x.,.x.,.x.,.x.];
0->[.x.,.x.,.x.,.x.,.x.,.x.];
1->[.x.,.x.,.x.,.x.,.x.,.x.];
0->[.x.,.x.,.x.,.x.,.x.,.x.];
1->[.x.,.x.,.x.,.x.,.x.,.x.];
0->[.x.,.x.,.x.,.x.,.x.,.x.];
1->[.x.,.x.,.x.,.x.,.x.,.x.];
END
```

12.- Simulaciones



Simulación en Test_vectors mostrando cada uno de los seis tiempos del ciclo



Simulación en PROTEUS mostrando los valores de salidas para cada uno de los seis tiempos del ciclo

13.- Representación Gráfica

Diagrama esquemático

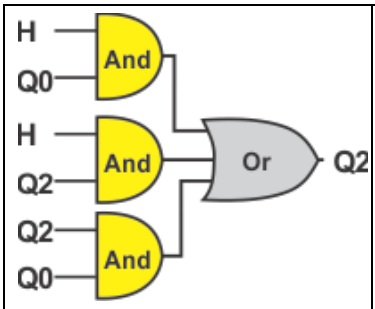
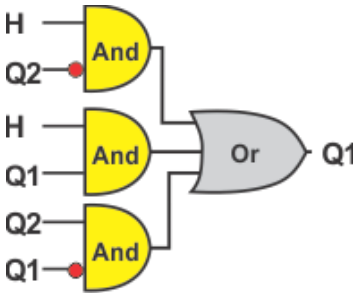
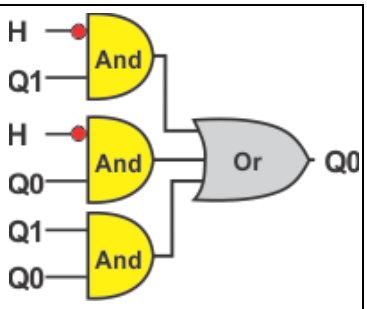
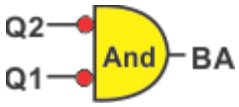
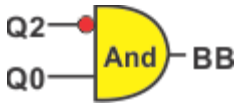
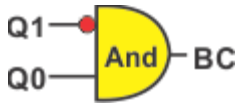
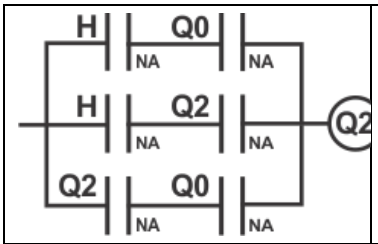
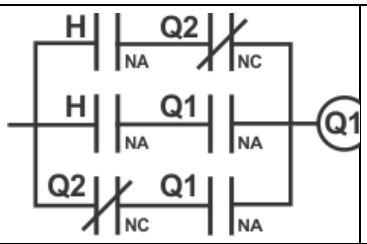
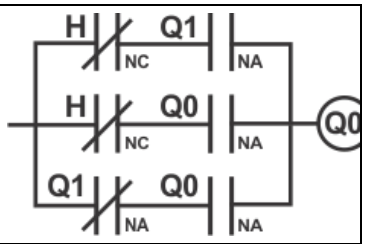
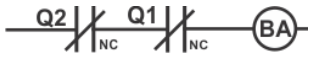
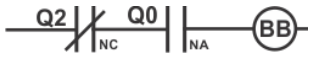
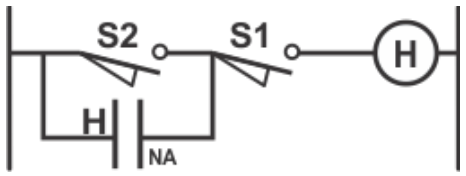
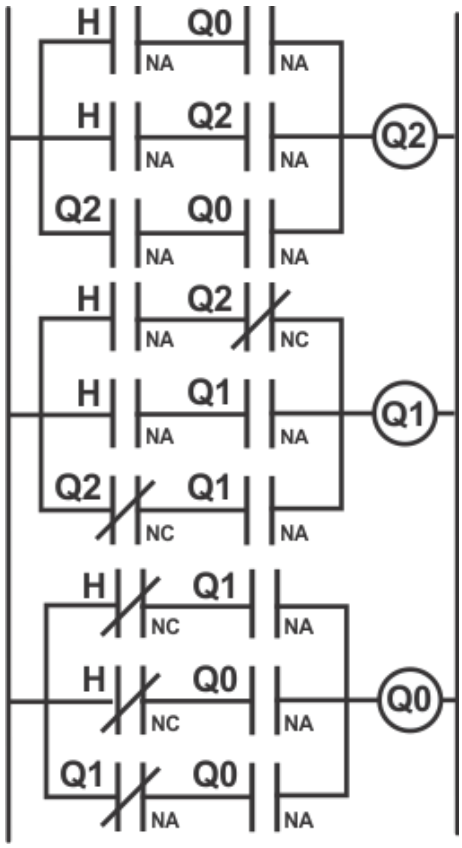
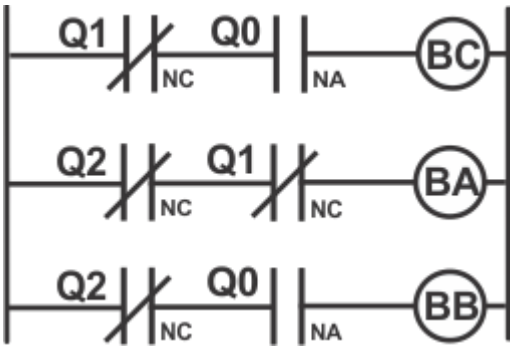
		
Q2=HQ0+ Q2+Q2Q0	Q1 =H **Q2'**+HQ1+**Q2'**Q1	Q0= **H'**Q1+**H'**Q0+Q1Q0
		
BA= **Q2´Q1'**	BB= **Q2'** Q0	BC=**Q1'** Q0

Diagrama Escalera

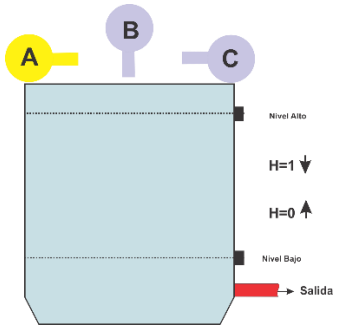
		
Q2=HQ0+ Q2+Q2Q0	Q1 =H **Q2'**+HQ1+**Q2'**Q1	Q0= **H'**Q1+**H'**Q0+Q1Q0
		
BA= **Q2´Q1'**	BB= **Q2'** Q0	BC=**Q1'** Q0

	
Sensor del nivel	
	
Sistema Combinacional	Sistema Secuencial

Problema 6, Sistema alternativo de 2 de 3 bombas

Diseñe, efectúe la simulación y construya un prototipo de un sistema secuencial asíncrono para control de llenado de un tanque por medio de tres Bombas llamadas A, B y C, el tanque cuenta con un sensor de nivel H, en donde H=0 detecta nivel bajo y H=1 detecta Nivel alto.

El sistema deberá de trabajar con la siguiente lógica:



Tanque con sensores de nivel.

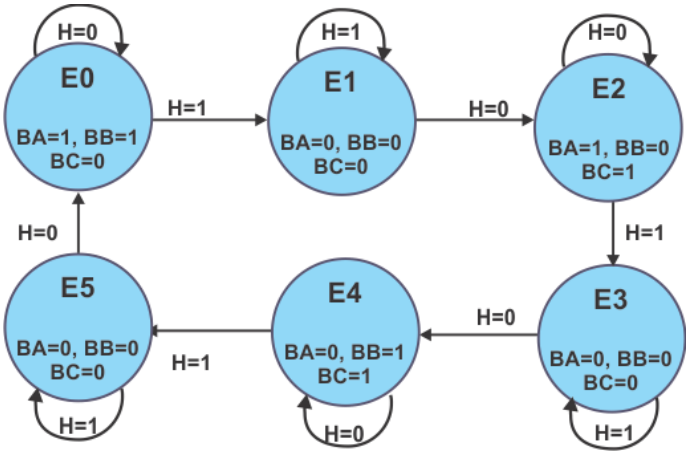
a) Partiendo que el tanque se encuentre vacío (H=0), el llenado deberá de iniciarse con las bombas A y B, y apagarse cuando se detecte nivel alto (H=1).

b) Si de nuevo el sensor indica nivel bajo (H=0), ahora trabajarán las bombas A y C y apagarse cuando se detecte nivel alto (H=1).

c) Si de nuevo el sensor indica nivel bajo (H=0), ahora trabajaran las bombas B y C y apagarse cuando se detecte nivel alto (H=1).

d) Si de nuevo el sensor indica nivel bajo (H=0), ahora trabajaran de nuevo las bombas A y B, y apagarse cuando se detecte nivel alto (H=1). Y así sucesivamente con la finalidad de que las bombas se alternen en su funcionamiento.

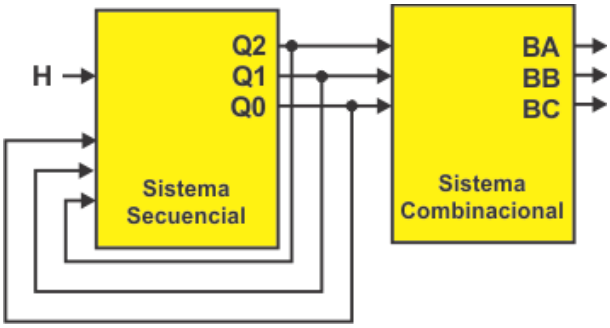
1. **Especificar el sistema**



2.- Tabla de flujo primitiva

H	0	1	BA	BB	BC
0	**E0**	E1	1	1	0
1	E2	**E1**	0	0	0
2	**E2**	E3	1	0	1
3	E4	**E3**	0	0	0
4	**E4**	E5	0	1	1
5	E0	**E5**	0	0	0

Comparando la tabla de flujo primitiva con el problema anterior se puede observar que es el mismo comportamiento en la parte secuencial por lo que solo obtendremos las ecuaciones mínimas de la parte combinacional



Omitiremos los pasos del 3 al 7 del método y utilizando la asignación 2 del problema anterior tenemos:

	Q2	Q1	Q0
a	0	0	0
b	0	1	0
c	0	1	1
d	1	1	1
e	1	0	1
f	1	0	0

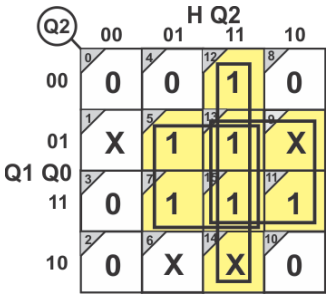
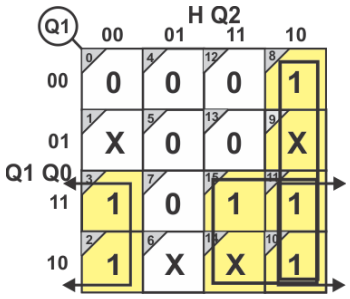
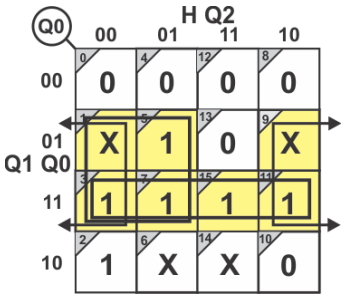
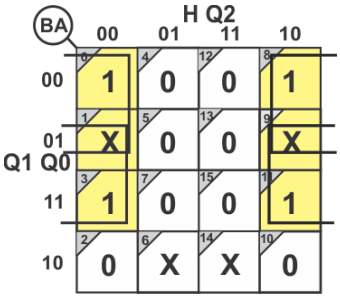
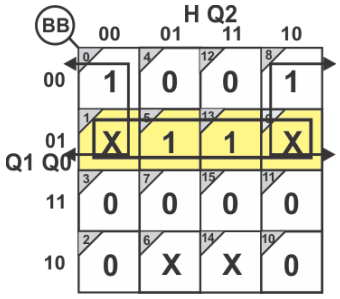
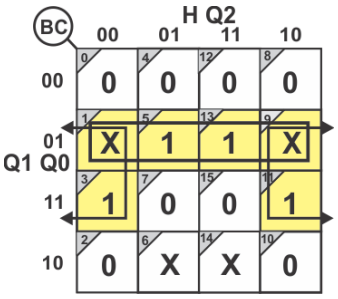
8.- Tabla de Estados Totales.

H	0	1	BA	BB	BC
000	000	010	1	1	0
010	011	010	0	0	0
011	011	111	1	0	1
111	101	111	0	0	0
101	101	100	0	1	1
100	000	100	0	0	0

Tabla de verdad

m	H	Q2	Q1	Q0	Q2+	Q1+	Q0+	BA	BB	BC
0	0	0	0	0	0	0	0	1	1	0
1	0	0	0	1	X	X	X	X	X	X
2	0	0	1	0	0	1	1	0	0	0
3	0	0	1	1	0	1	1	1	0	1
4	0	1	0	0	0	0	0	0	0	0
5	0	1	0	1	1	0	1	0	1	1
6	0	1	1	0	X	X	X	X	X	X
7	0	1	1	1	1	0	1	0	0	0
8	1	0	0	0	0	1	0	1	1	0
9	1	0	0	1	X	X	X	X	X	X
10	1	0	1	0	0	1	0	0	0	0
11	1	0	1	1	1	1	1	1	0	1
12	1	1	0	0	1	0	0	0	0	0
13	1	1	0	1	1	0	0	0	1	1
14	1	1	1	0	X	X	X	X	X	X
15	1	1	1	1	1	1	1	0	0	0

10.- Ecuaciones Mínimas

		
Q2=HQ0+ HQ2+Q2Q0	Q1 =H **Q2'**+HQ1+**Q2'**Q1	Q0= **H'**Q1+**H'**Q0+Q1Q0
		
BA = **Q2'Q1'** +**Q2'**Q0	BB=**Q2'Q1'**+**Q1'**Q0	BC=**Q2'**Q0+**Q1'**Q0

Ecuaciones

$$Q2 = H\,Q0 + H\,Q2 + Q2\,Q0$$

$$Q1 = H\,\mathbf{Q2'} + H\,Q1 + \mathbf{Q2'}Q1$$

$$Q0 = \mathbf{H'}Q1 + \mathbf{H'}Q0 + Q1\,Q0$$

$$BA = \mathbf{Q2'Q1'} + \mathbf{Q2'}Q0$$

$$BB = \mathbf{Q2'Q1'} + \mathbf{Q1'}Q0$$

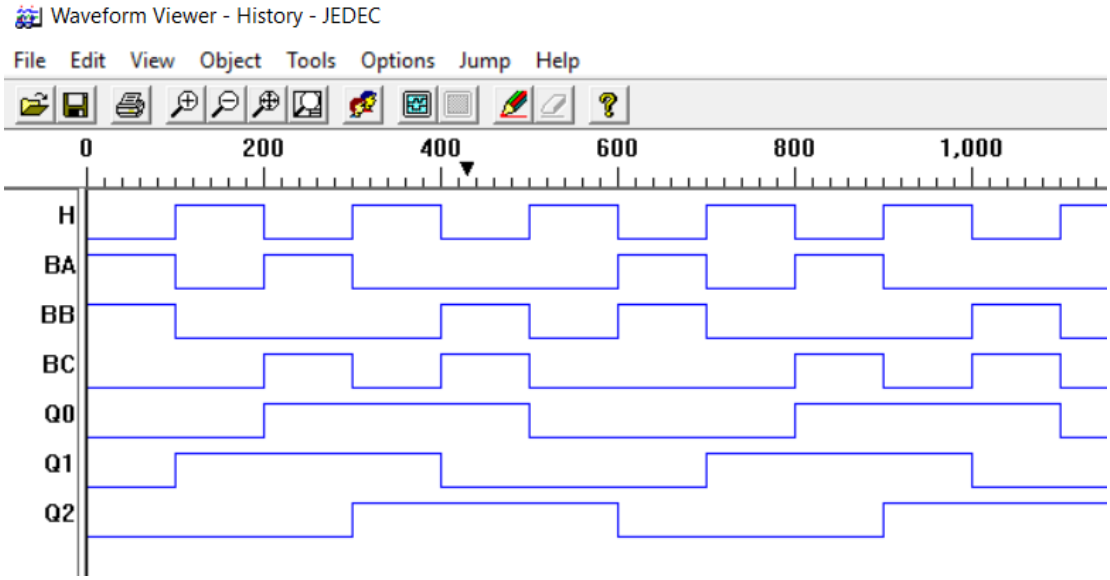
$$BC = \mathbf{Q2'}Q0 + \mathbf{Q1'}Q0$$

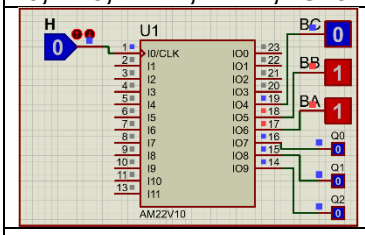
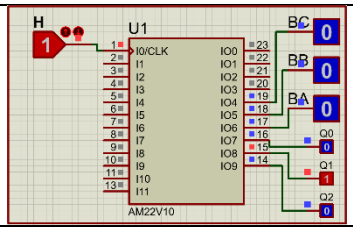
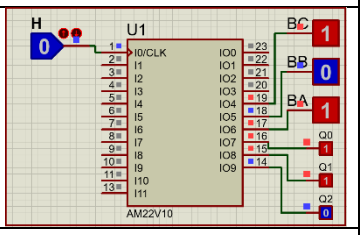
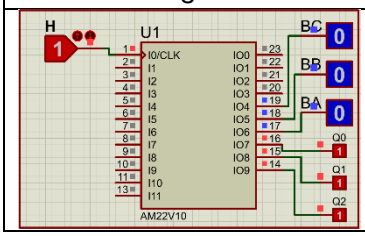
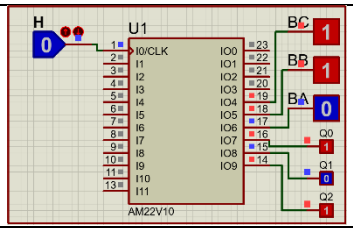
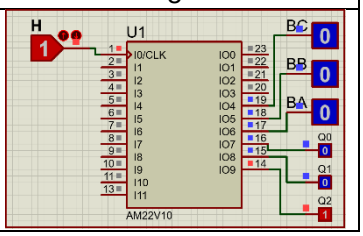
Código ABEL-HDL

```
MODULE pnueve
"2 de tres bombas
"6 mayo 2019
H pin 1;
"Salidas
Q2,Q1,Q0,BA,BB,BC pin 14..19 istype 'com';
equations
Q2=H&Q0#H&Q2#Q0&Q2;
Q1=H&!Q2#H&Q1#!Q2&Q1;
Q0=!H&Q1 #!H&Q0#Q1&Q0;
BA=!Q2&!Q1#!Q2&Q0;
BB =!Q2&!Q1#!Q1&Q0;
BC=!Q2&Q0#!Q1&Q0;
```

```
Test_vectors
(H->[Q2,Q1,Q0,BA,BB,BC])
0->[.x.,.x.,.x.,.x.,.x.,.x.];
1->[.x.,.x.,.x.,.x.,.x.,.x.];
0->[.x.,.x.,.x.,.x.,.x.,.x.];
1->[.x.,.x.,.x.,.x.,.x.,.x.];
0->[.x.,.x.,.x.,.x.,.x.,.x.];
1->[.x.,.x.,.x.,.x.,.x.,.x.];
0->[.x.,.x.,.x.,.x.,.x.,.x.];
1->[.x.,.x.,.x.,.x.,.x.,.x.];
0->[.x.,.x.,.x.,.x.,.x.,.x.];
1->[.x.,.x.,.x.,.x.,.x.,.x.];
0->[.x.,.x.,.x.,.x.,.x.,.x.];
1->[.x.,.x.,.x.,.x.,.x.,.x.];
0->[.x.,.x.,.x.,.x.,.x.,.x.];
1->[.x.,.x.,.x.,.x.,.x.,.x.];
0->[.x.,.x.,.x.,.x.,.x.,.x.];
1->[.x.,.x.,.x.,.x.,.x.,.x.];
0->[.x.,.x.,.x.,.x.,.x.,.x.];
1->[.x.,.x.,.x.,.x.,.x.,.x.];
END
```

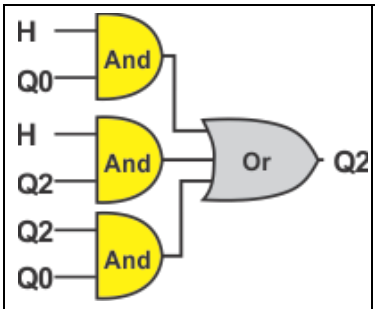
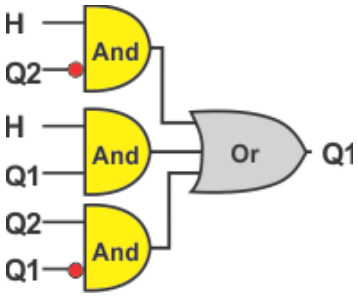
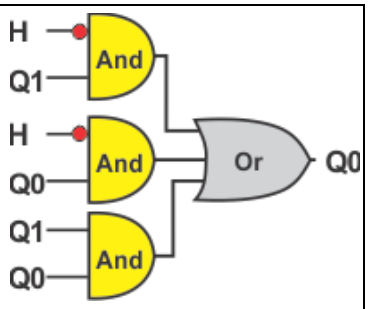
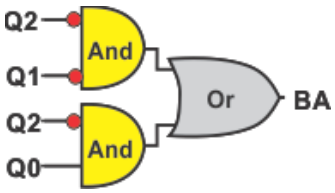
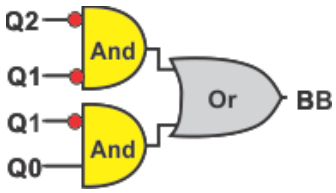
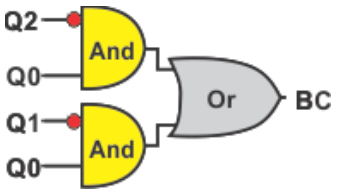
Simulaciones

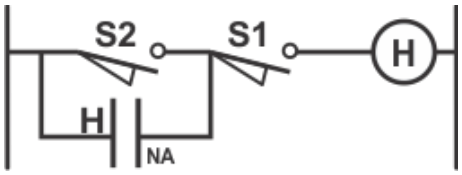


0, H=0, BA=1, BB=1,BC=0	1	2
		
3	4	5
		

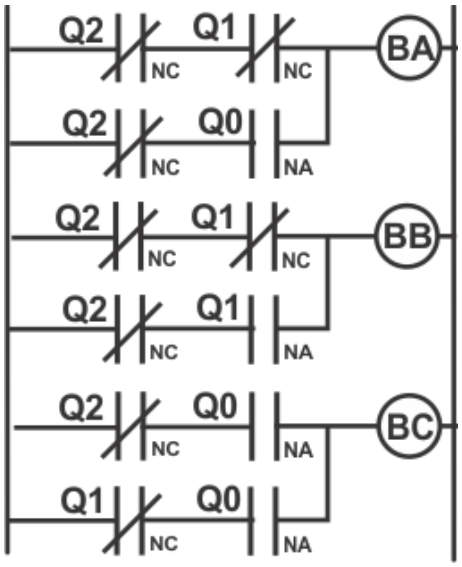
Simulación en PROTEUS mostrando los valores de salidas para cada uno de los seis tiempos del ciclo

13.- Representación Gráfica

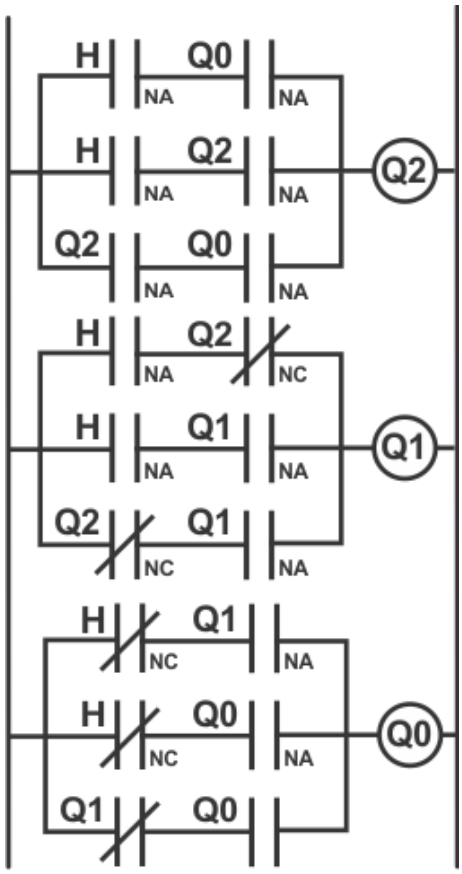
		
Q2=HQ0+ HQ2+Q2Q0	Q1 =H **Q2'**+HQ1+**Q2'**Q1	Q0= **H'**Q1+**H'**Q0+Q1Q0
		
BA = **Q2'Q1'** +**Q2'**Q0	BB=**Q2'Q1'**+**Q1'**Q0	BC=**Q2'**Q0+**Q1'**Q0



Sensor del nivel



Sistema Combinacional

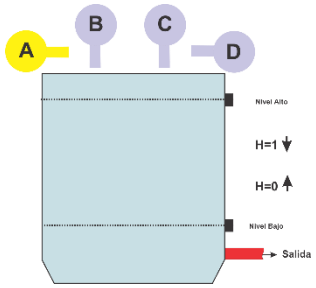


Sistema Secuencial

Problema 7, Llenado de un tanque cuatro bombas 3 de 4

Diseñe, efectúe la simulación y construya un prototipo de un sistema secuencial asíncrono para control de llenado de un tanque por medio de cuatro Bombas llamadas A, B, C y D, el tanque cuenta con un sensor de nivel H, en donde H=0 detecta nivel bajo y H=1 detecta Nivel alto.

El sistema deberá de trabajar con la siguiente lógica:



Tanque con sensores de nivel.

a)Partiendo que el tanque se encuentre vacío (H=0), el llenado deberá de iniciarse con las bombas A, B y C, y apagarse cuando se detecte nivel alto (H=1).

b) Si de nuevo el sensor indica nivel bajo (H=0), ahora trabajaran las bombas A, B y D y apagarse cuando se detecte nivel alto (H=1).

c) Si de nuevo el sensor indica nivel bajo (H=0), trabajaran las bombas A, C y D y apagarse cuando se detecte nivel alto (H=1).

d) Si de nuevo el sensor indica nivel bajo (H=0), trabajaran las bombas B, C y D y apagarse cuando se detecte nivel alto (H=1).

e) Y así sucesivamente alentándose las bombas en su funcionamiento siempre de tres en tres, con la siguiente secuencia ABC, ABD, ACD y BCD, en forma cíclica.

Diagrama de transición

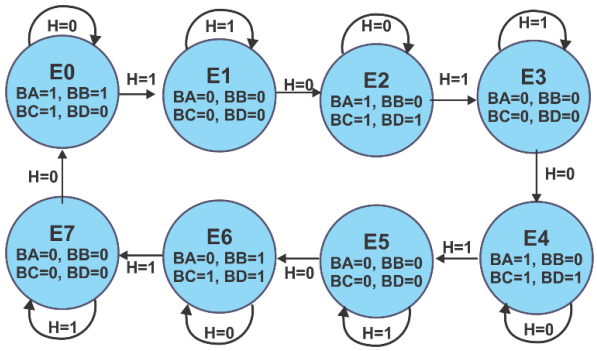


Tabla de flujo primitiva

H	0	1	BA	BB	BC	BD
0	**E0**	E1	1	1	1	0
1	E2	**E1**	0	0	0	0
2	**E2**	E3	1	1	0	1
3	E4	**E3**	0	0	0	0
4	**E4**	E5	1	0	1	1
5	E6	**E5**	0	0	0	0
6	**E6**	E7	0	1	1	1
7	E0	**E7**	0	0	0	0

H	0	1	BA	BB	BC	BD
a	**a**	b	1	1	1	0
b	c	**b**	0	0	0	0
c	**c**	d	1	1	0	1
d	e	**d**	0	0	0	0
e	**e**	f	1	0	1	1
f	g	**f**	0	0	0	0
g	**g**	h	0	1	1	1
h	a	**h**	0	0	0	0

Asignación de valores a los estados:

Mapa de asignaciones	Asignación propuesta

Q2 Q1

Q0	00	01	11	10
0	a	b	e	f
1	h	c	d	g

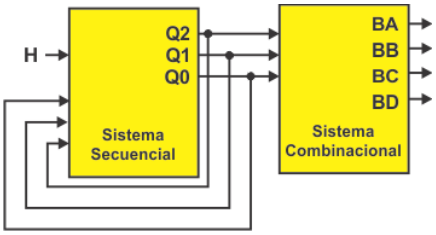
	Q2	Q1	Q0
a	0	0	0
b	0	1	0
c	0	1	1
d	1	1	1
e	1	1	0
f	1	0	0
g	1	0	1
h	0	0	1

H	0	1	BA	BB	BC	BD
000	**000**	010	1	1	1	0
010	011	**010**	0	0	0	0
011	**011**	111	1	1	0	1
111	110	**111**	0	0	0	0
110	**110**	100	1	0	1	1
100	101	**100**	0	0	0	0
101	**101**	001	0	1	1	1
001	000	**001**	0	0	0	0

Tabla de verdad

	H	Q2	Q1	Q0	Q2+	Q1+	Q0+	BA	BB	BC	BD
0	0	0	0	0	0	0	0	1	1	1	0
1	0	0	0	1	0	0	0	0	0	0	0
2	0	0	1	0	0	1	1	0	0	0	0
3	0	0	1	1	0	1	1	1	1	0	1
4	0	1	0	0	1	0	1	0	0	0	0
5	0	1	0	1	1	0	1	0	1	1	1
6	0	1	1	0	1	1	0	1	0	1	1
7	0	1	1	1	1	1	0	0	0	0	0
8	1	0	0	0	0	1	0	1	1	1	0
9	1	0	0	1	0	0	1	0	0	0	0
10	1	0	1	0	0	1	0	0	0	0	0
11	1	0	1	1	1	1	1	1	1	0	1
12	1	1	0	0	1	0	0	0	0	0	0
13	1	1	0	1	0	0	1	0	1	1	1
14	1	1	1	0	1	0	0	1	0	1	1
15	1	1	1	1	1	1	1	0	0	0	0

Diagrama de la máquina de Moore



De modo que el sistema combinacional estará definido por la siguiente tabla de verdad y que las salidas BA, BB, BC Y BD dependerán solo de las Q2, Q1 y Q0.

	Q2	Q1	Q0	BA	BB	BC	BD
0	0	0	0	1	1	1	0
1	0	0	1	0	0	0	0
2	0	1	0	0	0	0	0
3	0	1	1	1	1	0	1
4	1	0	0	0	0	0	0
5	1	0	1	0	1	1	1
6	1	1	0	1	0	1	1
7	1	1	1	0	0	0	0

10.- Ecuaciones Mínimas: utilizaremos en el código ABEL-HDL el comando Truth_Table, describiendo ambas tablas y de ahí se obtienen las ecuaciones mínimas.

```
MODULE pdiez
"3 de 4 bombas
"9 de Mayo 2019
"Entradas
H pin 1;
"Salidas
Q2,Q1,Q0 pin 14..16 istype 'com';
BA,BB,BC,BD pin 19..22 istype 'com';
Truth_Table
([H,Q2,Q1,Q0]->[Q2,Q1,Q0])
[0,0,0,0]->[0,0,0];
[0,0,0,1]->[0,0,0];
[0,0,1,0]->[0,1,1];
[0,0,1,1]->[0,1,1];
[0,1,0,0]->[1,0,1];
[0,1,0,1]->[1,0,1];
[0,1,1,0]->[1,1,0];
[0,1,1,1]->[1,1,0];
[1,0,0,0]->[0,1,0];
[1,0,0,1]->[0,0,1];
[1,0,1,0]->[0,1,0];
[1,0,1,1]->[1,1,1];
[1,1,0,0]->[1,0,0];
[1,1,0,1]->[0,0,1];
[1,1,1,0]->[1,0,0];
[1,1,1,1]->[1,1,1];
```

```
Truth_Table
([Q2,Q1,Q0]->[BA,BB,BC,BD])
[0,0,0]->[1,1,1,0];
[0,0,1]->[0,0,0,0];
[0,1,0]->[0,0,0,0];
[0,1,1]->[1,1,0,1];
[1,0,0]->[0,0,0,0];
[1,0,1]->[0,1,1,1];
[1,1,0]->[1,0,1,1];
[1,1,1]->[0,0,0,0];
Test_vectors
(H->[Q2,Q1,Q0,BA,BB,BC,BD])
0->[.x.,.x.,.x.,.x.,.x.,.x.,.X.];
1->[.x.,.x.,.x.,.x.,.x.,.x.,.X.];
0->[.x.,.x.,.x.,.x.,.x.,.x.,.X.];
1->[.x.,.x.,.x.,.x.,.x.,.x.,.X.];
0->[.x.,.x.,.x.,.x.,.x.,.x.,.X.];
1->[.x.,.x.,.x.,.x.,.x.,.x.,.X.];
0->[.x.,.x.,.x.,.x.,.x.,.x.,.X.];
1->[.x.,.x.,.x.,.x.,.x.,.x.,.X.];
0->[.x.,.x.,.x.,.x.,.x.,.x.,.X.];
1->[.x.,.x.,.x.,.x.,.x.,.x.,.X.];
0->[.x.,.x.,.x.,.x.,.x.,.x.,.X.];
1->[.x.,.x.,.x.,.x.,.x.,.x.,.X.];
END
```

En el archivo Chip report que genera el programa IspLever se obtiene las ecuaciones mínimas descritas en el leguaje ABEL-HDL, que se muestran a continuación**:**

Q2 =(**!Q0** & Q2 # Q0 & Q1 & H # Q2 & !H);

Q1 = (Q0 & Q1 # **!Q0** & **!Q2** & H # Q1 & !H);

Q0 = (Q0 & H # **!Q1** & Q2 & **!H** # Q1 & **!Q2** & **!H**);

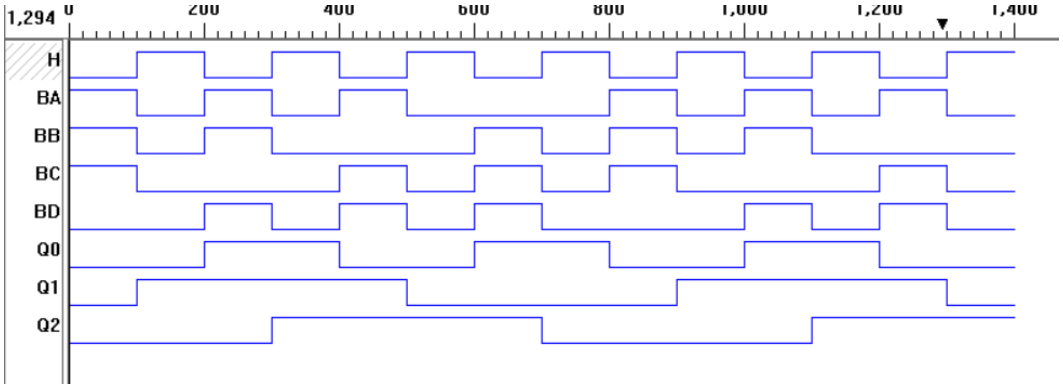
BA = (**!Q0** & Q1 & Q2 # Q0 & Q1 & !Q2 # **!Q0** & **!Q1** & **!Q2**);

BB = (Q0 & **!Q1** & Q2 # Q0 & Q1 & **!Q2** # **!Q0** & **!Q1** & **!Q2**);

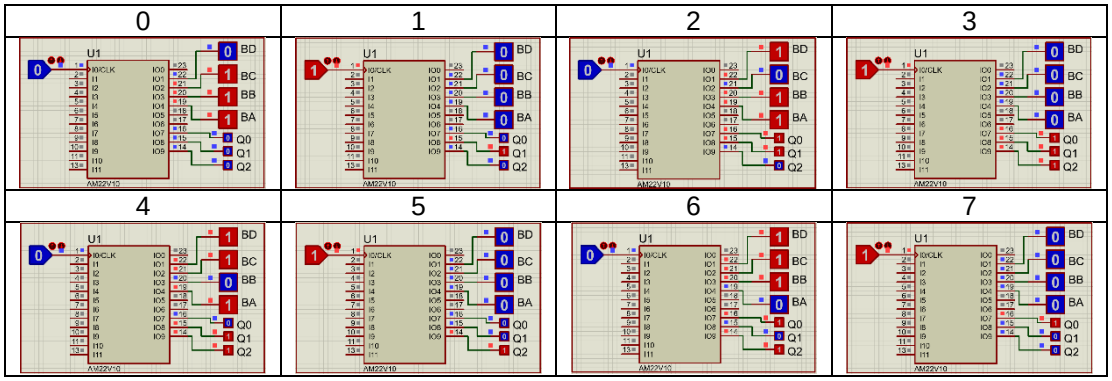
BC = (**!Q0** & Q1 & Q2 # Q0 & **!Q1** & Q2 # **!Q0** & **!Q1** & **!Q2**);

BD = (**!Q0** & Q1 & Q2 # Q0 & **!Q1** & Q2 # Q0 & Q1 & **!Q2**);

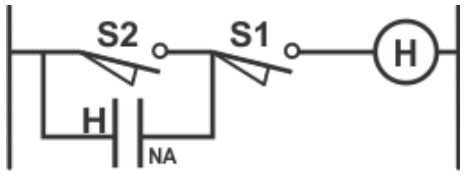
Simulación en Test_vectors mostrando cada uno de los ocho tiempos del ciclo



Simulación en PROTEUS mostrando los valores de salidas para cada uno de los ocho tiempos del ciclo.



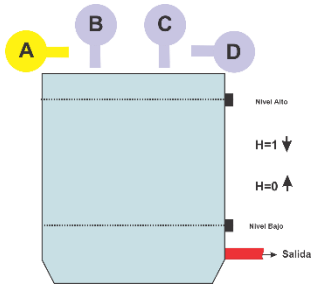
13.- Representación Gráfica

	
Sensor del nivel	
Sistema Combinacional	Sistema Secuencial

Problema 8, Llenado de un tanque cuatro bombas 1 de 4

Diseñe, efectúe la simulación y construya un prototipo de un sistema secuencial asíncrono para control de llenado de un tanque por medio de cuatro Bombas llamadas A, B, C y D, el tanque cuenta con un sensor de nivel H, en donde H=0 detecta nivel bajo y H=1 detecta Nivel alto (tanque lleno).

El sistema deberá de trabajar con la siguiente lógica:



Tanque con sensores de nivel.

a)Partiendo que el tanque se encuentre vacío (H=0), el llenado deberá de iniciarse solo con la bomba A y apagarse cuando se detecte nivel alto (H=1).

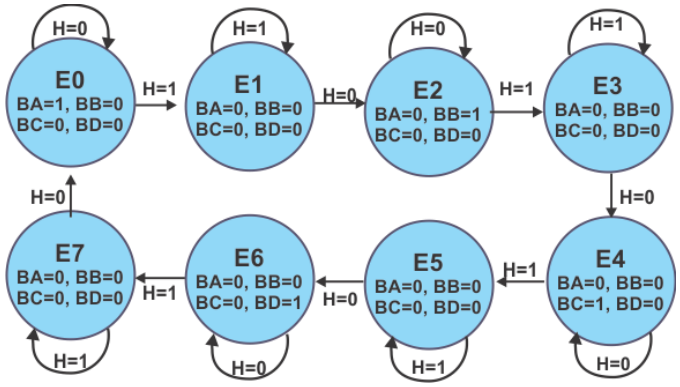
b) Si de nuevo el sensor indica nivel bajo (H=0), ahora trabajará la bomba B y apagarse cuando se detecte nivel alto (H=1).

c) Si de nuevo el sensor indica nivel bajo (H=0), trabajará solo la bomba C y apagarse cuando se detecte nivel alto (H=1).

d) Si de nuevo el sensor indica nivel bajo (H=0), trabajará solo la bomba D y apagarse cuando se detecte nivel alto (H=1).

e) Si de nuevo llega a nivel bajo, trabajara ahora la bomba A y apagarse cuando se llene o nivel alto, y así sucesivamente alentándose las bombas en su funcionamiento siempre de una en una, con la secuencia A, B, C y D, en forma cíclica.

1.-Diagrama de Transición



2.-Tabla de flujo primitiva

H	0	1	BA	BB	BC	BD
0	**E0**	E1	1	0	0	0
1	E2	**E1**	0	0	0	0
2	**E2**	E3	0	1	0	0
3	E4	**E3**	0	0	0	0
4	**E4**	E5	0	0	1	0
5	E6	**E5**	0	0	0	0
6	**E6**	E7	0	0	0	1
7	E0	**E7**	0	0	0	0

3.-Tabla de estados internos

H	0	1	BA	BB	BC	BD
a	**a**	b	1	0	0	0
b	c	**b**	0	0	0	0
c	**c**	d	0	1	0	0
d	e	**d**	0	0	0	0
e	**e**	f	0	0	1	0
f	g	**f**	0	0	0	0
g	**g**	h	0	0	0	1
h	a	**h**	0	0	0	0

En la parte secuencial de este ejemplo, si conservamos la misma asignación de valores a los estados que el problema 10 solo cambia el sistema combinacional.

Mapa de asignaciones	Asignación propuesta

Q2 Q1

Q0	00	01	11	10
0	a	b	e	f
1	h	c	d	g

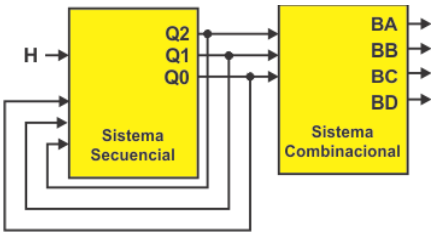
	Q2	Q1	Q0
a	0	0	0
b	0	1	0
c	0	1	1
d	1	1	1
e	1	1	0
f	1	0	0
g	1	0	1
h	0	0	1

H	0	1	BA	BB	BC	BD
000	**000**	010	**1**	**0**	**0**	**0**
010	011	**010**	**0**	**0**	**0**	**0**
011	**011**	111	**0**	**1**	**0**	**0**
111	110	**111**	**0**	**0**	**0**	**0**
110	**110**	100	**0**	**0**	**1**	**0**
100	101	**100**	**0**	**0**	**0**	**0**
101	**101**	001	**0**	**0**	**0**	**1**
001	000	**001**	**0**	**0**	**0**	**0**

Tabla de verdad

	H	Q2	Q1	Q0	Q2+	Q1+	Q0+	BA	BB	BC	BD
0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	1	0	0	0	0	0	0	0
2	0	0	1	0	0	1	1	0	0	0	0
3	0	0	1	1	0	1	1	0	1	0	0
4	0	1	0	0	1	0	1	0	0	0	0
5	0	1	0	1	1	0	1	0	0	0	1
6	0	1	1	0	1	1	0	0	1	0	0
7	0	1	1	1	1	1	0	0	0	0	0
8	1	0	0	0	0	1	0	1	0	0	0
9	1	0	0	1	0	0	1	0	0	0	0
10	1	0	1	0	0	1	0	0	0	0	0
11	1	0	1	1	1	1	1	0	1	0	0
12	1	1	0	0	1	0	0	0	0	0	0
13	1	1	0	1	0	0	1	0	0	0	1
14	1	1	1	0	1	0	0	0	1	0	0
15	1	1	1	1	1	1	1	0	0	0	0

Diagrama de la máquina de Moore



De modo que el sistema combinacional estará definido por la siguiente tabla de verdad.

	Q2	Q1	Q0	BA	BB	BC	BD
0	0	0	0	1	0	0	0
1	0	0	1	0	0	0	0
2	0	1	0	0	0	0	0
3	0	1	1	0	1	0	0
4	1	0	0	0	0	0	0
5	1	0	1	0	0	0	1
6	1	1	0	0	1	0	0
7	1	1	1	0	0	0	0

10.- Ecuaciones Mínimas: utilizaremos en el código ABEL-HDL el comando Truth_Table, describiendo ambas tablas y de ahí se obtienen las ecuaciones mínimas.

