**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**

**FACULTAD DE ESTUDIOS GENERALES CIENCIAS**



**PROYECTO CURSO 1IEE04 – DISEÑO DIGITAL**

**MÁQUINA EXPENDEDORA DE GASEOSAS**

**Presentado por el grupo MR-01**

**Integrantes**

**20170995 – Pacheco Auquis, Jose Maria**

**20171781 - Villavicencio Muñoz, Angel Arturo**

**20172058 – Boza Laurente, Brigith Forayma**

**20175456 – Gutierrez Hernandez, Antoni Máximo**

**20175637 - Lopinta Mucha, Dimar Anthony**

**Lima - Perú**

**Julio 2019**

**RESUMEN**

El presente trabajo consiste en el desarrollo e implementación de una máquina expendedora de gaseosas. Esta estará conformada por 3 módulos: uno que se encargará de la detección de las teclas presionadas en el teclado matricial a emplear, otro que se encarga de todas las operaciones de la máquina mediante una FSM-D y otro que se encarga de mostrar el funcionamiento de la máquina en un monitor mediante VGA.

La máquina venderá 3 productos: agua, Pepsi y Coca-cola, siendo sus precios 1.5, 2 y 2.5 soles respectivamente. Se podrán ingresar monedas de 0.5, 1, 2 y 5 soles (hasta 5 monedas de cualquier tipo). Asimismo, la máquina contará con dos modos de operación, uno cuando sea capaz de dar vuelto y otro en el que no. Por otra parte, se visualizará en un monitor el dibujo de la máquina con los productos a disposición (agua, Pepsi y Coca-Cola), la cantidad disponible de estos y su precio respectivo, el monto que se irá ingresando, el modo de operación de la máquina (si puede dar vuelto o no) y el vuelto que daría.

La descripción de hardware de la arquitectura diseñada, se realizó por medio del lenguaje de descripción de hardware VHDL en el entorno de desarrollo Quartus Prime 15.1 Lite Edition. Esto fue validado por medio de la herramienta *RTL Simulation,* con el software ModelSim-Altera.

El éxito no está en vencer siempre sino en no desanimarse nunca.

(Napoleón Bonaparte)**ÍNDICE**

**CAPÍTULO 1: DISEÑO DE LA ARQUITECTURA PROPUESTA**

**1.1. Introducción**

**1.2 Módulo de detección de teclado**

**1.2.1. Comportamiento funcional del módulo de detección de teclado**

**1.2.2. Diseño del módulo de detección de teclado**

**1.2.3.Simulación del módulo de detección de teclado**

**1.3. Módulo controlador (VGA)**

**1.3.1. Comportamiento funcional del VGA**

**1.3.2. Diseño del VGA**

**1.3.3. Simulación del módulo VGA**

**1.4 Máquina de estados principal(Módulo de FSM con Datapath)**

**1.4.1.Comportamiento funcional de la FSM**

**1.4.2. Diseño de la FSM**

**1.5. Resultados de la arquitectura completa.**

**1.5.1.Comportamiento funcional**

**CAPÍTULO 2 :RESULTADOS EXPERIMENTALES**

**CAPÍTULO 3 :CONCLUSIONES Y RECOMENDACIONES**

**3.1. Conclusiones**

**3.2. Recomendaciones**

**ANEXO A - PLANOS**

**ÍNDICE DE FIGURAS**

1. **Interacción del teclado con el FPGA**
2. **Módulo detección de teclado**
3. **Simulación de detección de teclado**
4. **Módulo controlador VGA de la máquina expendedora de gaseosas**
5. **Submódulo vga\_sync del controlador VGA**
6. **Simulación del VGA**
7. **Módulo máquina de estado**
8. **Diagrama de estados de la FSM**
9. **Máquina expendedora de gaseosas en puerto VGA en estado inicial**
10. **Selección de agua**
11. **Selección de Pepsi**
12. **Selección de Coca-Cola**
13. **Primer ingreso de moneda**
14. **Acumulación del ingreso de monedas hasta 8.5 soles**
15. **Se despacha el producto elegido**

**ÍNDICE DE TABLAS**

1. **Parámetros del módulo de detección de teclado**
2. **Parámetros del submódulo vga\_sync**
3. **Parámetros del VGA**

**LISTA DE ABREVIATURAS**

|  |  |
| --- | --- |
| FPGA | Arreglos de Compuertas Programables en el Campo |
| FSM | Máquina de estados finitos |
| HDL | Lenguaje de Descripción de Hardware |
| PIXEL | Elemento de imagen |
| VGA | Matriz de gráficos de vídeo |
| RTL | Register-Transfer Level |
| VHDL | Lenguaje de Descripción de Hardware de Circuitos Integrados  de alta velocidad |

**INTRODUCCIÓN**

En la actualidad el VHDL se ha vuelto un lenguaje trascendental para el desarrollo tecnológico. Este es un lenguaje de descripción de hardware que permite describir tanto circuitos síncronos como asíncronos y gracias a él se pueden realizar múltiples sistemas como una alarma, un semáforo o una máquina expendedora de dulces o bebidas. Este lenguaje inicia a quiénes están aprendiéndolo en el mundo de las tecnologías e inspira el desarrollo y creación de nuevas invenciones.

