

APPENDIX A ASSIGNMENT STATEMENT REFERENCE

A.1 ASSIGNMENT STATEMENTS

The assignment statement assigns a value to a given target. The target of an assignment can be an input name, a local attribute name, or one of several predefined names listed below. All assignment statements have an expression as their right hand side. The target will be assigned the value of that expression. Results of assignment statements depend upon the data type as listed below for the different assignment types. See **Display Block** for usage rules and an explanation of resources.

- Table A-1. Assignment Statement Results of Type 'srcname1 = srcname2'
For example, I1 = I2 or I1 = 4576, or RESULT = INPUT * RATIO + BIAS (where input, ratio and bias are unique names for display block inputs or local attribute names).
- Table A-2. Assignment Statements of Type 'display_resource = source_name'
For example, #RBAR = OUTPUT (where #RBAR is a display resource and output is the unique name of a display block input or local attribute name).
- Table A-3. Assignment Statements of Type 'display_resource = constant'
For example, #LINE1 = 37.94 (where #LINE1 is a display resource).
- Table A-4. Assignment Statements of Type 'display_resource.SRC = srcname'
For example, #RBAR.SRC = OUTPUT; (where #RBAR.SRC is a display resource having the suffix .SRC and output is the unique name of a display block input or local attribute name).

Table A-1. Assignment Statement Results of Type 'srcname1 = srcname2'

Destination Type	Source Type	Result
Discrete	Discrete:	= input
	Short State:	True if input non-zero
	Long State:	True if input non-zero
	Count:	True if input non-zero
	Msec Time:	True if input non-zero
	Floating Point	True if input non-zero
	Date: Hex:	True if input non-zero
Short State	Discrete:	= input
	Short State:	= input
	Long State:	limits at 15
	Count:	limits at 15
	Msec Time:	treats input as 4 byte integer, limits at 15
	Floating Point	drops fractional part, limits at 15
	Date: Hex:	treats input as 3 byte integer, limits at 15 treats input as n byte integer, limits at 15

APPENDIX A - ASSIGNMENT STATEMENT REFERENCE

Table A-1. Assignment Statement Results of Type 'srcname1 = srcname2' (Cont'd)

Destination Type	Source Type	Result
Long State	Discrete: Short State: Long State: Count: Msec Time: Floating Point Date: Hex:	= input = input = input limits at 255 treats input as 4 byte integer, limits at 255 drops fractional part, limits at 255 treats input as 3 byte integer, limits at 255 treats input as n byte integer, limits at 255
Count	Discrete: Short State: Long State: Count: Msec Time: Floating Point Date: Hex:	= input = input = input = input treats input as 4 byte integer, limits at 65535 drops fractional part, limits at 65535 treats input as 3 byte integer, limits at 65535 treats input as n byte integer, limits at 65535
Msec Time Treats output as 4 byte integer	Discrete: Short State: Long State: Count: Msec Time: Floating Point Date: Hex:	= input = input = input = input drops fractional part, limits at \$FFFFFFFF = input, treats input as 3 byte integer treats input as n byte integer, limits at \$FFFFFFFF
Floating Point	Discrete: Short State: Long State: Count: Msec Time: Floating Point Date: Hex:	= input = input = input = input treats input as 4 byte integer (lost resolution in large numbers) = input = input, treats input as 3 byte integer treats input as n byte integer (lost resolution in large numbers)
Date Treats output as 3 byte integer Accepts any input if output is local, but only valid dates if output is remote.	Discrete: Short State: Long State: Count: Msec Time: Floating Point Date: Hex:	= input = input = input = input treats input as 4 byte integer, limits at \$FFFFFF drops fractional part, limits at \$FFFFFF = input treats input as n byte integer, limits at \$FFFFFF
Ascii Output truncated to fit available field size	Discrete: Short State: Long State: Count: Msec Time: Floating Point Date: Ascii: Hex:	0 or 1 as ascii 0 - 15 as ascii 0 - 255 as ascii 0 - 65535 as ascii 0 - 4294967295 as ascii value to as many decimal places fit in output field ascii date format.Accepts any input, so can look strange. = input hex value as ascii

APPENDIX A - ASSIGNMENT STATEMENT REFERENCE

Table A-1. Assignment Statement Results of Type 'srcname1 = srcname2' (Cont'd)

Hex Output limits at available field size (all FF's)	Discrete: Short State: Long State: Count: Msec Time: Floating Point Date: Hex:	= input = input = input = input = input, treats input as 4 byte integer drops fractional part, input limits at \$FFFFFFFF = input, treats input as 3 byte integer = input
--	---	--

