PROGRAMMABLE FEATURES OVERVIEW AND SOFTWARE GUIDANCE FOR THE 5V9885

APPLICATION NOTE AN-240

TABLE OF CONTENTS

INTRODUCTION

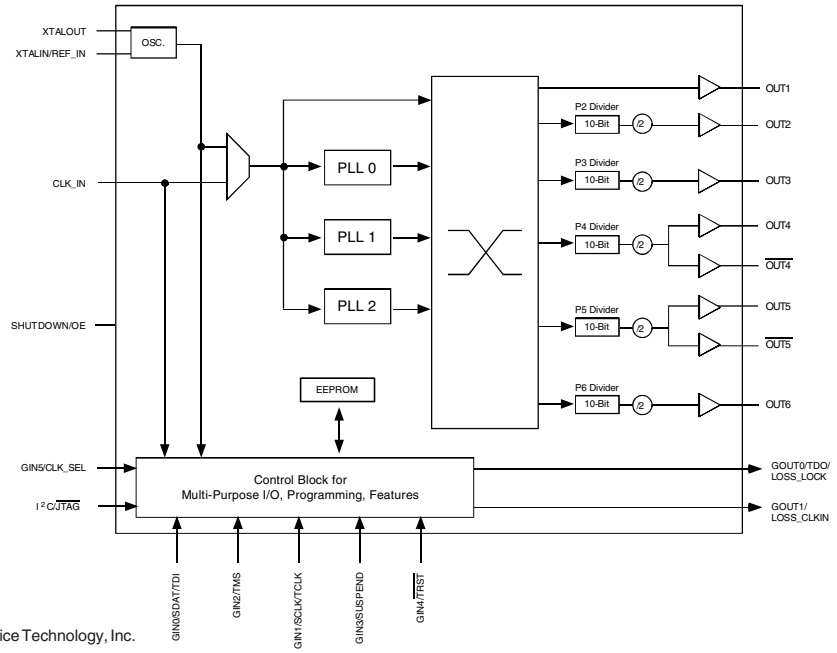
The Programmable Clock Generator 5V9885 was designed to provide the system engineer with many programmable features that can be used to support various applications. This application note will provide some guidelines for using the programmable features to achieve the best performance and will also describe the programmable software GUI (graphical user interface).

DEVICE OVERVIEW

The IDT5V9885 is a high performance programmable clock generator. It has three internal PLLs, each individually programmable, allowing for three unique non-integer related frequencies. It also has two inputs to support redundant clock networks. A crystal or a driven reference clock can be used with the inputs. The device also features an automatic or manual switchover mode that allows the system to switch between the two reference inputs; this enables redundant clock networks to be implemented easily and without complicated external logic. The device's output frequency ranges from 4.9KHz to 500MHz and the input frequency ranges from1MHz to 400MHz. The IDT5V9885 can be programmed through the I^2C or JTAG interfaces. Modifications to the configuration can be made dynamically during system operation to account for the operating environment or it can be made during the manufacturing of the device. The 5V9885 can be integrated into any standard JTAG programming flow; the device is fully JTAG compliant. An internal EEPROM allows the user to save and restore the configuration of the device without having to reprogram it every time on power-up.

Each of the three PLLs has an 8-bit pre-scaler and a 12-bit feedback divider. This allows the user to generate three unique non-integer related frequencies. The PLL loop bandwidth is programmable to allow the user to tailor the PLL response to the application. For instance, the user can tune the PLL parameters to minimize jitter generation or to maximize jitter attenuation. Spread spectrum generation is allowed on two of the PLLs. There are also 10-bit post dividers on five of the six outputs. Two of the six outputs are configurable to be LVTTL, LVPECL, or LVDS. The other four outputs are LVTTL. The outputs are connected to the PLLs via the switch matrix. The switch matrix allows the user to route the output clocks to any output pin. This feature can be used to simplify and optimize the board layout. In addition, each output's slew rate enable/disable function can be programmed. The Functional Block Diagram is shown below. For more information on this device, refer to the IDT5V9885 datasheet.

FUNCTIONAL BLOCK DIAGRAM



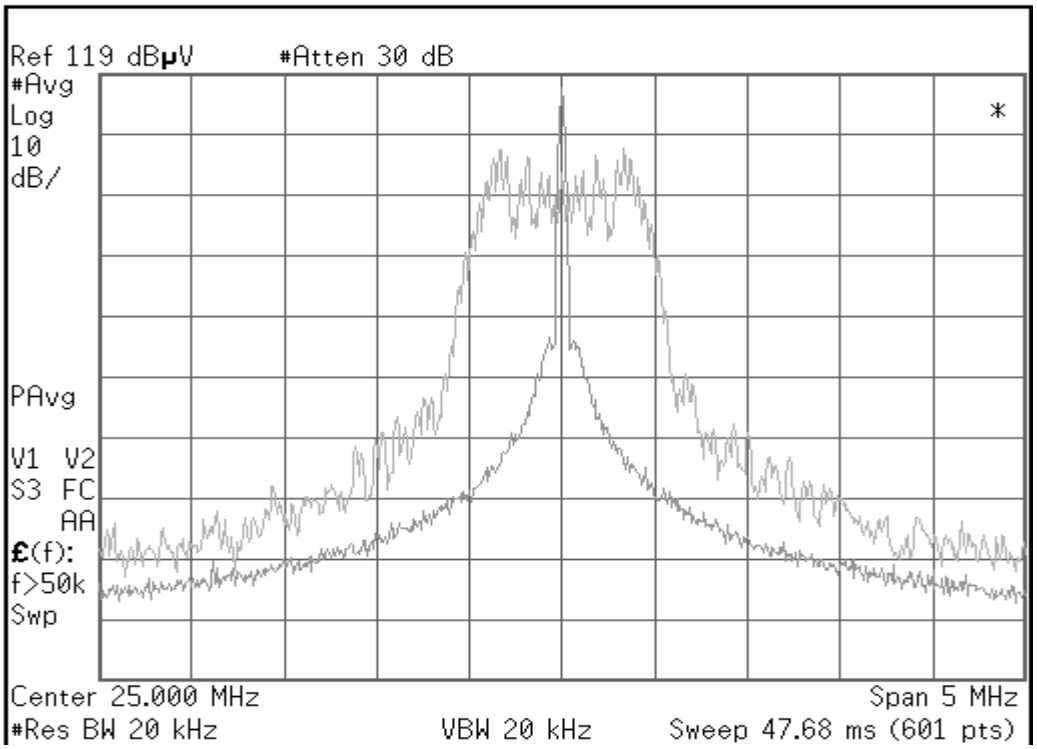
APRIL 2005

PROGRAMMABLE FEATURE GUIDE LINE

There are many programmable features in the IDT5V9885. In this section, we will provide guidelines for configuring the more advanced features: spread spectrum, loop-bandwidth, pre-scaler D, feedback-divider M, and post-divider P, slew rate and input crystal load capacitance.