En particular, el presente proyecto del curso 1IEE04 – Diseño Digital tiene por objetivo el diseño de una máquina expendedora de gaseosas implementada sobre FPGA. Además, se espera que los conocimientos teórico-prácticos que se adquirieron a lo largo del semestre durante las clases y laboratorios puedan ser aplicados y finalmente plasmados en el desarrollo de este proyecto, pues es importante que los conocimientos adquiridos por los alumnos pueda ser interiorizado y visto como algo tangible y que no quede como simple teoría sin utilidad . A continuación , se explicará el desarrollo de una máquina expendedora de gaseosas en VHDL y la tarjeta Altera DE10 Lite.

El primer capítulo se enfocará en el desarrollo del diseño de toda la arquitectura propuesta, en el segundo se mostrarán los resultados experimentales; es decir, algunas pruebas de funcionamiento y finalmente, se plantearán algunas conclusiones.

Los módulos descritos en la arquitectura total son:

* Un módulo de detección de teclado
* Un módulo controlador VGA
* Un módulo FSM con datapath

**CAPÍTULO 1: DISEÑO DE LA ARQUITECTURA PROPUESTA**

**1.1. Introducción**

Este capítulo explicará el diseño de los módulos descritos para poder obtener una arquitectura dedicada para una máquina expendedora de gaseosas implementada sobre FPGA, tal como fue mostrado en la figura 1 de la introducción.

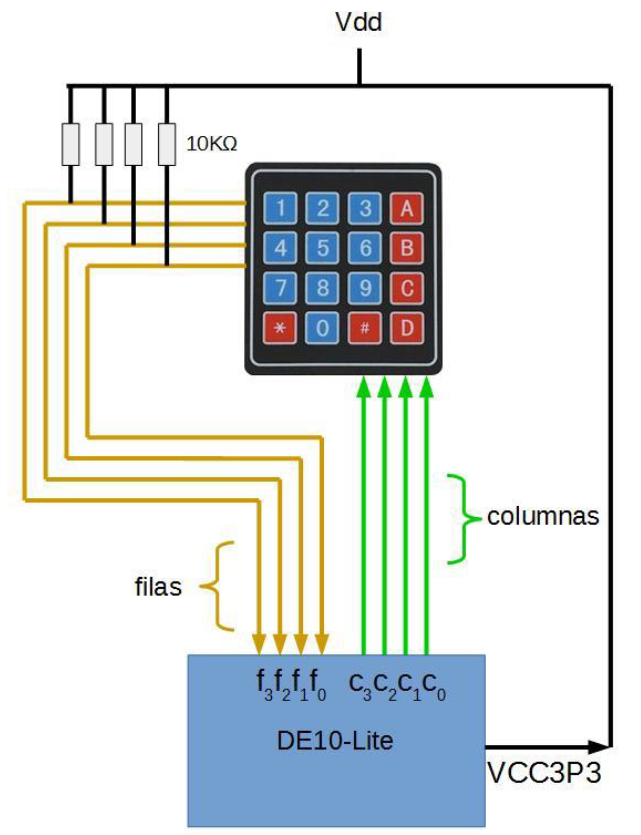
**1.2 Módulo de detección de teclado**

**1.2.1 Comportamiento funcional del módulo de detección de teclado**

Este módulo se encarga de detectar mediante coordenadas la tecla presionada. Esto se logra debido a la habilitación de una columna por vez para después poder detectar la fila de la tecla presionada.

El reconocimiento de las columnas y las filas sigue un estándar de lógica negada para poder clasificar qué coordenadas se están seleccionando.

El teclado deberá mostrar la tecla presionada en un display del FPGA y mientras no se haya presionado nada, el display mostrará “n”. Ademàs, la tecla “#” corresponde al valor “F” en el display y la tecla “\*” corresponde al valor de “E”.



**Figura 1: Interacción del teclado con el FPGA**

En la figura 2 se muestran las entradas y salidas de este módulo.



**Figura 2: Módulo detección de teclado.**

**1.2.2 Diseño del módulo de detección de teclado.**

Para la habilitación de las columnas necesitaremos un contador módulo 4 (pues son 4 columnas en el teclado), la salida de este contador será la entrada del decodificador 2 a 4 que siempre estará activado. La salida de este indicará la columna del teclado en alta (la salida “1 0 0 0” corresponde a la primera columna del teclado); sin embargo, al teclado matricial que se emplea trabaja en baja (lógica negada). Para que funcione como se espera, se deben negar las salidas del decodificador que se volverán la salida final de columnas que, a su vez, será entrada en el teclado. Por otra parte, al contador que se está usando trabaja con la frecuencia del FPGA, pero se quiere que esta cuenta sea cada 0.1 segundo. Por esto, se añade un habilitador al contador que es la salida de un divisor de frecuencia de 5 Millones

Por otro lado, en cuanto a las filas debemos negar cada bit de la entrada (por el mismo motivo que se negó la salida del decodificador para obtener las columnas) y sincronizarlo con el resto de los circuitos para no obtener un retraso .El sincronizador estará activo paralelamente al contador; es decir, estará controlado por un habilitador que es al mismo que se usa para al controlador. Luego , la señal de las filas ya sincronizada se usará como entrada de un codificador 4 a 2 y su salida como el selector de multiplexores que tienen como entrada a los valores de cada botón del teclado matricial (en esta parte se selecciona la fila). Seguido de esto, se elegirá la columna de donde se detectó la tecla. Para ello, se necesitará un multiplexor que tendrá como selector la resta entre el valor del contador y 2, debido a que este es quien asignará cada columna. Asimismo, la salida del último multiplexor mencionado será la entrada del circuito Hexa, el cual se encargará de convertir los valores que están en hexadecimal a decimal en el display de la tarjeta.

