1.0 Features

- Just-in-time customization of clock frequencies via internal non-volatile 128-bit serial EEPROM
- I^{2}C™-bus serial interface
- Three on-chip PLLs with programmable reference and feedback dividers
- Four independently programmable muxes and post dividers
- Programmable power-down of all PLLs and output clock drivers
- Tristate outputs for board testing
- One PLL and two mux/post-divider combinations can be modified via SEL_CD input
- 5V to 3.3V operation
- Accepts 5MHz to 27MHz crystal resonators

2.0 Description

The FS6370 is a CMOS clock generator IC designed to minimize cost and component count in a variety of electronic systems. Three EEPROM-programmable phase-locked loops (PLLs) driving four programmable muxes and post dividers provide a high degree of flexibility.

An internal EEPROM permits just-in-time factory programming of devices for end user requirements.

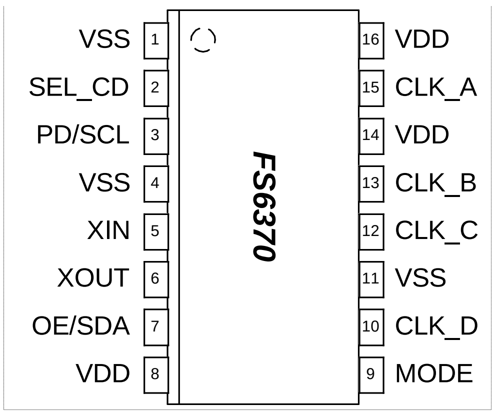


Figure 1: Pin Configuration

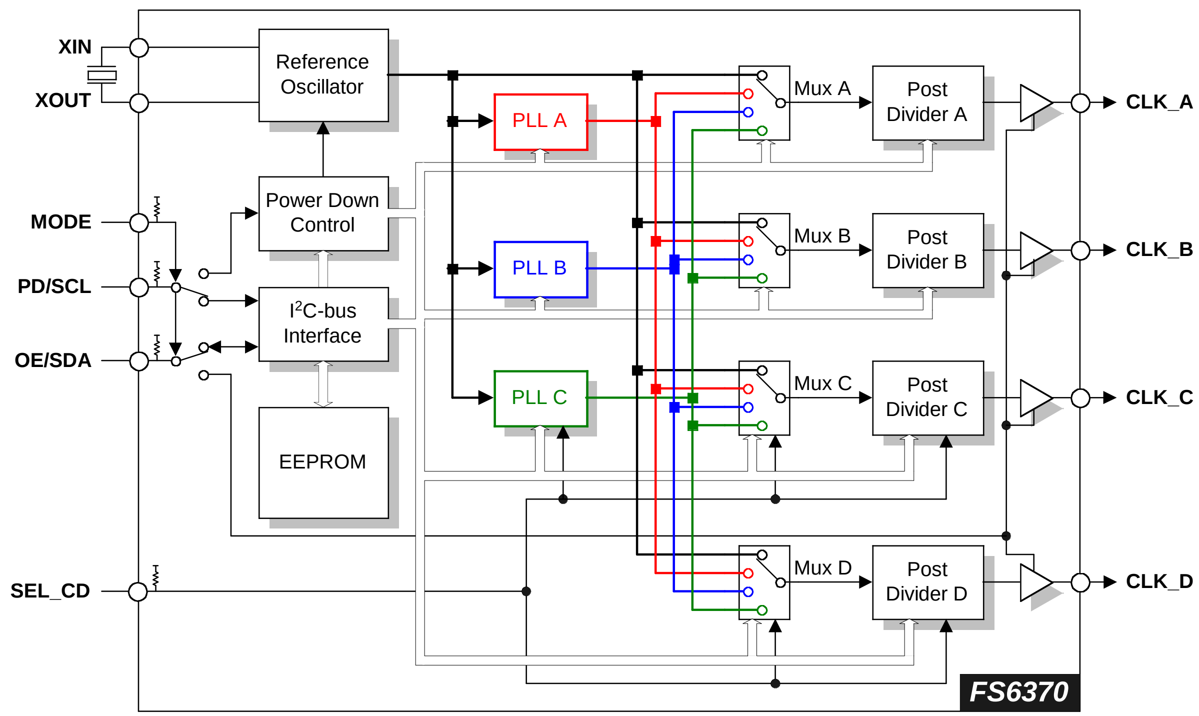


Figure 2: Block Diagram

Table 1: Pin Descriptions

Pin	Type	Name	Description
1	P	VSS	Ground
2	DI^{U}	SEL_CD	Selects one of two programmed PLL C, Mux C/D and post divider C/D combinations
3	DI^{U}	PD/SCL	Power-down input (run mode) or serial interface clock input (program mode)
4	P	VSS	Ground
5	AI	XIN	Crystal oscillator feedback
6	AO	XOUT	Crystal oscillator drive
7	DI^{U}O	OE/SDA	Output enable input (run mode) or serial interface data input/output (program mode)
8	P	VDD	Power supply (5V to 3.3V)
9	DI^{U}	MODE	Selects either program mode (low) or run mode (high)
10	DO	CLK_D	D clock output
11	P	VSS	Ground
12	DO	CLK_C	C clock output
13	DO	CLK_B	B clock output
14	P	VDD	Power supply (5V to 3.3V)
15	DO	CLK_A	A clock output
16	P	VDD	Power supply (5V to 3.3V)

Key: AI = Analog Input; AO = Analog Output; DI = Digital Input; DI^{U} = Input with Internal Pull-Up; DI$_{D}$ = Input with Internal Pull-Down; DIO = Digital Input/Output; DI-3 = Three-Level Digital Input, DO = Digital Output; P = Power/Ground; # = Active Low pin

3.0 Functional Block Description

3.1 Phase Locked Loops (PLLs)

Each of the three on-chip PLLs is a standard phase- and frequency-locked loop architecture that multiplies a reference frequency to a desired frequency by a ratio of integers. This frequency multiplication is exact.

As shown in Figure 3, each PLL consists of a reference divider, a phase-frequency detector (PFD), a charge pump, an internal loop filter, a voltage-controlled oscillator (VCO), and a feedback divider.

During operation, the reference frequency (f_{REF}), generated by the on-board crystal oscillator, is first reduced by the reference divider. The divider value is often referred to as the modulus, and is denoted as N_R for the reference divider. The divided reference is fed into the PFD.

The PFD controls the frequency of the VCO (f_{VCO}) through the charge pump and loop filter. The VCO provides a high-speed, low noise, continuously variable frequency clock source for the PLL. The output of the VCO is fed back to the PFD through the feedback divider (the modulus is denoted by N_F) to close the loop.

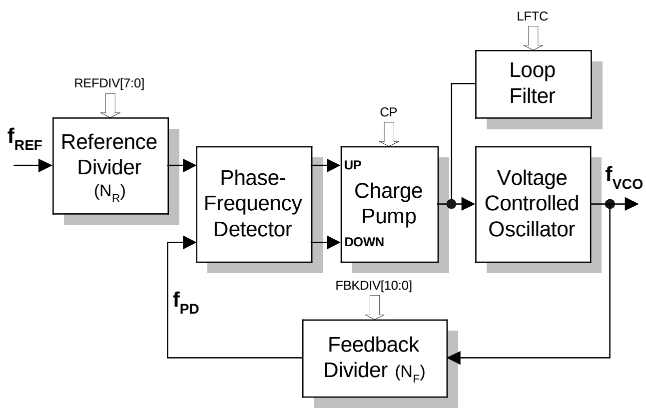


Figure 3: PLL Block Diagram

The PFD will drive the VCO up or down in frequency until the divided reference frequency and the divided VCO frequency appearing at the inputs of the PFD are equal. The input/output relationship between the reference frequency and the VCO frequency is:

$$f_{VCO} = f_{REF}\left(\frac{N_F}{N_R}\right)$$

3.1.1 Reference Divider

The reference divider is designed for low phase jitter. The divider accepts the output of the reference oscillator and provides a divided-down frequency to the PFD. The reference divider is an 8-bit divider, and can be programmed for any modulus from 1 to 255 by programming the equivalent binary value. A divide-by-256 can also be achieved by programming the eight bits to 00h.

3.1.2 Feedback Divider

The feedback divider is based on a dual-modulus pre-scaler technique. The technique allows the same granularity as a fully programmable feedback divider, while still allowing the programmable portion to operate at low speed. A high-speed pre-divider (also called a pre-scaler) is placed between the VCO and the programmable feedback divider because of the high speeds at which the VCO can operate. The dual-modulus technique insures reliable operation at any speed that the VCO can achieve and reduces the overall power consumption of the divider.

For example, a fixed divide-by-eight pre-scaler could have been used in the feedback divider. Unfortunately, a divide-by-eight would limit the effective modulus of the entire feedback divider to multiples of eight. This limitation would restrict the ability of the PLL to achieve a desired input-frequency-to-output-frequency ratio without making both the reference and feedback divider values comparatively large. Generally, very large values are undesirable as they degrade the bandwidth of the PLL, increasing phase jitter and acquisition time.

To understand the operation of the feedback divider, refer to Figure 4. The M-counter (with a modulus always equal to M) is cascaded with the dual-modulus pre-scaler. The A-counter controls the modulus of the pres-caler. If the value programmed into the A-counter is A, the pre-scaler will be set to divide by N+1 for A pre-scaler outputs. Thereafter, the prescaler divides by N until the M-counter output resets the A-counter, and the cycle begins again. Note that N=8, and A and M are binary numbers.