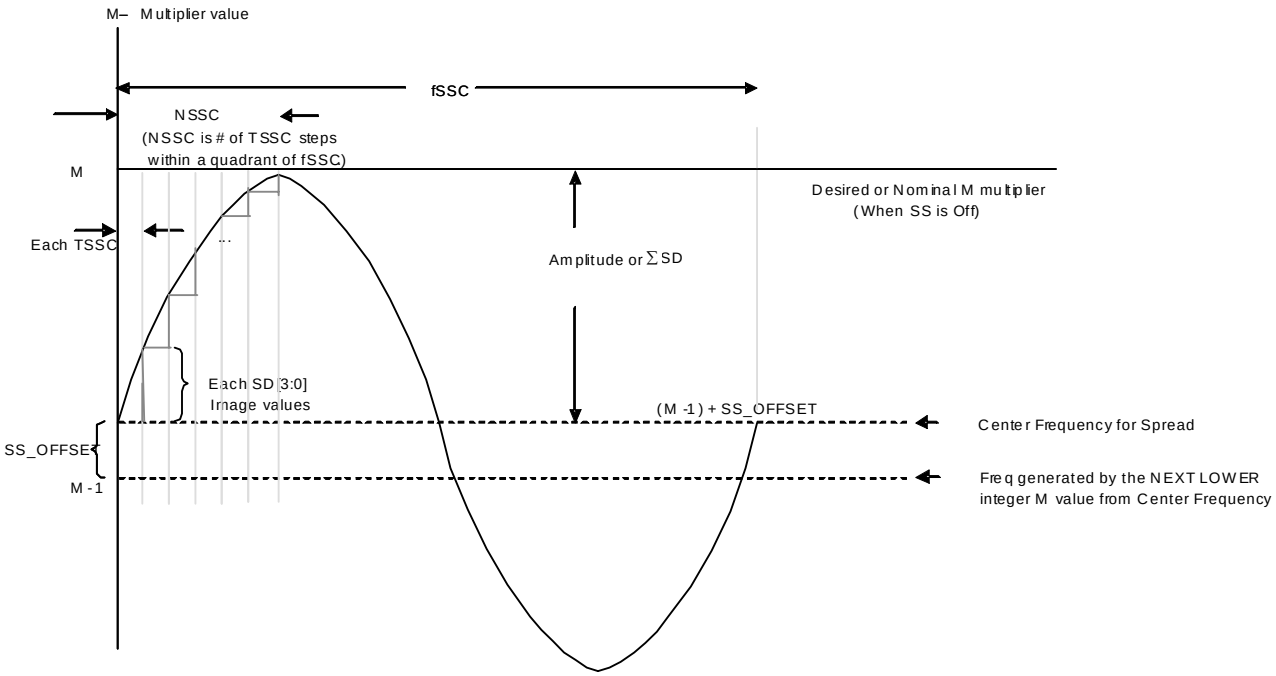
SPREAD SPECTRUM

The spread spectrum helps to reducing the radiated emission by spreading the emission over a wider frequency band. It reduces the fundamental clock frequency EMI (Electro Magnetic Interference) as well as the higher frequency harmonic components. PLL0 and PLL1 supports spread spectrum programmable capability e.g. center spread and down spread or up spread. The Spread Spectrum Output Clock diagram shows an example of spread spectrum spread.



Spread Spectrum Output Clock

Spread spectrum profile, frequency, and spread can be programmed to meet the EMI reduction requirement. The programmable spread spectrum generation parameters are TSSC[3:0], NSSC[3:0], SS_OFFSET[5:0], SD[3:0], DITH, and X2 bits. The spread spectrum generation on PLL0 & PLL1 can be enabled/disabled using the TSSC[3:0] bits. To enable spread spectrum, set TSSC > '0' and to disable, TSSC = '0'. The parameters are shown in the following diagram.



Spread Spectrum Parameters

The following equation determines how the divide value is set for spread spectrum.

M = 2 * N[11:0] + A[3:0] + 1 + SS_OFFSET[5:0] * 1/64 (Eq.1)

where SS_OFFSET are the bits used to program the fractional offset with respect to the nominal M integer value. For center spread, the SS_OFFSET should be set to '0' so the spread spectrum waveform is about the nominal M value. The SS_OFFSET[5:0] has integer values ranging from 1 to 63. The spread system works on 1/64^{THS} of the M value. The limits of the modulator are ±63 (i.e. SSCOffset + Sum(SD) < 64).

For example, reference frequency is 10 MHz, pre-scaler D divider is 1, F_{PFD} is 10MHz, M is 80, F_{VCO} is 800MHz and the output divider is 6 and output nominal frequency is 133MHz. Select N = 35 ,A= 9 and SS_OFFET = 0 for center spread.

M = 2N * A+1 = 2*35 + 9 + 1 = 80

The "A" value should be use for the spread spectrum. Spread Spectrum circuits utilize "A" value to achieve the modulation.

The percentage of spread spectrum can be calculated with the following formula:

$$\pm Spread\% = \frac{\Sigma\Delta * 100}{64 * (2*N[11:0] + A[3:0] + 1)} \quad (Eq.2)$$

$$\pm Max\ Spread\% = 1 / M \text{ or } 2 / M\ (X2=1) \quad (Eq.3)$$

$$\Sigma\Delta = SD0 + SD1 + SD2 + \ldots + SD11 \quad (Eq.4)$$

The number of samples used depends on the N_{SSC} value.

SD[3:0] are the bits used to shape the profile of the spread spectrum waveform. These are delta-encoded samples of the waveform. There are twelve samples (sets) of the SD bits for each PLL. The NSSC bits determine how many of these samples are used for the waveform. The sum of these delta-encoded samples (sigma-delta-encoded-samples) determine the amount of spread. The maximum spread is inversely proportional to the nominal M integer value.

The X2 bit will double the total value of the sigma-delta-encoded-samples which will increase the amplitude of the spread spectrum waveform by a factor of two. When X2 is '0', the amplitude remains nominal but if set to '1', the amplitude is increased by x2.

For example, the center percentage spread is ±0.5%, M is 80 then the amplitude can be calculated from equation 2.

$$0.5 = \frac{\Sigma\Delta * 100}{64 * 80}$$

$$\Sigma\Delta = (0.5/100)*64*80 = 26$$

The shape of the spread spectrum can be determined with the following formula.

$$T_{SSC} = TSSC[3:0] + 2 \quad (Eq.5)$$

$$N_{SSC} = NSSC[3:0] * 2 \quad (Eq.6)$$

This formula assumes that TSSC[3:0] are the bits used to determine the number of phase/frequency detector cycles per spread spectrum cycle (ssc) steps. The modulation frequency can be calculated with the TSSC bits along with the NSSC bits. Valid TSSC integer values for the modulation frequency range from 1 to 14. T_{SSC} should be greater than 5. It will give a better averaging of the modulation waveform sample with higher T_{SSC} .

The modulation frequency typically can be from 25KHz to 100KHz depending on the jitter performance and parameter limitation. NSSC[3:0] are the bits used to determine the number of delta-encoded samples used for the spread spectrum waveform. The modulation frequency is also calculated based off the NSSC bits along with the TSSC bits. Valid NSSC integer values range from 1 to 6.

$$SD[3:0]_M = S_{N+1}(unencoded) - S_N(unencoded) \quad (Eq.7)$$

where S_N is the unencoded sample out of a possible 12 and SDM is the delta-encoded sample out of a possible 12.

$$Amplitude = \frac{(2*N[11:0] + A[3:0] + 1) * Spread\% / 100}{2} \quad (Eq.8)$$

if 1 < Amp < 2, then set X2 bit to '1'.

For example, max number NSSC sample can be up to 12. To simplify, select the N_{SSC} = 8, then the register values for NSSC = 4 or 0100. SD Image sample can be chosen as below.

SD1[3:0][0] = 5 or 0101
SD1[3:0][1] = 4 or 0100
SD1[3:0][2] = 2 or 0010
SD1[3:0][3] = 1 or 0001
SD1[3:0][4] = 2 or 0000
SD1[3:0][5] = 3 or 0011
SD1[3:0][6] = 5 or 0101
SD1[3:0][7] = 4 or 0010

Sum of SD $\Sigma\Delta$ = 26

The modulation frequency can be calculated with the following formula:

$F_{PFD} = F_{IN} / D$ (Eq.9)
$F_{VCO} = F_{PFD} * M$ (Eq.10)

$F_{SSC} = F_{PFD} / (4 * N_{SSC} * T_{SSC})$ (Eq.9)

For example, determine TSSC value from the frequency F_{SSC} at 30KHz range

TSSC = F_{PFD} / (4 * F_{SSC} * NSSC) = 10000/ (4 * 30) / 8 = 10.41

Choose the nearest TSSC = 10 or 1010; therefore the actual F_{SSC} can be calculated

F_{SSC} = F_{PFD} / (4 * NSSC * TSSC) = 10000 / (4 * 8 * 10) = 31.25KHz

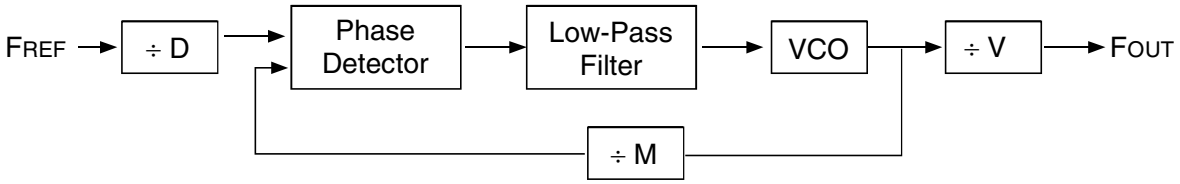
The DITH bit is for dithering the sigma-delta-encoded samples. This will enhance the profile of the spread spectrum waveform. Set the bit to '1' to enable dithering. Note that the 5v9885 should not be programmed with TSSC > '0', SS_OFFSET = '0', and SD = '0' in order to prevent an unstable state in the modulator.

