Relatório Processador MICO XII

Juliana Zambon - GRR20224168
Projetos Digitais e Microprocessadores - CI1210
Universidade Federal do Paraná – UFPR
Curitiba, Brasil
julianazambon@ufpr.br

***Resumo*—Este relatório apresenta a demonstração da implementação do processador Mico XII utilizando o simulador Digital. Além disso, descreve a implementação do programa teste que executa as dezessete instruções do processador, incluindo os dois casos possíveis do *branch if equal*.**

A implementação do processador Mico XII foi realizada por meio do simulador Digital, que oferece as ferramentas necessárias para a construção e simulação de circuitos digitais. Utilizando esse ambiente, foi possível criar os componentes principais do processador, como o Bloco de Registradores e a Unidade Lógica Aritmética (ULA), e interligá-los de acordo com a arquitetura especificada pela bibliografia.

***Index Terms*—Microprocessador, bloco de registradores, instruções, circuitos, programa teste.**

I. Introdução

O presente relatório é resultado de um trabalho realizado no curso de Ciência da Computação da Universidade Federal do Paraná (UFPR) durante a disciplina de Projetos Digitais e Microprocessadores (CI1210), no segundo período. O objetivo deste relatório é apresentar uma abordagem para a implementação do processador Mico XII, juntamente com o desenvolvimento de um programa teste na linguagem assembly. O programa teste foi criado para executar o conjunto de dezessete instruções do processador. Além disso, o relatório aborda a conversão das instruções para formatos binário, formato de máquina e hexadecimal, a fim de permitir o carregamento do programa na memória de instruções ROM.

II. Implementação de Subcircuitos

O processador Mico XII é composto por quatro componentes principais que foram implementados com o auxílio de subcircuitos. O primeiro componente é o Bloco de Registradores, que consiste em dezesseis registradores de trinta e dois bits cada. É importante observar que o registrador zero (r0) contém a constante zero, o que significa que qualquer escrita nesse registrador não terá efeito. Por isso, na implementação, o valor constante zero de 32 bits foi utilizado em vez do registrador r0.

Além disso, o processador inclui uma Unidade Lógica Aritmética (ULA), que é responsável por realizar as operações lógicas e aritméticas necessárias para processar os dados. Essa unidade executa operações como adição, subtração, comparação e operações lógicas, como AND, OR e NOT. Os dados para essas operações são fornecidos à ULA pelos registradores de dados e pelos registradores de endereços, e o resultado das operações é armazenado novamente nos registradores de dados.

Para garantir que todas as operações lógicas e aritméticas fossem realizadas corretamente, foi implementada uma ordem específica do conjunto de instruções correspondente no subcircuito do processador. Esse subcircuito foi desenvolvido utilizando os componentes disponibilizados pelo simulador Digital.

Esses componentes, o Bloco de Registradores e a ULA, são cruciais para o funcionamento adequado do processador Micro XII. Eles desempenham papéis essenciais no processamento de dados e na execução das operações lógicas e aritméticas necessárias para o correto funcionamento do processador.

III. Implementação da Memória de Controle

Para a implementação da memória de controle, foram utilizados quatorze (14) bits de dados. A configuração e organização dessa memória foram realizadas por meio da construção de uma tabela, como ilustrado na Tabela 1, com o objetivo de obter o valor binário correspondente a cada configuração e, em seguida, traduzi-los para o formato hexadecimal.

Os quatorze bits de dados foram definidos levando em consideração os seguintes elementos: *halt*, *wb*, *imm*, *ula*, *str*, *ld*, *content*, *jumps*, *branch* e *show*. Cada um desses elementos contribuiu com um determinado número de bits. Por exemplo, a *ula* possui 3 bits de dados, enquanto o *content* e o *jumps* possuem 2 bits de dados. Os demais elementos possuem apenas um bit de dado.

Essa organização dos bits de dados na memória de controle é importante para que o processador possa interpretar corretamente as instruções e realizar as operações adequadas. Cada configuração binária representa uma combinação específica de elementos e, por meio desses bits, o processador é capaz de determinar as ações a serem executadas em cada ciclo de clock.

Portanto, a definição dos quatorze bits de dados na memória de controle, levando em conta a distribuição dos elementos mencionados, é fundamental para o correto funcionamento do processador. Essa estrutura permite que o processador interprete e execute as instruções corretamente, garantindo o fluxo adequado das operações.

A partir disso foi realizada a concatenação dos quatorze bit que foram convertidos de binário para hexadecimal, a fim

de que fossem armazenados na memória de controle (ROM), como poder ser observado na Tabela 1.

