



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Programa de Graduação em Sistemas de Informação

Bryan Diniz Rodrigues
Luiz Henrique Gomes Guimarães
Maria Luiza Moura Rocha
Thais Barcelos Lorentz

**RELATÓRIO DO TRABALHO UNIDADE LÓGICA E ARITMÉTICA (ULA) DE
PROGRAMAÇÃO ORIENTADA POR OBJETOS E ARQUITETURA DE
COMPUTADORES**

Belo Horizonte
2019/2

Bryan Diniz Rodrigues
Luiz Henrique Gomes Guimarães
Maria Luiza Moura Rocha
Thais Barcelos Lorentz

**TRABALHO UNIDADE LÓGICA E ARITMÉTICA (ULA) 32 BITS DE
PROGRAMAÇÃO ORIENTADA POR OBJETOS E ARQUITETURA DE
COMPUTADORES**

Trabalho apresentado à disciplina Programação
Orientada a Objetos e arquitetura de computadores,
do curso de Sistemas de Informação da Pontifícia
Universidade Católica de Minas Gerais.

Orientadores: Prof. Paulo Cesar do Amaral Pereira e
Prof. Claudiney Vander Ramos.

Belo Horizonte
2019

SUMÁRIO

SUMÁRIO	3
RESUMO	4
1.INTRODUÇÃO	4
1.1 OBJETIVOS	4
2.4 NÚMEROS DE PONTO FLUTUANTE	7
2.4.1 A NORMA IEEE 754	7
2.4.2 MANTISSA	7
2.4.3 EXPOENTE	7
2.5. NORMALIZAÇÃO DO PONTO FLUTUANTE	8
2.6. OPERAÇÕES ARITMÉTICAS	8
2.6.1. ADIÇÃO E SUBTRAÇÃO	8
2.6.2. MULTIPLICAÇÃO E DIVISÃO	8
2.7 REPRESENTAÇÃO DE INTEIROS	9
2.8 PORTAS E ÁLGEBRA BOOLEANA	9
2.9 EXPLICAÇÃO DA PORTA LÓGICA AND	9
3.0 EXPLICAÇÃO DA PORTA LÓGICA OR	10
3.1 EXPLICAÇÃO DA PORTA LÓGICA NOT	10
3.2 EXPLICAÇÃO DA PORTA LÓGICA XOR	10
3.3 EXPLICAÇÃO DA CLASSE SOMADOR	10
3.4 EXPLICAÇÃO DA CLASSE SOMADOR 32BIT	10
3.5 EXPLICAÇÃO DA CLASSE COMPLEMENTO DE 2	10
5. CONCLUSÃO	11

RESUMO

O principal objetivo do trabalho foi desenvolver um programa que convertesse números inteiros em binários, para depois obter a soma, subtração e divisão destes respectivos números convertidos. Um sistema para representar números nos quais a faixa de números que podem ser expressos é independente do número de dígitos significativos. Dentre outras coisas que incrementamos no projeto.

Palavras-chave: Orientação a Objetos e Arquitetura de Computadores. Desenvolver um sistema de Unidade Lógica Aritmética. Aplicação prática da programação orientada a objeto e conhecimentos específicos da disciplina de Arquitetura.

1.INTRODUÇÃO

A Unidade Lógica e Aritmética (ULA), trabalho que integra as disciplinas de Arquitetura de Computadores e Programação Orientada a Objetos, propõe a implementação do código de um dos componentes da Unidade Central de Processamento de um computador, sendo esta, a responsável por realizar operações lógicas e aritméticas do computador.

Veremos a seguir, a implementação de cada uma das classes que tornam possível o funcionamento da ULA, sendo estas, as classes de portas lógicas básicas e portas derivadas destas, conforme a necessidade. Além disso, veremos classes de unidade lógica, somador e decodificador, que com o auxílio das portas lógicas, implementam uma ULA de 1 bit.

Além disso, temos a implementação de mais três ULAs, sendo elas de 8, 27 e 32 bits, sendo esta última destinada a números de ponto flutuante.

1.1 OBJETIVOS

Este trabalho tem como objetivos:

- Aprimorar conhecimentos de arquitetura de computadores.

- Utilizar conhecimentos de Programação Orientada a Objetos (POO) e Algoritmos e Lógica de Programação (ATP) para realizar uma ULA com ponto flutuante, mantissa, etc.
- Utilização de portas lógicas.
- Aprimoração do uso de Windows Forms, e outros conhecimentos passados pelos respectivos professores das disciplinas.

1.2 ORGANIZAÇÃO DO TRABALHO

O principal objetivo do trabalho foi relacionar os conteúdos trabalhados nas disciplinas de Programação Orientada a Objetos e Arquitetura de Computadores para a criação de um programa correlacionado a Unidade Lógica e Aritmética (ULA), onde precisaria de criar no sistema uma ULA de 32 bits. No qual, deveria representar números nos quais a faixa de números que podem ser expressos é independente do número de dígitos significativos. Contendo também a soma, subtração, multiplicação e divisão, convertendo e separando a mantissa e o sinal. Dentre outras coisas que foram especificadas pelos professores em sala de aula.

