FINAL PROJECT

Model and control drive-through purchases (similar to the type by KFC and MacDonald)

With reference to the observations made physically and video watched online, it was found that most of the drive-throughs use customized "phone booths" that allow customers to make purchase of food without leaving their cars. In some decades past, a lot of drive throughs have operated traditionally using microphone and speaker. Today, the face of fast food service is improving, there are LCD screens being used as menu-boards, and they come with touch screen for customers to input combinations of the various types of food available, shows full order and total cost of purchase. A few places such as the McDonalds have these digital implementations for ordering by passengers. The major challenge faced by these food service with drive thru is time delay. When the orders become more complicated, time-consuming sets in as passengers wait to get their food ready. In view of this, we would like to propose that a project that is focused on modelling and controlling the drive through digital system with the following information.

Design Specifications and Methodology

Inputs:

A. Select food
B. Pay
C. Print slip

Use three switch inputs to simulate the options and

Output: Two indicators:

x. Printslip: If food is paid, it outputs a chit.
y. Redlight: If food is unpaid or money is not sufficient, it outputs nothing.

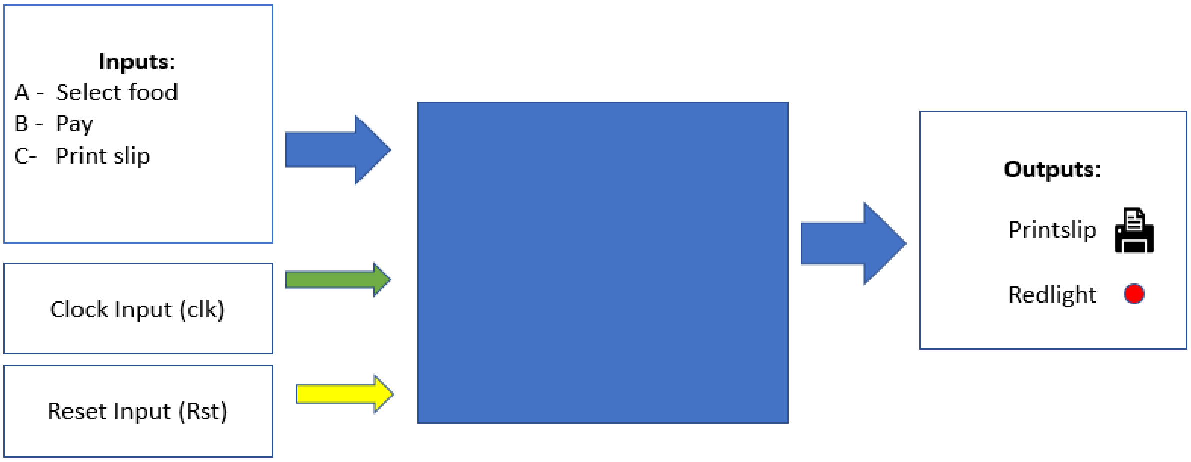


Figure 1 Entity of the drive thru machine representation.

State Assignment

00 – Reset

01 – No food

11 – Serve Food

At Reset state “00”, when all three inputs are high move to Serve Food “11”

At Reset state “00” when all A and C are high, but B is low, move to No Food “01”.

At No Food “01” when all inputs are high, move to Serve Food “11” else remain at ‘01’

At No Food “01” when all inputs are low, move to Reset “00”

At Serve Food “11”, when all inputs are low, move to Reset “00”

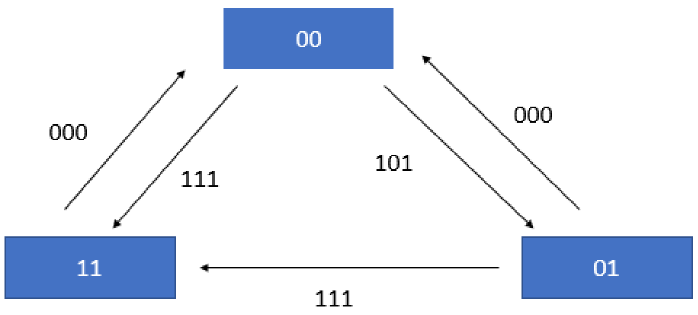


Figure 2. The state diagram of the drive through.

State table.