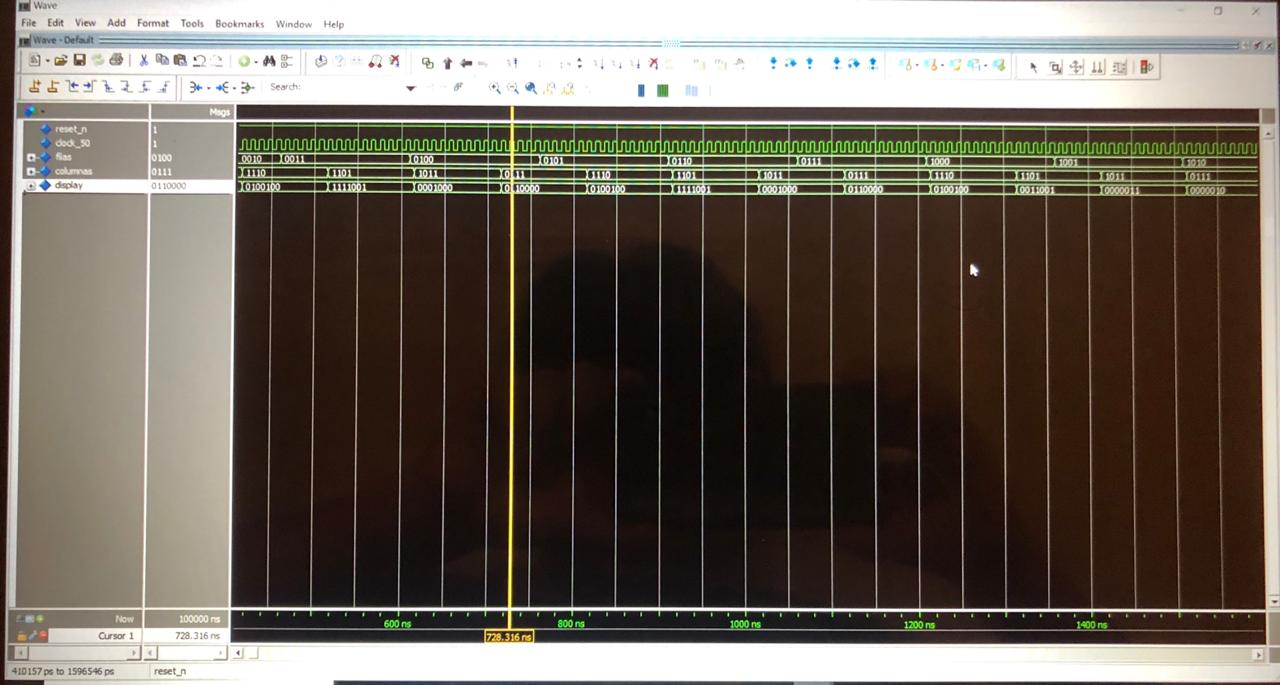
Finalmente, la salida del bloque Hexa pasará a un último multiplexor de 2 a 1, donde la otra entrada será el valor correspondiente para que se muestre “n” y cuyo selector será la salida negada del codificador de las filas que indica si se ha presionado alguna tecla.

La tabla 1 presenta la descripción de los parámetros de este módulo.

**Tabla 1: Parámetros del módulo de detección de teclado.**

|  |  |  |
| --- | --- | --- |
| **Entrada** | **Función** | **Cantidad de bits** |
| reset\_n | Señal de reinicio asíncrono del módulo, activa por baja | 1 |
| clk | Señal de sincronía de reloj, funciona a una frecuencia de 50MHz | 1 |
| filas | Señal que indica la fila seleccionada | 4 |
| **Salida** | **Función** | **Cantidad de bits** |
| display | Señal que sirve para mostrar la tecla presionada | 7 |
| columnas | Señal que indica la columna seleccionada | 4 |

**1.2.3.Simulación del módulo de detección de teclado**



**Figura 3: Simulación de detección de teclado**

**1.3. Módulo controlador (VGA)**

**1.3.1 Comportamiento funcional del VGA**

Esta parte del proyecto sirve para mostrar en pantalla las operaciones que el usuario realice en la máquina ;es decir, mostrará lo que este presione en el teclado , para lograr esto se realizó un circuito secuencial ,el cual muestra líneas negras que bordean a bloques de colores y dentro de estos se puede observar los nombres de los productos , la cantidad que hay de estos y el precio que ingresa el usuario ( en cada uno de estos bloques se usan visualizadores de 7 segmentos , los cuales irán cambiando mediante la variaciòn de cantidad de productos y el dinero que el usuario ingresará para comprar un determinado producto) .El rectángulo que corresponde al agua es de color verde, el correspondiente a Pepsi es azul y el de Coke es rojo. El último rectángulo ,en cambio, cambiará de color ,estará de color gris cuando no se ha entregado al producto y se pone de color amarillo cuando el usuario ha recibido el producto , los bloques celestes mostrarán al saldo y al vuelto . De esta manera es como funciona la máquina.