In general, the loop bandwidth frequency Fc should be close to F_{PFD}/10 and Fc should be at least six to ten times of F_{SSC} for the PLL to track the spread spectrum modulation.

PRE-SCALER, FEEDBACK DIVIDER AND POST DIVIDER VALUES

Each PLL has an 8-bit pre-scaler and a 12-bit feedback divider which allow the user to generate three unique non-integer related frequencies. Output banks OUT2-OUT6 each has a 10-bit post divider providing the ability to fanout divided down PLL outputs. For instance:

$$Fout = \frac{Fref \cdot \frac{M}{D}}{V}$$



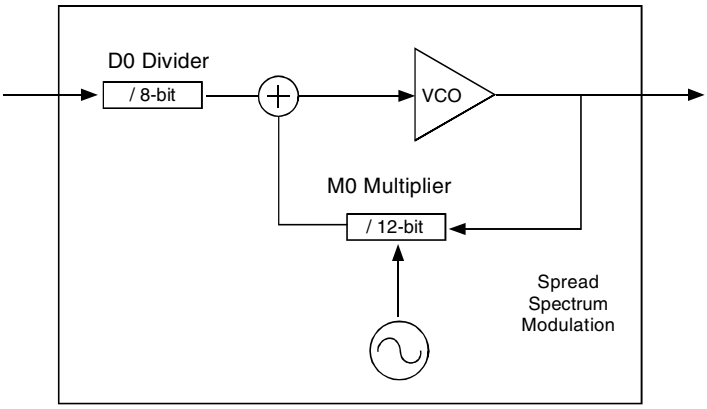
5V9885 Block Diagram

In the 5V9885 Block Diagram, the pre-scaler D divides down the reference clock with integer values ranging from 1 to 255. To achieve the best jitter performance, the divided down clock must be higher than 1MHz (it is best to use the smallest D divider value possible.) The VCO frequency should be maximized; the VCO's jitter decreases as its operating frequency increases. There are many output options from one PLL, as the high VCO frequency can be divided down more output dividers. The feedback divider M has integer values ranging from 1 to 4095. In a closed loop PLL, the input frequency is multiplied by the value M to get the output frequency. The VCO has a frequency range of 10MHz to 1000MHz and to maintain low jitter and a stable operation, it's best to use the largest M value possible. The post-divider will divide down the output banks' frequency with integer values ranging from 1 to 1023. When the post-divider is disabled, no clock will appear at the outputs. The output frequency range is from 4.9KHz up to 500MHz. LVTTL outputs can only go up to 200MHz. LVDS/LVPECL outputs can go up to 500MHz.

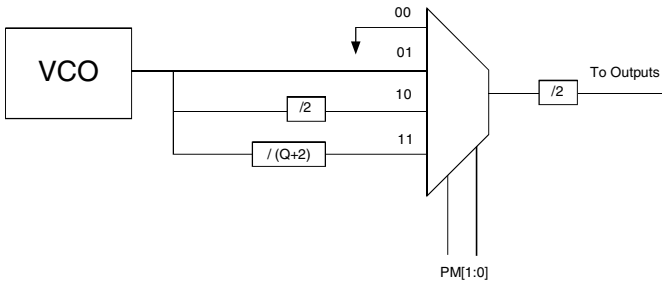
Due to the complexity and high programmability of this device, M and V are not constants, but rather functions of register settings. This information is detailed on pages 6 and 7 of the datasheet, including the fractional divider. Note that M and V are abstractions. They are only used for simplicity, and there are no register mappings to M and V. There are only registers for N, A, and Q. Because PLL0 and PLL1 have spread spectrum generation capabilities, their M values are governed by a separate function than that of PLL2. Also note that for PLL0 and PLL1, M cannot take on all integer values between 2 and 8190.

To program the device, determine D, M, and V such that the desired output frequency is generated. Then use the equations in the table to determine N, A, and P. If there are no solutions for N, A, or Q, then try again with different M or V values. Our development software will make this process effortless and easy. The N, A, and Q values are needed to be written into the registers.

	M	**V**
PLL0	M = 2.N, if A = 0 M = 2.N + A + 1, if A>0	V = 2*P
PLL1	M = 2.N, if A = 0 M = 2.N + A + 1, if A>0	V = 2*P
PLL2	M = N	V = 2*P



Pre-Scaler and Feedback Divider

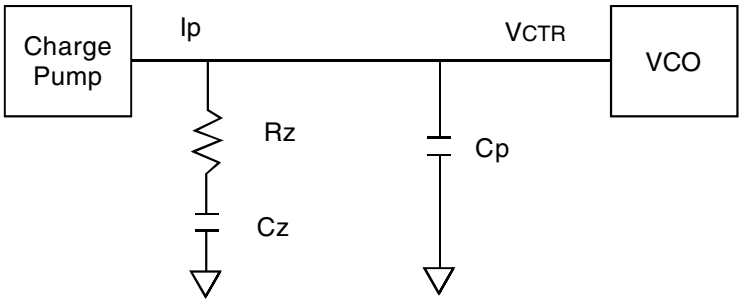


Post Divider

LOOP FILTER

The loop filter configuration is shown in the following diagram. This loop filter accumulates the average charge from the charge pump which then converts the charge into the voltage to set the frequency of the voltage control oscillator (VCO). All three PLLs support fully programmable loop filter settings. Loop filter parameters are:

Ip: Charge pump current: 5 to 640uA
Cp, pole capacitor: 1.3 to 12.55pF
Cz, zero capacitor: 6 to 414pF
Rz, zero resistor: 0.35 to 15.3kO



Loop Filter

The loop filter for each PLL can be programmed to optimize either jitter attenuation or jitter generation. The low-pass frequency response of the PLL is the mechanism that dictates the jitter transfer characteristics. The loop bandwidth can be extracted from the jitter transfer. A narrow loop bandwidth is good for jitter attenuation, while a wide loop bandwidth is best for low jitter generation. The specific loop filter components that can be programmed are the resistor via the RZ[3:0] bits, pole capacitor via the CZ[3:0] bits, zero capacitor via the CP[3:0] bits, and the charge pump current via the IP[2:0] bits.

Charge Pump and Loop Filter Configuration

Resistor (Rz) = 0.3KΩ + RZ[3:0] * 1KΩ (Eq.10)

Zero capacitor (Cz) = 6pF + CZ[3:0] * 27.2pF (Eq.11)

Pole capacitor (Cp) = 1.3pF + CP[3:0] * 0.75pF (Eq.12)

Charge pump current (Ip) = $5 * 2^{IP[2:0]}$ μA (Eq13)

Parameter	Step	Min	Max	Units
RZ	1	0.3	15.3	KΩ
CZ	27.2	6	414	pF
CP	0.75	1.3	12.55	pF
IP	2^n	5	640	μA

The PLL loop bandwidth (Fc) can be estimated with given the programmable loop filter parameters using the simplify method as follows.

PLL Loop Bandwidth:

Charge pump gain (KΦ) = Ip / 2π (Eq.14)
VCO gain (K_{VCO}) = 950MHz/V * 2π (Eq.15)
M = Total multiplier value (See the PRE-SCALER, FEEDBACK-DIVIDER, and POST DIVIDER VALUES section for more detail)

$$\omega c = \frac{Rz * K\Phi * K_{VCO} * Cz}{M * (Cz + Cp)} \quad \text{(Eq.16)}$$

$$Fc = \omega c / 2\pi \quad \text{(Eq.17)}$$

Note, the minimum limit of the loop bandwidth is restricted to F_{PFD} / Fc >10. F_{PFD} is the phase/frequency detector frequency.

To determine if the loop is stable, the phase margin (Φm) would need to be calculated as follows.

Phase Margin:

$$\omega z = 1 / (Rz * Cz) \quad \text{(Eq.18)}$$

$$\omega p = \frac{Cz + Cp}{Rz * Cz * Cp} \quad \text{(Eq.19)}$$

$$\Phi m = (360 / 2\pi) * [\tan^{-1}(\omega c/\omega z) - \tan^{-1}(\omega c/ \omega p)] \quad \text{(Eq.20)}$$

To ensure stability in the loop, the phase margin should be > 60°, but too high will result in the lock time being excessively long.
An example of where we are given the total M value and loop filter parameter settings is as follows.

