INSTITUTO FEDERAL DO ESPÍRITO SANTO
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

JOÃO MANOEL SEVILHA OLIVEIRA BARBOSA
JOÃO VICTOR DA SILVA DE MIRANDA
RAMON SOUSA ALVES

IMPLEMENTAÇÃO EM HARDWARE DE INSTRUÇÃO DE PROCESSADOR ARM MONOCÍCLO

SERRA
2021

JOÃO MANOEL SEVILHA OLIVEIRA BARBOSA
JOÃO VICTOR DA SILVA DE MIRANDA
RAMON SOUSA ALVES

IMPLEMENTAÇÃO EM HARDWARE DE INSTRUÇÃO DE PROCESSADOR ARM MONOCÍCLO

Projeto Final apresentado ao curso de Engenharia de Controle e Automação, no Instituto Federal do Espirito Santo, como requisito parcial de obtenção de nota da disciplina Arquitetura de Computadores.
Professor: Me. Rafael Emerick Zape de Oliveira

SERRA
2021

RESUMO

O trabalho tem o objetivo de proporcionar aos alunos a compreensão e expansão do funcionamento do processador ARM single cycle através do uso de linguagem de programação SystemVerilog em simulação por meio do ModelSim.

Palavras-chave: Arquitetura de Computadores, ARM Single-Cycle, SystemVerilog, ModelSim.

SUMÁRIO

1 INTRODUÇÃO

O grupo teve como ponto de partida as aulas e os materiais disponibilizados pelo professor, entendendo os comandos do trabalho e compartilhando entre si ideias iniciais do que fazer. Foram necessárias as instalações dos softwares recomendados, bem como seus respectivos testes para garantir a aprendizagem nos mesmos. Afim de simplificar o processo de elaboração do código em conjunto e atendendo a recomendação do professor, fizemos um curso gratuito de Git disponibilizado na plataforma Udemy. Assim, o desenvolvimento e implementação do projeto foram inicados.

1.1 Processador ARM Single-Cycle

Utilizamos o esquemático do processador ARM monociclo de referência disponibilizado pelo professor.

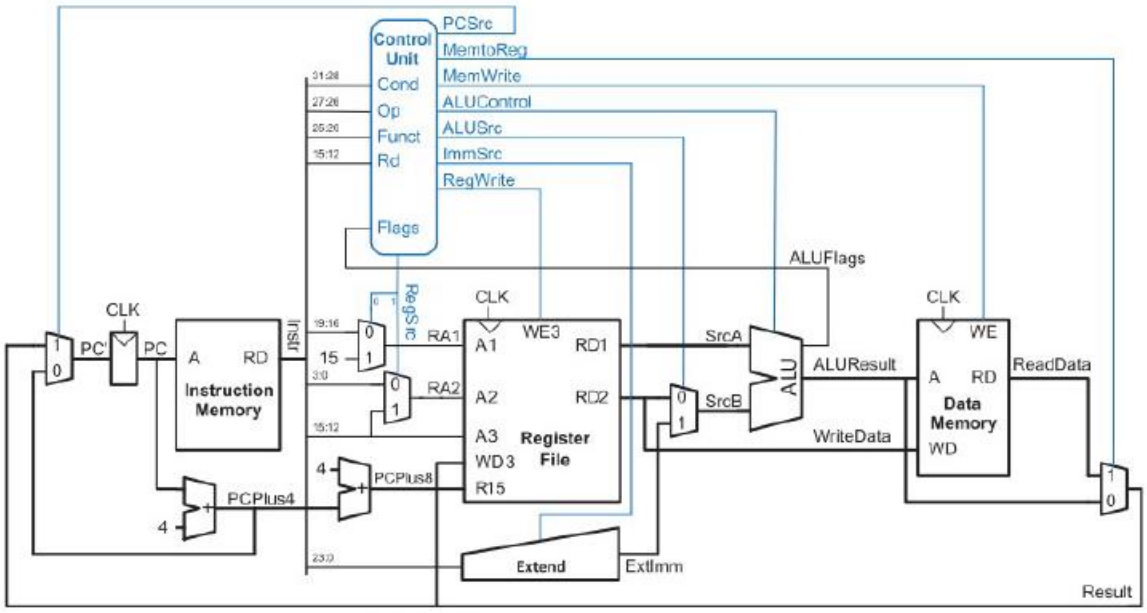

Figura 1 - Processador Single-Cycle completo

2 TESTANDO O PROCESSADOR ARM MONOCICLO

Afim de prever o que deve acontecer em cada ciclo ao executar o programa, desenhamos o esquemático correspondente ao HDL em SystemVerilog do processador ARM, utilizando a ferramenta Drawio.

