

**Residência Tecnológica em FPGA - EmbarcaTech / Turma Beta**  
**Residente: Hirislayne Batista Ramos dos Santos**  
**Matrícula: 20251RSE.MTC0041**

## **Unidade 3 - Capítulo 1: Linguagem VHDL - Descrição de Hardware com VHDL**

### **1. Objetivo**

Desenvolver a capacidade de projetar, modelar, simular e implementar circuitos digitais por meio da linguagem VHDL (*VHSIC Hardware Description Language*), promovendo a compreensão dos conceitos de lógica digital e sua aplicação prática no desenvolvimento de sistemas embarcados e dispositivos programáveis, como FPGAs.

### **2. Enunciado**

Para que um alimento seja classificado como *light*, seu valor calórico deve corresponder, **no máximo**, a **50%** das calorias presentes no produto original. Considera-se, neste contexto, a adição de ingredientes opcionais utilizados para realçar o sabor e a coloração do alimento, os quais contribuem com diferentes percentuais calóricos em relação ao produto normal, conforme descrito a seguir:

- A contém 40%
- B contém 30%
- C contém 20%
- D contém 10%

A partir dessas definições, monte a tabela-verdade, o mapa de Karnaugh e implemente em linguagem VHDL a solução do problema. Projete um circuito para **acender uma lâmpada** cada vez que a combinação dos produtos misturados em um tanque **ultrapassar 50%** das calorias de um produto normal.

### **3. Desenvolvimento**

a. Dedução da Tabela Verdade:

Queremos acender uma lâmpada ( $L = 1$ ) quando a soma das calorias dos ingredientes presentes na mistura for maior que 50%. Cada ingrediente pode estar presente (1) ou não (0). Portanto, temos 4 entradas binárias: A, B, C, D, totalizando em 16 combinações possíveis ( $2^4$ ).

Com isso, calculamos a soma para cada combinação possível verificando quais ultrapassam 50%. Sendo assim, foi deduzido a seguinte tabela verdade:

A (40%)	B (30%)	C (20%)	D (10%)	Soma (%)	L (>50%)
0	0	0	0	0	0
0	0	0	1	10	0
0	0	1	0	20	0
0	0	1	1	30	0
0	1	0	0	30	0
0	1	0	1	40	0
0	1	1	0	50	0
0	1	1	1	60	1
1	0	0	0	40	0
1	0	0	1	50	0
1	0	1	0	60	1
1	0	1	1	70	1
1	1	0	0	70	1
1	1	0	1	80	1
1	1	1	0	90	1
1	1	1	1	100	1

b. Dedução da expressão lógica correspondente à saída:

Em seguida, simplificamos os mintermos pelo mapa de Karnaugh encontrando-se a seguinte expressão na forma SOP:

CD AB	00	01	11	10
00	0 0	0 1	0 3	0 2
01	0 4	0 5	1 7	0 6
11	1 12	1 13	1 15	1 14
10	0 8	0 9	1 11	1 10

Fonte: <https://karnaughmapsolver.com/pt>.

Expressão lógica:  $L = A \cdot B + A \cdot C + B \cdot C \cdot D$

c. Implementação dos Códigos em VHDL:

Após isso, foram desenvolvidos os códigos do Módulo Principal “*classificacao\_alimentos*” e da Bancada de Teste “*tb\_classificacao\_alimentos*” em extensão (.vhd);

“*classificacao\_alimentos.vhd*”

```
-- Residencia Tecnologica em FPGA - Residente: Hirislayne Batista
-- Data: 10/09/2025
-- Descrição: Sistema que aciona uma lâmpada sempre que o limite de
-- 50% de composição calórica dos ingredientes for ultrapassado.

-- Arquivo do Módulo principal
library ieee;
use ieee.std_logic_1164.all;

entity classificacao_alimentos is
    port (
        A : in std_logic;
        B : in std_logic;
        C : in std_logic;
        D : in std_logic;
        L : out std_logic
    );
end entity;
```

```
architecture behavior of classificacao_alimentos is
begin
    L <= (A and B) or (A and C) or (B and C and D);
end architecture;
```