```
MODULE ponce
"9 de Mayo 2019
"Entradas
H pin 1;
"Salidas
Q2,Q1,Q0 pin 14..16 istype 'com';
BA,BB,BC,BD pin 19..22 istype 'com';
Truth_Table
([H,Q2,Q1,Q0]->[Q2,Q1,Q0])
[0,0,0,0]->[0,0,0];
[0,0,0,1]->[0,0,0];
[0,0,1,0]->[0,1,1];
[0,0,1,1]->[0,1,1];
[0,1,0,0]->[1,0,1];
[0,1,0,1]->[1,0,1];
[0,1,1,0]->[1,1,0];
[0,1,1,1]->[1,1,0];
[1,0,0,0]->[0,1,0];
[1,0,0,1]->[0,0,1];
[1,0,1,0]->[0,1,0];
[1,0,1,1]->[1,1,1];
[1,1,0,0]->[1,0,0];
[1,1,0,1]->[0,0,1];
[1,1,1,0]->[1,0,0];
[1,1,1,1]->[1,1,1];
```

```
Truth_Table
([Q2,Q1,Q0]->[BA,BB,BC,BD])
[0,0,0]->[1,0,0,0];
[0,0,1]->[0,0,0,0];
[0,1,0]->[0,0,0,0];
[0,1,1]->[0,1,0,0];
[1,0,0]->[0,0,0,0];
[1,0,1]->[0,0,0,1];
[1,1,0]->[0,0,1,0];
[1,1,1]->[0,0,0,0];

Test_vectors
(H->[Q2,Q1,Q0,BA,BB,BC,BD])
0->[.x.,.x.,.x.,.x.,.x.,.x.,.X.];
1->[.x.,.x.,.x.,.x.,.x.,.x.,.X.];
0->[.x.,.x.,.x.,.x.,.x.,.x.,.X.];
1->[.x.,.x.,.x.,.x.,.x.,.x.,.X.];
0->[.x.,.x.,.x.,.x.,.x.,.x.,.X.];
1->[.x.,.x.,.x.,.x.,.x.,.x.,.X.];
0->[.x.,.x.,.x.,.x.,.x.,.x.,.X.];
1->[.x.,.x.,.x.,.x.,.x.,.x.,.X.];
0->[.x.,.x.,.x.,.x.,.x.,.x.,.X.];
1->[.x.,.x.,.x.,.x.,.x.,.x.,.X.];
0->[.x.,.x.,.x.,.x.,.x.,.x.,.X.];
1->[.x.,.x.,.x.,.x.,.x.,.x.,.X.];
0->[.x.,.x.,.x.,.x.,.x.,.x.,.X.];
1->[.x.,.x.,.x.,.x.,.x.,.x.,.X.];
END
```

En el archivo Chip report que genera el programa IspLever se obtiene las ecuaciones mínimas descritas en el leguaje ABEL-HDL, que se muestran a continuación**:**

Q2 =(**!Q0** & Q2 # Q0 & Q1 & H # Q2 & !H);

Q1 =(Q0 & Q1 # **!Q0** & **!Q2** & H # Q1 & !H);

Q0 = (Q0 & H # **!Q1** & Q2 & **!H** # Q1 & **!Q2** & **!H**);

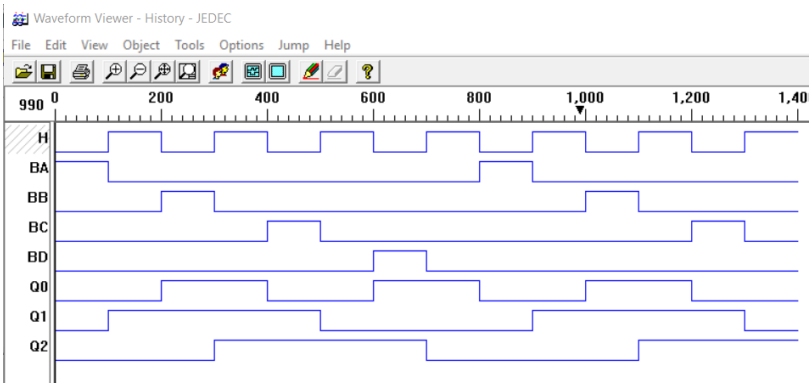
BA= (**!Q0** & **!Q1** & **!Q2**);

BB = (Q0 & Q1 & **!Q2**);

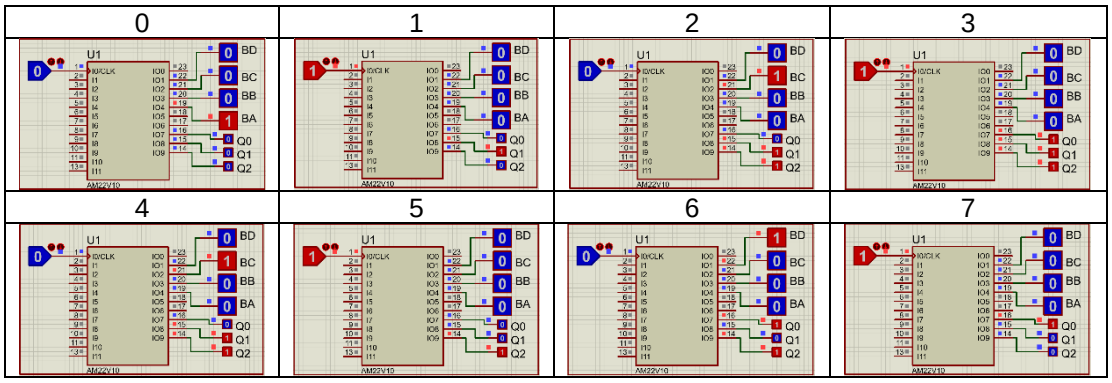
BC = (**!Q0** & Q1 & Q2);

BD = (Q0 & **!Q1** & Q2);

Simulación en Test_vectors mostrando cada uno de los ocho tiempos del ciclo

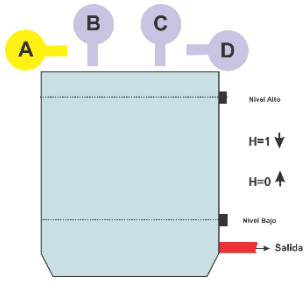


Simulación en PROTEUS mostrando los valores de salidas para cada uno de los ocho tiempos del ciclo.



Problema 9, llenado de un tanque cuatro bombas 2 de 4

Diseñe, efectúe la simulación y construya un prototipo de un sistema secuencial asíncrono para control de llenado de un tanque por medio de cuatro Bombas llamadas A, B, C y D, el tanque cuenta con un sensor de nivel H, en donde H=0 detecta nivel bajo y H=1 detecta Nivel alto.



Tanque con sensores de nivel.

El sistema deberá de trabajar con la siguiente lógica:

f) Partiendo que el tanque se encuentre vacío (H=0), el llenado deberá de iniciarse con las bombas A y B, y apagarse cuando se detecte nivel alto (H=1).

g)Si de nuevo el sensor indica nivel bajo (H=0), ahora trabajaran las bombas A y C y apagarse cuando se detecte nivel alto (H=1).

h)Si de nuevo el sensor indica nivel bajo (H=0), trabajaran las bombas A y D y apagarse cuando se detecte nivel alto (H=1).

i)Si de nuevo el sensor indica nivel bajo (H=0), trabajaran las bombas B y C y apagarse cuando se detecte nivel alto (H=1).

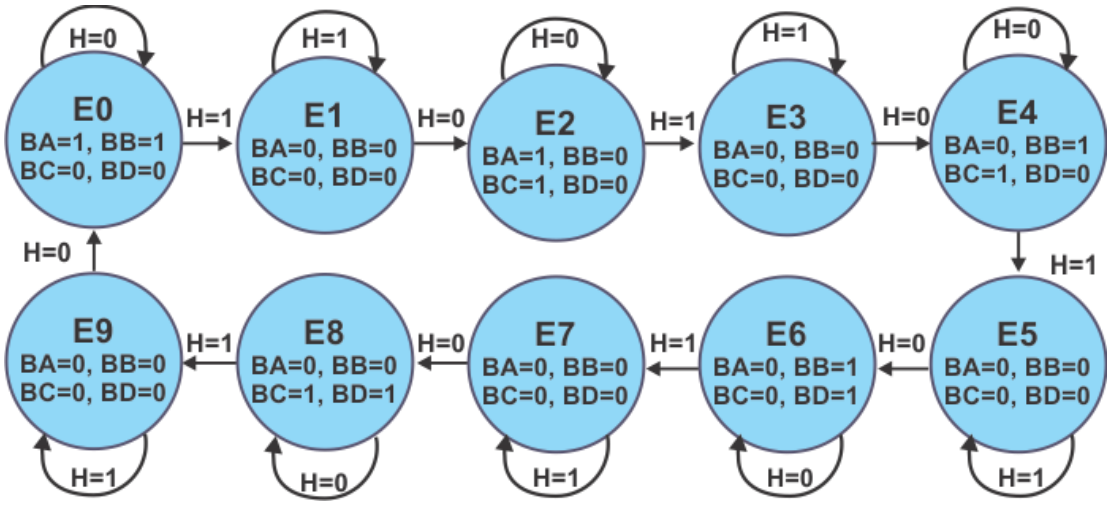
j)Si de nuevo el sensor indica nivel bajo (H=0), trabajaran las bombas B y D y apagarse cuando se detecte nivel alto (H=1).

k)Si de nuevo el sensor indica nivel bajo (H=0), trabajaran las bombas C y D y apagarse cuando se detecte nivel alto (H=1).

l)Si de nuevo el sensor indica nivel bajo (H=0), trabajaran de nuevo las bombas AB y apagarse cuando se detecte nivel alto (H=1).

m) Y así sucesivamente alentándose las bombas en su funcionamiento siempre de dos en dos, con la siguiente secuencia AB, AC, AD, BC, BD y CD en forma cíclica.

1.- Especificar el Sistema.



2 Tabla de flujo primitiva

H	0	1	BA	BB	BC	BD
0	**E0**	E1	1	1	0	0
1	E2	**E1**	0	0	0	0
2	**E2**	E3	1	0	1	0
3	E4	**E3**	0	0	0	0
4	**E4**	E5	1	0	0	1
5	E6	**E5**	0	0	0	0
6	**E6**	E7	0	1	1	0
7	E8	**E7**	0	0	0	0
8	**E8**	E9	0	1	0	1
9	E10	**E9**	0	0	0	0
10	**E10**	E11	0	0	1	1
11	E0	**E11**	0	0	0	0

6.- Tabla de Estados Internos

H	0	1	BA	BB	BC	BD
a	**a**	b	1	1	0	0
b	c	**b**	0	0	0	0
c	**c**	d	1	0	1	0
d	e	**d**	0	0	0	0
e	**e**	f	1	0	0	1
f	g	**f**	0	0	0	0
g	**g**	h	0	1	1	0
h	i	**h**	0	0	0	0
i	**i**	j	0	1	0	1
j	k	**j**	0	0	0	0
k	**k**	l	0	0	1	1
l	a	**l**	0	0	0	0

7.- Asignación de valores a los estados

Q1 Q0 \ Q3 Q2	00	01	11	10
00	a	h	i	l
01	b	g	j	k
11	c	f		
10	d	e		

Asignación resultante

	Q3	Q2	Q1	Q0	V_{10}
a	0	0	0	0	0
b	0	0	0	1	1
c	0	0	1	1	3
d	0	0	1	0	2
e	0	1	1	0	6
f	0	1	1	1	7
g	0	1	0	1	5
h	0	1	0	0	4
i	1	1	0	0	12
j	1	1	0	1	13
k	1	0	0	1	9
l	1	0	0	0	8

Nota: Las combinaciones 10, 11, 14 y 15 no son utilizadas en la asignación

8.- Tabla de Estados Totales

H	0	1	BA	BB	BC	BD
0000	**0000**	001	1	1	0	0
0001	011	**0001**	0	0	0	0
011	**0011**	0010	1	0	1	0
0010	0110	**0010**	0	0	0	0
0110	**0110**	0111	1	0	0	1
0111	0101	**0111**	0	0	0	0
0101	**0101**	0100	0	1	1	0
0100	1100	**0100**	0	0	0	0
1100	**1100**	1101	0	1	0	1
1101	1001	**1101**	0	0	0	0
1001	**1001**	1000	0	0	1	1
1000	0000	**1000**	0	0	0	0

9.- Completar tabla de salidas: no es necesario

10.- Obtención de las ecuaciones mínimas

Tabla de verdad

	H	Q3	Q2	Q1	Q0	Q3+	Q2+	Q1+	Q0+	BA	BB	BC	BD
0	0	0	0	0	0	0	0	0	0	**1**	**1**	**0**	**0**
1	0	0	0	0	1	0	0	1	1	**0**	**0**	**0**	**0**
2	0	0	0	1	0	0	1	1	0	**0**	**0**	**0**	**0**
3	0	0	0	1	1	0	0	1	1	**1**	**0**	**1**	**0**
4	0	0	1	0	0	1	1	0	0	**0**	**0**	**0**	**0**
5	0	0	1	0	1	0	1	0	1	**0**	**1**	**1**	**0**
6	0	0	1	1	0	0	1	1	0	**1**	**0**	**0**	**1**
7	0	0	1	1	1	0	1	0	1	**0**	**0**	**0**	**0**
8	0	1	0	0	0	0	0	0	0	**0**	**0**	**0**	**0**
9	0	1	0	0	1	1	0	0	1	**0**	**0**	**1**	**1**
10	0	1	0	1	0	0	0	1	0	**1**	**0**	**1**	**0**
11	0	1	0	1	1	X	X	X	X	X	X	X	X
12	0	1	1	0	0	1	1	0	0	**0**	**1**	**0**	**1**
13	0	1	1	0	1	1	0	0	1	**0**	**0**	**0**	**0**
14	0	1	1	1	0	X	X	X	X	X	X	X	X
15	0	1	1	1	1	X	X	X	X	X	X	X	X
16	1	0	0	0	0	0	0	0	1	**1**	**1**	**0**	**0**
17	1	0	0	0	1	0	0	0	1	**0**	**0**	**0**	**0**
18	1	0	0	1	0	0	0	1	0	**0**	**0**	**0**	**0**
19	1	0	0	1	1	X	X	X	X	X	X	X	X
20	1	0	1	0	0	0	1	0	0	**0**	**0**	**0**	**0**
21	1	0	1	0	1	0	1	0	0	**0**	**1**	**1**	**0**
22	1	0	1	1	0	0	1	1	1	**1**	**0**	**0**	**1**
23	1	0	1	1	1	0	1	1	1	**0**	**0**	**0**	**0**
24	1	1	0	0	0	1	0	0	0	**0**	**0**	**0**	**0**
25	1	1	0	0	1	1	0	0	0	**0**	**0**	**1**	**1**
26	1	1	0	1	0	X	X	X	X	X	X	X	X
27	1	1	0	1	1	X	X	X	X	X	X	X	X
28	1	1	1	0	0	1	1	0	1	**0**	**1**	**0**	**1**
29	1	1	1	0	1	1	1	0	1	**0**	**0**	**0**	**0**
30	1	1	1	1	0	X	X	X	X	X	X	X	X
31	1	1	1	1	1	X	X	X	X	X	X	X	X

Ecuaciones mínimas:

Q3=H Q3 + Q3 Q0 + **H'**Q2 **Q1'Q0'**

Q2=H Q2 + Q2 **Q0'** + **Q3'**Q2 + **H'Q3'**Q1 **Q0'**

Q1= H Q1 + Q1 **Q0'** + **H'Q3'Q2'**Q0

Q0=H Q3 Q2 + H Q2 Q1 + H **Q3'Q2'Q1'** + **H'**Q0

BA= Q2 Q1 **Q0'** + **Q3'Q2'Q1'Q0'** + Q3 Q1 + **Q2'**Q1 Q0

BB = Q3 Q2 **Q0'** + **Q3'**Q2 **Q1'**Q0 + **Q3'Q2'Q1'Q0'**

BC = Q3 **Q2'**Q0 + **Q3'**Q2 **Q1'**Q0 + Q3 Q1 + **Q2'**Q1 Q0

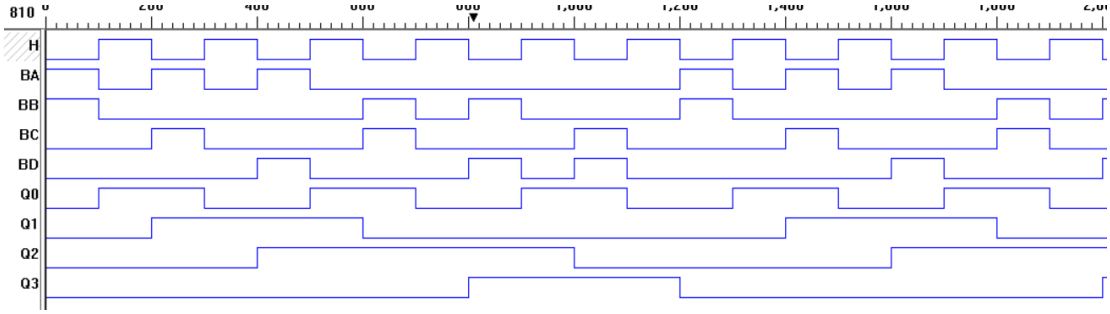
BD = Q3 Q2 **Q0'** + Q3 **Q2'**Q0 + Q2 Q1 **Q0'**

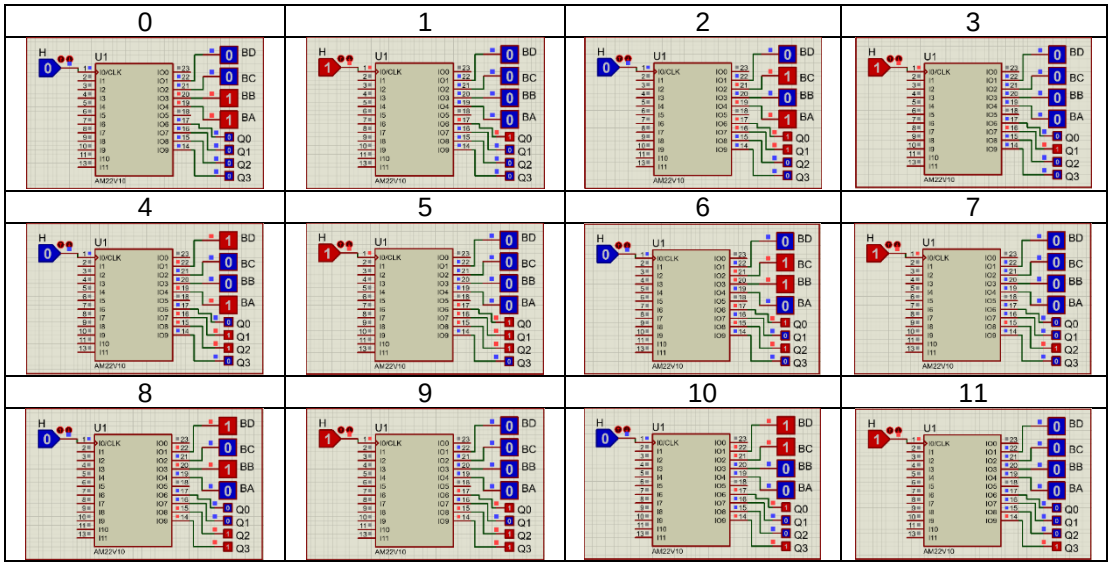
11.- Código ABEL-HDL

```
MODULE pnueve
"1 de Mayo 2019
"Entradas
H pin 1;
"Salidas
Q3,Q2,Q1,Q0,BA,BB,BC,BD pin 14..17,19.. istype 'com';
equations
Q3=H&Q3#Q3&Q0#!H&Q2&!Q1&!Q0;
Q2=H&Q2#Q2&!Q0#!Q3&Q2#!H&!Q3&Q1&!Q0;
Q1=H&Q1#Q1&!Q0#!H&!Q3&!Q2&Q0;
Q0=H&Q3&Q2#H&Q2&Q1#H&!Q3&!Q2&!Q1#!H&Q0;
"BA=  Q2 Q1 Q0' + Q3'Q2'Q1'Q0' + Q3 Q1 + Q2'Q1 Q0
BA=Q2&Q1&!Q0#!Q3&!Q2&!Q1&!Q0#Q3&Q1#!Q2&Q1&Q0;
BB=Q3&Q2&!Q0#!Q3&Q2&!Q1&Q0#!Q3&!Q2&!Q1&!Q0;
BC=Q3&!Q2&Q0#!Q3&!Q2&!Q1&Q0#Q3&Q1#!Q2&Q1&Q0;
BD=Q3&Q2&!Q0#Q3&!Q2&Q0#Q2&Q1&!Q0;
Test_vectors
(H->[Q3,Q2,Q1,Q0,BA,BB,BC,BD])
0->[.x.,.x.,.x.,.x.,.x.,.x.,.x.,.X.];
1->[.x.,.x.,.x.,.x.,.x.,.x.,.x.,.X.];
0->[.x.,.x.,.x.,.x.,.x.,.x.,.x.,.X.];
1->[.x.,.x.,.x.,.x.,.x.,.x.,.x.,.X.];
0->[.x.,.x.,.x.,.x.,.x.,.x.,.x.,.X.];
1->[.x.,.x.,.x.,.x.,.x.,.x.,.x.,.X.];
0->[.x.,.x.,.x.,.x.,.x.,.x.,.x.,.X.];
1->[.x.,.x.,.x.,.x.,.x.,.x.,.x.,.X.];
0->[.x.,.x.,.x.,.x.,.x.,.x.,.x.,.X.];
1->[.x.,.x.,.x.,.x.,.x.,.x.,.x.,.X.];
0->[.x.,.x.,.x.,.x.,.x.,.x.,.x.,.X.];
```

1->[.x.,.x.,.x.,.x.,.x.,.x.,.X.];
0->[.x.,.x.,.x.,.x.,.x.,.x.,.X.];
1->[.x.,.x.,.x.,.x.,.x.,.x.,.X.];
0->[.x.,.x.,.x.,.x.,.x.,.x.,.X.];
1->[.x.,.x.,.x.,.x.,.x.,.x.,.X.];
END

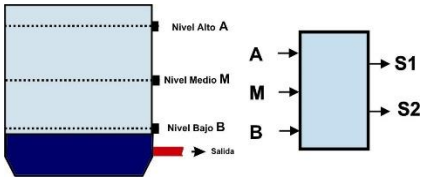
12.- Simulaciones





Problema 10, Detector de tres niveles en un tanque.

Diseñe un sistema secuencial asíncrono para la detección del nivel de un tanque que por medio de tres sensores llamados **B** bajo, **M** medio, **A** alto se obtengan las salidas S1 y S2 con los valores presentados en la siguiente gráfica.



descripción del tanque y sus sensores de nivel

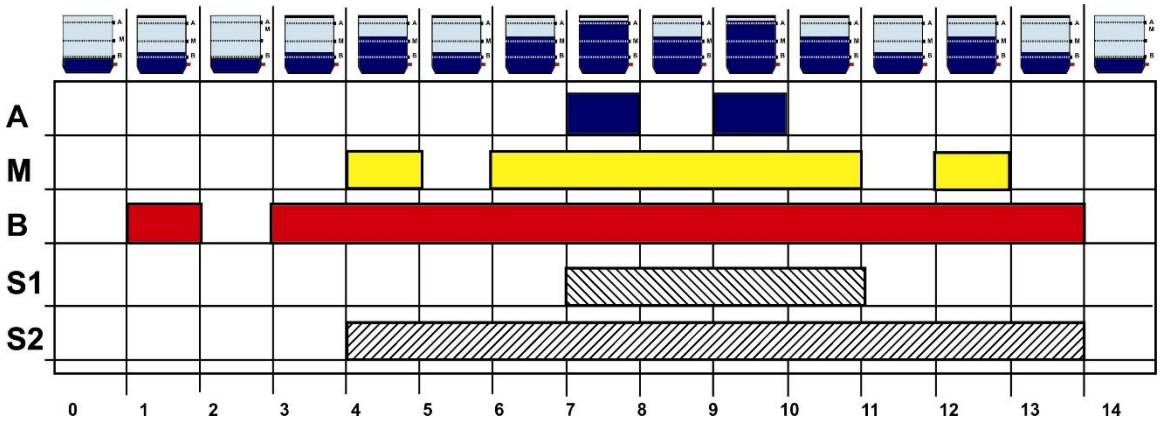
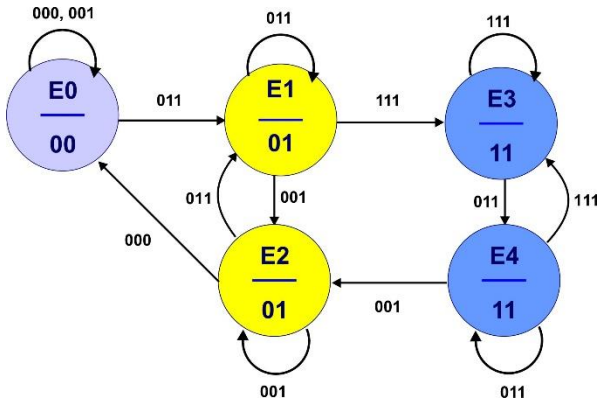


Diagrama de tiempos

1.- Especificar el Sistema.

Del comportamiento descrito en el diagrama de tiempos proponemos el siguiente diagrama de transición:



2.- Tabla de flujo primitiva

	Entradas AMB								Salidas	
	000	**001**	**010**	**011**	**100**	**101**	**110**	**111**	**S1**	**S2**
0	**E0**	**E0**	-	E1	-	-	-	-	**0**	**0**
1	-	E2	-	**E1**	-	-	-	E3	**0**	**1**
2	E0	**E2**	-	E1	-	-	-	-	**0**	**1**
3	-	-	-	E4	-	-	-	**E3**	**1**	**1**
4	-	E2		**E4**				E3	**1**	**1**

3.- Reducción de Estados.

No es posible la reducción de estados.

4.- Mezcla de Filas.

Mezclando las filas 1 con 2 y 3 con 4 obtenemos

	Entradas AMB								Salidas	
	000	**001**	**010**	**011**	**100**	**101**	**110**	**111**	**S1**	**S2**
0	**E0**	**E0**	-	E1	-	-	-	-	**0**	**0**
1, 2	E0	**E2**	-	**E1**	-	-	-	E3	**0**	**1**
3, 4	-	E2	-	**E4**	-	-	-	**E3**	**1**	**1**

Como las salidas son las mismas en cada fila no es necesario expandir la tabla de salida.

5.- Expandir Tabla de Salidas.

No es necesario expandir la tabla de salidas. **No es necesario expandir salidas ya que Q1 = S1 y Q2 = S2.**

6.- Estados Internos.

Sustituyendo E0 por a, E1 y E2 por b y E3 y E4 por c obtenemos:

	Entradas AMB								Salidas	
	000	**001**	**010**	**011**	**100**	**101**	**110**	**111**	**S1**	**S2**
a	**a**	**a**	-	b	-	-	-	-	**0**	**0**
b	a	**b**	-	**b**	-	-	-	c	**0**	**1**
c	-	b	-	**c**	-	-	-	**c**	**1**	**1**

7.- Asignación de Valores a los Estados.

Asignación de valores a los Estados.

Requisitos: (a $\rightarrow$ b), (b $\rightarrow$ c), (c $\rightarrow$ b)

		Q2 = 0	Q2 = 1
Q1	0	**a**	
	1	**b**	**c**

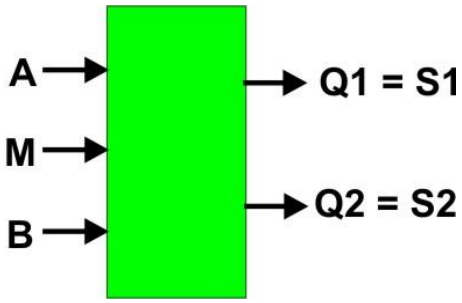
Asignación: a =00, b = 01, c = 11

8.- Tabla de Estados.

	Entradas AMB								Salidas	
Q1 Q2	**000**	**001**	**010**	**011**	**100**	**101**	**110**	**111**	**S1**	**S2**
00	**00**	**00**	-	01	-	-	-	-	**0**	**0**
01	00	**01**	-	**01**	-	-	-	11	**0**	**1**
11	-	01	-	**11**	-	-	-	**11**	**1**	**1**
10	-	-	-	**-**	-	-	-	**-**	**-**	**-**

9.- Completar Tabla de Salidas.

No es necesario completar tabla de salidas.



10.- Obtención de las ecuaciones por medio de minimización.

	Entradas					Salidas	
m	A	M	B	Q1	Q2	Q1+	Q2+
0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0
2,3	0	0	0	0	X	X	X
4	0	0	1	0	0	0	0
5	0	0	1	0	1	0	1
6	0	0	1	1	0	X	X
7	0	0	1	1	1	0	1
8 a 11	0	1	0	X	X	X	X
12	0	1	1	0	0	0	1
13	0	1	1	0	1	0	1
14	0	1	1	1	0	X	X
15	0	1	1	1	1	1	1
16 a 27	1	1	0	X	X	X	X
28	1	1	1	0	0	X	X
29	1	1	1	0	1	1	1
30	1	1	1	1	0	X	X
31	1	1	1	1	1	1	1

Ecuaciones:

$$Q1 = M\,Q1 + A,$$

$$Q2 = B\,Q2 + M$$

m	Q1	Q2	S1	S2
0	**0**	**0**	**0**	**0**
1	**0**	**1**	**0**	**1**
2	**1**	**0**	**X**	**X**
3	**1**	**1**	**1**	**1**

$$S1 = Q1,$$

$$S2 = Q2$$

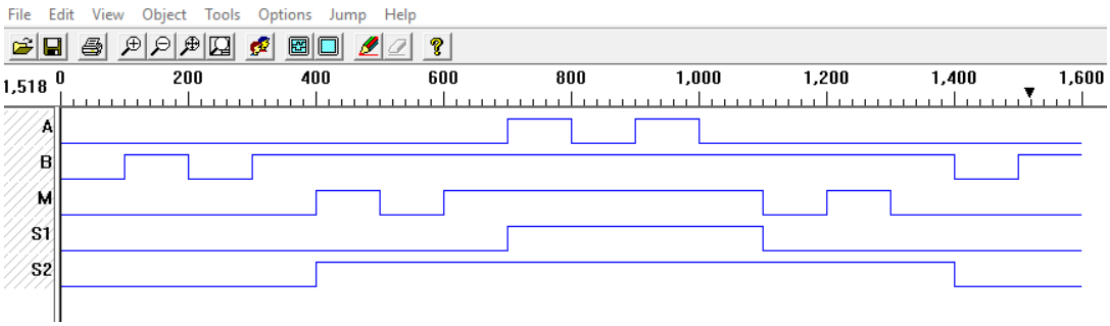
11.- Archivo en formato ABEL-HDL

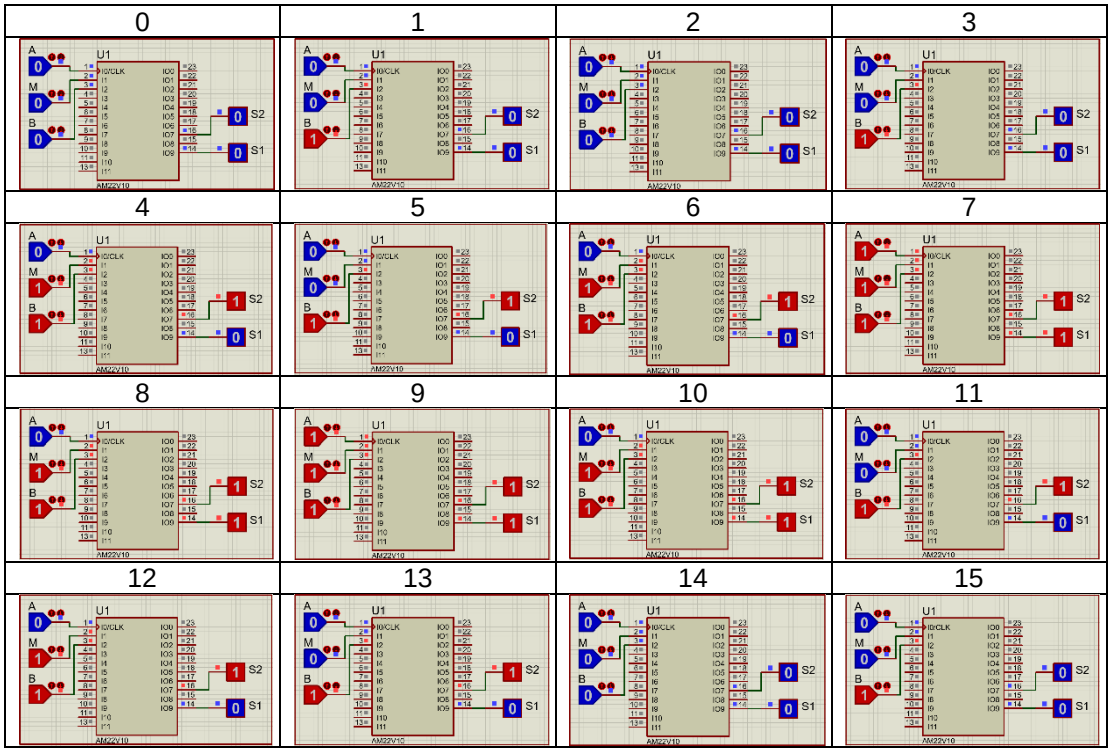
```
MODULE ptrece
"9de mayo 2019
"detector de 3 niveles
B,M,A pin 3..1;
S1,S2 pin 14,16 istype 'com';

