



CENTRO UNIVERSITÁRIO SALESIANO de SÃO PAULO
UNIDADE de ENSINO de CAMPINAS

“DISPENSADOR DE ALIMENTOS”

Luis Fernando Martins – RA: 04031123 – Turma: GEEN7A
Ricardo Biazotto Tiene – RA: 05030729 – Turma : GEEN7A
Renato Abrantes – RA: 02030357 – Turma: GEEN7A

Prof.: Wlamir de Almeida Passos

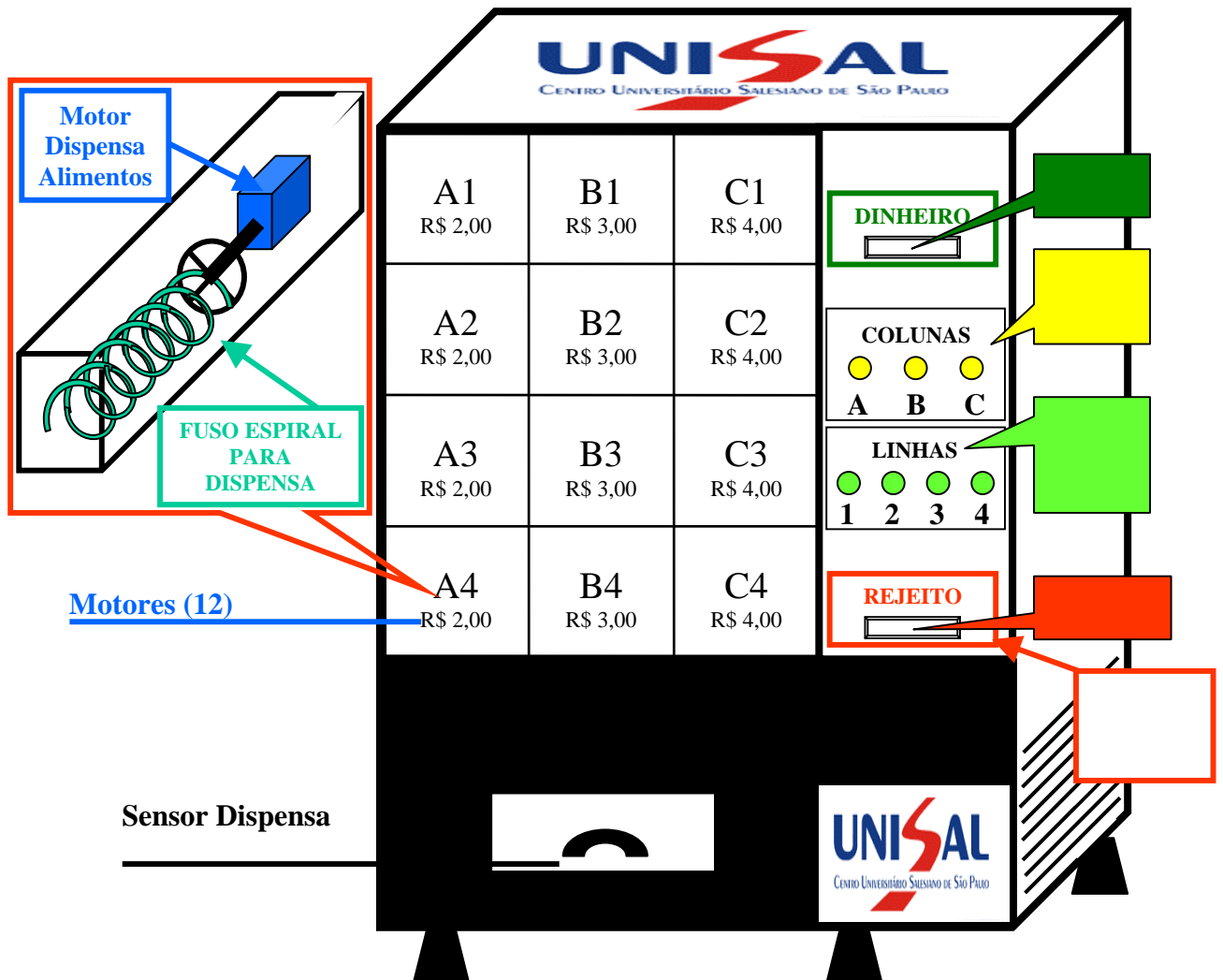
Sistemas Digitais

1. Objetivo

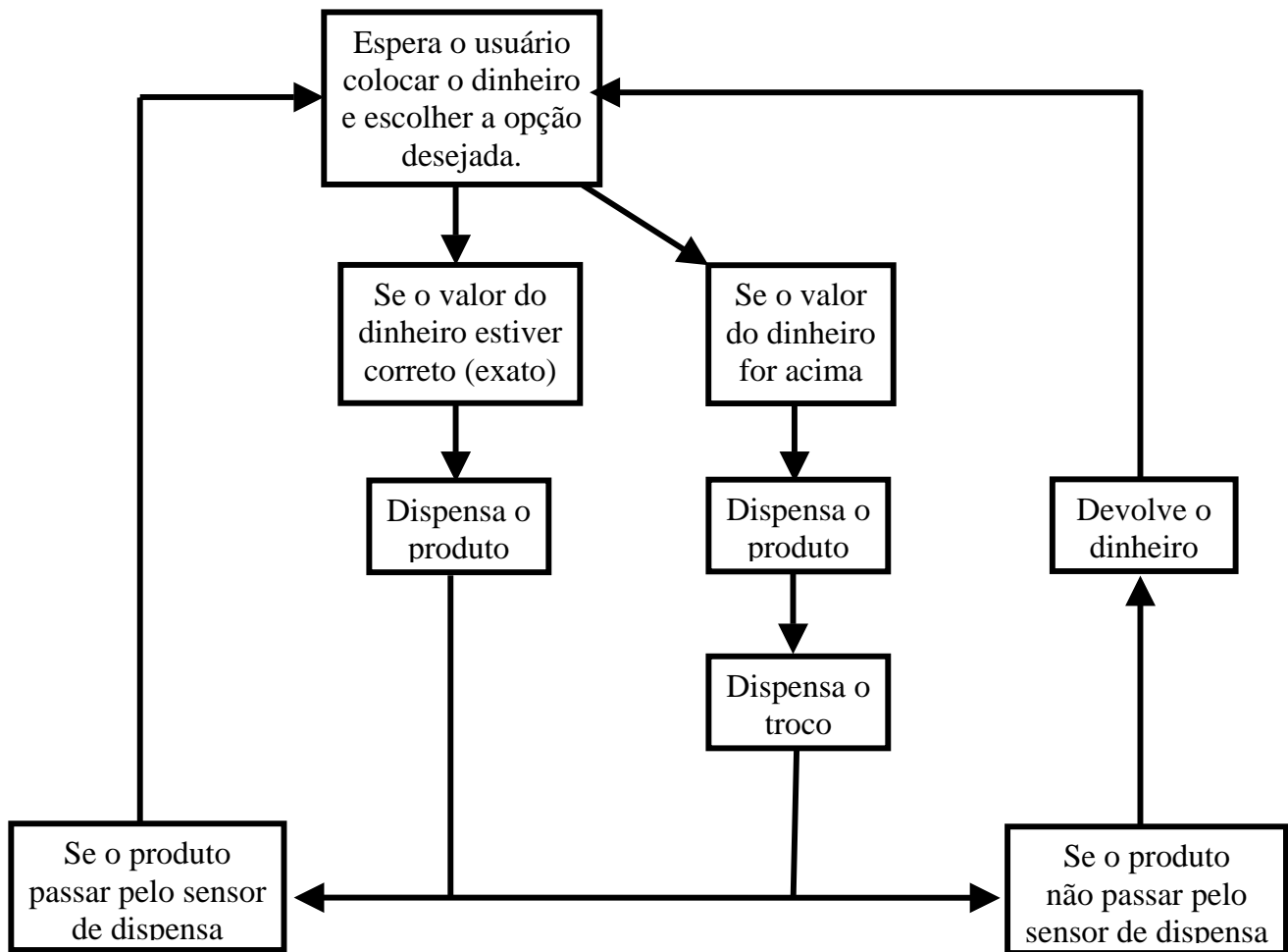
Escolhemos para este projeto uma MÁQUINA DISPENSADORA DE ALIMENTOS, que nos dias de hoje é bastante utilizada em vários estabelecimentos comerciais, como: shoppings, mercados, parques, escolas, etc. Portanto, o nosso objetivo nesse projeto, será descrever o funcionamento desta máquina. Apresentaremos a seguir o seu diagrama de funcionamento e também sua linguagem lógica de programação através do VHDL (feito através do software ALTERA).