“tb\_classificacao\_alimentos.vhd”

```
-- Arquivo de teste do módulo principal
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all; -- Para usar to_unsigned

entity tb_classificacao_alimentos is
end entity tb_classificacao_alimentos;

architecture sim of tb_classificacao_alimentos is
    signal A, B, C, D, L : std_logic;
begin
    DUT: entity work.classificacao_alimentos port map (A => A, B =>
B, C => C, D => D, L => L);

    stimulus: process
    begin
        for i in 0 to 15 loop
            (A, B, C, D) <= std_logic_vector(to_unsigned(i, 4));
            wait for 10 ns;
        end loop;
        wait;
    end process;
end architecture;
```

#### d. Compilação e Simulação

Execute os seguintes comandos para compilação e simulação:

```
ghdl -a classificacao_alimentos.vhd
ghdl -a tb_classificacao_alimentos.vhd
ghdl -e tb_classificacao_alimentos
ghdl -r tb_classificacao_alimentos --vcd=waveform.vcd
```

Por meio do comando *dir* é possível ver o diretório completo, em que deve conter os 4 arquivos necessários para a próxima etapa (.vhd, .vcd e .cf):

```
Diretório: C:\Users\Hirislayne Batista\Documents\Residencia Embarcatech - FPGA\Tarefas\Classificador_Alimentos

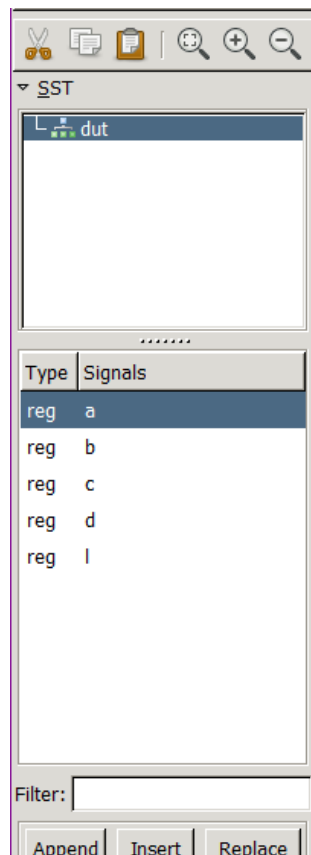
Mode                LastWriteTime         Length Name
----                -
-a----          10/09/2025   16:09             625 classificacao_alimentos.vhd
-a----          10/09/2025   16:26             649 tb_classificacao_alimentos.vhd
-a----          10/09/2025   16:26             824 waveform.vcd
-a----          10/09/2025   16:26             431 work-obj93.cf
```

Fonte: Terminal no VSCode.

Por fim, execute o GTKWave para a Visualização das Formas de Ondas a partir do arquivo .vcd por meio do comando:

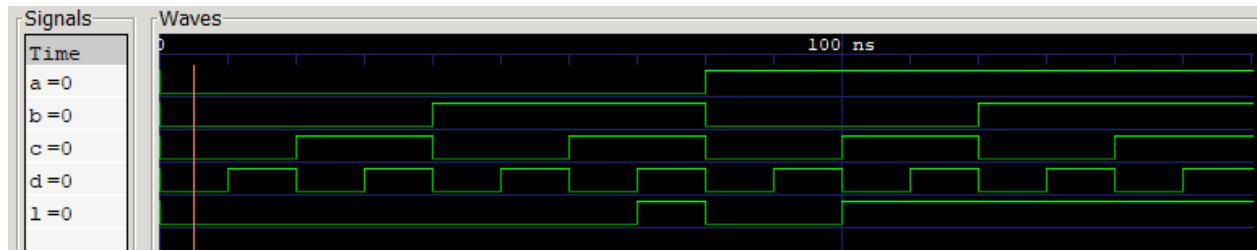
*gtkwave waveform.vcd*

Na tela do GtkWave adicione os sinais (contidos na instância *uut* criada na aba do canto superior esquerdo) e ajuste as formas de ondas por meio da lupa "Zoom Fit" para melhor visualização.



Fonte: GTKWave no VSCode.

Sendo assim podemos ver as combinações das saídas para cada entrada, conforme tabela.



Fonte: GTKWave no VSCode.

Link para acesso à pasta do projeto: [Arquivo Tarefa U3C1 - Google Drive](#).