Application Note 200
A Programmatic Approach to Calculating DS1085 Programmable-Oscillator Register Values

Introduction

The DS1085, unlike previous members of the EconOscillator™ family, is unique in that it allows the designer to generate any frequency between 8.2kHz to 133MHz by programming the master oscillator and then dividing it using a prescaler/divider chain. Whereas with previous family members such as the DS1077, the designer would purchase say a 100MHz version of the part and then program the prescaler and divider chain using the 2-wire serial interface. While this is fine if the desired frequency can be obtained simply by an integer divide of the available master frequencies, the DS1085 on the other hand allows the designer to program a unique master frequency in addition to the prescaler/divider chain. Since the DS1085 master frequency is user programmable, the designer no longer needs to select a fixed master frequency. Instead, the designer is now given a choice of step size. Master frequency step sizes of 10kHz, 25kHz, and 50kHz are available. The small, 8-pin DS1085 all-silicon solution is obviously the superior part when a design calls for unique frequencies. Unique (although common) frequencies such as 11.0592MHz can now be generated easily using the DS1085. Finally, the 5V DS1085 is also available in a 3V version, the DS1085L.

This application note will show how to calculate the values needed to program into the DS1085 registers to generate a specified frequency. The reason why an application note has been dedicated to a task that is usually simple is because, first of all, part of the calculation is interactive , meaning that one of the internal registers (a factory programmed default that can vary from device to device or lot to lot) must be read before the other registers can be calculated. Second, there are often many register combinations capable of generating a given frequency. And finally, to show how a solution can be generated with no user intervention required.

The particular DS1085 used in this application note is the DS1085-10, which is the 5V, 10kHz step-size version. Throughout the remainder of this document, it will simply be referred to the DS1085 for easier reading. Where appropriate, differences will be pointed out so that minor modifications can be made to use other flavors of the device.

Inside the DS1085

Before going much further, it is beneficial to look at the DS1085 block diagram located in the data sheet. Items of importance are the internal oscillator, prescaler, and the divider. Notice that one prescaler/divider goes to OUT0 and another prescaler/divider to OUT1. This application note will only discuss OUT1 since it is capable of generating a much wider range of frequencies.

In order to provide the amount of flexibility that the DS1085 provides, several registers are required, including four of which are two bytes wide. The registers of importance are DAC (two bytes), OFFSET (one byte), MUX (two bytes), DIV (two bytes), and RANGE (two bytes). The ADDR has no effect on the frequency and will not be discussed. Brief descriptions of the important registers follows.

OFFSET

This one byte register, containing a right justified 5-bit value is used to select a range of master frequencies. This register is used in conjunction with the DAC register. The OFFSET selects a range, while the DAC

selects a master frequency within that range. Table 1 is an excerpt of the Frequency vs. Offset Table from the data sheet. Please refer to the DS1085 or DS1085L data sheet for the official table. Several items to point out are, first, that the table shows frequencies outside of the master oscillator specification of 66.5MHz to 133MHz (for the 5V part). Second, notice that many of the ranges do have some overlap. Also, a formula can be written to calculate the range of frequencies given an offset (initial frequency = 51.2MHz, delta = 5.12MHz between offset values, and span 1023*step size), and the default offset. And finally, the value to be programmed into OFFSET depends on the factory default offset, referred to as OS.

During production testing and calibration, a default offset value is calculated and written into the OFFSET register. This value will vary from device to device and lot-to-lot. Since the OFFSET register is EEPROM, and to prevent the risk of loosing this factory default, a copy of it resides in the read-only RANGE register. It is therefore recommended that whenever programming the DS1085, begin by reading the RANGE register instead of the OFFSET register since you may have programmed the device before, overwriting the default. Otherwise, if you are positive that OFFSET has never been modified, it can be read as well to obtain the factory default offset.

In order to program the DS1085 to generate a desired frequency, the factory default offset must be read and saved until the new OFFSET is calculated. As an example, say we want to generate 82MHz. The default offset is read out of the RANGE register and returns say 12 decimal (after shifting and masking). Then looking at the Frequency vs. Offset Table, 82MHz can be generated in either OS-3 or OS-2. For this example we will choose OS-3 (since 82MHz is closer to the center of that range). Finally, the new value to be written into OFFSET is calculated by subtracting 3 (hence OS-3) from the this devices default offset, 12. The value 9 is then converted to binary, shifted and written to OFFSET.