2. REFERENCIAL TEÓRICO

Neste capítulo são apresentados conceitos básicos sobre os conteúdos utilizados neste trabalho, como portas lógicas, operações aritméticas realizadas pelo computador, números de ponto flutuante, entre outras pesquisas necessárias para a compreensão deste sistema de ULA de 32 BITS.

2.1. O QUE É UMA ULA?

A ULA ou Unidade Lógica e Aritmética, é uma das peças fundamentais da CPU/UCP (unidade central de processamento). Ele é um dispositivo que realiza operações lógicas e aritméticas sobre números representados em circuitos lógicos. Ela executa as principais operações lógicas de um computador, tais como a soma, subtração, divisão e multiplicação, além de determinar se um número é positivo ou negativo ou se é zero. Além de executar funções aritméticas, a ULA deve ser capaz de determinar se uma quantidade é menor ou maior que outra e quando quantidades são iguais. Ela pode executar funções lógicas com letras e com números.

As operações simples de uma ULA consistem em:

- Operações aritméticas com inteiros.
- Operações lógicas bit a bit AND, NOT, OR, XOR.
- Operações de deslocamento de bits (deslocamento, rotação por um número específico de bits para esquerda ou direita, com ou sem sinal); deslocamentos podem ser interpretados como multiplicações ou divisões por 2.

2.2. ENTRADAS E SAÍDAS DA ULA



2.3. REPRESENTAÇÃO EM SINAL – MAGNITUDE

Existem várias convenções alternativas usadas para representar números inteiros negativos e também positivos, todas envolvem o tratamento do bit mais significativo (mais à esquerda) na palavra como um bit de sinal. Se o bit de sinal for 0, o número é positivo, se o bit de sinal for 1, o número é negativo.

+ 18	=	00010010	
- 18	=	10010010	(sinal-magnitude)

Existem diversas desvantagens na representação de sinal – magnitude. Uma é que adição e subtração exigem uma consideração dos números e de suas relativas magnitudes para executar a operação exigida. Outra desvantagem é que existem duas representações do **0**:

+0 ₁₀	=	00000000	
-0 ₁₀	=	10000000	(sinal-magnitude)

Isso é inconveniente porque é ligeiramente mais difícil de testar se um valor é igual a 0 (uma operação realizada frequentemente nos computadores) do que se houvesse uma única representação. Por causa dessas desvantagens, a representação sinal -magnitude raramente é usada na implementação da parte inteira da ALU. Em vez disso, o esquema mais comum é a representação de complemento a dois.

2.4 NÚMEROS DE PONTO FLUTUANTE

2.4.1 A NORMA IEEE 754

A norma 754 do Instituto de Engenheiro Eletricistas e Eletrônicos (IEEE) define padrões que precisam ser seguidos pelos desenvolvedores de software e fabricantes de computadores no que se trata da aritmética binária de ponto flutuante. Esta norma traz como consequência uma portabilidade maior de softwares e hardwares desenvolvidos sobre esta norma, evitando assim problemas de mal condicionamento destes. A norma traz designações sobre o armazenamento, underflow e overflow, métodos de arredondamento e, principalmente a aplicação e realização de operações com esta aritmética binária. Este padrão já está ativa desde o ano de 1985, passou por algumas mudanças, porém mantém o mesmo objetivo: evitar a o formato de representação diferente entre os fabricantes de computadores e derivados.

2.4.2 MANTISSA

O significado, coeficiente ou mantissa, é a porção de dígitos de um sistema de vírgula flutuante que contém os dígitos significativos. Para “achar” a mantissa é necessário mover a vírgula e ficar entre o 1 e o 2. Por exemplo, o número 123,45 pode ser representado como um número em ponto flutuante decimal com significando inteiro 12345 e expoente -2. O seu valor é dado pela fórmula: 12345×10^{-2} O mesmo acontece com o número em binário.

2.4.3 EXPOENTE

O expoente da mantissa deve ser ajustado de acordo com as normas IEEE, sendo somado ao número padronizado 127. A seguir o número deve ser convertido para binário e assim poderá representar o expoente.

2.5. NORMALIZAÇÃO DO PONTO FLUTUANTE

Para a normalização do ponto flutuante em binário devemos “mover” a vírgula até que o primeiro 1. Isso significa que devemos move-lá para a esquerda ou para a direita até que ela fique depois do primeiro bit “1” da mantissa. Se a vírgula é deslocada para a esquerda devemos somar o expoente, e se ela for deslocada para a direita devemos subtrair o expoente.

Valores decimais de x entre $0 < x < 1$ fazem com que a vírgula seja movida para a direita no número binário, enquanto valores fora dessa faixa fazem com que ela seja movida para a esquerda.

Tanto os operandos quanto o resultado devem ser normalizados.