APPENDIX A - ASSIGNMENT STATEMENT REFERENCE

Table A-2. Assignment statements of type 'display_resource = source_name' (ex. #LINE1 = I1)

Resource	Format	Displayed Result
#LINE1	Discrete:	Integer1 0, 1
#LINE2		State Note 1
#LINE3	Short_State:	Integer1 0 - 15
		State Note 1
	Long_State:	Integer1 0 - 255
		State Note 1
	Count:	Integer2 0 - 65535
		State Note 1
	Msec_time:	Integer4 0 - 99999999 (0 - 99999999), first 8 digits (10000000 and up)
		Time1 hh:mm:ss (00:00:00.000 to 99:59:59.999), hhhh:mm (100:00:00.000 and up)
		Time2 m:ss.sss (0:00.000 to 9:59.999), mm:ss.ss (10:00.000 to 99:59.999), hh:mm:ss (00:00:00.000 to 99:59:59.999), hhhh:mm (100:00:00.000 and up)
		Time3 hhhh:mm
		Time4 mm:ss.ss (10:00.000 to 99:59.999), hh:mm:ss (00:00:00.000 to 99:59:59.999), hhhh:mm (100:00:00.000 and up)
	Floating_point	Floatn 0.0 - 99999999 (0.0 - 99999999.0), first 8 digits (10000000.0 and up)
	Date:	Date mm/dd/yy or dd/mm/yy
	Asc:	Ascii truncated to 8 characters.
	Hex:	Hex leftmost characters discarded if more than 8.
#LINE3	Discrete:	Integer1 0, 1
#LINE4		State Note 1
#LINE5	Short_state:	Integer1 0 - 15
		State Note 1
	Long_state:	Integer1 0 - 255
		State Note 1
	Count:	Integer2 0 - 999 (0 - 999), First 3 digits (1000 - 65535)
		State Note 1
	Msec_time:	Integer4 0 - 999 (0 - 999), first 3 digits alternate (1000 and up)
		Time1 :ss (even if > 0:00:99.999)
		Time2 sss (even if > 0:00:00.999)
		Time3 :mm (even if > 0:99:59.999)
		Time4 .ss (even if > 0:00:00.999)
	Floating_point	0.0 - 999 (0.0 - 999.0), first 3 digits (1000.0 and up)
	Date:	/yy (right justified)
	Asc:	truncated to 3 characters
	Hex:	leftmost characters discarded if more than 3.

Note 1 If the source value is found in the state table, the associated mnemonic is displayed. Otherwise the default mnemonic is displayed. All mnemonics in a state table have the same configured width. If not entered full width, they are padded with trailing spaces during compile. If the mnemonic field is wider than the display it is truncated to fit. Otherwise it (the entire field) is right justified.

APPENDIX A - ASSIGNMENT STATEMENT REFERENCE

Table A-2. Assignment statements of type 'display_resource = source_name' (ex. #LINE1 = 11)

Resource	Format	Displayed Result	
#LBAR #MBAR #RBAR	Discrete:	None	off, segment 1 on
		Bar	0 - 50 segments on (lo_limit - hi_limit), 50 segments on (> hi_limit)
		Dev	
	Short_state:	None	0 - 15 segments on
		Bar	0 - 50 segments on (lo_limit - hi_limit), 50 segments on (> hi_limit)
		Dev	
	Long_state:	None	0 - 50 segments on (0 - 50), 50 segments on (51 - 255)
		Bar	0 - 50 segments on (lo_limit - hi_limit), 50 segments on (> hi_limit)
		Dev	
	Count:	None	0 - 50 segments on (0 - 50), 50 segments on (51 - 65535)
		Bar	0 - 50 segments on (lo_limit - hi_limit), 50 segments on (> hi_limit)
		Dev	
	Msec_time:	None	0 - 50 segments on (00:00:00.000 - 00:00:00.050), 50 segments on (00:00:00.051 and up)
		Bar	0 - 50 segments on (lo_limit - hi_limit), 50 segments on (> hi_limit)
		Dev	
	Floating_point	None	0 - 50 segments on (0.0 - 50.0), 50 segments on (51.0 and up)
		Bar	0 - 50 segments on (lo_limit - hi_limit), 50 segments on (> hi_limit)
		Dev	

DEV format is not valid for intensified segments and causes a 'INVALID DISP OP' diagnostic

