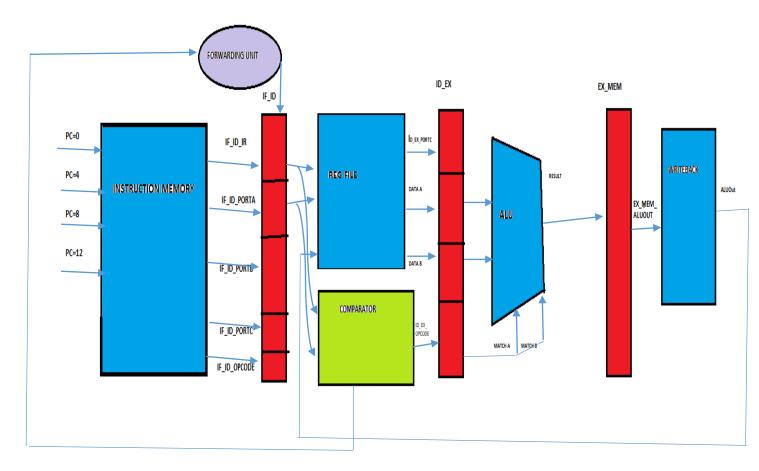
# **MINI PROJECT 2**

# IMPLEMENTATION OF 5 STAGED PIPELINED OF MIPS ISA OVERCOMING DATA HAZARDS

PARTA: IMPLEMENTED USING CONCEPT OF STALLING

•	/tb1/dk	0																								
	/tb1/pc	0044	0000		0004				0008				000c		0010		0014		0018		001c		0020		0024	
1	/tb1/inst	00022d2a	X 00008c		000118a				0001a06				00022d2													
		0004		0001		0002				0003				0004					8							
	/tb1/if_id_ir	2d2a		0c41		18a3				2066				2d2a												
	/tb1/if_id_portC	0b		03		06				08				0b												
1	/tb1/if_id_portA	09	_	02		05				03				09												
1	/tb1/if_id_portB	0a		01		03				06				0a												
1	/tb1/enableportA	1																								
1	/tb 1/enableportB	1																								
1	/tb1/enableportC	1																								
1	/tb1/portC_addr	0b			03		06				08				0b											
1	/tb1/portA_addr	09			02		05				03				09											
1	/tb1/portB_addr	0a			01		03				06				0a											
1	/tb1/dataA	ffff765e			0000003		000000	78			000000	0			ffff765e	Ŷ.										
1	/tb1/dataB	1fff756f			0000002	В	000000	00							1fff756											
	/tb1/dataC	00008880											0000006	4			0000001	4			fffffff		0000888	0		
	/tb1/matchA	0												**				<i>10</i>					0.000000			
j	/tb1/matchB	0																								
_	A STATE OF THE STA	0																								
1	/tb1/id_ex_portC	0b				03		00		06		100		08		)0b										
	/tb1/id_ex_A	ffff765e				0000003	c	000000	00	000000	78	000000	0			ffff765e										
	/tb1/id_ex_B	1fff756f				0000002	8	000000	00							1fff756f										
		0004				0001		0000		0002		10000		0003		0004										
	/tb1/result	00008880					000000				1000000	4			HHHH		0000888	0								
4								000000	64			1000000	4			fffffff		0000888	0							
	/tb1/ALUout	00008880						33330	0000006	4		20000	0000001	4			fffffff	THE OWNER OF THE OWNER,	0000888	0						
_	/tb1/comparematch	0							000000				3333001						5550000							
_	/ a specimpure motor																									
10	Now	200 ps	10 ps	11111	liiiil Ops	1111	ps	1111	) ps	1111	ps	60	1111	1111	) ps	1111	DS DS	90	1111	100	1111	110	1111	120	TITL	130

