EA773 LABORATÓRIO DE CIRCUITOS LÓGICOS
FEEC-UNICAMP - 1º Semestre de 2012 – Turmas G, H, O e P

Turma:_____ **Grupo:**_____ **Data:**_____

RA:_____ **Nome:**_____

RA:_____ **Nome:**_____

Experiência 3: ULA - Unidade Lógico-Aritmética e VHDL

Objetivo: Os objetivos desta experiência são: apresentar os princípios de funcionamento de uma Unidade Lógico-Aritmética, capaz de realizar operações aritméticas básicas com números inteiros com sinal em complemento de 2, esquematizada na Figura 1, bem como iniciar o estudo da linguagem VHDL e seu uso no projeto de circuitos lógicos.

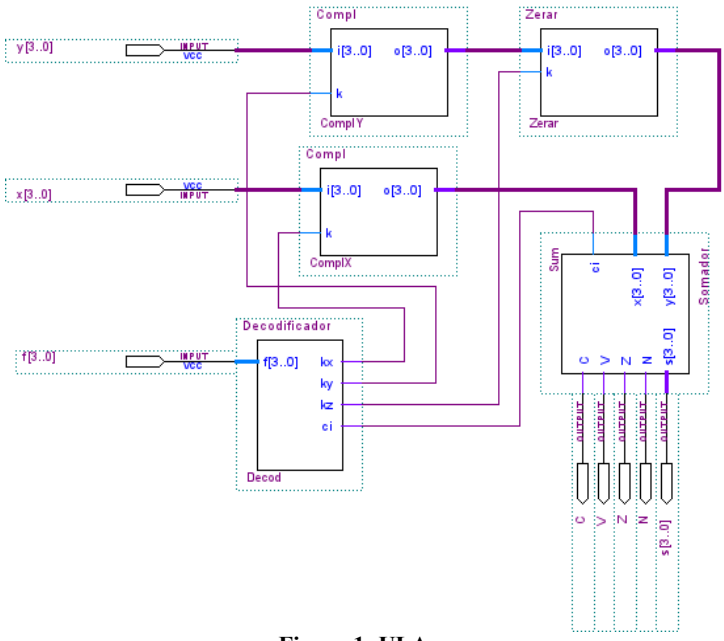

Figura 1: ULA.

O circuito suporta as quatro operações definidas na Tabela 1.

Tabela 1: Sinais de Controle.

Operação		Definição
ADD	adição	s = x + y
SUB	subtração	s = x - y
CMX	complemento	s = x'
CS	troca de sinal	s = -x

1. Preparo

Atividades Práticas:

(a) Analise o circuito da Figura 1 e entenda seu funcionamento.

(b) Procure na literatura um circuito que implemente o módulo "compl" (complemento simples de um número de 4 bits).

(c) Procure na literatura um circuito que implemente o módulo "zerar".

(d) Projete um circuito que implemente o módulo "somador". Inicialmente, procure na literatura um circuito que implemente um somador completo de 1 bit (com *carry* e *overflow*), e em seguida componha-o para gerar o somador de 4 bits.

(e) Projete 8 casos de teste para a ULA (operações da Tabela 1 com valores pré-definidos), de tal forma que os resultado das operações produzam os valores de bits de condição (N, Z, C, V) da tabela 2, onde N = Negativo (setado se o resultado da operação for um número negativo), Z = Zero (setado se o resultado da operação for zero), C = Carry (setado se a operação gerou um *carry*) e V = Overflow (setado se o resultado da operação gerou um *overflow*).

Tabela 2: Dados de teste.

1)	N=0; Z=0; C=1; V=0;	5)	N=1; Z=0; C=1; V=0;
2)	N=0; Z=0; C=1; V=1;	6)	N= 1; Z=0; C=0; V=0;
3)	N =0; Z=1; C=1; V=0;	7)	N= 1; Z=0; C= 0; V=1;
4)	N =0; Z=1; C=0; V=0.	8)	N= 0; Z=0; C=0; V=0.

(f) Projete um circuito combinacional que implemente o módulo "decodificador", de acordo com a tabela 3. Para isso, defina, para cada operação, os sinais kx, ky, kz e ci necessários para implementar cada uma das operações (observe que o sinal ci é necessário para implementar a subtração em complemento de dois).

Estudo Teórico: (Referência: Sistemas Digitais: Princípios e Aplicações, R. J. Tocci et al. 10a. edição)

(a) O que é um PLD e um FPGA? (Sec. 4.14)

(b) Descreva a diferença entre uma linguagem de programação e uma linguagem de descrição de *hardware* (Sec. 3.17).

(c) Descreva a estrutura básica de VHDL (Sec. 3.19).

(d) Explique a diferença entre SIGNAL e VARIABLE para declarar os sinais "intermediários". (Sec. 3.20).

(e) Sintetize os tipos de dados comuns em VHDL (Sec. 4.15).

(f) Cite a biblioteca que define os tipos de dados `std_logic` e que contém os blocos primitivos lógicos e componentes mais comuns.

(g) Mostre como uma tabela-verdade pode ser representada com uso de VHDL (Sec. 4.16).

