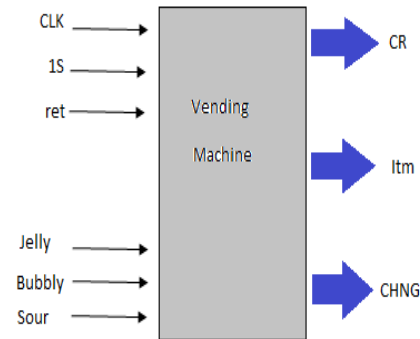


Abstract:

The aim of our project is to design a vending machine using quatus software written in Verilog HDL (Hardware Description Language).

Inputs:

- 1- **CLK**: The system clock that organize the process of entry and buying and it is useful for internal purposes. It generates one clock per second.
- 2- **1S**: A switch that indicates the entry of currency to the system. When it's zero, no currency is entered to the system, while when it's one the currency is increased by one.
- 3- **Bubbly**: A switch that indicates if a buying request came for bubbly or not.
- 4- **Sour**: A switch that indicates if a buying request came for sour or not.
- 5- **Jelly**: A switch that indicates if a buying request came for jelly or not.
- 6- **ret**: A switch that returns the vending machine to its initial state.



Outputs:

- 1- **CR**: Contains the number of entered coins.
- 2- **Itm**: Contains the item bought.
- 3- **CHNG**: Contains the change after a successful buying operation or after a return operation (using ret).

Procedure:

First, the clock rate from the FPGA (Field Programmable Gate Array) we use is 27 MHZ. So, it was reduced to 1 HZ using a frequency divider. Then the new generated clock was used as input(CLK) for the vending machine. Each input was wired to its specific port and each output was wired into a seven segment display decoder in order to display the data. Finally, each seven segment decoder output was wired into specified port on the FPGA.

Algorithm:

The process was divided into 3 states:

- ❖ **State 0**: all outputs are zero
- ❖ **State 1**: It handles the process of data entry, and it remains in the same state until a buy order or return order arrives.
- ❖ **State 2**: It deals with the buy orders. If there is no enough money, then it waits for more money to be entered. But, if there is enough money then it gives the item to the buyer and the change then it returns to the initial state. This process needs 10 seconds to be completed.

Notes:

- 1- My Verilog HDL code was not explicitly based on the previous algorithm. However, I was able to implement it using a different way that was more efficient and more general using if – else statements.
- 2- Multiple selection of item is ignored and the system remains in its current state.
- 3- ret has the highest priority of all inputs.