

TP1 — Verilog

Documentação

Disciplina: Organização de Computadores II

Professor: Omar Paranaíba

Grupo: Clarice Oliveira, João Victor Tamm e Renato Junior

Data: 16/10/17

1. Introdução

Esta atividade consiste em uma das etapas do desenvolvimento da descrição RTL de um processador que será prototipado em uma FPGA. Nesta etapa foi elaborada a unidade lógico aritmética e a sua conexão com o banco de registradores realizado na etapa anterior.

2. Implementação

2.1. Detalhes Importantes

- O tamanho da palavra é 16 bits.
- Todos os códigos fonte das unidades implementadas se encontram dentro da pasta ./src encontrada na raiz do projeto.

2.2. Unidades

2.2.1 Banco de registradores

Arquivo: Banco_registradores.v

Entradas: RW, [4:0] regA, [4:0] regB, [4:0] regC, [15:0] dado, clk, imediato, flagImediato.

Saídas: [15:0] regsaidaA, [15:0] regsaidaB

Especificação das entradas e saídas:

- **RW** é a flag que determina se a unidade funcionará em modo de *leitura* (0) ou *escrita* (1).
- **regA** é o endereço de um dos registradores que será lido.
- **regB** é o endereço do outro registrador de leitura.
- **regC** é o endereço do registrador a ser escrito.
- **imediato** é o valor do imediato que será utilizado em algumas operações.
- **flagImediato** é a flag que determina quando um imediato é usado na operação.
- **dado** é o dado a ser escrito no registrador.
- **clk** é o clock.
- **regsaidaA** é o dado lido do registrador a.
- **regsaidaB** é o dado lido do registrador b.

2.2.2 Unidade Lógico-Aritmética (ALU)

Arquivo: ALU.v

Entradas: [3:0] codop, [15:0] operando1, [15:0] operando2, [15:0] imm, clk

Saídas: [15:0] resultado, neg, zero, overflow

Especificação das entradas e saídas:

- **codop** é a flag que especifica *o modo de operação da ALU*. **0** determina operação de *adição*, **1** determina operação de *subtração*, **2** determina a operação de *comparação*, **3** determina a operação *conjunção lógica* (and), **4** determina a operação *disjunção lógica* (or), **5** determina a operação *xor*, **6** determina a operação *and com um operador imediato*, **7** determina a operação *or com um operador imediato*, **8** determina a operação *xor com um operador imediato*, **9** determina a operação de *adição com um operador imediato* e **10** determina a operação de *subtração com um operador imediato*.
- **operando1** é o primeiro operando.
- **operando2** é o segundo operando.
- **imm** é o operando imediato.
- **clk** é o clock.
- **resultado** é o resultado da operação executada.
- **neg** é a flag que determina se o resultado é negativo.
- **zero** é a flag que determina se a o resultado de saída é zero.

2.2.3 Unidade de Controle

Arquivo: Processador.v

Variáveis auxiliares: bancoRW

Entradas: [3:0] KEY, [17:0]SW, CLOCK_50

Saídas: [6:0] HEX0, [6:0] HEX1, [6:0] HEX2, [6:0] HEX3, [6:0] HEX4, [6:0] HEX5, [6:0] HEX6, [6:0] HEX7

Especificação das entradas e saídas:

- **KEY** é a flag relacionada aos botões de entrada da placa FPGA. A chave **KEY0** é utilizada quando se deseja imprimir no display de 7 segmentos o conteúdo dos registradores previamente definidos através dos switches. Já a **KEY3** tem como função principal realizar uma determinada operação, sendo que o opcode e os registradores, tanto de destino quanto de entrada, foram também pré determinados através dos switches.
- **SW** é a representação dos switches encontrados na FPGA Altera.
- **CLOCK_50** é um input determinado para acessar o clock de 50MHz do próprio dispositivo.
- **HEX0...HEX7** correspondem aos valores que devem ser impressos em cada um dos displays existentes.
- **clk** é o clock.

- **resultado** é o resultado da operação executada.
- **neg** é a flag que determina se o resultado é negativo.
- **zero** é a flag que determina se o resultado de saída é zero.