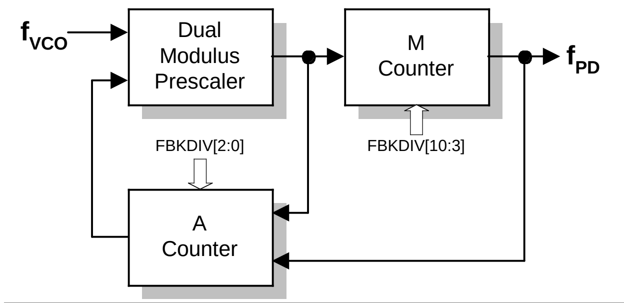


Figure 4: Feedback Divider

Suppose that the A-counter is programmed to zero. The modulus of the pre-scaler will always be fixed at N; and the entire modulus of the feedback divider becomes MxN.

Next, suppose that the A-counter is programmed to a one. This causes the pre-scaler to switch to a divide-by-N+1 for its first divide cycle and then revert to a divide-by-N. In effect, the A-counter absorbs (or "swallows") one extra clock during the entire cycle of the feedback divider. The overall modulus is now seen to be equal to MxN+1.

This example can be extended to show that the feedback divider modulus is equal to MxN+A, where A≤M.

3.1.3 Feedback Divider Programming

For proper operation of the feedback divider, the A-counter must be programmed only for values that are less than or equal to the M-counter. Therefore, not all divider moduli below 56 are available for use. This is shown in Table 2.

Above a modulus of 56, the feedback divider can be programmed to any value up to 2047.

Table 2: Feedback Divider Modulus Under 56

M-Counter: FBKDIV[10:3]	A-Counter: FBKDIV[2:0]							
	000	001	010	011	100	101	110	111
00000001	8	9	-	-	-	-	-	-
00000010	16	17	18	-	-	-	-	-
00000011	24	25	26	27	-	-	-	-
00000100	32	33	34	35	36	-	-	-
00000101	40	41	42	43	44	45	-	-
00000110	48	49	50	51	52	53	54	-
00000111	56	57	58	59	60	61	62	63
	Feedback Divider Modulus							

3.2 Post Divider Muxes

As shown in Figure 2, a mux in front of each post divider stage can select from any one of the three PLL frequencies or the reference frequency. The mux selection is controlled by bits in the EEPROM or the control registers.

The input frequency on two of the four multiplexers (muxes C and D in Figure 2) can be altered without reprogramming by a logic-level input on the SEL_CD pin.

3.3 Post Dividers

A post divider performs several useful functions. First, it allows the VCO to be operated in a narrower range of speeds compared to the variety of output clock speeds that the device is required to generate. Second, it changes the basic PLL equation to:

$$f_{CLK} = f_{REF}\left(\frac{N_F}{N_R}\right)\left(\frac{1}{N_P}\right)$$

where NP is the post divider modulus. The extra integer in the denominator permits more flexibility in the programming of the loop for many applications where frequencies must be achieved exactly.

The modulus on two of the four post dividers (post dividers C and D in Figure 2) can be altered without reprogramming by a logic level on the SEL_CD pin.

4.0 Device Operation

The FS6370 has two modes of operation:

- **Program mode:** during which either the EEPROM or the FS6370 control registers can be programmed directly with the desired PLL settings
- **Run mode:** where the PLL settings stored the EEPROM are transferred to the FS6370 control registers on power-up, and the device then operates based on those settings

Note that the EEPROM locations are not physically the same registers used to control the FS6370.

Direct access to either the EEPROM or the FS6370 control registers is achieved in program mode. The EEPROM register contents are automatically transferred to the FS6370 control registers in normal device operation (run mode).

4.1 MODE Pin

The MODE pin controls the mode of operation. A logic-low places the FS6370 in program mode. A logic-high puts the device in run mode. A pull-up on this pin defaults the device into run mode.

Reprogramming of either the control registers or the EEPROM is permitted at any time if the MODE pin is a logic-low.

Note, however, that a logic-high state on the MODE pin is latched so that only one transfer of EEPROM data to the FS6370 control registers can occur. If a second transfer of EEPROM data into the FS6370 is desired, power (VDD) must be removed and reapplied to the device.

The MODE pin also controls the function of the PD/SCL and OE/SDA pins. In run mode, these two pins function as power-down (PD) and output enable (OE) controls. In program mode, the pins function as the I^2C interface for clock (SCL) and data (SDA).

4.2 SEL_CD Pin

The SEL_CD pin provides a way to alter the operation of PLL C, muxes C and D, and post dividers C and D without having to reprogram the device. A logic-low on the SEL_CD pin selects the control bits with a "C1" or "D1" notation, per Table 3. A logic-high on the SEL_CD pin selects the control bits with "C2" or "D2" notation, per Table 3.

Note that changing between two running frequencies using the SEL_CD pin may produce glitches in the output, especially if the post-divider(s) is/are altered.

4.3 Oscillator Overdrive

For applications where an external reference clock is provided (and the crystal oscillator is not required), the reference clock should be connected to XOUT and XIN must be left unconnected (float).

For best results, make sure the reference clock signal is as jitter-free as possible, can drive a 40pF load with fast rise and fall times, and can swing rail-to-rail.

If the reference clock is not a rail-to-rail signal, the reference must be AC coupled to XOUT through a 0.01µF or 0.1µF capacitor. A minimum 1V peak-to-peak signal is required to drive the internal differential oscillator buffer.

5.0 Run Mode

If the MODE pin is set to a logic-high, the device enters the run mode. The high state is latched (see MODE pin). The FS6370 then copies the stored EEPROM data into its control registers and begins normal operation based on that data when the self-load is complete.

The self-load process takes about 89,000 clocks of the crystal oscillator. During the self-load time, all clock outputs are held low. At a reference frequency of 27MHz, the self-load takes about 3.3ms to complete.

If the EEPROM is empty (all zeros), the crystal reference frequency provides the clock for all four outputs.

No external programming access to the FS6370 is possible in run mode. The dual-function PD/SCL and OE/SDA pins become a power-down (PD) and output enable (OE) control, respectively.

5.1 Power-Down and Output Enable

A logic-high on the PD/SCL pin powers down only those portions of the FS6370 which have their respective power-down control bits enabled. Note that the PD/SCL pin has an internal pull-up.

When a post divider is powered down, the associated output driver is forced low. When all PLLs and post dividers are powered down the crystal oscillator is also powered down. The XIN pin is forced low, and the XOUT pin is pulled high.

A logic-low on the OE/SDA pin tristates all output clocks. Note that this pin has an internal pull-up.

6.0 Program Mode

If the MODE pin is logic-low, the device enters the program mode. All internal registers are cleared to zero, delivering the crystal frequency to all outputs. The device allows programming of either the internal 128-bit EEPROM or the on-chip control registers via I^{2}C control over the PD/SCL and OE/SDA pins. The EEPROM and the FS6370 act as two separate parallel devices on the same on-chip I^{2}C-bus. Choosing either the EEPROM or the device control registers is done via the I^{2}C device address.

The dual-function PD/SCL and OE/SDA pins become the serial data I/O (SDA) and serial clock input (SCL) for normal I^{2}C communications. Note that power-down and output enable control via the PD/SCL and OE/SDA pins is not available.

6.1 EEPROM Programming

Data must be loaded into the EEPROM in a most-significant-bit (MSB) to least-significant-bit (LSB) order. The register map of the EEPROM is noted in Table 3.

The device address of the EEPROM is:

A6	A5	A4	A3	A2	A1	A0
1	0	1	0	X	X	X

6.1.1 Write Operation

The EEPROM can only be written to with the random register write procedure (see Section 8.2.2). The procedure consists of the device address, the register address, a R/W bit, and one byte of data.

Following the STOP condition, the EEPROM initiates its internally timed 4ms write cycle, and commits the data byte to memory. No acknowledge signals are generated during the EEPROM internal write cycle.

If a stop bit is transmitted before the entire write command sequence is complete, then the command is aborted and no data is written to memory.
If more than eight bits are transmitted before the stop bit is sent, then the EEPROM will clear the previously loaded data byte and will begin loading the data buffer again.

6.1.2 Acknowledge Polling

The EEPROM does not acknowledge while it internally commits data to memory. This feature can be used to increase data throughput by determining when the internal write cycle is complete.

The process is to initiate the random register write procedure with a START condition, the EEPROM device address, and the write command bit (R/W=0). If the EEPROM has completed its internal 4ms write cycle, the EEPROM will acknowledge on the next clock, and the write command can continue.

If the EEPROM has not completed the internal 4ms write cycle, the random register write procedure must be restarted by sending the START condition, device address and R/W bit. This sequence must be repeated until the EEPROM acknowledges.

6.1.3 Read Operation

The EEPROM supports both the random register read procedure and the sequential register read procedure (both are outlined in Section 6).

For sequential read operations, the EEPROM has an internal address pointer that increments by one at the end of each read operation. The pointer directs the EEPROM to transmit the next sequentially addressed data byte, allowing the entire memory contents to be read in one operation.

6.2 Direct Register Programming

The FS6370 control registers may be directly accessed by simply using the FS6370 device address in the read or write operations. The operation of the device will follow the register values. The register map of the FS6370 is identical to that of the EEPROM shown in Table 3.

The FS6370 supports the random read and write procedures, as well as the sequential read and write procedures described in Section 8.

The device address for the FS6370 is:

A6	A5	A4	A3	A2	A1	A0
1	0	1	1	1	0	0

7.0 Cost Reduction Migration Path

The FS6370 is compatible with the programmable register-based FS6377 or a fixed-frequency ROM-based clock generator. Attention should be paid to the board layout if a migration path to either of these devices is desired.

7.1 Programming Migration Path

If the design can support I²C programming overhead, a cost reduction from the EEPROM-based FS6370 to the register-based FS6377 is possible.

Figure 5 shows the five pins that may not be compatible between the various devices if programming of the FS6370 or the FS6377 is desired.

