RTCA Special Committee 186, Working Group 3

ADS-B 1090 MHz MOPS

Meeting #9

Fort Lauderdale, Florida
9 January 2002

Transition Table for NL(lat) Function

Presented by James Maynard

SUMMARY
This paper is presented in (belated) support of Action Item #2-17, “Review the NL equation at A.7.2d and possibly reword for latitudes at 87.” With the assistance of Ed Bayliss and Randy Jacobson, I have re-examined the last row of Table 2-90, the table for testing the transition latitudes, and have prepared a table of transition latitudes in hexadecimal angular weighted binary notation.

1. Background. At meeting #2, WG-3 reviewed Working Paper 2-05 from Randy Jacobson. That paper recommended a change to the values in the last row of Table 2-90, which is used to describe a test for the transition latitude at which the NL(lat) function changes from a value of 2 to a value of 1. This transition latitude is <u>exactly</u> ±87 degrees, which, when expressed as an angular weighted binary numeral, is the repeating hexadecimal fraction,

$$0\mathbf{.}3DDDDDDDDDD\mathbf{\ldots} = (3DDDDDDD\mathbf{.}DDDD\mathbf{\ldots}) \times (2^{-32} \text{ circles}).$$

When rounded to the nearest multiple of 2^{-32} circles, this is

$$(\text{hexadecimal } 3DDDDDDE) \times (2^{-32} \text{ circles}).$$

In the minutes of meeting #2, it is recorded that WG-3 decided that the last row of Table 2-90 WG-3 should read as follows:

Latitude (rounded to nearest multiple of 2^{-32} circle)		Even Surface Position Encodings			
$\mathbf{Lat_{lower}}$	$\mathbf{Lat_{upper}}$	Lower		Upper	
		Latitude	**Longitude**	**Latitude**	**Longitude**
86.999999° = 3DDDDDDDD	87.00011° = 3DDDDE66	`00000`	`00000`	`00001`	`10000`

After studying these minutes, **it is my view that the table values recorded in minute #5 of WG-3 meeting #2 should be changed**. That minute shows a latitude <u>below</u> the transition latitude of exactly 87° being encoded as hexadecimal 00000, as if that latitude were at or slightly <u>above</u> the latitude zone boundary at 87°. The encoded latitude for 86.999999° should, I think, be hexadecimal 1FFFF, not hexadecimal 0000.

Also, I think it would be better for the values of lat_{lower} and lat_{upper} each to differ from 87° by slightly less than one-half the weight of the LSB in the surface position latitude encoding, that is, by a little less than $6° \times 2^{-20} = 0.000005722°$.

2. Teleconference with Ed Bayliss. Ed Bayliss (the original author of CPR), Randy Jacobson (the author of WP 2-05) and Jim Maynard conferred by telephone on 18 December 2001. The three of us reviewed both Table 2-90 and the NL equation at A.7.2.d, seeking ways to reduce confusion about the transition in the NL(lat) function that occurs at latitudes of ±87º. The three of us agreed:

(a) that the last row of Table 2-90 really ought to show values both below and above the transition latitude of 87° (lat_{lower} and lat_{upper}, respectively);

(b) that it would be helpful for Appendix A to give an explicit table of transition latitudes in 32-bit angular weighted binary notation; and

(c) that Table 2-90 should show the lat_{lower} and lat_{upper} values in hexadecimal AWB (angular weighted binary) notation to a resolution of 2^{-32} circles, and also in decimal degrees.

We agreed that more than just 6 places of decimals would be needed to represent the lat_{lower} and lat_{upper} values to the same resolution as in the AWB representation. (This is because 2^{-32} circles = 8.38×10^{-8} degrees.)

In that teleconference, I agreed to draft revised text for Table 2-90, to show the lat_{lower} and lat_{upper} test latitudes both in 32-bit angular weighted binary (8 hex digits) and in decimal degrees. I also agreed to draft proposed text for Appendix A to show an explicit table of transition latitudes.