2. Especificação

2.1 Desenho do projeto



2.2 Fluxograma sistêmico



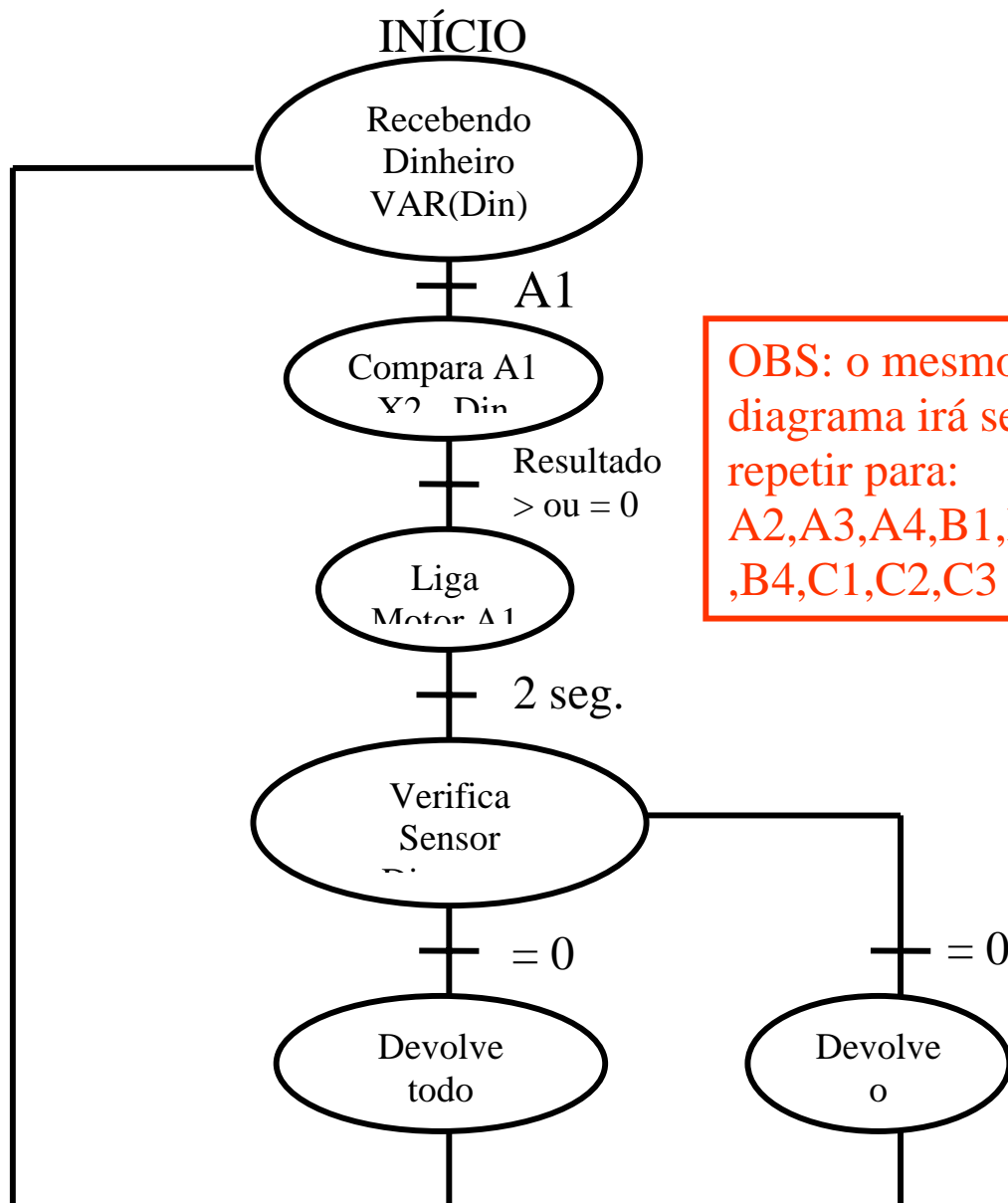
3. Projeto Hardware

3.1 Lista de Sensores e Atuadores

Nome	Tipo	Função
Sensor_Dinheiro	Entrada	Calcula (conta) a quantidade de dinheiro inserida na máquina.
Sensor_Rejeito		Verifica se o troco foi devolvido.
Sensor_Dispensa		Verifica se o produto foi entregue ao cliente.
Modo	Entrada de Barramento	Modo de operação da máquina (?)
Modo_Motor	Saída de Barramento	Tipo de operação do motor
Motor_Dispensa_Alimentos (1~12)	Saída	Rotaciona o fuso para a dispensa do alimento.
Motor_Rejeito_Dinheiro		Devolução do troco ou do dinheiro caso o produto (alimento) não seja entregue ao cliente.

