

# Laboratório de Sistemas Digitais

Ano Letivo 2020/21

## Proposta de Projeto Final

### Projeto nº 9 – Cronometro Digital

#### 1. Introdução

O objetivo deste trabalho é modelar em VHDL e testar na FPGA um Cronometro Digital. Assume-se que o Cronometro exibe informação de minutos, segundos, centésimos de segundo. Para a interação com o utilizador e para o acerto das horas devem ser consideradas as seguintes entradas e saídas:

**Entradas:** Seleção de modo (**SWs**), programar tempo (**KEYs**).

**Saídas:** Displays de 7 segmentos como mostrador de tempo e indicação de modo de funcionamento (conforme a descrição da última fase). LED vermelho indicador dos segundos.

#### 2. Descrição do funcionamento e requisitos

As principais especificações do relógio digital são as seguintes:

- Ligando o relógio o mostrador de tempo deve exibir a informação **00:00:00** (mapeados nos displays **HEX0** a **HEX5**). A indicação do modo “**r-**” nos displays **HEX6** a **HEX7**. O relógio funciona em regime normal contando a passagem do tempo no sentido decrescente.
- Atuando em **KEY(0)** deve dar entrada o modo de programação de tempo. A indicação de modo deve exibir “**PP**” nos displays **HEX6** a **HEX7**.
- No modo de programação, as **KEY(1)** e **KEY(2)** permitem a incrementação e decrementação do dígito selecionado, respetivamente.
- **KEY(3)** usado para arrancar/parar o cronometro. Com o cronometro parado, **KEY(3)** arranca o processo de contagem. Se pressionado outras vezes, permite parar e resumir a contagem.
- **KEY(0)** permite, no modo de programação, selecionar o dígito a acertar, sempre na direção dos minutos para os segundos (os centésimos de segundo não são programáveis).
- O dígito selecionado para programação deve piscar a uma frequência de **2Hz**.
- O modo de programação termina com o acerto do dígito dos minutos e pressão em **KEY(0)**, retomando para o modo normal de funcionamento.
- **SW(1)** permite selecionar o modo contagem crescente ‘**0**’ ou decrescente ‘**1**’. O indicador do modo (**HEX6** e **HEX7**) de contagem deve exibir “**r-**” ou “**r\_**” consoante a contagem seja decrescente ou crescente.
- O refrescamento do display é feito um dígito de cada vez com uma frequência de **1 kHz**.
- O controlo do refrescamento do display é feito pela máquina de estados **DispCtrl**.

- A passagem dos segundos deve ser indicada por um LED vermelho (**LEDR(0)**) a piscar à frequência de **1Hz**.
- Uma entrada de reset geral ativada com **SW(0)**.

Na execução do projeto, deve contemplar os seguintes requisitos de simulação e depuração do projeto:

- Simulação funcional com testbenches e Modelsim de todos os componentes do projeto.
- Depuração do funcionamento do relógio com signal-tap-logic-analyzer da condição **01:00:00 - > 00:59:99**.

### 3. Implementação

A implementação deste trabalho deve ser baseada em máquinas de estado comunicantes. Sugere-se uma estratégia faseada, de acordo com a descrição que se segue:

**Fase 1 (14 valores):** Implementação do relógio sem modo de programação, com contagem decrescente assumindo inicialização de 1 minuto e sem indicação do modo. Tendo por base uma só máquina de estados capaz de gerir o display (**DispCntrl**).

**Fase 2 (3 valores):** Implementação do modo de programação, tendo por base uma máquina de estados dedicada ao acerto (**ProgCntrl**) e uma máquina de estados geral (**MainCntrl**) capaz de alternar gerir os modos de funcionamento de cronometro.

- No modo de contagem crescente, **SW(1) = '0'**, a contagem deve ser feita até ao limite de representação do cronometro, altura em que o cronometro para.
- No modo decrescente **SW(1) = '1'**, a contagem só deve dar inicio depois de programado o valor inicial.