For example, RZ[3:0] = 0100, CZ[3:0] = 0111, CP[3:0] = 0101, IP[2:0] = 100, M = 22

Calculating the loop filter values
Rz = 4.3KΩ, Cz = 196pF, Cp = 5.05pF, Ip = 40μA

Solving for the PLL loop bandwidth
KΦ * K_{VCO} = Ip * 950MHz/V = 38000A/Vs

$$\omega c = \frac{4.3K\Omega * 38000A/Vs * 196pF}{22 * (196pF + 5.05pF)} = 5.79x10^6\ s^{-1}$$

$Fc = 5.79x10^6\ s^{-1} / 2\pi = 0.922MHz$

$\omega z = 1 / (4.3K\Omega * 196pF) = 1.18x10^6\ s^{-1}$

$$\omega p = \frac{196pF + 5.05pF}{4.3K\Omega * 196pF * 5.05pF} = 4.72x10^7\ s^{-1}$$

$\Phi m = (360 / 2\pi) * [\tan^{-1}(5.79x10^6\ s^{-1}\ 1 / 1.18x10^6\ s^{-1}) - \tan^{-1}(5.79x10^6\ s^{-1} / 4.72x10^7\ s^{-1})] = 71°$

In the above example, the loop showed to be fairly stable using the specified loop filter settings, however, this may not always be the case. Certain loop filter parameters would need to be compromised to not only meet a required loop bandwidth but to also maintain loop stability.

As we select parameters for another example, Fc = 150kHz is the desired loop bandwidth. The total M value is 850. A rule of thumb that will help to aid the way, the $\omega p / \omega c$ ratio should be about 4. Given Fc and M, we need to solve for an optimal loop filter setting that will meet both the PLL loop bandwidth and maintain loop stability.

The charge pump gain should be relatively small as possible to achieve a low loop bandwidth, therefore, Ip = 40μA is a good value to start with.

$K\Phi * K_{VCO} = 950MHz/V * 40\mu A = 38000A/Vs$

Loop Bandwidths

$\omega c = 2\pi * Fc = 9.42x10^5\ s^{-1}$

$\omega uz = \omega p / \omega c = 4$ (Eq.21)

$\omega c^2 = \omega p * \omega z$ (Eq.22)

$$\omega p = \frac{Cz + Cp}{Rz * Cz * Cp} = \omega z\ (1 + Cz / Cp)$$

Solving for Cz, Cp, and Rz

Knowing $\omega c = \frac{Rz * K\Phi * K_{VCO} * Cz}{M * (Cz + Cp)}$ and substituting in the equations from above,

Cz >>> Cp, therefore, we can easily derive Cp to be

$$Cp = \frac{K\Phi * K_{VCO}}{M * \omega c^2 * \omega uz} = 12.60pF$$

Similarly for Cz and Rz

$$Cz = \frac{K\Phi * K_{VCO} * (\omega uz^2 - 1)}{M * \omega c^2 * \omega uz} = Cp * (\omega uz^2 - 1) = 189pF$$

$$Rz = \frac{M * \omega c * \omega uz^2}{K\Phi * K_{VCO} * (\omega uz^2 - 1)} = 22.48K\Omega$$

Based on the loop filter parameter equations from above, since there are no possible values of 12.60pF for Cp, 189pF for Cz, and 22.48KΩ for Rz, the next possible values are 12.55pF (CP[3:0]=1111), 196.4pF (CZ[3:0]=0111), and 15.3KΩ (RZ[3:0]=1111), respectively. This loop filter setting will yield a loop bandwidth of about 102KHz. Last thing to check before the actual settings are final is the phase margin for loop stability.

$\Phi m = (360 / 2\pi) * [\tan^{-1}(6.41x10^5\ s^{-1} / 3.33x10^5\ s^{-1}) - \tan^{-1}(6.41x10^5\ s^{-1} / 5.54x10^6\ s^{-1})] = 56°$

Although slightly below 60°, the phase margin is acceptable and the loop should be stable.

The optimum loop filter parameter values are incorporated in IDT Programmable Clock software to insure loop stability.

SLEW RATE

There are four settings for the LVTTL output programmable slew rate; 0.7V/ns, 1.25V/ns, 2V/ns, and 2.75V/ns. The LVDS and LVPECL output slew rates are not programmable. The default slew rate for differential outputs is 2.75V/ns. In general, 2V/ns or 2.75V/ns should be selected for the output frequency at higher than 100MHz. The slow edge rate will cause more jitter at the output because the slower edge has more variant at the thred-hold level of the signal, especially in a noisy environment.

INPUT CRYSTAL LOAD CAPACITANCE

A total crystal capacitance load is programmable. A quartz crystal oscillator fundamental mode should be used. Crystal frequency should be specified for parallel resonance with 50Ω maximum equivalent series resonance. The internal load capacitors are true parallel-plate capacitors for ultra-linear performance, so an external non-linear crystal load is not necessary. The value of the internal load capacitors are determined by XTALCAP[7:0] bits at address 0x07. The load capacitance can be set with a resolution of 0.125pF for a total crystal load range of 3.5pF to 35.4pF. The internal load capacitance can be calculated with the following equation.

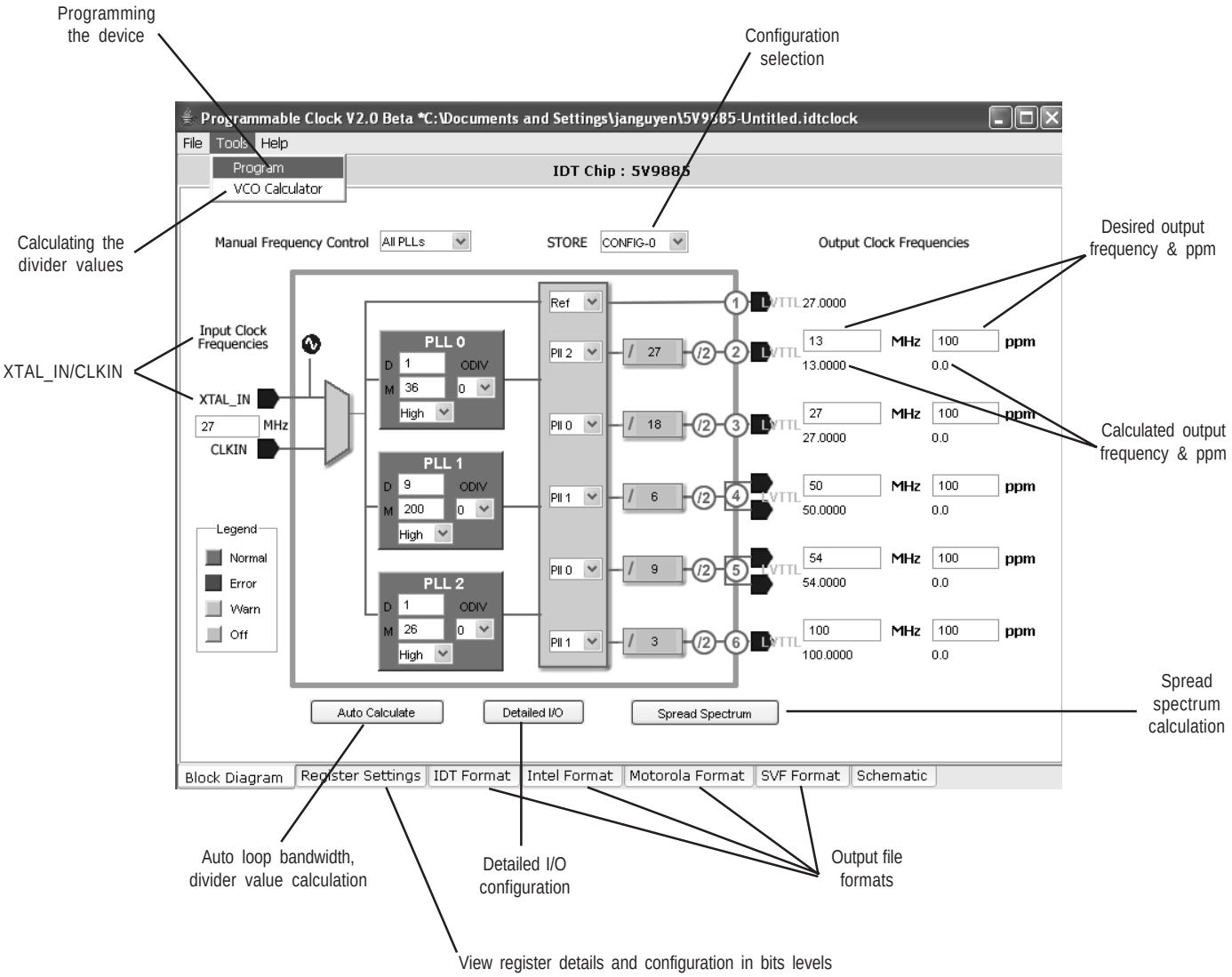
XTAL load cap = 3.5pF + XTALCAP[7:0] * 0.125pF (Eq.21)

When using an external reference clock instead of a crystal on the XTAL/REF_IN pin, the input load capacitors may be completely bypassed. The XTALOUT pin must be left floating, XTLCAP must be programmed to the default value of '0', and crystal drive strength bit, XDRV address 0x06, must be set to the default value of '11'.