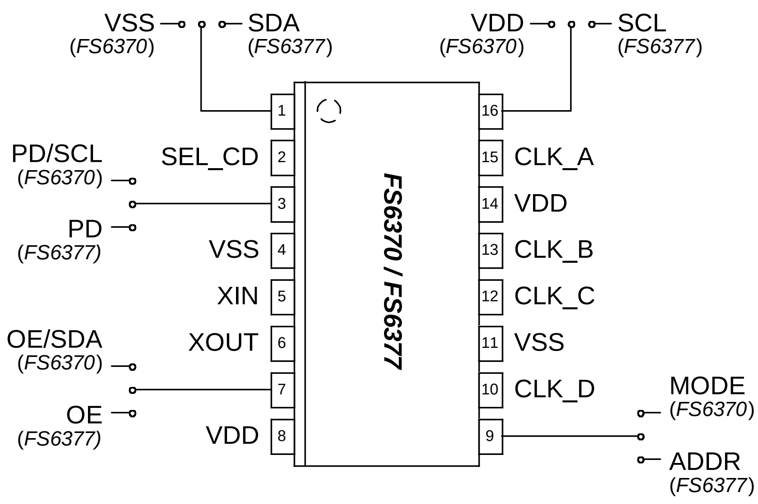


Figure 5: FS6370 to FS6377

7.2 Non-Programming Migration Path

If the design has solidified on a particular EEPROM programming pattern, the EEPROM pattern can be hard-coded into a ROM-based device. For high-volume requirements, a ROM-based device offers significant cost savings over the FS6370. Contact an AMIS sales representative for more detail.

8.0 I²C-bus Control Interface

This device is a read/write slave device meeting all Philips I²C-bus specifications except a "general call." The bus has to be controlled by a master device that generates the serial clock SCL, controls bus access and generates the START and STOP conditions while the device works as a slave. Both master and slave can operate as a transmitter or receiver, but the master device determines which mode is activated. A device that sends data onto the bus is defined as the transmitter, and a device receiving data as the receiver.

I²C-bus logic levels noted herein are based on a percentage of the power supply (V_{DD}). A logic-one corresponds to a nominal voltage of V_{DD}, while a logic-low corresponds to ground (V_{SS}).

8.1 Bus Conditions

Data transfer on the bus can only be initiated when the bus is not busy. During the data transfer, the data line (SDA) must remain stable whenever the clock line (SCL) is high. Changes in the data line while the clock line is high will be interpreted by the device as a START or STOP condition. The following bus conditions are defined by the I²C-bus protocol.

8.1.1 Not Busy

Both the data (SDA) and clock (SCL) lines remain high to indicate the bus is not busy.

8.1.2 START Data Transfer

A high to low transition of the SDA line while the SCL input is high indicates a START condition. All commands to the device must be preceded by a START condition.

8.1.3 STOP Data Transfer

A low to high transition of the SDA line while SCL is held high indicates a STOP condition. All commands to the device must be followed by a STOP condition.

8.1.4 Data Valid

The state of the SDA line represents valid data if the SDA line is stable for the duration of the high period of the SCL line after a START condition occurs. The data on the SDA line must be changed only during the low period of the SCL signal. There is one clock pulse per data bit.

Each data transfer is initiated by a START condition and terminated with a STOP condition. The number of data bytes transferred between START and STOP conditions is determined by the master device, and can continue indefinitely. However, data that is overwritten to the device after the first 16 bytes will overflow into the first register, then the second, and so on, in a first-in, first-overwritten fashion.

8.1.5 Acknowledge

When addressed, the receiving device is required to generate an acknowledge after each byte is received. The master device must generate an extra clock pulse to coincide with the acknowledge bit. The acknowledging device must pull the SDA line low during the high period of the master acknowledge clock pulse. Setup and hold times must be taken into account.

The master must signal an end of data to the slave by not generating an acknowledge bit on the last byte that has been read (clocked) out of the slave. In this case, the slave must leave the SDA line high to enable the master to generate a STOP condition.

8.2 I²C-bus Operation

All programmable registers can be accessed randomly or sequentially via this bi-directional two wire digital interface. The device accepts the following I²C-bus commands.

8.2.1 Device Address

After generating a START condition, the bus master broadcasts a seven-bit device address followed by a R/W bit.

The device address of the FS6370 is:

A6	A5	A4	A3	A2	A1	A0
1	0	1	1	1	0	0

Any one of eight possible addresses are available for the EEPROM. The least significant three bits are don't care's.

A6	A5	A4	A3	A2	A1	A0
1	0	1	0	X	X	X

8.2.2 Random Register Write Procedure

Random write operations allow the master to directly write to any register. To initiate a write procedure, the R/W bit that is transmitted after the seven-bit device address is a logic-low. This indicates to the addressed slave device that a register address will follow after the slave device acknowledges its device address. The register address is written into the slave's address pointer. Following an acknowledge by the slave, the master is allowed to write eight bits of data into the addressed register. A final acknowledge is returned by the device, and the master generates a STOP condition.

If either a STOP or a repeated START condition occurs during a register write, the data that has been transferred is ignored.

8.2.3 Random Register Read Procedure

Random read operations allow the master to directly read from any register. To perform a read procedure, the R/W bit that is transmitted after the seven-bit address is a logic-low, as in the register write procedure. This indicates to the addressed slave device that a register address will follow after the slave device acknowledges its device address. The register address is then written into the slave's address pointer.

Following an acknowledge by the slave, the master generates a repeated START condition. The repeated START terminates the write procedure, but not until after the slave's address pointer is set. The slave address is then resent, with the R/W bit set this time to a logic-high, indicating to the slave that data will be read. The slave will acknowledge the device address, and then transmits the eight-bit word. The master does not acknowledge the transfer but does generate a STOP condition.

8.2.4 Sequential Register Write Procedure

Sequential write operations allow the master to write to each register in order. The register pointer is automatically incremented after each write. This procedure is more efficient than the random register write if several registers must be written.

To initiate a write procedure, the R/W bit that is transmitted after the seven-bit device address is a logic-low. This indicates to the addressed slave device that a register address will follow after the slave device acknowledges its device address. The register address is written into the slave's address pointer. Following an acknowledge by the slave, the master is allowed to write up to 16 bytes of data into the addressed register before the register address pointer overflows back to the beginning address. An acknowledge by the device between each byte of data must occur before the next data byte is sent.

Registers are updated every time the device sends an acknowledge to the host. The register update does not wait for the STOP condition to occur. Registers are therefore updated at different times during a sequential register write.

8.2.5 Sequential Register Read Procedure

Sequential read operations allow the master to read from each register in order. The register pointer is automatically incremented by one after each read. This procedure is more efficient than the random register read if several registers must be read.

To perform a read procedure, the R/W bit that is transmitted after the seven-bit address is a logic-low, as in the register write procedure. This indicates to the addressed slave device that a register address will follow after the slave device acknowledges its device address. The register address is then written into the slave's address pointer.

Following an acknowledge by the slave, the master generates a repeated START condition. The repeated START terminates the write procedure, but not until after the slave's address pointer is set. The slave address is then resent, with the R/W bit set this time to a logic-high, indicating to the slave that data will be read. The slave will acknowledge the device address, and then transmits all 16 bytes of data starting with the initial addressed register. The register address pointer will overflow if the initial register address is larger than zero. After the last byte of data, the master does not acknowledge the transfer but does generate a STOP condition.