### 4. Requisitos para obtenção de classificações superiores a 17 valores

- No modo de acerto reduzir o número de botões de pressão para o processo para dois, um para seleccionar o dígito – **KEY(0)** - e outro para acertar **KEY(1)**. Se **KEY(1)** é pressionado de forma continuada a dígito é decrementado. Se **KEY(1)** é pressionado de forma fixa o dígito é incrementado à frequência de **2Hz**.
- Implementação do modo de contagem oculto. Se parado durante a contagem com **KEY(3)**, o cronometro exhibe o valor de contagem fixo, mas fica a contar em background (sem refrescar o display) até que seja resumida a operação.

**Nota:** O Top-level do Relógio Digital deverá ser implementado com recurso a representação estrutural em VHDL.

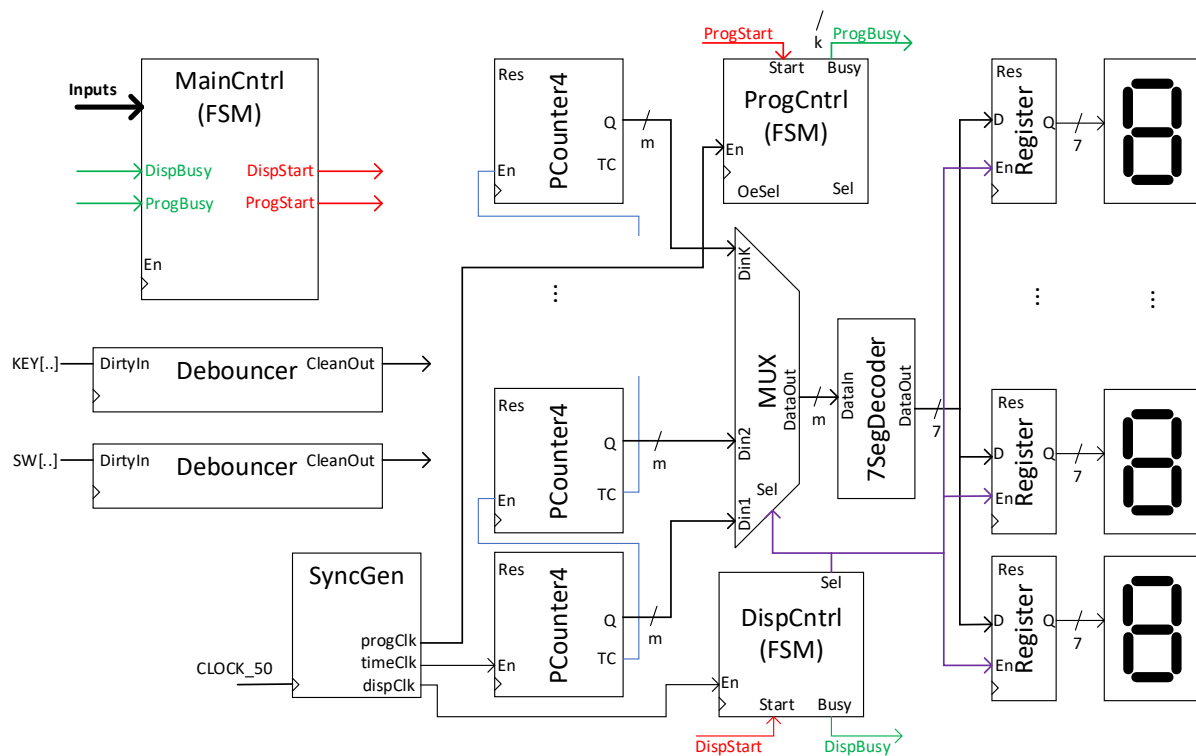


Fig. 1: Arquitetura do cronometro a implementar.

#### Notas sobre a arquitetura:

- A arquitetura não se encontra totalmente especificada. Em particular os sinais indicados a cores.
- O sinal de **clk** nos blocos sequenciais é obrigatoriamente o relógio de 50 MHz, **CLOCK\_50**.
- O sinal **Res** nos blocos sequenciais corresponde à implementação do reset e deve ser de diferentes formas consoante as fases de desenvolvimento:
  - Sinal reset geral com **SW(0)** na fase 1 e no evento de alguma contagem de interesse.
  - A Máquina de estados **MainCntrl** deve gerir o sinal de reset para os demais componentes, na fase 2 e posteriores.
- Os contadores a utilizar devem ter fim de contagem parametrizável, bem como uma saída de final de contagem, ativa por um pulso de relógio.