#LISEGn #MISEGn #RISEGn	Discrete:	None	off, segment 1 on
		Bar	off (< lo_limit), one of segments 1 - 50 on (lo_limit - hi_limit), segment 50 on (> hi_limit)
	Short_state:	None	off (0), one of segments 1 - 15 on (1 - 15)
		Bar	off (< lo_limit), one of segments 1 - 50 on (lo_limit - hi_limit), segment 50 on (> hi_limit)
	Long_state:	None	off (0), one of segments 1 - 50 on (1 - 50), segment 50 on (51 - 255)
		Bar	off (< lo_limit), one of segments 1 - 50 on (lo_limit - hi_limit), segment 50 on (> hi_limit)
	Count:	None	off (0), one of segments 1 - 50 on (1 - 50), segment 50 on (51 - 65535)
		Bar	off (< lo_limit), one of segments 1 - 50 on (lo_limit - hi_limit), segment 50 on (> hi_limit)
	Msec_time:	None	off (0), one of segments 1 - 50 on (00:00:00.001 - 00:00:00.050), segment 50 on (00:00:00.051 and up)
		Bar	off (< lo_limit), one of segments 1 - 50 on (lo_limit - hi_limit), segment 50 on (> hi_limit)
	Floating_point	None	off (0.x), one of segments 1 - 50 on (1.0 - 50.0), segment 50 on (51.0 and up)
		Bar	off (< lo_limit), one of segments 1 - 50 on (lo_limit - hi_limit), segment 50 on (> hi_limit)
#LED #BEEPER	Discrete:	na	off, on
	Short_state:	na	off (0), on (1), system use (2 - 15)
	Long_state:	na	off (0), on (1), system use (2 - 255)
	Count:	na	off (0), on (1), system use (2 - 65535)
	Msec_time:	na	off (00:00:00.000), on (00:00:00.001), system use (00:00:00.002 and up)
	Floating_point	na	off (0.x), on (1.x), system use (2.0 and up)
#UPDN	Discrete:	na	off, up arrow on
	Short_state:	na	dn arrow, slash and up arrow show binary value (0 - 7), system use (8 - 15)
	Long_state:	na	dn arrow, slash and up arrow show binary value (0 - 7), system use (8 - 255)
	Count:	na	dn arrow, slash and up arrow show binary value (0 - 7), system use (8 - 65535)
	Msec_time:	na	dn arrow, slash and up arrow show binary value (0 - 00:00:00.007), system use (00:00:00.008 and up)
	Floating_point	na	dn arrow, slash and up arrow show binary value of integer part (0.x - 7.x), system use (8.0 and up)

APPENDIX A - ASSIGNMENT STATEMENT REFERENCE

Table A-3. Assignment statements of type 'display_resource = constant' - ex. #LINE1 = 37.94

Resource	Constant Type	Displayed Result
#LINE1 #LINE2 #LINE6	Discrete: Short_state: Long_state: Count: Msec_time: Floating_point Date: Asc: Hex:	0, 1 0 - 15 0 - 255 0 - 65535 hh:mm:ss rounded to nearest storeable value (23 bit precision) and truncated to fit 8 characters. mm/dd/yy or dd/mm/yy truncated to 8 characters. leftmost characters discarded if more than 8.
#LINE3 #LINE4 #LINE5	Discrete: Short_state: Long_state: Count: Msec_time: Floating_point Date: Asc: Hex:	0, 1 0 - 15 0 - 255 0 - 999 (0 - 999), First 3 digits (1000 - 65535) :ss rounded to nearest storeable value (23 bit precision) and truncated to fit 3 characters. /yy (pretty useless) truncated to 3 characters leftmost characters discarded if more than 3.
#LBAR #MBAR #RBAR	Discrete: Short_state: Long_state: Count: Msec_time: Floating_point	off, segment 1 on 0 - 15 segments on 0 - 50 segments on (0 - 50), 50 segments on (51 - 255) 0 - 50 segments on (0 - 50), 50 segments on (51 - 65535) 0 - 50 segments on (00:00:00.000 - 00:00:00.050), 50 segments on (00:00:00.051 and up) 0 - 50 segments on (0.0 - 50.x), 50 segments on (51.0 and up)
#LISEGn #MISEGn #RISEGn	Discrete: Short_state: Long_state: Count: Msec_time: Floating_point	off, segment 1 on off (0), one of segments 1 - 15 on (1 - 15) off (0), one of segments 1 - 50 on (1 - 50), segment 50 on (51 - 255) off (0), one of segments 1 - 50 on (1 - 50), segment 50 on (51 - 65535) off (0), one of segments 1 - 50 on (00:00:00.001 - 00:00:00.050), segment 50 on (00:00:00.051 and up) off (0.x), one of segments 1 - 50 on (1.0 - 50.x), segment 50 on (51.0 and up)
#LED #BEEPER	Discrete: Short_state: Long_state: Count: Msec_time: Floating_point	off, on off (0), on (1), system use (2 - 15) off (0), on (1), system use (2 - 255) off (0), on (1), system use (2 - 65535) off (00:00:00.000), on (00:00:00.001), system use (00:00:00.002 and up) off (0.x), on (1.x), system use (2.0 and up)
#UPDN	Discrete: Short_state: Long_state: Count: Msec_time: Floating_point	off, up arrow on dn arrow, slash and up arrow show binary value (0 - 7), system use (8 - 15) dn arrow, slash and up arrow show binary value (0 - 7), system use (8 - 255) dn arrow, slash and up arrow show binary value (0 - 7), system use (8 - 65535) dn arrow, slash and up arrow show binary value (0 - 00:00:00.007), system use (00:00:00.008 and up) dn arrow, slash and up arrow show binary value of integer part (0.x - 7.x), system use (8.0 and up)