3. Detailed computation for test latitudes in last row of Table 2-90. The lower and upper test latitude values, lat_{lower} and lat_{upper} in the last row of table 2-90 should differ from exactly 87° by slightly less than one-half of the weight of the LSB in the surface CPR encoding. The latter value, "half the weight of the LSB in the even surface CPR encoding," is

$$0.5 \times 2^{-19} \times 6° = 5.722045898 \times 10^{-6} \text{ degrees} = 68.26666\ldots \times 2^{-32} \text{ circles.}$$

So, let us choose test latitudes, lat_{lower} and lat_{upper}, that differ from exactly 87° by 67×2^{-32} circles = (hexadecimal 43) $\times 2^{-32}$ circles.

$$87° = (\text{hexadecimal 3DDDDDDD}\mathbf{.}\text{DDDD}\ldots) \times (2^{-32} \text{ circles})$$
$$= (\text{hexadecimal 3DDDDDDE}) \times (2^{-32} \text{ circles}), \text{ when rounded to 32 bits.}$$

Thus suitable candidates for the lower and upper test latitudes for the last row of the Table might be

$$lat_{lower} = (\text{hex 3DDDDDDD} - \text{hex 44}) \times (2^{-32} \text{ circles})$$
$$= (\text{hex 3DDDDD99}) \times (2^{-32} \text{ circles})$$

$$lat_{lupper} = (\text{hex 3DDDDDDD} + \text{hex 43}) \times (2^{-32} \text{ circles})$$
$$= (\text{hex 3DDDDE20}) \times (2^{-32} \text{ circles})$$

In decimal degrees, lat_{lower} = 86.999994228°, and lat_{upper} = 87.000005543°.

4. Detailed computation for the encoded test latitudes in the last row of Table 2-90. Since lat_{lower} is just below the transition latitude of exactly 87° by less than the weight of the LSB in the encoded surface position latitude, the encoded latitude for lat_{lower} should consist of 17 binary "1" digits: hexadecimal 1FFFF. Likewise, since lat_{upper} is just above the transition latitude of 87°, by less than the weight of the LSB in the encoded latitude, the encoded value of the upper latitude should consist of 17 binary "0" digits: hexadecimal 00000.

This may be verified as follows.

For the lower latitude, lat_{lower}, encoded latitude (YZ_0, the encoded Y-coordinate within the Zone for even encoding) is computed as follows:

$$
\begin{aligned}
YZ_0 &= floor\left(2^{19} \cdot \frac{MOD\left(lat_{lower}, Dlat_0\right)}{Dlat_0} + \frac{1}{2}\right) \\
&= floor\left(2^{19} \cdot \frac{MOD\left(86.999994228°, 6°\right)}{6°} + \frac{1}{2}\right) \\
&= floor\left(2^{19} \cdot \frac{2.999994228°}{6°} + 0.5\right) \\
&= floor\left(2^{19} \cdot 0.499999038 + 0.5\right) \\
&= floor(262143.95609) \\
&= 262143
\end{aligned}
$$

Thus, the 19-bit value of YZ_0 is decimal 262143 = hexadecimal 3FFFF. Since only the lower 17 bits are used, the encoded value of lat_{lower} is hexadecimal 1FFFF.

Likewise, for the upper latitude, lat_{upper} = 87.000005543°,

$$
\begin{aligned}
YZ_0 &= floor\left(2^{19} \cdot \frac{MOD\left(lat_{up}, Dlat_0\right)}{Dlat_0} + \frac{1}{2}\right) \\
&= floor\left(2^{19} \cdot \frac{MOD\left(87.000005543°, 6°\right)}{6°} + \frac{1}{2}\right) \\
&- floor\left(2^{19} \cdot 0.500000924 + 0.5\right) \\
&= floor(262144.484390 + 0.5) \\
&= 262144
\end{aligned}
$$

Thus, the 19-bit value of YZ_0 for lat_{upper} = decimal 262144 = hexadecimal 40000. Since only the lower 17 bits are used, the encoded value of lat_{upper} is hexadecimal 00000.