En la figura 4 se muestran las entradas y salidas de este módulo



**Figura 4 : Módulo controlador VGA de la máquina expendedora de gaseosas**

**1.3.2. Diseño del VGA**

Se empleó un submódulo “vga\_sync” que se encarga de recorrer todos los píxeles de la pantalla, que se puede dividir en una “zona visible” y otra “no visible”. Para esto se emplean 3 salidas en el submódulo: VGA\_BLANK, VGA\_HS Y VGA\_VS cuyo funcionamiento específico se puede ver en la tabla 2. Además, contará con 2 señales más de salida (count\_v y count\_h) que indicarán en qué posición de la pantalla se encuentra. Para esto se necesita que la frecuencia de refresco de la pantalla coincida con la cuenta de cada pixel, se calcula la “frecuencia de píxel” mediante la siguiente fórmula:

Donde N es la cantidad de píxeles en total (tanto de la zona visible como los de la zona no visible).

La tabla 2 presenta la descripción de los parámetros del submódulo del VGA llamado vga\_sync.

**Tabla 2: Parámetros del submódulo vga\_sync.**

|  |  |  |
| --- | --- | --- |
| **Entrada** | **Función** | **Cantidad de bits** |
| reset\_n | Señal de reinicio asíncrono del módulo, activa por baja | 1 |
| clock\_50 | Señal de sincronía de reloj, funciona a una frecuencia de 50MHz | 1 |
| **Salida** | **Función** | **Cantidad de bits** |
| VGA\_HS | Señal que indica el sincronismo horizontal | 1 |
| VGA\_VS | Señal que indica el sincronismo vertical | 1 |
| VGA\_blank | Señal que indica si el recorrido de la pantalla se encuentra en una zona no visible | 1 |
| count\_v | Señal que indica la cuenta de las filas | 10 |
| count\_h | Señal que indica la cuenta de las columnas | 10 |

En la figura 4 se muestran las entradas y salidas del submódulo del VGA

****

**Figura 5: Submódulo vga\_sync del controlador VGA**

En el caso del monitor empleado, se tiene 800x520 píxeles (la zona visible solo es 640x480 de los píxeles) por lo que f\_pixel=24960000 Hz, lo que se puede aproximar a 25 MHz. Para conseguir esta frecuencia en la cuenta de los píxeles, se emplea un divisor de frecuencia de factor 2 (pues la frecuencia del reloj del FPGA es de 50 MHz). La salida de este funcionará como un habilitador en los 2 contadores que se emplean para llevar a cabo la cuenta. Ambos contadores serán de 10 bits, uno contará de 0 a 799 que serían las 800 columnas que se tienen en total, mientras que el otro contará de 0 a 519 que serían las 520 filas que se tienen en total.

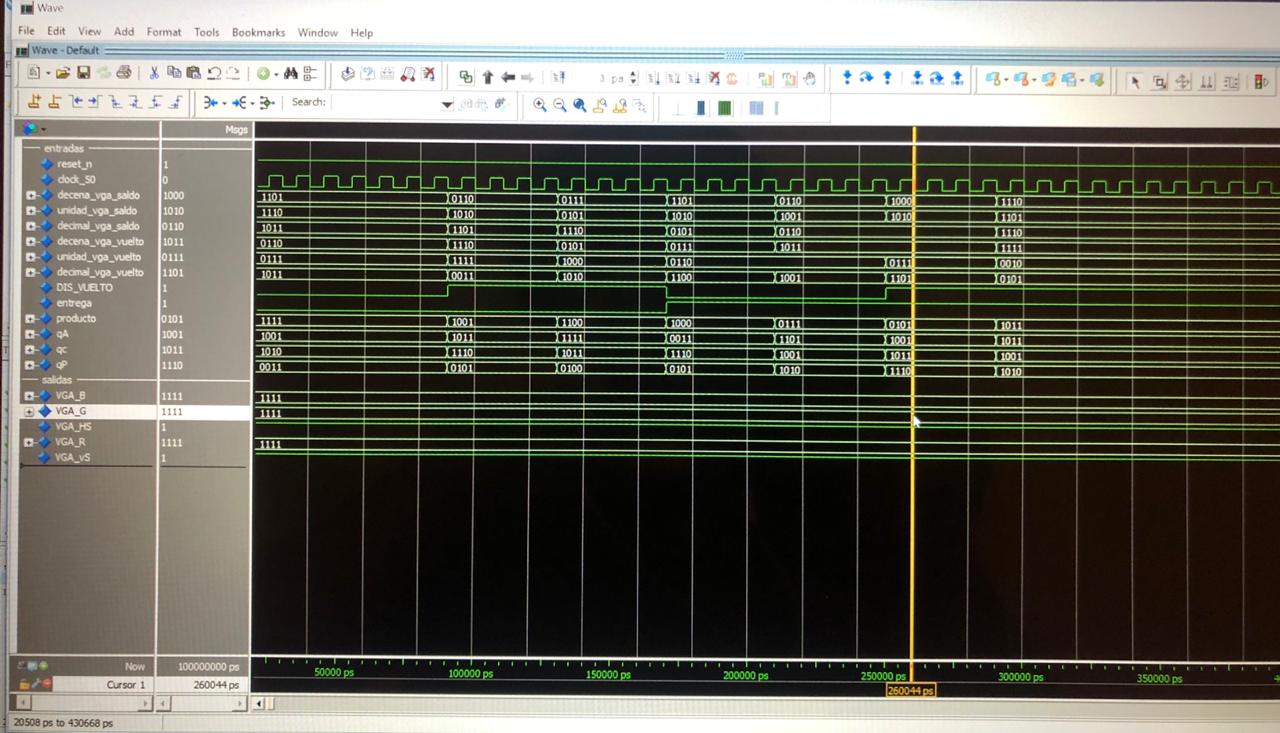
Como ya se mencionó, en el módulo principal del VGA se utiliza el sub-módulo “vga\_sync”, cuyas salidas VGA\_HS Y VGA\_VS serán las misma que del módulo principal. Por su parte count\_v, count\_h y VGA\_BLANK se emplean para el funcionamiento del módulo en sí. Toda la logica de este modulo se basa en comparaciones. Por ejemplo, si los contadores se encuentran en la zona donde debe ir la cantidad de botellas de agua y la señal qA es “1001”, entonces aparecerá en el monitor el número 9.