Table 1. Frequency vs. Offset Excerpt

DS1085Z-10	
Offset	Frequency Range
OS - 10	–
OS - 9	–
OS - 8	51.20–61.43
OS - 7	56.32–66.55
OS - 6	61.44–71.67
OS - 5	66.56–76.79
OS - 4	71.68–81.91
OS - 3	76.8–87.03
OS - 2	81.92–92.15
OS - 1	87.04–97.27
OS*	92.16–102.39
OS + 1	97.28–107.51
OS + 2	102.4–112.63
OS + 3	107.52–117.75
OS + 4	112.64–122.87
OS + 5	117.76–127.99
OS + 6	122.88–133.11

* Factory default. This is the integer value of the 5MSBs of the RANGE register.

DAC
This two-byte register with a left justified 10-bit value is used in conjunction with the OFFSET register to select a master frequency. OFFSET selects a range, while DAC selects a specific frequency within that range. When DAC is zero, the master frequency will be the minimum frequency of the selected range. For example, looking at Table 1, if OFFSET is OS-1 and DAC is zero, the master frequency will be 87.04MHz. Each increment of DAC will increase the master frequency by the step size (10kHz in this case). If the DAC is set to 1023 for the same OFFSET, the master frequency will be 97.27MHz.

MUX
This two-byte register not only contains the prescalers of both outputs, but also contains bits for configuring the CTRL0 and CTRL1 inputs, shutdown mode, output enable, and also a bit for bypassing the divider (for OUT1). Since each prescaler divides by 1, 2, 4, or 8, only two bits are required for each. This application note refers to the OUT1 prescaler as M. Refer to the data sheet for the bit locations.

DIV
This two-byte register with a left-justified 10-bit value contains the value of N minus 2. Since DIV is a 10-bit number, values of DIV can be 0–1023. In terms of N, this is equivalent to 2–1025. In order to obtain N = 1, N must be bypassed using the DIV1 bit in the MUX register.

RANGE
This two-byte read-only register contains a left-justified 5-bit value that is a copy of the factory programmed default offset value. The remaining 11 bits are don't cares and should be ignored. This register needs to be read before calculating a new OFFSET. For more information, see to the section describing the OFFSET register above.

Determining the Output Frequency When Given Register Values

For a given set of register values, the output frequency can be calculated by first determining the master oscillator frequency, and then dividing by the prescaler and divider.

To calculate the master frequency the following must be known: the step size of the device, the programmed value of the OFFSET register, the factory default OFFSET (read either from the RANGE register or from a brand new part), and the value of the DAC register (see Equation 1).

Equation 1

$$\text{master frequency} = \text{minimum frequency of selected offset} + (\text{step size} * \text{DAC})$$

where minimum frequency of selected offset is read from the Frequency vs. Offset Table, step size is .01 for the 10kHz version, and DAC is the decimal representation of the DAC value.

Once the master frequency is known, M can be determined from the MUX register and N can be determined from DIV. However, it is important to read the DIV1 bit of the MUX register to determine if N is bypassed (N = 1), or enabled. The frequency at OUT1 can be calculated as shown in Equation 2.

Equation 2

$$\text{OUT1} = \frac{\text{master frequency}}{M \times N}$$

where master frequency is the result from Equation 1, and M, N are the decimal values of the prescaler and divider.

Determining the Register Values When Given a Desired Output Frequency

On the other hand, if a desired frequency is given, it is possible to calculate the values needed to program the DS1085 registers. However, it is very important to understand that for a given frequency, there are many possible register combinations that will generate the desired frequency. Furthermore, before even beginning to write a subroutine that will output just one of the many solutions, some guidelines must be defined. The following guidelines can be modified to suit a particular application.

Design considerations for this application

1) Only OUT1 will be considered.
2) Prefer smaller master oscillator frequencies (lower current consumption).
3) Prefer higher values of N and M since the effective step size decreases.
4) The 5V, 10kHz step size version will be used.
5) The first valid combination of registers will be output, no need to search entire range for best combination.

An example function prototype can look similar to:

```
int ProgramDS1085Frequency( double DesiredFrequency);
```

where DesiredFrequency is an input parameter and can be either an integer, float, or double depending on the desired resolution. The function will read RANGE, convert the frequency into register values, and write the new register values into the device. The return value of the function is an integer that can pass back pass/fail information to the calling program. Of course, this is just an example and can be tailored to fit any application.

Example

Calculate values needed to generate a 1MHz output at OUT1.