Table 2-90would then look like the following:

Table 2-90: Verification of Transition Table.

Latitude (rounded to nearest multiple of 2^{-32} circle)				Even Surface Position Encodings			
Lat_{lower}		Lat_{upper}		Lower		Upper	
AWB (hex)	Dec. Degrees	AWB (hex)	Dec. Degrees	Latitude	Longitude	Latitude	Longitude
077216FB	10.470463800	07721777	10.470474204	1F5EB	10000	1F5EC	00000
0A8B62AB	14.828166990	0A8B6333	14.828178389	1C559	00000	1C55A	10000
0CEEB4CD	18.186252611	0CEEB555	18.186264010	03F93	10000	03F94	00000
0EF44889	21.029388467	0EF44911	21.029399866	00A08	00000	00A09	10000
10BE3E66	23.545040097	10BE3EEF	23.545051580	164B5	10000	164B6	00000
125E1222	25.829246510	125E12AB	25.829257993	07062	00000	07063	10000
13DE22AB	27.938976316	13DE2333	27.938987715	14081	10000	14082	00000
15453222	29.911354054	154532AB	29.911365537	1E1BE	00000	1E1BF	10000
1697EEEF	31.772094732	1697EF77	31.772106132	05CE0	10000	05CE1	00000
17D9C222	33.539932240	17D9C2AB	33.539943723	0B84C	00000	0B84D	10000
190D2DDE	35.228988659	190D3E66	35.229000058	0F8D4	10000	0F8D5	00000
1A346222	36.850250233	1A3462AA	36.850261716	12238	00000	12239	10000
1B50C444	38.412414528	1B50C4CD	38.412426012	13770	10000	13771	00000
1C63AE66	39.922565427	1C63AEEF	39.922576910	13AE7	00000	13AE8	10000
1D6E2F77	41.386516532	1D6E3000	41.386528051	12E99	10000	12E9A	00000
1E712A21	42.809131611	1E712AAA	42.809143094	1142F	00000	11430	10000
1F6D5EEF	44.194541937	1F6D5F77	44.194553336	0ED12	10000	0ED13	00000
2063719A	45.546260867	20637222	45.546272267	0BA75	00000	0BA76	10000
2153F000	46.867332458	2153F089	46.867343942	07D62	10000	07D63	00000
223F54CD	48.160388963	223F5555	48.160400363	036BF	00000	036C0	10000
23260C44	49.427753426	23260CCD	49.427764909	1E757	10000	1E758	00000
240876EF	50.671497351	24087777	50.671508750	18FDF	00000	18FE0	10000
24E6E889	51.893417398	24E6E911	51.893428797	130F4	10000	130F5	00000
25C1ADDE	53.095161449	25C1AE66	53.095172849	0CB26	00000	0CB27	10000
26990A22	54.278171528	26990AAB	54.278183011	05EF3	10000	05EF4	00000
276D3B33	55.443775160	276D3BBC	55.443786643	1ECCF	00000	1ECD0	10000
283E799A	56.593185458	283E7A22	56.593196858	17524	10000	17525	00000
290CF6EF	57.727466589	290CF777	57.727477988	0F84F	00000	0F850	10000
29D8E2A9	58.847637045	29D8E333	58.847648604	076A9	10000	076AA	00000
2AA26666	59.954589810	2AA266EF	59.954601293	1F080	00000	1F081	10000
2B69A99A	61.049171481	2B69AA22	61.049182881	1661E	10000	1661F	00000
2C2ED089	62.132160226	2C2ED111	62.132171625	0D7C7	00000	0D7C8	10000
2CF1FC44	63.204265572	2CF1FCCD	63.204277055	045B9	10000	045BA	00000
2DB34C44	64.266162850	2DB34CCD	64.266174333	1B02F	00000	1B030	10000
2E72DC44	65.318447091	2E72DCCD	65.318458574	1175D	10000	1175E	00000
2F30C777	66.361701926	2F30C800	66.361713409	07B76	00000	07B77	10000
2FED26EF	67.396465307	2FED2777	67.396476706	1DCA9	10000	1DCAA	00000
30A81111	68.423217768	30A8119A	68.423229251	13B20	00000	13B21	10000
31619B33	69.442417128	31619BBC	69.442428611	09703	10000	09704	00000
3219DA22	70.454509724	3219DAAB	70.454521207	1F079	00000	1F07A	10000
32D0DEEF	71.459861761	32D0DF77	71.459973160	147A2	10000	147A3	00000
3386BAAB	72.458839444	3386BB33	72.458850844	09C9E	00000	09C9F	10000
343B7C44	73.451763131	343B7CCD	73.451774614	1EF89	10000	1EF8A	00000
34EF319A	74.438930545	34EF3222	74.438941944	1407D	00000	1407E	10000
35A1E4CD	75.420558946	35A1E555	75.420570376	08F8D	10000	08F8E	00000
36539EEF	76.396842962	36539F77	76.396854362	1DCCA	00000	1DCCB	10000
370464CD	77.367885606	37046555	77.367897006	1283D	10000	1283E	00000
37B43889	78.333732644	37B43911	78.333744043	071EA	00000	071EB	10000
38631555	79.294280978	386315DE	79.294292461	1B9C8	10000	1B9C9	00000
3910ECCD	80.249221819	3910ED55	80.249233218	0FFBC	00000	0FFBD	10000
39BDA555	81.198005648	39BDA5DE	81.198017132	04396	10000	04397	00000
3A690D55	82.139568301	3A690DDE	82.139579784	184F9	00000	184FA	10000
3B12CB33	83.071987135	3B12CBBC	83.071998619	0C33D	10000	0C33E	00000
3BBA3A22	83.991725910	3BBA3AAB	83.991737394	1FD2D	00000	1FD2E	10000
3C5E0DDE	84.891654979	3C5E0E66	84.891666379	1305A	10000	1305B	00000
3CFB4BBC	85.755409263	3CFB4C44	85.755420662	1572E	00000	1572F	10000
3D894889	86.535369912	3D894911	86.535381312	16168	10000	16169	00000
3DDDDD99	86.999994228	3DDDDE20	87.000005543	1FFFF	00000	00000	10000

