ADVANCED COMPUTER ARCHITECTURE **MARCH 5, 2015**
Problem Set #6: MIPS64 Pipeline Execution (Control Hazard) **Prof RLUy**

I. MIPS pipeline execution tracing (Control Hazard): Shown below is a snapshot of the memory and some registers.
1.) Show the instruction format; 2.) Show the pipeline map; 3.) Show the contents of the internal **pipeline** registers after every cycle for all the instructions. Assume: PC = 0000000000000000 and COND= 0

	Before execution	After exectuion
100F	55	
100E	44	
100D	33	
100C	22	
100B	11	
100A	EF	
1009	CD	
1008	AB	
1007	89	
1006	67	
1005	45	
1004	23	
1003	01	
1002	EF	
1001	CD	
1000	AB	

R1 :	0000 0000 0000 0002	R2	0000 0000 0000 0008	R3	0000 0000 0000 0004	R4	0000 0000 0000 0005
R5 :	0000 0000 0000 0008	R6	0000 0000 0000 0001	R7	0000 0000 0000 0000	R8	0000 0000 0000 0004

Addr	Instruction	Opcode (Hex)	$IR_{0..5}$	$IR_{6..10}$	$IR_{11..15}$	$IR_{16.31}$
00	BEQ R0, R0, L1					
04	AND R0, R0, R0					
08	OR R0, R0, R0					
0C	XOR R0, R0, R0					
10	DSUBU R0, R0, R0					
14	L1: DADDU R9, R1, R3					

***Pipeline Map (Pipeline flush)**

Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
BEQ R0, R0, L1																
AND R0, R0, R0																
OR R0, R0, R0																
XOR R0, R0, R0																
DSUBU R0, R0, R0																
L1: DADDU R9, R1, R3																

***Pipeline Map (Pipeline freeze)**

Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
BEQ R0, R0, L1																
AND R0, R0, R0																
OR R0, R0, R0																
XOR R0, R0, R0																

DSUBU R0, R0, R0																
L1: DADDU R9, R1, R3																

***Pipeline Map (Predict not taken)**

Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
BEQ R0, R0, L1																
AND R0, R0, R0																
OR R0, R0, R0																
XOR R0, R0, R0																
DSUBU R0, R0, R0																
L1: DADDU R9, R1, R3																

***Pipeline Map (Modified pipeline, i.e., pipeline #2)**

Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
BEQ R0, R0, L1																
AND R0, R0, R0																
OR R0, R0, R0																
XOR R0, R0, R0																
DSUBU R0, R0, R0																
L1: DADDU R9, R1, R3																

Cycle 1			**Cycle 2**		
IF	IF/ID.IR =		IF	IF/ID.IR =	
	IF/ID.NPC =			IF/ID.NPC =	
	PC =			PC =	
ID	ID/EX.A =		ID	ID/EX.A =	
	ID/EX.B =			ID/EX.B =	
	ID/EX.IMM =			ID/EX.IMM =	
	ID/EX.IR =			ID/EX.IR =	
	ID/EX.NPC=			ID/EX.NPC=	
EX	$EX/MEM.ALU_{Output}$ =		EX	$EX/MEM.ALU_{Output}$ =	
	EX/MEM.COND =			EX/MEM.COND =	
	EX/MEM.IR =			EX/MEM.IR =	
	EX/MEM.B =			EX/MEM.B =	
MEM	MEM/WB.LMD =		MEM	MEM/WB.LMD =	
	Range of memory locations affected =			Range of memory locations affected =	
	MEM/WB.IR =			MEM/WB.IR =	
	$MEM/WB.ALU_{OUTPUT}$ =			$MEM/WB.ALU_{OUTPUT}$ =	
WB	R_n =		WB	R_n =	
Cycle 3			**Cycle 4**		
IF	IF/ID.IR =		IF	IF/ID.IR =	
	IF/ID.NPC =			IF/ID.NPC =	
	PC =			PC =	
ID	ID/EX.A =		ID	ID/EX.A =	
	ID/EX.B =			ID/EX.B =	
	ID/EX.IMM =			ID/EX.IMM =	
	ID/EX.IR =			ID/EX.IR =	
	ID/EX.NPC=			ID/EX.NPC=	
EX	$EX/MEM.ALU_{Output}$ =		EX	$EX/MEM.ALU_{Output}$ =	
	EX/MEM.COND =			EX/MEM.COND =	
	EX/MEM.IR =			EX/MEM.IR =	
	EX/MEM.B =			EX/MEM.B =	
MEM	MEM/WB.LMD =		MEM	MEM/WB.LMD =	
	Range of memory locations affected =			Range of memory locations affected =	
	MEM/WB.IR =			MEM/WB.IR =	
	$MEM/WB.ALU_{OUTPUT}$ =			$MEM/WB.ALU_{OUTPUT}$ =	

