A Five Stage Pipelined Mini MIPS Processor

Ram Gopal Srikar Katakam, Rishabh Singh Thakur, Ejiroghene Omojimite, Akhil Chowdary Movva

UNIVERSITÉ Concordia UNIVERSITY

Field Size	6 bits	5 bits	5 bits	5 bits	5 bits	6 bits	Comment
R - Format	Op	Rs	Rt	Rd	ShAmt	Funct	Arithmetic Instruction Format
I - Format	Op	Rs	Rt	Address/Immediate			Branch, Immediate Format
J - Format	Op	Target Address					Jump Instruction Format

	Field Size					
	6 bits	5 bits	5 bits	5 bits	5 bits	6 bits
	Opcode	Rs	Rt	Rd	ShAmt	Funct
INSTRUCTION	[31:26]	[25:21]	[20:16]	[15:11]	[10:6]	[5:0]
ADD	000001	Rs	Rt	Rd	00000	100000
SUB	000001	Rs	Rt	Rd	00000	100010
AND	000001	Rs	Rt	Rd	00000	100100
XOR	000001	Rs	Rt	Rd	00000	101000
JR	000001	Rs	00000	00000	00000	000000

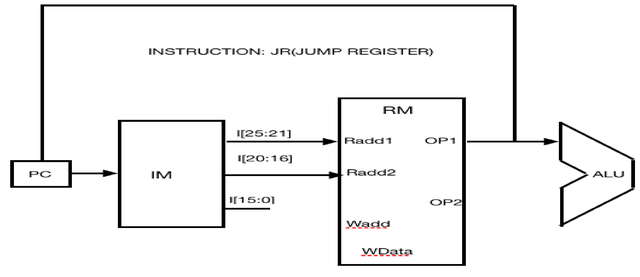
Table 3: Mapping of Specified R-type Instruction Set into bits

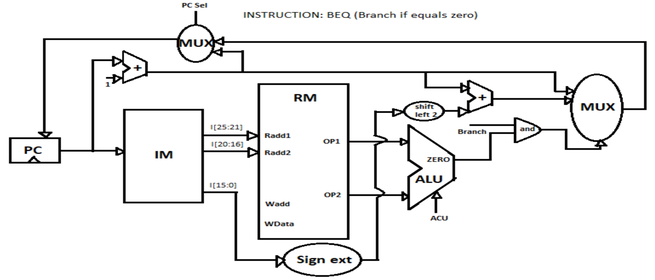
	Field Size					
	6 bits	5 bits	5 bits	16 bits		
	Opcode	Rs	Rt	Rd	ShAmt	Funct
INSTRUCTION	[31:26]	[25:21]	[20:16]	[15:0]		
ANDI	001100	Rs	Rt	Immediate		
SUBI	001001	Rs	Rt	Immediate		
LW	100011	Rs	Rt	2's Complement Constant		
SW	101011	Rs	Rt	2's Complement Constant		
BEQ	000100	Rs	Rt	2's Complement Constant		
ADDI	001000	Rs	Rt	Immediate		

Table 4: Mapping of Specified I-type Instruction Set into bits

No	Instruction	Name	Type	Syntax	Description
1	ADD	Add	R	ADD $D, $S, $T	Adds content of two Registers and stores the result into Registers.
2	SUB	Subtract	R	SUBI $D, $S, $T	Subtracts the contents of the two Registers and stores the result into Register
3	ANDI	AND Immediate	I	ANDI $D, $S, #i	Perform AND operation of content of Register and the sign extended value and stores the result into Register
4	SUBI	Subtract Immediate	I	SUBI $D, $S, #i	Subtracts the contents of the Register and sign extended value and stores the result into Register
5	AND	Logical AND	R	AND $D, $S, $T	Performs Logical AND operation of the contents of two Registers and stores the result in Register
6	XOR	Logical XOR	R	XOR $D, $S, $T	Performs Logical XOR operation of the contents of two Registers XOR stores the result in Register
7	BEQ	Branch equal	I	BEQ $D, $S, Label	Compares the content of two Registers and if equal, increment PC to PC+4+offset else increment to PC+4
8	JR	Jump Register	R	JR $D	Jump to the address stored in the specified Register
9	LW	Load Word	I	LW $D, $S, #i	Load Data from Data Memory into Register
10	SW	Store Word	I	SW $D, $S, #i	Store the Register content to Data Memory

Table 2: Description of specified ISA





Opcode	Rs	Rt	Rd	ShAnt	Function
			Immediate		
00 0100	0 1100	0 1001	0 0000 0000 0000 0111		
stall					
10 0011	0 1011	0 1100	0 0000 0000 0000 0000		
10 0011	0 1011	0 1101	0 0000 0000 0000 0001		
stall					
stall					
stall					
stall					
00 0001	0 1100	0 1101	0 1110	0 0000	10 0000
stall					
stall					
stall					
stall					
10 1011	0 1011	0 1110	0 0000 0000 0000 0001		
00 0001	0 1100	0 0001	0 1100	0 0000	10 0000
stall					
stall					
stall					
stall					
10 1011	0 1011	0 1100	0 0000 0000 0000 0000		
00 0001	0 0000	0 0000	0 0000	0 0000	01 1111
stall					

Table 7: Machine code