2.6. OPERAÇÕES ARITMÉTICAS

2.6.1. ADIÇÃO E SUBTRAÇÃO

Para se somar ou subtrair dois números representados em Ponto Flutuante no Padrão IEEE 754 Normalizado, seguiremos as mesmas regras do Padrão IBM, tomando o cuidado de lembrar que, na Norma IEEE 754, a mantissa é formada pela parte fracionária com o número 1 antes da vírgula e que, tanto a mantissa quanto o expoente tem quantidade fixa de bits para representá-los.

2.6.2. MULTIPLICAÇÃO E DIVISÃO

Para se multiplicar ou dividir dois números representados em Ponto Flutuante no Padrão IEEE 754 Normalizado, seguiremos as mesmas do Padrão IBM, tomando o cuidado de lembrar que, na Norma IEEE 754, a mantissa é formada pela parte fracionária com o número 1 antes da vírgula e que, tanto a mantissa quanto o expoente tem quantidade fixa de bits para representá-los.

Para encontrarmos o expoente resultante de uma divisão, devemos subtrair a potência do segundo número (p_2) da potência do primeiro número (p_1) e somar a potência resultante da razão das mantissas (p_m), quando houver.

2.7 REPRESENTAÇÃO DE INTEIROS

No sistema numérico binário, 1 número quaisquer podem ser representados apenas com os dígitos zero e um, o sinal de menos e a vírgula, ou vírgula fracionada.

$$-1101,0101_2 = -13,3125_{10}$$

Para as finalidades de armazenamento e processamento no computador, porém, não temos o benefício dos sinais de menos e vírgulas. Somente dígitos binários (0 e 1) podem ser usados para representar os números. Se estivermos limitados a inteiros não negativos, a representação é direta.

Uma palavra de 8 bits pode representar os números de 0 a 255, incluindo

$$00000000 = 0$$

$$00000001 = 1$$

$$00101001 = 41$$

$$10000000 = 128$$

$$11111111 = 255$$

Em geral, se uma sequência de n bits de dígitos binários $a_{n-1} a_{n-2} \dots a_1 a_0$ for interpretada como um inteiro sem sinal A , seu valor é:

$$A = \sum_{i=0}^{n-1} 2^i a_i$$

2.8 PORTAS E ÁLGEBRA BOOLEANA

Circuitos digitais podem ser construídos com um pequeno número de elementos primitivos combinando-os de inúmeras maneiras.

2.9 EXPLICAÇÃO DA PORTA LÓGICA AND

Esta classe, possui dois métodos, que recebem por parâmetro dois ou quatro valores, sendo eles falsos ou verdadeiros, e realiza uma operação lógica (and) sobre eles e retorna um valor lógico verdadeiro quando os dois (ou quatro) valores são verdadeiros.

3.0 EXPLICAÇÃO DA PORTA LÓGICA OR

Esta classe, possui três métodos, que recebem por parâmetro dois, quatro ou oito valores, e realizam uma operação lógica (or) sobre eles, retornando verdadeiro quando pelo menos um dos valores é verdadeiro.

3.1 EXPLICAÇÃO DA PORTA LÓGICA NOT

Esta classe, possui um método que recebe um valor por parâmetro, e retorna um valor oposto a esse.

3.2 EXPLICAÇÃO DA PORTA LÓGICA XOR

Esta classe, possui um método que recebe por parâmetro dois valores, realiza uma operação lógica (and) sobre o valor da operação lógica (not) do primeiro número e o segundo número original, então, realiza-se uma operação (and) sobre o valor da operação lógica (not) do segundo número e o primeiro número original, finalmente, retorna-se a operação lógica (or) das duas operações feitas acima.

3.3 EXPLICAÇÃO DA CLASSE SOMADOR

Essa classe, tem como objetivo, auxiliar na implementação da unidade lógica e aritmética de 1 bit, ela é a responsável por somar os valores de dois operandos recebidos por parâmetro.

3.4 EXPLICAÇÃO DA CLASSE SOMADOR 32BIT

Essa classe, utilizando do somador de 1 bit repetidas vezes faz a soma de uma mantissa de 23 bits +1 implícito, com um tratamento de overflow. É usado um vetor com 25 posições justamente pela possibilidade de overflow, caso haja o overflow o bit mais significativo não seja perdido.

3.5 EXPLICAÇÃO DA CLASSE COMPLEMENTO DE 2

Essa classe é semelhante ao somador só que especifica para fazer o complemento de dois na subtração.

5. CONCLUSÃO

Esse trabalho foi fundamental para aprimorarmos tanto os conhecimentos de Programação Orientada a Objetos quanto na disciplina de Arquitetura de Computadores. Concluimos que esse trabalho nos ajudou em ambas as matérias, pois ao mesmo tempo que utilizamos programação e Windows Forms para fazer a representação da ULA, tivemos que juntar conhecimentos teóricos para sabermos como as contas e as conversões necessárias são realizadas. Isso facilitou com que desenvolvêssemos conhecimento para sabermos o que está por trás das operações que são feitas pelo computador, e nos ajudou também na questão de conversões de bases, já que trabalhamos com decimais, binários e números hexadecimais. Concluimos portanto, que este trabalho foi muito complexo, mas muito produtivo.