

UNIVERSIDADE DE BRASÍLIA



INSTITUTO DE CIÊNCIAS EXATAS

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

ORGANIZAÇÃO E ARQUITETURA DE COMPUTADORES

TURMA C

Projeto Final

Nome:

Johannes Peter Schulte
Mateus de Moura Ramos
Bittencourt

Matrícula:

15/0132662
16/0079284

JULHO DE 2019

1 Descrição

Projeto Final da disciplina no qual foi projetado e implementado uma versão de um processador Uniciclo seguindo a arquitetura básica do RISC-V. O projeto foi escrito usando a linguagem de programação VHDL e utilizando as ferramentas de desenvolvimento Quartus II e ModelSim.

2 Implementação

O processador Uniciclo foi implementado de uma forma que os módulos principais foram agrupados em blocos, e depois esses blocos foram integrados em um único módulo do processador. Essa estratégia foi adotada de forma que os blocos podem ser testados separadamente, facilitando a depuração e integração entre os módulos.

- Módulo processador: Módulo que representa o funcionamento do processador como um todo, responsável por ligar os sinais entre os blocos, receber o clock do processador, distribuir os sinais de controle e direcionar o fluxo dos sinais utilizando somadores e módulos MUX.
 - Módulo fetch: Função de ler uma instrução da memória de instruções e incrementar o PC do processador.
 - * Módulo memoriaIns: Responsável por receber o PC e retornar a instrução correspondente armazenada no arquivo memoria.mif.
 - * Módulo PC: Contador do processador.
 - Módulo breg_ula: Realiza ambas as funções de *decode* e *execute*, faz a ligação da ULA com os valores de registradores e imediatos.
 - * Módulo xregs: Armazena e atualiza o valor de 32 registradores de 32 bits, tendo o registrador 0 sempre com o valor de 0.
 - * Módulo ula: Executa as instruções e retorna o resultado e um sinal para resultados iguais a 0.
 - * Módulo controleULA: Responsável por indicar a ula qual operação deve ser feita baseada no opcode, func3 e func7 recebido.
 - * Módulo genImm32: Gera os valores de imediatos para as operações e saltos.
 - Módulo memoria: É o módulo da memória de dados com o MUX implementado. Armazena dados ou envia para o write back de acordo com os sinais de controle recebidos.
 - Módulo riscv_pkg: Um módulo para declaração dos outros módulos. Serve para fazer a integração dos módulos.
 - Módulo de teste processador_tb: Módulo utilizado para ver o comportamento do processador. Inicializa e faz os ciclos de clock do processador e dá memória.
 - Módulos mux e somador: Controle de fluxo dos sinais dos módulos.
 - Arquivos memoria.mif e dados.mif: Dump de memória de algum programa executado no RARS. Lidos pelos módulos memoriaIns e memoriaDados, respectivamente.

3 Testes

Para testar o projeto foram realizados testes utilizando arquivos VHDL para testbench e arquivos .mif com instruções e dados de programas executados pelo RARS. Durante o desenvolvimento dos módulos os seguintes foram testados com um arquivo testbench: genImm32, memoriaDados, memoriaIns, xregs e ula. Quando os módulos foram integrados utilizamos alguns programas de risc-v, como o disponibilizado pelo professor para a apresentação e outros que foram desenvolvidos ao longo da disciplina.

4 Simulações

/clock_general	1									
/i1/fetch/pc/pcout	00...	00000000		00000004				00000008		
/i1/fetch/q_out	0F...	0FF06293			0F02F293			00002437		
/i1/fetch/rs1	0	0			5			0		
/i1/fetch/rs2	31	31			16			0		
/i1/fetch/rd	21	5						8		
/i1/breg_ula/ula/A	0	0				255			0	
/i1/breg_ula/ula/B	255	255			240			8192		
/i1/breg_ula/ula/Z	00...	000000FF			00000000	000000F0		00002000		
/i1/breg_ula/genImm32/imm32	255	255			240			8192		
/i1/memoria/dataout	00...	000000FF			00000000	000000F0		00002000		

Figura 1: PC 0 até 8.

