

Universidade do Vale do Itajaí
Disciplina: Projeto de Sistemas Digitais
Professores: Douglas Rossi de Melo
Avaliação 6 – Projeto de processadores em VHDL

Instruções:

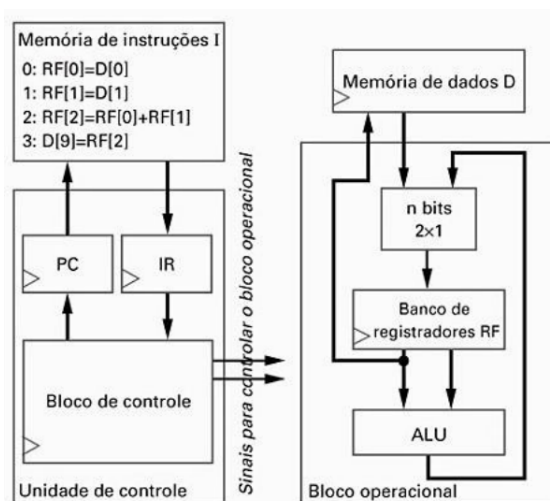
1. Esta avaliação tem por objetivo consolidar o aprendizado sobre o projeto de processadores em VHDL.
2. O sistema digital a ser implementado consiste em um processador programável básico de 4 instruções.
3. O processador a ser desenvolvido consiste no processador de 3 instruções (básico) apresentado no livro “sistemas digitais - projetos de otimização e hdl’s”, acrescido da instrução de carga de constante (load constant).
4. O processador descrito em VHDL deve ser verificado com o uso de testbench no simulador ModelSim e sintetizado no Quartus para verificação com o uso do visualizador de RTL para conferir os circuitos inferidos.
5. Para a implementação, deve-se utilizar o Quartus para identificar os custos de implementação do sistema expresso pelo número de LUTs e FFs.
6. Aplique o seguinte guia de estilo na codificação VHDL:
 - a. Tabulação com 02 espaços (substituindo tabs por espaços);
 - b. Palavras reservadas da linguagem em caixa baixa;
 - c. Nomes de sinais em caixa alta (exceto os prefixos e sufixos);
 - d. Utilize os seguintes prefixos para identificar o tipo sinal:
 - i_NOME : pino de entrada
 - o_NOME : pino de saída
 - b_NOME : pino de entrada-e-saída (bidirecional)
 - w_NOME : fio interno
 - v_NOME : variável de processo
 - r_NOME : registrador
 - t_NOME : tipo de dado
 - p_NOME : processo
 - u_NOME : instância de componente
 - e. Utilize o sufixo “n” para sinalizar que o sinal é ativo em 0 (negado).
Ex: i_RSTn
7. Devem ser apresentados as seguintes estruturas na defesa do trabalho:
 - a. Código fonte do projeto e testbench
 - b. Diagrama de formas de onda da simulação
 - c. Diagrama RTL gerado pelo Quartus
 - d. Tabela de custos do circuito (LUTs e FFs)
 - e. Frequência máxima do circuito

Descrição do projeto a ser desenvolvido

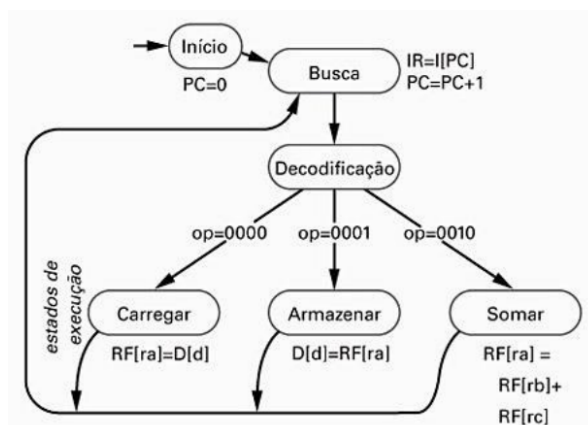
Lista de instruções:

- Instrução **Carregar**– **0000** $r_3r_2r_1r_0$ **d₇d₆d₅d₄d₃d₂d₁d₀**: Essa instrução especifica uma movimentação de dados que vai desde uma posição da memória de dados, cujo endereço é especificado pelo bits $d_7d_6d_5d_4d_3d_2d_1d_0$, até um registrador do banco de registradores, cuja posição é especificada pelo bits $r_3r_2r_1r_0$. Por exemplo, a instrução “0000 0000 00000000” especifica uma movimentação de dados desde a posição 0 da memória de dados, ou $D[0]$, até a posição 0 do banco de registradores, ou $RF[0]$; em outras palavras, essa instrução representa a operação $RF[0]=D[0]$. De modo similar, “0000 0001 00101010” especifica $RF[1]=D[42]$. Inserimos espaços em branco entre alguns bits para facilitar a leitura do leitor—esses espaços não têm nenhum outro significado e não existem na memória de instruções.
- Instrução **Armazenar**– **0001** $r_3r_2r_1r_0$ **d₇d₆d₅d₄d₃d₂d₁d₀**: Essa instrução especifica uma movimentação de dados no sentido oposto ao da instrução anterior, ou seja, uma movimentação que vai do banco de registradores até a memória de dados. Assim, “0001 0000 00001001” especifica $D[9]=RF[0]$.
- Instrução **Somar**– **0010** $ra_3ra_2ra_1ra_0$ $rb_3rb_2rb_1rb_0$ $rc_3rc_2rc_1rc_0$: Essa instrução especifica somar os conteúdos de dois registradores do banco de registradores, especificados por $rb_3rb_2rb_1rb_0$ e $rc_3rc_2rc_1rc_0$. O resultado é armazenado em um banco de registradores, no registrador especificado por $ra_3ra_2ra_1ra_0$. Por exemplo, “0010 0010 0000 0001” especifica a instrução $RF[2]=RF[0]+RF[1]$. Observe que *somar* é uma operação de ALU.

