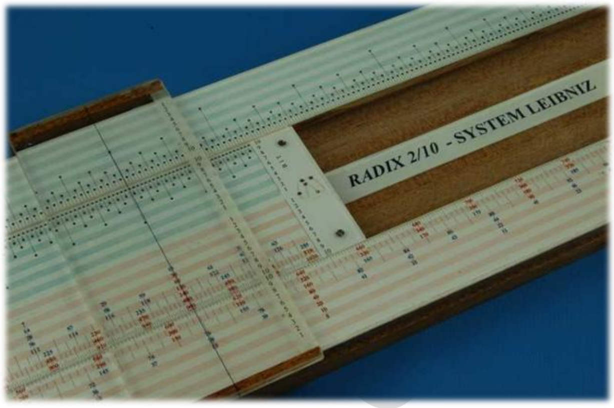
October 2013

RADIX 2/10 – SYSTEM LEIBNIZ

10 Bit Binary/Decimal Desktop Slide Rule *By C Tombeur*

OVERVIEW



The Radix 2/10 is a logarithmic 10 bit binary desktop slide rule with decimal equivalent scales. Where a 'normal' slide rule is used to perform decimal multiplication and division, the Radix 2/10 does the same for binary numbers, as well as allowing conversion between binary and decimal numbers.

The series is named Radix, radix being another term for the base of a number. The binary/decimal (base 2/base 10) system of the Radix 2/10 is especially named SYSTEM LEIBNIZ after Gottfried Leibniz who discovered the modern binary number system. Other models with any primary/equivalent base pairing are possible in this series, eg Radix 8/16 would be a base 8 slide rule with base 16 equivalent scales.

SCALE LAYOUT

The closed frame, mahogany/Perspex rule features a pair of logarithmic binary primary scales and a pair of decimal equivalent scales printed on paper, and a hairline cursor. The binary scales are positioned on the upper rail of the stock above the slide and on the adjacent upper half of the slide (BIN1 and BIN2 respectively), with the decimal scales on the lower half of the slide and the adjacent stock lower rail (DEC1 and DEC2 respectively). Each scale comprises 10 BIT-lines, with each line representing a bit in a binary number. The BIT-lines are 390mm long giving a total scale length of 3.9m. BIT-lines of the BIN1 and DEC1 scales are ordered from the most significant bit at the top to least significant bit at the bottom, whereas the BIT-lines of the BIN2 and DEC2 scales are in the reverse order (bottom to top, most to least significant bit). As such, the least significant BIT-line of each scale pair is adjacent. This layout of a primary scale pair in one base and an equivalent scale in pair a different base, where the scales comprise multiple lines representing digits, is the defining feature of the Radix model series.

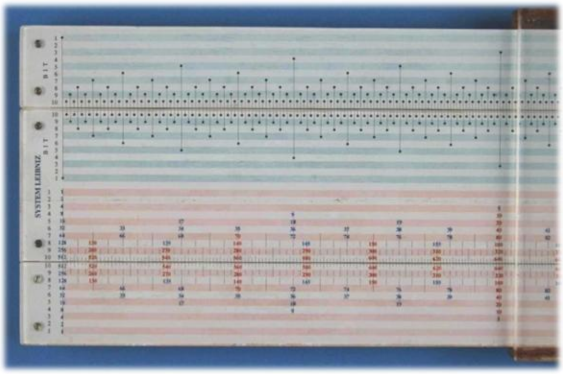
Alternate BIT-lines are shaded blue on the BIN scales and red on the DEC scales for ease of tracking along the lines. BIT-lines on each scale are labelled in black from 1 (most significant) to 10 (least significant) at both ends, and on the right-hand end of the cursor pane.

Both pairs of scales have all integers from 1 to 1024 indicated with tick marks (10 bits = 2^{10} = 1024). The BIN scale tick marks indicate a '1' bit value with a •. The DEC scales are labelled as follows: all 1-6 bit integers (1-63); even 7-BIT integers (64-127); 8-BIT integers divisible by 5 (128-255); 9-BIT integers divisible by 10 (256-511); 10-BIT integers divisible by 20 (512-1023). All powers of 2 at the start of each BIT-line are also labelled on the DEC scales. Integer labels are blue, except multiples of 10 which are red for ease of location.