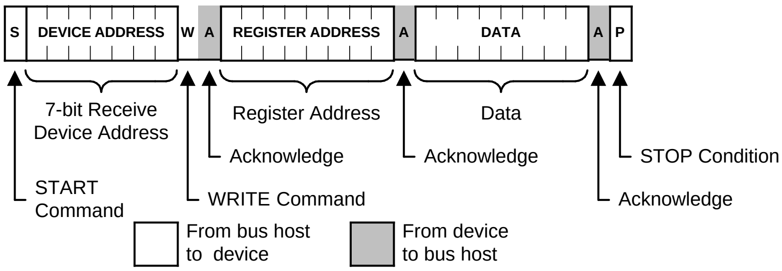


Figure 6: Random Register Write Procedure

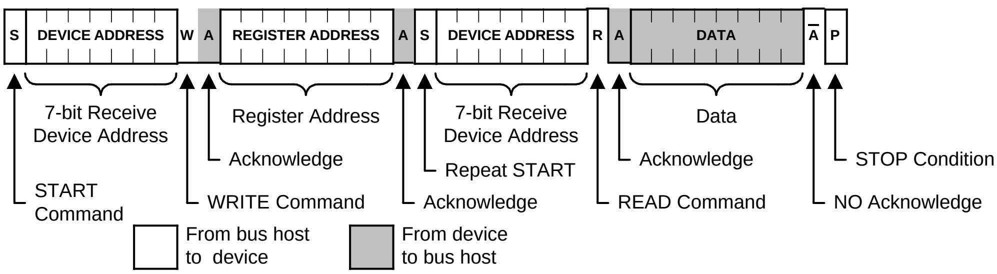


Figure 7: Random Register Read Procedure

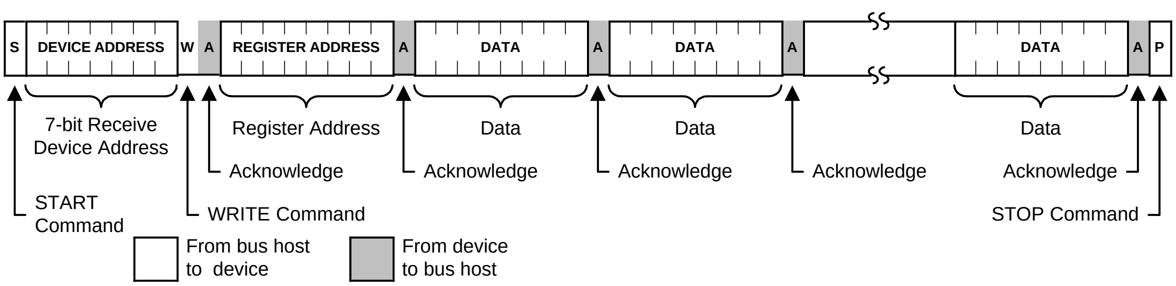


Figure 8: Sequential Register Write Procedure

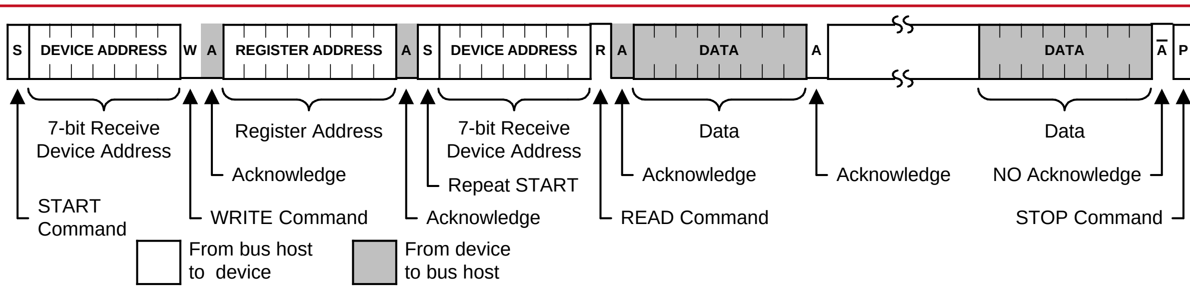


Figure 9: Sequential Register Read Procedure

9.0 Programming Information

Table 3: Register Map (Note: All register bits are cleared to zero on power-up.)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 15	MUX_D2[1:0] (selected via SEL_CD = 1)		MUX_C2[1:0] (selected via SEL_CD = 1)		PDPOST_D	PDPOST_C	PDPOST_B	PDPOST_A
Byte 14	POST_D2[3:0] (selected via SEL_CD = 1)				POST_C2[3:0] (selected via SEL_CD = 1)			
Byte 13	POST_D1[3:0] (selected via SEL_CD = 0)				POST_C1[3:0] (selected via SEL_CD = 0)			
Byte 12	POST_B[3:0]				POST_A[3:0]			
Byte 11	MUX_D1[1:0] (selected via SEL_CD = 0)		Reserved (0)	LFTC_C2 (SEL_CD=1)	CP_C2 (SEL_CD=1)	FBKDIV_C2[10:8] *M-Counter* (selected via SEL_CD pin = 1)		
Byte 10	FBKDIV_C2[7:3] *M-Counter* (selected via SEL_CD pin = 1)					FBKDIV_C2[2:0] *A-Counter* (selected via SEL_CD pin = 1)		
Byte 9	REFDIV_C2[7:0] (selected via SEL_CD pin = 1)							
Byte 8	MUX_C1[1:0] (selected via SEL_CD = 0)		PDPLL_C	LFTC_C1 (SEL_CD=0)	CP_C1 (SEL_CD=0)	FBKDIV_C1[10:8] *M-Counter* (selected via SEL_CD = 0)		
Byte 7	FBKDIV_C1[7:3] *M-Counter* (selected via SEL_CD = 0)					FBKDIV_C1[2:0] *A-Counter* (selected via SEL_CD = 1)		
Byte 6	REFDIV_C1[7:0] (selected via SEL_CD = 0)							
Byte 5	MUX_B[1:0]		PDPLL_B	LFTC_B	CP_B	FBKDIV_B[10:8] *M-Counter*		
Byte 4	FBKDIV_B[7:3] *M-Counter*					FBKDIV_B[2:0] *A-Counter*		
Byte 3	REFDIV_B[7:0]							
Byte 2	MUX_A[1:0]		PDPLL_A	LFTC_A	CP_A	FBKDIV_A[10:8] *M-Counter*		
Byte 1	FBKDIV_A[7:3] *M-Counter*					FBKDIV_A[2:0] *A-Counter*		
Byte 0	REFDIV_A[7:0]							

9.1 Control Bit Assignments

If any PLL control bit is altered during device operation, including those bits controlling the reference and feedback dividers, the output frequency will slew smoothly (in a glitch-free manner) to the new frequency. The slew rate is related to the programmed loop filter time constant.

However, any programming changes to any mux or post divider control bits will cause a glitch on an operating clock output.

9.1.1 Power-Down

All power-down functions are controlled by enable bits. That is, the bits select which portions of the FS6370 to power-down when the PD input is asserted. If the power-down bit contains a one, the related circuit will shut down if the PD pin is high (run mode only). When the PD pin is low, power is enabled to all circuits.

If the power-down bit contains a zero, the related circuit will continue to function regardless of the PD pin state.

Table 4: Power-Down Bits

Name	Description	
	Power-Down PLL A	
PDPLL_A (Bit 21)	Bit = 0	Power on
	Bit = 1	Power off
	Power-Down PLL B	
PDPLL_B (Bit 45)	Bit = 0	Power on
	Bit = 1	Power off
	Power-Down PLL C	
PDPLL_C (Bit 69)	Bit = 0	Power on
	Bit = 1	Power off
Reserved (0) (Bit 69)	Set these reserved bits to zero (0)	
	Power-Down POST divider A	
PDPOST_A (Bit 120)	Bit = 0	Power on
	Bit = 1	Power off
	Power-Down POST divider B	
PDPOST_B (Bit 121)	Bit = 0	Power on
	Bit = 1	Power off
	Power-Down POST divider C	
PDPOSTC (Bit 122)	Bit = 0	Power on
	Bit = 1	Power off
	Power-Down POST divider D	
PDPOSTD (Bit 123)	Bit = 0	Power on
	Bit = 1	Power off

Table 5: Divider Control Bits

Name	Description	
REFDIV_A[7:0] (Bits 7-0)	Reference Divider A (N_R)	
REFDIV_B[7:0] (Bits 31-24)	Reference Divider B (N_R)	
REFDIV_C1[7:0] (Bits 55-48)	Reference Divider C1 (N_R) *selected when the SEL-CD pin = 0*	
REFDIV_C2[7:0] (Bits 79-72)	Reference Divider C2 (N_R) *selected when the SEL-CD pin = 1*	
	Feedback Divider A (N_F)	
FBKDIV_A[10:0] (Bits 18-8)	FBKDIV_A[2:0]	A-Counter value
	FBKDIV_A[10:3]	M-Counter value
	Feedback Divider B (N_F)	
FBKDIV_B[10:0] (Bits 42-32)	FBKDIV_B[2:0]	A-Counter value
	FBKDIV_B[10:3]	M-Counter value
	Feedback Divider C1 (N_F) *selected when the SEL-CD pin = 0*	
FBKDIV_C1[10:0] (Bits 66-56)	FBKDIV_C1[2:0]	A-Counter value
	FBKDIV_C1[10:3]	M-Counter value
	Feedback Divider C2 (N_F) *selected when the SEL-CD pin = 1*	
FBKDIV_C2[10:0] (Bits 90-80)	FBKDIV_C2[2:0]	A-Counter value
	FBKDIV_C2[10:3]	M-Counter value

Table 6: Post Divider Control Bits

Name	Description
POST_A[3:0] (Bits 99-96)	POST divider A (see Table 7)
POST_B[3:0] (Bits 103-100)	POST divider B (see Table 7)
POST_C1[3:0] (Bits 107-104)	POST divider C1 (see Table 7) *selected when the SEL_CD pin = 0*
POST_C2[3:0] (Bits 115-112)	POST divider C2 (see Table 7) *selected when the SEL_CD pin = 1*
POST_D1[3:0] (Bits 111-108)	POST divider D1 (see Table 7) *selected when the SEL_CD pin = 0*
POST_D2[3:0] (Bits 119-116)	POST divider D2 (see Table 7) *selected when the SEL_CD pin = 1*

Table 7: Post Divider Modulus

Bit [3]	Bit [2]	Bit [1]	Bit [0]	Divide By
0	0	0	0	1
0	0	0	1	2
0	0	1	0	3
0	0	1	1	4
0	1	0	0	5
0	1	0	1	6
0	1	1	0	8
0	1	1	1	9
1	0	0	0	10
1	0	0	1	12
1	0	1	0	15
1	0	1	1	16
1	1	0	0	18
1	1	0	1	20
1	1	1	0	25
1	1	1	1	50

Table 8: PLL Tuning Bits

Name	Description	
	Loop Filter Time Constant A	
LFTC_A (Bit 20)	Bit = 0	Short time constant: 7µs
	Bit = 1	Long time constant: 20µs
	Loop Filter Time Constant B	
LFTC_B (Bit 44)	Bit = 0	Short time constant: 7µs
	Bit = 1	Long time constant: 20µs
	Loop Filter Time Constant C1 *selected when the SEL_CD pin = 0*	
LFTC_C1 (Bit 68)	Bit = 0	Short time constant: 7µs
	Bit = 1	Long time constant: 20µs
	Loop Filter Time Constant C2 *selected when the SEL_CD pin = 1*	
LFTC_C2 (Bit 92)	Bit = 0	Short time constant: 7µs
	Bit = 1	Long time constant: 20µs
	Charge Pump A	
CP_A (Bit 19)	Bit = 0	Current = 2µA
	Bit = 1	Current = 10µA
	Charge Pump B	
CP_B (Bit 43)	Bit = 0	Current = 2µA
	Bit = 1	Current = 10µA
	Charge Pump C1 *selected when the SEL_CD pin = 0*	
CP_C1 (Bit 67)	Bit = 0	Current = 2µA
	Bit = 1	Current = 10µA
	Charge Pump C2 *selected when the SEL_CD pin = 1*	
CP_C2 (Bit 91)	Bit = 0	Current = 2µA
	Bit = 1	Current = 10µA