First, the default offset must be read through the 2-wire interface. If the device has never been programmed, the value can be read directly from the OFFSET register. Otherwise, if the OFFSET no longer contains the factory default value, then read the RANGE register that is read-only and programmed at the factory. Read the value, convert it (by shifting the bits accordingly), and then store it until the offset is calculated later.

The master frequency needs to be greater than 66.5MHz (but less than 133MHz), so try 67MHz for the master frequency. Using Equation 2 with a known OUT1 of 1MHz and trying 67MHz, M x N must equal 67. See Equation 3. With M = 1 and N = 67, we have a valid solution.

Equation 3

$$1\text{MHz} = \frac{67\text{MHz}}{1 \times 67}$$

This portion of the calculation, which is a trade-off between the master frequency and prescaler/divider, is mostly trial and error and goes back to the design guidelines. If current is of concern, then it is best to start at the beginning of the valid oscillator frequencies, 66.5MHz, and work upward until a master divided by the divider chain yields the desired frequency. Otherwise, if precision is of more importance than current consumption, then it is best to begin at 133MHz and work downward. The benefit of higher master

frequencies is that M x N will be larger, resulting in even a smaller step size at the output. Remember to provide sufficient decoupling.

Now that the master frequency, M, and N are known, find OFFSET using the Frequency vs. Offset Table in the data sheet or the example table here in Table 1.

To generate a master frequency of 67MHz using the 5V, -10 device, both OS-6 (61.44–71.67) and OS-5 (66.56–76.79) will work. It is wise to choose the range where the master frequency will be closest to the center of the range. In this case, OS-6 is the smart choice. This range can produce frequencies from 61.44 when the DAC register is 0, up to 71.67MHz when DAC is 1023. Now in order to convert OS-6 to a value that can be written into the OFFSET register, we need to use the default offset that was obtained earlier. Since we need OS-6, we simply subtract 6 from the default. The value read from the RANGE register of the device used in this example was 12. So for this device 12 - 6 = 6. Six will be written into OFFSET (once converted into hex and the three unused MSBs masked off).

Finally, calculate the value of DAC as shown in Equation 4.

Equation 4

$$\text{DAC} = \frac{\text{master frequency} - \text{minimum frequency of selected offset}}{\text{step size}} = \frac{67 - 61.44}{.01} = 556$$

The answer, 556, is then converted into hex and shifted left six places since the DAC register is left justified.

Flowchart

The flowchart of the algorithm used to perform the calculation is included in Appendix A (due to its size). Comments accompany the flowchart and also point out values that need to be changed for other variants of the device.

Improvements

The most obvious improvement that can be made to the software and flowchart included in this application note is the addition of the other variants, such as the DS1085L and other step sizes. Again, however, the intent of this application note was to describe one flavor in detail while pointing out where differences lie.

Another improvement would be to have an option in the software to choose between 66.5MHz or 133MHz as the starting point when searching for the master oscillator frequency. Fortunately, after having discussed the general procedure, adding this should be straightforward.

