

Instruction Description	Current State cur_cycle	opcode	Inputs subfunc zero	Next State next_cycle halt	mem_rd	mem_wrt	ldPC	ldMAR	ldIR	ldMDSR	ldMDLR	ldALUout	Outputs ldReg wreg_addr_mux pc_mux	addr_mux	a_mux	b_mux	reg_write_mux	func
All Default outputs	X	X	X X X	X false	false	false	false	false	false	false	false	false	false WA_RD PC_ALU	ADDR_PC	A_PC	B_REG	RW_ALU_OUT	ADD
Fetch instruction IR=Mem[PC] PC=PC+4	0	X	X X X	1 (false)	true	(false)	true	(false)	true	(false)	(false)	(false)	(false) X (PC_ALU)	(ADDR_PC)	(A_PC)	B_4	X	(ADD)
halt	1	0x0	0xc X	X true	(false)	(false)	(false)	(false)	(false)	(false)	(false)	(false)	X X X X X X X					
Set PC to jump address PC=PC[31:28] IR[25:0]<2	1	0x2	X X X	0 (false)	(false)	(false)	true	(false)	(false)	(false)	(false)	(false)	X PC_IMM	X X X X X X				
Set ALUout to addition result A_reg=Reg[IR[25-21]] alu=A_reg+sign_extend(IR[15-0]) ALUout=alu	1	0x8	X X X	2 (false)	(false)	(false)	(false)	(false)	(false)	(false)	(false)	true	(false) X X X A_REG	B_IR_16	X		(ADD)	
Store ALUout to Register File Reg[IR[20-16]]=ALUout	2	0x8	X X X	0 (false)	(false)	(false)	(false)	(false)	(false)	(false)	(false)	(false)	true WA_RT	X X X X X RW_ALU_OUT	X			
Set ALUout to addition result A_reg=Reg[IR[25-21]] B_reg = Reg[IR[20-16]] alu=A_reg+B_reg ALUout=alu	1	0x0	0x20 X	2 (false)	(false)	(false)	(false)	(false)	(false)	(false)	(false)	true	(false) X X X A_REG	(B_REG)	X		(ADD)	
Store ALUout to Register File Reg[IR[15-11]]=ALUout	2	0x0	0x20 X	0 (false)	(false)	(false)	(false)	(false)	(false)	(false)	(false)	(false)	true WA_RD	X X X X X RW_ALU_OUT	X			
Set ALUout to addition result A_reg=Reg[IR[25-21]] B_reg = Reg[IR[20-16]] alu=A_reg+B_reg ALUout=alu	1	0x0	0x22 X	2 (false)	(false)	(false)	(false)	(false)	(false)	(false)	(false)	true	(false) X X X A_REG	(B_REG)	X		SUB	
Store ALUout to Register File Reg[IR[15-11]]=ALUout	2	0x0	0x22 X	0 (false)	(false)	(false)	(false)	(false)	(false)	(false)	(false)	(false)	true WA_RD	X X X X X RW_ALU_OUT	X			
Set MAR to Memory Address which is the addition result A_REG = Reg[IR[25-21]] alu = A_REG + sign_extend(IR[15-0]) MAR = Reg[20-16] MDSR = Reg[20-16]	1	0x23	X X X	2 (false)	(false)	(false)	(false)	true	(false)	(false)	(false)	(false)	X X X A_REG	B_IR_16	X		(ADD)	
Store the word from R1 to the MDLR = Mem[MAR]	2	0x23	X X X	3 (false)	true	(false)	(false)	(false)	(false)	(false)	(false)	(false)	X X ADDR_MEM	X X X X X				
Reg[15-0] = MDLR	3	0x23	X X X	0 (false)	(false)	(false)	(false)	(false)	(false)	(false)	(false)	(false)	true WA_RT	X X X X RW_MEM	X			
Set MAR to Memory Address which is the addition result A_REG = Reg[IR[25-21]] alu = A_REG + sign_extend(IR[15-0]) MAR = alu Store the word from R1 into MDSR from R1 MDSR = Reg[20-16]	1	0x2b	X X X	2 (false)	(false)	(false)	(false)	true	(false)	true	(false)	(false)	X X X A_REG	B_IR_16	X		(ADD)	
Store word from MDSR into memory Mem[MAR] = MDSR	2	0x2b	X X X	0 (false)	(false)	(false)	true	(false)	(false)	(false)	(false)	(false)	X X ADDR_MAR	X X X X X				
Subtract A_REG = REG[IR[25-21]] and B_REG = REG[IR[20-15]] alu = A_REG - B_REG Zero will be set to either to true or false depending on the subtraction	1	0x4	X X X	2 (false)	(false)	(false)	(false)	(false)	(false)	(false)	(false)	(false)	X X X A_REG	B_REG	X		(SUB)	
Branch taken as numbers are equal PC is set to to branch the address. Then A_REG = PC B_REG = sign_extend(IR[15-0])<2 alu = A_REG - B_REG PC=alu	2	0x4	X true	0 (false)	(false)	(false)	(false)	true	(false)	(false)	(false)	(false)	X (PC_ALU)	X A_PC	B_IR_16	X X	(ADD)	
Branch not taken.	2	0x4	X false	0 (false)	(false)	(false)	(false)	(false)	(false)	(false)	(false)	(false)	X X X X X					
Subtract A_REG = REG[IR[25-21]] and B_REG = REG[IR[20-15]] alu = A_REG - B_REG Zero will be set to either to true or false depending on the subtraction	1	0x5	X X X	2 (false)	(false)	(false)	(false)	(false)	(false)	(false)	(false)	(false)	X X X A_REG	B_REG	X		(SUB)	
Branch taken as numbers are not equal PC is set to to branch the address. Then A_REG = PC B_REG = sign_extend(IR[15-0])<2 alu = A_REG - B_REG PC=alu	2	0x5	X false	0 (false)	(false)	(false)	(false)	true	(false)	(false)	(false)	(false)	X (PC_ALU)	X A_PC	B_IR_16	X X	(ADD)	
Branch not taken.	2	0x5	X true	0 (false)	(false)	(false)	(false)	(false)	(false)	(false)	(false)	(false)	X X X X X					