The accuracy of the Radix 2/10 is limited to 10 bits, but operations on numbers with more than 10 bits can be performed in a similar way to a 'standard' slide rule.

The Radix 2/10 initially appears somewhat unfamiliar, confusing and complicated, so special care should be taken when reading the scales to avoid errors!

Reading the Scales and Conversions



Conversions between binary and decimal numbers effectively demonstrate how the scales are read. Either of the fixed scale pairs, BIN1/DEC2 on the stock or BIN2/DEC1 on the slide, can be used for conversion. The scales on the slide are closer together, but it is preferable to use the scales on the stock for conversion as it is easier to read the BIN scales from most to least significant bit by scanning downwards.

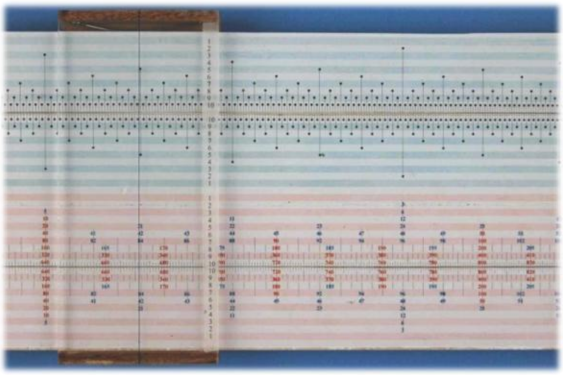
Note that while fractional bits can be read from the BIN scales, on the DEC scales there are no tick marks for any fractional component, which should be estimated if required.

To convert a **binary number to decimal**, first construct the binary number on the BIN1 scale. Position the cursor hairline over the left-hand index end of the 1-BIT-line (over the •). This represents the most significant '1' bit of the binary number, any leading '0' bits' are ignored. For each remaining bit in the number, if it is a '0' move on to the next bit, if it is a '1' move the cursor hairline to the right from its last position, along the corresponding BIT-line until it reaches a •, then position the hairline over the tick mark. When complete, note the number of integer bits in the original binary number, ignoring any leading '0' bits. Refer to the DEC2 scale. On the BIT-line corresponding to the noted number of integer bits, read the decimal integer at the tick mark under (or immediately to the left of) the cursor hairline. With practice it is possible to build the binary number moving the cursor only once across to its final position.

Example 1: convert 00001010_2 to base 10.

Set the cursor hairline over the left-hand index end of the BIN1 scale on the stock upper rail. Ignore the leading '0' bits. The • under the hairline on the 1-BIT line represents the 1^{st} '1' bit in the binary number. The 2^{nd} bit is a '0' and so is ignored. The 3^{rd} bit is a '1', so move the cursor hairline to the right along the 3-BIT line of the BIN1 scale until a • is reached, and align the hairline over it. The final 4^{th} bit is a '0' and so is also ignored. In the original binary number, 00001010, ignoring the leading '0's, there are 4 integer bits. Refer now to the DEC2 scale on the stock lower rail. Under the cursor hairline on the 4-BIT line, read the answer of '10'. Therefore $1010_2 = 10_{10}$.

To convert a **decimal number to binary**, first locate the integer tick mark for the number on the DEC2 scale and position the cursor hairline over it (approximate any fractional component). Refer to the BIN1 scale. Scrutinise each BIT-line in turn starting from the 1-BIT line (most significant bit), and ending at either the 10-BIT line or a BIT-line where there is a • under the hairline. For each of these BIT-lines, look the range starting from underneath the hairline, and ending to the left at either the nearest • on the BIT-line or at a tick mark that crosses the BIT-line and extends to a lower BIT-line, whichever comes first. If there is a • directly under the hairline, this is both the start and end of the range. At the left end of each BIT-line range scrutinised, if there is a • write a '1' bit, if there is a crossing tick mark write a '0' bit. When complete, note the number of the BIT-line on the DEC2 scale in which the decimal number is located, this indicates the number of integer bits in the equivalent binary number. Append trailing '0's to the binary number written as required so that its length is equal to the number of integer bits noted (or insert a point after the noted number of bits if appropriate). The string of bits written is the equivalent binary to the decimal number. Note that the first bit will always be '1' from the leftmost end of the 1-BIT-line.