Table 9: Mux Select Bits

Name	Description		
	MUX A Frequency Select		
MUX_A[1:0] (Bits 23-22)	Bit 23	Bit 22	
	0	0	Reference frequency
	0	1	PLL A frequency
	1	0	PLL B frequency
	1	1	PLL C frequency
	MUX B Frequency Select		
MUX_B[1:0] (Bits 47-46)	Bit 47	Bit 46	
	0	0	Reference frequency
	0	1	PLL A frequency
	1	0	PLL B frequency
	1	1	PLL C frequency
	MUX C1 Frequency Select *selected when the SEL_CD pin = 0*		
MUX_C1[1:0] (Bits 71-70)	Bit 71	Bit 70	
	0	0	Reference frequency
	0	1	PLL A frequency
	1	0	PLL B frequency
	1	1	PLL C frequency
	MUX C2 Frequency Select *selected when the SEL_CD pin = 1*		
MUX_C2[1:0] (Bits 125-124)	Bit 125	Bit 124	
	0	0	Reference frequency
	0	1	PLL A frequency
	1	0	PLL B frequency
	1	1	PLL C frequency
	MUX D1 Frequency Select *selected when the SEL_CD pin = 0*		
MUX_D1[1:0] (Bits 95-94)	Bit 95	Bit 94	
	0	0	Reference frequency
	0	1	PLL A frequency
	1	0	PLL B frequency
	1	1	PLL C frequency
	MUX D2 Frequency Select *selected when the SEL_CD pin = 1*		
MUX_D2[1:0] (Bits 127-126)	Bit 127	Bit 126	
	0	0	Reference frequency
	0	1	PLL A frequency
	1	0	PLL B frequency
	1	1	PLL C frequency

10.0 Electrical Specifications

Table 10: Absolute Maximum Ratings

Parameter	Symbol	Min.	Max.	Units
Supply Voltage, dc (V_{SS} = ground)	V_{DD}	V_{SS}-0.5	7	V
Input Voltage, dc	V_I	V_{SS}-0.5	V_{DD}+0.5	V
Output Voltage, dc	V_O	V_{SS}-0.5	V_{DD}+0.5	V
Input Clamp Current, dc (V_I < 0 or V_I > V_{DD})	I_{IK}	-50	50	mA
Output Clamp Current, dc (V_I < 0 or V_I > V_{DD})	I_{OK}	-50	50	mA
Storage Temperature Range (non-condensing)	T_S	-65	150	°C
Ambient Temperature Range, Under Bias	T_A	-55	125	°C
Junction Temperature	T_J		150	°C
Re-Flow Solder Profile				Per IPC/JEDEC J-STD-020B
Input Static Discharge Voltage Protection (MIL-STD 883E, Method 3015.7)			2	kV

Stresses above those listed under absolute maximum ratings may cause permanent damage to the device. These conditions represent a stress rating only, and functional operation of the device at these or any other conditions above the operational limits noted in this specification is not implied. Exposure to maximum rating conditions for extended conditions may affect device performance, functionality and reliability.

CAUTION: ELECTROSTATIC SENSITIVE DEVICE

Permanent damage resulting in a loss of functionality or performance may occur if this device is subjected to a high-energy electrostatic discharge.

Table 11: Operating Conditions

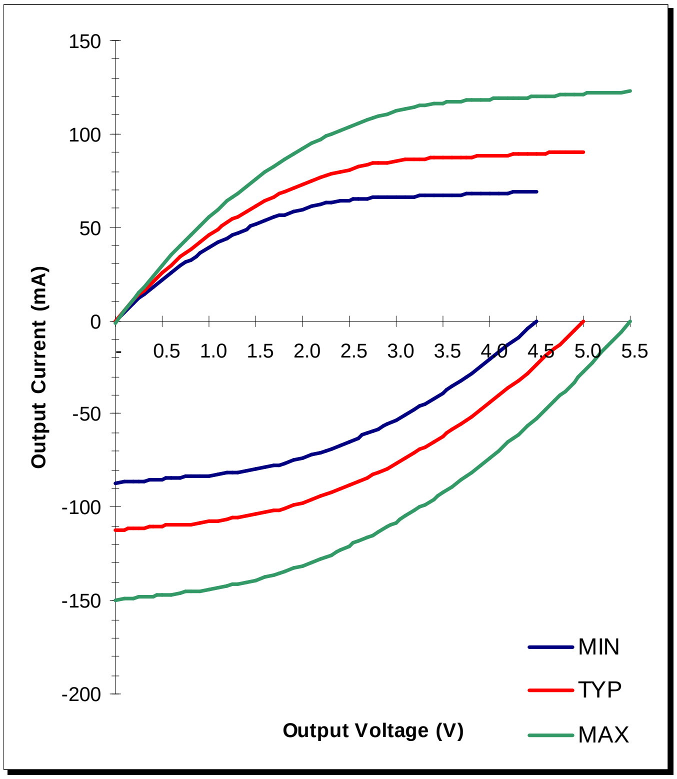
Parameter	Symbol	Conditions/Description	Min.	Typ.	Max.	Units
Supply Voltage	V_{DD}	5V ± 10%	4.5	5	5.5	V
		3.3V ± 10%	3	3.3	3.6	
Ambient Operating Temperature Range	T_A		0		70	°C
Crystal Resonator Frequency	f_{XIN}		5		27	MHz
Crystal Resonator Load Capacitance	C_{XL}	Parallel resonant, AT cut		18		pF
Serial Data Transfer Rate		Standard mode	10		100	kb/s
Output Driver Load Capacitance	C_L				15	pF

Table 12: DC Electrical Specifications

Parameter	Symbol	Conditions/Description		Min.	Typ.	Max.	Units
Overall							
Supply Current, Dynamic	I_{DD}	V_{DD} = 5.5V, f_{CLK} = 50MHz, C_L = 15pF *See Figure 11 for more information*			43		mA
Supply Current, Write	$I_{DD(write)}$	Additional operating current demand, EEPROM program mode, V_{DD} = 5.5V			2		mA
Supply Current, Read	$I_{DD(read)}$	Additional operating current demand, EEPROM program mode, V_{DD} = 5.5V			1		mA
Supply Current, Static	I_{DDL}	V_{DD} = 5.5V, powered down via PD pin			0.3		mA
Dual Function I/O (PD/SCL, OESDA)							
High-Level Input Voltage	V_{IH}	Run mode (PD, OE)	V_{DD} = 5.5V	3.85		V_{DD}+0.3	V
			V_{DD} = 3.6V	2.52		V_{DD}+0.3	
		Register program mode (SDA, SCL)	V_{DD} = 5.5V	3.85		V_{DD}+0.3	
			V_{DD} = 3.6V	2.52		V_{DD}+0.3	
		EEPROM program mode (SDA, SCL)	V_{DD} = 5.5V	3.85		V_{DD}+0.3	
			V_{DD} = 3.6V	2.52		V_{DD}+0.3	
Low-Level Input Voltage	V_{IL}	Run mode (PD, OE)	V_{DD} = 5.5V	V_{SS}-0.3		1.65	V
			V_{DD} = 3.6V	V_{SS}-0.3		1.08	
		Register program mode (SDA, SCL)	V_{DD} = 5.5V	V_{SS}-0.3		1.65	
			V_{DD} = 3.6V	V_{SS}-0.3		1.08	
		EEPROM program mode (SDA, SCL)	V_{DD} = 5.5V	V_{SS}-0.3		1.65	
			V_{DD} = 3.6V	V_{SS}-0.3		1.08	
Hysteresis Voltage	V_{hys}	Run mode (PD, OE)	V_{DD} = 5.5V		2.20		V
			V_{DD} = 3.6V		1.44		
		Register program mode (SDA, SCL)	V_{DD} = 5.5V		2.20		
			V_{DD} = 3.6V		1.44		
		EEPROM program mode (SDA, SCL)	V_{DD} = 5.5V		0.275		
			V_{DD} = 3.6V		0.18		
High-Level Input Current	I_{IH}	Run/register program mode		-1		1	µA
		EEPROM program mode		-1		1	
Low-Level Input Current (pull-up)	I_{IL}	V_{IL} = 0V		-20	-36	-80	µA
Low-Level Output Sink Current (SDA)	I_{OL}	Run/register program mode, V_{OL} = 0.4V			26		mA
		EEPROM program mode, V_{OL} = 0.4V			3.0		
Mode and Frequency Select Inputs (MODE, SEL_CD)							
High-Level Input Voltage	V_{IH}	V_{DD} = 5.5V		2.4		V_{DD}+0.3	V
		V_{DD} = 3.6V		2.0		V_{DD}+0.3	
Low-Level Input Voltage	V_{IL}	V_{DD} = 5.5V		V_{SS}-0.3		0.8	V
		V_{DD} = 3.6V		V_{SS}-0.3		0.8	
High-Level Input Current	I_{IH}			-1		1	µA
Low-Level Input Current (pull-up)	I_{IL}			-20	-36	-80	µA
Crystal Oscillator Feedback (XIN)							
Threshold Bias Voltage	V_{TH}	V_{DD} = 5.5V			2.9		V
		V_{DD} = 3.6V			1.7		
High-Level Input Current	I_{IH}	V_{DD} = 5.5V			54		µA
		V_{DD} = 5.5V, oscillator powered down		5		15	mA
Low-Level Input Current	I_{IL}			-25	-54	-75	µA
Crystal Loading Capacitance*	$C_{L(xtal)}$	As seen by an external crystal connected to XIN and XOUT			18		pF
Input Loading Capacitance*	$C_{L(XIN)}$	As seen by an external clock driver on XOUT; XIN unconnected			36		pF
Crystal Oscillator Drive (XOUT)							
High-Level Output Source Current	I_{OH}	V_{DD} = V(XIN) = 5.5V, V_O = 0V		10	21	30	mA
Low-Level Output Sink Current	I_{OL}	V_{DD} = 5.5V, V(XIN) = V_O = 5.5V		-10	-21	-30	mA
Clock Outputs (CLK_A, CLK_B, CLK_C, CLK_D)							
High-Level Output Source Current	I_{OH}	V_O = 2.4V			-125		mA
Low-Level Output Sink Current	I_{OL}	V_O = 0.4V			23		mA
Output Impedance	Z_{OH}	V_O = 0.5V_{DD}; output driving high			29		Ω
	Z_{OL}	V_O = 0.5V_{DD}; output driving low			27		
Tristate Output Current	I_Z			-10		10	µA
Short Circuit Source Current*	I_{SCH}	V_{DD} = 5.5V, V_O = 0V; shorted for 30s, max			-150		mA
Short Circuit Sink Current*	I_{SCL}	V_{DD} = V_O = 5.5V; shorted for 30s, max			123		mA