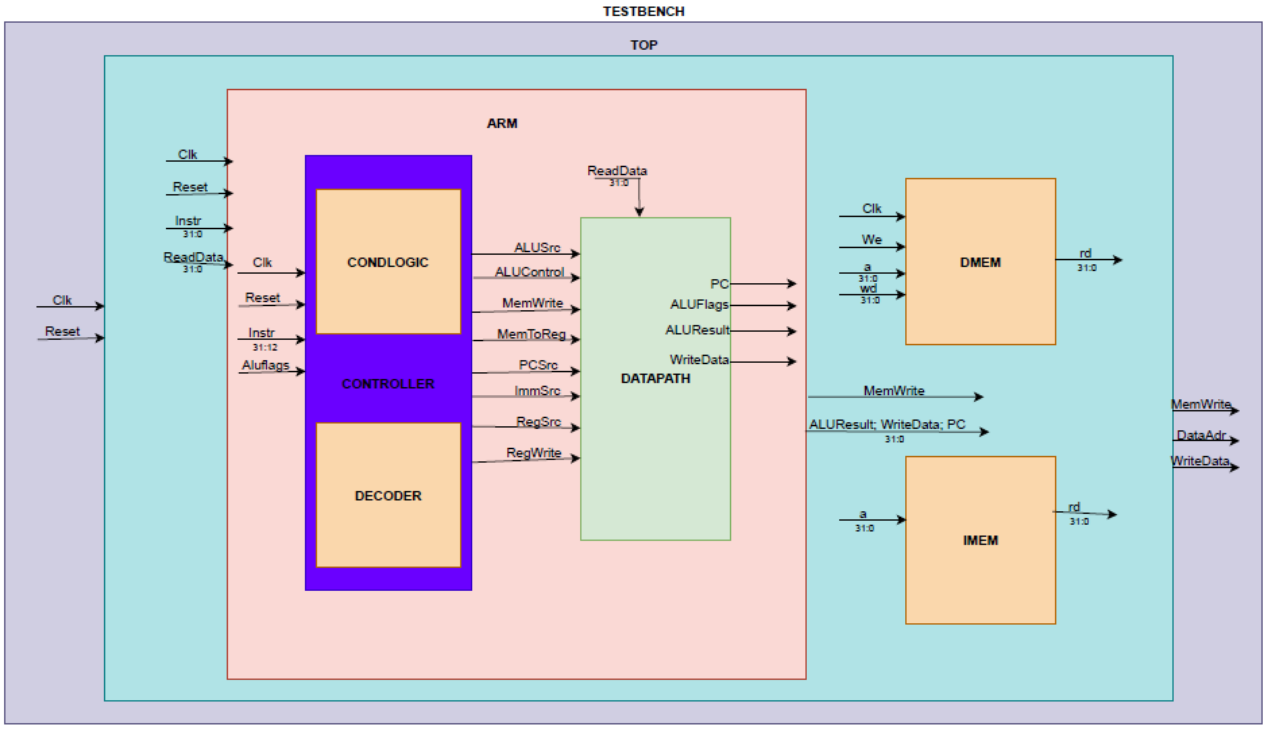

Figura 2 - Esquemático do Processador ARM

Também com o intuito de prever o que deve ocorrer, preenchemos uma tabela com as previsões da execução do código memfile.s. Com isso iriamos conseguir comparar a execução do programa com a tabela.

Ciclo	Reset	PC	Instrução	SrcA	SrcB	Branch	AluResult	Flags3:0	CondEx	WriteData	MemWrite	ReadData
1	1	0	E04F000F	8	8	0	0	0	1	8	0	X
2	0	4	E2802005	0	5	0	5	0	1	X	0	X
3	0	8	E280300C	0	0C	0	0C	0	1	X	0	X
4	0	0C	E2437009	0C	9	0	3	0	1	X	0	X
5	0	10	E1874002	3	5	0	7	0	1	5	0	X
6	0	14	E0035004	C	7	0	4	0	1	7	0	X
7	0	18	E0855004	4	7	0	B	0	1	7	0	X
8	0	1C	E0558007	B	3	0	8	0	1	3	0	X
9	0	20	0A00000C	28	30	1	58	2	0	X	0	X
10	0	24	E0538004	C	7	0	5	2	1	7	0	X
11	0	28	AA000000	30	0	1	30	2	1	0	0	X
12	0	30	E0578002	3	5	0	FFFFFFFE	2	1	5	0	X
13	0	34	B2857001	B	1	0	C	8	1	X	0	X
14	0	38	E0477002	C	5	0	7	8	1	5	0	X
15	0	3C	E5837054	C	54	0	60	8	1	7	1	X
16	0	40	E5902060	0	60	0	60	8	1	0	0	7
17	0	44	E08FF000	4C	0	0	4C	8	1	0	0	X
18	0	4C	EA000001	54	4	1	58	8	1	X	0	X
19	0	58	E5802064	0	64	0	64	8	1	7	1	X

Figura 3 - Tabela do Single Cycle ARM

Afim de validar o teste nota-se que a instrução E5802064 (*STR R2, [R0, #0x64]*) escreveu o valor 7 no endereço 100 (0x64) conforme esperado.

