

- Group -	- Encoding -	#S = immediate (I=1). S = register. #D = immediate (L=1). D = register.	* Z = (result == 0). ** If #S and cogex, PC += signed(S). If #S and hubex, PC += signed(S*4). If S, PC = register S.
Branch S - Call	EEEE 1011001 CZ1 DDDDDDDDD SSSSSSSSS	CALLD D, (#)S {WC/WZ/WCZ}	Call to S** by writing (C, Z, 10'b0, PC[19:0]) to D. C = S[31], Z = S[30].
Branch S - Call	EEEE 1011010 0L1 DDDDDDDDD SSSSSSSSS	CALLPA (#)D, (#)S	Call to S** by pushing (C, Z, 10'b0, PC[19:0]) onto stack, copy D to PA.
Branch S - Call	EEEE 1011010 1L1 DDDDDDDDD SSSSSSSSS	CALLPB (#)D, (#)S	Call to S** by pushing (C, Z, 10'b0, PC[19:0]) onto stack, copy D to PB.
Branch D - Call	EEEE 1011011 CZ0 DDDDDDDDD 000101101	CALLD D {WC/WZ/WCZ}	Call to D by pushing (C, Z, 10'b0, PC[19:0]) onto stack. C = D[31], Z = D[30], PC = D[19:0].
Branch D - Call	EEEE 1011011 CZ0 DDDDDDDDD 000101110	CALLA D {WC/WZ/WCZ}	Call to D by writing (C, Z, 10'b0, PC[19:0]) to hub long at PTRB+.
Branch D - Call	EEEE 1011011 CZ0 DDDDDDDDD 000101111	CALLB D {WC/WZ/WCZ}	Call to D by writing (C, Z, 10'b0, PC[19:0]) to hub long at PTRB+.
Branch A - Call	EEEE 1011011 RAA AAAAAAAAA AAAAAAAAA	CALL #A	Call to A by pushing (C, Z, 10'b0, PC[19:0]) onto stack. If R = 1, PC += A, else PC = A.
Branch A - Call	EEEE 1011011 RAA AAAAAAAAA AAAAAAAAA	CALLA #A	Call to A by writing (C, Z, 10'b0, PC[19:0]) to hub long at PTRB+.
Branch A - Call	EEEE 1011011 RAA AAAAAAAAA AAAAAAAAA	CALLB #A	Call to A by writing (C, Z, 10'b0, PC[19:0]) to hub long at PTRB+.
Branch A - Call	EEEE 11100W RAA AAAAAAAAA AAAAAAAAA	CALLD PA/PB/PTRA/PTRB, #A	Call to A by writing (C, Z, 10'b0, PC[19:0]) to PA/PB/PTRA/PTRB (per W). If R = 1, PC += A, else PC = A.
Instruction Prefix	0000 -----	RET <inst> <ops>	Execute <inst> always and return if no branch. If <inst> is not branching then return by popping stack[19:0] into PC.
Branch Return	EEEE 1101011 CZ1 000000000 000101101	RET {WC/WZ/WCZ}	Return by popping stack (K). C = K[31], Z = K[30], PC = K[19:0].
Branch Return	EEEE 1101011 CZ1 000000000 000101110	RETA {WC/WZ/WCZ}	Return by reading hub long (L) at --PTRB. C = L[31], Z = L[30], PC = L[19:0].
Branch Return	EEEE 1101011 CZ1 000000000 000101111	RETB {WC/WZ/WCZ}	Return by reading hub long (L) at --PTRB. C = L[31], Z = L[30], PC = L[19:0].
Branch S - Return	EEEE 1011001 110 111111111 111110001	RETI3	Return from INT3. (CALLD \$1FF, \$1F1 WC, WZ)
Branch S - Return	EEEE 1011001 110 111111111 111110011	RETI2	Return from INT2. (CALLD \$1FF, \$1F3 WC, WZ)
Branch S - Return	EEEE 1011001 110 111111111 111110101	RETI1	Return from INT1. (CALLD \$1FF, \$1F5 WC, WZ)
Branch S - Return	EEEE 1011001 110 111111111 111111111	RETI0	Return from INT0. (CALLD \$1FF, \$1FF WC, WZ)
Branch S - Resume	EEEE 1011001 110 111110000 111110001	RESI3	Resume from INT3. (CALLD \$1F0, \$1F1 WC, WZ)
Branch S - Resume	EEEE 1011001 110 111110010 111110011	RESI2	Resume from INT2. (CALLD \$1F2, \$1F3 WC, WZ)
Branch S - Resume	EEEE 1011001 110 111110100 111110101	RESI1	Resume from INT1. (CALLD \$1F4, \$1F5 WC, WZ)
Branch S - Resume	EEEE 1011001 110 111111110 111111111	RESI0	Resume from INT0. (CALLD \$1FE, \$1FF WC, WZ)
Branch D - Jump	EEEE 1101011 CZ0 DDDDDDDDD 000101100	JMP D {WC/WZ/WCZ}	Jump to D. C = D[31], Z = D[30], PC = D[19:0].
