

INSTITUTO FEDERAL
Espírito Santo

Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo
Campus Serra

CURSO: Engenharia de Controle e Automação

Disciplina: Arquitetura de computadores

Implementação do Processador SingleCycle

Alunos: Ian Zotelli Silva e Hewerton Folli
Prof. Rafael Emerick Zape de Oliveira

Serra, 2023

SUMÁRIO

1.INTRODUÇÃO.....	3
2. DESENVOLVIMENTO	4
3 CONCLUSÃO.....	7
4 REFERÊNCIAS BIBLIOGRÁFICAS	8

1. INTRODUÇÃO

Este trabalho tem o objetivo de ampliar o entendimento sobre o funcionamento de um processador Arm Single Cycle em ambiente simulado, utilizando o software Modelsim e linguagem Systemverilog. Nele, modificaremos um processador para ampliar seu leque de execução de instruções, implantando o MOV, CMP, TST, EOR, LDRB, STRB e BL com base nos assuntos discutidos em sala de aula.

2. DESENVOLVIMENTO

Fizemos a instalação e configuração dos programas necessários e os utilizamos para nos familiarizarmos com o Modelsim, e a utilização do site Armconverter para converter os códigos em Assembly para hexadecimal, necessário para alterar o arquivo MEMFILE.dat, que serve como nossa memória de instruções, e o site CPUlator, para melhor entender o passo a passo do funcionamento do CPU ao serem solicitadas determinadas instruções.

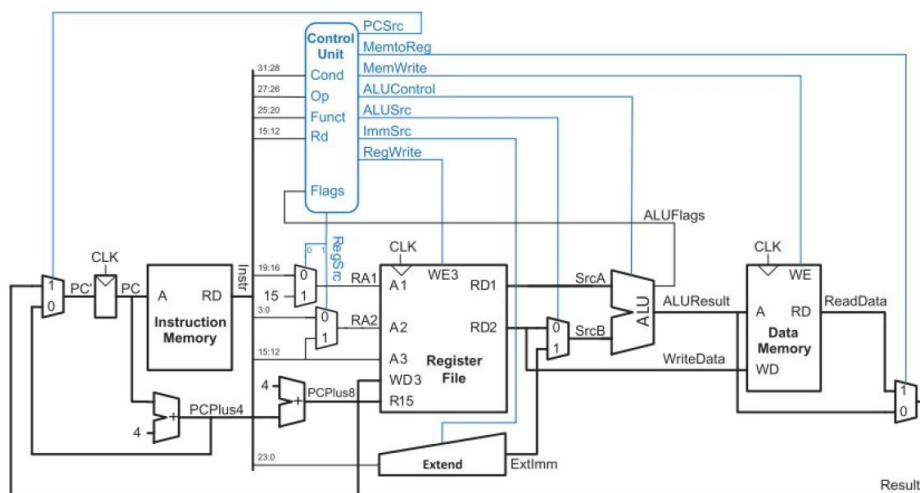


Figura 1 - Esquemático Processador Single-Cycle completo

Além disso fizemos um esquemático para compreender melhor o funcionamento da arquitetura do processador, utilizando o programa e o esquemático do livro. O esquemático não está completo, mas é um importante passo inicial para entender como os módulos estão organizados e se comunicam entre si, além do que deve ser alterado para a implementação das novas instruções.