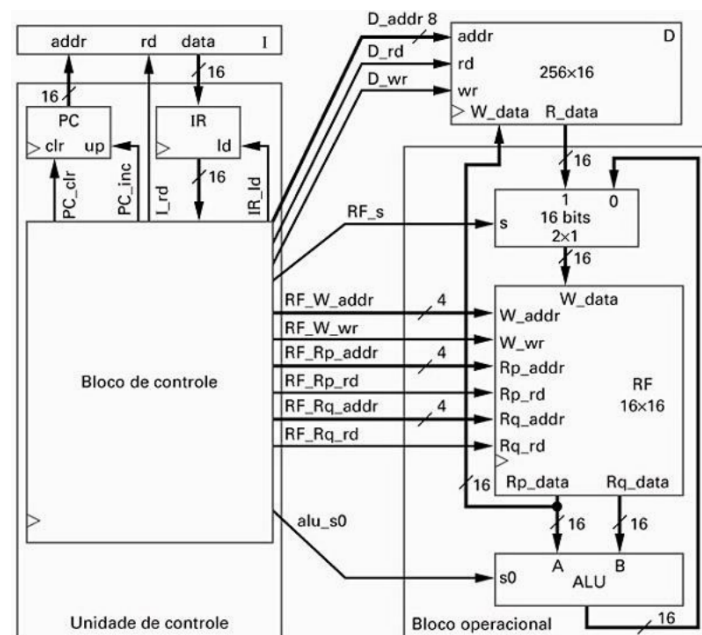
Visão geral do processador programável de 3 instruções:



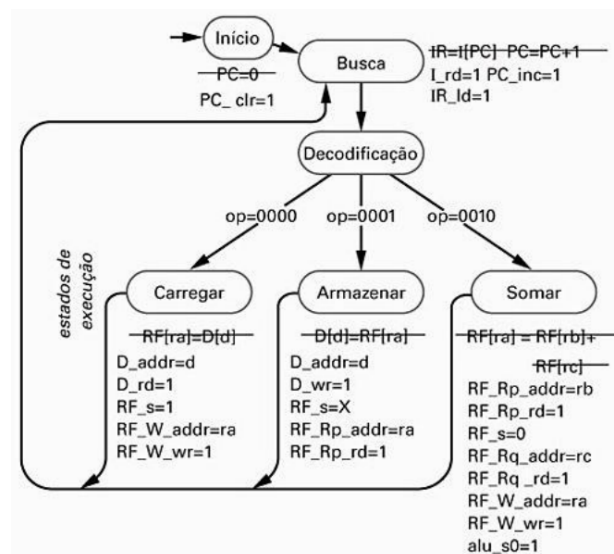
Controle básico do processador:



Caminho de dados do processador:



Máquinas de estados do controle:



A adição de carga de constante (load constant) deve ser baseada na instrução presente no processador de 6 instruções, apresentado no mesmo livro.

Requisitos:

1. O sistema deve ser composto de um controle (máquina de estados) e um caminho de dados;
2. O projeto deve seguir uma abordagem hierárquica, com a seguinte estrutura sugerida
 - Topo (CPU)
 - Controle (máquina de estados)
 - Caminho de dados
3. Ao realizar o projeto, identifique previamente e claramente os nomes das entradas, saídas e sinais internos nos diagramas esquemáticos de modo a facilitar a documentação e a descrição do projeto em VHDL;
4. Para sinais internos, utilize como referência a porta de saída e o nome do bloco ao qual o sinal está conectado. Exemplo: para um sinal de 8 bits conectado à saída de um somador, utilize o seguinte identificador: `w_ADDER_OUT`. Caso o sistema contenha mais de uma instância de um mesmo bloco, diferencie-os com algum sufixo (ex. `w_ADDER0_out`, `w_ADDER1_out`, ...);
5. Para a verificação, deve ser elaborado um programa que realize as seguintes operações, evidenciadas nas formas de onda:
`D[0] := 4`
`D[1] := D[0] + 2`
`D[0] := D[0] + D[1]`