/clock_general	1									
/i1/fetch/pc/pcout	00...	0000000C		00000010				00000014		
/i1/fetch/q_out	00...	00042483			00442903			012489B3		
/i1/fetch/rs1	8	8						9		
/i1/fetch/rs2	0	0			4			18		
/i1/fetch/rd	9	9			18			19		
/i1/breg_ula/ula/A	0	0	8192						15	
/i1/breg_ula/ula/B	0	0			4			0	63	
/i1/breg_ula/ula/Z	00...	00000...	00002000		00002004			00002000	0000004E	
/i1/breg_ula/genImm32/imm32	0	0			4			0		
/i1/memoria/dataout	00...	0000000F						0000003F	00002000	0000004E

Figura 2: PC C até 14.

/clock_general	1									
/i1/fetch/pc/pcout	00...	00000024		00000028				0000002C		
/i1/fetch/q_out	0F...	0FF00A93			014AFB33			014AEB83		
/i1/fetch/rs1	0	0			21					
/i1/fetch/rs2	31	31			20					
/i1/fetch/rd	21	21			22			23		
/i1/breg_ula/ula/A	0	0				255				
/i1/breg_ula/ula/B	255	255			0	2032				
/i1/breg_ula/ula/Z	00...	000000FF			00000000	000000F0		000007FF		
/i1/breg_ula/genImm32/imm32	255	255			0					
/i1/memoria/dataout	00...	000000FF			00000000	000000F0		000007FF		

Figura 3: PC 24 até 2C.

/clock_general	1								
/i1/fetch/pc/pcout	00...	00000030			00000034			00000038	
/i1/fetch/q_out	01...	014ACC33			004A9313			FF0003B7	
/i1/fetch/rs1	21	21						0	
/i1/fetch/rs2	20	20			4			16	
/i1/fetch/rd	24	24			6			7	
/i1/breg_ula/ula/A	255	255						0	
/i1/breg_ula/ula/B	2032	2032			4			-16777216	
/i1/breg_ula/ula/Z	00...	0000070F			00000FF0			FF000000	
/i1/breg_ula/genImm32/imm32	0	0			4			-16777216	
/i1/memoria/dataout	00...	0000070F			00000FF0			FF000000	

Figura 4: PC 30 até 38.

/clock_general	1								
/i1/fetch/pc/pcout	00...	0000003C			00000040			00000044	
/i1/fetch/q_out	00...	0043DE13			4043DE93			0062A433	
/i1/fetch/rs1	7	7						5	
/i1/fetch/rs2	4	4						6	
/i1/fetch/rd	28	28			29			8	
/i1/breg_ula/ula/A	0	0	-16777216					240	
/i1/breg_ula/ula/B	4	4						0	4080
/i1/breg_ula/ula/Z	00...	00000... 0FF00000			FFF00000			00000001	00000001
/i1/breg_ula/genImm32/imm32	4	4						0	
/i1/memoria/dataout	00...	00000... 0FF00000			FFF00000			00000001	00000001

Figura 5: PC 3C até 44.

/clock_general	1								
/i1/fetch/pc/pcout	00...	00000048			0000004C			00000050	
/i1/fetch/q_out	00...	005324B3			005039E3			0002BA33	
/i1/fetch/rs1	6	6			0			5	
/i1/fetch/rs2	5	5						0	
/i1/fetch/rd	9	9			19			20	
/i1/breg_ula/ula/A	240	240	4080		0			240	
/i1/breg_ula/ula/B	4080	4080	240					0	
/i1/breg_ula/ula/Z	00...	00000... 00000000			00000001			00000000	
/i1/breg_ula/genImm32/imm32	0	0							
/i1/memoria/dataout	00...	00000... 00000000			00000001			00000000	

Figura 6: PC 48 até 50.

/clock_general	1								
/i1/fetch/pc/pcout	00...	00000054			0000005C			00000060	
/i1/fetch/q_out	00...	008000EF			40628E33			00008067	
/i1/fetch/rs1	0	0			5			1	
/i1/fetch/rs2	8	8			6			0	
/i1/fetch/rd	1	1			28			0	
/i1/breg_ula/ula/A	240	240	0		240			88	
/i1/breg_ula/ula/B	0	0	1		4080			0	
/i1/breg_ula/ula/Z	00...	00000... 00000001			FFFFFF FFFF 100			000000F0	00000058
/i1/breg_ula/genImm32/imm32	8	8			0				
/i1/memoria/dataout	00...	00000... 00000001			FFFFFF FFFF 100			000000F0	00000058

Figura 7: PC 54 até 60.