5. Computation of NL(lat) function for latitudes in the vicinity of 87 degrees.

$$
\begin{aligned}
NL(lat_{lower}) &= floor\left(\frac{2 \cdot p}{\arccos\left(1 - \dfrac{1-\cos\left(\dfrac{p}{2 \cdot NZ}\right)}{\cos^2\left(\dfrac{p}{180°} \cdot \left|86.999994228°\right|\right)}\right)}\right) \\
&= floor\left(\frac{2 \cdot p}{\arccos\left(1 - \dfrac{1-\cos(6°)}{\cos^2(86.999994228°)}\right)}\right) \\
&= floor\left(\frac{2 \cdot p}{\arccos\left(1 - \dfrac{1-0.994521895}{(0.052336055685)^2}\right)}\right) \\
&= floor\left(\frac{2 \cdot p}{\arccos\left(1 - \dfrac{0.00547810463}{0.00273906285}\right)}\right) \\
&= floor\left(\frac{2 \cdot p}{\arccos(-0.9999922311)}\right) \\
&= floor\left(\frac{360°}{179.77531564°}\right) \\
&= floor(2.002499613) \\
&= 2
\end{aligned}
$$

Thus lat_{lower} is in the band of latitudes for which there are 2 longitude zones; it is just below the transition latitude between 2 longitude zones and 1 longitude zone.

$$NL(87° exactly) = floor\left(\frac{2\cdot p}{\arccos\left(1-\frac{1-\cos\left(\frac{p}{2\cdot NZ}\right)}{\cos^2\left(\frac{p}{180°}\cdot|87°|\right)}\right)}\right)$$

$$= floor\left(\frac{2\cdot p}{\arccos\left(1-\frac{1-\cos(6°)}{\cos^2(87°)}\right)}\right)$$

$$= floor\left(\frac{2\cdot p}{\arccos\left(1-\frac{1-0.994521895}{\sin^2(3°)}\right)}\right)$$

$$= floor\left(\frac{2\cdot p}{\arccos\left(1-\frac{0.00547810463}{0.00273905232}\right)}\right)$$

$$= floor\left(\frac{360°}{\arccos(-1.000000000+e)}\right)$$

Depending on the slight round-off error in computation, ε, the inverse cosine function may either return an angle very slightly smaller than 180° (if ε is positive), or it may be *undefined* (if ε is negative). In the former case, the NL(lat) function will return 2, but in the latter case, the computation breaks down.