Example 2 : Convert 86_{10} to base 2.

Find the '86' label on the 7-BIT line of the DEC2 scale and position the cursor hairline over the tick mark. Refer to the BIN1 scale. Look in the 1-BIT line from beneath the hairline and to the left. There is no • under the hairline and no tick marks cross the BIT-line before the • at the left end, so write a '1' as represented by the •. Next look at the 2-BIT line, there is no • under the hairline nor to the left before the tick mark from the 1-BIT line crosses it at the left end, so write a '0'. The 3-BIT line there is clear under the hairline, but there is a • to the left of it before a crossing tick mark, so write a '1'. The 4-BIT line has no • under the hairline or to the left before the tick mark from the 3-BIT line crosses it, so write a '0'. The 5-BIT line, has nothing under the hairline but to the left of it there is a • before a crossing tick mark, so write a '1'. The 6-BIT line has a tick mark with a • under the hairline which means this is the last BIT-line that needs to be scrutinised, so write a final '1'. The original number '86' was found on the 7-BIT line of the DEC2 scale, meaning there are 7 integer bits in the equivalent binary number. The binary string written is '101011' which has six bits, so an additional '0' must be appended to give it the required 7 integer bits. Therefore 86_{10} is 1010110 in binary.

If the number has more than 10 bits, only the first 10 bits from the first '1' bit can be built on the BIN scales. Numbers on the DEC scales must be factored by 2 for each additional bit over 10.

MULTIPLICATION AND DIVISION

Multiplication and division are performed in similar way to using the C and D scales on a 'standard' logarithmic slide rule. Either the BIN, DEC or a combination of both scales can be used, and conversions read.

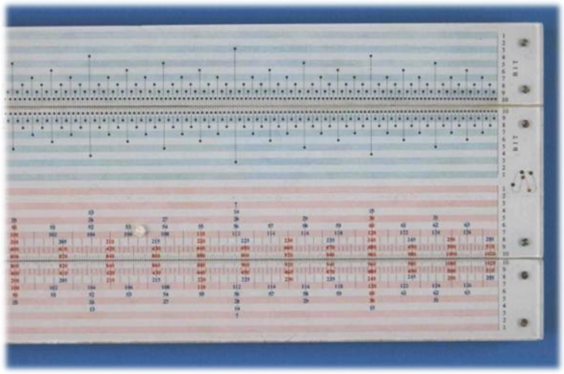
To **multiply** two numbers, construct the first factor on the BIN1 scale (or locate it on the DEC2 scale) and position the cursor hairline over the tick mark. Move the slide so that either its left or right-hand index end is underneath the hairline (as appropriate to enable the answer to be read in the stock scale range). Construct the second factor on the BIN2 scale (or locate it on the DEC1 scale) and position the hairline over the tick mark. Read the answer on the BIN1 scale (or the DEC2 scale), determining the number of integer bits as follows:

(№ of integer bits in answer) = (№ integer bits in 1^{st} factor) + (№ of integer bits in 2^{nd} factor)
- (0 if slide protrudes to the left, or 1 if slide protrudes to the right).

Example 3: What is 10011_2 multiplied by 11_2 in binary, and what is the completed sum in decimal?

Construct the 5 bit first factor on the BIN1 scale. The • on the left hand end of the 1-BIT line represents the 1^{st} '1' bit in the factor, so position the cursor hairline over it. Ignore the 2^{nd} and 3^{rd} bit as they are '0's. The 4^{th} bit is a '1' so move the hairline to the right along the 4-BIT line until it is over a •. The last (5^{th}) bit is also a '1', so continue moving the hairline until it is over a • on the 5-BIT line. Refer to the 5-BIT line on the DEC2 scale and read the decimal equivalent of the first factor to be '19'. Move the slide so that the left-hand index end is under the hairline. Now construct the second factor on the BIN2 scale. The • on the 1-BIT line under the hairline represents the 1^{st} '1' bit. The 2^{nd} (last) bit is also a '1', so move the hairline to the right along the 2-BIT line until it is over a •. Refer to the 2-BIT line on the DEC1 scale, and under the hairline read the decimal equivalent of the second factor to be '3'. Read the answer to the multiplication on the BIN1 scale. The first 3 BIT-lines have a • to the left of the hairline, so write '111'. The 4-BIT and 5-BIT lines have the tick mark from the 3-BIT line crossing them first to the left of the hairline, so write '00'. On the 6-BIT line there is a • under the hairline, so write a final '1'. The first factor has 5 bits, the second factor has 2 bits and the slide protrudes to the right, so there are 5+2-1=6 integer bits in the result. The binary string written, '111001', also has 6 bits and so is the final answer. Refer to the DEC2 scale to read the decimal equivalent of the answer to be '57' under the hairline on the 6-BIT line. The decimal sum is therefore 19 x 3 = 57.