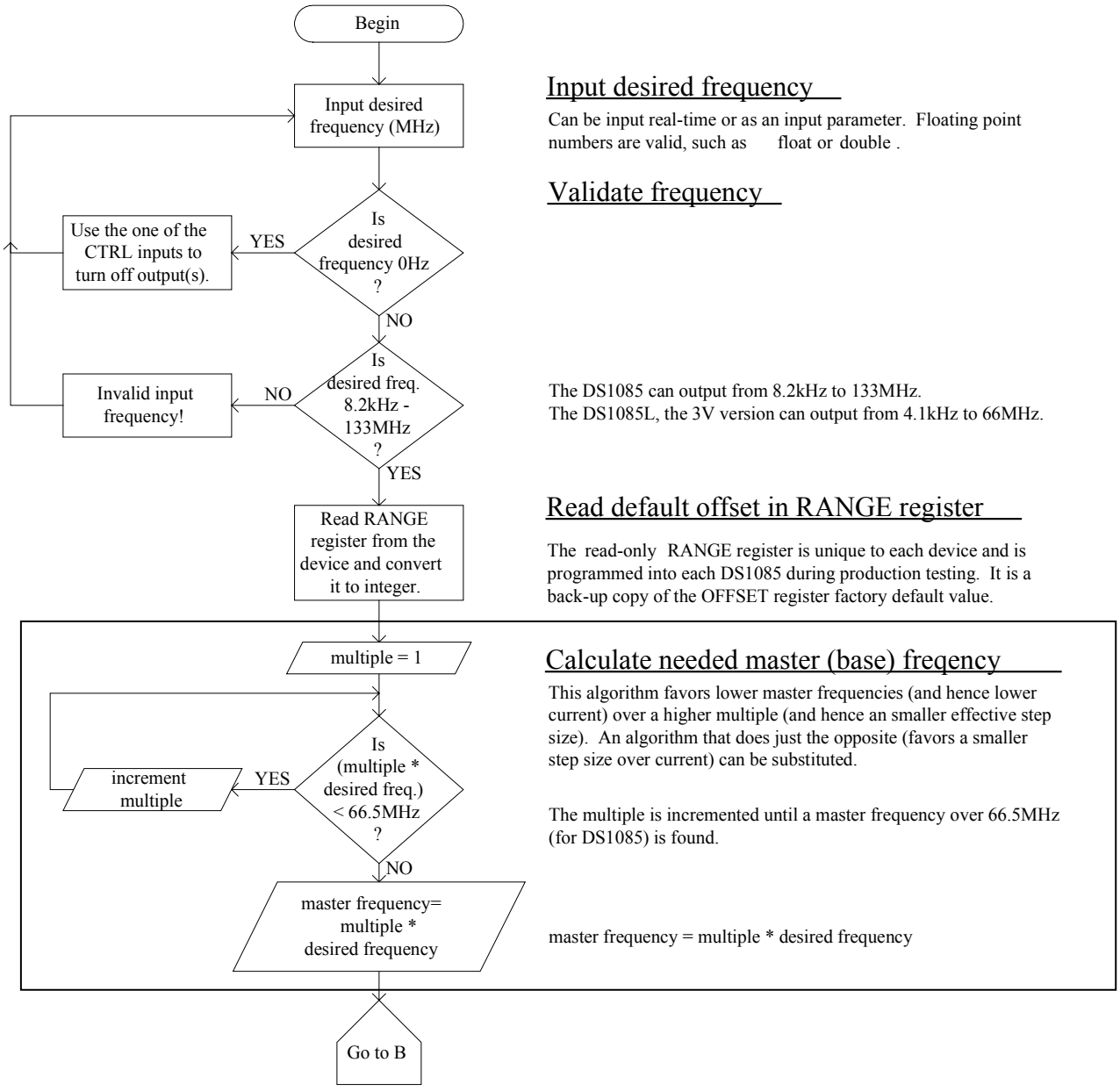
Probably the most important improvement, at least to some, would be to continue searching for a master frequency until the smallest percent error is found. And even better, maybe several valid solutions can be displayed. While such a task is not difficult, it sure would have added several pages and complexity to this application note and flowchart.

While many other improvements can be thought of, the vast majority are fairly application specific, and thus left as an exercise to the reader.

Conclusion

The DS1085 is a welcomed addition to the EconOscillator family. It is the most flexible oscillator to date. The challenge of working backwards and calculating the register values from a given frequency has been solved and explained in this application note. Best of all, using the provided flowchart makes a nice starting point for new projects.

Appendix A



Input desired frequency

Can be input real-time or as an input parameter. Floating point numbers are valid, such as float or double .

Validate frequency

The DS1085 can output from 8.2kHz to 133MHz.
The DS1085L, the 3V version can output from 4.1kHz to 66MHz.