Tabela I
CONCATENAÇÃO DE BITS DA MEMÓRIA DE CONTROLE

OPCODE	BINÁRIO	HEXADECIMAL
0000	000011000000000000	3000
0001	000111000100000000	7100
0010	001011001000000000	b200
0011	001111001100000000	f300
0100	010011010000000000	13400
0101	010111010100000000	17500
0110	011011011000000000	1b600
0111	011111100000000000	1f800
1000	100011100000000000	23800
1001	100110100001010000	26850
1010	101010000010000000	2a080
1011	101110000000000001	2e000
1100	110011000000001000	33008
1101	110110000000000100	36004
1110	111010000100000010	e840
1111	111100000000000000	3c000

IV. IMPLEMENTAÇÃO DO PROCESSADOR MICO XII

A construção do processador foi realizada pela integração dos subcircuitos necessários para a execução das instruções, como citado anteriormente. Dessa forma, o processador é composto por diferentes componentes, incluindo um Bloco de Registradores, uma Unidade Lógica Aritmética, uma Memória de Dados e circuitos auxiliares.

O circuito de controle é responsável por coordenar as operações do processador. Ele é composto por vários elementos, como o ponteiro de instrução (IP), que determina a próxima instrução a ser executada. Um somador é utilizado para incrementar o valor do ponteiro de instrução, permitindo a progressão sequencial do programa.

Outro componente importante do circuito de controle é a Memória de Instruções. Essa memória armazena as instruções do programa em formato binário, permitindo que o processador as acesse de acordo com o valor do ponteiro de instrução.

Além disso, o circuito de controle inclui uma memória ROM, que contém os sinais de controle necessários para coordenar as operações do processador. Esses sinais são essenciais para garantir que as instruções sejam executadas corretamente e que os componentes do processador sejam acionados no momento adequado.

Em conjunto, esses componentes e circuitos auxiliares formam o processador, permitindo a execução das instruções e o processamento de dados. A integração cuidadosa desses elementos é fundamental para garantir o funcionamento adequado do processador Micro XII.

V. IMPLEMENTAÇÃO DE CASOS OMISSOS

O *branch if equal* (beq) é uma instrução de desvio condicional no Mico XII que permite que o fluxo de execução de um programa seja desviado para uma determinada localização de memória se uma condição de igualdade for verdadeira. Utilizando comparadores internos para verificar se dois valores em registradores específicos são iguais. Caso a igualdade seja confirmada, o desvio condicional é realizado e o próximo endereço de instrução a ser executado será definido como o endereço especificado pelo desvio.

Requer a especificação de dois registradores e um endereço de destino. Os valores nos registradores são comparados para igualdade e, se a igualdade for verdadeira, o fluxo de execução é desviado para o endereço de destino especificado.

VI. IMPLEMENTAÇÃO DO PROGRAMA TESTE

A implementação do programa teste que executa cada uma das dezessete instruções do microprocessador ocorreu por meio da criação de uma tabela contendo o código das instruções na linguagem assembly, juntamente com suas respectivas traduções para binário, formato de máquina e hexadecimal. Essas traduções permitiram que o arquivo das instruções fosse carregado na memória ROM, que é a memória de instruções do processador.

A ISA, ou Arquitetura de Conjunto de Instruções, desempenha um papel fundamental na interface entre o software e o hardware. Ela é composta pelo opcode (código de operação) que identifica cada instrução, além dos registradores R(a), R(b) e R(c), e uma constante. Esses elementos, conforme ilustrado na Tabela 4, representam as operações fundamentais que o programa pode executar no processador.

Na Tabela 2, é possível observar as instruções escritas em *assembly* que foram utilizadas no programa.

Tabela II
TABELA DO CÓDIGO ASSEMBLY

ASSEMBLY	LINE CODE
addi R1, R2,3	0
addi R2, R1, 1	1
add R3, R1, R2	2
show R3	3
sub R4, R3, R2	4
show R4	5
mul R5, R1,R4	6
show R5	7
and R6, R1,R3	8
show R6	9
or R7, R1,R3	10
show R7	11
xor R8, R1,R3	12
show R8	13
slt R9, R1, R3	14
show R9	15
not R10, R4	16
show R10	17
st R2, R9	18
ld R8, R2	19
show R8	20
beq R1, R2, 24	21
addi R1, R1, 1	22
j 21	23
addi R11, R11,25	24
beq R2, R3,	25
addi R2, R2, 1	26
jr R11	27
beq R6, R5,	28
addi R6, R6, 1	29
jal R12	30
show R12	31
halt	32