3.2 Diagramas das Maquinas de estado

Diagrama de Estados A1:



OBS: o mesmo diagrama irá se repetir para: A2,A3,A4,B1,B2,B3 ,B4,C1,C2,C3 e C4

4. Projeto VHDL

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_unsigned.ALL;
ENTITY disp_alimen IS
    PORT
    (
        clk          : IN STD_LOGIC;
        S1           : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
        S2,S3       : IN STD_LOGIC;
        Bot1234: IN STD_LOGIC_VECTOR(2 DOWNTO 0);
        BotABC: IN STD_LOGIC_VECTOR(1 DOWNTO 0);
        M1,M2,M3,M4,M5,M6,M7,M8,M9,M10,M11,M12,M13 : OUT STD_LOGIC
    );
END disp_alimen;

ARCHITECTURE projeto OF disp_alimen IS
    TYPE estados_maq IS (start,recebe_din,comp_A1,comp_A2,comp_A3,comp_A4,comp_B1,comp_B2,comp_B3,
    comp_B4,comp_C1,comp_C2,comp_C3,comp_C4,M1_Disp_Prod,M2_Disp_Prod,M3_Disp_Prod,M4_Disp_Prod,
    M5_Disp_Prod,M6_Disp_Prod,M7_Disp_Prod,M8_Disp_Prod,M9_Disp_Prod,M10_Disp_Prod,M11_Disp_Prod,
    M12_Disp_Prod,Verif_S3,Devolve_Todo_Din,Devolve_Troco);
    SIGNAL estados : estados_maq;
    SIGNAL Produto: STD_LOGIC_VECTOR(3 DOWNTO 0);
    SIGNAL Liga_Timer: STD_LOGIC;
    SIGNAL Fim_Timer: STD_LOGIC;
    SIGNAL P1SEG : std_logic;

BEGIN

    Process(clk)
    variable din: STD_LOGIC_VECTOR(3 DOWNTO 0);
    variable troco: STD_LOGIC_VECTOR(3 DOWNTO 0);

    BEGIN

        IF clk'event and clk='1' then

            case estados is

                WHEN start =>
                    troco:=x"0";
                    DIN:= x"0";
                    IF S1="01" THEN
                        DIN := DIN+x"1";
                    ELSIF S1="10" THEN
                        DIN := DIN+x"2";
                    ELSIF S1="11" THEN
                        DIN := DIN+x"5";
                    END IF;

                    IF din /= x"0" then
                        estados <=recebe_din;
                    END IF;

                WHEN recebe_din =>

                    IF S1="01" THEN
                        DIN := DIN+x"1";
                    ELSIF S1="10" THEN
```

```
DIN := DIN+x"2";
ELSIF S1="11" THEN
DIN := DIN+x"5";
END IF;
```

```
IF BotABC = "01" and Bot1234 = "001" then estados <= comp_A1;
ELSIF BotABC = "01" and Bot1234 = "010" then estados <= comp_A2;
ELSIF BotABC = "01" and Bot1234 = "011" then estados <= comp_A3;
ELSIF BotABC = "01" and Bot1234 = "100" then estados <= comp_A4;
ELSIF BotABC = "10" and Bot1234 = "001" then estados <= comp_B1;
ELSIF BotABC = "10" and Bot1234 = "010" then estados <= comp_B2;
ELSIF BotABC = "10" and Bot1234 = "011" then estados <= comp_B3;
ELSIF BotABC = "10" and Bot1234 = "100" then estados <= comp_B4;
ELSIF BotABC = "11" and Bot1234 = "001" then estados <= comp_C1;
ELSIF BotABC = "11" and Bot1234 = "010" then estados <= comp_C2;
ELSIF BotABC = "11" and Bot1234 = "011" then estados <= comp_C3;
ELSIF BotABC = "11" and Bot1234 = "100" then estados <= comp_C4;
END IF;
```

```
WHEN comp_A1 =>
troco := din - "0010";
IF troco >= x"00" then estados <= M1_Disp_Prod;
ELSE estados <= Devolve_Todo_Din;
END IF;
```

```
WHEN comp_A2 =>
troco := din - "0010";
IF troco >= x"00" then estados <= M2_Disp_Prod;
ELSE estados <= Devolve_Todo_Din;
END IF;
```

```
WHEN comp_A3 =>
troco := din - "0010";
IF troco >= x"00" then estados <= M3_Disp_Prod;
ELSE estados <= Devolve_Todo_Din;
END IF;
```

```
WHEN comp_A4 =>
troco := din - "0010";
IF troco >= x"00" then estados <= M4_Disp_Prod;
ELSE estados <= Devolve_Todo_Din;
END IF;
```

```
WHEN comp_B1 =>
troco := din - "0011";
IF troco >= x"00" then estados <= M5_Disp_Prod;
ELSE estados <= Devolve_Todo_Din;
END IF;
```

```
WHEN comp_B2 =>
troco := din - "0011";
IF troco >= x"00" then estados <= M6_Disp_Prod;
ELSE estados <= Devolve_Todo_Din;
END IF;
```

```
WHEN comp_B3 =>
troco := din - "0011";
IF troco >= x"00" then estados <= M7_Disp_Prod;
ELSE estados <= Devolve_Todo_Din;
END IF;
```

```
WHEN comp_B4 =>
troco := din - "0011";
```

```

        IF troco >= x"00" then estados <= M8_Disp_Prod;
        ELSE estados <= Devolve_Todo_Din;
        END IF;

WHEN comp_C1 =>
    troco := din - "1000";
    IF troco >= x"00" then estados <= M9_Disp_Prod;
    ELSE estados <= Devolve_Todo_Din;
    END IF;