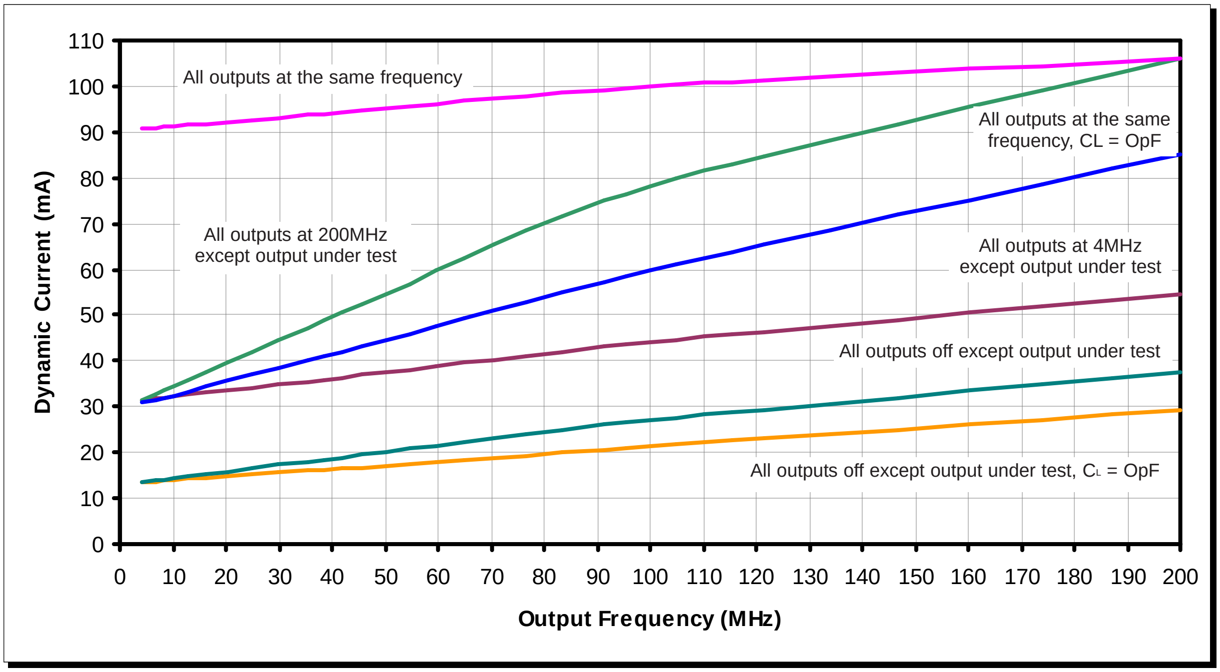
Unless otherwise stated, V_{DD} = 5.0V ± 10%, no load on any output, and ambient temperature range T_A = 0°C to 70°C. Parameters denoted with an asterisk (*) represent nominal characterization data and are not currently production tested to any specific limits. Min. and Max. characterization data are ± 3σ from typical. Negative currents indicate current flows out of the device.

Voltage (V)	Low Drive Current (mA)			Voltage (V)	High Drive Current (mA)		
	Min.	Typ.	Max.		Min.	Typ.	Max.
0	0	0	0	0	-87	-112	-150
0.2	9	11	12	0.5	-85	-110	-147
0.5	22	25	29	1	-83	-108	-144
0.7	29	34	40	1.5	-80	-104	-139
1	39	46	55	2	-74	-97	-131
1.2	44	52	64	2.5	-65	-88	-121
1.5	51	61	76	2.7	-61	-84	-116
1.7	55	66	83	3	-53	-77	-108
2	60	73	92	3.2	-48	-71	-102
2.2	62	77	97	3.5	-39	-62	-92
2.5	65	81	104	3.7	-32	-55	-85
2.7	65	83	108	4	-21	-44	-74
3	66	85	112	4.2	-13	-36	-65
3.5	67	87	117	4.5	0	-24	-52
4	68	88	119	4.7		-15	-43
4.5	69	89	120	5		0	-28
5		91	121	5.2			-11
5.5			123	5.5			0

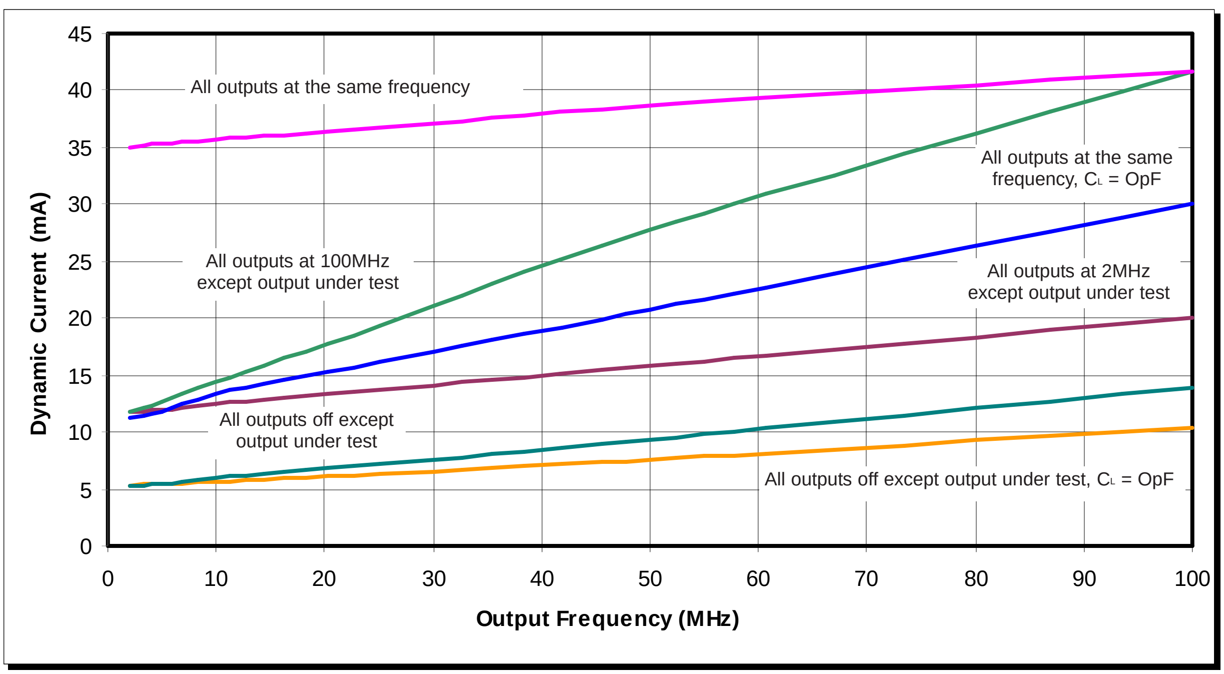


The data in this table represents nominal characterization data only.

Figure 10: CLK_A, CLK_B, CLK_C, CLK_D Clock Outputs



V_{DD} = 5.0V; Reference Frequency = 27.00MHz; VCO Frequency = 200MHz, C_L = 17pF except where noted



V_{DD} = 3.3V; Reference Frequency = 27.00MHz; VCO Frequency = 100MHz, C_L = 17pF except where noted

