MIPS Instruction Reference List
Connor Baker, January 20th 2017

Core Instruction Set

Name	Mnemonic	Format	Operation (in Verilog)	OPCODE/Funct (Hex)
Add	`add`	R	R[rd]=R[rs]+R[rt]	(1) $0/20_{16}$
Add Immediate	`addi`	I	R[rt]=R[rs]+SigExtImm	$(1,2)$ 8_{16}
Add Imm. Unsigned	`addiu`	I	R[rt]=R[rs]+SigExtImm	(2) 9_{16}
Add Unsigned	`addu`	R	R[rd]=R[rs]+R[rt]	$0/21_{16}$
And	`and`	R	R[rd]=R[rs]&R[rt]	(1) $0/24_{16}$
And Immediate	`andi`	I	R[rt]=R[rs]&ZeroExtImm	(3) C_{16}
Branch On Equal	`beq`	I	if(R[rs]==R[rt]) PC=PC+4+BranchAddr	(4) 4_{16}
Branch On Not Equal	`bne`	I	if(R[rs]!=R[rt]) PC=PC+4+BranchAddr	(4) 5_{16}
Jump	`j`	J	PC=JumpAddr	(5) 2_{16}
Jump And Link	`jal`	J	R[31]=PC+8; PC=JumpAddr	(5) 3_{16}
Jump Register	`jr`	R	PC=R[rs]	$0/08_{16}$
Load Byte Unsigned	`lbu`	I	R[rt]={24'b0,M[R[rs]] +SignExtImm](7:0)}	(2) 24_{16}

Name	`Mnemonic`	Format	Operation (in Verilog)	OPCODE/Funct (Hex)
Load Halfword Unsigned	`lhu`	I	R[rt]={16'b0,M[R[rs]] +SignExtImm](15:0)}	(2) 25_{16}
Load Linked	`ll`	I	R[rt]=M[R[rs]] +SignExtImm]	$(2,7)$ 30_{16}
Load Upper Imm.	`lui`	I	R[rt]={imm, 16'b0}	F_{16}
Load Word	`lw`	I	R[rt]=M[R[rs] +SignExtImm]	(2) 23_{16}
Nor	`nor`	R	R[rd]=$\sim$ ($R[rs]$ | $R[rt]$)	$0/27_{16}$
Or	`or`	R	R[rd]=(R[rs] | R[rt])	$0/25_{16}$
Or Immediate	`ori`	I	R[rt]=R[rs] | ZeroExtImm	(3) D_{16}
Set Less Than	`slt`	R	R[rd]=(R[rs]<R[rt])? 1 : 0	$0/2A_{16}$
Set Less Than Imm.	`slti`	I	R[rt]=(R[rs]<SignExtImm) ? 1 : 0	(2) A_{16}
Set Less Than Imm. Unsigned	`sltiu`	I	R[rt]=(R[rs]<SignExtImm) ? 1 : 0	$(2,6)$ B_{16}
Set Less Than Unsigned	`sltu`	R	R[rd]=(R[rs]<R[rt]) ? 1 : 0	(6) $0/2B_{16}$
Shift Left Logical	`sll`	R	R[rd]=R[rt]<<shamt	$0/00_{16}$
Shift Right Logical	`srl`	R	R[rd]=R[rt]>>>shamt	$0/02_{16}$