PROGRAMMABLE CLOCK SOFTWARE OVERVIEW

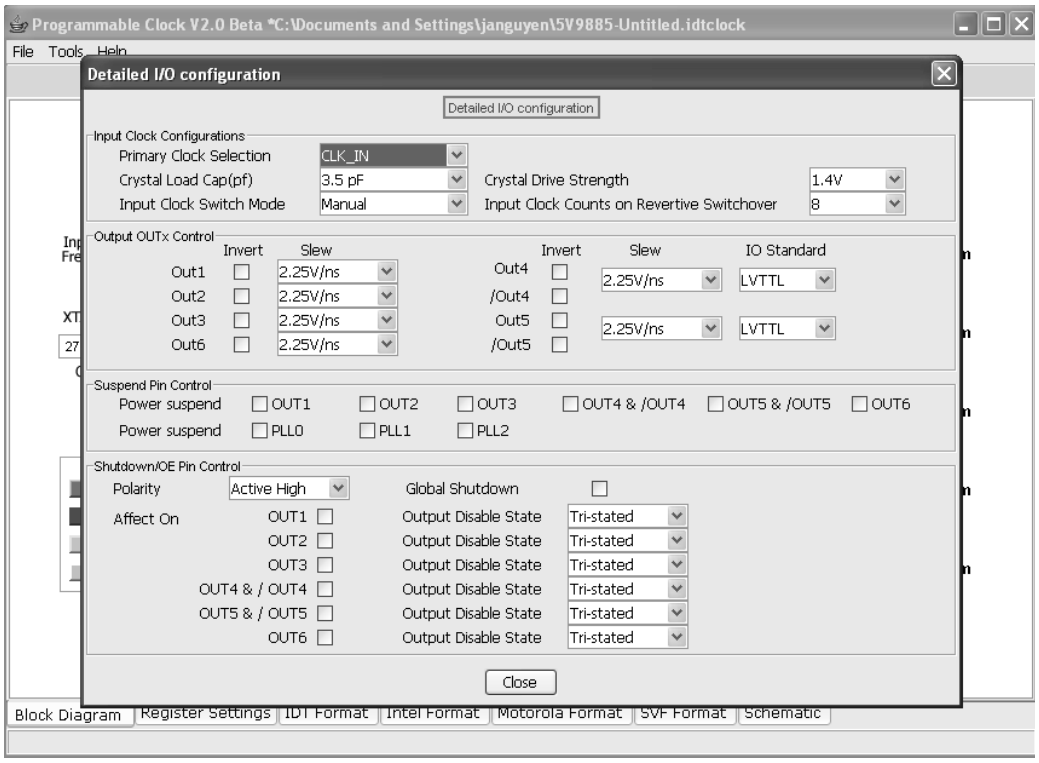
IDT provides software to configure the device and its features easily. Combined with the Timing Device Programmer kit and/or evaluation kit, the software allows device programming and testing in a lab environment. The Programmable Clock software is included in the Timing Device Programmer kit, and can also be downloaded from IDT website.

The Programmable Clock software provides seven screens to view all the features and bits: Block Diagram, Register Settings, IDT Format, Intel Format, Motorola Format SVF Format and Schematic. These individual screens are accessed via the tabs on the bottom of the screen. The Block Diagram and Register Settings tabs alternate between configurations. The IDT Format, the Intel Format, Motorola Format and the SVF Format screens shows configuration bits in a user-defined format. The Schematics tab shows the high level block diagram of the device.

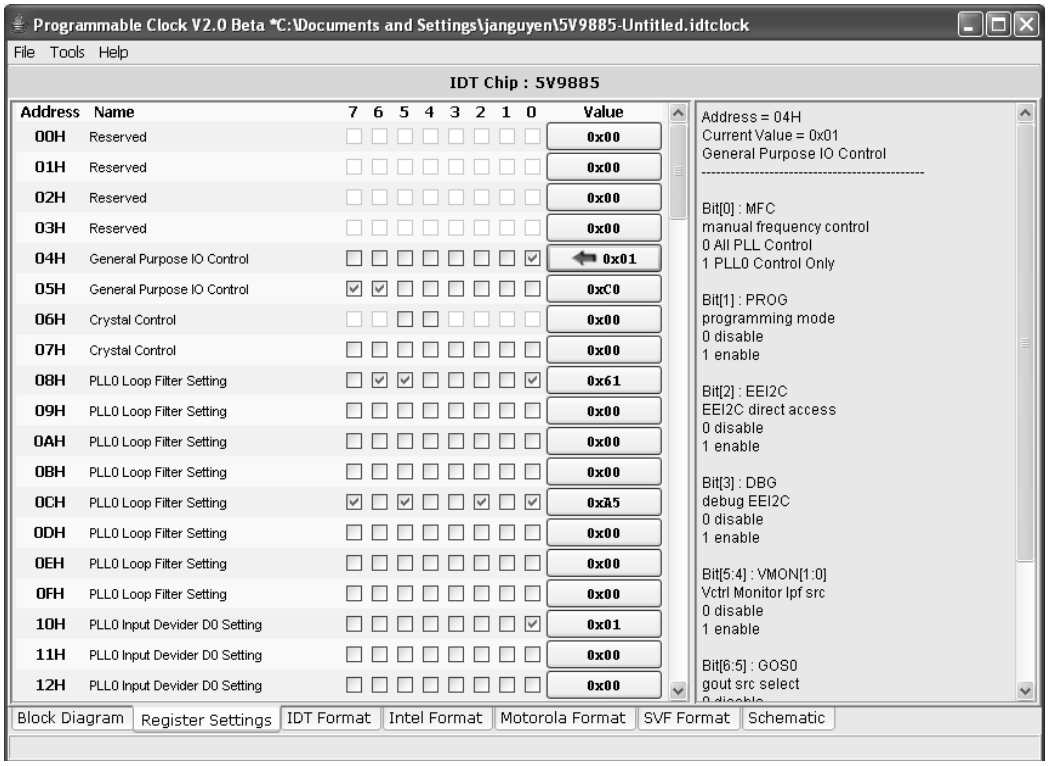


In the Block Diagram tab, XTAL_IN/CLKIN, Output Clock Frequency and ppm boxes are required values. Based on the selection of the input/output configuration, MFC mode, and spread spectrum boxes, the software will update the results and bits automatically.