WB	R_n =		WB	R_n =	
Cycle 5			**Cycle 6**		
IF	IF/ID.IR =		IF	IF/ID.IR =	
	IF/ID.NPC =			IF/ID.NPC =	
	PC =			PC =	
ID	ID/EX.A =		ID	ID/EX.A =	
	ID/EX.B =			ID/EX.B =	
	ID/EX.IMM =			ID/EX.IMM =	
	ID/EX.IR =			ID/EX.IR =	
	ID/EX.NPC=			ID/EX.NPC=	
EX	EX/MEM.ALUOutput =		EX	EX/MEM.ALUOutput =	
	EX/MEM.COND =			EX/MEM.COND =	
	EX/MEM.IR =			EX/MEM.IR =	
	EX/MEM.B =			EX/MEM.B =	
MEM	MEM/WB.LMD =		MEM	MEM/WB.LMD =	
	Range of memory locations affected =			Range of memory locations affected =	
	MEM/WB.IR =			MEM/WB.IR =	
	MEM/WB.ALUOUTPUT =			MEM/WB.ALUOUTPUT =	
WB	Rn =		WB	Rn =	
Cycle 7			**Cycle 8**		
IF	IF/ID.IR =		IF	IF/ID.IR =	
	IF/ID.NPC =			IF/ID.NPC =	
	PC =			PC =	
ID	ID/EX.A =		ID	ID/EX.A =	
	ID/EX.B =			ID/EX.B =	
	ID/EX.IMM =			ID/EX.IMM =	
	ID/EX.IR =			ID/EX.IR =	
	ID/EX.NPC=			ID/EX.NPC=	
EX	EX/MEM.ALUOutput =		EX	EX/MEM.ALUOutput =	
	EX/MEM.COND =			EX/MEM.COND =	
	EX/MEM.IR =			EX/MEM.IR =	
	EX/MEM.B =			EX/MEM.B =	
MEM	MEM/WB.LMD =		MEM	MEM/WB.LMD =	
	Range of memory locations affected =			Range of memory locations affected =	
	MEM/WB.IR =			MEM/WB.IR =	

	MEM/WB.ALUOUTPUT =			MEM/WB.ALUOUTPUT =	
WB	Rn =		WB	Rn =	

Cycle 9			**Cycle 10**		
IF	IF/ID.IR =		IF	IF/ID.IR =	
	IF/ID.NPC =			IF/ID.NPC =	
	PC =			PC =	
ID	ID/EX.A =		ID	ID/EX.A =	
	ID/EX.B =			ID/EX.B =	
	ID/EX.IMM =			ID/EX.IMM =	
	ID/EX.IR =			ID/EX.IR =	
	ID/EX.NPC=			ID/EX.NPC=	
EX	EX/MEM.ALUOutput =		EX	EX/MEM.ALUOutput =	
	EX/MEM.COND =			EX/MEM.COND =	
	EX/MEM.IR =			EX/MEM.IR =	
	EX/MEM.B =			EX/MEM.B =	
MEM	MEM/WB.LMD =		MEM	MEM/WB.LMD =	
	Range of memory locations affected =			Range of memory locations affected =	
	MEM/WB.IR =			MEM/WB.IR =	
	MEM/WB.ALUOUTPUT =			MEM/WB.ALUOUTPUT =	
WB	Rn =		WB	Rn =	
Cycle 11			**Cycle 12**		
IF	IF/ID.IR =		IF	IF/ID.IR =	
	IF/ID.NPC =			IF/ID.NPC =	
	PC =			PC =	
ID	ID/EX.A =		ID	ID/EX.A =	
	ID/EX.B =			ID/EX.B =	
	ID/EX.IMM =			ID/EX.IMM =	
	ID/EX.IR =			ID/EX.IR =	
	ID/EX.NPC=			ID/EX.NPC=	
EX	EX/MEM.ALUOutput =		EX	EX/MEM.ALUOutput =	
	EX/MEM.COND =			EX/MEM.COND =	
	EX/MEM.IR =			EX/MEM.IR =	
	EX/MEM.B =			EX/MEM.B =	
MEM	MEM/WB.LMD =		MEM	MEM/WB.LMD =	

	Range of memory locations affected =			Range of memory locations affected =	
	MEM/WB.IR =			MEM/WB.IR =	
	MEM/WB.ALUOUTPUT =			MEM/WB.ALUOUTPUT =	
WB	Rn =		WB	Rn =	

Cycle 13			**Cycle 14**		
IF	IF/ID.IR =		IF	IF/ID.IR =	
	IF/ID.NPC =			IF/ID.NPC =	
	PC =			PC =	
ID	ID/EX.A =		ID	ID/EX.A =	
	ID/EX.B =			ID/EX.B =	
	ID/EX.IMM =			ID/EX.IMM =	
	ID/EX.IR =			ID/EX.IR =	
	ID/EX.NPC=			ID/EX.NPC=	
EX	EX/MEM.ALUOutput =		EX	EX/MEM.ALUOutput =	
	EX/MEM.COND =			EX/MEM.COND =	
	EX/MEM.IR =			EX/MEM.IR =	
	EX/MEM.B =			EX/MEM.B =	
MEM	MEM/WB.LMD =		MEM	MEM/WB.LMD =	
	Range of memory locations affected =			Range of memory locations affected =	
	MEM/WB.IR =			MEM/WB.IR =	
	MEM/WB.ALUOUTPUT =			MEM/WB.ALUOUTPUT =	
WB	Rn =		WB	Rn =	
Cycle 15			**Cycle 16**		
IF	IF/ID.IR =		IF	IF/ID.IR =	
	IF/ID.NPC =			IF/ID.NPC =	
	PC =			PC =	
ID	ID/EX.A =		ID	ID/EX.A =	
	ID/EX.B =			ID/EX.B =	
	ID/EX.IMM =			ID/EX.IMM =	
	ID/EX.IR =			ID/EX.IR =	
	ID/EX.NPC=			ID/EX.NPC=	
EX	EX/MEM.ALUOutput =		EX	EX/MEM.ALUOutput =	
	EX/MEM.COND =			EX/MEM.COND =	
	EX/MEM.IR =			EX/MEM.IR =	
	EX/MEM.B =			EX/MEM.B =	

MEM	MEM/WB.LMD =		MEM	MEM/WB.LMD =	
	Range of memory locations affected =			Range of memory locations affected =	
	MEM/WB.IR =			MEM/WB.IR =	
	MEM/WB.ALUOUTPUT =			MEM/WB.ALUOUTPUT =	
WB	Rn =		WB	Rn =	