Let's look at this again, a little closer, using trigonometric identities:

$$NL(87°exactly) = floor\left(\frac{2\cdot p}{\arccos\left(1-\dfrac{1-\cos\left(\dfrac{p}{2\cdot NZ}\right)}{\cos^2\left(\dfrac{p}{180°}\cdot|87°|\right)}\right)}\right)$$

$$= floor\left(\frac{2\cdot p}{\arccos\left(1-\dfrac{1-\cos(6°)}{\cos^2(87°)}\right)}\right)$$

$$= floor\left(\frac{2\cdot p}{\arccos\left(1-\dfrac{1-\cos(2\cdot 3°)}{\sin^2(3°)}\right)}\right)$$

$$= floor\left(\frac{2\cdot p}{\arccos\left(1-\dfrac{1-(1-2\cdot\sin^2(3°))}{\sin^2(3°)}\right)}\right)$$

$$= floor\left(\frac{360°}{\arccos(-1)}\right) = floor\left(\frac{360°}{180°}\right)$$

$$= floor(2) = 2$$

It appears, then, that the defining formula for the NL(lat) function returns 2 if the latitude is exactly ±87°. However, it returns 1 if the latitude is ever-so-slightly greater than 87°.

Now, the closest possible 32-bit angular weighted binary numeral to 87° is hexadecimal 3DDDDDDE (rounded off from hex 3DDDDDDD**.**DDD**…**), and that is greater than 87° exactly. Thus

NL(87° represented in 32-bit angular weighted binary) = 1,

whereas

NL(87° represented in decimal degrees) = 2.

6. We need to settle on a definition for the value to be returned by the NL(lat) function when the latitude is exactly 87°. In doing so, we have two choices:

(a) We may choose the smaller of the values, as if the latitude were slightly larger than the transition latitude: i.e., NL(87°) = 1.

(b) We may choose the larger of the values, as if the latitude were slighter smaller than the transition latitude: i.e., NL(87°) = 2.

The former choice, (a), favours sources of latitude and longitude, such as ARINC 743A GNSS receivers, that deliver the latitude in angular weighted binary notation. For such devices, the closest possible representation of 87° will be the 32-bit angular weighted binary representation of that angle, (hexadecimal 3DDDDDDE) $\times$ (2^{-32} circles), which is slightly greater than 87° exactly.

The former choice, (a), also has the advantage that the transition at latitude 87° for the NL(lat) function behaves similarly to the transition at latitude +87° (87° N) for the even CPR encoding of the latitude. In the even surface position CPR encoding, latitude 87° N is encoded as hexadecimal 00000 (rather than 1FFFF), and therefore may be considered to grouped with positions immediately to the north of it. In a similar way, NL(87° N) would be 1 so that the transition latitude, 87°, is grouped with nearby latitudes that are closer to the pole.

The latter choice, (b) favours sources of latitude and longitude that provide latitude and longitude in degrees (or degrees, minutes, and seconds); such devices can represent the angle of 87° exactly. I think, however, that our navigation data sources are very likely to provide latitude as ARINC 429 binary labels in angular weighted binary, rather than as BCD labels in decimal degrees.

It is desirable, I think, for us to define the value that the NL( ) function returns in such a way that it doesn't matter whether or not the navigation data source uses angular weighted binary notation or decimal degrees to return the own-ship latitude. Thus, I recommend that we treat the latitudes of ±87° as special cases, just as latitudes of 0° and ±90° are treated as special cases.