Read default offset in RANGE register

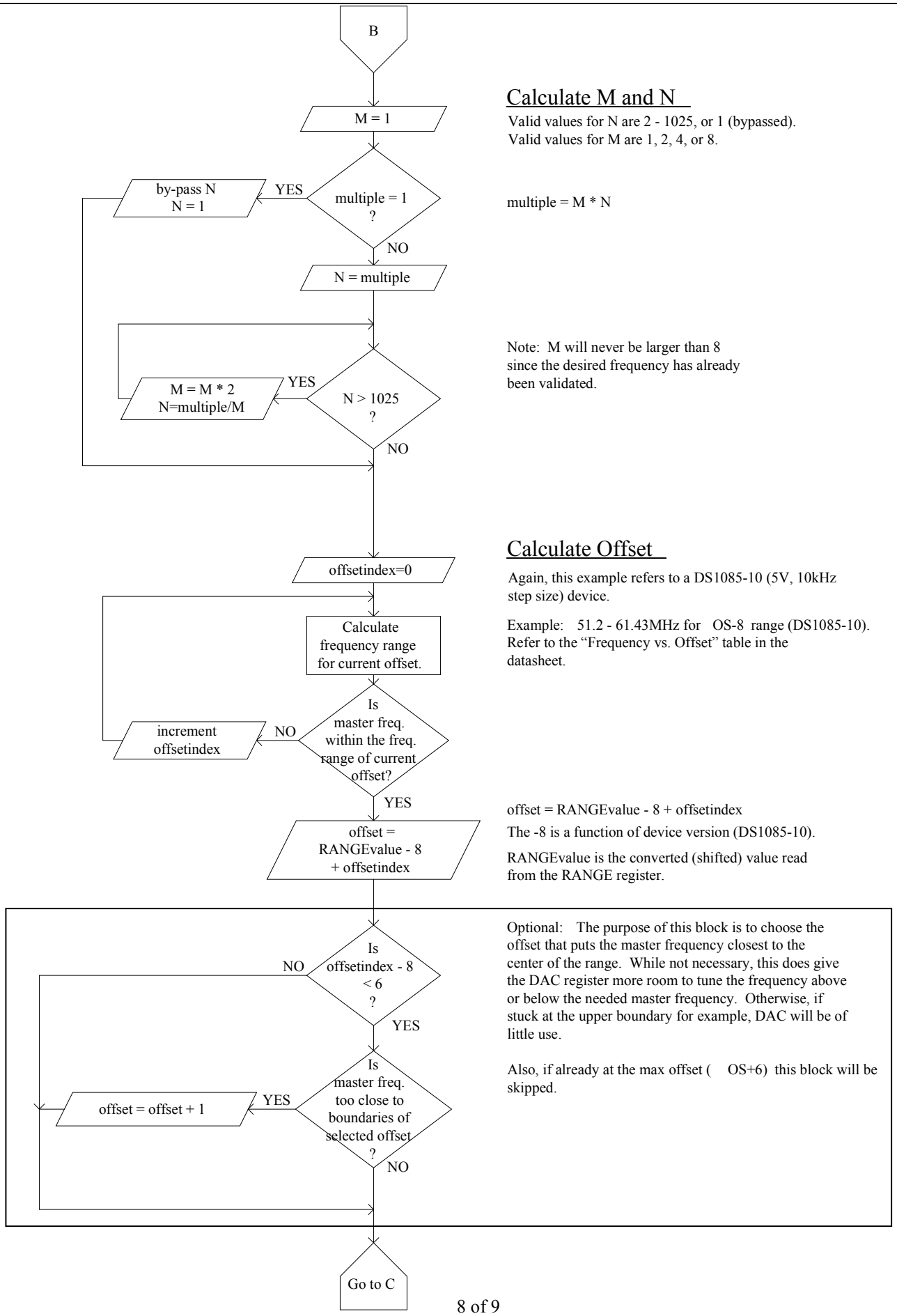
The read-only RANGE register is unique to each device and is programmed into each DS1085 during production testing. It is a back-up copy of the OFFSET register factory default value.

Calculate needed master (base) freqency

This algorithm favors lower master frequencies (and hence lower current) over a higher multiple (and hence an smaller effective step size). An algorithm that does just the opposite (favors a smaller step size over current) can be substituted.

The multiple is incremented until a master frequency over 66.5MHz (for DS1085) is found.

master frequency = multiple * desired frequency



Calculate M and N

Valid values for N are 2 - 1025, or 1 (bypassed).
Valid values for M are 1, 2, 4, or 8.

multiple = M * N

Note: M will never be larger than 8 since the desired frequency has already been validated.

Calculate Offset

Again, this example refers to a DS1085-10 (5V, 10kHz step size) device.

Example: 51.2 - 61.43MHz for OS-8 range (DS1085-10). Refer to the "Frequency vs. Offset" table in the datasheet.