To **divide** two numbers, construct the dividend on the BIN1 scale (or locate it on the DEC2 scale) and position the cursor hairline over the tick mark. Construct the divisor on the BIN2 scale (or locate it on the DEC1 scale) and move the slide so that the tick mark is underneath the hairline. Position the hairline over the index end of the slide that is inside the stock scale range. Read the answer on the BIN1 scale (or the DEC2 scale), determining the number of integer bits as follows:



(№ of integer bits in answer) = (№ integer bits in dividend) - (№ of integer bits in divisor)
+ (0 if slide protrudes to the left, or 1 if slide protrudes to the right).

Example 4: what is 100001_2 divided by 1011_2 in binary?

First construct the dividend 100001 on the BIN1 scale. Set the cursor hairline over the • at the left-hand end of the 1-BIT line, representing the 1st '1' bit. The 2nd to 5th bits are all '0' so ignore them. The last bit is a '1', so move the hairline to the right along the 6-BIT line until it is over a •. Construct the divisor on BIN2 scale of the slide by moving the slide rather than the cursor hairline. Move the slide under the cursor so the • on the left-hand end of the 1-BIT line is under the hairline, representing the 1st bit. The 2nd bit is '0' so ignore it. The 3rd bit is a '1' so move the slide to the left until there is a • on the 3-BIT line under the hairline. The final bit is also a '1', so continue moving the slide to the left until a • is under the hairline on the 4-BIT line. Now move the hairline over the right-hand index end of the slide. Refer to the BIN1 scale and read the answer. There is a • to the left of the hairline at the far end of the 1-BIT line, so write a '1'. On the 2-BIT line there is a • under the hairline, so finish by writing '1'. The dividend has 6 bits, the divisor has 4 bits and the slide protrudes to the left, so the answer has 6-4+0=2 integer bits. The number of bits written out is also 2, so the answer is 11_2.

Example 5: What is 355_{10} divided by 113_{10} in binary?

Locate the tick mark for 355 on the 9-BIT line of the DEC2 scale and position the cursor hairline over it. Locate the tick mark for 113 on the 7-BIT line of the DEC1 scale and move the slide so that the hairline is over it, thus aligning 355 on DEC2 with 113 on DEC1. Move the cursor so the hairline is over the right-hand index end of the slide. Now refer to the BIN1 scale to read the binary answer. On the 1-BIT line there is a •to the left of the hairline at the extreme left end, so write a '1'. On the 2-BIT line there is also a • to the left of the hairline, so write another '1'. The 3-BIT and 4-BIT lines have a tick mark from the 3-BIT line crossing them first to the left of the hairline so write '00'. The 5-BIT line has a • to the left of the hairline, so write a '1'. The 6-BIT and 7-BIT lines both have the tick mark from the 5-BIT line crossing them first to the left of the hairline, so write '00'. The 8-BIT line has a • left of the hairline so write a '1'. Finally the 9-BIT and 10-BIT lines both have the tick mark from the 8-BIT line crossing them first to the left of the hairline, so again write '00'. At no stage is there a • under the hairline and there are no more BIT-lines to scrutinise. The 10 bit binary string written is '1100100100'. The number of integer bits in the dividend is 9 (355 is on the 9-BIT line of DEC2) and the number of integer bits in the divisor is 7 (113 is on the 7-BIT line of DEC1), and the slide protrudes to the left. The number of integer bits in the answer is therefore 9-7+0=2 bits, so a point must be placed after the 2nd bit in the binary string, making the answer 11.001001.

When calculating with numbers greater than 10 bits, only the first 10 bits from the first '1' bit are used. The full number of bits in each number are then used in the rules to determine the magnitude of the final answer, as described above.