

**Departamento de Engenharia Elétrica e de Computação**

**SEL0384 – Laboratório de Sistemas Digitais I**

**Prof. Dr. Maximilian Luppe**

## **PRÁTICA Nº7**

### **Dispositivos de Lógica Programável tipo FPGA**

#### **Circuitos Sequenciais**

Objetivos:

Familiarização com a ferramenta Quartus Lite da Intel/Altera e Projeto e síntese de circuitos sequenciais em dispositivo reconfigurável (FPGA).

Equipamentos necessários:

- Kit Mercurio® IV

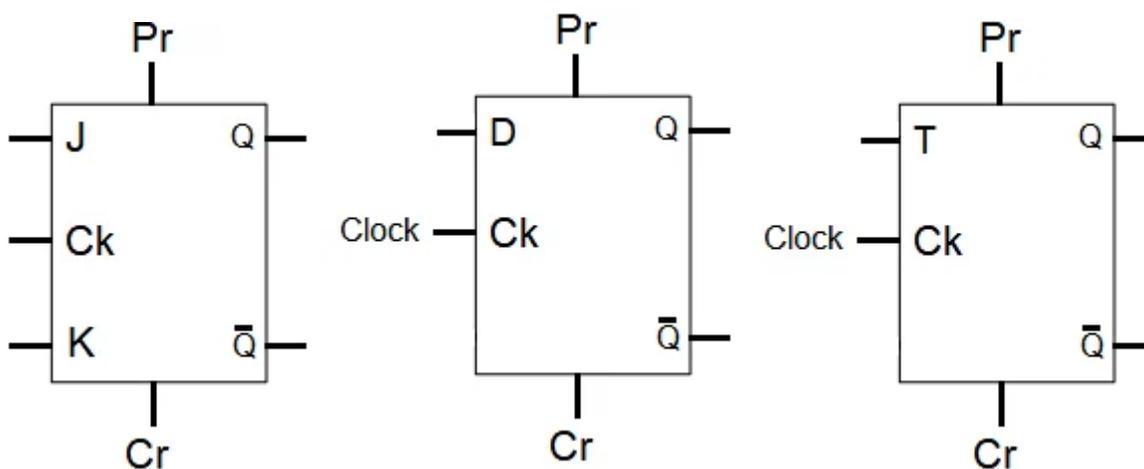
Introdução:

Esta prática de laboratório tem por objetivo a implementação de módulos de Flip-flop JK-MS, Tipo-D e Tipo-T, com reset assíncrono e borda positiva de clock, baseados em Equação Característica, no kit Mercurio® IV (Cyclone® IV EP4CE30F23).

Os Flip-flops (FF) são os elementos básicos dos Circuitos Sequenciais, assim como as portas lógicas AND, OR e NOT são dos Circuitos Combinacionais. A partir dos FF é possível implementar Registradores (Seriais ou Paralelos), Contadores (Assíncronos ou Síncronos) e Máquinas de Estados Finitos (tipo Mealy ou Moore), e, a partir destes, realizar diversas aplicações, como a implementação de Contadores de Programas (PC - *Program Counter*), Banco de Registradores e Unidades de Controle, na área de Arquitetura de Computadores, por exemplo.

Na figura 1 temos os principais FF: JK-MS, Tipo-D e Tipo-T. Estes podem possuir diversos sinais de controle, além, do Clock, como Reset (Clear ou Cr), que faz sua saída Q ser igual a '0', Set (ou Preset ou Pr), que faz sua saída Q ser igual '1', Clock-Enable, que habilita o funcionamento do Clock, Output-Enable, que habilita a saída do dado armazenado por meio de uma saída em Tri-state, sendo que estes sinais podem ser ativos em '1' ou '0'. Quanto ao Clock, este pode ser ativo em borda de subida (positiva) ou borda de descida (negativa), e quanto ao Reset e o Set, estes podem ser Síncronos, controlado pelo Clock, ou Assíncronos, sem controle pelo Clock.

Figura 1 - FF JK-MS. Tipo-D e Tipo-T



Fonte: [site da internet](#)

Destes, o mais simples de se implementar em VHDL é o FF Tipo-D. Os demais (JK-MS e Tipo-T), podem ser implementados por meio de sua Equação Característica:

$$Q_{JK} = J \cdot \bar{Q} + \bar{K} \cdot Q \quad (\text{eq. 1})$$

$$Q_T = T \cdot \bar{Q} + \bar{T} \cdot Q = T \oplus Q \quad (\text{eq. 2})$$

Procedimento Experimental:

Apresentar a implementação dos FF JK-MS, Tipo-D e Tipo-T, baseados na sua Equação Característica, com reset assíncrono, utilizando a linguagem de descrição de hardware VHDL.

Criar uma pasta denominada `MercurioIV_FF`, com as subpastas `docs`, `modelsim`, `quartus` e `src`, e, utilizando o Quartus Lite Edition, criar um projeto na pasta `quartus`, também denominado `MercurioIV_FF`.

Realizar a implementação das entidades dos FF JK-MS, Tipo-D e Tipo-T utilizando a linguagem de descrição de hardware VHDL, denominados `jk_ff.vhd`, `d_ff.vhd` e `t_ff.vhd`, respectivamente, com entrada de Clock `clk` de borda positiva, entrada de Clear `clr` ativa em '1', uma saída `q`, e as entradas `j` e `k`, `d`, e `t`, respectivamente.

Verifique o funcionamento dos FFs no kit Mercurio® IV incluindo o arquivo `MercurioIV_FF.vhd` ao projeto `MercurioIV_FF`, definindo este arquivo como "Top-Level Entity", incorporando o código dos FF (`jk_ff.vhd`, `d_ff.vhd` e `t_ff.vhd`) ao projeto principal (`MercurioIV_FF`), ligando as chaves SW(3 downto 0) às entradas `j`, `k`, `d` e `t`, respectivamente, o push-button KEY(0) à entrada `clk`, o push-button KEY(2) à entrada `clr`, e a saída `q` dos FF JK-MS, Tipo-D e Tipo-T aos LEDM\_C(2 downto 0), respectivamente, e executar o projeto no kit Mercurio® IV.

Apresentar código VHDL, circuito RTL, número de células lógicas utilizadas e foto do kit com o circuito funcionando.

#### Listagem 1 - Código de implementação do FF Tipo-D

```
entity d_ff is
    port (
        clk, clr, d : in bit;
        q           : out bit
    )
end d_ff;

architecture rtl of d_ff is
begin
    process (clk, clr)
    begin
        if (clr = '1') then
            q <= '0';
        elsif (clk'event and clk = '1') then
            q <= d;
        end if;
    end process;
end rtl;
```