

Wilson Cazarré Sousa

**Desenvolvimento e implementação de um
processador compatível com a Arquitetura 6502
em FPGA**

São José dos Campos - Brasil

Abril de 2024

Wilson Cazarré Sousa

Desenvolvimento e implementação de um processador compatível com a Arquitetura 6502 em FPGA

Relatório apresentado à Universidade Federal de São Paulo como parte dos requisitos para aprovação na disciplina de Laboratório de Sistemas Computacionais: Arquitetura e Organização de Computadores.

Docente: Prof. Dr. Tiago de Oliveira

Universidade Federal de São Paulo - UNIFESP

Instituto de Ciência e Tecnologia - Campus São José dos Campos

São José dos Campos - Brasil

Abril de 2024

Resumo

Palavras-chaves: 6502. FPGA. RISC. CISC.

Lista de ilustrações

Figura 1 – Endereçamento imediato	13
Figura 2 – Endereçamento absoluto. Note que o primeiro byte na memória é o menos significativo	14
Figura 3 – Endereçamento absoluto - Deslocado em X (ou Y)	14
Figura 4 – Endereçamento <i>Zero-Page</i>	15
Figura 5 – Endereçamento relativo	16
Figura 6 – Endereçamento indireto	17
Figura 7 – Datapath	20

Lista de tabelas

Tabela 1 – Tamanho da instrução por modo de endereçamento	18
Tabela 2 – Instruções de transferência	19

Lista de Abreviações

BD Barramento de Dados. [9](#)

BE Barramento de Endereços. [9](#)

Sumário

1	INTRODUÇÃO	9
1.1	Motivação	9
1.2	Metodologia	9
1.3	Organização do relatório	9
1.4	Objetivos	9
2	FUNDAMENTAÇÃO TEÓRICA	11
2.1	Visão geral de um sistema computacional	11
2.1.1	Arquitetura de von Neumann	11
2.1.2	Interfaces de entrada e saída	11
2.1.3	Microprocessadores	11
2.2	Microprocessador 6502	11
2.2.1	Arquitetura Original	11
2.2.2	Registrador de Status (SR)	11
2.2.3	Contador de Programa (PC)	12
2.2.4	<i>Stack Pointer</i> (SP)	12
2.2.5	Modos de endereçamento	12
2.2.5.1	Endereçamento imediato	12
2.2.5.2	Endereçamento absoluto	13
2.2.5.3	Endereçamento absoluto - Deslocado em X (ou Y)	13
2.2.5.4	Endereçamento <i>Zero-Page</i>	15
2.2.5.5	Endereçamento relativo	15
2.2.5.6	Endereçamento indireto	16
2.2.6	Endereçamento <i>Zero-Page</i> , deslocado em X (ou Y)	17
2.3	RTL - <i>Register-Transfer Logic</i>	17
2.4	FPGA - Field Programmable Array	18
2.5	Verilog Sintetizável	18
2.6	Completude de Turing	18
3	DESENVOLVIMENTO	19
3.1	Conjunto de instruções	19
3.2	Unidades funcionais	20
3.2.1	Registradores	20
3.2.2	Unidade de Controle	21
4	RESULTADOS OBTIDOS E DISCUSSÕES	23

5	CONSIDERAÇÕES FINAIS	25
	REFERÊNCIAS	27
	APÊNDICES	29

1 Introdução

Durante a disciplina de Laboratório de Sistemas Computacionais: Arquitetura e Organização de Computadores ofertada, é proposto que os discentes escolham um arquitetura para realizar sua implementação em um dispositivo FPGA. Esse relatório irá apresentar a fundamentação, bem como todo o processo de desenvolvimento de um sistema computacional baseado no microprocessador 6502.

1.1 Motivação

mandaram eu fazer esse trabalho ou eu seria reprovado na matéria

1.2 Metodologia

ai eu peguei lá o negócio e n sei oq lá ai eu fiz as coisas com meus manos e no final ficou tudo chave

1.3 Organização do relatório

ao leitor: basta ler

1.4 Objetivos

Geral

Descrever o objetivo geral do projeto

Específico

Descrever com detalhes os objetivos da etapa atual.