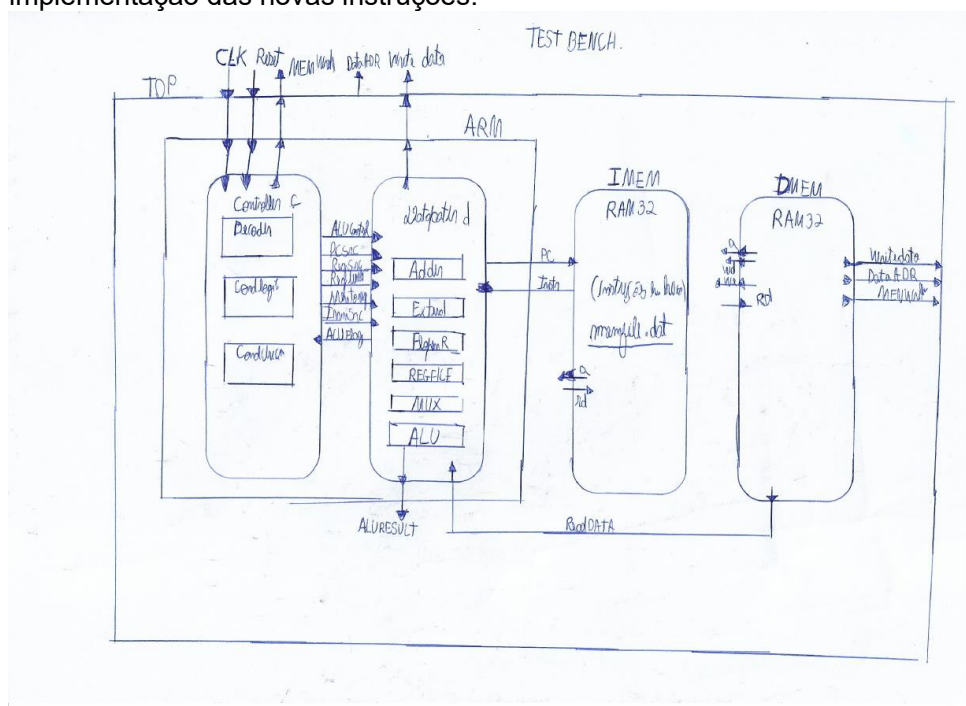


Figura 2 - Esquemático Processador Single-Cycle baseado no programa

Após isso realizamos simulações no testbench, observando a janela Wave, para preencher a tabela dos resultados esperados dos componentes do programa em cada ciclo de clock, utilizando as instruções armazenadas previamente no memfile.dat que foi fornecido pelo professor.

