

Organização e Arquitetura de Computadores

Laboratório 5: ULA e FPULA

Gabriel Levi - 16/0006490

Léo Moraes - 16/0011795

Luis Felipe - 16/0071569

Mikael Melo - 16/0015537

Gabriel Lobão - 16/0046424

Gabriel Nunes - 16/0006597

GRUPO 5

¹Dep. Ciência da Computação – Universidade de Brasília (UnB)
CiC 116394 - OAC - Turma A

Questão 2

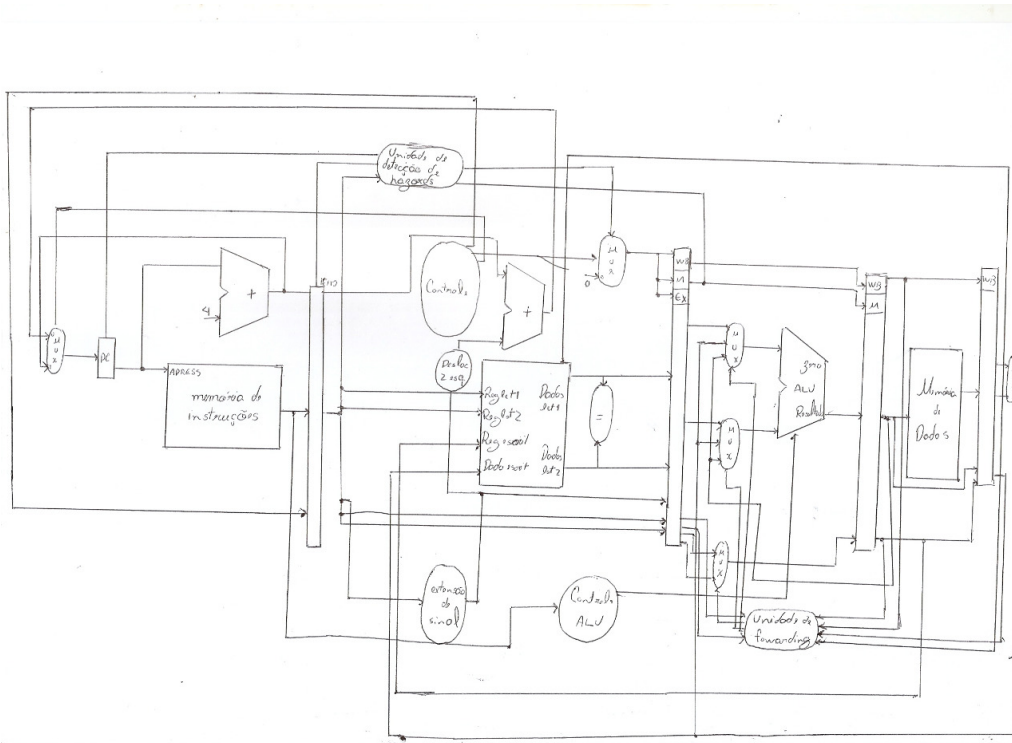


Figure 1. Datapath pipeline

Acima pode-se visualizar um esboço do *Datapath* com os registradores de *pipeline* inclusos.

A seguir, temos a tabela verdade do bloco de controle do pipeline. É uma estrutura muito mais simples se comparado ao controle do multiciclo ou uniciclo, visto que a ULA de ponto flutuante e o Coprocessador 0 (tratamento de exceções e gerenciamento de memória) não estão implementados no pipeline.

