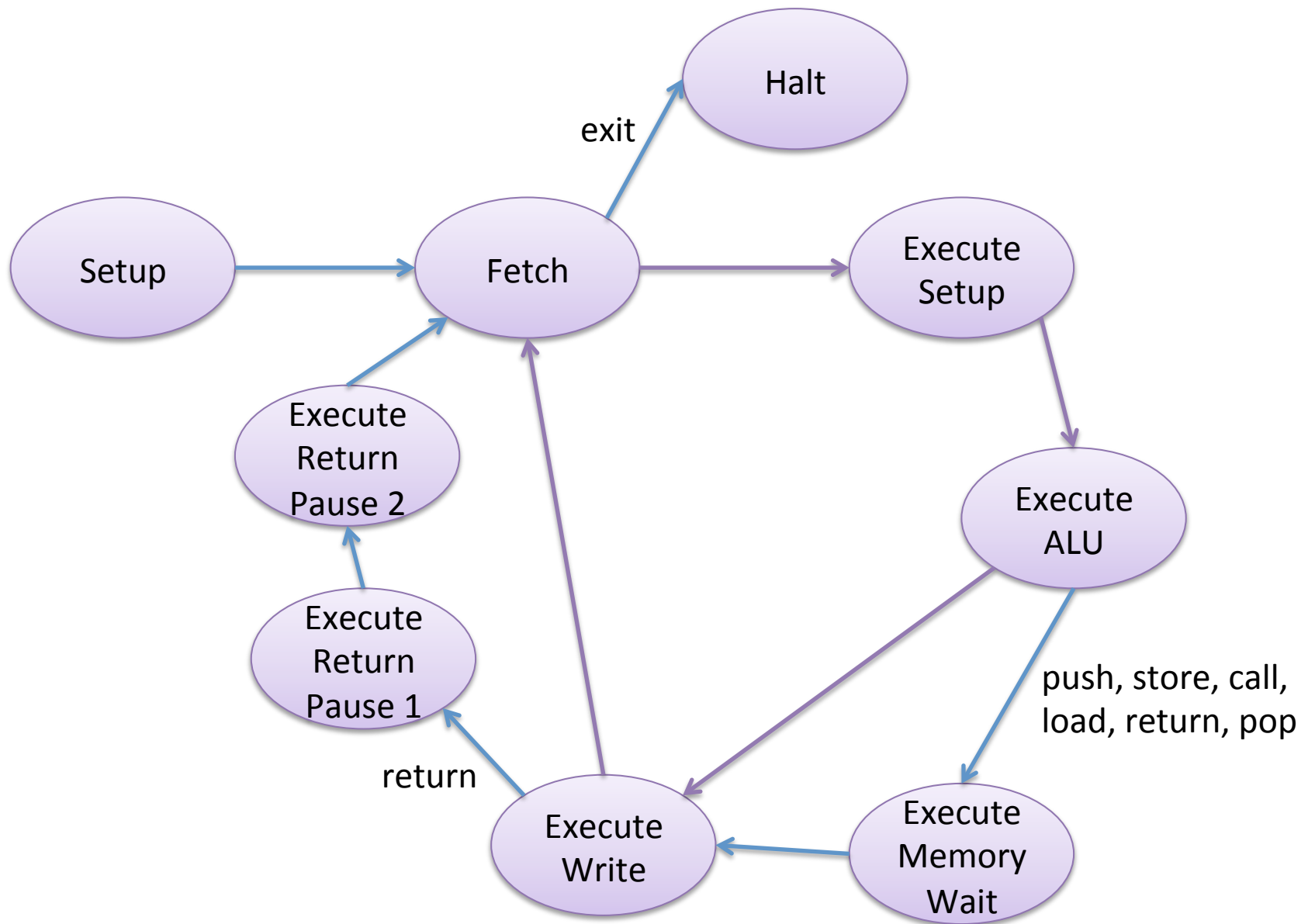




	Fetch	Execute Setup	Execute ALU	Execute MemWait	Execute Write	Execute Return Pause1	Execute Return Pause2
Load from RAM	IR gets ROM_output PC = PC+1	MAR=upper bits of IR, possibly added to RE	wait for RAM	wait for RAM	RA, RB, RC, RD, RE, or SP gets RAM_output	NOT USED	NOT USED
Store to RAM	IR gets ROM_output PC = PC+1	MAR=upper bits of IR, possibly added to RE MBR=RA, RB, RC, RD, RE, or SP	WE=1	wait for RAM	WE=0 wait for RAM	NOT USED	NOT USED
Unconditional Branch	IR gets ROM_output PC = PC+1	PC gets lower bits from IR	wait for ROM to find instruction at new PC	NOT USED	wait for ROM to find instruction at new PC	NOT USED	NOT USED
Conditional Branch	IR gets ROM_output PC = PC+1	If specified CR bit is 1, then PC gets lower bits from IR	wait for ROM to find instruction at new PC	NOT USED	wait for ROM to find instruction at new PC	NOT USED	NOT USED

	Fetch	Execute Setup	Execute ALU	Execute MemWait	Execute Write	Execute Return Pause1	Execute Return Pause2
Call	IR gets ROM_output PC = PC+1	PC gets lower bits from IR MAR = SP SP = SP+1 MBR gets CR and PC	WE=1	wait for RAM	WE=0 wait for RAM	NOT USED	NOT USED
Return	IR gets ROM_output PC = PC+1	MAR = SP SP = SP-1	wait for RAM	wait for RAM	PC and CR get values from RAM_output	wait for ROM to find instruction at new PC	wait for ROM to find instruction at new PC
Push	IR gets ROM_output PC = PC+1	MAR = SP SP = SP+1 MBR=RA, RB, RC, RD, RE, or SP	WE=1	wait for RAM	WE=0 wait for RAM	NOT USED	NOT USED
Pop	IR gets ROM_output PC = PC+1	MAR = SP-1 SP = SP - 1	wait for RAM	wait for RAM	RA, RB, RC, RD, RE, or SP gets RAM_output	NOT USED	NOT USED

	Fetch	Execute Setup	Execute ALU	Execute MemWait	Execute Write	Execute Return Pause1	Execute Return Pause2
Store to Output	IR gets ROM_output PC = PC+1	Wait or skip this state	Wait or skip this state	NOT USED	Output gets RA, RB, RC, RD, RE, SP, PC, or IR	NOT USED	NOT USED
Load from Input	IR gets ROM_output PC = PC+1	Wait or skip this state	Wait or skip this state	NOT USED	RA, RB, RC, RD, RE, or SP gets input port	NOT USED	NOT USED
Add, Subtract, And, Or, Xor, Shift, Rotate	IR gets ROM_output PC = PC+1	SRCA gets RA, RB, RC, RD, RE, 0's, or 1's SRCB gets RA, RB, RC, RD, RE, 0's, or 1's OP = IR bits	Wait for ALU to do operation	NOT USED	RA, RB, RC, RD, RE, or SP gets DEST (ALU output) CR gets cond (ALU flags)	NOT USED	NOT USED
Move	IR gets ROM_output PC = PC+1	SRCA gets RA, RB, RC, RD, RE, PC, or IR OP = "111"	Wait for data to pass through ALU	NOT USED	RA, RB, RC, RD, RE, or SP gets DEST	NOT USED	NOT USED