Present state		Inputs			Next state		Outputs	Flip flops	
A	B	In.1	In.2	In.3	A	B	0	Da	Db
0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0	0	1
0	0	1	1	0	0	0	0	0	0
0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	0	0	0	0
0	1	0	0	1	0	1	0	1	0

0	1	0	1	0	0	1	0	1	0
0	1	0	1	1	0	1	0	0	0
0	1	1	0	0	0	1	0	1	0
0	1	1	0	1	0	1	0	0	1
0	1	1	1	0	0	1	0	0	0
0	1	1	1	1	1	1	1	0	1
1	0	0	0	0	x	x	x	x	X
1	0	0	0	1	x	x	x	X	x
1	0	0	1	0	x	x	X	x	X
1	0	0	1	1	x	x	X	x	X
1	0	1	0	0	x	x	X	X	x
1	0	1	0	1	x	x	X	X	X
1	0	1	1	0	x	x	X	X	X
1	0	1	1	1	x	x	X	X	x
1	1	0	0	0	0	0	0	0	0
1	1	0	0	1	1	1	0	1	0
1	1	0	1	0	1	1	0	1	1
1	1	0	1	1	1	1	0	1	1
1	1	1	0	0	1	1	0	1	1
1	1	1	0	1	1	1	0	1	1
1	1	1	1	0	1	1	0	1	1
1	1	1	1	1	1	1	1	1	1

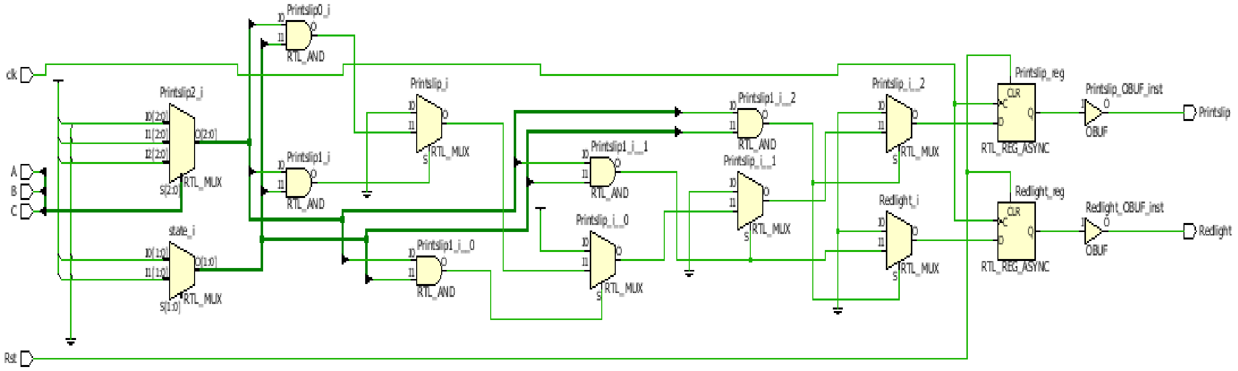
AB\C	0	1
00	0	0
01	0	0
11	1	1
10	1	1

Da = A

AB\C	0	1
00	0	0
01	0	0
11	1	1
10	1	1

Db = A

Behavioral Model of the logic circuit in Vivado



Behavioral Code

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity DrivThru is
    Port ( A : in STD_LOGIC;    ----- select food button
           B : in STD_LOGIC;    ---- select pay button
           C : in STD_LOGIC;    ---- print slip button
           Rst : in STD_LOGIC;  ----- reset active low
           clk : in STD_LOGIC;

           Printslip : out STD_LOGIC;
           Redlight : out STD_LOGIC);
end DrivThru;

architecture Behavioral of DrivThru is

signal state : STD_LOGIC_vector(1 downto 0);


begin

process (A, B, C, Rst, clk)

variable inputs: std_logic_vector(2 downto 0);
begin
inputs := A & B & C;

 if (Rst = '1')then
         printslip <='0';
         redlight <= '0';

     elsif (clk'event and clk = '1')then
     if  (inputs = "000" )and (state = "00") then
     printslip <= '0';
      redlight <= '0';
     elsif (inputs = "101" )and (state = "00") then
         printslip <= '0';
           redlight <= '1';

     elsif  (inputs = "111" )and (state = "00") then
             printslip <= '1';
              redlight <= '0';

     elsif  (inputs = "000" )and (state = "01") then
           printslip <= '0';
            redlight <= '0';

     elsif  (inputs = "111" )and (state = "01") then
           printslip <= '1';
            redlight <= '0';

     elsif  (inputs = "000" )and (state = "11") then
           printslip <= '0';
            redlight <= '0';

     else
           printslip <= '0';
           redlight <= '0';

     end if;
     end if;
 end process;


end Behavioral;
```

Reference articles

http://time.com/money/3086843/mcdonalds-one-minute-drive-thru-guarantee/

http://slideplayer.com/slide/1653929/