Branch D - Jump	EEEE 1101011 00L DDDDDDDDD 000110000	JMPREL (#)D	Jump ahead/back by D instructions. For cogex, PC += D[19:0]. For hubex, PC += D[17:0] << 2.
Branch A - Jump	EEEE 1101100 RAA AAAAAAAAA AAAAAAAAA	JMP #A	Jump to A. If R = 1, PC += A, else PC = A.
Branch S - Mod & Test	EEEE 1011011 00I DDDDDDDDD SSSSSSSSS	DJZ D, (#)S	Decrement D and jump to S** if result is zero.
Branch S - Mod & Test	EEEE 1011011 01I DDDDDDDDD SSSSSSSSS	DJNZ D, (#)S	Decrement D and jump to S** if result is not zero.
Branch S - Mod & Test	EEEE 1011011 10I DDDDDDDDD SSSSSSSSS	DJF D, (#)S	Decrement D and jump to S** if result is \$FFFF FFFF.
Branch S - Mod & Test	EEEE 1011011 11I DDDDDDDDD SSSSSSSSS	DJNF D, (#)S	Decrement D and jump to S** if result is not \$FFFF FFFF.
Branch S - Mod & Test	EEEE 1011100 00I DDDDDDDDD SSSSSSSSS	IJZ D, (#)S	Increment D and jump to S** if result is zero.
Branch S - Mod & Test	EEEE 1011100 01I DDDDDDDDD SSSSSSSSS	IJNZ D, (#)S	Increment D and jump to S** if result is not zero.
Branch S - Test	EEEE 1011100 10I DDDDDDDDD SSSSSSSSS	TJZ D, (#)S	Test D and jump to S** if D is zero.
Branch S - Test	EEEE 1011100 11I DDDDDDDDD SSSSSSSSS	TJNZ D, (#)S	Test D and jump to S** if D is not zero.
Branch S - Test	EEEE 1011101 00I DDDDDDDDD SSSSSSSSS	TJF D, (#)S	Test D and jump to S** if D is full (D = \$FFFF FFFF).
Branch S - Test	EEEE 1011101 01I DDDDDDDDD SSSSSSSSS	TJNF D, (#)S	Test D and jump to S** if D is not full (D != \$FFFF FFFF).
Branch S - Test	EEEE 1011101 10I DDDDDDDDD SSSSSSSSS	TJS D, (#)S	Test D and jump to S** if D is signed (D[31] = 1).
Branch S - Test	EEEE 1011101 11I DDDDDDDDD SSSSSSSSS	TJNS D, (#)S	Test D and jump to S** if D is not signed (D[31] = 0).
Branch S - Test	EEEE 1011110 00I DDDDDDDDD SSSSSSSSS	TJV D, (#)S	Test D and jump to S** if D overflowed (D[31] != C, C = 'correct sign' from last addition/subtraction).
Events - Branch	EEEE 1011110 01I 000000000 SSSSSSSSS	JINT (#)S	Jump to S** if INT event flag is set.
Events - Branch	EEEE 1011110 01I 000000001 SSSSSSSSS	JCT1 (#)S	Jump to S** if CT1 event flag is set.
Events - Branch	EEEE 1011110 01I 000000010 SSSSSSSSS	JCT2 (#)S	Jump to S** if CT2 event flag is set.
Events - Branch	EEEE 1011110 01I 000000011 SSSSSSSSS	JCT3 (#)S	Jump to S** if CT3 event flag is set.
Events - Branch	EEEE 1011110 01I 000000100 SSSSSSSSS	JSE1 (#)S	Jump to S** if SE1 event flag is set.
Events - Branch	EEEE 1011110 01I 000000101 SSSSSSSSS	JSE2 (#)S	Jump to S** if SE2 event flag is set.
Events - Branch	EEEE 1011110 01I 000000110 SSSSSSSSS	JSE3 (#)S	Jump to S** if SE3 event flag is set.
