

0279	32	BZ		;Branch to exit - no conversion needed
7A	8F			
7B	F0	LDX		;Get the same instruction from stack
7C	FB	XRI		
7D	70			;If = 70 (7N type)
7E	32	BZ		;Branch to exit
7F	8F			
0280	F0	LDX		
81	FB	XRI		
82	C0			;If = C0 (CN type)
83	32	BZ		;Branch to exit
84	8F			
85	F0	LDX		
86	FB	XRI		
87	F0			;If = F0 (FN type)
88	32	BZ		;Branch to exit
89	8F			
8A	0A	LDN	RA	;Else get the instruction
8B	F9	ORI		
8C	0F			; "OR" to convert to XF for look up table
8D	B7	PHI	R7	;Store in R7.1
8E	D5	SEP	R5	;Return
8F	0A	LDN	RA	;Some tests branch to here, loading R7.1
0290	B7	STR	R7	;With the unconverted instruction
91	D5	SEP	R5	;And returning

#### TEST FOR ARGUMENTS

0292	F8	LDI		
93	01			
94	AB	PLO	RB	;Initialize RB.0 = 01 -Argument count
95	F8	LDI		
96	02			
97	AF	PLO	RF	;RF.0 = Loop count - 2 part table
98	0A	LDN	RA	;Get instruction for disassembly
99	E2	SEX	2	;X = 2
9A	52	STR	R2	;Push value for comparison test
9B	F8	LDI		
9C	09			
9D	BE	PHI	RE	;Set RE = beginning
9E	F8	LDI		
9F	C0			;Of argument table @ 09C0
02A0	AE	PLO	RE	
A1	4E	LDA	RE	;Get entry from table - advance pointer
A2	32	BZ		;If 00 (null) found (end of part "N" table)
A3	A8			;Branch to done test
A4	F3	XOR		;Else compare with stacked instruction
A5	3A	BNZ		;If not equal, loop till equality or
A6	A1			;End of table part "N" found