Instrução	1	2	3	4	5	6	7	8	9	10	11	12	13	14
OPCRM														
FUNMADD	01	00	0	1	0	0	000	11	0	0	0	0	000	00
FUNMSUB	01	00	0	1	0	0	000	11	0	0	0	0	000	00
FUNMADDU	01	00	0	1	0	0	000	11	0	0	0	0	000	00
FUNMSUBU	01	00	0	1	0	0	000	11	0	0	0	0	000	00
DEFAULT	00	00	0	0	0	0	111	00	0	0	0	0	000	00
OPCLW	00	01	0	1	1	0	000	00	0	0	0	0	LW	00
OPCLH	00	01	0	1	1	0	000	00	0	0	0	0	LH	00
OPCLHU	00	01	0	1	1	0	000	00	0	0	0	0	LHU	00
OPCLB	00	01	0	1	1	0	000	00	0	0	0	0	LB	00
OPCLBU	00	01	0	1	1	0	000	00	0	0	0	0	LBU	00
OPCSW	00	01	0	0	0	1	000	00	0	0	0	0	000	SW
OPCSH	00	01	0	0	0	1	000	00	0	0	0	0	000	SH
OPCSB	00	01	0	0	0	1	000	00	0	0	0	0	000	SB
OPCBEQ	00	00	0	0	0	0	001	01	0	1	0	0	000	00
OPCBNE	00	00	0	0	0	0	101	01	0	0	1	0	000	00
OPCRFMT														
FUNJR	00	00	0	0	0	0	010	10	1	0	0	1	000	00
FUNSYS	10	00	1	1	0	0	100	10	1	0	0	0	000	00
DEFAULT	01	00	0	1	0	0	000	10	0	0	0	0	000	00
OPCJMP	00	00	0	0	0	0	010	00	1	0	0	0	000	00
OPCADDI	00	01	0	1	0	0	000	11	0	0	0	0	000	00
OPCADDIU	00	01	0	1	0	0	000	11	0	0	0	0	000	00
OPCANDI	00	10	0	1	0	0	000	11	0	0	0	0	000	00
OPCXORI	00	10	0	1	0	0	000	11	0	0	0	0	000	00
OPCORI	00	10	0	1	0	0	000	11	0	0	0	0	000	00
OPCIAL	10	00	1	1	0	0	010	00	1	0	0	0	000	00
OPCLUI	00	11	0	1	0	0	000	00	0	0	0	0	000	00
OPCSLTU	00	01	0	1	0	0	000	11	0	0	0	0	000	00
OPCSLTIU	00	10	0	1	0	0	000	11	0	0	0	0	000	00
INST NÃO RECONHECIDA	00	00	0	0	0	0	111	00	0	0	0	0	000	00

Figure 2. Tabela verdade do bloco de controle do pipeline

Legenda:

Registrador	Numero
oRegDst	1
oOrigALU	2
oSavePC	3
oEscreveReg	4
oLeMem	5
oEscreveMem	6
oOrigPC	7
oOpALU	8
oJump	9
oBranch	10
onBranch	11
oJr	12
oLoadType	13
oWriteType	14

Questão 3

Na unidade de Hazard detecta e trata, utilizando bolhas, hazards causados por acesso a memória de uma instrução e a utilização desse resultado logo em seguida, como por exemplo:

lw \$t0,0(\$s0)

add \$t1,\$t0,\$t2

Neste caso, o valor de \$t0 não está pronto quando requisitado pela instrução add, portanto a unidade de hazards mata a instrução add (coloca bolhas no pipeline). A unidade de hazards também trata hazards causados por desvios condicionais e incondicionais (beqs e jumps).

Já a unidade de forward controla os multiplexadores presentes no caminho de dados do pipeline que selecionam de onde vem o dado que vai ser utilizado pelas instruções que possivelmente podem causar hazards, por exemplo:

add \$t0,\$t0,\$t1

sub \$t2,\$t0,\$t2

Neste caso, o dado utilizado pela instrução sub está logo na saída da ULA, então a unidade de forward seleciona a opção em que o operador da ULA é sua última saída, permitindo assim fazer a subtração sem precisar de bolhas no pipeline.