equations
S1=M&S1#A;
S2=B&S2#M;
```

```
test_Vectors
([B,M,A]->[S1,S2])
[0,0,0]->[.x.,.x.];
[1,0,0]->[.x.,.x.];
[0,0,0]->[.x.,.x.];
[1,0,0]->[.x.,.x.];
[1,1,0]->[.x.,.x.];
[1,0,0]->[.x.,.x.];
[1,1,0]->[.x.,.x.];
[1,1,1]->[.x.,.x.];
[1,1,0]->[.x.,.x.];
[1,1,1]->[.x.,.x.];
[1,1,0]->[.x.,.x.];
[1,0,0]->[.x.,.x.];
[1,1,0]->[.x.,.x.];
[1,0,0]->[.x.,.x.];
[0,0,0]->[.x.,.x.];
[1,0,0]->[.x.,.x.];
END
```

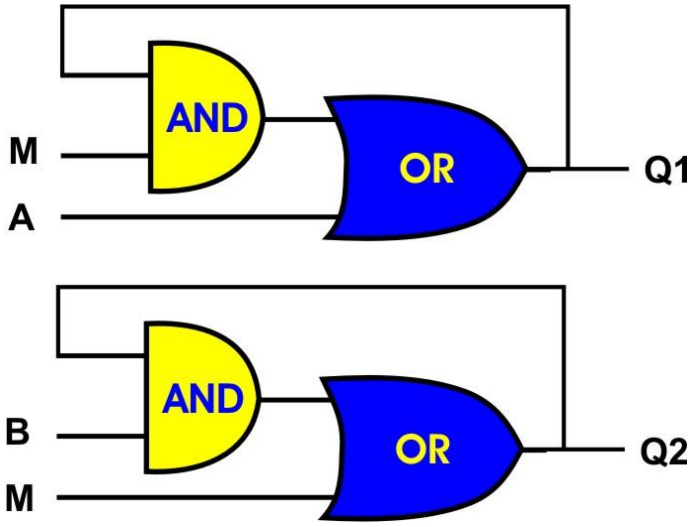
12.- Simulación.



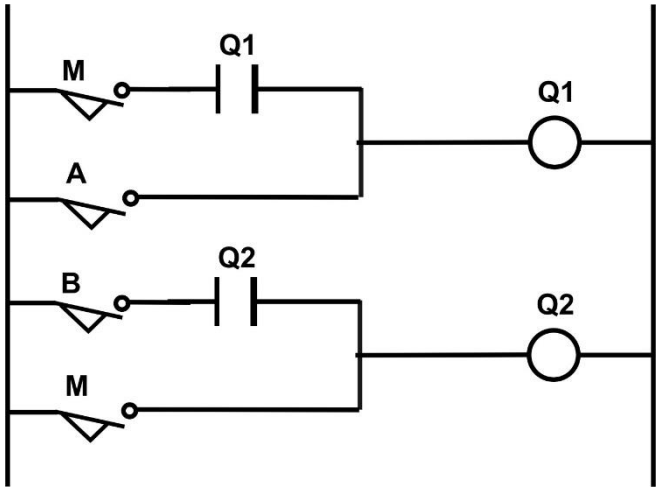


12.- Representación.

a) Diagrama esquemático



b) Diagrama escalera

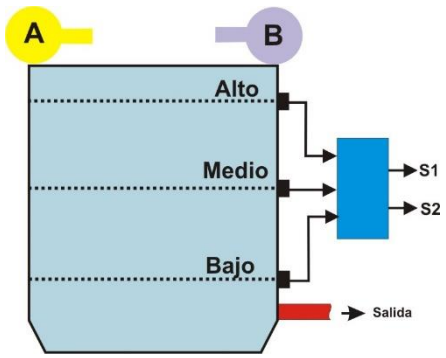


Problema 11, Dos bombas simultáneas caso 1

El tanque de la Figura se alimenta por medio de 2 bombas llamadas **A** y **B**.

El gasto de salida nunca será mayor al que proporcionen las dos bombas operando simultáneamente.

El tanque tiene un sistema detector de niveles que consta de 3 sensores de entrada nivel A (Alto), M (Medio) y B (Bajo), y dos salidas $\mathbf{S_2}, \mathbf{S_1}$. que indican lo siguiente:



	S2	S1
Bajo	**0**	**0**
Medio	**0**	**1**
Alto	**1**	**1**

Diseñar un sistema secuencial asíncrono, que controle la siguiente secuencia de operación de las bombas, en función de la salida del sistema de detección de niveles S2, S1.

1.- Partiendo de que el tanque se encuentra vacío (S2=0 y S1=0), se inicia el llenado funcionando ambas bombas **A** y **B,** hasta llegar al nivel medio (S2=0 y S1=1) y de ahí solo trabajar la bomba A, si se vacía de nuevo (S2=0 y S1=0) encenderán de nuevo ambas bombas y al llegar al nivel medio (S2=0 y S1=1) trabajara solo la bomba B.

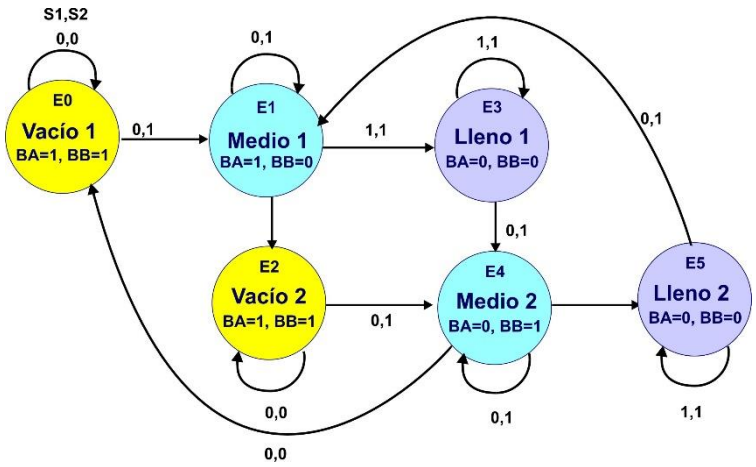
2.- Cada vez que se vacíe el tanque (S2=0 y S1=0), y de ahí pase al nivel medio (S2=1 y S1=0), deberá de alternarse el funcionamiento de las bombas A y B.

3.- Si el tanque se llena (S2=1 y S1=1) las bombas deberán de apagarse.

4.- Cada vez que se llene el tanque (S2=1 y S1=1), y de ahí pase al nivel medio (S2=1 y S1=0), deberá de alternarse el funcionamiento de las bombas A y B.

1.-Especificar el Sistema.

Diagrama de transición.



2.- Tabla de Flujo Primitiva.

	Entradas S2,S1				Salidas	
Hileras	00	01	11	10	A	B
0	**E0**	E1	-	-	1	1
1	E2	**E1**	E3	-	1	0
2	**E2**	E4	-	-	1	1
3	-	E4	**E3**	-	0	0
4	E0	**E4**	E5	-	0	1
5	-	E1	**E5**	-	0	0

3.- Reducción de Estados

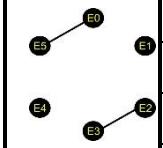
a) (E0, E2) (E1, E4)(E3, E5) Son estados estables en la misma columna

b) (E0, E2) (E3, E5) Tienen la misma salida

c) Sus estados siguientes no son equivalentes

por lo tanto, no hay reducción de estados.

4.- Mezcla de Filas y 5.- Expandir la Tabla de Salidas



	S1, S2				A, B			
Hileras	00	01	11	10	00	01	11	10
0,5	**E0**	E1	**E5**	-	11	--	00	--
1	E2	**E1**	E3	-	--	10	--	--
2,3	**E2**	E4	**E3**	-	11	--	00	--
4	E0	**E4**	E5	-	--	01	--	--

6.- Estados Internos.

	S1, S2				A, B			
Hileras	00	01	11	10	00	01	11	10
a	**a**	b	**a**	-	11	--	00	--
b	c	**b**	c	-	--	10	--	--
c	**c**	d	**c**	-	11	--	00	--
d	a	**d**	a	-	--	01	--	--

7.- Asignación de valores a los Estados.

Requisitos: 00).- (b → c), (d → a)

01).-, (a → b), (c → d)

11).- (b → c), (d → a)

Q1 \ Q2	0	1
0	a	d
1	b	c

	Q1	Q2
a	0	0
b	0	1
c	1	1
d	1	0

8.- Tabla de Estados.

	S2 S1				A B			
Q1 Q"	00	01	11	10	00	01	11	10
00	**00**	01	**00**	- -	11	- -	00	--
01	11	**01**	11	- -	- -	10	- -	--
11	**11**	10	**11**	- -	11	- -	00	--
10	00	**10**	00	- -	- -	01	- -	--

9.- Completando la tabla de salidas.

	S2 S1				A B			
Q1 Q"	00	01	11	10	00	01	11	10
00	**00**	01	**00**	- -	11	10	00	--
01	11	**01**	11	- -	1 -	10	- 0	--
11	**11**	10	**11**	- -	11	01	00	--
10	00	**10**	00	- -	- 1	01	0 -	--

10.- Obtención de las ecuaciones por medio de minimización.

Utilizando LogicAid obtenemos:

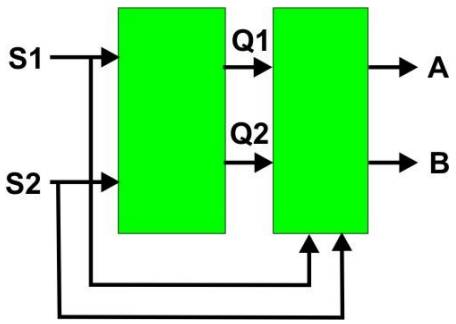
Ecuaciones

Q1=**S1'**Q2 +**S2'**S1 Q1 +S2 Q2

Q2+=**S1'**Q2+**S2'**S1**Q1'**+S2 Q2

A = S1' + S2' Q1'

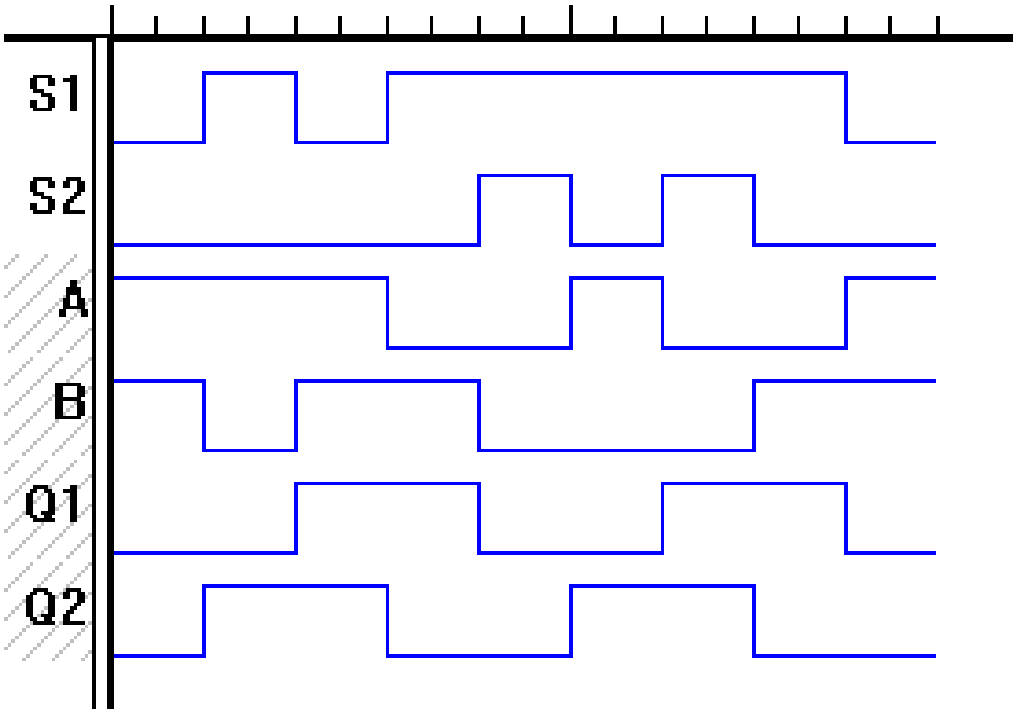
B = S1' + S2' Q1



11.- Elaborar el Archivo en formato ABEL-HDL incluyendo la Simulación.

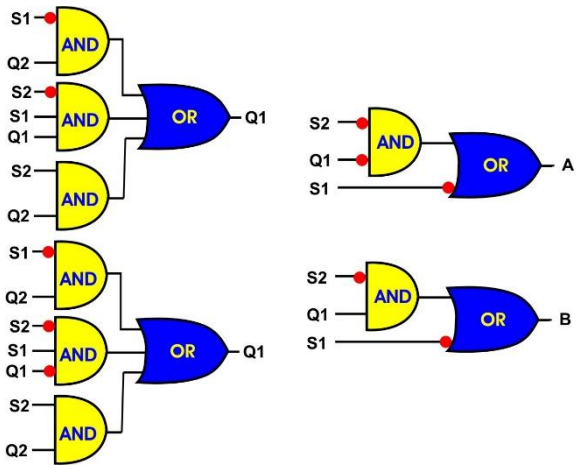
```
MODULE basinc
"Entradas
S1,S2 pin 1,2;
"salidas
A,B,Q1,Q2 pin 19..16 istype 'com';
equations
Q1=!S1&Q2#!S2&S1&Q1#S2&Q2;
Q2=!S1&Q2#!S2&S1&!Q1#S2&Q2;
A=!S1#!S2&!Q1;
B=!S1#!S2&Q1;
test_vectors
([S1,S2]->[A,B])
[0,0]->[.x.,.x.];
[1,0]->[.x.,.x.];
[0,0]->[.x.,.x.];
[1,0]->[.x.,.x.];
[1,1]->[.x.,.x.];
[1,0]->[.x.,.x.];
[1,1]->[.x.,.x.];
[1,0]->[.x.,.x.];
[0,0]->[.x.,.x.];
END
```

12.- Simulación.

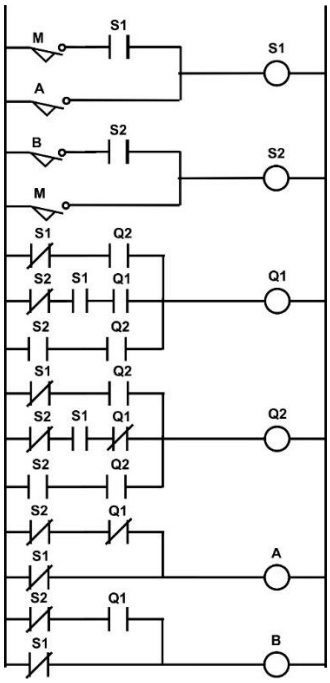

S1
S2
A
B
Q1
Q2

13.- Representación

a) Diagrama esquemático



b) Diagrama escalera

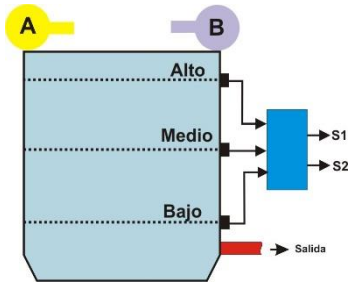


Problema 12, Dos bombas simultáneas caso 2

El tanque de la Figura se alimenta por medio de 2 bombas llamadas **A** y **B**.

El gasto de salida nunca será mayor al que proporcionen las dos bombas operando simultáneamente.

El tanque tiene un sistema detector de niveles que consta de 3 sensores de entrada nivel A (Alto), M (Medio) y B (Bajo), y dos salidas $\mathbf{S_2}, \mathbf{S_1}$. que indican lo siguiente:



	S2	S1
Bajo	**0**	**0**
Medio	**0**	**1**
Alto	**1**	**1**

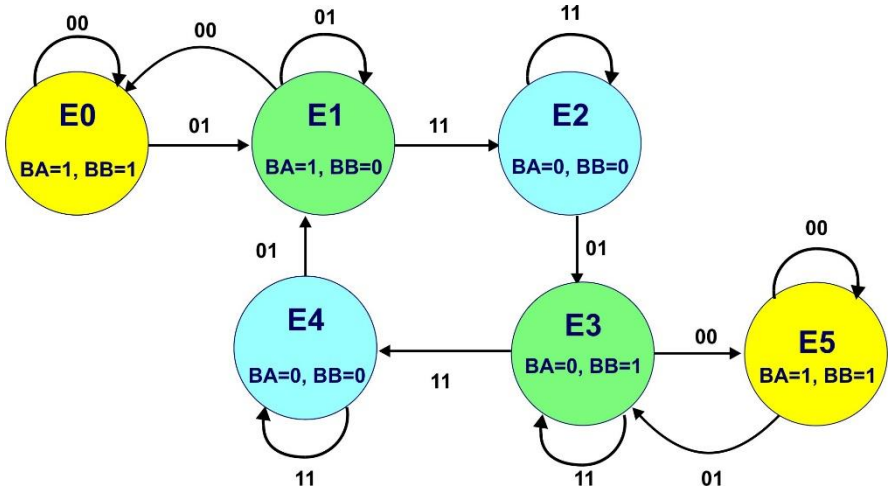
Diseñar un sistema secuencial asíncrono, que controle la siguiente secuencia de operación de las bombas, en función de la salida del sistema de detección de niveles S2, S1.

1.- Partiendo de que el tanque se encuentra vacío (S2=0 y S1=0), se inicia el llenado con ambas bombas **A** y **B,** hasta llegar al nivel medio (S2=0 y S1=1) y de ahí solo trabajar la bomba A, si se vacía de nuevo encenderán ambas bombas y al llegar al nivel medio (S2=0 y S1=1) trabajara de nuevo la bomba A, cuando el tanque se llene (S2=1 y S1=1) se deberán de apagar ambas bombas, al llegar de nuevo al nivel medio (S2=0 y S1=1) deberá de trabajar ahora solo la bomba B y si no es suficiente de modo que se vacié el tanque deberán de trabajar ambas bombas hasta llegar de nuevo al nivel medio (S2=0 y S1=1) y de ahí trabajar de nuevo solo la bomba B hasta llenarlo (S2=1 y S1=1).

2.- Cada vez que se llene el tanque (S2=1 y S1=1) y de ahí pase al nivel medio (S2=1 y S1=0), deberá de trabajar una sola bomba alternándose en su funcionamiento.

3.- Las bombas no se alternarán en su funcionamiento cuando el nivel pase de bajo a medio.

1.- Especificar el sistema



2.- Tabla de flujo primitiva

	Entradas S2,S1				Salidas	
Hileras	00	01	11	10	A	B
0	**E0**	E1	-	-	1	1
1	E0	**E1**	E2	-	1	0
2	-	E3	**E2**	-	0	0
3	E5	**E3**	E4	-	0	1
4	-	E1	**E4**	-	0	0
5	**E5**	E3	-		1	1

3.- Reducción de Estados.

No hay reducción de estados.

4.- Mezcla de Filas y 5.- Expansión de Tabla de salidas.

	Entradas S2,S1				Salidas A, B			
Hileras	00	01	11	10	00	01	11	10
0, 1	**E0**	**E1**	E2	-	11	10	-	-
2, 5	**E5**	E3	**E2**	-	11	-	00	-
3	E5	**E3**	E4	-	-	01	-	-
4	-	E1	**E4**		-	-	00	

6.- Estados Internos.

	Entradas S2,S1				Salidas A, B			
Hileras	00	01	11	10	00	01	11	10
a	**E0**	**E1**	E2	-	11	10	-	-
b	**E5**	E3	**E2**	-	11	-	00	-
c	E5	**E3**	E4	-	-	01	-	-
d	-	E1	**E4**		-	-	00	

	Entradas S2,S1				Salidas A, B			
Hileras	00	01	11	10	00	01	11	10
a	**a**	**a**	b	-	11	10	-	-
b	**b**	c	**b**	-	11	-	00	-
c	b	**c**	d	-	-	01	-	-
d	-	a	**d**		-	-	00	

7.- Asignación de Valores a los Estados.

Requisitos: 00).-(b → c), 01).- (b → c), (d → a) y 11).- (a → b), (c → d)

Q1 \ Q2	0	1
0	**b**	**a**
1	**c**	**d**

	Q1	Q2
b	**0**	**0**
c	**0**	**1**
d	**1**	**1**
a	**1**	**0**

8.- Tabla de Estados.

	Entradas S2,S1				Salidas A, B			
Hileras	00	01	11	10	00	01	11	10
10	**10**	**10**	00	-	11	10	-	-
00	**00**	01	**00**	-	11	-	00	-
01	00	**01**	11	-	-	01	-	-
11	-	10	**11**		-	-	00	

Acomodando Hileras en orden obtenemos

	Entradas S2,S1				Salidas A, B			
Hileras	00	01	11	10	00	01	11	10
00	**00**	01	**00**	-	11	-	00	-
01	00	**01**	11	-	-	01	-	-
11	-	10	**11**		-	-	00	
10	**10**	**10**	00	-	11	10	-	-

9.- Completando la Tabla de Salidas.

	Entradas S2,S1				Salidas A, B			
Hileras	00	01	11	10	00	01	11	10
00	**00**	01	**00**	-	11	01	00	-
01	00	**01**	11	-	11	01	0X	-
11	-	10	**11**		-	X0	00	
10	**10**	**10**	00	-	11	10	X0	-

10.- Obtención de las ecuaciones por medio de minimización.

	Entradas				Salidas			
m	S2	S1	Q1	Q2	Q1+	Q2+	A	B
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	0	X	1
2	0	0	1	0	1	0	1	1
3	0	0	1	1	X	X	X	X
4	0	1	0	0	0	1	0	1
5	0	1	0	1	0	1	0	1
6	0	1	1	0	1	0	1	0
7	0	1	1	1	1	0	X	0
8	1	0	0	0	X	X	X	X
9	1	0	0	1	X	X	X	X
10	1	0	1	0	X	X	X	X
11	1	0	1	1	X	X	X	X
12	1	1	0	0	0	0	0	0
13	1	1	0	1	1	1	0	X
14	1	1	1	0	0	0	0	0
15	1	1	1	1	1	1	X	0

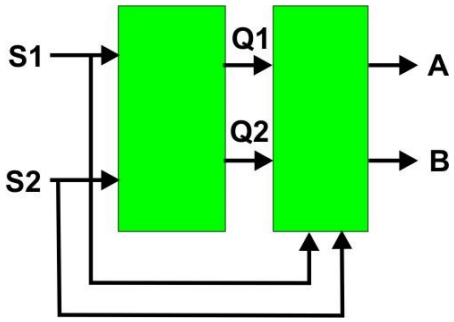
Ecuaciones

$$Q1=\mathbf{S1'}Q1 + S1\ Q2$$

$$Q2=\mathbf{S1'}S2\ \mathbf{Q1'} + S1\ Q2$$

$$A=\ \mathbf{S2'} + \mathbf{S1'}Q1$$

$$B=\ \mathbf{S2'} + \mathbf{S1'Q1'}$$



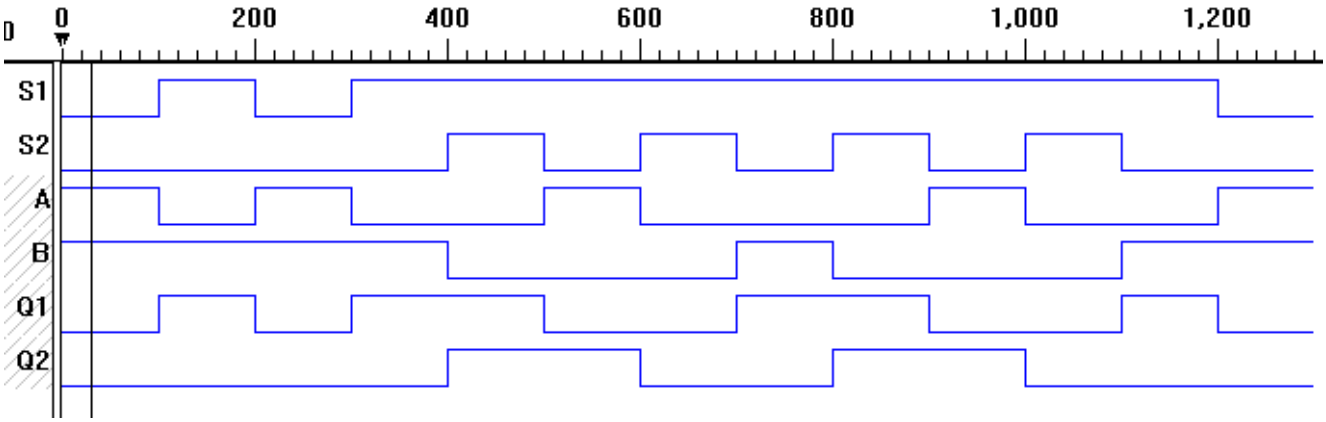
11.- Elaborar el Archivo en formato ABEL-HDL

```
MODULE ASINCT

"Entradas
S1,S2 pin 1,2;
"salidas
A,B,Q1,Q2 pin 19..16 istype 'com';
equations
Q2=!S2&Q2#S2&Q1;
Q1=!S2&S1&!Q2#S2&Q1;
A=!S1#!S2&Q2;
B=!S1#!S2&!Q2;
```

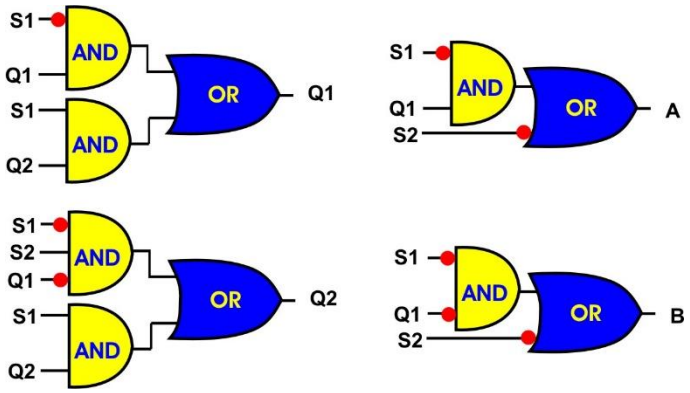
```
test_vectors
([S1,S2]->[A,B])
[0,0]->[.x.,.x.];
[1,0]->[.x.,.x.];
[0,0]->[.x.,.x.];
[1,0]->[.x.,.x.];
[1,1]->[.x.,.x.];
[1,0]->[.x.,.x.];
[1,1]->[.x.,.x.];
[1,0]->[.x.,.x.];
[1,1]->[.x.,.x.];
[1,0]->[.x.,.x.];
[1,1]->[.x.,.x.];
[1,0]->[.x.,.x.];
[0,0]->[.x.,.x.];
END
```

12.- Simulación

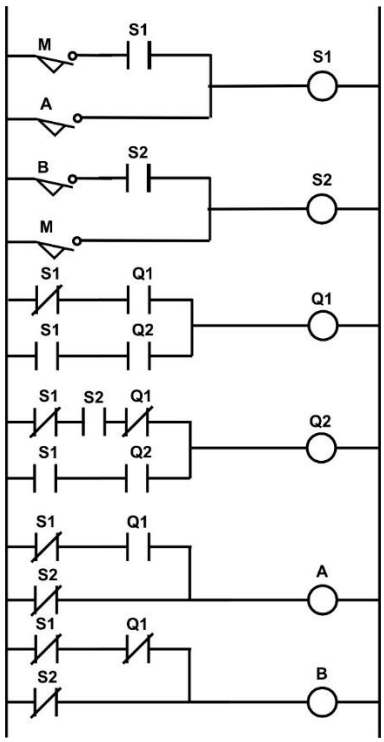


13.- Representación

a) Diagrama esquemático



b) Diagrama escalera

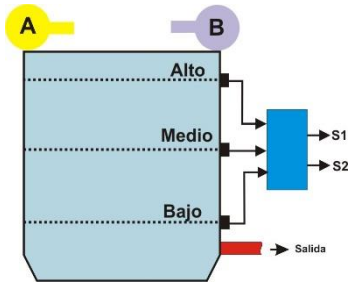


Problema 13, Dos bombas simultáneas caso 3.

El tanque de la Figura se alimenta por medio de 2 bombas llamadas **A** y **B**.

El gasto de salida nunca será mayor al que proporcionen las dos bombas operando simultáneamente.

El tanque tiene un sistema detector de niveles que consta de 3 sensores de entrada nivel A (Alto), M (Medio) y B (Bajo), y dos salidas $\mathbf{S_2}, \mathbf{S_1}$. que indican lo siguiente:



	S2	S1
Bajo	**0**	**0**
Medio	**0**	**1**
Alto	**1**	**1**

Diseñar un sistema secuencial asíncrono, que controle la siguiente secuencia de operación de las bombas, en función de la salida del sistema de detección de niveles S2, S1.

1.- Partiendo de que el tanque se encuentra vacío (S2=0 y S1=0), se inicia el llenado con ambas bombas **A** y **B,** hasta llenar el tanque (S2=1 y S1=1) y entonces desconectarlas.

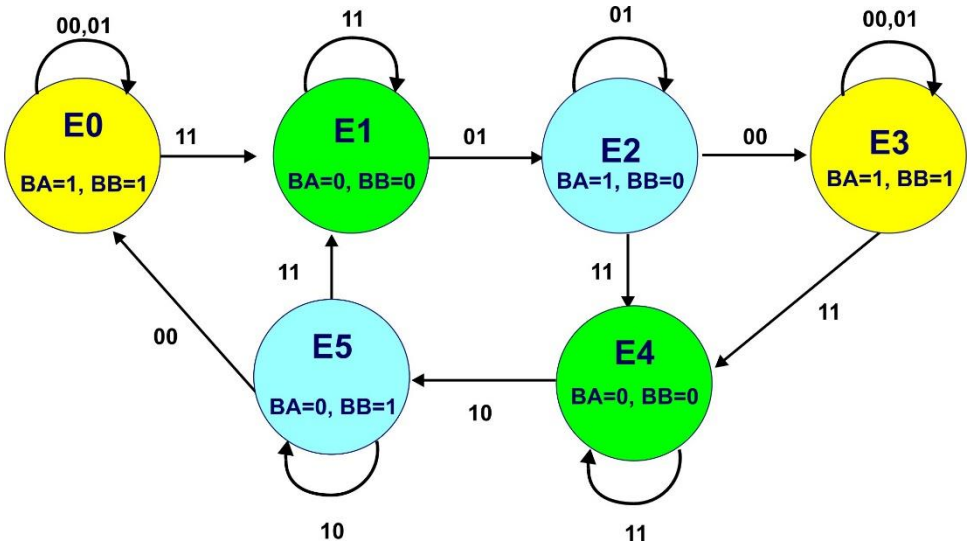
2.-una vez lleno si el nivel llega a medio (S2=0 y S1=1) deberá solo trabajar la bomba A, si el nivel sigue bajando hasta el nivel bajo (S2=0 y S1=0) deberán de trabajar de nuevo ambas bombas **A** y **B,** hasta llenar el tanque (S2=1 y S1=1) y entonces desconectarlas

3.- una vez lleno si el nivel llega de nuevo a medio (S2=0 y S1=1) deberá solo trabajar la bomba B, si el nivel sigue bajando hasta el nivel bajo (S2=0 y S1=0) deberán de trabajar de nuevo ambas bombas **A** y **B,** hasta llenar el tanque (S2=1 y S1=1) y entonces desconectarlas

4.- Cada vez que el nivel pase de lleno a medio, deberá de trabajar una sola bomba alternándose en su funcionamiento.

5.- Cada vez que se vacié trabajaran ambas bombas hasta llenar el tanque.

1.- Especificar el sistema



2.- Tabla de flujo primitiva

	Entradas S2,S1				
Hileras	00	01	11	10	AB
0	**E0**	**E0**	E1	-	11
1	-	E2	**E1**	-	00
2	E3	**E2**	E4	-	10
3	**E3**	**E3**	E4		11
4	-	E5	**E4**		00
5	E0	**E5**	E1		01

3.- Reducción de Estados.

No hay reducción de estados.

4.- Mezcla de Filas

	Entradas S2,S1				
Hileras	00	01	11	10	AB
0	**E0**	**E0**	E1	-	11
1	-	E2	**E1**	-	00
2	E3	**E2**	E4	-	10
3	**E3**	**E3**	E4		11
4	-	E5	**E4**		00
5	E0	**E5**	E1		01

5.- Expansión de salidas. En este caso no es necesario expandir las salidas.

6.- Tabla de Estados Internos.

	Entradas S2,S1				
Hileras	00	01	11	10	AB
a	**E0**	**E0**	E1	-	11
b	-	E2	**E1**	-	00
c	E3	**E2**	E4	-	10
d	**E3**	**E3**	E4		11
e	-	E5	**E4**		00
f	E0	**E5**	E1		01

Sustituyendo los estados de E0 a E5 por a hasta f respectivamente obtenemos:

	Entradas S2,S1				
Hileras	00	01	11	10	AB
a	**a**	**a**	b	-	11
b	-	c	**b**	-	00
c	d	**c**	e	-	10
d	**d**	**d**	e		11
e	-	f	**e**		00

f	a	f	b		01

7.- Asignación de Valores a los Estados.

	Entradas S2,S1				
Hileras	00	01	11	10	AB
a	a	a	b	-	11
b	-	c	b	-	00
c	d	c	e	-	10
d	d	d	e		11
e	-	f	e		00
f	a	f	b		01

Se require que:

Columna 00) (c→d) y (f→a), Columna 01) (b→c) y (e→f) y

Columna 11) (a→b), (c→e) (d→e) y (f→b)

Analizaremos el caso de b que tiene que tener un solo cambio con a, c y f y además f debe de tener un solo cambio con a, por lo que resulta imposible la asignación

Q1, Q2

Q3	00	01	11	10
0	a	b	f	
1		c		

Se propone para que sea posible la asignación incluir dos estados transitorios que llamaremos $\propto$ y β. De modo que para pasar de f $\rightarrow \propto \rightarrow$ a y de e$\rightarrow\beta\rightarrow$d como lo indica la siguiente tabla.

Q3 \ Q1, Q2	00	01	11	10
0	**a**	**b**	**f**	$\propto$
1	**d**	**c**	**e**	β

	Entradas S2,S1				
Hileras	00	01	11	10	AB
a	**a**	**a**	b	-	11
b	-	c	**b**	-	00
c	d	**c**	e	-	10
d	**d**	**d**	β		11
e	-	f	**e**		00
f	$\propto$	**f**	b		01
$\propto$	**a**	-	-		
β	-	-	**e**		

Asignación:

	Q1	Q2	Q3
a	0	0	0
d	0	0	1
b	0	1	0
c	0	1	1
∝	1	0	0
β	1	0	1
f	1	1	0
e	1	1	1

8.- Tabla de Estados.

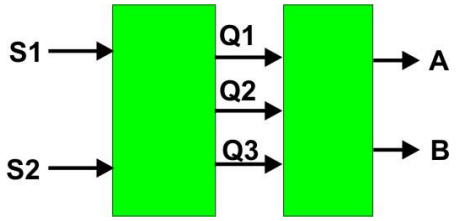
	Entradas S2,S1				
Hileras	00	01	11	10	AB
000	000	000	010	-	11
010	-	011	010	-	00
011	001	011	111	-	10
001	001	001	101		11
111	-	110	111		00
110	100	110	010		01
100	000	-	-		
101	-	-	111		

Acomodando hileras

	Entradas S2,S1				
Hileras	00	01	11	10	AB
000	000	000	010	-	11
001	001	001	101		11
010	-	011	010	-	00
011	001	011	111	-	10
100	000	-	-		
101	-	-	111		
110	100	110	010		01
111	-	110	111		00

9.- Completar Tabla de Salidas.

No es necesario completar la tabla de salidas.



10.- Obtención de las ecuaciones por medio de minimización.

Tabla para obtener los valores de Q1+, Q2+ y Q3+

m	S1 S2 Q1 Q2 Q3	Q1+ Q2+ Q3+
0	00000	000
1	00001	001
2	00010	---
3	00011	001
4	00100	000
5	00101	---
6	00110	100
7	00111	---
8	01000	000
9	01001	001
10	01010	011
11	01011	011
12	01100	---
13	01101	---
14	01110	110
15	01111	110
16	10000	---
17	10001	---
18	10010	---
10	10011	---
20	10100	---
21	10101	---
22	10110	---
23	10111	---
24	11000	010
25	11001	101
26	11010	010
27	11011	111
28	11100	---
29	11101	111
30	11110	010
31	11111	111

$$Q1+ = S1\ Q3 + \mathbf{S1'}Q1\ Q2$$

$$Q2+ = S2\ Q2 + S1\ \mathbf{Q3'} + Q1\ Q3$$

$$Q3+ = \mathbf{S1'Q1'}Q2 + S1\ Q3 + \mathbf{Q2'}Q3$$

Q1 Q2 Q3	A B
000	**11**
001	**11**
010	**00**
011	**10**
100	**--**
101	**--**
110	**01**
111	00

$$A = \mathbf{Q2'} + \mathbf{Q1'}Q3$$

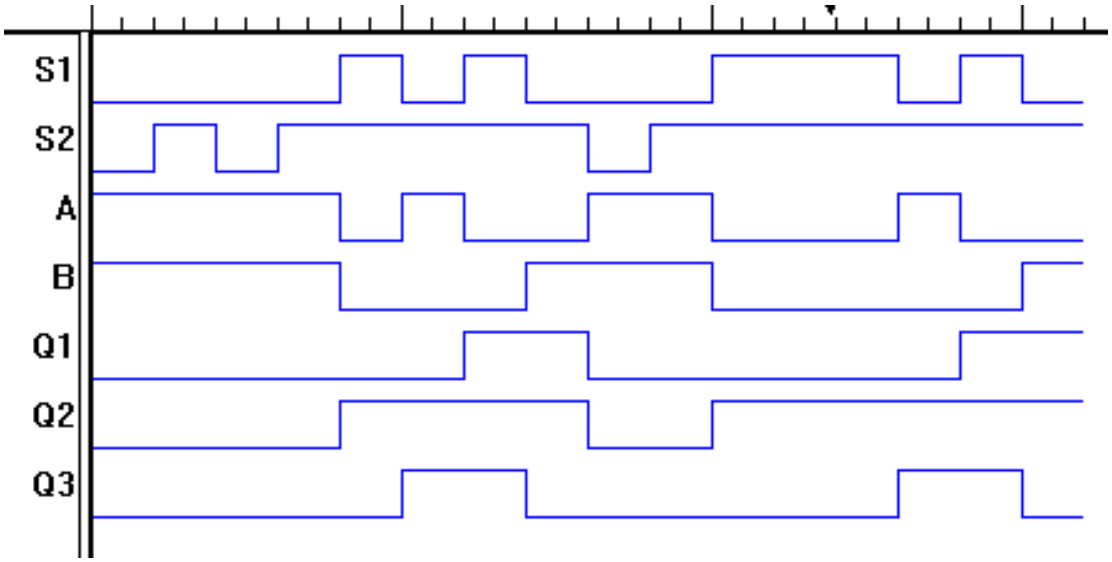
$$B = \mathbf{Q2'} + Q1\ \mathbf{Q3'}$$

11.- Archivo ABEL-HDL

```
MODULE dbasy
"Entradas
S1,S2 pin 1,2;
"salidas
A,B,Q1,Q2,Q3 pin 19..15 istype 'com';
equations
Q1=S1&Q3#!S1&Q1&Q2;
Q2=S2&Q2#S1&!Q3#Q1&Q3;
Q3=!S1&!Q1&Q2#S1&Q3#!Q1&Q3;
A=!Q2#!Q1&Q3;
B=!Q2#Q1&!Q3;
```

```
test_vectors
([S1,S2]->[A,B])
[0,0]->[.x.,.x.];
[0,1]->[.x.,.x.];
[0,0]->[.x.,.x.];
[0,1]->[.x.,.x.];
[1,1]->[.x.,.x.];
[0,1]->[.x.,.x.];
[1,1]->[.x.,.x.];
[0,1]->[.x.,.x.];
[0,0]->[.x.,.x.];
[0,1]->[.x.,.x.];
[1,1]->[.x.,.x.];
[1,1]->[.x.,.x.];
[1,1]->[.x.,.x.];
[0,1]->[.x.,.x.];
[1,1]->[.x.,.x.];
[0,1]->[.x.,.x.];
END
```

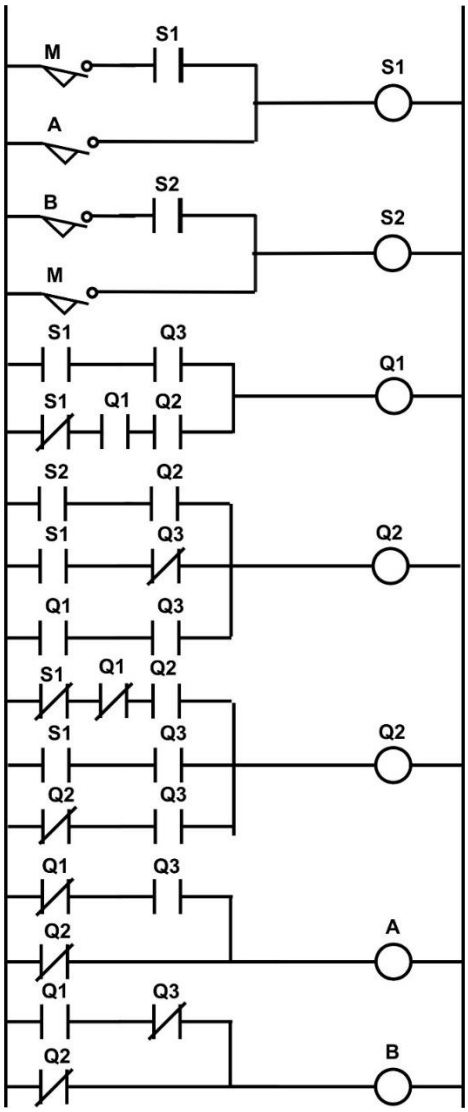
12.- Simulación



13.- Representación

Diagrama esquemático

Diagrama escalera



Problema 14, Sistema de Arranque y Paro

Diseñe un sistema secuencial asíncrono que contenga dos botones de entrada llamados A (arranque) y **P** (paro) de modo que al oprimir el botón A la salida deberá tomar el valor de uno mientras que con el botón P la salida Y deberá de tomar el valor de cero como lo indica la siguiente gráfica:

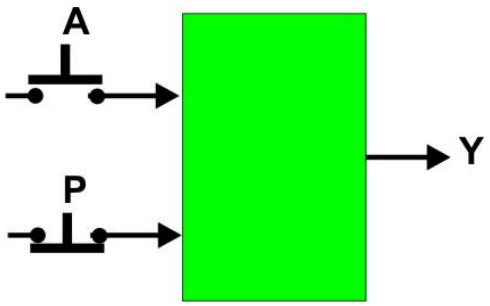


Diagrama de bloques

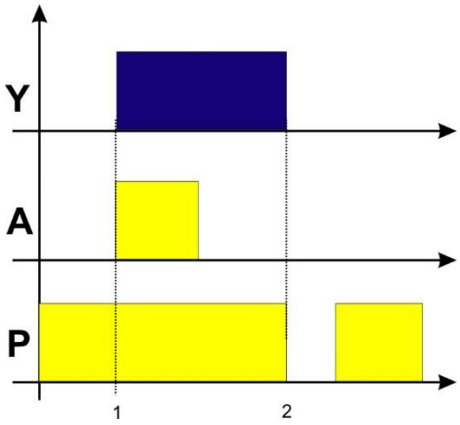
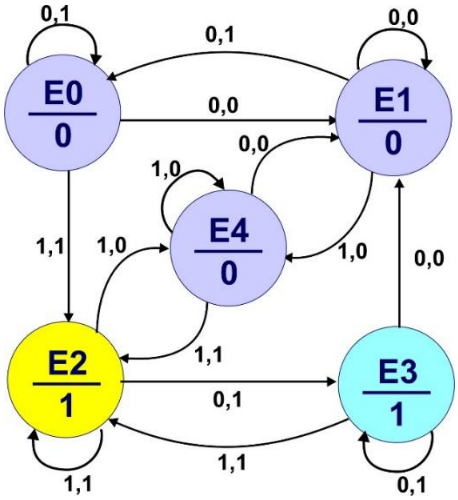


Diagrama de tiempos

1.- Especificar el sistema.

Diagrama de transición



2.- Tabla de flujo primitiva

A P	00	01	10	11	Y
0	E2	**E0**	**-**	E1	**0**
1	**-**	E3	E4	**E1**	**1**
2	**E2**	E0	E4	**-**	**0**
3	E2	**E3**	**-**	E1	**1**
4	E2	**-**	**E4**	E1	**0**

3.- Reducción de Estados.

a) Son estados estables en la misma columna (misma combinación de entradas)

E0 y E3

b) Tienen la misma salida.

E0 y E3 tienen diferente salida por lo que no hay reducción de estados

4.- Mezcla de Filas

Mezclando las filas 1con 2 y con 5 además 3 con 4 obtenemos la siguiente tabla:

	A P 00	01	10	11	Y
0,2,4	**E2**	**E0**	**E4**	E1	**0**
1,3	E2	**E3**	E4	**E1**	**1**

5.- Expandir Tabla de Salidas.

No es necesario expandir tablado salidas porque 1,2 y 5 tienen salida Y=0 y 3 y 4 salida Y=1.

6.- Tabla de Estados Internos.

Asignando el valor a la primera hilera de a y a la segunda el de b obtenemos:

	A P 00	**01**	**10**	**11**	**Y**
a	**E2**	**E0**	**E4**	E1	**0**
b	E2	**E3**	E4	**E1**	**1**

Sustituyendo los nombres de (E0, E2, E4) $\rightarrow$ a, (E3, E1) $\rightarrow$ b, obtenemos la tabla de estados internos:

	A P 00	**01**	**10**	**11**	**Y**
a	**a**	**a**	**a**	b	**0**
b	a	**b**	a	**b**	**1**

7.- Asignación de Valores a los Estados.

En este ejemplo la asignación es muy simple a=0 y b=1

8.- Tabla de Estados Totales.

Sustituyendo los valores de la asignación obtenemos:

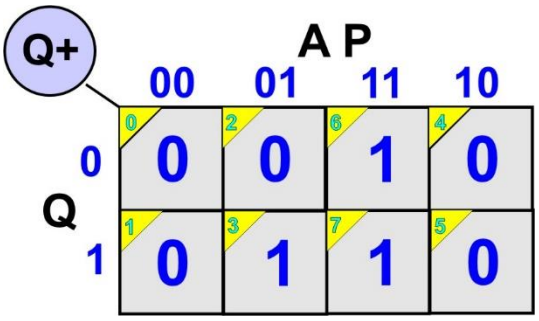
	A P 00	**01**	**10**	**11**	**Y**
0	**a**	**a**	**a**	b	**0**
1	a	**b**	a	**b**	**1**

	A P				
Q1	**00**	**01**	**10**	**11**	**Y**
0	**0**	**0**	**0**	**1**	**0**
1	**0**	**1**	**0**	**1**	**1**

9.- Completar Tabla de Salidas.

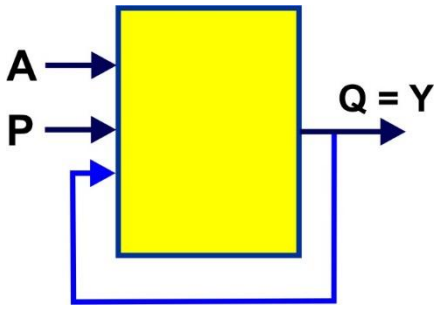
No es necesario completar tabla de salidas.

10.- Obtención de las ecuaciones por medio de minimización. (mapa de karnaugh)



Y = Q

Q+= P(A+Q)

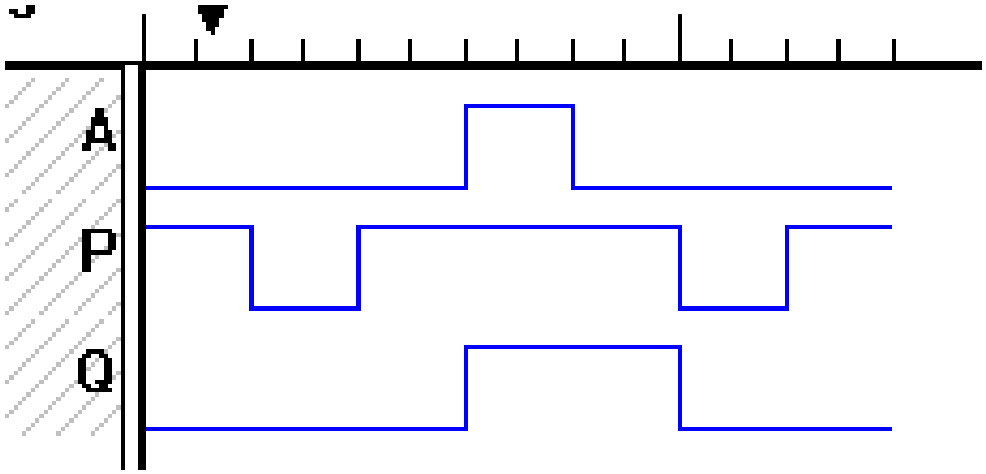


11.- Archivo en formato ABEL-HDL

```
MODULE ap
"entradas
A,P pin 1,2;
"Salidas
Q pin 19 istype 'com';
equations
Q=P&(A#Q);
```

```
test_vectors
([A,P]->Q)
[0,1]->.x.;
[0,0]->.x.;
[0,1]->.x.;
[1,1]->.x.;
[0,1]->.x.;
[0,0]->.x.;
[0,1]->.x.;
end
```

12.-Simulación



13.- Representación.

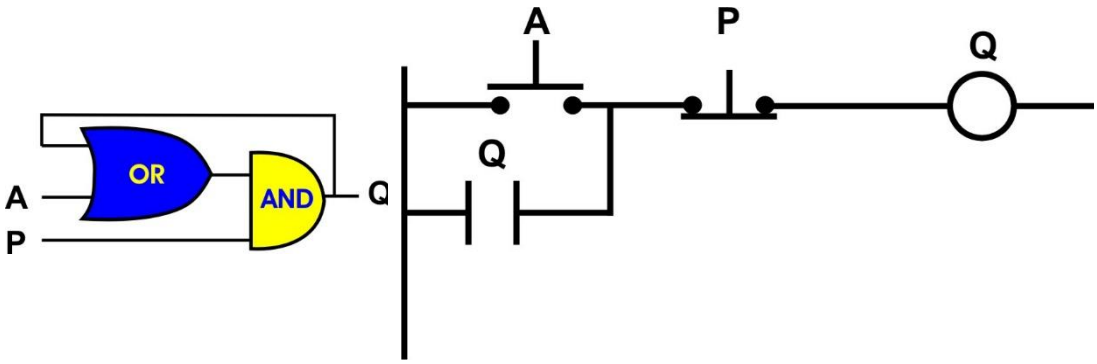


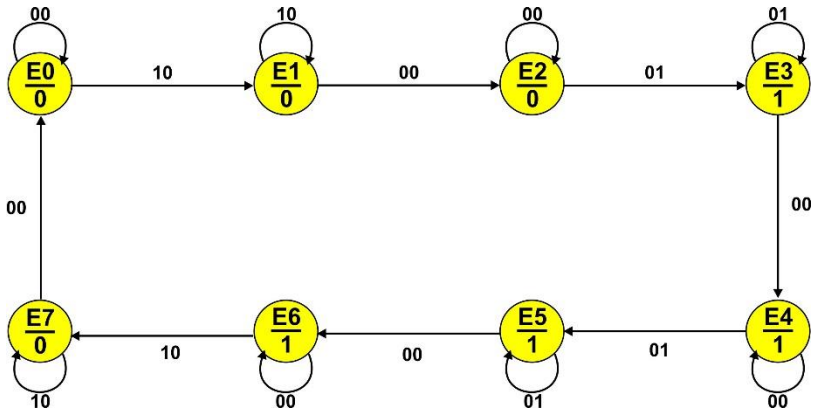
Diagrama esquemático Diagrama escalera

Problema 15, Arranque con secuencia AB y paro con secuencia BA

Diseñe un sistema secuencial asíncrono que contenga dos botones de entrada llamados **A** y **B** y una salida Y de modo que al oprimir los botones en la secuencia A soltar y después B la salida Y deberá de cambiar de un valor inicial cero a uno, al soltar el botón B la salida deberá de mantenerse en uno y una vez que Y=1 al oprimir los botones en la secuencia B soltar y después A la salida Y deberá de cambiar de un valor uno a cero, al soltar el botón A la salida deberá de mantenerse en cero.

1.- Especificar el Sistema.

A continuación se muestra solo la secuencia descrita en la redacción anterior:



Hay que considerar también que estando en el estado E0 se puede oprimir primero B (0,1) o en el estado E1 si se oprimen los dos botones a la vez (1,1), una vez considerando todos los posibles eventos se puede representar en el siguiente diagrama de transición:

2.- Tabla de Flujo Primitiva.

	Entradas A,B				Salida
	00	01	10	11	Y
0	**E0**	E8	E1	-	0
1	E2	-	**E1**	E10	0
2	**E2**	E3	E12	-	0
3	E4	**E3**	-	E14	1
4	**E4**	E5	E16	-	1
5	E6	**E5**	-	E18	1
6	**E6**	E20	E7	-	1
7	E0	-	**E7**	E22	0
8	E0	**E8**	-	E9	0
9	-	E8	E1	**E9**	0
10	-	E11	E1	**E10**	0
11	E2	**E11**	-	E10	0
12	E2	-	**E12**	E13	0
13	-	E3	E12	**E13**	0
14	-	E3	E15	**E14**	1
15	E4	-	**E15**	E14	1
16	E4	-	**E16**	E17	1
17	-	E5	E16	**E17**	1
18	-	E5	E19	**E18**	1
19	E6	-	**E19**	E18	1
20	E6	**E20**	-	E21	1
21	-	E20	E7	**E21**	1
22	-	E23	E7	**E22**	0
23	E0	**E23**	-	E22	0

3.- Eliminación de Estados Redundantes.

No es posible la reducción de estados.

4.- Mezcla de Filas.

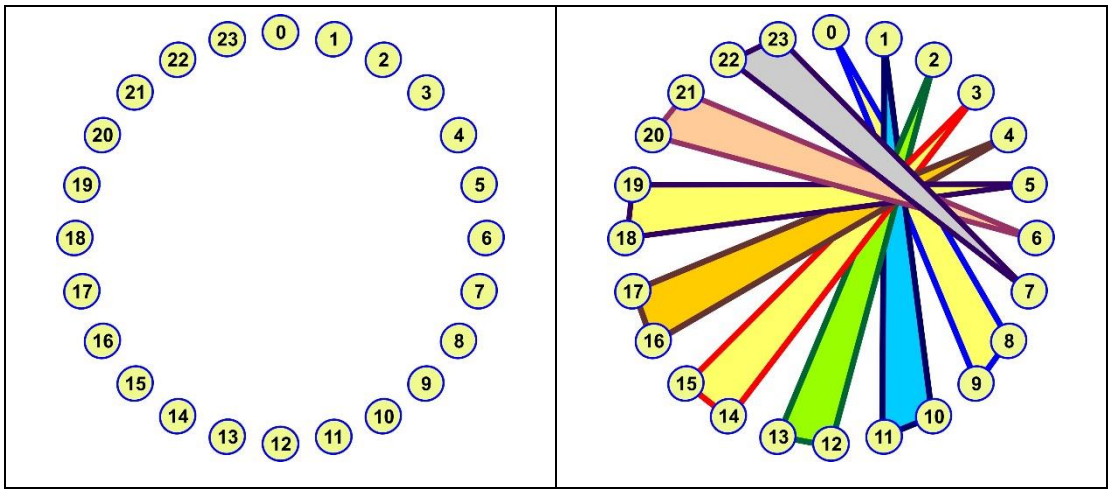


Tabla mezclada

Hileras	Entradas A, B				Salida Y
	00	01	10	11	
0, 8, 9	E0	E8	E1	E9	0
1, 10, 11	E2	E11	E1	E10	0
2, 12, 13	E2	E3	E12	E13	0
3, 14, 15	E4	E3	E15	E14	1
4, 16, 17	E4	E5	E16	E17	1
5, 18, 19	E6	E5	E19	E18	1
6, 20, 21	E6	E20	E7	E21	1
7, 22, 23	E0	E23	E7	E22	0

5.- Expandir Tabla de Salidas.

No es necesario expandir la tabla de salidas

6.- Tabla de estados internos.

	Entradas A,B				Salida Y
	00	01	10	11	
a	E0	E8	E1	E9	0
b	E2	E11	E1	E10	0
c	E2	E3	E12	E13	0
d	E4	E3	E15	E14	1
e	E4	E5	E16	E17	1
f	E6	E5	E19	E18	1
g	E6	E20	E7	E21	1
h	E0	E23	E7	E22	0

	Entradas A,B				Salida Y
	00	01	10	11	
a	a	a	b	a	0
b	c	b	b	b	0
c	c	d	c	c	0
d	e	d	d	D	1
e	e	f	e	e	1
f	g	f	f	f	1
g	g	g	h	g	1
h	a	h	h	h	0

7.- Asignación de Valores a los Estados.

Requisitos: 00).- (b $\rightarrow$ c), (d $\rightarrow$ e), (f $\rightarrow$ g), (h $\rightarrow$ a)

01).-, (c $\rightarrow$ d), (e $\rightarrow$ f)

10).- (a $\rightarrow$ b), (g $\rightarrow$ h), 11).- no hay transiciones

Q1, Q2

Q3	00	01	11	10
0	a	h	f	g
1	b	c	d	e

Asignación

	Q1	Q2	Q3
a	0	0	0
b	0	0	1
h	0	1	0
c	0	1	1
g	1	0	0
e	1	0	1
f	1	1	0
d	1	1	1

8.- Tabla de Estados Totales.

	00	01	10	11	Y
a	a	a	b	a	0
b	c	b	b	b	0
c	c	d	c	c	0
d	e	d	d	d	1
e	e	f	e	e	1
f	g	f	f	f	1
g	g	g	h	g	1
h	a	h	h	h	0

Sustituyendo los valores asignados obtenemos

Q1, Q2, Q3	Entradas A,B				Salida Y
	00	01	10	11	
000	000	000	001	000	0
001	011	001	001	001	0
011	011	111	011	011	0
111	101	111	111	111	1
101	101	100	101	101	1
100	110	100	100	100	1
110	110	110	010	110	1
010	000	010	010	010	0

Acomodando valores en orden ascendente obtenemos:

Q1, Q2, Q3	Entradas A,B				Salida Y
	00	01	10	11	
000	000	000	001	000	0
001	011	001	001	001	0
010	000	010	010	010	0
011	011	111	011	011	0
100	110	100	100	100	1
101	101	100	101	101	1
110	110	110	010	110	1
111	101	111	111	111	1

9.- Completar Tabla de Salidas. No es necesario completar tabla de salidas

10.- Obtención de las ecuaciones por medio de minimización.

Q1+ = Q2 Q3 **A'**B + Q1 **Q2'** + Q1 **A'** + Q1 B + Q1 Q3

Q2+ = **Q1'**Q3 **A'B'** + Q2 B + Q2 A + Q1 **Q3'A'B'**

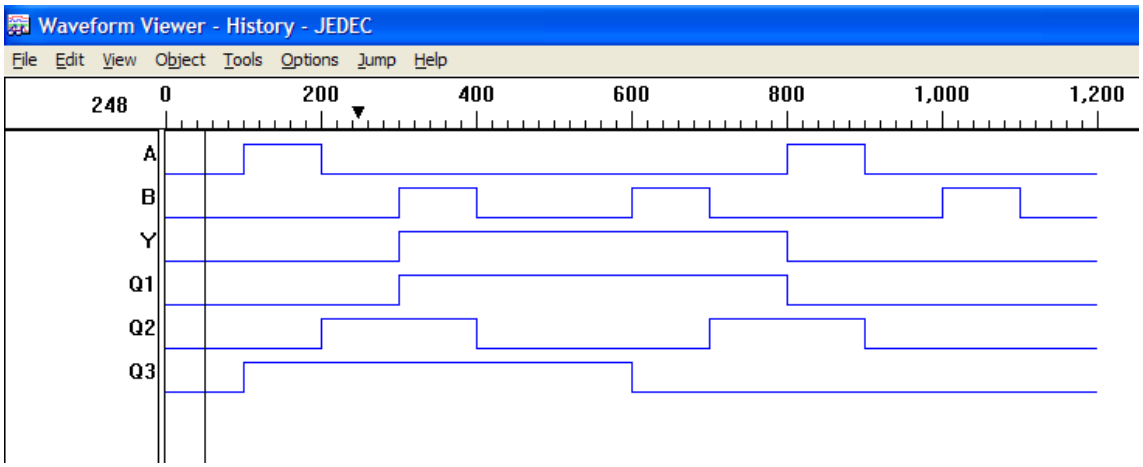
Q3+ = **Q1'Q2'**A B' + **Q1'**Q3 + Q3 **B'** + Q3 A + Q2 Q3

Y=Q1

11.- Elaborar el Archivo en formato ABEL-HDL.