La tabla 3 presenta la descripción de los parámetros del módulo VGA.

**Tabla 3: Parámetros del VGA.**

|  |  |  |
| --- | --- | --- |
| **Entrada** | **Función** | **Cantidad de bits** |
| reset\_n | Señal de reinicio asíncrono del módulo, activa por baja | 1 |
| clock\_50 | Señal de sincronía de reloj, funciona a una frecuencia de 50MHz | 1 |
| entrega | Señal que indica si el producto fue entregado | 1 |
| DIS\_VUELTO | Señal que indica si la máquina es capaz de dar vuelto o no | 1 |
| decena\_vga\_saldo | Señal que indica el valor de las decenas del saldo que se ingresa | 4 |
| unidad\_vga\_saldo | Señal que indica el valor de las unidades del saldo que se ingresa | 4 |
| decimal\_vga\_saldo | Señal que indica el valor de la parte decimal del saldo que se ingresa | 4 |
| decena\_vga\_vuelto | Señal que indica el valor de las decenas del vuelto | 4 |
| unidad\_vga\_vuelto | Señal que indica el valor de las unidades del vuelto | 4 |
| decimal\_vga\_vuelto | Señal que indica el valor de la parte decimal del vuelto | 4 |
| qA | Señal que indica la cantidad de botellas de agua | 4 |
| qP | Señal que indica la cantidad de botellas de Pepsi | 4 |
| qC | Señal que indica la cantidad de botellas de Coca-Cola | 4 |
| producto | Señal que indica qué producto se ha elegido | 4 |
| **Salida** | **Función** | **Cantidad de bits** |
| VGA\_R | Señales que en conjunto indican qué color se visualizará en el monitor | 4 |
| VGA\_G | 4 |
| VGA\_B | 4 |
| VGA\_HS | Señal que indica el sincronismo horizontal | 1 |
| VGA\_VS | Señal que indica el sincronismo vertical | 1 |

**1.3.3. Simulación del módulo VGA**

****

**Figura 6:Simulación del VGA**

**1.4 Máquina de estados principal(Módulo de FSM con Datapath)**

**1.4.1.Comportamiento funcional de la FSM**

Dependiendo de los valores de las señales “valido” y “valor” (donde válido quiere decir que se ha presionado una tecla válida y valor indica qué tecla se ha presionado), se tendrán las salidas que se conectarán con el controlador VGA.

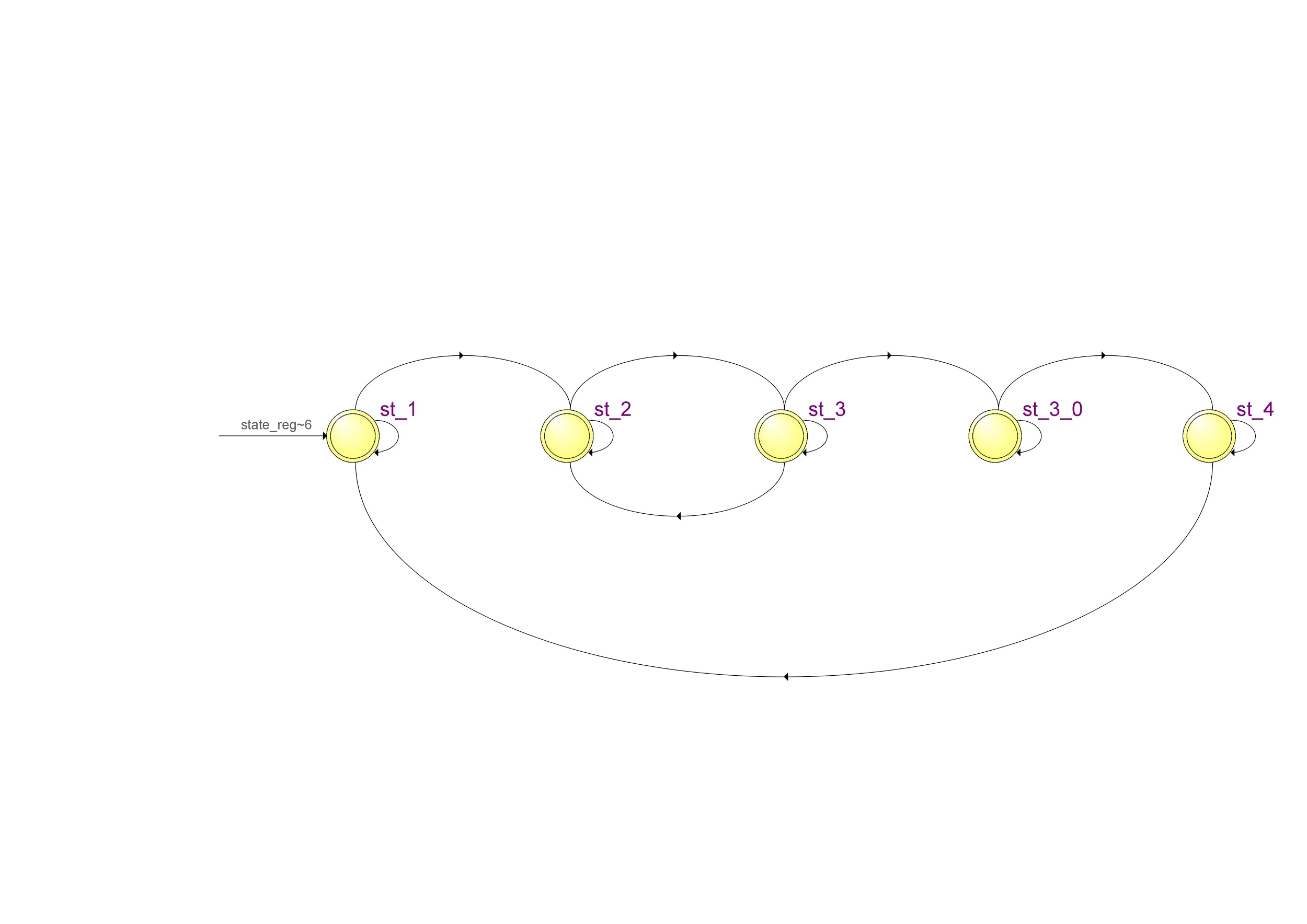
En la figura 7 se muestran las entradas y salidas de este módulo