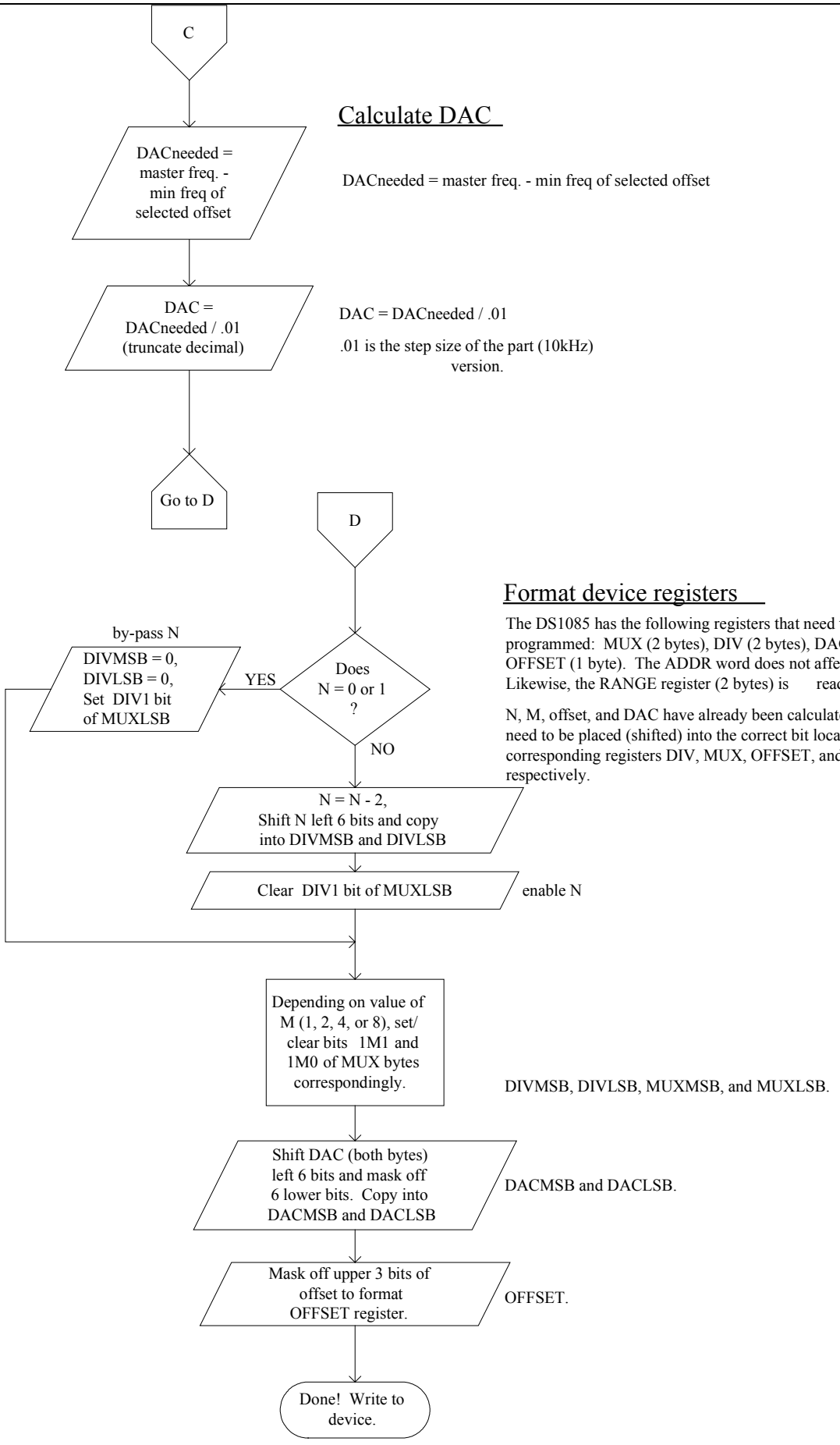
offset = RANGEvalue - 8 + offsetindex

The -8 is a function of device version (DS1085-10).

RANGEvalue is the converted (shifted) value read from the RANGE register.

Optional: The purpose of this block is to choose the offset that puts the master frequency closest to the center of the range. While not necessary, this does give the DAC register more room to tune the frequency above or below the needed master frequency. Otherwise, if stuck at the upper boundary for example, DAC will be of little use.

Also, if already at the max offset (OS+6) this block will be skipped.



Calculate DAC

DACneeded = master freq. - min freq of selected offset

DAC = DACneeded / .01

.01 is the step size of the part (10kHz) version.

Format device registers

The DS1085 has the following registers that need to be programmed: MUX (2 bytes), DIV (2 bytes), DAC (2 bytes), and OFFSET (1 byte). The ADDR word does not affect the frequency. Likewise, the RANGE register (2 bytes) is read-only .

N, M, offset, and DAC have already been calculated, but now they need to be placed (shifted) into the correct bit locations of their corresponding registers DIV, MUX, OFFSET, and DAC, respectively.