2 Fundamentação Teórica

2.1 Visão geral de um sistema computacional

2.1.1 Arquitetura de von Neumann

2.1.2 Interfaces de entrada e saída

2.1.3 Microprocessadores

2.2 Microprocessador 6502

O microprocessador 6502 é o segundo membro da família MCS650X. Essa família de microprocessadores de 8 bits foi lançada em 1975 pela *MOS Technology*. Os processadores dessa família apresentam o mesmo conjunto de instruções e modos de endereçamento, com pequenas diferenças em recursos e sua utilização. Por conta de sua eficácia e baixo custo, o microprocessador se popularizou rapidamente ao ser usado em diversos sistemas da época como *O Nintendo Entertainment System (NES)*, *Apple II*, *Commodore 64* e muitos outros.

2.2.1 Arquitetura Original

O microprocessador conta com um [Barramento de Dados \(BD\)](#) de 8 bits e um [Barramento de Endereços \(BE\)](#) de 16-bits. Qualquer operação que o processador precisa executar normalmente é iniciada colocando o endereço de acesso no [BE](#) e posteriormente lendo (ou escrevendo) um valor de 8-bits no [BD](#).

Internamente, 3 registradores de propósito geral podem ser usados.

- **Acumulador (A):** Usado também para armazenar o resultado das operações lógicas e aritméticas;
- **Index X e Y:** Ambos os registradores podem ser usados para operações com modos de endereçamento especiais, que serão abordados mais a frente no relatório.

Além dos 3 registradores que podem ser acessados diretamente, o 6502 também possui alguns registradores usados por funções específicas do processador.

2.2.2 Registrador de Status (SR)

O **registrador de status** é responsável por armazenar *flags* usadas para o controle do fluxo de programa do processador. Elas normalmente são atualizadas durante operações

lógicas, aritméticas e de transferência de dados.

- **Carry (C)**: Indica se a operação gerou um *carry*;
- **Negativo (N)**: Indica se a operação gerou um valor com o bit mais significativo ativo;
- **Overflow (V)**: Indica se a operação gerou um...;
- **Zero (Z)**: Indica se a operação gerou o valor zero;
- **Decimal (D)**: Indica se o processador está em modo aritmético decimal BCD;
- **Bloqueio de interrupções (I)**: Indica se o processador está ignorando as requisições de interrupções;
- **Break (B)**: Indica se a interrupção atual foi disparada via *software* pela instrução BRK, ao invés de uma interrupção via *hardware*.

2.2.3 Contador de Programa (PC)

O único registrador de 16-bits definido pela arquitetura. Esse registrador é responsável por manter o endereço de memória atualmente acessado pelo processador.

2.2.4 Stack Pointer (SP)

O *stack* é uma região de memória destinada para rápido acesso e escrita. A eficácia nessas operações vem do fato de que o processador utiliza o endereço no SP para saber exatamente onde a próxima leitura e escrita vai ocorrer. O registrador é incrementado ou decrementado de acordo após cada operação. O 6502 também utiliza o *stack* para armazenar os endereços de retorno quando subrotinas ou interrupções são executadas.