```
MODULE AC
A,B PIN 1,2;
Q1,Q2,Q3,Y PIN 19,18,17,16 ISTYPE 'COM';
EQUATIONS
Q1=(Q2&Q3&!A&B)#(Q1&!Q2)#(Q1&!A)#(Q1&B)#(Q1&Q3);
Q2=(!Q1&Q3&!A&!B)#(Q2&B)#(Q2&A)#(Q1&!Q3&!A&!B);
Q3=(!Q1&!Q2&A&!B)#(!Q1&Q3)#(Q3&!B)#(Q3&A)#(Q2&Q3);
Y=Q1;
TEST_VECTORS
([A,B]->[Y,Q1,Q2,Q3])
[0,0]->[.X.,.X.,.X.,.X.];
[1,0]->[.X.,.X.,.X.,.X.];
[0,0]->[.X.,.X.,.X.,.X.];
[0,1]->[.X.,.X.,.X.,.X.];
[0,0]->[.X.,.X.,.X.,.X.];
[0,0]->[.X.,.X.,.X.,.X.];
[0,1]->[.X.,.X.,.X.,.X.];
[0,0]->[.X.,.X.,.X.,.X.];
[1,0]->[.X.,.X.,.X.,.X.];
[0,0]->[.X.,.X.,.X.,.X.];
[0,1]->[.X.,.X.,.X.,.X.];
[0,0]->[.X.,.X.,.X.,.X.];
END
```

12.- Simulación.



13.- Representación gráfica.

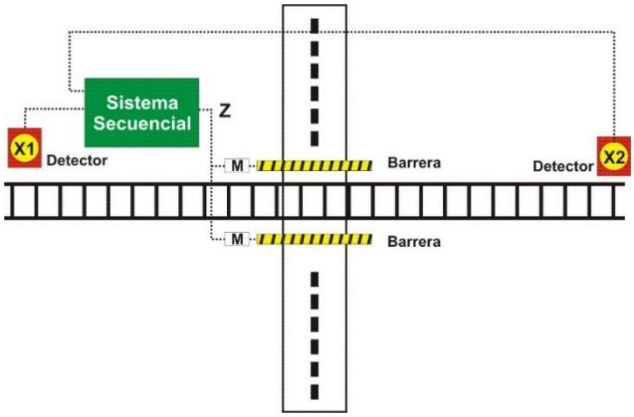
Problema 16, Barreras de Seguridad de una Vía de Ferrocarril

En una vía de ferrocarril con tráfico en ambos sentidos corta una carretera en la cual se colocan barreras que son activadas por una salida Z por medio de un sistema secuencial como se muestra en la figura de abajo.

A 500 m del punto de cruce en ambas direcciones se colocan detectores X1 y X2 respectivamente. A partir de un estado inicial en donde Z = 0, este valor debe de pasar al estado uno (Z =1) cuando se acerca un ferrocarril en cualquier sentido, al rebasar la máquina los 500 m del cruce y debe de volver al estado cero cuando el último vagón se aleja más de dicha distancia independientemente de la longitud del ferrocarril.

Considere que no está permitido hacer maniobras es decir no hay cambio de dirección y que sólo pasa un tren a la vez.

Diseñe el sistema secuencial asíncrono.



descripción del problema y sus sensores

Con las condiciones descritas anteriormente se puede presentar cuatro casos

- Caso 1 tren corto con dirección X1 a X2
- Caso 2 tren corto con dirección X2 a X1
- Caso 3 tren largo con dirección X1 a X2
- Caso 4 tren largo con dirección X2 a X1

Detallaremos a continuación la secuencia de cada uno de los casos:

Caso 1	
X1 X2	Z
00	0
10	1
00	1
01	1
00	0

Caso 2	
X1 X2	Z
00	0
01	1
00	1
10	1
00	0

Caso 3	
X1 X2	Z
00	0
10	1
11	1
01	1
00	0

Caso 4	
X1 X2	Z
00	0
01	1
11	1
10	1
11	0

1.- Especificar el Sistema.

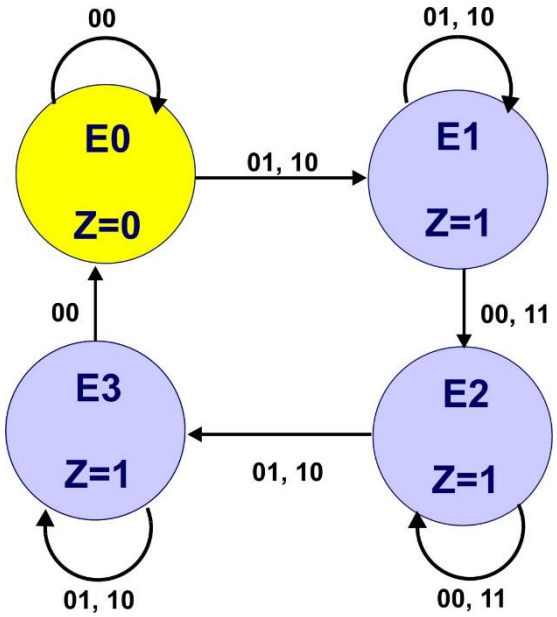


diagrama de transición

2.- Tabla de Flujo Primitiva.

	Entradas X1, X2				Salida
Hileras	00	01	11	10	Z
0	**E0**	E1	-	E1	0
1	E2	**E1**	E2	**E1**	1
2	**E2**	E3	**E2**	E3	1
3	E0	**E3**		**E3**	1

3.- Reducción de Estados Redundantes.

No hay posibilidad de Reducción de Estados.

4.- Mezcla de Filas.

No es posible la mezcla de filas.

5.- Expandir la tabla de salidas.

En este caso no hay expansión de tablas de salida.

6.- Tabla de Estados Internos.

	Entradas X1, X2				Salida
Hileras	00	01	11	10	Z
a	**E0**	E1	-	E1	0
b	E2	**E1**	E2	**E1**	1
c	**E2**	E3	**E2**	E3	1
d	E0	**E3**		**E3**	1

	Entradas X1, X2				Salida
Hileras	00	01	11	10	Z
a	**a**	b	-	b	0
b	c	**b**	c	**b**	1
c	**c**	d	**c**	d	1
d	a	**d**		**d**	1

7.- Asignación de Valores a los Estados.

Análisis por columna

00) b→c, d→a, 01) a→b, c→d, 11) b→c, 10) a→b, c→d

		Q2 = 0	Q2 = 1
Q1	0	**a**	**d**
	1	**b**	**c**

Asignación

a =00, b = 01, c = 11, d= 10

8.-Tabla de Estados

	Entradas X1, X2				Salida
Q1 Q2	**00**	**01**	**11**	**10**	**Z**
00	**00**	**01**	**-**	**01**	**0**
01	**11**	**01**	**11**	**01**	**1**
11	**11**	**10**	**11**	**10**	**1**
10	**00**	**10**	**-**	**10**	**1**

9.- Completar Tabla de Salidas.

No es necesario completar la tabla de salidas.

10.- Obtención de las ecuaciones por medio de minimización.

M	X1	X2	Q1	Q2	Q1+	Q2+
0	**0**	**0**	**0**	**0**	**0**	**0**
1	**0**	**0**	**0**	**1**	**1**	**1**
2	**0**	**0**	**1**	**0**	**0**	**0**
3	**0**	**0**	**1**	**1**	**1**	**1**
4	**0**	**1**	**0**	**0**	**0**	**1**
5	**0**	**1**	**0**	**1**	**0**	**1**
6	**0**	**1**	**1**	**0**	**1**	**0**
7	**0**	**1**	**1**	**1**	**1**	**0**
8	**1**	**0**	**0**	**0**	**0**	**1**
9	**1**	**0**	**0**	**1**	**0**	**1**
10	**1**	**0**	**1**	**0**	**1**	**0**
11	**1**	**0**	**1**	**1**	**1**	**0**
12	**1**	**1**	**0**	**0**	**X**	**X**
13	**1**	**1**	**0**	**1**	**1**	**1**
14	**1**	**1**	**1**	**0**	**X**	**X**
15	**1**	**1**	**1**	**1**	**1**	**1**

Ecuaciones mínimas

Q1=**X1'X2'**Q2 + X1X2Q2 + X2Q1 + X1Q1

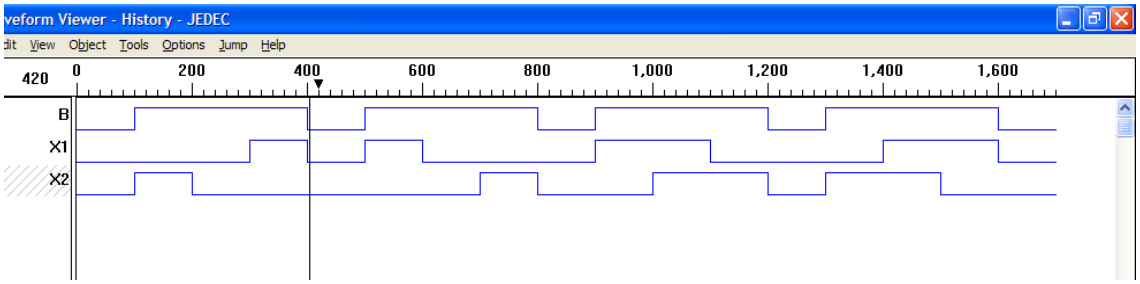
Q2=**X1'X2'**Q2 + X1X2 + X2**Q1'** + X1**Q1'**

Z=Q1+Q2

11.- Elaborar el archivo en formato ABEL-HDL.

```
MODULE tasinc
“entradas
X1,X2 pin 1,2;
“salidas
Q1,Q2,Z pin 19..17 istype ‘com’;
equations
Q1=!X1&!X2&Q2#X1&X2&Q2#X2&Q1#X1&Q1;
Q2=!X1&!X2&Q2#X1&X2#X2&!Q1#X1&!Q1;
Z=Q1#Q2;
test_vectors
([X1,X2]->[Z])
[0,0]->.x.;
[0,1]->.x.;
[0,0]->.x.;
[1,0]->.x.;
[0,0]->.x.;
[1,0]->.x.;
[0,0]->.x.;
[0,1]->.x.;
[0,0]->.x.;
[1,0]->.x.;
[1,1]->.x.;
[0,1]->.x.;
[0,0]->.x.;
[0,1]->.x.;
[1,1]->.x.;
[1,0]->.x.;
[0,0]->.x.;
END
```

12.- Simulación.



Problema 17, Detector de entrada y salida de estacionamiento

Sensor de entrada y salida

.

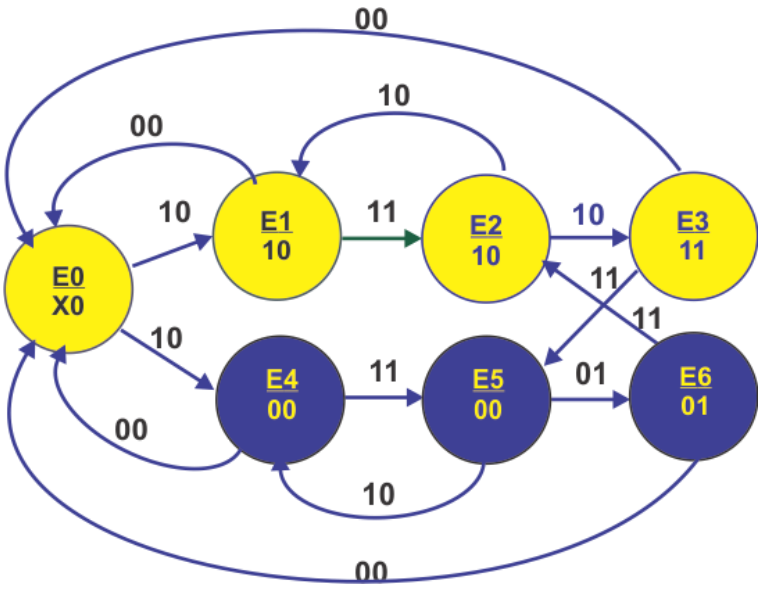
1.- Especificar el Sistema.

Se consideran 2 casos

1 caso de A hacia B			
A	**B**	**E/S**	**Clk**
0	**0**	**X**	**0**
1	**0**	**1**	**0**
1	**1**	**1**	**0**
0	**1**	**1**	**1**
0	**0**	**X**	**0**

2 caso de B hacia A			
A	**B**	**E/S**	**Clk**
0	**0**	**X**	**0**
0	**1**	**0**	**0**
1	**1**	**0**	**0**
1	**0**	**0**	**1**
0	**0**	**X**	**0**

Diagrama de transición



2.- Tabla de Flujo Primitiva.

	A, B				E/S	Clk
	00	**01**	**10**	**11**		
0	**E0**	E4	E1	-	**X**	**0**
1	E0	-	**E1**	E2	**1**	**0**
2	**-**	E3	E1	**E2**	**1**	**0**
3	E0	**E3**	-	E5	**1**	**1**
4	E0	**E4**	-	E5	**0**	**0**
5	-	**E4**	E6	**E5**	**0**	**0**
6	E0	**-**	**E6**	E2	**0**	**1**

3.- Eliminación de estados redundantes.

4.- Mezcla de Filas.

	A, B				E/S	Clk
	00	**01**	**10**	**11**		
0	**E0**	E4	E1	-	**X**	**0**
1,2	E0	E3	**E1**	**E2**	**1**	**0**
3	E0	**E3**	-	**E5**	**1**	**1**
4,5	E0	**E4**	E6	**E5**	**0**	**0**
6	E0	**-**	**E6**	E2	**0**	**1**

	A, B				E/S	Clk
	00	**01**	**10**	**11**		
a	**E0**	E4	E1	-	**X**	**0**
b	E0	E3	**E1**	**E2**	**1**	**0**
c	E0	**E3**	-	**E5**	**1**	**1**
d	E0	**E4**	E6	**E5**	**0**	**0**
e	E0	**-**	**E6**	E2	**0**	**1**

a= E0,: b= E1,E2: c=E3: d=E4,E5: e=E6.

5.- Expandir tabla de salidas.

6.- Tabla de estados internos.

	A, B					
	00	**01**	**10**	**11**	**E/S**	**Clk**
a	**a**	d	b	-	**X**	**0**
b	a	c	**b**	**b**	**1**	**0**
c	a	**c**	-	**d**	**1**	**1**
d	a	**d**	e	**d**	**0**	**0**
e	a	**-**	**e**	b	**0**	**1**

7.- Asignación de Valores a los Estados.

La transición requerida es de

Columna	Transición
00	b→a, c→a, d→a y d→a
01	a→d, b→c
10	a→b, d→e
11	c→d, e→b

Como a requiere de tener transiciones con b, c, d y e para asignar esta posibilidad se propone la asignación con cuatro variables Q3, Q2, Q1 Y Q0

	Q3, Q2			
	00	**01**	**11**	**10**
00		**c**		
01	**d**	**a**	**b**	
11		**e**		
10				

Q1,Q0

No es suficiente ya que con esa asignación no se cumplen las transiciones de b→c, d→e, c→d, e→b por lo que tenemos que recurrir a los estados transitorios R, S, V y T de modo que: b→R→c, d→S→e, c→V→d y e→T→b

Q1,Q0 \ Q3, Q2	00	01	11	10
00	**V**	**c**	**R**	
01	**d**	**a**	**b**	
11	**S**	**e**	**T**	
10				

	A, B					
	00	**01**	**10**	**11**	**E/S**	**Clk**
a	**a**	d	b	-	**X**	**0**
b	a	**R**	**b**	**b**	**1**	**0**
c	a	**c**	-	**V**	**1**	**1**
d	a	**d**	**S**	**d**	**0**	**0**
e	a	-	**e**	**T**	**0**	**1**
R	-	**c**		-	**1**	**0**
S	-	-	**e**	-	**0**	**0**
T	-	-	**-**	b	**0**	**0**
V				d	**1**	**0**

Asignación propuesta

	Q3	Q2	Q1	Q0
a	**0**	**1**	**0**	**1**
b	**1**	**1**	**0**	**1**
c	**0**	**1**	**0**	**0**
d	**0**	**0**	**0**	**1**
e	**0**	**1**	**1**	**1**
R	**1**	**1**	**0**	**0**
S	**0**	**0**	**1**	**1**
T	**1**	**1**	**1**	**1**
V	**0**	**0**	**0**	**0**

8.- Tabla de Estados totales.

Q3 Q2 Q1 Q0	A, B					
	00	01	10	11	E/S	Clk
0101	0101	0001	1101	-	X	0
1101	0101	1100	1101	1101	1	0
0100	0101	0100	-	0000	1	1
0001	0101	0001	0011	0001	0	0
0111	0101	-	0111	1111	0	1
1100	-	0100		-	1	0
0011	-	-	0111	-	0	0
1111	-	-	-	1101	0	0
0000	-	-	-	0001	1	0

	Q3 Q2 Q1 Q0	A, B					
		00	01	10	11	E/S	Clk
0	0000	-	-	-	0001	1	0
1	0001	0101	0001	0011	0001	0	0
2	0010						
3	0011	-	-	0111	-	0	0
4	0100	0101	0100	-	0000	1	1
5	0101	0101	0001	1101	-	X	0
6	0110						
7	0111	0101	-	0111	1111	0	1
8	1000						
9	1001						
10	1010						
11	1011						
12	1100	-	0100		-	1	0
13	1101	0101	1100	1101	1101	1	0
14	1110						
15	1111	-	-	-	1101	0	0

m	A	B	Q3	Q2	Q1	Q0	Q3+	Q2+	Q1+	Q0+	E/S	Clk
0	0	0	0	0	0	0	-	-	-	-	1	0
1	0	0	0	0	0	1	0	1	0	1	0	0
2	0	0	0	0	1	0	-	-	-	-	-	-
3	0	0	0	0	1	1	-	-	-	-	0	0
4	0	0	0	1	0	0	0	1	0	1	1	1
5	0	0	0	1	0	1	0	1	0	1	-	0
6	0	0	0	1	1	0	-	-	-	-	-	-
7	0	0	0	1	1	1	0	1	0	1	0	1
8	0	0	1	0	0	0	-	-	-	-	-	-
9	0	0	1	0	0	1	-	-	-	-	-	-
10	0	0	1	0	1	0	-	-	-	-	-	-
11	0	0	1	0	1	1	-	-	-	-	-	-
12	0	0	1	1	0	0	-	-	-	-	1	0
13	0	0	1	1	0	1	0	1	0	1	1	0
14	0	0	1	1	1	0	-	-	-	-	-	-
15	0	0	1	1	1	1	-	-	-	-	0	0
16	0	1	0	0	0	0	-	-	-	-	1	0
17	0	1	0	0	0	1	0	0	0	1	0	0
18	0	1	0	0	1	0	-	-	-	-	-	-
19	0	1	0	0	1	1	-	-	-	-	0	0
20	0	1	0	1	0	0	0	1	0	0	1	1
21	0	1	0	1	0	1	0	0	0	1	-	0
22	0	1	0	1	1	0	-	-	-	-	-	-
23	0	1	0	1	1	1	-	-	-	-	0	1
24	0	1	1	0	0	0	-	-	-	-	-	-
25	0	1	1	0	0	1	-	-	-	-	-	-
26	0	1	1	0	1	0	-	-	-	-	-	-
27	0	1	1	0	1	1	-	-	-	-	-	-
28	0	1	1	1	0	0	0	1	0	0	1	0
29	0	1	1	1	0	1	1	1	0	0	1	0
30	0	1	1	1	1	0	-	-	-	-	-	-
31	0	1	1	1	1	1	-	-	-	-	0	0
32	1	0	0	0	0	0	-	-	-	-	1	0
33	1	0	0	0	0	1	0	0	1	1	0	0
34	1	0	0	0	1	0	-	-	-	-	-	-
35	1	0	0	0	1	1	0	1	1	1	0	0
36	1	0	0	1	0	0	-	-	-	-	1	1
37	1	0	0	1	0	1	1	1	1	0	-	0
38	1	0	0	1	1	0	-	-	-	-	-	-
39	1	0	0	1	1	1	0	1	1	1	0	1
40	1	0	1	0	0	0	-	-	-	-	-	-
41	1	0	1	0	0	1	-	-	-	-	-	-
42	1	0	1	0	1	0	-	-	-	-	-	-
43	1	0	1	0	1	1	-	-	-	-	-	-

44	**1**	**0**	**1**	**1**	**0**	**0**	**-**	**-**	**-**	**-**	**1**	**0**
45	**1**	**0**	**1**	**1**	**0**	**1**	**1**	**1**	**0**	**1**	**1**	**0**
46	**1**	**0**	**1**	**1**	**1**	**0**	**-**	**-**	**-**	**-**	**-**	**-**
47	**1**	**0**	**1**	**1**	**1**	**1**	**-**	**-**	**-**	**-**	**0**	**0**
48	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**0**
49	**1**	**1**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**0**
50	**1**	**1**	**0**	**0**	**1**	**0**	**-**	**-**	**-**	**-**	**-**	**-**
51	**1**	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**
52	**1**	**1**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**
53	**1**	**1**	**0**	**1**	**0**	**1**	**-**	**-**	**-**	**-**	**-**	**0**
54	**1**	**1**	**0**	**1**	**1**	**0**	**-**	**-**	**-**	**-**	**-**	**-**
55	**1**	**1**	**0**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**1**
56	**1**	**1**	**1**	**0**	**0**	**0**	**-**	**-**	**-**	**-**	**-**	**-**
57	**1**	**1**	**1**	**0**	**0**	**1**	**-**	**-**	**-**	**-**	**-**	**-**
58	**1**	**1**	**1**	**0**	**1**	**0**	**-**	**-**	**-**	**-**	**-**	**-**
59	**1**	**1**	**1**	**0**	**1**	**1**	**-**	**-**	**-**	**-**	**-**	**-**
60	**1**	**1**	**1**	**1**	**0**	**0**	**-**	**-**	**-**	**-**	**1**	**0**
61	**1**	**1**	**1**	**1**	**0**	**1**	**1**	**1**	**0**	**1**	**1**	**0**
62	**1**	**1**	**1**	**1**	**1**	**0**	**-**	**-**	**-**	**-**	**-**	**-**
63	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**1**	**0**	**0**

10.- Obtención de las ecuaciones por medio de minimización.

Q3M= B Q3 Q0 + A Q2 **Q1'**Q0 + B Q1

Q3M = B Q3 Q0 + A B'Q2 Q1' + B Q1

Q2M = A'B' + A'Q0' + Q3 + Q1 + B'Q2

Q1M = A B'Q2' + A Q3'Q1

Q0M = B' + Q3'Q0 + Q2' + A Q3

Q0M = B' + Q3'Q0 + Q2' + A Q0

ES = Q0' + Q2 Q1'

ES = **Q0'** + Q3 Q1'

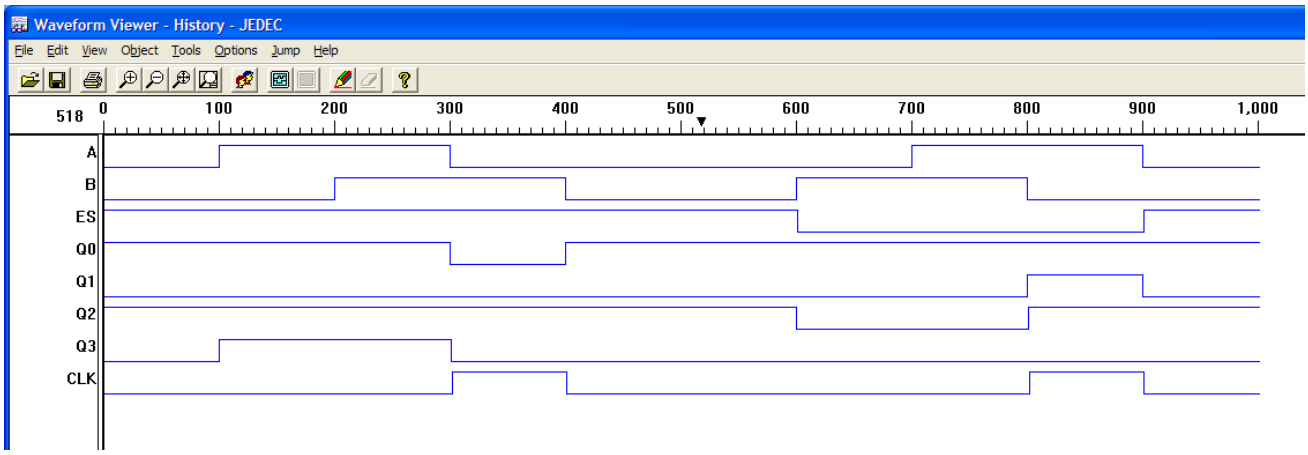
CLK = Q3'Q2 Q0' + Q3'Q2 Q1

11.- Archivo en formato ABEL-HDL.

```
 MODULE ENTSAL
 "Entradas
 A,B pin 2,3;
 "Salidas
 Q3,Q2,Q1,Q0,ES,CLK  pin  19..14  istype
'com';
 Equations
 "Q3M  =  B Q3 Q0 + B Q1 + A B'Q2 Q1'
 Q3=B&Q3&Q0#B&Q1#A&!B&Q2&!Q1;
 "Q2M  =  A'B' + A'Q0' + Q3 + Q1 + B'Q2
 Q2=!A&!B#!A&!Q0#Q3#Q1#!B&Q2;
 "Q1M  =  A B'Q2' + A Q3'Q1
 Q1=A&!B&!Q2#B&!Q3&Q1;
 "Q0M  =  B' + Q3'Q0 + Q2' + A Q3
  Q0=!B#!Q3&Q0#!Q2#A&Q3;
 "ES   =  Q0' + Q2 Q1'
 ES=!Q0#Q2&!Q1;
 "CLK  =  Q3'Q2 Q0' + Q3'Q2 Q1
 CLK=!Q3&Q2&!Q0#!Q3&Q2&Q1;

test_vectors
([A,B]->[ES,CLK])
[0,0]->[.x.,.x.];
[1,0]->[.x.,.x.];
[1,1]->[.x.,.x.];
[0,1]->[.x.,.x.];
[0,0]->[.x.,.x.];
[0,0]->[.x.,.x.];
[0,1]->[.x.,.x.];
[1,1]->[.x.,.x.];
[1,0]->[.x.,.x.];
[0,0]->[.x.,.x.];
END
```

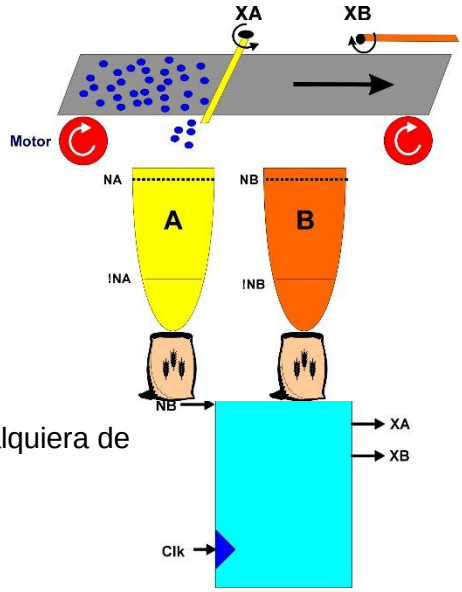
12 Simulaciones.



13.- Representación gráfica.

14.- Implementación

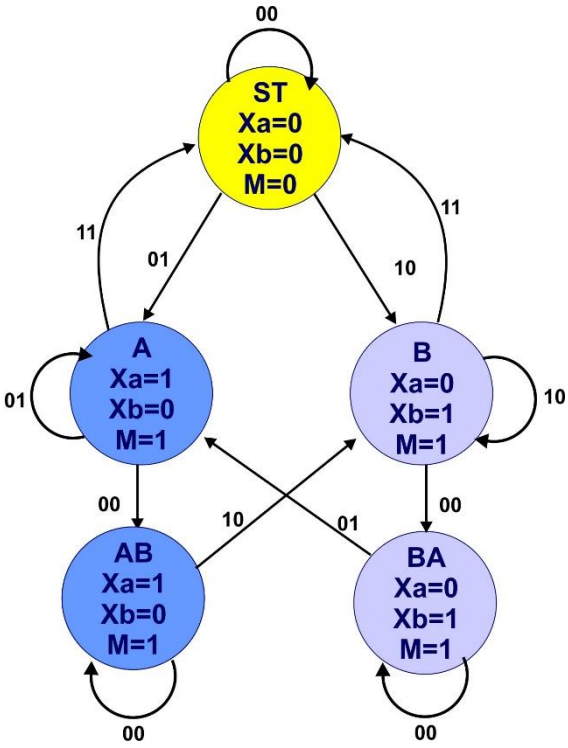
Problema 18, Dos tolvas de Ensacado



Se desea mantener llenas de material dos tolvas de ensacado (tolvas para el llenado se sacos) llamadas **A** y **B,** las cuales son alimentadas por una banda transportadora que pasa por encima de las tolvas y por medio de desviadores del material (**XA** y **XB**) se alimentan las tolvas, siguiendo el orden en que se hallan vaciado de acuerdo a los sensores de Nivel alto y bajo de cada tolva (**NA, ¡NA, NB,¡NB**).

En el caso de que las tolvas se encuentren llenas el motor de la banda transportadora deberá de parar y arrancar cuando cualquiera de las tolvas se vacíe.

1.-Especificar el Sistema.



2.- Tabla de Flujo Primitiva.

	Entradas Na, Nb				Salidas		
Hileras	00	01	11	10	Xa	Xb	M
1	AB	A	**ST**	B	0	0	0
2	AB	**A**	ST	-	1	0	1
3	BA	-	ST	**B**	0	1	1
4	**AB**	-	-	B	1	0	1
5	**BA**	A	-	-	0	1	1

3.-Reducción de Estados.

No es posible la reducción de estados.

4.- Mezcla de Filas y 5.- Expansión de la tabla de Salidas.

	Entradas Na, Nb				Salidas Xa, Xb, M			
Hileras	00	01	11	10	00	01	11	10
1, 2, 4	**AB**	**A**	**ST**	B	101	101	000	---
3, 5	**BA**	A	ST	**B**	011			011

6.-Tabla de Estados Internos.

7.- Asignación de los Valores a los Estados.

8.- Tabla de Estados.

9.- Completar Tabla de Salidas.

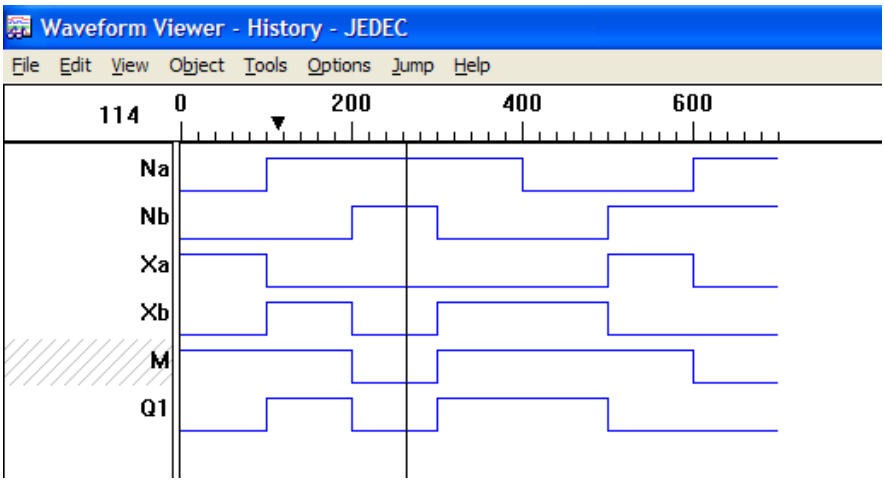
10.- Obtención de las ecuaciones por medio de minimización.

11.- Elaborar el Archivo en formato ABEL-HDL.

```
MODULE dtolvas
“Entradas
Na,Nb pin 1,2;
“Salidas
Q1,Xa,Xb,M pin 19..16 istype ‘com’;
equations
Q1=!Nb&Q1#Na&!Nb;
Xa= ¡Na&!Q1;
Xb= !Nb&Q1;
M= !Na#!Nb;