I conferred once again with Randy Jacobson and Ed Bayliss, on Wednesday, 2 January 2002. As a result of that conversation, we three recommend that the **NL(lat) function should return 1 when the latitude is ±87° exactly.**

7. Recommended changes to Appendix A. The text of A.7.2, subparagraph d, would be changed to read as follows:

d. The notation **NL(*x*)** denotes the "number of longitude zones" function of the latitude angle *x*. The value returned by **NL**(*x*) is constrained to the range from 1 to 59. **NL**(*x*) is defined for most latitudes by the equation,

$$NL(lat) = \mathit{floor}\left(\frac{2 \cdot p}{\arccos\left(1 - \dfrac{1 - \cos\left(\dfrac{p}{2 \cdot NZ} \right)}{\cos^2\left(\dfrac{p}{180^\circ} \cdot |lat| \right)} \right)} \right) ,$$

where *lat* denotes the latitude argument in degrees. For latitudes at or near the N or S pole, where the above formula would either be undefined or yield **NL**(*lat*) = 0, the value returned by the NL( ) function shall be 1. Likewise, at the equator, where the above formula might otherwise yield **NL**(*lat*) = 60, the value returned by the NL( ) function shall be 59. At latitudes of exactly 87° N or 87° S, where the above formula might otherwise yield **NL**(*lat*) = 2, the value returned by the NL( ) function shall be 1.

Note 5: *This equation for NL( )is impractical for a real-time implementation. A table of transition latitudes, such as Table A-21 below, can be pre-computed using the following equation:*

$$lat = \frac{180^\circ}{p} \cdot \arccos\left(\sqrt{\frac{1 - \cos\left(\dfrac{p}{2 \cdot NZ} \right)}{1 - \cos\left(\dfrac{2p}{NL} \right)}} \right) \textit{for } NL = 2 \textit{ to } 4 \times NZ\text{-}1,$$

and a table search procedure used to obtain the return value for NL( ).

Table A-21. Look-Up Table for Number of Longitude Zones, NL.

Condition	Transition Latitude		Number of Longitude Zones, NL	
	Degrees (decimal)	32-bit AWB (hexadecimal)		
If |lat| <	10.47047130	07 72 17 54	**Then NL(lat) =**	59
Else if |lat| <	14.82817437	0A 8B 63 03	**Then NL(lat) =**	58
Else if |lat| <	18.18626357	0C EE B5 50	**Then NL(lat) =**	57
Else if |lat| <	21.02939493	0E F4 48 D6	**Then NL(lat) =**	56
Else if |lat| <	23.54504487	10 BE 3E 9F	**Then NL(lat) =**	55
Else if |lat| <	25.82924707	12 5E 12 29	**Then NL(lat) =**	54
Else if |lat| <	27.93898710	13 DE 23 2C	**Then NL(lat) =**	53
Else if |lat| <	29.91135686	15 45 32 43	**Then NL(lat) =**	52
Else if |lat| <	31.77209708	16 97 EF 0B	**Then NL(lat) =**	51
Else if |lat| <	33.53993436	17 D9 C2 3B	**Then NL(lat) =**	50
Else if |lat| <	35.22899598	19 0D 3E 35	**Then NL(lat) =**	49
Else if |lat| <	36.85025108	1A 34 62 2C	**Then NL(lat) =**	48
Else if |lat| <	38.41241892	1B 50 C4 78	**Then NL(lat) =**	47
Else if |lat| <	39.92256684	1C 63 AE 77	**Then NL(lat) =**	46
Else if |lat| <	41.38651832	1D 6E 2F 8C	**Then NL(lat) =**	45
Else if |lat| <	42.80914012	1E 71 2A 88	**Then NL(lat) =**	44
Else if |lat| <	44.19454951	1F 6D 5F 49	**Then NL(lat) =**	43
Else if |lat| <	45.54626723	20 63 71 E6	**Then NL(lat) =**	42
Else if |lat| <	46.86733252	21 53 F0 01	**Then NL(lat) =**	41
Else if |lat| <	48.16039128	22 3F 54 E9	**Then NL(lat) =**	40
Else if |lat| <	49.42776439	23 26 0C C7	**Then NL(lat) =**	39
Else if |lat| <	50.67150166	24 08 77 22	**Then NL(lat) =**	38
Else if |lat| <	51.89342469	24 E6 E8 E0	**Then NL(lat) =**	37
Else if |lat| <	53.09516153	25 C1 AD DF	**Then NL(lat) =**	36
Else if |lat| <	54.27817472	26 99 0A 48	**Then NL(lat) =**	35
Else if |lat| <	55.44378444	27 6D 3B A2	**Then NL(lat) =**	34
Else if |lat| <	56.59318756	28 3E 79 B3	**Then NL(lat) =**	33
Else if |lat|	57.72747354	29 0C F7 42	**Then NL(lat) =**	31
Else if |lat| <	58.84763776	29 D8 E2 B2	**Then NL(lat) =**	30
Else if |lat| <	59.95459277	2A A2 66 89	**Then NL(lat) =**	30
Else if |lat| <	61.04917774	2B 69 A9 E5	**Then NL(lat) =**	29
Else if |lat| <	62.13216659	2C 2E D0 D5	**Then NL(lat) =**	28
Else if |lat| <	63.20427479	2C F1 FC B2	**Then NL(lat) =**	27