APPENDIX A - ASSIGNMENT STATEMENT REFERENCE

Table A-4. Assignment statements of type 'display_resource.SRC = source_name' (ex. #LINE1.SRC = I1)

Resource	Format	Displayed Result
#LINE1.SRC	Discrete:	Integer1 0, 1
#LINE2.SRC		State NOTE 1
#LINE3.SRC	Short_state:	Integer1 0 - 15
		State NOTE 1
	Long_state:	Integer1 0 - 255
		State NOTE 1
	Count:	Integer2 0 - 65535
		State NOTE 1
	Msec_time:	Integer4 0 - 99999999 (0 - 99999999), first 8 digits alternate with 'OVERFLOW' (100000000 and up)
		Time1 hh:mm:ss (00:00:00.000 to 99:59:59.999), hhhh:mm alternates with 'OVERFLOW' (100:00:00.000 and up)
		Time2 m:ss.sss (0:00.000 to 9:59.999), 'OVERFLOW' alternates with mm:ss.ss (10:00.000 to 99:59.999), hh:mm:ss (00:00:00.000 to 99:59:59.999), hhhh:mm (100:00:00.000 and up)
		Time3 hhh:mm
		Time4 mm:ss.ss (10:00.000 to 99:59.999), 'OVF' alternates with hh:mm:ss (00:00:00.000 to 99:59:59.999), hhhh:mm (100:00:00.000 and up)
	Floating_point	Floatn 0.0 - 99999999 (0.0 - 99999999.0), first 8 digits alternate with 'OVERFLOW' (100000000.0 and up)
	Date:	Date mm/dd/yy or dd/mm/yy
	Asc:	Ascii = input, marquee'd if necessary
	Hex:	Hex = input, marquee'd if necessary
#LINE3.SRC	Discrete:	Integer1 0, 1
#LINE4.SRC		State NOTE 1
#LINE5.SRC	Short_state:	Integer1 0 - 15
		State NOTE 1
	Long_state:	Integer1 0 - 255
		State NOTE 1
	Count:	Integer2 0 - 999 (0 - 999), first 3 digits alternate with 'OVF' (1000 - 65535)
		State NOTE 1
	Msec_time:	Integer4 0 - 999 (0 - 999), first 3 digits alternate with 'OVF' (1000 and up)
		Time1 :ss, alternates with 'OVF' if > 0:00:99.999
		Time2 sss, alternates with 'OVF' if > 0:00:00.999
		Time3 :mm, alternates with 'OVF' if > 0:99:59.999
		Time4 .ss, alternates with 'OVF' if > 0:00:00.999
	Floating_point	0.0-999 (0.0-999.0), first 3 digits alternate with 'OVERFLOW' (1000.0 & up)
	Date:	/yy (right justified)
	Asc:	= input, marquee'd if necessary
	Hex:	= input, marquee'd if necessary

Note 1: If the source value is found in the state table, the associated mnemonic is displayed. Otherwise the default mnemonic is displayed. All mnemonics in a state table have the same configured width. If not entered full width, they are padded with trailing spaces during compile. If the mnemonic field is wider than the display it is truncated to fit. Otherwise it (the entire field) is right justified.