WHEN comp_C2 =>
    troco := din - "1000";
    IF troco >= x"00" then estados <= M10_Disp_Prod;
    ELSE estados <= Devolve_Todo_Din;
    END IF;

WHEN comp_C3 =>
    troco := din - "1000";
    IF troco >= x"00" then estados <= M11_Disp_Prod;
    ELSE estados <= Devolve_Todo_Din;
    END IF;

WHEN comp_C4 =>
    troco := din - "1000";
    IF troco >= x"00" then estados <= M12_Disp_Prod;
    ELSE estados <= Devolve_Todo_Din;
    END IF;

WHEN M1_Disp_Prod =>
    Liga_Timer <= '1';
    IF Fim_Timer = '1' then estados <= Verif_S3;
    Liga_Timer <= '0';
    END IF;

WHEN M2_Disp_Prod =>
    Liga_Timer <= '1';
    IF Fim_Timer = '1' then estados <= Verif_S3;
    Liga_Timer <= '0';
    END IF;

WHEN M3_Disp_Prod =>
    Liga_Timer <= '1';
    IF Fim_Timer = '1' then estados <= Verif_S3;
    Liga_Timer <= '0';
    END IF;

WHEN M4_Disp_Prod =>
    Liga_Timer <= '1';
    IF Fim_Timer = '1' then estados <= Verif_S3;
    Liga_Timer <= '0';
    END IF;

WHEN M5_Disp_Prod =>
    Liga_Timer <= '1';
    IF Fim_Timer = '1' then estados <= Verif_S3;
    Liga_Timer <= '0';
    END IF;

WHEN M6_Disp_Prod =>
    Liga_Timer <= '1';
    IF Fim_Timer = '1' then estados <= Verif_S3;

```

```

        Liga_Timer <= '0';
        END IF;

    WHEN M7_Disp_Prod =>
        Liga_Timer <= '1';
        IF Fim_Timer = '1' then estados <= Verif_S3;
        Liga_Timer <= '0';
        END IF;

    WHEN M8_Disp_Prod =>
        Liga_Timer <= '1';
        IF Fim_Timer = '1' then estados <= Verif_S3;
        Liga_Timer <= '0';
        END IF;

    WHEN M9_Disp_Prod =>
        Liga_Timer <= '1';
        IF Fim_Timer = '1' then estados <= Verif_S3;
        Liga_Timer <= '0';
        END IF;

    WHEN M10_Disp_Prod =>
        Liga_Timer <= '1';
        IF Fim_Timer = '1' then estados <= Verif_S3;
        Liga_Timer <= '0';
        END IF;

    WHEN M11_Disp_Prod =>
        Liga_Timer <= '1';
        IF Fim_Timer = '1' then estados <= Verif_S3;
        Liga_Timer <= '0';
        END IF;

    WHEN M12_Disp_Prod =>
        Liga_Timer <= '1';
        IF Fim_Timer = '1' then estados <= Verif_S3;
        Liga_Timer <= '0';
        END IF;

    WHEN Verif_S3 =>
        Liga_Timer <= '1';
        IF Fim_Timer = '1' then
            IF S3 = '1' then estados <= Devolve_Troco;
            ELSE estados <= Devolve_Todo_Din; Liga_Timer <= '0';
            END IF;
        END IF;

    WHEN Devolve_Troco =>
        Liga_Timer <= '1';
        Troco := Troco - "0001";
        IF troco = 0 then
            din := x"0";
            estados <= start;
        end if;

    WHEN Devolve_Todo_Din =>
        DIN := DIN - x"1";
        Liga_Timer <= '1';
        IF DIN = 0 then estados <= start;
        END IF;

    END CASE;
END IF;

```

```
END PROCESS;
```

```
PROCESS (estados,s1,fim_timer)
```

```
BEGIN
```

```
M1 <= '0';
```

```
M2 <= '0';
```

```
M3 <= '0';
```

```
M4 <= '0';
```

```
M5 <= '0';
```

```
M6 <= '0';
```

```
M7 <= '0';
```

```
M8 <= '0';
```

```
M9 <= '0';
```

```
M10 <= '0';
```

```
M11 <= '0';
```

```
M12 <= '0';
```

```
M13 <= '0';
```

```
CASE estados IS
```

```
    WHEN M1_Disp_Prod =>
```

```
        M1 <= '1';
```

```
    WHEN M2_Disp_Prod =>
```

```
        M2 <= '1';
```

```
    WHEN M3_Disp_Prod =>
```

```
        M3 <= '1';
```

```
    WHEN M4_Disp_Prod =>
```

```
        M4 <= '1';
```

```
    WHEN M5_Disp_Prod =>
```

```
        M5 <= '1';
```

```
    WHEN M6_Disp_Prod =>
```

```
        M6 <= '1';
```

```
    WHEN M7_Disp_Prod =>
```

```
        M7 <= '1';
```

```
    WHEN M8_Disp_Prod =>
```

```
        M8 <= '1';
```

```
    WHEN M9_Disp_Prod =>
```

```
        M9 <= '1';
```

```
    WHEN M10_Disp_Prod =>
```

```
        M10 <= '1';
```

```
    WHEN M11_Disp_Prod =>
```

```
        M11 <= '1';
```

```
    WHEN M12_Disp_Prod =>
```

```
        M12 <= '1';
```

```
    WHEN Devolve_Todo_Din =>
```

```
        M13 <= '1';
```

```
--    IF Fim_Timer = '1' then
```

```
--        M13 <= '0';
```

```
--    END IF;
```

```

        WHEN Devolve_Troco =>
            M13 <= '1';
--        IF Fim_Timer = '1' then
--            M13 <= '0';
--        END IF;

        when others =>
            null;

END CASE;
END PROCESS;

PROCESS (CLK,Liga_Timer)
VARIABLE CNT_VAL : STD_LOGIC_VECTOR (3 DOWNTO 0);
VARIABLE FIM_VAL : STD_LOGIC;
BEGIN
IF Liga_Timer = '0' then
CNT_VAL := x"0";
FIM_VAL := '0';
ELSIF (CLK'EVENT AND CLK = '1') THEN
    IF CNT_VAL < x"5" THEN
        CNT_VAL := CNT_VAL + 1;
        FIM_VAL := '0';
    ELSE
        CNT_VAL := CNT_VAL;
        FIM_VAL := '1';
    END IF;
END IF;
FIM_TIMER <= FIM_VAL;
END PROCESS;