(h) Quais são as estruturas de controle de decisão em VHDL? (Sec. 4.17)

(i) Como se descreva uma multiplexação em VHDL? (Sec. 9.18)

(j) Mostre como um componente/bloco primitivo pode ser reutilizado como uma "caixa preta" em VHDL (Sec. 5.26).

(k) Como se descreve *flip-flops* e *latches* em VHDL? (Sec. 5.25 e 5.26)

RA:	**Visto:**	**Data:**

2. Implementação dos Módulos Compl, Zerar e Somador

a) Capture os esquemáticos dos módulos "compl", "zerar" e "somador" desenvolvidos no preparo do experimento no Quartus II, encapsulando cada um deles como um módulo.

b) Teste cada um dos módulos de maneira independente, utilizando o simulador

RA:	**Visto:**	**Data:**

3. Decodificador

a) Cada operação mostrada na Tabela 1 é representada por um código binário de 3 *bits* $\underline{f} = (f_2\ f_1\ f_0)$, conhecido por **Código de Operação**. A Tabela 3 apresenta a correspondência entre as operações e os códigos de operação. Capture o esquemático do circuito decodificador projetado durante o preparo e simule o mesmo para todas as possíveis combinações de $\underline{f} = (f_2\ f_1\ f_0)$.

Tabela 3: Códigos de Operação.

Operação	Código de Operação ($f_2\ f_1\ f_0$)
ADD	Nº do grupo%8
SUB	(Nº do grupo%8+3)%8
CMY	(Nº do grupo%8+2)%8
CSX	(Nº do grupo%8+1)%8

RA:	**Visto:**	**Data:**

4. Implementação da ULA

a) Integre todos os módulos da ULA. Simule-a no ambiente Quartus II, testando-a com todos os casos de teste desenvolvidos durante o preparo do experimento.

RA:	Visto:	Data:

5. Implementação em VHDL

a) Implemente uma versão em VHDL para o módulo "compl". Utilize a estrutura de decisão IF/THEN/ELSE do VHDL para descrevê-lo . Chame-o de "compl-vhdl". Simule-o.

b) Implemente uma versão em VHDL para o o módulo "zerar". Utilize a estrutura de multiplexação do VHDL para descrevê-lo. Chame-o de "zerar-vhdl". Simule-o.

c) Utilizando a linguagem de especificação VHDL, o circuito de **somador completo** de 1 bit pode ser especificado na seguinte forma:

```
library ieee;
use ieee.std_logic_1164.all;
entity FULLADDER is
    port(
        a          : in bit;      -- porta de entrada;
        b          : in bit;
        cin        : in bit;      -- carry in
        s          : out bit;     -- soma
        cout   : out bit          -- carry out
    );
end FULLADDER;
architecture a of FULLADDER is
    begin
      s <= (a xor b) xor cin;    -- soma
     cout <= (a and b) or (cin and (a xor b));      -- carry
end a;
```

Utilize o módulo FULLADDER para gerar uma implementação em VHDL do módulo "somador". Chame-o de "somador-vhdl". Simule-o.

d) Construa uma versão em VHDL para o circuito do decodificador. Simule-o

e) Substitua os circuitos "compl", "zerar", "somador" e "decodificador" da ULA, por suas versões em VHDL. Repita o item 4 com as versões dos módulos em VHDL.

RA:	Visto:	Data:

6. Decodificador Hexadecimal

a) Sabendo que os sinais HEX0[0] a HEX0[6] da placa FPGA alimentam os segmentos do display HEX0 (e também HEX1, HEX2 e HEX3) conforme a figura 2 abaixo, utilize a linguagem VHDL para descrever um módulo decodificador que visualize os números 0 a 9, e A a F conforme a figura. Simule-o para todas as possíveis entradas e teste-o na placa FPGA.

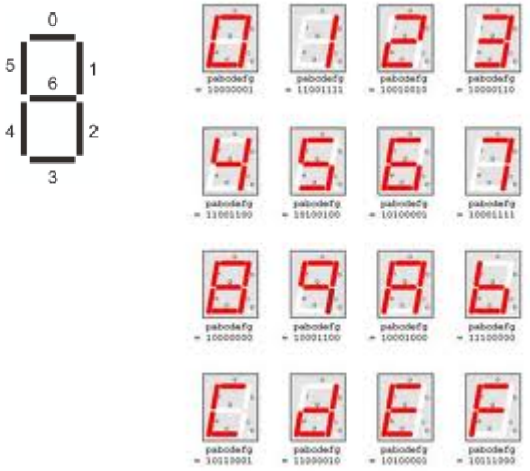

Figura 2 - Visualização de Números Hexadecimais no Display de 7 Segmentos

RA:	Visto:	Data:

7. Teste da ULA no FPGA

a) Utilizando as chaves e botões disponíveis na placa FPGA, implemente um circuito para a entrada de dados na ULA e a visualização das operações nos displays de 7 segmentos (números) e leds (bits de condição).

b) Programe a placa FPGA e teste o circuito com os testes desenvolvidos durante o preparo do experimento.

RA:	Visto:	Data: