DSC21XX & DSC22XX

Low-Jitter I²C/SPI Programmable Multiple-Output Oscillators

Programming Guide for USB-910H Evaluation Board

discera

1. Device Architecture

The DSC21XX & DSC22XX families of low jitter, configurable single and dual output oscillators consist of a MEMS resonator and a high performance PLL IC, programmable via I^2C/SPI interface. This document will explain how to program the output frequencies, CMOS drive strength, and power control modes of the devices using the USB-910H Evaluation Board.

The basic block diagram of the dual and single output oscillators including the integrated PLL and clock distribution path are shown in Fig. 1. These oscillators are equipped with a high performance fractional-N PLL that locks an integrated high frequency VCO to the internal MEMS oscillator. The output of the VCO is then divided down through independent even integer dividers, with divide values ranging from 4 to 254, to generate the desired output clock frequencies. Each clock output is buffered by a low-noise output driver, available in CMOS, LVPECL, LVDS, and HCSL formats.

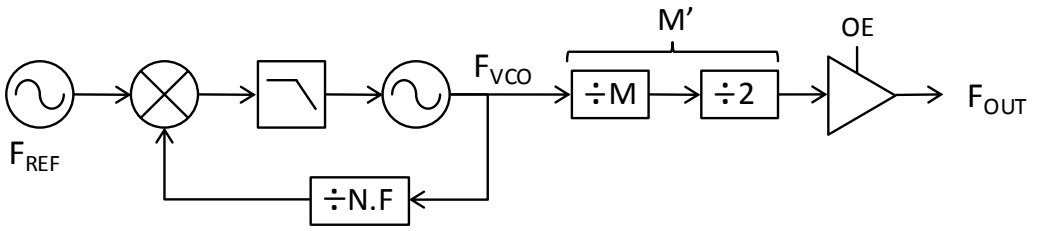
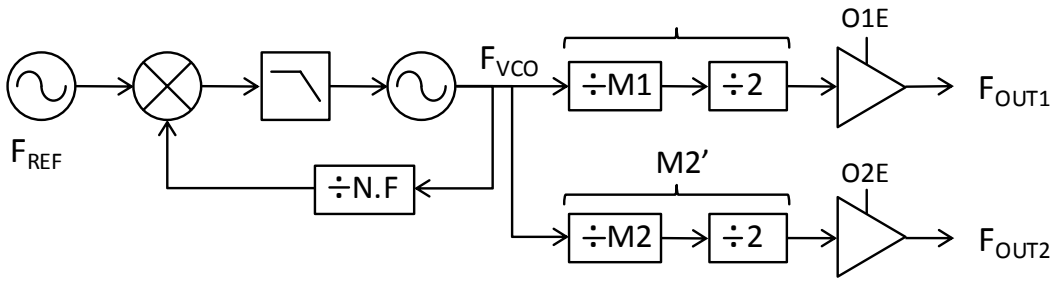


Fig. 1 a) Single output oscillator



b) Dual output oscillator

Figure 1. Block diagrams of single and dual output DSC20XX oscillators

To facilitate system level board testing or to reduce consumed power, the output drivers can be enabled or disabled, independently or as a pair, using I^2C/SPI interface. Pin 1 acts as a master enable pin and the oscillator has to be enabled (pin 1 high) before any programming can impact the output drivers. The DSC21XX and DSC22XX can, also, be programmed into and out of a low current mode (Hibernate).

2. Default Conditions

Upon power-up, the initial output frequency configuration is controlled by an internal pre-programmed memory (OTP). This memory stores all coefficients required by the PLL for two independent default frequencies for a single output oscillator, or two frequency combinations for a dual output version. The control pin (FS) selects which default frequency is the initial setting. After

power-up, a new output frequency configuration can be programmed using I^2C/SPI interface. Standard default frequencies are described in the datasheets. Discera supports customer defined versions of the DSC2XXX.

The CMOS output drive strength, which is programmable using 3 control bits, is set by default to maximum strength or the value of 111. After power-up, a new CMOS drive-strength setting can be programmed using I^2C/SPI interface.

3. Programming Clock Frequencies

This section describes the PLL variables and registers used to configure the oscillator to produce the desired output frequencies.

As shown in Fig. 1, several parameters need to be programmed to determine the internal VCO and the output clock frequencies. The PLL consists of a feedback divider with a fractional value described as N.F, where N (7-bit value) and F (17-bit value) are the integer and fractional portions of the divider value, respectively. The PLL is then followed by M divider (8-bit value) integer divider and a final divide-by-2 stage to ensure 50% output clock duty cycle. The combination of these two dividers creates an even integer M' divider block with divider value range of 4 to 254. It must be noted that M must have a minimum value of 2. To summarize, the output frequency is specified by three programmable values, N, F & M.

a. Calculation of N,F & M for Single-Output Oscillator (DSC21X0 & DSC22X0)

To optimize performance of the PLL, it is desirable to program the internal PLL to the lowest possible valid VCO frequency that can generate the desired output clock frequency. The minimum valid VCO frequency for the device is 1135MHz while the optimum VCO frequency range is from 1175 to 1700 MHz. Below is an example illustrating how to calculate the required N, F & M values.

Example:

Desired Clock Frequency: F_{OUT} = 25 MHz
Lowest Optimum VCO Frequency: $F_{VCO} = F_{OUT} \times M' = 25 \times 48 = 1200$ MHz
$M = M' \div 2 = 24$

Once the VCO frequency is determined, the N.F value can be easily calculated by dividing the VCO frequency by the input reference frequency from the MEMS oscillator, which is calibrated in the fabrication process to 18 MHz.

$N.F = F_{VCO} \div F_{REF} = 1200 \text{ MHz} \div 18 \text{ MHz} = 66.666667$
$N = 66, F = 0.666667$

Since F is described using a 17-bit register, we need to calculate it as the numerator value that will create our desired fraction with $(2^{17} - 1)$ in the denominator. This number will then need to be rounded to the nearest integer value.

$F[16:0] = 0.666667 \times (2^{17} - 1) = 87380.71 \rightarrow F[16:0] = 87381$ (rounded)

The final step before programming the values into the appropriate registers is to convert M, N, and F to the appropriate binary or hexadecimal values.

$M\ [7:0] = 24_{dec} = 0001\ 1000_{\ bin} = 18_{hex}$
$N\ [6:0] = 66_{dec} = 1000\ 0101_{\ bin} = 85_{hex}$
$F\ [16:0] = 87381_{dec} = 1\ 0101\ 0101\ 0101\ 0101_{bin} = 15555_{hex}$

We now load the resulting values into the appropriate registers to program the DSC21X0 or DSC22X0 to the desired output clock frequency. Table 1 describes the register map for the variables described above.

Table 1. Register map for the single-output DSC2XXX oscillator

Registers	7	6	5	4	3	2	1	0
0x13	F[7]	F[6]	F[5]	F[4]	F[3]	F[2]	F[1]	F[0]
0x14	F[15]	F[14]	F[13]	F[12]	F[11]	F[10]	F[9]	F[8]
0x15	N[6]	N[5]	N[4]	N[3]	N[2]	N[1]	N[0]	F[16]
0x16	M1[7]	M1[6]	M1[5]	M1[4]	M1[3]	M1[2]	M1[1]	M1[0]

Table 2 provides a list of commonly used output clock frequencies, corresponding PLL variables, and register parameters that need to be programmed into the oscillator via I^2C/SPI interface.

Table 2. Common clock frequencies and corresponding parameters

Freq (MHz)	M	N	F	FVCO (MHz)	Registers (address & value in HEX)			
					0x13	0x14	0x15	0x16
19.2	31	66	0.133333	1190.4000	44	44	84	1F
24	25	66	0.666667	1200.0000	55	55	85	19
25	24	66	0.666667	1200.0000	55	55	85	18
33.3333	18	66	0.666666	1200.0000	55	55	85	12
40	15	66	0.666667	1200.0000	55	55	85	15
48	13	69	0.333333	1248.0000	AA	AA	8A	0D
50	12	66	0.666667	1200.0000	55	55	85	0C
54	11	66	0	1188.0000	00	00	84	0B
62.5	10	69	0.444444	1250.0000	8E	E3	8A	0A
66.6666	9	66	0.666666	1200.0000	55	55	85	09
74.25	8	66	0	1188.0000	00	00	84	08
75	8	66	0.666667	1200.0000	55	55	85	08
77.76	8	69	0.12	1244.1600	71	3D	8A	08
100	6	66	0.666667	1200.0000	55	55	85	06
106.25	6	70	0.833333	1275.0000	AA	AA	8D	06
125	5	69	0.444444	1250.0000	8E	E3	8A	05
133.333	5	74	0.074074	1333.3333	ED	25	94	05
148.5	4	66	0	1188.0000	00	00	84	04
150	4	66	0.666667	1200.0000	55	55	85	04
155.52	4	69	0.12	1244.1600	71	3D	8A	04

156.25	4	69	0.444444	1250.0000	8E	E3	8A	04
200	3	66	0.666667	1200.0000	55	55	85	03
212.5	3	70	0.833333	1275.0000	AA	AA	8D	03
400	2	88	0.888889	1600.0000	1C	C7	B1	02

b. Calculation of N,F & M for Dual-Output Oscillator (DSC21XX & DSC22XX)

Calculation of the PLL parameters (N, F & M) for the dual output oscillator devices is very similar to that described in the previous section. However, the oscillator will now need to simultaneously generate two output clock frequencies from a single VCO frequency. This constraint alters the computation of the common VCO frequency. We need a valid VCO frequency that when divided by two independent even integer values, each ranging from 4 to 254, will generate both desired output clock frequencies. This analysis is best performed with an iterative approach using a spreadsheet. Please refer to Discera's DSC21XX Configuration Calculator spreadsheet (available on the Discera website at (www.discera.com/datasheets/DSC2XXX_Configuration_Calculator.xlsx).

Example:

Desired Clock Frequency 1: F_{OUT1} = 125 MHz
Desired Clock Frequency 2: F_{OUT2} = 25 MHz
Lowest Common VCO Frequency (from spreadsheet): F_{VCO} = 1250 MHz

Once the VCO frequency is determined, all other parameters can be easily calculated as shown in previous section.

M1′ = F_{VCO} ÷ F_{OUT1} = 1250 MHz ÷ 125 MHz = 10
M1 = M1′ ÷ 2 = 5
And
M2′ = F_{VCO} ÷ F_{OUT2} = 1250 MHz ÷ 25 MHz = 50
M2 = M2′ ÷ 2 = 25

N.F = F_{VCO} ÷ F_{REF} = 1250 MHz ÷ 18 MHz = 69.444444
N = 69, F = 0.444444

Since F is described using a 17-bit register, we need to calculate it as the numerator value that will create our desired fraction with $(2^{17} - 1)$ as the denominator. This number will then need to be rounded to the nearest integer value.

F [16:0] = 0.444444 x $(2^{17} - 1)$ = 58253.71 → F [16:0] = 58254 (rounded)

The final step before programming the values into the appropriate registers is to convert the M, N, and F to the appropriate binary or hexadecimal values.

M1 [7:0] = 5_{dec} = 0000 0101 $_{bin}$ = 05_{hex}
M2 [7:0] = 25_{dec} = 0001 1001 $_{bin}$ = 19_{hex}
N [6:0] = 69_{dec} = 1000 1010 $_{bin}$ = $8A_{hex}$
F [16:0] = 58254_{dec} = 0 1110 0011 1000 1110_{bin} = $0E38E_{hex}$

We now load the resulting values into the appropriate registers to program the DSC21XX or DSC22XX to our desired output clock frequencies. Table 3 describes the register map for the variable described above.

Table 3. Register map for the dual-output DSC21XX & DSC22XX oscillators

Registers	7	6	5	4	3	2	1	0
0x13	F[7]	F[6]	F[5]	F[4]	F[3]	F[2]	F[1]	F[0]
0x14	F[15]	F[14]	F[13]	F[12]	F[11]	F[10]	F[9]	F[8]
0x15	N[6]	N[5]	N[4]	N[3]	N[2]	N[1]	N[0]	F[16]
0x16	M1[7]	M1[6]	M1[5]	M1[4]	M1[3]	M1[2]	M1[1]	M1[0]
0x17	M2[7]	M2[6]	M2[5]	M2[4]	M2[3]	M2[2]	M2[1]	M2[0]

Table 4 provides a list of commonly used output clock frequency combinations, corresponding PLL variables, and register values that need to be programmed into the oscillator via I^2C/SPI interface. Contact factory for frequency pairs not included in the table below.

Table 4. Common clock frequency pairs and corresponding parameters

Freq1 (MHz)	Freq2 (MHz)	M1	M2	N	F	FVCO (MHz)	Registers (address & value in HEX)				
							0x13	0x14	0x15	0x16	0x17
27	24	24	27	72	0	1296.0000	00	00	90	18	1B
27	25	25	27	75	0	1350.0000	00	00	96	19	1B
33.3333	25	18	24	66	0.66666	1199.9999	54	55	85	12	18
40	25	15	24	66	0.66667	1200.0000	55	55	85	0F	18
50	25	12	24	66	0.66667	1200.0000	55	55	85	0C	18
75	25	8	24	66	0.66667	1200.0000	55	55	85	08	18
100	25	6	24	66	0.66667	1200.0000	55	55	85	06	18
106.25	25	8	34	94	0.44444	1700.0000	8E	E3	BC	08	22
125	25	5	25	69	0.44444	1250.0000	8E	E3	8A	05	19
150	25	4	24	66	0.66667	1200.0000	55	55	85	04	18
156.25	25	4	25	69	0.44444	1250.0000	8E	E3	8A	04	19
212.5	25	4	34	94	0.44444	1700.0000	8E	E3	BC	04	22
40	33.3333	20	24	88	0.88889	1600.0000	1C	C7	B1	14	18
48	40	15	18	80	0	1440.0000	00	00	A0	0F	12
100	75	6	8	66	0.66667	1200.0000	55	55	85	06	08
50	106.25	17	8	94	0.44444	1700.0000	8E	E3	BC	11	08
50	125	15	6	83	0.33333	1500.0000	AA	AA	A6	0F	06
148.5	74.25	5	10	82	0.5	1485.0000	00	00	A5	05	0A
156.25	125	4	5	69	0.44444	1250.0000	8E	E3	8A	04	05

c. Executing a Frequency WRITE for (DSC21XX & DSC22XX)

When executing a frequency WRITE to the DSC21XX & DSC22XX devices it is required to place the device in Hibernate by setting (0x10h<5>), execute the frequency WRITE to addresses x13-x17h, then bring the device out of Hibernate by clearing (0x10h<5>). Allow a 100 usec delay after applying the Hibernate signal before taking the subsequent Read / Write activity with the device.

4. Programming CMOS Output Drive Strength & Output Control modes

As describes earlier, drive strength for CMOS output drivers can be optimized via I^2C/SPI interface to help reduce EMI and improve performance. Each CMOS output driver has 8 drive strength settings (3 bits) and can also be independently enabled (logic high) or disabled (logic low) by writing into the corresponding bit, O1E for output 1 and O2E for output 2. An enabled output passes the generated output while a disabled output is in its tri-state condition (high impedance). Please note that oscillator has to be already enabled via ENABLE pin prior to any serial programming. Please refer to the datasheet of the specific DSC21XX device for the typical rise/fall times for the different output strength settings.

Table 5 describes the output control register map. Register bits [7:4] are only applicable to dual-output oscillators and can be ignored when programming a single-output oscillator.

Table 5. Register map for CMOS output drive control

Registers	7	6	5	4	3	2	1	0
0x11	O2S[2]	O2S[1]	O2S[0]	O2E	O1S[2]	O1S[1]	O1S[0]	O1E

5. Hibernate (Low Current) Mode

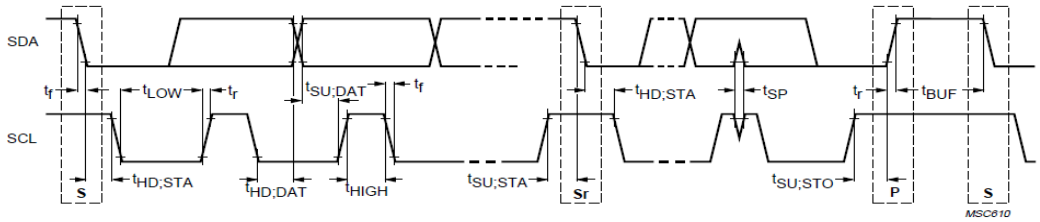
Using serial programming, the DSC21XX or DSC22XX devices can be put into a low current mode to save power. By programming the Hibernate bit (0x10<5>) to a high, the device will enter the low current mode. During Hibernate, all outputs are in the tri-state condition and the PLL and oscillator circuitry are powered down. Less than 100ua are consumed. Writing a 0 into the Hibernate bit (0x10<5>) will return the device to normal operation.

6. I^2C Bus Control Interface

The DSC21XX can be configured as a read/write slave device that conforms to Phillips I^2C bus specifications except a "general call". The DSC21XX employs a three pin I^2C bus configuration. A Chip Select (Cs_bar) enables I^2C communications with DSC21XX. The I^2C bus transmits data and clock with SDA and SCL. SDA and SCL are open-drain, that is the device can only drive these lines low or leave them to float. The bus is controlled by a master device that generates the serial clock SCL, controls bus access and generates the START and STOP conditions while the DSC21XX works as a slave. When Cs_bar is low, the accessed device will respond to SDA and SCL. When Cs_bar is high, the accessed device will not respond to I^2C signals. The I^2C slave interface follows the Philips Fast Mode (400 KHz) format.

The DSC21XX uses standard I^2C data structures and timing sequences. Because the DSC21XX employs a three pin configuration where Cs_bar controls access to the device, any 7 bit slave address can be used in the read or write commands with the exception of all bits equal 0. Figure 2 displays the I^2C timing diagram and specification.

Figure 2 I^2C Timing Diagram and Specification



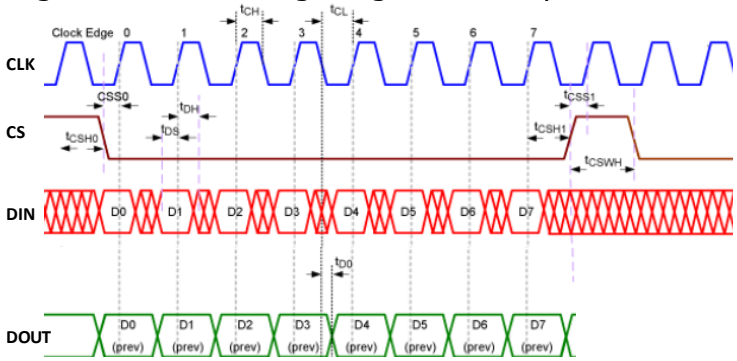
PARAMETER	SYMBOL	MIN	MAX	UNIT
SCL	t_{SCL}		3.4	MHZ
SET-UP Start	t_{SU_STA}	160		ns
Hold Time Start	t_{HD_STA}	160		ns
Low Period SCK	t_{LOW}	160		ns
High Period SCK	t_{HIGH}	160		ns
Data set-up time	t_{SU_DAT}	10		ns
Data Hold time	t_{HD_DAT}	0	70	ns
Rise Time: SCL	t_{rCL}	10	40	ns
Fall Time: SCL	t_{fCL}	10	40	ns
Rise Time: SDA	t_{rDA}	10	80	ns
Fall Time: SDA	t_{fDA}	10	80	ns
Set-UP Stop	t_{SU_STO}	160		ns

7. SPI Bus Control Interface

The DSC22XX can be used as a read/write slave device that conforms to SPI bus link protocol. The DSC22XX employs the standard four pin SPI bus configuration. The SS (Chip Select) signal (Active Low) enables communications with DSC22XX. The bus transmits full duplex synchronous data over the MOSI (Master Out Slave In) and MISO (Master In Slave Out) pins. Data is clocked by SCLK. MISO can be tri-stated (high impedance) to permit multiple devices to share the bus

The DSC22XX uses standard SPI data structures and timing sequences. Figure 3 displays the SPI timing diagram and specification.

Figure 3 SPI Timing Diagram and Specification



PARAMETER	SYMBOL	MIN	MAX	UNIT
SCLK	t_{SCL}		20	MHZ
Low Period SCLK	t_{LOW}	25		ns
High Period SCLK	t_{HIGH}	25		ns
DIN set-up time	t_{DS}	10		ns
DIN Hold time	t_{DH}	0		ns
SCLK fall to Dout valid	t_{DO}		22	ns
SCLK high to CS high Hold	t_{CSH1}	22		ns
SCLK high to CS low Hold	t_{CSH0}	0		ns
CS low to SCLK high Setup	t_{CSS0}	10		ns
CS high to SCLK high Setup	t_{CSS0}	5		ns
CS Pulse Width High	t_{CSS1}	10		ns

Figure 4 Example SPI Read Operation

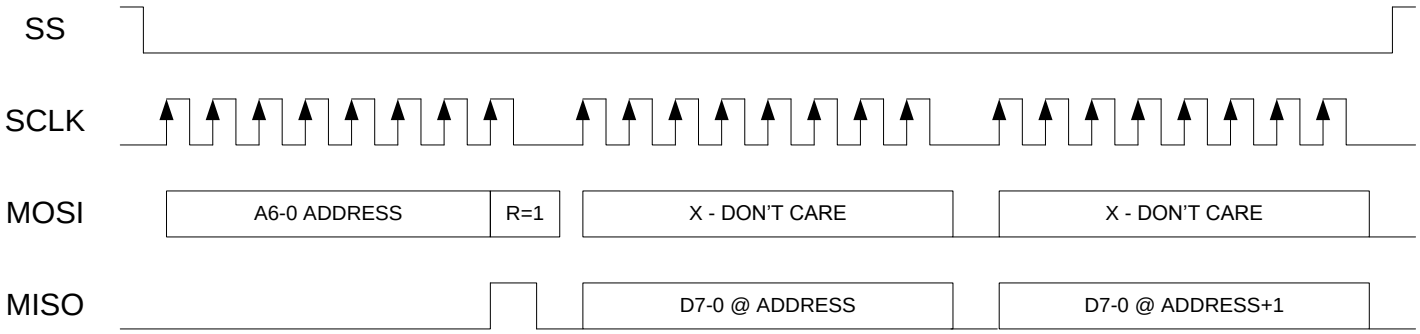


Figure 4 Example SPI Read Operation