END projeto;

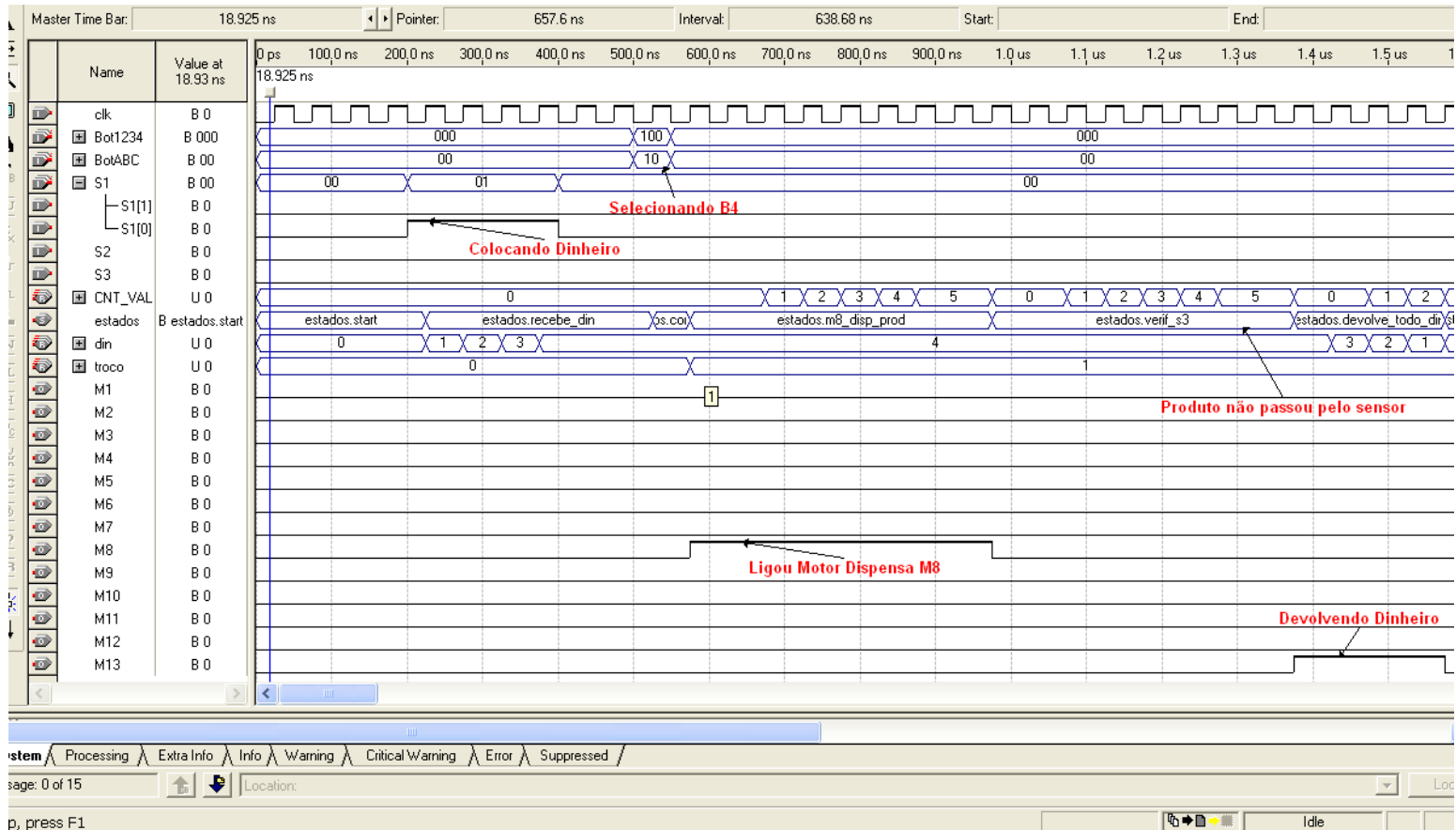
```

4.1 Colocar resposta da simulação de cada bloco do projeto e a simulação final.

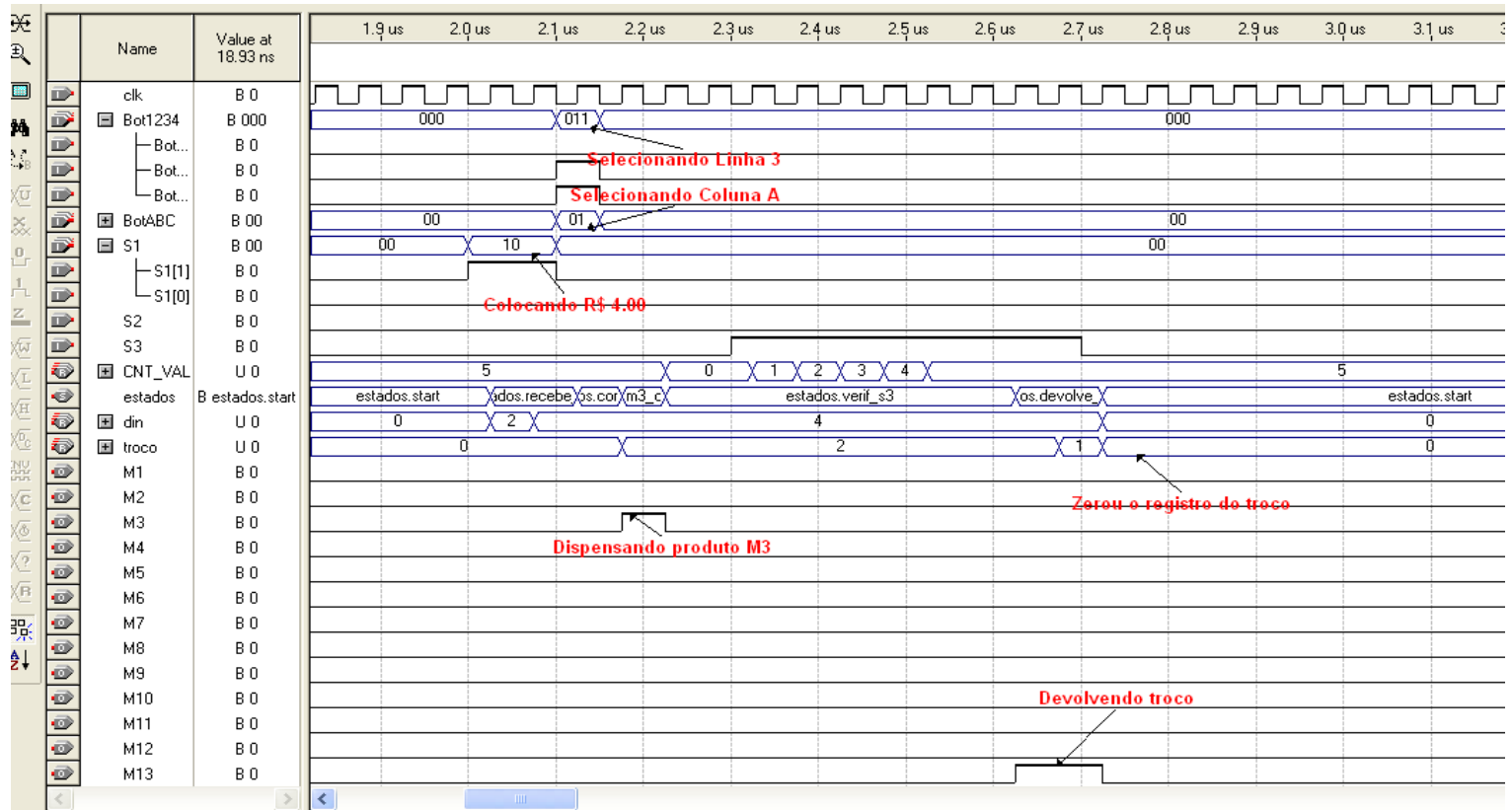
Desenhos do simulador.

5. Resultados obtidos

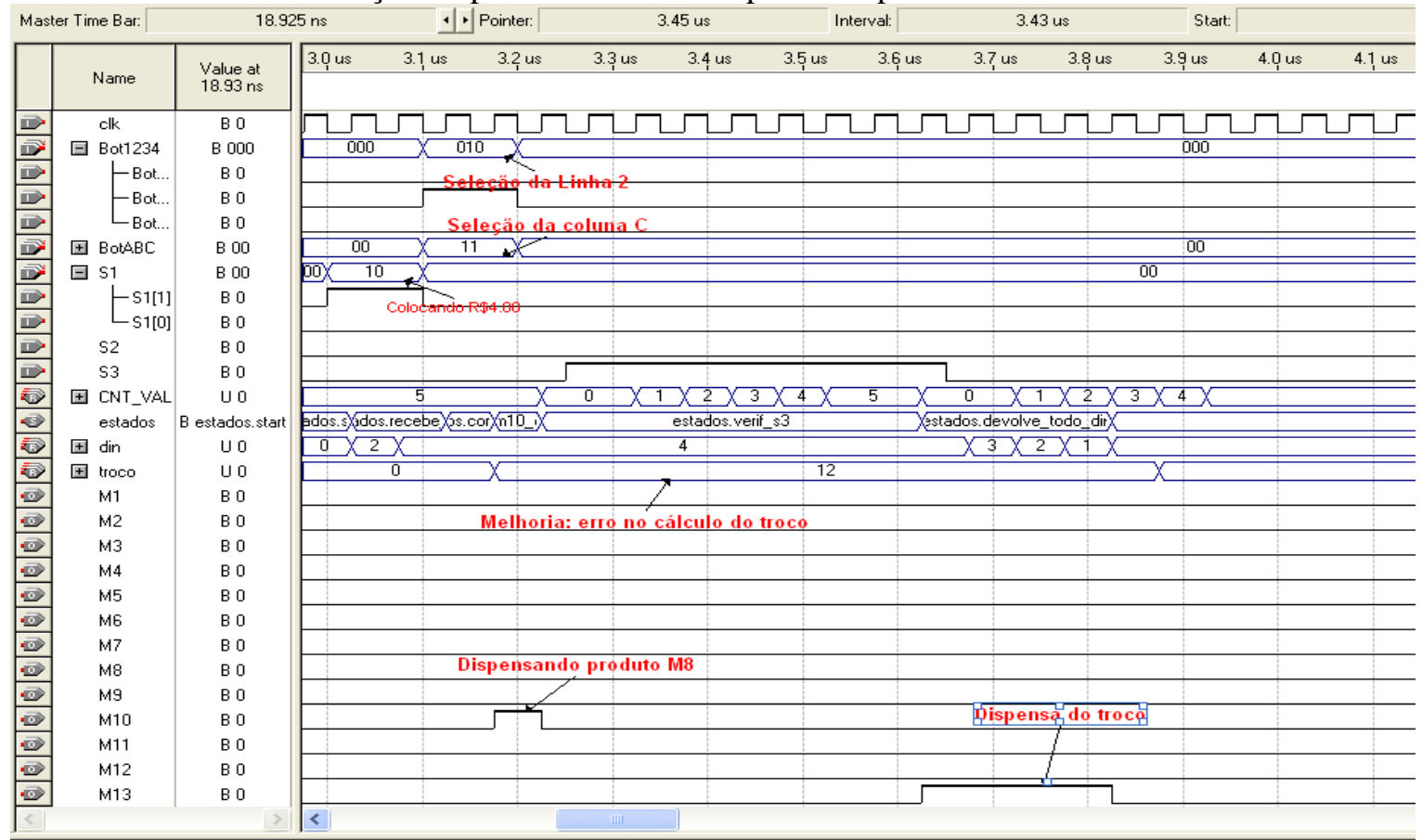
Forma de Onda:



Seleção do produto B4 com erro na dispensa

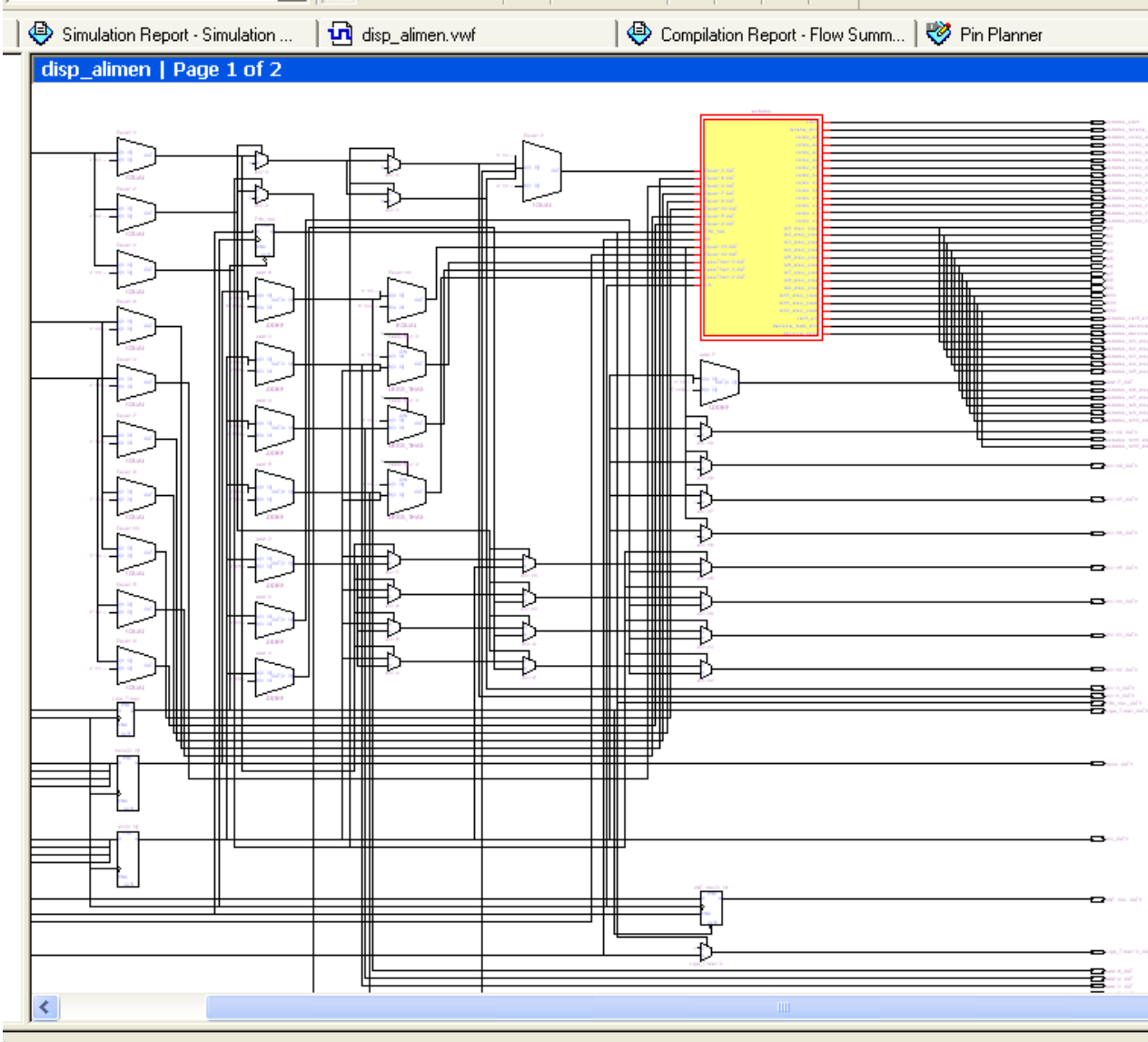


Seleção do produto A3 com dispensa do produto e do troco de R\$2.00



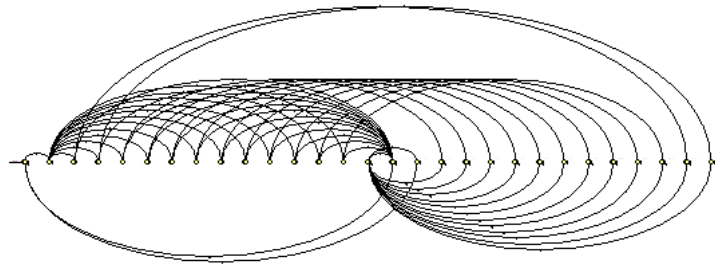
Seleção do produto C2 com valor exato (R\$ 4.00)

RTL View



Máquina de Estado

State Machine: |disp_alimenleestados



	Source State	Destination State	Condition
1	start	start	(Equal~0:OUT)
2	start	recebe_din	(!Equal~0:OUT)
3	recebe_din	comp_a4	(!Equal~6:OUT),(!Equal~7:OUT),(!Equal~8:OUT),(Equal~10:OUT),(Equal~4:OUT)
4	recebe_din	comp_b1	(!Equal~4:OUT),(!Equal~6:OUT),(!Equal~5:OUT)
5	recebe_din	comp_b2	(!Equal~6:OUT),(!Equal~4:OUT),(!Equal~7:OUT),(!Equal~5:OUT)
6	recebe_din	comp_b3	(!Equal~6:OUT),(!Equal~7:OUT),(!Equal~4:OUT),(Equal~8:OUT),(Equal~5:OUT)
7	recebe_din	comp_b4	(!Equal~6:OUT),(!Equal~7:OUT),(!Equal~8:OUT),(!Equal~4:OUT),(Equal~10:OUT),(Equal~5:OUT)
8	recebe_din	comp_c1	(!Equal~4:OUT),(!Equal~5:OUT),(Equal~9:OUT),(Equal~6:OUT)
9	recebe_din	comp_c2	(!Equal~6:OUT),(!Equal~4:OUT),(!Equal~5:OUT),(Equal~9:OUT),(Equal~7:OUT)
10	recebe_din	comp_c3	(!Equal~6:OUT),(!Equal~7:OUT),(!Equal~4:OUT),(!Equal~5:OUT),(Equal~9:OUT),(Equal~8:OUT)
11	recebe_din	comp_c4	(!Equal~6:OUT),(!Equal~7:OUT),(!Equal~8:OUT),(!Equal~4:OUT),(!Equal~5:OUT),(Equal~9:OUT),(Equal~10:OUT)
12	recebe_din	recebe_din	(!Equal~6:OUT),(!Equal~7:OUT),(!Equal~8:OUT),(!Equal~10:OUT) + (!Equal~4:OUT),(!Equal~5:OUT),(!Equal~9:OUT)
13	recebe_din	comp_a1	(Equal~6:OUT),(Equal~4:OUT)
14	recebe_din	comp_a2	(!Equal~6:OUT),(!Equal~7:OUT),(Equal~4:OUT)
15	recebe_din	comp_a3	(!Equal~6:OUT),(!Equal~7:OUT),(!Equal~8:OUT),(Equal~4:OUT)
16	comp_a1	m1_disp_prod	(LessThan~0:OUT)
17	comp_a1	devolve_todo_din	(!LessThan~0:OUT)

Transitions / Encoding /

Pinagem

Digital 2/VHDL/actual/disp_alimen - disp_alimen - [Pin Planner]

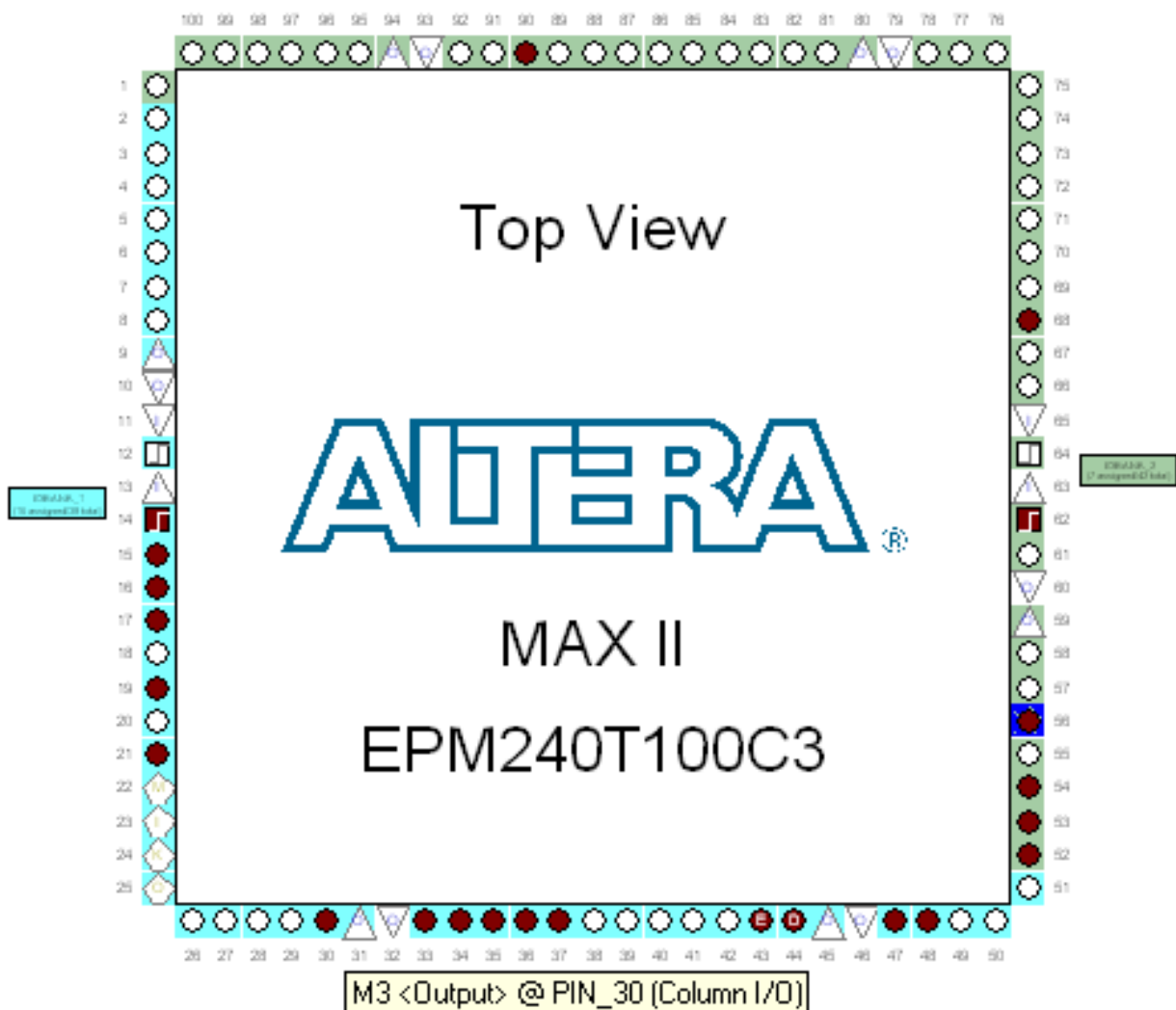
Help



Simulation Waveforms

disp_alimen.vwf

Compilation Report - Flow Summary



Named: *		All Pins						Filter: F
	Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Group
1	Bot1234[0]	Input	PIN_56	2		LVTTTL (default)		Bot1234
2	Bot1234[1]	Input	PIN_19	1		LVTTTL (default)		Bot1234
3	Bot1234[2]	Input	PIN_53	2		LVTTTL (default)		Bot1234
4	BotABC[0]	Input	PIN_21	1		LVTTTL (default)		BotABC
5	BotABC[1]	Input	PIN_54	2		LVTTTL (default)		BotABC
6	clk	Input	PIN_14	1		LVTTTL (default)		
7	M1	Output	PIN_34	1		LVTTTL (default)		
8	M2	Output	PIN_35	1		LVTTTL (default)		