Condition	Transition Latitude		Number of Longitude Zones, NL	
	Degrees (decimal)	32-bit AWB (hexadecimal)		
Else if |lat| <	64.26616523	2D B3 4C 60	**Then NL(lat) =**	26
Else if |lat| <	65.31845310	2E 72 DC 8C	**Then NL(lat) =**	25
Else if |lat| <	66.36171008	2F 30 C7 D8	**Then NL(lat) =**	24
Else if |lat| <	67.39646774	2F ED 27 0C	**Then NL(lat) =**	23
Else if |lat| <	68.42322022	30 A8 11 2E	**Then NL(lat) =**	22
Else if |lat| <	69.44242631	31 61 9B A1	**Then NL(lat) =**	21
Else if |lat| <	70.45451075	32 19 DA 2E	**Then NL(lat) =**	20
Else if |lat| <	71.45986473	32 D0 DF 12	**Then NL(lat) =**	19
Else if |lat| <	72.45884545	33 86 BA F3	**Then NL(lat) =**	18
Else if |lat| <	73.45177442	34 3B 7C CB	**Then NL(lat) =**	17
Else if |lat| <	74.43893416	34 EF 31 C5	**Then NL(lat) =**	16
Else if |lat| <	75.42056257	35 A1 E4 F8	**Then NL(lat) =**	15
Else if |lat| <	76.39684391	36 53 9E FA	**Then NL(lat) =**	14
Else if |lat| <	77.36789461	37 04 65 38	**Then NL(lat) =**	13
Else if |lat| <	78.33374083	37 B4 38 EB	**Then NL(lat) =**	12
Else if |lat| <	79.29428225	38 63 15 64	**Then NL(lat) =**	11
Else if |lat| <	80.24923213	39 10 ED 48	**Then NL(lat) =**	10
Else if |lat| <	81.19801349	39 BD A5 B3	**Then NL(lat) =**	9
Else if |lat| <	82.13956981	3A 69 0D 67	**Then NL(lat) =**	8
Else if |lat| <	83.07199445	3B 12 CB 8A	**Then NL(lat) =**	7
Else if |lat| <	83.99173563	3B BA 3A 96	**Then NL(lat) =**	6
Else if |lat| <	84.89166191	3C 5E 0E 31	**Then NL(lat) =**	5
Else if |lat| <	85.75541621	3C FB 4C 0F	**Then NL(lat) =**	4
Else if |lat| <	86.53536998	3D 89 48 8A	**Then NL(lat) =**	3
Else if |lat| <	87.00000000	3D DD DD DE	**Then NL(lat) =**	2
Else			**NL(lat) =**	1