Events - Branch	EEEE 1011110 01I 000000111 SSSSSSSSS	JSE4 (#)S	Jump to S** if SE4 event flag is set.
Events - Branch	EEEE 1011110 01I 000001000 SSSSSSSSS	JPAT (#)S	Jump to S** if PAT event flag is set.
Events - Branch	EEEE 1011110 01I 000001001 SSSSSSSSS	JFBW (#)S	Jump to S** if FBW event flag is set.
Events - Branch	EEEE 1011110 01I 000001010 SSSSSSSSS	JXMT (#)S	Jump to S** if XMT event flag is set.
Events - Branch	EEEE 1011110 01I 000001011 SSSSSSSSS	JXFI (#)S	Jump to S** if XFI event flag is set.
Events - Branch	EEEE 1011110 01I 000001100 SSSSSSSSS	JXRO (#)S	Jump to S** if XRO event flag is set.
Events - Branch	EEEE 1011110 01I 000001101 SSSSSSSSS	JXRL (#)S	Jump to S** if XRL event flag is set.
Events - Branch	EEEE 1011110 01I 000001110 SSSSSSSSS	JATN (#)S	Jump to S** if ATN event flag is set.
Events - Branch	EEEE 1011110 01I 000001111 SSSSSSSSS	JQMT (#)S	Jump to S** if QMT event flag is set.
Events - Branch	EEEE 1011110 01I 000010000 SSSSSSSSS	JNINT (#)S	Jump to S** if INT event flag is clear.
Events - Branch	EEEE 1011110 01I 000010001 SSSSSSSSS	JNCT1 (#)S	Jump to S** if CT1 event flag is clear.
Events - Branch	EEEE 1011110 01I 000010010 SSSSSSSSS	JNCT2 (#)S	Jump to S** if CT2 event flag is clear.
Events - Branch	EEEE 1011110 01I 000010011 SSSSSSSSS	JNCT3 (#)S	Jump to S** if CT3 event flag is clear.
Events - Branch	EEEE 1011110 01I 000010100 SSSSSSSSS	JNSE1 (#)S	Jump to S** if SE1 event flag is clear.
Events - Branch	EEEE 1011110 01I 000010101 SSSSSSSSS	JNSE2 (#)S	Jump to S** if SE2 event flag is clear.
Events - Branch	EEEE 1011110 01I 000010110 SSSSSSSSS	JNSE3 (#)S	Jump to S** if SE3 event flag is clear.
Events - Branch	EEEE 1011110 01I 000010111 SSSSSSSSS	JNSE4 (#)S	Jump to S** if SE4 event flag is clear.
Events - Branch	EEEE 1011110 01I 000011000 SSSSSSSSS	JNPAT (#)S	Jump to S** if PAT event flag is clear.
Events - Branch	EEEE 1011110 01I 000011001 SSSSSSSSS	JNFBW (#)S	Jump to S** if FBW event flag is clear.
Events - Branch	EEEE 1011110 01I 000011010 SSSSSSSSS	JNXMT (#)S	Jump to S** if XMT event flag is clear.
Events - Branch	EEEE 1011110 01I 000011011 SSSSSSSSS	JNXFI (#)S	Jump to S** if XFI event flag is clear.
Events - Branch	EEEE 1011110 01I 000011100 SSSSSSSSS	JNXRO (#)S	Jump to S** if XRO event flag is clear.
Events - Branch	EEEE 1011110 01I 000011101 SSSSSSSSS	JNXRL (#)S	Jump to S** if XRL event flag is clear.
Events - Branch	EEEE 1011110 01I 000011110 SSSSSSSSS	JNATN (#)S	Jump to S** if ATN event flag is clear.
Events - Branch	EEEE 1011110 01I 000011111 SSSSSSSSS	JNQMT (#)S	Jump to S** if QMT event flag is clear.
Branch D - Skip	EEEE 1101011 00L DDDDDDDDD 000110001	SKIP (#)D	Skip instructions per D. Subsequent instructions 0..31 get cancelled for each '1' bit in D[0]..D[31].
Branch D - Jump+Skip	EEEE 1101011 00L DDDDDDDDD 000110010	SKIPF (#)D	Skip cog/LUT instructions fast per D. Like SKIP, but instead of cancelling instructions, the PC leaps over them.
Branch D - Call+Skip	EEEE 1101011 00L DDDDDDDDD 000110011	EXECF (#)D	Jump to D[9:0] in cog/LUT and set SKIPF pattern to D[31:10]. PC = {10'b0, D[9:0]}.