

Introducción

La empresa Bayer de México desea introducir en México la aspirina envasada en bote de plástico, por lo que le ha pedido al TEC de Monterrey el apoyo para el diseño de un sistema de control para llenado automático de los botes. Al hacerlo, la empresa incrementará sustancialmente la eficiencia en la operación de llenado y permitirá que el armado de lotes para venta se realice de forma continua. Para esto es fundamental conocer algunos conceptos que nos harán llegar con mayor facilidad a una solución para la empresa Bayer.

El uso de sensores los cuales son dispositivos que detectan un cambio en el entorno y responde a alguna salida en otro sistema, convierte un fenómeno físico en un voltaje medible o una señal digital el cual se visualizará en una pantalla legible o transmitida para su lectura existen diferentes tipos de sensores como de contacto, ópticos, térmicos, de humedad, magnéticos e infrarrojos estos dispositivos serán una herramienta fundamental para una posible solución debido a que necesitamos saber el momento exacto cuando se detecte el llenado del bote. Los actuadores son dispositivos que transforman diferentes tipos de energía para generar algún funcionamiento dentro de un sistema automatizado, normalmente estos actuadores generan una fuerza mecánica a partir de distintos tipos de energía, ambos dispositivos antes mencionados son parte de un sistema de control.

Máquinas de estados finitos es una abstracción computacional que describe el comportamiento de un sistema reactivo mediante un número determinado de estados y de transiciones entre dichos estados, estas transiciones generan en respuesta a eventos de entrada externos e internos. Hay dos tipos de máquinas de estado:

- Máquinas de Moore: las salidas solo dependen del estado presente y las entradas intervienen en la decisión del próximo estado.
- Máquina de Mealy: Las salidas dependen del estado presente y del valor de las entradas.

Los circuitos digitales están contruidos por otros circuitos más simples, interconectados entre sí, estos son los circuitos lógicos combinatorios son aquellos circuitos digitales que en cualquier tiempo el estado de las salidas solo depende de los valores de las entradas en ese momento y no de las entradas anteriores, estos circuitos tienen N entradas y P salidas, cada una lleva a cabo una función digital o lógica este tipo de circuitos los vemos en comparadores, codificadores y multiplexores; También podemos encontrar los circuitos lógicos secuenciales los cuales funcionan de manera contraria a los combinatorios, pues estos circuitos dependen de su entrada para cambiar su salida, de su estado anterior y en algunas ocasiones de un pulso de reloj, hay dos tipos de circuitos, los síncronos que necesitan de un pulso de reloj para cambiar su estado y los asíncronos donde no es necesario un pulso de reloj debido a que el sistema cambia de manera automática cuando sus entradas sufren una modificación, ejemplo de esto son los flip-flops.

Por medio de lenguajes de descripción de hardware nos ayudan a describir el comportamiento de un sistema mediante palabras y símbolos, dentro de este lenguaje existen diferentes tipos de

abstracción como el “behavioral”, RTL (diseño de sucesión de registros conectados por lógica combinacional) y estructural (switches y compuertas), un ejemplo de estas aplicaciones es la programación VHDL que permite una descripción de la estructura del circuito y especificaciones de la funcionalidad , unido a esto tenemos los dispositivos lógicos programables los cuales son componentes electrónicos usados para construir circuitos digitales reconfigurables, estos dispositivos nos brindan ventajas como la integración de una gran cantidad de funcionalidades en un solo chip, existen diferentes arquitecturas de los PLDs, como arreglo lógico programable (PAL) que nos permite programar un arreglo de AND y OR se mantiene fijo, facilitando su desarrollo, a pesar de esto, esta estructura fue desapareciendo con la aparición de otros dispositivos como los FPGAs los cuales son circuitos integrados que contienen un arreglo de bloques lógicos (basados en LUTs), bloques de entrada/salida e interconexiones programables para establecer funciones que se realizarán por cada bloque lógico según las necesidades del usuario, una ventaja del uso de FPGAs es la integración de RAM, procesadores, entre otros sistemas, estos dispositivos utilizan lenguaje VHDL debido a sus pruebas de funcionalidad a nivel hardware antes de su implementación .

En cuanto a las memorias RAM las arquitecturas son:

- Random Access Memory (RAM): memoria usada por el CPU donde se almacena programas y datos, es volátil, son construidas por SRAM y DRAM al igual que implementa estas dos tecnologías en computadoras.
- Read Only Memory (ROM) : Almacena información necesaria para operar el sistema, no es volátil y se almacena en sistemas embebidos, utiliza ROM, PROM, EPROM, EEPROM y memoria Flash.
- Dynamic RAM (DRAM) : capacitor que almacena datos.
- Static RAM (SRAM): transistor que forma flip-flops para almacenamiento de datos.