3. Instruções Implementadas

CODOP	Instrução	Mnemônico	Operação
0	Add	Add \$s4, \$s3, \$s2	$\$s4 = \$s3 + \$s2$
1	Sub	Sub \$s4, \$s3, \$s2	$\$s4 = \$s3 - \$s2$
2	Slti	Slti \$s4, Imm, \$s2	If ($\$s2 > Imm$) ? $\$s4=1 : \$s4=0$
3	And	And \$s4, \$s3, \$s2	$\$s4 = \$s3 \& \$s2$
4	Or	Or \$s4, \$s3, \$s2	$\$s4 = \$s3 \$s2$
5	Xor	Xor \$s4, \$s3, \$s2	$\$s4 = \$s3 \wedge \$s2$
6	Andi	Andi \$s4, Imm, \$s2	$\$s4 = \$s2 \& Imm$
7	Ori	Ori \$s4, Imm, \$s2	$\$s4 = \$s2 Imm$
8	Xori	Xori \$s4, Imm, \$s2	$\$s4 = \$s2 \wedge Imm$
9	Addi	Addi \$s4, Imm, \$s2	$\$s4 = \$s2 + Imm$
10	Subi	Subi \$s4, Imm, \$s2	$\$s4 = \$s2 - Imm$

4. Especificação dos Arquivos

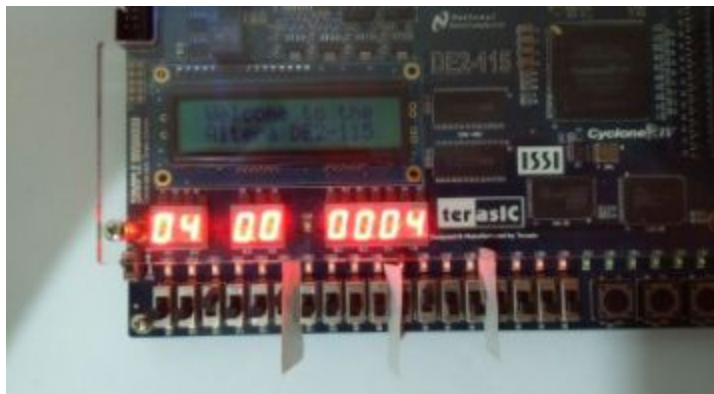
Arquivo	Descrição
Banco_registradores.v	Módulo responsável por realizar a implementação de um banco com 32 registradores de tamanho 16 bits.
ALU.v	Módulo responsável por realizar a implementação de uma Unidade Lógico-Aritmética contemplando todas as operações

	destacadas na tabela apresentada na seção anterior.
Processador.v	Módulo principal do sistema. Responsável por realizar a integração e controle dos diversos módulos existentes na aplicação além de também gerenciar a comunicação com os componentes da placa, tais como Displays, Keys e Switches. Para que tais coisas sejam possíveis, variáveis auxiliares foram criadas a fim de passar o dado de maneira correta para cada um dos módulos. Sendo assim tais variáveis são setadas de acordo com os dados obtidos através dos componentes da FPGA.

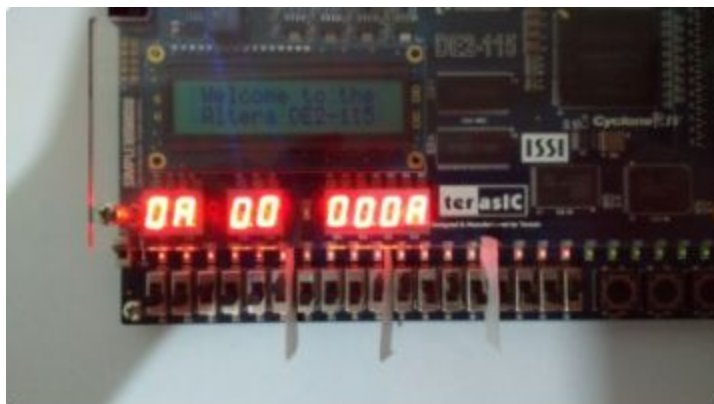
5. Testes

No final do processo, de modo a testar e validar tudo que foi implementado, foram realizados testes na placa FPGA de modo a validar os módulos desenvolvidos até então. Cabe pontuar que os testes foram feitos de maneira sequencial, sendo assim qualquer alteração em um registrador afetará as instruções seguintes. Alguns destes testes estão representados nas figuras abaixo:

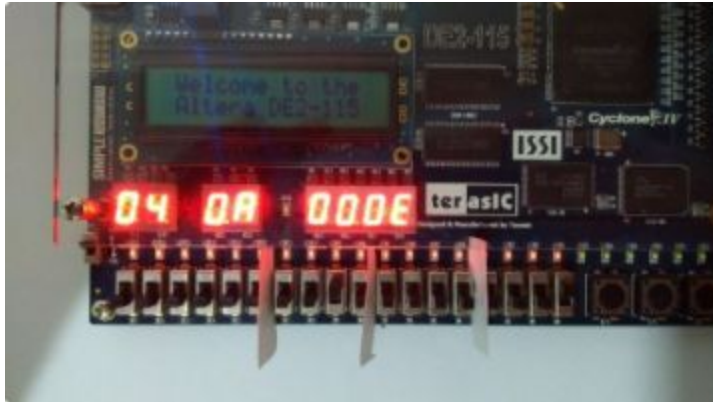
1. `addi $s0, 4, $s0`



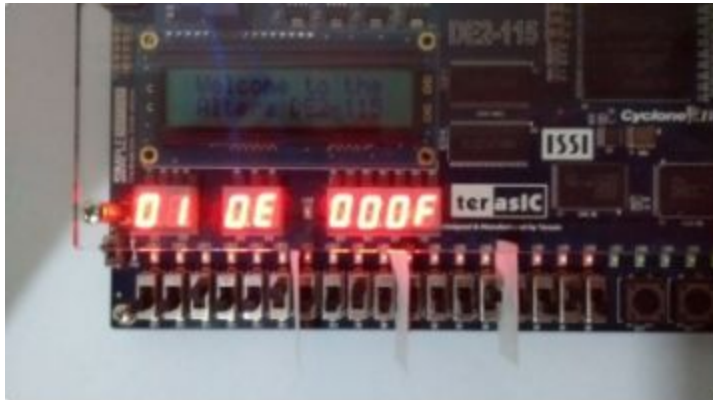
2. `addi $s1, A, $s1`



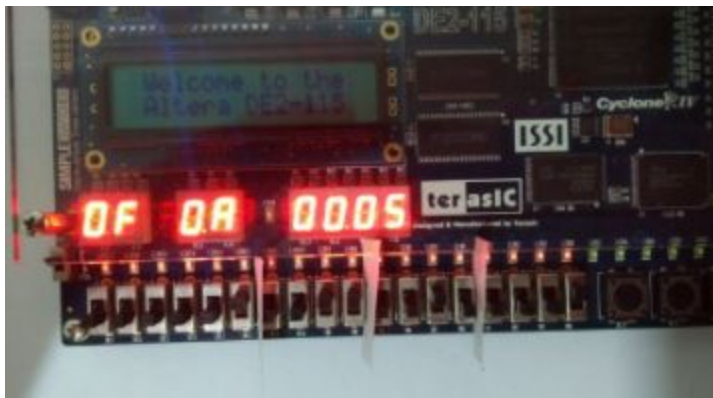
3. add \$s2, \$s0, \$s1



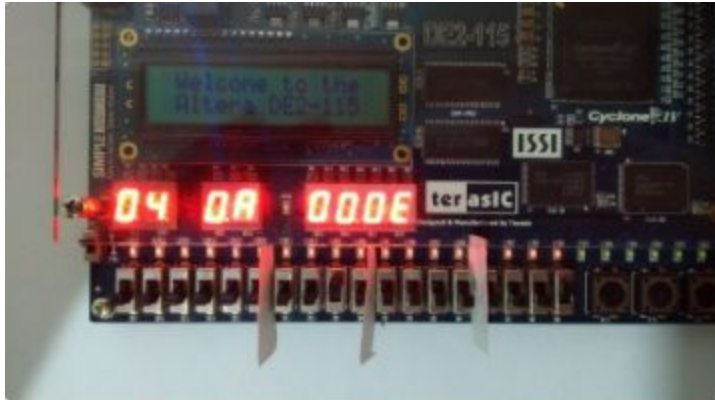
4. addi \$s2, 1, \$s2



5. sub \$s3, \$s2, \$s1



6. `subi $s4, 2, $s3`



6. Bibliografia

- [1] D. A. Patterson, J. L. Henessy. Computer Organization and Design, Fourth Edition, Fourth Edition: The Hardware/Software Interface (The Morgan Kaufmann Series in Computer Architecture and Design) 4th