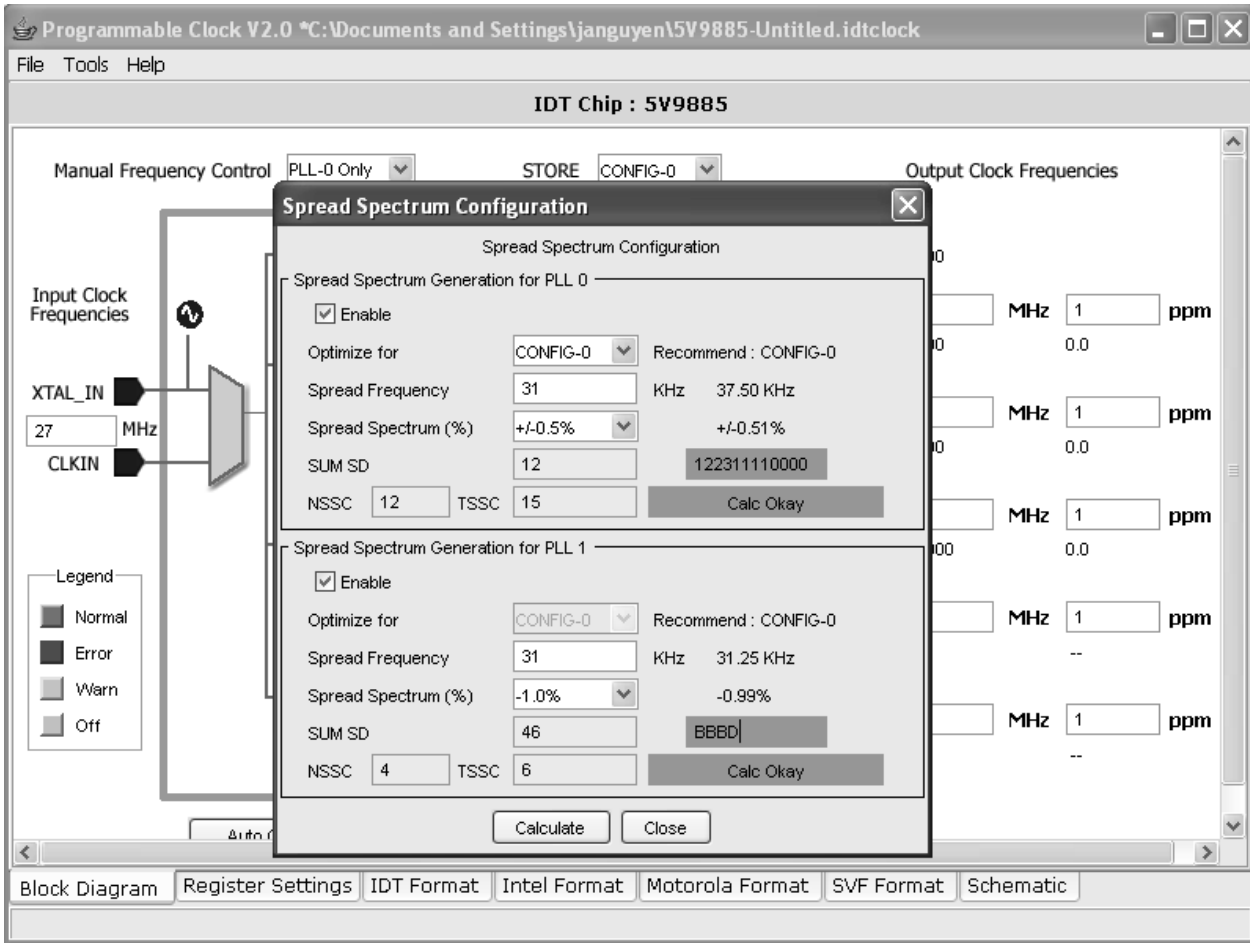
The pre-scaler values, feedback divider values, and output divider values and VCO frequency are automatically calculated when the 'Auto calculate' button is clicked. Individual VCO frequency and corresponding divider values can also be calculated with the pull-down VCO calculator menu. Any changes in the Block Diagram tab will change the configuration bit automatically, which will also be reflected in the Register Setting tab. The desired configuration in the Register Setting menu can also be changed with a given bit configuration. The detail input and output configuration can be updated in the Detail I/O Configuration button.



The Programmable Clock Software currently supports four file formats: IDT Format, Intel Hex Format, Motorola S-record Format, and SVF Format. The IDT Format is used to program the device when using Timing Device Programming kit. The file also includes more detail info about the configuration bits that can be useful for debugging purposes. Once the IDT Format file is saved, files can be exported into Intel Hex, Motorola S-record and/or SVF format. These formats are used to program the device through the I^2C and JTAG interfaces. Most automatic programming equipment can accept these file formats and program the device.



The desired spread frequency and spread percentage values should be entered for spread spectrum configuration calculation. When the 'Calculate' button is clicked, the software will generate SUM SD, Nssc, and Tssc. It will prompt the user to set the profile image in the box with all zeros. Each zero represents one SD value. Sets the image of the spread spectrum by replacing the zero value with a desired number such that sum of all SDs is equal to the calculated value in SUM SD box. Once the the 'Close' button is clicked, the image is saved.

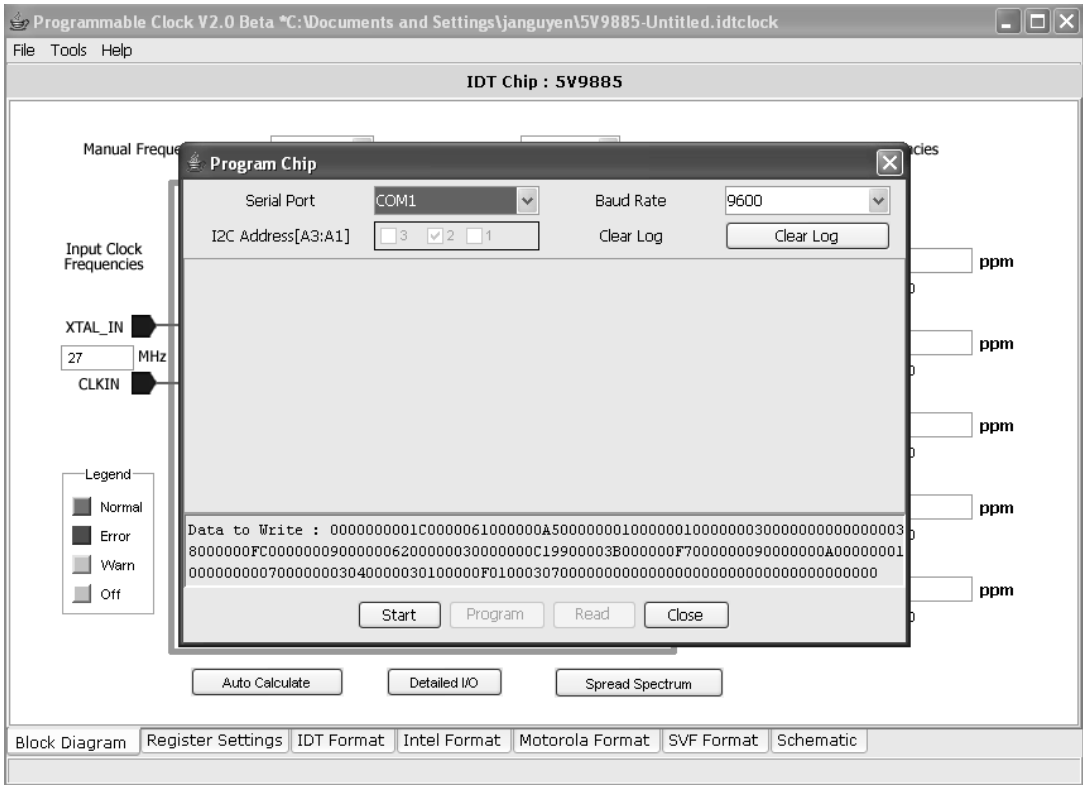


PROGRAMMING WITH TIMING DEVICE PROGRAMMING KIT

To program this device using the Timing Device Programmer, the Programmable Clock Software is needed and it must be set up correctly.

Setting COM Port

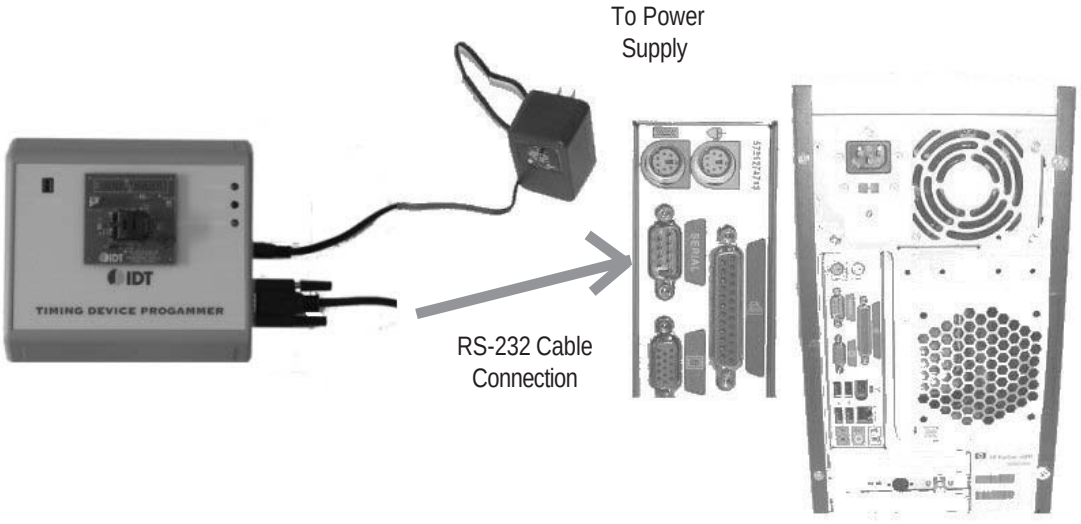
Once the features selections are done on the Block Diagram and Register Settings tabs, users will need to choose the appropriate COM port setting to program the device. To set the COM port, select Tools from the menu bar and then select Program. The 'Program Chip' menu will popup. Click on the Serial Port for appropriate COM port setting.