2.1 Simulação do Processador ARM Single-Cycle

Através do ModelSim simulamos o arquivo arm_single.sv. Com isso concluímos que o processador está funcionando corretamente e validamos que aconteceu o que era esperado de acordo com a tabela .

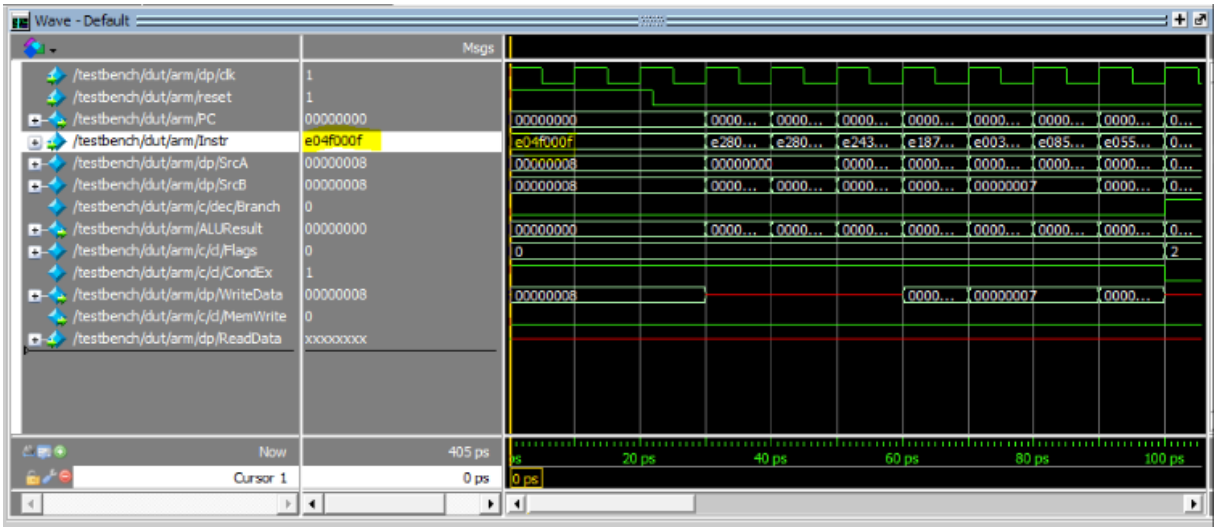

Figura 4 - Formas de onda antes da implementação

3 MODIFICANDO O PROCESSADOR ARM MONOCICLO

Afim de aumentar a quantidade de funções executadas pelo processador ARM, implementamos novas funções dentro das possibilidades MOV, CMP, TST, EOR, LDRB e STRB. Optamos pela implementação das instruções MOV, CMP, TST e EOR. Abaixo listamos as instruções e para o que são designadas:

Instrução MOV: copia o valor de determinado operando para um registrador.

Instrução CMP: compara dois valores, faz a subtração e atualiza as flags. O resultado é descartado uma vez que interessa é a comparação entre eles.

Instrução TST: compara o valor no operando 2 com o registrador pré-determinado, fazendo AND bit a bit. As flags são atualizadas porém o valor é descartado.

Instrução EOR: executa “ou exclusivo” entre dois operandos, bit a bit.

3.1 Ajustes no código em SystemVerilog

Afim de inserir novas funções, incluímos novas variáveis para coletar ou armazer bytes na memória e também transmiti-los entre os componentes. Foi necessário ampliar a dimensão de certos dispositivos tais qual a ALU. Utilizamos os conhecimentos de instruções ARM para Dataprocessing, Branch e Memory.

3.2 Nova Simulação

Finalizadas as mudanças no código fizemos a simulação e temos como resultado a seguinte tabela:

Ciclo	Reset	PC	Instrução	SrcA	SrcB	Branch	AluResult	Flags3:0	CondEx	WriteData	MemWrite	ReadData
1	1	0	E3A01000	X	0	0	X	0	1	X	0	X
2	0	4	E3A03004	X	4	0	X	0	1	X	0	X
3	0	8	E3A07007	X	7	0	X	0	1	X	0	X
4	0	C	E0235007	4	7	0	3	0	1	7	0	X
5	0	10	E3130001	4	1	0	0	0	1	0	0	X
6	0	14	E0471003	7	4	0	3	4	1	4	0	X
7	0	18	E1530007	4	7	0	FFFFFFFD	4	1	7	0	X
8	0	1C	10831007	4	7	0	B	8	1	7	0	X
9	0	20	E04F000F	28	28	0	0	8	1	28	0	X
10	0	24	E2802005	0	5	0	5	8	1	3	0	X
11	0	28	E280300C	0	C	0	C	8	1	X	0	X
12	0	2C	E2437009	C	9	0	3	8	1	X	0	X
13	0	30	E1874002	3	5	0	7	8	1	5	0	X
14	0	34	E0035004	C	7	0	4	8	1	7	0	X
15	0	38	E0855004	4	7	0	B	8	1	7	0	X
16	0	3C	E0558007	B	3	0	8	8	1	3	0	X
17	0	40	0A00000C	48	30	1	78	2	0	X	0	X
18	0	44	E0538004	C	7	0	5	2	1	7	0	X
19	0	48	AA000000	50	0	1	50	2	1	0	0	X
20	0	50	E0578002	3	5	0	FFFFFFFE	2	1	5	0	X
21	0	54	B2857001	B	1	0	C	8	1	B	0	X
22	0	58	E0477002	C	5	0	7	8	1	5	0	X
23	0	5C	E5837054	C	54	0	60	8	1	7	1	X
24	0	60	E5902060	0	60	0	60	8	1	0	0	7
25	0	64	E08FF000	6C	0	0	6C	8	1	0	0	X
26	0	6C	EA000001	74	4	1	78	8	1	B	0	X
27	0	78	E5802064	0	64	0	64	8	1	7	1	X

Figura 5 - Tabela após implementação

Após implementar as novas funções, temos as seguintes formas de onda do processador monociclo:

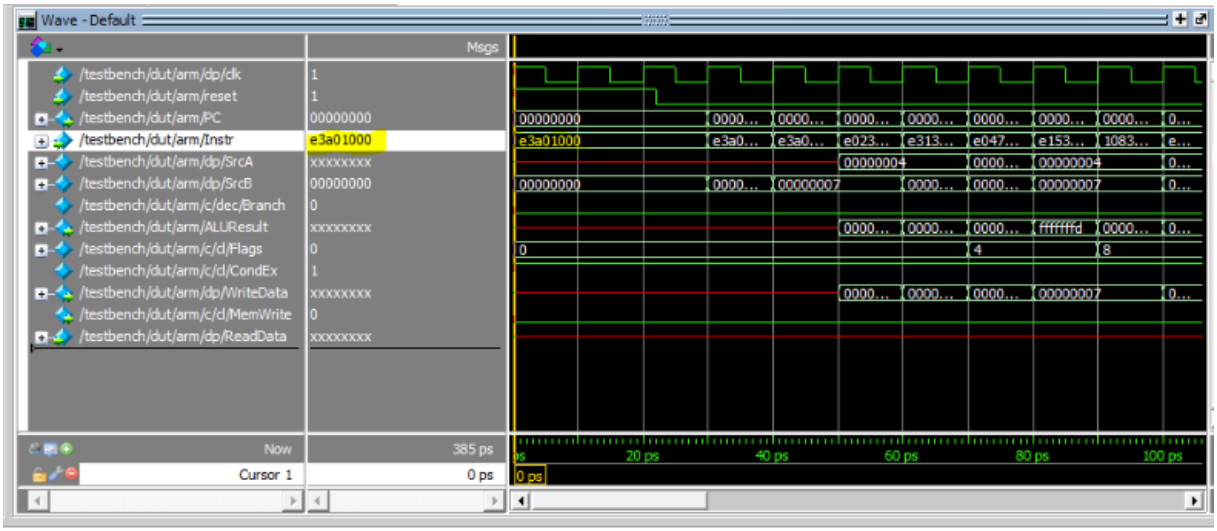

Figura 6 - Formas de onda depois da implementação

4 CONCLUSÃO

Com a conclusão do trabalho o grupo, em consenso, considera de extrema importância essa atividade para a consolidação dos conhecimentos adquiridos ao longo da disciplina. Para executá-lo tivemos que realizar consulta nos materiais do professor, procurar vídeos e explicações na internet, o que não seria realizado se não tivéssemos esse estimulo de necessidade. Também, graças ao estímulo do professor, pudemos aprender sobre Git, que foi muito útil para execução de tarefas em conjunto, e será mais útil ainda pelo resto do curso.

Vimos que por mais que na teoria o processador ARM monociclo seja simples, para conseguir realizar o trabalho precisamos de bastante estudo e esforço para dominar a lógica e conseguir implementar as instruções.

REFERÊNCIAS

Git e contribuições para projetos Open Source. Disponível em: <www.udemy.com/course/git-e-github/>. Acesso em 01 de abril de 2021.

Github. Disponível em: <https://gitlab.com/prof.rafael.emerick/armprocessors>. Acesso em: 10 de abril de 2020

David, Money Harris, Sarah, L. Harris. Digital Design and Computer Architecture: ARM® Edition © Chapter 7, 2015.