#### **SIMULATION**

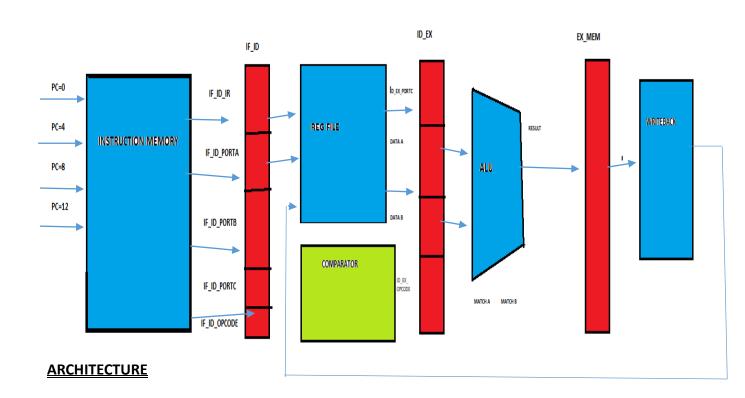


## **ARCHITECTURE**

### PARTB: <u>IMPLEMENTED USING CONCEPT OF DATA FORWARDING</u>

	/tb2/dk	0																	
<b>⊕</b> ♦	/tb2/pc	004c		0000	0004	00	108	000c		0010		0014	0018		001c	0020	0024	0028	002c
⊕ ♦	/tb2/inst	00022d2a	Xxxxxxxx	00008c41	000118a	3 00	01a066	000220	12a										
<b>.</b> ♦	/tb2/if_id_opcode	0004		0001		0002	0003		0004										
± 🔷	/tb2/if_id_ir	2d2a		0c41		18a3	2066		2d2a										
<b>±</b> ♦	/tb2/if_id_portC	0b		03		06	08		0b										
<b>±</b> ♦	/tb2/if_id_portA	09		02		05	03		09										
<b>±</b> ♦	/tb2/if_id_portB	0a		01		03	06		0a										
-	/tb2/enableportA	1																	
1	/tb2/enableportB	1																	
-	/tb2/enableportC	1																	
<b>⊕</b>	/tb2/portC_addr	0b			03	06	;	08		0b									
± 🔷	/tb2/portA_addr	09			02	0:		03		09									
<b>⊕</b> ♦	/tb2/portB_addr	0a			01	(03	3	06		0a									
	/tb2/dataA	ffff765e			0000003	c (00	1000078	000000	000	ffff765e									
	/tb2/dataB	1fff756f			0000002	8 00	000000			1fff756f									
	/tb2/dataC	00008880								0000006	4	00000014	(FIFTIFFE		00008880				
	/tb2/id_ex_portC	0b				03	06		08		0b								
_	/tb2/id_ex_A	ffff765e				0000003c	00000	078	000000	do	ffff765e								
	/tb2/id_ex_B	1fff756f				00000028	00000	000			1fff756	f							
	/tb2/matchA	0																	
_	/tb2/matchB	0																	
± •	/tb2/id_ex_opcode	0004				0001	0002		0003		0004								
	/tb2/result	00008880					000064	000000	)14	fffffff		00008880							
		00008880					00000		000000		(((((((		088800						
	/tb2/ALUout	00008880						000000	064	0000001	4	(fffffff	(000088	80					

#### **SIMULATION**



# **Sheet for uploading**

Fill the following table:

To detect data hazard using data forwarding and data stalling.

Memory	Instructions	Register status (Show the registers which are
Address		only affected). Remarks on hazards and what
		is the solution?
		() → represents Decimal
0000	ADD R3 , R2, R1	R2 = 3C (60), R1= 28 (40)
		R3 = 64 ( 100 )
		First instruction doesn't have any data hazard.
0004	SUB R6 , R5 ,R3	$R5 = 78 \ (120)$ , $R3 = 0(0)$ [Initially stored as 0 in the register file and later overwritten by the previous inst. Destination register),
		R6 = 14 (20)
		RAW hazard found at R3 register with respect to first instruction.
		R3 =0 , R6 =0 (Initially)
		Overwritten as $R3 = 64(100)$ , $R6 = 14(20)$
0008	NAND R8 ,R3 , R6	RAW hazard found at R6 register and R3 register with respect to second instruction.
		R8 = 0xFFFFFFF.
		R9 = 0xFFFF765E, $R10 = 0x1FFF756F$ ,
		No data hazard found destination register
000C	NOR R11 , R9 , R10	R11 = 0x0008880.

#### **Honor Code Declaration by student:**

- My answers to the above questions are my own work.
- I have not shared the codes/answers written by me with any other students. (I might have helped clear doubts of other students).
- I have not copied other's code/answers to improve my results. (I might have got some doubts cleared from other students).

Name: VARUN KUMAR S Date: 25/05/2020

ID No.: 2019H1400539G