

# Projeto 1: Relógio Digital Microprocessado

## Descrição

Este projeto será a criação de um relógio digital, em VHDL, com as seguintes características:

- Indica horas e minutos (segundos é opcional);
- Possui algum sistema para acertar o horário;
- Possui seleção da base de tempo:
  - Para mostrar a passagem das 24 horas em tempo reduzido:
    - Utilizado para verificar se o projeto funciona corretamente.

Opcionais que adicionam valor ao projeto:

- Montador (assembler) para o processador criado:
  - Deve ser em Python.
- Sistema de despertador;
- Temporizador com contagem regressiva;
- Indicação do horário com base em 12 horas - AM/PM;
- Mensagens enviadas ao Display de Cristal Líquido;
- Caso tenha ideia para algum opcional, confira antes se ele adiciona valor ao projeto.

Existem poucos critérios que limitem a liberdade de criação neste projeto:

- Deve utilizar um processador específico, projetado para essa finalidade:
  - O barramento de dados, pode ser de 4 ou 8 bits;
  - O barramento da memória de instruções pode ser de qualquer largura.
- O horário deverá ser mostrado através do display de sete segmentos.

Dica:

- A arquitetura *Harvard* pode diminuir a complexidade e ser mais eficiente para este projeto.

## Entrega

A entrega e apresentação será no dia da **Avaliação Intermediária**.

A avaliação do projeto será feita através dos seguintes itens:

- Apresentação do projeto, no kit de desenvolvimento:
  - Com arguição do(s) professor(es).
- Entrega de um resumo explicativo do funcionamento do circuito, contendo:
  - Diagrama de blocos do projeto;
  - Com comentários (opcional) sobre os problemas encontrados.
- Entrega do projeto do Quartus, com:
  - Código VHDL, **devidamente documentado** e com o nome dos participantes.

## Execução

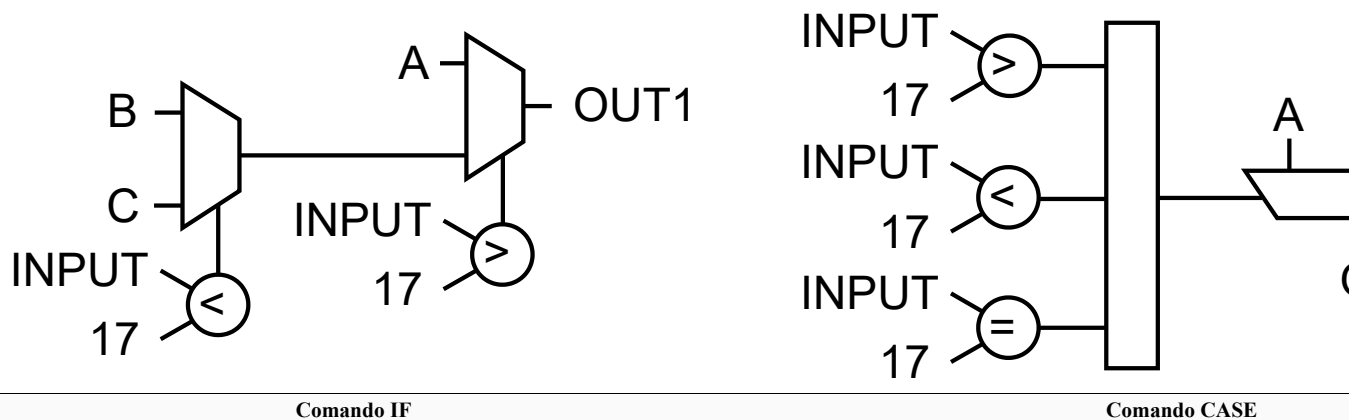
Em grupo, com no máximo 3 alunos e no mínimo 2 alunos.

## Rubrica

Rubricas para o Relógio			
Nota	Critério	Documentação	Montador
A	Entregou o projeto básico com dois ou três itens opcionais. Funciona sem erros.	Código bem comentado. Manual com descrição do projeto e instruções de uso.	Obrigatório
B	Entregou o projeto básico com um ou dois itens opcionais. Funciona com pequenos erros.	Código bem comentado. Manual com descrição do projeto e instruções de uso.	Opcional
C	Entregou o projeto básico. Funciona com pequenos erros.	Código comentado e um simples manual de instruções de uso.	Opcional
D	Entregou o projeto básico com erros graves ou incompleto.	N.A.	N.A.
I	Não entregou o projeto.	N.A.	N.A.