Figure 11: Dynamic Current vs. Output Frequency

Table 13: AC Timing Specifications

Parameter	Symbol	Conditions/Description	Clock (MHz)	Min.	Typ.	Max.	Units
Overall							
EEPROM Write Cycle Time	t_{WC}					4	ms
Output Frequency*	f_O	V_{DD} = 5.5V		0.8		150	MHz
		V_{DD} = 3.6V		0.8		100	
VCO Frequency*	f_{VCO}	V_{DD} = 5.5V		40		230	MHz
		V_{DD} = 3.6V		40		170	
VCO Gain*	A_{VCO}				400		MHz/V
Loop Filter Time Constant*		LFTC bit = 0			7		µs
		LFTC bit = 1			20		
Rise Time*	t_r	V_O = 0.5V to 4.5V; C_L = 15pF			2.0		ns
		V_O = 0.3V to 3.0V; C_L = 15pF			2.1		
Fall Time*	t_f	V_O = 4.5V to 0.5V; C_L = 15pF			1.8		ns
		V_O = 3.0V to 0.3V; C_L = 15pF			1.9		
Tristate Enable Delay*	t_{PZL}, t_{PZH}			1		8	ns
Tristate Disable Delay*	t_{PZL}, t_{PZH}			1		8	ns
Clock Stabilization Time*	t_{STB}	Output active from power-up, RUN mode via PD pin			100		µs
		After last register is written, register program mode				1	ms
Divider Modulus							
Feedback Divider	N_F	See also Table 2		8		2047	
Reference Divider	N_R			1		255	
Post Divider	N_P	See also Table 8		1		50	
Clock Outputs (PLL A clock via CLK_A pin)							
Duty Cycle*		Ratio of pulse width (as measured from rising edge to next falling edge at 2.5V) to one clock period	100	45		55	%
Jitter, Long Term ($\sigma_y(\tau)$)*	$t_{j(LT)}$	On rising edges 500µs apart at 2.5V relative to an ideal clock, C_L=15pF, f_{XIN}=14.318MHz, N_F=220, N_R=63, N_{Px}=50, no other PLLs active	100		45		ps
		On rising edges 500µs apart at 2.5V relative to an ideal clock, C_L=15pF, =14.318MHz, N_F=220, N_R=63, N_{Px}=50, all other PLLs active (B=60MHz, C=40MHz, D=14.318MHz)	50		165		
Jitter, Period (peak-peak)*	$t_{j(\Delta P)}$	From rising edge to the next rising edge at 2.5V, C_L=15pF, f_{XIN}=14.318MHz, N_F=220, N_R=63, N_{Px}=50, no other PLLs active	100		110		ps
		From rising edge to the next rising edge at 2.5V, C_L=15pF, f_{XIN}=14.318MHz, N_F=220, N_R=63, N_{Px}=50, all other PLLs active (B=60MHz, C=40MHz, D=14.318MHz)	50		390		
Clock Outputs (PLL B clock via CLK_B pin)							
Duty Cycle*		Ratio of pulse width (as measured from rising edge to next falling edge at 2.5V) to one clock period	100	45		55	%
Jitter, Long Term ($\sigma_y(\tau)$)*	$T_{j(LT)}$	On rising edges 500µs apart at 2.5V relative to an ideal clock, C_L=15pF, f_{XIN}=14.318MHz, N_F=220, N_R=63, N_{Px}=50, no other PLLs active	100		45		ps
		On rising edges 500µs apart at 2.5V relative to an ideal clock, C_L=15pF, f_{XIN}=14.318MHz, N_F=220, N_R=63, N_{Px}=50, all other PLLs active (A=50MHz, C=40MHz, D=14.318MHz)	60		75		
Jitter, Period (peak-peak)*	$T_{J(\Delta P)}$	From rising edge to the next rising edge at 2.5V, C_L=15pF, f_{XIN}=14.318MHz, N_F=220, N_R=63, N_{Px}=50, no other PLLs active	100		120		ps
		From rising edge to the next rising edge at 2.5V, C_L=15pF, f_{XIN}=14.318MHz, N_F=220, N_R=63, N_{Px}=50, all other PLLs active (A=50MHz, C=40MHz, D=14.318MHz)	60		400		
Clock Outputs (PLL_C clock via CLK_C pin)							
Duty Cycle*		Ratio of pulse width (as measured from rising edge to next falling edge at 2.5V) to one clock period	100	45		55	%
Jitter, Long Term ($\sigma_y(\tau)$)*	$T_{j(LT)}$	On rising edges 500µs apart at 2.5V relative to an ideal clock, C_L=15pF, f_{XIN}=14.318MHz, N_F=220, N_R=63, N_{Px}=50, no other PLLs active	100		45		ps
		On rising edges 500µs apart at 2.5V relative to an ideal clock, C_L=15pF, f_{XIN}=14.318MHz, N_F=220, N_R=63, N_{Px}=50, all other PLLs active (A=50MHz, B=60MHz, D=14.318MHz)	40		105		
Jitter, Period (peak-peak)*	$T_{J(\Delta P)}$	From rising edge to the next rising edge at 2.5V, C_L=15pF, f_{XIN}=14.318MHz, N_F=220, N_R=63, N_{Px}=50, no other PLLs active	100		120		ps
		From rising edge to the next rising edge at 2.5V, C_L=15pF, f_{XIN}=14.318MHz, N_F=220, N_R=63, N_{Px}=50, all other PLLs active (A=50MHz, B=60MHz, D=14.318MHz)	40		440		
Clock Outputs (Crystal Oscillator via CLK_D pin)							
Duty Cycle*		Ratio of pulse width (as measured from rising edge to next falling edge at 2.5V) to one clock period	14.318	45		55	%
Jitter, Long Term ($\sigma_y(\tau)$)*	$T_{j(LT)}$	On rising edges 500µs apart at 2.5V relative to an ideal clock, C_L=15pF, f_{XIN}=14.318MHz, N_F=220, N_R=63, N_{Px}=50, no other PLLs active	14.318		20		ps
		From rising edge to the next rising edge at 2.5V, C_L=15pF, f_{XIN}=14.318MHz, all other PLLs active (A=50MHz, B=60MHz, C=40MHz)	14.318		40		
Jitter, Period (peak-peak)*	$T_{J(\Delta P)}$	From rising edge to the next rising edge at 2.5V, C_L=15pF, f_{XIN}=14.318MHz, no other PLLs active	14.318		90		ps
		From rising edge to the next rising edge at 2.5V, C_L=15pF, f_{XIN}=14.318MHz, all other PLLs active (A=50MHz, B=60MHz, C=40MHz)	14.318		450		

Unless otherwise stated, V_{DD} = 5.0V ± 10%, no load on any output, and ambient temperature range T_A = 0°C to 70°C. Parameters denoted with an asterisk (*) represent nominal characterization data and are not currently production tested to any specific limits. Min. and Max. characterization data are ± 3σ from typical.

Table 14: Serial Interface Timing Specifications

Parameter	Symbol	Conditions/Description	Standard Mode		Units
			Min.	Max.	
Clock Frequency	f_{SCL}	SCL	0	100	kHz
Bus Free Time Between STOP and START	t_{BUF}		4.7		μs
Set-up Time, START (repeated)	$t_{su:STA}$		4.7		μs
Hold Time, START	$t_{hd:STA}$		4.0		μs
Set-up Time, Data Input	$t_{su:DAT}$	SDA	250		ns
Hold Time, Data Input	$t_{hd:DAT}$	SDA	0		μs
Output Data Valid From Clock	t_{AA}	Minimum delay to bridge undefined region of the falling edge of SCL to avoid unintended START or STOP		3.5	μs
Rise Time, Data and Clock	t_R	SDA, SCL		1000	ns
Fall Time, Data and Clock	t_F	SDA, SCL		300	ns
High Time, Clock	t_{HI}	SCL	4.0		μs
Low Time, Clock	t_{LO}	SCL	4.7		μs
Set-up Time, STOP	$t_{su:STO}$		4.0		μs

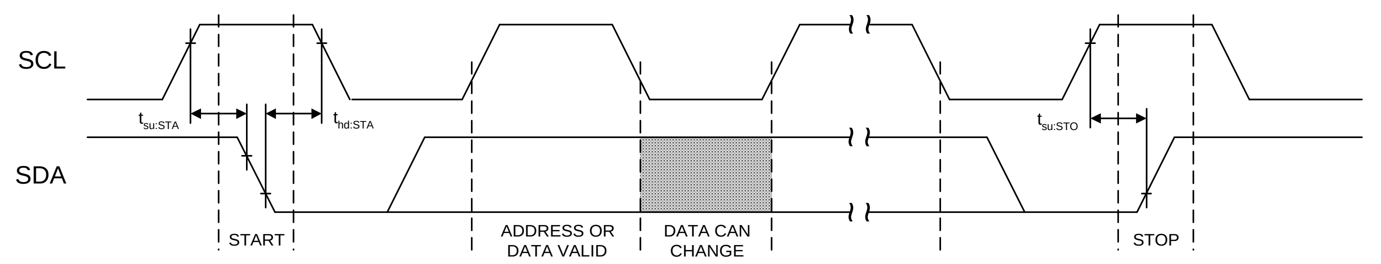


Figure 12: Bus Timing Data

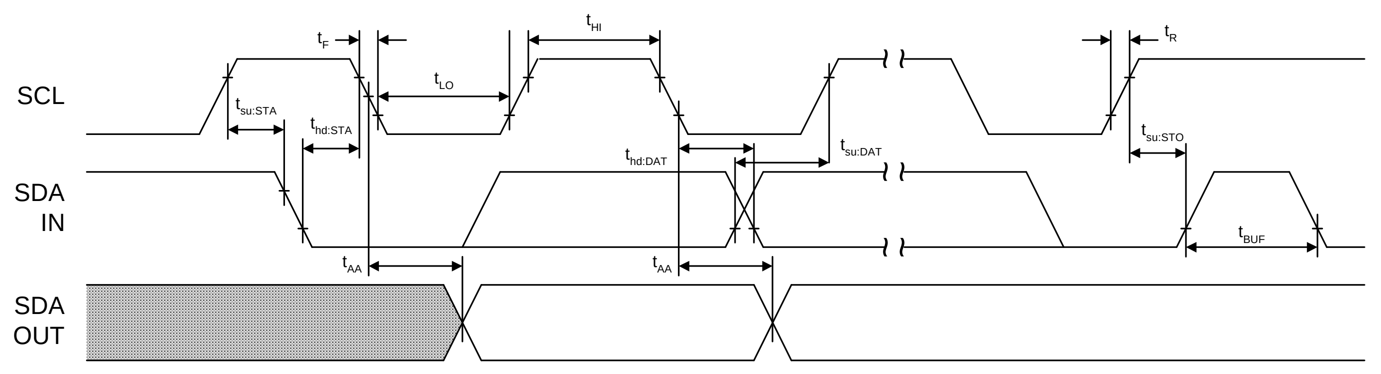


Figure 13: Data Transfer Sequence

11.0 Package Information For Both 'Green' and 'Non-Green'

Table 15: 16-pin SOIC (0.150") Package Dimensions

	Dimensions			
	Inches		Millimeters	
	Min.	Max.	Min.	Max.
A	0.061	0.068	1.55	1.73
A1	0.004	0.0098	0.102	0.249
A2	0.055	0.061	1.40	1.55
B	0.013	0.019	0.33	0.49
C	0.0075	0.0098	0.191	0.249
D	0.386	0.393	9.80	9.98
E	0.150	0.157	3.81	3.99
e	0.050 BSC		1.27 BSC	
H	0.230	0.244	5.84	6.20
h	0.010	0.016	0.25	0.41
L	0.016	0.035	0.41	0.89
Θ	0°	8°	0°	8°