**Figura 7 : Módulo máquina de estado**

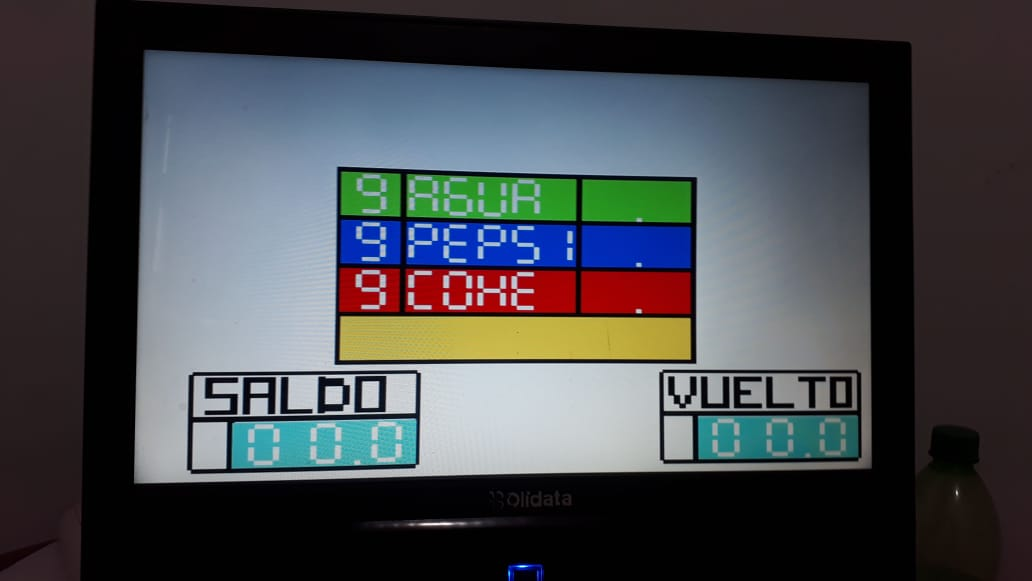
**1.4.2. Diseño del datapath**

Habrá un registro que tenga como dato de entrada a “valor”, la salida de este se comparará varias veces para ver si la tecla presionada es alguna moneda (teclas 1,2,3 o 4) y que moneda, si es un producto (A,B o C) y qué producto es, o si es para devolver el dinero (D). Por otra parte, se contará con un sumador que irá acumulando las monedas que se ingresan mediante 4 multiplexores 2 a 1, para esto se emplean las salidas de los comparadores del registro mencionado antes, es decir, si alguna de las comparaciones del tipo de moneda es verdadero, la salida del comparador respectivo será ‘1’ y el selector del multiplexor se pondrá a ‘1’ dejando pasar el valor del monto para que se sume y se acumule, esto último se consigue con otro registro que almacena la suma. La salida de este registro (el que acumula la suma de las monedas) entrará a otros 3 comparadores para evaluar si es que el monto acumulado alcanza para comprar alguno de los productos. Esta misma salida del registro, a su vez irá a un componente “bintobcd”, este se encarga de transformar el número binario que se tiene en números decimales que se mostrarán en el display. Por otra parte, también habrán 3 contadores decrecientes que se encargan de disminuir la cantidad de productos disponibles.

