# carbonated drinks production line (

# Main idea:

The main idea of a carbonated water production line is to efficiently and consistently produce carbonated water, which is water that has been infused with carbon dioxide (CO2) gas to create bubbles and a fizzy texture. So here we program a chip to control that production line, Hence the production line typically involves several key steps:

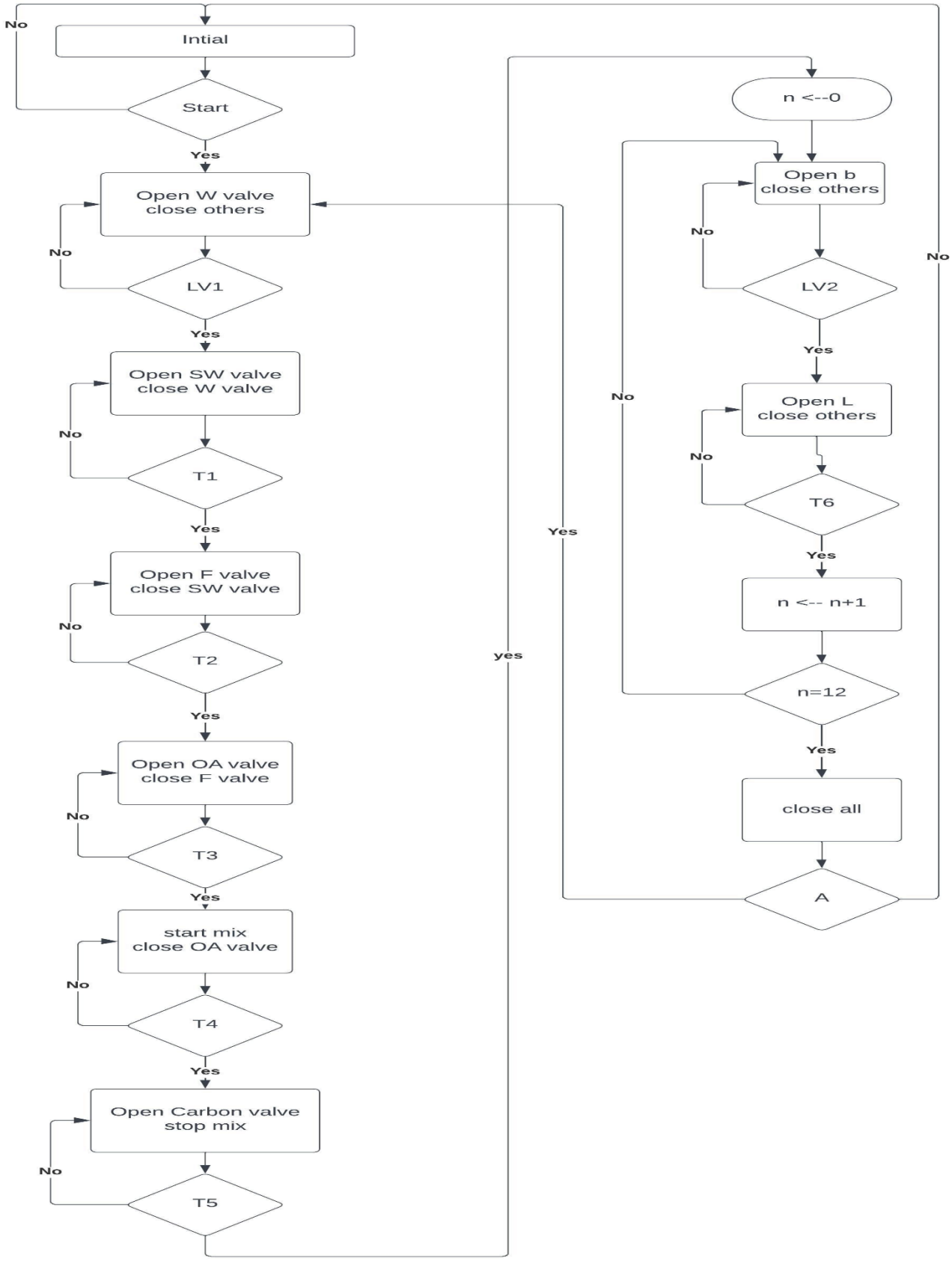
# YouTube video:

https://youtu.be/JT-cQm939XE?si=G9qaUJmmZIr7Rpv0

# Sequence and steps:

1- Start: The starting point of the carbonated drinks production process, where all valves are closed.
2- Adding ingredients: first we add water until it reaches a certain level, then we add sweeteners for certain time Ti1, then flavor for Ti2 and last, we add other additives for Ti3.
3- Ingredients Mixing: Mixing and blending the various ingredients, including water, sweeteners, flavors, and other additives for Ti4.
4- Carbonation: Introducing carbon dioxide into the mixture for certain time Ti5 to create the characteristic fizziness.
5- Bottling: Transferring the carbonated beverage into bottles then cover the bottles.
6- Labeling: Applying labels with product information and branding to the bottles.
7- Packaging: Packaging the bottles into cases or other packaging formats suitable for distribution, by iterating on steps 5 and 6 until the bottles number is equal to 12.
8- Finished Product: The result of the production process, ready for distribution and consumption.
9- Work for ever: This action (production process) will be repeated if the automatic is set.
10- End: The end of the carbonated drinks production process.