APPENDIX A - ASSIGNMENT STATEMENT REFERENCE

Table A-4. Assignment statements of type 'display_resource.SRC = source_name' (ex. #LINE1.SRC = I1)

Resource	Format	Displayed Result	
segment 1 remains on for source <= 0 to indicate bar is in use.			
#LBAR.SRC #MBAR.SRC #RBAR.SRC	Discrete:	None	segment 1 on
		Bar	1 - 50 segments on (lo_limit - hi_limit), 50 segments on (> hi_limit)
		Dev	
	Short_state:	None	1 - 15 segments on
		Bar	1 - 50 segments on (lo_limit - hi_limit), 50 segments on (> hi_limit)
		Dev	
	Long_state:	None	1 - 50 segments on (0 - 50), 50 segments on (51 - 255)
		Bar	1 - 50 segments on (lo_limit - hi_limit), 50 segments on (> hi_limit)
		Dev	
	Count:	None	1 - 50 segments on (0 - 50), 50 segments on (51 - 65535)
		Bar	1 - 50 segments on (lo_limit - hi_limit), 50 segments on (> hi_limit)
		Dev	
	Msec_time:	None	1 - 50 segments on (00:00:00.000 - 00:00:00.050), 50 segments on (00:00:00.051 and up)
		Bar	1 - 50 segments on (lo_limit - hi_limit), 50 segments on (> hi_limit)
		Dev	
	Floating_point	None	1 - 50 segments on (0.0 - 50.0), 50 segments on (51.0 and up)
		Bar	1 - 50 segments on (lo_limit - hi_limit), 50 segments on (> hi_limit)
		Dev	

segment 1 remains on for source <= 0 to indicate bar is in use. DEV format is not valid for iseg's and causes a 'INVALID DISP OP' diagnostic.

#LISEGn.SRC #MISEGn.SRC #RISEGn.SRC	Discrete:	None	segment 1 on
		Bar	one of segments 1 - 50 on (lo_limit - hi_limit), segment 50 on (> hi_limit)
	Short_state:	None	one of segments 1 - 15 on
		Bar	one of segments 1 - 50 on (lo_limit - hi_limit), segment 50 on (> hi_limit)
	Long_state:	None	one of segments 1 - 50 on (0 - 50), segment 50 on (51 - 255)
		Bar	one of segments 1 - 50 on (lo_limit - hi_limit), segment 50 on (> hi_limit)
	Count:	None	one of segments 1 - 50 on (0 - 50), segment 50 on (51 - 65535)
		Bar	one of segments 1 - 50 on (lo_limit - hi_limit), segment 50 on (> hi_limit)
	Msec_time:	None	one of segments 1 - 50 on (00:00:00.000 - 00:00:00.050), segment 50 on (00:00:00.051 and up)
		Bar	one of segments 1 - 50 on (lo_limit - hi_limit), segment 50 on (> hi_limit)
	Floating_point	None	one of segments 1 - 50 on (0.0 - 50.0), segment 50 on (51.0 and up)
		Bar	one of segments 1 - 50 on (lo_limit - hi_limit), segment 50 on (> hi_limit)
#LED.SRC #BEEPER.SRC	Discrete:	NA	off, on
	Short_state:	NA	off (0), on (1), system use (2 - 15)
	Long_state:	NA	off (0), on (1), system use (2 - 255)
	Count:	NA	off (0), on (1), system use (2 - 65535)
	Msec_time:	NA	off (00:00:00.000), on (00:00:00.001), system use (00:00:00.002 and up)
	Floating_point	NA	off (0.x), on (1.x), system use (2.0 and up)

APPENDIX A - ASSIGNMENT STATEMENT REFERENCE

Table A-4. Assignment statements of type 'display_resource.SRC = source_name' (ex. #LINE1.SRC = I1)

Resource	Format	Displayed Result	
#UPDN.SRC	Discrete:	NA	off, up arrow on
	Short_state:	NA	dn arrow, slash and up arrow show binary value (0 - 7), system use (8 - 15)
	Long_state:	NA	dn arrow, slash and up arrow show binary value (0 - 7), system use (8 - 255)
	Count:	NA	dn arrow, slash and up arrow show binary value (0 - 7), system use (8 - 65535)
	Msec_time:	NA	dn arrow, slash and up arrow show binary value (0 - 00:00:00.007), system use (00:00:00.008 and up)
	Floating_point	NA	dn arrow, slash and up arrow show binary value of integer part (0.x - 7.x), system use (8.0 and up)

APPENDIX A - ASSIGNMENT STATEMENT REFERENCE