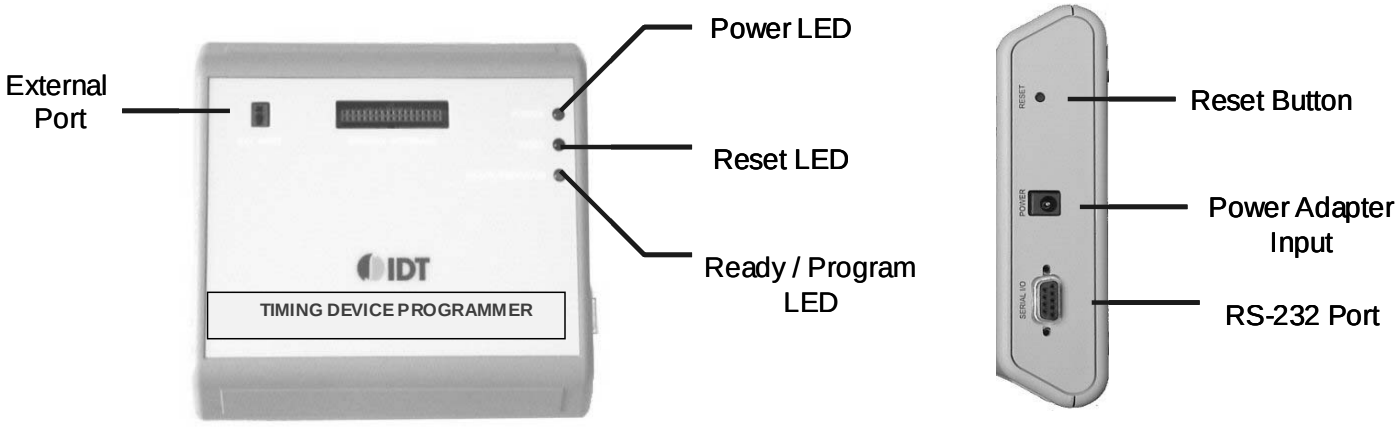
Setting Up Hardware Connections

All hardware needs to be connected correctly before programming starts. The module connector must be inserted into the Timing Device Programmer, and the correct device must be inserted into the socket. Note that the software checks for a device ID before programming; if the wrong device is inserted into the socket, programming will not begin.

Connect the RS-232 and Power Supply to the Timing Device Programmer

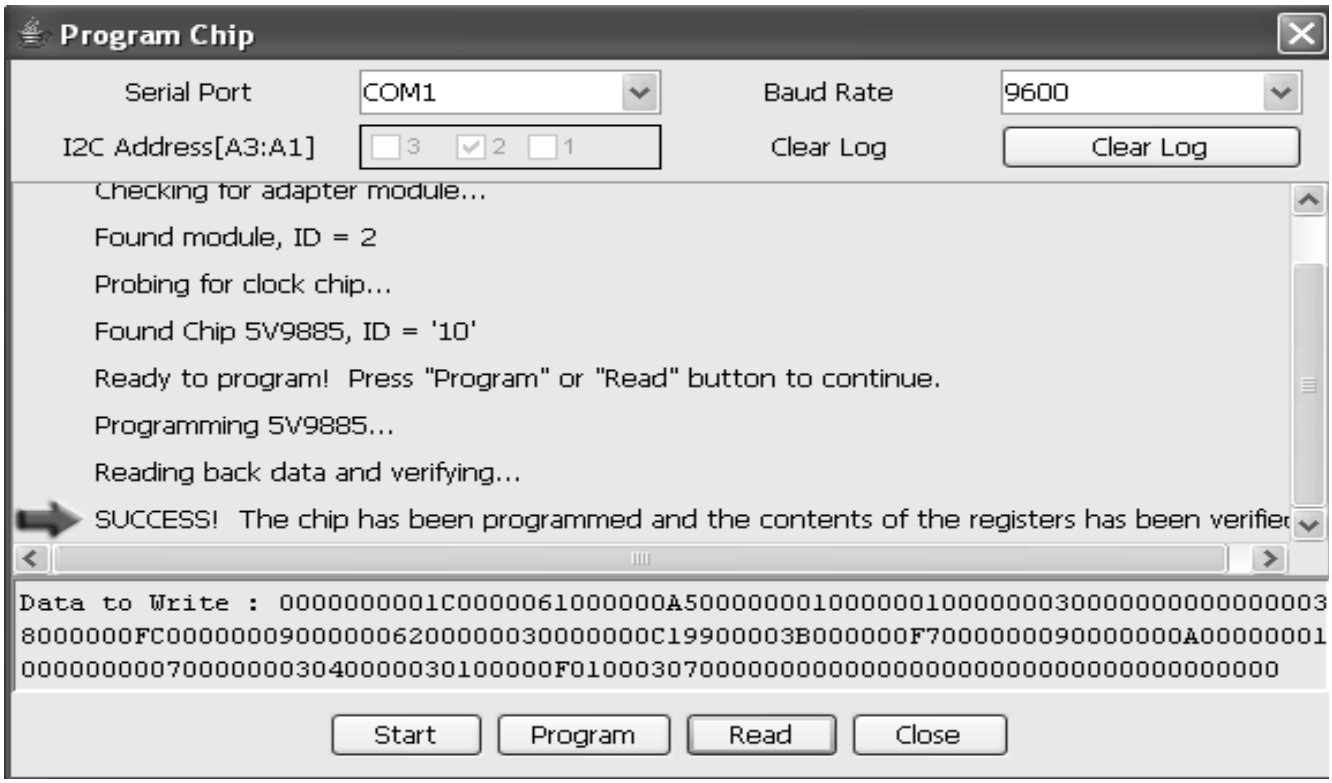


TIMING DEVICE PROGRAMMER OVERVIEW



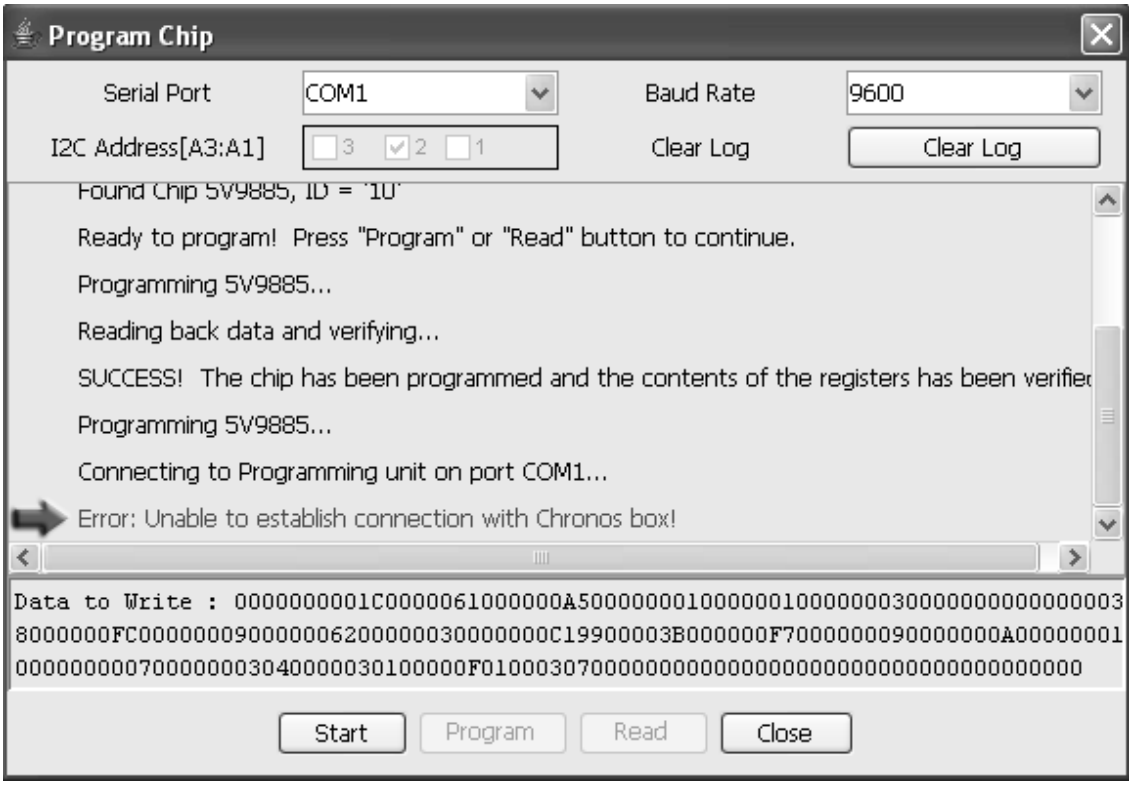
Start Programming

After the COM port setting selection and hardware are set up, the device is ready to be programmed. Select Tools from the menu bar, select Program Chip, and click on Start. The software will detect if the correct module is plugged-in or not, and then it will check if the device inside the socket matches the device ID from the configuration file. After successfully passing all these internal checks, the device is ready to be programmed. Click "Program" to program the device.