Tabela III
TABELA DA ARQUITETURA DE CONJUNTO DE INSTRUÇÕES

ISA				
OPCODE	**RA**	**RB**	**RC**	**CONSTANTE**
1000	0010	0000	0001	0000000000000011
1000	0001	0000	0010	0000000000000001
0000	0001	0010	0011	0000000000000000
1011	0011	0000	0000	0000000000000000
0001	0011	0010	0100	0000000000000000
1011	0100	0000	0000	0000000000000000
0010	0001	0100	0101	0000000000000000
1011	0101	0000	0000	0000000000000000
0011	0001	0011	0110	0000000000000000
1011	0110	0000	0000	0000000000000000
0100	0001	0011	0111	0000000000000000
1011	0111	0000	0000	0000000000000000
0101	0001	0011	1000	0000000000000000
1011	1000	0000	0000	0000000000000000
0110	0001	0011	1001	0000000000000000
1011	1001	0000	0000	0000000000000000
0111	0100	0000	1010	0000000000000000
1011	1010	0000	0000	0000000000000000
1010	0010	1001	0000	0000000000000000
1001	0010	0000	1000	0000000000000000
1011	1000	0000	0000	0000000000000000
1110	0001	0010	0000	0000000000011000
1000	0001	0000	0001	0000000000000001
1100	0000	0000	0000	0000000000010101
1000	1011	0000	1011	0000000000011001
1110	0010	0011	0000	0000000000011000
1000	0010	0000	0010	0000000000000001
1101	1011	0000	0000	0000000000000000
1110	0110	0101	0000	0000000000011111
1000	0110	0000	0110	0000000000000001
1100	1100	0000	0000	0000000000011100
1011	1100	0000	0000	0000000000000000
1111	0000	0000	0000	0000000000000000

Tabela IV
TABELA DA CONVERSÃO DE BINÁRIO PARA HEXADECIMAL DO ASSEMBLY

BINÁRIO	**HEXADECIMAL**
10000010000000010000000000000011	82010
10000001000000100000000000000001	81020
00000001001000110000000000000000	1230
10110011000000000000000000000000	b3000
00010011001001000000000000000000	13240
10110100000000000000000000000000	b4000
00100001010001010000000000000000	21450
10110101000000000000000000000000	b5000
00110001001101100000000000000000	31360
10110110000000000000000000000000	b6000
01000001001101110000000000000000	41370
10110111000000000000000000000000	91370
01010001001110000000000000000000	51380
10111000000000000000000000000000	b8000
01100001001110010000000000000000	61390
10111001000000000000000000000000	b9000
01110100000010100000000000000000	740a0
10111010000000000000000000000000	ba000
10100010100100000000000000000000	a2900
10010010000010000000000000000000	92080
10111000000000000000000000000000	b8000
11100001001000000000000000011000	e1200
10000001000000010000000000000001	81010
11000000000000000000000000010101	c0000
10001011000010110000000000011001	8b0b0
11100010001100000000000000011000	e2300
10000010000000100000000000000001	82020
11011011000000000000000000000000	db000
11100110010100000000000000011111	e6500
10000110000001100000000000000001	86060
11001100000000000000000000011100	cc000
10111100000000000000000000000000	bc000
11110000000000000000000000000000	f0000

VII. CONCLUSÃO

Com base na implementação do processador Mico XII, torna-se evidente como os componentes são interligados para implementar as instruções de forma eficiente e precisa. O processador Mico XII é composto por várias unidades funcionais, incluindo a Unidade de Controle, a Unidade Lógica e Aritmética (ULA), o Bloco de Registradores, o Registrador de Endereços, a Memória de Acesso Aleatório (RAM) e a Memória de Controle (ROM).

A Unidade de Controle desempenha um papel crucial na coordenação de todas as operações do processador. Ela interpreta as instruções e emite os sinais de controle adequados para cada componente, garantindo que a sequência correta de operações seja executada. Essa unidade é responsável por buscar instruções da memória, decodificá-las e controlar o fluxo de dados entre os registradores e a ULA.

A memória RAM desempenha um papel fundamental no armazenamento e recuperação de dados. Ela armazena instruções e dados utilizados pelo processador, permitindo que sejam acessados quando necessário.

REFERÊNCIAS

Relatório feito em LaTeX utilizando *Overleaf*.

HEXSEL, ROBERTO A. **Sistemas Digitais e Microprocessadores**. Editora UFPR, 2012.