9	M3	Output	PIN_30	1		LVTTTL (default)		
10	M4	Output	PIN_33	1		LVTTTL (default)		
11	M5	Output	PIN_48	1		LVTTTL (default)		
12	M6	Output	PIN_47	1		LVTTTL (default)		
13	M7	Output	PIN_43	1		LVTTTL (default)		
14	M8	Output	PIN_44	1		LVTTTL (default)		
15	M9	Output	PIN_36	1		LVTTTL (default)		
16	M10	Output	PIN_16	1		LVTTTL (default)		
17	M11	Output	PIN_15	1		LVTTTL (default)		
18	M12	Output	PIN_17	1		LVTTTL (default)		
19	M13	Output	PIN_62	2		LVTTTL (default)		
20	S1[0]	Input	PIN_68	2		LVTTTL (default)		S1
21	S1[1]	Input	PIN_90	2		LVTTTL (default)		S1
22	S2	Input	PIN_52	2		LVTTTL (default)		
23	S3	Input	PIN_37	1		LVTTTL (default)		
24	<<new node>>							

5.1 Conclusão

Conclusão do grupo indicando dificuldades e vantagens do projeto.

5.2 Referências

Material e sites pesquisados.

5.3 Melhorias propostas

Conforme indicado no gráfico de formas de onda (simulação)