Questão 4

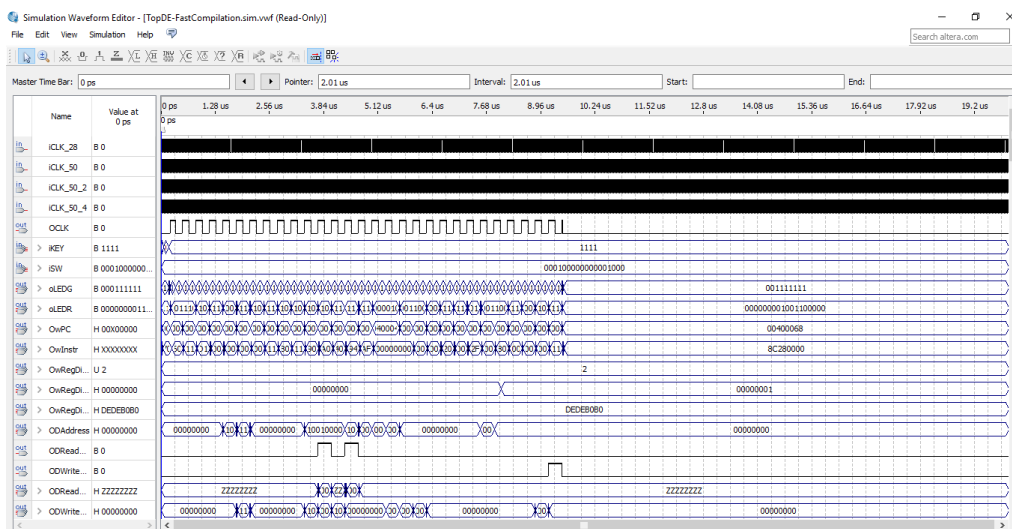
Os testes foram executados na placa utilizando o programa "teste.s", que executa cada instrução implementada e para a execução para que se possa observar o resultado do teste, o registrador \$v0 guarda o valor 1 caso a instrução testada tenha gerado os resultados desejados e -1 caso o contrário, o registrador \$a0 guarda um número que representa a instrução que foi testada de acordo com a tabela .

Para que todas as instruções restantes funcionem o clock máximo é de 12.5 MHz.

Instrução	Numero
add	1
mult	2
sub	3
and	4
or	5
mfhi	6
mflo	7
sll	8
srl	9
slt	10
sgt	11
sra	12
srav	13
sllv	14
srlv	15
xor	16
nor	17
sltu	18
lui	19
div	20
div(resto)	21
mthi	22
mtlo	23
andi	24
xori	25
ori	26
addi	27
sw/lw	28
sb/lb	29
lbu	30
sh/lh	31
lhu	32

Algumas instruções que causam desvio na ordem de execução do código, como jumps ou branches, foram verificadas nos algoritmos de testes das instruções restantes.

Forma de onda do teste.s executada por simulação temporal no QUARTUS:



Vídeo do teste.s sendo executado no Pipeline:

<https://www.youtube.com/watch?v=dnv00oP4R3c>

Questão 5

Vídeo do programa bandeiras.s sendo executado no pipeline:

<https://www.youtube.com/watch?v=HTvUkB4lhVI&feature=youtu.be>

Questão 7

Uniciclo:

O total de elementos lógicos utilizado com a FPULA foi de 25958, 7504 registradores e 858654 bits de memória. Ao retirar a FPULA, o total de elementos lógicos utilizados foi de 17011, 3881 registradores e 853691 bits de memória.

Multiciclo:

O total de elementos lógicos utilizado com a FPULA foi de 27269, 7862 registradores, 858654 bits de memória e o clock maximo é de 3.125 MHz para que a instrução mais demorada, sqrt, funcione corretamente. Ao retirar a FPULA, o total de elementos lógicos utilizados foi de 17892, 4089 registradores, 853691 bits de memória e o clock maximo é de 12.5 MHz para que a instrução mais demorada, div, funcione corretamente.

Pipeline:

O total de elementos lógicos utilizado com a FPULA foi de 25116, 7726 registradores, 858654 bits de memória. Ao retirar a FPULA, o total de elementos lógicos utilizados foi de 16835, 4006 registradores, 853691 bits de memória e o clock maximo é de 12.5 MHz para que a instrução mais demorada, div, funcione corretamente.

Para remover a FPULA basta comentar a oitava linha do arquivo verilog topDE, ”“define FPU”, para sintetizar a FPULA é necessário que está linha não esteja comentada no momento da compilação.

Questão 8

- a) No caminho de dados foi preciso implementar a FPULA, já que a mesma não está implementada no pipeline. Com isso, não foi preciso fazer mais nenhuma modificação.
- b) No bloco de controle foi necessário adicionar os functs das instruções junto ao cvt.w.s, pois usam os mesmos sinais de controle que o cvt.w.s.
- c) Apesar de ter sido implementada a FPULA, o grupo não conseguiu arrumar um bug no qual não grava/mostra o banco de registradores da fpula. Com isso, não foi possível gravar o vídeo e testar a forma de onda. O processador está em anexo a este relatório.