En relación a los FPGAs existen bloques especializados para la memoria RAM para almacenar datos necesarios de un proceso la capacidad de esta depende del fabricante.

Análisis y descripción del problema

El proceso de la embotelladora se realiza sobre una plataforma giratoria en la que una cierta cantidad de envases son colocados. Una vez que un envase está listo, una persona retira el envase sellado manualmente y coloca otro en la posición desocupada. El funcionamiento que lleva a cabo la máquina embotelladora de pastillas se puede describir por una máquina de estados finitos, en la que para pasar al siguiente estado es necesario cumplir con una condición. Dicha máquina de estados finitos se encuentra descrita por el siguiente diagrama de flujo.

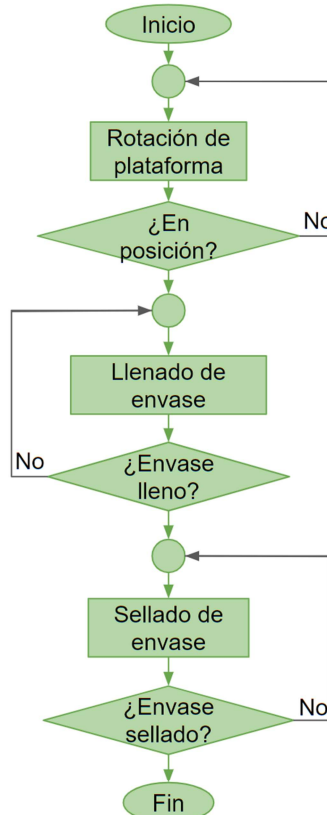


Figura 1. Mapa de flujo de maquina de estados

Como se puede observar en el diagrama de flujo, la máquina de estados finitos tiene tres estados, rotación, llenado y sellado, el proceso de la embotelladora es el siguiente. Inicia y por medio de un sensor se verifica que el envase esté en la posición correcta, si es así por medio del actuador pasa al llenado de envase, caso contrario regresa y rota la plataforma. En el llenado del envase verifica que esté lleno y pasa al sellado de envase, en caso de que no esté lleno continúa llenándolo. En el sellado del envase verifica que esté el envase sellado y termina el proceso, si el envase no se ha sellado continúa en el proceso de sellado.

Desarrollo de la propuesta de solución

Para su implementación se optó por hacer uso de una maquina de estados finita la cual se llevará a cabo por medio de código en la herramienta de Quartus y a su vez se utilizará la tarjeta DE10-Lite para corroborar su funcionamiento de manera física.

Para comenzar tenemos que nuestro sistema estará dado por tres diferentes estados, en el código los podemos ver definidos como S_0, S_1 y S_2 las cuales corresponden a la rotación, llenado y sellado correspondientemente. Se hace uso de tres entradas las cuales serán reloj, reset y X, esta última funcionara para el monitoreo de la condición ya que esta puede ser “se cumple” o “no se cumple” ‘0’ y ‘1’ respectivamente.

Para esta implementación se decidió utilizar un registro con reloj activo en alto y un reset asíncrono, en el cual se asignará el estado siguiente en la transición positiva del ciclo de reloj, y dado que se

implemento un reset disparado en alto '1' este deberá tener un uno lógico a la entrada para que el reset se dispare y la maquina nos devuelva al estado cero el cual sería la rotación. No obstante, se tuvo que tener en cuenta como se tenía que implementar este CLK ya que nuestra tarjeta cuenta con un CLK de 50Mhz en el pin P11 pero se pensó en esta opción y se descartó, ya que los cambios serían muy rápidos por lo cual se optó utilizar un switch que tuviera el funcionamiento de CLK al momento de realizarlo de manera física.

Tomemos el ejemplo del estado 1 (S0) en donde tenemos:

```

WHEN S0=>
  S_0<='1';
  S_1<='0';
  S_2<='0';
  IF X>='1' THEN
    NextState<=S1;
  ELSE
    NextState<=S0;
  END IF;

```

En el cual se puede observar que S_0 está en '1' en cual nos esta indicando que se encuentra en rotación y los estados restantes se mandan a '0', entonces si nuestro pin de monitoreo recibe un '1' este pasará al siguiente estado S1 el cual sería el llenado, y mientras no reciba un '1' nuestra "X" estará en el estado de rotación.

Los siguientes estados contienen la misma estructura del estado anterior, claramente en estos irán cambiando los estados dependiendo en el estado en que se encuentre.

Para corroborar el funcionamiento de nuestro código se decidió realizar un testbench para poder asignar valores y ver como este se comporta. En este se creó un loop para la generación de nuestro CLK y se mandaron datos a X para poder simular el proceso de sensado de cada estado. Se obtuvieron los siguientes resultados.



Figura 2. Resultados de la simulación.