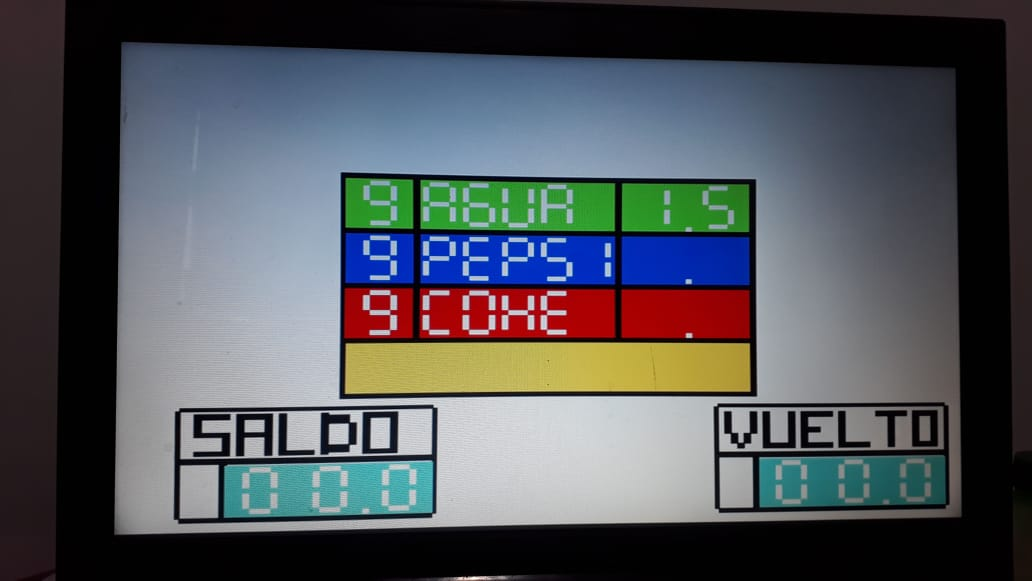


**Figura 8: Diagrama de estados de la FSM**

**CAPÍTULO 2: RESULTADOS EXPERIMENTALES**

****

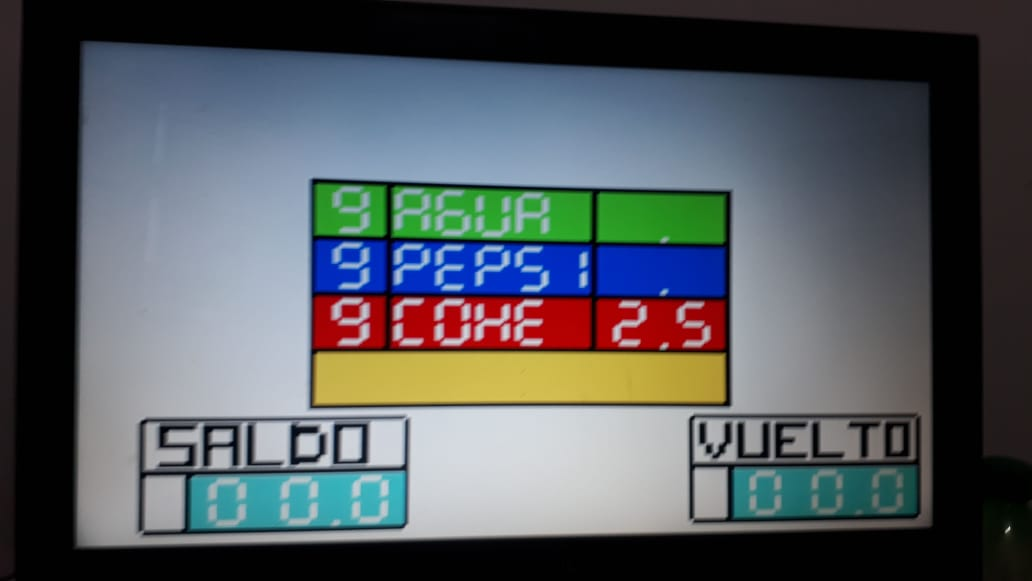
**Figura 9: máquina expendedora de gaseosas en puerto VGA en estado inicial**