/dock_general	1								
i1/fetch/pc/pcout	00...	00000058		00000064			00000068		
i1/fetch/q_out	00...	00C0006F		FFE00293			00228293		
i1/fetch/rs1	0	0					5		
i1/fetch/rs2	12	12		30			2		
i1/fetch/rd	0	0		5					
i1/breg_ula/ula/A	88	88	0						-2
i1/breg_ula/ula/B	0	0		-2			2		
i1/breg_ula/ula/Z	00...	00000... 00000000		FFFFFFFE			00000002 00000000		
i1/breg_ula/genImm32/imm32	12	12		-2			2		
i1/memoria/dataout	00...	00000... 00000000		FFFFFFFE			00000002 00000000		

Figura 8: PC 58 até 68.

/dock_general	1								
i1/fetch/pc/pcout	00...	0000006C		00000068			0000006C		
i1/fetch/q_out	FE...	FE028EE3		00228293			FE028EE3		
i1/fetch/rs1	5	5							
i1/fetch/rs2	0	0		2			0		
i1/fetch/rd	29	29		5			29		
i1/breg_ula/ula/A	-2	-2	0						2
i1/breg_ula/ula/B	0	0		2			0		
i1/breg_ula/ula/Z	FF...	FFFFF... 00000000		00000002			00000000 00000002		
i1/breg_ula/genImm32/imm32	-4	-4		2			-4		
i1/memoria/dataout	FF...	FFFFF... 00000000		00000002			00000000 00000002		

Figura 9: PC 6C até 6C.

/dock_general	1								
i1/fetch/pc/pcout	00...	00000070		00000074			00000070		
i1/fetch/q_out	FF...	FFF28293		FE029EE3			FFF28293		
i1/fetch/rs1	5	5							
i1/fetch/rs2	31	31		0			31		
i1/fetch/rd	5	5		29			5		
i1/breg_ula/ula/A	2	2			1				
i1/breg_ula/ula/B	-1	-1		0			-1		
i1/breg_ula/ula/Z	00...	00000001		00000000			00000000		
i1/breg_ula/genImm32/imm32	-1	-1		-4			-1		
i1/memoria/dataout	00...	00000001		00000000			00000000		

Figura 10: PC 70 até 70.

/dock_general	1								
i1/fetch/pc/pcout	00...	00000074		00000078			0000007C		
i1/fetch/q_out	FE...	FE029EE3		00000000					
i1/fetch/rs1	5	5		0					
i1/fetch/rs2	0	0							
i1/fetch/rd	29	29		0					
i1/breg_ula/ula/A	1	1	0						
i1/breg_ula/ula/B	0	0							
i1/breg_ula/ula/Z	00...	00000... 00000001		00000000					
i1/breg_ula/genImm32/imm32	-4	-4		0					
i1/memoria/dataout	00...	00000... 00000001		00000000					

Figura 11: Final do Programa.

(0)	0	0	(0)	0	0
(1)	0	0	(1)	0	88
(2)	0	0	(2)	0	0
(3)	0	0	(3)	0	0
(4)	0	0	(4)	0	0
(5)	240	240	(5)	240	0
(6)	4080	4080	(6)	4080	4080
(7)	000...	FF000000	(7)	000...	FF000000
(8)	8192	8192	(8)	8192	1
(9)	15	15	(9)	15	0
(10)	78	78	(10)	78	78
(11)	0	0	(11)	0	0
(12)	0	0	(12)	0	0
(13)	0	0	(13)	0	0
(14)	0	0	(14)	0	0
(15)	0	0	(15)	0	0
(16)	0	0	(16)	0	0
(17)	0	0	(17)	0	0
(18)	63	63	(18)	63	63
(19)	78	78	(19)	78	1
(20)	2032	2032	(20)	2032	0
(21)	255	255	(21)	255	255
(22)	240	240	(22)	240	240
(23)	2047	2047	(23)	2047	2047
(24)	1807	1807	(24)	1807	1807
(25)	0	0	(25)	0	0
(26)	0	0	(26)	0	0
(27)	0	0	(27)	0	0
(28)	000...	0FF00000	(28)	0	-3840
(29)	000...	0FF00000	(29)	000...	0FF00000
(30)	0	0	(30)	0	0
(31)	0	0	(31)	0	0

Figura 12: Estado do xregs no PC 68 e no final do programa.