

# CI1210 – Projetos Digitais e Microprocessadores

## Trabalho Semestral - Prof. Giovanni Venâncio

### Enunciado do Trabalho

O Mico X.1 é um processador simplificado que, a cada ciclo, busca e executa uma instrução de lógica/aritmética, ou uma instrução de controle de fluxo (salto/desvio). Sua tarefa é implementar o circuito de dados e de controle da versão X.1 do Mico. O conjunto de instruções do Mico X1 está definido na Tabela abaixo:

opcode	Instrução	Semântica	Comentário
0	nop	--	No operation
1,2	Op c, a, b	$R(c) \leftarrow R(a) \text{ op } R(b)$	op = {add, sub}
3,4,5	Op c, a, b	$R(c) \leftarrow R(a) \text{ op } R(b)$	op = {and, or, xor}
6	Not c, a	$R(c) \leftarrow \text{not } R(a)$	Complemento
7	sll	$R(c) \leftarrow R(a) \ll R(b)$	Deslocamento lógico esq. *
8	srl	$R(c) \leftarrow R(a) \gg R(b)$	Deslocamento lógico dir. *
9	ori	$R(c) \leftarrow R(a) \text{ OR extZero}(K)$	OR com constante lógica
a	xori	$R(c) \leftarrow R(a) \text{ XOR extZero}(K)$	XOR com constante lógica
b	addi	$R(c) \leftarrow R(a) + \text{extZero}(K)$	SOMA com constante lógica
c	Show a	Display R(a)	Exibe valor na saída
d	Jump E	$IP \leftarrow E$	Salto incondicional
e	Branch a, b, E	$IP \leftarrow ((R(a) == R(b)) ? E : IP + 1)$	Desvio condicional **
f	halt	$IP \leftarrow IP + 0$	Termina a simulação
* Completa com zeros. ** Você pode usar o subtrator da ULA e um fio de (ZERO?) para fazer a comparação. $E = IP + \text{SignExt}(\text{Const})$			

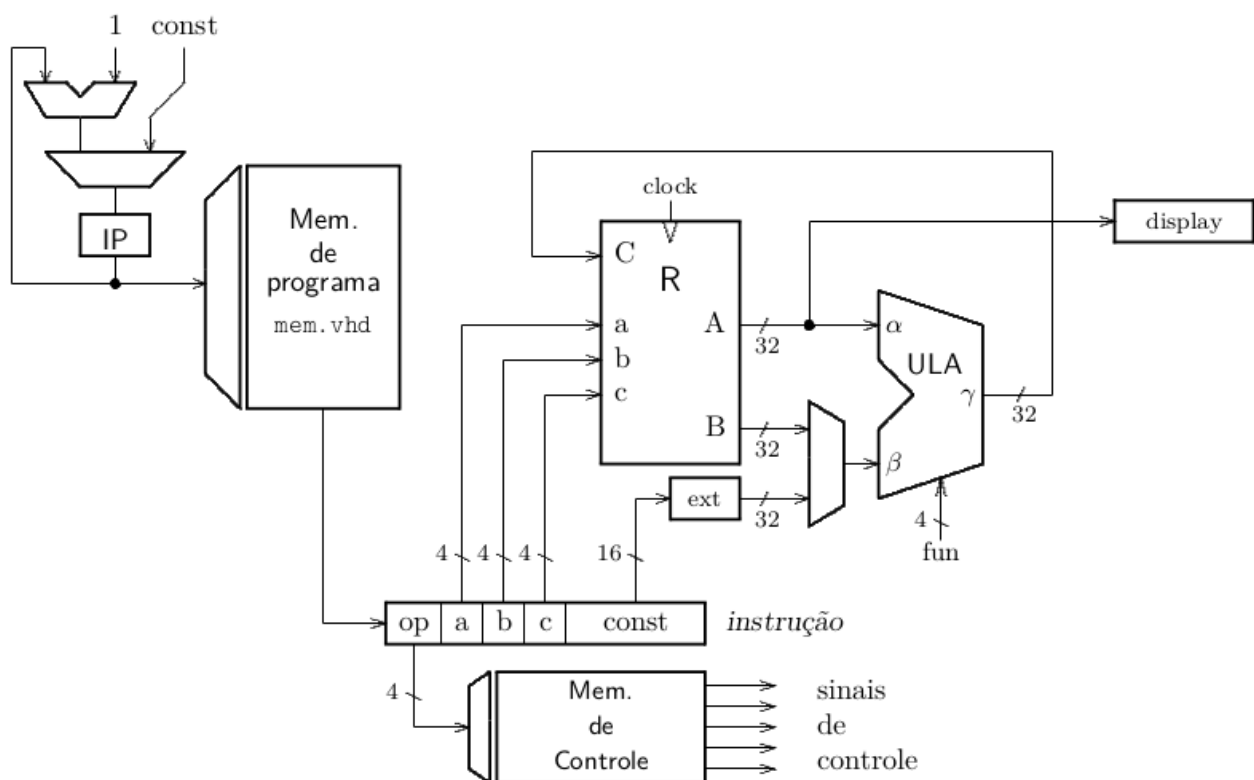
No diagrama e na Tabela 1, *a*, *b*, *c* são nomes de registradores (endereços dos registradores), enquanto que *A*, *B*, *C* são os conteúdos dos respectivos registradores (ex.:  $R(a) = A$ ). O registrador 0 contém a constante zero ( $R(0) = 0$ ); escritas neste registrador não tem nenhum efeito. A ULA executa as operações de lógica e aritmética definidas para as instruções. O bloco de registradores contém 16 registradores de 32 bits, duas portas de leitura (*A* e *B*) e uma porta de escrita (*C*).