test_vectors
([Na,Nb]->[Xa,Xb,M])
[0,0]->[.x.,.x.,.x.];
[1,0]->[.x.,.x.,.x.];
[1,1]->[.x.,.x.,.x.];
[1,0]->[.x.,.x.,.x.];
[0,0]->[.x.,.x.,.x.];
[0,1]->[.x.,.x.,.x.];
[1,1]->[.x.,.x.,.x.];
END
```

12.- Simulación

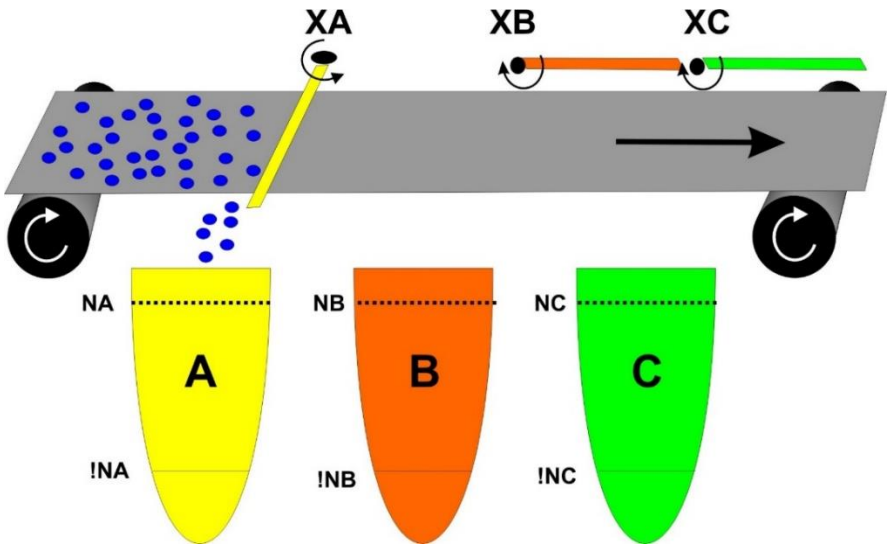


13.- Representación

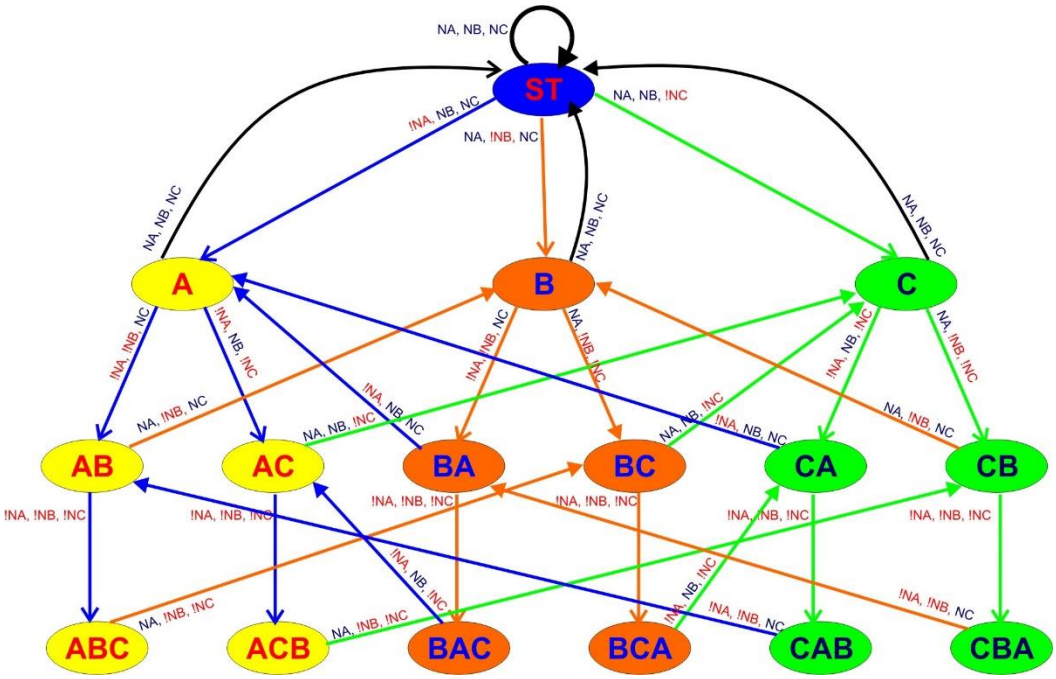
Problema 19, tres tolvas de ensacado

Diseñe, efectúe la simulación y construya un prototipo de un sistema digital secuencial asíncrono, para mantener llenas de material tres tolvas de ensacado (tolvas para el llenado de sacos) llamadas A, B y C, las cuales son alimentadas por una banda transportadora que pasa por encima de las tolvas y por medio de desviadores del material (XA, XB y XC) como se muestra en la figura, el control de llenado debe de alimentar solo una tolva a la vez y no cambiar el desviador de la tolva hasta llenarla, **el orden estricto del llenado es dando prioridad a la tolva que se halla vaciado primero.**

El nivel de las tolvas se detecta por medio de sensores de Nivel de cada tolva (NA, NB y NC) en donde un uno significa lleno y un cero significa vacío.
En el caso de que las tolvas se encuentren llenas el motor de la banda transportadora deberá de parar y arrancar cuando cualquiera de las tolvas se vacíe. **Recuerde que en el orden en que se vacíen deben de llenarse.**



1.- Especificar el sistema



2.- Tabla de flujo primitiva

3.- Reducción de Estados.

4.- Mezcla de Filas.

5.- Expandir Tabla de Salidas.

6.- Estados Internos.

	Na, Nb, Nc								Xa, Xb, Xc, M							
	000	**001**	**010**	**011**	**100**	**101**	**110**	**111**	**000**	**001**	**010**	**011**	**100**	**101**	**110**	**111**
a			A					**ST**								0000
b																
c																
d																
e								-								
f								-								
g																

7.- Asignación de Valores a los Estados.

000) ninguna

001) (b,c $\rightarrow$ e) (f $\rightarrow$ a)

011) (a, e $\rightarrow$ d) (b, c $\rightarrow$ f)

010) (b, f $\rightarrow$ e) (c , d $\rightarrow$ g)

110) (e, f $\rightarrow$ a)

111) (d $\rightarrow$ c)

101) (a, c $\rightarrow$ b)

100) (b,c $\rightarrow$ a)

	Q1 Q2			
Q3	**01**	**11**	**10**	**00**
0	b	e	d	a
1	c	$\propto$	β	f

8.- Tabla de Estados Totales.

	000	001	010	011	100	101	110	111	Xa, Xb, Xc, M
a	**a**	**a**	d	**a**		b	c→b	**a**	0 0 0 0
b	**b**	e	f→c	→a	**b**	**b**	C	a	1 0 0 1
c	**c**	e→b	f		**c**	b	**C**	a→b	0 1 0 1
d	**d**		**d**		c→b		c→b		0 0 1 1
e	**e**	**e**	d	a→b					1 0 0 1
f	**f**	a	**f**	a					0 1 0 1
$\propto$					c		c		0 0 1 1
β					$\propto$		$\propto$		

Q1,Q2	Na, Nb, Nc								Xa, Xb, Xc, M							
	000	**001**	**010**	**011**	**100**	**101**	**110**	**111**	**000**	**001**	**010**	**011**	**100**	**101**	**110**	**111**
000	**000**	**000**	**100**	000	---	010	010	000	1001	1001	1001	1001	----	0-01	00--	0000
001	**001**	000	001	000					0011	-0-1	0011	-0-1				
010	**010**	110	011	**000**	**010**	**010**	011	000	0101	0101	0—1	--01	0101	0101	0011	0000
011	**011**	010	001		**001**	010	**001**	010	0011	0101	0011		0011	0—1	0011	00--
100	**100**		**100**		101		101		1001		1001		0101		0011	
101					111		111						0101		0011	
110	**110**	**110**	100	010					0101	0101	--01	0101				
111					011		011						0101		0011	

9.- Completar la Tabla de Salidas.

10.- Obtención de las ecuaciones por medio de minimización.

Q1+ = **Na'Nb'**Nc Q2 **Q3'** + **Na'**Nb **Nc'Q2'Q3'** + **Nc'**Q1 **Q3'** + Q1 **Q2'**

Q2+=Nb'Q2+**Nc'Q1'**Q2 Q3'+ Nc Q1 +Na N**c'Q1'** +Na Q3 +Na**Nb'Q1'**

Q3+ = **Nc'**Q3 + Nb **Nc'Q1'**Q2 + Na Q1

Xa = **Na'Q2'Q3'**

Xb = Na **Nb'**Q1 + **Na'**Nc Q2 + **Nb'**Q2 **Q3'**

Xc = **Nc'Q1'**Q3 + Na Nb **Nc'**

M = **Na'** + **Nb'** + **Nc'**

11.- Elaborar el Archivo en formato ABEL-HDL.

12.- Simulación.

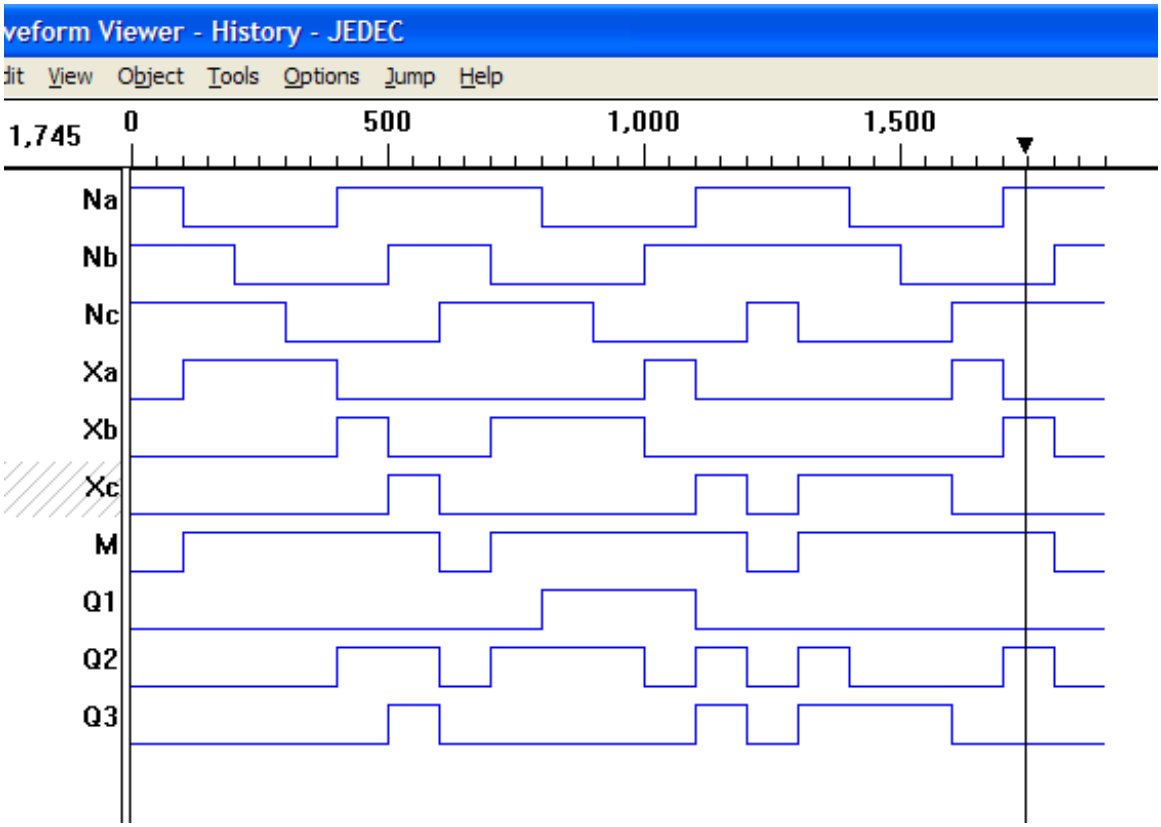


diagrama escalera

13.- Representación.

Diagrama de transición

tabla de flujo primitiva

Problema 20, Concurso de 2 participantes

Diseño efectué la simulación y construya un prototipo de un sistema secuencial asíncrono de control de luces para un concurso de preguntas y respuestas.

Descripción del problema:

Se desea diseñar un sistema secuencial asíncrono que controle el encendido de dos focos solo uno a la vez, denominados "Foco A" y "Foco B", en un concurso de preguntas y respuestas.

El sistema estará compuesto por tres botones de entrada: "A", "B" y "R", que representan a dos concursantes y al conductor del concurso R, respectivamente.

El funcionamiento del sistema es el siguiente:

Cuando el conductor "R" realiza una pregunta, el sistema se encuentra en estado de espera para recibir una respuesta de los concursantes.

Cuando uno de los concursantes (ya sea "A" o "B") tenga la respuesta, presionará su respectivo botón.

Es importante mencionar que no se contempla la posibilidad de que los dos botones se opriman simultáneamente, siempre habrá uno de los tres botones que se oprimirá primero.

El sistema deberá detectar cuál de los dos participantes presionó primero el botón y encenderá el foco correspondiente a ese concursante ("Foco A" si fue el botón "A", o "Foco B" si fue el botón "B"). La luz del foco se mantendrá encendida hasta que el conductor "R" oprima su botón de reset.

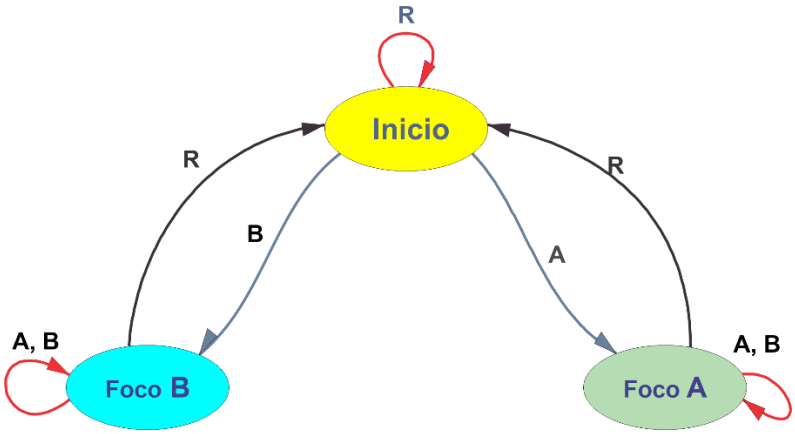
Después de que el conductor "R" haya presionado su botón de reset, el sistema se preparará para la siguiente pregunta y volverá al estado de espera.

Objetivo:

El objetivo es diseñar y simular un sistema secuencial asíncrono que cumpla con las condiciones mencionadas y que sea capaz de controlar adecuadamente la iluminación de los focos en función de las respuestas de los concursantes y las acciones del conductor del concurso.

.

1.- Especificar el sistema



2.- Tabla de Flujo Primitiva.

RAB	000	001	011	010	1XX	FA	FB
0	**E_0**	E5	X	E_1	E_0	0	0
1	E2	X	E3	**E_1**	E_0	1	0
2	**E2**	**E4**	X	E1	E_0	1	0
3	X	E4	**E3**	E1	E_0	1	0
4	E2	**E4**	E3	X	E_0	1	0
5	E6	**E5**	E7	X	E_0	0	1
6	**E6**	**E5**	X	**E8**	E_0	0	1
7	X	**E5**	**E7**	**E8**	E_0	0	1
8	E6	X	E7	**E8**	E_0	0	1

3.- Mezcla de filas

ABR	000	001	011	010	1XX	FA	FB
0	**E_0**	E5	X	E_1	E_0	0	0
1,2,3,4	**E2**	**E4**	**E3**	**E_1**	E_0	1	0
5,6,7,8	**E6**	**E5**	**E7**	**E8**	E_0	0	1

ABR	000	001	011	010	1XX	FA	FB
00	00	10	X	01	00	0	0
01	01	01	01	01	00	1	0
10	10	10	10	10	00	0	1

Tabla de estado siguiente

m	R	Q1	Q0	A	B	Q1+	Q0+	FA	FB
0	0	0	0	0	0	**0**	**0**	**0**	**0**
1	0	0	0	0	1	**1**	**0**	**0**	**0**
2	0	0	0	1	0	**0**	**1**	**0**	**0**
3	0	0	0	1	1	**X**	**X**	**X**	**X**
4	0	0	1	0	0	**0**	**1**	**1**	**0**
5	0	0	1	0	1	**0**	**1**	**1**	**0**
6	0	0	1	1	0	**0**	**1**	**1**	**0**
7	0	0	1	1	1	**0**	**1**	**1**	**0**
8	0	1	0	0	0	**1**	**0**	**0**	**1**
9	0	1	0	0	1	**1**	**0**	**0**	**1**
10	0	1	0	1	0	**1**	**0**	**0**	**1**
11	0	1	0	1	1	**1**	**0**	**0**	**1**
12	0	1	1	0	0	**X**	**X**	**X**	**X**
13	0	1	1	0	1	**X**	**X**	**X**	**X**
14	0	1	1	1	0	**X**	**X**	**X**	**X**
15	0	1	1	1	1	**X**	**X**	**X**	**X**
16	1	0	0	0	0	**0**	**0**	**0**	**0**
	1								
27	1	0	0	0	0	**0**	**0**	**0**	**0**
28	1	1	1	0	0	**X**	**X**	**X**	**X**
29	1	1	1	0	1	**X**	**X**	**X**	**X**
30	1	1	1	1	0	**X**	**X**	**X**	**X**
31	1	1	1	1	1	**X**	**X**	**X**	**X**

Ecuaciones mínimas

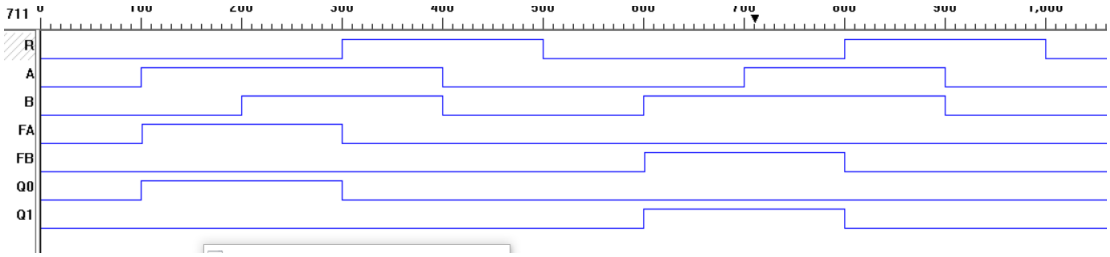
$$Q1 = \mathbf{R'Q0'}B + \mathbf{R'}Q1, \quad Q0 = \mathbf{R'Q1'}A + \mathbf{R'}Q0$$

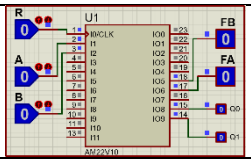
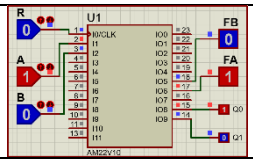
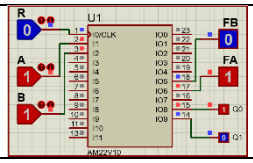
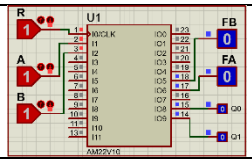
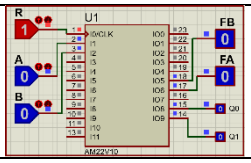
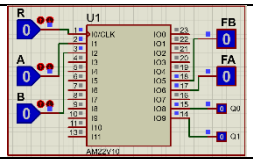
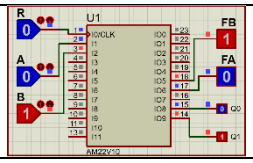
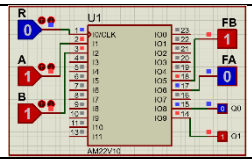
$$FA = \mathbf{R'}Q0, \quad FB = R'Q1$$

Código ABEL-HDL

```
MODULE DOSNUEVE
"15 DE MAYO 2019
R,A,B PIN 1..3;
Q1,Q0,FA,FB PIN 14,15, 17,18 ISTYPE 'COM';
EQUATIONS
"Q1= R'Q0'B + R'Q1
Q1=!R&!Q0&B#!R&Q1;
"Q0+=  R'Q1'A + R'Q0
Q0=!R&!Q1&A#!R&Q0;
"FA   =  R'Q0
FA=!R&Q0;
"FB   =  R'Q1
FB=!R&Q1;
TEST_VECTORS
([R,A,B]->[Q1,Q0,FA,FB])
[0,0,0]->[.X.,.X.,.X.,.X.];
[0,1,0]->[.X.,.X.,.X.,.X.];
[0,1,1]->[.X.,.X.,.X.,.X.];
[1,1,1]->[.X.,.X.,.X.,.X.];
[1,0,0]->[.X.,.X.,.X.,.X.];
[0,0,0]->[.X.,.X.,.X.,.X.];
[0,0,1]->[.X.,.X.,.X.,.X.];
[0,1,1]->[.X.,.X.,.X.,.X.];
[1,1,1]->[.X.,.X.,.X.,.X.];
[1,0,0]->[.X.,.X.,.X.,.X.];
[0,0,0]->[.X.,.X.,.X.,.X.];
END
```

Simulaciones



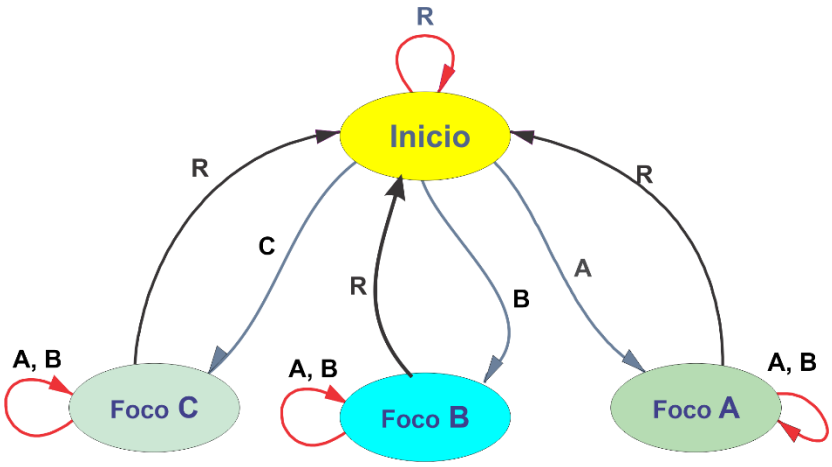
0	1	2	3
			
4	**5**	**6**	**7**
			

Problema 21, Concurso de 3 participantes

Diseñe, efectúe la simulación de un sistema Secuencial asíncrono, que contenga cuatro botones de entrada llamados A, B, C y R y tres salidas llamadas Foco A, Foco B y Foco C.

Los botones A, B y C, corresponden a tres concursantes y el botón R (Reset) al conductor de un concurso de preguntas y respuestas, en donde después de haber terminado de hacer la pregunta el sistema detectará cuál de los tres participantes oprimió primero el botón y encender la luz correspondiente y mantenerse encendida hasta que el conductor R oprima su botón, y el sistema estará listo para la siguiente pregunta.

1.- Especificar el sistema



2.- Tabla de flujo primitiva

RABC	0000	0001	0011	0010	0100	0101	0111	0110	1XXX	FA	FB	FC
0	**E_0**	E3	**X**	E2	E_1	**X**	**X**	**X**	**E_0**	**0**	**0**	
1	E1	E1	E1	E1	**E1**	E1	E1	E1	**E_0**	**1**	**0**	**0**
2	E2	E2		**E2**	E2	E2	E2	E2	**E_0**	**0**	**1**	**0**
3	E3	**E3**	E3	E3	E3	E3	E3	E3	**E_0**	**0**	**0**	**1**

RABC	0000	0001	0011	0010	0100	0101	0111	0110	1XXX	FA	FB	FC
a	$\underline{E_0}$	E3	X	E2	E_1	X	X	X	E_0	0	0	
b	E1	E1	E1	E1	**E1**	E1	E1	E1	E_0	1	0	0
c	E2	E2		**E2**	E2	E2	E2	E2	E_0	0	1	0
d	E3	**E3**	E3	E3	E3	E3	E3	E3	E_0	0	0	1

RABC	0000	0001	0011	0010	0100	0101	0111	0110	1XXX	FA	FB	FC
a	a	d	X	c	b	X	X	X	a	0	0	
b	b	b	b	b	b	b	b	b	a	1	0	0
c	c	c	c	c	c	c	c	c	a	0	1	0
d	d	d	d	d	d	d	d	d	a	0	0	1
	RABC											
Q2 Q1 Q0	0000	0001	0011	0010	0100	0101	0111	0110	1XXX	FA	FB	FC
000	000	100	X	010	001	X	X	X	000	0	0	
001	001	001	001	001	001	001	001	001	000	1	0	0
010	010	010	010	010	010	010	010	010	000	0	1	0
100	100	100	100	100	100	100	100	100	000	0	0	1

Tabla de Estados.

	R	Q2	Q1	Q0	A	B	C	Q2+	Q1+	Q0+	FA	FB	FC
	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	1	1	0	0	0	0	0
	0	0	0	0	0	1	0	0	1	0	0	0	0
	0	0	0	0	1	0	0	0	0	1	0	0	0
	0	0	0	1	0	0	0	0	0	1	1	0	0
	0	0	0	1	0	0	1	0	0	1	1	0	0
	0	0	0	1	0	1	0	0	0	1	1	0	0
	0	0	0	1	0	1	1	0	0	1	1	0	0
	0	0	0	1	1	0	0	0	0	1	1	0	0
	0	0	0	1	1	0	1	0	0	1	1	0	0
	0	0	0	1	1	1	0	0	0	1	1	0	0
	0	0	0	1	1	1	1	0	0	1	1	0	0
	0	0	1	0	0	0	0	0	1	0	0	1	0
	0	0	1	0	0	0	1	0	1	0	0	1	0
	0	0	1	0	0	1	0	0	1	0	0	1	0
	0	0	1	0	0	1	1	0	1	0	0	1	0
	0	0	1	0	1	0	0	0	1	0	0	1	0
	0	0	1	0	1	0	1	0	1	0	0	1	0
	0	0	1	0	1	1	0	0	1	0	0	1	0
	0	0	1	0	1	1	1	0	1	0	0	1	0
	0	1	0	0	0	0	0	1	0	0	0	0	1
	0	1	0	0	0	0	1	1	0	0	0	0	1
	0	1	0	0	0	1	0	1	0	0	0	0	1
	0	1	0	0	0	1	1	1	0	0	0	0	1
	0	1	0	0	1	0	0	1	0	0	0	0	1
	0	1	0	0	1	0	1	1	0	0	0	0	1
	0	1	0	0	1	1	0	1	0	0	0	0	1
	0	1	0	0	1	1	1	1	0	0	0	0	1
	1	X	X	X	X	X	X	0	0	0	0	0	0

Código ABEL-HDL

```
MODULE TRESc
"16 DE MAYO 2019
R,A,B,C PIN 1..4;
Q2,Q1,Q0,FA,FB,FC PIN 14,15,16, 18,19,20 ISTYPE 'DC,COM';
truth_table
([R,Q2,Q1,Q0,A,B,C]->[Q2,Q1,Q0,FA,FB,FC])
[0,0,0,0,0,0,0]->[0,0,0,0,0,0];
[0,0,0,0,0,0,1]->[1,0,0,0,0,0];
[0,0,0,0,0,1,0]->[0,1,0,0,0,0];
[0,0,0,0,1,0,0]->[0,0,1,0,0,0];
[0,0,0,1,0,0,0]->[0,0,1,1,0,0];
[0,0,0,1,0,0,1]->[0,0,1,1,0,0];
[0,0,0,1,0,1,0]->[0,0,1,1,0,0];
[0,0,0,1,0,1,1]->[0,0,1,1,0,0];
[0,0,0,1,1,0,0]->[0,0,1,1,0,0];
[0,0,0,1,1,0,1]->[0,0,1,1,0,0];
[0,0,0,1,1,1,0]->[0,0,1,1,0,0];
[0,0,1,0,1,1,1]->[0,1,0,0,1,0];
[0,0,1,0,0,0,0]->[0,1,0,0,1,0];
[0,0,1,0,0,0,1]->[0,1,0,0,1,0];
[0,0,1,0,0,1,0]->[0,1,0,0,1,0];
[0,0,1,0,0,1,1]->[0,1,0,0,1,0];
[0,0,1,0,1,0,0]->[0,1,0,0,1,0];
[0,0,1,0,1,0,1]->[0,1,0,0,1,0];
[0,0,1,0,1,1,0]->[0,1,0,0,1,0];
[0,0,1,0,1,1,1]->[0,1,0,0,1,0];
[0,1,0,0,0,0,0]->[1,0,0,0,0,1];
[0,1,0,0,0,0,1]->[1,0,0,0,0,1];
[0,1,0,0,0,1,0]->[1,0,0,0,0,1];
[0,1,0,0,0,1,1]->[1,0,0,0,0,1];
[0,1,0,0,1,0,0]->[1,0,0,0,0,1];
[0,1,0,0,1,0,1]->[1,0,0,0,0,1];
[0,1,0,0,1,1,0]->[1,0,0,0,0,1];
[0,1,0,0,1,1,1]->[1,0,0,0,0,1];
[1,.X.,.X.,.X.,.X.,.X.]->[0,0,0,0,0,0];
```

```
TEST_VECTORS
([R,A,B,C]->[Q2,Q1,Q0,FA,FB,FC])
[0,0,0,0]->[.X.,.X.,.X.,.X.,.X.,.X.];
[0,1,0,0]->[.X.,.X.,.X.,.X.,.X.,.X.];
[0,1,1,0]->[.X.,.X.,.X.,.X.,.X.,.X.];
[1,1,1,1]->[.X.,.X.,.X.,.X.,.X.,.X.];
[1,0,0,1]->[.X.,.X.,.X.,.X.,.X.,.X.];
[1,0,0,1]->[.X.,.X.,.X.,.X.,.X.,.X.];
[1,0,0,0]->[.X.,.X.,.X.,.X.,.X.,.X.];
[0,0,0,0]->[.X.,.X.,.X.,.X.,.X.,.X.];
[0,0,1,0]->[.X.,.X.,.X.,.X.,.X.,.X.];
[0,1,1,0]->[.X.,.X.,.X.,.X.,.X.,.X.];
[1,1,1,1]->[.X.,.X.,.X.,.X.,.X.,.X.];
[1,0,0,1]->[.X.,.X.,.X.,.X.,.X.,.X.];
[1,0,0,1]->[.X.,.X.,.X.,.X.,.X.,.X.];
[1,0,0,0]->[.X.,.X.,.X.,.X.,.X.,.X.];
[0,0,0,0]->[.X.,.X.,.X.,.X.,.X.,.X.];
[0,0,0,1]->[.X.,.X.,.X.,.X.,.X.,.X.];
[0,1,1,1]->[.X.,.X.,.X.,.X.,.X.,.X.];
[1,1,1,1]->[.X.,.X.,.X.,.X.,.X.,.X.];
[1,0,0,1]->[.X.,.X.,.X.,.X.,.X.,.X.];
[1,0,0,1]->[.X.,.X.,.X.,.X.,.X.,.X.];
[1,0,0,0]->[.X.,.X.,.X.,.X.,.X.,.X.];
[0,0,0,0]->[.X.,.X.,.X.,.X.,.X.,.X.];
END
```

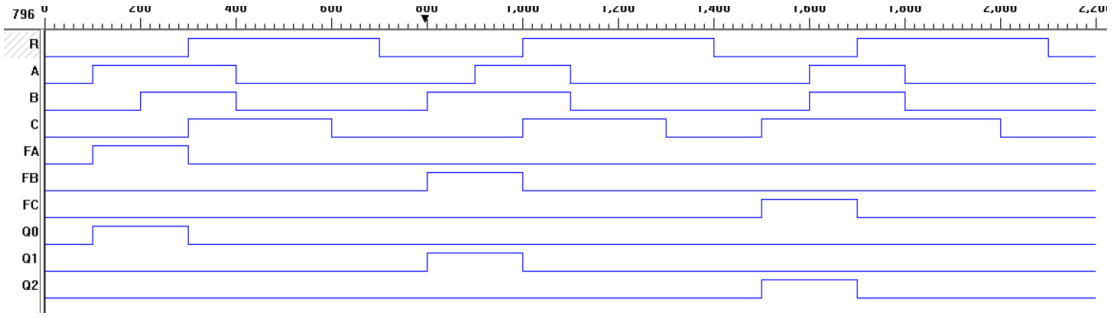
Ecuaciones mínimas

Q2= (C & **!Q0** & **!Q1** & **!R** # Q2 & **!R**);

Q1=(Q1 & **!R** # B & **!Q0** & **!Q2** & **!R**);

Q0= (Q0 & **!R** # A & **!Q1** & **!Q2** & **!R**);

FA = (Q0 & **!R**); , FB = (Q1 & **!R**); , FC = (Q2 & **!R**);



Formula general

Comparación de resultados:

Dos concursantes con R aparte:

Q1	**Q0'**B + Q1	Q0	**Q1'**A + Q0
FA	Q0	FB	Q1

Dos concursantes con R

Q1	B**R'Q0'**+**R'**Q1	Q0	A**R'Q1'** + **R'**Q0
FA	**R'**Q0	FB	**R'**Q1

Tres concursantes con R

Q2	C **R'Q1' Q0'**+**R'**Q2	Q1	B**R'Q2'Q0'**+**R'**Q1	Q0	A**R'Q1' Q2'**+**R'**Q0
FA	**R'**Q0	FB	**R'**Q1	FC	**R'**Q2

Por lo que se concluye que para cuatro concursantes serian:

Q3	D**R'Q2' Q1'Q0'**+**R'**Q3	Q2	C**R'Q3'Q1' Q0'**+ **R'**Q2
Q1	B **R'Q3' Q2' Q0'**+**R'**Q1	Q0	D**R'Q3' Q2' Q1'**+ **R'**Q0
FA	**R'**Q0	FB	**R'**Q1
FC	**R'**Q2	FD	**R'**Q3

Ecuaciones para cinco concursantes siguiendo la formula general

Q4	E**R'Q3'Q2' Q1'Q0'**+ **R'**Q4	Q3	D**R'Q4'Q2' Q1'Q0'**+**R'**Q3
Q2	C**R'Q4'Q3'Q1' Q0'**+**R'**Q2	Q1	B**R' Q4'Q3' Q2' Q0'**+**R'**Q2
Q0	D**R'Q4'Q3' Q2' Q1'**+ **R'**Q0		
FA	**R'**Q0	FB	**R'**Q1
FC	**R'**Q2	FD	**R'**Q3
FE	**R'**Q4		

Y para cinco concursantes seria:

$$Q4 = E\mathbf{R'Q3'Q2'\ Q1'Q0'} + \mathbf{R'}Q4,$$

$$Q3 = D\mathbf{R'Q4'Q2'\ Q1'Q0'} + \mathbf{R'}Q3,$$

$$Q2 = C\mathbf{R'Q4'Q3'Q1'\ Q0'} + \mathbf{R'}Q2,$$

$$Q1 = B\mathbf{R'\ Q4'Q3'\ Q2'\ Q0'} + \mathbf{R'}Q2,$$

$$Q0 = D\mathbf{R'Q4'Q3'\ Q2'\ Q1'} + \mathbf{R'}Q0,$$

$$FA = \mathbf{R'}Q0,\ FB = \mathbf{R'}Q1,\ FC = \mathbf{R'}Q2,\ FD = \mathbf{R'}Q3,\ FE = \mathbf{R'}Q4$$

Código ABEL-HDL

```
MODULE Tcincoc
"16 DE MAYO 2019
R,A,B,C,D,E PIN 1..6;
Q4,Q3,Q2,Q1,Q0,FA,FB,FC,FD,FE PIN 14..23 ISTYPE 'DC,COM';
equations
Q4=E&!R &!Q3 &!Q2 &!Q1& !Q0#!R&Q4;
Q3=D&!R&!Q4&!Q2&!Q1&!Q0#!R&Q3;
 Q2=C&!R&!Q4&!Q3&!Q1&!Q0#!R&Q2;
Q1=B&!R&!Q4&!Q3&!Q2&!Q0#!R&Q1;
Q0=A&!R&!Q4&!Q3&!Q2&!Q1#!R&Q0;
FA=!R&Q0;FB=!R&Q1;FC=!R&Q2;FD=!R&Q3;FE=!R&Q4;
END
```

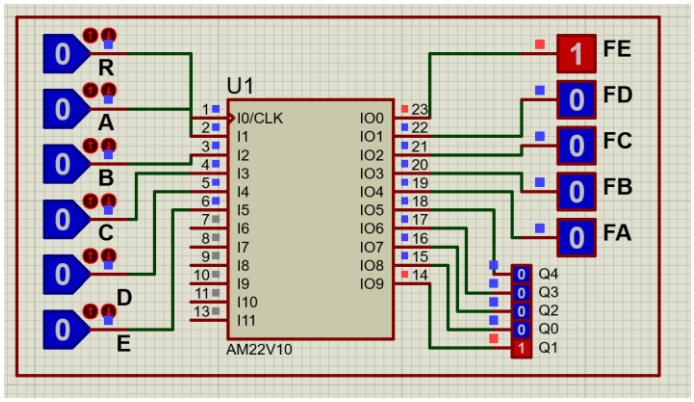


Ilustración 1

Y para seis concursantes las ecuaciones serian:

Q5 =F**R'Q4'Q3'Q2' Q1'Q0'**+ **R'**Q5

Q4 =E**R'Q5'Q3'Q2' Q1'Q0'**+ **R'**Q4

Q3= D**R'Q5'Q4'Q2' Q1'Q0'**+**R'**Q3,

Q2= C**R'Q5'Q4'Q3'Q1' Q0'**+**R'**Q2,

Q1= B**R'Q5'Q4'Q3' Q2' Q0'**+**R'**Q2,

Q0= A**R'Q5'Q4'Q3' Q2' Q1'**+ **R'**Q0,

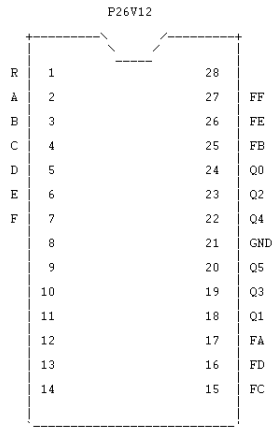
FA = **R'**Q0, FB= **R'**Q1, FC= **R'**Q2, FD= **R'**Q3, FE= **R'**Q4, FF= **R'**Q5

Este problema ya utilizaremos en su implementación en un PLD26V12 o un CPLD o un FPGA.

Código Abel-HDL para seis concursantes

```
MODULE Tseis
"16 DE MAYO 2019
R,A,B,C,D,E, F PIN 1..7;
Q5,Q4,Q3,Q2,Q1,Q0,FA,FB,FC,FD,FE, FF PIN ISTYPE 'DC,COM';
equations

Q5=E&!R &!Q4 &!Q2 &!Q1& !Q0#!R&Q5;
Q4=E&!R &!Q3 &!Q2 &!Q1& !Q0#!R&Q4;
Q3=D&!R&!Q4&!Q2&!Q1&!Q0#!R&Q3;
Q2=C&!R&!Q4&!Q3&!Q1&!Q0#!R&Q2;
Q1=B&!R&!Q4&!Q3&!Q2&!Q0#!R&Q1;
Q0=A&!R&!Q4&!Q3&!Q2&!Q1#!R&Q0;
FA=!R&Q0;FB=!R&Q1;FC=!R&Q2;FD=!R&Q3;FE=!R&Q4;
FF=!R&Q5;
END
```



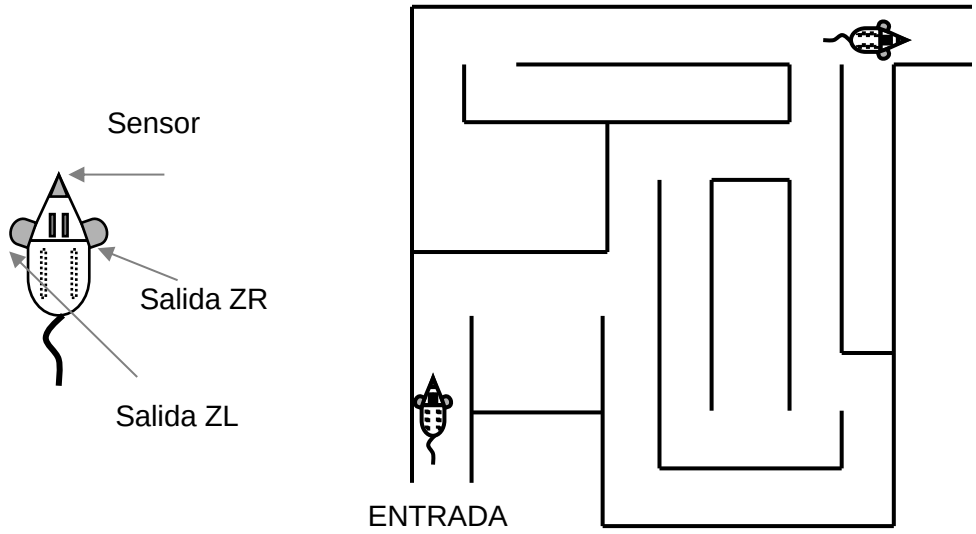
PLD26V12

Problema 22, Laberinto (caso1)

Diseñe el “cerebro” de un ratón. El objetivo es ayudar a que el ratón consiga la salida de un laberinto, tan rápido como sea posible.

Especificaciones:

Se diseñará un ratón que sea capaz de navegar tan rápido como sea posible a través de un laberinto. El laberinto se muestra en la siguiente figura:



El ratón deberá de tener las siguientes habilidades o capacidades:

Un sensor S sobre la nariz del ratón que detecta cuando toca una pared enfrente de él.

El ratón tiene la capacidad para girar a cualquiera de los dos lados ya sea a la izquierda o a la derecha, o en línea recta tan largo o extenso como sea el recorrido hasta que toque otra pared.

Así el ratón tiene dos motores de salida llamados ZL y ZR.

Solo está permitido una acción a la vez como seguir derecho, girar a la derecha o girar a la izquierda como lo indica la siguiente tabla.

Acción	**ZL**	**ZR**
Derecho	**1**	**1**
Giro derecha	**1**	**0**
Giro Izquierda	**0**	**1**

EJEMPLO: Después de un vistazo rápido al laberinto, ha notado una manera rápida para salir de él, es por medio de giros a la izquierda y a la derecha. Así que cuando el ratón alcance una pared, lo primero que hará será girar a la derecha; la próxima vez que alcance una pared, lo que hará será girar a la izquierda, la siguiente vez girará a la derecha, etc. Hasta que alcance la salida. Verifique que esta estrategia en efecto funcione. (Dibuje el camino que seguirá el ratón).

1.-Especificar el Sistema.

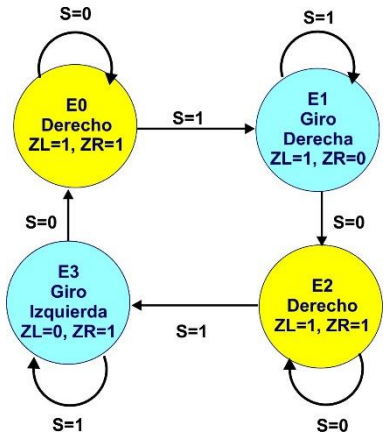


Figura 15 Diagrama de transición

2.- Tabla de Flujo Primitiva.

	Entrada s		Salidas	
	0	**1**	**ZL**	**ZR**
0	**E_0**	E1	**1**	**1**
1	E_2	**E_1**	**1**	**0**
2	**E_2**	E_3	**1**	**1**
3	E_0	**E_3**	**0**	**1**

Figura 16 Tabla de flujo primitiva

3.- Eliminación de Estados Redundantes.

No es posible la reducción de estados.

4.- Mezcla de Filas.

No es posible la mezcla de filas.

5.- Expandir Tabla de Salidas.

No es necesario expandir la tabla de salidas.

6.- Tabla de Estados Internos.

Asignación de variables a los estados

	Entrada s		Salidas		
	0	**1**	**ZL**	**ZR**	
0	**$\underline{E_0}$**	E1	**1**	**1**	**a**
1	E_2	**$\underline{E_1}$**	**1**	**0**	**b**
2	**$\underline{E_2}$**	E_3	**1**	**1**	**c**
3	E_0	**$\underline{E_3}$**	**0**	**1**	**d**

	Entrada S		Salidas	
	0	**1**	**ZL**	**ZR**
a	**<u>a</u>**	b	**1**	**1**
b	c	**<u>b</u>**	**1**	**0**
c	**<u>c</u>**	d	**1**	**1**
d	a	**<u>d</u>**	**0**	**1**

7.- Asignación de valores a los Estados.

Requisitos: (a $\rightarrow$ b), (b $\rightarrow$ c), (c $\rightarrow$ d), (d $\rightarrow$ a),

		Q2	
		0	**1**
Q1	**0**	**a**	**d**
	1	**b**	**c**

Asignación

a =00, b = 01, c = 11, d= 10

8.- Tabla de Estados.

	Entrada S		Salidas	
Q1 Q2	0	1	ZL	ZR
00	00	01	1	1
01	11	01	1	0
11	11	10	1	1
10	00	10	0	1

9.- Completar Tabla de Salidas.

No es necesario completar tabla de salidas.

10.- Obtención de las ecuaciones por medio de minimización.

Tabla

m	H	Q1	Q2	Q1+	Q2+
0	**0**	**0**	**0**	**0**	**0**
1	**0**	**0**	**1**	**1**	**1**
2	**0**	**1**	**0**	**0**	**0**
3	**0**	**1**	**1**	**1**	**1**
4	**1**	**0**	**0**	**0**	**1**
5	**1**	**0**	**1**	**0**	**1**
6	**1**	**1**	**0**	**1**	**0**
7	**1**	**1**	**1**	**1**	**0**

m	Q1	Q2	ZL	ZR
0	**0**	**0**	**1**	**1**
1	**0**	**1**	**1**	**0**
2	**1**	**0**	**0**	**1**
3	**1**	**1**	**1**	**1**

Ecuaciones

$$Q1+ = \mathbf{H'}Q2 + H\,Q1$$

$$Q2+ = \mathbf{H'}Q2 + H\,Q1'$$

$$ZL = \mathbf{Q1'} + Q2$$

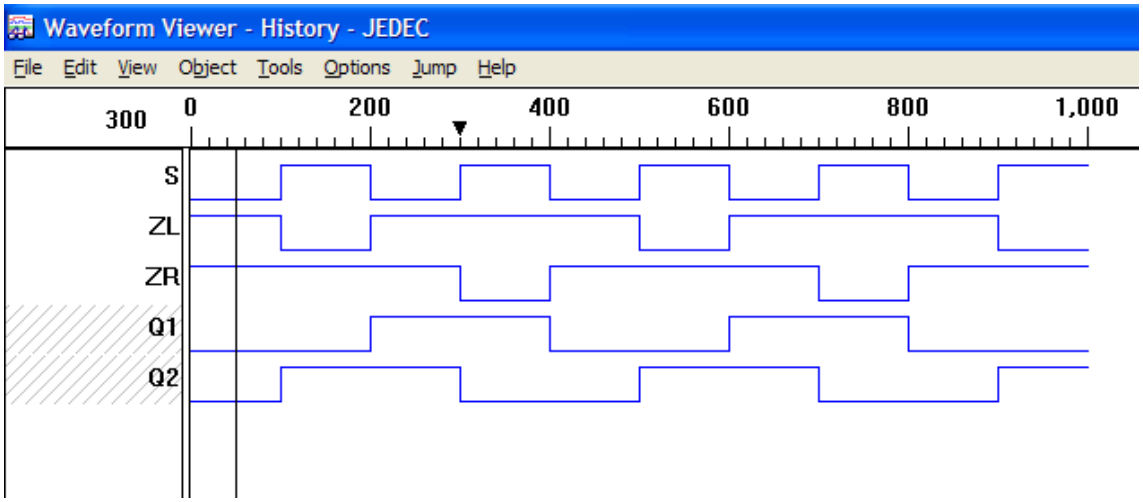
$$ZR = Q1 + \mathbf{Q2'}$$

11.- Archivo en formato ABEL-HDL.

```
MODULE raton
"Entrada
S pin 1;
"Salidas
Q1,Q2,ZL,ZR pin 19..16 istype 'com';
equations
Q1=!S&Q2#S&Q1;
Q2=!S&Q2#S&!Q1;
ZL=Q1#!Q2;
ZR=!Q1#Q2;
```

```
test_vectors
(S->[ZL,ZR])
0->[.x.,.x.];
1->[.x.,.x.];
0->[.x.,.x.];
1->[.x.,.x.];
0->[.x.,.x.];
1->[.x.,.x.];
0->[.x.,.x.];
1->[.x.,.x.];
0->[.x.,.x.];
1->[.x.,.x.];
END
```

12.- Simulación.



13.- Representación.

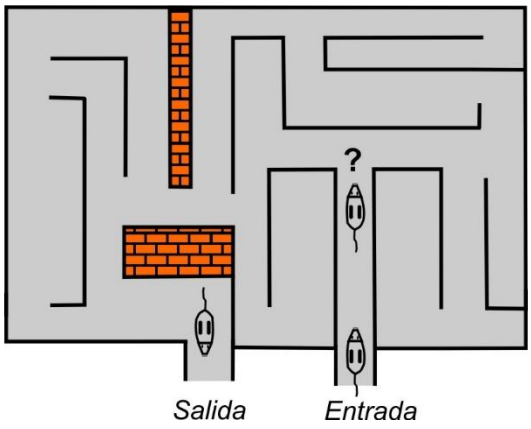
diagrama escalera

Problema 23, Laberinto (caso 2)

Diseñe el “cerebro” de un ratón. El objetivo es ayudar a que el ratón consiga la salida de un laberinto, tan rápido como sea posible.

Especificaciones:

Se diseñará un ratón que sea capaz de navegar tan rápido como sea posible a través de un laberinto. El laberinto se muestra en la siguiente figura:



El ratón deberá de tener las siguientes habilidades o capacidades:

Un sensor S sobre la nariz del ratón que detecta cuando toca una pared enfrente de él.

El ratón tiene la capacidad para girar a cualquiera de los dos lados ya sea a la izquierda o a la derecha, o en línea recta tan largo o extenso como sea el recorrido hasta que toque otra pared.

Así el ratón tiene dos motores de salida llamados ZL y ZR.

Solo está permitido una acción a la vez como seguir derecho, girar a la derecha o girar a la izquierda como lo indica la siguiente tabla.

Acción	**ZL**	**ZR**
Derecho	**1**	**1**
Giro Derecha	**1**	**0**
Giro Izquierda	**0**	**1**

1.- Especificar el Sistema.

2.- Tabla de Flujo Primitiva.

	S		SALIDAS	
	0	**1**	**ZL**	**ZR**
0	**E0**	**E1**	**1**	**1**
1	**E2**	**E1**	**1**	**0**
2	**E2**	**E3**	**1**	**1**
3	**E4**	**E3**	**0**	**1**
4	**E4**	**E5**	**1**	**1**
5	**E0**	**E5**	**0**	**1**

3.- Eliminación de Estados Redundantes. No hay reducción de estados.

4.- Mezcla de Filas. No es posible mezclar filas.

5.- Expandir Tabla de Salidas. No es necesario expandir tabla de salidas.

6.- Tabla de Estados Internos.

	S		SALIDAS	
	0	1	ZL	ZR
a	**E0**	E1	1	1
b	E2	**E1**	1	0
c	**E2**	E3	1	1
d	E4	**E3**	0	1
e	**E4**	E5	1	1
f	E0	**E5**	0	1

	S		SALIDAS	
	0	1	ZL	ZR
a	**a**	b	1	1
b	c	**b**	1	0
c	**c**	d	1	1
d	e	**d**	0	1
e	**e**	f	1	1
f	a	**f**	0	1

7.- Asignación de Valores a los Estados.

Requisitos

0) (b→c) (d→e) (f→a).

1) (a→b) (c→d) (e→f)

Q3	Q1 Q2				
		00	01	11	10
	0	a	-	-	f
	1	b	c	d	e

Asignación: a=000, b=001, c=011, d=111, e=101, f=100

8.- Tabla de Estados.

	S		SALIDAS	
	0	1	ZL	ZR
000	000	001	1	1
001	011	001	1	0
011	011	111	1	1
111	101	111	0	1
101	101	100	1	1
100	000	100	0	1

9.- Completar Tabla de Salidas.

No es necesario completar tabla de salidas.

10.- Obtención de las ecuaciones por medio de minimización.

$$ZL = \mathbf{Q1'} + \mathbf{Q2'}Q3$$

$$ZR = \mathbf{Q3'} + Q2 + Q1$$

$$Q1 = Q2\,S + Q1\,S + Q1\,Q3$$

$$Q2 = \mathbf{Q1'}Q3\,\mathbf{S'} + Q2S$$

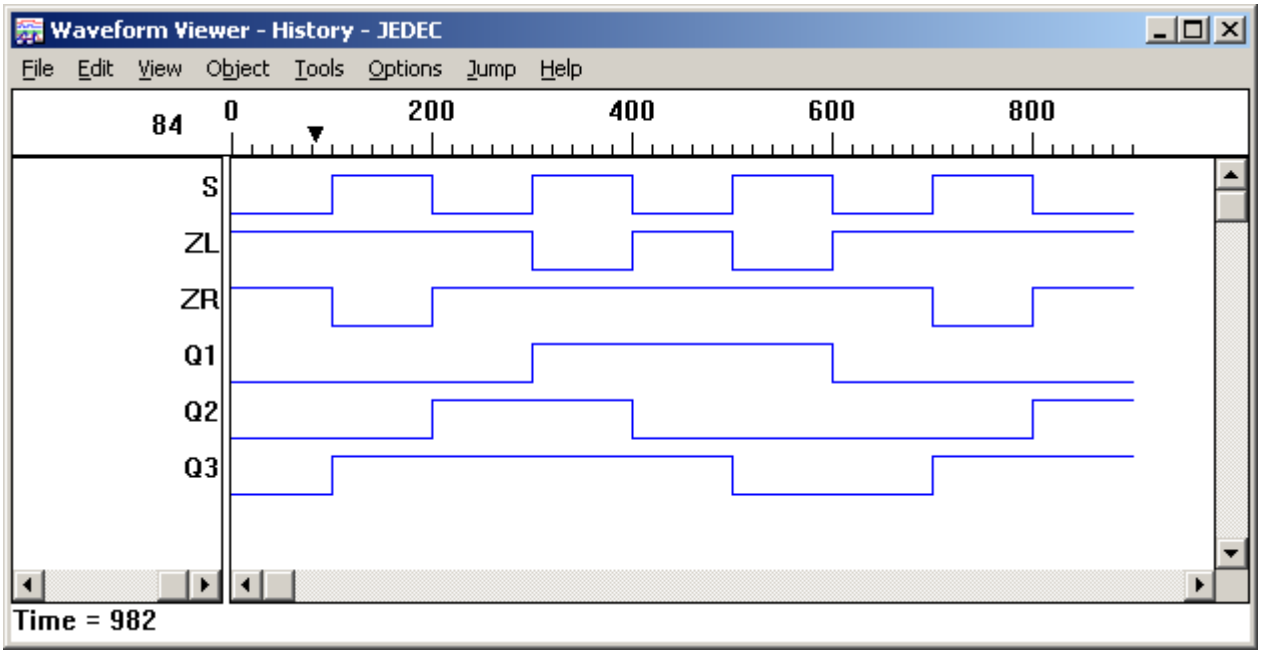
$$Q3 = \mathbf{Q1'}S + Q3\,\mathbf{S'} + Q2$$

11.- Archivo en formato ABEL-HDL.

```
MODULE RATON2
"ZL  =  Q1' + Q2'Q3

"ZR  =  Q3' + Q2 + Q1
"Q1+  =  Q2S + Q1S + Q1 Q3
"Q2+  =  Q1'Q3 S' + Q2 S
"Q3+  =  Q1'S + Q3 S' + Q2
S PIN 1;
Q1,Q2,Q3,ZL,ZR PIN 19..15 ISTYPE 'COM';
EQUATIONS
ZL=!Q1#(!Q2&Q3);
ZR=!Q3#Q2#Q1;
Q1=(Q2&S)#(Q1&Q3)#(Q1&S);
Q2=(!Q1&Q3&!S)#(Q2&S);
Q3=(!Q1&S)#(Q3&!S)#Q2;
TEST_VECTORS
([S]->[Q1,Q2,Q3,ZL,ZR])
 [0]->[.X.,.X.,.X.,.X.,.X.];
 [1]->[.X.,.X.,.X.,.X.,.X.];
   [0]->[.X.,.X.,.X.,.X.,.X.];
 [1]->[.X.,.X.,.X.,.X.,.X.];
 [0]->[.X.,.X.,.X.,.X.,.X.];
 [1]->[.X.,.X.,.X.,.X.,.X.];
 [0]->[.X.,.X.,.X.,.X.,.X.];
 [1]->[.X.,.X.,.X.,.X.,.X.];
 [0]->[.X.,.X.,.X.,.X.,.X.];
END
```

12.- Simulación.

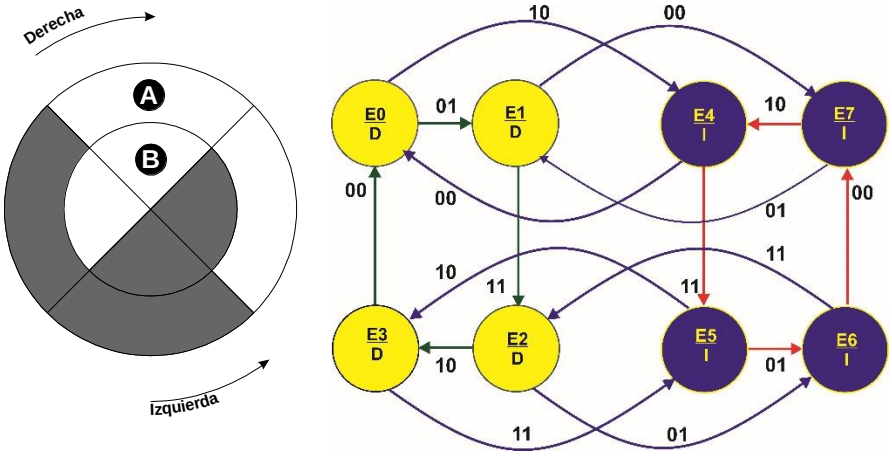


13.- Representación.

Problema 24, Detector de giro

Diseñe un sistema capaz de detectar el sentido de giro de un motor por medio de una salida Z.

1.- Especificar el Sistema.



SensorDiagrama de transición

Diseñe un sistema capaz de detectar el sentido de giro de un motor por medio de una salida Z

Diagrama de transición

Tabla de Flujo primitiva

	00	01	11	10	Z
0	**E0**	E1	-	E4	**D**
1	E7	**E1**	E2	-	**D**
2	-	E6	**E2**	E3	**D**
3	E0	-	E5	**E3**	**D**
4	E0	-	E5	**E4**	**I**
5	**-**	E6	**E5**	E3	**I**
6	E7	**E6**	E2	-	**I**
7	**E7**	E1	-	E4	**I**

Reducción de estados

E1							
E2							
E3							
E4							
E5							
E6							
E7							
	E0	**E1**	**E2**	**E3**	**E4**	**E5**	**E6**

Mezcla de filas

	00	01	11	10	00	01	11	10
0, 4	**E0**	E1	E5	**E4**	**D**			**I**
1, 7	**E7**	**E1**	E2	E4	**I**	**D**		
2, 6	E7	**E6**	**E2**	E3		**I**	**D**	
3, 5	E0	E6	**E5**	**E3**			**I**	**D**

Estados Internos

	00	01	11	10	00	01	11	10
a	**E0**	E1	E5	**E4**	**D**			**I**
b	**E7**	**E1**	E2	E4	**I**	**D**		
c	E7	**E6**	**E2**	E3		**I**	**D**	
d	E0	E6	**E5**	**E3**			**I**	**D**

Substituir: a = E0 y E4, b = E7 y E1, c= E6 y E2, d= E5 y E3

	00	01	11	10	00	01	11	10
a	**a**	b	d	**a**	**D**			**I**
b	**b**	**b**	c	a	**I**	**D**		
c	b	**c**	**c**	d		**I**	**D**	
d	a	c	**d**	**d**			**I**	**D**

Para la asignación de valores a los estados es necesario cumplir con:

Para la columna

00) $c \rightarrow b$ y $d \rightarrow a$

01) $a \rightarrow b$ y $d \rightarrow c$

11) $a \rightarrow d$ y $b \rightarrow c$

10) $b \rightarrow a$ y $c \rightarrow d$

	0	1
0	a	d
1	b	c

	Q1	Q0
a	0	0
b	0	1
c	1	1
d	1	0

	00	**01**	**11**	**10**	**00**	**01**	**11**	**10**
00	**00**	01	10	**00**	**D**			**I**
01	**01**	**01**	11	00	**I**	**D**		
11	b	**11**	**11**	10		**I**	**D**	
10	00	11	**10**	**10**			**I**	**D**

	00	**01**	**11**	**10**	**00**	**01**	**11**	**10**
00	**00**	01	10	**00**	**D**			**I**
01	**01**	**01**	11	00	**I**	**D**		
11	b	**11**	**11**	10		**I**	**D**	
10	00	11	**10**	**10**			**I**	**D**

Completar salidas

	00	**01**	**11**	**10**	**00**	**01**	**11**	**10**
00	**00**	01	10	**00**	**D**	**D**	**I**	**I**
01	**01**	**01**	11	00	**I**	**D**	**D**	**I**
11	b	**11**	**11**	10	**I**	**I**	**D**	**D**
10	00	11	**10**	**10**	**D**	**I**	**I**	**D**

So D=0 y I =1

	00	01	11	10	00	01	11	10
00	**00**	01	10	**00**	**0**	**0**	**1**	**1**
01	**01**	**01**	11	00	**1**	**0**	**0**	**1**
11	b	**11**	**11**	10	**1**	**1**	**0**	**0**
10	00	11	**10**	**10**	**0**	**1**	**1**	**0**

Obteniendo las ecuaciones

A B

Q1⁺	00	01	11	10
00	**0**	**0**	**1**	**0**
01	**0**	**0**	**1**	**0**
11	**0**	**1**	**1**	**1**
10	**0**	**1**	**1**	**1**

Q1Q2

$Q1^+ = A\,B + A\,Q1 + B\,Q1$

A B

Q2⁺	00	01	11	10
00	**0**	**1**	**0**	**0**
01	**1**	**1**	**1**	**0**
11	**1**	**1**	**1**	**0**
10	**0**	**1**	**0**	**0**

Q1Q2

Q2+ = A' B + A' Q2 + B Q2

S / Q1Q2	A B = 00	01	11	10
00	**0**	**0**	**1**	**1**
01	**1**	**0**	**0**	**1**
11	**1**	**1**	**0**	**0**
10	**0**	**1**	**1**	**0**

Ecuaciones

S = A B Q2' + A B'Q1' + A'B Q1 + A'B'Q2

2.- Tabla de Flujo Primitiva.

	00	**01**	**11**	**10**	**Z**
	1	2	-	3	**D**
	1	**2**	4	-	**I**
	6	-	5	**3**	**D**
	-	7	**4**	8	**I**
	-	7	**5**	8	**D**
	6	2	-	3	**I**
	1	**7**	4	-	**D**
	6	-	5	**8**	**I**

Comparación de estados

1=2, 3=6, 4=7, 5=8

Tabla de Estados

	00	01	11	10	00	01	11	10
a	**1**	**2**	3	4		1		
b	**6**	2	5	**3**				
c	1	**7**	**4**	8				
d	6	7	**5**	**8**				

Ecuaciones

$$FQ1 = A\,B + A\,Q1 + B\,Q1$$

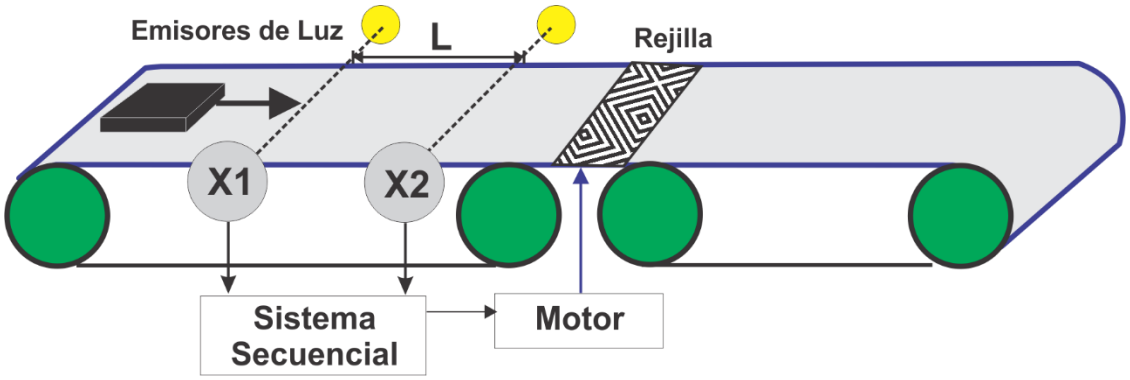
$$FQ2 = A\,Q2 + A\,\mathbf{B'} + \mathbf{B'}\,Q2$$

$$Z = A\,B\,\mathbf{Q2'} + A\,\mathbf{B'}Q1 + \mathbf{A'}B\,\mathbf{Q1'} + \mathbf{A'B'}Q2$$

$$Z = A\,Q1\,\mathbf{Q2'} + B\,\mathbf{Q1'Q2' + B'}Q1\,Q2 + \mathbf{A'Q1'}Q2$$

Problema 25, Medidor de longitud

El producto final de una fabricación son barras metálicas cuya longitud ha de ser inferior o igual a L. Para hacer la selección del producto terminado se utiliza el sistema mostrado en la siguiente figura:



Está constituido por una cinta transportadora que hace pasar las barras entre dos detectores fotoeléctricos separados por una distancia L y constituidos por un emisor y un receptor de luz. La salida de los receptores adopta dos niveles de tensión diferenciados según esté o no una barra situada entre él y su emisor respectivo. Se asigna por convenio el estado lógico 1 a la salida cuando la barra está situada delante del detector y el estado 0 en el caso contrario.

Después del segundo detector existe una trampilla accionada por un motor M. Si la barra tiene una longitud mayor que L, se ha de activar M y abrir la trampilla para dejar caer la barra; en caso contrario no ha de excitarse M. Una vez comprobada la barra, el motor M ha de volver a desactivarse y el sistema quedará preparado para una nueva detección.

Diseñar un sistema secuencial asíncrono cuyas entradas sean las salidas de los detectores, que denominaremos *x*1 y *x*2, y cuya salida *z* accione el motor M al ponerse en estado 1. Suponer que la distancia que separa dos barras sometidas a verificación es tal que nunca podrá entrar una en la zona de detección mientras se está comprobando la anterior.

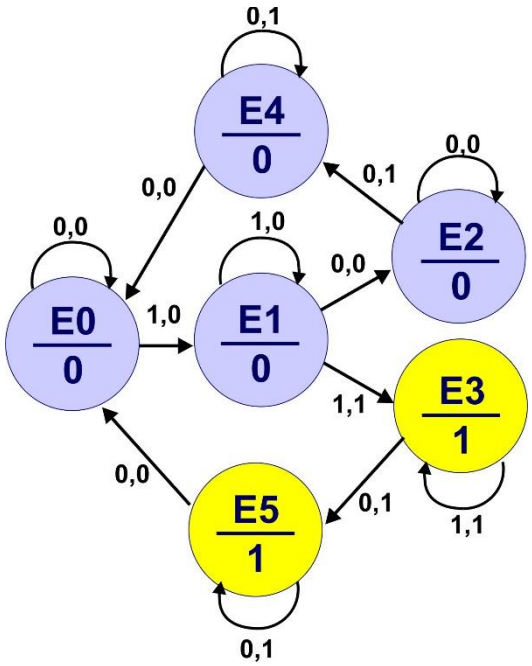
1.- Especificar el Sistema.

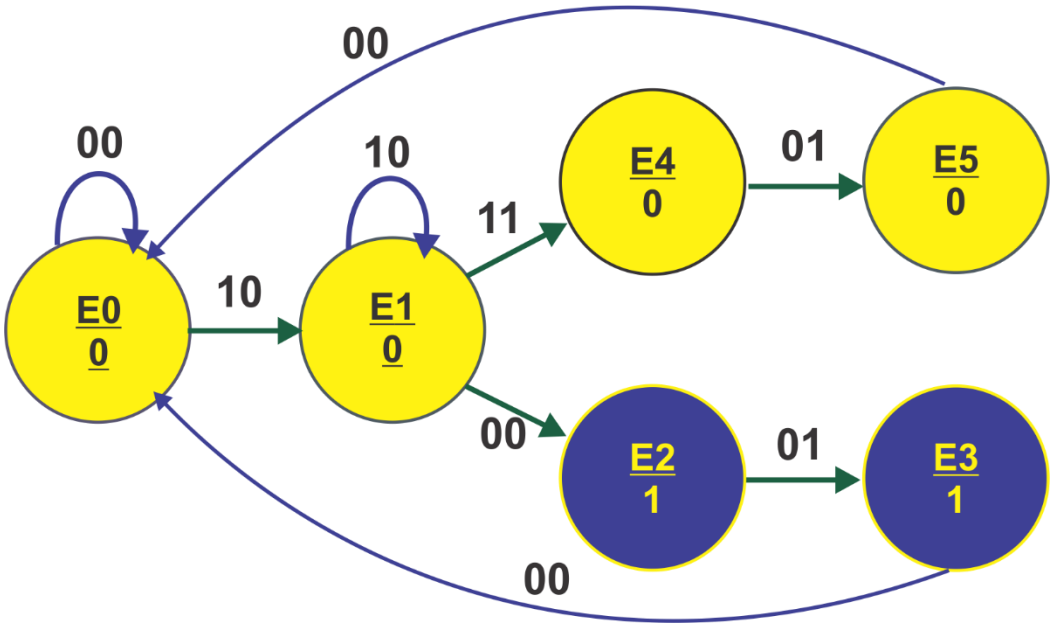
Se consideran 2 casos

1 caso no error		
X1	**X2**	**E**
0	**0**	**0**
1	**0**	**0**
0	**0**	**0**
0	**1**	**0**
0	**0**	**0**

2 caso si error		
X1	**X2**	**E**
0	**0**	**0**
1	**0**	**0**
1	**1**	**1**
0	**1**	**1**
0	**0**	**0**

Diagrama de transición





2.- Tabla de Flujo Primitiva.

	X1, X2				
	00	**01**	**10**	**11**	**E**
0	**E0**	-	E1	-	**0**
1	E2	-	**E1**	E3	**0**
2	**E2**	E4	-	-	**0**
3	-	E5	-	**E3**	**1**
4	E0	**E4**	-	-	**0**
5	E0	**E5**	-	-	**1**

3.- Eliminación de estados redundantes.

E0 = E2

4.- Mezcla de Filas.

	X1, X2				
	00	**01**	**10**	**11**	**E**
0, 4	**E0**	**E4**	E1	-	**0**
1,2	**E2**	E4	**E1**	E3	**0**
3, 5	E0	**E5**	-	**E3**	**1**

5.- Expandir tabla de salidas.

No es necesario

6.- Tabla de estados internos.

	X1, X2				
	00	**01**	**10**	**11**	**E**
a	**E0**	**E4**	E1	-	**0**
b	**E2**	E4	**E1**	E3	**0**
c	E0	**E5**	-	**E3**	**1**

	X1, X2				
	00	**01**	**10**	**11**	**E**
a	**a**	**a**	b	-	**0**
b	**b**	a	**b**	c	**0**
c	a	**c**	-	**c**	**1**

7.- Asignación de Valores a los Estados.

La transición requerida es de

00) c→a, 01) b→a, 10) a→b, 11) b→c

		Q2 = 0	Q2 = 1
Q1	0	**a**	**c**
	1	**b**	

Esta asignación no cumple la transición de b→c por lo que propondremos una transición de b→∝→c

		Q2 = 0	Q2 = 1
Q1	0	**a**	**c**
	1	**b**	∝

Asignación propuesta

	Q1	Q2
a	**0**	**0**
b	**0**	**1**
c	**1**	**0**
∝	**1**	**1**

	X1, X2				
	00	**01**	**10**	**11**	**E**
a	**a**	**a**	b	-	**0**
b	**b**	a	**b**	∝	**0**
c	a	**c**	-	**c**	**1**
∝				c	**0**

8.- Tabla de Estados totales.

Q1 Q2	**X1, X2**				
	00	**01**	**10**	**11**	**E**
00	**a**	**a**	b	-	**0**
01	**b**	a	**b**	∝	**0**
10	a	**c**	-	**c**	**1**
11	-	-	-	c	-

Q1 Q2	**X1, X2**				
	00	**01**	**10**	**11**	**E**
00	**00**	**00**	**01**	**-**	**0**
01	**01**	**00**	**01**	**11**	**0**
10	**00**	**10**	**-**	**10**	**1**
11	**-**	**-**	**-**	**10**	-

9.- Completar Tabla de Salidas.

10.- Obtención de las ecuaciones por medio de minimización.

$$Q1 = X1\,Q1 + X1\,X2$$

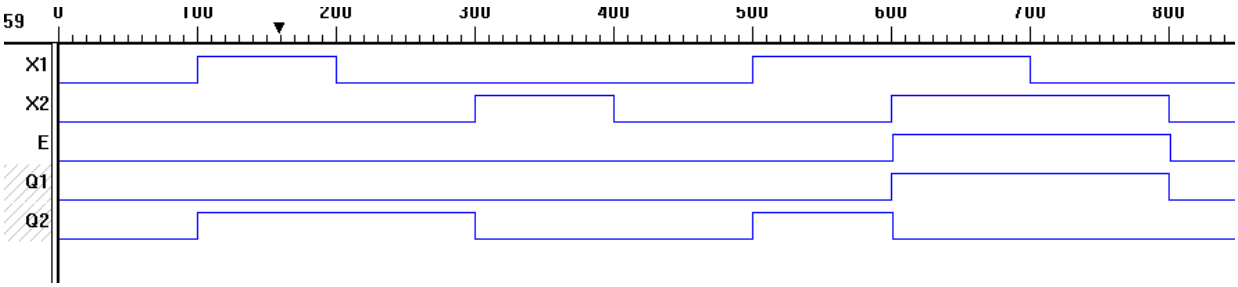
$$Q2 = \mathbf{X2'}\,Q2 + X1\,\mathbf{Q1'}$$

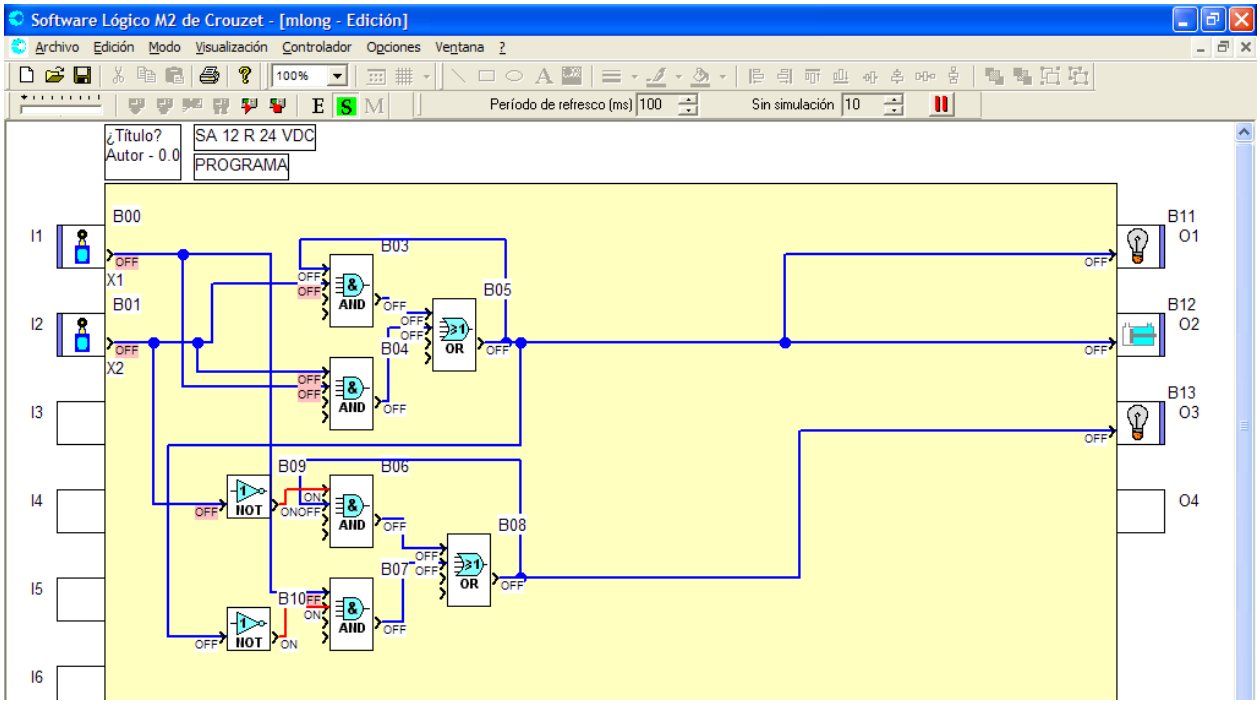
$$E = Q1$$

11.- Elaborar el Archivo en formato ABEL-HDL.

