Tópicos em Sistemas de Computação

Projeto de uma ULA Parametrizável

Objetivo: projetar e simular uma ULA com tamanho de palavra parametrizável.

Características:

- Duas entradas de n bits (A e B)
- Uma Saída de n bits (Z)
- Sinal Zero: detecta valor zero na saída
- Operações:

Operação	Significado	OpCode
add A, B	Saída Z recebe a soma das entradas A, B incluindo o vem-um	000
sub A, B	Saída Z recebe A - B	001
and A, B	Saída Z recebe a operação lógica A and B, bit a bit	010
or A, B	Saída Z recebe a operação lógica A or B, bit a bit	011
not A	Saída Z recebe a entrada A invertida bit a bit	100
xor A, B	Saída Z recebe a operação lógica A xor B, bit a bit	101
bypass A	Saída Z recebe A, sem alteração	110
slt A, B	Z = 1 se A < B	111

Arquivos:

Criar os seguintes arquivos:

- ula.h : define o SC_MODULE (ula)
- ula.cpp : implementa a função realizada pela ula
- ula_tb.h : define o *testbench* para verificação da ula
- ula_tb.cpp : contém a *thread* que gera os estímulos e imprime resultados
- sistema.h : definição do módulo de mais alto nível que instancia ula, ula_tb e interconecta-os
- main.cpp : módulo principal que instancia o sistema e inicia a simulação

Interfaces:

ula.h:

```
#ifndef __ULA_H
#define __ULA_H

#include "systemc.h"

enum OPCODE {
        ADD, SUB, AND, OR, NOT, XOR, BYPASS, SLT

};

#define SIZE 32
```

```
SC_MODULE(ula) {
        sc_in< sc_uint<3> > opcode;
        sc_in< sc_int<SIZE> > A, B;
        sc_out< sc_int<SIZE> > Z;
        sc_out<bool> zero;

        void proc(void);

        SC_CTOR(ula) {
             SC_METHOD(proc);
             sensitive << A << B << opcode;

        }
};
#endif
```

ula_tb.h:

```
#ifndef __ULA_TB
#define __ULA_TB

#include "systemc.h"
#include "ula.h"

SC_MODULE (ula_tb) {
        sc_out<sc_uint<3> > opcode;
        sc_out< sc_int<SIZE> > A, B;
        sc_in< sc_int<SIZE> > Z;
        sc_in<bool> zero;

        void aciona();

        SC_CTOR (ula_tb) {
             SC_THREAD(aciona);
        }
};
#endif
```

top.h:

```
#include "systemc.h"
#include "ula.h"
#include "ula_tb.h"

SC_MODULE ( top ) {
        ula_tb tb;
        ula u;

        sc_signal< sc_int<SIZE> > A, B, Z;
        sc_signal< sc_uint<3> > opcode;
        sc_signal< bool > zero;

        SC_CTOR ( top ) : tb("tb"), u("u") {

             u.opcode(opcode);   tb.opcode(opcode);
             u.A(A);             tb.A(A);
             u.B(B);             tb.B(B);
             u.Z(Z);             tb.Z(Z);
             u.zero(zero);       tb.zero(zero);
        }
};
```

Verificação:

- No testbench gerar entradas para testar cada operação da ULA
- Testar ao menos duas versões variando o tamanho da palavra (uma com 32 bits)