

ORGANIZAÇÃO E ARQUITETURA DE COMPUTADORES

Laboratório 4: CPU RISC-V MULTICICLO

Alexandre Bernardi Peres, 17/0171698

Caio Couto Moreira, 15/0155433

Valesca Gomes Soares, 16/0041091

Brasília, Janeiro de 2020



**ENGENHARIA
MECATRÔNICA**
UNIVERSIDADE DE BRASÍLIA

0.1 (3.0) Dado o processador MULTICICLO RISC-V v2.2b ISA RV32IMF escrito em Verilog:

0.1.1 (1.0) Desenhe o Caminho de Dados completo (conforme diagrama visto em aula);

0.1.1.1 Solução

INSTRUÇÕES	MEMZREG	LEMEM	ORIGPC	ALUCONTROL	ESCREVEMEM	ORIGULA	ORIGBULA	ESCREVEPCBACK	ESCREVEPC	ESCREVEPCCOND	load
LOAD	3b000	1b1	3b000	OPADD	1b0	3b000	3b010	1b0	1b0	1b0	1b1
IMM	3b000	1b0	3b000	X	1b0	3b000	3b010	1b0	1b0	1b0	1b0
AUIPC	3b000	1b0	3b000	OPADD	1b0	3b010	3b010	1b0	1b0	1b0	1b0
STORE	3b000	1b0	3b000	OPNULL	1b1	3b000	3b000	1b0	1b0	1b0	1b1
RTYPE	3b000	1b0	3b000	X	1b0	3b000	3b000	1b0	1b0	1b0	1b0
LUI	3b000	1b0	3b000	OPLUI	1b0	3b000	3b010	1b0	1b0	1b0	1b0
BRANCH	3b000	1b0	3b001	OPNULL	1b0	3b000	3b000	1b0	1b0	1b1	1b0
JAL	3b001	1b0	3b001	OPNULL	1b0	3b000	3b000	1b0	1b1	1b0	1b0
JALR	3b001	1b0	3b010	OPADD	1b0	3b000	3b010	1b0	1b1	1b0	1b0

Figura 1: Caminho de Dados

0.1.2 (1.0) Desenhe a máquina de estado do Bloco de Controle;

0.1.2.1 Solução

0.1.3 (1.0) Modificando as seleções no arquivo Parametros.v e recompilando o processador, faça duas tabelas comparativas dos requerimentos físicos e temporais das CPUs MULTICICLO com as ISAs RV32I, RV32IM e RV32IMF.

0.1.3.1 Solução

ISA	Número de ALMs	Número de Registradores	Quantidade de bits de memória	Número de DSPs
RV32I	4585	4035	3086257	12
RV2IM	7309	4025	3086257	24
RV32IMF	10243	6506	3133521	30

ISA	maior atraso tpd	maiores tempos tco, th, tsu	máxima frequência de clock utilizável	Requerimentos não atendidos (slacks)
RV32I	FF 24.644	tsu:14.225 th:6.903 tco:35.027	69.1 MHz	5.528
RV2IM	FF 25.308	tsu:12.936 th:6.945 tco:35.785	70.3 MHz	5.776
RV32IMF	FF 24.500	tsu:14.158 th:6.919 tco:34.708	68.58 MHz	5.419

É possível notar através da tabela que quão mais complexa for a ISA, maior vai ser o número de ALMs, provando o efeito das operações de ponto flutuante no desempenho do processador, custando mais espaço para a execução.

O número de registradores utilizados aumentou com a complexidade da ISA, pois o número de instruções foi aumentando. Assim como a quantidade de bits utilizados da memória.

O número de registradores utilizados aumentou com a complexidade da ISA, pois o número de instruções foi aumentando. Assim como a quantidade de bits utilizados da memória.

O número de DSPs também aumentou, já que quanto maior a complexidade da ISA, mais operações matemáticas são necessárias.

0.2 (7.0) Para seu programa do polígono fechado:

0.2.1 (3.0) Execute no Processador MULTICICLO com N=3,...,20. Filme a execução e tire fotos para colocar no relatório.

VIDEO Poligono Multiciclo

0.2.2 (2.0) Defina, através de testes (modificando o valor do divisor de frequências), qual a maior frequência de clock utilizável.

0.2.2.1 Solução

VIDEO Poligono Multiciclo (frequencia máxima)

Aqui nós fizemos os testes manuais e chegamos a conclusão que a frequencia máxima é de 25 MHz

0.2.3 (2.0) Para a estrutura V definida abaixo e para a maior frequência utilizável, conte o número de instruções (I) e o tempo de execução (texec) necessário para a execução dos seus procedimentos ORDENA(V) e POLIGONO(V,A), com A=0x00000007. Calcule a CPI média do seu programa.

V: .word 12, 0x00E6001C, 0x00A40052, 0x00A00042, 0x007B006D, 0x00760061, 0x004C007F, 0x003200D3, 0x005C00CB, 0x0080008C, 0x008600A4, 0x00A80068, 0x00AD007D

Dica: O endereço 0xFF20 0700 da memória contém o tempo absoluto do sistema em ms.

0.2.3.1 Solução

Quantidade de instruções é calculado por:

Text Segment				
Bkpt	Address	Code	Basic	Source
	0x00401470	0xffff30313	addi x6,x6,0xffffffff	1735: addi t1, t1, -1 # decrement...
	0x00401474	0xfe0314e3	bne x6,x0,0xffffffff4	1736: bne t1, zero, printIntUnsigned.loop2 # eh o ultimo?
	0x00401478	0x00020023	sb x0,0x00000000(x5)	1737: sb zero, 0(t0) # insere \N...
	0x0040147c	0x0fc0f517	auipc x10,0x0000fc0f	1739: la a0, TempBuffer # Endereco ...
	0x00401480	0xfdc50513	addi x10,x10,0xffff...	
	0x00401484	0xd08ff0ef	jal x1,0xfffffa84	1740: jal printString # chama o p...
	0x00401488	0x00012083	lw x1,0x00000000(x2)	1742: lw ra, 0(sp) # recupera a
	0x0040148c	0x00410113	addi x2,x2,0x00000004	1743: addi sp, sp, 4 # libera es...
	0x00401490	0x000080e7	jalr x0,x1,0x00000000	1744: printIntUnsigned.fim: ret

Figura 2: Quantidade de Instruções

Ultimo endereço menos primeiro endereço dividido por 4: $(4199568 - 4194304)/4 = 1316$ Instruções! (O mesmo do UNICICLO!)

Quantidade de tempo de execução: 127 ms

9	li s0,0xFF200700 # carrega o endereço do contador				
10	lv s1,0(s0) # lê o tempo inicial				
11	jal raio # procedimento que se quer medir o tempo				
12	lv s2,0(s0) # lê o tempo final				
13	sub s0,s2,s1 # calcula o tempo decorrido				
14					
15	raio:				
..					

Figura 3: 0xFF200700

$$CPI = (Tex/I) * 50x10^6/12 \quad (1)$$

$$CPI = (127x10^{-3}/1316) * 4,16x10^6 \quad (2)$$

$$CPI = 4,0145 \quad (3)$$

(aqui nós erramos no ultimo Lab, e colocamos o CPI diferente de 1)