```
MODULE mlong
"Entradas
X1,X2 pin 1,2;
"Salidas
E, Q1,Q2 pin 19..17 istype 'com';
equations
Q1=X2&Q1#X1&X2;
Q2=!X2&Q2#X1&!Q1;
E=Q1;
test_vectors
([X1,X2]->E)
[0,0]->.x.;
[1,0]->.x.;
[0,0]->.x.;
[0,1]->.x.;
[0,0]->.x.;
[1,0]->.x.;
[1,1]->.x.;
[0,1]->.x.;
[0,0]->.x.;
END
```

12.- Simulación.

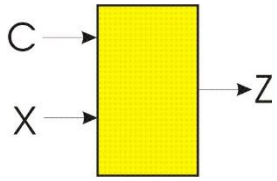




13.- Representación gráfica.

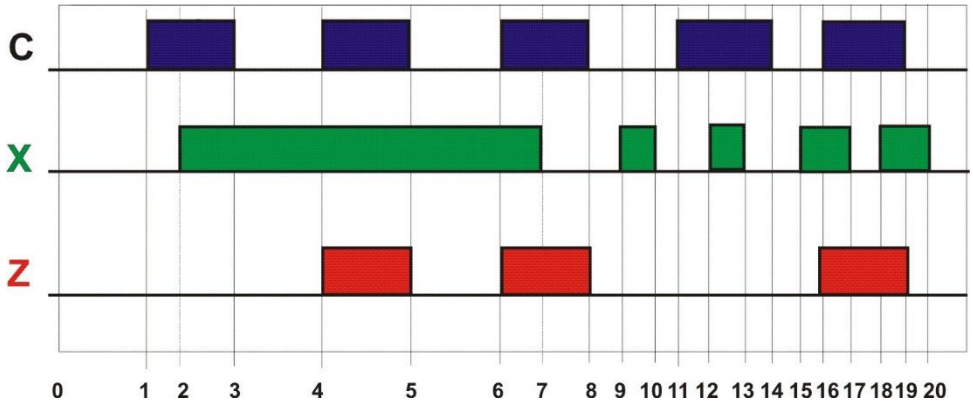
14.- Implementación

Problema 26, Pulsos Completos.



1.-Diseñe un sistema secuencial asíncrono en donde en la salida **Z** deberá aparecer solo Pulsos Completos de una señal **C** cuando la señal **X** tenga un valor de uno como lo muestra la gráfica del diagrama de tiempos.

1.- Especificar el Sistema.



2.- Tabla de Flujo Primitiva.

Para obtener la tabla de flujo primitiva se considera los posibles valores de las entradas **C** y **X** 00, 01, 11, 10 y en el orden descrito ya que si observamos sólo cambia un valor a la vez C o X pero no ambas en columnas consecutivas.

Iniciaremos la tabla de flujo primitiva con el análisis para el tiempo t =0.

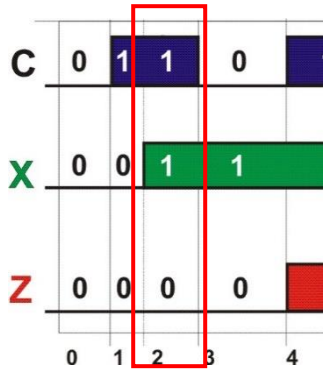
Partiendo de un estado estable inicial **E1** cuando C X = 00 cuya salida Z = 0

Hilera	CX				Salida
	00	**01**	**11**	**10**	**Z**
1	**E1**		**-**		**0**

También podemos incluir un guión – en la columna 11 ya que ésta no es posible que se presente partiendo de una entrada 00 tendría que pasar primero por 01 o 10 pero no 11(dos cambios a la vez no se permiten).

En t = 1

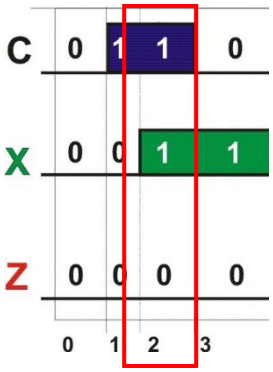
Si del **E1** se presenta una entrada 10 se cambiaría a un estado transitorio E2 en la misma hilera 1 y un estado estable **E2** en una hilera nueva 2 cuya salida también es Z = 0 ya que no se permite en una tabla de flujo primitiva tener dos estados estables en la misma columna..



Hilera	CX				Salida
	00	**01**	**11**	**10**	**Z**
1	**E1**		-	E2	**0**
2		-		**E2**	**0**

En t = 2

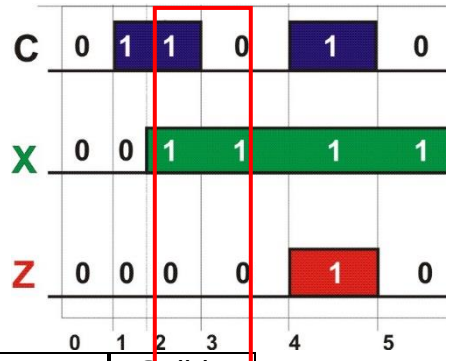
Si del **E2** se presenta una entrada 11 se cambiaría a un estado transitorio E3 en la misma hilera 2 y a un estado estable **E3** en una hilera nueva 3 cuya salida es Z = 0.



Hilera	CX				Salida	
	00	**01**	**11**	**10**	**Z**	
1		**E1**		-	E2	**0**
2			-	E3	**E2**	**0**
3		-		**E3**		**0**

En t = 3

Si del **E3 ,** se presenta una entrada 01 se cambiaría a un estado transitorio E4 en la misma hilera 3 y un estado estable **E4** en una hilera nueva 4 cuya salida también es Z = 0.



Hilera	CX				Salida
	00	**01**	**11**	**10**	**Z**
1	**E1**		-	E2	**0**
2		-	E3	**E2**	**0**
3	-	E4	**E3**		**0**
4		**E4**		-	

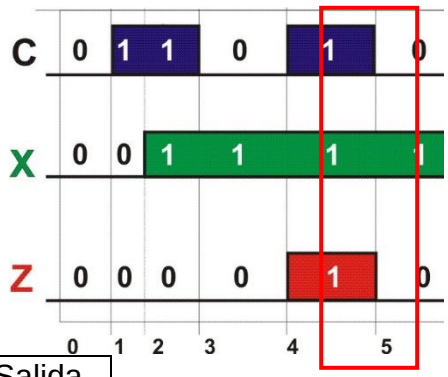
En t = 4

Si del **E4** se presenta una entrada 11 se podría regresar a E3, sin embargo la salida para este caso es Z = 0 y se requiere un valor de Z =1 a lo que necesitamos un estado diferente como transitorio E5 en la misma hilera 4 y un estado estable **E5** en una hilera nueva 5 cuya salida es Z = 1.

Hilera	CX				Salida
	00	**01**	**11**	**10**	**Z**
1	**E1**		-	E2	**0**
2		-	E3	**E2**	**0**
3	-	E4	**E3**		**0**
4		**E4**	E5	-	**0**
5	-		**E5**		**1**

En t = 5

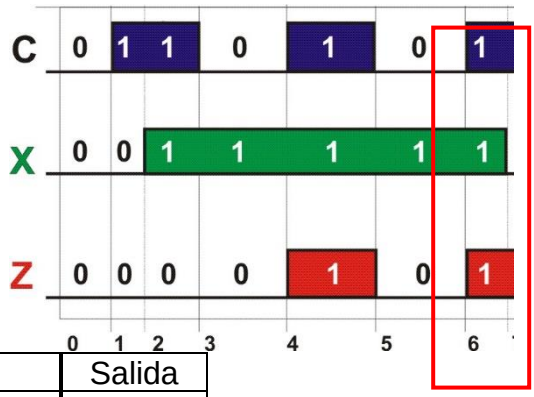
Si del **E5** se presenta una entrada 01 se cambiaría a un estado transitorio E6 en la misma hilera 5 y un estado estable **E6** en una hilera nueva 6 cuya salida es Z = 0.



Hilera	CX				Salida
	00	**01**	**11**	**10**	**Z**
1	**E1**		-	E2	**0**
2		-	E3	**E2**	**0**
3	-	E4	**E3**		**0**
4		**E3**	E5	-	**0**
5	-	E6	**E5**		**1**
6		**E6**		-	**0**

En t = 6

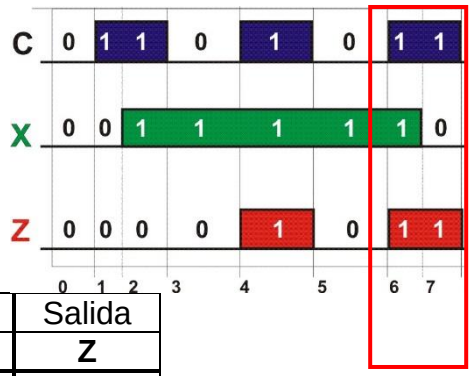
Si del **E6** se presenta una entrada 11 con salida Z=1 regresara por medio de un estado transitorio E5 en la misma hilera 6 a un estado estable **E5** en una hilera 5 cuya salida es Z = 1.



Hilera	CX				Salida
	00	**01**	**11**	**10**	**Z**
1	**E1**		-	E2	**0**
2		-	E3	**E2**	**0**
3	-	E4	**E3**		**0**
4		**E4**	E5	-	**0**
5	-	E6	**E5**		**1**
6		**E6**	E5	-	**0**

En t = 7

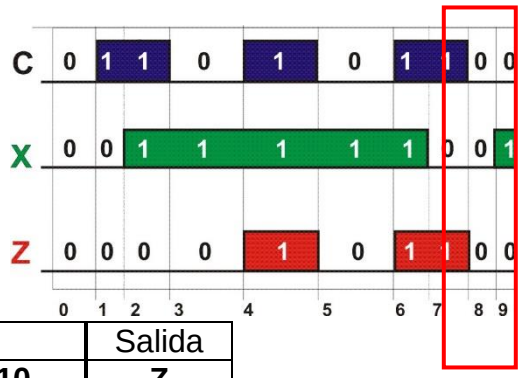
Si del **E5** se presenta una entrada 10 se con salida Z=1 cambiará a un estado transitorio E7 en la misma hilera 5 a estado estable **E7** en una nueva hilera 7 cuya salida es Z = 1. No es posible regresar a E2 por que la salida es diferente.



Hilera	CX				Salida
	00	**01**	**11**	**10**	**Z**
1	**E1**		-	E2	**0**
2		-	E3	**E2**	**0**
3	-	E4	**E3**		**0**
4		**E4**	E5	-	**0**
5	-	E6	**E5**	E7	**1**
6		**E6**	E5	-	**0**
7		-		**E7**	**1**

En t = 8

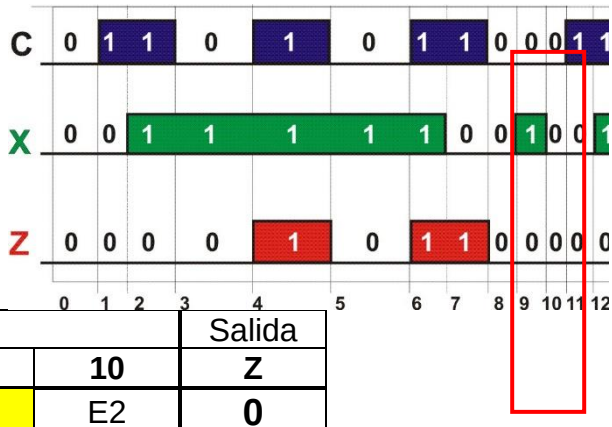
Si del **E7** se presenta una entrada 00 con salida Z=0 regresará por medio de un estado transitorio E1 en la misma hilera 7 a un estado estable **E1** en una hilera 1 cuya salida es Z =0.



Hilera	CX				Salida
	00	**01**	**11**	**10**	**Z**
1	**E1**		-	E2	**0**
2		-	E3	**E2**	**0**
3	-	E4	**E3**		**0**
4		**E4**	E5	-	**0**
5	-	E6	**E5**	E7	**1**
6		**E6**	E5	-	**0**
7	E1	-		**E7**	**1**

En t = 9

Si del **E1** se presenta una entrada 01 con salida Z=0, regresará por medio de un estado transitorio E4 en la misma hilera 1 a un estado estable **E4** ya existente en una hilera 4

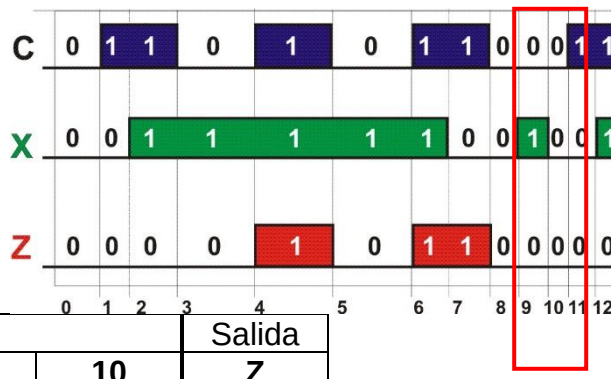


Hilera	CX				Salida
	00	**01**	**11**	**10**	**Z**
1	**E1**	E4	-	E2	**0**
2		-	E3	**E2**	**0**
3	-	E4	**E3**		**0**
4		**E4**	E5	-	**0**
5	-	E6	**E5**	E7	**1**

6		**E6**	E5	-	**0**
7	E1	-		**E7**	**1**

En t = 10

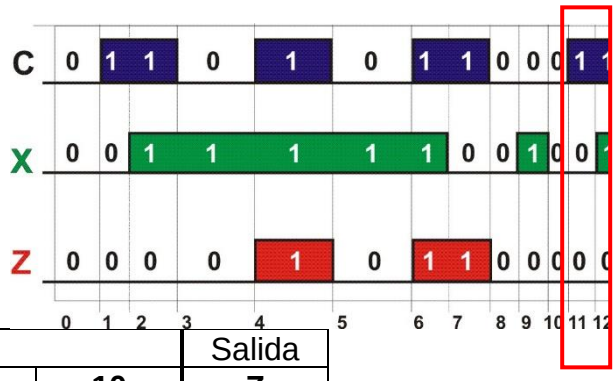
Si del **E4** se presenta una entrada 00 con salida Z=0 regresará por medio de un estado transitorio E1 en la misma hilera 4 a un estado estable **E1** ya existente en una hilera 1 cuya salida es Z =0.



Hilera	CX				Salida
	00	**01**	**11**	**10**	**Z**
1	**E1**	E4	-	E2	**0**
2		-	E3	**E2**	**0**
3	-	E4	**E3**		**0**
4	E1	**E4**	E5	-	**0**
5	-	E6	**E5**	E7	**1**
6		**E6**	E5	-	**0**
7	E1	-		**E7**	**1**

En t = 11

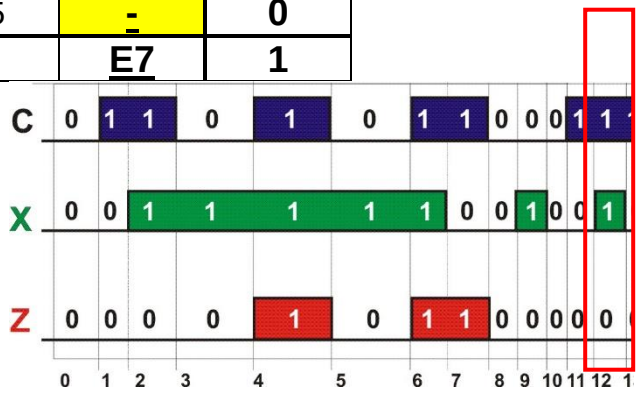
Si del **E1** se presenta una entrada 10 con salida Z=0, pasaría a un estado estable **E2** ya considerado.



Hilera	CX				Salida
	00	**01**	**11**	**10**	**Z**
1	**E1**	E4	-	E2	**0**

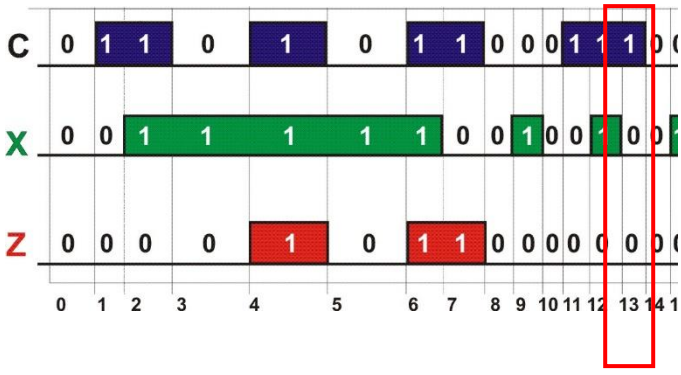
2		-	E3	**E2**	**0**
3	-	E4	**E3**		**0**
4	E1	**E4**	E5	-	**0**
5	-	E6	**E5**	E7	**1**
6		**E6**	E5	-	**0**
7	E1	-		**E7**	**1**

En t = 12, Si del **E2** se presenta una entrada 11 con salida Z=0 pasaría a un estado estable **E3** ya considerado.



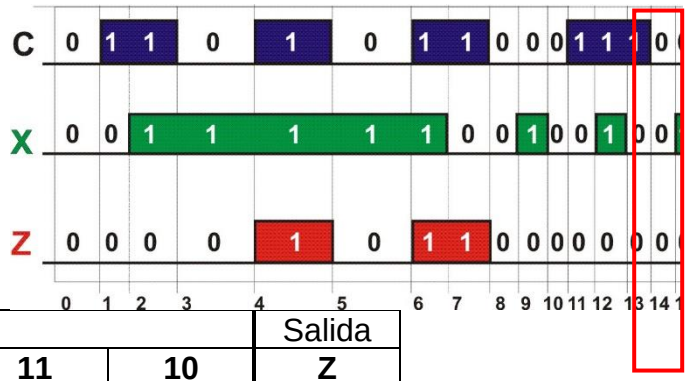
Hilera	CX				Salida
	00	**01**	**11**	**10**	**Z**
1	**E1**	E4	-	E2	**0**
2		-	E3	**E2**	**0**
3	-	E4	**E3**		**0**
4	E1	**E4**	E5	-	**0**
5	-	E6	**E5**	E7	**1**
6		**E6**	E5	-	**0**
7	E1	-		**E7**	**1**

En t = 13, Si del **E3** se presenta una entrada 10 con salida Z=0, regresara por medio de un estado transitorio E2 en la misma hilera 3 a un estado estable **E2** ya existente en una hilera 2.



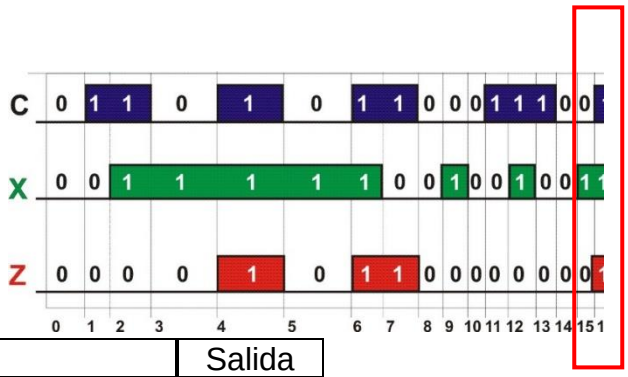
Hilera	CX				Salida
	00	**01**	**11**	**10**	**Z**
1	**E1**	E4	-	E2	**0**
2		-	E3	**E2**	**0**
3	-	E4	**E3**	E2	**0**
4	E1	**E4**	E5	-	**0**
5	-	E6	**E5**	E7	**1**
6		**E6**	E5	-	**0**
7	E1	-		**E7**	**1**

En t = 14, Si del **E2** se presenta una entrada 00 con salida Z=0, regresará por medio de un estado transitorio E1 en la misma hilera 2 a un estado estable **E1** ya existente en una hilera 1.



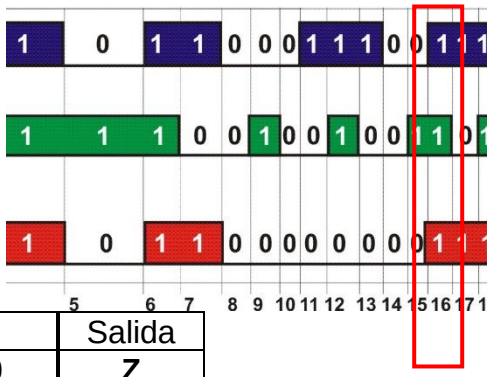
Hilera	CX				Salida
	00	**01**	**11**	**10**	**Z**
1	**E1**	E4	-	E2	**0**
2	E1	-	E3	**E2**	**0**
3	-	E4	**E3**	E2	**0**
4	E1	**E4**	E5	-	**0**
5	-	E6	**E5**	E7	**1**
6		**E6**	E5	-	**0**
7	E1	-		**E7**	**1**

En t = 15, Si del **E1** se presenta una entrada 01 con salida Z=0, a un estado estable **E4** ya considerado en la hilera 4.



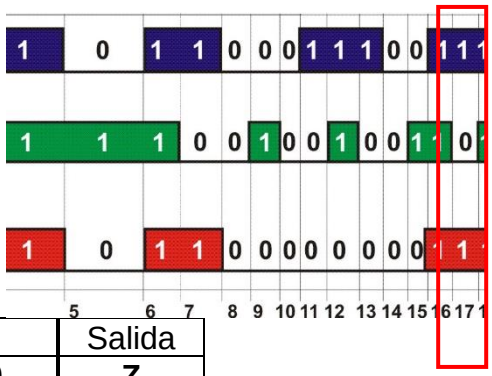
Hilera	CX				Salida
	00	**01**	**11**	**10**	**Z**
1	**E1**	E4	-	E2	**0**
2	E1	-	E3	**E2**	**0**
3	-	E4	**E3**	E2	**0**
4	E1	**E4**	E5	-	**0**
5	-	E6	**E5**	E7	**1**
6		**E6**	E5	-	**0**
7	E1	-		**E7**	**1**

En t = 16, Si del **E4** se presenta una entrada 11 con salida Z=1, a un estado estable **E5** ya considerado en la hilera 5.



Hilera	CX				Salida
	00	**01**	**11**	**10**	**Z**
1	**E1**	E4	-	E2	**0**
2	E1	-	E3	**E2**	**0**
3	-	E4	**E3**	E2	**0**
4	E1	**E4**	E5	-	**0**
5	-	E6	**E5**	E7	**1**
6		**E6**	E5	-	**0**
7	E1	-		**E7**	**1**

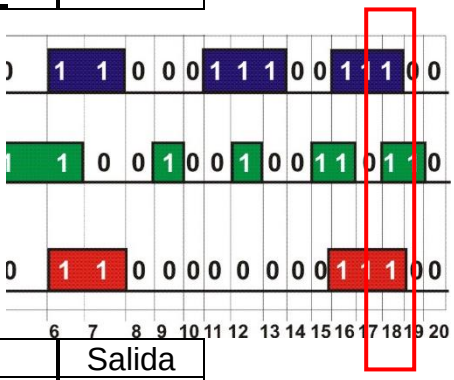
En t = 17, Si del **E5** se presenta una entrada 10 con salida Z=1, a un estado estable **E7** ya considerado en la hilera 7.



Hilera	CX				Salida
	00	**01**	**11**	**10**	**Z**
1	**E1**	E4	-	E2	**0**
2	E1	-	E3	**E2**	**0**
3	-	E4	**E3**	E2	**0**
4	E1	**E4**	E5	-	**0**
5	-	E6	**E5**	E7	**1**
6		**E6**	E5	-	**0**
7	E1	-		**E7**	**1**

En t = 18

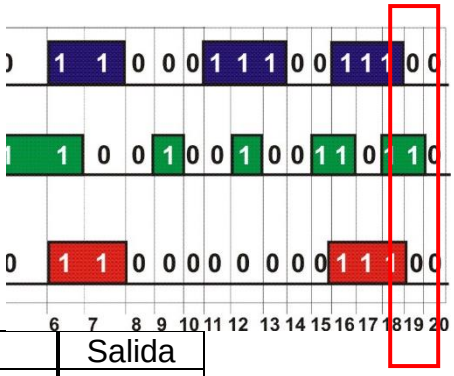
Si del **E7** se presenta una entrada 11 con salida Z=1, a un transitorio E5 en la misma hilera 7 y regresará estado estable **E5** ya considerado en la hilera 5.



Hilera	CX				Salida
	00	**01**	**11**	**10**	**Z**
1	**E1**	E4	-	E2	**0**
2	E1	-	E3	**E2**	**0**
3	-	E4	**E3**	E2	**0**
4	E1	**E4**	E5	-	**0**
5	-	E6	**E5**	E7	**1**
6		**E6**	E5	-	**0**
7	E1	-	E7	**E7**	**1**

En t = 19

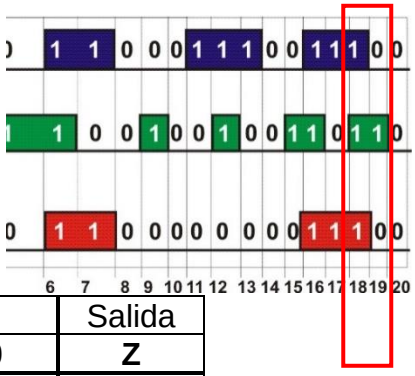
Si del **E5** se presenta una entrada 01 con salida Z=0, regresará estado estable **E6** ya considerado en la hilera 6.



Hilera	CX				Salida
	00	**01**	**11**	**10**	**Z**
1	**E1**	E4	-	E2	**0**
2	E1	-	E3	**E2**	**0**
3	-	E4	**E3**	E2	**0**
4	E1	**E4**	E5	-	**0**
5	-	E6	**E5**	E7	**1**
6		**E6**	E5	-	**0**
7	E1	-	E7	**E7**	**1**

En t = 20

Si del **E6** se presenta una entrada 00 con salida Z=0, pasaría a un estado transitorio E1 en la misma hilera y después al estado estable **E1** ya considerado en la hilera 6.



Hilera	CX				Salida
	00	**01**	**11**	**10**	**Z**
1	**E1**	E4	-	E2	**0**
2	E1	-	E3	**E2**	**0**
3	-	E4	**E3**	E2	**0**
4	E1	**E4**	E5	-	**0**
5	-	E6	**E5**	E7	**1**
6	E1	**E6**	E5	-	**0**
7	E1	-	E5	**E7**	**1**

Con esto completamos nuestra tabla de flujo primitiva,

3. Reducción de Estados.

No necesariamente los estados establecidos en la tabla de flujo primitiva son indispensables, puede haber estados redundantes. Por ello es necesario identificarlo (si los hay) y eliminarlos.

Un estado es redundante si existe uno equivalente.

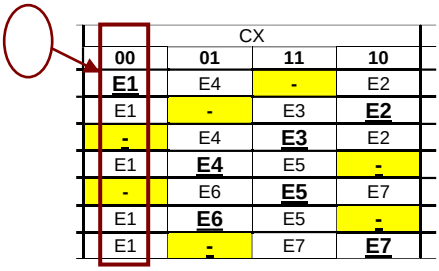
Dos estados son equivalentes si:

a) Son estados estables en la misma columna (misma combinación de entradas).

b) Tienen la misma salida.

c) Sus estados siguientes son equivalentes.

Analizaremos el inciso a) (Son estados estables en la misma columna)



	CX			
	00	**01**	**11**	**10**
	E1	E4	-	E2
	E1	-	E3	**E2**
	-	E4	**E3**	E2
	E1	**E4**	E5	-
	-	E6	**E5**	E7
	E1	**E6**	E5	-
	E1	-	E7	**E7**

En la columna 00) solo hay un estado estable **E1**

En la columna 01) **E4** y **E6** son estados estables

En la columna 11) **E3** y **E5** son estados estables

En la columna 00) **E2** y **E7** son estados estables

Tienen la misma salida

	CX				Salida
	00	**01**	**11**	**10**	**Z**
	E1	E4	-	E2	**0**
	E1	-	E3	**E2**	**0**
	-	E4	**E3**	E2	**0**
	E1	**E4**	E5	-	**0**
	-	E6	**E5**	E7	**1**
	E1	**E6**	E5	-	**0**
	E1	-	E7	**E7**	**1**

E4, Z=0 y **E6,** Z=0 si tienen la misma salida.

E3, Z=0 y **E5,** Z=1 no cumplen por tener salida diferente.

E2, Z=0 y **E7,** Z=1 no cumplen por tener salida diferente.

Sus estados siguientes son equivalentes.

El estado siguiente de **E4** es **E5** para entrada 11.

El estado siguiente de **E4** es **E1** para entrada 00.

El estado siguiente de **E6** es **E5** para entrada 11.

El estado siguiente de **E6** es **E1** para entrada 00.

Por lo tanto **E4** es equivalente **a E6.**

Si eliminamos **E6** tendremos la siguiente Tabla de flujo primitiva.

Hilera	CX				Salida
	00	**01**	**11**	**10**	**Z**
1	**E1**	E4	-	E2	**0**
2	E1	-	E3	**E2**	**0**
3	-	E4	**E3**	E2	**0**
4	E1	**E4**	E5	-	**0**
5	-	E6	**E5**	E7	**1**
6	E1	**E6**	E5	-	**0**
7	E1	-	E5	**E7**	**1**

En donde el E6 inestable de la Hilera 5 columna 01 pasaría a ser E4

Hilera	CX				Salida
	00	**01**	**11**	**10**	**Z**
1	**E1**	E4	-	E2	**0**
2	E1	-	E3	**E2**	**0**
3	-	E4	**E3**	E2	**0**
4	E1	**E4**	E5	-	**0**
5	-	E4	**E5**	E7	**1**
7	E1	-	E5	**E7**	**1**

4.-Mezcla de Filas.

Una vez eliminados los estados redundantes, las filas o hileras pueden mezclarse para reducir la tabla.

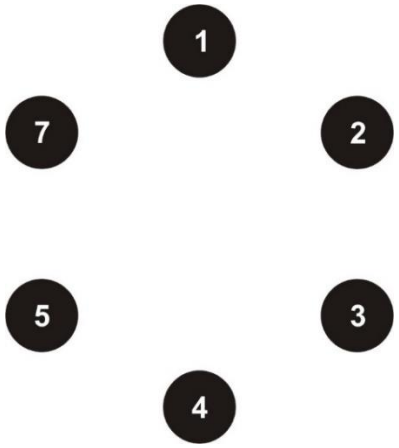
Dos filas o más se pueden mezclar siempre y cuando no hay ningún conflicto sobre qué estado debe ocupar cada columna.

Entendiéndose por conflicto la ocupación simultánea de una columna por dos estados diferentes.

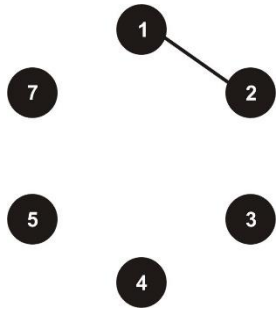
La salida no se considera como un factor de conflicto en la mezcla de filas. Esto es, dos filas con salidas diferentes pueden mezclarse.

Con el propósito de tener una visualización completa sobre las posibilidades de mezcla de las filas, se construye un diagrama de mezcla.

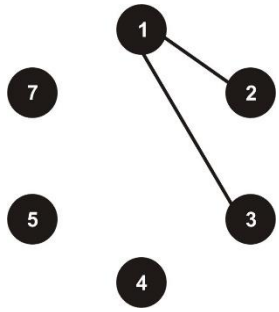
El diagrama de mezcla consiste en asignar un punto por cada fila y se unen esos puntos por líneas cuando estos pueden mezclarse.



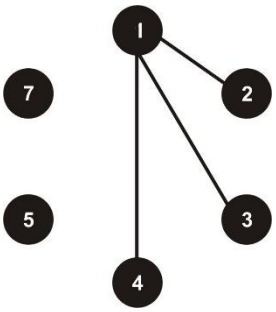
Por ejemplo la hilera 1 y 2 pueden mezclarse por que en la columna:

00	Tienen el mismo E1.	
01	Sólo existe E4.	
11	Sólo existe E3.	
10	Es el mismo E2.	

La hilera 1 y 3 pueden mezclarse por que en la columna:

00	Sólo existe E1.	
01	Tienen el mismo E4.	
11	Sólo existe E3.	
10	Es el mismo E2.	

La hilera 1 y 4 pueden mezclarse por que en la columna:

00 01 11 10	Tienen el mismo E1. Tienen el mismo E4. Sólo existe E5. Solo existe E2.	

La hilera 1 y 5 no se pueden mezclar porque para la columna 10 tiene diferente estado.

a hilera 1 y 7 no se pueden mezclar porque para la columna 10 tiene diferente estado.

La hilera 2 y 3 si se pueden mezclar.

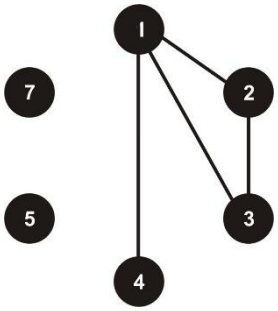
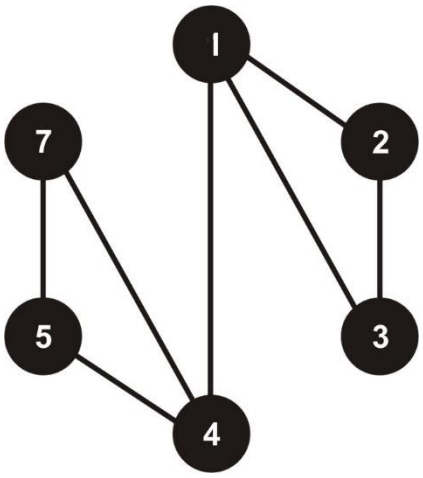


Diagrama de mezcla completo.



Existen dos opciones de mezcla:

a) En grupos de dos E1 con E4, E2 con E3 y E5 con E7.

b) En grupos de tres E1 con E2 y E3, E4 con E5 y E7.

Tomaremos la segunda opción:

Hilera	CX			
	00	**01**	**11**	**10**
1, 2 y 3	**E1**	E4	**E3**	**E2**
4, 5 y 7	E1	**E4**	**E5**	**E7**

Expandiendo la tabla incluyendo una columna para cada salida Z tenemos:

Hilera	CX				Z			
	00	**01**	**11**	**10**	**00**	**01**	**11**	**10**
a	**E1**	E4	**E3**	**E2**	**0**	**-**	**0**	**0**
b	E1	**E4**	**E5**	**E7**	**-**	**0**	**1**	**1**

Asignaremos la letra “a” a la hilera 1, 2 y 3, y la letra “b” a la hilera compuesta por 4, 5 y 7.

4. Completar Tabla de Salidas.

Para asignar los valores a las salidas que quedaron sin definir, se toma el siguiente criterio.

Si el paso del estado estable E4 de la hilera “b” columna 01 al estado estable E1 hilera “a” columna 00, se refleja en la parte de salidas y el cambio es de un valor cero a otro valor cero por lo que no afecta que le asignemos el mismo valor de cero.

Hilera	CX				Z			
	00	**01**	**11**	**10**	**00**	**01**	**11**	**10**
a	**E1**	E4	**E3**	**E2**	**0**	**-**	**0**	**0**
b	E1	**E4**	**E5**	**E7**	**0**	**0**	**1**	**1**

Hilera	CX				Z			
	00	**01**	**11**	**10**	**00**	**01**	**11**	**10**
a	**E1**	E4	**E3**	**E2**	**0**	**0**	**0**	**0**
b	E1	**E4**	**E5**	**E7**	**0**	**0**	**1**	**1**

5.- Expansión de Tabla de Salidas.

La expansión se hizo en el punto anterior.

6.- Tabla de Estados Internos.

Para obtener la tabla de estados internos sustituimos los estados estables de la hilera "a" por la letra, así mismo los de la hilera b.

Hilera	CX				Z			
	00	**01**	**11**	**10**	**00**	**01**	**11**	**10**
a	**a**	b	**a**	**a**	**0**	**0**	**0**	**0**
b	a	**b**	**b**	**b**	**0**	**0**	**1**	**1**

7.- Asignación de Valores a los Estados.

Asignaremos a=0 y b=1

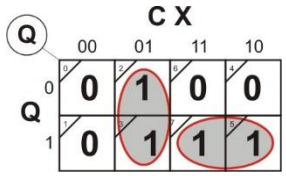
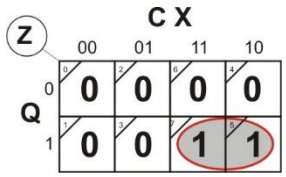
8.- Tabla de Estados.

Q	CX				Z			
	00	**01**	**11**	**10**	**00**	**01**	**11**	**10**
0	**0**	1	**0**	**0**	**0**	**0**	**0**	**0**
1	0	**1**	**1**	**1**	**0**	**0**	**1**	**1**

9.- Completar la Tabla de Salidas.

La tabla de salidas se completó en el punto número 4.

10.- Obtención de las ecuaciones por medio de minimización.

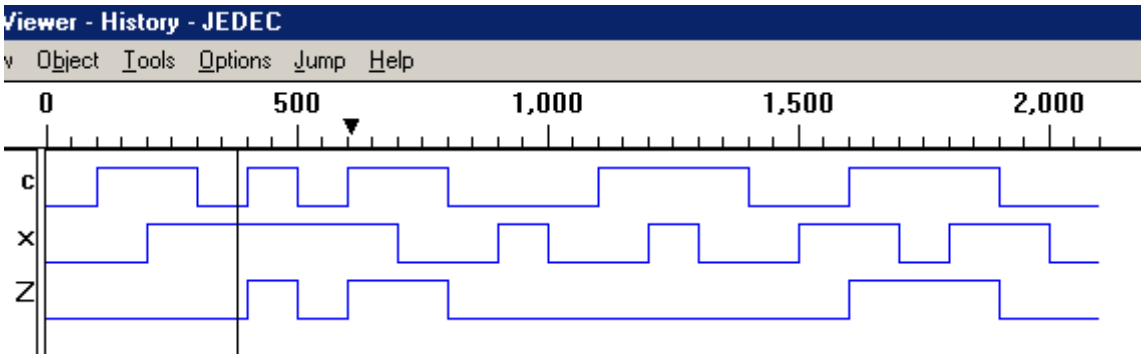
	Q= Q C + **C'** X
	Z= Q C

11.- Archivo en formato ABEL-HDL.

```
MODULE ejeuno
"Entradas
c,x pin;
"Salidas
Q, Z pin istype 'com';
Equations
Q= !c & x # c & Q;
Z =c & Q;
```

```
test_vectors
([c,x,Q]->[Q,Z])
[0,0,0]->[.x.,.x.];
[1,0,.x.]->[.x.,.x.];
[1,1,.x.]->[.x.,.x.];
[0,1,.x.]->[.x.,.x.];
[1,1,.x.]->[.x.,.x.];
[0,1,.x.]->[.x.,.x.];
[1,1,.x.]->[.x.,.x.];
[1,0,.x.]->[.x.,.x.];
[0,0,.x.]->[.x.,.x.];
[0,1,.x.]->[.x.,.x.];
[0,0,.x.]->[.x.,.x.];
[1,0,.x.]->[.x.,.x.];
[1,1,.x.]->[.x.,.x.];
[1,0,.x.]->[.x.,.x.];
[0,0,.x.]->[.x.,.x.];
[0,1,.x.]->[.x.,.x.];
[1,1,.x.]->[.x.,.x.];
[1,0,.x.]->[.x.,.x.];
[1,1,.x.]->[.x.,.x.];
[0,1,.x.]->[.x.,.x.];
[0,0,.x.]->[.x.,.x.];
END
```

12.- Simulación.



13.- Representación.

Problema 27, impulsos sucesivos

Un sistema secuencial asíncrono posee dos entradas de impulsos x1 y x2 (ambas entradas no pueden estar nunca en estado 1 simultáneamente) y una salida z. A partir de un estado inicial en el cual x1=x2=z=0, la salida z ha de tomar el valor 1 si se aplican dos impulsos sucesivos a la entrada x1 sin que se aplique ninguno a la entrada x2. La entrada x2 actúa como entrada de inicialización. La salida volverá a 0 sólo cuando se active x2. Se especifica que x1 y x2 no pueden cambiar simultáneamente.

1.- Especificar el Sistema.

2.- Tabla de Flujo Primitiva.

X1 X2				salida
00	01	10	11	Z
E0	E6	E1	-	0
E2	-	**E1**	-	0
E2	E7	E3	-	0
E4	-	**E3**	-	1
E4	E5	E8	-	1
E0	**E5**	-	-	0
E0	**E6**	-	-	0
E0	**E7**	-	-	0
E4		**E8**		1

3.- Eliminación de estados redundantes.

X1 X2				salida
00	01	10	11	Z
E0	EX	E1	-	0
E2	-	**E1**	-	0
E2	EX	EY	-	0
E4	-	**EY**	-	1
E4	EX	EY	-	1
E0	**EX**	-	-	0

4.- Mezcla de Filas.

	X1 X2				salida
	00	01	10	11	Z
0,5	**E0**	**EX**	E1	-	0
1	E2	-	**E1**	-	0
2	**E2**	EX	EY	-	0
3,4	**E4**	EX	**EY**	-	1

5.- Expandir tabla de salidas.

No es necesario expandir la tabla de salidas.

6.- Tabla de estados internos.

	X1 X2				salida
	00	01	10	11	Z
a	**E0**	**EX**	E1	-	0
b	E2	-	**E1**	-	0
c	**E2**	EX	EY	-	0
d	**E4**	EX	**EY**	-	1

	X1 X2				salida
	00	**01**	**10**	**11**	**Z**
a	**a**	**a**	**b**	**-**	**0**
b	**c**	**-**	**b**	**-**	**0**
c	**c**	**a**	**d**	**-**	**0**
d	**d**	**a**	**d**	**-**	**1**

7.- Asignación de Valores a los Estados.

Condiciones:

00)b-c

01) d-a, c-a

10) a-b, c-d

Q3 \ Q1 Q2	00	01	11	10
0	**a**			**d**
1	**b**	∝	**c**	**β**

00) b-∝-c

01) d-a, c- **β** -a

10) a-b, c- **β** -d

Asignación:

a=000, b=001, c=111, d=100, ∝=011 y β=101

8.- Tabla de Estados totales.

	X1 X2				salida
	00	01	10	11	Z
a	**a**	**a**	b	-	0
b	∝	-	**b**	-	0
c	**c**	β	β	-	0
d	**d**	a	**d**	-	1
∝	**c**			-	
β		a	d	-	

	X1 X2				salida
	00	01	10	11	Z
000	**000**	**000**	001	-	0
001	011	-	**001**	-	0
111	**111**	101	101	-	0
100	**100**	000	**100**	-	1
011	**111**			-	0
101		000	100	-	0-

9.- Completar Tabla de Salidas.

No es necesario completar la tabla de salidas.

10.- Obtención de las ecuaciones por medio de minimización.

Ecuaciones:

$$Q1+ = Q2 + Q1\,X\mathbf{2'}$$

$$Q2+ = Q3\ \mathbf{X1'X2'}$$

$$Q3+ = \mathbf{Q1'}X1 + Q2 + \mathbf{Q1'}Q3$$

$$Z = Q1\ \mathbf{Q2'}$$

11.- Elaborar el Archivo en formato ABEL-HDL.

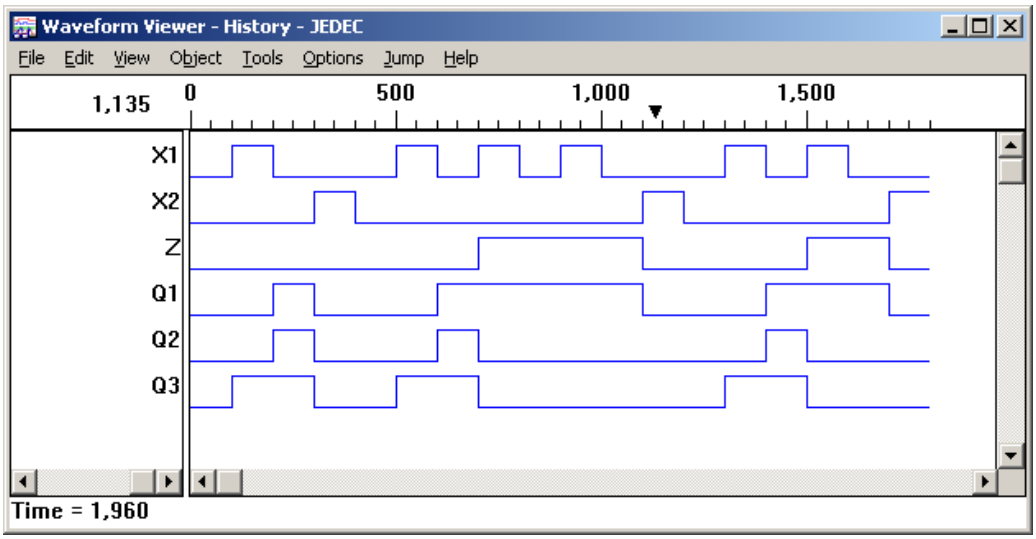
```
MODULE impulso

X1,X2 PIN 1,2;
Q1,Q2,Q3,Z PIN 19,18,17,16 ISTYPE 'COM';

EQUATIONS
Q1=Q2 # (Q1&!X2);
Q2=Q3&!X1&!X2;
Q3=(!Q1&X1)#(Q2)#(!Q1&Q3);
Z=Q1&!Q2&!Q3;

TEST_VECTORS
([X1,X2]->[Z,Q1,Q2,Q3])
[0,0]->[.X.,.X.,.X.,.X.];
[1,0]->[.X.,.X.,.X.,.X.];
[0,0]->[.X.,.X.,.X.,.X.];
[0,1]->[.X.,.X.,.X.,.X.];
[0,0]->[.X.,.X.,.X.,.X.];
[1,0]->[.X.,.X.,.X.,.X.];
[0,0]->[.X.,.X.,.X.,.X.];
[1,0]->[.X.,.X.,.X.,.X.];
[0,0]->[.X.,.X.,.X.,.X.];
[1,0]->[.X.,.X.,.X.,.X.];
[0,0]->[.X.,.X.,.X.,.X.];
[0,1]->[.X.,.X.,.X.,.X.];
[0,0]->[.X.,.X.,.X.,.X.];
[1,0]->[.X.,.X.,.X.,.X.];
[0,0]->[.X.,.X.,.X.,.X.];
[1,0]->[.X.,.X.,.X.,.X.];
[0,0]->[.X.,.X.,.X.,.X.];
[0,1]->[.X.,.X.,.X.,.X.];
END
```

12 Simulación.



13.- Representación gráfica.

Problema 28, Secuencia de puerta dos botones

se cuenta con dos botones llamados A y B de modo que si se oprime la secuencia A, B el sistema deberá de abrir una puerta, cualquier otra secuencia diferente de A, B por ejemplo B o A, A deberá de activar una alarma

1.- Especificar el sistema

Diagrama de transición

2.-Tabla de flujo primitiva

Hilera	A, B				Salida	
	00	**01**	**10**	**11**	**P**	**Al**
0	**E0**	E8	E1	**-**	**0**	**0**
1	E2	-	**E1**	E9	**0**	**0**
2	**E2**	E3	E10		**0**	**0**
3	E4	**E3**		E5	**1**	**0**
4	**E4**	E6	E7		**1**	**0**
5		E6	E7	**E5**	**1**	**0**
6	E0	**E6**		E5	**1**	**0**
7	E0		**E7**	E5	**1**	**0**
8	E11	**E8**		E9	**0**	**1**
9		E8	E10	**E9**	**0**	**1**
10	E11		**E10**	E9	**0**	**1**
11	**E11**	E12	E13		**0**	**1**
12	E0	**E12**		E9	**0**	**1**
13	E0		**E13**	E9	**0**	**1**

3.- reducción de estados

1.- estables en la misma columna

00) (E0, E2, E4, E11), 01) (E3, E6, E8, E12), 10) (E1, E7, E10, E13), 11) (E5, E9)

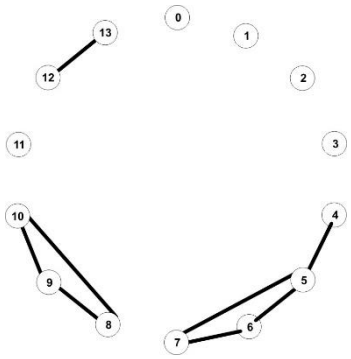
2.- Misma salida

(E0, E2), (E3, E6), (E8, E12), (E10, E13)

3.- el estado próximo equivalente

no tiene estados redundantes

4.- Mezcla de Filas

	Hilera	A, B				Salida	
		00	**01**	**10**	**11**	**P**	**Al**
	0	**E0**	E8	E1	**-**	**0**	**0**
	1	E2	-	**E1**	E9	**0**	**0**
	2	**E2**	E3	E10		**0**	**0**
	3	E4	**E3**		E5	**1**	**0**
	4	**E4**	E6	E7		**1**	**0**
	5		E6	E7	**E5**	**1**	**0**
	6	E0	**E6**		E5	**1**	**0**
	7	E0		**E7**	E5	**1**	**0**
	8	E11	**E8**		E9	**0**	**1**
	9		E8	E10	**E9**	**0**	**1**
	10	E11		**E10**	E9	**0**	**1**
	11	**E11**	E12	E13		**0**	**1**
	12	E0	**E12**		E9	**0**	**1**
	13	E0		**E13**	E9	**0**	**1**

Hilera	A, B				Salida	
	00	**01**	**10**	**11**	**P**	**Al**
0	**E0**	E8	E1	**-**	**0**	**0**
1	E2	-	**E1**	E9	**0**	**0**
2	**E2**	E3	E10		**0**	**0**
3	E4	**E3**		E5	**1**	**0**
4	**E4**	E6	E7		**1**	**0**
5, 6, 7	E0	**E6**	**E7**	**E5**	**1**	**0**
8, 9, 10	E11	**E8**	**E10**	**E9**	**0**	**1**
11	**E11**	E12	E13		**0**	**1**
12, 13	E0	**E12**	**E13**	E9	**0**	**1**

5.- Expandir Tabla de Salidas.

No es necesario expandir tabla de salidas.

6.- Tabla de Estados Internos.

Asignado variable a cada una de las hileras obtenemos:

Hilera	A, B				Salida	
	00	**01**	**10**	**11**	**P**	**Al**
a	**E0**	E8	E1	**-**	**0**	**0**
b	E2	-	**E1**	E9	**0**	**0**
c	**E2**	E3	E10		**0**	**0**
d	E4	**E3**		E5	**1**	**0**
e	**E4**	E6	E7		**1**	**0**
f	E0	**E6**	**E7**	**E5**	**1**	**0**
g	E11	**E8**	**E10**	**E9**	**0**	**1**
h	**E11**	E12	E13		**0**	**1**
i	E0	**E12**	**E13**	E9	**0**	**1**

Cambiando los estados por la correspondiente variable tenemos:

Hilera	A, B				Salida	
	00	**01**	**10**	**11**	**P**	**Al**
a	**a**	g	b	**-**	**0**	**0**
b	c	-	**b**	g	**0**	**0**
c	**c**	d	g		**0**	**0**
d	e	**d**		f	**1**	**0**
e	**e**	f	f		**1**	**0**
f	a	**f**	**f**	**f**	**1**	**0**
g	h	**g**	**g**	**g**	**0**	**1**
h	**h**	i	i		**0**	**1**
i	a	**i**	**i**	g	**0**	**1**

7.- Asignación de Valores a los Estados.

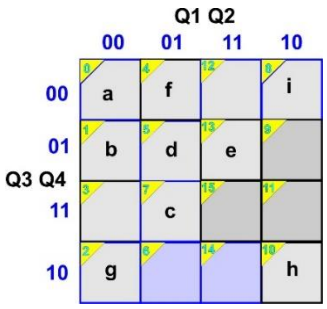
Transiciones

00) (b→c), (d→e), (f, i→a)

01) (a→g), (c→d), (e→f), (h→i)

10) (a→b), (c→g), (e→f) , (h→i)

11) (b, i→g), (d→f),



Como no es posible cumplir con las transiciones preponderemos agregar dos estados llamados $\propto$ y β, así como utilizar en caso necesario los lugares no asignados(huecos) en la tabla.

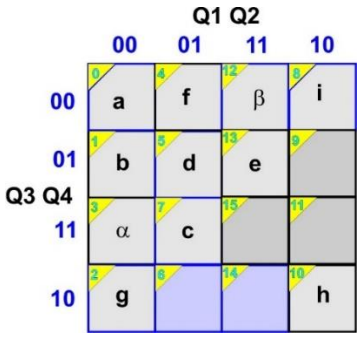
De modo que la asignación para cumplir con las transiciones seria:

00) (b→∝→c), (d→e), (f→β→a) (i→a)

01) (a→g), (c→d), (e→β→f), (h→i)

10) (a→b), (c→∝→g), (e→β→f) (h→i)

11) (b→∝→g), (i→h→g), (d→f),



Hilera	A, B				Salida	
	00	**01**	**10**	**11**	**P**	**Al**
a	**a**	g	b	-	**0**	**0**
b	∝	-	**b**	g	**0**	**0**
c	**c**	d	∝	∝	**0**	**0**
d	e	**d**		f	**1**	**0**
e	**e**	β	β		**1**	**0**
f	a	**f**	**f**	**f**	**1**	**0**
g	h	**g**	**g**	**g**	**0**	**1**
h	**h**	i	i	g	**0**	**1**
i	a	**i**	**i**	h	**0**	**1**
∝	c		g	g	**0**	**0**
β		f	f		**1**	**0**

Asignación

	Q1	Q2	Q3	Q4
a	**0**	**0**	**0**	**0**
b	**0**	**0**	**0**	**1**
c	**0**	**1**	**1**	**1**
d	**0**	**1**	**0**	**1**
e	**1**	**1**	**0**	**1**
f	**0**	**1**	**0**	**0**
g	**0**	**0**	**1**	**0**
h	**1**	**0**	**1**	**0**
i	**1**	**0**	**0**	**0**
$\propto$	**0**	**0**	**1**	**1**
β	**1**	**1**	**0**	**0**

8.- Tabla de Estados.

Sustituyendo los valores asignados obtenemos

Q1, Q2, Q3, Q4	A B				Salidas	
	00	**01**	**10**	**11**	**P**	**Al**
0000	**0000**	**0010**	**0001**	**-**	**0**	**0**
0001	**0011**	**-**	**0001**	**0010**	**0**	**0**
0111	**0111**	**0101**	**0011**	**0011**	**0**	**0**
0101	**1101**	**0101**		**0100**	**1**	**0**
1101	**1101**	**1100**	**1100**		**1**	**0**
0100	**0000**	**0100**	**0100**	**0100**	**1**	**0**
0010	**1010**	**0010**	**0010**	**0010**	**0**	**1**
1010	**1010**	**1000**	**1000**	**0010**	**0**	**1**
1000	**0000**	**1000**	**1000**	**1010**	**0**	**1**
0011	**0111**		**0010**	**0010**	**0**	**0**
1100		**0100**	**0100**		**1**	**0**

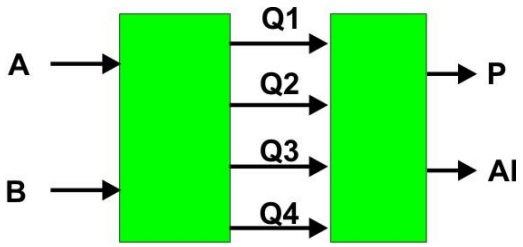
Acomodando hileras

Q1, Q2, Q3, Q4	A B				Salidas	
	00	**01**	**10**	**11**	**P**	**Al**
0000	**0000**	**0010**	**0001**	**-**	**0**	**0**
0001	**0011**	**-**	**0001**	**0010**	**0**	**0**
0010	**1010**	**0010**	**0010**	**0010**	**0**	**1**
0011	**0111**		**0010**	**0010**	**0**	**0**
0100	**0000**	**0100**	**0100**	**0100**	**1**	**0**
0101	**1101**	**0101**	**-**	**0100**	**1**	**0**
0110	**-**	**-**	**-**	**-**	**-**	**-**
0111	**0111**	**0101**	**0011**	**0011**	**0**	**0**
1000	**0000**	**1000**	**1000**	**1010**	**0**	**1**
1001	**-**	**-**	**-**	**-**	**-**	**-**
1010	**1010**	**1000**	**1000**	**0010**	**0**	**1**
1011	**-**	**-**	**-**	**-**	**-**	**-**
1100		**0100**	**0100**	**-**	**1**	**0**
1101	**1101**	**1100**	**1100**	**-**	**1**	**0**
1110	**-**	**-**	**-**	**-**	**-**	**-**
1111	**-**	**-**	**-**	**-**	**-**	**-**

9.- Completar Tabla de Salidas.

No es necesario completar tabla de salidas.

10.- Obtención de las ecuaciones por medio de minimización.



P=Q2 Q3', Al=Q4'Q3 + Q1 Q2',

Q1+=Q3Q4'A'B'+Q2Q3'Q4B'+Q1Q4+Q1Q2'AB'+Q1Q2'Q3'B+Q1 Q3 A'

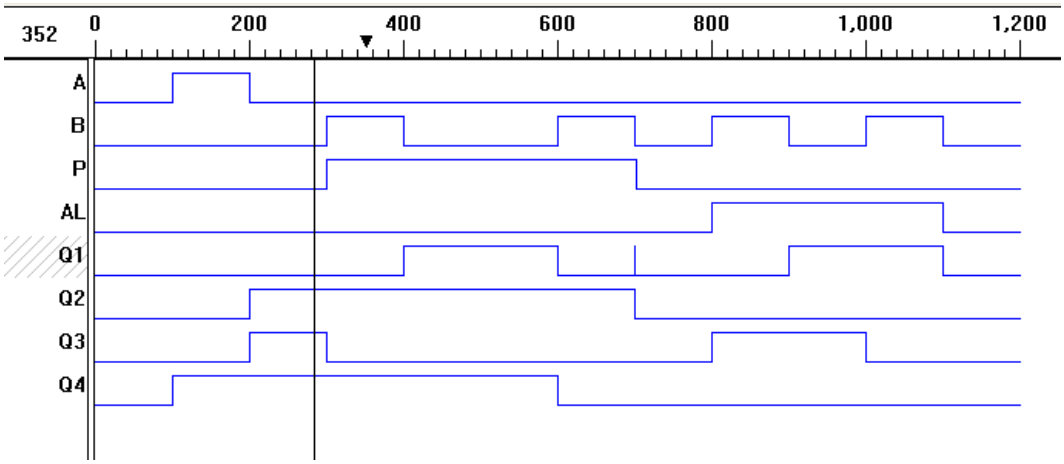
Q2+ = Q3 Q4 A' + Q2 Q3'B + Q2 Q3'A + Q2 Q3'Q4

Q3+ = Q1'Q2'B + Q2'Q4 A' + Q3 A'B' + Q1'Q3 A + Q2'A B

Q4+ = Q1'Q4 A' + Q2 Q3 + Q1'Q2'Q3'A B' + Q4 A'B'

11.- Archivo en formato ABEL-HDL

12.- Simulación

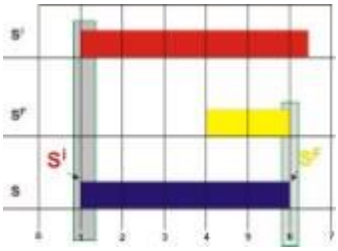


12.- Representación

Variable Lógica

La Variable Lógica es un método desarrollado por el ***Dr. Cesar Elizondo González*** que nos permite obtener las ecuaciones de un sistema secuencial asíncrono a partir de un diagrama de tiempos

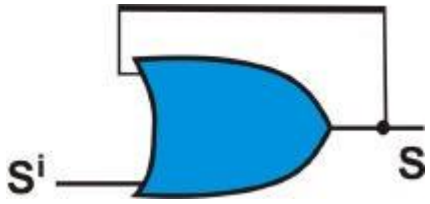
Teorema Fundamental Secuencial.



Toda señal está definida por un tiempo de creación $\mathbf{S^i}$ y un tiempo de extinción $\mathbf{S^f}$ en donde el período de tiempo **S** será igual al producto de $\mathbf{S^i}$ y por el complemento de $\mathbf{S^f}$.

$$S = S^i \ (S^f)'$$

Toda señal tiene un inicio propiciado por un evento, lo llamaremos $\mathbf{S^i}$, y para que la señal se mantenga activa se necesita memorizarse con la misma señal S.

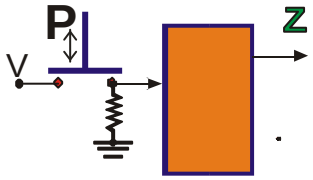
$\underline{S^i = S^i + S}$	

La operación OR es una retroalimentación en el mismo circuito con lo que asegura que la señal S^i se mantiene activada.

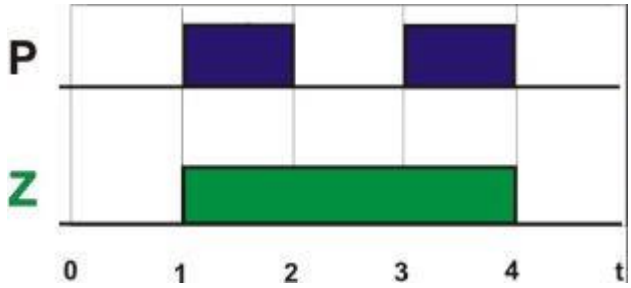
El fin de la señal $\mathbf{S^f}$ es propiciado por otro evento diferente al del inicio.

Para demostrar la eficacia del método lo aplicaremos a los problemas 1, 2, 3, 4, 5, 6, 7 y 11 de este documento.

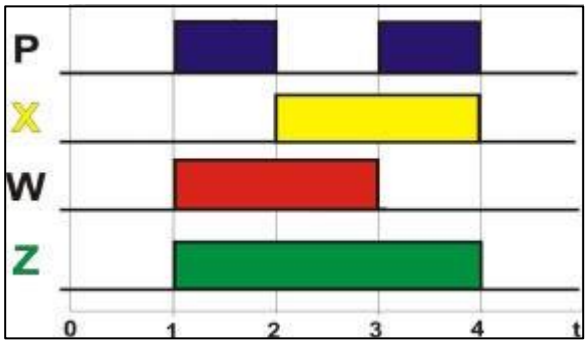
Problema 29, Botón caso 1.1



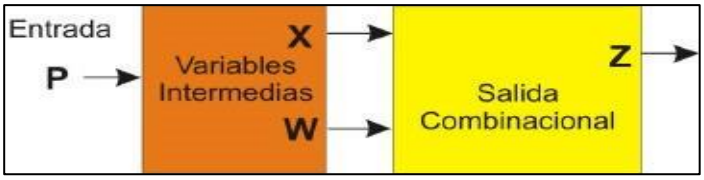
Por medio de un botón **P** de no retención, se obtiene una + salida **Z**, de modo que al oprimirlo la primera vez **Z=1**, y al oprimirlo en la segunda ocasión y soltar el botón Z**=0** como lo indica la siguiente figura:



Es necesario incluir dos variables adicionales las llamaremos **W** y **X**. **W** para indicar la primera vez que se oprime P y **X** para indicar la segunda vez que se oprime P como lo indica la figura

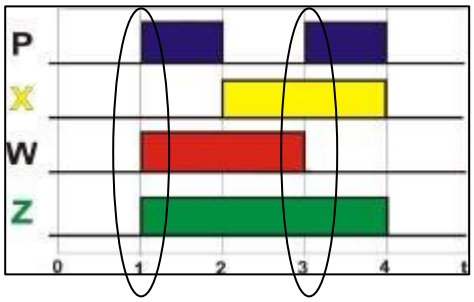


Las ecuaciones para **X** deberán de depender de la entrada **P** y de la variable **W**, así las ecuaciones de **W** deberán depender de **P** y **X**, en cambio la salida **Z** sólo dependerá de **X** y **W**.



Obtención de las Ecuaciones:

W inicia en el t=1 cuando empieza **P,** y no está **X** (X') para mantener la señal de W que se memoriza con la misma W, la ecuación de W^i sería:



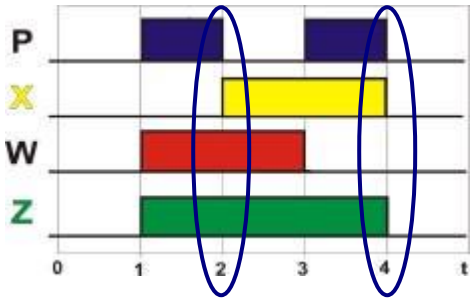
$$Wi = P\mathbf{X'} + W$$

W termina en t=3 cuando de nuevo se oprime P, y además esta X la ecuación de W^f sería:

$$Wf = P\ X$$

Aplicando el teorema fundamental tenemos: $W = (P\mathbf{X'} + W)(\mathbf{P'} + \mathbf{X'})$

X inicia en t=2 cuando no está **P** , si está **W** y se memoriza de modo que:



$X^i = P'W + X$ **x** termina en t=4 cuando no está P ni está W obtenemos : $Xf = \mathbf{P'\ W'}$

Aplicando el Teorema fundamental tenemos:

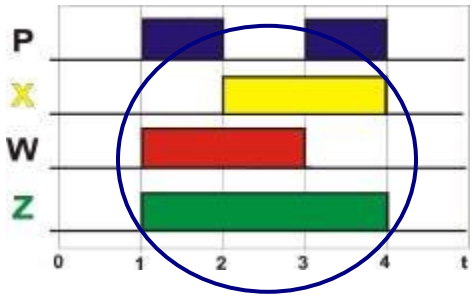
$$= (\mathbf{P'}W + X)(P + W)$$

La salida **Z**, vale uno del t=1 al t=4, cuando esta **W** o cuando está **X** o ambas, de modo que se puede hacer una tabla de verdad para la obtención de la ecuación de Z que es una Or de X y W:

$$Z = W + X$$

Tabla de Verdad

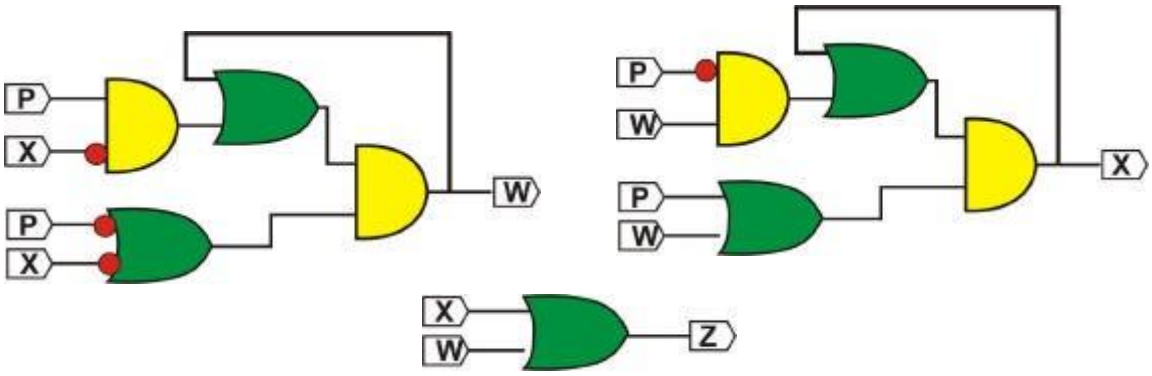
m	X	W	Z
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	1



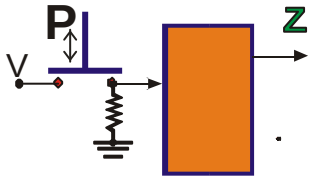
A continuación, se presenta el archivo en formato ABEL-HDL, incluyendo las ecuaciones, la instrucción test vectors y su simulación:

```
Module boton
“Entrada
 P pin 1;
“Salidas
 W,X,Z pin 19,18,17 istype ‘com’;
Equations
W= (P&!X#W)&(!P#!X);
X=(!P&W#X)&(P#W);
 Z=W#X;
test_vectors ([P]->[Z,X,W])
[0]->[.x.,.x.,.x.];
[1]->[.x.,.x.,.x.];
[0]->[.x.,.x.,.x.];
[1]->[.x.,.x.,.x.];
[0]->[.x.,.x.,.x.];
[1]->[.x.,.x.,.x.];
[0]->[.x.,.x.,.x.];
[1]->[.x.,.x.,.x.];
End
```

Diagrama Esquemático



Botón caso 1.2



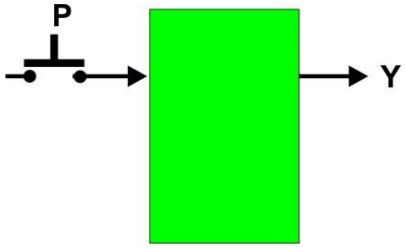


Diagrama de bloques

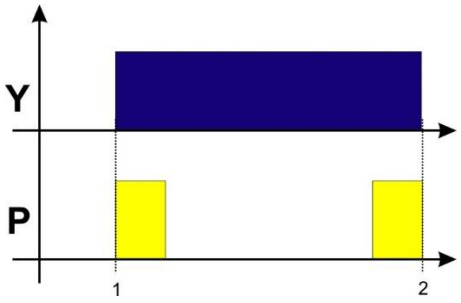
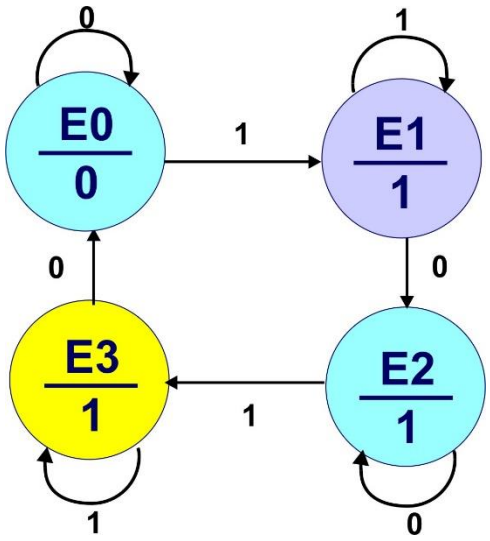


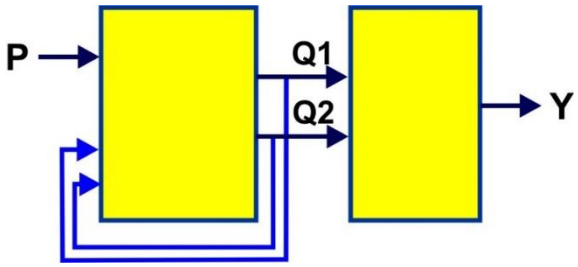
Diagrama de tiempos

1.- Especificar el sistema

Diagrama de transición



2.- Podemos observar en la tabla mostrada que con respecto al problema 29 el único cambio es la salida Y por tratarse del modelo de Moore, en donde la salida depende de los valores de Q1 y Q2 que no cambian como se muestra en la siguiente figura:

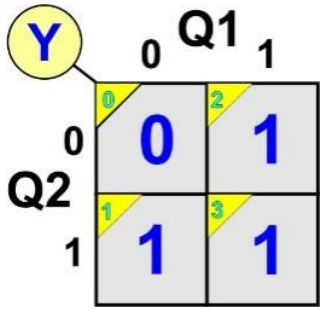


10.- Obtención de las ecuaciones por medio de minimización.

La salida Y está en función de Q1 y Q2 de modo que de la tabla anterior tenemos que:

Tabla de Verdad

m	X	W	Z
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	1

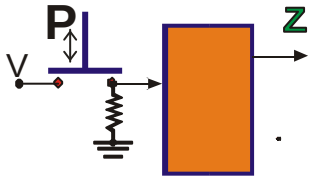


$$Y = Q1 + Q2$$

Archivo ABEL-HDL

```
Module boton
“Entrada
 P pin 1;
“Salidas
 W,X,Z pin 19,18,17 istype ‘com’;
Equations
W= (P&!X#W)&(!P#!X);
X=(!P&W#X)&(P#W);
Z=W#X;
test_vectors ([P]->[Z,X,W])
[0]->[.x.,.x.,.x.];
[1]->[.x.,.x.,.x.];
[0]->[.x.,.x.,.x.];
[1]->[.x.,.x.,.x.];
[0]->[.x.,.x.,.x.];
[1]->[.x.,.x.,.x.];
[0]->[.x.,.x.,.x.];
[1]->[.x.,.x.,.x.];
End
```

Botón caso 1.3



Diseñe un sistema secuencial asíncrono que contenga una entrada **P** de modo que al oprimir el botón por primera vez la salida Y deberá de permanecer en un valor de cero y al soltar el botón la salida deberá cambiar a uno, al oprimirlo por segunda vez la señal de salida deberá de continuar en uno y al soltarlo cambiar a cero como lo indica la siguiente gráfica:

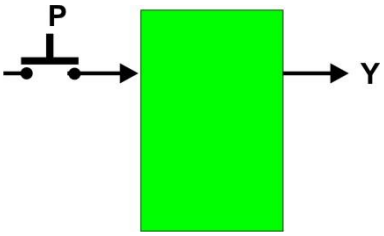


Diagrama de bloques

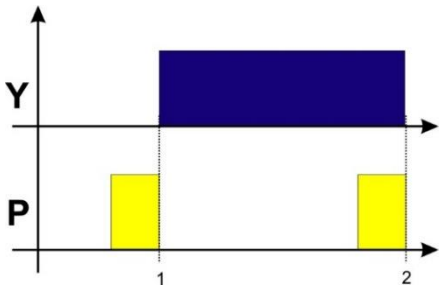
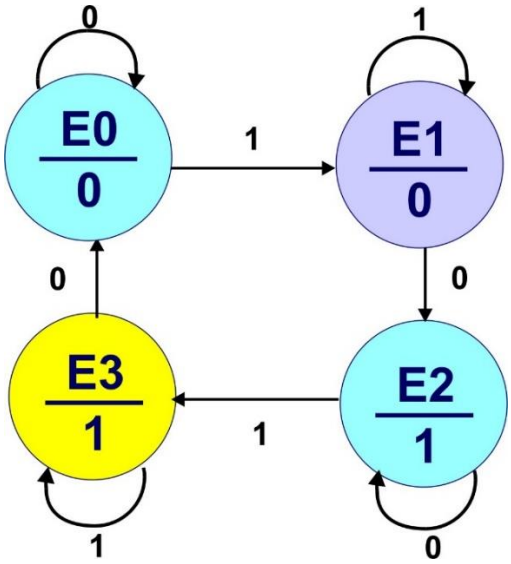
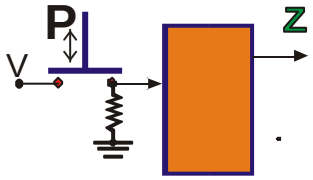


Diagrama de tiempos

1.-Diagrama de transición



Botón caso 1.4



Diseñe un sistema secuencial asíncrono que contenga una entrada **P** de modo que al oprimir el botón por primera vez la salida Y deberá de permanecer en un valor de cero y al soltar el botón la salida deberá cambiar a uno, al oprimirlo por segunda vez la señal de salida deberá de cambiar a cero y al soltarlo permanecer con el mismo valor como lo indica la siguiente gráfica:

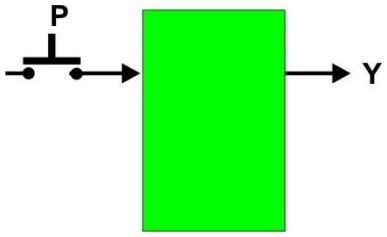


Diagrama de bloques

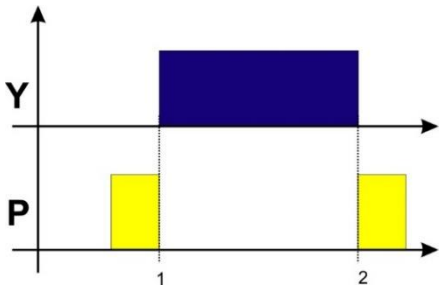
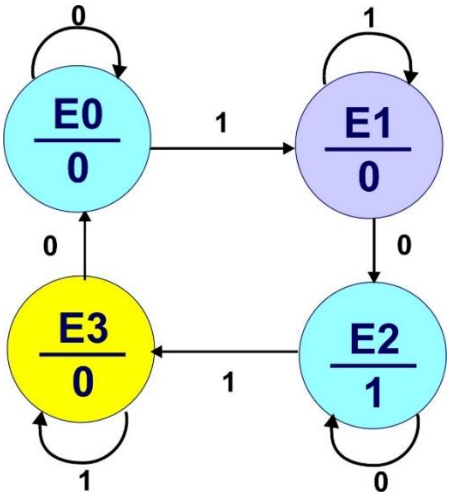


Diagrama de tiempos

1.- Diagrama de transición

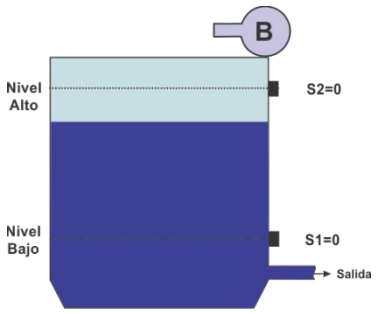


Circuito

Problema 30, Control de una bomba para el llenado de un tanque

Diseñe, efectúe la simulación y construya un prototipo de un sistema secuencial asíncrono para control de llenado de un tanque por medio de una Bomba (B), el tanque cuenta con dos sensores de nivel S2 detecta nivel alto y S1 detecta Nivel bajo.

El sistema deberá de trabajar con la siguiente lógica:



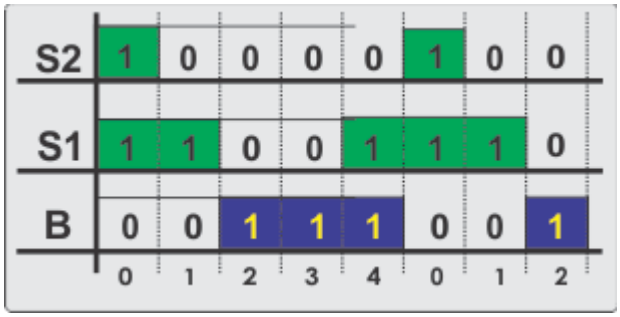
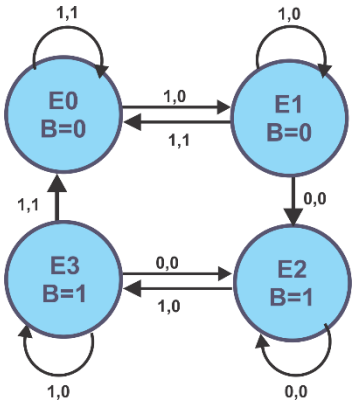
Tanque con sensores de nivel.

a) Partiendo que el tanque se encuentre lleno (S2=1, S1=1), la bomba B permanecerá apagada (B=0) y encenderá (B=1), hasta cuando el nivel del tanque esté vacío (S2=0, S1=0).

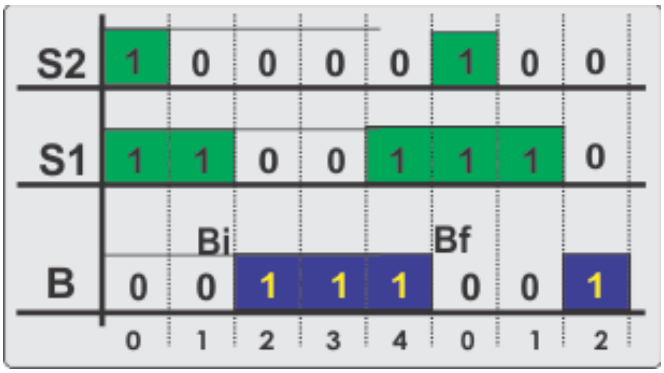
b)La bomba permanecerá encendida hasta que el nivel suba a tanque lleno (S2=1, S1=1).

1.- Especificar el Sistema.

Existen varios recursos para describir el funcionamiento del sistema secuencial, entre los que podemos mencionar son:

Diagrama de tiempos	Diagrama de Transición
	

.



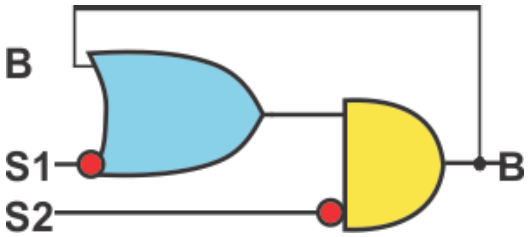
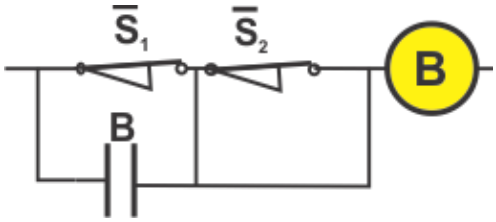
Obteniendo los valores de Bi y Bf

$$\underline{Bi = \mathbf{S1'} + B}$$

$$\underline{Bf = S2}$$

Aplicando el teorema fundamental secuencial $S = S^i$ **(S^f)' obtenemos**

B = (**S1'** + B) **S2'**

Diagrama esquemático	Diagrama escalera
	

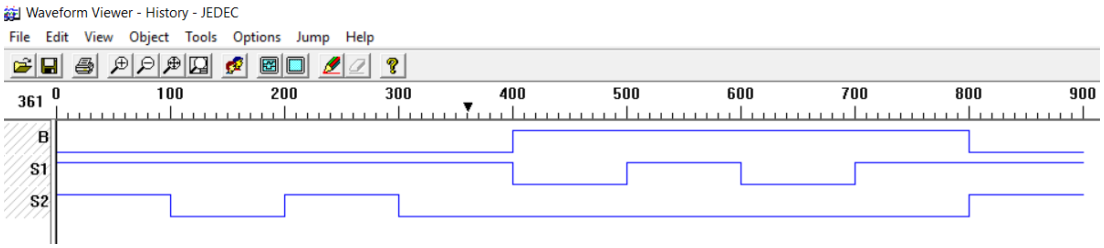
Código ABEL-HDL

```
MODULE pdnueve

"28 abril 2019
S2, S1 pin 1,2;
"Salidas
B pin 14 istype 'com';
equations
"Q+  =  S1' + Q S2'
B=(!S1 # B) &!S2;

test_vectors
([S2,S1]->[B])
[1,1]->[.x.];
[0,1]->[.x.];
[1,1]->[.x.];
[0,1]->[.x.];
[0,0]->[.x.];
[0,1]->[.x.];
[0,0]->[.x.];
[0,1]->[.x.];
[1,1]->[.x.];
END
```

Simulación Test_Vectors



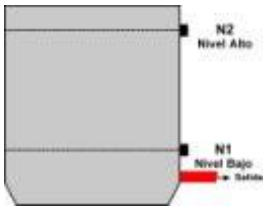
Simulación. PROTEUS

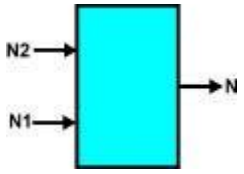
1	2	3
4	5	6
7	8	9

En el tiempo 1 (T1) partimos con tanque lleno (S_2=1, S_1=1) y la bomba apagada (B=0), en T2 el nivel pasa a nivel medio (S_2=0, S_1=1) y aun la bomba apagada, en T3 simulamos un oleaje de nuevo el nivel lleno y en T4 regresamos a nivel medio, en T5 se muestra por primera vez el nivel bajo (S_2=0, S_1=0) y el estado de la bomba cambia a B=1 encendiéndola, en T6 el nivel sube a medio (S_2=0, S_1=1) y la boba permanece encendida, en T7 simulamos un oleaje del nivel regresando a bajo (S_2=0, S_1=0) con la bomba encendida, en T8 de nuevo regresa a niel medio (S_2=0, S_1=1) manteniendo la bomba encendida B=1 y por último en T9 pasa a tanque lleno (S_2=1, S_1=1) apagando la bomba B=0. Demostrando por medio de la simulación el correcto funcionamiento del sistema propuesto.

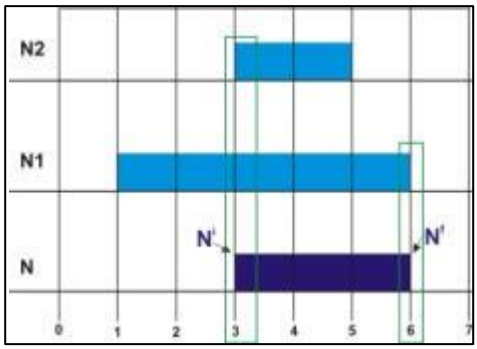
Problema 31, Detector de Nivel de un tanque

Diseñe un sistema secuencial asíncrono para detectar nivel de un tanque que cuenta con dos sensores llamados





N1 (Nivel Bajo), **N2** (Nivel Alto), se requiere de una señal de salida **N** *sea igual a cero* cuando el nivel va de **N1** hacia **N2** hasta que llegue a **N2** y **N** *sea igual a uno* cuando el nivel va de **N2** hacia **N1**. Hasta que llegue a **N1** como lo indica el siguiente diagrama de tiempos



Obtención de las Ecuaciones

La salida **N** inicia en T = 3 y se crea cuando **N2**=1, entonces la creación de la señal **N** está definida por $\mathbf{N^i}$ y depende del inicio de **N2** o también de la misma **N** para mantener la señal en un valor igual a uno. $\mathbf{N_i = N2 + N}$

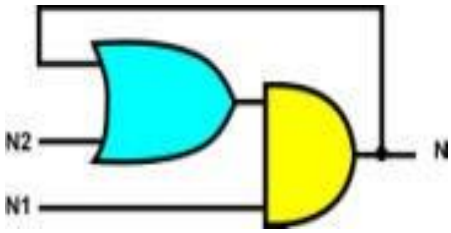
La salida **N** termina en T=6 porque **N1**=0 entonces la extinción de la señal **N** estará definida por $\mathbf{N^f}$ y depende solamente por la terminación **N1**. $\mathbf{N^f = (N1)'}$

Aplicando el teorema fundamental $\mathbf{N = N^i (N^f)'}$ obtenemos la ecuación para **N**

$$\underline{N = (N2 + N)\ \mathbf{(N1)''} =}$$

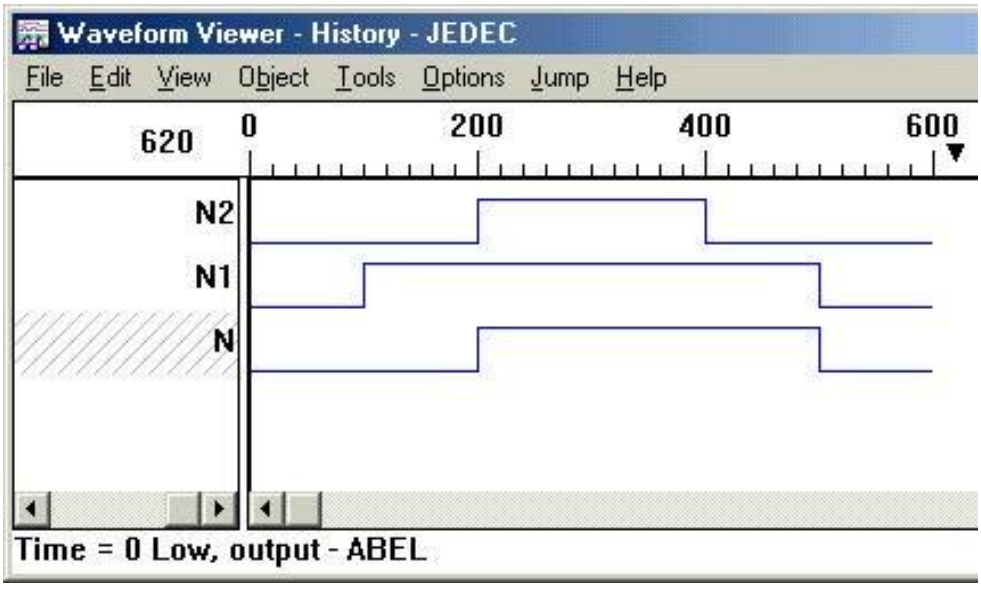
$$\underline{N = (N2 + N)N1}$$

CIRCUITO FINAL

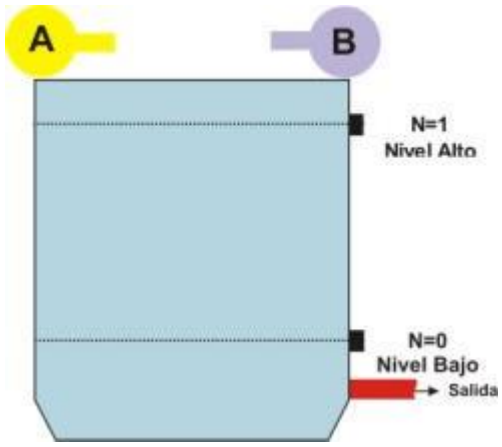


Obtenga el circuito implementado en un GAL22V10 utilizando la ecuación de **N** en ABEL y compruebe su funcionamiento.

```
MODULE nivel
"Entradas
 N1, N2 pin 1, 2 ;
"Salida
 N pin 19 istype 'com';
Equations
N = (N2 # N)&N1;
Test_vectors
([N1,N2]->[N])
[0,0]->[.x.];
[1,0]->[.x.];
[1,1]->[.x.];
[1,1]->[.x.];
[1,0]->[.x.];
[0,0]->[.x.];
END
```



Problema 32, Sistema alternativo de dos bombas no simultaneas



Diseñe un Sistema Secuencial que controle el llenado de un Tanque con las siguientes características:

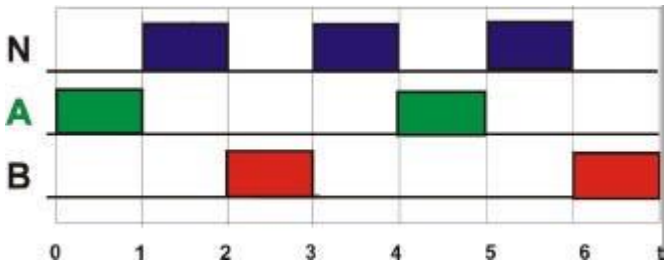
a) El sistema consta de dos bombas llamadas **"A"** y **"B**".

b) Un sensor de nivel **"N"** que indica con N=1 Tanque lleno y N=0 Tanque vacío.

c)Partiendo de que el Tanque se encuentra vacío, el llenado deberá iniciarse con la bomba "A" hasta llenar el Tanque (N=1) y entonces desconectar la bomba A.

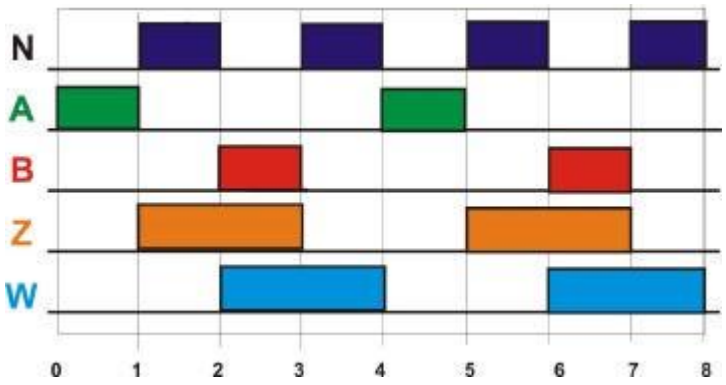
d)Si de nuevo se vacía el Tanque (N=0), el llenado deberá hacerse con la bomba "B" hasta llenar el Tanque y entonces desconectar la bomba B.

e) Si se vacía de nuevo el llenado deberá hacerse con la bomba "A" y así sucesivamente con la finalidad de que se las bombas se alternen en su funcionamiento.



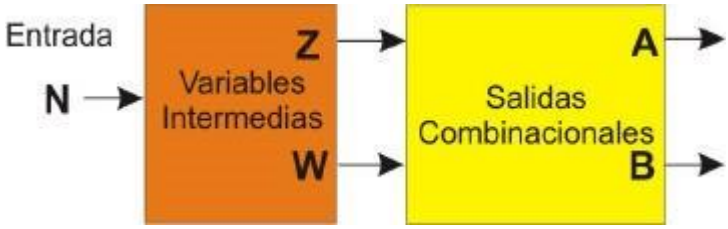
Solución:

Para distinguir cual fue la última bomba en trabajar es conveniente usar dos variables adicionales las llamaremos **W** y **Z** en donde ***Z=1*** desde que la bomba A deja de trabajar hasta que la bomba B deja de trabajar es decir del t=1 hasta t=3, y



W=1 desde que la bomba B empieza a trabajar hasta que la bomba A empiece también del t=2 hasta t=4.

Para obtener las ecuaciones deberemos de considerar que Z y W dependen de N y de ellas mismas mientras que A y B dependen del nivel N y las variables Z y W.



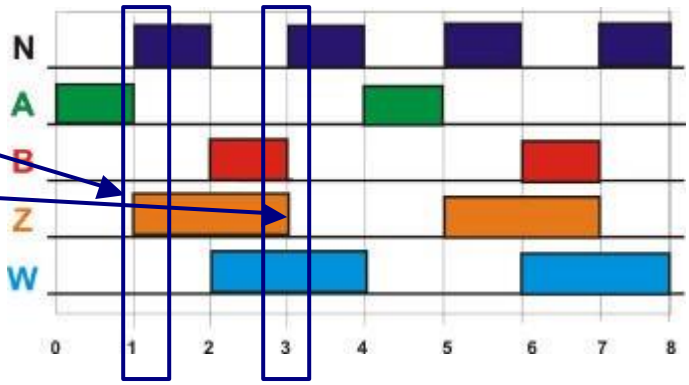
Ecuaciones:

Z inicia en t= 1

$Z^0 = N\,\mathbf{W'} + Z$

Z termina en t=3

$Z^1 = N\,W$



Aplicando el Teorema

Fundamental Secuencial obtenemos:

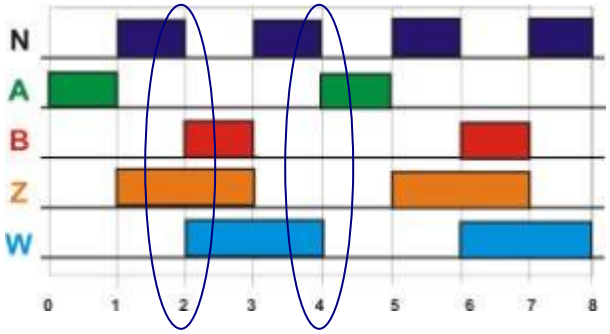
$$Z=(N\,\mathbf{W'} + Z)(\mathbf{N'} + \mathbf{W'})$$

W inicia en t=2

W^0= **N'** Z + W

W termina en t=4

W^1= **N' Z'**



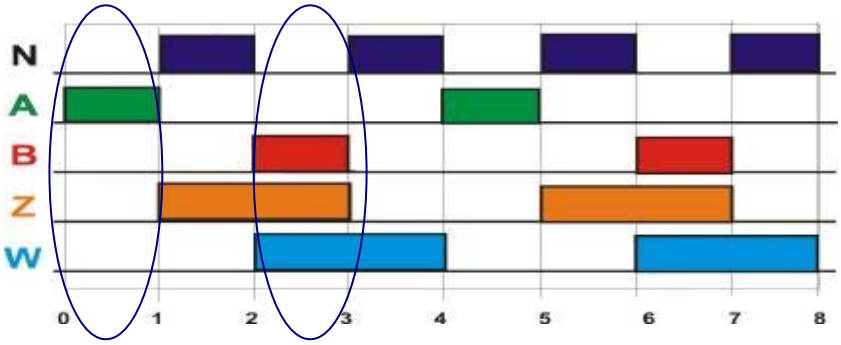
Aplicando el Teorema Fundamental Secuencial obtenemos:

W=(**N**' Z + w)(N + **Z**)

Las ecuaciones de A y B son tomadas como dependientes de N, Z y W de modo que:

A =**Z' W' N'**

B= Z W**N'**

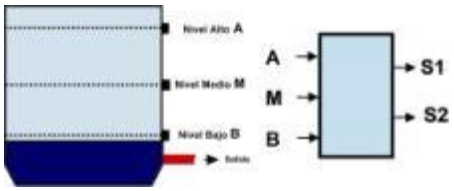


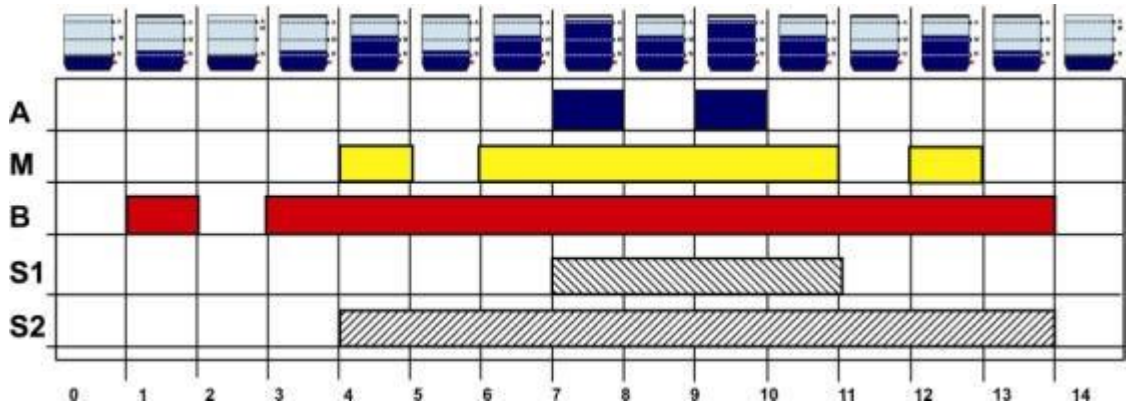
Archivo en ABEL-HDL

```
MODULE dosbom
"Entrada
 N pin 1;
"Salidas
 A,B,Z,W pin 19,18,17,16 istype 'com';
Equations
Z= ( N&!W#Z)&(!N#!W);
W = ( !N&Z#W)&(N#Z);
A= !Z&!W&!N; B= Z&W&!N;
test_vectors ([N]->[A,B])
[0]->[.x.,.x.];
[1]->[.x.,.x.];
[0]->[.x.,.x.];
[1]->[.x.,.x.];
[0]->[.x.,.x.];
[1]->[.x.,.x.];
END
```

Problema 33, Detector de tres niveles de un tanque

Diseñe un sistema secuencial asíncrono para la detección del nivel de un tanque que por medio de tres sensores llamados **B** bajo, **M** medio, **A** alto se obtengan las salidas S1 y S2 con los valores presentados en la siguiente gráfica.





Podemos observar en la gráfica que **S2** inicia cuando inicia **M** de modo que:

$S2^i = M + S2$ y S2 finaliza cuando termina **B** $S2^f = B'$

Aplicando el teorema fundamental obtenemos

$$S2 = (M + S2)\,B$$

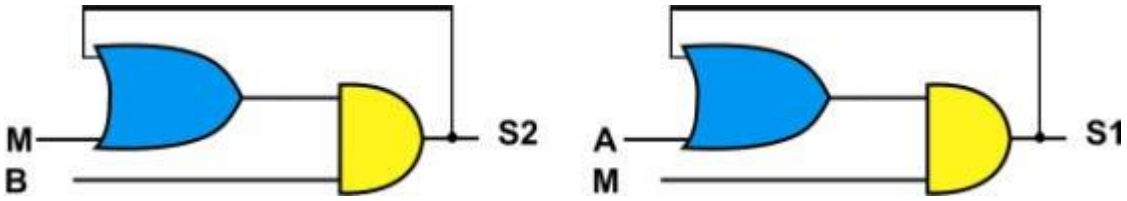
Podemos observar en la gráfica que **S1** inicia cuando inicia **A**

$S1^i = A + S1$, **S1** finaliza cuando termina M $S1^f = M'$

Aplicando el teorema fundamental obtenemos

$$S1 = (A + S1)\,M$$

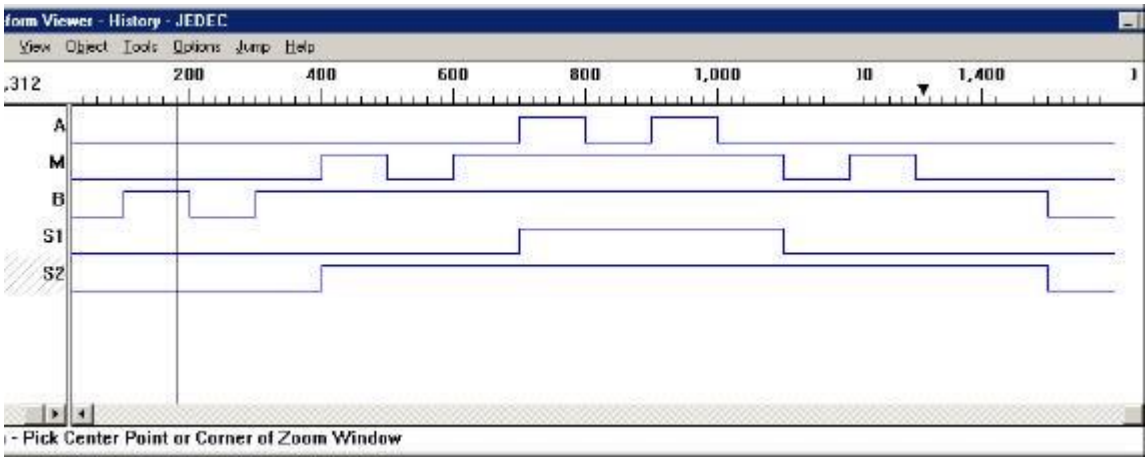
CIRCUITO DEL SENSOR DE NIVEL



Archivo en formato ABEL incluyendo simulación

```
MODULE nitres
"Entradas
 A,M,B PIN;
"Salidas
S1, S2 PIN ISTYPE 'COM';
EQUATIONS
S2 = (M # S2) &B; S1 = (A # S1) &M;
TEST_VECTORS
([A,M,B]->[S1,S2,])
[0,0,0]->[.x.,.x.];
[0,0,1]->[.x.,.x.];
[0,0,0]->[.x.,.x.];
[0,0,1]->[.x.,.x.];
[0,1,1]->[.x.,.x.];
[0,0,1]->[.x.,.x.];
[0,1,1]->[.x.,.x.];
[1,1,1]->[.x.,.x.];
[0,1,1]->[.x.,.x.];
[1,1,1]->[.x.,.x.];
[0,1,1]->[.x.,.x.];
[0,0,1]->[.x.,.x.];
[0,1,1]->[.x.,.x.];
[0,0,1]->[.x.,.x.];
[0,0,1]->[.x.,.x.];
[0,0,0]->[.x.,.x.];
END
```

Simulación


form Viewer - History - JEDEC
View Object Tools Options Jump Help
,312
200
400
600
800
1,000
1,400
A
M
B
S1
S2
- Pick Center Point or Corner of Zoom Window

Epílogo

Después de haber recorrido las páginas de "Diseño de Sistemas Secuenciales Asíncronos", es hora de reflexionar sobre el viaje educativo que hemos emprendido. Este libro ha sido una guía fundamental para estudiantes, profesores e ingenieros que buscan adentrarse en el apasionante mundo de los sistemas secuenciales asíncronos.

A lo largo de nuestra travesía, hemos explorado conceptos desafiantes y técnicas creativas para abordar sistemas cuyos eventos no están vinculados a un reloj común. Hemos aprendido que la sincronización y la coordinación son cruciales en el funcionamiento correcto de dispositivos y sistemas complejos, y hemos adquirido las habilidades analíticas necesarias para enfrentar estos desafíos únicos.

Una de las características más valiosas de este libro ha sido su enfoque en métodos alternativos. En lugar de limitarnos a una única perspectiva, hemos tenido la oportunidad de explorar una variedad de estrategias para resolver problemas relacionados con el diseño de sistemas asíncronos. Esta diversidad nos ha permitido ampliar nuestra comprensión y seleccionar las soluciones más apropiadas para cada situación.

Además, hemos profundizado en la aplicación práctica de nuestros conocimientos, descubierto cómo la simulación puede ser una herramienta esencial para validar y mejorar nuestros diseños antes de la implementación física, explorado la construcción de prototipos y adquirido las bases para programar Controladores Lógicos Programables (PLCs), conectando así los conceptos teóricos con el mundo real de la ingeniería.

Al concluir este viaje de aprendizaje emocionante y enriquecedor, debemos recordar que el conocimiento adquirido no es un destino final, sino un punto de partida para futuras exploraciones y logros. La ingeniería es un campo en constante evolución, y nuestras habilidades seguirán desarrollándose a medida que enfrentemos nuevos desafíos y abordemos proyectos más complejos.

En nombre de Juan Angel Garza Garza y de todos los que han contribuido en esta obra, les animo a seguir explorando, experimentando y aplicando lo que han aprendido. Que esta obra continúe siendo una herramienta valiosa en su trayectoria de formación profesional, ayudándoles a alcanzar el dominio de los conceptos vitales en el mundo de la ingeniería. ¡Adelante, ingenieros del futuro, el mundo está lleno de posibilidades y desafíos esperando ser conquistados!

Índice de figuras

Fig. 1. Gráfica del Método de detección y corrección de fallas 8
Fig. 2. Imagen gráfica de simulación por medio del Test_Vectors 9
Fig. 3. Imágenes de la Simulación por medio del programa PROTEUS 11
FIg. 4. Circuito asíncrono con flip flop's R-S 15
Fig. 5. Circuito asíncrono sin flip flop's 15
Fig. 6. Descripción a Bloques de un sistema secuencial asíncrono 17
Fig. 7. Diagrama de Tiempos 18
Fig. 8. Diagrama de transición 18
Fig. 9. Modelo de la máquina de Moore 20
Fig. 10. Modelo de la máquina de Mealy 20
Fig. 11. Diagrama de Transición 21
Figura 12. Estado o evento 21
Figura 13. Transición al mismo estado 21
Figura 14. Transición entre dos estados 21
Fig. 15 Diagrama de bloques 27
Fig. 16 Diagrama de tiempos 27
Fig. 17 Diagrama de Transición 28
Fig. 18 Diagrama de Transición e inicio de la tabla de flujo primitiva 29
Fig. 19 tabla de flujo primitiva completa 30
Fig. 20 tabla de estados internos 32
Fig. 21 Bloques del sistema 37
Fig. 21 Ciclos de las ocho formas estándar 40
Fig. 22 Ecuación, representación gráfica y diagrama escalera de la forma And/Or 41
Fig. 23 Ecuación, representación gráfica y diagrama escalera de la forma Nand/Nand 42
Fig. 24 Ecuación, representación gráfica y diagrama escalera de la forma Nor/Or 43
Fig. 25 Ecuación, representación gráfica y diagrama escalera de la forma Or/Nand 44
Fig. 26 representación gráfica de las primeras cuatro formas estándar 45
Fig. 27 Diagrama escalera de las primeras cuatro formas estándar 46
Fig. 28 Ciclo de las cuatro formas estándar correspondientes a POS 47

Fig. 29 Ecuación, representación gráfica y diagrama escalera de la forma Or/And 47
Fig. 29 Ecuación, representación gráfica y diagrama escalera de la forma Nor/Nor 48
Fig. 29 Imagen de la simulación Test_ Vectors 51
Fig. 30 Imágenes de la simulación en PROTEUS 51
Fig. 31 proceso de minimización en LogiAid 53