Crie a unidade lógico/aritmética (ULA) do processador Mico X.1. A ULA deve ser capaz de processar as seguintes operações:

Seletor	Operação
---------	----------

0	Somador: A+B
1	Subtrator: A-B
2	AND: A and B
3	OR: A or B
4	XOR: A xor B
5	Not: !A
6	Deslocador esquerda A«B
7	Deslocador direita A»B

A ULA deverá ter duas entradas: A e B de 32 bits cada e um seletor. A saída da ULA será o resultado e um fio indica se houve saída igual a zero em todos os bits (Saída ZERO).



**Figura 1: Circuito de dados e de controle do Mico X.**

O diagrama de blocos do Mico X.1 é ilustrado na Figura 1. O circuito de dados consiste de uma Unidade de Lógica e Aritmética (ULA), de um Bloco de Registradores (R), e de circuitos auxiliares. O circuito de controle consiste de um apontador de instrução (registrador Program Counter - PC), um somador, e de um comparador de igualdade, que não é mostrado no diagrama.

Uma vez que o processador esteja implementado, você deve traduzir o programa em C especificado para assembly do Mico X.1. O resultado final computado deverá ser exibido no display.

**Você deverá implementar um programa teste para seu processador Mico X.1.** O programa de teste é o Fibonacci iterativo calculando N números da sequência de Fibonacci. N é uma constante que deve ser inicializada no próprio programa. O programa deverá funcionar normalmente mesmo se a constante N for alterada.

Seu programa deve exibir o operando N (com show), e exibir cada número da sequência com a instrução show. Ao final, o programa deverá encerrar com a instrução halt.

Para tanto você deve:

1. Escrever o programa iterativo em C;
2. Traduzir este programa para o assembly do Mico X.1;
3. Traduzir o assembly para binário e editar a memória de instruções.

No Moodle serão adicionados vídeos que irão auxiliar na implementação da memória de instruções.

## **Regras Gerais de Entrega e Apresentação**

O trabalho consiste de duas etapas:

**Etapla 1. (60 pontos)** Projeto do processador Mico X.1, contendo a implementação de todos os componentes descritos no diagrama;

**Etapla 2. (40 pontos)** Desenvolvimento do programa em Assembly, codificado em binário, que será executado no processador implementado na Etapla 1.

A implementação pode ser feita no simulador **Digital** ou **Logisim-Evolution**. É permitido utilizar os componentes pré-existentis:

- Portas lógicas: AND, OR, NOT, etc.
- Memórias: RAM, ROM, FF
- Plexers: MUX, DEMUX, ENCODER, DECODER

O trabalho é individual e a entrega deverá ser feita através do Moodle da disciplina.

A entrega consiste de um relatório em PDF de no máximo 2 páginas A4 (frente e verso) com fonte tamanho mínimo 11 pontos, contendo:

- Nome
- Breve discussão sobre decisões de implementação (inclua descrições de bugs, caso exista)
- O código C do programa implementado
- O código Assembly Mico X.1 gerado a partir da tradução do código C

Além do relatório, também deverá ser entregue no Moodle o arquivo de implementação do processador (.dig ou .circ).

A cópia do trabalho (plágio) não será tolerada e acarretará em nota igual a Zero para todos os envolvidos. É do interesse de todos que alunos conversem sobre o projeto, mas cada um deverá desenvolver seu próprio trabalho.

Os trabalhos deverão ser apresentados de forma oral pelo aluno. A nota irá considerar domínio do tema, robustez da solução e rigorosidade da metodologia.