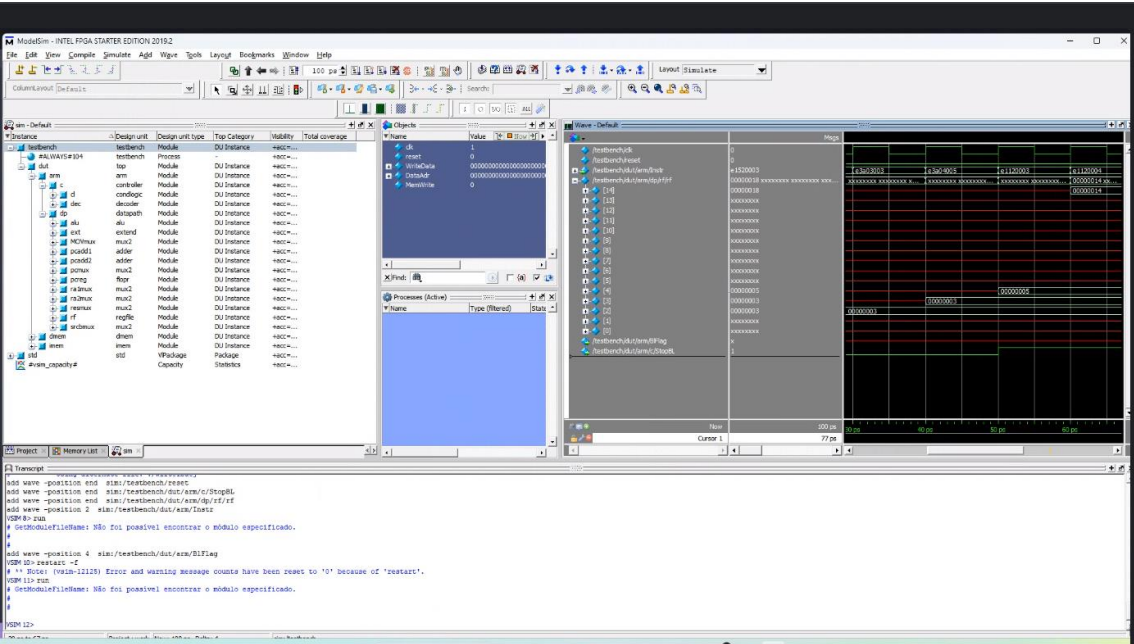


Figura 3 – Simulações no Modelsim

Ciclo	Reset	PC	Instrução	SRCA	SRCB	Branch	ALUresult	Flags3:0	CondEX	WriteData	MemWrite	ReadData
1	1	00000000	E04F000F	00000008	00000008	0	00000000	0000	1	00000008	0	xxxxxxxx
2	1	00000000	E04F000F	00000008	00000008	0	00000000	0000	1	00000008	0	xxxxxxxx
3	0	00000000	E04F000F	00000008	00000008	0	00000000	0000	1	00000008	0	xxxxxxxx
4	0	00000004	E2802005	00000000	00000005	0	00000005	0000	1	xxxxxxxx	0	xxxxxxxx
5	0	00000008	E280300C	00000000	0000000C	0	0000000C	0000	1	xxxxxxxx	0	xxxxxxxx
6	0	0000000C	E2437009	0000000C	00000009	0	00000003	0000	1	xxxxxxxx	0	xxxxxxxx
7	0	00000010	E1874002	00000003	00000005	0	00000007	0000	1	00000005	0	xxxxxxxx
8	0	00000014	E0035004	0000000C	00000007	0	00000004	0000	1	00000007	0	xxxxxxxx
9	0	00000018	E0855004	00000004	00000007	0	0000000B	0000	1	00000007	0	xxxxxxxx
10	0	0000001C	E0558007	0000000B	00000008	0	00000008	0000	1	00000003	0	xxxxxxxx
11	0	00000020	0A00000C	00000028	00000003	1	00000058	0010	0	xxxxxxxx	0	xxxxxxxx
12	0	00000024	E0538004	0000000C	00000030	0	00000005	0010	1	00000007	0	xxxxxxxx
13	0	00000028	AA000000	00000030	00000007	1	00000030	0010	1	00000000	0	xxxxxxxx
14	0	00000030	E0578002	00000003	00000000	0	FFFFFFFE	0010	1	00000005	0	xxxxxxxx
15	0	00000034	B2857001	0000000B	00000005	0	0000000C	0100	1	xxxxxxxx	0	xxxxxxxx
16	0	00000038	E0477002	0000000C	00000001	0	00000007	0100	1	00000005	0	xxxxxxxx
17	0	0000003C	E5837054	0000000C	00000005	0	00000060	0100	1	00000007	1	xxxxxxxx
18	0	00000040	E5902060	00000000	00000054	0	00000060	0100	1	00000000	0	00000007
19	0	00000044	E08FF000	0000004C	00000060	0	0000004C	0100	1	00000000	0	xxxxxxxx
20	0	0000004C	EA000001	00000054	00000004	1	00000058	0100	1	00000000	0	xxxxxxxx
21	0	00000058	E5802064	00000000	00000063	0	00000064	0100	1	00000007	1	xxxxxxxx
22	xxxxxxx	xxxxxxx	xxxxxxx	xxxxxxx	xxxxxxx	xxxxxxx	xxxxxxx	xxxxxxx	xxxxxxx	xxxxxxx	xxxxxxx	xxxxxxx

Figura 4 – Tabela de resultados preenchida

O próximo passo foi implementar as seguintes instruções no programa:

MOV (Move): O comando MOV é usado para copiar um valor de um local para outro. Por exemplo, MOV R1, R2 copia o valor do registrador R2 para o registrador R1.

CMP (Compare): O comando CMP é usado para comparar dois valores. Ele subtrai o segundo valor do primeiro, mas não armazena o resultado. O CMP define as flags do processador com base na comparação, permitindo que instruções condicionais sejam executadas posteriormente.

TST (Test): O comando TST é semelhante ao CMP, mas também atualiza as flags do processador. Ele realiza uma operação lógica AND bit a bit entre os dois operandos, definindo as flags com base no resultado.

EOR (Exclusive OR): O comando EOR executa uma operação lógica XOR (ou exclusiva) bit a bit entre os dois operandos e armazena o resultado no registrador de destino.

LDRB (Load Byte): O comando LDRB é usado para carregar um byte da memória para um registrador. Ele é comumente usado para acessar dados de 8 bits.

STRB (Store Byte): O comando STRB é usado para armazenar um byte de um registrador na memória. Ele é usado para escrever dados de 8 bits na memória.

BL (Branch and Link): O comando BL é usado para fazer um desvio (branch) para um endereço especificado e armazenar o endereço de retorno no registrador LR (Link Register). Geralmente, é usado para chamar sub-rotinas ou funções.

O processo de implementação pode ser visto com mais detalhes no link vídeo de youtube disponibilizado neste github.

3 CONCLUSÃO

Ao implementar essas instruções adicionais na microarquitetura do processador ARM monociclo, foi possível expandir sua capacidade de oferecer suporte a uma gama mais ampla de operações e funcionalidades. O Modelsim é um programa propenso a travamentos e não possui funcionalidades de programas mais novos, como acusar a não utilização de variáveis declaradas não utilizadas, o que pode gerar confusão e erros em um programa case sensitive. Porém, após compreender o básico do funcionamento do processador e a localização e sua organização em módulos, a modificação do código para implementar as novas funcionalidades se torna mais rápida e intuitiva.

4. REFERÊNCIAS BIBLIOGRÁFICAS

HARRIS, Sarah L.; HARRIS, David Money. Digital Design and Computer Architecture, ARM® Edition. [S.l.]: Morgan Kaufmann, 2015.