# ASM chart:


No
Intial
Start
Yes
Open W valve
close others
No
LV1
Yes
Open SW valve
close W valve
No
T1
Yes
Open F valve
close SW valve
No
T2
Yes
Open OA valve
close F valve
No
T3
Yes
start mix
close OA valve
No
T4
Yes
Open Carbon valve
stop mix
No
T5
yes
n <--0
Open b
close others
No
LV2
Yes
Open L
close others
No
T6
Yes
n <-- n+1
n=12
Yes
close all
A
No
Yes
No
No

# VHDL code:

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

entity carbonated_drinks is
  Port (clk, rst, s : in STD_LOGIC;
        Ti1, Ti2, Ti3, Ti4, Ti5, Ti6, LV1, LV2, A : in STD_LOGIC;
        W, SW, F, OA, M, C, B, L : out STD_LOGIC);
end carbonated_drinks;

architecture Behavioral of carbonated_drinks is
    TYPE state is (T0, T1, T2, T3, T4, T5, T6, T7, T8, T9, T10);
    SIGNAL pr, nxt : state;
begin
    seq: process(clk)
    begin
        if(rising_edge(clk)) then
            if(rst = '1') then
                pr <= T0;
            else
                pr <= nxt;
            end if;
        end if;
    end process seq;
```

```
comb: process(pr, s, Ti1, Ti2, Ti3, Ti4, Ti5, Ti6, LV1, LV2, A)
   variable n: integer;
begin
   case pr is
      when T0=> W <= '0'; SW <= '0'; F <= '0'; OA <= '0'; M <= '0'; C <= '0'; B <= '0'; L <= '0';
         if(s = '1') then nxt <= T1;
         else nxt <= T0;
         end if;
      when T1 => W <= '1'; SW <= '0'; F <= '0'; OA <= '0'; M <= '0'; C <= '0'; B <= '0'; L <= '0';
         if(LV1 = '1') then nxt <= T2;
         else nxt <= T1;
         end if;
      when T2 => W <= '0'; SW <= '1'; F <= '0'; OA <= '0'; M <= '0'; C <= '0'; B <= '0'; L <= '0';
         if(Ti1 = '1') then nxt <= T3;
         else nxt <= T2;
         end if;
      when T3 => W <= '0'; SW <= '0'; F <= '1'; OA <= '0'; M <= '0'; C <= '0'; B <= '0'; L <= '0';
         if(Ti2 = '1') then nxt <= T4;
         else nxt <= T3;
         end if;
      when T4 => W <= '0'; SW <= '0'; F <= '0'; OA <= '1'; M <= '0'; C <= '0'; B <= '0'; L <= '0';
         if(Ti3 = '1') then nxt <= T5;
         else nxt <= T4;
         end if;
      when T5 => W <= '0'; SW <= '0'; F <= '0'; OA <= '0'; M <= '1'; C <= '0'; B <= '0'; L <= '0';
         if(Ti4 = '1') then nxt <= T6;
         else nxt <= T5;
```

```
            end if;
         when T6 => W <= '0'; SW <= '0'; F <= '0'; OA <= '0'; M <= '0'; C <= '1'; B <= '0'; L <= '0';
            if(Ti5 = '1') then nxt <= T7; n:=0;
            else nxt <= T6;
            end if;
         when T7 => W <= '0'; SW <= '0'; F <= '0'; OA <= '0'; M <= '0'; C <= '0'; B <= '1'; L <= '0';
            if(LV2 = '1') then nxt <= T8;
            else nxt <= T7;
            end if;
         when T8 => W <= '0'; SW <= '0'; F <= '0'; OA <= '0'; M <= '0'; C <= '0'; B <= '0'; L <= '1';
            if(Ti6 = '1') then nxt <= T9;
            else nxt <= T8;
            end if;
        when T9 => n:=n+1; W <= '0'; SW <= '0'; F <= '0'; OA <= '0'; M <= '0'; C <= '0'; B <= '0'; L <=
'0';
            if(n = 12) then nxt <= T10;
            else nxt <= T7;
            end if;
         when T10 => W <= '0'; SW <= '0'; F <= '0'; OA <= '0'; M <= '0'; C <= '0'; B <= '0'; L <= '0';
            if(A = '1') then nxt <= T1;
            else nxt <= T0;
            end if;
      end case;
   end process comb;
end Behavioral;
```