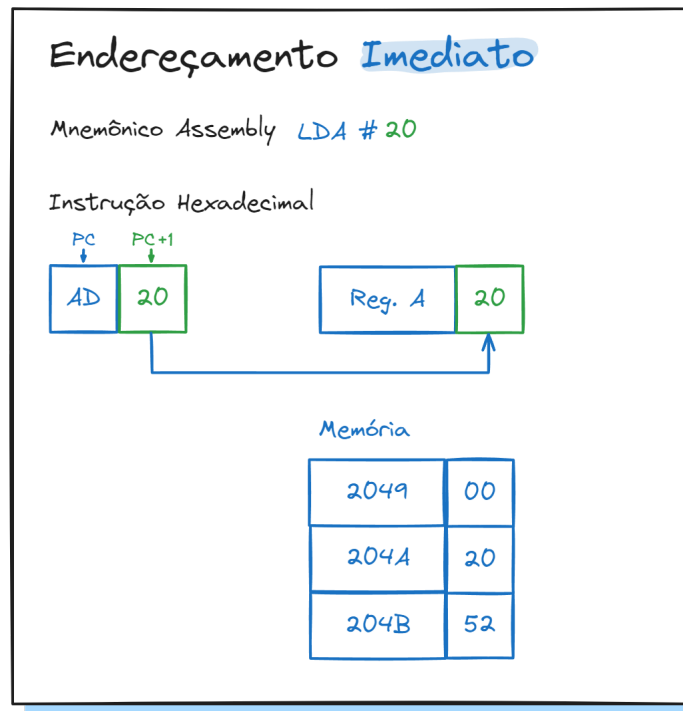
2.2.5 Modos de endereçamento

O 6502 é capaz de endereçar 65536 bytes de memória. Qualquer operação ou estrutura de dados dentro do processador compartilham esse mesmo espaço de memória. O processador também providencia 13 diferentes métodos de calcular o endereço efetivo de memória na qual a operação vai ser executada. Na computação, chamamos esses métodos de **modos de endereçamento** e aqueles disponíveis no 6502 serão descritos aqui.

2.2.5.1 Endereçamento imediato

Nesse tipo de instrução o operando é usado imediatamente após a instrução ter sido lida. Nenhum acesso a memória ou cálculo é realizado (Figura 1). Essa tipo de instrução utiliza 2 bytes de memória.

Figura 1 – Endereçamento imediato



Fonte: Autoria própria

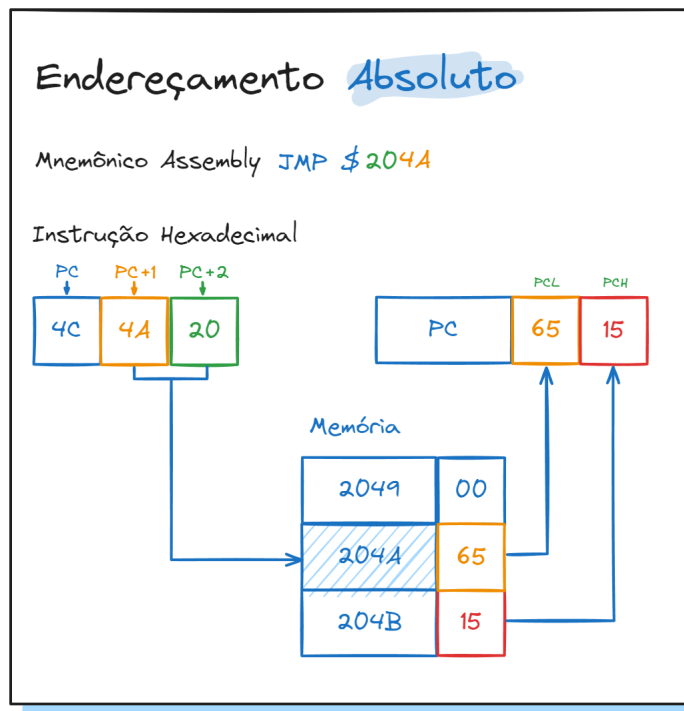
2.2.5.2 Endereçamento absoluto

Nesse tipo de instrução dois bytes são passados além do opcode. O processador usa esses bytes como um endereço de acesso a memória (Figura 2).

2.2.5.3 Endereçamento absoluto - Deslocado em X (ou Y)

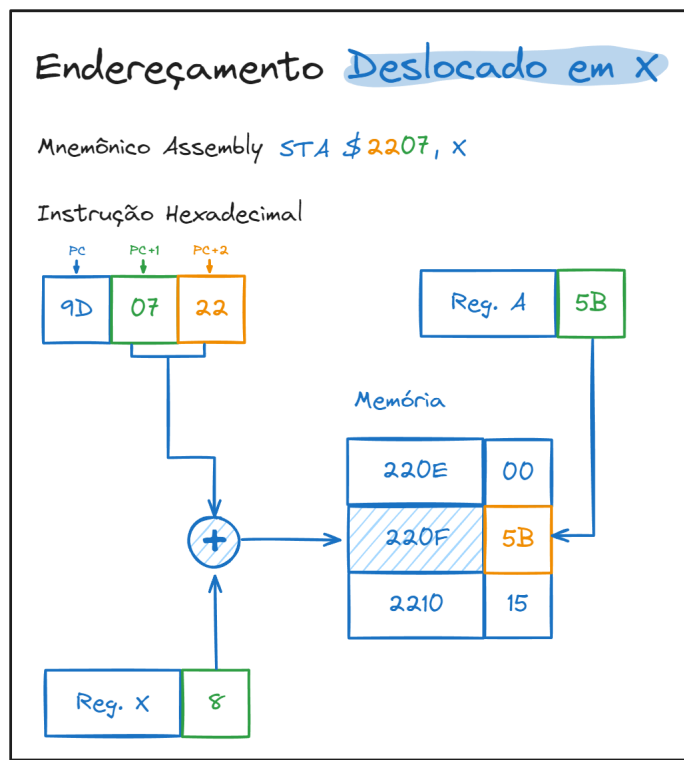
Esse modo é uma variação do endereçamento absoluto: dois bytes são buscados da memória e usados como endereço de acesso. A diferença está no fato de que o valor do registrador (X ou Y) é somado ao endereço de acesso. (Figura 3).

Figura 2 – Endereçamento absoluto. Note que o primeiro byte na memória é o menos significativo



Fonte: Autoria própria

Figura 3 – Endereçamento absoluto - Deslocado em X (ou Y)

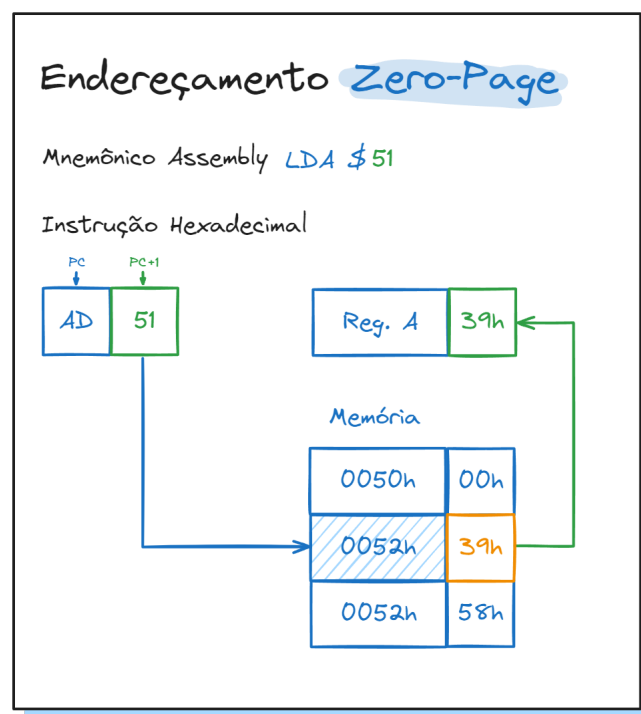


Fonte: Autoria própria

2.2.5.4 Endereçamento Zero-Page

Idêntico ao endereçamento absoluto, exceto que apenas um byte é lido da memória (o byte menos significativo). O byte mais significativo é inferido como 0 [Figura 4](#). Logo esse modo de endereçamento sempre retorna um dado localizado na primeira "página" da memória (os primeiros 256 bytes).

Figura 4 – Endereçamento Zero-Page

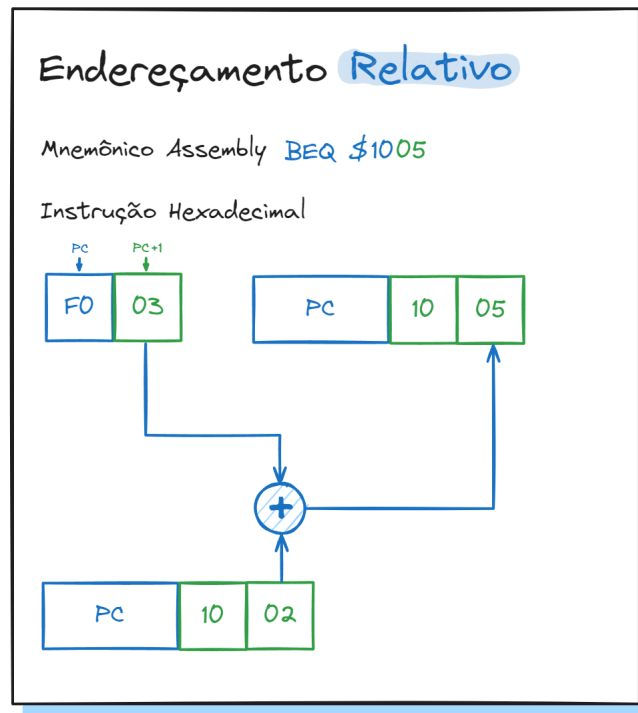


Fonte: Autoria própria

2.2.5.5 Endereçamento relativo

esse tbm

Figura 5 – Endereçamento relativo

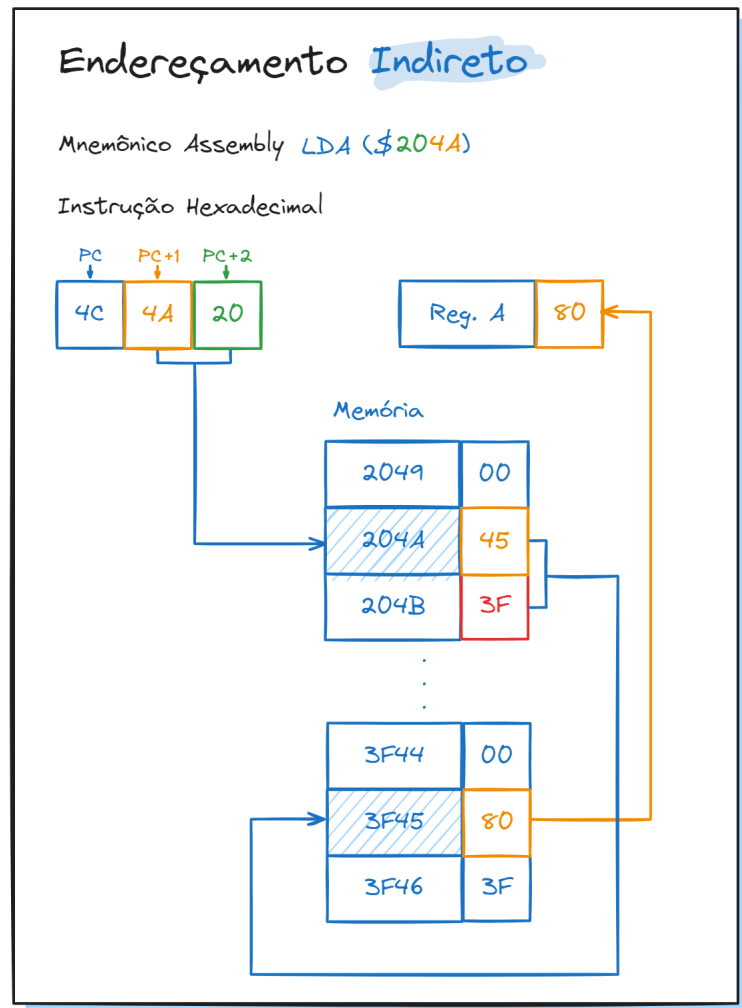


Fonte: Autoria própria

2.2.5.6 Endereçamento indireto

Figura 6

Figura 6 – Endereçamento indireto



Fonte: Autoria própria

2.2.6 Endereçamento Zero-Page, deslocado em X (ou Y)

Idêntico ao Endereçamento Absoluto deslocado em X (ou Y) com a diferença que o endereço de acesso está sempre dentro da primeira

2.3 RTL - *Register-Transfer Logic*

Quando tratamos do design de circuitos digitais complexos, é comum abstrairmos diferentes níveis do design com a intenção de tornar esses problemas mais simples de serem resolvidos.

3 níveis diferentes de abstração são definidos por Vahid (2011) na construção de circuitos digitais:

1. **Transistor Level:** Conectar transistores para construir componentes lógicos.

2. **Logic Level:** Utilizar-se de Portas Lógicas como bloco principal de construção para desenvolver circuitos combinacionais.
3. **Register-transfer Level:** Conectar uma rede de registradores e construir blocos que definem a lógica de transferência de estado entre esses registradores.

De maneira geral, no *Register-Transfer Level Design* (ou Design RTL) cada bloco do design deve desempenhar uma (e apenas uma) de duas possíveis funções:

1. **Lógica Combinacional:** São os blocos responsáveis pela computação do próximo estado. De maneira geral, esses blocos devem ser determinísticos e sempre apresentar a mesma saída para uma determinada entrada.
2. **Lógica Sequencial:** São blocos responsáveis por guardar e propagar o estado computado pelos blocos combinacionais de maneira síncrona.

Tabela 1 – Tamanho da instrução por modo de endereçamento

Modo de endereçamento	Tamanho em bytes
Acumulador (A)	1
Absoluto (abs)	3
Absoluto, deslocado em X (abs, x)	3
Absoluto, deslocado em Y (abs, y)	3
Imediato (#)	2
Implícito (impl)	1
Indireto (ind)	3
Indireto, deslocado em X (X, ind)	2
Indireto, deslocado em Y (ind, Y)	2
Relativo (rel)	2
Zero-Page (zpg)	2
Zero-Page, deslocado em X (zpg, x)	2
Zero-Page, deslocado em Y (zpg, y)	2

Fonte: Autoria Própria

2.4 FPGA - Field Programmable Array

2.5 Verilog Sintetizável

2.6 Completude de Turing

3 Desenvolvimento

O desenvolvimento da CPU se deu em algumas etapas. Primeiramente o conjunto de instrução foi definido como um subconjunto da família MCS650X original. Depois disso foram escolhidos alguns modos de endereçamento e um *datapath* foi definido.

3.1 Conjunto de instruções

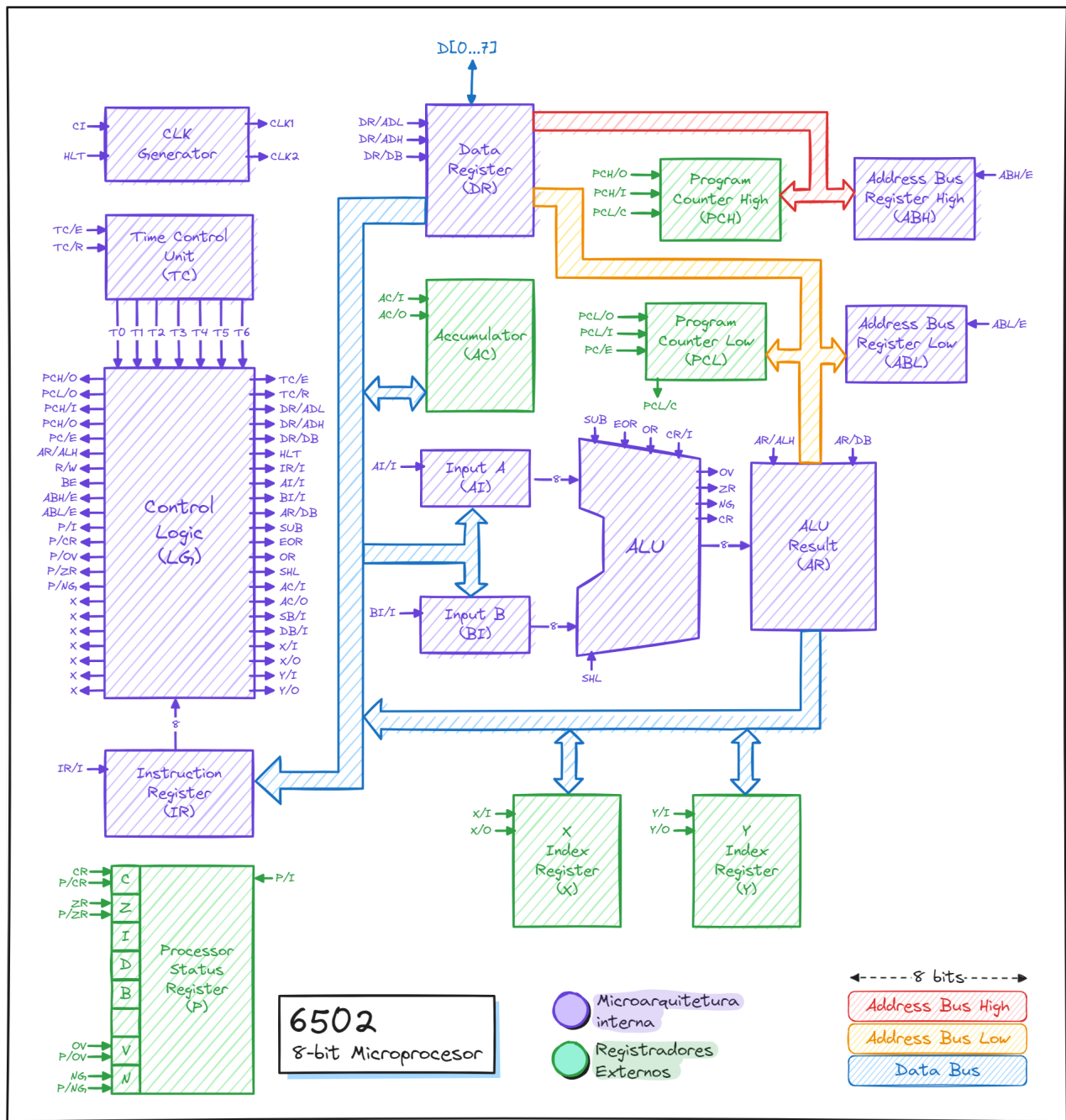
O conjunto de instruções apresentado é apenas um subconjunto da família MCS650X original. Os mesmos *opcodes* da arquitetura original serão mantidos aqui.

Tabela 2 – Instruções de transferência

<i>Opcode</i>	<i>Mod. End</i>	<i>Assembly</i>	<i>Operação</i>
2F	imm	LDA imm	RegAC \leftarrow imm
2F	abs	LDA addr	RegAC \leftarrow Mem[addr]
2F	zpg	LDA addrLow	RegAC \leftarrow Mem[00 + addrLow]
2F	(abs, x)	LDA addr, x	RegAC \leftarrow Mem[addr + x]
2F	(abs, y)	LDA addr, x	RegAC \leftarrow Mem[addr + x]

Fonte: Autoria Própria

Figura 7 – Datapath



Fonte: Autoria própria

3.2 Unidades funcionais

3.2.1 Registradores

- A
- X

- Y
- SP
- PC
- SR

3.2.2 Unidade de Controle

A unidade de controle é responsável por enviar os sinais de controle para o processador com o objetivo de executar o comando armazenado no Registrador de Instruções. Ela é composta de 3 blocos principais sendo eles:

- 1.

4 Resultados Obtidos e Discussões

5 Considerações Finais

Referências

VAHID, F. *Digital Design with RTL Design, VHDL, and Verilog*. 2nd edition. ed. Waltham/MA, EUA: John Wiley & Sons, Inc, 2011. Citado na página [17](#).

Apêndices

Lista de Abreviações

BD Barramento de Dados. [9](#)

BE Barramento de Endereços. [9](#)