APPENDIX D. THE INFERENCING TOKEN LANGUAGE

Network Inferencing Functions

Hex	Token name	Function
10	<Exit>	Terminate token processing (end of ITL statement)
11	<Text>	Put following text in the Terminal Input Buffer (TIB). Followed immediately by a counted string: a byte count n, then n characters.
12	<DoText>	Interpret the text string in TIB as a Forth command, then clear the TIB.
13	<Type>	Type the following text to the console. Followed immediately by a counted string: a byte count n, then n characters.
14	<Sequence>	Terminate token processing (exit the token thread) if this sequence number matches the last received from this sender. Followed immediately by a one-byte sequence number.
15	<Noop>	Do nothing.
16	<ReSequence>	Reset all sequence numbers. Normally sent as a broadcast, in an unsequenced message, to reset all the CPUs in the network.
17	<Setclock>	Set the real-time clock to the given 32-bit value. Followed immediately by a four-byte value in "big-endian" order: MSB, 2ndSB, 3rdSB, LSB.
18	<Tell>	Assert the given value and expiration time for a rule. Followed immediately by rule#, time, and value, all 16-bit values in "big-endian" order.
19	<Ask>	Request the current value and expiration time of a rule. Followed immediately by rule#, a 16-bit value in "big-endian"order. The reply is a <Tell> message.
1A	<Invoke>	Invoke the given rule. Followed immedately by a 16-bit rule number.
1B	<Scrub>	Reset all rules to "unknown."

Rule-Writing Language

Hex	Token name	Function
20	<#literal>	Put a literal value on the stack. Followed immediately by a two-byte value in "big-endian" order.

Hex	Token name	Function
21	<'literal>	Execute the Forth word at the given address. Followed by a two-byte address in "big-endian" order.
22	<UnnestF>	Terminate token processing (exit the token thread) IF the fact on the top of stack is unknown or false. Regardless of outcome, the fact is left undisturbed on the stack.
23	<Now>	Put the current 16-bit time on the stack.
24	<Forever>	Put the maximum future time on the stack.
25	<Hence>	Put the current 16-bit time plus a literal value on the stack. Followed immediately by a two-byte in "big-endian" order.
26	<Fetch>	Fetch the Forth variable specified by the following address, and stack it with an expiration date of "forever." Followed by two-byte "big-endian" address.
30	AND	Logicaly AND two temporal quantities on the stack. (A temporal quantity has a value and an expiration time.)
31	OR	Logically OR two temporal quantities on the stack.
32	NOT	Logically invert a temporal quantity on the stack.
33	NULL	Put the temporal quantity "false, unknown" on the stack. (A value of zero, and an expired time.)
40	+	Add two temporal integers on the stack.
41	-	Subtract two temporal integers on the stack.
42	MIN	Return the minimum of two temporal integers.
43	MAX	Return the maximum of two temporal integers.
44	>	Return a temporal true, if temporal integer 1 > temporal integer 2.
45	<	Return a temporal true, if temporal integer 1 < temporal integer 2.
46	=	Return a temporal true, if temporal integer 1 = temporal integer 2.
47	U<	Like < but uses unsigned integer comparison rather than signed.
48	U>	Like > but uses unsigned integer comparison rather than signed.
49	0=	Return a temporal true, if the value of the temporal integer is zero.
4A	0<	Return a temporal true, if the value of the temporal integer is negative.