****

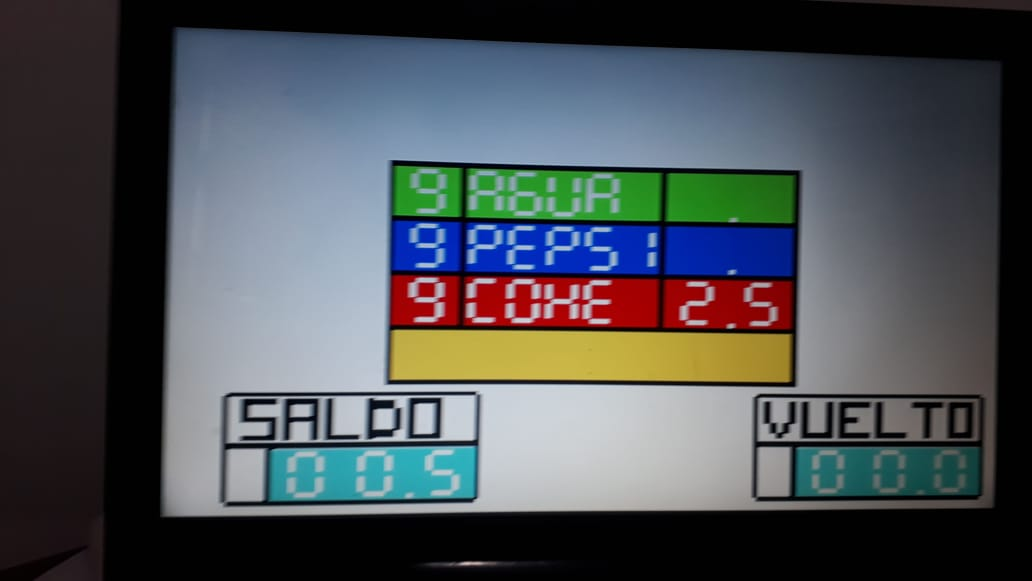
**Figura 10: Selección de agua**

****

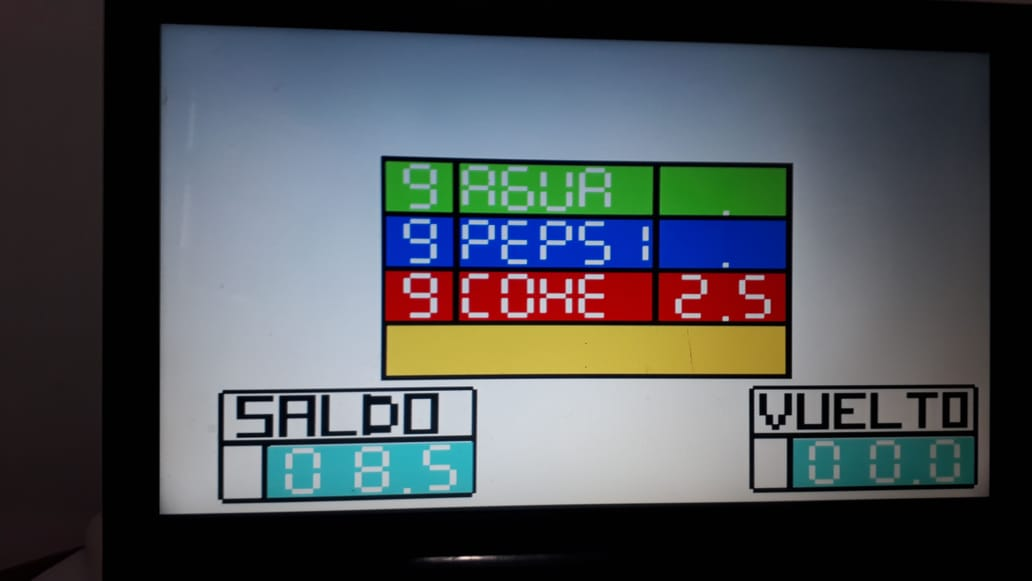
**Figura 11: Selección de Pepsi**

****

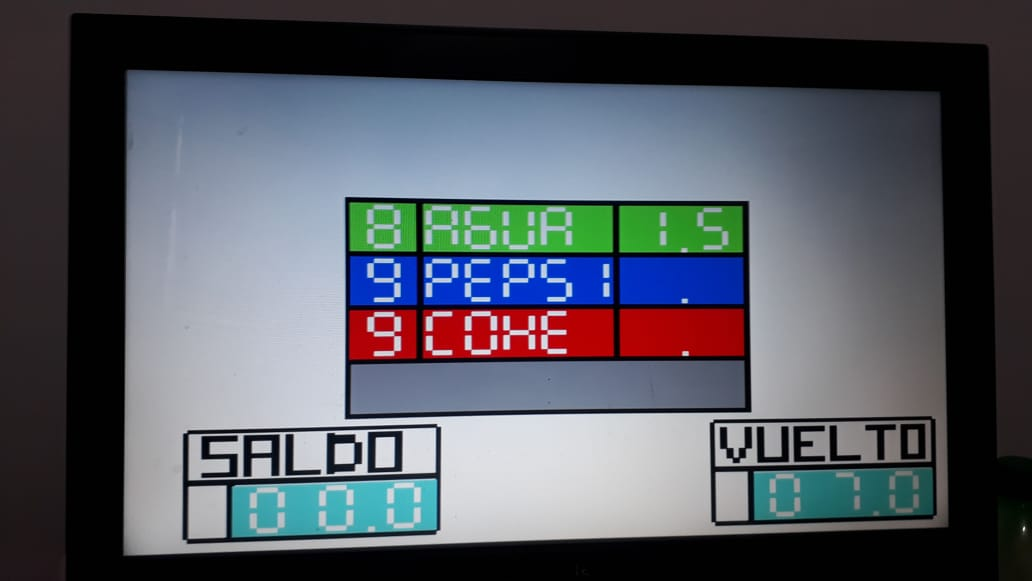
**Figura 12: Selección de Coca-Cola**

****

**Figura 13: Primer ingreso de moneda**

****

**Figura 14: Acumulación del ingreso de monedas hasta 8.5 soles**

****

**Figura 15: Se despacha el producto elegido**

Se presiona el botón de Agua y el rectángulo amarillo cambia a gris, además el saldo se reinicia pues ya se compró el producto y el vuelto correspondiente se muestra.

El funcionamiento es análogo con los demás productos.

**CAPÍTULO 3: CONCLUSIONES Y RECOMENDACIONES**

**3.1. Conclusiones**

* La descripción de los módulos es de funcionamiento independiente, comandados por una unidad de control propia. Esto permite que sean fácilmente utilizados al emplear el protocolo de comunicación Handshake.
* La frecuencia en la que trabaja cada modula afecta directamente al proceso y con ello al resultado.

**3.2. Recomendaciones**

* El datapath del FSM debe operar a 100 hz para que el tiempo de reconocimiento de datos sea instantantaneo, ya que al usar una frecuencia diferente, podríamos ampliar el tiempo de recepción de datos y la máquina podría detectar más veces la tecla presionada.
* Cuando se trabaja con muchos módulos, y se requiere reutilizar partes del código con variaciones en la longitud de las entradas o salidas (por ejemplo, contadores de módulo n, registros de distintas longitudes) se pueden emplear circuitos genéricos.

**ANEXO A - PLANOS**