## Um rascunho de metodologia para converter instruções em circuito digital

1. Liste as funções que o projeto terá, incluindo as entradas e saídas do circuito.
2. Crie o pseudocódigo para o programa do relógio:
  - A partir dele, liste as instruções necessárias para que o processador execute o código.
3. Liste as unidades funcionais (hardware) necessárias para executar o pseudocódigo do item anterior:
  - Converta o pseudocódigo lembrando que:
    - O comando *if* e o *case* são equivalentes ao multiplex (com ponto(s) de controle);



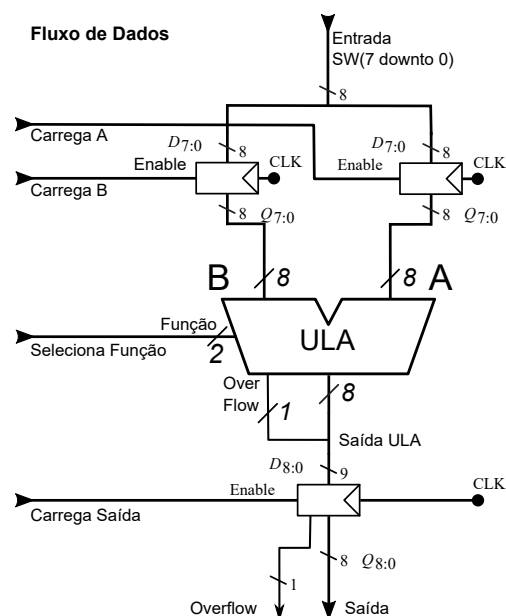
3. Continuação:

- Conversão do pseudocódigo:
  - As variáveis são equivalentes a registradores ou banco de registradores (com ponto(s) de controle);
  - Uma comparação é o resultado XNOR dos dois valores ou a utilização de um comparador;
  - Para detectar um resultado igual a zero, use uma porta NOR (todas entradas = 0, saída = 1);
  - As operações lógicas e aritméticas são feitas pela ULA (com ponto(s) de controle);
  - Uma operação de entrada ou saída requer o endereçamento do dispositivo de E/S;

4. Esboce um fluxo de dados (FD) que implemente a função básica:

- Esse esboço deve possuir a interligação dos componentes e a definição dos pontos de controle (crie um nome significativo para cada ponto de controle);
- Defina um vetor de bits que reúne todos os pontos de controle. Essa é a sua palavra de controle.

Exemplo de fluxo de dados com os pontos de controle:



5. Simule, com papel e lápis, o funcionamento do FD:

- Levante a sequência de ativação dos pontos de controle (palavras de controle), para cada função que deseja executar:
  - Cada combinação distinta de valores para os pontos de controle equivale a uma instrução do processador.
- Uma função pode ativar vários pontos de controle ao mesmo tempo e/ou em sequência.

6. Quando o FD básico estiver pronto, codifique em VHDL e simule no Quartus:

- Na simulação, verifique se as palavras de controle estão corretas, tanto no conteúdo quanto na sequência, e anote qualquer alteração necessária.

7. Repita o processo para criar novas funcionalidades:

- Pode ser feito partindo-se do FD atual;
- Ou com um FD novo, que depois terá de ser integrado ao existente com a remoção dos elementos que são comuns.

8. Com o FD completo e testado, utilize as anotações das palavras de controle para fazer o programa do relógio.

Uma forma de definir as operações do fluxo de dados é utilizando RTL (Register Transfer Language):

- Que possui uma notação (RTN) para exprimir as operações que ocorrem em um ciclo de *clock*. Ou seja, cada passo elementar do processo ou instrução.
- A explicação mais resumida, que encontrei, sobre essa notação está em:
  - [Notação](#).
  - [Operações](#).

---

## Dicas

Este [link](#) mostra alguns componentes que foram testados e são funcionais.

---

## Leituras

Livro texto (Organização e Projeto de Computadores: A Interface Hardware/Software):

- Capítulo 4, item 3 (construindo um fluxo de dados);
- [Apêndice D](#) completo. Trata de máquinas de estados, microprogramação e unidade de controle:
  - O link acima é para o site da editora, que disponibiliza, de forma gratuita, esse apêndice.

Caso queira ler mais, veja o livro do Stallings (Arquitetura e Organização de Computadores):

- Capítulos 14 e 15.

---

## Ferramentas

Acredito que o *Logic Friday* possa ser de alguma valia. Veja nos links úteis.

---

## Referências

[Página com links](#) de referências sobre VHDL, Quartus, etc ...

---

[Ir para o início do documento](#).