Error Messages

If the socket lid is not properly locked, the device not properly inserted (see pin 1 on the module PCB), or an incorrect clock device is used, then the software will prompt an error message as shown below.



PROGRAMMING SUMMARY (STEP-BY-STEP)

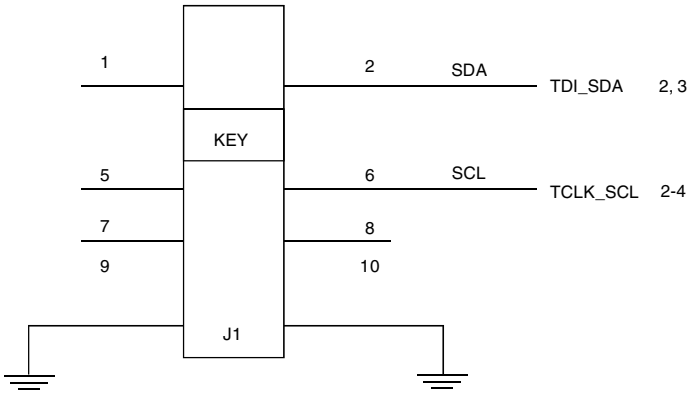
Below is the step-by-step procedure to program the device using Programmable Clock software and Timing Device Programmable kit.

1. Launch the Programmable Clock Software application
2. Select the device to be programmed or configured
3. Select all the feature settings through the Block Diagram tab and/or Register Settings tab
4. Save the configuration
5. Connect the Timing Device Programmer kit (see TDP Getting Started Guide)
6. Insert the Device Module into the Timing Device Programmer kit
7. Insert the correct device into the Device Module socket
8. Connect the power supply and the RS232 cable
9. From the software menu bar, select Tools -> Program, then Set Port to select the correct COM port setting
10. Select Tools -> Program Chip to program the device
    (The software will detect the connection, module, and device to make sure the device and the module matches the configuration file)
11. Click the Program button to program the device
12. To program through automatic programming equipment or standard JTAG programming software, the Intel Hex, Motorola S-record or SVF file format may be needed. To export to these files, go to File -> Export -> select Intel Hex or Motorola S-record or SVF.

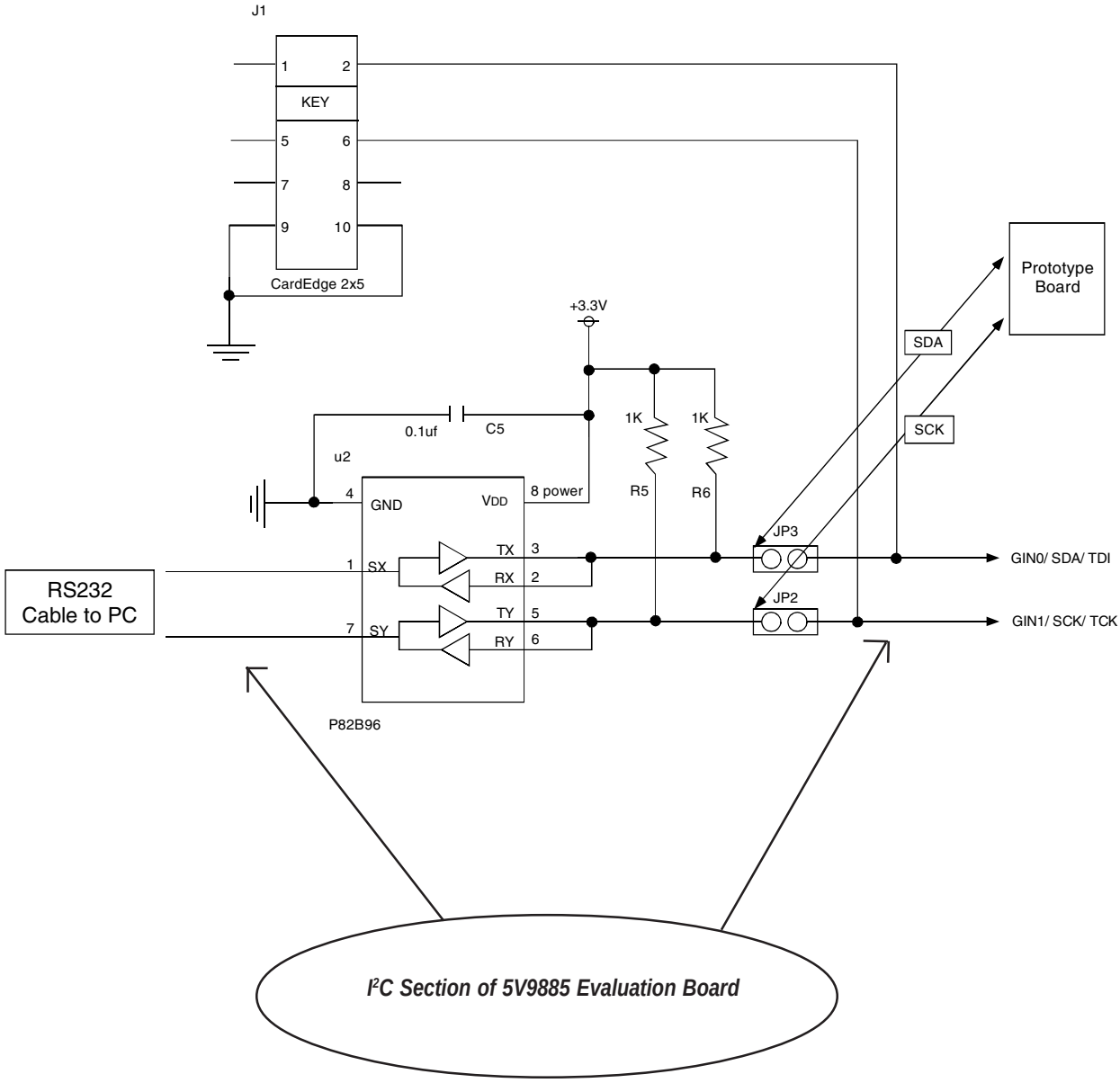
PROTOTYPE BOARD PROGRAMMING USING THE TIMING DEVICE PROGRAMMER OR 5V9885 EVALUATION BOARD

The Timing Device Programmer can be used to program a prototype board on the fly for testing purposes. First bring the I^2C pins of the device to a header or to the I^2C connector listed below. After the Device Programmer is set up according to the procedure in the "Setting Up Hardware Connection" section, connect the IDT I^2C cable from the I^2C external port (please see Timing Device Programmer Overview figure below) on the Timing Device Programmer to the prototype board. Since the Programmable Clock Software detects the module ID, it is important that the proper Socket Module (without the device inside) should be plugged into the programmer, even when the programming is through I^2C cable. (The I^2C cable that came with the Device Programmer can also be requested from IDT with the IDT part number below. The I^2C connector used for I^2C interface is made by Samtec, the part number of which is also shown below.) The device on the prototype board is now ready to be programmed by IDT's software.

Cable Part Number: 52-534-000
Connector Part Number: MEC1-105-02-S-D-LC (CONN,SM,2 x 5,Edge-Card Skt,1.0mm PT,STR)
Drawing: http://www.samtec.com/ftppub/cpdf/MEC1-1XX-XX-X-D-XX-XX-MKT.pdf



The 5V9885 Evaluation board can also program the prototype board. On the prototype board, bring the I^2C pins of the device to a header. After this, use two wires to connect the left side of jumpers JP2 and JP3 I^2C SDA/SCK pins on the evaluation board to the I^2C headers on the prototype board. The RS232 cable should be connected to the PC on the other side of the JP2/3, shown in detail in 5V9885 Evaluation Board User Guide. Once this setup is ready, the IDT software can directly control and program the device on the prototype board.



CORPORATE HEADQUARTERS
6024 Silver Creek Valley Road
San Jose, CA 95138

for SALES:
800-345-7015 or 408-284-8200
fax: 408-284-2775
www.idt.com

for Tech Support:
logichelp@idt.com