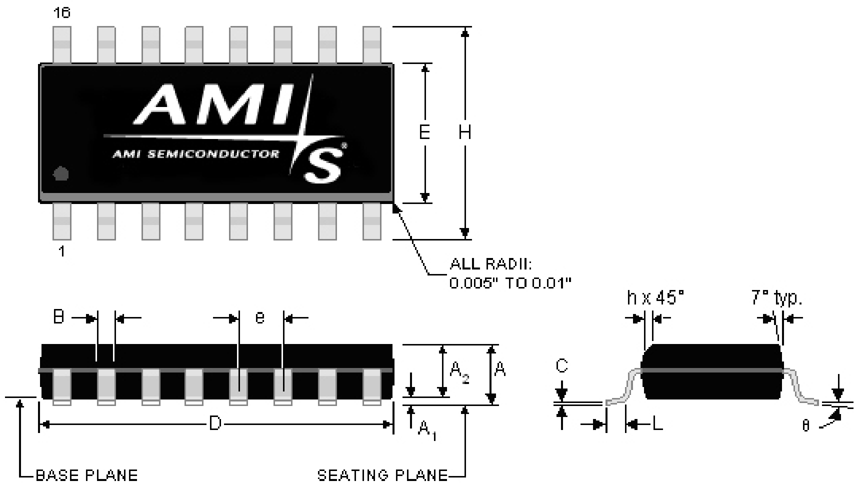


Table 16: 16-pin SOIC (0.150") Package Characteristics

Parameter	Symbol	Conditions/Description	Typ.	Units
Thermal Impedance, Junction to Free-Air 16-pin 0.150" SOIC	Θ_{JA}	Air flow = 0 m/s	109	°C/W
Lead Inductance, Self	L_{11}	Corner lead	4.0	nH
		Center lead	3.0	
Lead Inductance, Mutual	L_{12}	Any lead to any adjacent lead	0.4	nH
Lead Capacitance, Bulk	C_{11}	Any lead to V_{SS}	0.5	pF

12.0 Ordering Information

Table 17: Device Ordering Codes

Ordering Code	Device Number	Package Type	Operating Temperature Range	Shipping Configuration
11575-801-XTP (or -XTD)	FS6370-01	16-pin (0.150") SOIC (small outline package)	0°C to 70°C (Commercial)	-XTP (Tape & Reel) -XTD (Tube/Tray)
11575-819-XTP (or - XTD)	FS6370-01g	16-pin (0.150") SOIC (small outline package) 'Green' or lead-free packaging	0°C to 70°C (Commercial)	-XTP (Tape & Reel) -XTD (Tube/Tray)

13.0 Demonstration Software

Windows 3.1x/95/98-based software is available from AMIS that illustrates the capabilities of the FS6370. The software can operate under Windows NT.

Contact your local sales representative for more information.

13.1 Software Requirements

- PC running MS Windows 3.1x or 95/98. Software also runs on Windows NT in a calculation mode only.
- 1.8MB available space on hard drive C.

13.2 Software Installation Instructions

At the appropriate disk drive prompt (A:\) unzip the compressed demo files to a directory of your choice. Run setup.exe to install the software.

13.3 Demo Program Operation

Launch the fs6370.exe program. Note that the parallel port can not be accessed if your machine is running Windows NT. A warning message will appear stating: "This version of the demo program cannot communicate with the FS6370 hardware when running on a Windows NT operating system. Do you want to continue anyway, using just the calculation features of this program?" Clicking OK starts the program for calculation only.

The FS6370 demonstration hardware is no longer available nor supported.

The opening screen is shown in Figure 14.

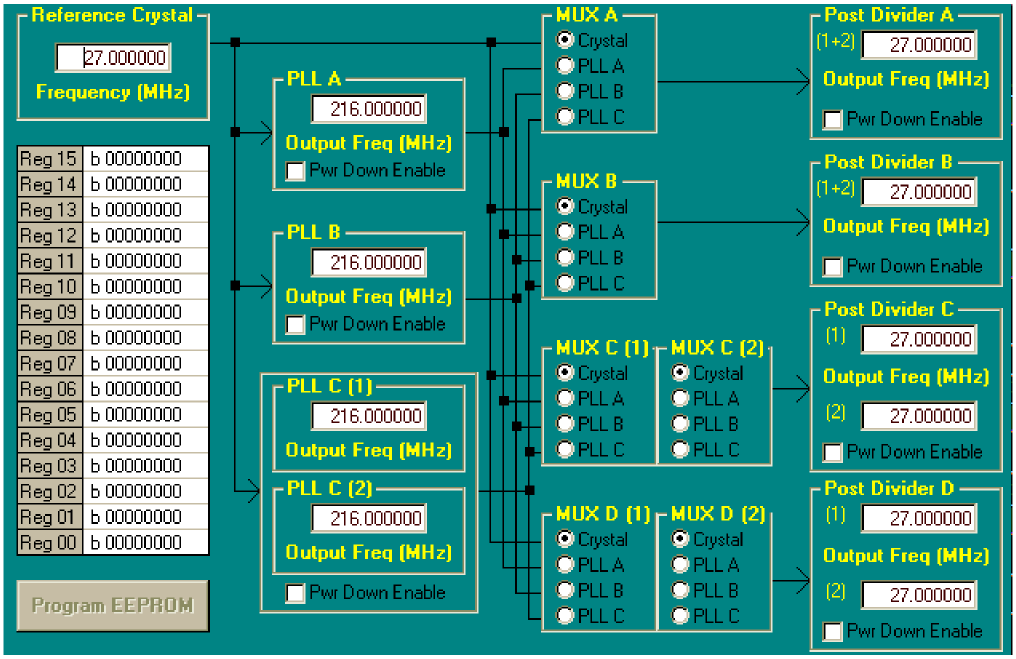


Figure 14: Opening Screen

13.3.1 Example Programming

Type a value for the crystal resonator frequency in MHz in the reference crystal box. This frequency provides the basis for all of the PLL calculations that follow.

Next, click on the PLL A box. A pop-up screen similar to Figure 15 should appear. Type in a desired output clock frequency in MHz, set the operating voltage (3.3V or 5V), and the desired maximum output frequency error. Pressing calculate solutions generates several possible divider and VCO-speed combinations.

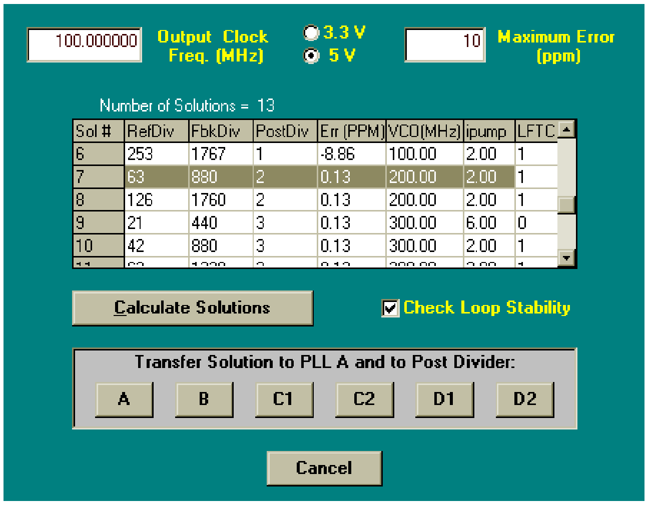


Figure 15: PLL Screen

For a 100MHz output, the VCO should ideally operate at a higher frequency, and the reference and feedback dividers should be as small as possible. In this example, highlight solution #7. Notice the VCO operates at 200MHz with a post divider of 2 to obtain an optimal 50 percent duty cycle.

Now choose which mux and post divider to use (that is, choose an output pin for the 100MHz output). Selecting A places the PostDiv value in solution #7 into post divider A and switches mux A to take the output of PLL A.

The PLL screen should disappear, and now the value in the PLL A box is the new VCO frequency chosen in solution #7. Note that mux A has been switched to PLL A and the post divider A has the chosen 100MHz output displayed.

Repeat the steps for PLL B.

PLL C supports two different output frequencies depending on the setting of the SEL_CD pin. Both mux C and mux D are also affected by the logic level on the SEL_CD pin, as are the post dividers C and D (see Section 4.2 for more detail).

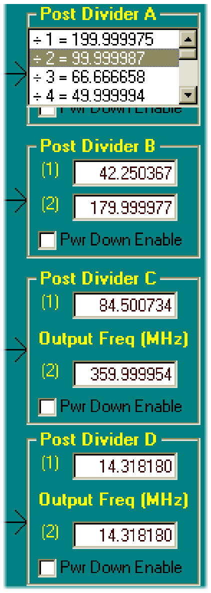


Figure 16: Post Divider Menu

Click on PLL C1 to open the PLL screen. Set a desired frequency, however, now choose the post divider B as the output divider. Notice the post divider box has split in two (as shown in Figure 16). The post divider B box now shows that the divider is dependent on the setting of the SEL_CD pin for as long as mux B is the PLL C output.

Clicking on post divider A reveals a pull-down menu provided to permit adjustment of the post divider value independently of the PLL screen. A typical menu is shown in Figure 16. The range of possible post divider values is also given in Table 7.

The EEPROM settings are shown to the left in the screen shown in Figure 14. Clicking on a register location displays a screen shown in Figure 17. Individual bits can be poked, or the entire register value can be changed.

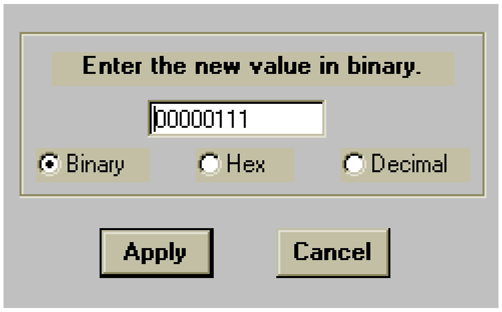


Figure 17: Register Screen