Type	Mne-monic	Assembly Example	Machine Code								Operation Summary	Detailed Description
			OPR (Upper)				OPA (Lower)					
Jump Unconditional	**JUN**	jun LABEL jun 0x123	0	1	0	0	A3				●PC+1→PCTEMP ●PCTEMP→PC	This is an unconditional branch instruction. The A3, A2, and A1 fields from the instruction word are transferred to the program counter (PC).
			A2				A1				●PC+1→PCTEMP ●A3→PC[11:8] ●A2→PC[7:4] ●A1→PC[3:0]	
Jump to Subroutine	**JMS**	jms LABEL jms 0x123	0	1	0	1	A3				●PC+1→PCTEMP ●PCTEMP→PC	This is a subroutine call instruction. The PC is first incremented by 2 to point to the instruction following the call, and this return address is stored in the stack. The stack pointer (SP) is incremented by 1, and the newly selected stack level becomes the active PC. The A3, A2, and A1 fields from the instruction word are then transferred to that PC. Since the stack has four levels, the JMS instruction supports nesting up to three levels.
			A2				A1				●PC+1→PCTEMP ●PCTEMP→(Stack) ●SP+1→SP ●A3→PC[11:8] in new Stack ●A2→PC[7:4] in new Stack ●A1→PC[3:0] in new Stack	
Jump Conditional	**JCN**	jcn 0x4 LABEL jcn 9 0x123 jcn TZ TARGET0 jcn TN TARGET1 jcn C0 TARGET2 jcn C1 TARGET3 jcn AZ TARGET4 jcn AN TARGET5 Reserved Words for Condition Bits TZ...CCCC=0001 TN...CCCC=1001 C0...CCCC=1010 C1...CCCC=0010 AZ...CCCC=0100 AN...CCCC=1100	0	0	0	1	C1	C2	C3	C4	●PC+1→PCTEMP ●PCTEMP→PC	This is a conditional branch instruction. Regardless of whether branching occurs, the PC is first incremented by 2 to point to the next instruction. Then, if the condition flags C1, C2, C3, and C4 specified in the instruction word are satisfied, the branch is taken. Otherwise, the instruction following JCN is executed. When branching, the A2 and A1 fields are transferred to PC[7:0]. Condition bit meanings (multiple conditions can be evaluated simultaneously): - C1 = 0: Do not invert the branch condition. - C1 = 1: Invert the branch condition. - C2 = 1: Branch if ACC equals 0. - C3 = 1: Branch if CY equals 1. - C4 = 1: Branch if TEST input signal equals 0. Branch evaluation logic: - JUMP = C1 XOR JUMP0 - JUMP0 = C2·(ACC==0) + C3·CY + C4·(TEST==0) Note: If PC[7:0] is 0xFE or 0xFF before executing the JCN instruction, incrementing PC by 2 causes a carry into PC[11:8], advancing the page number. However, during a branch, only PC[7:0] is updated. Therefore, if JCN is located at the end of a page (PC[7:0] == 0xFE or 0xFF), the branch target will reside within the next page.
			A2				A1				●PC+1→PCTEMP ●If (C1,C2,C3,C4) is true, PCTEMP[11:8]→PC[11:8] A2→PC[7:4] A1→PC[3:0] ●Else PCTEMP→PC	
Increment Index Register, Skip if Zero	**ISZ**	isz 9 LABEL isz 9 0x123	0	1	1	1	n				●PC+1→PCTEMP ●PCTEMP→PC	This is a conditional branch instruction. Regardless of whether branching occurs, the PC is first incremented by 2 to point to the next instruction. Then, index register Rn (designated by number n) is incremented, and if the result is non-zero, the branch is taken. If the result is zero, the instruction following ISZ is executed. When branching, the A2 and A1 fields are transferred to PC[7:0]. Note: If PC[7:0] is 0xFE or 0xFF before executing the ISZ instruction, incrementing PC by 2 causes a carry into PC[11:8], advancing the page number. However, during a branch, only PC[7:0] is updated. Therefore, if ISZ is located at the end of a page (PC[7:0] == 0xFE or 0xFF), the branch target will reside within the next page.
			A2				A1				●PC+1→PCTEMP ●Rn+1→Rn ●If (Rn != 0), PCTEMP[11:8]→PC[11:8] A2→PC[7:4] A1→PC[3:0] ●Else PCTEMP→PC	
Fetch Immediate from ROM	**FIM**	fim 1p 0xfe fim 1< 0xfe fim 2 0xfe These are identical.	0	0	1	0	m			0	●PC+1→PCTEMP ●PCTEMP→PC	The second word of the instruction is used as an immediate constant and stored into index register pair RmP.
			D2				D1				●PC+1→PCTEMP ●D2→RmP[7:4] ●D1→RmP[3:0] ●PCTEMP→PC	