

Estudo Preliminar: MOS 6502
Universidade Tecnológica Federal do Paraná
Câmpus Curitiba

Henrique G.P. Bubniak
Henrique R. Zambenedetti

Novembro de 2023

1 Introdução

Desenvolvido na década de 1970, O MOS 6502 é um microprocessador de 8 bits criado para ser uma versão mais barata e mais rápida dos Motorola 6800- Não coincidentemente, vários dos engenheiros do projeto haviam trabalhado na Motorola desenvolvendo a família 6800 de processadores-, sendo utilizado em sistemas extremamente populares como o Commodore 64, Apple II, NES e Atari 2600.

O 6502 apresenta um instruction set simplificado de apenas 56(?) instruções, um acumulador, 6 registradores - PC, ACC, X, Y, SR, SP - e 7 flags - Negative, Carry, Overflow, Decimal, Zero, Interrupt e Break - sendo utilizadas nas instruções de branch.

Como esse microprocessador foi projetado para minimizar os custos de fabricação, optou-se por utilizar somente 6 registradores, e por causa disso, fez-se necessário que as instruções de aritmética acessassem os 255 primeiros endereços da RAM - a *zeropage* - como se fossem uma extensão do banco de registradores.

Originalmente o processador da MOS apresenta instruções de 1, 2 e 3 bytes, mas nesta implementação todas as instruções têm um comprimento fixo de 14 bits por motivos de especificação.

2 Grupos de instruções: uma visão geral

É necessário explicitar que a classificação abaixo é de autoria própria e aborda somente as instruções aqui implementadas, diferindo de como a Fabricante do 6502 agrupa suas instruções.

2.1 Instruções Aritméticas

As instruções aritméticas utilizam o registrador acumulador como um dos operandos, e como o destino de armazenamento do resultado da operação. As variantes dessas instruções que foram implementadas são as do tipo *immediate* e *zeropage*. A variante *immediate* utiliza um imediato de um byte, escrito na instrução em si como o segundo operando. A variante *zeropage* utiliza um endereço de memória da *zeropage* da RAM que é escrito na instrução. O conteúdo desse endereço será utilizado como segundo operando. As instruções dessa categoria que foram implementadas são as ADC, de soma e SBC, de subtração.

2.2 Instruções de Load e Store

Foram implementadas as instruções STA, na variante *zeropage*, que armazena o conteúdo do acumulador em um endereço de memória de um byte, escrito na instrução, e as variantes *immediate* e *zeropage* da instrução LDA, que armazenam no acumulador um imediato de um byte, no caso da variante *immediate*, ou o conteúdo de um endereço de memória de um byte, no caso da variante *zeropage*.

2.3 Instruções de Branch

No 6502, as flags de SR - *Status Register* - são utilizadas para decidir se um branch será ou não tomado. Das 8 instruções de branch do processador, apenas 3 foram implementadas: BMI, *Branch on Minus*; BEQ, *Branch on Equals* e BCS, *Branch on Carry set*. Todo o endereçamento relacionado a estas instruções é relativo ao valor do PC. Além disso, foi implementada a instrução de Jump incondicional, cujo endereçamento é absoluto.

2.4 Lista de Instruções Implementadas

- ADC - Add to Accumulator with Carry
- SBC - Subtract from Accumulator with Carry
- STA - Store on Accumulator
- LDA - Load Accumulator
- JMP - Jump
- BEQ - Branch on Equals
- BMI - Branch on Minus
- BCS - Branch on Carry Set

As instruções ADC, SBC, STA e LDA apresentam diferentes modos de endereçamento, cada um recebendo um OPcode próprio, isto é, o OPcode de *ADC zpg* é diferente de *ADC imm*. Dos possíveis modos de endereçamento destas instruções, foram implementados - até o presente momento - 2 deles para cada uma delas: *zero page* e *immediate*.

Futuramente, quando da implementação da memória de Dados, serão implementados os modos absoluto e indireto.

3 Detalhes de Implementação

3.1 Registradores e Flags

Para facilitar a implementação do processador e tendo em vista que as limitações da época não se aplicam no caso em questão, os autores decidiram por implementar um banco de registradores - como na maioria dos processadores atuais - ao invés de utilizar as primeiras posições da memória de dados para simulá-los.

Além disso, foi criado separadamente um registrador de 8 bits para as flags, isto é, o *Status Register* é fisicamente separado do banco de registradores que contém o Acumulador, X, Y, e SP - Na realidade, sequer existe um *Stack Pointer* - bem como as demais posições que estariam na *zeropage*.

Das 7 flags do processador, apenas 4 estão presentes na implementação: N, Z, V, C - *Negative*, *Zero*, *Overflow* e *Carry* - simplesmente porque as demais - *Break*, *Interrupt* e *Decimal* - são utilizadas apenas em funcionalidades não implementadas.

3.2 OPCODEs e Tamanho de Instrução

Como explicitado na secção de introdução, diferentemente do MOS 6502 original que possui instruções de 1, 2 e 3 bytes, todas as instruções nesta implementação tem comprimento fixo de 14 bits.

Não obstante, os OPCODEs foram mantidos tais quais os originais, isto é, são representados por 8 bits ou dois dígitos hexadecimais. Os 6 bits remanecentes são utilizados, quando cabível, para endereçamento.

3.3 Manipulação de Flags

Tendo em vista que nem todas as instruções alteram as flags e que mesmo as que o fazem em geral modificam apenas algumas delas, será necessário implementar alguma lógica para realizar essa escolha.

Uma possibilidade pensada pelos autores foi a implementação de um novo sinal de controle de 8 bits - na realidade seriam necessário apenas 4 - correspondentes a cada uma das flags de maneira que um bit em 1 indica que a respectiva flag é alterada pela instrução sendo executada e um bit zero o contrário.

O sinal alimentaria 8 multiplexadores 2x1, um para cada flag, nos quais uma das entradas estaria conectada ao valor em *SR* e outro à flag calculada na ULA.

Referências

- Mass:werk. (n.d.). *6502 instruction set*. https://www.masswerk.at/6502/6502_instruction_set.html.
- MOS Technology. (1976a). *Mos microcomputers hardware manual*. https://bytecollector.com/archive/misc/6500-10A_MCS6500hwMan_Jan76.pdf.
- MOS Technology. (1976b). *Mos microcomputers programming manual*. https://archive.org/details/mos_microcomputers_programming_manual.