Podemos observar en las áreas marcadas como nuestros estados van cambiando de manera ordenada dependiendo el estado en el que se encuentre "X", por ejemplo, en el rectángulo rojo se observa que X='0' pero este cuando recibe '1', nos enviara al próximo estado, en este caso S1 el cual como el ejemplo anterior cuando X sea '1' cambia al estado S2 y este proceso se repite teniendo rotado, llenado y sellado. Cabe mencionar que el estado se mantendrá, siempre y cuando X no reciba el valor de '1' que indica que el proceso se completó.

Con las simulaciones en orden se procedió a la implementación de forma física en la tarjeta DE10-Lite por lo cual se hizo la siguiente configuración de pines. Haciendo énfasis en el uso de un switch de la tarjeta para la función del reloj y se descarto el uso del CLK interno de la tarjeta de 50MHz.

Control	Node name	Location	I/O Standard
Estado 0: Rotación	LEDR[0]	PIN_A8	3.3 LVTTL
Estado 1: Llenado	LEDR[1]	PIN_A9	3.3 LVTTL
Estado 2: Sellado	LEDR[2]	PIN_A10	3.3 LVTTL
Clock	SW[0]	PIN_C10	3.3 LVTTL
Reset	SW[1]	PIN_C11	3.3 LVTTL
Condición: X	SW[2]	PIN_D12	3.3 LVTTL

Al implementar la maquina de estados en la tarjeta surge la duda de cómo podríamos sensorar el correcto llenado de pastillas. Para esto se pueden utilizar diferentes métodos como lo podrían ser el sensado de peso, el sensado de nivel, el sensado del tiempo, entre otros. En lo personal optaría por el sensado de peso, implementándolo en la plataforma giratoria, por ejemplo, tener el peso de las botellas vacías que pueden caber en dicha plataforma y tomar ese valor como referencia, que al poner por ejemplo una pastilla de 10gr será el peso total sensado igual a [(peso de empaques vacíos) + peso de las pastillas]] y así configurar un peso relacionado con un numero de pastillas. El sensor a utilizar podría ser una galga extensiométrica la cual cambia su valor de resistencia conforme a la distorsión de su forma.

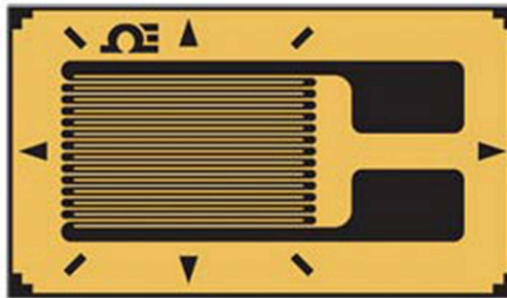


Figura 3. Galga extensiométrica.

Para el funcionamiento de la implementación se puede observar el siguiente video:

https://drive.google.com/file/d/14-fD1bY-wD9PmC3nrBey8FGsNaHd_PML/view?usp=sharing

El código de la maquina de estados como el del testbench se encuentra en la siguiente liga:

<https://github.com/lapr98/Situacion-Problema.git>

Conclusión:

El tener una problemática real referente a un sistema real como lo es en esta situación el llenado de botellas ayuda al mejor planteamiento de la solución que se va a emplear. Se pudo llevar a cabo la implementación de una máquina de estados capaz de realizar un proceso, esto a través del desarrollo de un código en VHDL y puesto en práctica de manera física por medio de la tarjeta de desarrollo DE10-Lite. El lenguaje VHDL nos ayuda a una mejor implementación de sistemas lógicos combinatorios y secuenciales ya que en este se sigue un mejor orden y mejor desarrollo de dichos sistemas, cabe mencionar que el poder simular y ver el comportamiento de nuestro sistema de manera grafica y secuencial es de gran aporte a la hora de buscar fallas o un mal funcionamiento. La búsqueda

Bibliografía.

Dispositivos lógicos programables | (Capitulo 2) Recuperado de <http://www.ptolomeo.unam.mx:8080/jspui/bitstream/132.248.52.100/658/8/A8.pdf>

¿Qué es un Sensor y Qué Hace? | Dewesoft. (s. f.). Recuperado de <https://dewesoft.com/es/daq/que-es-un-sensor>

Tecnologías, G. (s. f.). Arquitectura FPGA - GENERA Tecnologías. Recuperado de https://www.generatetecnologias.es/arquitectura_fpga.html

Mecafenix, I. (2020, noviembre 24). ¿Cuál es la diferencia entre los circuitos combinatoriales y secuenciales? Ingeniería Mecafenix. <https://www.ingmecafenix.com/electronica/circuitos-combinacionales-y-secuenciales/>

Orozco, J. A. G. (s. f.). Máquinas de Estados Finitos. 16.